Calibrer les comportements d'agents à partir de données réelles

Philippe Mathieu, Sébastien Picault

Calibrer les comportements d'agents à partir de données réelles

Philippe Mathieu, Sébastien Picault

Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022),
Université Lille 1, Cité Scientifique, 59655 Villeneuve d'Ascq, France
{philippe.mathieu,sebastien.picault}@univ-lille1.fr

RÉSUMÉ. *Les nouveaux domaines d'application des simulations multi-agents produisent de grandes quantités de données, qu'il convient de prendre en compte non seulement pour la validation des résultats de simulation, mais pour la calibration même des comportements des agents. En effet, la fiabilité des prédictions et des explications fournies par une simulation est fortement dépendante du réalisme statistique des agents qui y participent. Dans cet article, nous proposons une méthode pour extraire automatiquement des profils comportementaux à partir de données réelles, méthode que nous avons mise à l'épreuve dans le cadre de comportements de consommateurs dans un magasin. Les agents disposent des mêmes capacités globales d'interaction, mais sont munis de profils différenciés résultant de l'exploration des données. Placés dans un magasin virtuel réaliste, ils effectuent des achats qui reflètent la diversité des clients réels ainsi que les profils initiaux. Le processus de construction de connaissances que nous avons élaboré est assez général pour couvrir des domaines d'application variés. Aussi, nous défendons l'idée que de telles techniques sont indispensables pour renforcer le pouvoir prédictif des simulations multi-agents et en faire un puissant outil d'aide à la décision.*

ABSTRACT. *The new application domains of multiagent-based simulation provide very large databases, which must be take into account not only for validating simulation results, but also for calibrating the behaviors of the agents. Indeed, the confidence in simulation predictions and explanations highly depends on the statistical realism of the agents. In this paper, we propose a method for automatically retrieving behavioral prototypes from statistical measures. This method has been experimented within the context of consumer behavior. The agents are endowed with the same overall behavior, but are given different profiles based on the data analysis. They are put into a spatially realistic store, where their purchase reproduce the original clusters. The knowledge retrieving process is general enough to be used in various application domains. Thus, we argue that such techniques are crucial to enhance the predictive accuracy of multi-agent simulations and make them a powerful decision support tool.*

MOTS-CLÉS : *simulation multi-agents, exploration de données, marketing, interactions.*

KEYWORDS: *agent-based simulations, knowledge discovery, marketing, interactions.*

DOI:10.3166/RIA.28.

RIA – n° 4/2014, 463-484

1. Introduction

Depuis de nombreuses années déjà, les modèles centrés individus et les simulations multi-agents sont employés pour renforcer la compréhension de systèmes complexes large échelle, et ce dans des domaines variés, de la biologie moléculaire aux réseaux sociaux. Ces approches éclairent en effet les mécanismes qui font émerger des phénomènes collectifs à partir des interactions entre les entités du système, en plus de fournir des prédictions sur des variables macroscopiques. Or, les domaines abordés depuis quelques années par la simulation multi-agent sont de ceux qui produisent d'énormes quantités de données : biologie systémique (Rodin *et al.*, 2009), systèmes physiques ou chimiques (Torrel *et al.*, 2009 ; Troisi *et al.*, 2005), écosystèmes (Bousquet, Le Page, 2004), processus de diffusion (Fibich, Gibori, 2010), finance (Arthur *et al.*, 1997 ; Brandouy *et al.*, 2013), phénomènes de foule (Bonabeau, 2002 ; Narain *et al.*, 2009 ; Shao, Terzopoulos, 2007) ou encore réseaux sociaux (Roberts, Lee, 2012). Parmi les nouvelles problématiques qui en découlent, se pose de façon sensible la question de l'intégration de connaissances construites automatiquement à partir de ces données, en vue de compléter, voire remplacer une expertise humaine.

L'approche que nous proposons ici tente d'extraire autant d'information que possible de données enregistrées afin d'identifier des groupes d'agents dont les traces d'activité sont similaires. Nous construisons une description abstraite de leurs buts (sous forme de prototypes), laquelle permet de simuler une population d'agents qui reflète la diversité originelle. Nous adaptons pour ce faire diverses techniques de fouille de données au contexte particulier de l'initialisation de sous-groupes d'une population d'agents. Nous montrons que les prototypes ainsi construits, combinés à un modèle de comportement approprié et à une action située, produisent des traces d'activité similaires à celles de la population réelle.

L'article est structuré comme suit : la section 2 présente le cadre de nos travaux, notamment le contexte de l'analyse et de la simulation de comportements de clients, qui nous a servi de cadre applicatif. La section 3 décrit la manière dont nous représentons les informations pertinentes pour l'identification et la caractérisation des articles d'un magasin, des transactions effectuées (achats) et des prototypes qui peuvent en être inférés. La section 4 présente le processus d'exploration de données proprement dit, i.e. comment construire des prototypes à partir des transactions ; et la section 5, comment nous avons évalué sa robustesse et mis en œuvre notre approche au sein d'une simulation multi-agent. Enfin, nous discutons dans la section 6 des pistes permettant d'élargir le champ d'application de ces travaux.

2. Contexte scientifique

La question générale de l'identification statistique des caractéristiques d'une population d'agents à partir de données se pose avec une acuité croissante, comme en témoignent d'ailleurs plusieurs travaux récents, tels que la détection dynamique de groupes émergents (Caillou, Gil-Quijano, 2012) ou encore l'intégration dans les simulations de paramètres appris afin d'exprimer une diversité comportementale (Lacroix

et al., 2013). Dans cet article, nous nous plaçons dans la situation où les traces pertinentes de l'activité des agents sont représentables sous formes de *transactions* au sens d'Agrawal (Agrawal, Srikant, 1994). Il s'agit en l'occurrence du contexte applicatif de simulation de comportements de clients dans un magasin, où les transactions enregistrées sont des *tickets de caisse*, i.e. un ensemble d'articles ; nous discutons ultérieurement de la généralisation de notre approche à d'autres domaines. Nous commençons par exposer diverses approches utilisées pour l'analyse des paniers d'achats et la segmentation clientèle, puis nous décrivons le modèle multi-agent sur lequel nous nous appuyons pour tester notre approche.

2.1. Les techniques d'analyse et de simulation des comportements de clients

Les techniques classiques utilisées en marketing, par exemple pour partitionner les consommateurs en sous-groupes ayant des habitudes similaires, ou pour détecter des articles achetés fréquemment ensemble, consistent à extraire des informations globales à partir de très grandes bases au moyen d'algorithmes de fouille de données dédiés (Agrawal, Srikant, 1994 ; Mladenić *et al.*, 2001).

La principale technique de recherche d'informations dans des données réelles est l'analyse d'affinité (Agrawal, Srikant, 1994), qui s'appuie sur la co-ocurrence d'articles dans les achats enregistrés. Cette méthode peut être appliquée directement à des tickets de caisse réels et permet d'inférer des règles d'association entre articles (i.e. $X \rightarrow Y$ où X, Y sont des ensembles disjoints d'articles). Ces règles sont caractérisées par un *support* (proportion des achats qui contiennent à la fois X et Y) et une *confiance* (probabilité conditionnelle d'acheter les articles de Y lorsque ceux de X sont dans le panier).

Cette approche est très efficace pour la vente additionnelle (*cross-selling*) ou la montée en gamme (*up-selling*), et dans une certaine mesure donne des indications pour le placement des produits en rayon (par exemple, mieux vaut essayer d'associer dans les linéaires des produits achetés fréquemment ensemble). La première de ses limitations est le temps de calcul, qui croît comme le cube du nombre d'articles (Cavique, 2007). Mais surtout, il est assez difficile d'utiliser les règles d'association pour diriger les achats des agents, car une règle ne fait que suggérer des produits qui sont *associés à d'autres*, sans indiquer comment amorcer le panier (Sheth-Voss, Carreras, 2010).

Une autre méthode consiste à essayer de prédire des *listes de courses* à partir des tickets. Par exemple, dans (Cumby *et al.*, 2004), un assistant personnel tente d'apprendre les habitudes d'achats individuelles afin de rappeler au consommateur ce dont il va le plus certainement avoir besoin lors de ses prochaines courses, et de lui proposer des promotions sur mesure. Le but de cette application est très éloigné du nôtre, et les méthodes de classification employées ne construisent pas de représentation symbolique de la liste de courses : elles ne font qu'une prédiction probabiliste sur des catégories générales de produits. Néanmoins, ce travail démontre la possibilité d'opérer un apprentissage inductif sur des tickets réels dans le but d'identifier une liste de courses sous-jacente.

Par exemple, les règles d'association (Agrawal, Srikant, 1994) identifient des co-occurences entre articles dans les paniers des consommateurs, suggérant ainsi au distributeur une offre promotionnelle sur les produits fréquemment associés, ou de proposer des produits similaires pertinents. Toutefois, comme ces techniques ne font qu'observer des corrélations statistiques dans le comportement des clients sans offrir d'explication causale, leur utilisation peut s'avérer assez limitée. En outre, les données collectées dans les supermarchés résultent d'un processus de décision individuel complexe, affecté par des facteurs saisonniers, géographiques, environnementaux, ainsi que par la variation démographique et socio-culturelle des consommateurs, par la politique des marques, ou encore par des événements ponctuels dans le magasin comme les promotions ou les soldes. Aussi, il est difficile d'estimer la stabilité des règles construites de cette façon, et presque impossible de prédire comment des changements dans la gestion du magasin peut les affecter.

Quant aux modèles centrés individus, ils sont utilisés de façon croissante depuis quelques années dans le domaine de la distribution pour aider la prise de décision marketing (Schwaiger, Stahmer, 2003 ; Siebers *et al.*, 2007 ; Kubera *et al.*, 2010b ; Mathieu *et al.*, 2013), dans la mesure où la modélisation fine des comportements individuels permet d'élucider les raisons de l'efficacité (ou non) d'une technique commerciale donnée. Ces modèles permettent de prendre en compte les préférences au niveau individuel, et même d'introduire une expertise psychologique (Zhang, Zhang, 2007), de façon à construire une description précise des motivations et des besoins de chaque consommateur. Ce sont ensuite les actions de ces clients simulés qui sont responsables des achats prédits. Les hypothèses concernant les facteurs qui influencent les ventes peuvent être décrites explicitement dans le modèle : elles peuvent être comprises et examinées par les experts, et validées ou invalidées au moyen d'expériences appropriées.

En contrepartie, ces modèles multi-agents, tout particulièrement lorsqu'ils mettent en œuvre des agents cognitifs, requièrent souvent une expertise qui n'est facile ni à acquérir ni à implémenter. De plus, peu de modèles de simulation de magasins prennent en compte les aspects spatiaux qui sont pourtant considérés comme essentiels dans la grande distribution, comme l'allocation des linéaires, le placement des articles, le dimensionnement des caisses, la publicité sur le point de vente, etc. Dans de précédents travaux (Kubera *et al.*, 2010b), nous avons abordé ces questions de façon à concevoir une simulation de supermarchés où les agents sont situés dans un environnement spatialement réaliste. Ce caractère situé, comme nous le montrons plus loin, est nécessaire pour transformer une simulation *ad hoc* en un véritable outil d'aide à la décision, capable de prédire comment les clients réagissent aux changements de l'organisation spatiale du magasin, aux événements commerciaux ou encore à la pression concurrentielle. D'autre part, la fidélité des profils de consommateurs doit être respectée, afin d'assurer le réalisme non pas uniquement des actions effectuées par les agents simulés, mais également de la diversité de leurs objectifs d'achats.

2.2. Un modèle orienté interactions

L'approche que nous défendons ici s'inscrit dans la continuité de nos travaux antérieurs sur ce sujet (Kubera *et al.*, 2010b ; Mathieu *et al.*, 2013), dans lesquels l'acquisition d'une expertise et la conception de divers modèles de simulation imposaient une démarche incrémentale et empirique, d'où l'usage de la méthode IODA (Kubera *et al.*, 2011) pour aider les psychologues et les experts marketing à exprimer explicitement les règles comportementales adoptées par les agents.

Pour mémoire, cette méthode « orientée interactions » considère que toute entité du modèle est représentée par un agent (Kubera *et al.*, 2010a). Chaque comportement est modélisé de façon autonome par une « interaction », qui consiste en une séquence d'actions impliquant un agent source et un agent cible, soumise à des conditions d'exécution. Une interaction est réalisable si la source et la cible satisfont les conditions. Agents et interactions peuvent être développés en bibliothèques indépendantes, puis la simulation est conçue en assignant des interactions à des couples d'agents, au sein d'une « matrice d'interaction », qui constitue un moyen très visuel d'exprimer quelles familles d'agents sont autorisées à interagir, et au moyen de quelles interactions.

Par exemple dans la matrice présentée dans le tableau 1, l'intersection entre la ligne Customer et la colonne Item contient deux interactions, *Take* et *MoveTowards*, ce qui signifie qu'un agent Client peut soit prendre un agent Article, soit s'en approcher, en fonction de la priorité (premier des deux nombres — ici, *Take* a la plus forte priorité) et de la distance entre les agents (second nombre), sous réserve que les conditions de ces interactions soient satisfaites pour chacun des agents. La matrice d'interaction est traitée par un moteur de simulation générique, qui a pour fonction principale d'évaluer les interactions réalisables avec leurs sources et cibles respectives, de façon à déterminer quelles actions doivent être effectuées par chacun des agents.

En pratique, décrire les possibilités d'actions des entités du modèle en termes d'interactions pouvant être effectuées ou subies, au lieu de les représenter par des comportements encapsulés par les agents, s'apparente (sans chercher à en donner une implémentation exacte) à la théorie des affordances (Gibson, 1979) et facilite de fait l'acquisition de l'expertise quant aux comportements de ces entités.

Tableau 1. Matrice d'interaction qui définit le comportement de tous les agents de la simulation. La colonne ∅ contient les interactions réflexives (i.e. où l'agent cible est l'agent source lui-même)

Source \ Target	∅	Customer	Item	Checkout	Queue	Door
Customer	Wander (0) GoToPlace (1, ∞)		MoveTowards (2, 10) Take (4, 1)	MoveTowards (3, 10)	StepIn (5, 2) MoveOn (6, 1) WalkOut (7, 1)	Exit (8, 1)
Item		Notify (1, 10)				
Sign		Notify (1, 10)				
Checkout	Open (10) Close (10)	Notify (1, 15) CheckOut(7, 1)			Handle (8, 1) ShutDown (9, 1)	
Door	SpawnCustomer(1)	Notify(1, 10)				

Le modèle de comportement sur lequel nous nous appuyons dans les travaux présentés ici, a été développé pour la simulation d'un magasin de détail dans le but de former des vendeurs en les confrontant à des clients simulés au sein d'un *Serious Game* immersif (projet FormatStore (Mathieu *et al.*, 2013)). Le comportement générique des clients simulés a donc été validé par des experts en marketing et, dans toute la suite, il est *considéré comme figé*. La matrice d'interaction correspondante figure dans le tableau 1.

En résumé, les clients ont un même comportement d'ensemble, mais *diffèrent dans leurs besoins*, aussi sont-ils dotés au démarrage de la simulation d'une *liste de courses* qui spécifie, de façon plus ou moins détaillée, quels articles ils sont susceptibles d'acheter dans le magasin. Ces besoins peuvent être spécifiés avec précision (par exemple « SodaCola light, pack de 6× 2L ») ou au moyen d'une description vague (telle que « eau de source ») qui peut s'appliquer à de nombreux articles du magasin. Cette liste de courses est utilisée par les conditions de l'interaction *Take* (qui peut être effectuée par un agent Customer sur un agent Item) pour rendre compte de la décision d'achat.

Au cours de la simulation, les interactions ont lieu selon la matrice d'interaction et l'état et la position des agents, en respectant un comportement cohérent pour les consommateurs. Les clients artificiels (*Customer*) cherchent dans leur environnement (le magasin) les articles correspondant aux spécifications de leur liste, durant un laps de temps fixé. Ils peuvent connaître ou non (selon l'expérience à réaliser) la position de tout ou partie des produits dans le magasin (ce qui peut leur demander un effort d'exploration plus ou moins important). Lorsqu'ils perçoivent un article correspondant à un souhait sur leur liste, ils se dirigent vers lui et le prennent. D'autres agents (panneaux, affiches, caisses, etc.) peuvent leur transmettre diverses informations utiles pour accomplir leur tâche. Enfin, lorsqu'ils ont obtenu tous les articles de leur liste de courses ou lorsqu'ils ont passé trop de temps dans le magasin, les clients se dirigent vers les caisses et leur transaction est enregistrée.

Dans FormatStore (Mathieu *et al.*, 2013), ces listes de courses étaient soit aléatoires (selon une distribution de Poisson basée sur la taille moyenne des paniers), soit définies « à la main » selon une scénarisation particulière destinée à placer l'apprenti vendeur dans une situation problématique. Dans les expériences décrites dans la section 5, ces listes ont été construites au moyen de l'algorithme d'extraction de connaissances que nous proposons. La question que nous discutons dans la suite est donc essentiellement : comment construire efficacement de telles listes de courses ?

La réponse triviale consisterait à utiliser telles quelles les transactions enregistrées pour « rejouer » en simulation les achats réels. Mais les achats réels ne sont pas la cause du comportement des clients : seulement leur effet. Ils résultent précisément de la confrontation de deux mécanismes : d'une part, des besoins plus ou moins précis chez les clients, et d'autre part, une certaine organisation spatiale du magasin, la disponibilité ou la visibilité des articles, etc. autrement dit le caractère situé des choix que doivent faire les individus dans leur environnement. Les utiliser en dépit de ce caractère *fortuit* priverait la simulation de toute capacité d'extrapolation.

Nous tentons donc de combiner *l'identification de clients similaires* (comme dans la segmentation clientèle classique) en classifiant leurs transactions, et *l'induction d'une caractérisation abstraite de ces classes*, en l'occurrence par des listes de courses types (prototypes). Nous pouvons dès lors utiliser l'association entre des classes de clients et leurs listes de courses pour générer des profils d'agents susceptibles d'acheter des articles similaires. Notre démarche peut être vue comme une boucle méthodologique présentée sur la figure 1 : 1° à partir des transactions enregistrés (tickets de caisse), le processus d'exploration de données construit automatiquement des prototypes représentatifs de divers groupes de transactions similaires (listes de courses) ; 2° ces prototypes sont utilisés en simulation pour paramétrer les comportements des agents et produire des transactions simulées, qui à leur tour peuvent elles-mêmes être segmentées par le même procédé que précédemment, pour vérifier que les prototypes issus de la simulation sont les mêmes que ceux issus des données réelles.

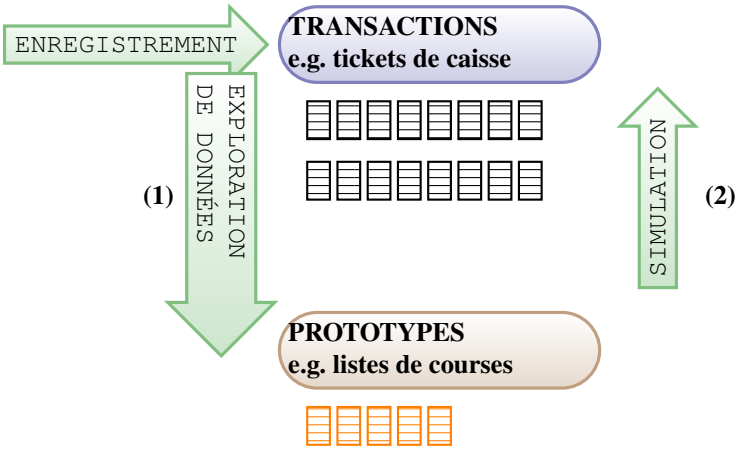


Figure 1. Une représentation de notre démarche méthodologique

Pour rentrer dans les détails de ce processus, nous devons d'abord expliquer comment nous allons décrire les articles, les tickets de caisse et les listes de courses pour leur appliquer le traitement le plus adapté.

3. Représentation des connaissances

Avant de décrire le processus d'exploration de données que nous avons élaboré pour construire les prototypes à partir des transactions, il nous faut d'abord expliquer comment ces deux sortes de données sont représentées. Cette étape peut requérir l'intervention d'un expert en marketing, mais après cela le processus de recherche d'information dans les données est automatique.

3.1. Identifiants des articles

Dans la grande distribution, chaque produit unique est identifié par une « unité de gestion des stocks » (UGS, en anglais SKU pour « stock-keeping unit ») afin de tracer sa disponibilité et la demande. Les UGS ne sont pas destinés à décrire la nature ou les caractéristiques des produits de façon standardisée, mais seulement à les référencer, comme d'autres méthodes fréquemment utilisées telles que l'UPC (*Universal Product Code*) ou l'EAN (*International Article Number*). Il arrive également que les achats réels soient anonymisés à partir d'un identifiant généré automatiquement, par exemple pour faire de l'analyse de panier sous de fortes contraintes de confidentialité.

Ces méthodes d'identification ne sont pas très adaptées pour extraire autre chose que des règles de co-occurrence. Il est en fait nécessaire, pour caractériser ces produits en vue d'une analyse explicative, de leur adjoindre des connaissances marketing pertinentes (famille de produits, qualité, prix relatif, marque, label bio, etc.). Cette étape peut requérir une expertise pour évaluer quelles sont les caractéristiques jugées pertinentes dans un contexte commercial donné.

Dans ce qui suit, nous identifions chaque produit unique par un ensemble de couples attributs-valeurs, ces attributs étant les caractères retenus comme pertinents par les experts (e.g. *{marque : "SodaCola", famille : "boisson", description :"soda goût cola"}*). Nous avons fait le choix, pour encoder les valeurs de ces caractères, de les discrétiser sous la forme d'entiers naturels non nuls (la valeur 0 servant, comme nous l'expliquons ci-dessous, à signifier l'absence de valeur spécifique). Un produit est donc représenté par un **tuple d'entiers strictement positifs**. Ainsi, supposons que les propriétés utiles soient la marque, la famille du produit et une description détaillée (e.g. respectivement « SodaCola » codé par exemple par la valeur 31, « boisson » codé par 4 et « soda goût cola » codé par 15) : alors les produits sont identifiés par un triplet d'entiers, e.g. (31, 4, 15). Cela permet une représentation de tous les produits à une granularité arbitrairement fine, incluant des qualifications très spécifiques comme « bio », « commerce équitable » ou « sans gluten ». De plus, des valeurs continues comme le prix ou le poids peuvent être encodées moyennant une discrétisation préalable (e.g. 1 pour « bon marché », 2 pour « moyen », 3 pour « cher » ; ou de 1 à 4 pour un conditionnement variant de « petit » à « familial »). La transformation des UGS ou d'autres méthodes d'identification en ce genre de tuples d'entiers peut être effectuée automatiquement, en effectuant les jointures appropriées entre bases de données.

3.2. Transactions et prototypes

Notre processus d'exploration de données s'appuie sur des informations d'achats enregistrées ; les plus accessibles sont les tickets de caisse. Une *transaction* au sens de (Agrawal, Srikant, 1994) peut être calculée à partir d'une simple énumération des produits uniques (sans doublons) présents sur les tickets, sans prendre en compte les quantités (comme (Agrawal, Srikant, 1994)). Ainsi, à partir d'une liste d'identifiants de type UGS, nous construisons un ensemble de tuples d'entiers.

À partir de ces transactions, le processus que nous appliquons consiste à extraire des *prototypes* qui visent à décrire un « ticket abstrait » caractérisant les groupes de tickets similaires. Pour ce faire, nous introduisons d'abord la notion d'*article prototype*. Un article prototype est également un tuple d'entiers, mais pour lequel **la valeur 0 est autorisée comme valeur générique (joker)**. Par exemple, un produit caractérisé par « n'importe quelle marque », « boisson », « soda goût cola » peut être décrit au moyen du triplet (0, 4, 15). Le triplet nul (0, 0, 0) signifie « n'importe quel produit ». Un *prototype* est alors simplement défini comme un ensemble d'articles prototypes.

Ces prototypes construits à partir des données peuvent également être utilisés comme « listes de courses » pour les cliens simulés, car il arrive fréquemment que seuls quelques traits des articles souhaités soient effectivement spécifiés. Ainsi, M. Dupont achète du « soda goût cola » sans égard pour la marque, tandis que M. Durand est susceptible d'acheter n'importe quel yaourt bio de la marque « Yoopla ». L'usage du 0 comme joker est fort utile pour exprimer de tels souhaits vagues.

Dans la section suivante, nous montrons comment de tels prototypes sont effectivement construits à partir des transactions.

4. Étapes du processus d'exploration de données

L'analyse des achats procède en deux étapes. Tout d'abord, la base de transactions est partitionnée en classes : il faut pour ce faire définir au préalable une mesure de distance entre tickets, elle-même basée sur une mesure de distance entre articles. Dans un second temps, pour chaque classe de transactions, tous les articles qui apparaissent sur les transactions sont à leur tour classés pour construire des articles prototypes, dont l'union constitue le prototype de la classe de transactions. La qualité de ce prototype est évaluée en mesurant sa similarité moyenne avec les transactions de la classe correspondante, grâce à la même mesure que dans la première étape.

4.1. Mesure de similarité entre articles

Comme certains articles souhaités par les clients peuvent n'être pas complètement spécifiés, il est à prévoir que des consommateurs dotés du même prototype (i.e. de la même liste de courses) *n'achètent pas exactement les mêmes produits*. Aussi, si l'on calcule une distance entre transactions uniquement en fonction des produits qu'elles ont en commun, il faut s'attendre à ce que la classification des transactions soit de piètre qualité. Nous proposons au contraire de moduler la comparaison des transactions en prenant en compte la distance entre articles.

Une façon simple de procéder est de calculer une distance « à la Hamming » (ou réciproquement un indice de similarité de type Hamming). Si deux articles (ou articles prototypes) sont représentés par les tuples d'entiers $I = (f_1, ..., f_n)$ et $I' = (f'_1, ..., f'_n)$, leur similarité est définie comme suit : $\sigma(I, I') = \frac{1}{n} \sum_{i=1}^{n} \varsigma(f_i, f'_i)$ où $\varsigma(f_i, f'_i) = 1$ si $f_i = f'_i$ ou $f_i = 0$ ou $f'_i = 0$, et $\varsigma(f_i, f'_i) = 0$ sinon.

Par exemple, prenons $I = (1, 1, 2)$ et $I' = (2, 1, 5)$: on a $\sigma(I, I') = \frac{1}{3}(\varsigma(1, 2) + \varsigma(1, 1) + \varsigma(2, 5)) = \frac{1}{3}$.

4.2. Mesure de similarité entre transactions

Afin de comparer les transactions, nous nous appuyons sur une mesure fort employée pour le calcul de similarités entre ensembles de tailles différentes : l'indice de Jaccard (Jaccard, 1901). Il est défini comme suit pour tout ensemble X, Y : $J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|}$

Comme nous l'avons expliqué ci-dessus, l'usage brut de l'indice de Jaccard ne peut satisfaire nos besoins, dans la mesure où il ne fait aucune différence entre des ensembles disjoints et des ensembles qui contiennent des éléments proches mais différents. Aussi, nous en proposons une extension, basée sur la similarité entre articles. Elle consiste à calculer le score des meilleurs appariements entre les articles de deux transactions, par similarité décroissante. Ce nouvel indice est noté dans la suite J_{BM} (pour « best-match Jaccard index »). Pour le calculer entre une transaction $\mathcal{T} = \{I_1, ..., I_p\}$ et une autre transaction $\mathcal{T}' = \{I'_1, ..., I'_q\}$, nous appliquons l'algorithme suivant :

1. Calculer la matrice d'appariement $(\sigma_{i,j})$ avec $\sigma_{i,j} = \sigma(I_i, I'_j)$
2. Pour k de 1 à min(p, q) :

   a) Calculer $\mu_k = \max_{i,j}(\sigma_{i,j})$

   b) Identifier un couple $(i^\star, j^\star)$ tel que $\sigma_{i^\star,j^\star} = \mu_k$ (si plusieurs couples (i, j) vérifient $\mu_k = \sigma_{i,j}$, sélectionner un couple $(i^\star, j^\star)$ qui minimise : $\big(\sum_{i \neq i^\star} \sigma_{i,j^\star} + \sum_{j \neq j^\star} \sigma_{i^\star,j}\big)$)

   c) Remplacer $(\sigma_{i,j})$ par la sous-matrice obtenue en supprimant la ligne $i^\star$ et la colonne $j^\star$

3. $\mu_{BM} = \sum_{k=1}^{\min(p,q)} \mu_k$ joue le même rôle que $|X \cap Y|$ dans l'indice de Jaccard classique, de sorte que nous avons :

$$J_{BM}(\mathcal{T}, \mathcal{T}') = \frac{\mu_{BM}}{p + q - \mu_{BM}}$$

Par exemple, prenons $\mathcal{T}$ ={(1, 1, 2), (3, 5, 8), (13, 21, 34)} et $\mathcal{T}'$={(1, 1, 2), (3, 6, 8), (12, 13, 14), (1, 1, 34)}. Comme on a $\mathcal{T} \cap \mathcal{T}'$ ={(1, 1, 2)} seulement, on aurait $J(\mathcal{T}, \mathcal{T}') \approx 0, 17$, tandis que l'on obtient $J_{BM}(\mathcal{T}, \mathcal{T}') = 0, 4$, car plusieurs articles de $\mathcal{T}$ et de $\mathcal{T}'$ sont assez proches comme on peut le voir sur le tableau 2 ci-après.

On notera par ailleurs que, tels que nous avons défini σ et J_{BM}, il est possible de comparer avec σ soit deux articles, soit un article et un article prototype, soit deux articles prototypes ; de même l'indice J_{BM} permet de comparer soit deux transactions, soit une transaction et un prototype, soit deux prototypes. Cette propriété est utilisée pour évaluer la qualité des prototypes qu'on cherche à construire à partir d'un ensemble de transactions (§ 4.4).

Tableau 2. Exemple de matrice de similarité entre les articles de deux transactions, utilisée pour calculer la valeur de J_{BM}. Les valeurs des μ_k sont en gras. On a ici $\mu_{BM} = 2$ d'où $J_{BM} = \frac{2}{3+4-2} = 0,4$

$\sigma(I, I')$	(1, 1, 2)	(3, 6, 8)	(12, 13, 14)	(1, 1, 34)
(1, 1, 2)	**1**	0	0	0,6666667
(3, 5, 8)	0	**0,6666667**	0	0
(13, 21, 34)	0	0	0	**0,3333333**

Il existe de nombreuses mesures de similarité et de distance qui peuvent se substituer à l'indice de Jaccard (Choi *et al.*, 2010) ; en pratique, la méthode que nous proposons est également adaptée pour les plus fréquents d'entre eux tels que les indices d'Ochiai (Ochiai, 1957) ou de Sørensen-Dice (Dice, 1945), qui peuvent être étendus suivant le même algorithme de meilleur appariement. Ce point précis a été vérifié expérimentalement au moyen de la même procédure que celle que nous présentons dans la section 5.

Quant à la complexité du calcul de J_{BM}, elle est fixée par le calcul des $p \times q$ similarités entre les articles de chaque paire $(\mathcal{T}, \mathcal{T}')$ de transactions. Elle est donc de l'ordre de $\bar{q}^2$ où $\bar{q}$ est le nombre moyen d'articles par transaction. Si l'on dispose d'une base de N transactions comptabilisant en tout A articles (soit $\bar{q} = \frac{A}{N}$), le calcul des similarités entre toutes les transactions est donc de l'ordre de $N^2 \times \bar{q}^2 = A^2$.

4.3. Classification des transactions

Nous utilisons ensuite l'indice J_{BM} pour calculer une matrice des distances entre toutes les transactions de la base de données : $\Delta_{BM} = (\mathrm{d}_{i,j})$ avec $\mathrm{d}_{i,j} = 1 - J_{BM}(\mathcal{T}_i, \mathcal{T}_j)$. Cette matrice de distances peut être soumise à de nombreuses techniques de classification, l'objectif étant d'obtenir les K classes les plus significatives. La valeur « correcte » de K n'est pas aisée à déterminer *a priori*. Nous avons proposé dans un premier temps une méthode reposant sur la détermination empirique de critères de classification, à partir d'un jeu de prototypes générés aléatoirement pour une valeur connue de K (Mathieu, Picault, 2013). Depuis nous avons pu renforcer la robustesse et le caractère automatique de cette étape en mesurant la qualité des classifications possibles grâce au critère de **silhouette** (Rousseeuw, 1987).

Ce critère permet une estimation rapide et efficace de la qualité de la classification en favorisant selon son auteur des classes « compactes et clairement séparées ». Si l'on choisit K classes notés $\mathcal{C}_i$, le score de silhouette d'une transaction au sein de cette classification est défini comme suit :

$$\forall p \in [1, K], \forall \mathcal{T}_i \in \mathcal{C}_p,\ \mathrm{silh}_K(\mathcal{T}_i) = \frac{b(\mathcal{T}_i) - a(\mathcal{T}_i)}{\max\{a(\mathcal{T}_i), b(\mathcal{T}_i)\}}$$

$$\text{où :} \quad a(\mathcal{T}_i) = \frac{1}{|\mathcal{C}_p| - 1} \sum_{\substack{\mathcal{T}_j \in \mathcal{C}_p \\ \mathcal{T}_j \neq \mathcal{T}_i}} \mathrm{d}_{i,j}, \qquad b(\mathcal{T}_i) = \min_{\mathcal{C}_q \neq \mathcal{C}_p} \frac{1}{|\mathcal{C}_q|} \sum_{\mathcal{T}_k \in \mathcal{C}_q} \mathrm{d}_{i,k}$$

Autrement dit, $a(\mathcal{T}_i)$ est la distance moyenne d'une transaction à toutes les autres transactions de la même classe, et $b(\mathcal{T}_i)$ la plus petite des distances moyennes aux transactions de chacune des autres classes.

Comme nous disposons d'un très grand nombre de transactions sans approximation préalable de la valeur de K, un procédé efficace consiste à appliquer d'abord un algorithme très classique de classification hiérarchique (Langfelder, Horvath, 2012), ce qui nous permet d'obtenir très simplement des dendrogrammes en fonction des similarités entre transactions. Ensuite, pour chaque valeur possible de K ce dendrogramme est découpé en K classes, ce qui permet de calculer le score de silhouette de chaque transaction dans le partitionnement en K classes à partir de la matrice de distances Δ_{BM}. Il ne reste plus qu'à garder la valeur de K qui maximise le score de silhouette moyen des transactions.

4.4. Induction des prototypes

Construire un prototype pour une classe de transactions consiste à trouver un ensemble $\mathcal{P}$ de tuples (autorisant le 0) qui obtient le meilleur score possible, quand on mesure sa similarité avec les N transactions de la classe au moyen de l'indice J_{BM}. Il doit donc être aussi précis que possible tout en introduisant de la généralisation lorsque c'est nécessaire.

Le processus commence par une analyse de fréquence : pour chaque article I qui apparaît sur l'une des N transactions de la classe à analyser, nous calculons $\phi(I)$ comme le rapport entre le nombre de transactions contenant I et N.

Les articles *rares*, i.e. pour lesquels $\phi(I) < \varepsilon$, peuvent être considérés comme des achats de circonstance ou du « bruit », et simplement ignorés (en pratique cela fonctionne assez bien pour $\varepsilon \approx \frac{1}{N}$, i.e. des articles qui n'apparaissent que sur un seul ticket). Inversement, les articles *fréquents*, i.e. pour lesquels $\phi(I) > \theta$, peuvent être considérés comme indispensables et sont maintenus tels quels dans le prototype à construire (empiriquement, nous utilisons $\theta \approx 0,95$).

En ce qui concerne les articles de fréquence intermédiaire, nous leur appliquons une nouvelle classification, afin de détecter certaines régularités, par exemple que les produits « SodaCola » sont toujours associés à du « yoghourt bio » de diverses marques. Pour cela nous calculons une matrice des distances entre les articles de la classe de transactions considérée : $(D_{i,j})$ avec $D_{i,j} = 1 - \sigma(I_i, I_j)$, et nous l'utilisons pour construire un dendrogramme des articles. À nouveau, le nombre pertinent de classes d'articles K_I n'est pas connu *a priori*. Pour l'estimer, nous utilisons l'algorithme suivant pour les valeurs possibles de K_I :

1. Pour chaque classe d'articles : calculer un article prototype en plaçant un 0 lorsque les caractéristiques diffèrent ; par exemple, si les articles de la classe sont (1,

5, 7), (1, 6, 7) and (1, 12, 7), l'article prototype correspondant est (1, 0, 7).

2. Faire l'union de tous les articles prototypes ainsi que des articles fréquents, ce qui donne un prototype candidat $\mathcal{P}_{K_I}$.

3. Calculer le score de $\mathcal{P}_{K_I}$ comme la moyenne de $J_{BM}(\mathcal{P}_{K_I}, \mathcal{T})$ pour toutes les transactions $\mathcal{T}$ de la classe analysée.

Nous conservons la valeur $K_I^\star$ pour laquelle le prototype associé $\mathcal{P}_{K_I^\star}$ maximise ce score. Cet algorithme est appliqué aux K classes de transactions pour générer K prototypes.

5. Validation

Les prototypes ainsi obtenus ont vocation à être utilisés par un processus de simulation pour produire des transactions artificielles. Les agents clients effectuent des comportements de façon autonome, en fonction de leurs listes de courses contenant des articles prototypes (i.e. avec des jokers), et ce dans le contexte situé d'un magasin réaliste. Dans la mesure où des articles peuvent manquer ou être difficiles à trouver, on ne peut toutefois s'attendre à ce que les transactions qui résultent de la simulation soient exactement identiques aux transactions réelles qui sont à l'origine des listes de courses.

Pourtant, nous devons nous assurer que la simulation est capable de reproduire le même *type de comportement d'achat* que celui observé chez les clients réels. Un moyen d'y parvenir est d'analyser les transactions produites par la simulation, de reconstruire les prototypes correspondants au moyen du même processus de fouille de données, et de les comparer aux prototypes issus des données réelles.

Toutefois, il est nécessaire avant d'analyser les résultats de simulations multi-agents de vérifier que la méthode de construction des prototypes est suffisamment robuste. Sans cela, d'éventuelles différences entre les prototypes qui reflètent l'activité des agents et ceux qui caractérisent les achats de consommateurs réels, pourraient trouver leur origine non pas dans le comportement des agents ou dans les particularités de l'environnement, mais seulement dans une forte sensibilité aux perturbations du processus d'exploration des données. Nous présentons donc ci-dessous comment la robustesse de notre méthode d'analyse a été évaluée.

5.1. Simulations stochastiques de l'instanciation des prototypes

Afin d'effectuer des tests assez complets, nous avons généré plusieurs ensembles de prototypes, chacun composé d'articles prototypes aléatoires. Puis, pour obtenir une simulation à gros grain, mais rapide, des achats induits par ces prototypes, nous les avons instanciés de façon aléatoire, sur la base des paramètres suivants :

– le nombre K de classes (donc de prototypes) à tester ;

– le nombre d'articles prototypes $N_I(i)$ dans chaque classe i ;

– le nombre de transactions par classe, $N_T(i)$, qui détermine combien de transactions sont instanciées pour le prototype i ;

– le nombre d'articles additionnels $N_A(i)$ qui indique combien d'articles aléatoires sont ajoutés dans chaque transaction de la classe i (cela permet de représenter des achats occasionnels qui ne sont pas représentatifs des habitudes des agents de cette classe) ;

– le nombre d'articles manquants $N_M(i)$ qui donne, pour la classe i, le nombre d'articles prototypes qui ne sont pas instanciés (possibilité que certains articles ne soient pas trouvés) ;

– le nombre N_O de transactions qui n'appartiennent à aucune classe (et sont générées de façon totalement aléatoire).

L'*instanciation* d'un article prototype consiste à remplacer chaque 0 par un entier aléatoire strictement positif, dans un certain domaine de valeurs pris par les caractéristiques des articles réels. Dans nos expériences nous avons utilisé des tuples de 5 entiers dont les valeurs maximales étaient (20, 100, 10, 5, 2) (valeurs choisies sur la base de travaux antérieurs (Mathieu *et al.*, 2013)).

Nous avons mené des expériences automatiques et leur évaluation pour les combinaisons des paramètres ci-dessus dans les intervalles suivants : K: 4 – 10 ; N_I : 5, 10, 20, 40 ; N_T: 50, 100, 200, 400, 800 ; N_A: 0, 5, 10 % du nombre d'articles dans les transactions ; N_M 0, 5, 10 % du nombre d'articles dans les transactions ; N_O: 0, 5, 10 % du nombre total de transactions.

Pour chacune de ces instantiations, nous avons mené une analyse des transactions selon le processus décrit plus haut, en utilisant notamment la bibliothèque `flashClust` de R (Langfelder, Horvath, 2012)) pour la classification hiérarchique, la fonction native `cutree` de R (Becker *et al.*, 1988) pour le découpage du dendrogramme en K classes, et la fonction `silhouette` de la bibliothèque `cluster` de R pour l'évaluation de la qualité de chaque découpage possible. Ces expériences ont donné des résultats concordants que nous résumons ci-après.

Nous avons par ailleurs mesuré le temps de calcul nécessaire pour chaque expérience, qui s'avère empiriquement proportionnel au carré du nombre total d'articles dans la base de transactions. Pour une base de $K = 5$ profils comptant chacun $N_I = 20$ articles prototypes et pour lesquels on génère $N_T = 400$ transactions, avec $N_A = 10$ % d'articles additionnels (ce qui représente 2 000 transactions en tout totalisant 44 000 articles), le traitement sur un ordinateur portable courant (MacBook-Pro i7 2.6 GHz) prend près d'une heure. Ce coût n'est pas prohibitif, d'autant que ce traitement est fait une fois pour toutes pour chaque jeu de données afin de déterminer les paramètres correspondants pour les comportements des agents.

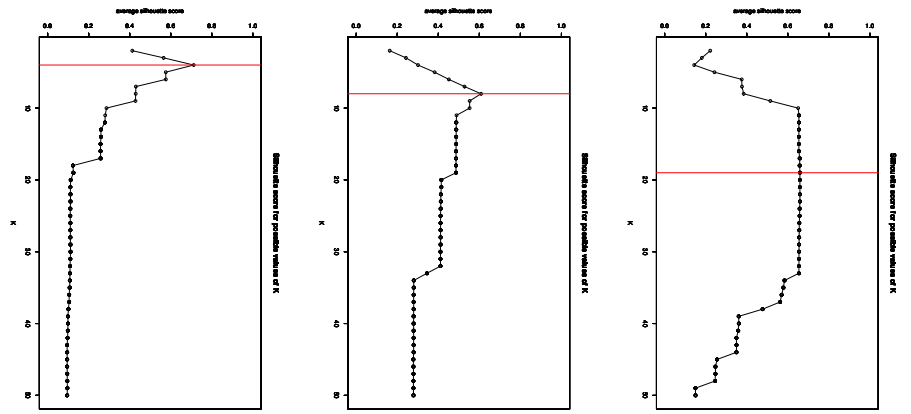
5.2. Résultats et discussion

Nous présentons sur la figure 2 la valeur du critère de silhouette en fonction du nombre K de classes possibles (la barre verticale donne l'abscisse de la valeur opti-

male de K) : comme on le voit, les situations sans transactions hors classes (a–b) ne posent pas de difficulté pour identifier la valeur d'origine de K, tandis que la présence de transactions totalement aléatoires produit un « plateau » avec quelques classes surnuméraires mais de très petits effectifs.

Sur la figure 3 sont présentés les dendrogrammes issus des similarités entre transactions pour ces trois expériences (les rectangles colorés représentent la coupe du dendrogramme pour le nombre optimal K de classes déterminé selon le critère de silhouette). On constate que les transactions produites par l'instanciation de prototypes aléatoires sont correctement discriminées, même lorsque les transactions sont construites avec des articles additionnels ou manquants.

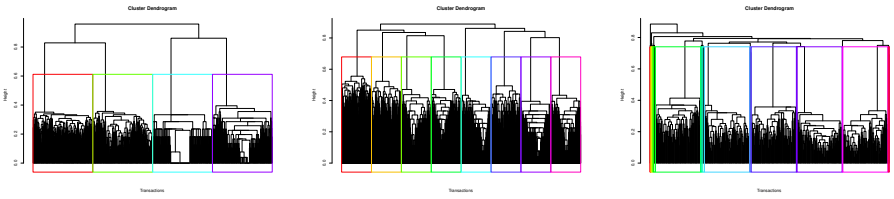
Lorsque la base contient également des transactions aléatoires (figures 2c et 3c), i.e. n'appartenant à aucune classe caractérisée par un prototype, le processus identifie toujours correctement les K classes d'origine, en ajoutant des classes supplémentaires très petites (cf. figure 4c), qui sont la plupart du temps réduites à une seule transaction.



(a) (b) (c)

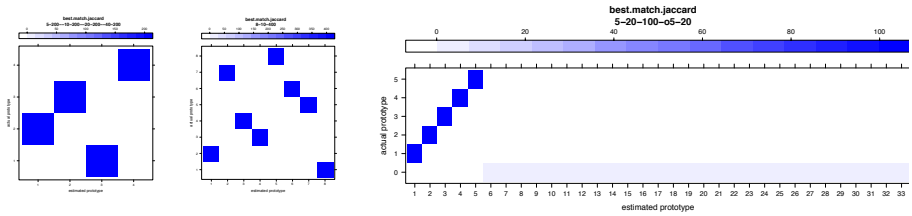
Figure 2. Valeurs du critère de silhouette (en ordonnée) en fonction du nombre K de classes possibles (en abscisse) dans 3 expériences. Paramètres :
(a) $K = 4, N_I = \{5, 10, 20, 40\}, N_T = 200, N_A = 5\,\%, N_M = 5\,\%, N_O = 0$
(b) $K = 8, N_I = 10, N_T = 400, N_A = 5\,\%, N_M = 5\,\%, N_O = 0$
(c) $K = 5, N_I = 20, N_T = 100, N_A = 5\,\%, N_M = 5\,\%, N_O = 5\,\%$

La figure 4 permet de comparer les classes de transactions estimées aux classes d'origine. L'intensité de chaque carré est proportionnelle au nombre de transactions classées dans la ligne et la colonne correspondantes. Tandis que (a) et (b) montrent une correspondance exacte entre classes estimées et classes d'origine, dans (c) on trouve en outre les transactions « bruit » générées de façon totalement aléatoire (ordonnée 0) : notre processus ne les répartit *pas* parmi les « véritables » classes, mais conduit à l'apparition de groupes supplémentaires facilement détectables par leur très faible effectif (1 ou 2 transactions). Ces classes peuvent être très facilement écartées lors de la phase de généralisation.



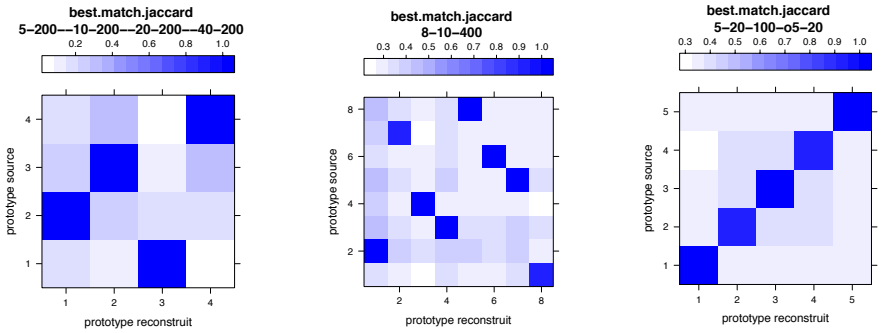
(a) (b) (c)

Figure 3. Dendrogrammes issus des similarités entre transactions dans les mêmes expériences que précédemment



(a) (b) (c)

Figure 4. Comparaison entre les classes de transactions estimées (abscisses) et les classes d'origine (ordonnées) pour les expériences (a), (b) et (c)



(a) (b) (c)

Figure 5. Comparaison entre les prototypes reconstruits (abscisses) et les prototypes d'origine (ordonnées) pour les expériences (a), (b) et (c)

La figure 5 compare les prototypes reconstruits aux prototypes d'origine. L'intensité du carré de la ligne i et de la colonne j est proportionnelle à la valeur de $J_{BM}(\mathcal{P}_i, \mathcal{P}_j^\star)$ où $\mathcal{P}_i$ désigne un des prototypes d'origine et $\mathcal{P}_j^\star$ un prototype reconstruit à partir des transactions de la classe j. Comme on peut le constater, les prototypes reconstruits sont identiques aux prototypes qui ont servi à produire les transactions de la classe correspondante. Les valeurs non-nulles ailleurs reflètent simplement le fait que des prototypes distincts ont rarement une similarité nulle deux à deux.

Dans toutes les expériences menées (jusqu'à $N_O = 10\ \%$ du nombre total de transactions), la construction de prototypes a été un succès, ce qui nous semble une bonne indication de robustesse.

5.3. Expérimentations multi-agents

Le simulateur conçu dans nos travaux antérieurs (cf. Mathieu *et al.* (2013) et § 2.2) a été modifié pour représenter les listes de courses au moyen de prototypes et les articles par des tuples d'entiers. Les comportements des agents sont régis par l'approche orientée interactions (IODA) à partir de la matrice d'interaction présentée dans le tableau 1.

Pour initialiser la population conformément aux connaissances extraites des données, les agents sont dotés à leur création d'une liste de course qui correspond à l'un des prototypes calculés, la probabilité de sortie de $\mathcal{P}_i$ étant le rapport entre le nombre de transactions de la classe $\mathcal{C}_i$ et le nombre total de transactions (selon les paramètres considérés, certains clients peuvent donc, comme dans les simulations stochastiques, être créés avec des listes de courses complètement aléatoires).

Dans un premier temps, afin de comparer les simulations multi-agents aux simulations stochastiques utilisées pour évaluer la robustesse du processus d'analyse des données, nous avons utilisé les mêmes prototypes, avec des agents connaissant la position des rayons dans le magasin. Dans ces conditions, les transactions simulées sont capables de reproduire les classes et les prototypes de départ.

Ces résultats changent légèrement lorsque l'on modifie les conditions environnementales ou certains paramètres de simulation. Ainsi, donner aux agents une *limite temporelle* pour effectuer leurs achats, lorsque ceux-ci ne connaissent pas l'agencement spatial du magasin, peut avoir pour effet que certains agents sortent du magasin sans avoir trouvé tous les articles de leur liste de courses. De la même façon, on peut faire en sorte que certains articles ne soient pas présents en rayon ou mal signalés. Introduites avec modération, ces modifications n'ont pas d'effet sur *l'identification des classes de transactions* (grâce à la robustesse du processus vis-à-vis des articles manquants).

En revanche, elles peuvent changer les prototypes reconstruits à partir des transactions simulées : dans une simulation stochastique, faire « disparaître » un article des transactions n'a statistiquement pas d'impact car chaque article a une probabilité uniforme d'être manquant. À l'inverse, lorsque les agents sont soumis au caractère

éminemment spatial de leur environnement et doivent y trouver les informations correctes pour atteindre leurs buts, il peut apparaître un biais statistique en défaveur des articles les moins faciles à trouver.

On peut observer les chemins suivis par les clients dans le magasin (cf. figure 6) : les couleurs représentent la fréquentation relative de chaque zone de l'environnement, du bleu, minimal au rouge, maximal (les zones restées en blanc n'ayant reçu aucune visite). Cela met en évidence l'existence de « points chauds » très fréquentés, où les articles sont donc trouvés facilement, ainsi que de « points froids » où la fréquentation est faible : les produits manquants sont donc souvent les mêmes, de sorte que les prototypes reconstruits dans ces conditions constituent un sous-ensemble des prototypes d'origine.

Ce phénomène met en lumière le côté crucial de la mise en situation spatiale des simulations, et illustre bien comment utiliser ces outils pour l'aide à la prise de décision quant au placement des articles en rayon, l'agencement du magasin, la signalisation, les événements commerciaux, etc.

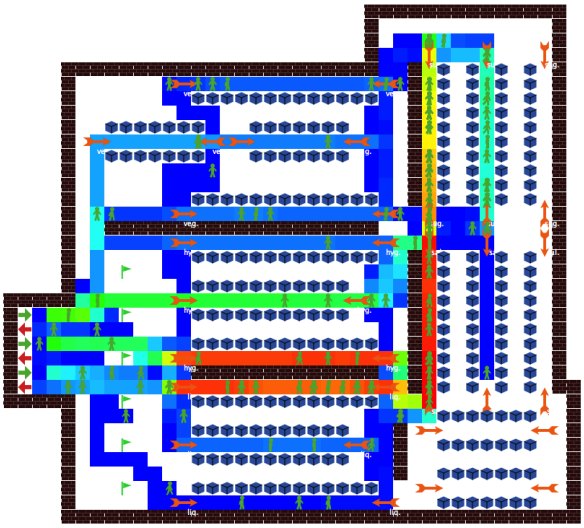


Figure 6. Tracé des chemins suivis par les clients simulés dans le magasin virtuel

Les expériences en cours visent à étudier de façon systématique l'impact de ces informations spatiales sur les changements qui surviennent dans les prototypes. Par ailleurs nous travaillons à l'intégration de bases de données de tickets réels et d'environnements de plus grande taille reproduisant les caractéristiques du magasin où ces tickets ont été collectés. Une grosse part des efforts porte sur la récupération des informations de positionnement des articles et des dispositifs d'information.

6. Conclusion et perspectives

La conception de simulations multi-agents puisant leur information au plus près des données réelles, en réduisant l'expertise humaine lorsque c'est possible, est évidemment un objectif de longue haleine, tout particulièrement si l'on souhaite élaborer des méthodes indépendantes du domaine d'application.

À ce stade, nous avons montré comment combiner une approche de simulation incrémentale (adaptée à la représentation d'hypothèses portant sur les comportements individuels, les interactions entre agents ou la configuration de l'environnement) et des algorithmes d'exploration de données (employés d'ordinaire pour extraire un « comportement moyen » valable pour l'ensemble de la population). Nous sommes ainsi en mesure 1° d'identifier diverses populations d'agents sur la base des traces laissées par leurs comportements, 2° d'associer automatiquement à ces groupes une représentation abstraite de leurs buts (prototypes), et 3° d'engendrer en simulation des populations d'agents différenciées mais présentant statistiquement les mêmes caractéristiques que les populations d'origine. La similarité entre résultats de simulation et données sources peut en outre être mesurée finement, en utilisant les mêmes moyens que pour le processus de construction de connaissances. Par ailleurs, nous avons montré que ces algorithmes sont plutôt robustes au bruit dans les données.

Notre méthode a ainsi montré son efficacité dans un cadre applicatif où nous bénéficions déjà d'une assez longue expérience. On notera que, contrairement à ce que visent nombre d'études en marketing, nous ne cherchons nullement à identifier des segments de clientèle « classiques » (par exemple suivant des critères socio-économiques, démographiques ou géographiques pré-établis), mais simplement à reproduire en simulation une « photographie » d'une population : l'instant du cliché, comme la durée de pose, c'est-à-dire le moment et la durée d'enregistrement des transactions, conditionnent évidemment les caractéristiques qui seront ensuite données aux agents. Ainsi, la même méthode peut, à volonté, intégrer des variations saisonnières, géographiques, etc. selon les données recueillies.

En pratique, notre méthode vise à capter des similarités dans les traces d'activité des individus, en construire une représentation abstraite et concise, dans l'hypothèse que cette représentation puisse être génératrice de comportements. Cette démarche rend notre méthode indépendante de cadres théoriques préexistants, et permet également d'envisager sa transposition à divers cadres applicatifs. En particulier, notre démarche s'apparente à la classification multi-label (Tsoumakas, Katakis, 2007), dans la mesure où nous cherchons à identifier des groupes dont chacun est caractérisé par un prototype constitué de plusiers articles prototypes, lesquels pourraient être vus comme des étiquettes. Évidemment, dans notre cas, l'espace des étiquettes à considérer est immense et rend de telles méthodes inapplicables. Néanmoins, cela nous donne des indications sur les domaines applicatifs qui, utilisant actuellement des méthodes de type classification multi-label ou identification de paramètres, pourraient bénéficier de notre approche : c'est le cas par exemple en génomique fonctionnelle (Barutcuoglu *et al.*, 2006) ou en écologie (Beaudouin *et al.*, 2008). Si la généralité de cette approche

se confirme, elle constituera une étape importante pour le renforcement de l'intégration automatique de données dans les simulations muti-agents.

Enfin, nous nous sommes restreints à des traces d'activités de type transaction, dans lesquelles on ne tient pas compte de l'ordre dans lequel l'agent a satisfait (ou tenté de satisfaire) ses buts. Or, il arrive fréquemment dans les systèmes naturels que l'on dispose d'informations *chronologiques* : une séquence d'événements accomplie par les agents. Les traces sont alors ordonnées, voire caractérisées par une durée. Dans les systèmes artificiels, ces informations peuvent évidemment être recueillies à des fins d'analyse. Afin de généraliser l'approche que nous avons proposée, une prochaine étape consistera à mettre au point des méthodes capables de traiter des traces chronologiques. Un certain nombre de pistes sont envisageables pour cela, la plus prometteuse étant semble-t-il à chercher du côté de *l'analyse de processus* (*process analysis*) ou de workflows (Aalst *et al.*, 2003 ; Aalst, 2011). La capacité à intégrer de telles traces permettra d'accroître considérablement le pouvoir prédictif des simulations multi-agents.

Remerciements

Nous remercions Guillaume Dauster pour sa participation à ces travaux dans le cadre de son stage de Master.

Bibliographie

Aalst W. M. van der. (2011). *Process mining: Discovery, conformance and enhancement of business processes*. Springer.

Aalst W. M. van der, Dongen B. F. van, Herbst J., Maruster L., Schimm G., Weijters A. J. M. M. (2003). Workflow mining: A survey of issues and approaches. *Data and Knowledge Engineering*, vol. 47, n° 2, p. 237–267.

Agrawal R., Srikant R. (1994). Fast algorithm for mining association rules. In *Proceedings of the 20th conference on very large data bases (VLDB'94)*, p. 487–499.

Arthur W. B., Holland J. H., LeBaron B., Palmer R., Tayler P. (1997). Asset pricing under endogenous expectations in an artificial stock market. *Economic Notes*, vol. 26, p. 297–330.

Barutcuoglu Z., Schapire R. E., Troyanskaya O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, vol. 22, n° 7, p. 830–836.

Beaudouin R., Monod G., Ginot V. (2008). Selecting parameters for calibration via sensitivity analysis: An individual-based model of mosquitofish population dynamics. *Ecological Modelling*, vol. 218, p. 29–48.

Becker R. A., Chambers J. M., Wilks A. R. (1988). *The new s language*. Wadsworth & Brooks/Cole.

Bonabeau E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences (PNAS)*, vol. 99, n° 3, p. 7280–7287.

Bousquet F., Le Page C. (2004). Multi-agent simulations and ecosystem management: a review. *Ecological Modelling*, vol. 176, n° 3–4, p. 313–332.

Brandouy O., Mathieu P., Veryzhenko I. (2013). On the design of agent-based artificial stock markets. In J. Filipe, A. Fred (Eds.), *Agents and artificial intelligence*, vol. 271, p. 350-364. Springer.

Caillou P., Gil-Quijano J. (2012). Description automatique de dynamiques de groupes dans des simulations à base d'agents. In *Actes des 20èmes journées francophones sur les systèmes multi-agents (JFSMA'12)*, p. 23–32. Cépaduès.

Cavique L. (2007, November). A scalable algorithm for the market basket analysis. *Journal of Retailing and Consumer Services*, vol. 14, n° 6.

Choi S.-S., Cha S.-H., Tappert C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, vol. 8, n° 1, p. 43–48.

Cumby C., Fano A., Ghani R., Krema M. (2004). Predicting customer shopping lists from point-of-sale purchase data. In *Proceedings of the 10th international conference on knowledge discovery and data mining (KDD'04)*, p. 402–409. New York, NY, USA, ACM. Consulté sur http://doi.acm.org/10.1145/1014052.1014098

Dice L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, vol. 26, n° 3, p. 297–302.

Fibich G., Gibori R. (2010). Aggregate diffusion dynamics in agent-based models with a spatial structure. *Operations Research*, vol. 58, n° 5, p. 1450–1468.

Gibson J. J. (1979). *The ecological approach to visual perception*. Hillsdale.

Jaccard P. (1901). Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, p. 547–579.

Kubera Y., Mathieu P., Picault S. (2010a). Everything can be agent! In *Proceedings of the ninth international joint conference on autonomous agents and multi-agent systems (AAMAS'2010)*, p. 1547–1548. Consulté sur http://www.cse.yorku.ca/AAMAS2010

Kubera Y., Mathieu P., Picault S. (2010b, September). An interaction-oriented model of customer behavior for the simulation of supermarkets. In *Proceedings of IEEE/WIC/ACM international conference on intelligent agent technology (IAT'10)*, p. 407-410. IEEE Computer Society. Consulté sur http://www.yorku.ca/wiiat10

Kubera Y., Mathieu P., Picault S. (2011). IODA: an interaction-oriented approach for multi-agent based simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, p. 1–41.

Lacroix B., Mathieu P., Kemeny A. (2013, January). Formalizing the construction of populations in multi-agent simulations. *Journal of Engineering Applications of Articifial Intelligence*, vol. 26, n° 1, p. 211-226.

Langfelder P., Horvath S. (2012). Fast R functions for robust correlations and hierarchical clustering. *Journal of Statistical Software*, vol. 46, n° 11, p. 1–17.

Mathieu P., Panzoli D., Picault S. (2013). Virtual customers in a multiagent training application. In *Transactions on edutainment IX*, vol. 7544, p. 97–114. Springer.

Mathieu P., Picault S. (2013). Des données aux agents : la simulation réaliste de populations diversifiées de clients. In S. Hassas, M. Morge (Eds.), *Actes des 21e journées francophones sur les systèmes multi-agents (JFSMA'2013)*, p. 41–50. Cépaduès.

Mladenić D., Eddy W. F., Ziolko S. (2001). Exploratory analysis of retail sales of billions of items. In A. Goodman, P. Smyth (Eds.), *Frontiers in data mining and bioinformatics. the 33rd symposium on the interface of computing science and statistics.*

Narain R., Golas A., Curtis S., Lin M. C. (2009). Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics*, vol. 28, n° 5, p. 122:1–122:8.

Ochiai A. (1957). Zoogeographic studies on the soleoid fishes found in japan and its neighbouring regions. *Bulletin of the Japanese Society for Fish Science*, vol. 22, p. 526–530.

Roberts S. C., Lee J. D. (2012). Using agent-based modeling to predict the diffusion of safe teenage driving behavior through an online social network. In *Proceedings of the human factors and ergonomics society annual meeting*, vol. 56, p. 2271–2275.

Rodin V., Querrec G., Ballet P., Bataille F.-R., Desmeulles G., Tisseau J. (2009). Multi-agents system to model cell signalling by using fuzzy cognitive maps. Application to computer simulation of multiple myeloma. In J. J. P. Tsai, P. C.-Y. Sheu, H. C. W. Hsia (Eds.), *Proceedings of the 9th IEEE international conference on bioinformatics and bioengineering (BIBE'2009)*, p. 236–241. IEEE Computer Society Press.

Rousseeuw P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, p. 53–65.

Schwaiger A., Stahmer B. (2003). Simmarket: Multiagent-based customer simulation and decision support for category management. In *Proceedings of MATES 2003: Multiagent system technologies*, vol. 2831, p. 74–84. Springer.

Shao W., Terzopoulos D. (2007). Autonomous pedestrians. *Graphical Models*, vol. 69, n° 5–6, p. 246–274.

Sheth-Voss P., Carreras I. E. (2010, Winter). How informative is your segmentation? a simple new metric yields surprising results. *Marketing Research*, p. 9–13.

Siebers P.-O., Aickelin U., Celia H., Clegg C. W. (2007, December). Using intelligent agents to understand management practices and retail productivity. In *Proceedings of the winter simulation conference (WSC'07)*, p. 2212–2220. Washington, D.C..

Torrel J.-C., Lattaud C., Heudin J.-C. (2009). Complex systems in cosmology: "The Antennae" case study. *Complex*, vol. 2, p. 1887–1897.

Troisi A., Wong V., Ratner M. A. (2005). An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Sciences (PNAS)*, vol. 102, n° 2, p. 255–260.

Tsoumakas G., Katakis I. (2007). Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, vol. 3, n° 3, p. 1–13.

Zhang T., Zhang D. (2007, August). Agent-based simulation of consumer purchase decision-making and the decoy effect. *Journal of Business Research*, vol. 60, n° 8, p. 912–922. (Complexities in Markets Special Issue)