Analyse de données symboliques et graphe de connaissances d'un agent

Philippe Caillou, Edwin Diday

Analyse de données symboliques et graphe de connaissances d'un agent

Philippe Caillou*, Edwin Diday**

*LAMSADE - Université Paris Dauphine
Place du maréchal de Lattre de Tassigny 75016 Paris
caillou@lamsade.dauphine.fr
**CEREMADE - Université Paris Dauphine
Place du maréchal de Lattre de Tassigny 75016 Paris
diday@ceremade.dauphine.fr

Résumé. Dans cet article nous appliquons l'analyse de données symboliques au graphe de connaissances d'un agent. Nous présentons une mesure de similarité entre des données symboliques adaptée à nos graphes de connaissances. Nous utilisons les pyramides symboliques pour extraire un nouvel objet symbolique. Le nouvel objet est ensuite réinséré dans le graphe où il peut être utilisé par l'agent, faisant ainsi évoluer sa sémantique. Il peut alors servir d'individu lors des analyses ultérieures, permettant de découvrir de nouveaux concepts prenant en compte l'évolution de la sémantique.

1 Introduction

Notre objectif est d'utiliser l'analyse de données symboliques (ADS) pour qu'un agent informatique enrichisse sa sémantique (et non pour qu'un utilisateur humain dispose de concepts décrivant ses données). L'analyse permettant d'extraire de nouveaux concepts pertinents à partir des données, l'agent va modifier progressivement sa sémantique pour s'adapter à son environnement. Le contexte et la représentation des connaissances sont décrits section 2, la modélisation en section 3. Nous avons défini une mesure de similarité entre graphes ainsi qu'un opérateur de fusion (voir section 4) afin de permettre l'application des différents outils de l'ADS (voir section 5). Un algorithme introduisant un objet symbolique dans le graphe est présenté section 6. Des résultats sont présentés en section 7.

2 Domaine d'application

Le modèle présenté ici a été développé pour être appliqué à un graphe de connaissances, lequel est utilisé et modifié en temps réel par un agent. Le graphe utilisé (Caillou 2003) est un graphe de concepts fluides inspiré de (Hofstadter 1995). Les liens entre les nœuds du graphe (qui sont appelés Connaissances) sont non-typés et ont une utilité fonctionnelle. Ils servent à transmettre un flux d'activation au sein du graphe. Ce flux génère les « états mentaux » de l'agent. L'intensité du lien mesure la part d'activation du nœud origine qui est transmise au nœud destination. La cible du lien sera donc activée si l'intensité est positive et inhibée si elle est négative. Une Frame (qu'elle représente une règle ou un objet) correspond à un sous-graphe centré autour d'une connaissance (appelée racine) et contenant toutes les connaissances potentiellement activées directement ou indirectement par cette racine.

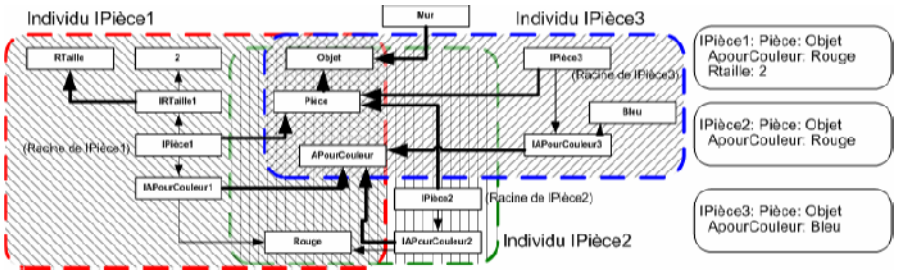


FIG. 1 - *Exemple de graphe contenant trois instances de Pièce correspondant aux individus IPièce1, IPièce2 et IPièce3 et leur représentation sous forme de Frame*

Les liens épais, correspondant à des intensités élevées, sont des liens de type implicite *Est-Un*. La lettre I au début d'un nom signifie « instance de ». Les attributs sont représentés grâce à des connaissances intermédiaires (*IAPourCouleur1*) liées à la fois à la connaissance correspondant au type de l'attribut (*APourCouleur*) et à la connaissance cible de l'attribut (*Rouge*). Notre agent est appliqué à une version simplifiée du jeu Tetris.

3 Modélisation

L'objectif de l'ADS est d'offrir un cadre d'analyse pour traiter tout type de données mais aussi pour modéliser et utiliser les concepts sous-jacents (Diday 2000). Les principales notations sont représentées figure 4 (à gauche). Notre objectif est d'analyser les connaissances de l'agent à partir de son graphe pour extraire de nouveaux concepts pertinents par rapport à l'environnement. *C* désigne l'ensemble des connaissances (nœuds) du graphe. *L* désigne l'ensemble des liens du graphe. Un lien *l* est orienté et possède une connaissance origine (*origine(l)*), une connaissance cible (*cible(l)*) et une intensité comprise entre -1 et 1 (*intensité(l)*). Ω désigne l'ensemble des sous-graphes. Pour illustrer notre modèle, nous allons considérer le graphe représenté figure 1. Ce graphe contient 15 connaissances, donc 15 individus analysables. Nous choisissons ici d'analyser les individus qui sont des instances de *Pièce*: *IPièce1*, *IPièce2* et *IPièce3*.

Les liens décrivant un individu sont ceux qui sont atteignables depuis la racine de l'individu. Lorsque l'origine est la racine de l'individu, ils sont décrit comme partant d'une connaissance générique *Racine*. Par exemple, la description de l'individu *IPièce2* est donnée figure 2. L'intensité des liens variant entre -1 et 1, on peut représenter cette variable sous une forme d'histogramme.

$$y_i(w) = \left\{ \lambda \in L \,/\, \begin{array}{l} \text{soit} \exists l_1 \ldots l_k \in L^k \; k \in \mathbb{N}^* \; avec \; origine(l_1) = racine(w), cible(l_k) = origine(\lambda), \forall t \in [\![1; k-1]\!], cible(l_t) = origine(l_{t+1}) \\ \text{soit} \, origine(\lambda) = racine(w) \end{array} \right\}$$

Les objets symboliques (OS) permettent de modéliser les concepts du monde, qu'ils soient issus directement d'un individu, d'une classe d'individus ou d'un outil d'analyse. L'ensemble des objets symboliques est noté *S*. Un OS est défini par : Une description $d \in D$; Une relation binaire *R* sur *D* permettant de comparer *d* à une autre description de *D* ; Une fonction *a* permettant d'évaluer le résultat de la comparaison (à l'aide de *R*) de la description d'un individu de Ω par rapport à la description donnée *d*.

Dans notre application, un concept peut être décrit au travers de liens minimaux qu'un individu doit posséder pour appartenir au concept. La description de l'OS est un « squelette » minimum de sous-graphe. Le concept d' « instance de Pièce » peut se décrire :

$d_{InstPiece} = \{\{ Racine, Pièce; 0, 9 \}\}$. Tout individu qui active suffisamment la connaissance *Pièce*, directement ou indirectement, à partir de sa connaissance racine est une instance de *Pièce*.

La relation *R* compare la description *d* de l'OS et une autre description *d'*. Intuitivement, on aura *d'Rd* si tous les liens de *d* existent dans *d'* avec une intensité supérieure et éventuellement des liens intermédiaires. La fonction caractéristique de *R*, h_R se définit ainsi :

$$h_R(d';d) = \begin{cases} 1 \text{ si } \forall lien \in d, \exists (l'_1 ... l'_k) \in d'^k, k \in \mathbb{N}^* \text{ avec } origine(l'_1) = origine(lien), cible(l'_k) = cible(l), \\ \quad \forall t \in [\![1; k-1]\!], cible(l'_t) = origine(l'_{t+1}) \text{ et } \min_{j=1..k}(intensité(l'_j)) \geq intensité(l) \\ 0 \text{ sinon} \end{cases}$$

a est une fonction binaire qui vérifie si d'Rd. Si les liens d'un individu sont au moins aussi fort que ceux de l'OS, *a(w)=vrai* ; *a(w)=faux* sinon. Ainsi, $a_{InstPièce}(IPièce1) = vrai$. La définition de l'objet symbolique InstPièce peut s'écrire $a_{InstPiece} = [\, y_1 \supseteq \{ (Racine, Piece; 0, 9) \}\,]$

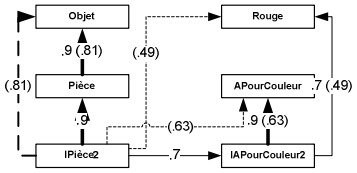
4 Mesure de similarité et opérateur de fusion

Pour plus d'efficacité, la similarité choisie nécessite un prétraitement des données:

Modification des intensités des liens : L'intensité modifiée décrite correspond à l'activation maximale qui peut passer par ce lien en provenant de la racine. La valeur est donc le produit des intensités des liens réels depuis la racine.

Ajout de liens virtuels : Les liens ayant une fonction de transmission d'activation, nous complétons la description avec des liens virtuels entre les connaissances liées par une chaîne de liens. L'intensité du lien ajouté est égale au produit des intensités des liens intermédiaires.

La nouvelle description de l'individu *IPièce2* est donnée figure 2.



$$d_{Piece1} = \begin{Bmatrix} \{ Racine, Pièce; 0, 9 \} ; \\ \{ Pièce, Objet; 0, 9 \} ; \\ \{ Racine, IAPourCouleur2; 0, 7 \} ; \\ \{ IAPourCouleur2, APourCouleur; 0, 9 \} ; \\ \{ IAPourCouleur2, Rouge; 0, 7 \} \end{Bmatrix} \quad d_{Piece2} = \begin{Bmatrix} \{ Racine, Pièce; 0, 9 \} ; \\ \{ Pièce, Objet; 0, 8 \} ; \\ \{ Racine, Objet; 0, 8 \} ; \\ \{ Racine, IAPourCouleur2; 0, 7 \} ; \\ \{ IAPourCouleur2, APourCouleur; 0, 63 \} ; \\ \{ IAPourCouleur2, Rouge; 0, 49 \} ; \\ \{ Racine, APourCouleur; 0, 63 \} ; \\ \{ Racine, Rouge; 0, 49 \} \end{Bmatrix}$$

FIG. 2 - *Individu IPièce2 et sa modélisation avant et après traitement*

Le principe de la similarité est de mesurer la somme des minima d'activation vers chaque connaissance commune depuis la racine. Une fois les prétraitements utilisés, elle peut s'écrire simplement en utilisant une fonction *lien* : *lien(c1,c2,d)* correspond à l'intensité modifiée du lien de c_1 vers c_2 pour la description *d*.

$$Sim(d_1; d_2) = \sum_{c_2 \in C} Min\{ lien(Racine, c_2, d_1); lien(Racine, c_2, d_2) \} - \sum_{c_2 \in C} \underset{c_1 \in C - \{Racine\}}{Max} \left(Min\{ lien(c_1, c_2, d_1); lien(c_1, c_2, d_2) \} \right)$$

Somme des intensités communes des liens depuis la racine (Base) : Du fait de la nature fonctionnelle des liens, l'élément important est la proportion des connaissances communes qui sera activée dans le cas d'une activation de la connaissance centrale (Racine) qui définit l'OS. Nous comparons non pas tous les liens entre les connaissances, mais uniquement les liens virtuels entre la racine et les autres connaissances communes.

Retrait de l'activation redondante (Redondance) : La mesure de base précédente entraîne l'apparition de redondance au delà des connaissances communes. Par exemple, entre

IPièce2 et *IPièce3*, nous volons mesurer ce qui est commun aux individus (ce sont des *Pièce*) et non les conséquences de ces éléments communs (ce sont des *Objet*). La suppression de ces éléments redondants favorise la découverte de nouveaux concepts caractérisés par des attributs qui n'ont pas encore été regroupés (voir les résultats obtenus section 7)

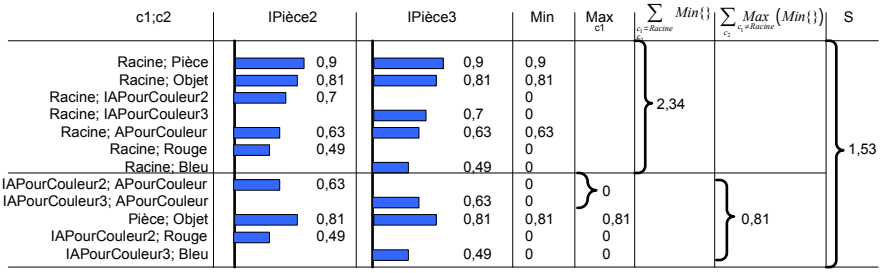


FIG. 3 - *Calcul de la similarité entre les individus IPièce2 et IPièce3*

Fusion : L'opérateur de fusion nous permet d'obtenir un nouvel objet symbolique correspondant à l'ensemble des Objets Symboliques fusionnés. Pour réaliser la fusion, nous prenons le minimum des intensités des liens communs (réels ou virtuels) aux objets fusionnés (Λ désigne l'ensemble des liens possibles). G={d1,… ;dn}

$$T(G) = \left\{ l \in \Lambda \middle/ \begin{array}{l} \forall c_1, c_2 \in C \times C \ / \forall k \in [\![1;n]\!], \exists l_k \in d_k / \left[origine(l_k) = c_1 \right] \wedge \left[cible(l_k) = c_2 \right] \\ l = \left\{ c_1 ; c_2 ; \underset{k}{Min} \left(\text{intensite} \left(l_k \right) \right) \right\} \end{array} \right\}$$

5 Sélection du concept appris

Du fait de la contrainte de rapidité de notre application (un agent temps réel), nous avons choisi de réaliser une pyramide et de sélectionner comme nouveaux concepts les paliers réalisant les sauts maximaux au niveau de la fonction d'indice.

$$Omax = \arg\max_{O \in P} \left\{ \min \left\{ f\left(SupG\left(O\right)\right) - f\left(O\right) ; f\left(SupD\left(O\right)\right) - f\left(O\right) \right\} \right\}$$

Où f(O) désigne une fonction d'indice d'un palier de la pyramide, SupD(O) désigne le palier supérieur droit de O et SupG(O) désigne le palier supérieur gauche et P l'ensemble des paliers de la pyramide. Pour la fonction d'indice, nous utilisons ici la similarité décrite précédemment en la généralisant à n individus.

6 Individualisation

Afin que l'agent puisse utiliser le nouvel objet pour raisonner, il faut l'introduire dans le graphe comme un nouvel individu. Cette opération d' « individualisation », c'est-à-dire de création d'un individu à partir d'un objet symbolique, n'est habituellement pas traitée par les modèles d'analyse de données, car ceux-ci ont pour objectif d'extraire des concepts et non de les réinsérer par la suite. Il nous faut donc compléter le modèle global (cf. figure 4).

Le nouveau concept est ajouté aux connaissances de l'agent. Comme il n'y a pas de différence de représentation en fonction de l'abstraction des concepts, le nouveau concept pourra être considéré comme un individu lors des analyses suivantes. Le nouvel ensemble des individus (Ω') peut alors être vu comme l'ensemble des concepts précédents (C). Celui-

ci contient à la fois le nouveau concept (c) et tous les concepts dont l'extension correspond à un seul individu de Ω. Card(Ω')=Card(Ω)+1 (le nouveau concept). Si l'on considère le nouvel objet symbolique (s') issu de la fusion des mêmes individus que lors de l'analyse précédente, l'extension de cet objet contiendra l'individu correspondant au nouveau concept (w_s). C'est-à-dire qu'on aura : $a_{s'}(w_s)=vrai$. Le graphe ne peut être modifié directement à partir de la description de l'objet symbolique. Pour être utilisable pour le raisonnement, il faut reconstruire les nœuds intermédiaires qui ont été perdus au cours de l'analyse, car ceux-ci sont spécifiques aux différents individus. Pour cela, nous utilisons un algorithme qui va chercher à compléter le graphe progressivement tout en plaçant les liens issus de la fusion.

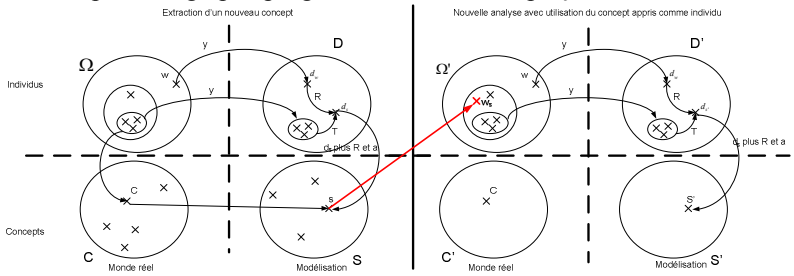


FIG. 4 - *Ajout d'un nouvel individu à partir du concept appris*

7 Résultats

Dans notre application, chaque individu de ce type d'analyse comporte entre 100 et 1200 Connaissances (nœuds du sous-graphe). On constate une première tendance, qui consiste à extraire un concept très général qui pourrait s'interpréter comme « instance du type général ». Par exemple, lorsqu'il analyse le concept de *Pièce*, il va extraire « Pièce observée » par opposition à la catégorie générale de *Pièce*. Cette distinction est utile et n'existe pas à l'origine dans le graphe, un sur-type n'étant pas une catégorie mais un ensemble de propriétés communes. Lorsque l'agent crée le concept de « Pièce observée », cela signifie qu'il rassemble toutes les propriétés communes de toutes les pièces qu'il a observé, ce qui permet à toute nouvelle pièce observée de les acquérir instantanément. On obtient par exemple en analysant les objectifs au début d'une exécution de l'agent la pyramide 1 de la figure 5. Le premier concept extrait est celui correspondant aux dix objectifs supérieurs, et qui correspond bien aux « objectifs exécutables » par opposition aux catégories d'objectifs, situés au dessous. Le deuxième concept extrait par l'agent est celui d' « objectif avec sous-objectif en attente » (les trois premiers en haut), catégorie elle aussi pertinente. L'intérêt de la suppression des redondances apparaît ici : à partir du moment où deux individus sont du même type, leur similarité augmente, mais tout ce qui est caractéristique du type est considéré comme redondant, donc retiré. Par exemple, dans la pyramide 2, bien qu'il y ait onze objectifs du même type *ana_brique*, on remarque que le palier correspondant à leur regroupement n'est même pas sélectionné dans les dix meilleurs. Des concepts plus intéressants peuvent alors apparaître, tels que ceux sélectionnés ici d'Objectif en cours et d'Objectif créé par une règle. Un deuxième avantage de la suppression de la redondance provient du fait qu'une fois qu'un concept a été créé (comme *PièceObservée*), il peut être lui-même analysé. Même s'il n'est pas analysé, il va influencer les résultats en supprimant toutes les valeurs communes aux *PièceObservée* futures.

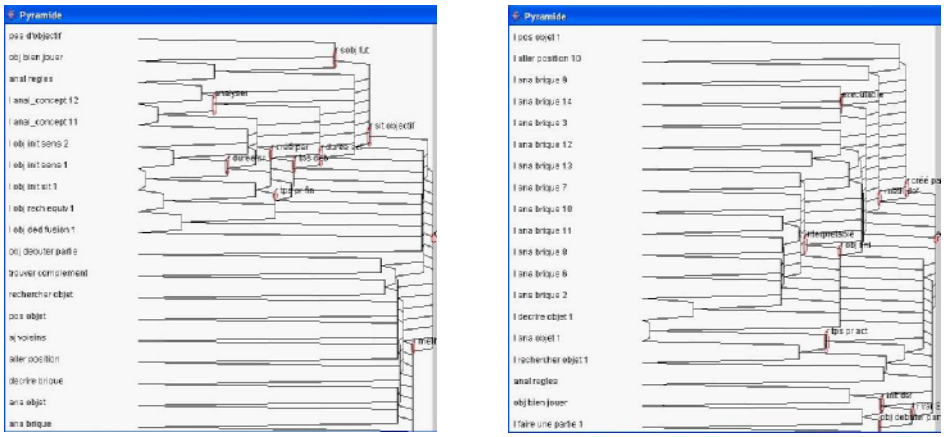


FIG. 5 - *Exemples issus de deux analyses du concept Objectif à des périodes différentes*

8 Conclusion

Dans cet article, nous avons proposé une méthode d'analyse de données symboliques permettant à la fois d'extraire des concepts pertinents du graphe de connaissances d'un agent et de les réintégrer dans le graphe pour que l'agent puisse les utiliser et les analyser à leur tour. Les résultats ont montré que les concepts extraits étaient pertinents par rapport au contexte et bien utilisés par l'agent. De nombreuses perspectives existent pour étendre le modèle. D'autres méthodes de sélection de la classe extraite sont utilisables, tout en gardant à l'esprit l'impératif de rapidité lié à l'application. L'utilisation de pyramides 3D (Diday 2004) pourrait notamment constituer une alternative intéressante. L'extension de la similarité et de l'individualisation à d'autres types de représentation sous forme de graphes, comme les graphes conceptuels (Sowa 2000), permettrait d'étendre leur domaine d'applicabilité.

Références

Caillou, P. (2003), "Raisonnement Heuristique Réflexif Dans Un Agent Temps Réel," in JNMR 03, Laval, France: INRIA, pp. 51-66.

Diday, E. (2000), "Symbolic Data Analysis and the Sodas Project: Purpose, History, Perspective," in Analysis of Symbolic Data, eds. H.-H. Bock and E. Diday, Springer.

Diday, E. (2004), "Spatial Pyramid Clustering Based on a Tessellation," in International Federation of Classification Societies (IFCS 04), Chicago: Springer Verlag, pp. 105-122.

Hofstadter, D. (1995), Fluid Concepts and Creative Analogies, New York: BasicBooks.

Sowa, J. F. (2000), Knowledge Representation, Brooks Cole.

Summary

In this paper, we use symbolic data analysis on the knowledge graph of a computer agent. The agent can thus learn new concepts that he will introduce in his graph. By doing it, he will progressively change his semantic. To achieve this process, we present a similarity measure between graphs and an operator to put the new concept in the graph.