Prédiction au plus tôt d'événements par règles d'épisodes

Lina Fahed, Armelle Brun, Anne Boyer

Université de Lorraine - LORIA - Équipe KIWI
Campus scientifique BP 239 54506 Vandoeuvre-lès-Nancy Cedex
{Lina.Fahed, Armelle.Brun, Anne.Boyer}@loria.fr

Résumé. Le nombre de messages déposés quotidiennement sur les réseaux sociaux et les blogs est colossal. Ces messages constituent une source d'informations précieuse. Leur exploration se place dans le domaine de la fouille de données temporelles. Cette fouille peut être utilisée dans un but de prédiction afin d'anticiper l'apparition de certaines informations. Notre objectif dans cet article est de proposer un algorithme permettant la prédiction d'informations au plus tôt et de façon fiable.

Dans ce travail préliminaire, nous nous focalisons sur l'extraction de règles d'épisodes minimales, pouvant être identifiées rapidement dans un flux de données. Ces règles auront aussi une conséquence éloignée temporellement de l'antécédent, pour anticiper au plus tôt la conséquence, tout en ayant une confiance élevée.

1 Introduction

Avec l'émergence du web 2.0, les internautes ne sont plus de simples consommateurs, ils sont également acteurs par le biais des messages qu'ils peuvent déposer, des tags ou commentaires qu'ils peuvent laisser et de toute action qu'ils peuvent effectuer. Dans ce cadre, les messages laissés dans les réseaux sociaux représentent une source importante et précieuse d'informations, que de nombreuses recherches cherchent à analyser dans le but d'en comprendre le contenu, d'en extraire les relations cachées, mais aussi de prédire de l'information.

Le flux de données (de messages) peut être considéré comme une séquence ordonnée par la date de création des messages. On appellera "item" un élément représentant un message (mot du message, opinion ou sujet extrait, etc.). À un temps t donné, plusieurs items apparaissent donc dans cette séquence : l'ensemble des items du message créé au temps t. Ce type de séquence est appelée "séquence complexe".

La fouille de données temporelles concerne la fouille de données séquentielles ordonnées selon certains critères comme par exemple le temps ou la position (Laxman et Sastry, 2006). Par conséquent, l'analyse du flux de messages issus des réseaux sociaux peut être vue comme une tâche de fouille de données temporelles. Les traces de navigation (Priya et Vadivel, 2012), les séquences biologiques (Bathoorn et al., 2010), ou les rapports météo (Basak et al., 2004), peuvent représenter d'autres exemples de données temporelles.

L'extraction de motifs (et surtout de motifs "séquentiels" dans lesquels l'ordre est considéré) est une tâche essentielle dans la fouille de données temporelles pour extraire des items

corrélés et des relations (ordonnées) entre items. Dans certains cas, les motifs sont utilisés pour former des règles d'association qui peuvent être exploitées pour prédire des informations : la conséquence des règles (Agrawal et al., 1994).

Une sous-tâche de l'extraction de motifs, adaptée aux données séquentielles représentée sous la forme d'une unique séquence, est l'extraction d'épisodes. Un épisode est un motif temporel composé d'items "relativement proches", qui apparaît souvent tout au long de la séquence ou sur une partie de cette séquence (Mannila et al., 1997). L'extraction d'épisodes diffère de celle de motifs séquentiels par la mesure de fréquence des épisodes. À l'opposé des premiers algorithmes d'extraction d'épisodes, qui extrayaient un ensemble complet d'épisodes, certains algorithmes récents visent à extraire, pour des raisons de complexité, un sous-ensemble d'épisodes respectant certaines caractéristiques. Par exemple, nous pouvons citer l'ensemble des épisodes fermés (Tatti et Cule, 2012) ou des épisodes maximaux (Iwanuma et al., 2005). Tout comme il est possible d'extraire des règles d'association à partir de motifs, des règles d'épisodes peuvent être extraites à partir d'épisodes. Ces règles d'épisodes peuvent aussi être utilisées dans un objectif de prédiction (Daurel, 2003).

La majorité des règles d'épisodes construites à partir d'épisodes fermés ou maximaux ont la caractéristique d'avoir un antécédent long (composé de nombreux items). De notre point de vue, elles ne sont pas adaptées à la prédiction au plus tôt d'informations. En effet, pour prédire une conséquence, il faut attendre l'apparition de la totalité des items de l'antécédent. Cependant, notre objectif est d'anticiper, au plus tôt, des événements. Dans ce cas, pour détecter rapidement des événements, une règle d'épisodes avec un antécédent plus petit est plus pertinente qu'une règle avec un antécédent plus long. Par conséquent, notre premier objectif est d'extraire des règles d'épisodes composées d'un antécédent de taille minimale, que nous appellerons "règles d'épisodes minimales" .

Dans la tâche d'identification au plut tôt d'informations, nous faisons l'hypothèse qu'il est inutile de chercher à former des règles d'épisodes très complexes ou très précises, mais des règles d'épisodes permettant d'identifier les premiers signaux. Nous faisons donc l'hypothèse qu'il est inutile d'extraire les règles d'épisodes contenant plusieurs fois le même item. Il est vrai qu'une apparition multiple d'un item porte plus d'informations, mais cette caractéristique permettra de diminuer la complexité de l'algorithme. Nous appelons "épisode sans répétition" un épisode composé d'items uniques. Notre second objectif est donc d'extraire des épisodes sans répétition.

Nous souhaitons pouvoir anticiper/prédire des informations "lointaines". Notre troisième objectif est donc d'extraire des règles d'épisodes ayant une conséquence éloignée temporellement de l'antécédent.

La suite de cet article est organisée comme suit : la section 2 présente plusieurs travaux de l'état de l'art sur la fouille de données temporelles et en quoi ils ne peuvent répondre à notre besoin. Dans la section 3, nous définissons quelques notions qui seront utilisées dans notre algorithme et la section 4 détaille l'algorithme EMMA, dont s'inspire notre travail. Notre algorithme est présenté dans la section 5. Nous conclurons et fournirons quelques perspectives dans la section 6.

2 État de l'art

Les méthodes d'extraction de motifs varient en fonction des données disponibles. Dans le cas de données transactionnelles, premier type de données auxquelles s'est intéressée la fouille de données, des motifs non séquentiels (itemsets) sont extraits de l'ensemble des transactions. Plusieurs algorithmes sont très populaires dans l'état de l'art, comme Apriori (Agrawal et al., 1994) et FPGrowth (Han et al., 2000).

Dans le cas de données séquentielles, dans lesquelles l'ordre entre les transactions est considéré, des motifs séquentiels sont extraits. Les algorithmes sont en règle générale une adaptation des algorithmes d'extraction d'itemsets, comme prefixspan (Han et al., 2001). Plusieurs méthodes proposent l'extraction de règles d'association à partir de motifs (règles d'association séquentielles dans le cas de données séquentielles). Une règle d'association est constituée de deux parties : un antécédent et une conséquence, et signifie que si l'antécédent est vrai, alors la conséquence est vraie. Pour mesurer l'efficacité de règles d'association, deux mesures principales sont utilisées : le support et la confiance (Agrawal et al., 1994), où le support est le nombre de transactions contenant à la fois l'antécédent et la conséquence, et la confiance est la probabilité d'apparition de la conséquence lorsque l'antécédent est apparu.

Dans le cas de données composées d'une unique longue séquence de données ordonnée par le temps, les motifs extraits sont des épisodes et des règles d'épisodes peuvent en être déduites (Mannila et al., 1997). À la différence des motifs séquentiels, dans lesquels seul l'ordre est utilisé, les méthodes d'extraction d'épisodes utilisent le temps d'apparition. De plus, la fréquence d'apparition des motifs séquentiels est évaluée comme étant le nombre de transactions dans lesquelles le motif est apparu, alors que la fréquence d'apparition des épisodes est évaluée en fonction du nombre d'occurrences de l'épisode dans l'unique séquence de données.

Mannila (Mannila et al., 1997) a proposé les premiers algorithmes d'extraction d'épisodes : $Winepi$ et $Minepi$ qui seront la base de nombreux autres algorithmes proposés par la suite. Les deux algorithmes extraient dans un premier temps les épisodes les plus petits, puis cherchent itérativement des épisodes plus grands en se basant sur leur fréquence. Les deux algorithmes diffèrent par la méthode utilisée pour compter le nombre d'occurrences des épisodes. $Winepi$ fait glisser une fenêtre de largeur w et compte le nombre de fenêtres qui contiennent l'épisode. $Minepi$ n'utilise pas la notion de fenêtre mais cherche les intervalles les plus petits de l'épisode qui ne contiennent aucune autre occurrence du même épisode ou d'un de ses sous-épisodes, ce qui est appelé "l'occurrence minimale d'un épisode". Ces méthodes ont la caractéristique d'extraire un ensemble complet d'épisodes.

La fouille d'épisodes dans des séquences de données complexes est une problématique récente. Elle nécessite un algorithme adapté qui doit prendre en compte l'existence de plusieurs items à chaque temps afin d'extraire les épisodes. Huang et Chang (Huang et Chang, 2008) proposent un algorithme appelé EMMA qui extrait un ensemble complet d'épisodes à partir d'une séquence complexe. Dans les deux premières phases, EMMA pré-traite la séquence complexe : il extrait un ensemble de motifs fréquents de chaque temps. À partir de ces motifs, les épisodes sont construits itérativement pendant une troisième phase. EMMA est un algorithme rapide et facile à adapter. La communauté d'extraction de motifs fréquents admet que dans la plupart des applications, il n'est pas nécessaire d'extraire tous les épisodes mais qu'il est suffisant d'extraire un ensemble d'épisodes condensé et significatif (Pasquier et al., 1999). En effet, non seulement les algorithmes qui extraient un ensemble complet d'épisodes sont

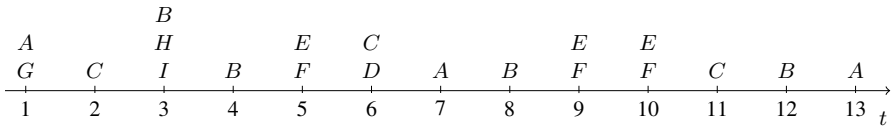

FIG. 1 – *Séquence complexe.*

complexes, mais de plus la taille de l'ensemble résultat affecte leur bonne compréhension et leur analyse. Afin d'avoir un ensemble d'épisodes plus concis et pertinent et un algorithme de plus faible complexité, les travaux récents se focalisent sur la détection d'épisodes comportant certaines caractéristiques et contraintes de façon à rendre cet ensemble plus significatif. Cette notion varie selon les auteurs. Cependant, dans la plupart des travaux, elle a pour but d'extraire un ensemble d'épisodes fermés (Tatti et Cule, 2012) ou maximaux (Iwanuma et al., 2005).

Certains travaux ont proposé d'extraire un ensemble condensé d'épisodes dans une séquence complexe, en impliquant certaines caractéristiques comme le fait que les épisodes soient maximaux (Gan et Dai, 2011) ou approximativement fermés et non dérivables (Gan et Dai, 2012). Cependant, dans ces épisodes, le même item peut apparaître plusieurs fois. Pour certaines applications, cette caractéristique peut rendre les règles non pertinentes, et on peut ne pas souhaiter l'apparition multiple d'un item dans un épisode. Dans ce cas, les algorithmes proposés dans ces deux travaux sont non applicables, et cette caractéristique ne peut pas être modifiée parce qu'elle représente le cœur des algorithmes.

Comme nous l'avons mentionné dans l'introduction, nous voulons extraire les règles d'épisodes ayant certaines caractéristiques, comme le fait d'être minimales. La notion "minimale" n'est pas nouvelle dans l'état de l'art. Elle a été proposée par Rahal (Rahal et al., 2004) dans le cadre d'extraction de règles d'association fiables, peu fréquentes ayant l'antécédent le plus petit et la conséquence fixée par l'utilisateur à partir de données transactionnelles dans le but de prédire au plus tôt la conséquence. Cette approche est différente de notre travail. En effet, nous voulons extraire les règles d'épisodes (et non pas des règles d'association). De plus, nous cherchons les règles d'épisodes sans répétition et avec une conséquence temporellement éloignée de l'antécédent.

Nous présentons maintenant quelques définitions, appuyées par un exemple de données, qui seront utiles pour la suite de cet article.

3 Définitions et exemples

Séquence complexe : Soit I un ensemble fini d'items de taille m. I_t est l'ensemble des items qui apparaissent au temps t. S est une suite ordonnée de paires (temps t, item i_t), appelée une séquence de données. S est qualifiée de séquence complexe quand apparaît au temps t un ensemble d'items I_t (voir figure 1). $S = < (t_1, I_{t_1}), (t_2, I_{t_2}), ..., (t_n, I_{t_n}) >$ avec $t_1 < t_2 < ... < t_n$. Le nombre de paires dans la séquence S est n .

Fenêtre : Dans l'extraction d'épisodes à partir d'une séquence S, la notion de fenêtre est souvent utilisée pour représenter un segment de la séquence. Notons $Win(S, st, w)$ le segment

dans la séquence S qui commence au temps st (start) et qui a la longeur w. Certains algorithmes d'extraction d'épisodes utilisent des fenêtres glissantes où st varie de 1 à n. Dans la figure 1, une fenêtre glissante de longeur 6 se décale tout au long de la séquence S, portant sur les intervalles suivants : ([1,6], ... ,[11,13],[12,13],[13,13]).

Épisode, sous-épisode, épisode sans répétition, concaténation :

Un épisode $P =< p_1, p_2, ..., p_k >$ sur I^k est une suite ordonnée d'ensembles d'items. Soit l'épisode $Q =< q_1, q_2, ..., q_s >$. P est un sous-épisode de Q si chaque $p_i \subseteq q_i$ (p_i peut être $\emptyset$). Par exemple, l'épisode $P = (< A >, < BD >)$ est un sous-épisode de $Q = (< AF >, < BD >)$. Au contraire, l'épisode $R = (< AB >)$ n'est pas un sous-épisode de Q. Dans certains cas, il est possible de trouver $p_i \subseteq p_j$ où $i \neq j$, ce qui signifie une apparition multiple d'un item dans le même épisode. Par exemple, dans l'épisode $(< A >, < AF >)$ l'item A apparaît plusieurs fois. Nous appelons "épisode sans répétition" un épisode qui ne contient pas d'items apparaissant plusieurs fois. Un épisode P peut apparaître plusieurs fois dans la séquence S et il peut être considéré comme un sous-épisode de la séquence S. On dit que $Win(S, st, w)$ contient $P =< p_1, ..., p_k >$ si p_1 est un sous-ensemble de I_{st} et le reste de P est contenu dans $< I_{st+1}, ..., I_{st+w} >$. Le nombre de fenêtres dans lesquelles P apparaît s'appelle le support de P. P est fréquent dans la séquence S si $support(P) \geq minsupp$ où $minsupp$ est le seuil minimum de support pré-déterminé.

Soient les deux épisodes P, Q. La concaténation de P et Q est $P.Q =< p_1, ..., p_k, q_1, ..., q_s >$ de taille $k + s$.

Règle d'épisodes : Soit un épisode fréquent $R = P.Q$. À partir de R, une règle d'épisodes $P \rightarrow Q$ signifie que Q apparaît après P avec une probabilité élevée. Nous appelons "règle d'épisodes sans répétition" une règle $P \rightarrow Q$ si $P.Q$ est un épisode sans répétition.

Mesures de règles d'épisodes : Les mesures d'évaluation de règles d'épisodes sont globalement les mêmes que pour les règles d'association. Le $support$ représente la fréquence de la règle et la $confiance$ représente la probabilité d'apparition de Q après P. Nous définissons le support et la confiance de la règle $P \rightarrow Q$ pour une séquence de taille n comme suit :

$$support(P \rightarrow Q) = \frac{support(P.Q)}{n} \qquad confiance(P \rightarrow Q) = \frac{support(P.Q)}{support(P)} \tag{1}$$

$(P \rightarrow Q)$ est considérée **fréquente** si $support(P \rightarrow Q) \geq minsupp$, et **fiable** si $confiance(P \rightarrow Q) \geq minconf$. Nous appelons **règle acceptable** une règle fréquente et fiable. Par exemple, l'épisode $< A >< B >$ a un nombre d'occurrences de 4 dans 2 fenêtres de largeur $w = 6$. Comme le support d'un épisode représente le nombre de fenêtres dans lesquelles il apparaît (pas l'occurrence de l'épisode), donc $support(< A >< B >) = 2/13 = 15.3\%$. À partir de cet épisode, une règle d'épisodes $RE =< A > \rightarrow < B >$ peut être construite. Le support de RE est $2/13 = 15.3\%$ et la confiance $2/3 = 66\%$. Pour un $minsupp = 15\%$ et $minconf = 60\%$, RE est fréquente et fiable et donc acceptable.

Liste de bornes : Une borne d'un épisode P dans la séquence S est un intervalle $[t_s, t_e]$, où $(t_e - t_s) < w$ et où p_1 (premier élément de P) est un sous-épisode de I_{t_s} (ensemble des items apparaissant au temps t_s), p_k (dernier élément de P) est un sous-épisode de I_{t_e}, et

$p_2, ...p_{k-1}$ est un sous-épisode de $(I_{t_s+1}, ..., I_{t_e-1})$. La liste de bornes $Borne(P)$ de P est la liste de tous les intervalles dans S dans lesquels P apparaît, et $Borne_i(P)$ est la $i^{ème}$ borne de P. Dans la figure 1, l'épisode $< A >$ a la liste de bornes $[1, 1], [7, 7], [13, 13]$, et l'épisode $< A >< B >$ a la liste de bornes $[1, 3], [1, 4], [7, 8], [7, 12]$ pour $w = 6$. Les bornes permettent de prendre en compte des mesures de temps et de distance entre items. Notons que l'ensemble final d'épisodes ne contient pas ces mesures, seul l'ordre est représenté.

Règle d'épisodes minimale : Soit la règle d'épisodes $RE = (P \rightarrow Q)$. Si RE est acceptable, RE est minimale s'il n'existe pas de règle $RE' = (R \rightarrow Q)$ acceptable où $R \subset P$.

Nous présentons maintenant l'algorithme de l'état de l'art EMMA, dont est inspiré notre algorithme.

4 Algorithme existant : EMMA

EMMA (Episode Mining using Memory Anchor) (Huang et Chang, 2008) est un algorithme d'extraction d'épisodes fréquents dans une séquence complexe. Il a la caractéristique d'extraire un ensemble complet d'épisodes. EMMA extrait d'abord les motifs fréquents, pour chaque temps t, dans l'ensemble des items apparaissant à t. Cette première caractéristique permet de diminuer la complexité de l'algorithme en diminuant la grande taille des motifs extraits. Les épisodes sont ensuite construits incrémentalement en prenant en compte la liste de bornes de chaque 1-uplet épisode. L'algorithme EMMA comporte trois phases résumées dans les sous-sections suivantes.

4.1 Extraction de motifs fréquents

EMMA utilise un algorithme appelé FIMA (Frequent Itemset mining using Memory Anchor) pour extraire un ensemble de motifs fréquents locaux : LFP (Local Frequent Patterns). Tous les items dans un motif fréquent ayant le même temps d'apparition t, le motif est dit "local". FIMA commence par une lecture de la séquence de données pour trouver les motifs locaux de taille 1. Ensuite, pour chaque motif local, une combinaison est faite avec un autre motif du même temps t. Cette procédure de combinaison est appliquée récursivement afin d'avoir des motifs plus longs. À chaque itération, les motifs sont choisis en fonction de leur support. Ensuite, une liste de bornes de chaque motif local fréquent est créée. Notons qu'un motif fréquent extrait représente un 1-uplet épisode.

4.2 Encodage de la séquence de données

Une fois l'ensemble des 1-uplet épisodes extrait, un identifiant id est associé à chacun (voir tableau 1). La séquence de données (figure 1) est encodée par les id de 1-uplet épisodes dans chaque intervalle selon leur liste de bornes (voir tableau 2). L'ensemble des id est appelé ID.

4.3 Phase principale : Extraction des épisodes fréquents

La dernière étape de l'algorithme EMMA extrait les épisodes fréquents en utilisant la séquence de données encodée. Chaque id représente un préfixe d'un épisode à construire (le

id	1-uplet épisode	Liste de bornes
#1	A	[1,1], [7,7], [13,13]
#2	B	[3,3], [4,4], [8,8], [12,12]
#3	C	[2,2],[6,6], [11,11]
#4	E	[5,5], [9,9], [10,10]
#5	F	[5,5], [9,9], [10,10]
#6	EF	[5,5], [9,9], [10,10]

TAB. 1 – *Encodage des* 1*-uplet épisodes.*

Temps	1	2	3	4	5	6	7	8	9	10	11	12	13
id	#1	#3	#2	#2	#4 #5 #6	#3	#1	#2	#4 #5 #6	#4 #5 #6	#3	#2	#1

TAB. 2 – *Encodage de la séquence de données.*

premier élément de l'épisode). Pour chaque id et selon sa liste de bornes, on cherche les autres id fréquents dans sa liste de bornes projetées définie comme suit :

Liste de bornes projetées : Pour une borne d'un épisode P, une borne projetée représente l'intervalle de temps (de taille au plus w) où les épisodes candidats (super-épisodes de P) sont recherchés. Pour l'épisode P, qui a sa $i^{ème}$ borne $Borne_i(P) = [t_{s_i}, t_{e_i}]$, la borne projetée de $Borne_i(P)$ est $ProjBorne_i(P) = [t_{s_i} + 1, t_{s_i} + w - 1]$. Par exemple, pour $w = 6$, le 1-uplet épisode $\#1$ a la liste de bornes projetées suivante : $ProjBorne(\#1) = \{[2, 6], [8, 12]\}$.

Un nouvel épisode candidat est alors construit en étendant chaque id avec chacun des id fréquents dans sa liste de bornes projetées. Dans ce cas, $\#1$, $\#2$, $\#3$, $\#4$, $\#5$, $\#6$ qui sont fréquents sont tous étendus, pour $minsupp = 15\%$ (tableau 2). Pour chaque épisode candidat, sa liste de bornes est ensuite calculée. Si cet épisode est fréquent, il est de nouveau étendu avec un nouvel id. Par exemple, pour construire l'épisode $< \#1, \#2 >$, sa liste de bornes est $Borne(< \#1, \#2 >) = [1, 3], [1, 4], [7, 8], [7, 12]$. $support(< \#1, \#2 >) = 2/13 = 15.3\%$, donc $< \#1, \#2 >$ est un épisode fréquent et il sera étendu avec d'autres id. EMMA utilise un parcours en profondeur pour étendre les épisodes, et l'itération s'arrête quand l'épisode candidat n'est pas fréquent ou s'il n'y a plus d'id avec lequel l'étendre. Cette façon de construire les épisodes peut mener à des épisodes contenant plusieurs occurrences du même item.

5 Notre approche : extraction de règles d'épisodes minimales

Rappelons que notre objectif est de pouvoir anticiper au plus tôt des événements. Pour cela, nous proposons un algorithme d'extraction de règles d'épisodes qui, en plus d'être fréquentes et fiables comme celles extraites par les algorithmes de l'état de l'art, sont sans répétition, minimales et avec une conséquence éloignée temporellement de l'antécédent.

Les règles d'épisodes extraites par notre algorithme sont minimales, c'est-à-dire avec un antécédent le plus petit possible, car nous partons de l'hypothèse que plus l'antécédent est petit, plus tôt nous l'identifierons dans une séquence, et plus tôt nous pourrons prévoir/anticiper la conséquence. À notre connaissance, cela est contraire aux algorithmes d'extraction de règles d'épisodes de l'état de l'art dont l'objectif est d'extraire des règles d'épisodes maximales ou les plus longues possibles tout en respectant certaines caractéristiques.

Les règles d'épisodes extraites par notre algorithme ont également la caractéristique d'avoir une conséquence la plus éloignée possible de l'antécédent, ce qui permettra de détecter les événements lointains.

Les règles d'épisodes extraites ont enfin la caractéristique d'être sans répétition car nous considérons que cette unicité permettra de réduire la complexité de l'algorithme.

Nous proposons une approche novatrice pour extraire des règles d'épisodes avec les caractéristiques souhaitées. Notre algorithme utilise les mêmes étapes d'initialisation et d'encodage qu'EMMA. Nous différons d'EMMA en deux points : notre algorithme extrait des règles d'épisodes, au contraire d'EMMA qui extrait des épisodes. De plus, grâce aux caractéristiques souhaitées, notre algorithme extrait un ensemble réduit pas complet de règles d'épisodes, au contraire d'EMMA qui extrait un ensemble complet d'épisodes.

Le principe de notre algorithme est le suivant : chaque règle d'épisodes est construite en fixant un 1-uplet épisode comme préfixe de la future règle, puis en fixant la conséquence, qui est la plus éloignée possible de ce préfixe, et enfin en complétant l'antécédent avec des 1-uplet épisodes les plus proches possibles du préfixe. L'algorithme 1 présente notre proposition. Nous détaillons ci-dessous cet algorithme.

5.1 Déroulement de l'algorithme

Identification du préfixe : Chaque id obtenu dans la deuxième étape d'EMMA représente un préfixe d'un épisode potentiel. Soit id_i un préfixe, nous construisons $P_{roj}ID_{fin}(id_i)$ la liste des id présents à la fin des bornes projetées de id_i (dernières positions des bornes projetées de id_i). Ces id sont stockés dans cette liste dans l'ordre décroissant du nombre de bornes dans lesquelles ils apparaissent. Nous construisons également $P_{roj}ID_{deb}(id_i)$, la liste des id présents au début des bornes projetées de id_i. Les id sont stockés dans cette liste dans l'ordre décroissant du nombre de bornes dans lesquelles ils apparaissent. La taille des intervalles "début" et "fin" est une proportion de w. Cette représentation nous permettra d'accélérer la construction des règles d'épisodes minimales car elle va éviter des itérations inutiles.

Par exemple, soit la taille de l'intervalle égale à 2, alors pour $\#1$, c'est $\#3$ qui est apparu le plus souvent à la fin des bornes projetées de $\#1$, il est apparu 2 fois. Ensuite, $\#2, \#4, \#5, \#6$ ne sont apparus qu'une seule fois à la fin des bornes projetées de $\#1$. Donc, $P_{roj}ID_{fin}(\#1)=$ $(\#3, \#2, \#4, \#5, \#6)$. C'est cette liste qui va nous servir pour identifier les conséquences de toutes les règles d'épisodes qui auront $\#1$ comme préfixe.

Identification de la conséquence : id_i est étendu avec les éléments de $P_{roj}ID_{fin}(id_i)$ qui représentent une conséquence potentielle d'un épisode dont le premier élément est id_i. id_i est étendu avec id_j (qui sera la conséquence de la règle), formant ainsi un épisode candidat $< id_i, id_j >$. La fréquence de cet épisode candidat est calculée en utilisant le nombre de bornes dans sa liste de bornes. Si $< id_i, id_j >$ n'est pas fréquent, on arrête cette itération et on

considère que id_j ne peut pas être une conséquence de id_i. On itère de nouveau pour étendre id_i avec d'autres $id \in P_{roj}ID_{fin}(id_i)$. Si $< id_i, id_j >$ est fréquent, alors la règle $id_i \rightarrow id_j$ est construite. Si elle est fiable, elle est considérée comme acceptable. Elle n'est donc plus étendue et elle est ajoutée à la liste finale des règles d'épisodes LRE. En effet, elle est non seulement fiable mais en plus elle est minimale, elle remplit donc notre objectif. Si la règle est fréquente mais pas fiable, alors le 2-uplet épisode $< id_i, id_j >$ est étendu pour compléter l'antécédent, comme suit :

Complétion de l'antécédent : Rappelons que nous cherchons à compéter l'antécédent de façon à obtenir non seulement une règle fiable mais aussi une règle minimale (avec un antécédent minimal). Les itérations de complétion de l'antécédent s'arrêtent donc dès que la règle d'épisodes est $fiable$, ou s'il n'y a plus d'épisode candidat.

À partir de $< id_i, id_j >$, id_i est étendu avec les id dans sa liste $P_{roj}ID_{deb}(id_i)$ itérativement. Soit $id_s \in P_{roj}ID_{deb}(id_i)$, si l'épisode candidat $< id_i, id_s, id_j >$ n'est pas fréquent, alors le préfixe de l'antécédent id_i est étendu avec un autre $id \in P_{roj}ID_{deb}(id_i)$, sinon la confiance de la règle $id_i, id_s \rightarrow id_j$ est calculée. Si la règle n'est pas fiable, alors l'épisode $< id_i, id_s >$ est étendu avec d'autres $id \in P_{roj}ID_{deb}(id_i, id_s)$, sinon elle est acceptable, et par définition minimale, son antécédent n'est donc plus étendu et elle est ajoutée à la liste LRE.

Épisode sans répétition : Comme nous l'avons mentionné, notre algorithme a pour but d'éviter l'apparition multiple d'un item dans les règles d'épisodes. À chaque fois qu'un épisode doit être étendu, une vérification est faite pour éviter de l'étendre avec des id ayant des items en commun, ce qui aura également l'avantage de réduire la complexité de l'algorithme.

En résumé : Notre algorithme permet non seulement d'obtenir des règles avec les trois caractéristiques souhaitées, mais il est également moins complexe qu'EMMA. Dans la plupart des algorithmes d'extraction d'épisodes, la complexité est dépendante de la taille de l'épisode. En extrayant les règles d'épisodes portant nos caractéristiques, l'algorithme consomme théoriquement moins de temps et de mémoire, ce qui signifie pour nous une complexité diminuée, par rapport aux autres algorithmes de l'état de l'art. En effet, en cherchant à extraire des règles d'épisodes minimales, sans répétition et avec une conséquence éloignée, les stratégies choisies pour former ces règles permettent d'éviter plusieurs itérations. Par exemple, les stratégies de recherche de la conséquence et de la construction de l'antécédent (utilisant les deux listes de début et de fin dans les bornes projetées) rendent l'algorithme moins complexe parce qu'elles réduisent les intervalles de recherche des id candidats.

5.2 Exemple

Reprenons l'exemple précédent. Nous avons $P_{roj}ID_{fin}(\#1)$=$(\#3, \#2, \#4, \#5, \#6)$, et $P_{roj}ID_{deb}(\#1)$ =$(\#2, \#3, \#4, \#5, \#6)$. Nous commençons par la phase d'identification de la conséquence du préfixe $\#1$, en l'étendant avec les id dans sa liste $P_{roj}ID_{fin}(\#1)$. Pour la conséquence candidate $\#3$, $< \#1, \#3 >$ est fréquent parce que $support(< \#1, \#3 >) = 15.3\% > minsupp\%$, donc nous fixons la conséquence $\#3$ pour le préfixe $\#1$. Ensuite, la règle $\#1 \rightarrow \#3$ est construite et sa confiance est calculée : $confiance(\#1 \rightarrow \#3) = 66\% <$

$minconf$, donc elle n'est pas fiable, dans ce cas il faut procéder à l'étape de complétion de l'antécédent afin de construire une règle acceptable et minimale.

Nous complétons donc $\#1$ avec $\#2$ (le premier id dans $P_{roj}ID_{deb}(\#1)$). $support(< \#1, \#2, \#3 >) = 15.3\% > minsupp$, donc l'épisode $< \#1, \#2, \#3 >$ est fréquent. $confiance(\#1, \#2 \rightarrow \#3) = 100\% > minconf$. Donc, la règle $\#1, \#2 \rightarrow \#3$ est acceptable et donc minimale, et elle est ajoutée à LRE. Le préfixe $\#1$ qui a une conséquence fixée $\#3$ est encore étendu avec d'autres id dans sa liste $P_{roj}ID_{deb}(\#1)$ pour compléter l'antécédent afin d'extraire d'autres règles d'épisodes minimales. Selon l'étape de vérification d'apparition multiple d'un item dans un épisode, $\#3 \in P_{roj}ID_{deb}(\#1)$ ne peut pas être un complément candidat de l'antécédent de la règle $\#1 \rightarrow \#3$. Lorsque l'on complète cette règle avec les id $\#4, \#5, \#6$ de $P_{roj}ID_{deb}(\#1)$, les règles construites ne sont pas acceptables et donc pas minimales. Nous procédons à une autre itération d'identification d'une autre conséquence de $\#1$. Cette procédure est répétée pour chaque $id \in P_{roj}ID_{fin}(\#1$.

Algorithme 1 : Extraction de règles d'épisodes minimales

Données : S : séquence complexe, LFP : motifs fréquents locaux, ID : liste des id
Résultat : LRE : liste de règles d'épisodes minimales
Procédure *extraction de règles épisodes minimales* **est**
  pour chaque $id_i \in LFP$ **faire**
    Construire $P_{roj}ID_{fin}(id_i), P_{roj}ID_{deb}(id_i)$
    `conséquence(`id_i`)`

Procédure *conséquence*(id_i) ; `/* `id_i`: préfixe */`
est
  pour chaque $id_j \in P_{roj}ID_{fin}(id_i)$ **faire**
    si $id_i \bigcap id_j = \emptyset$; `/* vérifier l'apparition multiple */`
    alors
      si $< id_i, id_j >$ *est fréquent* **alors**
        si $id_i \rightarrow id_j$ *est fiable* **alors**
          Ajouter $id_i \rightarrow id_j$ à LRE
        sinon
          `antécédent(`id_i, id_j`)`

Procédure *antécédent*(id_i, id_j) ; `/* `id_i`: préfixe, `id_j`: conséquence */`
est
  pour chaque $id_s \in P_{roj}ID_{deb}(id_i)$ **faire**
    si $id_i \bigcap id_s \bigcap id_j = \emptyset$; `/* vérifier l'apparition multiple */`
    alors
      si $< id_i, id_s, id_j >$ *est fréquent* **alors**
        si $id_i, id_s \rightarrow id_j$ *est fiable* **alors**
          Ajouter $id_i, id_s \rightarrow id_j$ à LRE
        sinon
          `antécédent(`$< id_i, id_s >, id_j$`)`

6 Conclusion et perspectives

Dans ce travail préliminaire, nous proposons un algorithme de fouille de flux de données dans le but de prédire au plus tôt et de façon fiable, des événements. Notre objectif à court terme est de fouiller des messages issus des réseaux sociaux et de prédire l'apparition d'informations. Nous avons mis en évidence que l'extraction de règles d'épisodes est adaptée à nos données, qui se présentent sous la forme d'une séquence complexe ordonnée selon le temps. Pour atteindre nos objectifs, nous avons déterminé plusieurs caractéristiques des règles d'épisodes à extraire : elles doivent être sans répétition, fréquentes, fiables, ayant l'antécédent le plus petit possible et la conséquence la plus éloignée temporellement de l'antécédent. Nous avons proposé un algorithme inspiré d'un autre de l'état de l'art "EMMA". Notre algorithme est bien plus rapide qu'EMMA, et donc que plusieurs autres algorithmes parce que les caractéristiques souhaitées permettent de réduire le nombre d'itérations nécessaires.

Nous envisageons de tester notre algorithme sur des données réelles de réseaux sociaux afin de prédire au plus tôt d'événements lointains. L'algorithme, tel que nous l'avons défini, ne permet d'extraire que des conséquences de taille 1 (1-uplet épisodes). Il est évident qu'une conséquence plus longue sera plus porteuse d'informations. Une de nos perspectives est donc d'adapter cet algorithme de façon à ce qu'il puisse extraire des conséquences plus longues. Cela va permettre d'anticiper des conséquences contenant plus d'informations sur les futurs événements. Nous envisageons également de modifier la première étape d'extraction des motifs fréquents locaux afin de ne pas extraire tous les motifs et d'éliminer les motifs longs, afin de favoriser l'extraction de règles minimales et de diminuer la complexité de l'algorithme. En effet, certains de ces motifs longs sont éliminés lors de la phase d'extraction de règles d'épisodes minimales de notre algorithme.

Références

Agrawal, R., R. Srikant, et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, Volume 1215, pp. 487–499.

Basak, J., A. Sudarshan, D. Trivedi, et M. Santhanam (2004). Weather data mining using independent component analysis. *The Journal of Machine Learning Research 5*, 239–253.

Bathoorn, R., M. Welten, M. Richardson, A. Siebes, et F. J. Verbeek (2010). Frequent episode mining to support pattern analysis in developmental biology. In *Pattern Recognition in Bioinformatics*, pp. 253–263. Springer.

Daurel, T. (2003). *Mining confident minimal rules with fixed-consequents*. Ph. D. thesis, Institut National des Sciences Appliquées de Lyon.

Gan, M. et H. Dai (2011). Fast mining of non-derivable episode rules in complex sequences. In *Modeling Decision for Artificial Intelligence*, pp. 67–78. Springer.

Gan, M. et H. Dai (2012). Mining condensed sets of frequent episodes with more accurate frequencies from complex sequences. *International Journal of Innovative Computing, Information and Control 8*(1), 453–470.

Han, J., J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, et M. Hsu (2001). Prefix*S*pan : Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of*

the 17th international conference on data engineering, pp. 215–224.

Han, J., J. Pei, et Y. Yin (2000). Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, Volume 29, pp. 1–12. ACM.

Huang, K.-Y. et C.-H. Chang (2008). Efficient mining of frequent episodes from complex sequences. *Information Systems 33*(1), 96–114.

Iwanuma, K., R. Ishihara, Y. Takano, et H. Nabeshima (2005). Extracting frequent subsequences from a single long data sequence a novel anti-monotonic measure and a simple on-line algorithm. In *Data Mining, Fifth IEEE International Conference on*, pp. 8–pp. IEEE.

Laxman, S. et P. S. Sastry (2006). A survey of temporal data mining. *Sadhana 31*(2), 173–198.

Mannila, H., H. Toivonen, et A. I. Verkamo (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery 1*(3), 259–289.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Discovering frequent closed itemsets for association rules. In *Database Theory-ICDT'99*, pp. 398–416. Springer.

Priya, V. et A. Vadivel (2012). User behaviour pattern mining from weblog. *International Journal of Data Warehousing and Mining (IJDWM) 8*(2), 1–22.

Rahal, I., D. Ren, W. Wu, et W. Perrizo (2004). Mining confident minimal rules with fixed-consequents. In *Tools with Artificial Intelligence, ICTAI*, pp. 6–13. IEEE.

Tatti, N. et B. Cule (2012). Mining closed strict episodes. *Data Mining and Knowledge Discovery 25*(1), 34–66.

Summary

The number of daily posted messages on social networks and blogs is huge, providing a valuable source of information. Temporal data mining is adapted to the exploration of this dataset. The result of this mining can be used for a prediction purpose, to anticipate the emergence of information. Our goal in this article is to propose an algorithm with the characteristic of early and confident information prediction.

In this preliminary work, we focus on extracting minimal episode rules to be quickly identified in a data stream. These rules also have the characteristic of having a consequence temporally distant of the antecedent, to be able to anticipate the consequence as soon as possible, and also have a high confidence.