Enrichissement de règles de Horn par des predicats numériques

Armita Khajeh Nassiri*, Nathalie Pernelle**, Fatiha Saïs*

* LISN, CNRS (UMR 9015), Université Paris Saclay , France
** LIPN, CNRS (UMR 7030), Université Sorbonne Paris Nord, France
firstname.lastname@lri.fr

Résumé. Dans cet article nous présentons REGNUM, un système qui enrichit le corps de règles déjà découvertes dans un graphe de connaissances avec des atomes impliquant des prédicats numériques dont les valeurs sont contraintes par des intervalles spécifiés. Les intervalles sont obtenus à l'aide de techniques de discrétisation supervisées, avec l'objectif d'augmenter la confiance des règles fournies par la méthode de découverte de règles. Nos résultats expérimentaux démontrent que les règles enrichies avec des prédicats numériques sont de meilleure qualité globale et sont mieux adaptées pour la tâche d'enrichissement de graphes de connaissances comparativement aux règles initiales.

1 Introduction

Les graphes de connaissances (GC) représentent des faits sur de nombreuses entités du monde exprimés dans un format interprétable par les machines. Ces graphes intègrent différentes formes de connaissances, et de nombreux travaux ont été consacrés à l'acquisition de ces connaissances. Un type d'approches sont celles permettant la fouille de règles logiques dans les GCs. Ces règles peuvent servir à compléter le GC, à détecter des données erronées et à aligner des ontologies.

AMIE(Lajus et al., 2020) est un système de fouille de règles pour les GCs très connu dans l'état de l'art. Il est efficace et exhaustif, c'est-à-dire qu'il extrait toutes les règles connectées et fermées en fonction de seuils définis sur des mesures de qualité (par exemple, la confiance et la couverture des têtes des règles) et d'un nombre maximal d'atomes spécifié. AMIE peut découvrir des règles qui impliquent des constantes (par exemple, `age(x,53)`). Cependant, ces règles peuvent être trop spécifiques et peu intéressantes lorsqu'il s'agit de ces prédicats. RuDiK (Ortona et al., 2018) propose une approche non exhaustive pour découvrir des règles logiques plus expressives. RuDiK peut prédire l'absence d'un fait et permet d'effectuer des comparaisons au-delà des égalités en utilisant les relations appartenant à l'ensemble $rel \in \{<, \leq, \neq, \geq, >\}$. Les valeurs impliquées ne sont pas des seuils mais proviennent du graphe de connaissances lui-même.

Il existe également des systèmes de fouille de règles tels que AnyBURL (Meilicke et al., 2019) se focalise uniquement sur les règles fondées sur des chemins dans le graphe. AnyBURL est une approche ascendante qui commence par l'échantillonnage de chemins spécifiques dans

le graphe et utilise des techniques de généralisation pour l'étendre de manière à ce que la règle obtenue soit la meilleure possible (i.e., une confiance élevée). AnyBURL, comme AMIE, peut seulement considérer les prédicats numériques comme des constantes. Une autre famille de systèmes de fouille de règles est constituée de méthodes d'inférence fondées sur des règles différentiables, telles que NeuralLP (Yang et al., 2017). Ces approches font correspondre chaque entité à un vecteur et chaque relation à une matrice d'adjacence. Une extension de NeuralLP a été proposé dans le modèle NeuralLP-num (Wang et al., 2020) qui peut apprendre des règles impliquant des prédicats numériques. Ces règles peuvent, comme RuDiK, impliquer des règles négatives ou effectuer des comparaisons par paire entre les valeurs numériques de différents atomes dans les règles. Les règles produites par NeuralLP-num peuvent également contenir des opérateurs de classification. Il s'agit de fonctions sigmoïdes sur les valeurs numériques des atomes avec prédicats numériques dans la règle.

À notre connaissance, RuDiK et NeuralLP-num sont les seuls travaux capables d'extraire des règles intéressantes avec des prédicats numériques. Cependant, ces deux approches ne peuvent pas considérer des intervalles numériques (ou des seuils).La difficulté de trouver de telles règles réside dans la taille exponentielle de l'espace de recherche, qui varie en fonction du biais de langage considéré. Des approches récentes se sont appuyées sur l'échantillonnage de chemins, un calcul de la confiance par approximation ou l'extraction de règles qui ne couvrent que des exemples positifs, entre autres, pour traiter ce problème. Nous pensons que les contraintes sur les valeurs de prédicats numériques peuvent être très pertinentes dans des domaines tels que la finance, la santé publique ou la science de la vie. Dans cet article, nous faisons un pas en avant vers l'introduction de prédicats numériques dans les règles logiques découvertes dans les graphes de connaissances.

Les principales contributions de ce travail sont les suivantes : (i) une approche qui enrichit les règles fermées générées par un système de fouille de règles en utilisant des contraintes numériques exprimées par des intervalles de valeurs. Comme d'autres systèmes de fouille de règles, ces règles ont l'avantage d'être explicables, interprétables et transférables à des entités inconnues ; (ii) un algorithme exploitant des techniques de discrétisation supervisée pour obtenir des intervalles qui distinguent les prédictions correctes et fausses produites par une règle. Il prend en compte les contraintes qui peuvent exprimer à la fois l'appartenance et la non-appartenance d'une valeur à un intervalle pour offrir plus de possibilités de générer une règle de bonne qualité ; et (iii) évaluation expérimentale sur trois graphes de connaissances de référence où nous démontrons les bénéfices de notre approche pour la qualité globale des règles, ainsi que les améliorations sur la tâche d'enrichissement de graphes de connaissances.

2 Preliminaries

Graphe de connaissances RDF. Un graphe de connaissances RDF $\mathcal{G}$ est un ensemble de faits (triplets) représentés par {(subject, property, object) | subject $\in \mathcal{I}$, property $\in \mathcal{P}$, object $\in \mathcal{I} \cup \mathcal{L}$}, où $\mathcal{I}$ est un ensemble d'entités, $\mathcal{P}$ un ensemble de propriétés, et $\mathcal{L}$ est un ensemble de littéraux.

Règle de Horn. Une règle $r : \mathcal{B} \Rightarrow H$ est une formule en logique du premier ordre qui a un corps $\mathcal{B}$ composé de conjonctions d'atomes $B_1, ..., B_n$ et une tête H composée d'un un seul atome.

Une règle est dite *fermée* si chaque variable apparaît au moins deux fois dans la règle.

Degré de fonctionnalité. Les propriétés prennant pour chaque sujet au plus un objet sont dites *fonctionnelles*. Le degré de fonctionnalité $fd(p) := \frac{\#x:\exists y:p(x,y)}{\#(x,y):p(x,y)}$ d'une propriété est une valeur comprise entre 0 et 1 et est défini par le rapport entre le nombre de sujets avec lesquels cette propriété est en relation, dans $\mathcal{G}$, et le nombre de triplets de cette relation dans $\mathcal{G}$. Le degré de fonctionnalité inverse $ifd(p)$ est le degré de fonctionnalité pour l'inverse de p.

Les graphes de connaissances ne contiennent que des exemples positifs et mettent en oeuvre l'hypothèse du monde ouvert. Dans cet article, pour considérer les contre-exemples, nous nous inspirons de l'approche l'AMIE (Lajus et al., 2020) et mettons en œuvre l'hypothèse de complétude partielle (PCA) stipulant que si un ensemble de faits tels que $p(x, y)$ est déclaré pour l'entité x, alors aucun autre fait concernant l'entité x avec la relation p ne peut être déclaré et peut être considéré comme un contre-exemple. Ceci est particulièrement vrai si p a un degré de fonctionnalité élevé et peut être étendu au cas où le $fd(p) > ifd(p)$.

Pour une règle $r : \mathcal{B} \Rightarrow H$ nous définissons les mesures de qualité suivantes telles qu'elles sont introduites dans (Lajus et al., 2020).

Support. Le support mesure le nombre de prédictions correctes faites par la règle.

Couverture de la tête de règle. Une version proportionnelle du support est la couverture de la tête de la règle (nommée *head coverage* pour l'anglais), qui représente la proportion de paires instanciées de H qui sont prédites correctement par la règle.

$$hc(r) = \frac{supp(r):=\#(x,y)\in\mathcal{G}:\mathcal{B}\wedge H(x,y)}{\#(x',y'):H(x',y')}$$

Confiance PCA. La *confidence PCA* mesure la précision de la règle sous l'hypothèse de complétude partielle (PCA). Si $fd(H) > ifd(H)$, elle est calculée comme suit.

$$pca\_conf(r) = \frac{supp(r)}{\#(x,y)\in\mathcal{G}:\exists y':\mathcal{B}\wedge H(x,y')}$$

Si $ifd(H) > fd(H)$ le dénominateur devient $\#(x, y) \in \mathcal{G} : \exists x' : \mathcal{B}\wedge H(x', y)$ dans l'équation ci-dessus.

3 Enrichissement des règles avec des prédicats numériques

Dans cette section, nous présentons REGNUM, un système qui enrichit automatiquement les règles (fermées), découvertes dans un graphe de connaissances, avec des prédicats numériques en limitant les valeurs numériques introduites à des intervalles spécifiés. Nous considérons les règles qui peuvent être fournies par tout système de fouille de règles existant sur les KG (par exemple, AMIE, AnyBURL). REGNUM vise à améliorer la confiance PCA des règles considérées tout en garantissant que les règles ne deviennent pas trop spécifiques.

3.1 Problème

Nous considérons un graphe de connaissance $\mathcal{G}$ et un ensemble de règles fermées $\mathcal{R}$ déjà découvertes de $\mathcal{G}$, appelées *règles-parentes* comme défini dans 2, et les seuils $marginPCA$ et $marginHC$ qui contrôlent la mesure de qualité des règles enrichies par rapport à la règle

parente. Nous voulons que la confiance PCA des règles enrichies augmente au moins de $marginPCA$ sans que la couverture de la tête ne diminue de plus de $marginHC$. REGNUM a pour objectif d'étendre les règles de $\mathcal{R}$ avec des prédicats numériques et produit un ensemble de règles enrichies $\mathcal{E}$ de confiance PCA augmentée.

Les règles enrichies $\mathcal{B} \Rightarrow H$ où les prédicats du coprs de la règles sont dans $\mathcal{P} \cup \{belongs\} \cup \{notbelongs\}$. Pour un atome avec un prédicat numérique $p_{num}(x, y)$, la conjonction avec l'atome $belongs(y, [inf, sup])$ exprime que y est instancié par des valeurs numériques qui appartiennent à l'intervalle $[inf, sup]$. Et $notbelongs(y, [inf, sup])$ exprime que y est instancié par des valeurs qui n'appartiennent pas à l'intervalle $[inf, sup]$.

3.2 Processus d'enrichissement

Nous expliquons ci-dessous le processus d'enrichissement pour chaque règle parente $r \in \mathcal{R}$.
(1) Identifier les prédicats numériques Dans un premier temps, l'ensemble des prédicats numériques $\mathcal{P}_{num}$ de $\mathcal{G}$ est identifié. Nous utilisons les axiomes de définition de domaine et de co-domaine s'ils sont disponibles dans l'ontologie. Sinon, une simple étape de pré-traitement pour trouver de tels prédicats en considérant le leurs valeurs.

(2) Sélectionner les prédicats numériques candidats. Pour chaque règle $r \in \mathcal{R}$, cette étape vise à trouver les prédicats dans $\mathcal{P}_{num}$ qui peuvent être utilisés pour enrichir la règle parente r. Plus précisément, nous considérons que l'ensemble de toutes les variables d'une règle r est $vars = \{x_1, .., x_n\}$. Cette étape sélectionne pour une règle r et une variable $x_i \in vars$, tous les prédicats p_{num} tels que l'ajout de l'atome $p_{num}(x_i, x_{n+1})$ à r résulte en la règle spécialisée

$$r_s : p_{num}(x_i, x_{n+1}) \wedge \mathcal{B} \Rightarrow H,$$

dont le support est supérieur à $minhc * size(H)$ avec $minhc = (1 - marginHC) * hc(r)$.

Exemple. Soit $r_1 : workPlace(x_1, x_2) \Rightarrow birthPlace(x_1, x_2)$, une règle parente. Le prédicat numérique $hasPopulation$ avec les variables x_1 sera retiré de l'espace de recherche comme $hasPopulation(x_1, x_3) \wedge workPlace(x_1, x_2) \Rightarrow birthPlace(x_1, x_2)$, ne satisfait pas le seuil $minhc$.

(3) Classification des entités fondées sur les prédictions des règles. Notre objectif est de construire des intervalles qui permettent de classer au mieux les instances selon qu'elles conduisent à des prédictions correctes ou incorrectes. Cette étape permet de définir l'ensemble des entités de $\mathcal{G}$ de chaque classe. Nous nous appuierons sur des techniques de discrétisation supervisées pour obtenir les intervalles et alimenter ces techniques avec des exemples positifs et négatifs en adhérant à l'hypothèse de complétude partielle (PCA). À cette fin, nous considérons le score de fonctionnalité fd et le score de fonctionnalité inverse ifd du prédicat de la tête de la règle H.

Considérer le prédicat de la tête H comme plus fonctionnel qu'inverse-fonctionnel $(fd(H) > ifd(H))$. Une prédiction produite par une règle est incorrecte si un objet prédit contredit un fait dans le graphe de connaissances (par exemple, un lieu de naissance différent pour quelqu'un qui a déjà un lieu de naissance dans le graphe de connaissances). En d'autres termes, nous classons les entités apparaissant comme sujets dans $H(x, y)$, et x est appelé la variable cible. Dans le cas où H est plus inverse fonctionnel que fonctionnel, nous classons les entités apparaissant comme des objets de $H(x, y)$, et y est la variable cible. Si nous devons

classer l'entité qui apparaît comme une valeur constante dans H, alors il ne sera pas utile de procéder à la classification ; par conséquent, nous ne procédons pas à l'enrichissement d'une telle règle.

Dans cette étape de classification, nous définissons la classe A pour représenter l'ensemble des entités qui conduisent à une prédiction correcte pour r_s.

$\mathcal{B}(x_1, ..., x_n) \wedge p_{num}(x_z, x_{n+1}) \wedge H(x_i, x_j) \Rightarrow A(x_t)$, $\{x_z, x_i, x_j\}\} \in vars$ et x_t est la variable cible (c'est-à-dire x_i ou x_j).

Nous construisons également la classe B pour représenter l'ensemble des entités qui conduisent à une prédiction incorrecte pour r_s. Plus précisément, une entité appartient à la classe B s'il existe au moins un fait dans le graphe de connaissance qui décrit H pour l'entité et si tous ces faits contredisent la prédiction de la règle. x_t est la variable cible et une variable de H, tandis que x_j est l'autre variable de H.

$\mathcal{B}(x_1, ..., x_n) \wedge p_{num}(x_z, x_{n+1}) \wedge (\forall x_k H(x_t, x_k)) \Rightarrow (x_k <> x_j) \wedge (\exists x_k H(x_t, x_k)) \Rightarrow B(x_t)$

Un individu peut appartenir à la fois à A et à B puisque les prédicats peuvent être multivalués.

(4) Règle avec variable existentielle. En vérifiant simplement si $pca\_conf(r_s) > (1 + marginPCA) * pca\_conf(r)$, nous pouvons ajouter r_s qui a une variable existentielle à $\mathcal{E}$ (c'est-à-dire assez pour vérifier qu'un fait avec le prédicat numérique existe).

(5) Discrétisation supervisée et spécification de règles. Étant donné les classes binaires (A et B) définies pour la variable cible x_t générée à l'étape 3, et les valeurs du prédicat p_{num} comme caractéristique d'une technique de discrétisation supervisée, différents intervalles $b_1, ..., b_k$ discrétisant les valeurs de p_{num} peuvent être obtenus. Chaque intervalle b_i contient un nombre de prédictions correctes e_i (c'est-à-dire appartenant à la classe A) et un nombre de prédictions incorrectes ne_i (c'est-à-dire appartenant à la classe B). Ces valeurs sont normalisées par le nombre total d'entités dans les deux classes (E_i et NE_i).

Comme nous ne voulons pas sur-spécialiser les règles tout en augmentant la confiance, nous ajouterons la contrainte $belongs$ (classe A) ou $notbelongs$ (classe B) seulement si le hc ne diminue pas de plus de $marginHC$ et si la confiance PCA est augmentée d'au moins $marginPCA$.

Si le i-ème intervalle $[inf, sup]$ identifie la classe A, nous vérifions $E_i > (1 + marginPCA) * pca\_conf(r)$ car E_i servira de confiance PCA de la règle enrichie, et nous vérifions également que $\frac{e_i}{size(H)} > (1 - marginHC) * hc(r)$. Si les équations ci-dessus sont vérifiées, nous ajoutons $p_{num}(x_i, x_{n+1}) \wedge belongs(x_{n+1}, [inf, sup])$ à la règle parentale r.

Par contre, si le i-ème intervalle identifie la classe B, on vérifie $NE_i > (1 + marginPCA) * pca\_conf(r)$ et $\frac{supp(r_s) - e_i}{size(H)} > (1 - marginHC) * hc(r)$. Si les équations sont satisfaites, nous ajoutons l'atome $p_{num}(x_i, x_{n+1}) \wedge notbelongs(x_{n+1}, [inf, sup])$. Un intervalle sera élagué chaque fois qu'il ne satisfait pas l'une des conditions ci-dessus.

Exemple. Considérons la règle parente r_1 de l'exemple 3.2 avec $pca\_conf(r_2) = 0.7$ et $hc(r_2) = 0.4$. Nous avons envisagé d'ajouter l'atome $dateOfBirth(x_1, x_2)$ au corps de cette règle et avons défini les classes A et B comme indiqué aux étapes (2) et (3). Le tableau ci-dessous montre un ensemble d'intervalles proposés par une technique de discrétisation pour le prédicat numérique $dateOfBirth$, E_i et NE_i. Les valeurs finales de pca_conf et hc obtenues en considérant chaque intervalle sont également indiquées. Dans cet exemple, les premiers et

Intervalle	E	NE	Contrainte sur le prédicat	$hc(r_{enr})$	$pca\_conf(r_{enr})$
(-∞, 1834]	0.1	0.9	$notbelongs$	0.19	0.84
[1834,1905]	0.6	0.4	$belongs$	0.09	0.60
[1905, +∞)	0.8	0.2	$belongs$	0.10	0.80

troisièmes intervalles sont sélectionnés pour générer deux règles car ils satisfont aux conditions définies à l'étape (5).

4 Évaluation expérimentale

Nous avons mené deux séries d'expériences. Premièrement, nous évaluons la qualité des règles enrichies par rapport à leurs règles parentes respectives. Deuxièmement, nous comparons les résultats de la tâche d'enrichissement de graphes de connaissances sur ces ensembles de règles. Nous considérons trois jeux de données de référence différents qui impliquent des valeurs numériques. Les statistiques de ces ensembles de données sont données dans le Tableau 1. Le code source de notre approche est accessible au public [1].

Dataset	$\|\mathcal{I}\|$	$\|\mathcal{P}\|$	$\|\mathcal{P}_{num}\|$	$\|\mathcal{G}\|$	$\|\mathcal{G}_t\|$
DB15K-num(García-Durán and Niepert, 2018)	12,867	278	251	79,345	9,789
FB15K-237-Num(García-Durán and Niepert, 2018)	14,541	237	116	272,115	1,215
LitWD19K(Gesese et al., 2021)	18,986	182	151	260,039	14,447

TAB. 1 – *Statistiques des jeux de données de référence.* $|\mathcal{G}_t|$ *est la taille de l'ensemble de test.*

Évaluation de la qualité des règles. Dans cette première série d'expériences, nous comparons la qualité globale des règles extraites par AMIE(Lajus et al., 2020) avec les règles enrichies de REGNUM sur les trois ensembles de données de référence. Pour mesurer la qualité globale des règles, nous utilisons $F(r) = 2 * \frac{pca\_conf(r) * hc(r)}{pca\_conf(r) + hc(r)}$, qui est une moyenne harmonique entre pca_conf et hc. En effet, un pca_conf ou un hc élevé n'est pas un bon indicateur de la qualité globale d'une règle (la règle peut être trop spécifique ou ne pas donner de bonnes prédictions).

Nous avons exécuté AMIE avec $minhc = 0.01$ et $min\_pca\_conf = 0.1$ sur les jeux de données *LitWD19K* et *DBPedia15K*, et avec $minhc = 0,1$ sur *FB15K-237-num*, et permettent d'obtenir l'ensemble des règles parentes $\mathcal{R}$. REGNUM enrichit ces règles parentes avec $marginPCA = 20\%$, $marginHC = 10\%$. Nous avons utilisé différentes techniques de discrétisation supervisée telles que la discrétisation optimale (Navas-Palencia, 2020) et MDLP (Fayyad and Irani, 1993). Dans le tableau 2, nous avons reporté les résultats avec MDLP.

On obtient la moyenne de pca_conf, hc, et F des règles parentes qui pourraient être enrichies $\mathcal{R}_{enr}$ et des règles enrichies $\mathcal{E}$. Dans le tableau 4, g_{conf}, g_{hc} et g_F représentent le pourcentage d'amélioration de chacune de ces mesures de qualité en comparant $\mathcal{E}$ à $\mathcal{R}_{enr}$. Par exemple, pour le jeu de données LitWD19K, la moyenne de pca_conf est de 0,40 pour les règles de l'AMIE et de 0,72 pour nos règles enrichies. Nous observons donc un gain significatif en termes de confiance. Globalement, sur ces trois jeux de données, nous observons que les règles enrichies ont un F-score plus élevé avec une augmentation significative de la confiance sans perdre trop de couverture de tête de règle.

1. `https://github.com/armitakhn/REGNUM`

Dataset	$\|\mathcal{R}\|$	$\|\mathcal{R}_{enr}\|$	$\|\mathcal{E}\|$	g_{conf}	g_{hc}	g_F
DB15K-num	2,689	352	2081	+19.2%	-4.0%	+4.1%
FB15K-237-num	9,590	2,394	17,639	+14.6%	-3.7%	+1.5%
LitWD19K	2,481	690	10772	+43.8%	-2.8%	+2.4%

TAB. 2 – *$|\mathcal{R}|$ est le nombre de règles parentes découvertes par AMIE, $\mathcal{R}_{enr}$ nombre de règles qui pourraient être enrichies, et $|\mathcal{E}|$ est le nombre de règles enrichies produites par REGNUM.*

Enrichissement de graphes de connaissances. Dans cette deuxième série d'expériences, nous avons dirigé notre attention vers la tâche d'enrichissement de graphes de connaissances, qui vise à prédire un objet manquant o dans un fait $(s, p, o) \notin \mathcal{G}$. La plupart des travaux pour l'enrichissement de graphes de connaissances reposent sur des techniques de plongement dans graphes de connaissances. Néanmoins, les règles peuvent également être utilisées pour faire ces prédictions, et elles ont l'avantage d'être explicables et interprétables. En utilisant les règles obtenues dans 4, nous montrons que l'ajout de $\mathcal{E}$ à $\mathcal{R}_{enr}$, augmente la précision de la complétion du KG. Pour faire l'évaluation, nous rapportons les résultats de Hits@k.

Pour chaque règle, nous exécutons une requête SPARQL et considérons les prédictions faites par les règles comme un ensemble de faits candidats. Chaque fait candidat peut être donné par un ensemble de règles $C = \{R_1, ..., R_n\}$. Nous avons utilisé une fonction d'agrégation F-pondérée qui pénalise les règles qui donnent lieu à de nombreuses prédictions $\mathcal{S}_c = \Sigma_{i=1}^{n} \frac{1}{\#Prediction(R_i)} * f(R_i)$. Le tableau 3 montre les résultats de l'enrichissement du graphe de connaissances en utilisant uniquement les règles générées par l'outil AMIE qui pourraient être enrichies $\mathcal{R}_{enr}$ vs. en utilisant ces règles ainsi que les règles enrichies $\mathcal{E}$. Ce choix nous permet d'observer de près l'impact que les règles enrichies apportent à la complétion du graphe de connaissances. Néanmoins, il ne nous permettra pas de comparer avec (Wang et al., 2020) et (Gesese et al., 2021) car nous ne considérons pas toutes les règles. Nous pouvons observer que la prise en compte des règles enrichies avec leurs règles parentes respectives augmente la précision de la complétion graphe de connaissances.

Dataset	**AMIE** ($\mathcal{R}_{enr}$)		**AMIE+REGNUM** ($\{\mathcal{R}_{enr} \cup \mathcal{E}\}$)	
	Hits@1	Hits@10	Hits@1	Hits@10
DBPedia15K	7.1	11.1	8.9	14.4
FB15K-237-num	9.2	24.0	12.5	30.1
LitWD19K	11.4	20.5	12.6	24.9

TAB. 3 – *Les résultats de Hits@1 et de Hits@10 de l'enrichissement de graphes de connaissances sur les jeux de données benchmark.*

5 Conclusion and travaux futurs

Dans cet article, nous avons présenté une nouvelle approche d'extension de règles découvertes de règles afin d'obtenir des règles de Horn incluant des prédicats numériques dont les valeurs sont contraintes par des intervalles spécifiques. Pour obtenir les intervalles, nous nous appuyons sur des techniques de discrétisation. Nous avons montré que les règles enrichies sont de meilleure qualité en moyenne et améliorent la précision des systèmes de découverte de règles pour la tâche de complétion graphes de connaissances. Dans les travaux futurs, nous

envisageons d'ajouter plus d'un prédicat numérique à une règle parente, d'appliquer de nouvelles stratégies d'optimisation et de considérer d'autres jeux de données plus adaptés.

Acknowledgements : Ce travail a été soutenu par le projet PSPC AIDA : 2019-PSPC-09 financé par BPI-France.

Références

Fayyad, U. M. et K. B. Irani (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*.

García-Durán, A. et M. Niepert (2018). Kblrn : End-to-end learning of knowledge base representations with latent, relational, and numerical features. In *Proc. of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Gesese, G. A., M. Alam, et H. Sack (2021). Literallywikidata - a benchmark for knowledge graph completion using literals. In *The Semantic Web – ISWC 2021*, Cham, pp. 511–527. Springer International Publishing.

Lajus, J., L. Galárraga, et F. Suchanek (2020). Fast and exact rule mining with amie 3. In *The Semantic Web*, Cham, pp. 36–52. Springer International Publishing.

Meilicke, C., M. W. Chekol, D. Ruffinelli, et H. Stuckenschmidt (2019). Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 3137–3143.

Navas-Palencia, G. (2020). Optimal binning : mathematical programming formulation. *CoRR abs/2001.08025*.

Ortona, S., V. V. Meduri, et P. Papotti (2018). Rudik : Rule discovery in knowledge bases. *Proc. VLDB Endow. 11*(12), 1946–1949.

Wang, P.-W., D. Stepanova, C. Domokos, et J. Z. Kolter (2020). Differentiable learning of numerical rules in knowledge graphs. In *International Conference on Learning Representations*.

Yang, F., Z. Yang, et W. W. Cohen (2017). Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Summary

In this work, we present REGNUM, a system that enriches the body of the rules mined on a knowledge graph with atoms involving numerical predicates whose values are constrained by specified intervals. The intervals are obtained using supervised binning techniques with the objective of increasing the confidence of the rules provided by the rule mining technique. Our experimental results demonstrate that the rules enriched with numerical predicates have a higher overall quality and are better suited for the knowledge graph completion task.