Modélisation de la dynamique de phénomènes spatio-temporels par des séquences de motifs

Loïc Mabit*, Nazha Selmaoui-Folcher* Frédéric Flouvat*

*PPME - Université de la Nouvelle-Calédonie, BP R4, Nouméa, Nouvelle-Calédonie
{loic.mabit,frederic.flouvat,nazha.selmaoui}@univ-nc.nc

Résumé. Dans ce papier, nous proposons un nouveau cadre théorique permettant de modéliser la dynamique de phénomènes spatio-temporels. Nous définissons le concept de séquences spatio-temporelles de motifs afin de capturer les interactions entre des ensembles de propriétés et un phénomène à observer. Un algorithme incrémental est proposé pour extraire des séquences spatio-temporelles de motifs sous contraintes, et une nouvelle structure de données est mise en place afin d'améliorer ses performances. Un prototype a été développé et testé sur des données réelles.

1 Introduction

Dans un grand nombre d'applications, la compréhension et la modélisation de la dynamique spatio-temporelle est un problème majeur. Yuan (2008) utilise le terme "dynamique" pour caractériser les forces qui affectent dans l'espace et le temps le comportement d'un système. Par exemple, une épidémie de dengue est caractérisée par un ensemble de facteurs en interaction et causant la propagation de la maladie dans l'espace et le temps. Lorsque la dengue est déclarée dans un quartier, la question est de savoir comment, et en fonction de quels facteurs, elle va se propager dans les autres quartiers. Même si cette propagation semble dépendante de l'environnement direct des zones (points d'eau, mangroves à proximité, etc.) ainsi que d'un ensemble de circonstances évoluant dans le temps (humidité, chaleur, précipitation, etc.), la dynamique globale de propagation est loin d'être maîtrisée si on considère toutes les interactions possibles entre facteurs. D'autres applications ont le même type de problèmes. Par exemple, le phénomène d'érosion des sols est aussi influencé par un ensemble de facteurs environnementaux (type de sol, type de végétation, etc.) et de circonstances temporelles (pluie forte, cyclone, etc.). L'étude des dynamiques de propagation de l'érosion est une problématique majeure pour une gestion efficace des risques naturels et le développement durable de beaucoup de territoires.

Face à ces questions, les experts ont besoins de méthodes formelles leur permettant de valider ou de découvrir les dynamiques de propagation de ces phénomènes. Les méthodes de fouille de données spatio-temporelles visent à apporter des solutions pour mieux comprendre et décrire ces phénomènes complexes. Le but est de chercher alors des relations entre variables et événements sans hypothèse a priori. Parmi les méthodes de fouille de données, l'extraction de motifs caractérisant l'évolution d'un phénomène dans l'espace et dans le temps reste un

problème ouvert en raison de la complexité des systèmes étudiés et de la complexité des données disponibles. En effet, nous sommes en présence d'une masse de données hétérogènes et complexes (données géographiques, données météorologiques, images satellites, etc.).

Comme le montre Yao (2003), pendant longtemps, les travaux de recherche se sont focalisés sur une seule dimension (spatiale ou temporelle) sans prendre en compte l'autre. Ces dernières années, de plus en plus de travaux en fouille de données spatio-temporelles analysent conjointement l'aspect spatial et temporel. Généralement, ces travaux traitent deux types de problèmes : les trajectoires d'objets en mouvement (Yao, 2003; Cao et al., 2005; Yuan, 2008; Du et al., 2009; Monreale et al., 2009; Verhein, 2009) et les séquences d'événements (Wynne Hsu, 2009a,b; Celik et al., 2008, 2006; Mohan et al., 2010).
Dans Cao et al. (2005); Yuan (2008); Du et al. (2009), les auteurs caractérisent les trajectoires d'objets en mouvement par des séquences de tuples (l, t), où l est la localisation de l'objet au temps t. Dans ce type de problème, les trajectoires de chaque objet sont explicitement décrites dans les données. Les auteurs cherchent alors à extraire les trajectoires les plus fréquentes dans la base de trajectoires. Notre problématique est plus complexe car la dynamique des objets (i.e. les trajectoires) n'est pas explicitement stockée dans la bases de données. De plus, les objets d'étude ne sont pas clairement identifiés. Nous avons uniquement une base de données géographique stratifiée en couches temporelles, avec au niveau de chaque couche des variables d'environnements et des variables temporelles localisées dans des zones.
Dans le cadre de l'extraction des séquences d'événements, Mohan et al. (2010) définissent la notion de motifs spatio-temporels en cascades, et les appliquent à une base de données de crimes. Leurs motifs se présentent sous la forme de graphes d'événements et leur algorithme d'extraction est basé sur le principe "générer-tester" (cf algorithme Apriori, Agrawal et Srikant (1994)). Leurs travaux s'appuient sur le concept de co-localisation spatio-temporelle (Celik et al., 2006, 2008; Wang et al., 2009; Qian et al., 2009; Lin et Lim, 2009). Une co-localisation spatio-temporelle est un *ensemble* d'événements dont les instances sont voisines dans l'espace et dans le temps. Ces motifs sont associés à une nouvelle mesure d'intérêt anti-monotone (l'index de participation), afin de pouvoir exploiter une stratégie de type $Apriori$. Les motifs étudiés dans ces travaux permettent de représenter l'évolution d'événements dans l'espace et le temps, mais ils ne permettent pas de prendre en compte l'environnement de ces évènements (p.ex. l'occupation du sol pour l'érosion). De plus, les mesures d'intérêt proposées sont difficiles à interpréter pour les experts. L'obtention de la propriété d'anti-monotonie s'est faite au détriment de l'interprétabilité des mesures. Pour finir, les algorithmes d'extraction proposés ne sont pas incrémentaux, i.e. l'ajout de données pour une nouvelle date nécessite de recalculer l'ensemble des solutions. Notre démarche est différente de ces travaux sur ces différents points. Nous avons défini un nouveau concept de séquences spatio-temporelles de motifs (i.e. des séquences d'ensembles), appelée STSP (*Spatio-Temporal Sequence of Patterns*), afin de capturer les interactions entre les différents facteurs. Nous utilisons une mesure d'intérêt facile à interpréter (proche du support des ensembles fréquents) mais qui n'est pas anti-monotone. Nous proposons un algorithme incrémental permettant l'extraction des séquences de motifs intéressantes grâce à un ensemble de contraintes (non nécessairement monotones). Une nouvelle structure de données, appelée STP-tree, est aussi définie afin de rendre cet algorithme performant.

La section 2 introduit le concept de séquence spatio-temporelle de motifs (*STSP*). La section 3 présente notre approche incrémentale de générations des STSP et son optimisation grâce

au STP-Tree. La section 4 présente un algorithme d'extraction de STSP sous contraintes. Pour finir, nous présenterons les résultats expérimentaux préliminaires.

2 Concepts et définitions

Soient un ensemble de temps ordonné $T = \{t_1 < t_2 < ... < t_{|T|}\}$, un ensemble de zones géographiques (une carte de zones) géo-référencées $Z = \{z_1, z_2, ..., z_{|Z|}\}$, et un ensemble de caractéristiques booléennes $I = \{i_1, i_2, ..., i_{|I|}\}$ décrivant ces zones à un temps donné.
Nous noterons par Ω_t le triplet $(Z, I, \Re_B)$ où $\Re_B \subseteq Z\text{x}I$ est la relation binaire telle que $\Re_B(z, i) = 1$ si la caractéristique $i \in I$ est associée à la zone $z \in Z$ au temps t.
Ω_t est appelé une **couche temporelle** et $\Omega = \bigcup_{t \in T} \Omega_t$ est appelé une **base de données spatio-temporelles**. Une base de données spatio-temporelles est donc une suite de couches temporelles décrivant l'évolution d'un ensemble de zones géographiques dans le temps.
Par exemple, dans la figure 1, $T = \{t_1, t_2, t_3\}$, $Z = \{z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}\}$ *et* $I = \{i_1, i_2, i_3, i_4, i_5, i_6\}$. *(a),(b) et (c) représentent respectivement* Ω_{t_1}, Ω_{t_2}, Ω_{t_3}, *et constituent la base de données spatio-temporelles* Ω *aux temps* t_1, t_2 *et* t_3.

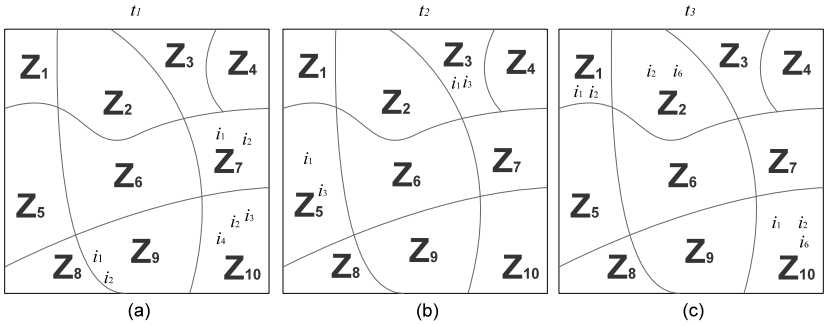

FIG. 1 – *Exemple de base de données spatio-temporelles* Ω

Soit $X \subseteq I$ un ensemble de caractéristiques booléennes appelé **motif** (*itemset*). Le **support** de X dans Ω_t, noté $support_{\Omega_t}(X)$, est égal au nombre de zones z associés à l'ensemble des caractéristiques X au temps $t \in T$, i.e. $support_{\Omega_t}(X) = |\{z \in Z / \Re_B(z, i) = 1 \; \forall i \in X\}|$.
Soit α un seuil de support minimum, X est appelé **motif fréquent** (FI) si $support_{\Omega_t}(X) \geq \alpha$. On note $FI(\Omega_t)$ l'ensemble de tous les motifs fréquents dans Ω_t, i.e. $FI(\Omega_t) = \{X \subseteq I / support_{\Omega_t}(X) \geq \alpha\}$. Pour simplifier les notations, un motif $\{i_k, ..., i_l\}$ sera noté $i_k...i_l$.

Soit une zone $z \in Z$, le couple (X, z), appelé **motif fréquent géoréférencé**, représente l'apparition du motif fréquent X dans la zone z. On notera $GFI_z(\Omega_t)$ l'ensemble des motifs fréquents caractérisant (apparaissant dans) la zone z au temps $t \in T$, i.e. $GFI_z(\Omega_t) = \{(X, z) \subseteq (I, Z) / X \in FI(\Omega_t) \; et \; \forall i \in X, \Re_B(z, i) = 1\}$. *Par exemple, dans la figure 1(a),* $GFI_{z_7}(\Omega_{t_1}) = \{i_1, i_2, i_1 i_2\}$, *le motif fréquent* $i_1 i_2$ *caractérise la zone* $z_7 \in \Omega_{t_1}$ *(pour* $\alpha = 2$*).*

Notons $\Re_n \subseteq Z\text{x}Z$ une **relation de voisinage** telle que $\Re_n(z_a, z_b) = 1$ si z_a et z_b sont voisins dans Z . *Par exemple, dans la figure 1, si* $\Re_n$ *est la relation d'adjacence, nous avons* $\Re_n(z_5, z_8) = 1$ *(car* z_5 *et* z_8 *sont des zones adjacentes) alors que* $\Re_n(z_1, z_3) = 0$.

Une **séquence spatio-temporelle de motifs** (STSP) $s = \langle s_1, s_2, ..., s_m \rangle_{\Re_n}$ est une liste ordonnée $(s_i)_{i=1..m}$ de motifs fréquents tels que s_k et s_{k+1} caractérisent deux zones voisines dans deux couches temporelles consécutives. Le nombre de motifs dans la séquence est appelé la taille de la STSP. Nous noterons $S_{\Re_n}$ l'ensemble de toutes les STSP de Ω pour la relation $\Re_n$. $S_{\Re_n}$ est un multi-ensemble car une même séquence peut apparaître plusieurs fois.
Par exemple, dans Ω_{t_1} (figure 1(a)), le motif fréquent $i_1 i_2$ caractérise les zones z_7 et z_8. Dans Ω_{t_2} (figure 1(b)), $i_1 i_3$ caractérise z_3 et z_5. Or z_7 est une zone voisine ($\Re_n$ relation d'adjacence) de z_3, $\langle i_1 i_2, i_1 i_3 \rangle_{\Re_n}$ est donc une STSP de taille 2 (voir figure 2(d)). De même, z_8 est une zone voisine de z_5, $\langle i_1 i_2, i_1 i_3 \rangle_{\Re_n}$ est donc une autre STSP (voir figure 2(d)). Ainsi, $\langle i_1 i_2, i_1 i_3 \rangle_{\Re_n}$ apparait deux fois dans $S_{\Re_n}$. Au temps suivant (Ω_{t_3}, figure 1(c)), le motif fréquent $i_2 i_6$ caractérise la zone z_2. Or z_3 est une zone voisine de z_2, la séquence STSP $\langle i_1 i_2, i_1 i_3 \rangle_{\Re_n}$ générée entre le temps t_1 et t_2 peut être prolongée par $i_2 i_6$ pour former la STSP $\langle i_1 i_2, i_1 i_3, i_2 i_6 \rangle_{\Re_n}$ de taille 3 (figure 2(e)).

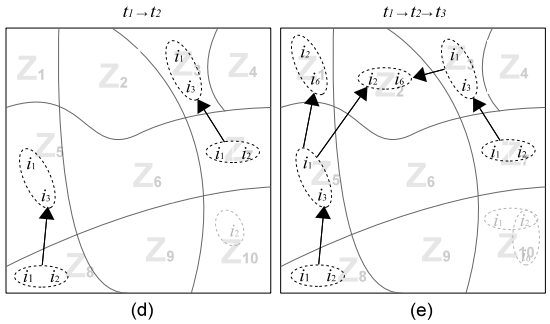

FIG. 2 – *Exemple de génération de STSP dans une base de données spatio-temporelles*

Une STSP $s' = \langle s'_1, s'_2, ...s'_n \rangle_{\Re_n}$ est une **sous-STSP** (sous-séquence) d'une STSP $s = \langle s_1, s_2, ..., s_m \rangle_{\Re_n}$ (et s une **sur-STSP** de s'), s'il existe des entiers $1 \leq j_1 < j_2 < ... < j_n \leq m$ tel que $s'_1 \subseteq s_{j_1}$, $s'_2 \subseteq s_{j_2}$,...,$s'_n \subseteq s_{j_n}$.

Soit $S_{\Re_n}$ un multi-ensemble de STSP. Le **support absolu de la STSP** $s' = \langle s'_1, s'_2, ...s'_n \rangle_{\Re_n}$ dans $S_{\Re_n}$, noté $support_{abs}(s', S_{\Re_n})$, est le nombre de chemins $\langle (z_l, t_k), (z_{l+1}, t_{k+1}), ..., (z_{l+n-1}, t_{k+n-1}) \rangle_{\Re_n}$ empruntés par s' dans Ω, i.e. $support_{abs}(s', S_{\Re_n}) = |\{\langle (z_l, t_k), (z_{l+1}, t_{k+1}), ..., (z_{l+n-1}, t_{k+n-1}) \rangle_{\Re_n} / \{t_k, t_{k+1}, ..., t_{k+n-1}\} \subseteq T, \forall z_a \in \{z_l, z_{l+1}, ...z_{l+n-1}\}, z_a \in Z, \Re_n(z_a, z_{a+1}) = 1\}$.
Notons que, contrairement au support d'un motif, le support d'une STSP n'est pas nécessairement décroissant (ni croissant). *Par exemple, dans la figure 2 (e), nous avons $support_{abs}(\langle i_1 i_2, i_1 i_3, i_2 i_6 \rangle_{\Re_n}, S_{\Re_n}) = 3$, alors que $support_{abs}(\langle i_1 i_2, i_1 i_3 \rangle_{\Re_n}, S_{\Re_n}) = 2$.*

Soit β un seuil minimum de support, une STSP s' est **fréquente** dans $S_{\Re_n}$ si $support_{abs}(s', S_{\Re_n}) \geq \beta$. Notons que le seuil α peut être assimilé à un seuil de support spatial et que β peut être assimilé à un seuil de support temporel.

3 Construction incrémentale des séquences spatio-temporelles de motifs

Cette section présente le principe de notre approche pour générer incrémentallement les STSP d'une base de données spatio-temporelles, ainsi que la structure de données développée pour optimiser cette génération.

3.1 Principe de l'approche

Le principe de notre algorithme (algorithme 1) est basé sur la construction bout à bout (incrémentallement) des séquences de $S_{\Re_n}$ (ensemble de toutes les séquences STSP apparaissant dans Ω) en suivant les étapes suivantes :

Etape 1 : Caractérisation des zones aux temps t et $t+1$ par les motifs fréquents. Cette étape consiste à extraire les motifs fréquents (avec un seuil minimum de support α) aux temps t et $t+1$ **(Algorithme 1, lignes 3-5)**. Chaque motif est ensuite associé aux zones dans lequel il apparait **(Algorithme 1, lignes 7-8)**. On obtient ainsi les motifs fréquents géo-référencés (GFI).

Etape 2 : Modélisation de la dynamique entre t et $t+1$. Cette étape génère toutes les séquences de taille 2 (2-STSP) apparaissant entre t et $t+1$. Nous rappelons qu'une séquence de taille 2 notée par $\langle s'_{from}, s'_{to}\rangle_{\Re_n}$ est telle que s'_{from} et s'_{to} sont respectivement des motifs fréquents de Ω_t et Ω_{t+1} géo-référencés dans des zones voisines par rapport à la relation $\Re_n$ **(Algorithme 1, ligne 9)**. En d'autres termes, étant donné deux zones voisines z_{from} et z_{to} (i.e. $\Re_n(z_{from}, z_{to}) = 1$), l'algorithme effectue un produit cartésien entre $GFI_{z_{from}}(\Omega_t)$ et $GFI_{z_{to}}(\Omega_{t+1})$.

Etape 3 : Extension des STSP Cette étape étend les STSP existantes et met à jour l'ensemble des STSP, noté $S_{\Re_n}$. Plus précisément, étant donné une séquence de taille 2 $\langle s'_{from}, s'_{to}\rangle_{\Re_n}$, chaque STSP $s = \langle s_1, ..., s_{last}\rangle_{\Re_n}$ de $S_{\Re_n}$ est prolongée par s'_{to} si s_{last} = s'_{from} au même temps (Ω_t) et dans la même zone (z_{from}) **(Algorithme 1, lignes 11 à 13)**.

S'il n y a pas d'extension possible, alors $\langle s'_{from}, s'_{to}\rangle_{\Re_n}$ est insérée dans $S_{\Re_n}$ comme une nouvelle séquence de taille 2 **(lignes 10 à 18)**.

Cette construction incrémentale des STSP est très importante car les bases de données spatio-temporelles peuvent être mises à jour régulièrement avec de nouvelles couches temporelles. Notre approche évite ainsi une re-construction de toutes les STSP à chaque nouvelle couche ajoutée. L'algorithme 1 présente le principe général de la construction des séquences. Cependant, le nombre des STSP peut être très grand (jusqu'à plusieurs millions), rendant leur construction impossible. Face à ce problème, il est donc nécessaire d'optimiser cette algorithme, et plus particulièrement le stockage et l'extension des STSP.

3.2 STP-tree : une structure de données optimisée pour les STSP

Dans cette section, nous proposons une nouvelle structure adaptée au traitement des STSP, et l'exploitons dans l'étape 3 de l'algorithme 1. Cette structure de données a été appelée **STP-tree** (*Spatio-Temporal Pattern tree*). Pour illustrer l'utilisation de cette structure de données,

Algorithm 1 Algorithme incrémental de génération des STSP

$S_{\Re_n} = \emptyset$
$t = 1$ // entier représentant les différents temps
$FI_t = mine\_freq\_itemsets(\Omega, t, \alpha)$
while $t < |T|$ **do**
  $FI_{t+1} = mine\_freq\_itemsets(\Omega, t+1, \alpha)$
  for each z_{from},$z_{to} \in Z$ tel que $\Re_n(z_{from}, z_{to}) =$ **true do**
    $GFI_{t,z_{from}} = georeference\_freq\_itemsets(FI_{t,z_{from}})$
    $GFI_{t+1,z_{to}} = georeference\_freq\_itemsets(FI_{t+1,z_{to}})$
    2-STSP= $generate\_size2\_STSP(GFI_{t,z_{from}}$,$GFI_{t+1,z_{to}})$
    for each STSP de taille 2 $< s'_{from}, s'_{to} >_{\Re_n} \in$ 2-STSP **do**
      for each STSP $< s_1, ..., s_{last} >_{\Re_n} \in S_{\Re_n}$ tel que $s_{last} = s'_{from}$ avec s_{last} dans z_{from} au temps t **do**
        Etendre $< s_1, ..., s_{last} >_{\Re_n}$ avec s'_{to}
      end for
      if $S_{\Re_n} = \emptyset$ ou si aucune séquence $s \in S_{\Re_n}$ n'a été étendue **then**
        Ajouter $< s'_{from}, s'_{to} >_{\Re_n}$ à $S_{\Re_n}$
      end if
    end for
  end for
  $t = t + 1$
end while
return $S_{\Re_n}$

nous considérons un nouvel exemple (figure 3). Les deux figures de gauche représentent les STSP obtenues lors des transitions $t1 - t2$ et $t2 - t3$. La figure à droite représente le STP-tree obtenu à la fin du temps $t3$. Notons que, dans cet exemple, les motifs fréquents on déjà été extraits et géoréférencés. *Pour simplifier les notations, ces motifs ont été notés $P1$, $P2$, $P3$,... (Pi représente donc un ensemble de caractéristiques).*

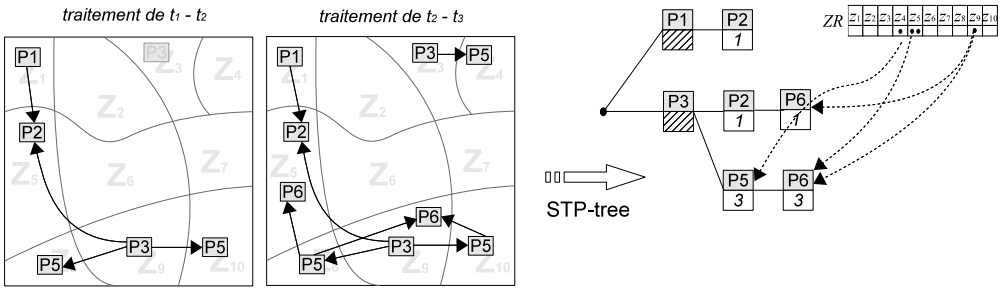

FIG. 3 – *Exemple de STSP et de STP-tree*

Comme le montre l'exemple, un STP-tree est un arbre préfixé associé à un tableau de zones ZR (*Zonal Referencing*). L'arbre est utilisé pour stocker les STSP et leur support. Chaque noeud n est composé d'un motif fréquent et d'un compteur. Le lien entre les noeuds représente le lien de précédence temporelle et de proximité spatial entre les motifs fréquents de la séquence. Le compteur représente le support de la STSP commençant à la racine de l'arbre et finissant au noeud n. *Par exemple, la STSP $< P3, P5 >_{\Re_n}$ apparaît trois fois (support de 3) et la STSP $< P3, P5, P6 >_{\Re_n}$ apparaît deux fois dans la base de données spatio-temporelles.* Le tableau ZR fait le lien entre les zones et les dernière STSP mises à jour. Plus précisément,

chaque cellule du tableau est associée à une zone géographique et contient une liste de liens vers des noeuds de l'arbre. Chacun de ces liens pointe sur le dernier motif d'une STSP "extensible", i.e. une STSP qui est toujours en construction (son dernier motif a été ajouté lors de l'itération précédente dans l'algorithme 1). Autrement dit, toutes les STSP extensibles à partir de la zone z_{from} sont référencées dans la cellule $ZR[z_{from}]$. *Par exemple, dans la figure 3,* ZR *montre qu'au temps* $t4 : < P3, P5 >_{\Re_n}$ *peut être étendue à partir de la zone* z_4*,* $< P3, P5, P7 >_{\Re_n}$ *peut être étendue à partir de la zone* z_5*, et* $< P3, P5, P6 >_{\Re_n}$ *peut être étendue à partir des zones* z_5 *et* z_9*.*

Le STP-tree a l'avantage de compresser l'ensemble des STSP stockés en factorisant les préfixes communs entre les STSP. *Par exemple,* $< P3, P5 >$ *n'est pas dupliquée bien qu'il y ait deux STSP commençant par cette sous-séquence (*$< P3, P5, P6 >_{S_{\Re_n}}$ *et* $< P3, P5, P7 >_{S_{\Re_n}}$*).* De plus, les compteurs maintenus au niveau des noeuds permettent de conserver le support associé à chaque STSP. A titre d'exemple, l'utilisation d'une liste classique pour stocker les STSP aurait augmenté les traitements à effectuer. *Par exemple, dans la figure 2 (d), la sous-séquence* $< (i_1, i_2), (i_1, i_3) >_{\Re_n}$ *aurait pu être comptabilisée 3 fois (car elle apparaît dans 3 sur-STSP stockée dans* $S_{\Re_n}$*), alors qu'en réalité elle n'apparaît que deux fois dans les données.* Pour finir, comme nous allons le montrer dans la suite, cette structure de données va permettre une mise à jour efficace des STSP (étape 3, lignes 11 à 16 de l'algorithme 1).

Pour rappel, pour chaque 2-STSP $< s'_{from}, s'_{to} >_{\Re_n}$ construite à l'étape 2, l'étape 3 de le génération (ligne 11) consiste d'abord à trouver les STSP "extensibles" à partir de $< s'_{from}, s'_{to} >_{\Re_n}$ parmi toutes celles générées. Comme nous l'avons vu précédemment, le STP-tree permet de les avoir directement. En fait, un simple parcours de la liste des liens contenus dans les cases Z_{from} du tableau ZR permet de les trouver, où Z_{from} est l'ensemble des zones utilisées pour construire $< s'_{from}, s'_{to} >_{\Re_n}$ à l'étape 2 et associées au motif s'_{from}. Ensuite, deux cas sont possibles :

1. Les cases ne contiennent aucun lien vers un motif s'_{from} (lignes 14-15). Autrement dit, il n'existe aucune STSP extensible à partir des zones de $< s'_{from}, s'_{to} >_{\Re_n}$. Dans ce cas, une nouvelle STSP $< s'_{from}, s'_{to} >_{\Re_n}$ est ajoutée dans le STP-tree. Cette insertion dans le STP-tree consiste à mettre à jour un compteur si $< s'_{from}, s'_{to} >_{\Re_n}$ est déjà dans le STP-tree ou à créer de nouveaux noeuds sinon. La figure 4 illustre ces deux sous-cas sur l'exemple de la figure 3.

2. Les cases contiennent des liens vers un motif s'_{from} (lignes 11-13). Autrement dit, il existe une STSP extensible $< s_1, s_2..., s'_{from} >_{\Re_n}$ à partir des zones de $< s'_{from}, s'_{to} >_{\Re_n}$. Dans ce cas, la STSP extensible est étendue avec le motif s'_{to}. Cette extension consiste à ajouter un nouveau noeud s'_{to} lié au dernier noeud de la STSP si celui-ci n'existe pas déjà, ou à mettre à jour le compteur du dernier noeud sinon. La figure 5 illustre ces deux sous-cas. Notons que lorsque $t_{i+1} > t_2$, il est nécessaire de conserver le tableau ZR de l'itération précédente pendant le traitement de $t_i - t_{i+1}$ afin de pouvoir toujours trouver les STSP extensible au temps t_i (à la fin du traitement de $t_i - t_{i+1}$, $ZR(t_i)$ pourra être supprimé). La figure 5 illustre ces deux sous-cas sur l'exemple de la figure 3.

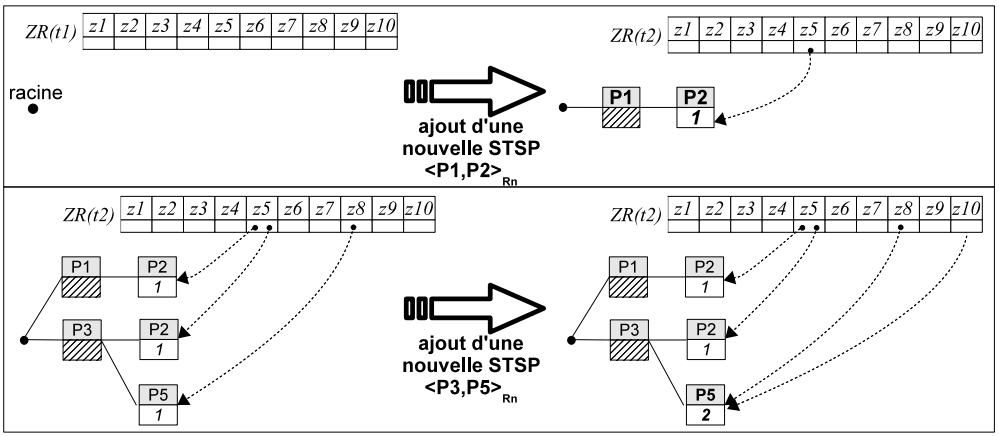

FIG. 4 – *Exemple d'insertions de nouvelles STSP dans le STP-tree entre t1-t2*

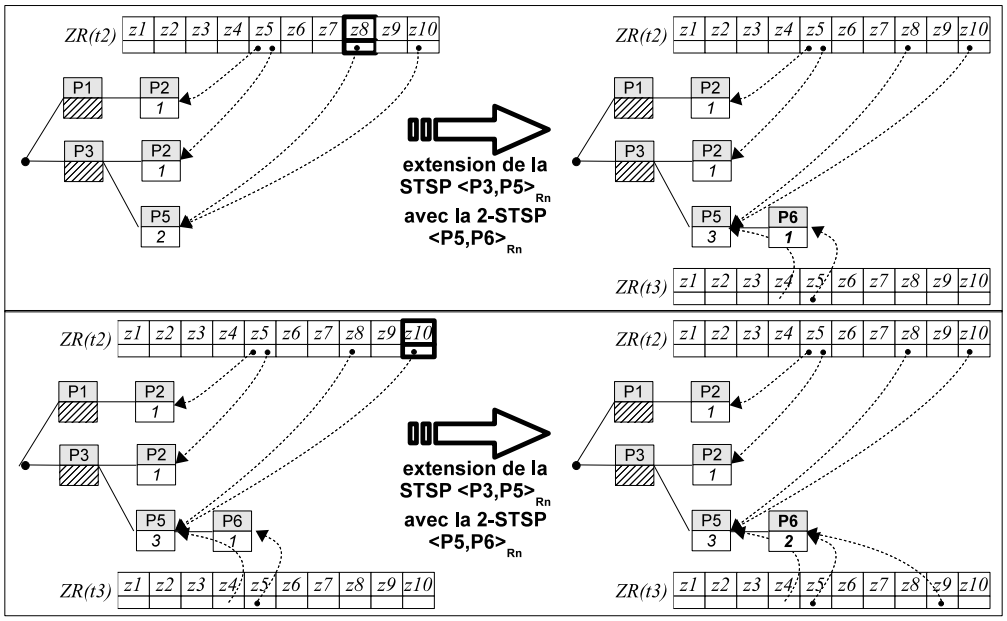

FIG. 5 – *Exemple d'extensions d'une STSP dans le STP-tree entre t2-t3*

4 Extraction de STSP sous-contraintes

4.1 Contraintes sur les sous-séquences

La définition de STSP donnée dans la section 2 est générale. Extraire toutes les sous-séquences reste un problème combinatoire. Toutefois, suivant les applications, toutes les sous-

séquences ne sont pas nécessairement utiles. Nous introduisons dans la suite de cette sous-section plusieurs contraintes permettant d'extraire des sous-séquences intéressantes pour nos applications.

Contrainte 1 : Seules sont extraites les sous-séquences $s' =< s'_1, s'_2, ...s'_n >_{\Re_n}$ de $s =< s_1, s_2, ..., s_m >_{\Re_n}$ tel qu'il existe des entiers $1 \leq j_1 < j_2 < ... < j_n \leq m$ avec $s'_1 = s_{j_1}$, $s'_2 = s_{j_2}$,...,$s'_n = s_{j_n}$. Cette contrainte considère les motifs comme étant atomique, i.e. les sous-motifs ne sont pas étudiés. En effet, dans certaines applications, un motif représente une interaction de caractéristiques qui n'auraient pas nécessairement eu lieu si une des caractéristiques avait été enlevée.

Contrainte 2 : Seules sont extraites les sous-séquences $s' =< s_1, s_2, ...s_i >_{\Re_n}$ de $s =< s_1, s_2, ..., s_m >_{\Re_n}$ avec $1 < i < m$. Les sous-séquences doivent ainsi avoir le même préfixe que la STSP originelle. Cette contrainte représente le fait que le motif s_i est apparu à cause de l'interaction de *tous* les autres motifs qui le précède dans la séquence. Par exemple, si on considère la séquence $< bulldozers, construction, erosion >_{\Re_n}$, $erosion$ n'aurait peut être pas eu lieu si $construction$ avait été absent (i.e. c'est l'association dans le temps de bulldozers et d'une construction qui a entraîné l'érosion).

Contrainte 3 : Cette contrainte fixe à 1 l'intervalle de temps entre deux motifs consécutifs de la séquence. On se focalise ainsi sur les dynamiques de propagation entre des temps consécutifs.

Contrainte 4 : Une sous-séquence est fréquente si son support relatif est supérieure ou égal au seuil de support $\beta \in [0, 1]$ donné. Le support relatif d'une STSP est égal au support absolu de la STSP (cf section 2) divisé par le nombre total de STSP générées.

4.2 Extraction de sous-séquences intéressantes

Nous avons introduit dans la section 3.2 la structure de données STP-tree qui permet de construire et de stocker efficacement les STSP. Dans cette partie, nous allons exploiter le STP-tree obtenu à la fin de la génération des STSP pour extraire efficacement des sous-séquences intéressantes (i.e. respectant les contraintes définies ci-dessus).

Algorithm 2 Extraction des STSP intéressantes à partir du STP-tree

1: $ISTSP = \emptyset$
2: $nb\_STSP = \sum_{node \in STP-tree} node.counter$
3: **for each** node that is not a child of the root node **do**
4: **if** $\frac{node.counter}{nb\_STSP} \geq \beta$ **then**
5: s = generate STSP starting from root and ending at the node
6: Add s to $ISTSP$
7: **end if**
8: **end for**
9: **return** $ISTSP$

L'algorithme 2 d'extraction prend en entrée le STP-tree généré à la fin de l'algorithme 1 ainsi qu'un seuil minimum de support relatif $\beta \in [0, 1]$, et retourne en sortie l'ensemble des sous-STSP intéressantes (noté $ISTSP$). Le principe de l'algorithme est le suivant :

1. calculer le nombre total de STSP dans le STP-tree (ligne 2)

2. parcourir tous les noeuds du STP-tree (ligne 3) et pour chacun
   – vérifier la contrainte de seuil minimum β à partir du support absolu stocker dans le noeud (ligne 4)
   – si la contrainte est vérifiée, générer la STSP correspondante simplement en parcourant l'arbre, du noeud en cours jusqu'à la racine (lignes 5-6)

Comme nous venons de le constater, l'extraction des STSP intéressantes est simple grâce au STP-tree. Sa complexité est en O($|S_{\Re_n}|$). En effet, cette structure de données permet de générer rapidement les STSP vérifiant les contraintes 1, 2 et 3 avec leur support absolu. Pour obtenir leur support relatif (et ainsi appliquer la contrainte 4), il suffit de calculer le nombre total de STSP dans l'arbre. En pratique, ce nombre est déjà calculé pendant la phase de génération (son calcul ne requiert donc aucun traitement supplémentaire).

5 Résultats expérimentaux préliminaires

Notre proposition a été implémentée dans un prototype en C++. Notons que pour améliorer la pertinence des motifs et diminuer leur nombre, ce prototype extrait les motifs fermés fréquents pour caractériser les zones à chaque date, et non pas tous les motifs fréquents. Pour faire cela, nous avons intégré l'implémentation de l'algorithme LCM de Uno et al. (2004).

Des expérimentations ont été réalisées à partir d'une bases de données spatio-temporelles associée à une surface de 12 km^2 (divisée en 42 zones). Ce jeu de données est constitué de 5 couches temporelles, chacune composée d'informations sur la nature du sol, l'occupation du sol, le type de végétation et un indice de végétation (NDVI). Au total, ce jeu de données contient 25 caractéristiques. La suite de cette section présente les résultats préliminaires de ces expérimentations.

Dans un premier temps, nous avons étudié le temps d'exécution de l'algorithme et plus particulièrement l'impact des STP-tree sur les performances (expérimentations réalisées avec Linux Ubuntu et un Intel Core 2 duo 2.66 Ghz). Nous avons notamment comparé deux variantes de notre approche : la première utilise une liste pour stocker les STSP et la deuxième utilise un STP-tree. Comme le montre le tableau 1, l'utilisation des STP-tree est beaucoup plus efficace que l'approche naïve consistant à utiliser une liste. Cette différence est d'autant plus importante que les expérimentations réalisées avec les listes ont été faite sur un sous-ensemble du jeu de données initial limité à 9 zones (au lieu de 42 zones pour les expérimentations avec les STP-tree).

α	**Utilisation d'une liste (9 zones)**	**Utilisation d'un STP-tree (42 zones)**
0.9	1 sec	1 sec
0.8	4 sec	1 sec
0.7	19 sec	2 sec
0.6	**332 sec**	**9 sec**
0.5	*process killed by OS*	30 sec
0.4	*process killed by OS*	39 sec
0.3	*process killed by OS*	135 sec
0.2	*process killed by OS*	177 sec
0.1	*process killed by OS*	283 sec

TAB. 1 – *Temps d'exécution de l'algorithme avec une structure de liste et avec les STP-tree*

Ensuite, nous avons étudié l'impact des seuils de support spatio-temporels α et β sur le nombre de STSP générées (tableau 2). Le tableau suivant montre que le nombre de STSP

générées augmente de manière exponentielle quand α diminue. En effet lorsque α diminue, le nombre de motifs fréquents augmente, et donc le nombre de STSP possibles augmente aussi. On peut même obtenir près de 80 millions de STSP avec un seuil de support α de 0.6 (seuil relatif). Notons néanmoins que, parmi ces 80 millions de STSP, seule 64 278 STSP différentes sont générées. Ce nombre important de STSP a pour conséquence d'entraîner une baisse du nombre STSP fréquentes (pour un même seuil de support β), qui est liée à la diminution du support absolu des STSP (qui dépend du nombre total de STSP). De manière plus classique, on constate aussi que le nombre de STSP fréquente augmente avec la diminution du seuil de support β.

α	**nb moyen de motifs fermés fréquents**	**nb de STSP**	**nb de STSP fréquentes**			
			$\beta = 0.01$	$\beta = 0.005$	$\beta = 0.001$	$\beta = 0.0001$
0.9	4	9350 (27 distinctes)	10	18	25	27
0.8	7	372313 (292 distinctes)	35	64	145	243
0.7	10	4640331 (2951 distinctes)	0	0	162	2061
0.6	19	82939975 (64278 distinctes)	0	0	0	1

TAB. 2 – *Impact des seuils de supports minimaux α et β sur le nombre de STSP*

Les résultats préliminaires mettent en avant des relations connues. Par exemple, une faible sensibilité à l'érosion est souvent observée près de forêts associées à un certain type de sol. Autre exemple, la savane arborée est souvent associée à une faible quantité de végétation lorsqu'elle est seule, alors qu'elle est souvent associée à une forte quantité de végétation lorsqu'elle est dans des zones avec des forêts.

6 Conclusion & Perspectives

Ce travail s'est focalisé sur la fouille de données spatio-temporelles. Dans ce domaine émergent, l'un des défis est d'extraire des motifs spatio-temporels dans des bases de données où la dynamique des objets n'est pas explicitement stockée (contrairement aux bases de trajectoires). Cette problématique est particulièrement importante dans un grand nombre d'applications telles que la prévention d'épidémies ou le suivi de l'impact humain sur l'environnement. Face à ce problème, nous avons proposé un nouveau concept : les séquences spatio-temporelles de motifs (STSP). Ce séquences ont pour objectifs de décrire la propagation de certains phénomènes dans le temps et dans l'espace. Afin d'extraire les séquences les plus intéressantes, nous avons développé un algorithme efficace s'appuyant sur une nouvelle structure de données, le STP-tree, et avons intégré des contraintes afin d'améliorer la pertinence des résultats. Les résultats expérimentaux préliminaires ont mis en avant l'intérêt de cette approche.

Les perspectives de ce travail sont nombreuses. L'une des premières est de faire une étude expérimentale plus poussée des STSP extraites, avec les experts, afin de mettre en avant de nouvelles contraintes permettant d'améliorer encore la pertinence des solutions. Par ailleurs, il serait particulièrement intéressant de relâcher certaines contraintes imposées dans ce papier, et plus particulièrement celles limitant les sous-STSP possibles (contraintes 1 et 2). De même, il pourrait être intéressant de relâcher la contrainte sur l'intervalle temporel entre les motifs de la séquence. La modification de ces contraintes impliquent de mettre en place une nouvelle stratégie pour extraire les STSP intéressantes à partir du STP-tree stockant l'ensemble des STSP générées. Une autre perspective importante est de développer une approche de visualisation adaptée aux STSP et aux besoins des experts. En effet, actuellement, les résultats sont présentés sous une forme textuelle ce qui rend leur analyse par les experts difficile.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, et C. Zaniolo (Eds.), *VLDB*, pp. 487–499. Morgan Kaufmann.

Cao, H., N. Mamoulis, et D. W. Cheung (2005). Mining frequent spatio-temporal sequential patterns. In *ICDM*, pp. 82–89. IEEE Computer Society.

Celik, M., S. Shekhar, J. P. Rogers, et J. A. Shine (2006). Sustained emerging spatio-temporal co-occurrence pattern mining : A summary of results. In *ICTAI*, pp. 106–115.

Celik, M., S. Shekhar, J. P. Rogers, et J. A. Shine (2008). Mixed-drove spatiotemporal co-occurrence pattern mining. *IEEE Trans. Knowl. Data Eng. 20*(10), 1322–1335.

Du, X., R. Jin, L. Ding, V. E. Lee, et J. H. T. Jr. (2009). Migration motif : a spatial - temporal pattern mining approach for financial markets. In *KDD*, pp. 1135–1144.

Lin, Z. et S. Lim (2009). Optimal candidate generation in spatial co-location mining. In S. Y. Shin et S. Ossowski (Eds.), *SAC*, pp. 1441–1445. ACM.

Mohan, P., S. Shekhar, J. A. Shine, et J. P. Rogers (2010). Cascading spatio-temporal pattern discovery : A summary of results. In *SDM*, pp. 327–338.

Monreale, A., F. Pinelli, R. Trasarti, et F. Giannotti (2009). Wherenext : a location predictor on trajectory pattern mining. In *KDD*, pp. 637–646.

Qian, F., Q. He, et J. He (2009). Mining spatial co-location patterns with dynamic neighborhood constraint. In *ECML/PKDD'09*, Volume 5782 of *LNCS*, pp. 238–253. Springer.

Uno, T., M. Kiyomi, et H. Arimura (2004). Lcm ver. 2 : Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI*.

Verhein, F. (2009). Mining complex spatio-temporal sequence patterns. In *SDM*, pp. 605–616.

Wang, L., L. Zhou, J. Lu, et J. Yip (2009). An order-clique-based approach for mining maximal co-locations. *Inf. Sci. 179*(19), 3370–3382.

Wynne Hsu, Mong Li Lee, J. W. (2009a). Mining generalized flow patterns. In *Temporal and spatio-temporal Data Mining*, pp. 189–208. IGI Publishing.

Wynne Hsu, Mong Li Lee, J. W. (2009b). Mining spatio-temporal trees. In *Temporal and spatio-temporal Data Mining*, pp. 209–226. IGI Publishing.

Yao, X. (2003). Research issues in spatio-temporal data mining. In *White paper UCGIS*.

Yuan, M. (2008). Toward knowledge discovery about geographic dynamics in spatiotemporal databases. In J. Han et H. J. Miller (Eds.), *Geographic Data Mining and Knowledge Discovery, Taylor and Francis*, pp. 347–365.

Summary

In this paper, we propose a new theoretical framework for modeling the dynamics of spatio-temporal phenomena. We define a new concept: spatio-temporal sequences of patterns. We propose a new incremental algorithm for the construction and mining of these sequences in spatio-temporal database. A prototype has been developed and tested on real data.