Extraction de motifs séquentiels. Problèmes et méthodes

Florent Masseglia, Maguelonne Teisseire, Pascal Poncelet

▶ **To cite this version:**

Florent Masseglia, Maguelonne Teisseire, Pascal Poncelet. Extraction de motifs séquentiels. Problèmes et méthodes. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, 2004, 9 (3/4), pp.183-210. 10.3166/isi.9.3-4.183-210 . lirmm-00108563

HAL Id: lirmm-00108563
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108563

Submitted on 3 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Aide au diagnostic de pannes guidée par l'extraction de motifs séquentiels

Julien Rabatel[*,**], Sandra Bringay[*,***], Pascal Poncelet[*], Maguelonne Teisseire[*]

[*]LIRMM, 161 rue Ada, 34392 Montpellier Cedex 5, France
{rabatel,bringay,poncelet,teisseire}@lirmm.fr
[**]Fatronik France, Cap Omega, Rond-point Benjamin Franklin - CS 39521
34960 Montpellier, France
[***]Université Montpellier 3, Route de Mende
34199 Montpellier Cedex 5, France

Résumé. La maintenance de systèmes complexes pose problème dans de nombreux domaines industriels. Le diagnostic de pannes à partir de données issues de capteurs fournissant de nombreuses informations complémentaires est un véritable défi. Nous nous intéressons ici à la caractérisation des comportements normaux de ces systèmes par des méthodes d'extraction de connaissances. Il s'agit d'un problème difficile. Les données contiennent diverses erreurs et proviennent de nombreuses sources pouvant correspondre à différents types d'informations propres aux systèmes étudiés. Nous étudions et proposons plusieurs solutions afin de traiter efficacement ce type de données et d'offrir des connaissances utiles et suffisamment complètes pour répondre aux exigences de la maintenance. Nous proposons donc une méthodologie complète, allant de l'acquisition des données brutes issues des capteurs, jusqu'à l'extraction des connaissances souhaitées. Ainsi, en premier lieu, nous nous intéressons au problème de la représentation de telles données pour dégager l'information contenue dans les données brutes. Puis, afin de fournir des connaissances utiles et valides, nous étudions les méthodes de fouille de données existantes pour les adapter à notre problématique. De plus, nous proposons une méthode de détection de tendances qui tient compte de l'évolution des comportements normaux au fil du temps due, par exemple, à l'usure du matériel. L'applicabilité des propositions développées est évaluée sur un jeu de données réel.

1 Introduction

De nombreux domaines industriels doivent faire face aux problèmes soulevés par la maintenance. En particulier, le manque de connaissances liées au comportement des systèmes surveillés rend difficile les tentatives de diagnostic ou de prévision de pannes des experts, et ce malgré l'usage de nombreux capteurs qui fournissent diverses informations sur le comportement global des systèmes étudiés.

Les données relevées par les capteurs sont difficiles à exploiter. Ce sont des informations comportementales diverses (e.g. températures, taux d'humidité, accélérations, etc.) auxquelles

s'ajoutent souvent des informations contextuelles tout aussi multiples (e.g. météo, géolocalisation, etc.). Intégrer l'ensemble de ces informations est également rendu difficile par les spécificités des données issues de capteurs : de grands volumes de données sont obtenus. Elles contiennent également de nombreuses erreurs : celles propres aux capteurs qui peuvent devenir défectueux (e.g. un capteur peut avoir des pannes intermittentes ou devenir complètement hors d'usage) ou être sujets au bruit (e.g. la transmission de l'information qui peut être perdue ou bruitée), etc.

Dans ce contexte, il est nécessaire de proposer des méthodes pour extraire des connaissances sur les comportements des systèmes dans une perspective d'aide au diagnostic des pannes. Caractériser un comportement normal (i.e. décrire les caractéristiques des comportements qui ne renferment pas d'anomalies : pannes, surchauffes, etc.) permet de détecter les comportements anormaux ayant des caractéristiques différentes des comportements attendus et donc d'anticiper des problèmes potentiels. Par exemple, si nous étudions l'évolution d'un capteur au cours du temps et que celui-ci commence à avoir un comportement différent de celui attendu, une alarme peut être déclenchée et un contrôle peut être réalisé afin d'anticiper une éventuelle panne. Dans cet article, nous nous focalisons uniquement sur la description de comportements normaux, en étudiant plus précisément le cas de la maintenance de trains.

Ces dernières années, de nombreux travaux ont abordé le problème de l'extraction de connaissances à partir de données issues de capteurs. Les approches développées sont diverses et liées aux caractéristiques particulières des données traitées : un nombre de capteurs élevé (Rodrigues et Gama (2006)), des données numériques (Yairi et al. (2001), Halatchev et Gruenwald (2005)) ou non numériques (Jakkula et Cook (2007), Guralnik et Haigh (2002)), etc. Chaque approche souhaite répondre à un problème précis, et propose des méthodes spécifiques et difficilement applicables dans un autre contexte. Ainsi, il existe plusieurs méthodes *(1)* de modélisation des données qui utilisent divers procédés (clustering de capteurs dans Rodrigues et Gama (2006), discrétisation dans Yairi et al. (2001), et *(2)* de fouille de données : règles d'association (Yairi et al. (2001)), motifs séquentiels (Guralnik et Haigh (2002)), etc. Nous souhaitons nous inscrire dans un contexte plus large : l'extraction de connaissances à partir de données temporisées issues de capteurs pour l'aide au diagnostic de pannes. Nous proposons donc des solutions générales, qui prennent en compte l'ensemble des caractéristiques que l'on peut retrouver dans de telles données pour produire des connaissances multiples et complémentaires. De manière plus précise, nous nous intéressons au problème de la maintenance ferroviaire, qui rassemble l'ensemble des problèmes que nous pouvons rencontrer.

Nous proposons une méthodologie complète qui englobe l'ensemble des tâches nécessaires pour obtenir, à partir de données brutes de capteurs, des connaissances utiles dans le cadre de la maintenance. Afin d'extraire des connaissances utiles, notre contribution aborde dans un premier temps le problème de la représentation des données et nous proposons différentes méthodes qui exploitent les caractéristiques propres des données. Nous nous intéressons ensuite à la fouille des données ainsi modélisées. De nombreux travaux existants répondent au problème de l'extraction de motifs descriptifs des données (e.g. règles d'association, motifs séquentiels). Nous étudions en détail l'apport de ces différentes approches dans le contexte particulier de la caractérisation des comportements pour la maintenance. En effet, les données manipulées étant complexes, de nombreuses connaissances peuvent être obtenues en fonction des méthodes utilisées. Il est alors indispensable, pour construire un ensemble de connaissances réellement représentatif et exploitable, de prendre en compte la complémentarité de ces

approches.

De plus, nous abordons un problème non évoqué dans les travaux existants. Le comportement normal des systèmes étudiés varie au cours du temps : l'impact de l'usure du matériel influe sur le comportement global observé, sans pour autant le rendre anormal. Par exemple, un capteur situé sur une roue de train sera plus sensible à l'environnement qui aura un impact plus marqué sur le comportement du capteur. Ainsi, les connaissances acquises lorsque le matériel est neuf ne sont valides que pour une certaine période de temps et deviennent inutiles lorsque le matériel impliqué est usé. Afin de remédier à ce problème, nous proposons une approche d'extraction des tendances dans les données, offrant des connaissances sur les comportements normaux, mais également sur leur évolution.

L'article est organisé de la manière suivante. Dans la section 2, nous nous intéressons à la représentations des données issues de capteurs. La section 3 présente les différentes méthodes de fouille de données employées et leur adéquation avec l'objectif de caractérisation des comportements normaux en considérant les différentes représentations. La section 4 introduit notre proposition d'extraction des tendances générales. Les méthodes proposées sont évaluées dans la section 5. Enfin, nous concluons dans la section 6.

2 Représentation des données

Les données brutes issues de capteurs fournissent des informations importantes sur le comportement des systèmes étudiés. Leurs spécificités les rendent difficiles à exploiter dans un processus d'extraction de connaissances sans une représentation adéquate. Dans cette section, nous proposons différentes approches de représentation.

2.1 Description des données

Les données issues de capteurs dans le contexte de la maintenance sont complexes pour deux raisons : *(1)* les caractéristiques propres aux données de capteurs (bruit, erreurs diverses) et *(2)* l'aspect multi-sources qui amène des informations diverses. Ces données comportementales donnent à manipuler les différents éléments listés ci-dessous.

Les capteurs. Chaque capteur décrit une propriété du comportement global du système étudié pouvant correspondre à différentes informations (e.g. températures, vitesses, accélérations, taux d'humidité, etc.).

Les mesures. Il s'agit des valeurs numériques relevées par les capteurs. Celles-ci sont soumises à diverses erreurs liées aux capteurs : bruit, défaillances, etc.

Les relevés. Ils sont définis comme l'ensemble de valeurs mesurées par l'ensemble des capteurs à une date donnée. L'information portée par un relevé est l'état du comportement global observé à un instant donné. En raison des erreurs liées au transfert des informations, certains relevés peuvent être incomplets, voire manquants.

Nous considérons que les données manipulées sont telles que celles décrites dans la figure 1, où un **relevé** pour une date donnée (la première colonne) est décrit par les **mesures** des **capteurs** (les cellules des autres colonnes). Les différents éléments de ce tableau offrent plusieurs possibilités de représentation.

TEMPS	Capteur 1	Capteur 2	Capteur 3	...
2008/03/27 06:36:39	0	16	16	...
2008/03/27 06:41:39	82.5	16	16	...
2008/03/27 06:46:38	135.6	19	21	...
2008/03/27 06:51:38	105	22	25	...

FIG. 1 – *Extrait de données brutes issues de capteurs.*

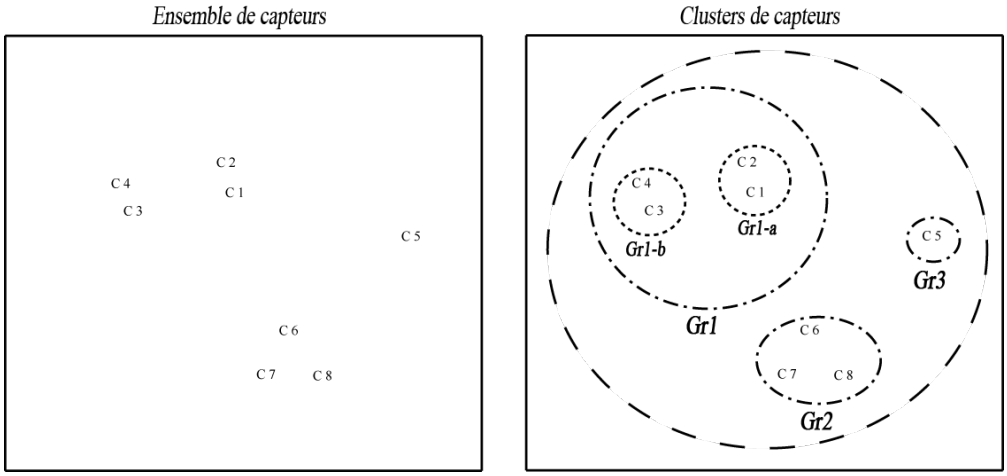

FIG. 2 – *Clustering de capteurs.*

2.2 Agrégation de capteurs

Dans de nombreux cas, l'étude des données montre que certains capteurs ont des comportements très similaires au cours du temps. En effet, dans un système complexe dont le comportement est décrit par des capteurs, il est fréquent que les valeurs de ces derniers soient très corrélées. Par exemple, deux capteurs de température placés sur une même roue d'un train subissent les mêmes conditions et mesurent donc sensiblement les mêmes valeurs. Ainsi, ces deux capteurs seront considérés comme similaires, car les informations qu'ils fournissent sont semblables.

Nous proposons de regrouper les capteurs similaires (i.e. dont les valeurs mesurées au cours du temps sont proches), les considérant alors comme un nouveau capteur, mesurant ses propres valeurs au cours du temps. Pour cela, nous appliquons une technique de clustering sur l'ensemble des capteurs pour créer une partition de cet ensemble en groupes telle que :

- les capteurs appartenant au même groupe se ressemblent,
- les capteurs appartenant à deux groupes différents soient peu ressemblants.

Nous souhaitons obtenir un partitionnement hiérarchisé de capteurs[1]. De manière générale,

[1] i.e. qui comporte des relations de subsomption entre groupes.

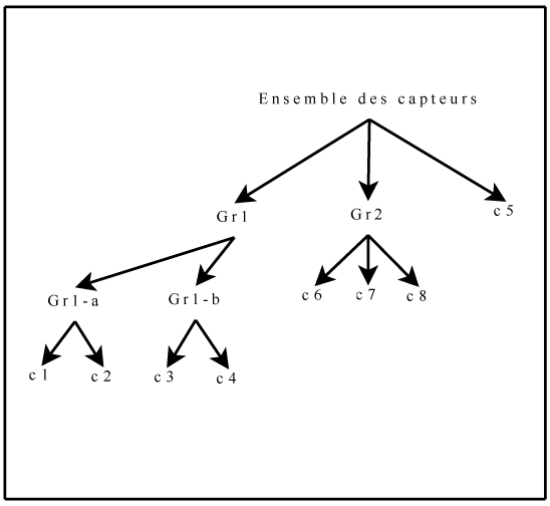

FIG. 3 – *Hiérarchie de capteurs.*

les algorithmes de clustering hiérarchique permettent d'obtenir ce type de partitionnement, représenté sous la forme d'un arbre. Cependant, l'utilisation de tels algorithmes pose le problème de la coupe de l'arbre obtenu, afin de conserver uniquement les groupes réellement significatifs. Nous pouvons par conséquent utiliser d'autres méthodes de clustering. Parmi les algorithmes fréquemment utilisés, *k-means* a l'inconvénient de partitionner en un nombre de groupes qui doit être fixé *a priori* par l'utilisateur. Pour corriger ce défaut, nous utilisons l'algorithme *X-means* décrit dans Pelleg et Moore (2000), qui estime lui-même le nombre de clusters adéquat. Afin de partitionner l'ensemble des capteurs, nous considérons l'espace à n dimensions $\{t_1, t_2, ..., t_i, ..., t_n\}$, où t_i est une date de relevé. Dans cet espace, un capteur est représenté par un vecteur dont chaque coordonnée correspond à la valeur mesurée par le capteur à la date associée à la dimension.

Pour l'ensemble des capteurs, nous appliquons alors l'algorithme *X-means*, puis sur chaque cluster obtenu, nous réappliquons l'algorithme de manière à partitionner en groupes de plus en plus précis. La figure 2 présente un ensemble de 8 capteurs placés dans un espace à deux dimensions. Dans un premier temps l'algorithme *X-means* partitionne l'ensemble des capteurs en trois groupes : *Gr1*, *Gr2*, et *Gr3*. On réapplique *X-means* pour segmenter *Gr1* en deux nouveaux groupes.

Avec cette méthode, nous obtenons un partitionnement hiérarchisé de l'ensemble des capteurs qui peut être représenté sous la forme d'un arbre orienté (cf. figure 3). Les feuilles de cet arbre correspondent aux capteurs réels du système et les ancêtres symbolisent les différents groupes révélés par le clustering.

Nous sommes désormais en mesure d'agréger les capteurs selon la hiérarchie construite avec une précision choisie. Ce prétraitement permet de résumer les données en perdant un minimum d'informations.

2.3 Agrégation de relevés

L'étude de données brutes a montré que des relevés consécutifs pouvaient contenir les mêmes valeurs. Nous cherchons donc ici à agréger les relevés successifs qui portent sensiblement la même information. Il est alors utile de fusionner ces relevés pour en créer un nouveau qui résumera les données contenues dans les relevés originaux. Pour mesurer la similarité de deux relevés, nous proposons la mesure de similarité suivante.

Un relevé est une liste de valeurs noté : $(v_1, v_2, ..., v_i, ...v_n)^t$ où :
- t est l'estampille temporelle associée au relevé,
- n est le nombre de capteurs qui constituent le relevé,
- v_i est la valeur mesurée par le capteur numéroté i.

Soient deux relevés $r = (v_1, v_2, ..., v_i, ...v_n)^t$ et $r' = (v'_1, v'_2, ..., v'_i, ...v'_n)^{t'}$. La fonction de similarité $\Phi(r, r')$ qui, à deux relevés r et r', associe leur mesure de similarité est définie par :

$$\Phi(r, r') = \frac{\sum_{i=1}^{n} \rho(v_i, v'_i)}{n}$$

où $\rho(v_i, v'i)$ mesure la similarité entre deux valeurs et est définie ainsi :

$$\rho(v_i, v'_i) = \begin{cases} 1 \; si \; v_i = v'_i = 0, \\ \frac{min(v_i, v'_i)}{max(v_i, v'_i)} \; sinon. \end{cases}$$

Notons que toutes les valeurs manipulées sont positives ou nulles. Ainsi définie, la mesure de similarité Φ obéit aux deux propriétés suivantes :
- la symétrie : $\Phi(r, r')$ = $\Phi(r', r)$,
- $\Phi(r, r')$ est maximum (et égal à 1) quand $r = r'$.

Nous fixons un seuil minimum de similarité nommé *minSim*. Si la similarité de plusieurs relevés consécutifs dépasse *minSim*, alors ils sont jugés similaires et sont agrégés. Ce cas intervient notamment lorsque le comportement des systèmes observés est stable (e.g. un engin à l'arrêt) : les relevés successifs sont redondants puisque le comportement qu'ils décrivent ne change pas.

2.4 Agrégation de mesures de capteurs

Il existe plusieurs méthodes d'agrégation des valeurs dans un groupe de relevés ou de capteurs, telles que la somme, la moyenne, le minimum, le maximum, la variance, etc. Nous choisissons le calcul de la moyenne qui fournit une valeur représentative des éléments agrégés. Ainsi, la valeur associée à un groupe de capteurs pour une date donnée sera la moyenne des valeurs mesurées par chacun des capteurs de ce groupe.

2.5 Transformation des valeurs brutes

La discrétisation ou l'extraction de variations sont deux autres méthodes de représentation des données.

Temps	t_1	t_2	t_3	t_4	...	t_{n-1}	t_n
Mesures brutes	24	24	35	52	...	47	35
Mesures discrétisées	bas	bas	moyen	haut	...	moyen	moyen
Variations		$\rightarrow$	$\nearrow$	$\nearrow$	...	$\searrow$	$\searrow$

FIG. 4 – *Comparaison des différentes représentations des comportements de capteurs.*

Discrétisation. Le comportement d'un capteur est décrit par les valeurs numériques qu'il mesure (variable quantitative). La discrétisation de ces valeurs est un prétraitement qui a par exemple été utilisé dans Halatchev et Gruenwald (2005) et Chong et al. (2008) pour regrouper des valeurs exprimant la même information en classes de valeurs (e.g. une valeur est *basse*, *moyenne* ou *haute*). Par exemple, les valeurs 25 et 26 sont différentes mais très proches, à tel point qu'on puisse les considérer comme porteuses de la même information. Pour discrétiser, on partitionne l'univers des valeurs mesurées par un capteur en intervalles. Le tableau 4 présente, à partir de valeurs brutes mesurées par un capteur, les mesures discrétisées correspondantes.

Extraction de variations. Jusqu'à présent, nous avons considéré uniquement la valeur mesurée par un capteur. Une autre possibilité, utilisée dans Ma et al. (2004), concerne la description de la variation associée à cette valeur par rapport à la précédente.

La valeur est alors remplacée par $\nearrow$ si elle est en hausse, $\rightarrow$ si elle est stable et $\searrow$ lorsque qu'elle est en baisse.

L'information portée par la variation est différente de celle apportée par les valeurs elles-mêmes. Comme le montre le tableau 4, deux mesures à des temps différents peuvent être égales (t_3 et t_n) mais ne pas être associées à la même variation et inversement.

Dans cette section, nous avons présenté différents moyens de représenter les données issues de capteurs en étudiant les éléments qui les composent. Nous avons ainsi proposé des méthodes pour *(1)* réduire le volume de données sans perte d'informations en agrégeant les éléments redondants, et *(2)* dégager de nouvelles informations à partir des mesures brutes de capteurs (discrétisation, extraction de variations).

Les différentes représentations des données proposées sont exploitées dans la section 3 pour extraire des connaissances différentes.

3 Caractérisation de comportements normaux

Dans cette section, nous nous intéressons à l'étape de fouille de données du processus d'extraction de connaissances. A cette étape, à partir des différentes représentations des données obtenues dans la section précédente, nous souhaitons extraire des motifs qui caractérisent les comportements normaux. Il existe plusieurs techniques ayant leurs objectifs propres, plus ou moins adaptées aux données et au but du procédé d'extraction de connaissances. Nous nous intéressons à deux de ces techniques : *(1)* la découverte de règles d'association, et plus spé-

cialement *(2)* l'extraction de motifs séquentiels qui nous permet de prendre en compte l'aspect temporel des données.

Nous étudions les données pour les adapter aux formats requis par les algorithmes traditionnels d'extraction de motifs.

3.1 Règles d'association

Introduite par Agrawal et al. (1993), la problématique des règles d'association peut être résumée ainsi : soit une base de transactions (les paniers), chacune étant composée d'items (les produits achetés). La découverte de règles d'association consiste à chercher des ensembles d'items fréquemment liés et des règles les combinant. Un exemple d'association pourrait être : «*75% des gens qui achètent de la bière achètent également des couches* ».

Plus formellement, le problème de la découverte de règles d'association peut être défini comme suit : soit $I = \{i_1, i_2, ..., i_n\}$ un ensemble d'items. Un *itemset* est un ensemble non vide d'items noté $(i_1...i_k)$, où i_j est un item de I. Etant donné DB un ensemble de transactions tel que chaque transaction est constituée d'un itemset, il s'agit de trouver toutes les règles d'association de la forme $X \Rightarrow Y$, où X (l'antécédent) et Y (le conséquent) sont des sous-ensembles de I et $X \cap Y = \emptyset$.

Une règle d'association est caractérisée par sa *confiance* (le pourcentage de transactions qui supportent Y parmi celles qui supportent X) et son *support* (le pourcentage des transactions de DB supportant à la fois X et Y).

Ainsi, à partir d'une base de transactions, le problème de la recherche de règles d'association consiste à identifier toutes les règles dont le support et la confiance dépassent deux seuils fixés par l'utilisateur, appelés respectivement *support minimum* ($minSupp$) et *confiance minimum* ($minConf$).

Les données que nous manipulons comportent plusieurs éléments distincts : des *capteurs*, des *relevés*, des *dates de relevés*, des *états de capteurs* (valeurs numériques, valeurs discrétisées, ou variations), et des *périodes*[2]. Différentes règles d'association peuvent être extraites à partir de cet ensemble de données. Les différences reposent sur la manière de construire la base de transactions sur laquelle sera appliqué l'algorithme d'extraction. Nous avons évalué de manière exhaustive les différents cas possibles de formats de données. Le tableau 5 présente un récapitulatif des règles présentant un intérêt pour notre problématique ainsi que leur sémantique.

Par exemple, considérons une transaction comme étant un relevé, et un item l'état associé à un capteur, nous pouvons extraire des règles différentes selon les trois représentations identifiées précédemment, telles que celles décrites ci-dessous :

- **valeurs brutes** : $(C1_{30}, C2_{32}) \Longrightarrow (C3_{57})$ qui signifie «*lorsque le capteur C1 mesure 30 et le capteur C2 mesure 32, alors le capteur C3 mesure 57* ».
- **valeurs discrétisées** : $(C1_{bas}, C2_{bas}) \Longrightarrow (C3_{haut})$ qui signifie «*lorsque C1 et C2 mesurent des valeurs basses, alors C3 mesure une valeur haute* ».

[2]La notion de *période* correspond à un intervalle de temps dont la sémantique est propre à chaque domaine. Ainsi, dans le cas de comportements d'engins de transport par exemple, une telle période pourra correspondre à un trajet.

Transaction Items	Règles d'association	Sémantique
relevé valeurs brutes	$(C1_{30}, C2_{32}) \Longrightarrow (C3_{57})$	«*Lorsque le capteur C1 mesure 30 et le capteur C2 mesure 32, alors le capteur C3 mesure 57* »
relevé valeurs discrétisées	$(C1_{bas}, C2_{bas}) \Longrightarrow (C3_{haut})$	«*Lorsque C1 et C2 mesurent des valeurs basses, alors C3 mesure une valeur haute* »
relevé variations	$(C1_{\rightarrow}, C2_{\nearrow}) \Longrightarrow (C3_{\nearrow})$	«*Lorsque la valeur mesurée par* $C1$ *est stable et celle mesurée par* $C2$ *est en augmentation, alors la valeur mesurée par* $C3$ *est en augmentation* »
relevé périodes	(période1, période5) $\Longrightarrow$ (période15)	«*Lorsqu'un relevé est mesuré pendant les périodes 1 et 5, alors il est également mesuré pendant la période 15* »

FIG. 5 – *Différents formats de données pour l'extraction de règles d'association.*

– **variations** : $(C1_{\rightarrow}, C2_{\nearrow}) \Longrightarrow (C3_{\nearrow})$ qui signifie «*lorsque la valeur mesurée par* $C1$ *est stable et celle mesurée par* $C2$ *est en augmentation, alors la valeur mesurée par* $C3$ *est en augmentation* ».

Les connaissances apportées par ces règles portent sur les corrélations qui existent entre les états de capteurs au sein d'un même relevé.

La représentation des comportements a un impact sur le sens de ces corrélations. Dans le cas des valeurs brutes, les règles sont très précises mais ont un intérêt limité.

La discrétisation des mesures de capteurs permet de découvrir des règles plus générales, souvent plus intéressantes, et plus facilement interprétables. Les variations de capteurs produisent en revanche des connaissances tout à fait différentes mais néanmoins pertinentes qui se révèlent complémentaires des règles précédentes.

3.2 Motifs séquentiels

Les motifs séquentiels ont été introduits dans Agrawal et Srikant (1995) et peuvent être considérés comme une extension du concept de règle d'association en prenant en compte la temporalité associées aux itemsets. La recherche de motifs séquentiels consiste à extraire des ensembles d'items couramment associés au cours du temps. Dans le contexte du «*panier de la ménagère* », un motif séquentiel peut par exemple être : «*60% des clients achètent une télévision, puis achètent plus tard un lecteur de dvd* ».

Soit DB une base de transactions telles qu'une *transaction* T est un triplet de la forme : *(*$Client_{id}$*,* $Date_{id}$*,* $itemset$*)*, où chaque élément caractérise respectivement : le *client* qui a réalisé l'achat, la *date*, et les *items* achetés.

Une *séquence* s est définie comme une liste ordonnée non vide d'itemsets notée $\langle s_1 s_2 ... s_n \rangle$ où s_i est un itemset. Une *n-séquence* est une séquence de taille n (i.e. composée de n items).

Un client supporte une séquence s si et seulement si s est incluse dans la séquence de données de ce client. Le *support* d'une séquence est alors défini comme le pourcentage de clients de DB qui supportent s. Une séquence est dite fréquente si son support est au moins égal à une valeur minimale $minSupp$ spécifiée par l'utilisateur.

La recherche de motifs séquentiels dans une base de séquences DB consiste alors à trouver, pour une valeur de support minimale, toutes les *séquences fréquentes maximales* contenues dans DB.

Comme pour la découverte de règles d'association, les algorithmes d'extraction de motifs séquentiels requièrent un format spécifique. Celui-ci doit correspondre aux concepts de clients, dates et items définis plus haut. La différence essentielle réside dans le concept de date. Celle-ci doit nécessairement correspondre à une notion d'ordre sur l'attribut concerné. Parmi les éléments que nous manipulons, seuls deux sont naturellement ordonnés : les dates de relevés et les valeurs mesurées par les capteurs. Nous avons étudié toutes les connaissances qu'il est possible d'obtenir par le biais des motifs séquentiels, en considérant les éléments qui composent notre base de transactions. Le tableau 6 récapitule celles qui ont une signification dans notre contexte. Nous considérons par exemple qu'un client est un capteur, une date est une date de relevé, et un item est l'état associé à un capteur. Le nombre de clients distincts correspond au nombre de capteurs (ou de groupes de capteurs) considérés. Tous les itemsets sont des 1-itemsets (i.e. des itemsets qui ne contiennent qu'un item). On peut alors parler de séquences d'items.

Le motif séquentiel $\langle(30)(34)(38)\rangle$ signifie : «*Beaucoup de capteurs mesurent 30, puis 34, puis 38* ». La connaissance apportée par ce type de résultats concerne des comportements qui sont partagés par beaucoup de capteurs.

L'extraction des motifs séquentiels telle que décrite précédemment implique un problème sémantique. Par exemple, dans la séquence $\langle(C1_{bas})(C2_{bas}, C3_{haut})\rangle$ la seule information apportée est que l'état décrit dans le deuxième itemset survient **après** celui révélé par le premier itemset, sans contrainte sur la durée. Or, une brève durée nous autorise à estimer qu'il existe une réelle corrélation entre les deux évènements, ce qui n'est plus le cas lorsque ceux-ci sont très éloignés dans le temps.

En utilisant la notion de séquence généralisée introduite dans Srikant et Agrawal (1996) nous pouvons prendre en compte différentes contraintes temporelles et :

- regrouper des itemsets lorsque leurs dates sont assez proches via la contrainte de *windowSize*,
- considérer des itemsets comme trop rapprochés pour apparaître dans la même séquence fréquente avec la contrainte de *minGap*,
- considérer des itemsets comme trop éloignés pour apparaître dans la même séquence fréquente avec la contrainte de *maxGap*.

Comme nous cherchons à limiter la durée qui sépare deux itemsets consécutifs, nous utilisons la contrainte *maxGap*.

Par exemple, si le paramètre *maxGap* est fixé à 15 minutes, la séquence $\langle(C1_{bas})(C2_{bas}, C3_{haut})\rangle$ signifie que l'état décrit par le second itemset intervient **au plus** 15 minutes après le premier

Client Date Items	Motifs séquentiels	Sémantique
période date de relevé valeurs brutes	$\langle (C1_{30})(C2_{32}, C3_{57})\rangle$	« *Beaucoup de périodes vérifient le comportement suivant : le capteur* $C1$ *mesure la valeur* 30*, puis les capteurs* $C2$ *et* $C3$ *mesurent respectivement les valeurs* 32 *et* 57 »
période date de relevé valeurs discrétisées	$\langle (C1_{bas})(C2_{bas}, C3_{haut})\rangle$	« *Beaucoup de périodes vérifient le comportement suivant : le capteur C1 mesure une valeur basse, puis C2 mesure une valeur basse tandis que C3 mesure une valeur haute* »
périodes date de relevé variations	$\langle (C1_{\rightarrow})(C2_{\nearrow}, C3_{\nearrow})\rangle$	« *Beaucoup de périodes vérifient le comportement suivant : C1 mesure une valeur stable, puis C2 et C3 mesure une valeur en hausse* »
capteur date de relevé valeurs brutes	$\langle (30)(34)(38)\rangle$	« *Beaucoup de capteurs mesurent 30, puis 34, puis 38* »
capteur date de relevé valeurs discrétisées	$\langle (bas)(moyen)(moyen)\rangle$	« *Beaucoup de capteurs mesurent une valeur basse, puis une valeur moyenne, puis de nouveau une valeur moyenne* »
capteur date de relevé variations	$\langle (\nearrow)(\rightarrow)(\searrow)\rangle$	« *Beaucoup de capteurs mesurent des valeurs en hausse, puis stables, puis en baisse* »
relevé valeur capteurs	$\langle (C1)(C2,\ C3)\rangle$	« *Souvent, le capteur C1 mesure une valeur inférieure à celle mesurée par les capteurs C2 et C3* »

FIG. 6 – *Différents formats de données pour l'extraction de motifs séquentiels.*

itemset. La durée idéale à attribuer au paramètre *maxGap* dépend bien entendu du domaine concerné.

3.3 Motifs séquentiels multidimensionnels

L'extraction de motifs séquentiels décrite dans la section précédente ne tient compte que d'une seule dimension d'analyse. Les motifs séquentiels multidimensionnels visent à corriger cet inconvénient. Un item est, dans ce contexte, défini sur plusieurs dimensions.

Dans Plantevit et al. (2005), un *item multidimensionnel* défini sur les dimensions d'analyse $D_1, ..., D_n$ est noté $\{d_1, ..., d_n\}$ tel que $d_i \in Dom(D_i)$. Un *itemset multidimensionnel* est alors un ensemble non vide d'items multidimensionnels où chaque item est défini sur les mêmes dimensions d'analyse. De même, une *séquence multidimensionnelle* est une liste ordonnée d'itemsets multidimensionnels.

D'après ces définitions, un item ne peut être trouvé que s'il existe une combinaison de valeurs de domaines des dimensions d'analyse se retrouvant fréquemment dans les données. Or, il peut arriver qu'aucune combinaison ne soit fréquente. Par exemple, il est possible que les items multidimensionnels $\{A, r\}$ et $\{B, r\}$ ne soient pas fréquents, alors que r l'est. Pour cette raison la valeur joker, notée $*$, est introduite. Elle signifie que l'on ne tient pas compte de la valeur sur la dimension d'analyse. L'item $\{*, r\}$ est appelé *item étoilé*.

Afin d'obtenir de nouvelles connaissances, les motifs séquentiels multidimensionnels permettent de décrire l'état d'un capteur sur trois dimensions : le capteur lui-même, sa mesure, et la variation associée. Nous pouvons alors extraire des motifs séquentiels multidimensionnels tels que :

$$\langle(\{C1, bas, *\})(\{C2, moyen, \searrow\}\{C3, *, \searrow\})\rangle$$

qui aura pour signification : «Souvent, le capteur C1 mesure une valeur basse, puis C2 mesure une valeur moyenne en baisse, tandis que C3 mesure une valeur en baisse également». Sur le plan sémantique, l'apport du cadre multidimensionnel est important car il permet de rechercher des corrélations portant sur plusieurs types de représentations des mesures.

3.4 Motifs séquentiels flous

Dans la théorie des ensembles classique, un élément appartient ou n'appartient pas à un ensemble de manière exclusive (i.e. son degré d'appartenance à cet ensemble est 0 ou 1). Le raisonnement humain fait toutefois preuve d'une richesse qui lui permet d'admettre des situations intermédiaires. Zadeh (1965) introduit alors la théorie des sous-ensembles flous afin de modéliser la représentation humaine des connaissances en formalisant les notions d'imprécision et d'incertitude. Ainsi, un ensemble flou est défini de manière à contenir des éléments de façon partielle : le degré d'appartenance d'un élément à un ensemble flou est compris entre 0 et 1.

Les notions liées à l'extraction de motifs séquentiels flous décrites dans Fiot (2008) étendent celles des motifs séquentiels traditionnels. Un item flou, noté $[x, a]$, est l'association d'un item x à un sous-ensemble flou a, défini sur l'univers des valeurs de x. Dans ce contexte,

un *itemset flou* est un ensemble non ordonné d'items flous et une *séquence floue* est une liste ordonnée d'itemsets flous.

Le comptage du support d'une *séquence floue* s revient à calculer le nombre de clients qui supportent s. Ce problème rejoint la problématique de la détermination de la cardinalité d'un sous-ensemble flou. Plusieurs solutions étant possibles, Fiot (2008) décrit trois algorithmes basés sur trois méthodes de comptage distinctes : *Speedy Fuzzy*, *Mini Fuzzy* et *Totally Fuzzy*.

En appliquant ces concepts aux données issues de capteurs, l'état d'un capteur C1 peut être représenté par un item flou tel que $[C1, bas]$ qui signifie que C1 mesure une valeur qui appartient au sous-ensemble flou *bas*.

Le motif séquentiel flou $\langle([C1, bas])([C2, bas][C3, haut])\rangle$ signifiera alors «Souvent, le capteur C1 mesure une valeur basse, puis C2 mesure une valeur basse tandis que C3 mesure une valeur haute ».

L'intérêt d'un tel motif réside dans la richesse qui définit la notion d'appartenance à un ensemble. De plus, la modélisation de l'imprécision liée aux items flous rend l'extraction de motifs séquentiels moins sensible au bruit qui réside dans les mesures de capteurs.

4 Extraction de tendances générales

Dans la section précédente, nous avons proposé différentes approches d'extraction qui permettent d'obtenir des connaissances complémentaires en fonction de la représentation des données et des techniques utilisées. Toutefois, en étudiant les données brutes issues de capteurs, nous avons observé que les comportements des capteurs pouvaient évoluer au cours du temps. Les techniques précédentes ne prennent malheureusement pas en compte ces évolutions. Nous proposons ici d'extraire des tendances dans les comportements étudiés.

Considérons l'exemple suivant. Sur une période de 7 mois, le capteur x possède le même comportement que le capteur y. En utilisant, par exemple, les motifs traditionnels et une représentation des données sous la forme de valeurs discrétisées, nous pouvons obtenir ce type de motifs commun à x et $y : < (bas)(moyen)(moyen)(haut)(bas) >$. Si nous revenons sur la manière dont ce motif a été obtenu, nous constatons que nous avons considéré la période de 7 mois et que nous avons recherché les motifs qui apparaissent fréquemment sur cette période. Supposons à présent que ce motif n'apparaît réellement que sur les 6 premiers mois et que, au cours du dernier mois, le comportement du capteur x, équivalent à celui du capteur t soit le suivant : $< (bas)(haut)(haut)(moyen)(bas) >$. Etant donné que l'analyse est effectuée sur une longue période de 7 mois, le dernier motif n'est probablement pas fréquent et a de grandes chances de ne pas apparaître, ayant une valeur de support trop faible. Du point de vue du diagnostic, la connaissance que nous avons obtenue n'est pas significative de ce qui se passe réellement sur le comportement des capteurs. Notre problème est donc de pouvoir détecter des tendances générales et plus particulièrement de faire émerger les périodes pendant lesquelles des capteurs ont le même comportement sur la globalité de l'historique.

Ces dernières années de nombreuses approches ont été proposées pour extraire ces tendances et Kontostathis et al. (2003) propose un état de l'art des principaux travaux. La majorité se sont intéressés à des données textuelles pour essayer d'extraire automatiquement de nouvelles thématiques ou au contraire de voir les moments où une thématique a tendance à

disparaître. Un exemple typique est l'apparition du langage XML qui est apparu au milieu des années 1990. En utilisant les résultats de la base de données INSPEC®[3], nous constatons qu'en 1994, le nombre de documents contenant XML est de 3, en 1996 de 8, en 1998 de 170 et en 1999 de 371.

Parmi les approches proposées, dans Lent et al. (1997), les auteurs proposent d'extraire les tendances à l'aide de motifs séquentiels. Le principe général est le suivant : considérons une base de données sur une période d'un an. La base est découpée en périodes de 1 mois. Pour chaque mois, un algorithme d'extraction de motifs est appliqué avec différentes valeurs de support (e.g. pour 1%, 2%, 3% etc.) et les résultats obtenus sont stockés dans une base de données. Cette dernière contient donc pour une valeur de support et un mois donné, l'ensemble des séquences fréquentes extraites. A la fin du traitement des douze mois, nous disposons donc d'une base de motifs séquentiels par période. L'obtention des tendances dans ce cas consiste à interroger la base de données afin de voir comment évoluent au cours du temps les séquences fréquentes obtenues.

Algorithme 1 : Extraction des tendances

Données : B_1, ...B_n les bases de données correspondantes à chaque sous-période (i.e. chaque mois) de la période d'observation totale (i.e. une année), $incrSupport$ la valeur d'incrémentation du paramètre $minSupp$ pour l'extraction de motifs séquentiels

début

```
/* Pour chaque base de données B_i                          */
pour chaque i ∈ {0, ..., n} faire
    minSupp = 0;
    /* les motifs séquentiels sont extraits pour
       différentes valeurs de support minimum                */
    tant que minSupp ≤ 1 faire
        minSupp = minSupp + incrSupport;
        extraire(B_i, minSupp);
        /* les motifs extraits sont enregistrés dans la base de
           données avec leur support et la liste des clients qui
           les supportent.                                    */
        stockeMotif(séquence, support, clients);
```

fin

Revenons à notre problématique. Afin d'extraire des tendances dans des données issues de capteurs, nous généralisons la proposition de Lent et al. (1997) (spécifique aux données textuelles) et proposons l'algorithme 1. La section 3 nous a montré que nous étions capables d'obtenir des séquences représentatives par extraction de motifs séquentiels sur les données comportementales issues de capteurs. L'algorithme 1 nous permet d'obtenir en fonction des

[3]http ://www.theiet.org/publishing/inspec/

différents résultats obtenus sur chaque période les tendances correspondantes.
Cet algorithme utilise les deux fonctions suivantes :

- $extraire(B_i, minSupp)$ effectue l'extraction de motifs sur la base de données B_i pour un support minimum fixé à $minSupp$,
- $stockeMotif(séquence, support, clients)$ stocke les séquences fréquentes obtenues dans une base de données. Chaque motif séquentiel extrait est stocké avec son support et la liste des clients qui le supportent. C'est cette liste de clients qui permet par la suite de rechercher les corrélations qui existent entre clients, et leurs évolutions.

L'algorithme est utilisable pour chaque représentation des données (cf. section 2) ou format de données (cf. section 3) choisi, mais également pour tout algorithme d'extraction de motifs séquentiels sélectionné (i.e. motifs séquentiels traditionnels, multidimensionnels ou flous). Cette généricité est due à la fonction $extraire(minSupp)$ qui est définie en fonction des choix faits. Ainsi, l'extraction de tendances générales dans les comportements s'adapte aux différents types de connaissances recherchées qui ont été décrits précédemment. Notons également que le découpage indispensable en périodes est flexible. Dans l'exemple donné précedemment, nous avons choisi un découpage en mois, mais en fonction des systèmes étudiés et des connaissances liées au domaine, celui-ci peut être adapté : dans certains cas, il sera nécessaire d'effectuer un découpage basé sur des périodes plus courtes (en semaines par exemple), ou encore sur des périodes de taille variable (dans ce cas des connaissances *a priori* seront indispensables).

Notons que l'extraction des tendances générales permet de faire apparaître à la fois des comportements émergents, symbolisés par des séquences dont le support augmente avec le temps, et des comportements qui disparaissent, i.e. dont le support diminue. Cet aspect a plusieurs avantages. D'abord, la connaissance des tendances extraites fournit réellement une meilleure compréhension des comportements étudiés aux experts. De plus, il est plus aisé dans certains cas de faire de la prévention, en extrapolant les tendances connues. Enfin, la mise en valeur des changements de comportements au cours du temps par l'extraction des tendances garantit en permanence la justesse des connaissances extraites malgré l'évolution des comportements normaux. En particulier, ceci s'avère indispensable dans le but de détecter des anomalies (i.e. des pannes de composants dans le cas des données ferroviaires).

5 Expérimentations

Afin d'évaluer nos propositions, nous les avons appliquées sur des jeux de données réels ferroviaires. La maintenance ferroviaire est un domaine d'application difficile du fait du gros volume de capteurs utilisés pour décrire le comportement des trains en route. Le manque de connaissances sur le comportement normal d'un train est alors un frein pour l'exploitation de ces données pour l'aide au diagnostic de pannes.

Les données comportementales utilisées sont telles que : sur un train sont installés des capteurs de différents types (températures, accélérations, vitesse) qui mesurent et enregistrent des valeurs dans une base de données de manière régulière (toutes les cinq minutes). Dans ce contexte, les périodes considérées correspondent aux trajets effectués par les trains. Plus précisément, il existe 249 capteurs par train, dont 232 capteurs de température partagés sur les différents composants (e.g. roues, moteurs, etc.), 16 capteurs d'accélération répartis sur les wagons et un capteur qui mesure la vitesse globale du train. Les expérimentations ont été ef-

fectuées sur 12 trains bénéficiant de cet équipement, pour une centaine de trajets enregistrés dans la base de données. Les trajets utilisés pour l'expérimentation ont été sélectionnés par des experts du domaine ferroviaire et ne comportent pas d'anomalies. Par conséquent, les connaissances que nous en retirons correspondent bien à des connaissances sur les comportements normaux.

Notons que l'objectif de ces expérimentations n'est pas de comparer la méthodologie présentée ici à d'autres approches, ni de mesurer la performance de nos algorithmes, mais de souligner l'intérêt des résultats obtenus, notamment leur qualité et leur complémentarité par rapport aux résultats que l'on peut obtenir avec les méthodes classiques.

5.1 Protocole expérimental

Le protocole expérimental reprend l'organisation des propositions :

1. **Représentation des données.** (cf. section 2) Nous étudions l'agrégation des capteurs et l'influence de cette méthode selon les différents niveaux d'abstraction qu'elle permet de manipuler, ainsi que l'agrégation des relevés et l'influence sur les données selon la mesure de similarité proposée.
2. **Caractérisation des comportements normaux.** (cf. sections 3 et 4) Nous étudions les types de résultats obtenus à partir des différentes méthodes d'extraction de connaissances (règles d'association, motifs séquentiels, motifs séquentiels multidimensionnels et motifs séquentiels flous).

L'ensemble des expérimentations ont été effectuées sur un PC équipé d'un processeur Pentium 4 3GHz et de 2Go de mémoire vive.

5.2 Représentation des données

Nous avons proposé, dans le but de traiter les données issues de capteurs, deux méthodes visant à résumer les données en éliminant les éléments redondants. Nous étudions ici l'influence de ces approches.

5.2.1 Agrégation de capteurs

L'application du clustering sur les 249 capteurs qui décrivent le comportement d'un train permet de faire ressortir 5 niveaux d'abstraction, où le niveau 0 correspond aux capteurs réels et le niveau 5 agrège l'ensemble des capteurs sous la forme d'un unique groupe. Le tableau 7 fournit pour chaque niveau d'abstraction le nombre de capteurs obtenu.

Dans l'ensemble total des capteurs (niveau 5 d'abstraction) le clustering de capteurs révèle des groupes qui correspondent d'abord (niveau 4) aux différents types de capteurs (vitesse, accélération et température). Parmi les capteurs de température, les groupes suivants (niveau 3) révèlent les trois types de composants présents sur chaque wagon et sur lesquels sont placés les capteurs de température, que nous appellerons composants A, B et C. Le niveau 2 révèle les sous-composants : A1, A2, A3, A4, C1 et C2, et le niveau 1 correspond aux emplacements précis des capteurs sur les composants. Le niveau 0 correspond à l'ensemble des capteurs, i.e. en considérant séparément chaque wagon.

Niveau d'abstraction	Nombre de capteurs total
0	249
1	32
2	9
3	5
4	3
5	1

FIG. 7 – *Nombre de capteurs en fonction du niveau d'abstraction.*

L'agrégation de capteurs présente deux intérêts majeurs. *(1)* Elle réduit considérablement le nombre de capteurs nécessaires pour décrire le comportement d'un train dans un relevé (cf. tableau 7). *(2)* Elle rend la compréhension des données plus aisée en produisant des informations plus compactes pour un sens identique.

Par exemple, le niveau 3 d'abstraction révèle des informations du type « *la température sur les composants A est haute* », tandis qu'au niveau 0, elles sont de la forme « *le capteur C1 du composant A1 sur le wagon 2 est haute* » pour chaque capteur des composants A du train. Ces deux cas fournissent la même information, mais sous une forme plus générale et plus facile à interpréter lorsque le niveau d'abstraction est plus élevé.

5.2.2 Agrégation de relevés

L'agrégation des relevés restreint les redondances dans les données. Par exemple, lorsqu'un train reste à l'arrêt son comportement général ne varie pas. Les informations contenues dans les relevés enregistrés dans une telle période sont alors peu intéressants car ils ne fournissent pas d'informations additionnelles par rapport aux précédents relevés. La valeur du paramètre *minSim* doit alors être fixée de manière à faire un compromis entre une perte d'informations pourtant significatives lorsque *minSim* est choisi trop bas, ou une quantité d'informations trop importante et parfois redondante s'il est trop élevé.

La figure 8 présente le nombre de relevés composant un trajet en fonction du paramètre *minSim*. Le trajet considéré est originellement constitué de 93 relevés. Plus le paramètre *minSim* est bas, et plus le nombre de relevés est réduit. Lorsque *minSim* est fixé à 0.9, le nombre de relevés est diminué de 88%. Une telle réduction est due au fait que la similarité entre deux relevés consécutifs est toujours très proche de 1, notamment car les valeurs mesurées par les capteurs de température évoluent lentement.

5.3 Découverte de règles d'association

L'extraction de règles d'association est effectuée à l'aide d'une implémentation de l'algorithme *Apriori*, décrite dans Bodon (2003). Nous avons dans la section 3 décrit plusieurs formats de données à partir desquels nous pouvons extraire différentes connaissances à l'aide de règles d'association. Ayant déjà présenté les apports sémantiques correspondants, nous nous concentrons ici sur l'impact des résultats en fonction des différentes représentations des données et des critères propres aux algorithmes de découverte de règles d'association. Pour cela,

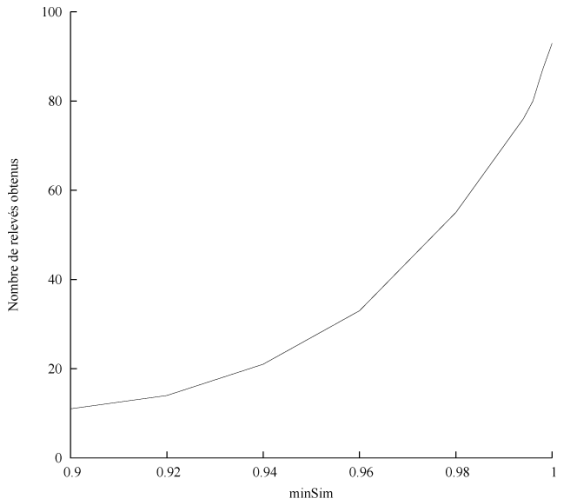

FIG. 8 – *Nombre de relevés d'un trajet en fonction du paramètre minSim.*

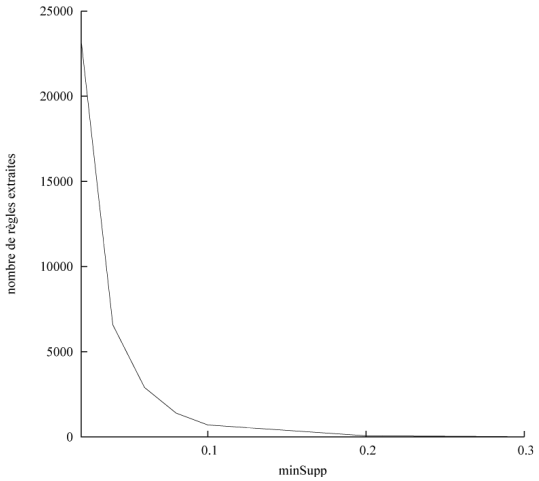

FIG. 9 – *Nombre de règles extraites en fonction du support minimum (items discrétisés).*

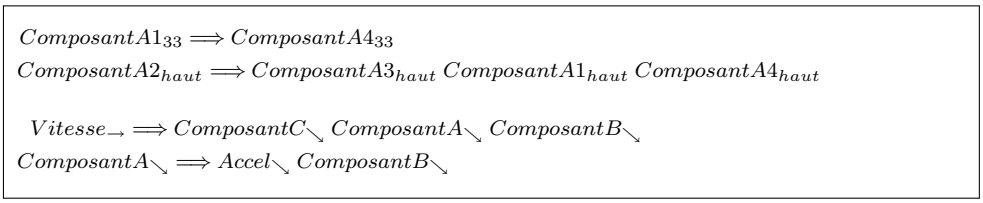

FIG. 10 – *Exemples de règles d'association extraites.*

nous considérons une transaction comme étant un relevé, et un item comme l'état associé à un capteur (cf. tableau 5).

La figure 10 présente des règles d'association extraites avec ce format de données. La courbe 9 présente le nombre de règles d'association extraites en fonction du support minimum (*minSupp*) choisi, lorsque les mesures de capteurs sont discrétisées. Nous constatons que ce nombre augmente exponentiellement lorsque *minSupp* diminue. Cette augmentation est due au fait que la recherche porte sur des règles de moins en moins fréquente et, par conséquent, de plus en plus nombreuses. Notons également que lorsque *minSupp* est inférieur à 75% environ, le nombre de règles extraites devient très grand et va à l'encontre de notre objectif d'obtenir un ensemble de règles facilement interprétables. En effet, une quantité de résultats trop élevée rend la compréhension de ceux-ci délicate.

5.4 Extraction de motifs séquentiels

La caractérisation des comportements ferroviaires normaux par l'extraction de motifs séquentiels a été effectuée par l'algorithme VPSP (L. Di-Jorio (2006)).

Nous faisons le choix d'un format de données parmi ceux étudiés dans la section 3. Nous considérons une transaction comme correspondant à l'ensemble des états associés à un capteur (les *items*) pour un relevé (le *client*) à la date de ce relevé (la *date*). La figure 11 expose quelques exemples de motifs séquentiels extraits à partir de ce format.

La courbe 12 montre le nombre de motifs séquentiels extraits lorsque l'état d'un capteur est obtenu par discrétisation en fonction du support minimum. Afin de dégager les comportements normaux, nous recherchons les motifs qui ont un support élevé (supérieur à 60%). En effet, les motifs séquentiels qui caractérisent les données apparaissent dans une majorité de trajets. Comme pour les règles d'association, le nombre de motifs séquentiels s'accroît à mesure que le support minimum diminue, occasionnant également une quantité de résultats trop importante lorsque celui-ci est trop bas.

De plus, la courbe 13 montre que le temps d'exécution nécessaire augmente également lorsque *minSupp* diminue[4].

Nous avons exposé dans la section 3 la nécessité d'introduire la contraite de temps *maxGap* dans l'extraction de motifs séquentiels ainsi que son apport sur le plan sémantique. Il s'agit

[4]Cette observation concorde également avec les résultats obtenus lors de l'extraction de règles d'association.

$\langle\ (Vitesse_0\ Accel_0)\ \rangle$
$\langle\ (Vitesse_{nulle}\ ComposantA_{bas}\ ComposantB_{bas})\ (ComposantA_{moyen}\ ComposantB_{haut})\ \rangle$
$\langle\ (Vitesse_{\nearrow}\ Accel_{\nearrow})\ (ComposantA_{\nearrow}\ ComposantB_{\nearrow}\ ComposantC_{\nearrow})\ \rangle$

FIG. 11 – *Exemples de motifs séquentiels extraits.*

également d'une nécessité sur le plan quantitatif. En effet, lorsque l'on n'utilise pas la contrainte temporelle *maxGap*, ou lorsque *maxGap* est élevé, le nombre de séquences fréquentes devient très grand, même pour un support minimum élevé et peut rendre l'extraction impossible en raison du trop grand nombre de séquences potentiellement fréquentes. Dans l'ensemble de nos expérimentations, nous avons fixé *maxGap* à 3, i.e. qu'au plus 3 relevés séparent deux itemsets consécutifs dans une séquence fréquente.

Dans la section 2, nous avons montré que plusieurs représentations de l'état d'un capteur sont possibles et avons présenté les caractéristiques sémantiques de chacune. L'expérimentation nous permet d'observer que des différences sont également présentes sur le plan quantitatif. En effet, pour *minSupp* fixé à 75% et des conditions identiques, le nombre de motifs séquentiels trouvés est de 4 lorsque l'on considère les mesures brutes des capteurs, 2708 pour des mesures discrétisées, et 49499 lorsque l'on s'intéresse aux variations. Un telle disparité provient des caractéristiques des données, et notamment du nombre d'items distincts. En effet, si dans le cas des valeurs brutes un capteur peut avoir un nombre d'états égal au nombre d'éléments de l'univers de ses valeurs, après discrétisation ce nombre descend à 6 (i.e. le nombre de classes choisi pour partitionner l'univers des valeurs), et à 3 après extraction des variations. Le nombre de motifs extraits augmente à mesure que le nombre d'items distincts diminue, car chaque séquence est alors potentiellement plus fréquente.

5.5 Motifs séquentiels multidimensionnels

L'extraction de motifs séquentiels multidimensionnels a été effectuée par l'algorithme M^2SP développé dans Plantevit et al. (2005). Les motifs séquentiels extraits permettent alors d'obtenir des informations plus complètes qui prennent en compte les différentes représentations que nous avons proposées.
Ainsi, nous pouvons extraire des motifs séquentiels tels que celui présenté ci-dessous :

$$\langle\ (\{Vitesse, bas, \nearrow\}\ \{ComposantA, bas, \rightarrow\})(\{ComposantA, bas, \nearrow\})\ \rangle$$

Le comportement décrit par ce motif séquentiel multidimensionnel signifie que souvent la vitesse est basse et en augmentation et la température des composants A est stable est basse puis la température sur les composants A, bien qu'encore basse, commence à augmenter.

Notons que les motifs séquentiels multidimensionnels étant une extension des motifs séquentiels classiques, les caractéristiques des algorithmes ont sensiblement le même profil en fonction des différentes représentations des données choisies ou des paramètres fixés par l'utilisateur (*minSupp* et *maxGap*).

5.6 Motifs séquentiels flous

L'extraction de motifs séquentiels flous est effectuée par l'algorithme *SpeedyFuzzy* décrit dans Fiot (2008). Nous partitionnons dans un premier temps l'univers des valeurs que peuvent mesurer les capteurs à l'aide d'une partition floue. Par exemple, sur l'univers des températures, il existe 6 ensembles flous que nous appelons $tbas$, bas, $moyen_-$, $moyen_+$, $haut$ et $thaut$, dont les définitions s'appuient sur celles des ensembles utilisés pour la discrétisation des mesures de capteurs. Ainsi, une température de 28°C appartient à l'ensemble $tbas$ (valeurs très basses) avec un degré d'appartenance de 0,6 et à bas (valeurs basses) avec un degré égal à 0,4.

L'application de l'algorithme extrait alors des motifs séquentiels flous tels que :

$$\langle([Vitesse, moyen_+])([ComposantA, moyen_+]\ [ComposantB, moyen_+])\rangle$$

Comme pour l'extraction de motifs séquentiels multidimensionnels, l'algorithme de découverte de motifs séquentiels flous a un comportement similaire à l'algorithme d'extraction de motifs séquentiels traditionnels.

5.7 Extraction de tendances dans les comportements ferroviaires

Dans la figure 14, nous présentons quelques résultats obtenus en analysant les tendances au cours du temps. Soit le motif séquentiel $< (bas)(moyen)(bas)(moyen)(haut) >$. Le comportement ainsi décrit est très fréquent (74% des capteurs le vérifient) au début de la période d'analyse, puis il devient de moins en moins fréquent jusqu'à atteindre un support de 48% au bout d'un an. A l'inverse, la séquence $< (bas)(moyen)(haut)(haut)(haut) >$ a au début de la période d'observation un support faible de 30%, qui va croître tout au long de l'année jusqu'à devenir plus important que le support de la première séquence et atteindre 62%. Ces résultats reflètent bien les changements de comportements des capteurs au cours du temps et montrent à quel point il est important de prendre en compte ces évolutions pour acquérir des connaissances réellement utiles. Cependant, valider l'approche nécessite de l'expérimenter dans un système de diagnostic de pannes, afin d'en évaluer l'apport dans un tel contexte.

5.8 Synthèse

Les méthodes que nous avons proposées et expérimentées sur des jeux de données réels permettent d'extraire des motifs représentatifs à partir de données issues de capteurs. Les différents types de représentations des données sont suffisamment adaptables pour s'accorder avec les diverses caractéristiques des données que l'on peut rencontrer, comme nous l'avons fait pour les données ferroviaires. Ainsi, dans certains domaines, des relevés sont enregistrés très fréquemment et décrivent pourtant des comportements qui évoluent peu avec le temps. Il est alors nécessaire de mettre l'accent sur l'agrégation de relevés pour résumer les données (en modifiant le paramètre $minSim$). De même, dans le cas où un nombre de capteurs trop élevé entraînerait un volume de données trop grand et trop redondant pour permettre l'extraction de connaissances utiles, l'agrégation de capteurs en adoptant un certain niveau d'abstraction est adéquate. Cette adaptabilité s'étend également aux connaissances recherchées. Par exemple, dans certains domaines, les simples valeurs mesurées par les capteurs ne permettent

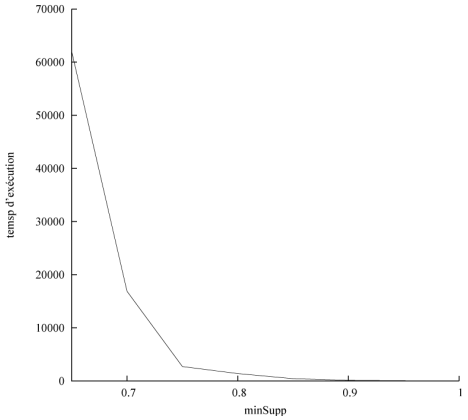

FIG. 12 – *Influence du support minimum sur le nombre de résultats extraits.*

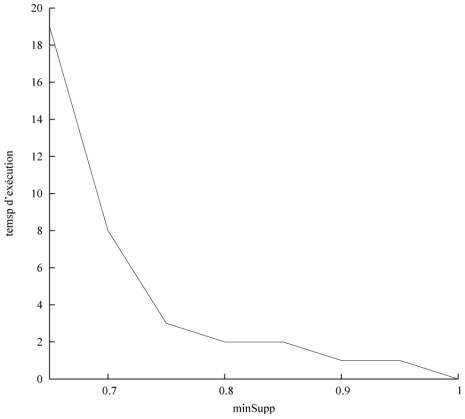

FIG. 13 – *Influence du support minimum sur le temps d'exécution (en secondes).*

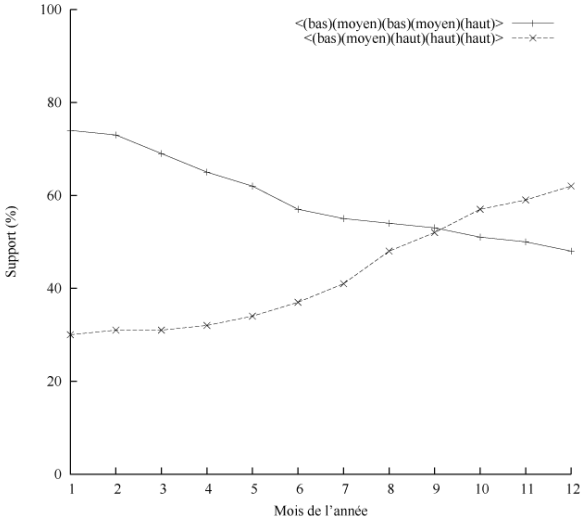

FIG. 14 – *Tendances extraites.*

pas d'obtenir des connaissances satisfaisantes, car elles sont trop dépendantes de conditions externes. C'est le cas avec les données ferroviaires, où les températures mesurées sur certains composants sont dépendantes à la fois du comportement du train, mais aussi de facteurs tels que la température externe. Ne disposant pas d'informations à propos de ce facteur dans nos données ferroviaires, les températures mesurées sont peu caractéristiques[5]. En revanche, les variations de ces températures jouent un rôle important. Cet exemple montre la nécessité d'obtenir des connaissances complémentaires pour répondre aux besoins propres au diagnostic de pannes.

Les résultats qui ont été extraits pour la caractérisation des comportements normaux de trains sont suffisamment complets et intelligibles pour offrir une meilleure compréhension des comportements aux experts du domaine ferroviaire. Cependant, les motifs obtenus étant nombreux et de différents types, leur consultation est difficile. De plus, afin de confirmer l'utilité et la validité des motifs extraits, il est nécessaire de les utiliser dans un contexte de reconnaissance de pannes capable, en comparant les comportements des trains avec nos résultats, d'identifier ceux qui comportent des anomalies (i.e. qui ne correspondent pas aux motifs caractérisant les comportements normaux). Cette perspective peut dans un premier temps être remplie dans un contexte statique, mais à terme un objectif intéressant concerne la création d'un système de diagnostic temps réel dans un contexte de flot de données. Dans ce contexte, des problèmes liés au nombre de motifs extraits ou à leur redondance pourront être soulevés. En effet, lorsque le temps d'exécution est primordial, l'ensemble des connaissances obtenues devra à la fois être concis et complet. Les problèmes que nous soulevons engendrent des questions intéressantes : combien de motifs conserver pour caractériser les comportements normaux pour faire de la détection en temps réel ? Quels sont les motifs les plus significatifs ? Comment détecter et éliminer la redondance dans les motifs extraits via les différentes méthodes et représentations proposées ?

6 Conclusion

Nous nous sommes intéressés au problème de l'extraction de connaissances à partir de données comportementales issues de capteurs dans le but de caractériser les comportements normaux de systèmes complexes. Cette problématique soulève de nombreux problèmes liés à la complexité des données (bruit, erreurs diverses, volume de données et aspect multi-sources). Nous avons dans un premier temps proposé des solutions visant à représenter les données liées aux capteurs en tenant compte des différentes spécificités.

Nous avons ensuite cherché à caractériser les comportements normaux portés par l'ensemble des capteurs. Pour ce faire, nous avons étudié l'adéquation de différentes techniques de fouille de données dans ce contexte, principalement basées sur l'extraction de motifs séquentiel. L'intérêt d'une telle méthodologie est d'offrir à l'utilisateur des connaissances qui s'avèrent complémentaires. De plus, le contexte lié aux capteurs nous a conduit à considérer l'évolution des comportements normaux dans le temps. Ainsi, l'extraction de tendances que nous avons proposée permet de considérer le fait que les comportements normaux évoluent au cours du

[5]Notons que cette information pourrait aisément être rajoutée en installant les capteurs adaptés sur les trains. Cette information contextuelle pourrait alors être prise en compte et apparaître dans les motifs caractérisant les données, pour fournir des connaissances plus complètes.

temps. En effet, l'une des particularités des données issues de capteurs est qu'un comportement considéré comme normal peut par la suite devenir anormal en raison, par exemple, de l'usure des capteurs ou du matériel observé. L'approche proposée répond à cette spécificité en considérant l'évolution des comportements les plus féquents au cours du temps.

Les travaux effectués ouvrant de nombreuses perspectives, nous étudions maintenant la possibilité d'utiliser les connaissances acquises afin de mettre en place une détection d'anomalies comportementales en temps réel. Un tel système de diagnostic doit prendre en considération deux aspects. Le premier est lié à la reconnaissance de comportements anormaux (i.e. de comportements déviants par rapport aux comportements normaux des trains que nous avons pu extraire). Dans ce cadre, il est indispensable de comparer chaque événement survenant dans le train afin de rechercher des correspondances dans les comportements appris. Le second point est lié à la nécessité de poursuivre l'apprentissage des comportements en prenant en compte les données issus des trains en cours d'analyse. Bien entendu, ces deux points rentrent dans le cadre plus général des nouvelles approches d'extraction définies pour les flots de données.

Références

Agrawal, R., T. Imieliński, et A. Swami (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec. 22*(2), pp. 207–216.

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In P. S. Yu et A. S. P. Chen (Eds.), *Eleventh International Conference on Data Engineering*, Taipei, Taiwan, pp. 3–14. IEEE Computer Society Press.

Bodon, F. (2003). A fast apriori implementation. In B. Goethals et M. J. Zaki (Eds.), *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, Volume 90 of *CEUR Workshop Proceedings*, Melbourne, Florida, USA.

Chong, S. K., S. Krishnaswamy, S. W. Loke, et M. M. Gaben (2008). Using association rules for energy conservation in wireless sensor networks. In *SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing*, New York, NY, USA, pp. 971–975. ACM.

Fiot, C. (2008). Fuzzy sequential patterns for quantitative data mining. In *Galindo, J. (Ed.), Handbook of Research on Fuzzy Information Processing in Databases*, pp. 727–744.

Guralnik, V. et K. Z. Haigh (2002). Learning models of human behaviour with sequential patterns. In *Proceedings of the AAAI-02 workshop "Automation as Caregiver"*, pp. 24–30. AAAI Technical Report WS-02-02.

Halatchev, M. et L. Gruenwald (2005). Estimating missing values in related sensor data streams. In J. R. Haritsa et T. M. Vijayaraman (Eds.), *Proceedings of the 11th International Conference on Management of Data (COMAD '05)*, pp. 83–94. Computer Society of India.

Jakkula, V. et D. Cook (2007). Learning temporal relations in smart home data. In *Proceedings of the Second International Conference on Technology and Aging*.

Kontostathis, A., L. Galitsky, W. M. Pottenger, S. Roy, et D. J. Phelps (2003). *A Survey of Emerging Trend Detection in Textual Data Mining*.

L. Di-Jorio, D. Jouve, D. K. A. S. C. R. A. L. M. T. P. P. (2006). VPSP : extraction de motifs séquentiels dans WEKA. In *Démonstration dans les 22èmes journées de Bases de Données*

avancées (BDA'06).

Lent, B., R. Agrawal, et R. Srikant (1997). Discovering trends in text databases. In *KDD*, pp. 227–230.

Ma, X., D. Yang, S. Tang, Q. Luo, D. Zhang, et S. Li (2004). Online mining in sensor networks. In H. Jin, G. R. Gao, Z. Xu, et H. Chen (Eds.), *NPC*, Volume 3222 of *Lecture Notes in Computer Science*, pp. 544–550. Springer.

Pelleg, D. et A. W. Moore (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 727–734. Morgan Kaufmann Publishers Inc.

Plantevit, M., Y. W. Choong, A. Laurent, D. Laurent, et M. Teisseire (2005). M^2SP: Mining sequential patterns among several dimensions. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, et J. Gama (Eds.), *PKDD*, Volume 3721 of *Lecture Notes in Computer Science*, pp. 205–216. Springer.

Rodrigues, P. P. et J. Gama (2006). Online prediction of streaming sensor data. In J. S. A. R. João Gama, J. Roure (Ed.), *Proceedings of the 3rd International Workshop on Knowledge Discovery from Data Streams (IWKDDS 2006), in conjuntion with the 23rd International Conference on Machine Learning*.

Srikant, R. et R. Agrawal (1996). Mining sequential patterns: Generalizations and performance improvements. In P. M. G. Apers, M. Bouzeghoub, et G. Gardarin (Eds.), *Proc. 5th Int. Conf. Extending Database Technology, EDBT*, Volume 1057, pp. 3–17. Springer-Verlag.

Yairi, T., Y. Kato, et K. Hori (2001). Fault detection by mining association rules from housekeeping data. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Zadeh, L. (1965). Fuzzy sets. *Information Control 8*, pp. 338–353.

Summary

Maintenance is a very challenging task in numerous industrial fields. Although the presence of many sensors provides much complementary information about the studied systems, but failure diagnosis still remains a difficult task. We investigate the characterization of normal systems behaviors by means of knowledge discovery techniques. This particular context implies difficulties like various errors and multi-source aspects of the data. We survey and propose several solutions to efficiently process the sensor data for providing useful knowledge. First, we are interested in the problem of representing the data. Then, in order to provide useful and valid knowledge, we investigate the existing data mining techniques to adapt them to our context. Besides, we propose a trend detection method to take into consideration the normal behaviors changes over time, for instance with equipment usury. In order to evaluate our approach, we apply it on a real dataset.