Actes de la 5e conférence nationale sur les Applications Pratiques de l'Intelligence Artificielle

Amal El Fallah-Seghrouchni, Juliette Mattioli

AfIA

Association française
pour l'Intelligence Artificielle

APIA

Conférence Nationale
sur les
Applications Pratiques de l'Intelligence Artificielle

PFIA 2019

Table des matières

Éditorial

Ce recueil est constitué des articles acceptés à la cinquième conférence APIA 2019 (Applications Pratiques de l'Intelligence artificielle) qui se tient à Toulouse les 4 et 5 Juillet comme conférence hébergée par la plateforme AFIA 2019. Les recherches en IA menées ces dernières années ont abouti à des résultats très prometteurs et l'IA se trouve au cœur de nombreuses applications, très performantes, qui révolutionnent notre vie quotidienne et d'autres très prometteuses sont en train de le devenir. L'objectif de la conférence est de faire un tour d'horizon des applications concrètes qui couronnent de succès l'opérationnalisation de l'IA et d'échanger autour des perspectives de l'IA en termes d'applications et de recherches. Les articles sélectionnés pour cette édition traitent de la santé, l'industrie manufacturière, l'industrie financière, le e-commerce, le tourisme, la défense et sécurité, l'optimisation des processus métier, la gestion documentaire, etc. Nous remercions tous les membres du comité de programme d'avoir participé à l'évaluation des soumissions, tous les auteurs pour leurs contributions et le conférencier invité Dr. Takayuki Ito chercheur à l'Institut de Technologie de Nagoya pour sa conférence invitée.

Enfin, cette édition n'aurait pas eu lieu sans le soutien des organisateurs de la plateforme AFIA et de son président. Qu'ils en soient chareuleusement remerciés.

Amal EL FALLAH-SEGHROUCHNI, Juliette MATTIOLI

Comité de programme

Présidents

— Amal EL FALLAH-SEGHROUCHNI, Sorbonne University
— Juliette MATTIOLI, Thales

Membres

— Carole ADAM, LIG CNRS UMR 5217 - Université Grenoble-Alpes
— Florence AMARDEILH, Ticket for Change
— Ghislain ATEMEZING, Mondeca
— Jérôme AZÉ, LIRMM-UM-CNRS
— Alain BERGER, Ardans
— Moez BOUCHOUICHA, Université du Sud Toulon-Var
— Bertrand BRAUNSCHWEIG, INRIA
— Stéphan BRUNESSAUX, Airbus
— Nathalie CHAIGNAUD, LITIS - INSA Rouen Normandie
— Caroline CHOPINAUD, craft ai
— Laurent COSSERAT, Renault
— Jean-Marc DAVID, Renault
— Philippe DAVID, SNCF
— Etienne de SEVIN, SANPSY - University of Bordeaux
— Yves DEMAZEAU, CNRS - LIG
— Sylvie DESPRES, LIMICS - Université Paris 13
— Sébastien DESTERCKE, CNRS, UMR Heudiasyc
— Hamza DIDARALY, IA pour Tous
— Christophe GUETTIER - SAFRAN ELECTRONICS & DEFENSE
— Béatrice FUCHS, LIRIS, IAE - Université Lyon 3
— Céline HUDELOT, Ecole Centrale Paris
— Christophe LABREUCHE, Thales
— Arnaud LALLOUET, Huawei
— Christine LARGOUËT, Irisa /Agrocampus Ouest
— Dominique LENNE, Heudiasyc - Université de Technologie de Compiègne
— Domitile LOURDEAUX, Heudiasyc - Université de Technologie de Compiègne
— Sylvain MAHE, EDF R&D
— Philippe MATHIEU, University of Lille 1
— Nada MATTA, University of Technology of Troyes
— Christophe MENICHETTI, IBM
— Eunika MERCIER-LAURENT, CRESTIC- Université Reims Champagne Ardennes
— Christophe MEYER, Thales
— Philippe MORIGNOT, SAFRAN
— Selmin NURCAN, Université Paris 1 Panthéon - Sorbonne
— Jean-Marc OGIER, L3i - University of La Rochelle
— Jean ROHMER, De Vinci Research Center (DVRC)
— Frédérique SEGOND, Bertin
— Catherine TESSIER, Onera
— Erwan TRANVOUEZ, LSIS - Polytech'Marseille - Université d'Aix-Marseille
— Brigitte TROUSSE, Université Côte d'Azur, INRIA Sophia Antipolis - Méditerranée
— Amel YESSAD, LIP 6 - Sorbonne University

P3M : une plate-forme logicielle pour la modélisation et la manipulation des cartes d'interactions moléculaires

J.M. Alliot[1] M. Cialdea[2] R. Demolombe[1] M. Dieguez[1]
L. Farinas[1] G. Favre[3] J.C. Faye[3] O. Sordet[3]

[1] IRIT
[2] Università degli Studi Roma
[3] INSERM

jean-marc.alliot@irit.fr

Résumé

Les réseaux métaboliques, composés d'une série de voies métaboliques, consistent en un ensemble de réactions cellulaires et extracellulaires qui déterminent les propriétés biochimiques d'une cellule, régulées par des interactions complexes d'activation et d'inhibition. Ces réseaux sont le plus souvent représentés sous la forme de graphe qui décrivent les liaisons entre les différents points de contrôle du cycle cellulaire. Dans cet article, nous décrivons une plate-forme logicielle, partant de la description des réactions sous forme de graphe, de leur modélisation en logique temporelle et aboutissant à un outil permettant de vérifier la cohérence du modèle avec les résultats expérimentaux, de trouver l'ensemble de conditions initiales aboutissant à un état donné, ou de réconcilier des données expérimentales avec le graphe.

Abstract

Metabolic networks, formed by a series of metabolic pathways, are made of intracellular and extracellular reactions that determine the biochemical properties of a cell, and by a set of interactions that guide and regulate the activity of these reactions. Most of these pathways are formed by an intricate and complex network of chain reactions, and can be represented in a human readable form using graphs (often called Molecular Interaction Maps) which describe the cell cycle checkpoint pathways. In this article, we describe a software workflow, starting from the graph description of these interactions, from their modeling in temporal logic and leading to a tool that can model reactions, find the initial conditions leading to a given final state, or modify the graph in order to correctly model the observed behaviour of the cell during real experiments.

Keywords : logique temporelle, médecine, biologie

1 Introduction

Les réseaux métaboliques sont formés par un ensemble de voies métaboliques, et consistent en un ensemble

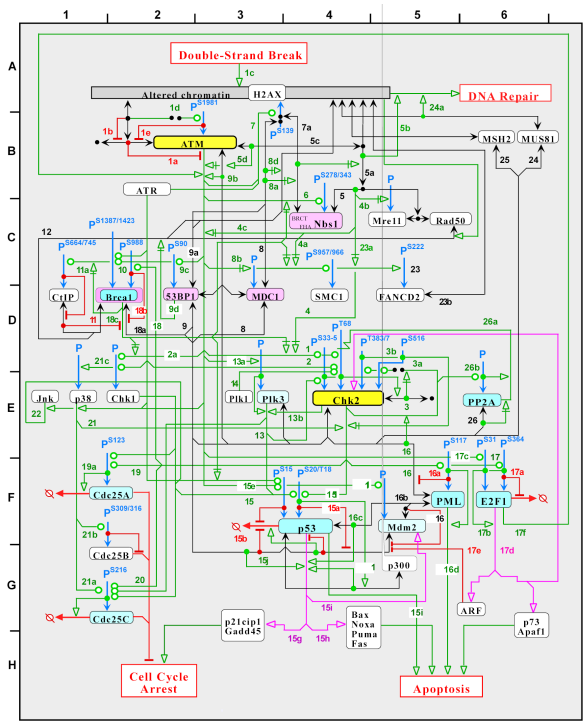


FIGURE 1 – atm-chk2/atr-chk1 molecular interaction map.

de de réactions cellulaires et extra-cellulaires qui déterminent les propriétés biochimiques d'une cellule, régulées par des interactions complexes d'activation et d'inhibition [KAWP06]. Ces réactions sont à la base du fonctionnement cellulaire et sont régulées par d'autres protéines, qui peuvent les activer ou les inhiber.

Ces réseaux sont le plus souvent représentés sous la forme de graphes qui décrivent les liaisons entre les différents points de contrôle du cycle cellulaire. Ces graphes peuvent devenir extrêmement importants et complexes (voir figure 1 [KP05]), et bien qu'essentiels à la formalisation de

la connaissance du fonctionnement de la cellule, ils sont difficiles à utiliser :

— Leur lecture est difficile en raison de leur taille.

— Ils contiennent souvent de la connaissance implicite, ce qui en modifie la compréhension suivant le niveau d'expertise de celui qui l'écrit et celui qui le lit.

— Ils peuvent contenir des incohérences, ou des manques, souvent difficiles, voire impossibles à détecter.

Dans cet article, nous présentons un logiciel permettant de formaliser, de vérifier, et de répondre à des questions complexes concernant ce type de graphe. Ce logiciel est basé sur l'utilisation de la logique temporelle linéaire pour modéliser le fonctionnement cellulaire, suivi de la transformation des formules temporelles en formules de la logique propositionnelle par réification, et enfin par l'utilisation d'un solveur SAT pour la résolution [1].

Dans la section 2, nous présentons un exemple classique (l'opéron Lac) sur lequel nous baserons la plupart de nos exemples, dans la section 3 nous montrons comment représenter une série de voies métaboliques dans notre formalisme et nous présentons l'ensemble des opérations, et les types de variable que notre modèle peut traiter, ainsi que le mécanisme général d'évolution de l'automate, la section 4 présente l'état actuel de l'implantation de l'outil la section 5 présente des exemples d'utilisation, et la section 6 présente les travaux actuellement en cours pour améliorer l'outil.

2 L'opéron lac

Nous allons tout d'abord présenter un graphe simple, qui décrit la régulation de l'opéron lactose, ou opéron *lac* [2]. Cet opéron a été le premier mécanisme de régulation génétique à être compris clairement et est maintenant un exemple "standard" de tous les cours de biologie.

L'opéron lac est un opéron utilisé par la bactérie pour permettre le transport et le métabolisme du lactose. En effet, si le glucose est la source de carbone "préférée" de la plupart des bactéries, l'opéron lac permet la digestion du lactose quand le glucose n'est pas disponible. L'opéron lac est une séquence de trois gènes (lacZ, lacY et lacA) qui encodent trois enzymes qui, à leur tour, permettent la transformation du lactose en glucose. Nous allons nous concentrer ici sur le gène lacZ qui encode la β-galactosidase, qui permet de cliver le lactose en glucose et galactose.

L'opéron lac utilise un système de contrôle en deux parties, pour s'assurer que la cellule n'utilise que l'énergie nécessaire. En l'absence de lactose, le répresseur lac arrête la production des enzymes encodés par l'opéron ; en présence de glucose, la production de l'adénosine monophosphate cyclique (CAMP) à partir de l'adénosine triphosphate (ATP) est bloquée.

1. Pour d'autres approches, on peut se référer à [CRFS04] et plus généralement à [FI14].

2. Le prix Nobel fut attribué à Monod, Jacob and Lwoff in 1965 en partie pour la découverte de l'opéron lac par Monod et Jacob [JM61]

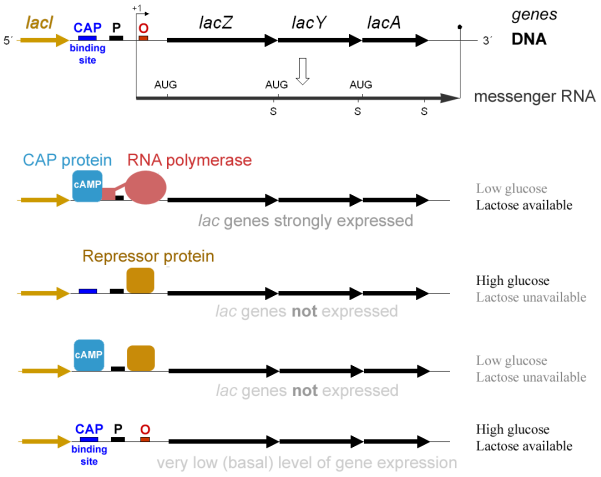


FIGURE 2 – Opéron Lac

La figure 2 décrit ce mécanisme. L'expression du gène lacZ n'est possible que si l'ARN polymérase (en rose) peut se lier au site promoteur (P, en noir) en amont du gène. Cette liaison est aidée par la molécule CAMP (en bleu) qui se lie avant le promoteur sur le site CAP (bleu foncé).

Le gène lacI (en jaune) est toujours exprimé, et produit une protéine répresseur LacI, qui se lie sur le site promoteur de l'ARN polymérase quand le lactose est disponible, empêchant l'ARN polymérase de s'y fixer, et bloquant ainsi l'expression des gènes suivants (LacZ, lacY, lacA) : il s'agit là d'une *régulation négative*, ou d'une *inhibition*, car elle bloque la production des protéines. Mais quand le lactose est présent, un de ses isomères (l'allolactose) se lie avec la protéine répresseur LacI, et le composé résultant ne peut plus se lier sur le site promoteur. L'ARN polymérase. peut alors se lier au site promoteur et permettre l'expression du gène lacZ *si* CAMP est liée à CAP.

CAMP exerce une *régulation positive*, ou activation, car sa présence est nécessaire à l'expression du gène lacZ. Mais CAMP est elle-même régulée négativement : quand du glucose est présent, CAMP n'est plus produite et ne peut pas se lier sur le site CAP, empêchant l'expression de lacZ.

3 Modélisation

Le mécanisme décrit dans la section précédente est résumé dans le graphe de la figure 3 [ADFdC16, ADD$^+$16]. Cet exemple contient toutes les relations et tous les types d'acteur que nous utilisons dans notre modèle, et nous allons les détailler maintenant, ainsi que leur mode d'évolution temporelle, et le type de problème que notre logiciel peut traiter.

3.1 Relations

Il existe deux grands types de relations : les productions et les régulations.

Productions. Les **productions** peuvent prendre deux formes différentes, suivant que les réactifs sont consom-

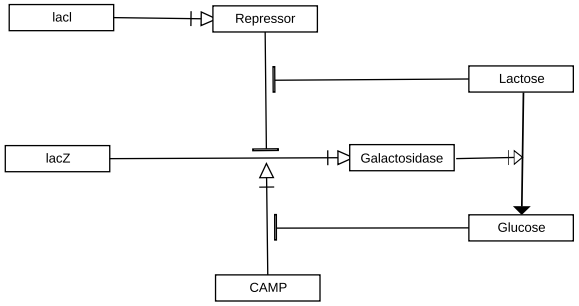


FIGURE 3 – Représentation fonctionnelle de l'opéron lac

més, ou non consommés [3] :

— Dans la figure 3, le lactose produit du glucose, et est consommé, ce que l'on note : ($lactose \rightarrow glucose$).

— A l'opposé, l'expression du gène lacZ pour produire la galactosidase (ou du gène lacI pour produire la protéine LacI) ne consomme pas le gène, et nous le notons : ($lacZ \rightarrow galactosidase$).

Définition :

— Si une réaction consomme ses réactifs, nous écrivons : $a_1, a_2 ... a_n \rightarrow b$. Ici la production de b consomme a_1,...,a_n

— Si les réactifs ne sont pas consommés, nous écrivons : $a_1, a_2, ... a_n \rightarrow b$. Ici b est produit mais a_1, a_2 ... a_n sont toujours présents après la production de b.

Régulations. Les **régulations** peuvent également prendre deux formes différentes : une réaction peut-être soit *inhibé* soit *activée* par d'autres protéines, ou d'autres conditions.

— Dans l'exemple ci-dessus, la production de la galactosidase à partir de l'expression du gène lacZ est activée par CAMP (nous écrivons $CAMP \rightarrow$ pour exprimer l'activation)

— D'autre part la même production de galactosidase est bloquée (ou inhibée) par la protéine répresseur LacI (ce que l'on note $Repressor \dashv$).

Définition :

— nous écrivons $a_1, a_2, ... a_n \rightarrow$ si la présence simultanée de $a_1, a_2, ... a_n$ active une production ou une autre régulation.

— nous écrivons $a_1, a_2, ... a_n \dashv$ si la présence simultanée de $a_1, a_2, ... a_n$ inhibe une production ou une autre régulation.

3. Dans les graphes Pathvisio nous employons $\dashrightarrow$ à la place de $\rightarrow$ pour des raisons techniques mais les deux opérateurs sont équivalents.

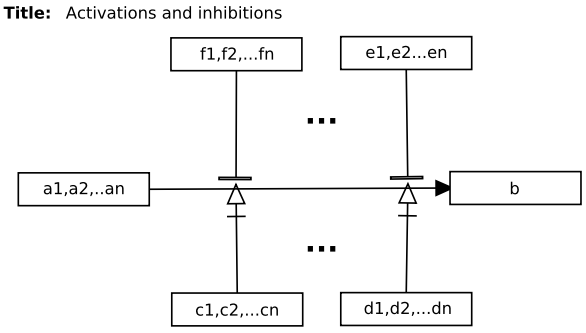


FIGURE 4 – Activations/Inhibitions

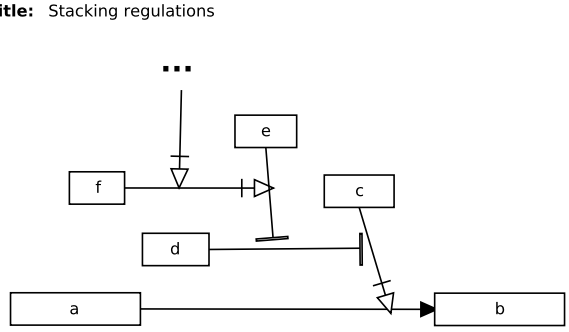


FIGURE 5 – Empilage

Forme générale d'un graphe. Sur la figure 4, nous avons un résumé de la plupart des cas possibles : la production de b à partir de $a_1, \cdots, a_n$ est activée par la présence simultanée de $c_1, \cdots, c_n$ **ou** par la présence simultanée de $d_1, \cdots, d_n$, et inhibée par la présence simultanée de $e_1, \cdots, e_n$ **ou** par la présence simultanée de $f_1, \cdots, f_n$. Les régulations sont souvent "empilées" sur plusieurs niveaux (voir figure 5). Par exemple, dans la figure 3, l'inhibition de la production de galactosidase par le répresseur LacI peut elle-même être inhibée par la présence de lactose, et l'activation de cette même production par CAMP est elle-même inhibée par la présence de glucose.

3.2 Types d'acteur

Après avoir détaillé les différents types d'interactions possibles entre les acteurs, nous allons maintenant détailler les deux différents types d'acteurs présents dans ces graphes :

Acteurs exogènes : un acteur *exogène* a sa valeur fixée par des conditions externes (température, présence ou absence d'un gène, protéine fournie en quantité suffisante par l'environnement) et sa valeur n'évolue jamais dans le temps. Ils sont fixés par les conditions expérimentales, et leur valeur n'évolue jamais dans le temps. S'ils sont consommés, l'environnement en fournira toujours en quantité suffisante. S'ils sont produits, ils sont supposés consommés instantanément par l'environnement.

Acteurs endogènes : à l'opposé les acteurs endogènes peuvent évoluer dans le temps en fonction de la dynamique du graphe. Ils peuvent apparaître s'ils sont produits, disparaître s'ils sont consommés. Leur valeur initiale est fixée par l'utilisateur.

Le statut d'un acteur est fixé par le biologiste en fonction de sa compréhension du processus biologique décrit par le graphe. En général, il existe des règles de bon sens permettant d'initialiser le type des acteurs : les acteurs exogènes n'apparaissent quasiment jamais à droite d'une règle de production (ils ne sont généralement pas produits), alors que les acteurs endogènes apparaissent eux à droite de ce même type de règles (ils sont produits).
Par exemple, dans la figure 3, les deux types d'acteurs sont présents : lacl, lacZ, CAMP et le lactose sont initialisés comme acteurs exogènes en fonction de la règle "de bon sens" énoncés ci-dessus ; toujours suivant la même règle, la galactosidase, la protéine répresseur et le glucose sont classés comme endogènes. Ces règles "de bon sens" permettant de classifier les variables en endogène ou exogène ne servent qu'à initialiser "simplement" les types de variables, mais ils doivent pouvoir être modifiés ultérieurement par le biologiste suivant les conditions expérimentales et le type de l'étude. Le lactose peut par exemple être considéré comme une variable exogène ou une variable endogène, suivant le type de dynamique que le biologiste veut étudier : soit un environnement saturé en lactose, soit un environnement où le lactose est présent à l'état initial, mais peut disparaître par consommation par la bactérie. De la même façon le glucose pourrait être considéré comme exogène si l'on veut simuler un environnement saturé en glucose.
Il faut également comprendre que les graphes décrivent les réactions que les biologistes estiment importantes pour décrire un mécanisme particulier du fonctionnement cellulaire. Ils peuvent être écrits de multiples façons, en fonction par exemple des blocs fonctionnels que l'on veut dégager, et certaines réactions ou relations ne sont simplement pas décrites parce qu'elles ne sont pas fondamentales pour l'objet de l'étude.

3.3 Évolution temporelle

Un graphe peut être considéré comme un automate qui produit une séquence d'états des acteurs. Les formules pour décrire les séquences d'état sont celles de la logique temporelle linéaire. Le temps est supposé discret, et toutes les relation de production/consommation qui peuvent s'exécuter s'exécutent simultanément en un pas de temps. Un acteur peut donc être vu comme une variable booléenne pouvant prendre les valeurs 0 (absent) ou 1 (présent). Lorsqu'un acteur est consommé, il devient absent ; lorsqu'il est produit, il devient présent. Lorsqu'un acteur (déjà présent) est à la fois consommé et produit, il reste présent par défaut. Ce fonctionnement peut sembler réducteur, puisqu'il ne prend en compte ni les quantités, ni les vitesses de réaction, mais nous allons voir qu'il permet déjà de traiter de nombreux problèmes.
Les valeurs des variables sont initialisées également à partir de règles de "bon sens". Les variables exogènes sont initialisées à *libre* : ce sont généralement les variables sur lesquelles on va "raisonner". Les variables endogènes sont initialisées à *absent*. Là aussi, cette initialisation sera modifiée par l'utilisateur à travers l'interface graphique en fonction de l'étude qu'il poursuit (conditions initiales, etc).

3.4 Types de problèmes traités

Le logiciel est capable de traiter trois grands types de "problème" sur un graphe donné :

Cohérence : le logiciel peut vérifier la cohérence du graphe avec des résultats expérimentaux. L'utilisateur fournit les conditions initiales et les conditions finales d'une expérimentation, et le système vérifie que les résultats expérimentaux sont en accord avec la modélisation.

Abduction : étant donné un ensemble de conditions finales, le logiciel peut trouver le ou les ensembles de conditions initiales qui permettent d'y aboutir. Dans ce cas l'utilisateur initialise les valeurs des variables dont il veut fixer la valeur, et laisse libres les valeurs initiales des variables sur lesquelles il veut raisonner. La complexité du système croit exponentiellement avec le nombre de variables libres : lorsque n variables sont libres, il y a 2^n chemins possibles pour l'automate. Nous verrons un exemple détaillé dans la section 5.1.

Mise à jour : lorsque les conditions finales d'une expérimentation ne sont pas en accord avec le modèle, le logiciel peut proposer des modifications du graphe permettant de mettre en cohérence les résultats observés avec le modèle. Cela peut se faire soit en fixant toutes les valeurs initiales des variables, soit en laissant certaines variables libres et en recherchant des solutions vérifiant certaines conditions sur ces variables. Dans ce cas, la mise à jour se combine avec l'abduction. Si N est le nombre total de variables, n le nombre total de variables libres, r le nombre de relations et p le nombre de relations nouvelles que nous cherchons à ajouter au graphe,la complexité du problème à résoudre croit comme $2^n(N(r+N))^p$, Nous verrons un exemple détaillé de mise à jour de graphe dans la section 5.2.

4 Implantation

Nous allons décrire ici la plate-forme logicielle que nous utilisons pour la représentation, l'interprétation et l'utilisation de nos graphes. Le mécanisme général, en quatre parties, est décrit dans la figure 6.

4.1 Saisie des graphes

Les graphes sont construits en utilisant un outil standard, *Pathvisio* [vIKP+08]. Il s'agit d'un logiciel du domaine public, bien connu dans la communauté des biologistes.

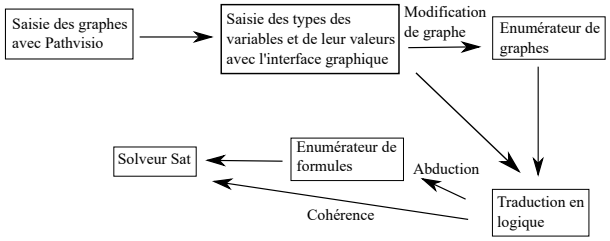


FIGURE 6 – Implantation

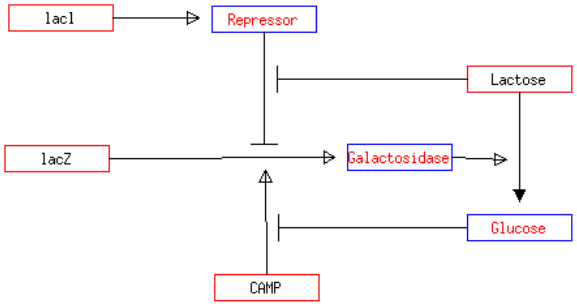


FIGURE 7 – Opéron lac après initialisation "simple" des types de variables et de leurs valeurs

Utiliser Pathvisio nous permet également de récupérer des graphes déjà écrits dans d'autres contextes et de les utiliser pour nos besoins propres sans avoir à les saisir à nouveau. Pathvisio permet de saisir des informations plus riches que celles utilisées par notre modélisation, mais elles ne sont pas utilisées pour l'instant. Les graphes sont sauvegardés en format XML par Pathvisio, et peuvent donc être relus par n'importe quel parseur XML.

4.2 Saisie des types des variables et de leurs valeurs avec l'interface graphique

Le fichier XML produit par Pathvisio est directement relu par notre logiciel. A travers une interface graphique, l'utilisateur peut modifier le type des variables (exogène / endogène) ainsi que leurs valeurs initiales. On peut voir sur la figure 7 comment le logiciel initialise les types et les valeurs des variables à partir des règles "de bon sens" décrites dans la section précédentes. Les boites entourant lacI, lacZ, CAMP et Lactose sont *rouges*, ce qui indiquent qu'elles sont *exogènes*, les boîtes entourant le glucose, la galactosidase et le répresseur sont en *bleues*, indiquant qu'il s'agit de variables *endogènes*. Les valeurs des variables sont initialisées à *absent* (*rouge*) pour les variables endogènes et à *libre* (*noir*) pour les variables exogènes.
Sur la figure 8, on voit comment l'utilisateur a modifié les valeurs des variables. lacI, lacZ et CAMP sont en *verts*, indiquant qu'ils *sont présents* au début (et le resteront, puis-

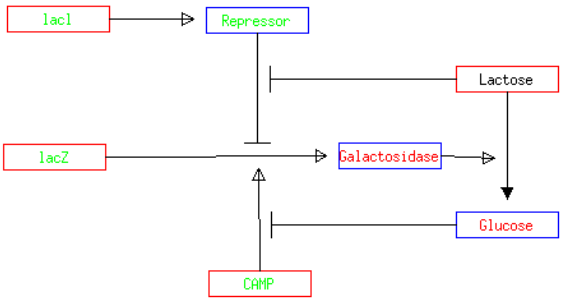


FIGURE 8 – Opéron lac après modifications

qu'il s'agit de variables exogènes). Le répresseur est *vert*, en effet la protéine répresseur est toujours présente dans la cellule, en particulier dans les conditions initiales. Le lactose reste *noir* car il s'agit de la variable *libre*, sur laquelle l'utilisateur va "poser des questions" au système.
Les questions à poser au système ainsi que les différents paramètres à saisir (nombre de pas de temps, ou nombre de modifications à effectuer sur le graphe pour les problèmes de mise à jour) doivent être posés au système à travers une interface textuelle pour l'instant.

4.3 Moteur de résolution

Le fonctionnement du moteur de résolution dépend du type de questions posées :

Cohérence : on appelle directement un solveur SAT standard (ici minisat [ES03]) qui vérifie la cohérence des formules décrivant le graphe et les conditions initiales avec les conditions finales de la question.

Abduction : on construit l'ensemble des conditions initiales possibles, et pour chacune d'entre elles, on appelle le solveur SAT afin de vérifier la cohérence avec les conditions finales.

Mise à jour de graphe : on construit l'ensemble des graphes possibles et on appelle le solveur SAT pour en vérifier la cohérence. S'il existe des variables libres, on ajoute l'étape d'abduction avant l'appel du solveur SAT.

La traduction en logique temporelle linéaire du graphe, puis la traduction de cette représentation en logique propositionnelle par réécriture et réification en associant à chaque état du modèle temporel un nombre naturel, s'effectue une fois que le graphe est fixé, c'est à dire après l'appel de l'énumérateur de graphes s'il a lieu ; dans ce cas, il faudra traduire successivement chacun des graphes énumérés. Les fondements logiques et le détail de la méthode de traduction sont décrits dans [ADFdC16, ADD$^+$16, CMD17], nous n'y reviendrons pas ici.

5 Exemples d'utilisation

Nous allons nous concentrer ici sur les deux types les plus intéressants et les plus complexes à traiter : l'abduction et la mise à jour de graphe.

5.1 Abduction

Dans cette section, nous allons nous intéresser à un exemple complexe ; nous allons reprendre une fraction significative de la carte présentée dans la figure 1 et la modéliser sous forme de graphe ; nous allons nous intéresser à la voie métabolique atm-chk2, qui amène à l'apoptose cellulaire lorsqu'une rupture du double brin d'ADN se produit.
La rupture du double brin d'ADN (Double strand break ou *dsb*) est une cause majeure de cancer, et les recherches médicales et pharmaceutiques [KP05, GMP$^+$11] ont montré que la rupture du double brin d'ADN peut apparaître dans une cellule comme la résultante d'une pathologie dans une voie métabolique.
Ce type de carte est utilisée pour étudier les déterminants moléculaires de la réponse tumorale aux cancers. Les paramètres moléculaires incluent la voies métabolique de la réparation de l'ADN, celle de l'apoptose programmée de la cellule et celle des points de contrôle du cycle cellulaire [PSR$^+$05, KP05, GMP$^+$11, LKLC07, PZL$^+$11]. Lorsque l'ADN est endommagée, les points de contrôle du cycle cellulaire sont activés et peuvent rapidement tuer la cellule par apoptose, ou arrêter la progression du cycle cellulaire pour permettre la réparation de l'ADN avant la reproduction ou la division cellulaire. Deux de ces points de contrôle sont les voies métaboliques atm-$chk2$ (que nous allons utiliser comme exemple ici) et atr-$chk2$ [PSR$^+$05].
Sur la figure 9 (construite à partir de la partie de la carte de la figure 1 représentant la voie atm-$chk2$) nous avons le graphe représentant la voie métabolique atm-$chk2$ qui conduit de trois façons différentes à l'apoptose cellulaire. Il y a dans ce graphe six variables exogènes : atm, dsb, $chk2$, $mdm2$, pml et $p53$. Toutes les autres variables sont endogènes. Certaines de ces variables sont des protéines, alors que d'autres comme dsb qui représente la rupture du double brin d'ADN, ou $apoptose$ qui représente la mort cellulaire, sont des conditions ou des états.
La possibilité de faire de l'abduction sur ce type de graphe dépend du nombre de pas de temps considérés ; sur cet exemple, si nous utilisons t pas de temps, nous avons $20 + 42t$ variables et $14 + 145t$ clauses. Cependant, si le nombre de variables libres reste relativement bas (ce qui est le cas ici), les temps de réponse du système sont extrêmement rapides. En effet, si nous avons n variables libres, nous ne devons réaliser que 2^n appels au solveur SAT, car seules les valeurs des variables libres ont un intérêt pour l'expérimentateur. Dans le cas présent, en utilisant par exemple 11 pas temporels (482 variables et 1609 clauses), le système répond en quelques millisecondes à la question lui demandant de trouver les conditions initiales menant à l'apoptose cellulaire [4].

4. L'utilisation d'un algorithme d'abduction "pur" comme celui de Tsiknis [KT90] amène en revanche à des temps de résolution très élevés. La raison est que l'algorithme de Tsiknis travaille sur l'ensemble des variables, alors que les seules valeurs intéressant les biologistes sont les valeurs des variables exogènes.

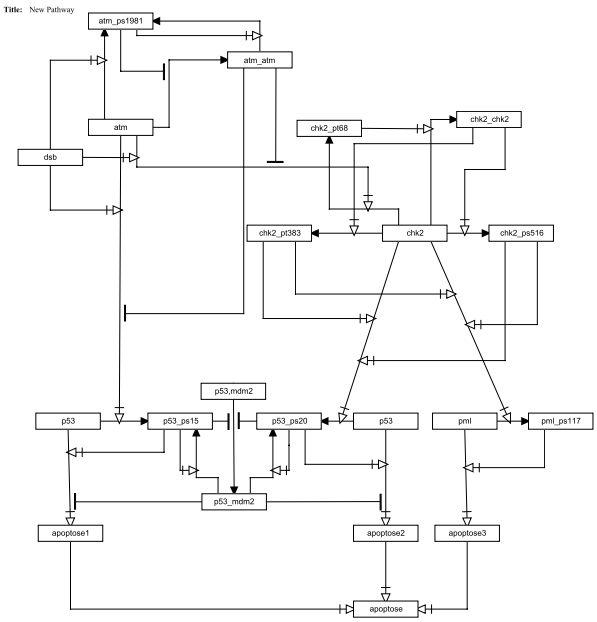


FIGURE 9 – carte d'interaction moléculaire atm-$chk2$ (avant initialisation des variables)

Il est même possible de raffiner les questions en demandant par exemple combien de temps il faut pour atteindre l'apoptose cellulaire sur chacun des trois voies métaboliques, et quelles sont les conditions initiales qui y mènent. Il suffit de poser les questions $apoptose1_i$, $apoptose2_i$ et $apoptose3_i$ pour tous les i en partant de $i = 0$.

- $apoptose1$ est la voie la plus courte et $apoptose1_2$ (second pas de temps) est vraie si atm, dsb and $p53$ sont présents, $mdm2$ est absent et les valeurs de pml et $chk2$ sont indifférentes (la réponse est correcte et peut facilement être vérifiée sur la carte). Pour $i \geq 3$, la réponse à $apoptose1_i$ est la même sauf que $mdm2$ devient aussi indifférent ce qui est cohérent avec le graphe ($p53\_mdm2$ est dissocié à l'étape 2).
- $apoptose2$ est le chemin le plus court et la première réponse positive est obtenue pour $apoptose2_5$; atm, $chk2$, dsb, $p53$ doivent être présents, et $mdm2$ et pml sont indifférents.
- $apoptose3$ demande le même nombre d'étapes que $apoptose2$ mais les conditions sont différentes : atm, $chk2$, dsb, et pml doivent être présents, $mdm2$ et $p53$ sont indifférents.

5.2 Mise à jour de graphe

Dans la figure 10 nous voyons à nouveau la représentation de l'opéron lac mais nous avons retiré l'inhibition exercée par le lactose sur la régulation négative du répresseur sur

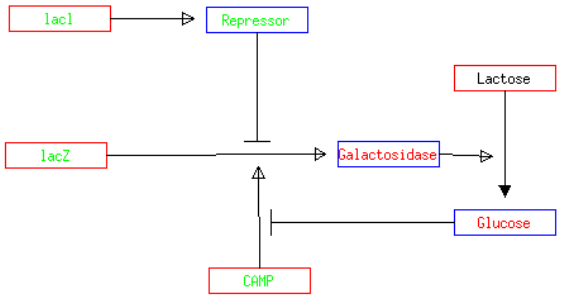


FIGURE 10 – Opéron lac sans l'inhibition par le lactose

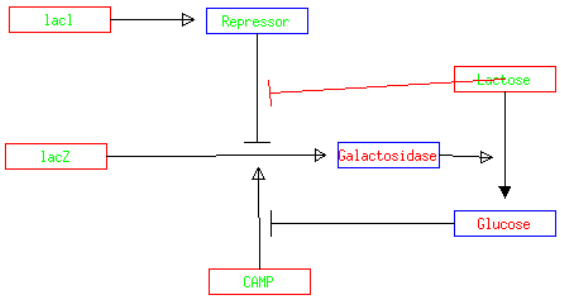


FIGURE 11 – Solution correcte

la production de la galactosidase. Dans ce cas, le glucose n'est pas produit en présence de lactose.

L'utilisateur peut demander au système quels ajouts peuvent être faits sur le graphe pour que le glucose soit produit. La solution exacte est trouvée instantanément (figure 11) avec une dizaine d'autres. Certaines sont sans intérêt, comme par exemple la production directe de glucose par un des gènes lacZ ou lacI.

En revanche le système propose aussi d'autres solutions, comme celle représentée sur la figure 12. Ici le glucose est nécessaire pour activer l'action inhibitrice de la protéine répresseur sur la réaction de production de la galactosidase. Donc, en présence de glucose, la production de galactosidase est stoppée, et elle est effectuée en l'absence de glucose. Il faut cependant noter que la nature a choisi la solution la plus "économique", puisque dans le cas proposé ici la galactosidase sera produite dès qu'il n'y a pas de glucose, qu'il y ait du lactose ou pas, alors que dans la solution standard, la galactosidase n'est produite qu'en l'absence de glucose et en présence de lactose.

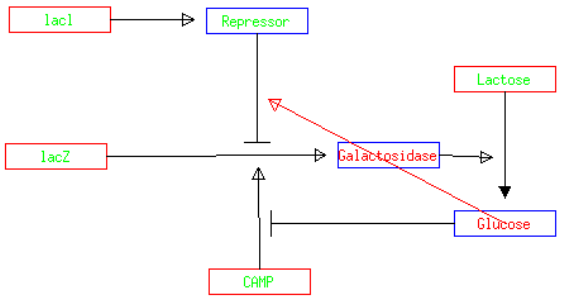


FIGURE 12 – Autre solution "intéressante"

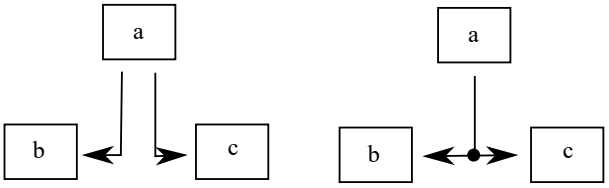


FIGURE 13 – Opérateur •

6 Conclusion

Nous avons présenté dans cet article une méthode pour traduire des graphes représentant des systèmes biologiques en logique temporelle, et un logiciel permettant de répondre à des questions complexes sur ces graphes. Le logiciel a aujourd'hui atteint le stade d'un prototype qui peut être testé sur des exemples réalistes de grande taille.

Il reste cependant un certain nombre de points à résoudre :

— Les graphes que nous utilisons sont limités par les relations élémentaires (productions, activation, inhibition) et la syntaxe limitée que nous autorisons dans les graphes. Ces relations élémentaires permettent de traduire des relations plus complexes (comme l'opérateur • qui permet par exemple de dupliquer une relation de production, d'activation ou d'inhibition, voir figure 13), mais dans l'état actuel du logiciel c'est l'utilisateur qui doit lui même décomposer les relations complexes en relations élémentaires. L'implantation d'un plus grand nombre de relations améliorerait l'expérience utilisateur.

— Les graphes sont créés par des biologistes, qui s'appuient sur leur "connaissance métier". Cette connaissance comprend souvent un ensemble de non-dits et d'implicites, qu'ils n'incluent pas dans les graphes. L'expérience montre que la construction formelle d'un graphe complexe comme celui de $atm\text{-}chk2$ (figure 9) a demandé de nombreuses itérations entre biologistes et informaticiens pour saisir la totalité de la connaissance nécessaire pour saisir la logique complète du système. Résoudre ce problème pourrait peut-être se faire à travers l'utilisation d'une interface graphique permettant de modifier directement le graphe et de le tester interactivement, alors qu'aujourd'hui toute modification de graphe impose de recourir à Pathvisio.

— La saisie des questions se fait à travers une interface

textuelle, et demande un minimum de compréhension de fonctionnement du système au niveau logique pour poser les "bonnes" questions. Là encore, le développement d'un outil graphique permettant de saisir les questions dans un format plus naturel serait largement appréciable.

— La vitesse des réactions n'est pas prise en compte. Une solution en cours d'étude est le passage au dual : au lieu de représenter la vitesse d'une réaction, nous pensons représenter la durée nécessaire pour qu'une réaction se déclenche.

— Le dernier point, plus complexe à résoudre, est que notre système repose sur l'hypothèse du "tout ou rien". Nous ne savons pas représenter les quantités intermédiaires, un acteur est soit présent, soit absent. Il s'agit à aussi d'un axe d'amélioration sur lequel nous travaillons.

Remerciements

Ce travail a bénéficié du soutien du projet ANR-11-LABX-0040-CIMI au sein du programme ANR-11-IDEX-0002-02, de l'IREP Associated European Laboratory et du projet CLE de la région Midi-Pyrénées .

Références

[ADD+16] Jean-Marc Alliot, Robert Demolombe, Martín Diéguez, Luis Fariñas del Cerro, Gilles Favre, Jean-Charles Faye, Naji Obeid, and Olivier Sordet. Temporal logic modeling of biological systems. In Seiki Akama, editor, *Towards Paraconsistent Engineering*, pages 205–226. Springer International Publishing, 2016.

[ADFdC16] Jean-Marc Alliot, Martín Diéguez, and Luis Fariñas del Cerro. Metabolic pathways as temporal logic programs. In Loizos Michael and Antonis Kakas, editors, *Logics in Artificial Intelligence*, pages 3–17. Springer International Publishing, 2016.

[CMD17] Serenella Cerrito, Marta Cialdea Mayer, and Robert Demolombe. Temporal abductive reasoning about biochemical reactions. *Journal of Applied Non-Classical Logics*, 27(3-4) :269–291, 2017.

[CRFS04] Nathalie Chabrier-Rivier, Francois Fages, and Sylvain Soliman. The Biochemical Abstract Machine BIOCHAM. In Vincent Danos and Vincent Schächter, editors, *CMSB'04 : Proceedings of the second Workshop on Computational Methods in Systems Biology*, volume 3082, pages 172–191, Paris, 2004. Springer-Verlag.

[ES03] N. Een and N. Sörensson. An extensible satsolver. In *SAT'03*, pages 502–518, 2003.

[FI14] L. Farinas and K. Inoue, editors. *Logical Modeling of Biological Systems*. John Wiley & Sons, 2014.

[GMP+11] V. Glorian, G. Maillot, S. Poles, J. S. Iacovoni, G. Favre, and S. Vagner. Hur-dependent loading of mirna risc to the mrna encoding the ras-related small gtpase rhob controls its translation during uv-induced apoptosis. *Cell Death & Differentiation*, 18(11) :1692–70, 2011.

[JM61] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3 :318–356, 1961.

[KAWP06] K. W. Kohn, M. I. Aladjem, J. N. Weinstein, and Y. Pommier. Molecular interaction maps of bioregulatory networks : A general rubric for systems biology. *Molecular Biology of the Cell*, 17(1) :1–13, 2006.

[KP05] K. W. Kohn and Y. Pommier. Molecular interaction map of the p53 and mdm2 logic elements, which control the off-on switch of p53 response to dna damage. *Biochemical and biophysical research communications*, 331(3) :816–27, 2005.

[KT90] A. Kean and G. Tsiknis. An incremental method for generating prime implicants/implicates. *Journal of Symbolic Computing*, 9 :185–206, 1990.

[LKLC07] W. Lee, D. Kim, M. Lee, and K. Choi. Identification of proteins interacting with the catalytic subunit of pp2a by proteomics. *Proteomics*, 7(2) :206–214, 2007.

[PSR+05] Y. Pommier, O. Sordet, V. A. Rao, H. Zhang, and K. W. Kohn. Targeting chk2 kinase : molecular interaction maps and therapeutic rationale. *Current pharmaceutical design*, 11(22) :2855–72, 2005.

[PZL+11] H. Pei, L. Zhang, K. Luo, Y. Qin, M. Chesi, F. Fei, P. L. Bergsagel, L. Wang, Z. You, and Z. Lou. MMSET regulates histone H4K20 methylation and 53BP1 accumulation at DNA damage sites. *Nature*, 470(7332) :124–128, 2011.

[vIKP+08] M. P. van Iersel, T. Kelder, A. R. Pico, K. Hanspers, S. Coort, B. R. Conklin, and C. Evelo. Presenting and exploring biological pathways with pathvisio. *BMC Bioinformatics*, page 399, 2008.

Multimodal deep networks for text and image-based document classification

Nicolas Audebert Catherine Herold Kuider Slimani Cédric Vidal

Quicksign, 38 rue du Sentier, 75002 Paris

{nicolas.audebert,catherine.herold,kuider.slimani,cedric.vidal}@quicksign.com

Résumé

La classification automatique de documents numérisés est importante pour la dématérialisation de documents historiques comme de procédures administratives. De premières approches ont été suggérées en appliquant des réseaux convolutifs aux images de documents en exploitant leur aspect visuel. Toutefois, la précision des classes demandée dans un contexte réel dépend souvent de l'information réellement contenue dans le texte, et pas seulement dans l'image. Nous introduisons un réseau de neurones multimodal capable d'apprendre à partir d'un plongement lexical du texte extrait par reconnaissance de caractères et des caractéristiques visuelles de l'image. Nous démontrons la pertinence de cette approche sur Tobacco3482 et RVL-CDIP, augmentés de notre jeu de données textuel QS-OCR[1]*, sur lesquels nous améliorons les performances d'un modèle image de 3% grâce à l'information sémantique textuelle.*

Mots-clés

Classification de documents, apprentissage multimodal, fusion de données.

Abstract

Classification of document images is a critical step for archival of old manuscripts, online subscription and administrative procedures. Computer vision and deep learning have been suggested as a first solution to classify documents based on their visual appearance. However, achieving the fine-grained classification that is required in real-world setting cannot be achieved by visual analysis alone. Often, the relevant information is in the actual text content of the document. We design a multimodal neural network that is able to learn from word embeddings, computed on text extracted by OCR, and from the image. We show that this approach boosts pure image accuracy by 3% on Tobacco3482 and RVL-CDIP augmented by our new QS-OCR text dataset[1]*, even without clean text information.*

Keywords

Document classification, multimodal learning, data fusion.

[1] https://github.com/Quicksign/ocrized-text-dataset

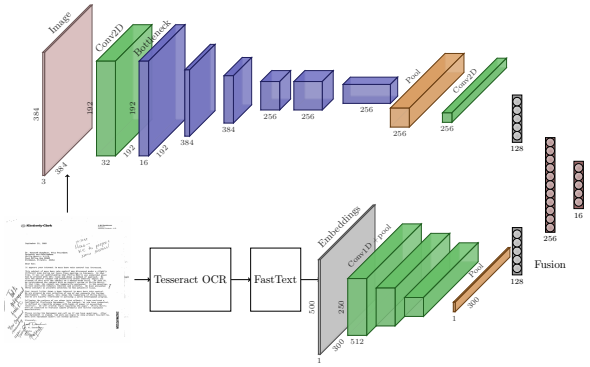
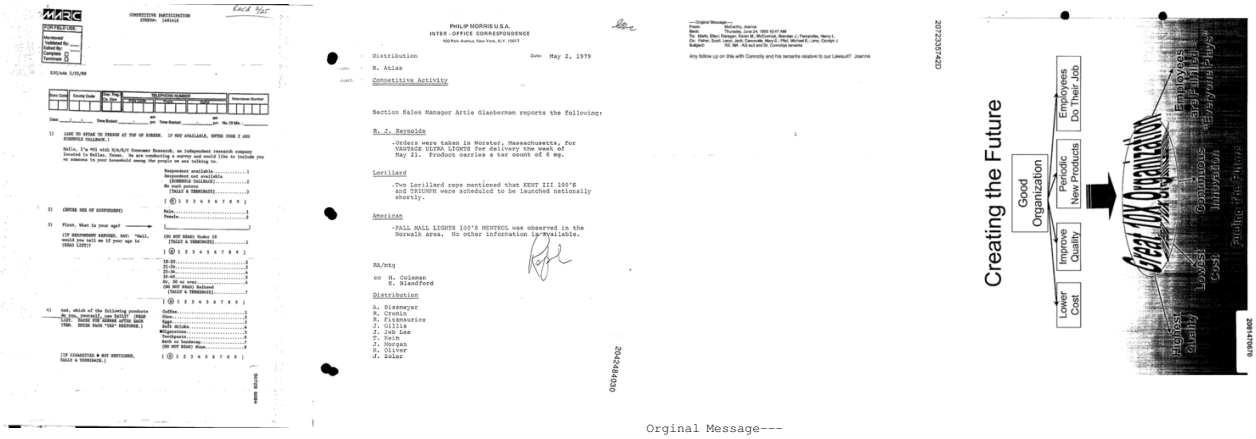


Figure 1: Multimodal classifier for hybrid text/image classification. Training is performed end-to-end on both textual and visual features.

1 Introduction

The ubiquity of computers and smartphones has incentivized governments and companies alike to digitize most of their processes. Onboarding new clients, paying taxes and proving one's identity is more and more done through a computer, as the rise of online banking has shown in the last few years. Industrial and public archives are also ongoing serious efforts to digitize their content in an effort for preservation, e.g. for old manuscripts, maps and documents with a historical value. This means that previously physical records, such as forms and identity documents, are now digitized and transferred electronically. In some cases, those records are produced and consumed by fully automated systems that rely on machine-readable formats, such as XML or PDF with text layers. However, most of these digital copies are generated by end-users using whatever mean they have access to, i.e. scanners and cameras, especially from smartphones. For this reason, human operators have remained needed to proofread the documents, extract selected fields, check the records' consistency and ensure that the appropriate files have been submitted. Automation through expert systems and machine learning can help accelerate this process to assist and alleviate the burden of this fastidious work for human workers.

A common task involved in data filing processes is document recognition, on which depends the class-specific rules that command each file. For example, a user might be asked to upload several documents such as a filled subscription form,



Keck Zr COMPETITIVE
PARTICIPATION
sTupYe: 1401618 Sic/sds
2/25/88 .
= Coren aE a Dale:__/{__j__
Time Ended: 4 'pm
Time Started:. pm
No. Of Min. :, 1)
(ASK To SPEAK TO PERSON AT
TOP OF SCREEN, IF NOT
AVAILABLE, ENTER CODE 2 AY
SCHEDULE CALLBACK
...

PHILIP MORRIS U.S.A. koe
INTER - OFFICE
CORRESPONDENCE
100 Park Avenua, New York,
N.Y. 10017
Distribution Date:
May 2, 1979
R, Atlas
Competitive Activity

Section Sales Manager Artie
Glacberman reports the
following:
...

Orginal Message---
From: McCarthy, Josnne
Sent. 'Thursday, June 24,
1990 10:47 AM
Tor Met, Eten; Daragan,
Karon M; McCormick, Brendan;
Femandez, Henry L.
(Ge: Fisher, Scot: Lon
Jock: Gamavalo, Mary ;
Pol, Michael. Ley, Ceo} J
Subject: RE: MA^{TM}-AG out
and Dr. Connolys remarks
'Any follow up on this with
Connolly and his remarks
relative to our Lawsuit?
Joanne Oztrlseezloz

Creating the Future
Good
Organization

Lower Improve Periodic
Employees e
Cost Quality New Products
Do Their Job

029021807

(a) Questionnaire (b) Memo (c) Email (d) Presentation

Figure 2: Document samples from the RVL-CDIP [1] dataset with corresponding text extracted by Tesseract OCR.

an ID and a proof-of-residence. In this work, we tackle the document classification task to check that all required files have been sent so that they are filed accordingly.

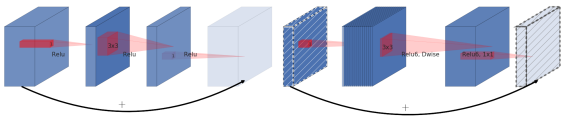
Yet, if discriminating between broad classes of documents can be achieved based on their appearance only (e.g. separating passports from banking information), fine-grained recognition often depends on the textual content of the documents. For example, different tax forms might share their layout, logos and templates while the content in itself vastly differs. Computer vision has been interested for some time in optical character recognition (OCR) to extract text from images. However, dealing with both the textual and visual contents remains an open problem. In the past years, deep learning has been established as the new state-of-the-art for image classification and natural language processing. For fine-grained document recognition, we expect the model to leverage both image and text information.

This work introduces a multimodal deep network that learns from both a document image and its textual content automatically extracted by OCR to perform its classification. We design a pragmatic pipeline for end-to-end heterogeneous feature extraction and fusion under time and cost constraints. We show that taking both the text and the document appearance into account improves both single modality baselines by several percents on two datasets from the document recognition literature. We detail some limitations of the current academic datasets and give leads for an application in an industrial setting with unclean data, such as photographed documents.

2 Related work

Analyzing digitized documents is an old task in computer vision that was boosted by the dissemination of computers in offices and then of digital cameras and smartphones in everyday life. To allow for textual search and easy indexing, the critical part of digitization is extracting text content from documents that have been scanned or photographed. Indeed, either when scanning or taking a picture of the document, its actual text is lost, although it is implicitly embedded in the pixel values of the image. Numerous optical character recognition (OCR) algorithms have been designed to transform images into strings of characters [2, 3]. Despite those efforts perfectly reading any type of document remains challenging due to the wide variety of fonts and languages. Layout analysis is a way to preprocess the data to detect text areas and find the text orientation in order to enforce a better local and global consistency [4, 5].

Document image analysis is also one of the first topic where modern deep learning has been applied. The first convolutional neural network (CNN) [6] was originally designed for classification of digits and letters. The computer vision community deployed consequent efforts to achieve image-based document classification without text, as shown by a 2007 survey [7] which focuses on document image classification without OCR results. As an example, [8] introduced SURF visual features with a bag-of-words scheme to perform document image classification and retrieval. In 2015, [1] introduced a large labeled image document dataset which sparked interest and generated several studies of deep CNN on this topic [9, 10, 11], inspired by the success of



(a) Residual block (b) Inverted residual block

Figure 3: MobileNetV2 uses inverted residual blocks to reduce the number of channels that are forwarded in subsequent layers. Figure from [23].

these networks on ImageNet and tuning data augmentation policies, transfer learning strategies and domain adaptation for document classification. In the same idea, [12] also investigated such deep architectures to classify identity documents. [13] goes even further by trying to segment the full layout of a document image into paragraphs, titles, ornaments, images etc. These models focus on extracting strong visual features from the images to classify the documents based on their layout, geometry, colors and shape.

On the other hand, text-based document classification has also long been investigated. In 1963, [14] introduced an algorithmic approach to classify scientific abstracts. More recently, [15] experimented with one-class SVM for document classification based on various text features, such as TF-IDF. [16] used Latent Dirichlet Allocation to perform topic modeling and used it as a generative approach to document classification. The recent appearance of learned word embeddings approaches such as word2vec [17] or ELMo [18] paved to way to a large body of works related to recurrent and attention mechanisms for text classification. For example, [19] proposed a bidirectional recurrent network with a hierarchical attention mechanism that learns both at the word and sentence levels to improve document classification.

Some works tried to reconcile the text-based and image-based approaches to exploit both information sources. [20] performs OCR to detect keywords in images which are then encoded as colored boxes before passing the image through a CNN. While a clever trick, this does not leverage the representation power of word embeddings. Closer to our approach, [21] goes further by generating text feature maps that are combined with visual feature maps in a fully convolutional network. However, the considered documents are synthetic and the network is trained using perfectly clean texts and images, which is unrealistic for practical uses. More similar to us, [22] learns to combine bag of words and bag of visual words features for industrial document images using a statistical model combining outputs of two single-modality classifiers. While using shallow features, they show that using both information allows for a better accuracy when the OCR is unreliable, which is often the case in an industrial setting.

In this paper, we go further in this direction and propose a new baseline with a hybrid deep model. In order to classify OCRized document images, we present a pragmatic pipeline perform visual and textual feature extraction using off-the-shelf architectures. To leverage the complementary information present in both modalities, we design an efficient end-to-end network that jointly learn from text and image while keeping computation cost at its minimum. We build on existing deep models (MobileNet and FastText) and demonstrate significant improvements using our fusion strategy on two document images dataset.

3 Learning on text and image

3.1 Visual features

There is a large literature both in general image recognition and in image document classification. Recent works have established deep convolutional neural networks as the *de facto* state of the art on many competitions in object recognition, detection and segmentation, e.g. ImageNet. Deep features, extracted by pretrained or fine-tuned deep CNNs, constitute a strong baseline for visual recognition tasks [24]. Based on this, we choose to fine-tune a CNN pretrained on ImageNet in order to extract visual features on our images, as suggested in several recent document classification publications [9, 10, 1] As we aim to perform inference on a large volume of data with time and cost constraints, we focus on a lightweight architecture with competitive classification performance, in our case the MobileNet v2 model [23].

MobileNetV2 [23] consists in a stack of bottleneck blocks. Based on the residual learning principle [25], each bottleneck block transforms a feature map first by expanding it by increasing its number of channels with a 1×1 convolutional layer with identity activation. Then, a 3×3 depthwise convolution is performed, followed by a ReLU and a final 1×1 convolution with ReLU. For efficiency issues, this block inverts the traditional residual block since the expansion is performed inside the block, whereas residual blocks compress and then reexpand the information, as illustrated in Fig. 3. The final MobileNetV2 contains 19 residual bottleneck layers. Compared to other state of the art CNNs, MobileNetV2's accuracy is on-par with VGG-16 while being significantly faster.

3.2 Textual features

Since our use case focuses on document images in which the text has not been transcribed, we need to perform an OCR step. To this end, we use the Tesseract OCR engine [3] in its 4.0 version which is based on an LSTM network. Tesseract is configured in English to use full page segmentation and the LSTM engine. In practice, this means that Tesseract will try to detect the text orientation in the image and perform the needed affine transformation and rotation if any. Tesseract also deals with the image binarization using Otsu's thresholding to identify black text on white background [26]. This will suffice on the datasets described in Section 4.1, although we found Tesseract challenging to apply on real-world images, especially pictures which are not flat and grayscale scans.

Recent literature in NLP suggests that pretrained word embeddings offer a strong baseline which surpasses traditional shallow learning approaches. Many word embeddings have been designed following the initial success of *word2vec* [17], such as GloVe [27] or more recently the contextualized word embeddings from ELMo [18].

However, those word embeddings assume a good tokenization of the words, i.e. most embeddings remove digits, ignore punctuation and do not deal with out-of-vocabulary (OOV) words. Since these embeddings are learned on clean corpus (e.g. Wikipedia or novels), tokenization is fairly straightforward. OOV words are either assigned a random embedding or mapped to the closest in-vocabulary word based on the Levenshtein distance.

Unfortunately, outputs of the Tesseract OCR are noisy and not as clean as the training data from these embeddings. Even in grayscale, well-oriented documents, OCR might have trouble dealing with diacritics, exotic fonts or curved text, as illustrated by the extracts from Fig. 2. Moreover, specific user domains (e.g. banking or medieval manuscripts) might use rare words, codes, abbreviations or overall jargon that is absent from general-purpose word embeddings. Since we face many possible misspellings in the extracted text, we cannot use the previous workarounds for OOV embeddings since it would inject a lot of non-discriminant features in our text representation In average, on the Tobacco3482 corpus, a document processed by Tesseract OCR contains 136 words with 4 characters or more. Of those, only 118 in average are in the GloVe embeddings [27][2] and only 114 are in Enchant's spellchecker US English dictionary. Overall, approximately 26% of the corpus is absent from the US English dictionary and 23% from the GloVe embeddings. The document distribution with respect to the proportion of out-of-vocabulary words is shown in Fig. 4a. Although most of the documents are concentrated around 10% of OOVs, there is a significant long tail including several dozens of documents that contain only words outside of the English language.

Therefore, we turn to character-based word embeddings that are able to deal with OOV words by assigning them plausible word vectors that preserve both a semantic and a spelling similarity. One possibility was to use the mimicking networks from [28] that learn to infer word embeddings such as GloVe, but based only on subword information. More complex embeddings such as FastText [29, 30] and ELMo [18], which produce vectors using respectively n-grams and subword information, can also address this problem. Finally, the Magnitude library [31] uses two alternative strategies to deal with OOV words:

- Assigning a *deterministic* random vector. These vectors do not capture semantic sense, however similar words based on the Levenshtein-Damerau distance will have similar vectors. Misspellings will therefore not be close to the original word, but similar lingo words will be close.
- Using character n-grams inspired by [29] and interpolation with in-vocabulary words, Magnitude can generate vectors for OOV words which are sensible based on existing learned embedding.

Preliminary data exploration shows that subword-aware embeddings perform better at preserving similarity despite misspellings, as illustrated in Fig. 4b. We therefore focus our interest on the FastText embedding, which is faster than ELMo since the latter requires passing the context through a bidirectionnal LSTM during inference. It is worth noting that this raises concern for characters that have not been seen by FastText. We found experimentally that Tesseract OCR generated no character that was OOV for FastText on the documents we considered.

Finally, it is necessary to convert those word embeddings into a document embedding. We consider two approaches:

- The simple baseline for sentence embedding suggested in [32], which consists in a weighted average of word embeddings altered by PCA.
- Using variable-length document embeddings consisting in a sequence of word embeddings.

The first approach is suitable as generic feature while the second requires a statistical model able to deal with sequences, such as recurrent or convolutional neural networks. For both methods, we use the SpaCy small English model [33] to perform the tokenization and punctuation removal. Individual word embeddings are then inferred using FastText [29] pretrained on the Common Crawl dataset.

3.3 Multimodal features

Once text and image features have been extracted, we feed them to a multi-layer perceptron following [34]. To do so, we need to combine both feature vectors into one. Two approaches can be envisioned:

- Adaptive averaging of both feature vectors. This aligns both feature spaces so that scalars at the same index become compatible by summation, i.e. that each dimension of the vectors have a similar semantic meaning.
- Concatenating both vectors. This does not imply that both feature spaces can be aligned and delegates to the fusion MLP the task of combining the two domains.

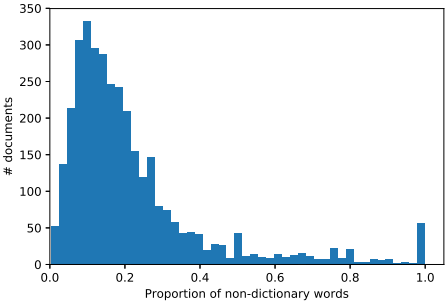
Both fusion strategies are differentiable, therefore the whole network can be trained in an end-to-end fashion. Moreover, the model is modular and each feature extractor can be swapped for another model, e.g. MobileNet can be exchanged with any other popular CNN and FastText could be replaced by subword-level NLP models, even differentiable ones that could allow fine-tuning the embeddings. In this work, we try to keep things simple and build on robust base networks in order to clearly understand how the data fusion impacts model performance. Preliminary experiments showed that the summation fusion significantly underperformed compared to pure image baseline. We suggest that this is provoked by the impossibility of aligning the text and image feature spaces without breaking their discriminating power, resulting in suboptimal space. Therefore, we move on with the concatenation strategy for the rest of this paper. The complete pipeline is illustrated in Fig. 1.

4 Experimental setup

4.1 Datasets

Tobacco3482. The Tobacco3482 dataset [8] contains 3482 black and white documents, a subset from the Truth

[2] Based on the Wikipedia 2014 + Gigaword 5 datasets.



(a) Document distribution w.r.t of non-dictionary words % in the Tobacco3482-Tesseract corpus.

Word pair	Similarities		
`specifically`	GloVe	ELMo	FastText
`Specificalily`	0.71	0.68	0.96
`filter`	GloVe	ELMo	FastText
`fiilter`	0.91	0.73	0.96
`alcohol`	GloVe	ELMo	FastText
`Aleohol`	0.40	0.69	0.88
`Largely`	GloVe	ELMo	FastText
`Largly`	0.25	0.81	0.98

(b) Word embeddings similarity for misspelled words.

Figure 4: Tesseract OCR outputs noisy text that does not entirely overlap with the assumptions usually held when training word embeddings for NLP.

Tobacco Industry Documents[3] archives of legal proceedings against large American tobacco companies. There are annotations for 10 classes of documents (e.g. email, letter, memo...). Following common practices, we perform k-fold cross-validation using 800 documents for training and the rest for testing. Results are averaged over 3 runs.

RVL-CDIP. The RVL-CDIP dataset [1] is comprised of 400000 grayscale digitized documents from the Truth Tobacco Industry Documents. There are annotations for 16 classes of documents (e.g. email, letter, invoice, scientific report...), each containing 25000 samples. We use the standard train/val/test split from [1] with 320000 documents for training, 40000 for validation and 40000 for testing.

Text generation. The Tobacco3482 and RVL-CDIP are image-based datasets. In order to evaluate our multi-modal networks, we wish to learn from both visual and textual content. Therefore we use the Tesseract OCR library[4] to extract text from the grayscales images. We perform this operation on both datasets. We release the OCR text dataset openly[5] to encourage other researchers to replicate our work or test their own model for post-OCR text classification or multi-modal text/image classification.

4.2 Models

This subsection describes the implementation details of our deep networks. All models are implemented in TensorFlow 1.12 using the Keras API and trained using a NVIDIA Titan X GPU. Hyperparameters were manually selected on a subset of Tobacco3482 and fixed for all experiments.

Text baseline. Seeing that our representation of textual data can be either a document embedding or a sequence of word embeddings, we compare two models for our text baseline.

The first model is an improved Multi-Layer Perceptron (MLP) with ReLU activations, Dropout and Batch Normalization (BN) after each layer. The network has a fixed width of 2048 neurons for all layers except the last one, which produces a 128 feature vector, classified by a softmax layer. Weights are randomly initialized using He's initialization [35]. The averaged document embedding [32] is used as an input for this classifier.

The second model is a one-dimensional convolutional neural network designed inspired by previous work for sentence classification [36]. The CNN is 4-layers deep and interlaces 1D convolutions with a window of size 12 with maxpooling with a stride of 2. Each layer consists in 512 channels with ReLU activation. The final feature map is processed by a max-pooling-through-time layer that extracts maximal features on the sequence on top of which we apply Dropout for regularization. A fully connected layer then maps the features to the softmax classifier. The input word sequence is zero-padded up to 500 words for documents with less 500 words.

We experiment on the Tobacco3482 dataset in order to evaluate which text model to choose. Results are reported in Table 1a. Without surprise, the CNN 1D outperforms significantly the MLP classifier. The pattern recognition abilities of the convolutional network makes it possible to interpret the word sequences by leveraging contextual information. Since only some part of the text might be relevant, averaging over all word embeddings dilute the discriminating information. Moreover, noisy embeddings due to garbage output from Tesseract (e.g. incoherent strings where OCR has failed) are included in the final document embedding. However, when dealing with word sequences, convolutional layers and temporal max-pooling help extracting only the relevant information. Therefore, we choose to include the 1D CNN as the text component in our multimodal architecture. This model is denoted **TEXT** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 100 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40[6].

Image baseline. We investigate as our base CNN the lightweight MobileNetV2 [23] which focuses on computing efficiency, albeit at the cost of a slightly lower top-1 accuracy on ImageNet compared to other state of the art CNN. We

[3] `https://www.industrydocuments.ucsf.edu/tobacco/`

[4] `https://github.com/tesseract-ocr/tesseract/`

[5] The QS-OCR dataset is available at: `https://github.com/Quicksign/ocrized-text-dataset`

[6] Hyperparameters are manually tuned on a small validation set.

(a) Preliminary experiments on Tobacco3482 for the text baseline.

Model	OA	F_1
MLP (document)	70.8%	0.69
CNN 1D (word sequence)	73.9%	0.71

OA = overall accuracy, F_1 = class-balanced F_1 score.

(b) Preliminary experiments on Tobacco3482 for the image baseline.

Model	OA	F_1
MobileNetV2	84.5%	0.82
MobileNetV2 (w/ DA)	83.9%	0.82

OA = overall accuracy, F_1 = class-balanced F_1 score, DA = data augmentation.

Table 1: Preliminary tuning of the single-modality baselines on Tobacco3482.

train the CNN on grayscale document images resized at 384×384. Although this warps the aspect ratio, [9] reports better accuracy than when using padding at the same resolution. As the model is designed for RGB images, the grayscale channel is duplicated three times. This allows us to initialize the network by loading its pretrained weights on ImageNet, which accelerates convergence and slightly improves accuracy through transfer learning.

This model is denoted **IMAGE** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 200 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40.

As reported in Table 1b, preliminary experiments on the Tobacco3482 with random JPEG artifacts, saturation and contrast alterations did not significantly alter the classifier's accuracy compared to no augmentation. This is explained by the low variability between the grayscale document images. All images are grayscale with dark text on white background with horizontal text lines, therefore color and geometric augmentation are not necessary. However, [9] report some success using shear transform, which we did not consider in this work. It is worth noting that compared with previous literature on the RVL-CDIP dataset, e.g. [9, 10, 1], we do not average predictions over multiple crops at inference time for speed concerns. This might explain why our visual baseline underperforms the current state of the art in this state (although this does not question the gains due to the multi-modal network).

Fusion. For our multimodal network, we consider the same model as our baselines except that the final layers are cut-off. For the **TEXT** model, the last layer produces an output vector of dimension 128 instead of the number of classes. For the **IMAGE** model, we aggregate the last convolutional features using global average pooling on each channel, which produces a feature vector of dimension 1280. We then map this feature vector using a fully connected layer to a representation space of dimension 128.

This model is denoted **FUSION** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 200 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40.

5 Discussion

5.1 Performances

Model performances scores on Tobacco3482 and RVL-CDIP are reported in Tables 2 and 3. Behaviour of all models is consistent both on the smaller dataset and on the very large one. In both cases, the **TEXT** baseline is significantly underperforming the **IMAGE** one. Indeed, as could be seen in Fig. 2, Tesseract OCR outputs noisy text. This includes words that have been misspelled – which are correctly dealt with by the FastText embeddings – and new words that are hallucinated due to poor binarization or salt-and-pepper noise in the image. Moreover, layout and visual information tends to be more informative based on how the classes were defined: scientific papers, news and emails follow similar templates while advertisements present specific graphics. However, in both cases, this simple document embedding is enough to classify more than 70% of the documents, despite its roughness.

Using the **IMAGE** model only, we reach accuracies competitive with the state of the art. MobileNetV2 alone does on-par is with the holistic CNN ensemble from [1] and is competitive with fine-tuned GoogLeNet and ResNet-50 [10] (90.97%).

On both datasets, the fusion scheme is able to improve the overall accuracy by $\simeq 1.5\%$ which demonstrates the relevance of our approach. While the document embedding we chose is simple, it appears to be at least partially robust to OCR noise and to preserve enough information about the document content to boosts CNN accuracy on document image classification even further. We also report the results from an oracle, which corresponds to the perfect fusion of the **TEXT** and **IMAGE** baselines, i.e. a model that would combine the predictions from both single-modality networks and always choose the right one. The oracle corresponds to the theoretical maximal accuracy boost that we could expect from the **FUSION** model. On Tobacco3482, the oracle corresponds to a 7.6% absolute improvement (9% relative). In our case, the **FUSION** model improves the best single-source baseline by an absolute 3.3% (4% relative), which is significant although still leaves the door open to further improvements. More importantly, the gains are consistent on all classes of interest, almost never underperforming one of the two base networks on any class. This confirm the proposed approach as the two sources, image and text, give complementary information to classify a document.

5.2 Processing time

Although some applications of document image recognition can be performed offline, most of the time users upload a document and expect near real-time feedback. User experience engineering [37] indicates than less than 1s is the maximum latency the user can suffer before the interface feels sluggish, and 10s is the maximum delay before they start loosing their attention. On the RVL-CDIP dataset, Tesseract processes a document image in $\simeq 910\text{ms}$ in average on an Intel Core i7-8550U CPU using 4 threads,

Table 2: Overall accuracy on the RVL-CDIP dataset.

Model	IMAGE	TEXT	FUSION	CNNs [1]	VGG-16 [10]	AlexNet+SPP [9]
OA	89.1%	74.6%	90.6%	89.8%	90.97%	90.94%

OA = Overall Accuray.

Table 3: Overall accuracy and F_1 scores on the Tobacco3482 datasets.

Model	OA	F_1	Adv.	Email	Form	Letter	Memo	News	Notes	Report	Res.	Sci.
CNNs [1]	79.9	–						–				
TEXT	73.8	0.71	0.60	0.96	0.76	0.71	0.79	0.67	0.62	0.43	**0.97**	0.57
IMAGE	84.5	0.82	**0.94**	0.96	0.85	0.83	0.90	0.89	0.83	0.61	0.80	0.62
FUSION	**87.8**	**0.86**	0.93	**0.98**	**0.88**	**0.86**	**0.90**	**0.90**	**0.85**	**0.71**	0.96	**0.68**
Oracle	92.1	0.91	0.94	0.99	0.94	0.92	0.93	0.93	0.89	0.81	0.97	0.79

Adv. = Advertisement, Res. = Resume, Sci. = Scientific.

including loading the image from disk. This means that every additionnal latency induced by the network inference time is critical since it will negatively affect the user experience.

On the same CPU, the full inference using the **FUSION** model takes $\simeq$360ms including loading, resizing and normalizing the image. The complete process including Tesseract OCR therefore takes less than $\simeq$1300ms which is acceptable in a system requiring user input. Of those, 130ms are spent in the 1D CNN (including reading the file and performing FastText inference) and 230ms in MobileNetV2 (including image preprocessing). The overhead added by the final fusion layer is negligible. We stress that this is using a standard TensorFlow without any CPU-specific compilation flags, which could speed up the inference further. On a NVIDIA Titan X GPU, the **FUSION** network runs in 110ms (50ms for **TEXT**, 60ms for MobileNetV2), which brings the total just above the 1s recommendation. In our case, using compute-efficient architectures allow us to avoid running on an expensive and power-hungry GPU.

As a comparison basis, other architecture choices that we dismissed earlier would have resulted in poorer performance and the network would not be usable in a near real-time user application. For example, the Xception network [38] takes 630ms to run during inference with the same parameters and hardware. For the text model, an LSTM-based RNN with a similar depth takes many seconds to run.

Note that, although this does not reduced the perceived delay for one user, the global throughput of the system can be improved by batching the images. Two Tesseract processes can leverage the full eight cores from an Intel Core i7-8550U CPU. In this setting, processing an image takes $\simeq$660ms in average. Thanks to the batch efficiency of neural networks, the average processing time becomes $\leq$750ms on GPU and $\leq$1000ms on CPU. This is particularly helpful when users have several documents to upload that can be processed concurrently.

5.3 Limitations

One of the main limitation of this work stems from the public document image datasets available. Indeed, in a real-world application, document images can be grayscale, RGB, scanned images and photographs with various rotations, brightness, contrast and hue values. The Tobacco documents are all oriented in the right way, which makes it easier for Tesseract to perform OCR. Moreover, documents have been scanned by professionals who tried to maximize their legibility while user-generated often presents poor quality.

While it was not required here, data augmentation is definitely required for practical applications to encompass the large variety of environmental conditions in which documents are digitized. This is especially true for rotations, since it is often not possible to ensure that users will capture the document with the right orientation and Tesseract does not always correctly detects it. For industrial-grade applications dealing with user-generated content, such a data augmentation is necessary to alleviate overfitting and reduce the gap between train and actual data. Preprocessing page segmentation and layout analysis tools, such as dhSegment [13] can also bring significant improvements by renormalizing image orientation and cropping the document before sending it to the classifier.

Moreover, as we have seen, the post-OCR word embeddings include lots of noisy or completely wrong words that generate OOV errors. In practical applications, we found beneficial to perform a semantic tokenization and named entity recognition using SpaCy. This allows us to perform a partial spellchecking, e.g. using symspell [7] to correct words that have been misread by Tesseract, without affecting proper nouns or domain-specific abbreviations and codes. If this can deal frequent mispellings of words, it might also suppress out-of-vocabulary words such as alphanumeric codes. Therefore, learning domain specific, character-based or robust-to-OCR embeddings [39] is an interesting lead for future research, as the current interest in the ICDAR2019

[7] https://github.com/wolfgarbe/SymSpell

competition on Post-OCR Text Correction shows[8].

6 Conclusion

In this work, we tackled the problem of document classification using both image and text contents. Based only on an image of a digitized document, we try to perform a fine-grained classification using visual and textual features. To do so, we first used Tesseract OCR to extract the text from the image. We then compute character-based word embeddings using FastText on the noisy Tesseract output and generate a document embedding which represents our text features. Their counterpart visual features are learned using MobileNetv2, a standard CNN from the state of the art. Using those pragmatic approaches, we introduce an end-to-end learnable multimodal deep network that jointly learns text and image features and perform the final classification based on a fused heterogeneous representation of the document. We validated our approach on the Tobacco3482 and RVL-CDIP datasets showing consistent gains both on small and large datasets. This shows that there is a significant interest into hybrid image/text approach even when clean text is not available for document image classification and we aim to further investigate this topic in the future.

References

[1] A. W. Harley *et al.*, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," in *ICDAR*, Aug. 2015.

[2] K. Y. Wong *et al.*, "Document Analysis System," *IBM J. Res. Dev.*, Nov. 1982.

[3] A. Kay, "Tesseract: An Open-Source Optical Character Recognition Engine," *Linux J.*, July 2007.

[4] D. X. Le *et al.*, "Classification of binary document images into textual or nontextual data blocks using neural network models," *Mach. Vis. Appl.*, Sept. 1995.

[5] S. Imade *et al.*, "Segmentation and classification for mixed text/image documents using neural network," in *ICDAR*, Oct. 1993.

[6] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, Nov. 1998.

[7] N. Chen and D. Blostein, "A survey of document image classification," *Int. J. Doc. Anal. Recogn.*, June 2007.

[8] J. Kumar *et al.*, "Structural similarity for document image classification and retrieval," *Pattern Recognit. Lett.*, 2014.

[9] C. Tensmeyer and T. Martinez, "Analysis of CNNs for Document Image Classification," in *ICDAR*, Nov. 2017.

[10] M. Z. Afzal *et al.*, "Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification," in *ICDAR*, Nov. 2017.

[11] A. Das *et al.*, "Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks," in *ICPR*, Aug. 2018.

[12] R. Sicre *et al.*, "Identity Documents Classification as an Image Classification Problem," in *ICIAP*, Sept. 2017.

[13] S. Ares Oliveira *et al.*, "dhSegment : A generic deep-learning approach for document segmentation," in *ICFHR*, Aug. 2018.

[14] H. Borko and M. Bernick, "Automatic Document Classification," *J. ACM*, Apr. 1963.

[15] L. M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *J. Mach. Learn. Res*, Dec. 2001.

[16] T. N. Rubin *et al.*, "Statistical topic models for multi-label document classification," *Mach. Learn.*, July 2012.

[17] T. Mikolov *et al.*, "Efficient Estimation of Word Representations in Vector Space," in *ICLR*, Jan. 2013.

[18] M. Peters *et al.*, "Deep Contextualized Word Representations," in *NAACL*, June 2018.

[19] Z. Yang *et al.*, "Hierarchical Attention Networks for Document Classification," in *NAACL*, 2016.

[20] L. Noce *et al.*, "Embedded Textual Content for Document Image Classification with CNNs," in *ACM DocEng*, 2016.

[21] X. Yang *et al.*, "Learning to Extract Semantic Structure from Documents Using Multimodal FCNNs," in *CVPR*, July 2017.

[22] O. Augereau *et al.*, "Improving Classification of an Industrial Document Image Database by Combining Visual and Textual Features," in *IAPR Workshop*, Apr. 2014.

[23] M. Sandler *et al.*, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *CVPR*, June 2018.

[24] A. S. Razavian *et al.*, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *CVPRW*, June 2014.

[25] K. He *et al.*, "Deep Residual Learning for Image Recognition," in *CVPR*, June 2016.

[26] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man. Cybern.*, Jan. 1979.

[27] J. Pennington *et al.*, "Glove: Global Vectors for Word Representation," in *EMNLP*, Oct. 2014.

[28] Y. Pinter *et al.*, "Mimicking Word Embeddings using Subword RNNs," in *EMNLP*, Sept. 2017.

[29] P. Bojanowski *et al.*, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, 2017.

[30] A. Joulin *et al.*, "Bag of Tricks for Efficient Text Classification," in *EACL*, 2017.

[31] A. Patel *et al.*, "Magnitude: A Fast, Efficient Universal Vector Embedding Utility Package," in *EMNLP*, Nov. 2018.

[32] S. Arora *et al.*, "A Simple but Tough-to-Beat Baseline for Sentence Embeddings," in *ICLR*, Nov. 2016.

[33] M. Honnibal and Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[34] A. Eitel *et al.*, "Multimodal deep learning for robust RGB-D object recognition," in *IROS*, Sept. 2015.

[35] K. He *et al.*, "Delving Deep into Rectifiers," in *ICCV*, 2015.

[36] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *EMNLP*, Oct. 2014.

[37] J. Nielsen, *Usability Engineering*. 1993.

[38] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *CVPR*, July 2017.

[39] V. Malykh *et al.*, "Robust Word Vectors: Context-Informed Embeddings for Noisy Texts," in *EMNLP W-NUT*, 2018.

[8] https://sites.google.com/view/icdar2019-postcorrectionocr

Contrôle qualité en radiothérapie externe basé sur une imagerie portale via des réseaux de neurones

Frédéric CHATRIE[1,2,3] Marie-Véronique LE LANN[1,2] Xavier FRANCERIES[2,3]

[1] LAAS-CNRS, F-31000, Toulouse, France
[2] Université de Toulouse, INSA, UPS, F-31000 Toulouse, France
[3] Inserm, UMR1037 CRCT, F-31000 Toulouse, France

fchatrie@laas.fr

Résumé

Les réseaux de neurones artificiels appliqués à la radiothérapie externe peuvent avoir un grand intérêt, notamment pour le contrôle qualité des traitements. Dans ce travail, une nouvelle approche a été investiguée basée sur des réseaux de neurones qui ont permis de reconstruire une distribution de dose absorbée 2D à partir de l'imageur portal dont le signal est récupéré durant la séance de prétraitement. Le modèle utilisé est un réseau de neurones supervisé multi-couches de type « feed-forward ». Il a permis d'obtenir un très bon critère d'évaluation clinique appelé gamma-index montrant sa capacité à reconstruire une distribution de dose absorbée 2D.

Mots Clef

Réseaux de neurones artificiels, radiothérapie externe, imageur portal, contrôle qualité.

Abstract

Artificial neural networks applied to external beam radiation therapy can be of great interest, especially for treatment quality assurance. In this work, a novel approach has been investigated, based on artificial neural networks, which allowed the 2D absorbed dose reconstruction from electronic portal imaging device whose the signal is retrieved during the pre-treatment session. The used model is a supervised multi-layers feed-forward neural network. It was given a good agreement assessed by the clinical gamma criterion highlighting the neural networks capability to reconstruct a 2D absorbed dose distribution.

Keywords

Artificial neural network, external beam radiation therapy, electronic portal imaging device, quality assurance.

1 Introduction

Les réseaux de neurones artificiels (RNAs) ont reçu une attention particulière ces dernières années de par la diversité de leurs champs d'application. Cet essor est également dû à la récente explosion technologique entrainant d'importantes capacités calculatoires via l'utilisation des GPUs et TPUs notamment.

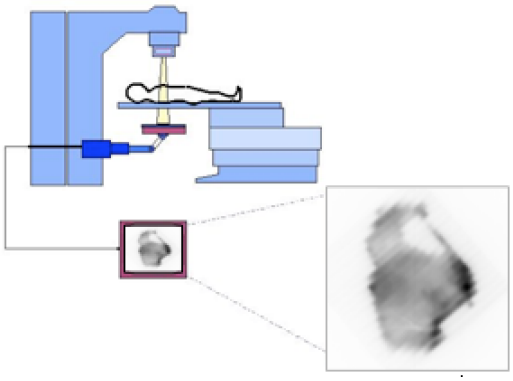


FIGURE 1 – Illustration de l'accélérateur linéaire (en bleu) avec l'EPID (en violet) durant le traitement d'un patient.

Le champ d'application de cette étude concerne un type de traitement contre le cancer : la radiothérapie externe. Elle consiste, par l'intermédiaire d'une source de rayons X placée à l'extérieur du patient, à concentrer un maximum d'énergie à l'endroit où sont situées les cellules cancéreuses, tout en minimisant l'irradiation des cellules saines ou des organes à risque situés à proximité. Pour calculer la distribution d'énergie déposée, appelée « dose absorbée » un système de planification de traitement (TPS) est utilisé. Il va permettre via des algorithmes d'optimisation, d'obtenir un plan de traitement spécifique qui correspondra au mieux à chaque patient. Cependant, l'objectif des entreprises développant les TPS est toujours d'améliorer la qualité des plans de traitement, ce qui entraine une plus grande complexité calculatoire, perceptible avec la radiothérapie conformationnelle à modulation d'intensité (RCMI) par exemple. Cette complexité implique des procédures de vérification pré et durant le traitement (in-vivo) afin de contrôler la précision de la distribution de

dose délivrée et ainsi pouvoir détecter, cliniquement, des anomalies conséquentes.
L'utilisation de l'imageur portal EPID (« Electronic Portal imaging Device »), qui est un détecteur plan de rayon X MV, monté directement sur l'accélérateur de particules (FIGURE 1), a été choisie pour estimer la distribution de dose absorbée [1]. Ce détecteur récupère un signal quantitatif contenant une information sur l'irradiation après la traversée du patient.
Plusieurs approches ont déjà été étudiées à des fins dosimétriques en utilisant l'EPID. La première d'entre elles, concerne les méthodes analytiques, qui consistent, à partir de kernels de calculs, de produits de convolution et d'une calibration de l'EPID, de reconstruire la distribution de dose absorbée [2, 3]. La deuxième concerne les méthodes Monte-Carlo qui se basent sur la simulation de toutes les interactions des particules selon un modèle physique bien établi [4, 5]. Ces approches présentent quelques inconvénients, pour la dernière présentée par exemple, le temps de calcul peut être significatif tandis que pour la première, plusieurs approximations et le besoin de validations routinières sont nécessaires. De plus, toutes ces approches sont dépendantes de la modélisation de chaque accélérateur linéaire, demandant au préalable des calibrations. Dans le cadre de ce travail, une nouvelle approche est proposée, celle de l'apprentissage automatique par l'utilisation des réseaux de neurones. Elle a permis de faire abstraction des complexités physiques, de les modéliser autrement et de manière plus efficace. Également, via des extensions facilement implémentables, il a été possible d'étendre le modèle vers différents accélérateurs linéaires.

2 Matériels et méthodes

2.1 Les réseaux de neurones artificiels

Les neurones artificiels ont été inspirés par le fonctionnement des neurones biologiques. Naturellement, ils restent une abstraction mathématique simplifiée, loin de la complexité du fonctionnement du neurone réel. Les RNAs sont composés de neurones artificiels organisés en plusieurs couches avec une arborescence qui diffère selon l'architecture utilisée. Plusieurs types d'architecture existent déjà et nombreuses sont celles qui restent à découvrir. Tout comme les humains, les RNAs fonctionnent en deux phases - l'apprentissage et la reconnaissance. L'apprentissage supervisé contient pour chaque donnée d'entrée, une donnée de sortie associée que l'on appelle souvent étiquette ou cible. L'étiquette donne une correspondance que l'algorithme va considérer comme étant la valeur exacte. Par analogie avec les humains, on parle souvent d'apprentissage avec l'expertise d'un professeur. Dans cette catégorie d'apprentissage, les étiquettes peuvent prendre différentes formes, soit des booléens dans le cadre de la classification, soit des valeurs réelles, pour de la régression. Le type de modèle considéré dans cette étude est la regression, il permet de modéliser des fonctions mathématiques complexes qui peuvent être non linéaires. Dans une organisation de type perceptron multi-couche, la valeur des poids est calculée par un algorithme d'identification de paramètres telle que la méthode de descente de gradient ou une méthode de Gauss-Newton. Il permet de minimiser, itérativement, l'erreur entre la valeur prédite par les réseaux de neurones et la valeur cible (étiquette) sur un maximum d'échantillons de données transmises pour l'apprentissage. L'erreur ici, est définie par une fonction coût qui peut prendre différentes formes selon la catégorie d'apprentissage supervisé utilisée (classification ou régression). Un des problèmes rencontrés avec les algorithmes d'optimisation, est l'existence de minima locaux pour les modèles non convexes tels que les RNAs. Il est alors nécessaire de trouver un minimum local contenant une faible valeur de la fonction coût afin d'approcher au mieux le minimum global [6]. Une étude a cependant montré que de récupérer un minimum global sur un modèle de taille conséquente pouvait montrer un très bon taux d'apprentissage mais entraînait un modèle non représentatif du système [7].

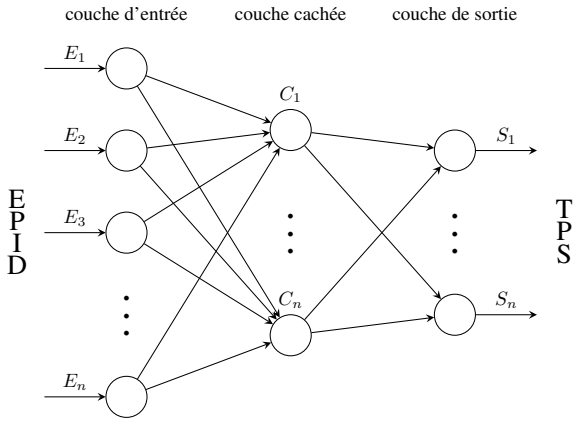


FIGURE 2 – Architecture du réseau de neurones utilisée.

La phase de reconnaissance consiste à fournir une nouvelle donnée d'entrée (ne faisant pas partie des données d'apprentissage) à l'algorithme qui lui, va donner le résultat prédit en utilisant la structure et les poids identifiés lors de l'étape précédente. Dans notre étude, l'architecture de RNAs utilisée est un réseau de neurones multi-couches « feed-forward » non-récurrent dont l'architecture est montré Figure 2. L'algorithme d'optimisation permettant de minimiser l'erreur est celui de Levenberg-Marquard (couplant les méthodes de Gauss-Newton et de gradient), la fonction coût associée est l'erreur quadratique moyenne.

2.2 Données utilisées pour les RNAs

Les données requises pour développer l'application RNA proposée ont été obtenues par différents accélérateurs linéaires (ELEKTA® et VARIAN®) et pour différentes techniques de traitement qui sont la radiothérapie conformationnelle (RTC) et celle à modulation d'intensité (RCMI). Autrement dit, un collimateur multi-lames est placé dans la tête de l'accélérateur et permet de se confor-

mer précisément à la forme de la tumeur. Dans le cadre de la RTC, les lames restent statiques au cours de l'irradiation (irradiation fixe et quasi-homogène en sortie de la tête de l'accélérateur), contrairement à la RCMI où les lames sont dynamiques au cours de l'irradiation pour un champ donné (l'irradiation est modulée dans le temps et l'espace). D'une part, les données d'entrées utilisées pour l'apprentissage et la reconnaissance correspondent exclusivement à des données qui proviennent de l'EPID. Aucune information ne provient du plan de traitement en tant que données d'entrées pour ne pas influencer la phase d'apprentissage. Le choix d'utiliser des données quasi-brutes a été fait. En effet, seules les corrections dites « flood field » et « dark field » ont été apportées. La correction « dark field » correspond à une moyenne faite du signal sans irradiation sur l'imageur portal, ce qui permet de retirer le bruit électronique notamment. La correction « flood field » correspond à une irradiation complète sur l'aire de détection qui supprime les différences de sensibilité du détecteur. D'autre part, les données de sorties utilisées pour l'apprentissage supervisé correspondent aux distributions de dose absorbée planifiées provenant du TPS, considérées comme exactes. Les RNAs supervisés ont besoin, architecturellement, d'autant d'échantillons de données d'entrées que d'étiquettes associées. Pour cela, un échantillonnage sur la distribution de dose absorbée TPS a été appliqué. En effet, le faisceau d'irradiation étant conique, la projection du champ à 100cm de la tête de l'accélérateur linéaire (position de la distribution de dose absorbée TPS à l'isocentre du traitement) sera spatialement rétrécie par rapport à une projection du champ à 150cm (position de l'EPID).

Une région d'intérêt, après application d'un seuil de 10% sur la distribution de dose absorbée TPS a été considérée. Le masque créé a été appliqué pour les deux ensembles de données (EPID et TPS) afin de garder et d'évaluer uniquement les informations pertinentes du traitement.

Pour permettre la mise en place d'un modèle correct du système, l'utilisation de chaque pixel a été privilégiée au lieu des images EPID et TPS comme échantillon de données. En effet, chaque pixel d'une image EPID correspond à un signal en niveau de gris qui est physiquement lié à chaque valeur de pixel de dose absorbée du TPS. Outre la relation physique directe qui a pu être établie entre les pixels de chaque image EPID et les distributions de dose absorbée TPS, les informations provenant des pixels voisins ont également été prises en compte. Cela a permis de modéliser intrinsèquement le rayonnement diffusé du patient (ici, équivalent à une cuve d'eau - milieu homogène) et d'autres composantes pendant le traitement. Toutes ces informations, ainsi que la localisation spatiale de ces pixels, ont été définies comme données d'entrées. L'intégralité des données d'apprentissage (EPID et TPS) ont été considérées et mises à l'échelle afin de rendre le modèle plus pertinent.

3 Résultats et discussions

Les résultats énoncés ici concernent l'accélérateur de particules de marque VARIAN®. La phase d'apprentissage a été effectuée avec 8 et 11 ensembles de données d'entrées/sorties pour la RTC et la RCMI respectivement. La phase de reconnaissance a été faite avec 2 ensembles de données. L'intégralité des images disposaient de 384x512 pixels avant l'application du masque (seuil à 10%) comme décrit dans la section précédente. Étant limité par le nombre de données récupérées, l'ensemble des données ont servi de manière croisée soit pour la phase d'apprentissage soit pour la phase de reconnaissance, ce qui nous a permis de faire 5 et 6 sets d'apprentissage, respectivement. Procéder de telle sorte a permis de pouvoir valider un plus grand nombre d'apprentissage.

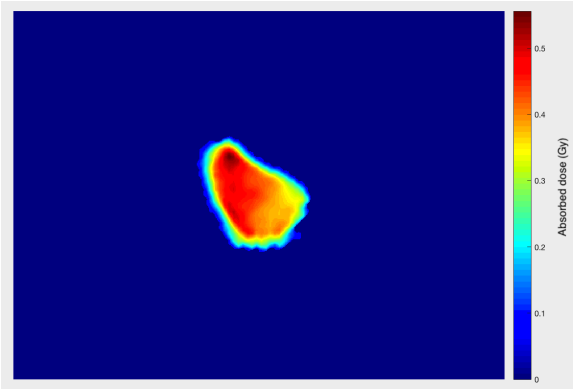


FIGURE 3 – Distribution de dose absorbée planifiée par le TPS.

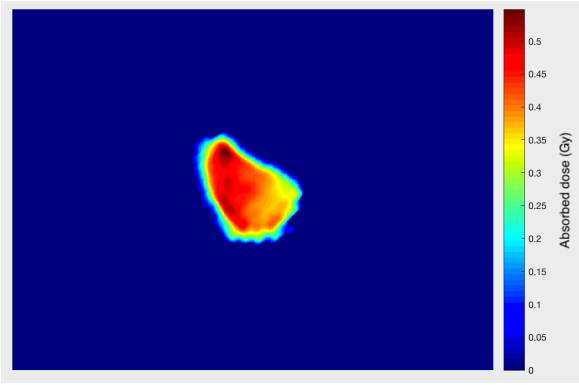


FIGURE 4 – Distribution de dose absorbée prédite par les RNAs.

Les FIG. 2 et FIG.3 montrent respectivement la distribution de dose absorbée planifiée par le TPS (résultat attendu) et la distribution de dose absorbée prédite par les RNAs durant la phase de reconnaissance, respectivement, pour un même champ d'irradiation en RCMI. Afin d'évaluer la qualité de l'apprentissage effectué, un critère communément utilisé

en clinique a été choisi. Ce critère est appelé gamma-index et peut être utilisé de différentes manières. Il donne en pourcentage, le nombre de pixels qui respectent l'objectif donné. Dans l'étude, l'objectif était d'avoir au maximum 3% de différence de niveau de dose absorbée ou une différence de localisation spatiale de 3mm. Le gamma-index global obtenu avec ces objectifs a été supérieur à 98%, sur l'intégralité des données testées, montrant la capacité des RNAs à reconstruire une distribution de dose absorbée. Sur l'ensemble des données récupérées, aucun cas avec un grand désaccord n'a été observé. De plus, des résultats similaires (non détaillés ici) ont été obtenus pour un accélérateur de particules de marque ELEKTA®, confirmant que les algorithmes peuvent être utilisés pour différentes machines.

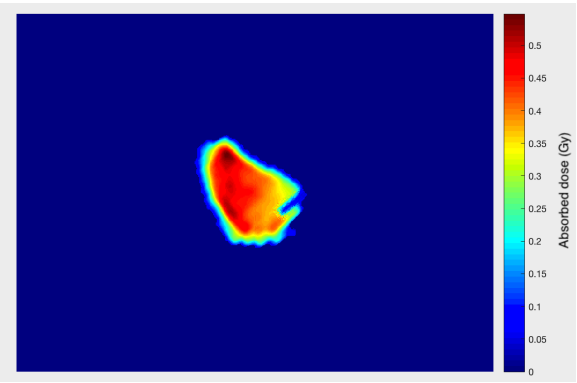


FIGURE 5 – Distribution de dose absorbée prédite par les RNAs avec mauvaise position d'une lame.

La FIG.4 montre la capacité des algorithmes développés à détecter un problème sur la machine. Ici, une mauvaise position de lame restée à sa position initiale a été simulée sur l'image EPID. Le plan de traitement considéré est le même que celui étudié précédemment et affiché en FIG.2. Le gamma-index global obtenu était de 93.8% montrant la détection de l'anomalie. La différence de 5% correspond à l'influence géométrique de la mauvaise position de lame sur le nombre de pixels considérés dans cette image. Il est intéressant de signaler qu'aucune mauvaise position de lame n'était présente dans l'ensemble des données d'apprentissage. Malgré cela, la phase de reconnaissance montre une reconstruction de la dose absorbée cohérente avec la réalité.

4 Conclusion

Les résultats obtenus ont montré que les RNAs pouvaient être choisis afin de reconstruire des distributions de dose absorbée délivrées lors du pré-traitement, à partir de l'imageur portal EPID, et cela indépendamment de l'accélérateur linéaire et du type de traitement (RTC ou RCMI) utilisé. L'architecture de RNAs « feed-forward » profond a été choisie et correctement paramétrée afin d'obtenir un modèle pertinent pour cette application. Le critère d'évaluation clinique gamma-index confirme la pertinence du modèle créé. Les RNAs ont permis de créer un « pattern » généralisé à de nouvelles données pour cette application spécifique. Ces algorithmes ont été développés pour la vérification pré-traitement de la dose absorbée délivrée, il serait maintenant intéressant de développer la vérification durant le traitement de la dose absorbée délivrée (*in-vivo*) en temps-réel. Des paramètres supplémentaires demandent à être considérés pour étendre les algorithmes vers la dosimétrie *in-vivo*. En effet, modéliser l'influence de la géométrie du patient avec les méthodes RNAs peut être fastidieux. La sélection des données les plus pertinentes pour un apprentissage correct sera nécessaire.

Références

[1] Wouter van Elmpt, Leah McDermott, Sebastiaan Nijsten, Markus Wendling, Philippe Lambin, and Ben Mijnheer. A literature review of electronic portal imaging for radiotherapy dosimetry. *Radiotherapy and Oncology*, 88(3) :289–309, 07 2008.

[2] Markus Wendling, Rob Louwe, Leah McDermott, Jan-Jakob Sonke, Marcel van Herk, and Ben Mijnheer. Accurate two-dimensional imrt verification using a back-projection epid dosimetry method. *Medical physics*, 33 :259–273, 03 2006.

[3] Markus Wendling, Leah McDermott, Anton Mans, Jan-Jakob Sonke, Marcel van Herk, and Ben Mijnheer. A simple backprojection algorithm for 3d in vivo epid dosimetry of imrt treatments. *Medical physics*, 36 :3310–3321, 08 2009.

[4] Wouter van Elmpt, Sebastiaan Nijsten, Robert F. H. Schiffeleers, Andre Dekker, Ben Mijnheer, Philippe Lambin, and Andre Minken. A monte carlo based three-dimensional dose reconstruction method derived from portal dose images. *Medical physics*, 33 :2426–2434, 08 2006.

[5] Geneviève Jarry and Frank Verhaegen. Patient-specific dosimetry of conventional and intensity modulated radiation therapy using a novel full monte carlo phase space reconstruction method from electronic portal images. *Physics in Medicine and Biology*, 52(8) :2277–2299, 04 2007.

[6] Ian Goodfellow, Oriol Vinyals, and Andrew Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*, 2015.

[7] Anna Choromanska, MIkael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 192–204, San Diego, California, USA, 09–12 May 2015. PMLR.

Recherche d'information pour l'aide à la résolution d'incident: De l'expérimentation à l'industrialisation d'une solution pour les métiers

Adèle Désoyer [1] Simon Devaradja[1]

[1]EDF, *Direction des Services Informatique et Télécoms*, 32 Avenue Pablo Picasso, 92000 NANTERRE

adele.desoyer@edf.fr; simon.devaradja@edf.fr

Résumé

Dans cet article, nous présentons un cas d'usage de recherche d'information appliquée à l'aide à la résolution d'incident informatique. Après avoir décrit ce cas d'usage, ses données et un bref état de l'art, nous exposerons les expérimentations menées pour y répondre, ainsi que les résultats obtenus. Avant de conclure, nous expliquerons comment la solution est passée à l'échelle, depuis un environnement expérimental, à un environnement de production, destiné au métier.

Mots Clef

Recherche d'information, mise à l'échelle, suivi de performance, industrialisation.

Abstract

In this paper, we present a use case of Information Retrieval, for assistance in resolving informatics incident. First, we explain more precisely that use case, its data and a short state-of-the-art. Then, we develop our experimentations and its results. Finally, before to conclude, we show how the solution has been scaled up in a production environment.

Keywords

Information Retrieval, scalability, performance monitoring, product industrialisation

1 Introduction

La croissance incessante du volume de données, notamment textuelles, fait aujourd'hui encore de la recherche d'information (RI), émergeant à la fin des années 40 [1,2], une activité au cœur des préoccupations des individus et des industriels. Née du besoin de retrouver efficacement des données non structurées dans une collection documentaire, en réponse à un besoin d'information spécifique, la RI et ses diverses applications permettent de faire émerger du sens de grands volumes de données textuelles, et pour EDF notamment, de valoriser un patrimoine riche de décennies de création et d'archivage d'information.

Discipline issue de la recherche documentaire, la RI se donne pour objectif de trouver des documents, régulièrement textuels donc non structurés, satisfaisant un besoin d'information donné, parmi un ensemble de documents composant une collection.

C'est précisément la problématique que nous développerons dans cet article, au travers d'un cas d'usage métier que nous avons approfondi : les exploitants des centrales hydrauliques d'EDF disposent d'une hotline à contacter en cas d'incident lié au système informatique. Le premier niveau de cette hotline est composé d'opérateurs qui ne sont pas experts du fonctionnement du SI ni des centrales. Aussi, pour les guider dans la résolution des problèmes remontés, les équipes du CIH[1] mettent régulièrement à jour une collection d'articles de résolution, dont chacun fourni les informations nécessaires à résoudre un problème donné. Depuis sa création, chacun des nouveaux événements survenus s'est vu associer un article de résolution de cette collection, via un outil dit de *ticketing*, (*cf.* **Figure 1**) recensant l'ensemble des événements. Nous disposons ainsi d'un ensemble de référence, construit manuellement par les opérateurs, auquel nous pourrons comparer les résultats automatiques pour mesurer les performances de l'IA développée.

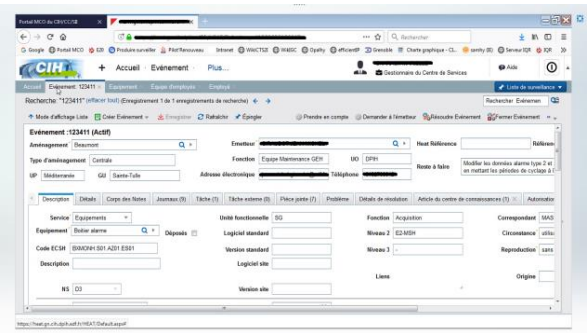


Figure 1. Outil de gestion des tickets du CIH

2 Travaux antérieurs

Un système de recherche d'information (SRI) se décompose globalement en deux grandes étapes : une première transforme les textes bruts en représentations intelligibles par une machine ; une seconde mesure la similarité entre ces représentations pour proposer à un utilisateur des réponses proches de sa requête.

2.1 Représentation des textes

1. *Centre d'Ingénierie Hydraulique*

Pour qu'une machine manipule un texte, il faut le transformer. Différentes méthodes, linguistiques et mathématiques peuvent être utilisées à cet effet.

Linguistiquement, il existe tout un ensemble de techniques, régulièrement spécifiques à la langue étudiée, pour découper le texte en phrases ou en mots, dit *tokenisation* [3]. Il existe également un ensemble de techniques pour normaliser ces mots : la racinisation [4,5] qui permet de ramener à une même forme différentes variations d'une même racine (*e.g. migration*, *émigrés*, *migrants*, *migratoire* partagent la même racine *migr-*). La lemmatisation, qui va souvent de pair avec une annotation en partie du discours (ou *part of speech* en anglais) [6], est une alternative à la racinisation, moins radicale. Les termes de même nature sont ramenés à une même forme non fléchie (*e.g.* les formes conjuguées *migrais* et *migrez* sont normalisées en la forme infinitive *migrer* ; de même, les adjectifs *migrants* et *migrante* sont normalisés en *migrant*). Il est également possible de filtrer ces mots par rapport à une liste de mots-vides, c'est-à-dire de termes qui participent plus à la syntaxe d'un texte qu'à sa sémantique (typiquement, les adverbes, déterminants, pronoms, …).

Mathématiquement, il s'agit de transformer ces mots en vecteurs de caractéristiques.
La méthode la plus ancienne produit une représentation de surface : c'est celle du *sac de mots* (*bag-of-words*). Cette méthode consiste à considérer tous les termes du vocabulaire de la collection documentaire comme dimension d'un espace vectoriel. Un texte est ensuite projeté dans l'espace en fonction des termes qu'il contient, et du poids de ceux-ci. Les fonctions de pondérations classiques sont binaire, fréquentielle [8], *TF*IDF* [9,10] ou *BM-25* [11].
Bien qu'ayant fait ses preuves, cette méthode de vectorisation ne considère pas la sémantique des termes. C'est pourquoi des méthodes de représentation plus récentes, dite profondes, se sont développées pour pallier cette lacune : on parle de plongements lexicaux ou *word-embeddings* [12]. Le principe est de représenter les mots des documents (et non plus les documents directement) dans un espace continu, dans le but de pouvoir les comparer et ainsi de saisir d'éventuels liens sémantiques. Il existe deux types de modèles (CBOW[2] et Skip-gram [12]), et différentes implémentations, toutes disponibles en open source : *Word2Vec* [13], *GloVe* [14] ou *FastText*, librairie d'apprentissage[3].

2.2 Similarité textuelle

Une fois les textes représentés dans l'espace, on souhaite pouvoir les comparer entre eux pour en mesurer le degré de similarité.

Différentes mesures de distance entre vecteurs existent, telles que la distance euclidienne, la distance cosinus [15] (*cf.* **Figure 2**) ou la *Latent Semantic Indexation* (*LSI*) [16]. Ces méthodes sont régulièrement utilisées sur des vecteurs de type *bag-of-words*. Pour les vecteurs de type *word-embeddings*, sont régulièrement utilisées les méthodes *Word Moving Distance* (*WMD*) [17] ou *Smooth Inverse Frequency*[4].
Cette section se termine en évoquant les métriques classiques utilisées pour évaluer les SRI [2]. Etant donné, pour une requête, un ensemble de réponses attendues et un ensemble de réponses prédites, on obtient la matrice[5] suivante :

PRED	REF	Pertinent	Non Pertinent
Pertinent		TP	FP
Non pertinent		FN	TN

Il est alors possible de calculer les métriques suivantes :

- Précision = $\frac{TP}{TP+FP}$
- Rappel = $\frac{TP}{TP+FN}$
- F-mesure = $\frac{2*\mathrm{Pr}ecision*Rappel}{\mathrm{Pr}ecision+Rappel}$

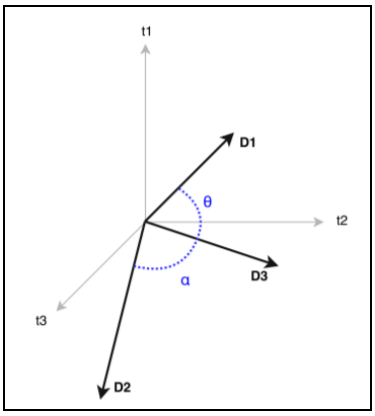


Figure 2 - Calcul du cosinus entre vecteurs documentaires (D1, D2, D3,) dans un espace de trois termes (t1, t2, t3)

3 Expérimentations

3.1 Dataset

On dispose d'un ensemble initial de 7873 paires événement/article, que l'on découpe en deux, de façon aléatoire :

- 6406 paires (80%) pour la phase d'expérimentation, qui servira à sélectionner le meilleur algorithme
- 1467 paires (20%) pour tester en fin de phase le meilleur algorithme sélectionné sur de nouvelles données jamais observées, et s'assurer de sa capacité de généralisation.

2 *Continuous Bag-Of-Words*
3 https://fasttext.cc/
4 https://github.com/PrincetonML/SIF
5 *TP* = True Positive ; *TN* = True Negative ; *FP* = False Positive; *FN* = False Negative

Par ailleurs, au moment de l'expérimentation, la base d'articles interrogée compte 1078 articles.

Un événement est décrit textuellement par un sujet (8 termes en moyenne) et une description (de longueur très hétérogène, parfois vide, parfois plusieurs paragraphes). Les articles ressemblent aux événements dans leur format textuel, puisqu'ils sont décrits par un titre (6 termes en moyenne) et une procédure de résolution (de même que pour les événements, ce champ peut être vide ou très long).

3.2 Implémentation

Le souhait du métier est de se voir proposer l'article le plus susceptible de répondre à un événement soumis en requête au système. En termes de métriques, la précision importe donc peu, tandis que le rappel est essentiel, ainsi que l'ordonnancement des résultats proposés.
La métrique utilisée pour mesurer la performance des algorithmes testés est donc le rappel à *k* documents, noté ***R@k***, régulièrement utilisé dans les systèmes de recommandation, pour qui l'ordonnancement des résultats présentés est crucial [18]. Cette métrique permet d'observer la proportion de résultats pertinents retournés par les *k* meilleurs résultats, parmi *n*.

$$\text{R@k(V)} = \frac{\sum_{i=1}^{k} rel(v_i)}{\sum_{i=1}^{n} rel(v_i)}$$

Dans le présent cas d'usage, il n'existe qu'un seul résultat pertinent pour une requête donnée (un événement n'est associé qu'à un seul article : *n* vaut donc 1. Quant à *k*, on le fixe à 10, en accord avec le métier. On mesure donc la performance des algorithmes sur la métrique de ***R@10.***

Différentes configurations d'algorithmes sont testées, en vue de trouver le meilleur sur ces données et ce cas d'usage.
Les paramètres que l'on fait varier sont de différentes nature :

- **Objets comparés** : événement/article vs. événement/événement
- **Sections considérées** : sujet seul ; description seule ; sujet et description ensemble
- **Pré-traitement des données** : tokenisation ; normalisation ; filtre des mots-vides
- **Représentation vectorielle** : *bag-of-words* ; *word2vec*
- **Pondération des termes** : binaire, fréquentielle, *TF*IDF*
- **Mesure de similarité** : euclidienne ; cosinus ; WMD

En combinant ces différents paramètres entre eux, nous sommes parvenus à tester et comparer les résultats de 576 algorithmes différents, afin d'observer l'influence de chaque paramètre indépendamment des autres.

3.3 Résultats

Nous n'exposons pas ici les résultats de l'ensemble des algorithmes testés. Notons cependant que, sur l'ensemble des configurations testées, à paramètres égaux, la représentation *bag-of-words* est toujours meilleure que la *word2vec*; de la même façon, la mesure de similarité cosinus est toujours meilleure que la WMD. Ceci s'explique par le fait que le vocabulaire employé dans les demandes de résolution est très spécifique à l'événement décrit, et qu'il y a peu de variations sémantiques pour décrire un même problème. Il n'est donc pas nécessaire de tenter de rapprocher des termes dans l'espoir d'élargir la recherche d'article. Au contraire, on souhaite trouver celui répondant spécifiquement à l'événement, sans en proposer d'autres qui pourraient s'avérer proches sémantiquement.

Par ailleurs, il est plus pertinent de comparer les événements entre eux (en proposant comme résultat les articles associés aux événements les plus proches), que les événements aux articles. L'algorithme est enfin plus pertinent en considérant les contenus du sujet et de la description ensemble, plutôt que l'un ou l'autre seul.

La **Figure 3** présente les courbes de quatre de ces configurations, sur l'ensemble de données de test. La meilleure d'entre elle parvient à atteindre un R@10 de 0.87, satisfaisant les attentes du métier.

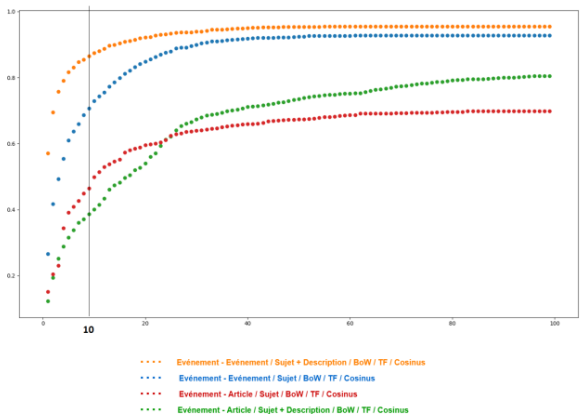


Figure 3 - Courbes de performance : Rappel de 1 à 100 documents

4 Passage à l'échelle

4.1 Enjeux

Il y a plusieurs enjeux du passage à l'échelle d'un tel type de solution. D'abord, la solution doit devenir accessible car à ce stade, il ne s'agit que de code, que le métier ne pourrait comprendre ni utiliser. Ensuite, la solution élaborée dans un environnement dit de développement que nous seuls utilisions dans un but exploratoire, doit maintenant fonctionner dans un environnement dit de production que cette fois plusieurs dizaines, voire une centaine de métiers vont utiliser. Cette considération a donc un impact fort sur l'infrastructure qui va héberger la solution et sur l'architecture du code. En effet, la solution doit pouvoir :

- **Répondre dans les délais imposés par le client** : il faut donc optimiser le code avec les notions de distributions de calculs sur CPUs, avant de réfléchir au dimensionnement de l'environnement (de combien de CPUs avons-nous besoin ?).
- **Permettre l'enregistrement de toute erreur** : il faut donc que l'équipe de maintenance de la solution puisse remonter à la source du problème très rapidement afin de débloquer la situation.
- **Être testée en continu** : la donnée de production est dynamique. La solution doit donc s'adapter à la donnée, et se mettre à jour afin de minimiser les erreurs. Si le contexte a tellement évolué que la solution devient obsolète et qu'il faut repasser par une phase de R&D, alors cette obsolescence doit être très rapidement identifiée par la mise en place d'un suivi de métriques.

4.2 Mise en place d'une application Web

Pour rendre notre algorithme accessible, il a été nécessaire de créer une IHM (*Interface Homme Machine*) qui permettrait à notre métier d'utiliser l'algorithme sans avoir à coder. D'un côté, nous avons dû mettre en place un *back office* qui rend accessible notre algorithme par micro service. De l'autre côté, un *front-end* a été développée puis connectée à ce *back* office pour offrir à l'utilisateur le parcours suivant :

- **Figure 4 :** l'utilisateur arrive sur l'application et doit renseigner le sujet et la description de l'événement qu'il traite
- **Figure 5 :** l'utilisateur accède d'une part à tous les articles recommandés par notre algorithme, et d'autre part aux événements similaires qui expliquent la remontée des articles. Cela lui permet de comprendre le fonctionnement de l'algorithme, et favorise ainsi son adhésion[6].

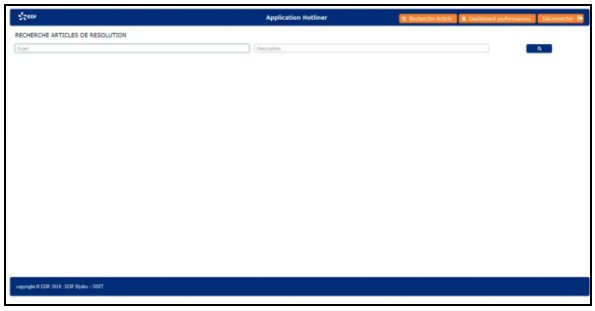


Figure 4 – Page de recherche dans l'application

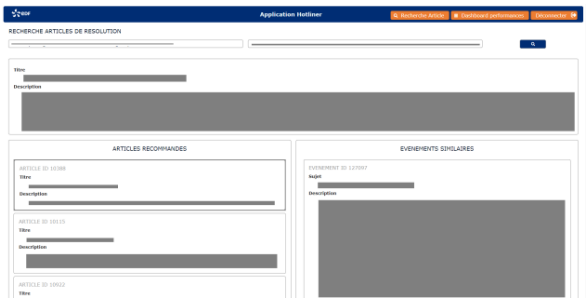


Figure 5 – Consultation des résultats

4.1 *Profiling* de code

Un parcours utilisateur, ce n'est pas seulement une question de visuel : le temps de réponse exigé par le métier est également un élément clé. Si la solution est trop lente, elle devient inefficace.

L'utilisation de l'outil ***cProfile***[7] en Python nous a permis de profiler (*i.e.* d'analyser) le code et de mettre en avant toutes les briques élémentaires longues en temps de réponse. Ces briques peuvent être optimisées, notamment si elle contient une boucle ***for***, grâce à une seconde librairie : ***multiprocessing***[8]. Elle permet de distribuer les calculs internes à la boucle à l'ensemble des cœurs disponibles sur l'environnement.

Dans notre cas, c'est le calcul de la similarité entre un nouvel événement et l'ensemble des événements de l'historique qui a dû être optimisé pour arriver au temps de réponse souhaité par le métier. La recherche d'une solution alternative en passant par des librairies qui font ce calcul optimisé de manière native pouvait nous faire éviter ce travail.

4.2 Enregistrement des erreurs

Un outil en production est un outil qui se doit d'être disponible tout le temps, sans quoi l'utilisateur risque de s'en désintéresser.

Pour couvrir l'ensemble des erreurs possibles, il est nécessaire de faire une revue globale du code avec un œil pessimiste sur tout objet Python manipulé : variable, code, classe, *etc*. C'est ainsi qu'en utilisant un *logger* (cf. la librairie native ***logging***[9]), chaque ligne de code qui peut être source d'erreur, à cause d'une donnée entrante erroné ou mal formée par exemple, va être enregistrée dans un fichier *log*. Ce protocole permet ainsi de contrôler tout point d'une solution.

Au moindre dysfonctionnement, il suffit d'aller consulter ce fichier log pour savoir quelle brique du code n'a pas fonctionné pour ensuite trouver la solution. L'identification du problème peut sinon être très coûteuse en temps.

6 Les éléments de réponses ont été ici grisés, dans un souci d'anonymisation.

7 https://docs.python.org/2/library/profile.html
8 https://docs.python.org/3.6/library/multiprocessing.html
9 https://docs.python.org/3.6/library/logging.html

4.3 Monitoring de la solution

Une fois la solution en production, les données analysées sont susceptibles de varier par rapport à l'ensemble de test initial, sur lequel elle avait atteint un *R@10* de 0.87.
Or, pour répondre à la demande des opérateurs, cette solution doit être capable de retrouver des articles pour de nouveaux événements en temps réel, événements jamais observés ni en phase de développement, ni en phase de test.

Nous avons identifié deux types de problèmes, susceptibles de détériorer les performances :

- Le dimensionnement de l'environnement n'est plus adapté au contexte de la solution : stockage insuffisant, capacité de calcul devenue faible à cause d'une plus forte volumétrie de donnée, problème de versions de technologies.
- La solution n'est plus efficace vis-à-vis de la problématique métier : les résultats ne sont plus à la hauteur de ce qu'ils étaient en condition d'expérimentation, car les données testées ont évolué.

Pour pallier la première difficulté, nous suivons quotidiennement quotidien le temps de mise à jour des des données et des modèles interrogés (*cf.* **Figure 6**, graphe de gauche) L'observation d'un temps anormalement long mettra en évidence un défaut de l'environnement. Nous évoquerons dans les perspectives d'autres métriques que nous envisageons d'ajouter à ce tableau de bord.

Pour la seconde difficulté soulevée, nous avons automatisé le lancement de l'algorithme, tous les matins, sur les événements de la veille. Disposant, en base de données, des articles ayant été réellement associés par les opérateurs à ces événements, il nous est possible de calculer le R@10 quotidiennement, et de suivre ainsi l'évolution des performances algorithmiques dans le temps. Le graphe de droite sur la **Figure 6** fait état de ces résultat : la courbe orange correspond au nombre d'événements analysés, par jour ; la courbe bleue au nombre d'événements pour lequel la solution a proposé le bon article dans les 10 premiers résultats. Plus les courbes sont proches l'une de l'autre, plus les performances sont bonnes.

Figure 6 – Monitoring de la solution

À l'inverse, si la courbe bleue s'éloigne de la courbe orange, il est nécessaire de s'interroger sur les raisons de cette baisse, et d'observer plus précisément les événements que l'algorithme n'a pas sur résoudre. Dans ce but, le tableau de bord offre également la possibilité d'observer les contenus textuels des événements manqués, ainsi que les contenus de ceux proposés en résultats, à des fins comparatives (**Figure 7**).

Figure 7 – Analyse qualitative des textes d'événements

5. Conclusions et perspectives

Dans cet article, nous avons présenté un cas d'usage de recherche d'information, appliqué à l'aide à la résolution d'incident informatique, dans un contexte industriel.

Nous avons proposé, dans une première partie, un bref état de l'art du domaine de la recherche d'information, présentant les différentes techniques de représentation documentaire, de vectorisation et de calcul de similarité textuelle. Ont ensuite été développées les expérimentations menées pour répondre à la problématique du métier, ainsi que les résultats obtenus.

Dans une seconde partie, nous avons discuté du passage de la solution d'un environnement expérimental à un environnement de production, et des difficultés soulevées à ce stade. Nous sommes notamment revenus sur la conception d'une IHM, pour interroger l'algorithme, et avons soulignés l'importance du temps de traitement, dans l'exécution en temps réel. Nous avons finalement présenté un tableau de bord permettant de suivre le comportement des composants de la solution, du point de vue machine (évolution du temps de calcul) et du point de vue utilisateurs (pertinence des résultats, analyse des erreurs, …).

En suivant les performances, grâce à ce tableau de bord, nous avons observé une baisse de performance par rapport à celle obtenue en phase d'expérimentation. Une nouvelle phase de R&D, testée sur ces événements nouveaux, est actuellement en cours. Elle vise à comprendre les différences entre ces événements, traités en temps réels, et ceux du précédent ensemble de test, ainsi qu'à trouver des pistes d'amélioration de la solution, tenant compte des spécificités de la production.

Par ailleurs, nous sommes également à l'étude sur de nouveaux indicateurs de performance, nous permettant d'améliorer d'avantage le suivi de la solution, afin de faciliter sa maintenance.

Nous envisageons, entre autres :

- De calculer des métriques sur des composants de plus bas niveaux (RAM, CPU, stockage)
- De configurer des environnements afin de pouvoir héberger la solution en mode *A/B testing*, afin de comparer deux algorithmes en temps réel, sur les mêmes données, sans écraser celle en production.
- D'ajouter au tableau de bord des indicateurs statistiques sur les termes et leur distribution dans les événements, afin d'affiner les analyses qualitatives
- D'automatiser la mise à jour de certaines briques de l'algortime, comme la liste des *stopwords*, via des seuils sur les pondérations de termes, par exemple.

La liste des postes envisagées n'est ici pas exhaustive, et nous continuons à réfléchir aux moyens d'optimiser les moyens d'analyse et de suivi d'une solution en production.

Au-delà du cas d'usage métier présenté, qui a servi de base à toutes ces réflexions, nous souhaiterions mutualiser l'ensemble des techniques et pratiques présentées ici, à l'ensemble des projets que nous sommes amenés à traiter. Dans l'idéal, nous souhaiterions pouvoir normaliser la phase de mise en production, en considérant un large éventail de possibilités. Bien que chaque projet aura toujours des exigences qui lui sont propres, la mise en place d'un socle commun devrait permettre d'accélérer et de fiabiliser cette phase complexe de mise en production des solutions IA.

Bibliographie

[1] Mooers, Calvin N. *Application of random codes to the gathering of statistical information*. Diss. Massachusetts Institute of Technology, 1948.

[2] Cleverdon, Cyril. "The Cranfield tests on index language devices." *Aslib proceedings*. Vol. 19. No. 6. MCB UP Ltd, 1967.

[3] Palmer, David D. "Tokenisation and sentence segmentation." *Handbook of natural language processing* (2000): 11-35.

[4] Porter, Martin F. "An algorithm for suffix stripping." *Program* 14.3 (1980): 130-137.

[5] Paternostre, Marjorie, et al. "Carry, un algorithme de désuffixation pour le français." *Rapport technique du projet Galilei* (2002).

[6] Schmid, Helmut. "Probabilistic Part-of-Speech Tagging Using Decision Trees, Intl." *Conference on New Methods in Language Processing. Manchester, UK*. 1994.

[7] Feldman, Ronen, and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.

[8] Luhn, Hans Peter. "The automatic creation of literature abstracts." *IBM Journal of research and development* 2.2 (1958): 159-165.

[9] Sparck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." *Journal of documentation* 28.1 (1972): 11-21.

[10] Jones, Karen Sparck. "Index term weighting." *Information storage and retrieval* 9.11 (1973): 619-633.

[11] Jones, K. Sparck, Steve Walker, and Stephen E. Robertson. "A probabilistic model of information retrieval: development and comparative experiments: Part 2." *Information processing & management* 36.6 (2000): 809-840.

[12] Ghannay, Sahar, et al. "Word embedding evaluation and combination." *LREC*. 2016.

[13] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*(2013).

[14] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[15] Salton, Gerard. "Automatic information organization and retrieval." (1968).

[16] Hofmann, Thomas. "Probabilistic latent semantic analysis." *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999.

[17] Kusner, Matt, et al. "From word embeddings to document distances." *International Conference on Machine Learning*. 2015.

[18] Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." *Recommender systems handbook*. Springer, Boston, MA, 2011. 1-35.

Link Prediction on Dynamic Attributed Knowledge Graphs for Maritime Situational Awareness

Jacques Everwyn[1,2] Abdel-Illah Mouaddib[1] Bruno Zanuttini[1] Sylvain Gatepaille[2] Stephan Brunessaux[2]

[1] Normandie Université, Université de Caen Normandie, GREYC
[2] Airbus Defence and Space

jacques.everwyn@unicaen.fr
Main research area: AI

July 2019

Résumé

Actuellement, les opérateurs de surveillance maritime parcourent à la main les quantités massives de données à leur disposition pour repérer les événements à surveiller. Les données maritimes viennent de sources variées et hétérogènes qui peuvent être fusionnées en un graphe de connaissance dynamique avec attributs, qui représente l'évolution d'une situation maritime. Via ce graphe, l'automatisation de la levée d'alerte revient à une tâche de prédiction de lien: étant donnés des labels venant de connaissance experte, y a-t-il d'autres situations similaires que l'on veut relever dans le graphe? Dans cet article, nous allons passer en revue plusieurs techniques de prédiction de lien dans un contexte de surveillance maritime et tirer des conclusions sur les bénéfices que pourrait apporter l'ajout d'attributs dans les modèles de graphes dynamiques pour l'exécution de cette tâche.

Keywords

Graphe de connaissance dynamique, situation maritime, attributs, apprentissage machine, prédiction de liens

Abstract

Currently, maritime surveillance operators have to monitor by hand the massive amount of data at their disposal to spot the events of interest, thus limiting their capabilities. Maritime data comes from various and heterogeneous sources, that can be merged into a dynamic attributed knowledge graph which represents an evolving maritime situation. Using this graph, the automation of alert rising comes through a link prediction task: given some labels from expert knowledge, are there similar situations of interest elsewhere in the graph? In this article, we review link prediction techniques for situation awareness in a maritime context, and draw conclusions on how the addition of attributes in a dynamic graph model could improve results on this task.

Keywords

Dynamic knowledge graph, maritime situation, attributes, machine learning, link prediction

1 Introduction

The maritime domain is the theater of many unlawful activities that may go unnoticed: terrorism, piracy, smuggling, illegal immigration... That's why Maritime Situational Awareness (MSA) is of first importance to maritime security. It is defined by NATO as "The understanding of military and non military events, activities and circumstances within and associated with the maritime environment that are relevant for current and future NATO operations and exercises, where the Maritime Environment (ME) is the oceans, seas, bays, estuaries, waterways, coastal regions and ports" [1]. MSA is often performed by surveillance operators who monitor the flow of data coming from maritime activities. This data is diverse, heterogeneous, and comes from several sources: AIS (Automatic Identification System), radars, satellites, intelligence, websites... With more than 50.000 vessels sailing the oceans each day, there is a need for automation in the detection of illicit events [2].

A maritime situation implies evolving entities: vessels, ports, countries... Such a situation can be represented by a *dynamic attributed* knowledge graph (DAKG), and understanding how its elements connect and jointly evolve gives valuable information pertaining to MSA. This task is here reduced to a link prediction problem. A *link*, or an *event*, is a relation between two entities at a given time point, for instance (Titanic ; :builtBy ; WhiteStarCompany ; 1909), and *attributed* means that entities can have attributes whose values can change over time, e.g. (Titanic ; :passengers ; 2,344 ; April 10th 1912).

Generally, link prediction is performed by learning an embedding for each entity of the graph, and predictions are

made by ranking the events in the graph using these embeddings. This can benefit to MSA in two ways:

- *data completion:* when monitoring an operational situation, the sensors and reports do not always have all the needed information at their disposal. Using link prediction, missing data can be inferred to improve MSA;
- *automated alerts:* link prediction can discover events that a human operator would not have noticed in the massive dataset. Illegal activities could also be anticipated by making prediction in the future and evaluating the risk a ship represents based on its current and past behavior.

In this article, we review (1) two models on a dynamic (but not attributed) knowledge graph, (2) the literature on static/dynamic/attributed knowledge graphs, (3) how to apply DAKGs to MSA.

2 Previous work

The previous work related to this study can be broadly divided into four categories: maritime related work, static graphs, dynamic graphs and attributed graphs.

MSA. MSA often focuses on anomaly detection [3]. It can be tackled with clustering [2], bayesian networks [4], self-organizing maps [5] and many others techniques [6]. Route estimation is also handled, e.g. with neural networks [7] or Extended Kalman filter [8]. To the best of our knowledge, this is the first attempt of using link prediction on DAKG to improve MSA.

Static Knowledge Graph. In a static setting, each node is represented by a single vector. This field is largely covered with a broad range of techniques. Translational models evaluate a fact by measuring the distance between the two entities, generally using the relation during the translation. TransE [9] is its most known representative. Semantic matching models are similarity-based and compare the latent semantics of entities and relations embeddings. RESCAL [10] was the first to do this and has been extended multiple times [11] [12]. Neural network architectures have also been tried with NTN [13] or VGAE [14]. These models achieve great performances on static knowledge graphs but are not suited to deal with dynamic ones.

Dynamic Knowledge Graph. In a dynamic setting, each node is represented by a time series of vector modeling its evolution. This topic is emerging and has less contributions but advances have already been made. Leblay et al. [15] predict time validity for unannotated edges using side information in the learning process. Esteban et al. [16] updates the knowledge graph using an event graph to add new information, and Trivedi et al. [17] extends the bilinear model (RESCAL) with a LSTM network in order to learn non-linearly evolving entities. Jiang et al. [18] incorporate the valid time of facts using a joint time-aware inference model based on Integer Linear Programming. Self-attention networks were tried by Sankar et al. [19].
Although these models are time-aware, they do not include attribute information in the relation prediction task and we will show that they are needed when dealing with MSA.

Attributes. Lin et al. [20] can predict discrete attribute values and find correlation between them. However, they are not included during the learning of relations and relations are not included in the learning of attributes. Tay et al. [21] propose a model that jointly learns KG^R and KG^A with a neural network and predicts continuous values with a regression task. However, neither model deals with temporal data.
Li et al. [22] propose a streaming model (SLIDE) on dynamic attributed networks using a sketching matrix that summarizes the currently observed links and node attributes. They review the challenges pertaining to such networks and real-world data, but they apply it on social networks (Epinions, DBLP, ACM) that have very different kinds of attributes and only a few widely separated timesteps (~20 timesteps from one month to one year each). All these models showed that the addition of attributes improves the results on link prediction.

Table 1 compares representative models from each category and shows that no model currently fits our needs perfectly (see Part 4 for detailed requirements).

	Static	Dynamic	Attributes	Near Real-Time
TransE[9]	X			
Know-Evolve[17]		X		
MT-KGNN[21]	X		Continuous	
SLIDE[22]		X	Discrete	$\sim$

Table 1: Application domain of models comparison

3 Problem statement

The relation and attribute prediction problems are formalized in this section.

3.1 Dynamic Knowledge Graphs

Before introducing dynamicity and attributes, we recall the definition of standard knowledge graphs.

Definition 1 (standard knowledge graph) *Let $E = \{e_1, \ldots, e_n\}$ and $R = \{r_1, \ldots, r_k\}$ be two finite sets, of* entities *and* relations*, respectively. A* knowledge graph *on E, R is a finite set $KG \subseteq E \times R \times E$. For a triple $t = (e^s, r, e^o) \in KG$, e^s is called the* subject *of t, r is called its* relation*, and e^o is called its* object*.*

We now introduce attributes and dynamicity. Note that the relation between an entity and (some value for) an attribute can be seen as a triple in a knowledge graph, but we define it differently because we want to handle them in a specific manner when predicting with knowledge graphs.

Definition 2 (frame) *A* frame *is a quadruple* $F = \langle E, R, A, D \rangle$ *where* E, R, *and* A *are finite sets of elements called* entities, relations, *and* attributes, *respectively, and* $D : A \to S_a$ *is a function assigning a* range $D(A)$ *to each attribute and* S_a *is a set of possible values for* $a \in A$ *(discrete or continuous).*

We write $|E|$ (resp. $|R|$, $|A|$) for the size of E (resp. of R, of A), and $|F|$ for $|E|+|R|+|A|$.
We are now in position to define a knowledge graph with attributes and time (which we simply call "knowledge graph" for simplicity).

Definition 3 (knowledge graph) *Let* $F = \langle E, R, A, D \rangle$ *be a frame. A* standard knowledge graph *on* F *is a couple* $KG = \langle KG^R, KG^A \rangle$, *where*

- KG^R *is a finite subset of* $E \times R \times E \times \tau$ *with* τ *the set of time points,*
- KG^A *is a finite subset of* $E \times A \times D \times \tau$ *such that for all quadruples* $(e, a, v, t) \in KG^A$, $v \in D(a)$ *holds.*

For $KG = \langle KG^R, KG^A \rangle$, KG^R *is called the* relational part *of* KG, *and* KG^A *is called its* attributional *part. The last component of each tuple in* KG^R *or* KG^A *is called its* timestamp *or* time point *(*$t \in \tau$*), the time at which the attribute's value or entity's relation is valid.*

Intuitively, (e^s, r, e^o, t) is read "entity e^s is in relation r with entity e^o at time t", and (e, a, v, u) is read "entity e has value v for attribute a at time u". Given a knowledge graph KG, we always write KG^R (resp. KG^A) for its relational (resp. attributional) part. Figure 1 is an example of the previously defined KG.

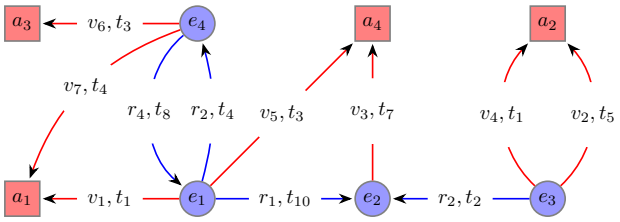


Figure 1: Example of knowledge graph KG on a frame $F = \langle E, R, A, D \rangle$. The nodes $e_i \in E$ are *entities* and the nodes $a_j \in A$ are *attributes*. The *relations* $r_n \in R$ are annotated on edges between two entities. The *values* v_k, annotated on the edges between two attributes, belong to the *range* $D(A)$ of the attribute they are attached to, and the $t'_l s$ are the *timestamps* of the edges. Attributes of entities can change their value over time (e_3 and a_2) and two entities can have common attributes (e_4, e_1 and a_1) but not necessarily with the same value. The blue nodes and edges are KG^R and the red edges with all the nodes are KG^A.

3.2 Prediction Problems with Knowledge Graphs

We are interested in predicting the missing relations between entities and values of attributes in knowledge graphs. We focus on the case where, for some timestamp, they can be predicted from the values of a subset of the relations (R') and attributes (A') at previous timestamps.
The relation r^* (*resp.* attribute a^*) is said to be determined by $KG^{R',A',\leq t^*}$ if for all timestamps t^*, there is a function $f(\cdot)$ such as $f(KG^{R',A',\leq t^*})$ outputs a relational (resp. attributional) quadruple comprised of r^* (*resp.* a^*) at time t^* that exists in KG, where $KG^{R',A',\leq t^*}$ denotes the restriction of KG to quadruples with a relation in R' or an attribute in A', and with a timestamp until t^*. This is the determined relation (*resp.* attribute) problem.

With this in hand, the learning problem which we tackle is the following. Intuitively, for a given knowledge graph KG^*, we are given all the information just before timestamp t^* together with some information at timestamp t^*, and the problem is to induce some target relation r^* (or attribute a^*) at time t^*.

4 Application to MSA

"Real-world" datasets often have more constraints than the academic ones (YAGO [23], Wikidata...) because of their specificities. Maritime datasets are no exception and the following challenges must be overcome.

4.1 Evolution of attributes

A maritime situation is a fast evolving world with very little time between two events. For instance, the event databases ICEWS [24] and GDELT [25] respectively have a temporal granularity of one day and fifteen minutes. In MSA, a good evolutionary model is needed for change detection and the granularity depends on the task. For a change in the position/course/speed of a vessel (dynamic attributes), the information must be given within minutes (e.g., rapid response needed in case of piracy). But to detect a change in a vessel particulars (identifier, name...), the granularity needed can be in hours or days. When modeling the evolution of a vessel's attributes, they can be divided into two categories [26]:

- Static attributes: related to static information about a given vessel (name, flag, length...). They are not supposed to change but their evolution must nevertheless be monitored to report modifications (e.g. change of owner) or anomalies (e.g. identity fraud).
- Dynamic attributes: these can be divided into two subcategories:
  - Kinematic attributes that refers to location, speed, course...
  - Non-kinematic attributes such as passengers, cargo, crew...

Both types of attributes must be handled in the DAKG. Note that they can be discrete (e.g. flag) or continuous (e.g. position).

4.2 Event and threat detection

An event (or quadruple) represents a new relation between two entities or an abovementionned attribute evolution. A maritime relation can be proximity between two vessels, an exchange of goods, harbouring in a port, an attack on another ship... Such events find their roots in both KG^R and KG^A. For instance, two cargo ships from allied countries stopped at the same position are likely to be performing a transhipping (proximity, speed, flag). Currently, most of these events are found using rule-based systems. Using knowledge graphs and machine learning, it could be possible to find events using latent features that cannot be perceived by a human or a rule.

If event mining extracts raw facts, threat detection is a task highly related to its context and definition. A nation will not consider a transhipping between two fishing vessels as a threat since they are more likely to exchange fish than warheads, but an NGO for ocean conservation can suspect illicit fishing of an endangered species. Performing this task still requires either expert knowledge or labeled events.

4.3 Streaming

MSA requires a constant monitoring of maritime areas, meaning that the model must deal with a continuous flow of data. Even if the model does not change after training, the representations of entities and relations must be updated regularly with the incoming information to keep an up-to-date view of the situation. Recent work on the subject can be found in the literature [22, 27].

4.4 Uncertainty

Maritime data often results from hard (sensors) and soft (websites, intelligence) data fusion. However, this data is not always 100% certain: an intelligence report may have a typo, sensors have a range and precision (e.g. +/- 500 meters), or collisions may happen when satellites receive signals. Errors and approximations are inherent to real-world data and the uncertainty of facts must be taken into account when making link prediction [28, 29].

4.5 Explainability

Link prediction models are often black boxes when it comes to the origin of the prediction. However, a surveillance operator needs to know why a prediction was made in order to understand it and justify any upcoming response to an event. Because operators still do not trust AI-based systems to take decisions, explainability is needed to take DAKG-based decisions for MSA [30].

An illustration of all these concepts can be found in Figure 2.

5 Reviewed models

A dynamic and a static link prediction methods are presented in this section. They use embeddings to represent elements of the graph i.e. continuous vector representations for entities, attributes ans sometimes relations ($[c_1, \ldots, c_n]$ with $\forall i \in [1, n], c_i \in \mathbb{R}$ and n the dimension of the embedding). Algorithm 1 shows the high level mechanisms of the two following models.

Algorithm 1: High-level learning algorithm

Result: Up-to-date embeddings
Input: training set *S* (triples/quadruples), entities *E*, relations *R*, number of iterations *nb_it*
Initialization
  Initialize embeddings;
for $i \leftarrow 0$ **to** nb_it **do**
  Sample batch from S;
  Update embeddings of E (and R if relations have embeddings) using score function;
  Update model parameters (if any) using score function;
end

5.1 Know-Evolve

Proposed by Trivedi et al. [17], this model uses a **temporal point process** framework for temporal reasoning over dynamically evolving knowledge graphs that models the **occurence of a fact**. They propose a novel deep learning architecture that evolves over time based on availability of new facts. The dynamically evolving network (Recurrent Neural Network) ingests the incoming new facts, learns from them and **updates the embeddings of involved entities** based on their recent relationships and temporal behavior. Their model can predict the occurence of a fact, but also the time when a fact may potentially occur. It supports the Open World Assumption and can predict over unseen entities.

The point process is characterized by the following conditional intensity function:

$$\lambda_r^{e_s,e_o}(t|\bar{t}) = exp(g_r^{e_s,e_o}(\bar{t}))*(t-\bar{t}) \quad (1)$$

$\lambda_r^{e_s,e_o}(t|\bar{t})$ represents intensity of event involving triplet (e^s, r, e^o) at time t given previous time point $\bar{t}$ when either e^s or e^o was involved in an event. The exp function ensures that intensity is positive and well-defined, and the model is learned by minimizing the joint negative log likelihood of intensity function.
The relational score function $g_r^{e_s,e_o}$ is computed using a bilinear formulation as follows:

$$g_r^{e_s,e_o} = v^{e_s}(t-)^T . R_r . v^{e_o}(t-) \quad (2)$$

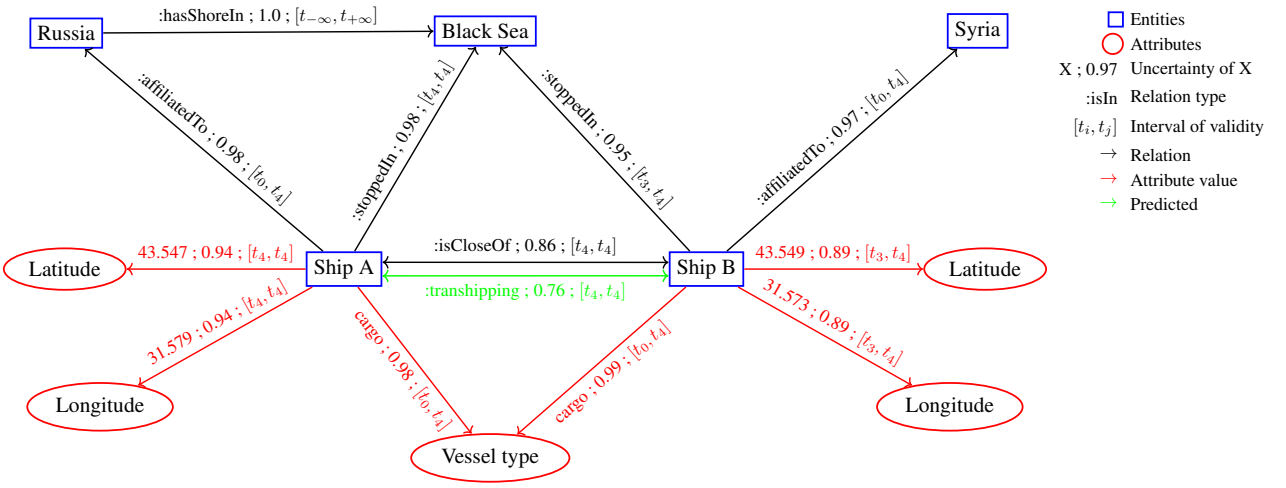


Figure 2: Example of KG at time t_4 (best viewed in color). Events in red represent KG^A and the other events KG^R. Ship A and Ship B are vessels from different countries, moving in the Black Sea. They have a static attribute (vessel type) setting them as cargos and dynamic attributes (latitude and longitude) revealing their positons. Before t_4, the two vessels were moving in the Black Sea and had different positions, but now (t_4) they have stopped and are close to each other. A possible link prediction from $KG^{<t_4}$ is that the two vessels are performing transhipping. Note that the *:isStopped* relation can be deduced from the *speed* attribute going to zero, not represented here for the sake of clarity. If relations such as Russia having a shore on the Black Sea are 100% sure, some are more uncertain: the position of Ship B is only 89% sure because the signal was picked up by a satellite in an area with high ship density. More, the uncertainty of predicted relations (*:transhipping*) depends on the uncertainty of the root events. Finally, to predict the transhipping action in time, the model must be updated with the root causes as soon as they are available, hence the need for streaming link prediction.

with $v^{e_s}, v^{e_o} \in \mathbb{R}^d$ the latent feature embeddings of entities, $R_r \in \mathbb{R}^{d*d}$ the relationship weight matrix and $t-$ represents time point just before time t.
In Know-Evolve, events are included in KG^R and the model partially solves the determined relation problem (only using KG^R). KG^A is not included as Know-Evolve does not consider attribute nodes.
The authors proposed a more recent model [31] that improves the first one with an attention mechanism ; however in the absence of source code, it is Know-Evolve that is evaluated here[1].

5.2 TransE

TransE [9] is the most representative translational distance model and now has many extensions [32]. It represents both entities and relations as vectors in the same space.
Given en event (e^s, r, e^o), the relation is interpreted as a **translation vector** r **between** e^s **and** e^o so that r connects the two embedded entities with low error, i.e., $e^s + r \approx e^o$ when (e^s, r, e^o) holds. The scoring function is then defined as the (negative) distance between $e^s + r$ and e^o , i.e.,

$$f_r(e^s, e^o) = -\|e^s + r - e^o\|_{1/2}$$

where $_{1/2}$ refers to the L_1 or L_2 norm. The score is expected to be large if (e^s, r, e^o) holds. But this method has problems dealing with 1-to-N, N-to-1 and N-to-N relations, and can not process a temporal graph nor KG^A. It can learn over new relations but not over new entities. In our study, it only tackles the determined relation problem without t and KG^A. The evaluation was performed using OpenKE [33][2].

6 Experiments

In these experiments, the performed task is the prediction of the position of a vessel at the next time points. Obviously, there are many better fitted methods to do this (like regression or a Kalman filter), but the ultimate goal is to use the full capacities of the knowledge graph i.e. exploit all the relationships, events and attributes in the maritime surveillance ecosystem to perform better link predictions. Position prediction is just a reduction of this task to test knowledge graphs capabilities on MSA. As we could not find a method handling both time and attributes that can be tested on our data, the attribute *:location* is replaced by a relation *:isLocatedIn* between a vessel and an area.

6.1 Dataset

In the absence of publicly available maritime knowledge graph, we created our own in order to evaluate the models.

AIS data. The dataset used in our experiments is based on real maritime data: AIS messages transmitted by vessels. AIS is a short range (37-74km) ship-to-ship and ship-to-shore navigational data exchange system. It is currently

[1] https://github.com/rstriv/Know-Evolve

[2] https://github.com/thunlp/OpenKE

Dataset	#Vessels	#Areas	#Events	Train	Test
Gibraltar 1M	2,545	1,556	955k	720k	235k

Table 2: Dataset composition

the main source of information available in support of maritime surveillance. The satellite version of AIS (S-AIS) gives a broader range (~5000km) but the transmissions are less regular and more subject to signal collision [2]. AIS provides the following non-exhaustive list of information about ships:

- the unique identifier of the vessel (called MMSI),
- its longitude/latitude,
- its speed and course,
- the timestamp of the report,
- the type of ship,
- the destination.

AIS to KG. A knowledge graph can be built using these AIS messages, where vessels are entities with attributes. Other entities can be added like nations (flag of the ship) or ports. However, the reviewed methods can only handle time, not attributes, hence the need to consider attributes as entities. In our work, the focus is on the evolution of the positions of vessels. Positions being continuous values, they need to be discretized to be casted as entities in the graph. Therefore, the studied area is converted into a grid made of 1km $\times$ 1km squares and each square is an entity (further referred to as "areas").

Moreover, AIS messages are on average received every three minutes so it can be a reasonable choice to separate each time point by three minutes, instead of having a time point every second as it happens in the data (different events can be attached to the same time point). Finally, as the chosen models can not always handle entities or relations not encountered during the training phase, the test set is filtered to remove any event involving an entity or relation not present in the train set. Note that only one relation type is considered here: "vessel :isLocatedIn area" ($|R| = 1$) and each event is represented by a quadruple $(e^s, r, e^o, t) \in KG^R$.
To summarize, we build the knowledge graph consisting of entities = {vessels, areas} and relation = {:isLocatedIn} over one month, we divide it into train/test sets and run the methods to predict the relation ":isLocatedIn" between vessels and areas.

The dataset covers the Gibraltar Strait from February 2^{nd}, 2017 to March 2^{nd}, 2017 and the test set is comprised of the eight last day of the studied period. It means that we are predicting positions at time t given all positions at times $< t$. More information is given in Table 2 and Figure 3 illustrates the trajectories recorded during the considered period.

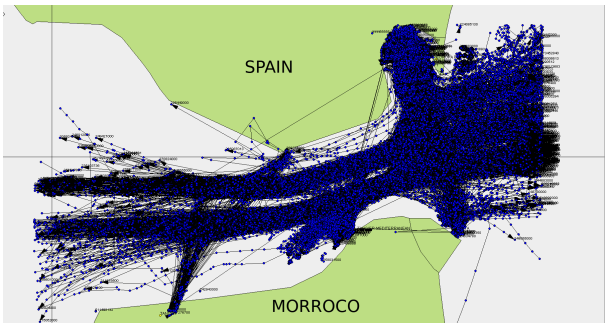


Figure 3: One week of maritime traffic in the Gibraltar Strait (best viewed in color)

6.2 Evaluation task

Link prediction. The evaluation is performed on the link prediction task: given a quadruple (e^s, r, e^o, t), e^o is replaced by every possible entity and the resulting quadruplet is evaluated by the model. All the quadruples are then ranked in descending order of plausiblity and we record the Mean Average Rank (MAR) and the @Hits10 measure (one of the 10 best ranked quadruples is the true one). A lower rank means that the quadruple is classified better (the best rank being 1 and the worst the number of entites) and @Hits10 is expressed in percentage of correctly ranked quadruples i.e. higher is better. The filtering method of TransE [9] is applied, i.e. the quadruple is not ranked against corrupted quadruples that are true.

Sliding window evaluation. The performance is tested using the sliding window evaluation from Know-Evolve. We divide the test set into 8 different slides, each slide including one day of time (Know-Evolve uses 12 slides of two weeks each). This method is said to *"help to realize the effect of modeling temporal and evolutionary knowledge"* [17].

Static method on dynamic data. As it is a static method, the evaluation of TransE required some modifications of the dataset. All the timestamps t are removed and as a result, multiple occurences of the same triples (e^s, r, e^o) appear. Those are removed in order to have a unique representant for each triple and the dataset is then comprised of 102,470 (train) and 16,807 (test) events. The test set still only contains entities seen during training.

6.3 Results

Experimental settings. We used the settings reported in [17] to run Know-Evolve. For TransE, we set batch size=200, learning rate = 0.001 and embedding dimension = 64.

Quantitative Analysis. Figure 4 show the results of the reviewed models over the Gibraltar1M dataset. Know-Evolve, being a temporal model, performs way better than TransE which struggle in @Hits10 prediction despite being not so far from Know-Evolve in Mean Rank. The reason is

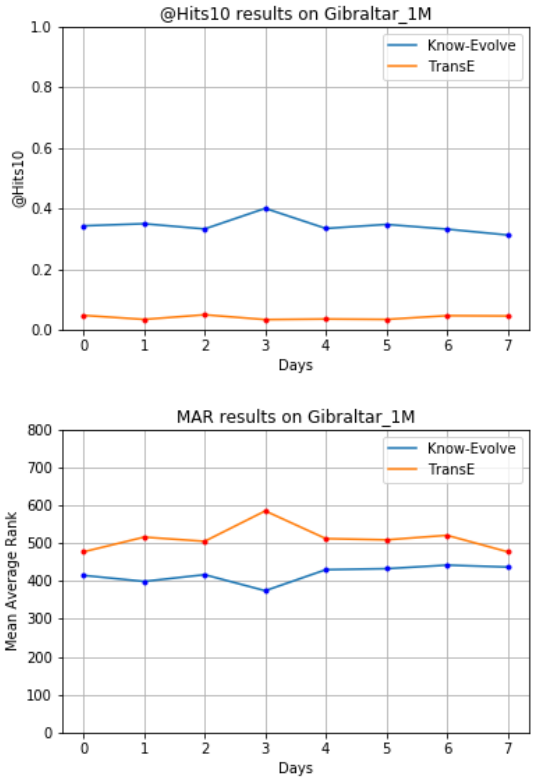


Figure 4: @Hits10 and MAR results of tested models

that TransE depend only on static entity embeddings to perform prediction. With an average @Hits10 of 34%, Know-Evolve captured the relationship between the vessels and the areas better than TransE but do not excels at the task.

Contextual analysis. This evaluation was made on a single task: predict in which area a vessel will be next. This task is made harder by the discretization of the positions: areas are independant and the graph does not tell if areas are close to each other or not. The only way to extract proximity is the analysis of a vessel's track (the succession of relations with area entities), meaning that two areas having a relation with a vessel in a short timespan may be close. More, a proximity relationship between two vessels in the same area could not be established because areas are too wide to consider two vessels as close (e.g. enough to perform an exchange of goods). At last, areas not seen in training cannot be predicted as next location due to the limitations of TransE. Know-Evolve somehow managed to find some connections between vessels and areas but the results are very unsatisfactory: a Mean Rank of 400 means that the correct area is on average ranked 400th, against MR = 20 on ICEWS [17]. Despite the difficulty induced by the discretization, position prediction is a simple task and the models performed poorly: they are not adequated to address this problem. The use of positions as continuous attributes could solve the abovementionned issues and improve the results on position prediction with knowledge graphs.

7 Conclusion and future work

In this article, we reviewed two link prediction techniques for a task: the evolution of the positions of vessels using a dynamic knowledge graph for Maritime Situational Awareness. We showed that relational data (KG^R) is not sufficient for modelling the movement of a vessel and that attributional information should be used (KG^A). We also exhibited the challenges that need to be overcome to apply DAKGs on MSA, and formalized the relation and attribute value prediction problem.

We foresee several tasks for future work: (1) make the prediction task more realistic by adding more entity and relation types in the dataset, such as ships going in and out of ports, or encounters between ships, (2) find a model that can handle both KG^R and KG^A for link and attribute prediction in a temporal setting, (3) perform threat and/or anomaly detection on DAKGs. These are the three requirements to fully evaluate the use of DAKGs on operational maritime data.

8 Aknowledgements

This work was partially supported by the French National Association for Research and Technology and by Airbus Defence and Space.

References

[1] North Atlantic Treaty Organisation, "MC MSA draft definition", 2007

[2] N. Le Guillarme and X. Lerouvreur, "Unsupervised Extraction of Knowledge from S-AIS Data For Maritime Situational Awareness", *FUSION*, 2013

[3] M. Riveiro, G. Pallotta and M. Vespe, "Maritime anomaly detection: A review", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018

[4] S. Mascaro, A. E. Nicholso, and K. B. Korb, "Anomaly detection in vessel tracks using Bayesian networks", *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 2014.

[5] M. Riveiro, F. Johansson, G. Falkman, and T. Ziemke, "Supporting maritime situation awareness using selforganizing maps and gaussian mixture models", *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, vol. 173, p. 84, 2008

[6] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, G.-B. Huang, "Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey", *CoRR*, abs/1606.00981, 2016

[7] D. Zissis, E. K. Xidias, and D. Lekkas, "Real-time vessel behavior prediction", *Evolving Systems*, vol. 7, no. 1, pp. 29-40, 2016

[8] L. Perera, C. Guedes Soares, "Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter", *2nd International Conference on Adaptive and Self-adaptive Systems and Applications*, 2010

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data", *NIPS*, 2013

[10] M. Nickel, V. Tresp, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data", *ICML*, 2011

[11] M. Nickel, L. Rosasco, and T. Poggio, "Holographic Embeddings of Knowledge Graphs", *AAAI*, 2016

[12] H. Liu, Y. Wu, and Y. Yang, "Analogical Inference for Multi-relational Embeddings", *ICML*, 2017

[13] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion", *NIPS*, 2013

[14] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders", *NIPS Workshop on Bayesian Deep Learning*, 2016

[15] J. Leblay and M. W. Chekol, "Deriving Validity Time in Knowledge Graph", *in Companion of the The Web Conference 2018 on The Web Conference*, pp. 1771-1776, 2018

[16] C. Esteban, V. Tresp, Y. Yang, S. Baier, and D. Krompaß, "Predicting the Co-Evolution of Event and Knowledge Graphs", *19th International Conference on Information Fusion*, 2016.

[17] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs", *ICML*, 2017

[18] T. Jiang et al., "Towards Time-Aware Knowledge Graph Completion", *International Conference on Computational Linguistics*, 2016

[19] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dynamic Graph Representation Learning via Self-Attention Networks", *arXiv:1812.09430*, 2018

[20] Y. Lin, Z. Liu, and M. Sun, "Knowledge Representation Learning with Entities, Attributes and Relations", *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016

[21] Y. Tay, L. A. Tuan, M. C. Phan, and S. C. Hui, "Multi-Task Neural Network for Non-discrete Attribute Prediction in Knowledge Graphs", *CIKM*, 2017

[22] J. Li, K. Cheng, L. Wu, and H. Liu, "Streaming Link Prediction on Dynamic Attributed Networks", *in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM, Marina Del Rey, CA, USA*, pp. 369-377, 2018

[23] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia", *Artificial Intelligence*, vol. 194, pp. 28–61, 2013

[24] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, M. Ward, "ICEWS Coded Event Data", 2015

[25] K. Leetaru and P. A. Schrodt, "GDELT: Global Data on Events, Location and Tone", *ISA Annual Convention*, 2013

[26] J. Roy, "Anomaly detection in the maritime domain", *Proceedings of SPIE - The International Society for Optical Engineering*, 2008

[27] P. Zhao, C. Aggarwal, and G. He, "Link prediction in graph streams", *in IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland*, pp. 553-564, 2016

[28] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models", *in International Conference on Machine Learning*, pp. 1278-1286, 2014

[29] M. W. Chekol, G. Pirrò, J. Schoenfisch and H. Stuckenschmidt, "Marrying Uncertainty and Time in Knowledge Graphs", *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017

[30] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)", *IEEE Access*, vol. 6, pp. 52138–52160, 2018

[31] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Representation Learning over Dynamic Graphs", *To be published in ICLR*, 2019

[32] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications", *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, 2017

[33] X. Han et al., "OpenKE: An Open Toolkit for Knowledge Embedding", *EMNLP*,2018

Une approche multi-séries pour la prévision de la demande sur des données d'E-Commerce

Rémy Garnier [1] and Arnaud Belletoile[2]

[1]Université Paris Seine, Laboratoire AGM UMR 8088, 95000 Cergy-Pontoise & CDiscount 33000 Bordeaux, France.
[2]CDiscount 33000 Bordeaux, France

June 19, 2019

Abstract

Prévoir les ventes à l'avance est indispensable pour les chaînes d'approvisionnements modernes du E-commerce. Les outils habituellement utilisés font de la prévision univariée, en traitant chaque série de vente indépendemment. Toutefois, en raison de la faible longueur des séries chronologiques sur les ventes dans le commerce électronique, les méthodes univariées ne s'appliquent pas bien. Dans cet article, nous proposons un modèle global qui surpasse des modèles utilisés dans l'industrie sur un jeu de données réel. Pour ce faire, il utilise des méthodes de renforcement des arbres qui exploitent les informations de non-linéarité et de séries croisées. Nous avons également proposé un cadre de pré-traitement pour surmonter les difficultés inhérentes aux données de commerce électronique. En particulier, nous utilisons différentes méthodes pour limiter l'impact de la volatilité des données.

Mot-clefs: E-Commerce, Prévision de la demande , Boosting Trees, Applied Machine Learning, prévisions de séries temporelles multiples.

1 Introduction

Le E-commerce repose sur des prévisions opérationnelles de la demande au niveau des produits. En effet, les standard modernes des chaînes logistiques modernes exigent un réapprovisionnement *à flux tendus*, pour diminuer les coûts de stockage et les invendus. Une meilleure précision des prévisions peut entraîner d'importantes économies et une réduction du coût et de la place nécessaire au stockage.

Cependant, l'environnement commercial dans le commerce électronique rend cette prévision complexe en raison de la volatilité des ventes. Par exemple, les ventes sont affectées par le calendrier (jour fériés, vacances), les comportements des concurrents, les modifications de prix, *etc* .... Les données relatives à la demande comportent divers défis, tels que des données historiques non stationnaires, des séries chronologiques courtes et des effets de cannibalisation entre produits similaires.

On dispose généralement d'un arbre hiérarchique naturel entre produits, selon le type de produits. Cet arbre se constitue à l'aide d'information sur le famille, le type de produit, la marque, etc... Dans cet arbre hiérarchique, des produits proches auront des comportements similaires. Ainsi, les produits de la famille 'Jouets' auront une saisonnalité similaire et connaîtront souvent leurs meilleures ventes de l'année avant Noël.

Les méthodes existantes traitent généralement différentes séries séparément. Cela fonctionne dans la vente au détail physique, mais la rotation rapide des produits et la volatilité de la demande dans la vente en ligne nécessitent de fournir des modèles qui partagent les informations entre les séries chronologiques[Yelland, 2010, Chapados, 2014, Trapero et al., 2015, Bandara et al., 2019].

Dans cette étude, nous proposerons un cadre pour le problème de prévision de la demande du monde réel dans le commerce électronique. Notre objectif est d'exploiter la corrélation entre les séries pour améliorer la précision des prévisions. En particulier, nous cherchons à surmonter le problème de la faible longueur des séries chronologiques.

Dans la section 2, nous définissons formellement le problème et proposons un bilan rapide des travaux antérieurs sur le terrain. Nous présentons un pré-traitement des données dans la section 3. Dans la section 4, nous présentons le modèle de boosting qui nous donne les meilleures performances. Enfin, nous présentons la configuration et les résultats de notre expérience sur un ensemble de données réelles sur la section 5.

2 Contexte

2.1 Position du problème

Nous avons un ensemble I de produits, divisé en K différentes catégories I_k telles que $I = \biguplus_k I_k$ (I union disjointe de I_k). Nous avons également N compté fois série $(y_{i,t})$, où $y_{i,t}$ représente le nombre de ventes du produit $i \in I$ au cours de la semaine t. Cette série s'observe pendant les T semaines.

Le support de cette série, c'est-à-dire le nombre de semaines non nulles pour chaque série est relativement faible par rapport à T. Cela signifie que nous n'observons pas beaucoup d'historique pour chaque produit individuellement. Nous supposons que les séries suivent une saisonnalité de période τ, la plupart du temps annuelle ($\tau = 52$), bien qu'une période entière soit rarement observée pour un produit donnée.

Certaines caractéristiques externes $Z = ((z_{i,t})_i)_t$ sont importantes. Trois types de covariables peuvent être utilisés:

- **Variables temporelles:** Covariables dépendant de la date t uniquement. Elles sont communnes à tous les produits. Par exemple, les événements spéciaux (Noël, Black Friday) et les covariables liées aux conditions météorologiques entrent dans cette catégorie.
- **Variables longitudinales:** Covariables dépendant du produit i uniquement. Par exemple, le type de produit, sa marque. Les caractéristiques longitudinales permettent de creer une hiérarchie entre les produits.
- **Variables mixtes**: Covariables dépendantes des deux. Par exemple, le prix d'un produit peut varier chaque semaine.

Notre objectif est de prévoir les valeurs de cette série pour un horizon h. Plus formellement, nous souhaitons développer un modèle de prévision f, tel que, si nous considérons les ventes passées d'un produit i $y_{i,:t} = (y_{i,0}, \ldots, y_{i,t})$, la valeur $f(y_{i,:t}, z_{i,t}, \theta)$ est un estimateur de $y_{i,t+h}$ pour un ensemble de paramètres θ pouvant être appris .

2.2 Travaux Similaires

Un grand nombre de travaux ont été publiés concernant les méthodes de prévision de la demande, pour différentes applications (installations, vente au détail physique et en ligne, ...). Les méthodes les plus largement utilisées sont les modèles de séries chronologiques classiques tels que les modèles ARIMA [Ediger and Akar, 2007] et les variantes de lissage exponentiel [Taylor, 2003]. Cependant, les prévisions dans le domaine du commerce électronique doivent généralement faire face à des problèmes tels que les tendances des ventes irrégulières, la présence de données de vente extrêmement volumineuses et éparses, *etc*. Certaines de ces limitations peuvent être surmontées grâce à la fonction de vraisemblance modifiée et aux modèles linéaires étendus [Seeger et al., 2016]. Mais cette méthode ne parvient pas à obtenir de bonnes performances lorsque les séries sont petites.

D'autres méthodes de régression ont été proposées. Par exemple [Pierrot and Goude, 2011] utilisent des modèles additif généralisées pour la demande d'electricité, [Chen et al., 2004] utilise,t des SVR(régression à support de vecteur) et [Borovykh et al., 2017] des réseaux de neurones récurrents. Toute ces méthodes ne croisent pas les informations disponibles entre séries et s'étendent mal au problème de prévision de la demande pour le E-commerce.

Récemment, [Bandara et al., 2019] ont proposé d'utiliser des réseaux de neurones profonds pour réaliser des prédictions en transférant des informations entre séries dans le cadre du E-commerce. Ils adaptent une architecture de réseau de neurones(Long Short Term Memory ou LSTM) pour traiter toutes les séries en même temps. Ils séparent également les effets des caractéristiques longitudinales et temporelles pour obtenir de bonnes performances. Cela suggère que le partage d'informations entre séries permet d'améliorer les performances en prédiction.

Les modèles hiérarchiques bayésiens sont un autre modèle prometteur [Yelland, 2010, Chapados, 2014]. Ces modèles expriment les ventes d'un produit comme issues d'une distribution dont les paramètres sont eux même issue d'une distribution dépendant caractéristiques du produits (prix, type). On a donc des équations à plusieurs niveaux, à la fois au niveau du produit et au niveau de sa catégorie. Ces modèles permettent d'établir des relations plus explicites entre les prédictions et les features, ainsi que de donner des bornes pour les intervalles de confiance.

3 Pré-traitement des données

3.1 Données de Ventes

Il existe deux types de problèmes avec les données de vente dans le commerce électronique. Le premier est la présence de valeurs anormalement basses, ou "faux zéros". Ces faibles valeurs peuvent être dues à des ruptures de stock, à des problèmes de réseau ou à la modification du moteur de recherche sur le site Web. Notre objectif est de prédire une demande, qui a pu se reporter sur autre chose durant les semaines anormales (autre produits, concurrents,...) .Nous devons donc identifier et remplacer ces valeurs par

des valeurs 'raisonnables' pour la demande. La correction des faux zéros se fait en remplaçant les valeurs anormalement 'basses' par des valeurs fictives issues d'algorithme univariés simples sur chaque séries temporelles . Ces valeurs serviront à entrainer les modèles, mais ne seront pas utilisées pour l'évaluation.

Le deuxième problème est la présence de valeurs anormalement élevées. Ces valeurs sont informatives, car elles nous renseignent sur l'effet des ventes. Cependant, ces valeurs sont problématiques lorsqu'elles sont utilisées en tant que variables explicatives, car elles peuvent suggérer un niveau de ventes supérieur aux prévisions ou donner des informations erronées sur les tendances et la saisonnalité. Par conséquent, nous construisons des 'ventes lissées' $x_{i,t}$ en éliminant les valeurs supérieures à γ multipliées par la variation standard. Plus précisément:

- Pour chaque produit, nous calculons une moyenne et un écart-type mobile

$$\overline{y_{i,t}} = \frac{1}{M}\sum_{k=0}^{M} y_{i,t-k}$$

$$\overline{\sigma_{i,t}} = \left(\frac{1}{M}\sum_{k=0}^{M}(y_{i,t-k} - \overline{y_{i,t}})^2\right)^{\frac{1}{2}}$$

.

- Si $y_{i,t} > \overline{y_{i,t}} + \gamma\overline{\sigma_{i,t}}$, alors $x_{i,t} = \overline{y_{i,t}} + \gamma\overline{\sigma_{i,t}}$.
- Sinon $x_{i,t} = y_{i,t}$.

3.2 Features temporelles: Saisonnalité et Tendance

La saisonnalité et la tendance d'une série temporelle sont deux caractéristiques essentiellement temporelles, et qui peuvent difficilement être déduites par un algorithme de machine learning utilisé en régression, comme ce qui sera présenté en section 4. Afin d'enrichir notre apprentissage, et d'introduire des aspects temporels dans notre régression, nous allons construire des features temporelles dans notre modèle correspondant à des tendances locales, des tendances annuelles et à des saisonalités pour chaque produit.

Construire une tendance locale est relativement évident. Etant donné un produit i, on peut calculer une tendance locale par régression (linéaire) sur une fenêtre glissante le long de la série. Cependant, il vaut mieux prendre une fênêtre assez large pour ne pas tenir compte des mouvemens les plus violents. Il est possible d'effectuer des calculs de saisonnalités annuelles de manière analogue lorsque l'information est disponible. Lorsque ce n'est pas le cas, on peut calculer une saisonnalité annuelle unique pour chaque catégorie de produits, représentant l'évolution annuelle du marché pour un produit particulier.

Le traitement des saisonnalités est plus complexe en raison de la brièveté des séries considérées. Nous utilisons une variante de la procédure décrite dans [Kumar et al., 2002] pour produire un facteur de saisonnalité pour chaque produit. Esquissons cette procédure.

Tout d'abord, nous normalisons les chiffres de vente pour chaque année. Nous voulons nous assurer que chaque produit a le même niveau moyen. Pour chaque produit i, en considérant N_i le nombre de semaines de vente au cours de l'année, nous notons pour une date t cette année (c.-à-d. $T \in 0, \ldots, \tau - 1$):

$$x_{t,i}^{std} = \frac{N_i}{\tau} \cdot \frac{x_{t,i}}{\sum_{t=0}^{\tau} x_{t,i}}$$

Deuxièmement, nous calculons la moyenne des valeurs normalisées dans chaque catégorie I_k. Nous avons donc une saisonnalité standardisée pour chaque catégorie de produit. L'idée centrale est de supposer qu'il existe une saisonnalité multiplicative commune $s_{I_k}(t)$ pour tous les produits de cette catégorie. Par conséquent, si la date à laquelle le produit a été mis sur le marché est uniformément répartie, la moyenne calculée est directement proportionnelle à la saisonnalité.

Toutefois, en raison de la nature erratique des données sur les ventes dans le E-commerce, à ce stade, la saisonnalité calculée n'est souvent pas assez informative et est souvent très bruité par la présence d'évènement particuliers propres à une année et une catégorie de produits (par exemple, les coupes du monde de football pour les ventes de téléviseurs) .

C'est pourquoi nous utilisons un algorithme de clustering de séries chronologiques pour regrouper les saisonnalité des différentes catégories. Ce regroupement est basé sur la distance euclidienne entre les modèles de saisonnalité, mais prend également en compte la variabilité des saisonnalités constaté dans chaque catégories.

Après clustering, nous obtenons un faible nombre de pattern de saisonnalités normalisées différentes, que nous introduisons comme variables explicatives features dans l'algorithme de machine learning.

3.3 Traitement des autres features

Traitement des features catégorielles Le E-commerce dispose naturellement de nombreuses variables longitudinales catégorielles sur les produits (famille de produits, type, marque,gamme de prix, avis clients,...) qu'on aimerait pouvoir exploiter. Plusieurs méthodes sont possibles:

Tout d'abord, on peut séparer les produits par catégories, et entrainer un modèle pour chaque catégorie de produits. Cependant, on se heurte rapidement à la multiplicité des catégories considérées. Cela impose d'utiliser des algorithmes très simples, et empêche de capturer des effets plus faibles qui apparaîtraient en considérant un plus grand nombre de produits. Dans les exemples, on distinguera un modèle 'Par Magasin', qui entraîne un algorithme de machine learning par magasin, et un modèle global, qui entraîne un modèle sur l'ensemble du site.

Pour traiter des features plus complexes, on est donc amener à encoder les features catégorielles, c'est à dire à coder numériquement chaque catégorie. La méthode classique, dite de One-Hot Encoding, consiste à introduire une colonne booléenne pour catégorie. Cependant, du fait du grand nombre de catégories, cette méthode est impraticable en pratique.

Deux possibilités subsistent. Tout d'abord, il est possible d'encoder les variables catégorielles sur plusieurs colonnes via une fonction de hashage. Cette méthode limite l'espace nécessaire pour l'apprentissage, mais rend plus difficile l'interprétation de l'importance relative des différentes variables.

Une autre possibilité consiste à utiliser un encoding ordinal, c'est à dire à associer un entier à chaque catégorie. Cette méthode est extrêmement simple, et permet une interprétation facile des résultats, mais introduit un ordre sur les features qui ne repose sur rien. Pour limiter cet effet, on peut agréger les résultats issus de plusieurs permutations possibles des features catégorielles.

Traitement des features imprévisibles Certaines featuress mixtes ou temporelles, telles que la météo ou les prix, ne peuvent pas être utilisées pour la prédiction, car elles ne peuvent pas être prédites pour l'horizon sur lequel nous voulons prévoir des informations. Cependant, ces fonctions peuvent être utilisées pour former le modèle sur les données passées, afin d'expliquer des valeurs anormalement basses (ou élevées) dans le passé. Nous pouvons ensuite effectuer une prédiction en utilisant une estimation des valeurs futures. Par exemple, nous pouvons prendre la valeur saisonnière des données météorologiques ou le prix passé moyen observé des produits considérés . Ce schéma a une faiblesse : le fait que nous utilisions des valeurs antérieures exactes conduit l'algorithme d'apprentissage automatique à donner beaucoup d'importance à ces features.

4 Modèle

4.1 Schéma d'apprentissage

Nous considérons notre problème de prévision de séries chronologiques multiples comme un problème de régression. Notre objectif est les valeurs de vente corrigées des "faux zéros" à l'horizon $y_{i,t+h}$. Nous utilisons les valeurs passées des ventes lissées comme des features, comme décrit dans 3.1. Nous avons donc une prédiction

$$\widehat{y_{i,t+h}} = f(x_{i,:t}, z_{t+h,i}, \theta)$$

L'hypothèse est que $x_{i,t}$ représente le niveau de vente 'normal'. Il est supposé supprimer les effets des effets ponctuels, comme les offres spéciales. Les variables externes $z_{i,t}$ nous donnent des informations sur la différence $\delta_{i,t} = y_{i,t} - x_{i,t}$. Par conséquent, nous préférons utiliser les valeurs décalées de la variable lissée $x_{i,t}$ comme variable explicatives plutôt que $y_{i,t}$.

Nous avons utilisé comme ensemble d'apprentissage les valeurs des tuples $(x_{i,:t}, z_{t+h,i})$ pour tous les produits i avant une date donnée. Les hyper-paramètres sont sélectionnés en utilisant une simple période de validation.

Nous résumons tout sur la figure 1.

4.2 Métriques d'évaluation et métriques d'apprentissages

Nous devons différencier les métriques d'évaluation des métriques utilisées lors de l'apprentissage de la fonction de machine learning f. Nous avons utilisé comme métriques d'évaluation de l'erreur quadratique moyenne (RMSE) et de l'erreur absolue moyenne (MAE) en tant que métriques d'évaluation, en utilisant le prix du produit p_i comme poids.

$$RMSE(f, \theta) = \sqrt{\frac{1}{n}\sum_{i \in I} p_i^2 \cdot (y_{i,t+h} - f(x_{i,:t}, z_{t+h,i}, \theta))^2}$$

$$MAE(f, \theta) = \frac{\sum_{i \in I} p_i \cdot |y_{i,t+h} - f(x_{i,:t}, z_{t+h,i}, \theta)|}{\sum_{i \in I} p_i \cdot f(x_{i,:t}, z_{t+h,i}, \theta)}$$

Ces métriques sont couramment utilisées dans les prévisions de chaîne d'approvisionnement. Cependant, ces métriques souffrent de plusieurs faiblesse pour les utiliser comme métriques d'apprentissage. En effet, le RMSE et le MAE ont tendance à sous-estimer la prévision en raison du caractère positif de la série. De plus, l'erreur de prédiction des produits les plus vendus a tendance à l'emporter sur les autres erreurs.

On va utiliser une métrique différente pour modéliser la dispersion de la série. On va en effet supposer que,

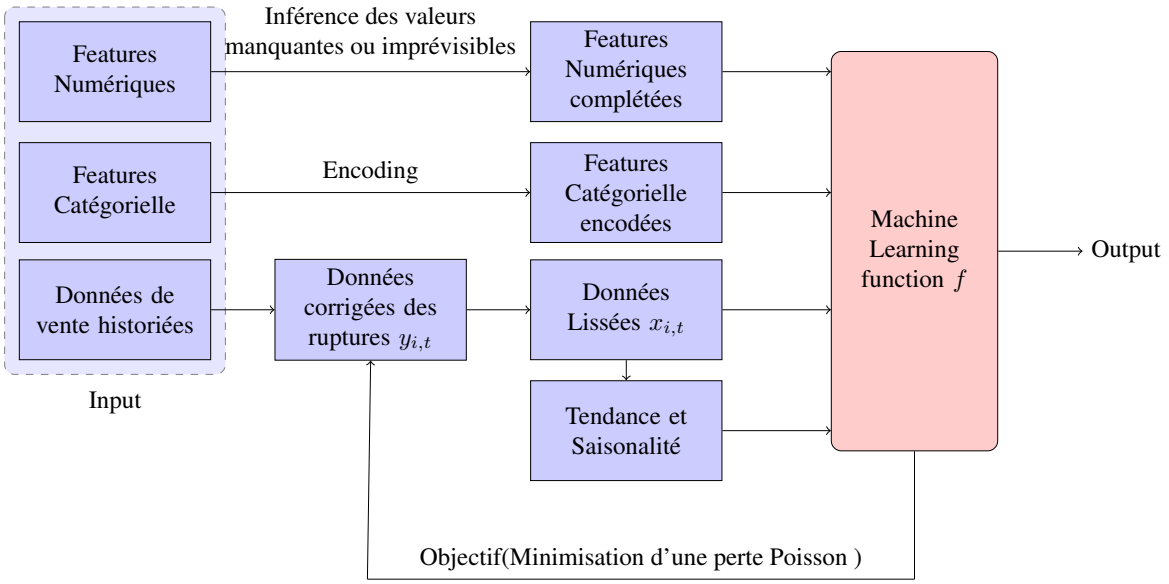


Figure 1: Schéma général

les valeurs $y_{i,t}$ sont tirées par des distributions de Poisson indépendantes du paramètre $lambda_{i,t}$ pour tout produit i et toute date t.

Ce choix a déjà été utilisé, par exemple par [Borovykh et al., 2017] et est naturel pour trois raisons. Premièrement, nous avons observé que la variance des séries est proportionnelle à la moyenne empirique des séries, avec des coefficients de proportionnalité proche de 1, ce qui est ce qu'on attend dans le cas d'une distribution de Poisson.
Deuxièmement, cela nous permet de limiter les effets de la présence de valeurs aberrantes dans nos données. En effet, des valeurs plus élevées sont plus probable que dans une modélisation par un bruit blanc gaussien, par exemple.
Troisièmement, les valeurs entières positives sont naturellement modélisées par un processus de comptage. Nous pouvons supposer que pour chaque semaine t et chaque produit i, les dates d'arrivées du client suivent un processus de Poisson et que le paramètre de ce processus change chaque semaine.

C'est pourquoi on peut utiliser une perte de Poisson lors de la phase d'apprentissage de notre modèle . Si l'on suppose que $y_{i,t}$ est distribué selon une distribution de Poisson de paramètre $\lambda_{i,t}(\theta) = f(x_{i,:t}, z_{t+h,i}, \theta)$, le critère que nous voulons optimiser est alors la log-vraisemblance de la valeur $y_{i,t}$ en faisant varier les paramètres θ de la fonction d'apprentissage:

$$\text{Poisson}(f, \theta) = \sum_{i,t} \lambda_{i,t}(\theta) - y_{i,t} \log(\lambda_{i,t}(\theta))$$

4.3 Algorithme

Le choix de l'algorithme d'apprentissage automatique pour calculer f et θ est crucial. D'une part, il doit être suffisamment souple pour utiliser différents types de features et pour sélectionner les variables explicatives les plus utiles. En particulier, il devrait pouvoir résister à une redondance des features . En revanche, il doit être suffisamment consistant pour éviter le sur-apprentissage. Enfin, en raison du grand nombre de séries et de features, il doit être suffisamment rapide pour gérer des données volumineuses.

Nous avons testé différents modèles. Pour chacun, nous avons sélectionné les hyper-paramètres par validation croisée. Nous essayons également de normaliser les features dans les différents cas.

Modèles linéaires Le choix le plus simple est de chercher une fonction f linéaire. Dans ce cas, la simplicité du modèle autorise même d'utiliser le One-Hot Encoding. Il est également possible d'ajouter une régularisation L1 ou L2 pour éviter l'overfitting. Cependant, les effets étudiés semblent profondément non-linéraire, et dépendent d'effets croisés entre features que les modèle linéaires ne prennent pas en compte.

Modèle additifs généralisées (GAM) On peut étendre un peu les modèle linéaires, et utiliser des modèles linéaires généralisées(GAM). L'idée est de chercher la fonction f dans une base de fonctions relativement simples. Les modèles GAM sont fréquemment utilisés pour prédire des séries temporelles(voir [Hastie, 2017]) et peuvent prendre en compte des effets croisées. Cependant, il n'est pas évident de choisir une bonne base de fonction f, et nous ne sommes pas parvenus à trouver une bonne modélisation par ces modèles.

Forêts aléatoires Les forêts aléatoires sont un type d'algorithme de bagging, qui consiste à construire différents arbres de régression par boostrap, puis à produire une prédiction basée sur les prédictions des différents arbres. Il permet de prendre en compte les effets de seuil et des effets croisés. Il peut être parallélisé, ce qui permet un calcul rapide.

Les forêts aléatoires conviennent bien à l'estimation de f et permettent donc d'obtenir de bonnes performances sur les jeux de données.

Arbres boostés Contrairement aux méthodes de forets aléaires, les méthodes d'arbres boostées implémentent un regroupement séquentiel de la prédiction de différents arbres. Ils ont récemment fait l'objet de beaucoup d'attention, en raison de leurs performances sur des cas réels. Ici, nous utilisons principalement XgBoost [Chen and Guestrin, 2016], qui en est une implémentation à gradient rapide.

Il conserve les avantages des forêts aléatoires, mais offre de meilleures performances. Le prix est généralement un temps de formation plus long, car la formation ne peut pas être mise en parallèle. Les hyper-paramètres XgBoost sont sélectionnés via validation. Les domaines de validation des hyper-paramètres sont présentés sur la table 1. Nous utilisons un arrêt précoce pour réduire le temps d'entraînement.

hyper-paramètres	Min value	Max value
learning rate	0.01	0.3
min split loss	0.01	0.2
max depth	5	8
round evaluation	1000	5000

Table 1: Domaine utilisées pour rechercher les hyper-paramètres pour XgBoost

5 Experience

5.1 Dataset

Pout tester le cadre que nous proposons pour la prévision des ventes, nous allons prendre un ensemble de données provenant de *Cdiscount.com*. Il rassemble les ventes de 99305 produits, répartis dans 10 magasins et 1031 catégories, sur une période d'environ 4 ans. Sur la figure 2, nous avons représenté la répartition des produits par durée de vie. Seule une minorité des produits depasse les 52 semaines de ventes.

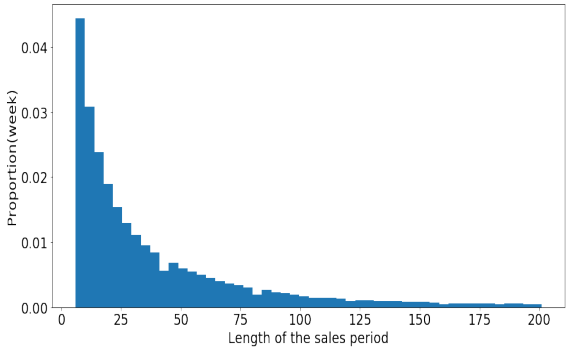


Figure 2: Répartition des produits par durée de vie

En figure 3 quelques exemples de courbes de ventes pour des téléviseurs.

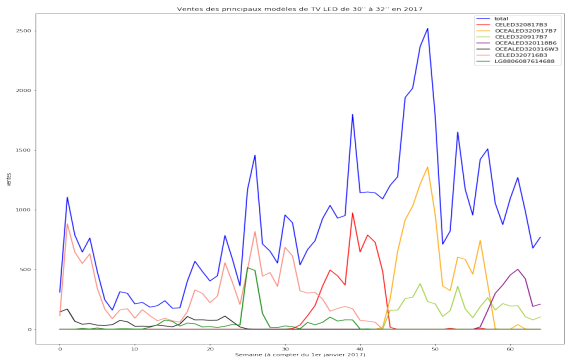


Figure 3: Ventes de quelques modèles de téléviseurs

Nous prenons un horizon h de 6 semaines, puis formons le modèle sur les 170 premières semaines, puis utilisons les 10 prochaines semaines pour valider les hyper-paramètres. Pour évaluer le modèle, nous utilisons les 19 dernières semaines. Ces semaines correspondent aux dernières semaines de l'année 2018 et au début de 2019. Elles contiennent donc beaucoup de variabilité (Black Friday, Noël,

soldes d'hiver).

Il existe 3 ensembles de produits, nommés A, B et C. Le premier regroupent les produits qui se vendent le plus, le dernier les produits qui se vendent le moins.

5.2 Benchmark et variantes

Nous comparons notre approche à des algorithmes "maison", ainsi qu'à des solution industrielles.

Plus précisement, nous utilisons un simple algorithme de lissage exponentiel comme référence (ES). Par ailleurs nous comparons avec un algorithme commercialisé, développé par la société Relex[Rel,]. Cet algorithme effectue une prédiction pour chaque série en utilisant une classification des séries chronologiques et des connaissances métier.

Nous présentons les performances de l'algorithme XgBoost dans différentes configurations. Nous faisons varier l'encodage des variables catégorielles, ainsi que l'ensemble de données considérées. Dans le cadre global, nous entrainons l'algorithme sur l'ensemble des produits disponibles. Dans le cadre mixte, on entraîne un faible nombre (4) de magasins très particuliers séparément. Enfin, on étudie l'impact de l'ajout d'une feature modélisant la saisonnalité sur nos prédictions.

Un avantage de la méthode de prédiction globale est qu'elle permet la prédiction *à froid*, c'est-à-dire la prédiction de nouvelles séries sans historique. Pour obtenir une évaluation similaire à celle du benchmark, nous ignorons les 6 premières semaines de vie des produits, où notre algorithme est capable de prédire, mais pas les algorithmes de référence.

5.3 Résultats

Le tableau 4 présente les performances de la prévision pour deux mesures d'évaluation pour l'ensemble des produits et pour les différents ensembles A, B et C. Les valeurs RMSE sont exprimées en milliers d'euros (k€). Nous présentons différentes versions de notre algorithme en fonction du codage des caractéristiques catégorielles (ordinal ou hachage) et de l'utilisation d'une feature exprimant la saisonnalité (décrites en section 3.2).

Nous pouvons voir que XgBoost surpasse le benchmark pour toutes les catégories. Il réduit le MAE d'environ 5% et le RMSE de 10% sur l'ensemble des jeux de données. Globalement, le gain relatif est plus important dans le RMSE que dans le MAE, ce qui montre qu'il réduit généralement le plus grand écart de performances plus qu'il n'améliore la prédiction moyenne.

L'introduction d'une variable modélisant la saisonnalité améliore les performances en général pour les produits les plus vendus, notamment en RMSE. Cependant, elles semblent dégrader les performances sur les produits les moins vendus. On peut supposer que l'ajout d'une saisonnalité n'est vraiment sensible qu'à partir d'un certain niveau de ventes.

Bizarrement, l'encodage ordinal semble plus efficace que le hashing.

Si nous examinons attentivement les performances, nous constatons que notre algorithme est particulièrement performant au cours des semaine du cycle du produit. Nous présentons ces résultats dans 3 pour le benchmark et un XgBoost avec saisonnalité . La forte variabilité de la RMSE est due au faible nombre de produits concernés et à la forte variabilité de la période étudiée. Néanmoins, nous pouvons constater que, au début du cycle de produit, notre structure dépasse fortement la référence. Nous diminuons par exemple le MAPE de $24,0$ % et le RMSE de $42,5$ % pour le produit avec 10 semaines de données historiques. Cette différence diminue avec le temps, à mesure que le benchmark acquiert un historique suffisant pour sa prédiction.

6 Conclusion

L'amélioration de la prévision de la demande dans le commerce électronique est possible grâce à l'utilisation de méthodes globales, qui partagent des informations entre des séries temporelles. Dans notre article, nous avons proposé d'utiliser une méthode de renforcement de gradient pour le faire, mais d'autres méthodes sont en développement[Bandara et al., 2019]. Cela nous permet d'exploiter les caractéristiques croisées et les effets non linéaires existant dans les données de commerce électronique. En outre, nous pouvons également effectuer la prédiction *à froid*, avec très peu d'historique sur nos produits.

Nous avons également proposé plusieurs astuces pour résoudre les difficultés inhérentes aux données de commerce électronique. En particulier, nous avons proposé un moyen de calculer la saisonnalité des produits grâce au comportement des catégories de produits considérées.

Enfin, nous évaluons notre méthodologie sur un ensemble de données du monde réel, avec un nombre de produits réaliste, et surpassons les solutions de pointe en matière de prévision de la demande.

Cependant, nous n'avons pas de modélisation de la compétition 'entre produits' au sein d'un site de E-commerce. D'autres travaux seront nécessaires pour développer des modèles pouvant prendre en compte cette compétition.

Framework				Tous		A		B		C	
ML Algo.	Configuration	Encodage	Méthode	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ES				3.83	1.09	5.68	1.03	1.12	1.06	1.28	1.31
RF	with seas.	Ordinal	Global	3.09	0.831	5.27	0.796	1.66	0.892	1.32	0.92
XgBoost	Poisson/with seas.	Ordinal	Global	**2.67**	0.725	**4.59**	0.674	1.41	0.801	**1.20**	0.874
XgBoost	Poisson/without seas.	Ordinal	Global	2.76	0.730	4.72	0.681	1.39	0.800	1.21	0.872
XgBoost	Poisson/with seas.	Hashing	Global	2.78	0.728	4.75	0.685	1.41	0.816	1.21	0.893
XgBoost	Poisson/without seas.	Hashing	Global	2.79	0.740	4.77	0.689	1.40	0.817	1.22	0.893
XgBoost	Poisson/with seas.	Ordinal	Mixte	2.80	**0.707**	4.67	**0.656**	**1.33**	0.788	1.23	0.852
XgBoost	Poisson/without seas.	Ordinal	Mixte	2.79	**0.707**	4.77	0.658	1.36	**0.785**	1.25	**0.85**
Benchmark				3.01	0.758	4.97	0.688	1.77	0.907	1.56	0.982

Table 2: Comparaison de différents modèles

Product cycle	Framework		Benchmark	
Longueur	RMSE	MAE	RMSE	MAE
8	3.71	1.04	6.04	1.77
9	3.28	0.879	6.43	1.36
10	3.67	0.920	6.39	1.21
11	11.48	1.15	11.97	1.31
12	5.33	0.867	7.19	0.928

Table 3: Performance sur le début du cycle de vie des produits

References

Relex. `https://www.relexsolutions.com/fr/`.

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., and Seaman, B. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. *arXiv preprint arXiv:1901.04028*.

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *stat*, 1050:16.

Chapados, N. (2014). Effective bayesian modeling of groups of related count time series. In *International Conference on Machine Learning*, pages 1395–1403.

Chen, B.-J., Chang, M.-W., et al. (2004). Load forecasting using support vector machines: A study on eunite competition 2001. *IEEE transactions on power systems*, 19(4):1821–1830.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Ediger, V. Ş. and Akar, S. (2007). Arima forecasting of primary energy demand by fuel in turkey. *Energy policy*, 35(3):1701–1708.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Kumar, M., Patel, N. R., and Woo, J. (2002). Clustering seasonality patterns in the presence of errors. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 557–563. ACM.

Pierrot, A. and Goude, Y. (2011). Short-term electricity load forecasting with generalized additive models. *Proceedings of ISAP power*, 2011.

Seeger, M. W., Salinas, D., and Flunkert, V. (2016). Bayesian intermittent demand forecasting for large inventories. In *Advances in Neural Information Processing Systems*, pages 4646–4654.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8):799–805.

Trapero, J. R., Kourentzes, N., and Fildes, R. (2015). On the identification of sales forecasting models in the presence of promotions. *Journal of the operational Research Society*, 66(2):299–307.

Yelland, P. M. (2010). Bayesian forecasting of parts demand. *International Journal of Forecasting*, 26(2):374–396.

Clustering et interactions multi-agents pour la création de groupes de vacanciers

E. Grislin Le Strugeon E. Adam

Univ. Polytechnique Hauts-de-France, CNRS, UMR 8201 - LAMIH -
Laboratoire d'Automatique de Mécanique et d'Informatique Industrielles et Humaines,
F-59313 Valenciennes, France

emmanuelle.grislin@uphf.fr, emmanuel.adam@uphf.fr

Résumé

L'objectif applicatif de cette étude en cours est de proposer une aide à la mise en relation de personnes âgées souhaitant partir en vacances. L'approche proposée associe une pré-sélection basée sur des mesures de compatibilité relatives aux préférences des personnes, et des interactions par appels d'offres entre agents.

Mots Clef

Agent, interactions multi-agents, fouille de données, clustering, groupes d'utilisateurs, apprentissage de préférences

Abstract

The application goal of this ongoing study is to offer an assistance to elderly who aim at going on holiday. The proposed approach includes a first selection based on preference related compatibility measures and agent interactions by bidding mechanisms.

Keywords

Agent, multiagent interaction, data mining, clustering, user group, preference learning

1 Introduction

Les personnes âgées de plus de 60 ans représentent une part croissante de la population française. Parmi celles-ci, nombreuses sont celles qui vivent seules, elles constituent par exemple 27% des 60-79 ans et 52% des 80 ans et plus dans les Hauts-de-France. Ces chiffres sont à mettre en regard des 59% de voyages en couple et 31% seules chez les personnes âgées de 62 à 89 ans [1]. L'isolement constitue un frein pour leurs activités et en particulier concernant leurs possibilités de partir en vacance hors du domicile. Le projet PartEns ("Partir Ensemble Sénior") [Naveteur and Secq, 2018] s'est attaché à spécifier une plate-forme dont l'objectif est de faire se rencontrer de potentiels co-vacanciers parmi ces personnes âgées isolées. Les travaux présentés ici constituent une partie des réponses apportées dans le cadre de ce projet.

La problématique présentée ici consiste à former des regroupements parmi les "candidats aux vacances" en tenant compte de leurs préférences. En effet, une étude préliminaire dans le cadre du projet PartEns [Delelis et al., 2017] a mis en évidence l'importance des souhaits des candidats, non seulement concernant le séjour de vacances lui-même mais aussi concernant les caractéristiques des personnes partageant le séjour. Il s'agit de plus de permettre, en continu, l'adaptation des regroupements à l'arrivée de nouveaux candidats.

L'approche que nous avons adoptée est basée sur des agents représentant les candidats aux vacances. Par exemple, l'agent représentant Annie cherche des co-vacanciers non fumeurs pour partir une semaine à la mer en juin ; l'agent représentant Valérie cherche une co-vacancière pour partir randonner à la montagne ; l'agent représentant Bernard cherche des co-vacanciers avec véhicule pour partir visiter les châteaux de la Loire. Les agents interagissent par appel d'offre pour constituer des groupes aux intérêts communs. Ce type de méthode est bien connu dans le domaine des Systèmes Multi-Agents [Davis and Smith, 1983]. Toutefois, ce type de méthode génère un nombre important de messages. Nous avons donc cherché à améliorer l'efficacité de la méthode de regroupement en réalisant une première étape de filtrage permettant de limiter le nombre d'interlocuteurs pour chaque agent. L'approche utilisée pour cette étape de filtrage est issue des travaux en apprentissage non supervisé dans lesquels des groupes homogènes sont recherchés.

Ainsi, l'approche proposée permet de bénéficier à la fois d'un traitement rapide d'une masse de données, regroupées par des techniques de clustering, et d'un traitement plus fin de regroupement au sein de chaque cluster par une approche multi-agent.

2 Travaux connexes

Le contexte applicatif de ces travaux concerne différentes thématiques de recherche, en particulier : le traitement de

1. https://www.economie.gouv.fr/entreprises/tourisme-seniors-chiffres

données à l'aide d'agents et la constitution de groupes en fonction de préférences.

2.1 Agents et exploration de données

Les agents sont couramment intégrés à des systèmes qui réalisent de l'exploration de données par le recours à diverses capacités, utilisées isolément ou en association. La mobilité par exemple, est une capacité utilisée en collecte de données distribuées, essentiellement des données qui concernent l'usage ou le contenu des réseaux, avec de nombreuses applications dans le domaine de la sécurité ou dans celui des réseaux sociaux [del Val et al., 2016]. Le traitement des données peut faire appel à des capacités de résolution distribuée de problèmes comprenant ou non de l'apprentissage [Warkentin et al., 2012], en faisant appel à de la coordination ou de la négociation basée sur des mécanismes de communication [Chihab et al., 2019]. Enfin l'interaction avec les utilisateurs de ces systèmes peut également intégrer des agents communicants, particulièrement lorsqu'il s'agit de systèmes d'aide à la décision ou de recommandation [Morais et al., 2012].

Parmi les systèmes qui réalisent du traitement de données à l'aide d'agents, certains sont spécifiquement dédiés à l'apprentissage non supervisé basé sur des agents ou des systèmes multi-agents [Bu et al., 2017, Guériau et al., 2018]. L'objectif du clustering est de diviser l'ensemble de données en groupes homogènes (les clusters) qui se distinguent les uns des autres. Ces méthodes semblent pertinentes relativement à notre problématique en ce qu'ils réalisent des regroupements au sein de la population initiale. Toutefois des adaptations sont nécessaires car :

— les distances utilisées dans les méthodes de clustering, qu'il s'agisse d'une distance Euclidienne, de Minkowski, de Chebyshev ou autre, sont toutes par essence symétriques : la distance de A à B est la même que celle de B à A. Or, ce n'est pas vrai dans le cas qui nous intéresse lorsque l'avis de A sur B peut différer de l'avis de B sur A.
— les personnes donnent une préférence quant au nombre souhaité de co-vacanciers. Ceci va se traduire par une contrainte sur le nombre d'éléments par cluster, contrainte absente des méthodes de clustering.

2.2 Constitution de groupes basés sur les préférences

L'objectif applicatif peut être recherché selon d'autres approches, en particulier via la thématique des appariements. Cette autre approche est abordée de façon complémentaire par nos partenaires dans le cadre du projet, voir par exemple les travaux [Morge and Nongaillard, 2017] qui réalisent des regroupements liés aux activités.

3 Approche multi-agent pour la constitution de groupes de vacanciers basés sur les préférences

L'approche proposée est une approche essentiellement décentralisée, basée sur un mécanisme d'appel d'offre entre agents représentant les préférences des candidats aux vacances. Ce type de mécanisme est très répandu dans le domaine des systèmes multi-agents, mais il génère un volume important de messages. Nous avons donc cherché à en améliorer l'efficacité en permettant aux agents de cibler plus précisément leurs interlocuteurs par une pré-sélection fondée sur les compatibilités entre vacanciers. L'approche utilisée pour cette étape de filtrage est issue des travaux en apprentissage non supervisé dans lesquels des groupes homogènes sont recherchés (clustering).
L'approche proposée comporte ainsi deux étapes consécutives : sélection d'interlocuteurs représentant des préférences compatibles ; interactions par appel d'offre au sein de chaque sous-groupe pour former les groupes finaux de co-vacanciers.

3.1 Etape 1 : filtrage des co-vacanciers potentiels

L'ensemble $\mathcal{C}$ des données comporte N candidats. Chaque candidat X est représenté par un ensemble d'attributs appartenant à trois catégories différentes : un profil personnel po_X, des préférences quant au projet de vacances ho_X et des souhaits quant au profil des co-vacanciers co_X :

$$X = (\mathrm{ho}_X, \mathrm{po}_X, \mathrm{co}_X)$$

L'objectif est de trouver des groupes de candidats qui sont des sous-ensembles $\mathcal{G}_i \subseteq \mathcal{C}^N$ tels qu'ils satisfont au mieux les préférences des candidats appartenant à ces groupes. Ces groupes ne forment pas forcément une partition de l'ensemble $\mathcal{C}$: il peut y avoir des candidats "isolés", qui n'appartiennent à aucun groupe.
Les candidats peuvent former un groupe sous les conditions suivantes : leurs projets de vacances sont compatibles ; leurs profils personnels sont compatibles.
Les projets et profils sont évalués selon une échelle de trois valeurs : -1 signifiant non souhaité, 0 signifiant l'indifférence et 1 signifiant souhaité. La compatibilité entre deux candidats est évaluée de façon normalisée dans $[0, 1]$, avec 1 pour totalement compatibles, 0 pour totalement incompatibles.
Considérant que chaque candidat ne donne pas la même importance à chaque critère de choix et qu'un candidat X peut être rejeté par Y alors que Y peut ne pas rejeter X, nous distinguons l'évaluation de X par Y notée $\mathrm{Eval}^Y(X)$ de l'évaluation $\mathrm{Eval}^X(Y)$ de Y par X. La compatilité Cp entre X et Y est ainsi évaluée sur la base du candidat le moins satisfait entre X et Y : $\mathrm{Cp}(X, Y) = \min\{\mathrm{Eval}^X(Y), \mathrm{Eval}^Y(X)\}$ L'évaluation $\mathrm{Eval}^X(Y)$ du point-de-vue de X est calculée à partir de l'évaluation de la compatibilité du projet de vacances de Y avec celui de

X et de la compatibilité du profil de Y avec celui des co-vacanciers recherchés par X.

Projet de vacances. Le projet de vacances d'un candidat X comporte cinq critères, concernant la destination, le temps, le coût, les activités et la taille du groupe. Pour chacun de ces critères h, l'évaluation de la compatibilité du projet de vacances de Y est définie sur la base du nombre de souhaits s^i_X de X non rejetés par Y :

$$\mathrm{Eval}^X_h(Y) = \frac{|\{s^i_X = 1 \,|\, s^i_Y \neq -1\}_i|}{|\{s^i_X = 1\}_i|}$$

Sur cette base, l'évaluation de la compatibilité du projet de vacances de Y du point-de-vue de X est une combinaison linéaire de ces évaluations liées aux critères h de destination, temps, coût, activités et taille du groupe, pondérées selon le point-de-vue de X :

$$\mathrm{Eval}^X_{\mathrm{ho}}(Y) = \sum_h w^X_h \, \mathrm{Eval}^X_h(Y)$$

Dans un premier temps, les valeurs des poids w^X_h sont considérées fournies par les candidats, tout en envisageant en perspective d'ajuster ceux-ci par apprentissage.

Profils personnels. Chaque profil de candidat po_X est composé de critères relatifs à son âge, son sexe, sa capacité de marche et le fait qu'il soit fumeur ou non. Ces mêmes critères mais avec éventuellement des valeurs différentes sont utilisés pour déterminer le profil co_X des co-vacanciers souhaités. L'évaluation de la compatibilité du profil de Y est définie sur la base des valeurs de critères représentant Y qui ne sont pas rejetées par X :

$$\mathrm{Eval}^X_{\mathrm{po}}(Y) \quad = \quad 1 \; - \; \frac{|\{\mathrm{po}^i_Y = 1 \,|\, \mathrm{co}^i_X \neq -1\}_i|}{|\{\mathrm{po}^i_Y = 1\}_i|}$$

Sur la base de ces évaluations relatives au projet de vacances et aux profils des candidats, l'évaluation $\mathrm{Eval}^X(Y)$ est calculée comme étant la moyenne géométrique de l'évaluation des projets de vacances de Y par X et de l'évaluation du profil de Y par X :
$\mathrm{Eval}^X(Y) = \sqrt{\mathrm{Eval}^X_{\mathrm{ho}}(Y) \times \mathrm{Eval}^X_{\mathrm{po}}(Y)}$.
Ainsi si l'une de ces évaluations est à 0 cela signifie qu'il existe une incompatibilité majeure et l'évaluation globale est nulle.
Enfin, la compatibilité $\mathrm{Cp}(X, Y)$ est déterminée par l'évaluation la plus faible entre X et Ycomme énoncé précédemment : $\mathrm{Cp}(X, Y) = min(\mathrm{Eval}^X(Y) \; , \; \mathrm{Eval}^Y(X))$.
Cette dernière mesure est symétrique et peut ainsi constituer une distance utilisable par une méthode de clustering.
La généralisation au groupe consiste en l'évaluation de la compatibilité entre les membres d'un groupe $\mathcal{G}_i$ déterminée également par l'évaluation la plus faible entre deux membres distincts de $\mathcal{G}_i$:

$$\mathrm{Cp}(\mathcal{G}_i) = \min_{X,Y \in \mathcal{G}_i, X \neq Y} \{\mathrm{Cp}(X, Y)\}$$

Cette première phase permet ainsi de fournir pour chaque candidat X une liste des autres candidats Y ordonnés par $\mathrm{Cp}(X, Y)$ et une méthode de calcul de la compatibilité au sein d'un groupe $\mathcal{G}_i$.

3.2 Etape 2 : création des groupes par appels d'offres

Chaque candidat est représenté par un agent qui porte les préférences du candidat. L'agent est chargé de trouver d'autres candidats susceptibles de convenir en tant que co-vacanciers du candidat qu'il représente. Le comportement de l'agent suit le mécanisme par appel d'offre : il propose son profil aux agents de son groupe et reçoit en retour les intérêts des autres agents envers lui-même.
Ainsi, chaque agent X construit la liste des agents Y tels que $\mathrm{Cp}(X, Y) > \gamma$ avec γ un seuil de compatibilité donné (0.5 par défaut), ainsi que les valeurs $\mathrm{Cp}(X, Y)$.
Parmi les réponses positives reçues, l'agent X tente de composer des groupes pour atteindre la taille de groupe souhaitée par le candidat qu'il représente. Par exemple, si la taille de groupe souhaitée est de 3, et qu'il a reçu des réponses positives de Y, Z et T, il propose à ceux-ci les groupes (X, Y, Z), (X, Y, T) et (X, Z, T). Les destinataires répondent en fonction de leurs propres valeurs de compatibilité entre eux. En effet, X ne connaissant pas $\mathrm{Cp}(Y, Z)$, il ne peut pas présager de la compatibilité du groupe $\mathcal{G} = (X, Y, Z)$: par exemple si X accepte les fumeurs alors que Y ne les accepte pas et Z est fumeur. Les agents partagent alors leurs préférences, à l'instar d'une résolution par Asynchronous Back-Tracking par exemple.
Lorsqu'un cluster est défini, un processus de vote est établi pour chaque projet de vacances.

3.3 Dynamicité de l'approche

L'arrivée de nouveaux candidats au fil de l'eau, est susceptible de modifier les solutions envisagées. A chaque nouveau candidat, des mesures de compatibilité sont calculées avec les candidats qui n'ont pas encore pris connaissance des propositions calculées auparavant. En effet, il ne s'agit pas de changer des groupes en cours de formation. L'étape de filtrage permet de sélectionner les seuls candidats "compatibles" avec le nouvel arrivant, permettant à l'agent qui le représente d'émettre son appel d'offre vers une sélection d'autres agents, et réciproquement seule une partie des agents contactera le nouvel agent.

3.4 Interaction avec les candidats-utilisateurs

Les personnes s'inscrivent à la plate-forme et y sont alors représentées en tant que "candidats aux vacances". L'acquisition d'un ensemble initial minimal de préférences est réalisée lors de l'inscription du candidat : je suis non fumeur, je veux bien aller à la mer, etc. Les autres préférences seront initialisées à un état "neutre", puis ajustées au fil des interactions.
Après les calculs de compatibilité, le candidat peut être

intégré à un ou plusieurs groupes potentiels. Cependant, pour chaque groupe, tant qu'aucun membre du groupe n'a pris connaissance et sélectionné la proposition de regroupement, le groupe est susceptible d'être dissout au profit de meilleures solutions.

Lorsqu'un candidat prend connaissance de propositions de groupes/projets de vacances, il en sélectionne certaines et en rejettent d'autres. Nous prévoyons également qu'il puisse les ré-ordonner. Cette sélection donne des informations au système pour ajuster les préférences liées au candidat.

Lorsque plusieurs candidats sélectionnent un projet/groupe commun, le système les met en contact afin qu'ils se rencontrent pour confirmer ou non la possibilité de séjour partagé. Cette dernière phase se passe hors support du système, si ce n'est que les candidats doivent informer le système de la constitution ou non du groupe.

3.5 Perspective

Une première base de données réelles, issues d'une collecte de préférences réalisée par des participants au projet [Delelis et al., 2017] a permis de valider les possibilités de clustering. Toutefois, les autres propositions présentées ici n'ont pas encore été évaluées. Nous envisageons deux types d'évaluation : une évaluation intrinsèque et une évaluation de l'utilisation.

De façon intrinsèque, l'objectif est d'évaluer l'intérêt de la phase de clustering relativement à un fonctionnement uniquement par appel d'offre. Les critères suivants pourrait permettre la comparaison entre les deux approches :

- mesures (moyenne, min et écart-type) de compatibilité dans les groupes formés ;
- passage à l'échelle : mesures relatives au nombre de messages échangés, à la vitesse de résolution

Concernant l'utilisation, l'objectif est d'évaluer les actions des utilisateurs du système. Tout d'abord, est-ce que les groupes recommandés par le système vont être validés par les utilisateurs ? Vont-ils aboutir ensuite à de réels regroupements pour des séjours de vacances ? En particulier, un risque provient du poids des préférences "non-dites" sur la sélection des co-vacanciers (avec les problèmes de privacté associés).

4 Conclusion

Les systèmes de recommandations peuvent permettre de mettre en relation des personnes exclues de certaines activités comme les vacances, du fait de leur isolement social. Dans le cadre du projet PartEns, l'objectif est de former des groupes de co-vacanciers tenant compte des préférences des personnes âgées participantes. Une approche multi-agent a été proposée, consistant en une étape de présélection suivie d'un mécanisme d'appel d'offres. Si des données réelles ont permis de vérifier sa faisabilité, l'évaluation de l'approche reste à compléter par un ensemble d'expérimentations.

Remerciements

PartEnS a été financée par la région Hauts de France dans le cadre de l'appel à projets Chercheur-Citoyen 2015-2018 et a bénéficié d'un soutien du groupe Humanis et de la Fondation Caisse d'Epargne

Références

[Bu et al., 2017] Bu, Z., Gao, G., Li, H.-J., and Cao, J. (2017). CAMAS : A cluster-aware multiagent system for attributed graph clustering. *Information Fusion*, 37 :10 – 21.

[Chihab et al., 2019] Chihab, Y., Bousbaa, Z., Jamali, H., and Bencharef, O. (2019). An approach based on heterogeneous multiagent system for stock market speculation. *Journal of Theoretical and Applied Information Technology*, 97(3) :835–845.

[Davis and Smith, 1983] Davis, R. and Smith, R. G. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1) :63–109.

[del Val et al., 2016] del Val, E., Martínez, C., and Botti, V. (2016). Analyzing users' activity in online social networks over time through a multi-agent framework. *Soft Computing*, 20(11) :4331–4345.

[Delelis et al., 2017] Delelis, G., Naveteur, J., Antoine, P., and Secq, Y. (2017). « partir ensemble séniors » (partens) : une plate-forme de repérage d'affinités et d'appariement pour promouvoir les vacances des séniors. In *Congrès International de Psychologie Sociale Appliquée (CIPSA)*, Lille, F.

[Guériau et al., 2018] Guériau, M., Armetta, F., Hassas, S., Billot, R., and El Faouzi, N. (2018). Apprentissage constructiviste à base de systèmes multiagents une application au problème complexe de la régulation coopérative du trafic. *Revue d'Intelligence Artificielle*, 32(2) :249–277.

[Morais et al., 2012] Morais, A. J., Oliveira, E., and Jorge, A. M. (2012). A multi-agent recommender system. In *Distributed Computing and Artificial Intelligence*, volume 151 of *Advances in Intelligent and Soft Computing*, pages 281–288. Springer Berlin Heidelberg.

[Morge and Nongaillard, 2017] Morge, M. and Nongaillard, A. (2017). Affectation distribuée d'individus à des activités avec des préférences additivement séparables. In Garbay, C. and Bonnet, G., editors, *Journées Francophones sur les Systèmes Multi-Agents*, pages 19–28, Caen. Cépaudès édition.

[Naveteur and Secq, 2018] Naveteur, J. and Secq, Y. e. (2018). Partir ensemble senior, rapport final. Technical report, LAMIH-CNRS8201, UPHF.

[Warkentin et al., 2012] Warkentin, M., Sugumaran, V., and Sainsbury, R. (2012). The role of intelligent agents and data mining in electronic partnership management. *Expert Systems with Applications*, 39(18) :13277–13288.

Robust Reinforcement Learning for Autonomous Driving

Y. Jaafra[1,2,3] J. L. Laurent [2] A. Deruyver [1] M. S. Naceur [3]

[1] ICube Laboratory, Université de Strasbourg, 300 bd Sébastien Brant, 67412 Illkirch, France
[2] Segula Technologies, Parc d'activité de Pissaloup, 8 avenue Jean d'Alembert, 78190 Trappes, France
[3] LTSIRS Laboratory, ENIT, 1002 Tunis, Tunisie

yasmina.jaafra@etu.unistra.fr

Abstract

Autonomous driving is still considered as an "unsolved problem" given its inherent important variability and that many processes associated with its development like vehicle control and scenes recognition remain open issues. Despite reinforcement learning algorithms have achieved notable results in games and some robotic manipulations, this technique has not been widely scaled up to the more challenging real world applications like autonomous driving. In this work, we propose a deep reinforcement learning (RL) algorithm embedding an actor critic architecture with multi-step returns to achieve a better robustness of the agent learning strategies when acting in complex and unstable environments. The experiment is conducted with CARLA simulator offering customizable and realistic urban driving conditions. The developed deep actor RL guided by a policy-evaluator critic distinctly surpasses the performance of a standard deep RL agent.

Keywords

Neural networks, Deep reinforcement learning, Actor-critic model, Autonomous driving, CARLA simulator.

1 Introduction

An important approach for goal-oriented optimization is reinforcement learning (RL) inspired from behaviorist psychology [24]. The frame of RL is an agent learning through interaction with its environment driven by an impact (reward) signal. The environment return reinforces the agent to select new actions improving learning process, hence the name of reinforcement learning [24]. RL algorithms have achieved notable results in many domains as games [16], [23] and advanced robotic manipulations [13], [15] beating human performance. However, standard RL strategies that randomly explore and learn faced problems lose efficiency and become computationally intractable when dealing with high-dimensional and complex environments[25].

Autonomous driving is one of the current highly challenging tasks that is still an "unsolved problem" more than one decade after the promising 2007 DARPA Urban Challenge [5]. The origin of its difficulty lies in the important variability inherent to the driving task (e.g. uncertainty of human behavior, diversity of driving styles, complexity of scene perception...).

Our key contribution in this paper consists in extending RL application to a complex and dynamic real-life task relatively more difficult than current RL benchmarks commonly used in RL research. Subsequently, we introduce an advantage actor-critic approach with multi-step returns for autonomous driving. This type of RL has demonstrated good convergence performance and faster learning in several applications which make it among the preferred RL algorithms [8]. The derived actor-critic algorithm consolidates the robustness of the agent learning strategy by using a temporal difference (TD) update to control returns and guide exploration. We implement a generalized version of TD learning that allows the agent to collect more data on the environment before computing the critic error approximation and adjusting the policy accordingly. Technically it consists in bootstrapping value states (sampled returns) over multiple time steps into the future and reuse the resulting TD error in policy evaluation.

The training and evaluation of the approach are conducted with the recent CARLA simulator [7]. Designed as a server-client system, where the server runs the simulation commands and renders the scene readings in return, CARLA is an interesting tool since physical autonomous urban driving generates major infrastructure costs and logistical difficulties. It particularly offers a realistic driving environment with challenging properties variability as weather conditions, illumination and density of cars and pedestrians. The experiment results reported through episodic average and cumulative rewards demonstrate a substantial performance enhancement of the RL agent guided by multi-step TD learning in a dynamic environment with several changing variables.

The remainder of this paper is organized as follows. In Sect. 2 we review the definition and the common application areas of actor-critic RL. Sect. 3 describes the proposed method focusing on the techniques used to improve robustness and fast converging learning. In Sect. 4 we

present CARLA simulator and the experiment steps and results to evaluate our model using this environment. Last, conclusions and open research direction are discussed in Sect. 5.

2 Related Work

Various types of RL algorithms have been introduced and are classified into three categories, actor, critic or actor-critic depending on whether they rely on a parameterized policy, a value function or a combination of both to predict actions [12]. In the actor-only methods, a gradient is generated to update the policy parameters in a direction of improvement [27]. Despite policy gradients offer tough convergence guarantees, they may suffer from high variance resulting in slow learning [3]. On the other hand, critic-only methods built on value function approximation, use TD learning and show lower variance of estimated returns [4]. However, they lack reliable guarantee of converging and reaching the real optimum [8].

Actor-critic methods combine the advantages of the two previous ones by inducting a repetitive cycle of policy evaluation and improvement. [2] is considered as the starting point that defined the basics of actor-critic algorithms commonly used in recent research. Since then, several algorithms have been developed with different directions of improvements. [26], introduced the Fuzzy Actor-Critic Reinforcement Learning Network (FACRLN), which involves one neural network to approximate both the actor and the critic. Based on the same strategy, [18] developed the Consolidated Actor-Critic Model (CACM). [11] used for the first time a natural gradient [1] for the policy updates in their actor-critic algorithm. [22] presented the Deterministic Policy Gradient algorithm (DPG) that assign a learned value estimate to train a deterministic policy. Recently, [17] proposed the Asynchronous Advantage Actor-Critic (A3C) algorithm where multiple agents operate in parallel allowing data decorrelation and learning experience diversity.

Despite that several actor-critic methods have been developed, most of them were tested on standard RL benchmarks. The latter generally include basic tasks with low-level complexity comparatively to real world applications, like cart-pole balancing [26], [11], maze problems [18], multi-armed bandit [22], Atari games [17], [9] and OpenAI Gym tasks [19], [15].

Our work contribution consists in extending actor-critic RL application to a very challenging task which is urban autonomous driving. The domain setting is particularly difficult to handle due to intricate and conflicting dynamics. Indeed, the driving agent must interact, in changing weather and lighting conditions and through a wide action space, with several actors that may behave unexpectedly, identify traffic rules and street lights, estimate appropriate speed and distance... Our approach, that will be detailed in the next section, incorporates an actor and a multi-step TD critic component to improve the stability of the RL method.

3 Advantage Actor Critic with multi-step returns

The RL task considered in this work is a Markov Decision Process (MDP) T_i defined according to the tuple $(S, A, p, r, \gamma, \rho_0, H)$ where S is the set of states, A is the set of actions, $p(s_{t+1}|s_t, a_t)$ is the state transition distribution predicting the probability to reach a state s_{t+1} in the next time step given current state and action, r is a reward function, γ is the discount factor, ρ_0 is the initial state distribution and H the horizon. Consider the sum of expected rewards (return) from a trajectory $\tau_{(0,H-1)} = (s_0, a_0, ..., s_{H-1}, a_{H-1}, s_H)$. A RL setting aims at learning a policy π of parameters θ (either deterministic or stochastic) that maps each state s to an optimal action a maximizing the return R of the trajectory.

$$R_t = r_{t+1} + \gamma R_{t+1} = \sum_{i=t}^{t+H-1} \gamma^{i-t} r_{i+1} \tag{1}$$

Following the discounted return expressed above, we can define a state value function $V(s) : S \rightarrow R$ and a state-action value function $Q(s, a) : A \times S \rightarrow R$ to measure, respectively, the current state and state-action returns estimated under policy π:

$$V(s_t) = \mathbb{E}[R_t|s_t = s] \tag{2}$$

$$Q(s_t, a_t) = \mathbb{E}[R_t|s_t = s, a_t = a] \tag{3}$$

In value-based RL algorithms such as Q-learning, a value function is approximated to select the best action according to the maximum value attributed to each state and action pair. On the other hand, policy-based methods directly optimize a parameterized policy without using a value function. They use instead gradient descents like in the family of REINFORCE algorithms [27] updating the policy parameters θ in the direction:

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t|a_t) R(t) \tag{4}$$

The main problem with policy based methods is that the score function R_t uses the averaged rewards calculated at the end of a trajectory which may lead to the inclusion of "bad" actions and hence slow learning. The solution provided in actor-critic framework is to replace the reward function R_t in the policy gradient (equation 4) with the action value function that will enable the agent to learn the long-term value of a state and therefore enhance its prediction decision:

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t|a_t) Q(s_t, a_t) \tag{5}$$

Then train a critic to approximate this value function parameterized with ω and update the model accordingly. At this point, we can conclude that an efficient way to derive an optimal control of policies is to evaluate them using

Figure 1: CARLA environments. Left: Clear Noon weather in Town 2. Right: Hard Rainy in Town 1.

approximated value functions. Hence, building accurate value function estimators results in better policy evaluation and faster learning.

TD learning combining Monte Carlo method and dynamic programming [24] has proved to be an effective way to calculate good approximations of value functions by allowing an efficient reuse of rewards during policy evaluation. It consists in taking an action according to the policy and bootstrapping the 1-step sampled return from the value function estimate resulting in the below 1-step TD target:

$$G_t = r_t + \gamma * V_t(s_{t+1}) \tag{6}$$

Given the last return estimation, we obtain the 1-step TD update rule that allows the adjustment of the value function according to the TD error δ_t with step size β:

$$V(s_t) = V(s_t) + \beta(\underbrace{r_t + \gamma V_t(s_{t+1}) - V(s_t)}_{\delta_t}) \tag{7}$$

At this level, the actor-critic algorithm still suffers from high variance. In order to reduce the variance of the policy gradient and stabilize learning, we can subtract a baseline function, e.g. the state value function, from the policy gradient. For that, we define the advantage function $A(s_t, a_t)$ which calculates the improvement in predicting an action compared to the average $V(s_t)$:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) \tag{8}$$

An approximation of the advantage function is required since it involves two value functions $Q(s_t, a_t)$ and $V(s_t)$. Therefore let's reformulate $A(s_t, a_t)$ as the difference between the expected future reward and the actual reward that the agent receives from the environment [10]:

$$A(s_t, a_t) = R(s_t, a_t) - V(s_t) \tag{9}$$

When used in the previous policy gradient (equation 5), this gives us the advantage of the actor policy gradient:

$$\Delta\theta = \alpha\nabla_\theta \log \pi_\theta(s_t|a_t)(G_t - V(s_t)) \tag{10}$$

We can subsequently assume that TD error is a good candidate to estimate the advantage function. Accordingly, we deduce the final actor policy gradient:

$$\Delta\theta = \alpha\nabla_\theta \log \pi_\theta(s_t|a_t)\delta_t \tag{11}$$

Given the complex nature of the autonomous urban driving task, we will use a generalized version of TD learning by extending the bootstrapping over multiple time steps into the future. Algorithmically, we will define configurable multi-step returns within the TD target. Hence, TD error becomes:

$$\delta_t = [\sum_{i=t}^{t+H-1} \gamma^{i-t} r_i] + \gamma^H V(s_{t+H}) - V(s_t) \tag{12}$$

Multi-step returns have been demonstrated to improve the performance of learning especially with the advent of deep RL [17]. Indeed, it allows the agent to gather more information on the environment before calculating the error in the critic estimates and updating the policy.

So far, we have a good theoretical basis to launch our agent. The experiments carried out by the application of this approach in the CARLA simulator will be presented in the next section.

4 Experiments

In this section we investigate the performance of an advantage actor-critic (A2C) algorithm embedding multi-step TD target updates on the challenging task of urban autonomous driving. The goal of our experimental evaluation is to demonstrate that the incorporation of a multi-step returns critic (MSRC) component in a deep RL framework consolidates the robustness of the agent by controlling and guiding its learning strategy. We expect a reduction of the actor gradient variance, an ascendant trend of episodic average returns and more generally a better performance comparatively to the case where the MSRC component is deactivated in the A2C algorithm.

4.1 The setup

Environment. We conduct the experiments using CARLA simulator for autonomous driving which provides an interesting interface allowing our RL agent to control a vehicle and interact with a dynamic environment. Comparatively

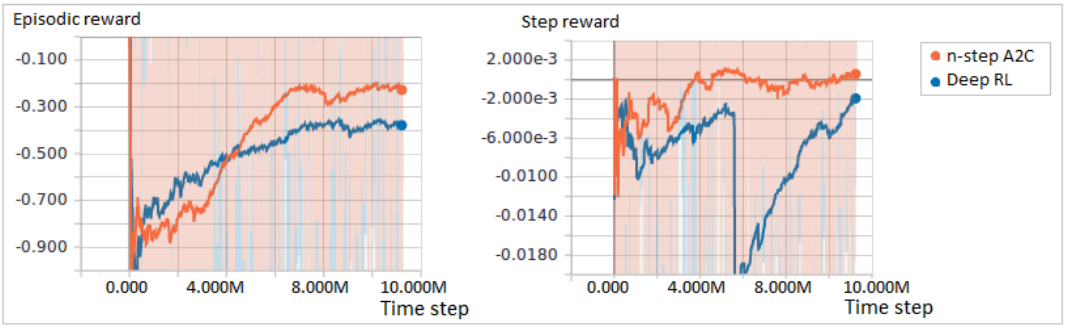


Figure 2: Training phase - Comparison between n-step A2C and standard deep RL performance trained in Town 2.

to existing platforms, CARLA offers a customizable and quite realistic urban driving conditions with a set of advanced features for controlling the vehicle and gathering the environment feedback. It is designed as a server-client system where the server implemented in Unreal Engine 4 (UE4) [1] runs the simulation commands and returns the scene readings. The client implemented in Python sends the agent predicted actions mapped as driving commands and receives the resulting simulation measures that will be interpreted as the agent rewards.

CARLA $3D$ environment consists of static objects as buildings, roads and vegetation and dynamic non-player characters, mainly pedestrians and vehicles. During training, we can episodically vary server settings as the traffic density (number of dynamic objects) and visual effects (weather and lightening conditions, sun position, cloudiness, precipitation...). Some examples of resulting environments are illustrated in figure 1.

Observation and action spaces. The agent interacts with the environment by generating actions and receiving observations over regular time steps. The action space selected for our experiments is built on the basis of three discrete driving instructions (steering, throttle, and brake) extended with some combinations in-between (turn left and accelerate/decelerate...). The observation space includes sensors outputs as color images produced by RGB cameras and derived depth and semantic segmentations. The second type of available observations consists of a range of measurements reporting the vehicle location (similarly to GPS) and speed, number of collisions, traffic rules and positioning of non-player dynamics characters.

Rewards. A crucial role is played by rewards in building driving policies as they orient the agent predictions. In order to further optimal learning, the reward is shaped as a weighted sum of measurements extracted from the observations space described in the previous paragraph. The idea is to compute a difference between the current and the previous measure of the selected observation then impact it positively or negatively on the aggregated reward. For example, the agent will get a reward if the distance to goal decreases and a penalty each time a collision or an intersection with the opposite lane is recorded.

Settings. The agent training follows a goal-directed navigation on straight roads from scratch. An episode is terminated when the target destination is reached or after a collision with a dynamic non-player character. The A2C networks are trained with 10 millions steps for 72 hours of simulated continuous driving. Motivated by the recent success achieved by deep RL in challenging domains [17], we use convolutional neural networks (CNN) to approximate both the value function of the critic and the actor policy where the parameters are represented by the deep network weights.

The CNN architectures consist of 4 convolutional layers, 3 max-pooling layers and one fully connected layer at the output. The discount factor is set as 0.9. We used 10-step rollouts, with initial learning rate set as $0,01$. Learning rate is linearly decreased to zero over the course of training. While training the approach, a stochastic gradient descent is operated each 10 time steps and the resulting policy model is stored only if its performance (accumulated rewards) exceeds the last retained model. The final stored model is then used in the test phase.

4.2 Comparative evaluation.

In the absence of various state-of-the-art works on the recent CARLA simulator, we choose to compare 2 versions of our algorithm: the original deep actor RL guided by the MSRC policy-evaluator versus a standard deep actor RL resulting from the deactivation of the MSRC component in the original algorithm. In fact the few available state-of-the-art results in CARLA environment [7], [14] report the percentage of successfully completed episodes. This type of quantitative evaluation doesn't meet our experiment objectives mentioned in the beginning of this section to evaluate and interpret the MSRC contribution in complex tasks like autonomous driving. Guided by the several works on RL strategies in different domains [17], [19], we selected episodic average and cumulative rewards metrics to evaluate our approach.

[1] https://www.unrealengine.com

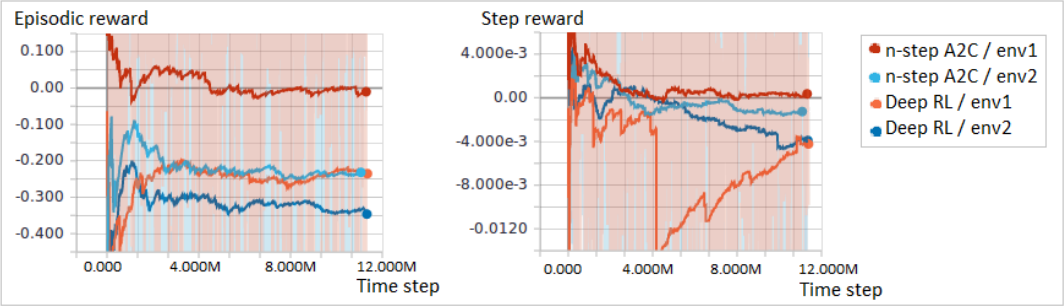


Figure 3: Testing Phase - Evaluation of n-step A2C and standard deep RL tested in 2 different environments env1 and env2. (Both have been trained in env1).

Figure 2 shows the generated reward in training phase. We use average episodic reward to describe the methods global performance and step reward to emphasize the predictions return variance. We can make few observations in this regard. In term of performance, our n-step A2C approach is dominant over almost all the 10000 training episodes confirming the efficiency of the RL strategy controlled by the MSRC. Furthermore, we noticed that regarding the best retained models, the A2C stored just few models (5) in the 2000 first episodes, then this number drastically increased to 100 retained models in the remaining 8000 episodes. This means that our method early achieved the exploration phase and moved to exploitation from the training level of 2000 episodes. On the other hand, the standard deep RL totalized only 10 best models over the training phase reflecting the weak efficiency of a random strategy to solve a very complex and challenging problem like autonomous driving. A last visual interpretation that we can deduce from the step reward graph is that the variance of A2C predictions is significantly reduced relatively to the standard deep RL confirming the TD learning contribution in accomplishing a faster learning.

Figure 3 recaps the testing phase evaluation following two different scenarios. First, the testing was conducted in the same environment and conditions as the training: Town 2 and Clear Noon weather (env1). From the episodic reward graph we can observe that our approach substantially outperforms the standard deep RL which means that training with multi-step returns critic leads to more efficient RL models. In the second scenario, both methods agents are tested in a different environment than training: Town 1 and in hard rainy conditions (env2). The n-step A2C is still more competitive than the standard deep RL showing superior generalization capabilities in the new unseen setting. Nevertheless, its performance has decreased in the second test scenario reflecting a certain fragility to changing environment. On the other side, the standard deep RL is still showing higher prediction return variance in the step reward graph confirming training phase conclusions.

5 Conclusion

In this paper we addressed the limits of RL algorithms in solving very complex tasks within dynamic environments. Combining both actor and critic methods advantages, the proposed approach implemented a continuous process of policy assessment and improvement using multi-step TD learning. The application domain selected for our work is the challenging task of urban autonomous driving. Compared to common RL research benchmarks, It implies very conflicting dynamics, changing conditions and unexpected non-player characters behaviors.

Our deep actor-critic algorithm was evaluated with CARLA simulator for autonomous driving, a recent server-client system offering a customizable and quite realistic urban driving conditions with a set of advanced features for controlling the vehicle and gathering the environment feedback. It has generated higher returns than a standard deep RL and demonstrated a more steady performance evolution across training and testing episodes reflecting faster and more efficient learning capabilities. On the other side, the results showed a certain vulnerability of the approach when raising the environment variability and testing on unseen cases.

In future work, we will tackle the issue of non-stationary environments impact on RL methods robustness as a multi-task learning problem [6]. In such context, we will explore recently applied concepts and methodologies such as novel adaptive dynamic programming (ADP) approaches, context-aware and meta-learning strategies. The latter is currently attracting a keen research interest and particularly achieving promising advances in designing generalizable and fast adapting RL algorithms [21], [20].

References

[1] S. Amari and S. C. Douglas, Why natural gradient?, *ICASSP*, IEEE, pp. 1213-1216, 1998.

[2] A. G. Barto and R. S. Sutton and C. W. Anderson, Neuronlike Adaptive Elements That Can Solve Diffi-

cult Learning Control Problems, *Artificial Neural Networks*, pp. 81-93, 1990.

[3] H. R. Berenji and D. Vengerov, A convergent actor-critic-based FRL algorithm with application to power management of wireless transmitters, *IEEE Transactions on Fuzzy Systems*, Vol. 4, pp. 478–485, 2003.

[4] J. A. Boyan, Technical update: Least-squares temporal difference learning, *Machine Learning*, pp.233-246, 2002.

[5] M. Buehler and K. Iagnemma and S. Singh, The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, *Springer Publishing Company, Incorporated*, 2009.

[6] R. Caruana, Learning to Learn, *Multitask Learning*, Kluwer Academic Publishers, pp. 95-133, 1998.

[7] A. Dosovitskiy and G. Ros and F. Codevilla and A. Lopez and V. Koltun, CARLA: An Open Urban Driving Simulator, *Proceedings of the 1st Annual Conference on Robot Learning*, Vol. 78, pp. 1-16, 2017.

[8] I. Grondman and L. Busoniu and G. A. D. Lopes and R. Babuska, A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients, *Trans. Sys. Man Cyber Part C*, Vol. 42, pp. 1291-1307, 2012.

[9] A. Gruslys and M. G. Azar and M. G. Bellemare and R. Munos, The Reactor: A Sample-Efficient Actor-Critic Architecture, *ICLR*, 2018.

[10] N. Heess and D. Silver and Y. W. Teh, Actor-Critic Reinforcement Learning with Energy-Based Policies, *Proceedings of the Tenth European Workshop on Reinforcement Learning*, Vol. 24, pp. 45-58, 2013.

[11] P. Jan and V. Sethu and S. Stefan, Reinforcement learning for humanoid robotics, *IEEE-RAS International Conference on Humanoid Robots (Humanoids2003)*, Karlsruhe, Germany, 2003.

[12] V. R. Konda and J. N. Tsitsiklis, On Actor-Critic Algorithms, *SIAM J. Control Optim.*, Vol. 42, pp. 1143-1166, 2003.

[13] S. Levine and C. Finn and T. Darrell and P. Abbeel, End-to-end Training of Deep Visuomotor Policies, *J. Mach. Learn. Res.*, Vol. 17, pp. 1334-1373, 2016.

[14] X. Liang and T. Wang and L. Yang and E. Xing, CIRL: Controllable Imitative Reinforcement Learning for Vision-Based Self-driving, *Computer Vision - ECCV 2018 - 15th European Conference, Part VII*, pp. 604–620, 2018.

[15] T. P. Lillicrap and J. J. Hunt and A. Pritzel and N. Heess and T. Erez and Y. Tassa and D. Silver and D. Wierstra, Continuous control with deep reinforcement learning, *ICLR*, 2016.

[16] V. Mnih and K. Kavukcuoglu and D. Silver and A. A. Rusu and J. Veness and M. G. Bellemare and A. Graves and M. Riedmiller and A. K. Fidjeland and g. Ostrovski and S. Petersen and C. Beattie and A. Sadik and I. Antonoglou and H. King and D. Kumaran and D. Wierstra and S. Legg and D. Hassabis, Human-level control through deep reinforcement learning, *Nature*, Vol. 518, pp. 529-533, 2015.

[17] V. Mnih and A. P. Badia and M. Mirza and A. Graves and T. Lillicrap and T. Harley and D. Silver and K. Kavukcuoglu, Asynchronous Methods for Deep Reinforcement Learning, *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48, pp. 1928-1937, 2016.

[18] C. Niedzwiedz and I. Elhanany and Z. Liu and S. Livingston, A Consolidated Actor-Critic Model with Function Approximation for High-Dimensional POMDPs, *AAAI 2008 Workshop for Advancement in POMDP*, pp. 37–42, 2008.

[19] S. Parisi and V. Tangkaratt and J. Peters and M. E. Khan, TD-regularized actor-critic methods, *Machine Learning*, 2019.

[20] R. Sachin and L. Hugo, Optimization as a model for few-shot learning, *In International Conference on Learning Representations (ICLR)*, 2017.

[21] A. Santoro and S. Bartunov and M. Botvinick and D. Wierstra and T. Lillicrap, Meta-Learning with Memory-Augmented Neural Networks, *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48, pp. 1842-1850, 2016.

[22] D. Silver and G. Lever and N. Heess and T. Degris and D. Wierstra and M. Riedmiller, Deterministic Policy Gradient Algorithms, *Proceedings of the 31st International Conference on International Conference on Machine Learning*, Vol. 32, pp. 387-395, 2014.

[23] D. Silver and A. Huang and C. J. Maddison and A. Guez and L. Sifre and G. Driessche and J. Schrittwieser and I. Antonoglou and V. Panneershelvam and M. Lanctot and S. Dieleman and D. Grewe and J. Nham and N. Kalchbrenner and I. Sutskever and T. Lillicrap and M. Leach and K. Kavukcuoglu and T. Graepel and D. Hassabis, Mastering the Game of Go with Deep Neural Networks and Tree Search, *Nature*, Vol. 529, pp. 484-489, 2016.

[24] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, *The MIT Press*, 2018.

[25] N. Wahlström and T. B. Schon and M. P. Deisenroth, From Pixels to Torques: Policy Learning with Deep Dynamical Models, *Deep Learning Workshop at the 32nd International Conference on Machine Learning*, France,2015.

[26] X. S. Wang and Y. H. Cheng and J. Q. Yi, A Fuzzy Actor-Critic Reinforcement Learning Network, *Inf. Sci.*, Vol. 177, pp. 3764-3781, 2007.

[27] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learning*, pp. 229–256, 1992.

Synchronisation d'horloge dans un système multi-agents

M. Limame[1] J. Henriet[2] C. Lang[2] N. Marilleau[1]

[1] IRD/UMMISCO – {prenom.nom@ird.fr}

[2] Univ. Bourgogne Franche-Comté FEMTO-ST Institute, CNRS {prenom.nom@univ-fcomte.fr}

Résumé

Nous présentons dans cette publication un algorithme permettant au sein d'un système multi agents (SMA) d'instaurer une cohérence au niveau des données recueillies par chacun des agents à travers un nouveau mode de synchronisation. Nous nous focalisons essentiellement sur une flotte de drones comme application de l'algorithme de synchronisation. Notre approche est fondée sur l'assimilation de données.

Mots-clés : Synchronisation, Cohérence des données, assimilation de données, systèmes multi-agents.

Abstract

We present in this publication an algorithm allowing within a multi-agent system (SMA) to establish coherence at the level of the data collected by each agent through a new synchronization mode. We focus essentially on a fleet of drones as application of the synchronization algorithm. Our approach is based on data assimilation.

Keywords: Synchronization, data consistency, data assimilation, multi-agent systems.

1 Introduction

Un système distribué est constitué d'un ensemble d'entités autonomes interconnectées, pouvant communiquer ensemble et dont chacune dispose d'une horloge locale. Néanmoins, il est souvent nécessaire que ces entités obtiennent une notion commune du temps via une synchronisation. En effet, les systèmes distribués sont majoritairement synchronisés afin d'assurer le service pour lesquels ils ont été conçus, notamment dans dans le domaine de supervision utilisant des systèmes distribuées avec entités mobiles.

1.1 L'algorithme de synchronisation un domaine exploré

Sur la base des horloges physiques, trois orientations majeures sont proposées dans la littérature pour la synchronisation :

(i) les architectures centralisées dans lesquelles il existe une hiérarchie distinguant un noeud par une unique horloge permettant de synchroniser l'ensemble du système distribué. Comme détaillé dans [1], parmi les techniques de synchronisation utilisant cette architecture, nous pouvons citer le système de positionnement global (GPS);

(ii) les architectures distribuées de synchronisation dans lesquelles il y a réplication d'horloge sur l'ensemble des nœuds du système ce qui rend cette architecture non hiérarchique puisque donnant à chaque agent la capacité de calculer l'horloge. Parmi les techniques de synchronisation utilisant cette architecture, nous pouvons citer l'algorithme de Cristian dans lequel, comme présenté dans [2], un agent du système envoie une demande au serveur de temps pour recevoir l'horloge actuelle. Lorsqu'il reçoit la réponse, il trouve le délai de transmission (délai entre l'envoi de la demande et la réception de la réponse), le divise par deux et l'ajoute au délai reçu du serveur. Une limite de cet algorithme se situe dans ses implémentations qui se basent sur un serveur unique, le rendant impropre à une utilisation dans les applications distribuées où la redondance peut s'avérer critique. Il existe aussi l'algorithme de Berkeley qui, comme précisé dans [3], contient un serveur temps actif qui interroge périodiquement chaque agent du système distribué pour lui demander l'heure. Sur la base des réponses, il calcule une durée moyenne et demande à tous les agents du système d'avancer leurs horloges vers la nouvelle

heure ou de ralentir leur horloge jusqu'à ce qu'une réduction spécifiée soit atteinte. Avec cette approche il y a un risque fort d'avoir un goulot d'étranglement au niveau du serveur temps.

(iii) les architectures distribuées hybrides de synchronisation dans lesquelles il y a réplication d'horloge uniquement sur certains nœuds du système ce qui fait que cette architecture est un mixte des deux architectures (i) et (ii). Parmi les techniques de synchronisation pouvant être appliquées à cette architecture, nous pouvons citer le protocole Network Time Protocol (NTP) qui, selon [2], utilise un système hiérarchique stratifié de sources de temps se basant sur l'UTC comme temps de référence. Lorsqu'un serveur est défini comme horloge maître, un message est envoyé à tous les esclaves (clients) pour synchroniser l'horloge. Ensuite, les esclaves calculent leur heure locale et la dérive de l'horloge maître. Mais le problème dans cette approche est qu'il existe un délai de propagation. Le décalage est égal à la différence entre l'horloge de l'esclave et l'horloge du serveur maître.

Il existe une approche alternative à la synchronisation d'horloges physiques qui se base sur le concept de l'horloge logique utilisant des algorithmes de synchronisation. Une horloge logique est un dispositif logiciel qui sert à établir et mesurer une notion de temps établie selon la relation de causalité arrivé-avant dans un système réparti asynchrone. Différents méthodes de synchronisation d'horloges logiques existent. Parmi ces méthodes : l'horloge de Lamport qui, comme détaillé dans [4], attribue une horloge logique ou estampille à tous les événements d'un système distribué de manière à ce que si un événement E1 précède un événement E2 passé sur un même agent du système, alors H(E1) < H(E2). Néanmoins, quand deux événements sont concurrents, on ne peut rien conclure quant à leurs horloges logiques respectives;la seule certitude est que Si H(E1) = H(E2) alors E1 et E2 sont concurrents. Il y a aussi l'horloge de Mattern avec laquelle chaque agent *e* possède un vecteur d'entiers appelé estampille dans lequel chaque composant estampille[i] est l'estimation par *e* de la valeur de l'horloge de Lamport de l'agent *i*. En particulier, estampille[*e*] est exactement l'horloge de Lamport de *e*. Les horloges de Mattern donnent une information plus précise que les horloges logiques de lamport pour un coût plus élevé en mémoire.

1.2 Les limites des algorithmes de synchronisation dans les systèmes autonomes

Quelle que soit la technique de synchronisation d'horloges physiques utilisée, une synchronisation ne peut être obtenue qu'au travers d'une approche algorithmique évoluée et de protocoles de synchronisation dont la complexité est intimement liée à la précision souhaitée de l'horloge. Plus le degré de précision est important plus le système a besoin de capacités énergétique et calculatoire pour : (i) exécuter les algorithmes ; (ii) communiquer en vue d'échanger de l'information d'horodatage ; (iii) et utiliser les horloges physique. Ceci constitue un véritable frein pour un usage dans les systèmes embarqués où l'énergie, les communications et la puissance de calcul sont limitées. En conséquence, la synchronisation d'horloges physiques est inadaptée pour un système multi-agents mobile riche en capteurs à l'instar d'une flotte de drones.

Quelle que soit la technique de synchronisation d'horloges logiques utilisée, elles se caractérisent toutes par un coût élevé en mémoire pour pouvoir gérer la logique d'ordonnancement des événements survenant dans un système distribué. En conséquence, la synchronisation d'horloges logiques constitue une difficulté dans un système multi-agents mobile riche en capteurs dont la capacité en mémoire est limitée.

1.3 Tirer parti des concepts de connaissance dans les SMA pour synchroniser des entités autonomes

D'une manière générale, en dehors des domaines d'application critiques, la synchronisation oeuvre pour que les agents du système multi-agents soient en accord et puissent prendre des décisions cohérentes en se basant sur des ressources et événements du système. En conséquence, la synchronisation des horloges logiques est nécessaire pour garantir le bon fonctionnement du système et une synchronisation des horloges physiques n'est pas indispensable.

Les méthodes de synchronisation d'horloges physiques et logiques étant inadaptées (non optimisées) pour un SMA mobile, nous proposons dans cet article de décrire une nouvelle approche de synchronisation respectant l'architecture de synchronisation

présentée dans la partie 1.1, paragraphe (ii) et s'appuyant sur le principe d'ordonnancement des événements et faisant appel à la technique d'assimilation de données. Dans ce qui suit, nous faisons d'abord un état des lieux des mécanismes de synchronisation dans les systèmes distribués pour passer ensuite à une description d'une nouvelle approche synchronisation appelée SMASDEV.

2 Nouvelle approche de synchronisation

2.1 Principe

La nouvelle approche apportée s'inspire de l'approche de Lamport basée sur les événements pouvant survenir dans un SMA sauf qu'elle ne s'intéresse pas à leurs ordres de survenance mais elle s'intéresse plutôt au contenu d'un événement et précisément à la donnée qui lui est associée.

N'étant pas gourmande en ressources, nous utiliserons aux techniques d'assimilation des données pour les analyser et les ré-organiser dans un chronographe. Ainsi, cette nouvelle approche a pour objectif d'instaurer une cohérence globale au niveau d'un SMA en permettant à chaque agent de rétablir un ordre chronologique des données qu'il reçoit des autres agents, sur la base de ses connaissances et s ans faire appel à une horloge (physique ou logique).

Le principe consiste à ce que chaque agent du système enregistre en mémoire sa perception personnelle de l'évolution d'une donnée qu'il recueille à une fréquence donnée par rapport à son horloge locale ce qui va lui permettre, grâce au principe de l'assimilation, de prévoir son évolution dans le futur. Pour cela, l'agent doit disposer du modèle d'évolution en adéquation avec la donnée en question. Pour passer d'une perception personnelle vers une perception globale, chaque agent du système doit, dans un premier temps, demander aux autres agents de lui transmettre la donnée qu'ils ont recueillie puis dans un deuxième temps il doit les positionner par rapport à sa perception personnelle. Ainsi l'agent sera en mesure d'identifier le positionnement de l'ensemble des données reçues de la part des autres agents par rapport à son horloge. En conséquence, les agents du système seront synchrones et en phase par rapport à l'évolution de la donnée dans le temps.

2.2 Algorithme SMASDEV

Soit :

- A^d un agent distant
- A^l l'agent local
- H^{Ad} horloge de l'agent distant (A^d)
- H^{Al} horloge de l'agent local (A^l)
- t^{ε} erre ur maximale entre deux horloges du système
- $M^{Ad} = (p_1, p_2,\ldots,p_i,\ldots, p_n)$ une mesure réelle de l'agent A^d des paramètres p_i à la date t selon l'horloge H^{Ad} (t^{est} selon l'horloge H^{Al}, valeur estimée après exécution de l'algorithme)
- $M^{est} = (p^{est}_1, p^{est}_2,\ldots,p^{est}_i,\ldots, p^{est}_n)$ une mesure estimée de l'agent A^l des paramètres p_i à la date t selon l'horloge H^{Al}
- $K^{Al} = (k_1, k_2,\ldots, k_i,\ldots, k_n)$ un tuple ordonné de connaissances de l'agent local (A^l), tel que $k_i = \{(p^{k}_i,t_1), (p^{k}_i,t_2),\ldots, (p^{k}_i,t_j),\ldots, (p^{k}_i,t_m))\}$ et (p^{k}_i,t_j) une mesure admise du paramètre p_i à une date t_j selon l'horloge H^{Al}
- $F = (f_1, f_2,\ldots, f_i,\ldots, f_n)$ un tuple ordonnées de fonctions d'estimation $f_i(k_i,t)$ permettant d'estimer la valeur p^{est}_i d'un paramètre p_i, à un instant t selon une base de connaissances k_i et l'horloge H^{Al}
- f^{eval} une fonction d'évaluation multicritère $f^{eval}(M^{est}, M^{Ad})$ qui permet de mesurer la distance entre un tuple de paramètres estimés $M^{est}{}_t$ et un tuple de paramètres mesurés M^{Ad}

Avec :

- $t_{\varepsilon} > | t_{Ad} - t_{Al} |$

L'algorithme de la fonction baptisé SMASDEV (Multi Agent System Synchronisation based on Data Evolution) est décrit ci-dessous :

```
var eval ← +inf
M^est ← tableau(n)
var t^est ← 0
POUR TOUT t_id ∈ [t_Al - t_ε, t_Al + t_ε] FAIRE
        var M^temp ← tableau(n)
        POUR TOUT i ∈ [1,n] FAIRE
                M^temp[i] ← F[i](K^Al[i], t_id)
        FIN POUR
        var eval^temp ← f^eval(M^temp, M^Ad)
        SI eval^temp < eval ALORS
                eval ← eval^temp
                M^est ← M^temp
                t^est ← t_id
        FIN SI
FIN POUR
RETOURNE t^est
```

Algorithme SMASDEV

Dans un premier temps, l'algorithme définit la plage de temps sur laquelle va se baser la fonction d'estimation $f_i(k_i,t)$ et ceci en prenant en compte le taux d'erreur maximale entre deux horloges en l'occurrence entre l'horloge de l'agent distant et l'horloge de l'agent local. Ce taux d'erreur varie en fonction des caractéristiques techniques des horloges utilisées.

Dans un deuxième temps, après avoir obtenu un ensemble de tuples de données estimées par rapport au contexte de l'agent local, l'objectif de l'algorithme est d'identifier le tuple dont les données estimées sont les plus proches par rapport aux tuples des données mesurées par l'agent distant d'où l'utilisation de la fonction f^{eval} qui sert pour calculer les différences entre les tuples de données estimés et les tuples de données mesurées. La variable eval est utilisée pour stocker le tuple de données estimées dont la différence avec le tuple de données mesurées est la moins importante. L'obtention de ce tuple permettra d'identifier l'instant t^{est}. Nous pouvons en conséquence identifier le tuple de données le plus récent en faisant une comparaison entre les instants t^{est} selon l'horloge H^{Al} et t selon l'horloge H^{Ad}. Le tuple dont l'instant t de prise est le plus important correspond au tuple dont les données sont les plus récentes.

La fonction $f_i(k_i,t)$ permettant d'obtenir un tuple de données estimées au cours d'une plage temps est une fonction qui dépend de la nature de la donnée et précisément de son modèle d'évolution dans le temps. Par exemple dans le cas où la donnée objet de l'estimation porte sur la température la fonction d'estimation peut se baser sur une fonction affine. En revanche, une fonction estimant la densité des particules dans l'air suit un modèle plus complexe.

3 Conclusion & Perspectives

Notre objectif est d'introduire un protocole basé sur le contenu des observations d'un agent pour rétablir une chronologie des événements observés et valeurs recueillies. Cette nouvelle approche permettrait en particulier d'apporter une solution à la synchronisation des horloges d'un système multi-agents tels qu'une flotte de drones de surveillance d'un territoire. En perspective, nous souhaitons également explorer la possibilité d'introduire des outils de raisonnement par analogie ou de classification proposés par le paradigme de l'intelligence artificielle dans ce nouveau protocole.

Références

[1] Hofmann¬Wellenhof, B., Lichtenegger, H., and Collins, J., 2001, "Global Positioning System: Theory and Practice," Text Book, Fifth edition

[2] M. Leela, D. Manoj Kumar, G. Bhavana, 2018, Clock Synchronisation in Distributed Systems: A Review

[3] D.Adithya Chandra Varma, Praveen Kumar Reddy.M, Prof.Gopinath, 2013, Performance Comparison of Physical Clock Synchronization Algorithms

[4] Lamport, L., 1978, Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7):558–565

LinTO : Assistant vocal open-source respectueux des données personnelles pour les réunions d'entreprise

Jean-Pierre Lorré[1], Isabelle Ferrané[2], Francisco Madrigal[3], Michalis Vazirgiannis[4], Christophe Bourguignat[5]

[1]LINAGORA
[2]IRIT
[3]LAAS-CNRS
[4]Laboratoire d'Informatique de l'Ecole Polytechnique
[5]Zelros

jplorre@linagora.com, isabelle.ferrane@irit.fr, jfmadrig@laas.fr, mvazirg@lix.polytechnique.fr, christophe.bourguignat@zelros.com

Résumé

Cet article présente les premiers résultats du projet de recherche PIA Grands Défis du Numérique LinTO[1] dont l'objectif est de réaliser un assistant vocal permettant d'aider les employés d'une entreprise en particulier lors des réunions. Dispositif interactif doté de micros, écran et caméra 360°, il permet de piloter la salle, d'interroger le système d'information, d'aider à l'animation de la réunion et propose un environnement d'aide à la rédaction du compte rendu. Diffusé suivant un modèle ouvert respectueux des données personnelles, LinTO[2] est le premier assistant d'entreprise open-source conçu pour favoriser la prise en compte des exigences du RGPD.

Mots Clef

Assistant Vocal Conversationnel, Intelligence Artificielle, Reconnaissance Automatique de la Parole, Traitement du Langage Naturel, Reconnaissance de personnes

Abstract

This paper presents the first results of the PIA « Grands Défis du Numérique » research project LinTO. The goal of this project is to develop a conversational assistant to help the company's employees, particularly during meetings. LinTO is an interactive device equipped with microphones, a screen and a 360° camera, which allows to control the room, query company's information system, helps facilitate the meeting and provides an environment to aid minute writing. Distributed according to an open model that respects private data LinTO is the first open-source enterprise's assistant designed to comply with the GDPR requirements.

Keywords

Conversational Voice Assistant, Artificial Intelligence, Automatic Speech Recognition, Natural Language Processing, People Recognition.

1 Introduction

Nous assistons à une prolifération d'outils dans le domaine des assistants personnels. Présents sur les téléphones portables, ils aident à écrire des sms, des mails, prendre des rendez-vous ; associés aux outils de communication collaboratifs ils jouent le rôle d'un être humain connecté pour aider. Le marché associé à ces technologies est identifié par de nombreux analystes, certains l'évaluant à 5,1 milliards de dollars en 2022 [5] en croissance annuelle de 32% entre 2015 et 2022.

LinTO est un assistant conversationnel vocal offrant des fonctionnalités avancées et adaptées au milieu professionnel. À la différence des assistants personnels grand public, LinTO est conçu pour s'interfacer avec une plateforme de productivité (telle que OpenPaaS, Office 365 ou G Suite), les autres briques du système d'information de l'entreprise ainsi que des services externes. Il dispose ainsi de fonctions propres à l'assistance dans un contexte de travail : gestion de réunion, compte-rendu, accès à des données pour la prise de décision, etc.

Remarquons que toutes les entreprises des "GAFAM" (Google, Amazon, Facebook, Apple et Microsoft) proposent une offre d'assistant intelligent. Les acteurs désirant se positionner sur ce marché se retrouvent souvent face à un choix cornélien: ou bien entrer en compétition R&D avec ces acteurs possédant des moyens bien supérieurs, ou bien utiliser et dépendre des services de leurs concurrents directs tout en les enrichissant de nouvelles données. Une autre voie est donc nécessaire. Le modèle open-source a démontré dans d'autres domaines qu'il était possible à une communauté de "petits" acteurs collaborant et mutualisant leurs connaissances de mettre à mal la situation hégémonique d'un acteur dominant un marché.

L'objectif de ce papier, est de présenter les premiers résultats du projet collaboratif LinTO soutenu par le Programme des Investissements d'Avenir.

2 Assistant LinTO

Le dispositif matériel "LinTO" inclut carte CPU de type Raspberry Pi, écran tactile, haut-parleurs, matrice de microphones et suivant les configurations, une caméra 360°. Il est complété par une plateforme logicielle d'assistant conversationnel supportant différentes modalités d'interaction en fonction des besoins et des configurations matérielles.

Deux groupes de fonctionnalités complémentaires sont étudiés :

- un ensemble de fonctionnalités de type "assistant personnel" pour aider l'utilisateur à accéder à l'information qu'elle soit interne au système

[1] Projet soutenu par Bpifrance N° P169201
[2] https://linto.ai/

d'information (mail, rendez-vous, document, etc.) ou externe par l'intermédiaire de la connexion avec un service disponible sur Internet ;
- un second ensemble de fonctionnalités dédiées au contexte de la réunion, incluant des mécanismes d'aide à la modération de réunion, de reconnaissance de participants et de suivi de leurs échanges, de recommandations contextuelles et de génération semi-automatique de résumé. C'est dans ce groupe que sont situés les principaux verrous du projet, en particulier ceux liés à la détection de plusieurs personnes (contexte multi-participants), à la définition d'une signature audio-visuelle (caractériser sans chercher à identifier) et à la fusion d'indicateurs issus d'analyses bas niveau (sonore/visuel) [2] pour enrichir les traitements sur l'analyse des interactions.

Compte tenu de l'usage dans des entreprises disposant d'un nombre élevé de salles de réunions, un outil d'administration est également proposé afin de gérer à distance la flotte des LinTO.

Le contexte professionnel induit des contraintes. Outre la problématique de la sécurisation des données confidentielles, l'accès aux informations pertinentes est souvent conditionné par l'intégration avec les systèmes et processus existants déjà mis en place au sein de l'entreprise. Enfin, dans un contexte professionnel les informations utiles à l'assistant ne sont pas nécessairement regroupées autour d'un seul individu, mais peuvent nécessiter de croiser celles de plusieurs participants (par exemple la sélection de date pour une réunion nécessite l'accès aux calendriers des différents participants). Pour obtenir une expérience utilisateur correcte dans un tel cadre, il faut donc être capable d'offrir des performances satisfaisantes en termes de reconnaissance de la parole et de traitement du langage naturel (justesse de la transcription, temps de réponse et montée en charge), mais il est également important de pouvoir s'interfacer avec une plateforme d'entreprise capable de fournir les informations nécessaires à ces fonctionnalités avancées.

3 Reconnaissance de la parole en réunion

Malgré les grandes avancées dans le domaine, la reconnaissance de la parole reste un défi technique important [3]. Bien que cette dernière soit désormais suffisamment bien maîtrisée pour donner des résultats satisfaisants suivant le type d'usage (dialogue de commande ou dictée vocale), son application au dialogue entre humains dans le cadre d'une réunion reste un verrou. Notons qu'il n'existe pas de solution open-source disponible pour le français.

LINAGORA développe un moteur de reconnaissance automatique de la parole (ASR, Automatic Speech Recognition) nommé LinSTT. Ce dernier est basé sur la boîte à outils open-source Kaldi [9].

Les deux types d'usages proposés précédemment reposent sur deux modes de fonctionnement du composant ASR : (i) mode "commande", il s'agit alors de reconnaître une commande et ses attributs dans un contexte bien défini ; (ii) mode "large vocabulaire" où le dispositif doit être à même de transcrire un flux de parole concernant des sujets ouverts.

4 Mode commande

La chaîne de traitements mise en œuvre par la plate-forme LinTO en mode commande est illustrée ci-dessous. L'outil traite en permanence le son provenant d'un ou plusieurs microphones, se déclenche à l'énoncé du « hotword », détecte l'activité vocale, repère l'énoncé (utterance), extrait de la transcription les principaux concepts (intention, entités) qui permettent de constituer l'action à exécuter (skill).

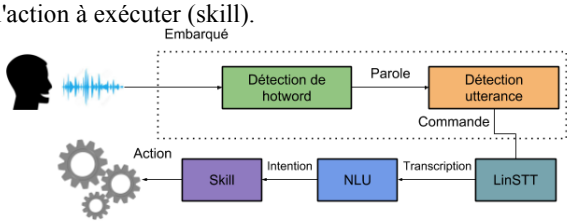


Figure 1 : Chaîne de traitement en mode commande.

4.1 Détection du « hotword »

Le mode commande est déclenché à l'aide d'un mécanisme de réveil qui consiste à reconnaître un ou plusieurs mots spécifiques prononcés par l'utilisateur afin de s'activer.

Dans cette approche la détection d'un mot-clef (*hotword*) prononcé par l'utilisateur précède une commande ou une question. Le dispositif sait que la phrase suivant le hotword lui est destinée et il ne traite que celle-ci. La reconnaissance du mot de réveil se base sur un classifieur neuronal de type réseaux récurrents. Les LSTM (Long Short Term Memory) et GRU (Gated Recurrent Units) sont alors deux solutions possibles. Le réseau GRU est une généralisation d'un réseau LSTM et offre de meilleurs résultats sur des jeux de données restreints, de plus, à performance égale il est plus léger qu'un réseau LSTM [12]. C'est donc ce type de réseau qui a été retenu.

Quand le mot-clef est détecté, le signal est analysé afin de détecter l'activité vocale pour identifier le début et la fin d'une commande. Cette commande est ensuite envoyée à LinSTT qui transcrit l'audio en texte à partir duquel est extraite une intention. Cette dernière est associée à une compétence matérialisée par un composant logiciel (un *skill*) qui effectue une action comme par exemple projeter un document.

4.2 Détection d'intention

L'analyse du texte transcrit permet d'identifier les intentions et leurs paramètres en vue d'activer les actions associées (exemple : LinTO *allume* la lumière).

Pour cette étape nous nous appuyons sur le composant open-source TOCK [15] qui permet de construire des modèles d'analyse du langage naturel. Ce dernier propose une interface graphique qui permet d'associer à chaque phrase l'intention et les entités correspondantes afin de construire des modèles. Cette étape utilise les librairies Stanford CoreNLP ou Apache OpenNLP.

Nous obtenons des performances respectables, tant dans le cas "commande" (taux de reconnaissance des intentions supérieur à 97%) que dans le cas large vocabulaire (Word Error Rate, WER inférieur à 14% évalué sur le corpus ESTER [4], modèle DNN-HMM).

5 Fonctionnalités avancées en réunion

Il s'agit ici d'analyser la conversation afin de proposer l'aide à la décision, l'extraction de thèmes en temps réel et le résumé automatique.

5.1 Interaction langagière multi-locuteurs

Les services d'aide à la gestion de réunions ont été identifiés dans [11] comme des scénarios complexes, centrés sur l'interaction conversationnelle. La multimodalité en situation multipartite a été étudiée dans et représente toujours un verrou important [2].

A partir des transcriptions fournies par les étapes précédentes, le projet prévoit une analyse des interactions, pour répondre à trois questions : (1) à qui s'adresse-t-on dans le cas d'un dialogue à plus de deux personnes (2) quels sont les buts conversationnels d'une intervention (réponse à une question, accusé de réception, élaboration d'une question passée, acceptation d'une proposition, etc.), ce que l'on désigne par le terme d'actes de dialogue et (3) quels sont les fils conversationnels qui composent l'interaction à plusieurs. Pour pallier le coût de constituer des données annotées utilisées dans la plupart des approches automatiques qui reposent sur de l'apprentissage supervisé, le projet a pour but de développer de la supervision distante, en suivant l'approche décrite dans [10]. Ces travaux sont en cours et font l'objet du focus particulier sur le deuxième année.

5.2 Synthèse de réunion

Afin de produire un résumé à la fin de la réunion, deux approches seront comparées. La première vise à produire d'un seul jet un résumé pouvant être lu et sauvegardé tel quel, en se passant de toute intervention humaine. La seconde approche, produit une proposition de résumé sous la forme d'un modèle pré-rempli, demandant ensuite à être corrigé et réorganisé par un ou plusieurs participants. Lors de la première année du projet, c'est la première approche qui a été évaluée.

Nous avons travaillé sur la compréhension par la machine du texte issu du système ASR. Cette première étape indispensable permet ensuite de grouper les utterances de telle sorte que pour chaque groupe, une phrase résumant l'ensemble du groupe puisse être produite. Le compte rendu final est alors composé de l'ensemble de ces phrases résumées.

Plus précisément, un encodeur neuronal d'utterances basé sur des réseaux récurrents et trois types de mécanismes attentionnels [1] a été développé. Cet encodeur a ensuite été incorporé dans des architectures siamoises et triplettes [8] et entraîné sur le corpus AMI [6]. Les évaluations montrent que l'encodeur proposé permet d'obtenir des groupes d'utterances proches des solutions humaines, et généralement meilleurs que ceux retournés par d'autres systèmes de l'état de l'art [14].

Par ailleurs, le corpus AMI est en cours de traduction via une approche de *crowdsourcing* afin de pouvoir développer une version française du modèle.

6 Reconnaissance visuelle

LinTO doit inclure des services de reconnaissance de participants et de gestes à l'aide d'une caméra 360° pour apporter une assistance supplémentaire. Le traitement audio est incapable seul de répondre à une partie des besoins comme le comptage lors d'un vote collectif ou la demande d'une prise de parole en réunion.

La reconnaissance visuelle vise également à inférer une localisation topologique des participants à la réunion afin de permettre à LinTO d'agir sur la matrice de microphones et renforcer la reconnaissance audio.

De manière globale, deux stratégies de fusion des percepts audio et visuel sont à considérer pour notre assistant: d'un coté, l'information provenant du module vision est récupérée puis synthétisée vocalement par LinTO comme par exemple lors d'une demande de parole à distance. De l'autre côté, c'est un participant qui demande à LinTO d'exécuter une tâche nécessitant la ressource vision par exemple le comptage des personnes « pour » dans le cas d'un vote.

Cependant, l'intégration de la reconnaissance visuelle dans LinTO pose plusieurs défis :

- traitement du flux vidéo d'images 360° ;
- construction d'un corpus d'images pour des personnes en réunion, annotation et entraînement de modèles d'apprentissage profond ;
- contrainte temps réels.

Notons que nous nous focalisons sur la localisation des participants à la réunion et pas à l'identité des individus.

6.1 Dispositif dédié à l'analyse visuelle

Notre dispositif privilégie une caméra RICOH THETA pour son faible coût, sa popularité auprès des développeurs et la possibilité de streaming vidéo. Les images sont acquises au format «dual-ficheyes» puis transformées en images panoramiques prêtes pour le traitement. Il est judicieux d'étudier les modèles d'apprentissage profond existants afin d'évaluer leurs performances dans notre contexte applicatif. De plus, ces modèles sont déjà entraînés et feront gagner un temps considérable quand à la construction de corpus, l'annotation et l'entraînement.

6.2 Reconnaissance de participants

L'objectif est de détecter les personnes en réunions et les repérer spatialement en temps réel à partir de notre caméra. Comme évoqué, la localisation des personnes en environnement encombré donc en présence d'occultations éventuelles (écran, autres personnes, etc.), est un verrou important.

Notre détecteur de personnes est basé sur le modèle SSD Mobilenet (mieux adapté à des cartes matérielles de faible ressources) et un filtrage spatio-temporel pour le suivi (*tracking*). Le principe est le suivant :

- le filtrage spatial repose sur le recouvrement entre les régions images associées aux détections de personnes entre deux instants d'image ;
- le filtrage temporel repose sur l'hypothèse de faible déplacement des personnes entre images successives. On estime cette valeur à 6 secondes après des expérimentations pour une personne qui quitte sa place.

Ce filtrage est possible grâce à l'historique des détections mémorisé durant le déroulement de la réunion. L'ajout de l'algorithme DELF (*DEep Local Feature*) de *matchings* d'images permet de lever les doutes quand il y a confusion. Les problèmes qui persistent sont les non-détections sporadiques et les ambiguïtés lors d'un chevauchement de zones de détection durant le déplacement de personnes.

6.3 Reconnaissance de gestes

Le modèle présenté dans [7], entraîné sur un million d'images de gestes, a été expérimenté avec succès sur

nos images et semble pertinent pour la reconnaissance de gestes. Cependant nos investigations actuelles ne permettent pas de cibler à ce jour le pipeline adapté à l'intégration de nos modalités compatibles avec les ressources matérielles limitées.

6.4 Discussion sur l'analyse visuelles

Ces évaluations qualitatives préliminaires ont permis d'exhiber quelques verrous listés ci-après :

- mouvements conjoints : expressions faciales, rotations extrêmes de la tête, etc. induisant des erreurs d'interprétation, par exemple, une personne qui se tourne pour parler à la personne d'à côté ;
- variation d'éclairage de la salle de réunion induisant des problèmes de robustesse, par exemple, des salles avec des fenêtres donnant sur l'extérieur ou dans des situations où la lumière s'éteint pendant une présentation et s'illumine pendant la discussion ;
- distance importante caméra / scène induisant des résolutions images trop faibles pour inférer certains percepts comme des gestes faciaux, cela se produit généralement lorsque on place la caméra sur un mur ou un coin de la salle de réunion.
- variabilité du point de vue caméra / scène induisant des problèmes de robustesse, les participants sont occultés par la personne la plus proche de la caméra.

La question de la performance demeure centrale, notamment pour un couplage de traitement audio/vidéo. Plusieurs solutions sont en cours d'expérimentation comme par exemple la délégation de certains traitements à distance.

7 Respect des données personnelles

Un facteur important du projet concerne la sécurité et la confidentialité des données personnelles. En effet, l'objectif de LinTO est d'aider les participants présents en salle de réunion. Afin de pouvoir les suivre efficacement, LinTO est susceptible d'écouter leurs conversations et d'analyser diverses données les concernant (emails, calendriers, etc.). Afin de minimiser les risques d'atteinte à la vie privée, le traitement doit être effectué en local (dispositif LinTO dédié).

Pour cela LinTO traite toutes les données à l'intérieur du système d'information de l'entreprise, éliminant ainsi le besoin d'envoyer les données vers un Cloud externe et ne stocke pas les données vocales. Par conséquent, personne ne dispose d'informations sur les paroles échangées, ce qui protège contre le piratage et la surveillance de masse.

8 Prochaines étapes

Le projet a commencé en avril 2018 et doit se terminer en mars 2021. La première année a permis de mettre en place le projet et de valider le mode assistant personnel qui s'appuie sur le modèle commande. Des prototypes sont disponibles pour l'aide à la rédaction de compte-rendu de réunions [13] ainsi que pour la localisation spatiale des participants. Les prochaines étapes concernent l'étude des stratégies de fusion des percepts audio et visuel, l'intégration des approche d'analyse des interactions langagière avec la démarche de génération de résumé et enfin la qualification définitive de l'architecture du dispositif LinTO afin de prendre en compte les contraintes aussi bien d'un point de vue scientifique que du traitement des données personnelles.

Bibliographie

[1] Bahdanau et al, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473 (2014).

[2] Chen L. et al. (2006) VACE Multimodal Meeting Corpus. In: Renals S., Bengio S. (eds) Machine Learning for Multimodal Interaction. MLMI 2005. Lecture Notes in Computer Science, vol 3869. Springer, Berlin, Heidelberg

[3] J. Bengio, *Reaching new records in speech recognition*, Mars 2017, https://www.ibm.com/blogs/watson/2017/03/reaching-new-records-in-speech-recognition/

[4] Galliano et al, *The ester 2 evaluation campaign for the rich transcription of French radio broadcasts*, Interspeech-2009, 2583-2586

[5] *Global Intelligent Virtual Assistant Market*, juin 2018 https://www.transparencymarketresearch.com/pressrelease/intelligent-virtual-assistant-industry.htm

[6] Hoffer, Elad, and N. Ailon. *Deep metric learning using triplet network*, International Workshop on Similarity-Based Pattern Recognition. Springer, Cham, 2015.

[7] O. Koller et al. *Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3793-3802, Las Vegas, NV, USA, June 2016.

[8] McCowan, Iain, et al. *The AMI meeting corpus*. Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research. Vol. 88. 2005.

[9] D. Povey et al, *The Kaldi Speech Recognition Toolkit*, Workshop on Automatic Speech Recognition and Understanding, 2011

[10] A. Ratner et al, *Rapid Training Data Creation with Weak Supervision*, Proceedings of the VLDB Endowment, 11(3), 269-282, 2017

[11] S. Renals et al, *ROCKIT: roadmap for conversational interaction technologies*. Proceedings of the 2014 Workshop on Roadmapping the Future of Multimodal Interaction Research including Business Opportunities and Challenges. ACM, 2014. p. 39-42.

[12] A. Sercan O et al. *Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting*, Interspeech 2017

[13] G. Shang, W. Ding, Z. Zhang, A. J.-P. Tixier, P. Meladianos, M. Vazirgiannis, J.P. Lorré, *Unsupervised Abstractive Meeting Summarization with Multi-Sentence Compression and Budgeted Submodular Maximization*, ACL 2018.

[14] G. Shang et al, *Energy-based Self-attentive Learning of Abstractive Communities for Spoken Language Understanding, arXiv:1904.09491 [cs.CL]*

[15] *Tock (The Open Conversation Kit)*: https://voyages-sncf-technologies.github.io/tock

La gestion d'actifs augmentée par l'intelligence artificielle

Juliette MATTIOLI[1] Sarah LAMOUDI[2] Pierre-Olivier ROBIC [3]

[1] Thales,
[2] Consultant Expert pour le Pôle Finance-Innovation,
[3] Thales Global Services

juliette.mattioli@thalesgroup.com
sarah.lamoudi@gmail.com
pierre-olivier.robic@thalesgroup.com

Résumé

L'intelligence artificielle (IA) bouleverse tous les secteurs mais surtout le secteur secondaire. En effet, les moyens de productions industriels sont déjà équipées de capteurs qui servent à collecter des informations capitales. Les données vont alors aider à optimiser la production. Les banques, elles-aussi collectent de nombreuses données, comme celles relatives aux cours de bourse, à la santé des entreprises... L'analyse prédictive et prescriptive va permettre de prévoir et de quantifier la valeur des actifs et cela afin de prendre les meilleures décisions possibles. Quelle soit statistique, connexionniste ou symbolique, cet article présente comment l'IA contribue à la gestion d'actifs (GDA) tant dans le monde de la finance que celui de l'industrie.

Mots Clef

Gestion d'actifs, apprentissage automatique, décision dans l'incertain, optimisation multicritère, IA connexionniste, IA symbolique.

Abstract

Today, Artificial Intelligence (AI) disrupts all sectors and especially industry. Indeed, industrial assets are already equipped with sensors that are used to collect relevant data. Such data will help to optimize production. Finance collected such data for a long time, such as stock market prices, corporate data for enterprise quotation, etc. Predictive and prescriptive analysis will predict and quantify the value of assets in order to take the best possible decision. Whether AI is statistic, connectionist or symbolic, this article underlines how AI contributes to the management of assets both in the domains : finance and industry.

Keywords

Asset management, machine learning, decision under uncertainty, multicriteria optimisation, connexionnist AI, symbolic AI.

1 La gestion d'actifs financiers et industriels

1.1 Contexte applicatif

Les avancées en intelligence artificielle (IA) de ces dernières années ont permis la mise en oeuvre de résultats prometteurs dans de nombreux domaines. Il devient possible de prédire, d'anticiper ou d'optimiser des comportements ou des processus métiers, dotant un système de capacité d'apprentissage, de raisonnement et de décision. Ainsi, l'IA peut contribuer à la gestion d'actifs allant de la gestion de portefeuille de biens aux décisions d'investissement tout en prenant en compte la conformité et les risques. Cependant, le terme **gestion d'actifs** (GDA - *Asset management* en anglais) est utilisé de manière très différente suivant le domaine d'usage concerné.

En effet, *en finance*, la GDA consiste à faire prospérer le patrimoine de ses clients via l'investissement, en respectant les obligations réglementaires et contractuelles et en appliquant des stratégies d'investissements, afin de dégager le meilleur rendement possible en fonction du risque choisi [1].

L'*industrie manufacturière* a repris ce terme pour décrire la maîtrise du cycle de vie de ses actifs. Une installation industrielle produit de la valeur grâce à ses performances techniques de disponibilité, de rendement et la qualité de sa production. Il est donc important de les préserver. Les actifs considérés sont alors les biens physiques. Cette approche est connexe aux approches de type *Product Life Cycle Management* (PLM littéralement gestion du cycle de vie des produits) ou *Service Life Cycle Management* (SLM), sur la composante services [2]. La GDA couvre ici les activités réalisées directement sur les actifs au cours de leur cycle de vie, de la conception ou acquisition à leur déclassement ou destruction en passant par leur exploitation, maintenance et renouvellement. Les activités de gestion nécessaires pour diriger, planifier, coordonner, favoriser, faciliter, soutenir et améliorer en continu l'efficacité et l'efficience des activités effectuées directement sur ou par les actifs sont aussi

à prendre en compte. De plus, un actif physique industriel possède une valeur d'usage (valeur à neuf du bien, corrigée de la vétusté et/ou des maintenances ou réparation réalisés sur ce bien), et une valeur vénale, constituant ainsi un patrimoine de l'entreprise. Enfin, avec le mouvement "usine 4.0", l'usine intelligente est une usine productrice de données, ces dernières étant majoritairement délivrées par les technologies de l'Internet of Things (IoT). Il devient alors possible d'avoir des informations sur l'utilisation des actifs, sur leur degré d'obsolescence, de dégradation, ou sur leurs éventuelles défaillances pour optimiser la gestion des actifs de production en minimisant le coût du cycle de vie constitué des coûts directs et indirects (manque à gagner...) et obtenant le meilleur compromis entre les besoins à court et à long terme. L'IFRAMI (Institut Français d'Asset Management Industriel et Infrastructures) définit la GDA comme un processus impliquant l'équilibre des coûts, la prise en compte des risques et des opportunités ainsi que les avantages liés à une meilleure performance des actifs.

Un système de GDA est donc un système d'aide à la décision permettant d'aligner toutes les activités induites pour générer de la valeur, garantir une meilleure qualité et cohérence dans la prise de décision, définir des priorités dans les mesures à prendre, et permettre une meilleure gestion des risques, des incertitudes et des changements.

Le champ des possibles de l'IA est alors immense et ne cesse de s'étendre. Qu'elle soit symbolique, connexionniste ou statistique, et/ou combinée à la science des données (voir fig. 1), l'IA apporte des solutions à toutes les étapes de la GDA, de la gestion de portefeuille, aux décisions d'investissement en passant par la conformité réglementaire et la gestion des risques.

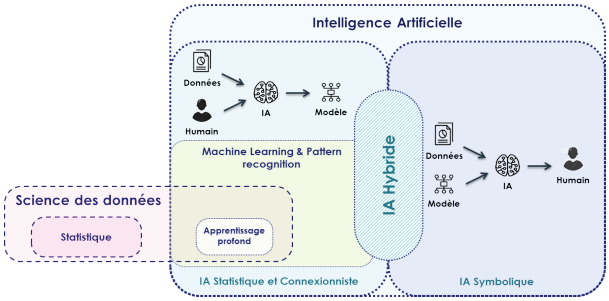


FIGURE 1 – Intelligence artificielle et science des données

1.2 Structure de l'article

L'objectif de ce papier est de montrer comment les différentes techniques de l'intelligence artificielle (IA) sont déjà opérationnelle pour la GDA tant dans le domaine de la finance que dans l'industrie.

Notons que ce panorama pour le domaine de la finance est une synthèse des travaux menés dans le cadre du livre blanc "*Intelligence Artificielle, Blockchain et Technologies Quantiques au service de la finance de demain*" [3]. Pour le domaine de l'industrie, cet article reflète les actions autour de la transformation numérique de Thales.

Face à la complexité croissante des systèmes (financiers ou industriels) et aux évolutions souvent imprévisibles des environnements dans lesquels ils sont plongés, le décideur a de plus en plus besoin d'outils pour une GDA efficiente et efficace. Mais pour prendre de bonnes décisions, il est nécessaire de bien comprendre la situation courante, puis de faire de bonnes prévisions [1]. Dans ce contexte, les algorithmes d'apprentissage apportent d'excellents résultats, comme le souligne la section 2.

Le caractère optimal de la GDA a amené les chercheurs à proposer une modélisation classique (ou monocritère) des problèmes de décision sous contraintes dont la solution représente le meilleur choix. Mais il parait illusoire de parler d'optimalité car les critères de décisions devant être pris en compte sont multiples et parfois conflictuels. Ainsi, la GDA étant par essence de nature multicritère [4], l'aide à la décision multicritère fournit le cadre naturel à la résolution de tels problèmes comme le souligne la section 3.

Enfin, compte tenu de la complexité, les approches hybrides (mixant IA des données avec l'IA symbolique) prennent tout leur sens (voir section 4). Par exemple, pour obtenir une estimation de la valeur du portefeuille aussi réaliste que possible, le couplage de la prédiction estimé par apprentissage avec de la connaissance métier est pertinent. D'autres techniques d'IA peuvent apporter une vraie valeur ajoutée. En effet, la lecture et l'interprétation de toutes les règlementations requièrent beaucoup de temps de personnels qualifiés. Il est donc envisageable d'effectuer une lecture automatique par le biais de techniques du traitement automatique de la langue (TAL) autorisant une sélection ciblée des articles pertinents. De plus, couplé aux arbres de décision, le TAL est de plus en plus utilisés en GDA en particulier au travers des chatbots avec l'exemple des robots advisors (voir paragraphe 4.2) très répandus en finance.

2 Apprentissage automatique au service de la GDA

En finance, un nom incontestablement associé à la gestion d'actifs, est celui de l'économiste Harry Markowitz qui reçut le prix Nobel d'économie pour ses travaux en 1990. Son idée repose sur la construction d'un portefeuille d'actions qui permet d'obtenir un rendement donné avec un risque moindre comparé à celui de chaque action prise séparément, et cela en répartissant de manière optimisée son argent entre les différents actifs financiers. Ce problème d'ingénierie financière nécessite de déterminer une stratégie d'investissement parmi un ensemble d'actifs afin d'atteindre certains objectifs, tels que la maximisation de la richesse cumulée ou le rendement ajusté du risque à

1. Par exemple, dans le contexte de la finance, une prévision est un pronostic effectué sur le comportement futur de variables d'intérêt. Par exemple, dans le cas de la finance, ces variables modélisent les rendements du marché boursier et la décision est l'action prise sur les marchés, comme l'action d'investir dans certains titres plutôt que d'autres.

long terme. H. Markowitz propose [5] en 1952 son modèle Moyenne-Variance reposant sur l'assertion que *tout investisseur poursuit deux objectifs contradictoires qui sont la maximisation du rendement espéré et la minimisation du risque mesuré par la variance du rendement*. Sur cette base, de nombreuses variantes ont été développées [6] parmi lesquelles on trouve la sélection de portefeuille, la théorie de l'utilité espérée, les modèles d'équilibre d'actifs financiers ou la théorie du prix d'arbitrage. Chaque action est décrite par son rendement moyen et sa volatilité. Les estimations du rendement moyen et de la volatilité s'appuient en général, sur des approches d'IA statistique mais les arbres de décisions et les réseaux de neurones donnent aujourd'hui des résultats plus performants [7].

Dès 1997, l'utilisation de réseaux de neurones (RN) à base de perceptron multi-couches est proposée dans [8] pour la détection automatique d'entreprises en difficulté, et Y. Bengio [9] applique des techniques d'apprentissage par renforcement sur un problème réaliste d'allocation de 35 actifs. Il obtient d'excellente performance par rapport à un modèle entraîné à minimiser l'erreur de prévision (erreur quadratique). Dans les années 2000, des approches à base de *Pattern-Matching* ont été utilisés pour indentifier dans l'historique du comportement du marché, une partie similaire à la situation actuelle pour optimiser le portefeuille [10]. En 2017, dans [11], les auteurs reprennent l'idée de Y. Bengio sur le *deep learning*, et appliquent avec succès des techniques d'apprentissage profond par renforcement (*reinforcement deep learning*) à la gestion de portefeuille, et cela sans aucune connaissance des marchés financiers.

Dans le cadre de la GDA industrielle, comme celle déployée dans Thales, le problème d'évaluation des risques est lié à la défaillance des actifs physiques. Cela consiste alors à détecter et à diagnostiquer des dégradations ou des baisses de performance des moyens de production. L'objectif est de pouvoir prédire (pronostiquer) des défaillances dans le futur, en inférant l'état de santé actuel des actifs dans le temps. Cette fonction *pronostic* se concentre sur l'évolution des défaillances progressives et exploite généralement l'état de santé actuel de l'actif (résultats du diagnostic prédictif) pour évaluer son état de santé futur, au regard du profil d'emploi estimé. Bien qu'il n'existe pas de définition unique de la notion de pronostic dans le domaine de la GDA industrielle, les différentes définitions existantes convergent vers une fonction capable de déterminer la durée de vie restante avant la défaillance, nommée RUL (*Remaining Useful Life*), ou la probabilité d'un système à fonctionner pendant un certain temps. Ce concept est spécifique à la maintenance prédictive [12]. Par nature, le pronostic vise à anticiper l'état d'un système dans le futur et amène des incertitudes sur la valeur du RUL. En effet, le pronostic de l'état futur de l'actif considéré doit non seulement prendre en compte son état actuel, mais également des données relatives à ce système dans le futur (prévisions d'utilisation, informations sur l'environnement, futures opérations de maintenance exécutées...), qui influent sur son état de dégradation [13]. De nombreux travaux ont été réalisés ces dernières années afin d'améliorer la performance de l'estimation du RUL. [14] présente un état de l'art assez complet des méthodes de calcul du RUL parmi lesquelles figurent les techniques de régression, les Hidden Markov Models (HMM) et Hidden Semi-Markov Models (HSMM). [15] utilise une approche de *Multiobjective Deep Belief Network Ensemble* basée sur de l'apprentissage automatique non supervisé. Toutefois, les méthodes les plus prometteuses sont celles qui combinent plusieurs approches, comme le souligne [16]. En effet, le pronostic doit prévoir l'état futur et fournir une durée de vie résiduelle à partir d'une connaissance à priori sur ce système (lois de dégradations, par exemple), d'une connaissance d'informations passées (historique des modes de fonctionnement passés), présentes (état courant) et futures, ce qui nécessite une hybridation de techniques d'IA symboliques avec des méthodes d'apprentissages (voir section 4).

3 GDA : problème de décision multicritère sous contraintes en avenir incertain

3.1 Principes généraux de la décision multicritère en avenir incertain

Aide à la décision. L'aide à la décision est sans doute l'un des domaines où l'émergence de l'intelligence artificielle (IA) apporte un bouleversement majeur, tant sur la nature de l'aide fournie que sur l'échelle temporelle où cette information est susceptible d'être disponible. Elle peut même conduire à un changement de paradigme. Cela concerne tous les métiers de la finance mais aussi ceux de l'industrie. Mais qu'est-ce qu'une décision ? C'est le fait qu'un acteur (ou un ensemble d'acteurs) effectue un choix, si possible après réflexion, entre plusieurs solutions (*alternatives*) pour affronter une situation difficile, résoudre un problème délicat ou répondre à une question complexe.

Décision en avenir incertain. La décision en avenir incertain prend tout son sens lors d'opérations de gestion d'événements rares ou exceptionnels, et cela en raison des bouleversements de l'environnement dans lequel les organisations évoluent. En effet, on ne peut prévoir ni leur forme (ex. krach boursier, panne subite d'un équipement), ni le moment précis de leur déclenchement. De plus, la capacité de l'organisation à prendre des décisions justes et rapides est un facteur clé de succès vers l'issue la plus favorable. L'incertitude est alors le cadre de cette prise de décision. Il est impossible d'y échapper. Elle renvoie au fait que les conséquences d'une décision ne sont pas connues à l'avance, ce qui expose le décideur à un risque. Cependant, le risque encouru peut être quantifiable quand la probabilité d'occurrence des différentes conséquences possibles est objectivement connue à l'avance. On estime alors la perte encourue en multipliant le montant de la perte éven-

tuelle par la probabilité que se réalise l'éventualité défavorable, en utilisant par exemple le RUL dans le cadre industriel ou la Value at Risk (VaR) mesure probabiliste de la perte financière possible sur un horizon donné. Toutefois, selon Knight [17], l'incertitude devient intrinsèque si les risques ne sont pas objectivement mesurables à l'avance. Ainsi, en finance, la rentabilité des investissements par le crédit bancaire et le rendement futur des titres obligataires ou des actions sont affectés par cette incertitude. Les méthodes sous-jacentes à cette estimation ont été présentées dans la section 2 et sont en général soit basées sur de l'IA statistique ou connexioniste soit sur de l'IA hybride.

Décision multicritère. De plus, un problème de décision est souvent multicritère. Il se caractérise par la prise en compte explicite de plusieurs objectifs à optimiser simultanément dans l'analyse des préférences, la comparaison des solutions et la détermination d'une ou des solution(s) optimale(s). Les problèmes induits varient selon la question posée. On peut distinguer les problèmes de choix où l'on cherche à déterminer les meilleures solutions, les problèmes de classement où l'on veut ordonner, au moins partiellement, les solutions selon un ou plusieurs critères et les problèmes de classification où l'on cherche à affecter les solutions à des catégories prédéfinies selon leur valeur intrinsèque. La problématique du choix est celle que l'on rencontre le plus fréquemment. Elle vise à trouver une solution qui optimise au mieux les différents critères, ou un sous-ensemble de solutions, aussi réduit que possible, contenant les meilleures solutions. Formulée ainsi, l'optimisation multicritère est alors un problème mal posé. En effet, de part la nature potentiellement conflictuelle des critères, comme c'est le cas pour la GDA, il n'existe généralement pas de solution optimisant tous les critères simultanément. La plupart des méthodes existantes en décision multicritère se basent sur la somme pondérée, et ce pour des raisons évidentes de simplicité, mais cet opérateur présente des biais qu'il n'est pas possible d'éliminer. Par exemple, comme le montre [18], la somme pondérée ne permet pas de modéliser des phénomènes de véto ou des phénomènes de compensation entre critères. Le problème de l'agrégation des préférences consiste alors à synthétiser des informations traduisant des aspects ou des points de vues différents, parfois conflictuels, au sujet d'un même ensemble d'objets ou d'actions (performances, utilités, préférences). Il se pose de manière cruciale dans nombre de procédures d'évaluation, de comparaison ou de classification utilisées en aide à la décision multicritère [19]. Que ce soit un problème de choix ou de rangement, la question centrale est toujours un problème de comparaison. Ainsi, dans un problème de choix, l'identification du meilleur candidat nécessite d'être capable de le comparer à tous les autres ; dans un problème de rangement, on espère pouvoir comparer toute paire d'actions de manière à obtenir un classement complet. Un problème d'agrégation multicritère consiste alors à exploiter l'information préférentielle (performances, indices d'importance), pour construire un modèle de comparaison globale des alternatives sous la forme d'une relation de préférence $\succeq$ sur l'ensemble des dites alternatives. Pour cela, on utilise nécessairement une règle d'agrégation permettant de construire la relation $\succeq$ à partir des vecteurs de performances.

3.2 Décision multicritère appliquée à la GDA

En matière de gestion financière de portefeuille, l'ensemble des actions présentes sur un maché financier est discret, fini et défini en extention. Chaque action est alors représentée dans l'espace des critères. En général, six critères sont utilisés :

- Le *Return on Equity* (ROE), critère de rentabilité des capitaux propres à maximiser ;
- Le *current ratio,* critère de liquidité au sens strict qui doit être maximisé ;
- Le ratio cash flow/dette, critère de solvabilité à maximiser ;
- Le rendement mensuel moyen ;
- Le *Price Earning Ratio* (PER) mensuel qui doit être minimisé ;
- Et le bénéfice par action (EPS, *Earnings Per Share*) annuel, qui doit être maximisé.

Dans le monde de l'industrie, lors de l'élaboration d'un plan de renouvellement des actifs de production, les décideurs sont confrontés à plusieurs questions, parmi lesquelles :

- Quand est-il approprié de remplacer les actifs existants au lieu de continuer à les entretenir ?
- Que faire si le potentiel de production de l'actif est insuffisant ?
- Quels critères devraient être considérés lors du remplacement de l'actif ?

Ces problèmes font partie de la GDA et devraient être traités de façon appropriée lorsque l'on considère la question de renouvellement. Le taux de remplacement d'un actif peut en effet, varier en raison de plusieurs facteurs dont la maintenance excessive, l'obsolescence, la détérioration physique, la vie de l'actif soumis à l'usure et les défaillances subites.

Dans ces deux cadres (finance et industrie), l'application de techniques d'aide à la décision multicritères permet de définir une fonction d'aggrégation des diffrérents KPI (*Key Performance Indicator*) intègrant les différentes dimensions de la GDA pour une optimisation multicritère. Les techniques de planification et de résolution de problèmes combinatoires comme la programmation par contraintes permettent alors de proposer au décideur, une ou des solutions valides.

4 IA hybride en GDA

4.1 Estimation de la valeur d'un portefeuille d'actifs

Si les modèles analytiques traditionnels basés sur les processus de fabrication du produit ou du service, restent encore largement utilisés en industrie, les modèles statistiques s'imposent progressivement. Ainsi, pour estimer la valeur d'un portefeuille d'actifs (industriels ou financiers) au cours du temps, le recours de techniques faisant appel aux approches stochastiques est devenue une pratique courante [20]. Ces modèles stochastiques doivent cependant être calibrés (par apprentissage par exemple) à partir des données ou mesures observables les plus récentes possibles [21]. Cependant, la complexité des approches sous-jacentes, surtout si elles sont dynamiques et stochastiques, rend difficile une implantation numérique, à moins de simplifier les modèles. Ainsi, en renonçant à un modèle très précis, [22] propose une approche reposant sur l'existence de moyennes ou de tendances et sur des séries temporelles pour l'estimation de la valeur d'un portefeuille d'actifs financiers. Signalons que [23] utilise dès 2002 des approches neuronales pour aborder ce problème, ou plus récemment avec l'apprentissage profond [11]. En 2004, CMU (*Carnegie Mellon University*) propose même *Warren*, un système de gestion intelligente de portefeuille reposant sur des approches multi-agents combinées à de l'analyse textuelle de rapports financiers [24].

Mais avec l'IA hybride couplant apprentissage et connaissances, de nouvelles méthodologies apparaissent, notamment en matière de pricing [20]. On considère alors que le prix d'un actif est corrigé par l'ajout de données complémentaires, comme les agrégats des vues, avis et réflexions délivrées par de nombreux individus (clients, prospects, partenaires, etc.) permettant de comprendre sa dynamique et d'anticiper son évolution future.

4.2 Robo-advisors support à la GDA

Depuis plusieurs années, une nouvelle forme de conseil financier a émergé bousculant le monde traditionnel : les *robo-advisors* [25]. Littéralement "robot-conseiller", le robo-advisor est un conseiller en gestion de patrimoine, impliquant un strict minimum d'intervention humaine, les arbitrages étant effectués sur la base d'algorithmes croisant de grandes quantités de données. Plus précisément, il s'agit d'examiner comment segmenter les clients en matière de gestion de patrimoine, en fonction de leurs besoins et de leurs préférences, et développer une offre de conseil partiellement automatisée et personnalisée [26]. Pour ce faire, l'IA peut être utilisés dans le fonctionnement de ses services pour contrecarrer les deux points de blocages majeurs :

- Le coût : dans la finance traditionnelle, moins le client a de ressources, plus le conseil lui revient cher. Grâce à l'apprentissage automatique, les temps de calcul sont réduits et la recommandation du CGP (Conseil en Gestion de Patrimoine) est automatisée, baissant ainsi les coûts ;
- Le choix de produits et la personnalisation : la plupart des acteurs de l'épargne cherchent aujourd'hui à faire rentrer leurs clients dans des cases prédéfinies et n'offrent pas de personnalisation du service : c'est en effet la solution la plus couramment utilisée pour industrialiser la distribution. Grâce à l'aide à la décision multicritère, des systèmes de recommandations sur-mesure adaptées à leurs objectifs, à leurs ressources financières, à leur situation patrimoniale et à leur volonté de prise de risques seront possibles.

De nombreuses banques ont été séduites par les assistants virtuels, comme la banque singapourienne DBS qui a décidé de mettre Watson au service de ses conseillers en gestion du patrimoine. Dans ce cas, une combinaison de techniques d'apprentissage, de raisonnement à base de connaissances (modélisées par le biais d'ontologie) a permis de concevoir des robot-advisors ou des chatbots pertinents. Ainsi un système, reposant sur une hybridation d'algorithmes d'apprentissage comme le deep learning et IA symbolique, permet ainsi de bâtir une recommandation personnalisée. Par ailleurs, la collecte ciblée de données améliorera la pertinence de la recommandation.

5 Conclusions

Le secteur de la GDA aussi bien financière qu'industrielle est actuellement en pleine transformation, touché par l'augmentation de l'incertitude, de nouvelles contraintes réglementaires et l'objectif d'augmenter l'efficacité du portefeuille. Ainsi dans un environnement de plus en plus concurrentiel, l'IA devient alors incontournable. La mise en place de méthodes d'analyse de données historiques à base de machine learning permet ainsi d'évaluer les performances des actifs, l'estimation de leur valeur et de détecter des schémas qui se répètent et réallouer le portefeuille en conséquence. Dans ce processus de GDA, la dimension de l'affect humain est aussi extrêmement importante. L'apport de la décision multicritère (voir section 3) couplé à des techniques d'apprentissage est très attendu pour permettre de s'adapter à un environnement changeant où il n'existe pas de règles précises. Deux approches se profilent :

- L'approche *exo-squelette* consiste à utiliser des outils d'aide à la décision, permet d'augmenter la productivité du gérant d'actif par le recours à des outils techniques utilisant l'IA.
- La seconde approche, à plus long terme, inclura l'industrialisation du processus d'investissement et donc de la prise de décision. C'est un changement de paradigme significatif, analogue à la révolution industrielle qui a vu la disparition de la fabrication artisanale au profit d'unités de production où les tâches rationalisées et optimisées sont effectuées par des machines contrôlées par l'homme.

Pour y parvenir, il faut

— Travailler sur la qualité des données : les données considérées comme utiles à la prise de décision sont celles sur les actifs, son contexte d'emploi, et le sentiment des décideurs à l'égard de la valeur de l'actif. L'utilisation de données externes factuelles alternatives est intéressante et peut donner un avantage concurrentiel, mais elles sont sujettes à interprétation positive ou négative selon le contexte (par exemple : des stocks importants peuvent vouloir dire qu'il y a une surproduction et donc des invendus - signal négatif ou que l'entreprise a engrangé beaucoup de commandes - signal positif). Le lien entre l'information et la décision à prendre n'est pas linéaire et l'enjeu est donc de sélectionner attentivement l'information pertinente. Avant toute mesure de mise en oeuvre, il est important de définir les objectifs et usages des données à traiter en fonction des enjeux selon le principe du "fitness for use" : dans certains cas, une tolérance à l'erreur est acceptable (comme des systèmes de recommandation d'achat en B2C) alors que dans d'autres (comme les systèmes critiques ayant un impact juridique, médical, financierâĂę), l'ensemble du système d'information devra ÃŁtre traité avec la plus grande rigueur. Il s'agit donc d'une problématique liée au domaine métier et à l'usage :
  — en amont, il faut que les producteurs de données deviennent des sources qualifiées, voire certifiées. On parle alors de cotation des sources ;
  — en aval, il s'agit de fournir aux utilisateurs tous les éléments utiles afin qu'ils puissent en faire un usage pertinent et durable.
— Etre sélectif sur les techniques d'IA : comme nous l'avons vu, il n'existe pas de méthode unique pour adresser le problème de la GDA et il faut donc trouver la meilleure hybridation. Cela implique de décomposer le processus en tâches simples et d'appliquer la technique la plus efficace. Pour cela, une logique holistique et systémique est extrêmement importante ;
— Et incorporer l'humain : l'estimation de la valeur des actifs résulte de l'agrégation de plusieurs points de vue et donc de l'affect des diffèrentes parties prenantes intègrant parfois des facteurs à la fois rationnels et émotionnels. L'ambition est de comprendre et d'anticiper les motivations de décisions basés sur des perceptions ou des sentiments.

Remerciements

Ces réflexions ont été nourries par de nombreuses discussions avec Bertrand Braunschweig (directeur de la mission Inria de coordination du programme national de recherche en intelligence artificielle) que les auteurs remercient chaleureusement. En effet, pendant les réunions de travail du Livre Blanc sur *Intelligence Artificielle, Blockchain et Technologies Quantiques au service de la finance de demain* [3], l'analyse de retours d'expériences issus d'autres industries que la finance a permis d'identifier de nombreuses synergies, en particulier celui de la gestion d'actifs.

Références

[1] Robert A Haugen and Robert A Haugen. *Modern investment theory*, volume 5. Prentice Hall Upper Saddle River, NJ, 2001.

[2] Hasan Burak Cavka, Sheryl Staub-French, and Erik A Poirier. Developing owner information requirements for bim-enabled project delivery and asset management. *Automation in construction*, 83 :169–183, 2017.

[3] Pôle Finance Innovation, editor. *Intelligence Artificielle, Blockchain et Technologies Quantiques au service de la finance de demain,*, Livre blanc, 2019.

[4] Christian Hurson and Constantin Zopounidis. *Gestion de portefeuille et analyse multicritere*. FeniXX, 1996.

[5] Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1) :77–91, 1952.

[6] Mark Rubinstein. Markowitz's portfolio selection : A fifty-year retrospective. *The Journal of finance*, 57(3) :1041–1045, 2002.

[7] Bo Wahlberg, Stephen Boyd, Mariette Annergren, and Yang Wang. An admm algorithm for a class of total variation regularized estimation problems. *IFAC Proceedings Volumes*, 45(16) :83–88, 2012.

[8] Philippe Paquet et al. L'utilisation des réseaux de neurones artificiels en finance. *Document de recherche*, 1, 1997.

[9] Yoshua Bengio. Training a neural network with a financial criterion rather than a prediction criterion. In *Decision Technologies for Financial Engineering : Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets (NNCM'96), World Scientific Publishing*, pages 36–48, 1997.

[10] László Györfi, Gábor Lugosi, and Frederic Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance : An International Journal of Mathematics, Statistics and Financial Economics*, 16(2) :337–357, 2006.

[11] Zhengyao Jiang and Jinjun Liang. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 905–913. IEEE, 2017.

[12] Simon Fossier and Pierre-Olivier Robic. Maintenance of complex systems : From preventive to predictive. In *Live Maintenance (ICOLIM), 2017 12th International Conference on*, pages 1–6. IEEE, 2017.

[13] Camille Baysse, Didier Bihannic, Anne Gégout-Petit, Michel Prenat, and Jérôme Saracco. Hidden markov model for the detection of a degraded operating mode of optronic equipment. *arXiv preprint arXiv :1212.2358*, 2012.

[14] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1) :1–14, 2011.

[15] Chong Zhang, Pin Lim, AK Qin, and Kay Chen Tan. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE transactions on neural networks and learning systems*, 28(10) :2306–2318, 2017.

[16] Juliette Mattioli, Pierre-Olivier Robic, and Thomas Reydellet. L'intelligence artificielle au service de la maintenance prévisionnelle. In *4ème conférence sur les Applications Pratiques de l'Intelligence Artificielle APIA2018*, 2018.

[17] Frank H. Knight. Risk, uncertainty and profit, 1921. *Library of Economics and Liberty*, 1971.

[18] Michel Grabisch and Christophe Labreuche. A decade of application of the choquet and sugeno integrals in multi-criteria decision aid. *4OR*, 6(1) :1–44, 2008.

[19] Jacky Montmain and Christophe Labreuche. Amélioration multicritère d'options dans les systèmes complexes. *LFA'2009, Rencontres Francophones sur la Logique Floue et ses Applications*, 2009.

[20] Arash Bahrammirzaee. A comparative survey of artificial intelligence applications in finance : artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, 19(8) :1165–1195, 2010.

[21] Michel Fliess, Cedric Join, and Frédéric Hatt. A-t-on vraiment besoin d'un modèle probabiliste en ingénierie financière ? In *Conférence Méditerranéenne sur l'Ingénierie Sûre des Systèmes Complexes, MISC 2011*, 2011.

[22] Michel Fliess. A mathematical proof of the existence of trends in financial time series. *Systems Theory : Modelling, Analysis and Control*, pages 43–62, 2009.

[23] Yang Liu, Xiaohui Yu, and Jiqing Han. Sharpe ratio-oriented active trading : A learning approach. In *Mexican International Conference on Artificial Intelligence*, pages 331–339. Springer, 2002.

[24] Young-Woo Seo, Joseph Giampapa, and Katia Sycara. Financial news analysis for intelligent portfolio management. Technical report, Carnegie-Mellon Univ Pittsburgh PA Robotics Inst, 2004.

[25] Marika Salo and Helena Haapio. Robo-advisors and investors : Enhancing human-robot interaction through information design, 2017.

[26] Claire Castanet and Camille Planes. Finance et intelligence artificielle : une révolution en marche. *Enjeux numériques*, page 15, 2018.

Application du Clustered Deep Q-Network aux Politiques Tarifaires

Simon Pageaud[1,2]
Vassilissa Lehoux[2]
Véronique Deslandres[1]
Salima Hassas[1]

[1] Université de Lyon - Université Claude Bernard Lyon 1 - LIRIS CNRS UMR 5205, Lyon, France
[2] NAVER LABS Europe - Meylan, France

simon.pageaud@liris.cnrs.fr

Résumé

Ce travail présente une nouvelle approche multi-agent et multi-niveaux, nommée Clustered Deep Q-Network (CDQN), avec pour objectif de répondre au problème de passage à l'échelle et de la non-stationnarité dans des contextes d'apprentissages décentralisés. Notre approche repose sur : 1) une gestion de chaque agent dans des clusters dynamiques avec une action jointe contrainte pour réduire la non-stationnarité et 2) l'attribution d'un score de confiance joint pour évaluer la contribution individuelle de chaque agent. Les expérimentations et les résultats sur une politique urbaine montrent que notre modèle permet une coordination efficace d'agents indépendants en utilisant l'apprentissage par renforcement profond multi-agent et la réutilisation d'expériences pour augmenter à la fois le gain individuel et global.

Mots-clés

Simulation multi-agent, Apprentissage par renforcement multi-agent, Apprentissage par renforcement profond, Politiques urbaines.

1 Introduction

Les progrès en apprentissage par renforcement profond a permis aux agents d'atteindre un niveau de contrôle humain sur de nombreux domaines comme les jeux Atari [14]. Cependant, ces jeux requièrent uniquement un agent apprenant. Les améliorations au Deep Q-Network (DQN) dans un contexte coopératif permet à deux agents de jouer à des jeux Atari ensemble [22]. Leur approche repose sur l'Independent Q-learning où les agents apprennent leur propre Q-fonctions indépendamment en parallèle [21]. Les limites de ce leur travail est l'hypothèse que chaque agent est dans un environnement entièrement observable. Les travaux avec les Q-networks profonds récurrents [7] permettent à des agents de jouer à des jeux dans des environnements 3D partiellement observables [9] dans un cadre mono-agent. Plus récemment, certains travaux se concentrent sur la combinaison des observations partielles avec un cadre multi-agent. Le Q-network profond récurrent distribué [3] fournit au cadre multi-agents des outils pour initier des protocoles de communications et s'organiser entre eux pour résoudre des énigmes.

Les cadres mono-agents passent difficilement à l'échelle avec l'augmentation de l'espace d'états. L'utilisation des approches multi-agents surmontent ce problème en utilisant des politiques décentralisées où les agents choisissent leurs actions uniquement à partir de leur historique local d'états/action. L'entrainement centralisé de politiques décentralisées est une approche classique de l'apprentissage par renforcement multi-agent [16] et l'apprentissage par renforcement profond [3]. Dans ce papier, nous considérons un apprentissage décentralisé sans informations supplémentaires sur l'état et sans communications entre agents. Une des approches les plus populaires en apprentissage par renforcement multi-agent, l'Independent Q-Learning [23], permet à l'agent d'apprendre ses Q-valeurs à partir de son ensemble d'états/actions en considérant les autres agents comme une partie de l'environnement. Le problème avec l'utilisation de l'apprentissage par renforcement profond avec les apprenants indépendants est que l'environnement reste non-stationnaire et impacte fortement la réutilisation d'expériences nécessaire au DQN.

Un autre défi lié à l'utilisation de politiques décentralisées est l'identification de la contribution des agents (*multi-agent credit assignment* [2], où les actions jointes ne génèrent généralement que des avantages globaux, ce qui rend difficile pour chaque agent d'inférer sa propre contribution au gain global. Une solution consiste à concevoir des fonctions de récompense individuelles pour chaque agent. Cependant, dans les contextes coopératifs, cela incite les agents à adopter des comportements individuels plutôt que coopératifs.

Les travaux précédents mentionnés dans les contextes partiellement observables et multi-agents reposent sur le fait que les agents ont une connaissance des autres agents et qu'ils peuvent communiquer entre eux. Nous considérons le cas où les agents ne peuvent pas communiquer et ont aucune connaissance des actions effectuées par les autres agents. Dans ce contexte, les agents doivent pouvoir se coordonner afin de maximiser une mesure de performance ciblée.

Ce document présente l'architecture CDQN (Clustered Deep Q-Network) permettant aux agents contrôleurs de gérer des clusters d'agents apprenants et de les coordonner efficacement pour améliorer leurs politiques. Ils apprennent des politiques décentralisées tandis que les agents contrôleurs ont pour rôle de réduire la non-stationnarité lors de la collecte de l'historique de l'action-observation locale. Nous montrons que la combinaison d'agents DQN indépendants avec des agents contrôleurs permet de résoudre le problème de la communication limitée dans des environnements multi-agents partiellement observables. Il permet également de réutiliser l'expérience et d'identifier la contribution individuelle de chaque agent plus facilement.

Nous introduisons quatre éléments d'importance primordiale pour l'efficacité du CDQN : i) un modèle de populations multi-niveaux et multi-agents dans lequel les agents contrôleurs gèrent un groupe d'agents apprenant ; ii) une attribution de récompense unique au lieu d'une récompense jointe. Les agents apprenants ont ainsi une récompense locale basée sur leurs propres observations ; iii) un score de confiance : la sélection et la gestion du comportement des agents apprenants via l'attribution de scores de confiance par les agents contrôleurs ; iv) des actions de contrôle permettant la réorganisation dynamique de groupes d'agents d'apprenants avec un mécanisme de fusion / séparation. La population d'agents de contrôle évolue à travers les ajouts et suppressions d'agents contrôleurs.

L'évaluation est réalisée au moyen de trois expériences basées sur des problèmes de politique de prix du monde réel utilisant les approches IQL et CDQN. Un intervenant recherche une répartition satisfaisante des prix de stationnement afin de maximiser le gain dans une ville avec un découpage approprié. Nos résultats montrent que le CDQN augmente efficacement le gain cumulé global et parvient à coordonner les agents apprenants de manière décentralisée, grâce à leur organisation dynamique en clusters sans aucune communication entre eux.

2 Travaux similaires

Les méthodes d'apprentissage par renforcement multi-agents étaient auparavant axées sur les méthodes tabulaires [1] avant d'utiliser des techniques d'apprentissage par renforcement profond. Ces méthodes permettent d'avoir des espaces d'actions et/ou d'états plus vastes [22]. Pour la planification d'une politique urbaine, nous considérons un cadre coopératif.

Dans les environnements d'apprentissage par renforcement multi-agents profonds, nous distinguons deux approches principales. UUne première s'attaque à l'apprentissage centralisé d'actions jointes, ce qui permet une meilleure coordination et évite la non-stationnarité. La limite de cette approche est la croissance exponentielle de l'espace d'actions jointes avec le nombre d'agents. Certains travaux antérieurs reposent sur la communication entre les agents [19] ou nécessitent une connaissance préalable étendue de la relation entre les agents [5].

Une autre approche considère l'apprentissage décentralisé d'actions jointes pour améliorer la gestion d'un espace d'actions jointes croissant. Le Q-Learning (IQL) indépendant [23] propose que chaque agent apprenne sa propre fonction Q à partir de son propre historique d'observation/action en considérant les autres agents comme faisant partie de l'environnement. Chaque agent i observe l'état actuel s_t, choisit une action $a_i(t)$ en fonction de sa politique, puis reçoit une récompense d'équipe r_t partagée entre tous les agents. Dans des environnements partiellement observables, l'IQL peut être implémenté en demandant à chaque agent d'apprendre son historique d'observation/action. L'IQL a récemment été utilisé dans les contextes d'apprentissage par renforcement profond [22] où il est combiné au DQN, produisant ainsi un DQN indépendant appliqué dans un jeu compétitif de Pong sur Atari. Tous les agents apprennent en parallèle une Q-fonction $Q_i(s, a_i; \theta_i)$ où θ_i est le paramètre du réseau de neurones. Cependant, le Q-learning indépendant rencontre des problèmes de convergence en raison de l'environnement non stationnaire induit par l'apprentissage des agents par rapport aux autres agents. Dans les approches tabulaires, les résultats empiriques montrent que la non-stationnarité est surmontée [12], mais reste présent dans l'apprentissage par renforcement profond. L'approche Deep Q-Network (DQN) [15] s'appuie sur la réutilisation d'expériences pour stocker l'expérience dans une mémoire tampon et l'utiliser pour entrainer le modèle [11]. Au fur et à mesure que les agents améliorent leurs politiques, les autres agents le font également, ce qui entraîne le problème du *moving target*, où l'expérience antérieure devient obsolète. Certains travaux proposaient un moyen d'éviter ce problème en limitant la taille du tampon d'expérience [10] ou même en les désactivant [3].

Il existe des approches hybrides utilisant un apprentissage centralisé et une exécution décentralisée pour entrainer des agents décentralisés [4, 6].

3 Clustered Deep Q-Network

Le Clustered Deep Q-Network (CDQN) est composé de populations multi-agents, multi-niveaux utilisant l'attribution d'un score de confiance et des récompenses locales pour coordonner les actions, dans des environnements partiellement observables et ne permettant pas de communication. Le but de la planification dans CDQN est de trouver une politique commune $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ qui maximise le gain cumulé global. Le modèle formel CDQN est une extension du Dec-POMDP factorisé [17].

3.1 Agents multi-niveaux

La coordination multi-agents de notre approche repose sur deux populations d'agents décentralisés. Nous considérons l'ensemble fini d'agents augmentés $\hat{\mathcal{D}} = \mathcal{D}^l \cup \mathcal{D}^c$ avec un ensemble fini d'agents $\mathcal{D}^l = (d^l_1, \ldots, d^l_m)$, appelée *apprenants* dont leur nombre m est fixé par la simulation, et l'ensemble fini d'agents $\mathcal{D}^c = (d^c_1, \ldots, d^c_n), n \in [1, m]$,,

appelés *contrôleurs*, chacun gèrant un clusters d'apprenants. Le nombre de contrôleurs varie au cours de la simulation avec l'ajout et la suppression des contrôleurs pour une meilleure répartition des clusters. À tout moment, un apprenant est géré par un seul contrôleur. Par conséquent, le nombre possible de configurations de contrôleurs correspond au nombre de partitions de l'ensemble des apprenants en sous-ensembles disjoints et non vides, c'est-à-dire le *mème* nombre de Bell.

Seuls les contrôleurs peuvent communiquer entre eux et les apprenants connaissent uniquement leur agent contrôleur courant.

L'ensemble fini d'actions augmentées est défini par $\hat{\mathcal{A}} = \mathcal{A}^c \cup \mathcal{A}^l$ où $\mathcal{A}^c$ est l'ensemble fini d'actions de contrôle disponibles pour les contrôleurs et $\mathcal{A}^l$ l'ensemble fini d'actions environnementales disponibles pour les apprenants. Une action de contrôle $a^c \in \mathcal{A}^c$ est une action qui façonne des clusters par opposition à une action environnementale $a^l \in \mathcal{A}^l$ qui modifie l'état. À chaque pas de temps, un apprenant i suggère une action $a_i^s \in \mathcal{A}_i^l$ à son contrôleur. Les actions suggérées par les apprenants d'un cluster $j \in [1, n]$ produit $\boldsymbol{a}_j^s$, l'*action jointe suggérée* des apprenants du cluster j. Actuellement, le contrôleur j utilise un système de vote pour choisir l'action élue a_j^l parmi les actions $\boldsymbol{a}_j^s$ avec le plus grand nombre de suggestions. L'action choisie résultante a_j^l est appliquée par chaque apprenant de ce cluster sur l'environnement formant l'action jointe *appliquée* $\boldsymbol{a}_j^a = (a_j^l)_{i \in j}$. C'est la première contribution du modèle CDQN : à un moment donné, chaque apprenant d'un même cluster agit dans la même direction en appliquant la même action, chacun ayant ainsi la même transition en mémoire. Cela réduit la non-stationnarité pendant l'apprentissage.

Un apprenant $d_i^l, i \in [1, m]$ a un ensemble de variables d'états x_i et un réseau représentant sa fonction de valeur individuelle $Q_i(o_i, a_i^l)$. Nous représentons le réseau de l'apprenant par un DQN recevant l'observation individuelle actuelle $o_i \in \mathcal{O}_i$ avec $\mathcal{O} = x_i \mathcal{O}_i$ l'ensemble des observations jointes et son action appliquée a_i^l comme entrée à chaque pas de temps (figure 1). La fonction d'observation O spécifie les probabilités d'observations $\Pr(o|a, s')$.

Chaque apprenant possède la même fonction de récompense mais reçoit un gain individuel $g_i^l(t)$ à chaque pas de temps en fonction de son état factorisé actuel $s_m \in \mathcal{S}$ avec $\mathcal{S} = \chi_1 \times \cdots \times \chi_{|\chi|}$ l'espace d'état factorisé. Les contrôleurs perçoivent le gain cumulé $g_j^c(t)$ de leurs apprenants comme unique perception. Ceci est motivé par les *différence de récompenses* [25] où chaque agent apprend d'une récompense modifiée plutôt que d'une récompense globale et réduit l'impact du problème de *multi-agent credit assignment*. Nous utilisons des *récompenses tronquées* [24] $r = \{-1, 1\}$ sur l'évolution relative du gain entre l'état précédent et l'état actuel. Cela présente deux avantages : (i) une représentation générique de l'attribution de récompense, quelle que soit la fonction de récompense utilisée et (ii) leur impact sur la vitesse d'apprentissage. Un apprenant d_i^l enregistre son gain local le plus élevé gb_i^l obtenu pendant l'exploration, son gain local au pas de temps précédent $g_i^l(t-1)$ et de son gain local actuel $g_i^l(t)$. La récompense est égale à 1 si $g_i^l(t) > g_i^l(t-1)$ ou si $g_i^l(t) = g_i^l(t-1)$ et $g_i^l(t) = gb_i^l$ et égal à -1 si $g_i^l(t) \leq g_i^l(t-1)$ et $g_i^l(t) < gb_i^l$. L'ensemble des récompenses factorisées est $\mathcal{R} = \{R_1, \ldots, R_n\}$.

Les contrôleurs utilisent un score de confiance $t_s \in [-1; 1]$ pour suivre le comportement de leurs apprenants avec l'action appliquée jointe $\boldsymbol{a}_j^a$. Le contrôleur j met à jour le score de confiance de l'apprenant i en utilisant la fonction de score de confiance incrémentielle $\tau : \mathcal{A}^l \times \mathcal{A}^l \times S \times S \rightarrow \{-0.1, 0, 0.1\}$, l'action proposée $a_i^s \in A^l$, l'action appliquée $a_j^a \in A^l$, le gain cumulé à l'itération précédente $g_j^c(t-1) \in S$ et le gain cumulatif de l'itération actuelle $g_j^c(t) \in S$. Alors que les récompenses sont fournies par l'environnement et utilisées par les apprenants, l'attribution du score de confiance incombe aux contrôleurs. La confiance est modifiée chaque fois qu'un apprenant propose une action. $\tau = 0$ si l'action proposée est choisie au hasard ; $\tau = 0.1$ si $a_i^s = a_j^l$ et $g_j^c(t) \geq g_j^c(t-1)$; $\tau = -0,1$ si $a_i^s = a_j^l$ et $g_j^c(t) < g_j^c(t-1)$ ou si $a_i^s \neq a_j^l$ et $g_j^c(t) \geq g_j^c(t-1)$. Le décrément $a_i^s \neq a_j^l$ et $g_j^c(t) \geq g_j^c(t-1)$ est plus subtile à comprendre, il traduit notre hypothèse que toute action augmentant le gain cumulé du contrôleur est la meilleure pour l'état actuel. Cependant, cela est faux si deux actions différentes ont le même effet sur les performances. Pour résoudre ce problème, nous avons élaboré un moyen d'identifier de telles actions similaires. Comme le contrôleur ne dispose pas d'informations claires sur l'état actuel, nous considérons plutôt les transitions d'actions offrant le meilleur gain. Ces transitions fournissent des informations sur l'état précédent et l'état actuel via l'observation du gain du contrôleur. Une transition est décrite par $tr = (g_j^c(t-1), a_j^l(t-1), g_j^c(t))$ et T est une fonction de transition spécifiant les probabilités de transition d'état $\Pr(s'|s, a)$. Deux actions sont considérées comme similaires si, pour deux transitions, tr_1 et tr_2, $g_j^c(t-1, tr_1) \neq g_j^c(t-1, tr_2)$ et $g_j^c(t, tr_1) = g_j^c(t, tr_2)$ et$g_j^c(t, tr_1) = gb_j^c$ où gb_j^c est le meilleur gain obtenu par le contrôleur j. Avec cette notion d'actions similaires, nous ajoutons $\tau = 0.1$ si $a_i^s \equiv a_i^l$ et $g_j^c(t) = gb_j^c$ pour le score de confiance.

3.2 Actions de contrôle

Les actions disponibles pour les contrôleurs sont définies par $\mathcal{A}^c = \{\texttt{Fusion}, \texttt{Separation}, \texttt{Do\_Nothing}\}$. Les contrôleurs utilisent ces actions pour gérer la répartition des apprenants dans les clusters afin d'améliorer leurs gains cumulés. Lorsqu'une `Fusion` ou une `Sparation` est appliquée, le score de confiance de chaque apprenant dans les clusters concernés est réinitialisé et la configuration avant l'application de l'action de contrôle est enregistrée. Avec ces actions vient un moyen d'identifier la pertinence de la nouvelle distribution sur l'environnement. Lorsqu'une action est appliquée, les contrôleurs disposent de plusieurs épisodes pour évaluer l'efficacité de l'action de contrôle.

Pendant cette période d'évaluation, aucune autre action de contrôle (fusion, séparation) ne sera acceptée sur le cluster en cours d'évéluation pour éviter de fausser l'évaluation de l'action de contrôle. Après évaluation, les contrôleurs concernés choisissent de conserver cette configuration ou de revenir à la configuration précédente. Dans tous les cas, toutes les actions de contrôle sont à nouveau disponibles. Nous voulons minimiser le nombre de clusters afin de réduire l'impact de la non-stationnarité dans un contexte d'apprentissage Q-Learning indépendant.

Toutefois pour être considérée comme efficace, une action de contrôle doit avoir un impact positif sur le gain cumulé global.

L'action `Separation` sépare les apprenants avec des scores de confiance positifs de ceux ayant des scores de confiance négatifs, créant ainsi deux clusters : l'actuel et un nouveau, augmentant d'un le nombre de contrôleurs. Le nombre d'apprenants reste le même. Pour autoriser une séparation, nous introduisons une zone dite de *clivage* dans laquelle aucun score de confiance ne doit entrer pendant un nombre spécifique d'itérations. Si un score de confiance entre dans la zone interdite, le compteur d'itérations permettant d'appliquer l'action de séparation est réinitialisé. Pour un contrôleur d_1^c qui se divise en contrôleurs d_2^c et d_3^c, l'action de séparation est efficace si $g^c_{d_1^c}(bs) < g^c_{d_2^c}(as) + g^c_{d_3^c}(as)$ est vrai, où bs correspond aux cas avant la séparation et as après la séparation. La séparation est considérée comme non pertinente si $g^c_{d_1^c}(bs) \geq g^c_{d_2^c}(as) + g^c_{d_3^c}(as)$. L'égalité des gains est jugée non pertinente car généralement il est plus efficace de minimiser le nombre de contrôleurs.

L'action de `Fusion` fusionne deux contrôleurs en vérifiant des séquences équivalentes d'actions d'environnement. Le cluster du contrôleur résultant contient les apprenants des deux contrôleurs confondus : leur nombre est ainsi réduit.

À chaque étape, les contrôleurs vérifient si les autres contrôleurs ont la même séquence d'actions environnementales auquel cas, ils appliquent une action de fusion. Lorsqu'un contrôleur reçoit son gain cumulé le plus élevé, il commence à rechercher des *séquences équivalentes* chez les autres contrôleurs. Les séquences sont dites équivalentes si elles sont composées des mêmes actions, même non ordonnées. Ainsi pour les trois actions environnementales a, b et c, les séquences abc et séquence acb sont équivalentes.

Cependant, considérer uniquement les séquences similaires ne suffit généralement pas pour réussir les fusions. Si nous considérons que les états initiaux des contrôleurs sont différents, ils subiront un changement dans leurs états. Ensuite, nous ne pouvons utiliser la fusion que si les états des contrôleurs sont homogènes, ce qui signifie que l'état factorisé a les mêmes variables d'état. Lorsque la simulation réinitialise les états de l'environnement, les contrôleurs se trouvent dans un état initial homogène. Ainsi, nous considérons des séquences d'action qui débutent de tout nouvel état aléatoire généré de l'environnement et se terminent quand le contrôleur pense avoir atteint son état optimal. Une deuxième contrainte est liée au fait que des séquences équivalentes peuvent conduire à des états différents, en fonction du contexte. C'est pourquoi pour la fusion, nous nous limiterons aux séquences équivalentes qui conduisent aux mêmes états.

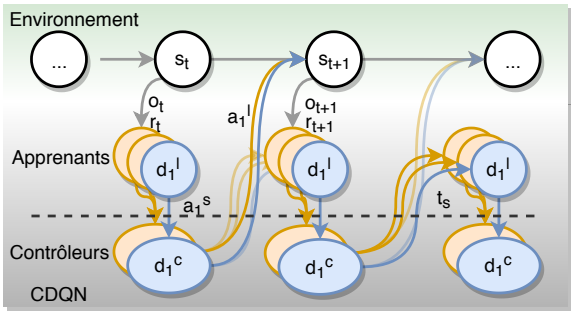


FIGURE 1 – Les apprenants sont séparés en deux clusters (bleu et orange). Un apprenant d_1^l observe o_t et suggère une action a_1^s à son contrôleur. Le contrôleur d_1^c, avec un système de vote, choisit une action a_1^l appliqué par les apprenants. À l'itération suivante, d_1^l observe o_{t+1} et reçoit la récompense r_{t+1}. Avec l'évolution de son gain, d_1^c modifie le score de confiance t_s des apprenants de son cluster.

Dans notre contexte, certaines actions environnementales peuvent avoir des actions qui annulent une action précédente, appelée action opposée. Par exemple, si nous considérons nos trois actions environnementales précédentes et posons a le contraire de b. Alors les séquences abc et c débutant dans le même état initial conduiraient au même état mais seraient considérées comme différentes. Cela signifie que le contrôleur avec la séquence c a un cluster avec de meilleures actions suggérées que le contrôleur avec la séquence abc. Nous considérons donc que les deux groupes ne devraient pas fusionner.

Pour deux contrôleurs d_1^c et d_2^c fusionnés en d_3^c, l'action de fusion est pertinente si $g^c_{d_1^c}(bf) + g^c_{d_2^c}(bf) \leq g^c_{d_3^c}(af)$ est vrai, où bf signifie avant la fusion et af pour après le `Fusion`. Cette règle a également pour conséquence de minimiser le nombre de contrôleurs car la fusion est considérée comme pertinente même si les gains avant et après `Fusion` sont égaux.

Pendant une période spécifiée, le contrôleur effectuant l'action observe si ses performances augmentent. À la fin de cette période, les contrôleurs utilisent les critères de pertinence introduits précédemment pour évaluer l'action et conserver la configuration actuelle ou revenir à la configuration précédant l'application de l'action de contrôle.

3.3 Algorithme du CDQN

Le CDQN lance une simulation avec la configuration initiale b_0 décrivant l'état factorisé de chaque apprenant et sa répartition dans les différents clusters. À chaque pas de temps, un apprenant i reçoit une observation o_t^i et suggère l'action a_i^s produisant la Q-valeur la plus élevée avec une

probabilité de $1 - \epsilon$. Chaque contrôleur reçoit des propositions de son cluster et choisit, avec son système de votes, l'action a_t^l à appliquer sur l'environnement par les apprenants du cluster. A l'itération suivante, les apprenants reçoivent une observation o_{t+1} et une récompense r_t. La récompense est déterminée par l'évolution du gain local entre l'itération actuelle et l'itération précédente. Les contrôleurs reçoivent leurs gains cumulés et mettent à jour le score de confiance ts_{t+1} de chaque apprenant de leur cluster.

Si tous les apprenants ont un score de confiance de 1 pendant plusieurs épisodes, le cluster est alors considéré comme pertinent car chaque apprenant propose l'action qui améliore le gain cumulé. D'autre part à chaque pas de temps, l'apprenant stocke une expérience (o_t, a_t, r_t, o_{t+1}) dans sa mémoire tampon.

Les contrôleurs ne peuvent appliquer qu'une et une seule action à chaque itération. Les agents apprenants sont pénalisés s'ils avaient suggéré une action différente de celle finalement choisie pour le cluster, et la mise à jour du score de confiance permet aux contrôleurs d'apprendre sur l'apport de chaque agent au gain du cluster.

Dans les environnements de vidéo-ludiques [13], les agents ont un état terminal lorsqu'ils apprennent. Il est souvent induit par la perte du jeu. Cela provoque un nouvel état initial et impose à l'agent d'accumuler de nouvelles connaissances via une exploration forcée. Dans notre contexte, l'environnement n'a pas d'état terminal. Si jamais c'est nécessaire, une idée serait d'ajouter un horizon fini avant lequel le contrôleur doit atteindre son état optimal, mais cela dépend de l'environnement. De plus, cela compromettrait l'autonomie de la simulation. Cependant, nous observons que sans états terminaux, lorsque les apprenants parviennent à une solution, ils l'exploitent et remplace la mémoire d'expériences par la meilleure solution, menant à l'oubli de la politique précédemment apprise. Par conséquent, nous introduisons un nouvel état factorisé aléatoire généré avec des variables d'états initiaux homogènes. Nous avons appelé un épisode le nombre d'itérations entre la réinitialisation de deux environnements. Au cours d'un épisode, chaque contrôleur tente de maximiser la somme des gains cumulés.

Le modèle CDQN est donc décrit par une version augmentée du Dec-POMDP factorisé $\langle \hat{\mathcal{D}}, \mathcal{S}, \hat{\mathcal{A}}, T, \mathcal{O}, O, \mathcal{R}, t, \tau, \gamma \rangle$ où $\gamma \in [0, 1)$ est un facteur de réduction.

4 Expérimentations

Le cadre principal de l'environnement est basé sur le modèle SmartGov [18] pour produire une simulation à plusieurs niveaux pour la régulation de la politique urbaine, dans laquelle des agents de personnalités différentes interagissent dans des environnements réalistes construits avec des données d'Open Street Map. Pour évaluer le CDQN, nous menons des expériences sur une instance de SmartGov et considérons les agents indépendants profonds (IDQL) comme base de comparaison.

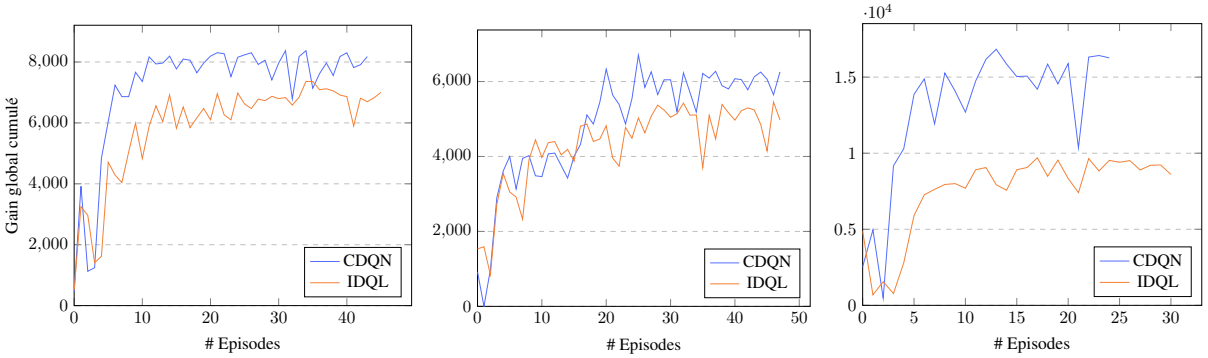
SmartGov fournit au CDQN les états de l'environnement en tant qu'entrées pour les apprenants et en sortie, il reçoit l'action jointe à appliquer sur l'environnement.

Nos expériences se situent dans un contexte de politique urbaine où les parties prenantes souhaitent améliorer des politiques urbaines. L'étude de cas concerne une politique de tarification des places de stationnement de la ville, pour des personnes se rendant chaque jour en ville, appelées navetteurs, en fonction de la personnalité et des préférences de stationnement des conducteurs. Chaque agent local contrôle le prix du stationnement d'un certain nombre de places d'un périmètre restreint, avec un front de rue composée de routes, d'emplacements payants et de bâtiments hétérogènes (maisons, bureaux, magasins, etc.) [18]. Les agents locaux appliquent des actions locales (augmenter, diminuer, etc., décrites ci-après), en fonction de la perception limitée de leur environnement. Ils doivent apprendre la politique qui optimise les récompenses. Dans ce contexte, les apprenants ont un espace d'états défini par des perceptions limitées et des variables d'état telles que $\chi = \{\texttt{price}, \texttt{occupation}\}$. La fonction de récompense choisie ici est le gain représenté par le produit du prix par place et du nombre de places occupées. Nous appelons *quartier* un espace où chaque navetteur a la même personnalité et les mêmes attentes.Nous notons C_1 et C_2 les ensembles de navetteurs avec des personnalités identiques. Sur la base de la configuration initiale du système, un cluster représente un ensemble d'apprenants ayant le même prix de stationnement initial. Le décideur politique a pour objectif de maximiser le gain cumulé global obtenu par le produit du prix et de l'occupation des emplacements, et il souhaite également que les clusters proposent un découpage de la politique de prix du stationnement en vigueur dans la ville. Le décideur juge de la pertinence du découpage de la politique proposée. Le CDQN gère efficacement ses clusters pour maximiser le gain cumulé sur ceux-ci.

Pour obtenir des résultats réalistes, nous avons utilisé un ensemble de trajectoires et de populations du monde réel pour essayer les actions de contrôle `Fusion` et `Separation` de manière indépendante dans un premier temps, puis ensembles. L'objectif principal de chaque expérience est de maximiser le gain global cumulé et de minimiser le nombre de clusters, les deux critères étant pris en compte par ordre lexicographique.

L'ensemble des actions disponibles pour l'apprenant i est $\mathcal{A}_i^l = \{\nearrow, \searrow, =, +, -\}$ où $\nearrow$ augmente le prix d'un emplacement, $\searrow$ le baisse, $+$ ajoute un emplacement, $-$ en supprime un et $=$ n'a aucun impact. Dans notre contexte, l'action $-$ n'est jamais pertinente, mais elle est nécessaire pour évaluer la robustesse de la méthode actuelle et d'ajouter du bruit. Le nombre d'apprenants par clusters est déterminé par la topologie de la carte.

Nous avons évalué l'efficacité des actions `separation` et `fusion` sur trois scénarios différents (Tableau 1). Pour chaque scénario, nous avons comparé les performances du CDQN avec des IDQL dans lequel les agents IDQL sont des apprenants sans contrôleur.



(a) `Fusion` (b) `Separation` (c) `Fusion` et `Separation`

FIGURE 2 – Gain global cumulé pour l'IDQL et le CDQN sur les trois scénarios.

Actions	Contrôleurs	Apprenants	Navetteurs
`Fusion`	2	8/9	C1
`Separation`	1	17	C1, C2
Both	2	14/18	C1, C2

TABLE 1 – Apprenants par contrôleur pour chaque scénario et distribution des navetteurs.

Le scénario de fusion vise à identifier des séquences d'actions équivalentes pour fusionner et minimiser efficacement le nombre de clusters. Les résultats attendus pour ce scénario sont un cluster avec le gain cumulé le plus élevé.

Le scénario de séparation montre comment l'attribution de scores de confiance est utilisée pour identifier les quartiers et pour appliquer une séparation de cluster appropriée. Les résultats attendus pour ce scénario sont deux clusters avec un gain cumulé global plus élevé qu'avant la séparation du contrôleur.

L'objectif du scénario avec les deux actions est d'identifier les apprenants associés à chaque quartier et d'augmenter ainsi le gain cumulé en les regroupant efficacement. Le résultat attendu est une séparation par contrôleur, puis deux fusions, ou une fusion/séparation.

Le dernier scénario montre comment une combinaison des deux actions de contrôle permet d'augmenter le gain cumulé global et d'assurer une gestion efficace des clusters dans un cadre d'apprentissage par renforcement multi-agent utilisant des DQN indépendants et la réutilisation d'expériences.

Tous les réseaux sont entrainés à l'aide de l'algorithme ADAM [8] et de mini-lots de 64 échantillons. Le facteur de réduction a été défini par $\gamma = 0.95$. Nous avons utilisé une règle ϵ-greedy lors de la formation, où ϵ décroissait de 1,0 à 0,1 chaque fois que les apprenants proposaient une action avec $\epsilon = \epsilon \times \epsilon_{decay}$ et où $\epsilon_{\text{decay}} = 0.995$.

5 Résultats

La figure 2a représente deux contrôleurs possédant les mêmes types navetteurs. Seule l'action de fusion est disponible. Nous observons que le CDQN atteint un meilleur gain cumulé global que l'IDQL avec l'aide de la configuration initiale du cluster. La non-stationnarité est réduite, cela est due aux apprenants d'un même cluster qui appliquent une action identique, et la vitesse d'apprentissage permet d'atteindre une stratégie individuelle optimale plus rapidement que les IDQL. Après un certain nombre d'itérations, les apprenants proposent des actions qui augmentent les gains. Lorsque les deux contrôleurs ont la même séquence un certain nombre de fois, ils fusionnent à l'épisode 5. a configuration est conservée après l'évaluation montrant que la fusion est efficace.

Dans la figure 2b, nous observons comment le score de confiance est utilisé pour proposer une séparation efficace des clusters. Au début de la simulation, CDQN et IQL explorent l'environnement. Avant 15 épisodes, un cluster identifie un groupe d'apprenants en fonction de l'évolution du score de confiance. À ce moment, IDQL obtient de meilleurs résultats avec des actions individuelles plus faciles à appliquer. L'impact de la séparation après 15 épisodes est immédiat, car les apprenants proposent des actions plus pertinentes dans un cluster, ce qui permet à CDQN d'obtenir de meilleures performances que IDQL avec la nouvelle configuration des clusters. Certains agents apprennent que les actions appliquées par leur contrôleur j n'améliorent pas leur gain local, alors que d'autres apprennent que les actions appliquées augmentent leur gain local. L'action ($\nearrow$) améliore le gain et donc le score de confiances des apprenants proposant cette action. Le phénomène s'inverse avec ceux qui proposent l'action ($\searrow$) . À l'aide des règles spécifiées, le contrôleur effectue une séparation à l'épisode 15 en conservant les apprenants avec un score de confiance positif et en créant un nouveau contrôleur qui reçoit un groupe d'apprenants dotés d'un score de confiance négatif. ces scores sont reinitialisés une fois le nouveau cluster constitué. Désormais, les contrôleurs

peuvent appliquer différentes actions environnementales sur l'environnement en même temps. Un contrôleur applique donc ($\nearrow$) et l'autre ($\searrow$) et la somme des gains devient plus élevée après la séparation, la configuration est donc conservée.
Avec les deux scénarios, nous voyons que la fusion et la séparation sont pertinentes pour le contrôle dynamique et le regroupement des apprenants dans l'environnement. L'action de fusion utilise les gains et la séquence d'actions équivalentes, et l'action de séparation repose sur le score de confiance et l'apprentissage indépendant de chaque apprenant. Vers la fin de la simulation, les apprenants proposent des actions qui maintiennent le contrôleur dans un état stable et optimisent les gains à chaque étape. L'utilisation du score de confiance en plus de l'attribution de récompenses aide le système à apprendre des informations locales sur l'environnement.
Le dernier scénario (figure 2c) contient un total de 32 apprenants répartis dans deux groupes. Les contrôleurs fusionnent à l'épisode 4 et le cluster résultant et le cluster résultant obtient de meilleurs résultats. Après un certain temps d'exploration, le contrôleur obtient de meilleurs résultats et identifie les apprenants responsables de cette amélioration. L'écart entre les scores de confiance permet une nouvelle action de contrôle avec une séparation en deux contrôleurs qui, une fois validés, modifient l'environnement pour atteindre un nouveau gain global maximal.
Dans les trois cas, la convergence de la simulation est observée avec l'évolution des scores de confiance et du gain.

6 Discussion

Les résultats affichés dans la section précédente montrent que notre approche permet aux agents de se coordonner sans communication ni connaissance des autres agents, avec des résultats plus pertinents que pour IDQL : le CDQN converge vers un état stable avec un gain global plus élevé.
Un point important est que les apprenants de notre modèle n'ont pas à considérer l'influence des actions des autres apprenants avec la gestion des clusters par les contrôleurs. La réorganisation des clusters diminue l'espace d'action joint à prendre en compte dans le processus d'apprentissage. À l'exception des approches IDQL, nous avons, à notre connaissance, aucune approche existante similaire pour comparer nos résultats avec le contexte initial considéré.
Une perspective d'amélioration serait d'utiliser différentes combinaisons de vecteurs d'entrée pour étudier leur impact sur l'apprentissage des apprenants, par exemple avec un niveau moindre de connaissance (ici disposer de moins d'informations sur les personnalités et préférences des navetteurs). Comme chaque apprenant enregistre son modèle de réseau neuronal pendant la simulation, nous pourrions aussi envisager de réutiliser un apprentissage déjà effectué et modifier la configuration de l'environnement pour évaluer la capacité d'adaptation des apprenants avec la réutilisation d'expérience prioritaire [20]. Nous pourrions également essayer différents algorithmes d'apprentissage par renforcement en profond pour observer l'impact sur la vitesse de convergence et les configurations de clusters. Enfin, une autre piste consisterait à configurer le système initial avec des états non homogènes.

7 Conclusion

Dans cet article, nous avons présenté le Clustered Deep Q-Network, une architecture d'apprentissage par renforcement multi-agents pour apprendre des politiques décentralisées dans des environnements partiellement observables sans communication entre les agents. Le modèle proposé distingue différents niveaux d'agents et de récompenses. La récompense de l'apprenant façonne les comportements individuels alors que la configuration des clusters évolue en fonction des scores de confiance des apprenants. Ce travail propose la recherche de configurations de clusters hiérarchiques d'agents avec des actions de fusion et de séparation, et l'apprentissage de leurs Q-valeurs individuelles. Une gestion efficace des agents de contrôle via des clusters d'apprenants contribue à augmenter les gains locaux et globaux. Les résultats sur un problème de politique tarifaire montrent comment la combinaison des scores de confiance et des récompenses individuelles permet un apprentissage efficace et une coordination émergente entre apprenants. Nous comparons ces résultats à un apprentissage profond individuel avec DQN afin de démontrer comment nous pouvons surmonter la non-stationnarité dans un apprentissage décentralisé. Toutes les applications de planification décentralisées ne pourront pas utiliser notre approche, du fait du choix d'une action unique par cluster pour réduire la non-stationnarité. Cependant, elle reste valable pour de nombreux domaines tels que l'élaboration de politiques, urbaines ou autre (santé, etc.), qui ont besoin d'apprendre vite et efficacement. Nous réfléchissons actuellement à la notion d'actions hiérarchiques, impactant la manière dont les apprenants pourraient agir indépendamment des autres apprenants du même groupe sans altérer le mécanisme d'attribution du score de confiance.

8 Remerciements

Ce projet est financé par la Région Auvergne-Rhône-Alpes (Contrat ADR ARC7). Il est le fruit d'une collaboration entre le LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) à Lyon et NLE (NAVER LABS Europe), anciennement XRCE (Xerox ResearchCenter Europe) à Meylan.

Références

[1] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems Man and Cybernetics*, pages 156–172, 2008.

[2] Yu-Han Chang, Tracey Ho, and Leslie Pack Kaelbling. All learning is local : Multi-agent learning in

global reward games. *Advances in neural information processing systems*, pages 807–814, 2004.

[3] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks. *arXiv preprint arXiv :1602.02672*, 2016.

[4] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradient. *arXiv preprint arXiv :1705.08926*, 2017.

[5] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. *Advances in Neural Information Processing Systems NIPS-14*, 2001.

[6] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative Multi-Agent Control Using Deep Reinforcement Learning. *International Conference on Autonomous Agents and Multiagent Systems*, 2017.

[7] Matthew Hausknecht and Peter Stone. Deep Recurrent Q-Learning for Partially Observable MDPs. *Association for the Advancement of Artificial Intelligence*, 2015.

[8] Diederik P. Kingma and Jimmy Lei Ba. ADAM : A Method for Stochastic Optimization. *International conference on machine learning*, 2015.

[9] Guillaume Lample and Devendra Singh Chaplot. Playing FPS Games with Deep Reinforcement Learning. *Association for the Advancement of Artificial Intelligence*, pages 2140–2146, 2017.

[10] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. pages 464–473.

[11] Long-Ji Lin. *Reinforcement Learning for Robots Using Neural Networks*. PhD thesis, Carnegie Mellon University, 1993.

[12] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games : a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1) :1–31, 2012.

[13] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *International conference on machine learning*, pages 1928–1937, 2016.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv :1312.5602*, 2013.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540) :529, 2015.

[16] Frans A. Oliehoek, Matthijs T.J. Spaan, Shimon Whiteson, and Nikos Vlassis. Exploiting locality of interaction in factored dec-pomdps. *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[17] Frans A. Oliehoek, Shimon Whiteson, and Matthijs T.J. Spaan. Approximate solutions for factored dec-pomdps with many agents. *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, May 2013.

[18] Simon Pageaud, Véronique Deslandres, Vassilissa Lehoux, and Salima Hassas. Couplage de simulations multi-agents pour la conception de politiques urbaines. *26èmes Journées Francophones sur les Systèmes Multi-Agents*, 2018.

[19] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv :1703.10069*, 2017.

[20] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. *International Conference on Learning Representations*, 2016.

[21] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[22] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent Cooperation and Competition with Deep Reinforcement Learning. *PloS ONE*, 12(4), April 2017.

[23] Ming Tan. Multi-agent reinforcement learning : Independent vs. cooperative agents. *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[24] Hado van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. Learning Values Across many Orders of Magnitude. *Advances in Neural Information Processing Systems*, 2016.

[25] David H. Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Modeling complexity in economic and social systems*, pages 355–369, 2002.

TSRuleGrowth : Extraction de règles de prédiction semi-ordonnées à partir d'une série temporelle d'éléments discrets, application dans un contexte d'intelligence ambiante

Benoit Vuillemin[1,2] Lionel Delphin-Poulat[2] Rozenn Nicol[2] Laetitia Matignon[1] Salima Hassas[1]

[1] Univ Lyon, Université Lyon 1, CNRS, LIRIS, UMR5205, F-69622, France
[2] Orange Labs, Lannion, France

{benoit.vuillemin, laetitia.matignon, salima.hassas}@liris.cnrs.fr
{lionel.delphinpoulat, rozenn.nicol}@orange.com

Résumé

Cet article présente un nouvel algorithme : TSRuleGrowth, recherchant des règles de prédiction semi-ordonnées sur une série temporelle. Cet algorithme reprend les principes de l'état de l'art de la fouille de règles et les applique aux séries temporelles via une nouvelle notion de support. Nous l'appliquons à des données réelles provenant d'un environnement connecté. Cet algorithme extrait les habitudes des utilisateurs à travers différents objets connectés.

Mots Clef

Fouille de règles, Intelligence Ambiante, Habitudes, Automatisation, Support, Séries temporelles

Abstract

This paper presents a new algorithm : TSRuleGrowth, looking for partially-ordered rules over a time series. This algorithm takes principles from the state of the art of rule mining and applies them to time series via a new notion of support. We apply this algorithm to real data from a connected environment, which extract user habits through different connected objects.

Keywords

Rule Mining, Ambient Intelligence, Habits, Automation, Support, Time series

1 Introduction

La fouille de règles de prédiction sur une série temporelle est un problème majeur en data mining. Utilisé notamment dans l'analyse du cours des actions et la recommandation d'achats, ce problème est de plus en plus étudié à mesure que le domaine de l'intelligence ambiante (AmI) se développe. L'AmI est la fusion entre l'intelligence artificielle et l'internet des objets, et peut être décrit comme : "Un environnement numérique qui soutient les personnes dans leur vie quotidienne de façon proactive, mais raisonnable" [2]. Ce travail entre dans le domaine de l'AmI : nous voulons faire un système qui trouve les habitudes des utilisateurs dans un environnement connecté, autrement dit un environnement dans lequel des objets connectés sont présents, afin de fournir aux utilisateurs des automatisations. Par exemple, si une personne allume habituellement la lumière après être entrée chez elle, ce qui peut être vu par une ampoule connectée et un capteur de porte, le système pourrait détecter cette règle de prédiction et proposer d'allumer la lumière à chaque entrée de l'utilisateur.
Cet article décrit un nouvel algorithme utilisé dans notre système d'AmI, qui vise à rechercher les règles de prédiction sur une série temporelle. Ici, la série temporelle représente les événements envoyés par les objets connectés. Ces règles de prédiction seront ensuite proposées aux utilisateurs comme suggestions d'automatisation. Dans le cadre de cet article, les séries temporelles sont composées uniquement de valeurs catégorielles, plutôt que continues. Les données peuvent survenir à tout moment, c'est-à-dire qu'il n'y a pas de fréquence d'échantillonnage fixe dans les séries temporelles. La structure de ces règles est expliquée, ainsi que l'état de l'art des domaines concernés, ce qui justifie les choix effectués pour cet algorithme.

2 Contexte et définitions

2.1 Entrée

Il existe deux types d'objets connectés : les capteurs et les actionneurs. Les **capteurs** observent des grandeurs physiques de l'environnement. Un capteur renvoie des événements, correspondant aux changements d'état de la variable observée. Un capteur d'ouverture de porte, par exemple, peut renvoyer des événements d'ouverture et de fermeture. Les capteurs peuvent mesurer des variables **continues** ou **catégorielles**. Par exemple, la température d'une pièce, exprimée en degrés, peut être considérée comme variable continue, tandis que la sélection d'une station radio ou l'ouverture d'une porte sont des variables catégorielles. **Dans cette étude, seuls les capteurs renvoyant des variables catégorielles sont considérés.** Les

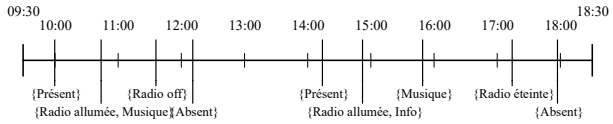


FIGURE 1 – Représentation d'une série temporelle

actionneurs agissent sur l'environnement. Un actionneur renvoie un événement lorsqu'il a effectué une action. Par exemple, un volet connecté renvoie un événement lorsqu'il s'ouvre ou se ferme. De la même manière que les capteurs, les actionneurs peuvent effectuer des actions catégorielles, comme ouvrir un volet, ou continues, comme augmenter la température à une certaine valeur. **Comme pour les capteurs, seuls les actionneurs qui effectuent des actions catégorielles sont pris en compte.**

Chaque objet, qu'il soit un capteur ou un actionneur, renvoie des événements primaires. Dans cet article, ils sont nommés **éléments**. Tous les éléments envoyés par tous les objets sont regroupés dans un ensemble noté E.

Prenons l'exemple d'une pièce contenant deux objets connectés : un détecteur de présence, permettant de savoir si une personne se trouve dans une pièce ou non, et un actionneur : une radio. Le détecteur de présence peut renvoyer les éléments suivants : "Présent" et "Absent". La radio peut agir de deux façons : son état peut être "Radio allumée" or "Radio éteinte", et elle peut sélectionner l'une des stations suivantes : "Musique", "Info", "Débat". Ainsi, l'ensemble de tous les éléments est E = {Présent, Absent, Radio allumée, Radio éteinte, Musique, Info, Débat}.

Le système AmI que nous proposons recueille des flux de données à partir de plusieurs objets connectés. Chaque flux de données est composé d'une succession d'éléments, dont chacun peut se produire une ou plusieurs fois. Chaque occurrence est estampillée d'un timestamp, une donnée temporelle. Ainsi, chaque élément est potentiellement associé à plusieurs timestamps correspondant à ses multiples occurrences. Pour la suite du traitement, toutes les données collectées par les différents capteurs sont regroupées en une seule **série temporelle**. En d'autres termes, une série temporelle est obtenue par une juxtaposition dans le temps d'éléments fournis par tous les objets. Elle est notée TS=$\langle(t_1, I_1), ..., (t_n, I_n)\rangle, I_1, ..., I_n \subseteq E$, où :

- t_i est un **timestamp**, un point fixe dans le temps.
- $I_i \subseteq E$ est un **itemset**. C'est l'ensemble des éléments uniques provenant de E observés au timestamp t_i.

Il est à noter qu'un élément ne peut être vu qu'une seule fois dans un itemset. De plus, les timestamps ne sont pas nécessairement espacés de de manière uniforme. La figure 1 est un exemple de série temporelle créée à partir de l'environnement connecté mentionné dans la section 2.1. Sa représentation mathématique est :

$TS = \langle$(10 :00 am, {Présent}), (10 :44 am, {Radio allumée, Musique}), (11 :36am, {Radio éteinte}), (12 :11 am, {Absent}), (2 :14 pm, {Présent}), (2 :52 pm, {Radio allumée, Informations}), (3 :49 pm, {Musique}), (5 :14 pm, {Radio éteinte}), (5 :57 pm, {Absent})$\rangle$.

Cette série temporelle représente certaines actions d'un utilisateur dans l'environnement. La section suivante détaille ce que le système doit trouver dans une série temporelle.

2.2 Sortie

Le système proposé doit trouver des **règles de prédiction**, pour exprimer les habitudes observées. Une règle de prédiction est notée $R : E_c \Rightarrow E_p$, où E_c est la **condition**, et E_p est la **prédiction** de la règle. R décrit que si E_c est observé, E_p sera observé après un certain temps.

Dans le cas d'utilisation étudié, nous voulons limiter la recherche de règles **pour lesquelles la partie prédiction E_p doit être composée uniquement d'éléments provenant d'actionneurs**. En effet, la recherche de règles étant fortement combinatoire, cela permet de limiter cet aspect tout en s'adaptant au cas d'utilisation : proposer des automatisations d'actions en fonction de situations. D'après la série temporelle de la section 2.1, une règle peut être {Présent} $\Rightarrow$ {Radio allumée}. Il s'agit d'une règle basique, où la condition et la prédiction ne sont composés que d'un seul élément. Nous ne voulons pas, par exemple, trouver la règle {Radio éteinte} $\Rightarrow$ {Absent} car la partie prédiction, {Absent}, provient d'un capteur (de présence) et non d'actionneurs (comme la radio).

Une règle, pour être validée, doit être fréquente et fiable. Il est aussi possible de faire de la détection d'anomalies, c'est-à-dire rechercher des règles très peu fréquentes et très intéressantes, mais cela n'entre pas dans le cadre de cet article.

Plusieurs types de règles de prédiction sont possibles [8] :

- **Règles séquentielles complètement ordonnées**, où la condition E_c et la prédiction E_p sont des séquences, c'est-à-dire des successions temporelles d'éléments,
- **Règles séquentielles semi-ordonnées** [9], où la condition E_c et la prédiction E_p ne sont pas ordonnées. Un ordre existe toujours entre la condition et la prédiction, d'où le nom "semi-ordonné". Deux structures mathématiques sont possibles pour E_c et E_p : les **ensembles**, où un élément ne peut apparaître qu'une fois, et les **multiensembles**, où plusieurs instances d'éléments sont possibles. Le nombre d'instances d'un élément dans un multiensemble est appelé **multiplicité**. Par exemple, la multiplicité de l'élément x dans le multiensemble $\{x, x, y\}$ est 2.

Après avoir testé chacun des types, nous avons choisi d'utiliser des **règles séquentielles semi-ordonnées contenant des multiensembles**. Le problème avec les règles séquentielles complètement ordonnées est que plusieurs règles peuvent caractériser la même situation. Par définition, l'extraction de règles semi-ordonnées génère moins de candidats et moins de règles. De plus, elles sont décrites comme étant plus générales, avec une plus grande précision de prédiction que l'autre type de règles, et elles ont été utilisées dans des applications réelles [8]. Dans le cas d'utilisation proposé, la description d'une situation ne nécessite pas né-

cessairement un ordre, mais la multiplicité d'un élément peut être significative. Pour expliquer ce choix, prenons l'exemple d'une lampe à détection sonore. Lorsque l'on tape deux fois dans les mains, que l'on fait deux fois le même son, la lampe s'allume.

3 Travaux associés

Comme dit ci-dessus, le système proposé doit rechercher des règles de prédiction semi-ordonnées sur une série temporelle d'éléments provenant de capteurs et d'actionneurs. Ainsi, dans l'état de l'art, deux grands domaines de recherche doivent être considérés : la fouille de règles sur les séries temporelles et la fouille de règles semi-ordonnées. Mais avant cela, rappelons quelques définitions.
Une règle de prédiction $R : E_c \Rightarrow E_p$ doit être fréquente et fiable. Pour vérifier qu'une règle est fréquente, son **support** est calculé. La notion de support dépend de la structure des données en entrée, mais il estime la fréquence d'une règle, d'un ensemble d'éléments ou d'un élément. Pour vérifier qu'une règle est fiable, son **intérêt** est calculé. Plusieurs mesures permettent de connaître l'intérêt d'une règle. La plus connue est la confiance [3], mais il existe des alternatives, telles que la conviction, le lift [3] ou netconf [7, 1]. Ces mesures dépendent des supports de R, E_c et E_p.

3.1 Fouille de règles sur séries temporelles

[5] propose un système de fouille de règles basiques, où un élément en prédit un autre, sur une séquence d'éléments. Ces éléments représentent des variations basiques de données boursières. Il peut aussi rechercher des règles plus complexes, où la condition est une séquence. Ce système permet donc de trouver des règles sur une série temporelle. Cependant, il cherche des règles complètement ordonnées, plutôt que des règles semi-ordonnées. De plus, la partie prédiction des règles est limitée à un seul élément, une limitation que nous voulons éviter dans ce système d'AmI.
[11] peut être considéré comme une amélioration de [5], car il recherche des règles où la prédiction n'est pas limitée à un élément. Mais, recherchant des règles complètement ordonnées, il ne peut pas être appliqué dans notre cas.
[10] introduit une notion de support pour série temporelle, via une fenêtre glissante à durée fixe. Le support d'un élément, d'un ensemble d'éléments ou d'une règle est le nombre de fenêtres dans lesquelles cet élément, ensemble ou règle apparaît. L'algorithme trouve des règles semi-ordonnées, en cherchant des ensembles fréquents d'éléments, puis en les combinant pour générer des règles. D'autres algorithmes utilisent ce support, dont [6] qui trouve des règles dont la prédiction est composée d'un élément.
L'algorithme présenté dans [10] peut donc s'appliquer dans notre cas. Cependant, la définition du support peut être problématique. En effet, les éléments E_p étant strictement postérieurs à ceux de E_c, le nombre de fenêtres recouvrant la règle R est strictement inférieur à celui de E_c. Ainsi, même si E_p apparaît toujours après E_c dans un temps donné, le support de la règle est inférieur à celui de E_c, réduisant son intérêt. De plus, comme la recherche est structurée en deux étapes (recherche d'ensembles fréquents, puis recherche de règles), l'algorithme n'est pas totalement efficace.

3.2 Fouille de règles semi-ordonnées

A notre connaissance, peu d'algorithmes de fouille de règles semi-ordonnées existent. Les plus connues sont RuleGrowth [9], et ses variations, TRuleGrowth [9] et ERMiner [8]. Ces algorithmes prennent en entrée un ensemble de transactions. Une transaction est une séquence d'ensembles d'éléments appelés itemsets, ordonnée dans le temps, mais contrairement aux séries temporelles, sans timestamp associé. Pour vérifier qu'une règle est fréquente, ces algorithmes calculent son support, lié à la structure des transactions. Pour vérifier qu'une règle est fiable, ils calculent son intérêt, lié au support de la règle, de la condition et de la prédiction. TRuleGrowth est une extension de RuleGrowth qui accepte la contrainte d'une fenêtre glissante, définie comme un nombre d'itemsets consécutifs. Il permet de limiter la recherche aux règles qui ne peuvent se produire que dans cette fenêtre. ERMiner est une version plus efficace de RuleGrowth, mais sans fenêtre glissante.
Ces algorithmes cherchent directement des règles de prédiction, contrairement à [10] qui recherche des ensembles fréquents et recherche ensuite des règles sur ces ensembles. De plus, l'architecture commune à RuleGrowth, TRuleGrowth et ERMiner permet de limiter la taille des règles recherchées. Il est également possible de limiter les éléments sur lesquels les règles sont recherchées. Dans notre cas d'utilisation, nous recherchons des règles dont les prédictions ne sont faites qu'à partir d'actionneurs. Ces algorithmes permettent cette limitation directement dans la recherche, ce qui réduit le temps total de calcul.
Cependant, ils ont un problème majeur dans le cas d'utilisation visé : ils prennent des transactions en entrée, au lieu d'une série temporelle. La notion de support dépend directement de la structure des transactions, et ne peut être appliquée en tant que telle sur une série temporelle. Ainsi, malgré les avantages de ces algorithmes, ils ne peuvent être appliqués en tant que tels à nos données d'entrée.

3.3 Problèmes scientifiques

A notre connaissance, les algorithmes de l'état de l'art ne sont pas assez satisfaisants pour résoudre le problème initial. Deux problèmes majeurs doivent être résolus :

1. Comment définir le support d'une règle dans une série temporelle qui évite le problème de la section 3.1 ?
2. Comment construire un algorithme de fouille de règles sur cette nouvelle mesure de support ?

De plus, cet algorithme doit aborder les points suivants :

3. Comment limiter la durée des règles trouvées ?
4. Comment limiter la recherche à certains éléments dans la condition ou de la prédiction ?
5. Comment éviter qu'une règle soit trouvée deux fois ?

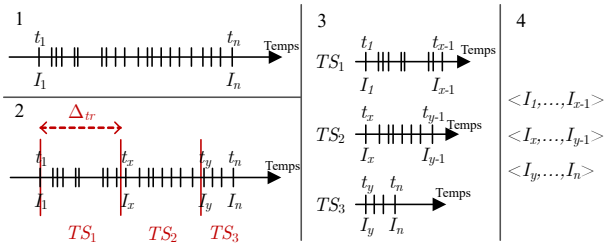


FIGURE 2 – Exemple de conversion d'une série temporelle en transactions

RuleGrowth répond aux points 4 et 5, mais ne prend que des transactions en entrée. TRuleGrowth, utilise une fenêtre glissante qui peut être utilisée pour répondre au problème 3 avec quelques modifications. Notre système utilise les principes de TRuleGrowth, mais les applique aux séries temporelles, pour traiter les deux premiers problèmes. La section suivante explique ces principes en détail.

4 TRuleGrowth

4.1 Principes

TRuleGrowth est un algorithme de recherche de règles semi-séquentielles sur des transactions. Il prend en entrée :

- $TR = \{tr_1, ..., tr_{nb_{tr}}\}$: Un ensemble de transactions
- min_{sup} : La valeur minimale du support pour qu'une règle soit considérée comme fréquente
- min_{int} : La valeur minimale de l'intérêt pour qu'une règle soit acceptée comme règle de prédiction
- $window$: Un nombre maximal d'itemsets consécutifs dans lesquels les règles doivent se produire

Cet algorithme produit des règles semi-ordonnées, dont la condition et la prédiction sont des ensembles, comme indiqué dans la section 2. Le support permet de vérifier qu'une règle est fréquente. Il existe deux types de support : absolu et relatif. Le support absolu sup d'une règle est le nombre de transactions contenant cette règle. Il en va de même pour les éléments et les ensembles. Le support relatif $relSup$ d'un élément, règle ou ensemble est son support absolu divisé par le nombre total de transactions. Puis, l'intérêt d'une règle, via la mesure de confiance $conf$ [3], permet vérifier sa fiabilité. Pour une règle $R : E_c \Rightarrow E_p$,

$$conf(R : E_c \Rightarrow E_p) = \frac{relSup(E_c \Rightarrow E_p)}{relSup(E_c)} \qquad (1)$$

TRuleGrowth recherche les règles de manière incrémentale. Il cherche d'abord des règles basiques, dont la condition et la prédiction sont composées d'un seul élément. Puis, il essaie de les étendre progressivement, en y ajoutant un élément. TRuleGrowth est composé de trois sous-algorithmes : l'algorithme principal, ExpandLeft et ExpandRight. L'algorithme principal recherche les règles basiques dans la fenêtre, dont la condition et la prédiction sont composées d'un seul élément. Si une règle a un support supérieur à min_{sup}, l'algorithme principal va essayer de l'étendre dans sa partie condition via ExpandLeft, et dans sa partie prédiction, via ExpandRight.
ExpandLeft et ExpandRight tentent d'étendre la règle en y ajoutant un élément, puis en calculant le support de la nouvelle règle formée. Si ce support est supérieur à min_{sup}, ExpandLeft et ExpandRight s'appelleront mutuellement, par récursivité, pour essayer de développer à nouveau la règle en ajoutant un élément. Pour éviter de chercher dans toutes les transactions, l'algorithme enregistre les transactions où la règle est apparue et recherche uniquement dans celles-ci. Ensuite, pour toutes les autres règles, leurs valeurs d'intérêt sont calculées pour les valider ou non.
Avec cette architecture, il est possible de trouver des doublons, c'est-à-dire la même règle plusieurs fois. TRuleGrowth évite de les trouver, grâce à deux mécanismes expliqués dans [9]. Premièrement, après une expansion de la condition, il n'est plus possible d'étendre la prédiction. Ainsi, ExpandLeft ne peut être suivi par ExpandRight, mais ExpandRight peut être suivi par ExpandLeft. Deuxièmement, une expansion n'est faite que sur des éléments plus grands selon l'ordre lexicographique. Par exemple, pour $\{b, c\} \Rightarrow \{x, y\}$, $\{b, c\}$ peut être étendu en ajoutant d, ou e, mais pas a, car il est inférieur à c selon cet ordre.
Une idée pour appliquer TRuleGrowth à notre problème serait d'adapter les données en entrée, afin qu'elles puissent être acceptées par l'algorithme. Voici une proposition de modification et les inconvénients qui en découlent.

4.2 Adaptation de la série temporelle

Pour résoudre le problème de structure d'entrée de ces algorithmes, on peut simplement convertir la séries temporelles en transactions, comme dans la figure 2. Pour cela, la série temporelle est divisée (1 dans la figure) en séries plus petites d'une durée notée Δ_{tr} (2 et 3 dans la figure). Ensuite la notion de temps des petites séries est supprimée, pour ne garder que l'ordre d'apparition des éléments (4 dans la figure). Sans cette notion de temps, ce sont des séquences d'éléments, des transactions. Mais le principal problème de cette implémentation est le calcul du support d'une règle. Prenons un exemple avec trois transactions :

$$\langle\{x\},\{x\},\{y\},\{x\},\{x\}\rangle$$
$$\langle\{x\},\{y\},\{x\},\{x\},\{x\}\rangle$$
$$\langle\{x\},\{x\},\{y\},\{x\},\{x\}\rangle$$

Ici, $x \Rightarrow y$ est jugée valide, car son support est de 3, le même que x et y. Si une règle n'a été vue qu'une fois dans une transaction, elle est jugée valide tout au long de cette transaction, même si elle aurait pu être invalidée, comme dans l'exemple : x peut être vu sans y après dans toutes les transactions. Le découpage d'une série en transactions peut conduire à des règles qui sont validées par erreur. Il y a d'autres problèmes, inhérents à Δ_{tr}. Avoir un petit Δ_{tr} augmente le risque qu'une règle soit "coupée en deux", c'est-à-dire dont l'occurrence est séparée entre deux transactions, ce qui réduit l'intérêt. Avoir un gros Δ_{tr}, sur une série temporelle, peut réduire le support absolu des règles

recherchées par le système. La conversion d'une série temporelle en transactions peut être appliquée dans le cas d'utilisation proposé. Cependant, les limitations ci-dessus nous ont conduit à créer un nouvel algorithme, inspiré de TRuleGrowth, qui est pleinement adapté aux séries temporelles.

5 TSRuleGrowth

5.1 Entrées, Sorties

Ce nouvel algorithme recherche de règles de prédiction à partir d'une série temporelle d'éléments discrets. Cet algorithme est incrémental, et permet de limiter la recherche de règles à certains éléments dans la condition et de la prédiction. TSRuleGrowth prend en entrée :

- $TS = \langle (t_1, I_1), ..., (t_n, I_n) \rangle, I_1, ..., I_n \subseteq E$: Une série temporelle d'éléments discrets
- min_{sup} : La valeur minimale du support
- min_{int} : La valeur minimale de l'intérêt
- $window$: Une durée dans laquelle les règles doivent se produire

TSRuleGrowth produit des règles de prédiction semi-séquentielles utilisant des multiensembles, détaillées dans la section 2.2. Dans le cas d'utilisation proposé, la prédiction des règles ne contient que des éléments d'actionneurs.

5.2 Métriques

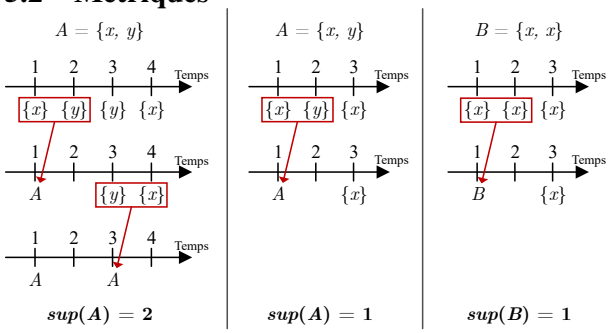


FIGURE 3 – Exemples de calcul de support

Support. Pour une série temporelle TS notée $\langle (t_1, I_1), ..., (t_{n_s}, I_{n_s}) \rangle$ où I_i est un itemset et t_i un timestamp associé, le support de x, noté $sup(x)$, est défini comme le nombre d'itemsets contenant x.

$$sup(x) = \left| (t_z, I_z) \in TS | x \in I_z \right| \tag{2}$$

Le support absolu d'un ensemble d'éléments A est le nombre d'occurrences distinctes de tous les éléments de A dans la fenêtre temporelle. Si une occurrence d'un élément de A a contribué à une occurrence du multiensemble A, elle ne peut plus contribuer aux autres occurrences de A. Les exemples de la figure 3 peuvent aider à comprendre ce concept plus facilement. L'algorithme de comptage de support, Count, fait glisser une fenêtre sur la série temporelle. Si tous les éléments de A sont vus, leurs occurrences sont mises sur liste noire, autrement dit blacklistés, pour les empêcher d'être impliqués dans une autre occurrence de A. Ceci permet de respecter la définition du support. Si plusieurs occurrences du même élément A sont vues dans la même fenêtre, seules les plus anciennes seront blacklistées. Ceci laisse aux plus récentes, via une itération ultérieure, la possibilité de contribuer à une possible future occurrence de A. Le support absolu d'une règle $R : E_c \Rightarrow E_p$ est le nombre d'occurrences distinctes où tous les éléments de E_c sont observés, suivi par tous les éléments de E_p. Les éléments de E_c et E_p ont aussi des listes noires, regroupées en deux ensembles, pour la condition et la prédiction.
Le support relatif d'un élément x, d'un multiensemble A ou d'une règle R, noté $relSup$, est son support absolu divisé par le nombre total d'itemsets de la série temporelle.

$$relSup(R) = \frac{sup(R)}{|(t_z, I_z) \in TS|} \tag{3}$$

Cette notion de support peut donc s'appliquer à des règles semi-ordonnées, contrairement à [5, 11], et évite le cas exprimé à la section 3.1.

Algorithme 1 : Count

Données : A : multiensemble, $TS = \langle (t_1, I_1), ..., (t_n, I_n) \rangle, I_1, ..., I_n \subseteq E$: série temporelle, $window$: durée
// Initialisation
Assigner liste noire $b(a)$ à chaque élément unique $a \in A$;
$sup(A) \leftarrow 0$; // Support de A
// Fenêtre glissante sur la série temporelle
tant que *la fenêtre n'a pas atteint la fin de* TS **faire**
  $dist \leftarrow$ vrai;
  Scanner la fenêtre, enregistrer les timestamps de $a \in A$ dans $T(a)$;
  pour chaque *élément* $a \in A$ **faire**
    $T(a) \leftarrow T(a) \setminus b(a)$;
    si $|T(a)| <$ *multiplicité de* a *dans* A **alors**
      $dist \leftarrow$ faux ; // Pas d'occ. distincte
  si *dist est vrai* **alors**
    $sup(A)$ += 1;
    pour chaque *élément* $a \in A$ **faire**
      // Ajouter les plus anciens timestamps $T(a)$ à la liste noire de a
      $m \leftarrow$ multiplicité de a dans A;
      $b(a) \leftarrow b(a) \cup m$ plus anciens timestamps de $T(a)$;
  Itérer la fenêtre d'un itemset;
Renvoyer $sup(A)$;

Intérêt. Dans TSRuleGrowth, on peut calculer l'intérêt d'une règle par sa confiance, conviction ou lift comme dit dans la section 3.2. Dans le cas d'utilisation proposé, nous avons choisi netconf [7, 1]. Pour une règle $R : E_c \Rightarrow E_p$:

$$netconf(R) = \frac{relSup(R) - relSup(E_c) \times relSup(E_p)}{relSup(E_c) \times (1 - relSup(E_c))} \tag{4}$$

Contrairement à la confiance, netconf teste l'indépendance entre les occurrences de E_c et celles de E_p [1]. Aussi, il est délimité entre -1 et 1, contrairement à conviction et lift, 1 montrant que E_p a une forte probabilité d'apparaître après E_c, -1 que E_p a une forte probabilité de ne pas apparaître après E_c, et 0 que cette probabilité est inconnue.

5.3 Enregistrement d'occurrences de règles

Prenons l'exemple d'une règle $R : \{a, b, c\} \Rightarrow \{x, x, y\}$. Une occurrence de R est décomposée comme l'occurrence de sa condition et de sa prédiction. En effet, un élément peut se trouver à la fois dans la condition et dans la prédiction, et il est nécessaire de distinguer les occurrences

de cet élément dans la condition de celles dans la prédiction. L'occurrence d'un multiensemble est enregistrée dans un tableau associatif où les clés sont les éléments distincts du multiensemble, et leurs valeurs associées sont l'ensemble des timestamps où ces éléments sont observés. Dans la figure 4, l'occurrence de la condition de R est$\{a:\{2\}, b:\{2\}, c:\{1\}\}$ et l'occurrence de la prévision est $\{x:\{5,6\}, y:\{4\}\}$. Ici, deux timestamps sont enregistrés pour x, car il est présent deux fois dans la prédiction de R. Les occurrences d'un multiensemble sont stockés dans une liste de ces tableaux associatifs. Les occurrences de une règle sont enregistrées dans deux listes, pour la condition et la prédiction.

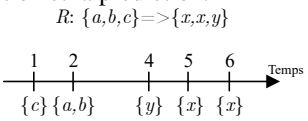


FIGURE 4 – Exemple de règle et de série temporelle

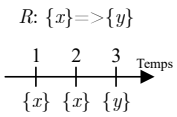


FIGURE 5 – Exemple de règle et de série temporelle

5.4 Principes

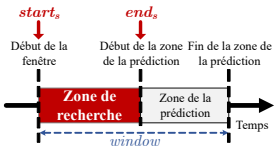
Principes partagés avec TRuleGrowth. TSRuleGrowth reprend les principes de TRuleGrowth et les applique aux séries temporelles. Il utilise une fenêtre glissante, mais, contrairement à TRuleGrowth où la fenêtre est un nombre d'itemsets consécutifs, TSRuleGrowth a une fenêtre temporelle. Elle permet de restreindre la recherche, et d'avoir une estimation de la durée d'une règle. En outre, cet algorithme trouve des règles de base, où un élément en prédit un autre. Ensuite, récursivement, il les étendra, en ajoutant un élément dans la condition ou la prédiction, via ExpandCondition et ExpandPrediction. Ce mécanisme permet de limiter la longueur des règles à chercher, c'est-à-dire le nombre maximal d'éléments dans la condition et la prédiction.
Ensuite, TSRuleGrowth applique les deux principes de TRuleGrowth mentionnés dans la section 4.1, pour éviter de trouver des règles doublons. Premièrement, ExpandPrediction ne peut pas être appelé par ExpandCondition. De plus, ExpandCondition et ExpandPrediction ne peuvent ajouter un élément que s'il est plus grand que les éléments de la condition ou de la prédiction, selon l'ordre lexicographique. Puisque TSRuleGrowth utilise une série temporelle comme entrée au lieu d'une liste de transactions, certaines notions doivent être redéfinies : le support, l'intérêt, et le stockage des occurrences d'une règle trouvée.

Nouveaux principes. Prenons l'exemple illustré dans la figure 5. Pour cette règle R, même si $sup(R) = 1$, deux occurrences sont possibles : $\{x:\{1\}, y:\{3\}\}$ et $\{x:\{2\}, y:\{3\}\}$. Ce problème est inhérent aux séries temporelles : nous ne pouvons pas savoir *a priori* quelle occurrence sera utile pour étendre cette règle. Pour ce faire, TSRuleGrowth essaie d'étendre toutes les occurrences vues de cette règle. En outre, TSRuleGrowth n'utilise pas la même structure de règles que TRuleGrowth. Au lieu d'être des ensembles, la condition et la prédiction d'une règle sont des multiensembles, où les éléments peuvent apparaître plusieurs fois. Par conséquent, un principe issu de TRuleGrowth doit être modifié : ExpandCondition et ExpandPrediction peuvent ajouter un élément s'il est plus grand que les éléments de la condition ou prédiction, mais aussi s'il est égal au plus grand élément de ceux-ci, selon l'ordre lexicographique.
Mais un nouveau problème de doublons se pose. Prenons l'exemple de la figure 5. Deux occurrences de la règle $\{x\}\Rightarrow\{y\}$ sont observés : $\{x:\{1\}, y:\{3\}\}$ et $\{x:\{2\}, y:\{3\}\}$. Si nous étendons cette règle vers la règle $\{x, x\} \Rightarrow \{y\}$, la même occurrence sera trouvée deux fois. $\{x:\{1\}, y:\{3\}\}$ sera étendue à $\{x:\{1, \mathbf{2}\}, y:\{3\}\}$, en ajoutant le timestamp 2, et $\{x:\{2\}, y:\{3\}\}$ sera étendue à $\{x:\{\mathbf{1}, 2\}, y:\{3\}\}$, en ajoutant le timestamp 1. Pour éviter cette situation, et donc éviter la duplication, TSRuleGrowth fait la chose suivante : si la règle s'étend au plus grand élément de la condition ou de la prédiction, elle ne doit enregistrer que les timestamps de cet élément qui apparaissent **strictement plus tard** que le timestamp de ce même élément dans la règle de base. Ainsi, dans l'exemple précédent, la première occurrence est enregistrée, et non la seconde.

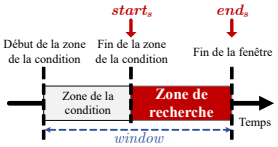
5.5 Algorithme

Boucle principale. Comme TRuleGrowth, la boucle principale de TSRuleGrowth tente de trouver des règles de base, c'est-à-dire des règles dont les conditions et les prédictions sont composées d'un seul élément. Pour ce faire, il calcule le support de toutes les règles basiques qui peuvent être créées dans la série temporelle. Si l'une de ces règles a un support supérieur à min_{sup}, elle essaie d'abord de l'étendre, en ajoutant un élément dans la condition (ExpandCondition), et dans la prédiction (ExpandPrediction). Enfin, elle calcule l'intérêt de cette règle pour la valider. Comme mentionné précédemment, l'algorithme recherche toutes les occurrences distinctes de la règle pour son support, mais aussi toutes les occurrences vues pour étendre la règle. Pour ce faire, TSRuleGrowth utilise un système de liste noire pour distinguer les occurrences.

Extension des règles. ExpandCondition essaie d'étendre une règle en ajoutant un élément à sa condition. Il passe en revue toutes les occurrences possibles de la règle, de la



(a) Zone de recherche de ExpandCondition



(b) Zone de recherche de ExpandPrediction

FIGURE 6 – Zone de recherche pour étendre une règle

plus ancienne à la plus récente. Pour respecter la contrainte de temps imposée par $window$, la condition d'une règle ne peut s'étendre qu'entre deux timestamps, noté $start_s$ et end_s, montrés sur la figure 6a : entre le début de la fenêtre, et le début de la prédiction. Comme pour ExpandCondition, ExpandPrediction cherche de nouveaux éléments pour la partie prédiction de la règle, à partir de la fin de la condition, et en respectant la taille de $window$ (figure 6b). Après avoir trouvé de nouvelles règles, ExpandCondition et ExpandPrediction essaient de les étendre à nouveau, et vérifient leur intérêt. Ici, les pseudocodes simplifiés de TSRuleGrowth et ExpandPrediction sont décrits.

Algorithme 2 : TSRuleGrowth

Données : TS : série temporelle, min_{sup} : support minimal, min_{int} : intérêt minimal, $window$: durée
Scanner TS une fois. Pour chaque élément e, stocker les timestamps des itemsets contenant e dans $T(e)$;
// Création de règles basiques
pour chaque *paire d'éléments i, j* **faire**
  $sup(i \Rightarrow j) \leftarrow 0$; // Support de la règle
  $O_c(i \Rightarrow j), O_p(i \Rightarrow j) \leftarrow []$; // Occurrences
  $b(i), b(j) \leftarrow \emptyset$; // Listes noires
  pour chaque $t_i \in T(i)$ **faire**
    pour chaque $t_j \in T(j)$ **faire**
      si $0 < t_j - t_i \leq window$ **alors**
        // Nouvelle occurrence
        Ajouter t_i à $O_c(i \Rightarrow j)$;
        Ajouter t_j à $O_p(i \Rightarrow j)$;
        si $t_i \notin b(i)$ *et* $t_j \notin b(j)$ **alors**
          // Nouvelle occurrence distincte
          $sup(i \Rightarrow j)$ += 1;
          $b(i) \leftarrow b(i) \cup \{t_i\}$;
          $b(j) \leftarrow b(j) \cup \{t_j\}$;
  // Expansion des règles basiques
  si $sup(i \Rightarrow j) \geq min_{sup}$ **alors**
    Lancer ExpandCondition et ExpandPrediction;
    si *netconf*$(\frac{|T(i)|}{|TS|}, \frac{|T(j)|}{|TS|}, \frac{sup(i \Rightarrow j)}{|TS|})) \geq min_{sup}$ **alors** Afficher la règle;

Algorithme 3 : ExpandPrediction

Données : TS : série temporelle, $E_c \Rightarrow E_p$: règle, $sup(E_c)$, occurrences de $E_c \Rightarrow E_p$, min_{sup} : support minimal, min_{int} : intérêt minimal, $window$: durée
// Expansion de la règle basique $E_c \Rightarrow E_p$
pour *chaque occurrence de la règle* $E_c \Rightarrow E_p$ **faire**
  pour chaque *élément k vu dans la zone de recherche* **faire**
    si *k n'a jamais été vu avant* **alors**
      Créer une règle $E_c \Rightarrow E_{pk}$, sa liste d'occurrences et ses listes noires;
      $sup(E_c \Rightarrow E_{pk}) \leftarrow 0$;
    pour chaque *timestamp de k* t_k *dans la fenêtre (ordre croissant)* **faire**
      si $k > \max(e), e \in E_p$ *où* $t_k >$ *occurrences de k dans la partie prédiction de la règle* **alors**
        Créer nouvelle occurrence de $E_c \Rightarrow E_{pk}$;
        si *timestamps pas dans listes noires* **alors**
          $sup(E_c \Rightarrow E_{pk})$ += 1;
          Ajouter timestamps aux listes noires;
// Expansion des nouvelles règles trouvées
pour chaque *k où* $sup(E_c \Rightarrow E_{pk}) \geq min_{sup}$ **faire**
  $sup(E_{pk}) \leftarrow Count(E_{pk}, TS, window)$;
  Lancer ExpandCondition et ExpandPrediction;
  si *netconf*$(\frac{sup(E_c)}{|TS|}, \frac{sup(E_{pk})}{|TS|}, \frac{sup(E_c \Rightarrow E_{pk})}{|TS|}) \geq min_{int}$ **alors** Afficher la règle;

6 Expérimentations et résultats

Nous avons testé cet algorithme sur la base de données Orange4Home [4], qui enregistre les activités quotidiennes d'un occupant. Elle contient 180 heures de données, sur une période de 4 semaines consécutives, à partir de 236 objets connectés intégrés dans un appartement. Pour les besoins de l'expérience, certains objets ont été spécifiés manuellement comme actionneurs : volets, portes et luminaires par exemple. De plus, un processus de discrétisation de l'amplitude a été effectué sur des objets qui rapportaient des données continues, comme un capteur de température. Pour rappel, seuls les actionneurs peuvent fournir des éléments pour la prédiction des règles. TSRuleGrowth a été implémenté en Python [1], avec $min_{sup} = 20$, $min_{int} = 0.9$, et un $window$ de 1, 2, 5, 10, 15, 20, 25 et 30 secondes. TSRuleGrowth trouve des règles simples lorsque $window$ est petit. Les règles suivantes ont été trouvées par TSRuleGrowth dans une fenêtre de deux secondes :

— {'bedroom switch top right : ON'} ⇒ {'bedroom light 1 : 0', 'bedroom light 2 : 0'} : le bouton en haut à droite de la chambre éteint les lumières de cette pièce.

— {'livingroom switch 2 top right : ON'} ⇒ {'livingroom shutter 1 : 100', 'livingroom shutter 2 : 100', 'livingroom shutter 3 : 100', 'livingroom shutter 4 : 100', 'livingroom shutter 5 : 100'} : le bouton en haut à droite du deuxième interrupteur du salon commande tous les volets.

Ces règles décrivent des prédictions à court-terme, telles que les actions des interrupteurs dans l'environnement. Ainsi, avec une petite fenêtre temporelle, l'algorithme peut déjà décrire certains des mécanismes de l'environnement connecté. Ensuite, lorsque la fenêtre est plus grande, des règles plus complexes sont découvertes en plus des règles simples, prenant en compte des objets plus diversifiés, pour caractériser des situations plus complexes. Ces règles, puisque la fenêtre d'observation est plus grande, peuvent révéler les habitudes de l'utilisateur. La règle {'bathroom door : OPEN', 'kitchen presence : OFF', 'walkway light : 0'} ⇒ {'bathroom light1 : 100', 'bathroom light2 : 100'}, vue dans une fenêtre de 30 secondes, décrit la situation où l'occupant quitte la cuisine et entre dans la salle de bains. Il est à noter que le nombre de règles trouvées par l'algorithme augmente de façon exponentielle à mesure que la fenêtre grandit, comme le montre la figure 7. En effet, la plupart des règles trouvées sur une fenêtre temporelle seront trouvées sur une fenêtre plus grande, en plus des nouvelles règles. De plus, lorsque la fenêtre grandit, davantage d'objets peuvent être utilisés pour décrire une situation, et davantage de règles peuvent être validées en conséquence. Regardons maintenant les résultats rapportés par TRuleGrowth, sur la figure 8. TRuleGrowth a été exécuté avec les mêmes paramètres que TSRuleGrowth. Deux variations ont été faites : la longueur Δ_{tr} des transactions, et la taille de $window$. De plus, la mesure d'intérêt utilisée est netconf. Selon la longueur Δ_{tr} assignée aux transactions lors du découpage de la série temporelle, le nombre de règles trouvées peut varier considérablement, comme expliqué dans la section 4.2. TSRuleGrowth se libère de cette limitation. Plus la fenêtre est grande, plus l'espace de

1. Python 3.7.3, CPU : Intel(R) Xeon(R) Gold 5118 @ 2.30GHz, RAM : 128GiB, Ubuntu 18.04.2 LTS, Multiprocessing ajouté au code

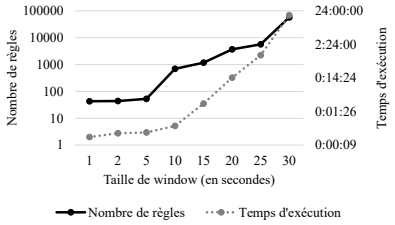


FIGURE 7 – Nombre de règles trouvées par TSRuleGrowth et temps d'exécution

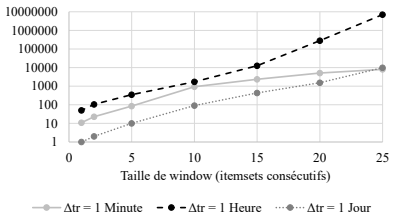


FIGURE 8 – Nombre de règles trouvées par TRuleGrowth

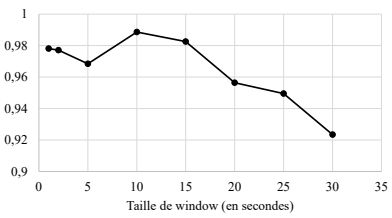


FIGURE 9 – Moyenne de l'intérêt des règles trouvées par TSRuleGrowth

recherche est grand. Par conséquent, le temps d'exécution de TSRuleGrowth augmente exponentiellement à mesure que la fenêtre augmente, comme le montre la figure 7. On trouve de plus en plus de règles pour décrire les situations. Ces situations, impliquant l'utilisateur, ne peuvent pas être aussi certaines que les règles simples vues avant, telles que celles d'un interrupteur. En conséquence, l'intérêt moyen des règles tend à diminuer au fur et à mesure que la fenêtre augmente, comme le montre la figure 9.

7 Conclusion

Cet article décrit deux contributions : une nouvelle notion de support sur une série temporelle, et un algorithme de recherche de règles de prédiction semi-ordonnées sur une série temporelle d'éléments discrets. La notion de support est libéré des limites exprimées dans l'état de l'art, et l'algorithme se distingue également par ses caractéristiques. Tout d'abord, une architecture incrémentale, inspirée de TRuleGrowth, permettant de limiter la recherche à certains éléments si nécessaire, comme dans le cas d'utilisation proposé. Une fenêtre glissante permet de limiter la durée des règles recherchées. Un nouveau mécanisme évite de trouver plusieurs fois la même règle. Les résultats présentés permettent de tester et valider l'algorithme sur des données réelles provenant d'un environnement connecté. Ils montrent des règles de prédiction simples, telles que l'action d'un interrupteur dans une pièce donnée, et d'autres plus complexes, impliquant des objets connectés différents. Ces dernières règles ouvrent sur des propositions d'automatisation pertinentes aux utilisateurs d'un système d'intelligence ambiante.

Références

[1] K.-I. Ahn and J.-Y. Kim. Efficient Mining of Frequent Itemsets and a Measure of Interest for Association Rule Mining. *J. Inf. Knowl. Manag.*, 03(03) :245–257, Sept. 2004.

[2] J. C. Augusto and P. McCullagh. Ambient intelligence : Concepts and applications. *Comput. Sci. Inf. Syst.*, 4(1) :1–27, 2007.

[3] P. J. Azevedo and A. M. Jorge. Comparing Rule Measures for Predictive Association Rules. In *Machine Learning : ECML 2007*, Lecture Notes in Computer Science, pages 510–517. Springer Berlin Heidelberg, 2007.

[4] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley. A Dataset of Routine Daily Activities in an Instrumented Home. In *11th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI)*, Nov. 2017.

[5] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule Discovery from Time Series. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 16–22. AAAI Press, 1998.

[6] J. Deogun and L. Jiang. Prediction Mining – An Approach to Mining Association Rules for Prediction. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Lecture Notes in Computer Science, pages 98–108. Springer Berlin Heidelberg, 2005.

[7] J. K. Febrer-Hernández, R. Hernández-León, C. Feregrino-Uribe, and J. Hernández-Palancar. SPaC-NF : A classifier based on sequential patterns with high netconf. *Intell. Data Anal.*, 20(5) :1101–1113, Sept. 2016.

[8] P. Fournier-Viger, T. Gueniche, S. Zida, and V. S. Tseng. ERMiner : Sequential Rule Mining Using Equivalence Classes. In *Advances in Intelligent Data Analysis XIII*, pages 108–119. Springer International Publishing, 2014.

[9] P. Fournier-Viger, C.-W. Wu, V. S. Tseng, L. Cao, and R. Nkambou. Mining Partially-Ordered Sequential Rules Common to Multiple Sequences. *IEEE Transactions on Knowledge and Data Engineering*, 27(8) :2203–2216, Aug. 2015.

[10] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Min Knowl Discov*, 1(3) :259–289, Jan. 1997.

[11] T. Schlüter and S. Conrad. About the analysis of time series with temporal association rule mining. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 325–332, Apr. 2011.