GPoID : Extraction de Motifs Graduels pour les Bases de Données Imprécises

Michael Chirmeni Boujike*, Jerry Lonlac**
Norbert Tsopze*, Engelbert Mephu Nguifo***

*Département d'informatique - Université de Yaoundé 1
Sorbonne University, IRD, UMMISCO, F-93143, Bondy, France
michael.chirmeni@facsciences-uy1.cm, norbert.tsopze@facsciences-uy1.cm
** IMT Lille Douai, IMT, Univ. Lille, Center for Digital Systems, F-59000 Lille, France
jerry.lonlac@imt-lille-douai.fr
***LIMOS CNRS UMR 6158 - Univ. Clermont-Auvergne, F-63000 Clermont-FD
engelbert.mephu_nguifo@uca.fr

Résumé. Ces dernières années, les motifs graduels ont attiré l'attention de la communauté de la science des données et plusieurs algorithmes ont été conçus pour extraire ces motifs à partir de différents modèles de données. Sur certaines données, comme les données imprécises, l'un des biais dans les algorithmes traditionnels est le fait qu'ils définissent la gradualité comme une augmentation/diminution de valeur. Par conséquent, certains motifs extraits ne sont qu'un effet de bruit dans les données. Pour remédier à ce problème, nous proposons dans cet article, une méthode qui introduit dans le processus de fouille, un seuil graduel à partir duquel considérer une gradualité. Les expérimentations sur différentes bases de données montrent que notre proposition réduit les temps de calcul et le nombre de motifs générés en focalisant sur les motifs d'intérêt. De plus, elle extrait les motifs graduels dans certains cas où les approches traditionnelles échouent.

1 Introduction

Le raisonnement humain est le plus souvent basé sur les données imprécises ou incomplètes ; en effet, il est facile pour un humain de déterminer si une personne est de petite ou de grande taille sans pour autant connaitre sa taille exacte ; ce qui n'est pas le cas pour un ordinateur car ce dernier traite les données exactes. Transmettre les facultés du raisonnement humain à un ordinateur a été initié par (Zadeh et al., 1996) dont le but était de faire traiter les données imprécises par un ordinateur. Ainsi, des travaux basés sur la logique floue (Zadeh et al., 1996) et la logique classique (Agrawal et Srikant, 1994) ont été réalisés pour l'extraction des connaissances dans les bases de données catégorielles mais rencontrent de nombreuses difficultés dans les bases de données numériques. Récemment, un mouveau type de motif a émergé (motifs graduels) pour l'extraction de connaissance dans ces bases de données numériques et plusieurs algorithmes efficaces furent proposés pour extraire automatiquement ces motifs à

partir de différents modèles de données (données temporelles (Lonlac et al., 2018), données incomplètes (Shah et al., 2020), données incertaines (Flores et al., 2012), etc...). Sur certaines données, comme les données imprécises, l'un des biais dans ces algorithmes est le fait qu'ils définissent la gradualité comme une augmentation/diminution de valeur. Par conséquent, certaines gradualités extraites ne sont qu'un effet de bruit dans les données. En médecine par exemple, la température corporelle normale est d'environ 37°C et les professionnels considèrent comme signe de fièvre la température supérieure à 38°C. Cela signifie qu'ils ne peuvent prendre aucune décision sur la base d'une variation de température entre 37°C et 38°C.

Afin d'éviter les motifs graduels inintéressants et de réduire par conséquent le nombre de motifs graduels à analyser par l'utilisateur, nous proposons une approche d'extraction de motifs graduels qui tient compte de la distribution (échelle des valeurs) de chaque attribut de la base de données pour intégrer au processus de fouille un seuil de variation à partir duquel considérer une gradualité (augmentation ou diminution). Intégrée aux sémantiques de gradualités proposées par (Lonlac et al., 2018) ou par (Négrevergne et al., 2014), ou encore par (Di-Jorio et al., 2009), cette approche permet de réduire considérablement le nombre de motifs générés et focaliser la recherche sur les motifs d'intérêt.

Le reste de cet article est organisé comme suit : nous présentons quelques travaux sur les différentes approches traitant des motifs graduels dans la section 2. Dans la section 3, nous décrivons notre approche d'extraction des motifs graduels sous contrainte sur la variation des valeurs d'attributs. Avant de conclure, nous présentons et discutons des résultats expérimentaux dans la section 4.

2 Etat de l'art

La notion *de motif graduel*, ainsi que *d'extension* (liste ordonnée de tuples respectant le motif graduel) et de *support* associé à un motif graduel ont été largement étudiées dans (Di-Jorio et al., 2009; Laurent et al., 2009; Négrevergne et al., 2014; Lonlac et al., 2018). La plupart des méthodes d'extraction de motifs graduels proposées diffèrent généralement dans leur application en fonction du type de données (données temporelles, séquences de données temporelles, flux de données, données multirelationnelles, données graphe, etc.) à partir duquel l'extraction du motif a été effectuée. Dans ce qui suit, nous donnons une brève description de quelques méthodes efficaces d'extraction de motifs graduels en fonction du modèle de données considéré, tout en soulignant leurs avantages et leurs limites.

2.1 Sur la fouille des motifs graduels

Approche basée sur les graphes de précédence : pour faire face à l'inconvénient révélé dans l'approche (Di-Jorio et al., 2008), (Di-Jorio et al., 2009) propose une approche plus complète nommée GRITE (GRadual ITemset Extraction). Les auteurs considèrent la même définition du support proposée dans (Di-Jorio et al., 2008) et proposent une nouvelle méthode basée sur des graphes de précédence. Dans cette méthode, les données sont représentées par un graphe dont les nœuds sont définis par les objets et les liens représentent la relation de précédence dérivée des items pris en compte. Les auteurs adoptent une représentation binaire du graphe par une matrice. Le support (Di-Jorio et al., 2009) du motif graduel considéré est défini

comme la longueur du chemin le plus long dans le graphe. Cette approche permet de générer efficacement les motifs graduels de taille $n+1$ à partir des motifs graduels de taille n.

Approche basée sur les correlations de rang : dans (Laurent et al., 2009), les auteurs extraient les gradualités comme des corrélations de rang entre variables. Ils exploitent le coefficient de corrélation de Kendall pour calculer le support d'un motif graduel comme le nombre de couples d'objets ordonnables (concordantes) ou non (discordantes) dans la base de données pour être en accord avec le motif graduel considéré. Cette approche permet de prendre en compte dans une base de données l'amplitude de la discordance des données qui ne satisfont pas les motifs graduels.

2.2 Sur la fouille des motifs graduels sous contraintes

La plupart des algorithmes présentés dans la section 2.1, utilisent des techniques de data mining pour extraire les motifs graduels. Cependant, ils ne sont pas efficaces pour l'extraction de motifs graduels dans certains domaines d'application où les données numériques présentent des formes particulières (par exemple, des données temporelles, de flux, relationnelles ou bruitées). Ainsi, certains travaux récents se sont plutôt focalisés sur l'extraction de variantes de motifs graduels sur les données numériques fournies avec des contraintes spécifiques pour exprimer un autre type de connaissance.

Extraction de motifs graduels à partir de données bruitées : les motifs graduels flous sont revisités dans (Ayouni et al., 2010) pour les données bruitées où il est souvent difficile de comparer les valeurs d'attribut, soit parce que les valeurs sont prises à partir de données bruitées, soit parce qu'il est difficile de considérer qu'une petite différence entre deux valeurs est significative. Un exemple de motif graduel flou pourrait être exprimé comme suit : "plus l'âge d'un employé est proche de 46 ans, plus son revenu est élevé".

Extraction des motifs graduels à partir des bases de données temporelles : récemment, dans (Lonlac et al., 2018; Lonlac et Nguifo, 2020), les auteurs ont proposé une approche pour l'extraction des motifs graduels dans des bases de données temporelles avec une application sur des données paléoécologiques pour appréhender des regroupements fonctionnels de coévolution d'indicateurs paléoécologiques qui modélisent l'évolution de la biodiversité dans le temps. (Owuor et al., 2019) ont proposé une approche d'extraction de motifs graduels temporels flous pour intégrer le fait qu'un décalage temporel peut exister entre les changements de certains attributs et leur impact sur d'autres. Ces motifs graduels temporels flous permettent de détecter les cas de corrélations pertinentes entre les attributs d'une base de données dont les changements dans la valeur d'un attribut provoquent un effet d'entraînement sur d'autres attributs par rapport au temps.

3 Fouille de motifs graduels sous contrainte du seuil graduel

Nous debutons cette section par un rappel de la définition de motifs graduels.

Definition 1 (Item graduel). Un item graduel est defini sous la fomre i^* à partir d'un attribut i et d'un sens de variation $* \in \{\leq, \geq\}$ (ascendant ou descendant). Il est linguistiquement exprimé comme "plus la valeur de i augmente" pour $i^{\geq}$ et "plus la valeur de i diminue" pour $i^{\leq}$.

Un motif graduel $(i_1^{*_1}, \ldots, i_k^{*_k})$ est un ensemble non vide d'items graduels.

La gradualité ou le seuil graduel est très lié à la connaissance du domaine. A partir de cette observation, nous définissons la gradualité comme suite :

Definition 2 (Seuil graduel) : Soit Δ une base de données définie sur un ensemble d'attributs à valeurs numériques I, un seuil graduel est une valeur $\sigma_i (i \in I)$ définie par l'utilisateur ou à partir de la distribution des données de telle sorte que la variation de i entre deux tuples t_1 et t_2 de Δ est considérée si et seulement si $|t_1.i - t_2.i| \geq \sigma_i$, $t.i$ est la valeur de i sur le tuple t.

Nous proposons de calculer $\sigma_i (i \in I)$, à partir de la distribution des valeurs de i.

1. Soit $sd(i)$ l'écart type des valeurs de l'attribut i. σ_i défini à partir de la distribution des valeurs de i est appelé seuil graduel de i. Il est déterminé comme suit :
$$\sigma_i = k_1 \times sd(i) + k_2 \tag{1}$$
2. Soit $cv(i)$ le coefficient de variation (l'écart type relatif) des valeurs de l'attribut i. σ_i déterminé à partir de la distribution des valeurs de i est appelé seuil graduel de i et est défini comme suit :
$$\sigma_i = k_1 \times cv(i) + k_2 \tag{2}$$
3. Soit la composante de i (les valeurs de i sur chaque objet) trié dans l'ordre croissant, σ_i peut également être déterminé comme l'écart type des différents écarts entre deux valeurs consécutives de i. Donc σ_i est calculé comme suit :
$$\sigma_i = k_1 \times st(\Delta_{i_p}) + k_2,\ \Delta_{i_p} = t_{p+1}.i - t_p.i \tag{3}$$

k_1 et k_2 sont deux nombres réels. Lorsque $k_1 = k_2 = 0$, la gradualité est considérée en terme d'augmentation et de diminution des valeurs d'attributs, ce qui ramène au cas des approches de l'état de l'art (Lonlac et al., 2018; Négrevergne et al., 2014; Di-Jorio et al., 2009).

3.1 Algorithme

L'algorithme 1 présente les différentes étapes de notre approche proposée. Nous décrivons ces étapes dans cette section.

Algorithme 1 : GPoID

Input : Δ : base de données numériques, $minSupp$: seuil de support.
Output : M : motifs graduels fréquents

$F \leftarrow SetThreshold(\Delta)$;
$\Delta' \leftarrow Num2Cat(\Delta, F)$;
$M \leftarrow MiningAlgo(\Delta', minSupp)$
$Return\ M$;

Initialisation du seuil graduel : *SetThreshold*. En fonction de la connaissance du domaine, l'expert peut le définir. Dans ce travail, nous proposons les formules (1), (2), (3) pour le calcul de *SetThreshold*.

Transformation de la base de données numériques en une base de données catégorielles : *Num2Cat*. Pour les algorithmes *GRITE* (Di-Jorio et al., 2009) et *T-GPatterns* (Lonlac et al., 2018), qui transforment d'abord la base de données numériques en une base de données catégorielles, le seuil graduel est appliqué lors de la transformation comme suit :

1. Cas de l'approche *T-GPatterns* présentée dans (Lonlac et al., 2018).
   La base de données $\Delta' = \mathcal{T}' \times \mathcal{I}'$ ($|T'| = n-1$ et $|I'| = |I|$) résultant de l'application de la fonction $Num2Cat$ sur la base de données numérique $\Delta = \mathcal{T} \times \mathcal{I}$, est calculée comme suit :
   — $\forall t'_j \in T', t'_j.i_k = " + " \iff t_{j+1}.i_k > t_j.i_k + \sigma_{i_k}$
   — $\forall t'_j \in T', t'_j.i_k = " - " \iff t_{j+1}.i_k < t_j.i_k - \sigma_{i_k}$
   — $t'_j.i_k =$ "o" Sinon
   Cette fonction permet de générer plus de symboles "o" que la fonction ***Num2Cat*** de l'approche (Lonlac et al., 2018) parce que la gradualité est considérée si et seulement si la différence de valeur d'attribut dépasse le seuil graduel.
2. Cas de l'algorithme GRITE présentée dans (Di-Jorio et al., 2009).
   L'étape "binary matrices generation" de GRITE est modifiée par l'introduction du seuil graduel. Soit t_1 et t_2 deux tuples et i un attribut. Les matrices binaires sont calculées comme suit :
   — La matrice M_1 de $i^{\geq}$: $M_{t_1,t_2} = 1 \iff t_2.i - t_1.i \geq \sigma_i$ et 0 sinon.
   — La matrice M_2 de $i^{\leq}$: $M_{t_1,t_2} = 1 \iff t_2.i - t_1.i \leq \sigma_i$ et 0 sinon.
   Comme dans le premier cas, les matrices résultantes seront moins denses que les matrices obtenues avec l'algorithme GRITE.

Fouille des motifs graduels : *MiningAlgo* Pour l'algorithme *T-GPatterns*, l'étape restante est la procédure $searchCoevolution(Apriori(\Delta', minSupp))$. Dans notre proposition, cette étape est effectuée exactement comme proposé par les auteurs. L'introduction des contraintes de seuil dans l'algorithme GRITE ou Paraminer consiste à changer l'étape de traitement où la matrice binaire associée à chaque attribut est calculée. Toutes les autres étapes (*initialisation, jointure avec l'opérateur AND et suppression des tuples*) ne sont pas modifiées.

3.2 Propriétés de l'algorithme *GPoID*

Exactitude : comme l'algorithme *GPoID* est basé sur les algorithmes existants qui se révèlent être corrects, l'étape d'introduction du seuil permet de rendre moins dense la matrice transformée, paramètre d'entrée de l'algorithme de fouille de motifs graduels fréquents (de type Apriori). Comme ces algorithmes s'avèrent corrects, *GPoID* l'est également ;

Complétude : Les algorithmes sur lesquels est basé l'algorithme *GPoID* sont complets, l'étape d'introduction du seuil graduel ne modifie pas cette propriété pour l'algorithme obtenu. La principale conséquence est la réduction des motifs graduels fréquents ;

Complexité : si le seuil des différents attributs est fixé par l'utilisateur, la complexité théorique reste celle de l'algorithme choisi (*GRITE, T-GPatterns, ...*). Mais si ces seuils sont calculés par les formules (1), (2), ou (3), la complexité en temps est augmentée du facteur $n \times m$ où n est le nombre d'objets et m le nombre d'attributs de la base de données. Mais comme ces algorithmes sont basés sur *Apriori*, cette complexité théorique reste exponentielle (2^m), identique à celle obtenue lorsque $k_1 = k_2 = 0$.

3.3 Impact du seuil graduel

L'application d'un seuil graduel affecte les résultats du processus de fouille tant au niveau de la quantité des motifs extraits que sur le support de ces derniers.

Proposition 1. *Soit Δ une base de données numériques, N_1 l'ensemble des motifs graduels fréquents extraits à partir de l'algorithme T-GPatterns ou GRITE, et N_2 l'ensemble des motifs graduels extraits à partir de notre approche, on aura toujours $N_2 \subseteq N_1$.*

Il est clair qu'après l'application du seuil de gradualité, le support de certains motifs graduels diminue. Ainsi, ceci doit être pris en compte au cours de l'exploration.

Propriété 1. *Soit Δ une base de données numériques et i un attribut de Δ. Soit $minSupp$ un seuil de support minimum et σ un seuil de gradualité. L'algorithme T-GPatterns vérifie la relation suivante : si $\sum_{k=1}^{n-1} |t_{k+1}.i - t_k.i| < \sigma \times minSupp$ alors i^* $(* \in \{\leq, \geq\})$ ainsi que tous ses sur-ensembles ne sont pas fréquents dans Δ.*

Preuve. Supposons que $\sum_{k=1}^{n-1} |t_{k+1}.i - t_k.i| < \sigma \times minSupp$, i^* $(* \in \{\leq, \geq\})$. Supposons également que i^* est un item graduel fréquent, selon la définition du motif graduel proposée dans (Lonlac et al., 2018), cela signifie qu'il existe une liste de séquence d'objets consécutifs $s = \langle s_1, \ldots, s_m \rangle$ tel que $|t_{k+1}.i - t_k.i| \geq \sigma$, pour $t_k, t_{k+1} \in s_j (1 \leq j \leq m)$ et $\sum_{j=1}^{m} |s_j| \geq minSupp$. Donc $\sum_{j=1}^{m} \sum_{k=1}^{|s_j|-1} |t_{k+1}.i - t_k.i| \geq \sigma \times minSupp$. Donc, $\sum_{k=1}^{n-1} |t_{k+1}.i - t_k.i| \geq \sigma \times minSupp$, ce qui contredit l'hypothèse initiale.

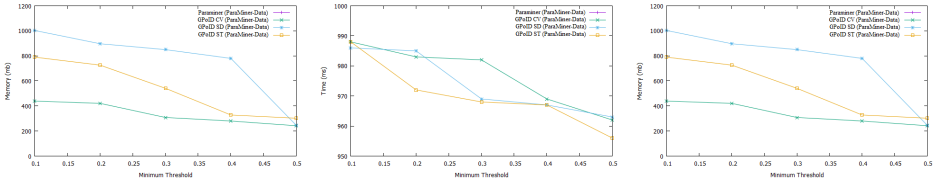
4 Expérimentations

Cette section présente une étude expérimentale du temps d'exécution et du nombre de motifs graduels extraits à l'aide de *GPoID*. Nous évaluons également les performances en termes de consommation mémoire. Nous avons utilisé deux jeux de données pour des soucis d'espace. Le premier jeu de données nommé *Paleo* contient 87 attributs et 111 objets est issu de (Lonlac et al., 2018) et le deuxième nommé *ParaMiner-Data* est constitué de 4413 attributs et 109 objets issu de (Négrevergne et al., 2014). Toutes les expériences sont menées sur un ordinateur de 8Go de RAM, de processeur Intel(R) Core(TM) i5-8250U. Nous comparons d'une part l'implémentation R de *GPoID* avec l'implémentation R originale de *T-Gpatterns*, et d'autre part l'implémentation C++ de *GPoID* (*GPoID-ParaMiner*) avec l'implémentation C++ originale de *ParaMiner*.

Le code source de notre algorithme proposé *GPoID* (respectivement *GPoID-ParaMiner*) peut être obtenu à partir de `https://github.com/chirmike/GPoID` (respectivement `https://github.com/Chirmeni/GPoID-Paraminer`).

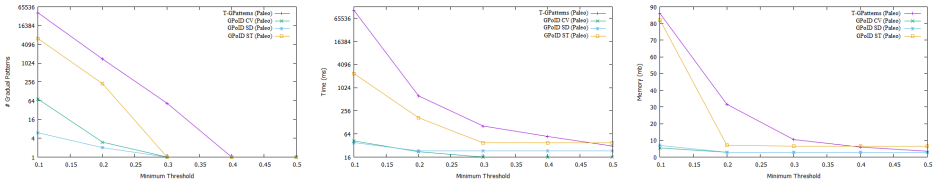
Les résultats sont présentés en deux étapes : nous présentons premièrement les résultats obtenus sur des données non temporelles (figures 1), ensuite, ceux obtenus sur des données temporelles (figures 2). Nous avons fait varier le seuil de support dans l'intervalle $[0.1, 0.5]$ avec un pas de 0.1. Tout au long de nos expériences, nous avons fixé k_1 à 1 et k_2 à 0.

Quel que soit le type de données, nous observons des figures 1 et 2 que le nombre de motifs graduels fréquents extraits avec *GPoID-ParaMiner* (resp. *GPoID*) est considérablement réduit par rapport au nombre de motifs extraits avec *ParaMiner* (resp. *T-GPatterns*) sur des données non temporelles (resp. temporelles). Par exemple, sur la figure 2 pour une valeur de seuil de support égale à 0.1, *GPoID* extrait 72 motifs graduels fréquents lorsque le seuil graduel est défini par l'équation 2, alors que *T-GPatterns* extrait 41867. Ceci est un avantage considérable pour l'expert car il est facile d'analyser 72 motifs par rapport à 41867. De plus, les motifs



(a) (b) (c)

FIG. 1: Etude comparative en nombre de motifs graduels fréquents (a), en temps (b) et en usage mémoire (c) de quatre algorithmes sur les données ParaMiner-Data.



(a) (b) (c)

FIG. 2: Etude comparative en nombre de motifs graduels fréquents (a), en temps (b) et en usage mémoire (c) de quatre algorithmes sur la base de données paléoécologique.

graduels fréquents extraits avec *GPoID* ont des informations supplémentaires : ce sont des motifs avec des gradualités supérieures à un seuil. Sur les temps d'exécution, l'introduction du seuil graduel rend *GPoID-ParaMiner* (resp. *GPoID*) plus rapide que *Paraminer* (resp. *T-GPatterns*) sur les deux jeux de données utilisés. De plus, *GPoID* est évolutif car il arrive à extraire les motifs graduels dans les bases de données (cas paramMiner-Data) où Paraminer échoue (c'est la raison pour laquelle sur la figure 1 nous observons uniquement trois courbes). *GPoID* reste meilleur en consommation mémoire par rapport à *ParaMiner* (resp. *T-GPatterns*).

5 Conclusion

Cet article propose une approche pour extraire les motifs graduels à partir des données imprécises. Elle prend en compte les préférences utilisateurs sur la distribution des données et les introduit dans le processus de fouille. Cette préférence est un seuil graduel à partir duquel considérer une gradualité. L'impact de la prise en compte du seuil graduel sur l'exploitation des résultats de fouille par l'utilisateur a été aussi étudié. Les résultats expériementaux montrent que l'introduction du seuil graduel dans le processus de fouille permet non seulement de capturer l'imprécision dans les données lors de la génération des motifs à partir de données numériques imprécises, mais également de réduire considérablement la quantité de motifs extraits, le temps d'extraction, et la quantité de mémoire consommée. En outre, l'introduction du seuil graduel supprime les motifs graduels bruités. Néanmoins, il serait intéressant d'étudier l'impact de l'introduction du seuil graduel sur la qualité des motifs extraits, ainsi que sur le choix du seuil de support.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In *VLDB*, pp. 487–499.

Ayouni, S., S. B. Yahia, A. Laurent, et P. Poncelet (2010). Fuzzy gradual patterns : What fuzzy modality for what result ? In *SoCPaR*, pp. 224–230.

Di-Jorio, L., A. Laurent, et M. Teisseire (2008). Fast extraction of gradual association rules : a heuristic based method. In *CSTST*, pp. 205–210.

Di-Jorio, L., A. Laurent, et M. Teisseire (2009). Mining frequent gradual itemsets from large databases. In *IDA*, pp. 297–308.

Flores, P. M. Q., F. del Razo Lopez, N. Sicard, et A. Laurent (2012). Consommation mémoire et puissance de calcul en fouille de motifs graduels basée sur les ordres flous multi-précisions. In *LFA*, pp. 1–7.

Laurent, A., M. Lesot, et M. Rifqi (2009). GRAANK : exploiting rank correlations for extracting gradual itemsets. In *FQAS*, pp. 382–393.

Lonlac, J., Y. Miras, A. Beauger, V. Mazenod, J.-L. Peiry, et E. M. Nguifo (2018). An approach for extracting frequent (closed) gradual patterns under temporal constraint. In *FUZZ-IEEE*, pp. 878–885.

Lonlac, J. et E. M. Nguifo (2020). A novel algorithm for searching frequent gradual patterns from an ordered data set. *Intell. Data Anal. 24*(5), 1029–1042.

Négrevergne, B., A. Termier, M. C. Rousset, et J. F. Méhaut (2014). Para miner : a generic pattern mining algorithm for multi-core architectures. *DMKD 28*(3), 593–633.

Owuor, D., A. Laurent, et J. Orero (2019). Mining fuzzy-temporal gradual patterns. In *FUZZ-IEEE*, pp. 1–6.

Shah, F., A. Castelltort, et A. Laurent (2020). Handling missing values for mining gradual patterns from nosql graph databases. *Future Gener. Comput. Syst. 111*, 523–538.

Zadeh, L. A., G. J. Klir, et B. Yuan (1996). *Fuzzy sets, fuzzy logic, and fuzzy systems : selected papers*, Volume 6. World Scientific.

Summary

In recent years, gradual patterns have gained the attention of the data science community and several algorithms have been designed to extract these patterns from different data models. On some data, like imprecise data, one of the biases in traditional algorithms is that they define graduality as an increase/decrease in value. Therefore, some of the graduations extracted are just a noise effect in the data. To remedy this problem, this paper proposes a method which introduces into the mining process, a gradual threshold from which to consider a graduality. Experiments on different databases show that our proposal reduces computation times and the number of patterns generated by focusing on patterns of interest. Additionally, she extracts gradual patterns in some cases where traditional approaches fail.