EGC 2006

6èmes Journées
Extraction et Gestion de Connaissances

Atelier 9 : Fouille de données temporelles

Organisation : Georges Hébrail, René Quiniou

EGC 2006

Atelier 9 : Fouille de données temporelles

Dans de nombreux domaines applicatifs, tels la biologie, la santé, les télécommunications, la vidéo-surveillance, etc., des données sont enregistrées de manière continue. Les bases de données concernées peuvent atteindre des tailles gigantesques ou ne retenir que les données les plus récentes. Les données représentent, par exemple, les valeurs prises par des variables mesurées à intervalles réguliers ou des événements se produisant de manière irrégulière. Dans la plupart des cas, les données présentent un caractère temporel qu'il est intéressant de caractériser : relations entre les tendances de plusieurs variables, relations temporelles entre occurrences de certains types d'événements, etc. L'exploitation de cette dimension temporelle introduit une complexité supplémentaire dans les tâches de fouille de données et d'extraction de connaissances. Ainsi, il faut tenir compte

- des aspects métriques ou symboliques des relations temporelles traitées,
- de l'irrégularité ou du manque de synchronisation des mesures,
- du volume des données à traiter,
- de la fugacité des données et de la nécessité d'un traitement en temps-réel,
- de la nécessité/possibilité ou non d'un encodage explicite des relations temporelles des données avant leur exploitation,
- de la granularité temporelle et du caractère hétérogène des types des données pouvant avoir un impact sur les motifs susceptibles d'être découverts,
- de la nature et de l'utilisation des connaissances extraites,
- de la possibilité de prendre en compte la connaissance générale sur le domaine.

Les approches généralement suivies consistent, soit à étendre les approches classiques de la fouille de données pour prendre en compte la dimension temporelle, soit à proposer de nouvelles solutions et algorithmes appropriés aux données temporelles. Dans les deux cas, elles doivent tenir compte de la complexité des algorithmes utilisés et de leur possibilité de "passer à l'échelle".

L'objectif de cet atelier est de rassembler des chercheurs, du domaine académique ou de l'industrie, travaillant sur des problèmes cités ci-dessus ou sur des applications confrontées à ces problèmes. Parmi les 14 communications soumises, 11 ont été retenues et 3 ont été rejetées car un peu éloignées des thèmes de l'atelier.

Organisateurs

- Georges Hébrail (ENST Paris) hebrail@enst.fr
- René Quiniou (IRISA/INRIA Rennes) Rene.Quiniou@irisa.fr

Comité de programme

- Fabrice Clérot (FT R & D Lannion)
- Marie-Odile Cordier (IRISA/Université de Rennes 1)
- Florian De Vuyst (Ecole Centrale de Paris)
- Michel Dojat (Unité mixte INSERM-UJF U594 "Neuroimagerie Fonctionnelle & Métabolique" Grenoble)
- Catherine Garbay (CLIPS-IMAG Grenoble)
- Georges Hébrail (ENST Paris)
- René Quiniou (IRISA/INRIA Rennes)

Sommaire

MULTIPLE TIME SERIES: NEW APPROCHES AND NEW TOOLS IN DATA MINING APPLICATIONS TO CANCER EPIDEMIOLOGY

Mireille Gettler Summa*, Frédérick Vautrain****
Laurent Schwartz**, Mathieu Barrault****
Jean Marc Steyaert***, Nicolas Hafner****

* Centre de recherche en Mathématiques de la Décision
Université Paris Dauphine 1 Pl. du Ml de Lattre de Tassigny 75016 Paris France
Summa@ceremade.dauphine.fr
** Service de Radiothérapie, Hôpital Pitié Salpêtrière 47 boulevard de l'Hôpital Paris
*** LIX – Ecole Polytechnique
steyaert@polytechnique.fr
**** Isthma, 14-16 rue Soleillet 75020 Paris France
vautrain@isthma.fr, barrault@isthma.fr, hafner@isthma.fr

Abstract. Innovating data mining tool for complex data provide new and comprehensive viewpoints to the epidemiologist who can derive original results and perform interactive treatments. New algorithms working in a multidimensional space on curves (such as a set of multiple time series in our study) or on discrete distributions, without loss of information as it is the case with more classical encoding techniques (e.g. by data aggregation: means, quintiles etc.) have been studied and have been implemented in Delta Suite software. Each cell of the table under study is a function (in this case time series). Delta Suite software is applied for comparing epidemiological trends w.r.t. time, on two illustrative studies of the WHO data:
- the simultaneous visualization and exploration of the time series for cancer death rates on many entries, geographical information, temporal data, age, sex and pathologies.
- the geographic comparison of lung cancer evolutions over 21 years by building automatic classifications.

1 INTRODUCTION

1.1 A new exploratory approach for time series

The data of the death certificates that the World Health Organization (WHO) provides officially on its servers contain five main entries: death main cause, sex, age class, certificate death country (state or province in some cases), death year. These five variables define therefore a multidimensional space in which the statistical investigations are performed; a great number of multidimensional arrays (usually 2 entries, but quite often 3 and even 4) (Huang 1999) can be built in order to perform Multiple Array Analysis (Pardoux 2001).

Such an approach is seldom used in epidemiology where the main tools are based on the juxtaposition of monovalued statistics (Groupe d'étude et de réflexion interrégional 1996).

Innovating data mining tools for complex data (Gettler- Summa Pardoux 2000) provide new and comprehensive viewpoints to the epidemiologist who can derive original results and perform interactive treatments. They allow working in a multidimensional space on curves (such as a set of multiple time series in our study) instead of sets of single curves (Last and al. 2004) or on discrete distributions (Bock Diday 2000), without loss of information as it is the case with more classical encoding techniques (e.g. by data aggregation: means, quintiles etc.). Therefore the data can be kept and integrally treated. These original methods have been applied using Delta Suite software which was developed by the Research Centre for Decision Mathematics of Paris Dauphine and by Isthma Company (France) on two illustrative studies of the WHO data:

- The simultaneous visualization and exploration of the temporal series for cancer death rates on many entries, geographical information, temporal data, age, sex and pathologies.
- The geographic comparison of lung cancer evolution over 30 years by building automatic classifications.

1.2 Questions in Cancer epidemiology methodology

It appears that the overall cancer death rate has been about constant in the past 30 years, both in Europe and in the United States, with the notable exception of mortality among children and young adults. At the same time, there has been a global rise in mortality by lung cancer and slightly less important rises of melanoma and brain tumor, whereas one observes a concomitant decrease for cancers of the stomach, esophagus and head-and-neck (Gettler Summa and al. 2001).

Smoking explains certainly the increased lung cancer death rate, but in the meanwhile there is a partial decrease in other malignancies such as head-and-neck tumors which are also mainly caused by tobacco. The reasons for these multiple and apparently contradictory shifts remain largely unknown. The goal of this paper is to address these cancers in cancer epidemiology using statistical and mathematical techniques rarely used in the medical community.

2 Data description

2.1 The initial array

The data analyzed in the present work come from a data base of deaths caused by cancer which has been extracted from files obtained from the World Health Organization (WHO1981-2000). We extracted the number of deaths caused by cancer for 122 countries, both sexes, 21 age-classes (ranging from birth to 100+), 13 cancer types and 50 years of data. The original data were organized in 1,521,828 lines, 63% of which had missing data and 14,074 of which were containing out of the range figures. Odd data were processed by the 'missing value' menu in Delta software which is described further. 574,200 lines array have been eventually validated.

2.2 Initial data pre-processing

Since we want to compare country evolutions, we have to normalize the data over the age-classes; for this purpose, we compute new homogenous ASRs (Average Standardized Ratios) that are commonly used in epidemiology (National Center for Health Statistics 1998). These ratios are modified by so-called direct or indirect adjustments in epidemiology (SEGI reference). Since we want to compare temporal evolutions between countries, we have to synchronize the various data in order to have the same periods of time for all the countries. We have to reconstruct a common denomination for the various types of cancer since the nomenclature, ICD, varies along time. These evolutions can be found in the 'International Classification of Diseases' (ICD): we have taken into account 5 classifications including the 10th ICD from 1999 (Anderson and al. 2001). 155 countries have been retained globally at the beginning of our study since their data can be exploited on common periods of time. Furthermore, we have proved for some countries that the data variations follow a Gaussian law: what is effectively measured (in the case of Iceland, e.g.) is more the average value of mortality than real fluctuations. Such countries were excluded.

3 DELTA: New Approaches in Data Mining

The DELTA platform, allows incorporating variation on a single data: probability distribution, functional variation etc.: multiple time series are characteristic examples of complex data, where variation consists of time evolution. Delta allows managing, editing and analyzing such complex data in a multidimensional statistical framework.

3.1 DELTA GRAPHIC data manager and editor in the context of epidemiology

DELTA GRAPHIC allows a number of operations on a classical data array in order to build new data arrays, significantly more complex since it is possible to define functions as entries of these new arrays. It is composed of a graphic editor on complex data which allows a visual exploratory analysis, which is completed by statistical numerical indices (regression coefficients etc.) and data management options (sorting, filtering etc.)

3.1.1 The SYNTHESIS operator: taking into account the variations of basic data and building the complex data base

The initial data analyzed in this study concern 51 countries and 21 years of reporting over 9 pathologies and 16 age classes for both sexes.

country	Year	cause of death	sex	ASR at all ages	...	ASR at age 45 49 years	...	ASR at age 85 89 years
England and Wales	1980	Malignant neoplasm of stomach	female	17.91536	...	0.24453	...	0.98982
England and Wales	1980	Malignant neoplasm of stomach	male	26.86425		0.56563		1.33101
...								
Italy	1999	Malignant neoplasm of trachea, bronchus and lung	female	19.02513	...	0.43480	...	0.42265
Italy	1999	Malignant neoplasm of trachea, bronchus and lung	male	92.82309		1.57175		2.36752

TAB. 1 – *Initial data*

We have built time series by synthesizing the observed data according to time (Year); we obtained new statistical units where the variables are temporal curves (functions of the time variable), such as shown below.

FIG. 1 - *Automatic generation of time series for countries, diseases and males, females*

One can equally synthesize raw data according to the variable country, thus obtaining new variables in the shape of discrete distributions defined on the domain of countries or on the variable sex, yielding binary distributions; the age classes could also be used as the synthesis variable. The synthesis process produces an array of complex data in which statistical units are described in terms of generalized variables: continuous functions, intervals, distributions, lists, discrete mapping. Figure 2 presents a sample of a complex array

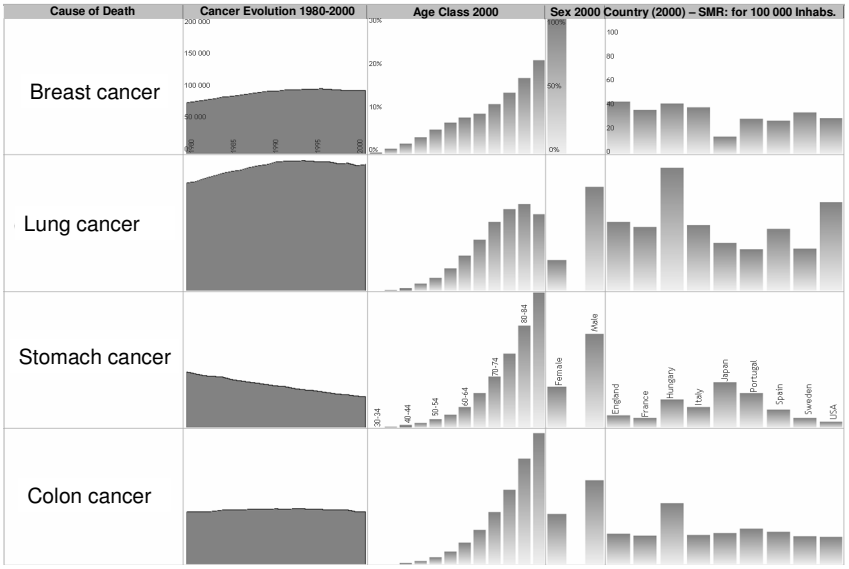


FIG. 2 - *Joint edition of variables from various types (discrete applications, curves)*

3.1.2 Options for the re-encoding of complex data

The encoding of data is a key step in any statistical analysis: the methods, hence the results ultimately depend on the values that have been retained to represent the initial information.

In the case of a time series, it is usual, during the exploratory phase, to smoothen, deseasonalize or decompose the series. DELTA offers these operators on any function obtained during the SYNTHESIS process. Another re-encoding allows making more precise a visual feeling on the variations of a curve, by means of elementary numerical operators. A polynomial approximation of degree 1 was performed (higher degrees can be used) the coefficients of which (steepness 'a' and origin step 'b') characterize each statistical unit and allow an easy interpretation of the cancer growth rate. Other re-encoding options have been implemented according to the type of the complex data: polynomial approximation, Fourier transformation, regression, BSplines and other smoothing options etc.

3.1.3 Filtering options

With DELTA, the filtering options are queries that are specified from a dedicated predicate editor. They are necessary for any interactive exploratory search.

For example the result of a double filter onto the curves of the initial data, in order to select all the countries and diseases such that their death curves for the age class 50-54 are above the similar death curves for the age class 80-84, reveals that this situation occurs for 13 countries over 18, mainly for head and neck, esophagus and lung cancers.

3.1.4 The multiple time series editor

DELTA contains a complex data worksheet editor for a visual screening of the data.

Graphic scales Four modes are possible to render a cell:

- "Local" mode, each cell has its own scale (and cannot be visually compared to another).
- "Global" mode, the cells of the same column have a common scale and can be compared to the others of the same column. But it is not possible to compare the columns.
- " Common" mode, all the cells have the same scale and can be compared to the others.
- "Manual" mode, the user may emphasize some visual effects. The scales are defined by the user at convenience on both axes.

The Cohort approach The data manager allows to build, among other arrays, cohorts and to visualize them (Breslow 1987). Let us show an example on the WHO data.

Let us imagine tracking persons born between 1916 and 1920. Those who died in 2000 pertain to the death statistics of the 80-84 age class, whereas a death in 1980 would put them in the age class 60-64. Figure 3 exhibits 8 cohorts specified in rows: people born during 4 periods (1920-1924, 1925-1929, 1930-1934 and 1935-1939), two countries (England and Japan), 4 diseases (head and neck, prostate, stomach, lung); in each cell, a discrete mapping: raw death ratios per disease. So doing, one can read, at the intersection of row England-1930-1934 and of column age-class-60-64, the quotient of the number of deaths of this age class during the years 1990-1994 and the population of this age class at this period.

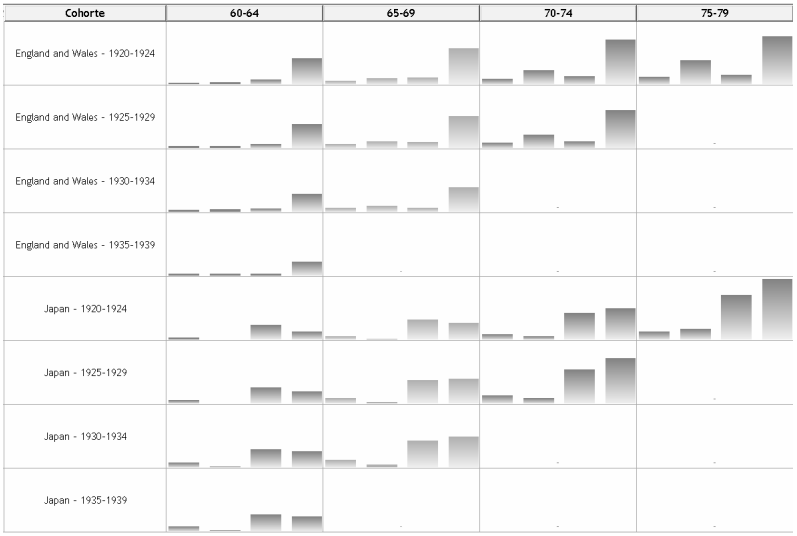


FIG. 3 - *Visualization of age class cohorts*

Missing data management Missing data are dealt with by DELTA and are denoted ''n/a''.

In some circumstances it is possible to re-encode them by a number of smoothing and interpolating functions, especially on discrete variables. 36 countries have thus been interpolated and/or extrapolated, depending on the reason for which data were missing (no census data for a year, delayed ICD, etc.).

3.1.5 Statistical results and their interpretation of a study Countries/Cancers/Years

We now present the major conclusions of a case study that will illustrate DELTA functionalities for elementary statistical mining. This study works out statistics on 51 industrialized countries, with a western style of life, based on 21 years of data concerning 10 types of cancer and 16 age classes (each class being of 5 years duration).

The conclusions are issued from the analysis of graphs similar to figure 1. They first come from a visual analysis of the graphics and then are formally validated by the computation of various numerical indicators obtained from the curves, using DELTA tools.

From the resulting figures, one can assert the following principal facts:

- the epidemiology of ''Cancer'' is NOT the epidemiology of each ''type of cancer''; for instance ''all cancers'' for a given country shows some trends that are contradictory according to the pathology (lung cancer is increasing while stomach cancer decreases); for a given pathology, very different trends can be observed depending on the country (for the lung cancer one can see an important increase in Hungary contrasting with an equivalent decrease in England).
- The proportion of oldest people is the highest for prostate cancer and smallest for lung cancer; it increases globally with years, especially for lung cancer.
- Lung cancer remains the most important in the number of deaths; stomach cancer has diminished in a significant manner.
- All tobacco linked cancers do not follow the same evolutions

3.2 DELTA METRICS multidimensional time series processor in the context of epidemiology

In order to estimate in a same analysis the influence of the various dimensions (age, country, sex, pathology) on the trajectories, a multidimensional approach for curve analysis is necessary. DELTA METRICS was used to perform factorial analyses and unsupervised classifications. We present in this study the clustering phase results.

3.2.1 Automatic classifications for the geographic analysis of lung cancer from 1969 to 1998

DELTA contains several tools for performing unsupervised classifications: one can use a 'k-means' approach or build hierarchical classifications for temporal series. These algorithms have been designed on recent results and generalize previously known methods for quantitative or qualitative data to the new universe of functions (Ramsey 1997).

Methodology. One of the main methodological elements which distinguish the classical classification methods of mono-valued data from the classification of functional data (probability distributions, time series, etc.) is the definition of the metrics over the data space.

This metrics can be defined in two steps. In the first step one computes the dissimilarities between the curves representing two distinct statistical units on a single variable. In the second step, one computes the dissimilarities aggregates over all the variables.

Dissimilarities computation. The classical dissimilarities have been implemented in DELTA, but they have been adapted to the structure of the data dealt with on the platform.

In the case of discrete applications, three distances between the graphs have been implemented: the L_1 distance, the L_2 distance and the L_{max} distance.

In the case of continuous functions, defined on subsets of RxR, values intermediate between the sampling points which are obtained by a linear interpolation have been used.

When the envelop of the sample abscissa is not the same for the two mappings, we use by default the intersection of the supports. (Other options are possible, such as giving the value 0 when the mapping is not defined).

$$Diss(D_1, D_2) =$$

$$\frac{1}{Xs - Xe}\int_{Xs}^{Xe}\left(Y^1(t) - Y^2(t)\right)^2 dt$$

$$= \frac{1}{Xs - Xe}\sum_i \int_{X_i}^{X_{i+1}} \left(\left(Y^2_{i+1} - Y^2_i\right)\frac{t - X^2_i}{X^2_{i+1} - X^2_i} + Y^2_i - \left(Y^1_{i+1} - Y^1_i\right)\frac{t - X^1_i}{X^1_{i+1} - X^1_i} - Y^1_i \right)^2 dt$$

$$Xs = Max(Min(X_1), Min(X_2))$$

$$Xe = Min(Max(X_1), Max(X_2))$$

For interval data (the value is a real interval) the Hausdorff distance was implemented

In the case of finite discrete distributions for the same variable, to build a hierarchical classification, the formula used for each variable is inspired from the chi2 distance.

Hierarchical classification of complex data with DELTA. DELTA provides tools for building ascending hierarchical classification of complex data. The algorithm presents the remarkable feature that the final result proposes an order on the terminals which optimizes the distance to the initial matrix of their dissimilarities (Pak 2005).

The menu presents various possibilities at the three levels of formula choice for the algorithm of hierarchical classification for complex data. The aggregation links currently

implemented are δ_{Max} , δ_{Min} and the mean link. The distances between statistical units for the same variable are listed at 3.2.2.1 and the aggregation functions between variables can be the maximum, the geometric mean or the weighted quadratic sum.

Main results. We now present and comment a hierarchical classification of the ASRs concerning lung cancer for 51 countries over 21 years.

The classes of curves have been computed by the "Minimum link" method and yield a 8 classes partitions; the countries are ordered as it can be seen on the following dendrogram.

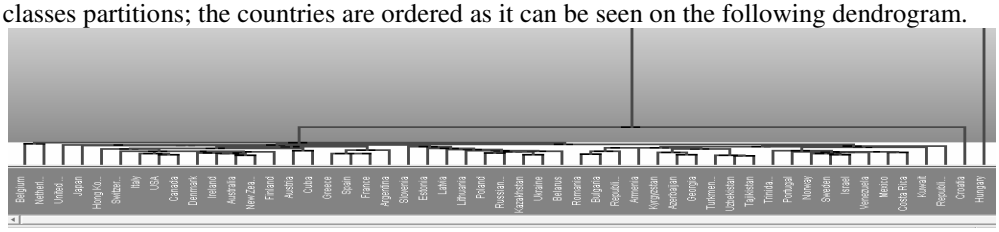


FIG. 5 - *Classification of countries*

We have chosen eight homogeneous groups among all possible partitions:

- Netherlands, Belgium, United Kingdom (value: high then medium, trend: decreasing)
- Italy, USA, Canada, Ireland, Australia, New Zealand, Austria, Switzerland, Finland, Denmark, Hong Kong (value: medium-high, trend: steady then decreasing)
- France, Spain, Greece, Argentina, Cuba (value: medium, trend: steady)
- Russian Federation, Estonia, Poland, Kazakhstan, Ukraine, Lithuania, Latvia, Belarus, Slovenia, Romania, Bulgaria, Rep. of Moldavia, Armenia (value: medium then high, trend: increasing)
- Trinidad & Tobago, Tajikistan, Uzbekistan, Turkmenistan, Georgia, Azerbaijan, and Kyrgyzstan (Value: very low)
- Portugal, Norway, Israel, Sweden (low value, trend: steady then slightly increasing)
- Kuwait, Costa Rica, Mexico, Venezuela (Value : Low; trend: slightly decreasing)
- Rep. of Korea, Croatia, Hungary (Highly increasing)

Japan has particular features: low and steady values for 40-74 then slightly increasing (value and trend).

We have next computed the mean curve for each class, then the distances (see 3.2.2.1) of each time series to the mean of its class and the paragon for the curves of each class is the country, which has the closest time series to the mean (height paragons plus Japan on Figure 6)

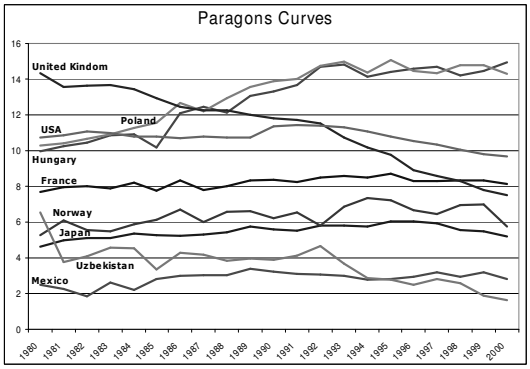


FIG. 6 - *9 typical time series for lung cancer epidemiology 1980-2000*

In addition to the trend typology being classified in 6 groups, the major observation is that the paragon curves of the "western style" countries migrate towards an interval of values much smaller in 2000 than twenty years before, with the "exceptions" of Hungary and Russia. This observation is coherent with and refines the result shown in Figure 7: the inter-geographic variation coefficient of lung cancer for these countries decreases between 1980 and 2000.

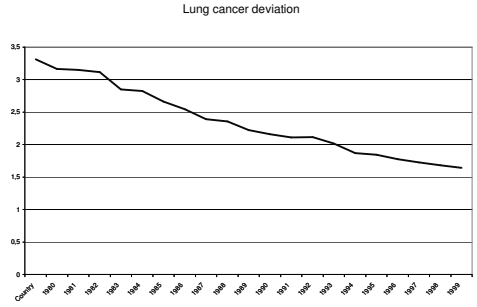


FIG. 7 - *Evolution of Lung cancer inter-geographical deviation between 1980 and 2000*

4 CONCLUSION AND PERSPECTIVES

From a methodological point of view, this paper demonstrates the bonus that the tools of multidimensional analysis bring into the domain of epidemiology on huge mass of information in the exploratory phases. Furthermore, the epidemiological data are temporal by essence and DELTA SUITE software gives the opportunity to take time into account. The main innovation presented in these studies is the fact that each cell of the table under study is a function (in this case time series). We have thus been able to study and to compare epidemiological trends w.r.t. time in several dimensions (geographic, age, disease type, sex) and, particularly, to build typologies for countries in terms of the mortality evolutions of various cancer types since 20 years. The results show stability and at the same time some clear evolutions in the mortality induced by cancer. If the global mortality, all the sites being put together, stands almost unchanged, the mortality by sites shows big variations. The exploratory data mining of large volumes of mortality data by cancer in the world is therefore useful to reveal the links between various factors or groups of factors. It should also allow estimating the respective weights of these factors on mortality by cancer. In order to achieve this objective, we have to develop new specific modules for the treatment of complex time series that will extend DELTA capabilities: quantitative modeling, forecasting tools in particular. As a natural extension in the epidemiological study, we should add a number of new variables addressing new potentially relevant domains: life style, food, growth speed, psychological trends, etc.

5 BIBLIOGRAPHY

Anderson R.N., Arialdi M.N.,Hoyert D.L., Rosenberg H.M. (2001) *Comparability o f cause of death between ICD-9 and ICD-10: preliminary estimates* National Vital Statistics Reports Volume 49, Number 2

Breslow NE, Day NE. (1987) *Rates and Rate Standardization The Design and Analysis of Cohort Studies* Statistical Methods in Cancer Research, Vol. II, IARC Scientific Publications No. 82, Lyon, International Agency for Research on Cancer 48-79.

Bock H.-H, Diday E.(2000) Analysis of Symbolic Data , Editions Springer, 2000

Huang F. (1999) *Knowledge Discovery in Data Mining highly multiple time series of astronomical observations* CTAC99 Camberra Australia

Gettler-Summa M. , Pardoux C.(2000) *Symbolic Approaches for Three-way Data Analysis of Symbolic Data* , Editions Springer, 342-352

Gettler Summa M., Goupil-Testu F., Goupil J., A.Sasco A., Schwartz L., Vautrain F., (2001) Application de l'Analyse de données aux causes de décès de 1950 à 1997 ; position de la mortalité par tumeur Actes du colloque de la Société Francophone de Classification,

Groupe d'étude et de réflexion interrégional (1996) *L'analyse des données évolutives Méthodes et applications,* Edition technip

Last M., Kandel A., Bunke H. (2004) *Data Mining in time series data bases* World scientific

National Center for Health Statistics. (1998) *Report of the workshop on age adjustment*. Vital and Health Statistics. Series 4. No. 30. December 1998.

Pak Kutluhan Kemal (2005) *Classifications Hiérarchique et Pyramidale Spatiales; Nouvelles Techniques d'Interprétation*, Thèse de Doctorat Université Paris dauphine

Pardoux. C. (2001) *Analyse exploratoire d'une suite de tableaux de données indicés par le temps.* La Revue de MODULAD.. n° 47. 67-102.

Ramsay J.O., Silverman B.W. (1997) *Functional Data Analysis* Springer Verlag

WHO (1981-2000) International histological classification of tumours, 2nd ed. Geneva, World Health Organization

Aknowlegments to Ecole Polytechnique FX Conseil and Phillips Morris International

Résumé

Des résultats innovants en fouille de données complexes fournissent des approches originales pour les épidémiologistes qui bénéficient de traitements interactifs pour aborder les données d'un point de vue nouveau et en tirer des résultats. L'étude présente des algorithmes qui travaillent sur des espaces multidimensionnels de fonctions (ici des chroniques) ou sur des distributions discrètes (sans perte d'information sur les valeurs comme c'est le cas des codages habituels par agrégation – quantiles etc.) ; ils ont été implémenté dans le logiciel DELTA Suite : chaque cellule d'une table étudiée contient une donnée complexe (en particulier une série temporelle). Delta Suite est utilisé dans deux études épidémiologique de l'évolution des cancers dans le temps et dans l'espace: en un premier temps la visualisation simultanée et l'exploration des chroniques de taux de mortalité par cancer sur plusieurs entrées conjointement (géographiques, temporelles, âge, sexe et pathologies) puis en un deuxième temps la comparaison géographique des évolutions du cancer du poumon pour 51 pays et 21 années par classification automatique.

Agrégation d'alarmes faiblement structurées

Alexandre Vautier*, Marie-Odile Cordier*,
Mireille Ducassé**, René Quiniou***

*Irisa/Université de Rennes 1, **Irisa/Insa, ***Irisa/Inria
Campus de Beaulieu 35042 Rennes Cedex, France
{Alexandre.Vautier,Marie-Odile.Cordier, Mireille.Ducasse,Rene.Quiniou}@irisa.fr

Résumé. La contribution principale de ce document[1] est une approche plaçant l'opérateur au coeur de l'analyse de journaux d'alarmes faiblement structurées en lui permettant d'utiliser ce qu'il sait, même si ses connaissances sont partielles, et sans le submerger d'informations. Des motifs temporels structurés sont extraits par agrégation d'alarmes généralisées et corrélation se basant sur la date des alarmes et sur la similarité d'attributs autres que la date. L'approche est appliquée aux alarmes produites par un concentrateur VPN (Virtual Private Network). Une étude de cas montre comment 5000 d'alarmes peuvent être regroupées en 50 motifs.

1 Introduction

La profusion des alarmes produites par les systèmes informatiques rend l'analyse des journaux d'alarmes de plus en plus difficile pour l'opérateur. Ces journaux sont générés, par exemple, par des composants réseau, des dispositifs de détection d'intrusion ou le système d'exploitation. Ils sont complexes et difficiles à appréhender complètement par un opérateur qui n'en a, en général, qu'une connaissance partielle. Confronté à cette masse de données, l'opérateur est, la plupart du temps, contraint de ne s'intéresser qu'à certaines alarmes qu'il sélectionne de façon plus ou moins pertinente. C'est, malgré tout, l'opérateur qui doit comprendre les informations contenues dans les journaux et qui sait de quoi il a besoin. Peu d'outils permettent de le seconder dans cette tâche risquée. L'augmentation grandissante des échanges d'informations, d'une part, et de la demande en sécurité, d'autre part, réclame la conception et la mise en œuvre de tels outils.

Les alarmes ainsi produites sont souvent faiblement structurées. Elles sont composées de champs mais les valeurs de ces champs sont souvent simplement copiées dans un fichier sous forme de chaînes de caractères sans marqueur syntaxique, reflétant le fait que l'analyse automatique de ces alarmes n'a pas été prévue lors de leur génération. Cet état de fait évolue, au moins pour les applications critiques, en particulier grâce à l'utilisation de XML. Cela étant, comme la structuration des données d'audit demande un certain effort, il reste encore de nombreuses applications où un journal est simplement un texte faiblement structuré. L'opérateur peut souvent paramétrer, au moins partiellement, les systèmes de génération des alarmes mais il en a rarement la maîtrise totale. De plus, c'est souvent à la suite d'un incident qu'il se penche sur un journal. Dans ce cas, il est trop tard pour paramétrer le système et il lui faut faire avec ce qui existe.

La contribution principale de ce document est de proposer une approche permettant de placer l'opérateur au coeur de l'analyse de ces journaux en lui permettant d'utiliser ce qu'il sait, même si

[1]Ces travaux sont partiellement financés par France Telecom R&D

ses connaissances sont partielles, et sans le submerger d'informations. Des motifs temporels sont extraits de journaux faiblement structurés. Ces motifs sont des agrégations d'alarmes généralisées et corrélées. Dans une première étape, certains attributs d'alarmes sont extraits en s'appuyant sur des expressions régulières spécifiées par l'opérateur. Comme l'opérateur n'a pas forcément la connaissance pour spécifier directement les bonnes expressions, la spécification peut être affinée en plusieurs itérations. Dans une deuxième étape, les alarmes structurées issues de l'étape précédente sont regroupées. Les alarmes sont corrélées temporellement, en se basant sur la date des alarmes, et rationnellement, en se basant sur la similarité entre attributs autres que la date. On obtient ainsi une partition du journal.

Notre approche est appliquée aux alarmes produites par un concentrateur VPN (Virtual Private Network) de marque Cisco dans le réseau de France Télécom. Un concentrateur est un matériel réseau acceptant un grand nombre de connexions VPN simultanées. Dans ce cadre, les motifs temporels extraits constituent des modèles de connexion VPN présents implicitement dans le journal. Une connexion est composée de transactions, elles-mêmes constituées d'alarmes. Une étude de cas montre comment 5000 d'alarmes sont regroupées en 50 motifs.

L'extraction des attributs d'alarmes faiblement structurées est présentée dans la section 2. Le regroupement des alarmes est décrit dans la section 3. Les résultats d'une étude de cas sont présentés en section 4. Des travaux voisins sont discutés en section 5.

2 Extraction assistée des attributs

Les attributs sont générés à la suite de plusieurs interactions entre l'opérateur et un processus d'extraction automatique. À partir des résultats d'une extraction automatique et de ses propres connaissances, l'opérateur spécifie approximativement ou précisément la manière dont les attributs doivent être automatiquement extraits et le processus est réitéré. Les attributs générés sont utilisés dans la phase suivante (cf. section 3) pour construire des transactions.

Date	VPN	Type	Client	Groupe
Message				
05/09/2004 23 :11 :02.750	VPN-1	L2TP/46	82.83.84.85	
Tunnel to peer 82.83.84.85 closed, reason : Peer no longer responding				
05/09/2004 23 :12 :44.530	VPN-1	IKE/24	80.13.14.15	Group [Group1]
Received local Proxy Host data in ID Payload : Address 85.75.65.55, Protocol 17, Port 0				
05/10/2004 07 :46 :46.950	VPN-2	AUTH/37	20.21.22.23	
User [Unknown] Protocol [SNMP] attempted ADMIN logon.. Status : <ACCESS GRANTED> !				
06/02/2004 22 :37 :04.510	VPN-1	IKE/41		
IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer 81.71.61.51 local Proxy Address 85.75.65.55 , remote Proxy Address 81.71.61.51, SA (WindowsServer1)				
06/06/2004 15 :13 :45.640	VPN-1	IKE/41	81.251.55.54	Group [Group1]
IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer 81.251.55.54 local Proxy Address N/A, remote Proxy Address N/A, SA (N/A)				

FIG. 1 – *Exemples d'alarmes produites par le concentrateur VPN.*

Le journal étudié contient 1.262.117 alarmes issues du concentrateur VPN pendant environ 1 mois. Elles ont le format suivant : *date, nom de concentrateur VPN, type d'alarme, message* ainsi que les champs optionnels *adresse IP du client* et *groupe du client*. L'extraction des attributs est ai-

```
- (\b(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}
                  (?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b)(?:$|\W)
- (?:^|\W|0x)([a-fA-F0-9]{3,})(?:$|\W)
- User\s\[(\w*)\]
```

FIG. 2 – *Un ensemble $\mathcal{A}$ de trois expressions régulières : adresse IP constituée de 4 groupes de 2 à 3 chiffres, nombre hexadécimal supérieur à 99 et identificateur d'utilisateur.*

sée pour tous les champs, sauf pour champ *message* qui, bien que le plus riche, n'est pas structuré : l'information contenue ne dépend pas seulement du type de l'alarme associée mais aussi de l'état du concentrateur. Ils contiennent des attributs de différents types (adresse IP, nombres hexadécimaux ou décimaux, nom d'utilisateur, etc.). La figure 1 présente quelques exemples d'alarmes du journal. Par exemple, la première indique qu'un incident de type L2TP/46 a eu lieu le 9 mai 2004 à 23h11min2.75s indiquant que le tunnel du client 82.83.84.85[2] a été fermé car il ne répondait plus.

Dans un premier temps, l'opérateur établit un ensemble $\mathcal{A}$ d'expressions régulières qui représentent la forme générale des attributs à extraire. La figure 2 montre trois expressions régulières[3] permettant de rechercher des adresses IP, des nombres comportant trois chiffres ou plus et des noms d'utilisateur. Chaque alarme est convertie en une *alarme normalisée* composée d'une date, d'un type et d'une liste d'attributs. La date et le type sont extraits des champs date et type de l'alarme. Les attributs des champs adresse IP du client, groupe du client et message sont extraits en utilisant $\mathcal{A}$. De cette façon, l'opérateur exprime approximativement ses connaissances dans $\mathcal{A}$, qu'il va pouvoir ensuite affiner au vu des résultats de l'extraction automatique. Les attributs et un ensemble $\mathcal{S}$ de *signatures d'alarme* sont ainsi générés.

Date	Type	Attributs
05/09/2004 23 :11 :02.750	L2TP/46	82.83.84.85
05/09/2004 23 :12 :44.530	IKE/24	80.13.14.15, 85.75.65.55
05/10/2004 07 :46 :46.950	AUTH/37	20.21.22.23
06/02/2004 22 :37 :04.510	IKE/41	81.71.61.51,85.75.65.55
06/06/2004 15 :13 :45.640	IKE/41	81.251.55.54

FIG. 3 – *Alarmes de la figure 1 converties en alarmes normalisées*

Définition 1 (Signature d'alarme)
Une signature est un triplet (t, l, e) où t est un type d'alarme, l une liste de types d'attribut et e une liste de doublets (r, f) formé d'une expression régulière r et d'une fonction $f : l \rightarrow \{0, 1\}$. Les expressions régulières r sont appliquées sur la concaténation des champs adresse IP du client, groupe du client et message pour extraire des alarmes de type t les attributs dont le type appartient à l. La première expression régulière r de la liste e aboutissant à une extraction est utilisée. $\forall x \in l$ si $f(x) = 1$ alors l'attribut correspondant au type x est présent dans l'expression r, sinon $f(x) = 0$ et cet attribut est absent de l'expression r.

Une signature d'alarme synthétise la façon dont les attributs ont été automatiquement extraits pour un type d'alarme donné. Elle peut être utilisée pour extraire les attributs de ce même type d'alarme lors de l'extraction automatique. L'opérateur examine ensuite les signatures générées et évalue si l'ensemble $\mathcal{A}$ est suffisant. Si tel n'est pas le cas pour un type d'alarme donné, il

[2]Les noms d'utilisateur et les adresses IP apparaissant dans les exemples sont fictifs.
[3]Le format des expressions régulières est celui du package java.util.regex qui utilise la même syntaxe que perl.

peut soit modifier l'ensemble $\mathcal{A}$, en ajoutant ou modifiant des expressions régulières, soit modifier la signature de ce type d'alarme dans $\mathcal{S}$. L'extraction automatique suivante est basée soit sur la signature extraite pour S si l'opérateur l'a décidé, soit sur l'ensemble $\mathcal{A}$ des formes d'attribut.

La figure 3 montre les alarmes normalisées construites à partir des alarmes de la figure 1 et de l'ensemble $\mathcal{A}$ de la figure 2. Notons que l'attribut 82.83.84.85 est présent dans le champ client et le champ message de l'alarme de type L2TP/46 mais un tel attribut n'est représenté qu'une fois dans l'alarme normalisée correspondante. Dans la suite du document, le terme *alarme normalisée* est substitué par le terme *alarme* quand il n'y a pas d'ambiguïté.

L'exemple suivant montre comment l'opérateur peut interagir avec le processus d'extraction. Les deux alarmes du type «IKE/41» de la figure 1 sont normalisées en deux alarmes de même type (voir figure 3) dont les listes de types d'attribut sont différentes. Les alarmes d'un même type doivent avoir la même liste de types d'attribut. L'examen, par l'opérateur, de la signature (figure 4) générée par l'extraction automatique lui montre que les alarmes de type IKE/41 ont en fait quatre attributs (et non un ou trois). Il peut alors modifier cette signature comme le montre la figure 5.

Plusieurs interactions entre l'opérateur et l'extraction automatique peuvent s'avérer nécessaires pour générer des alarmes normalisées utilisables dans l'étape de construction de transactions. Ainsi l'opérateur introduit, de façon aisée et incrémentale, ses connaissances sur le journal et améliore l'abstraction du journal en vue de la phase de construction des transactions.

Expression régulière *(IP) remplace une expression régulière*	Attributs IP	IP	IP
_ IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer (IP) local Proxy Address (IP), remote Proxy Address (IP), SA \(WindowsServer1\)	1	1	1
(IP) Group \[Group1\] IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer (IP) local Proxy Address N/A, remote Proxy Address N/A, SA \(N/A\)	1	0	0

FIG. 4 – *Signature générée pour les alarmes de type IKE/41. La colonne Attributs décrit la liste l , ici trois attributs de type IP. Les lignes de la matrice décrivent les éléments (r, f) de la liste e.*

Expression régulière *(IP) remplace une expression régulière*	Attributs IP	IP	IP	IP
_ IKE Initiator : Rekeying Phase 2, Intf 2, IKE Peer (IP) local Proxy Address (IP), remote Proxy Address (IP), SA \(WindowsServer1\)	0	1	1	1
(IP) Group \[Group1\] IKE Initiator : Rekeying Phase 1, Intf 2, IKE Peer (IP) local Proxy Address N/A, remote Proxy Address N/A, SA \(N/A\)	1	0	0	0

FIG. 5 – *Signature modifiée par l'opérateur pour les alarmes de type IKE/41*

3 Construction des transactions

Une connexion VPN entre un client et un concentrateur se compose de transactions réseaux qui provoquent des rafales d'alarmes apparaissant de manière isolée et dispersée dans le journal. Le but est de reconstruire les transactions et les connexions. Les alarmes d'une transaction sont proches dans le temps et partagent des attributs, par exemple, ceux propres au client ayant initié la transaction. Or, a priori, l'opérateur ne connaît pas ces attributs. C'est pourquoi, dans un premier temps *tous* les attributs sont considérés afin d'obtenir des transactions primitives. Celles-ci vont

Date	Type	Attributs
05/13/2004 14 :15 :50.910	IKE/25	82.81.80.79*
05/13/2004 14 :15 :50.910	IKE/24	82.81.80.79*, 85.75.65.55
05/13/2004 14 :15 :50.910	IKE/66	82.81.80.79*
05/13/2004 14 :15 :50.910	IKE/75	82.81.80.79*, 3600
05/13/2004 14 :15 :50.960	IKE/49	82.81.80.79*, 53e9fac2
05/13/2004 14 :15 :50.960	IKE/120	82.81.80.79*, 5da6fd05
05/13/2004 14 :15 :53.960	IKE/170	82.81.80.79*, d81626c
05/13/2004 14 :16 :18.140	IKEDBG/64	217.128.126.9*
05/13/2004 14 :16 :18.450	IKE/172	217.128.126.9*
05/13/2004 14 :16 :18.460	AUTH/12	483$^{\diamond}$
05/13/2004 14 :16 :18.560	AUTH/41	217.128.126.9*, 483$^{\diamond}$
05/13/2004 14 :16 :18.560	AUTH/13	483$^{\diamond}$
05/13/2004 14 :16 :18.630	IKE/79	217.128.126.9*, 6cf24368000000000f41
05/13/2004 14 :16 :23.320	AUTH/12	484^{+}
05/13/2004 14 :16 :23.430	AUTH/4	217.128.126.9*, 10.169.25.80, 484^{+}, user2

FIG. 6 – *Deux transactions primitives partitionnant une partie du journal*

ensuite servir de base à la construction des transactions, assistée par l'opérateur.L'objectif de cette construction est de réduire les informations présentes dans la synthèse présentées à l'opérateur sans pour autant réduire la quantité d'information. Ceci répond à l'objectif du Minimum Description Length (complexité de Kolmogorov, Vitanyi (2005)) i.e. la synthèse de taille minimale (vis à vis de l'opérateur) et perdant le moins d'information vis à vis du journal qu'elle résume.

3.1 Partition du journal en transactions primitives

Les transactions primitives sont extraites des alarmes normalisées du journal.

Définition 2 (α-transaction)
Une α-transaction est :

- soit, une séquence composée d'une alarme normalisée,

- soit, une séquence $< T, x >$ où T est une α-transaction et x une alarme telles qu'il existe une alarme y de T qui possède au moins n_a attributs en commun avec l'alarme x et l'intervalle de temps qui les sépare est inférieur à $maxGap$ (l'écart temporel maximum).

Définition 3 (Transaction primitive)
Une transaction primitive est une α-transaction maximale (aucune α-transaction la contient).

Les paramètres n_a et $maxGap$ sont fixés par l'opérateur. La figure 6 montre un exemple de découpage du journal en deux transactions primitives. Les attributs partagés qui permettent la construction de la transaction primitive sont identifiés par une même marque en exposant ($\diamond$,* ,$^{+}$).
.

Théorème 1
Pour tout journal d'alarmes normalisées et des paramètres n_a et $maxGap$ donnés, il existe un unique partitionnement définissant les transactions primitives représentant ce journal.

En faisant varier les paramètres n_a et $maxGap$, on peut générer un partitionnement comprenant autant de transactions primitives que d'alarmes normalisées jusqu'à un partitionnement constitué d'une seule transaction primitive couvrant tout le journal. Il faut donc souvent contraindre plus fortement le partitionnement afin de satisfaire le critère de MDL. C'est le rôle de la phase suivante qui permet de construire des modèles de transactions à partir des transactions primitives.

3.2 Construction des modèles de transactions

Nous définissons tout d'abord un modèle d'alarme afin de pouvoir définir un modèle de transaction qui correspond à une généralisation, soit des alarmes, soit des transactions, par extension de leur domaine au niveau des attributs. Une relation d'ordre partiel sur de tels modèles est ensuite introduite. Enfin, la génération des modèles de transaction est décrite.

Définition 4 (Modèle d'alarme)
Un modèle d'alarme est un couple (t, l) où t est un type d'alarme et l est une liste de variables. Une variable $v \in l$ possède un type t_v et son domaine d_v est contraint.

Il faut noter que le temps n'est pas représenté explicitement dans un modèle d'alarme.Le temps est considéré comme un élément permettant d'ordonner les alarmes au niveau d'un modèle de transaction. Un modèle d'alarme couvre (généralise) des alarmes appelées instances de ce modèle d'alarme. Par exemple, l'alarme (date = 05/13/2004 14 :19 :46.940, type = PPPDECODE/16, attributs = [80.14.52.129, user1, 85.75.65.55]) est une instance du modèle d'alarme (t = PPPDECODE/16, $l = [X, Y, Z]$) où X et Z sont des variables de type adresse IP qui correspondent respectivement aux adresses 80.14.52.129 et 85.75.65.55 et Y est une variable de type identifiant qui correspond à l'attribut user1. L'ensemble des valeurs que peuvent prendre les variables est restreint par leur domaine associé. Par exemple, au modèle précédent on peut ajouter la contrainte $X \in$ 80.14.*.* qui impose à la variable X de prendre ses valeurs entre $80.14.0.0$ et $80.14.255.255$.

Définition 5 (Modèle de transaction)
Un modèle de transaction est un doublet (S,V) où S est une séquence de modèles d'alarme et V est l'ensemble des variables apparaissant dans les modèles d'alarme de la transaction.

Type	Variables
IKEDBG/64	X
IKE/172	X
AUTH/12	Z
AUTH/41	X, Z
AUTH/13	Z,
IKE/79	X, W
AUTH/12	A
AUTH/4	X, B, A, C
$X[IP] \in$ 217.128. $*.*$, $B[IP], W[int], Z[int], A[int], C[String]$	

FIG. 7 – *Ce modèle de transaction généralise la seconde transaction de la figure 6. Les lignes du tableau sont les éléments de S et la dernière ligne décrit les variables de V et leur domaine.*

Un modèle de transaction couvre (généralise) des transactions appelées instances.Ainsi, la deuxième transaction de la figure 6 est une instance du modèle de transaction de la figure 7. Ce

Type	Variables
IKEDBG/64	X
AUTH/12	Z
AUTH/41	X, Z
AUTH/13	Z
$X[IP] \in 217.*.*.*, Z[int]$	

FIG. 8 – *Un modèle de transaction généralisant le modèle de transaction de la figure 7*

dernier comporte une variable X partagée par 5 modèles d'alarme. Une telle variable apparaissant dans plusieurs modèles d'alarme de la transaction contraint les attributs correspondant des instances à posséder un type (un domaine) identique.

De plus, une relation de généralité peut être définie sur les modèles de transaction. Ainsi, le modèle de transaction de la figure 8 est plus général que le modèle de la figure 7.

Définition 6 (Relation d'ordre partiel sur les modèles d'alarme)
Soient A_g et A_s deux modèles d'alarme. A_g est plus général que A_s si et seulement si le type de A_g est identique au type de A_s et si pour toute variable v_g de A_g il existe une variable v_s de A_s telles que le domaine de v_s est inclus dans le domaine de v_g.

Définition 7 (Relation d'ordre partiel sur les modèles de transactions)
Un modèle de transaction $T_g = (S_g, V_g)$ est plus général qu'un modèle de transaction $T_s = (S_s, V_s)$ si et seulement si S_g est une sous-séquence de S_s telle que chaque alarme de S_g est plus générale que l'alarme correspondante de S_s.

Les modèles de transaction sont générés à partir du partitionnement décrit dans la partie 3. Les transactions primitives contenant des alarmes de type identique et des attributs éventuellement différents mais de même type sont généralisées en un modèle unique : le modèle le plus spécifique généralisant ces transactions (aussi appelé *lgg* - least general generalisation Plotkin (1970)). Ainsi, les types des modèles d'alarme de ce modèle sont identiques à ceux des transactions primitives et les domaines des variables du modèle sont aussi contraints que possible.

Par exemple, le modèle de la figure 8 est le lgg des deux transactions de la figure 9. Le domaine de la variable X du lgg est contraint à être le plus petit possible, soit $217.*.*.*$.

$t1$			$t2$		
Date	Type	Attributs	Date	Type	Attributs
...	IKEDBG/64	217.128.16.9	...	IKEDBG/64	217.56.200.1
...	AUTH/12	483	...	AUTH/12	513
...	AUTH/41	217.128.16.9, 483	...	AUTH/41	217.56.200.1, 513
...	AUTH/13	483	...	AUTH/13	513

FIG. 9 – *Le modèle de transaction de la figure 8 est le lgg des transactions primitives t_1 et t_2.*

4 Étude de cas et perspectives

Les 1.262.117 alarmes du journal du concentrateur VPN ont été entièrement normalisées après trois passages de l'extraction automatique des attributs. 37 signatures générées automatiquement

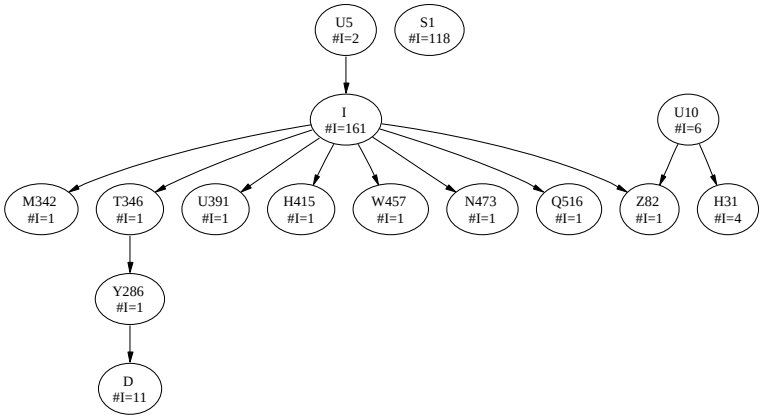


FIG. 10 – *Une partie du graphe des modèles de transaction extraits à partir des 5000 premières alarmes du journal (*$maxGap = 10s, n_a = 1$*)*

ont dû être modifiées et 4 formes d'attribut ont été ajoutées aux expressions régulières de la figure 2. Les modifications de signatures font apparaître qu'un dispositif d'inférence grammaticale pourrait être utile. En effet, en corrélant les expressions régulières d'une même signature, il serait possible d'associer automatiquement les attributs correspondants L'exemple d'interaction entre l'opérateur et la base de signatures présenté en partie 2 pourrait ainsi être automatisé.

La construction des transactions, expérimentée sur les 5000 premières alarmes du journal avec $maxGap = 10s$ et $n_a = 1$, a produit 628 transactions primitives qui ont généré 81 modèles de transaction. La figure 10 montre une partie du graphe généré à partir de cette construction : un nœud représente un modèle de transaction et affiche le nombre d'instances qui lui sont associées, un arc représente la relation de généralité. Si le modèle A est plus général que le modèle B alors il existe un arc de A vers B. On constate que le modèle de transaction nommé «I» (dont une instance est la première transaction de la figure 6) est très fréquent, de même que le modèle «S1» constitué de 8 modèles d'alarme. Ces deux modèles couvrent 41% de la partie de journal analysé.

80% des modèles n'ont qu'une seule instance et devraient être fractionnés en modèles plus petits ou fusionnés avec d'autres modèles. Certains modèles de transaction ont été générés par des variables qui n'étaient pas propres à un client mais plutôt à des serveurs, d'autres par des identifiants communs à toutes les alarmes, etc. Une analyse statistique sur les variables des modèles de transactions pourrait déterminer celles qui participent à la construction de transactions primitives mais qui ne doivent pas participer à la construction des transactions finales. Un modèle de transaction généré à partir de l'une de ces variables doit être fractionné. Enfin, certaines transactions primitives ne présentant aucun attribut commun ont des occurrences proches temporellement dans le journal pourraient être fusionnées.

5 Travaux voisins

La profusion des alarmes provenant des outils de détection d'intrusions ou d'attaques nécessite un pré-traitement pour fournir à l'opérateur de sécurité une vision synthétique des alarmes. De nombreux travaux présentent des méthodes visant à enrichir, structurer des informations provenant

d'alarmes ou corréler les alarmes ainsi obtenues. La méthode présentée ici étant proche de la fouille de données, nous nous focalisons sur des travaux ayant des liens avec ce thème.

La plupart des travaux n'abordent pas l'étape de préparation des données et considèrent que l'ensemble des attributs utilisés pour décrire les alarmes et les domaines associés sont connus *a priori*. Par exemple, les données d'alarmes fournies par les IDS sont parfois au format XML[4] ce qui permet une analyse syntaxique simple et l'intégration des données dans une base de données relationnelle, par exemple. Nous proposons une méthode d'assistance à l'opérateur pour constituer un ensemble d'attributs à partir de données faiblement structurées.

Une première étape pour le résumé ou la synthèse des alarmes consiste à les regrouper ou les agréger. L'agrégation locale vise à regrouper, sur une fenêtre temporelle de taille limitée, les alarmes issues de la même attaque. L'agrégation globale s'attache à caractériser un type d'attaque en regroupant, sur la totalité de la base, les alarmes ayant des attributs communs. Un groupe peut ensuite être *synthétisé* en méta-alarme puis *qualifié* en adjoignant des informations contextuelles, telles que le type de réseau ou les équipements observés, contribuant à améliorer leur sémantique. Différents critères basés sur une notion de similarité sont utilisés pour l'agrégation : similarité probabiliste (pondérée) basée sur la proximité des valeurs d'une partie (Dain et Cunningham (2001)) ou de la totalité des attributs (Valdes et Skinner (2001)), règles expertes définissant une similarité logique (Cuppens (2001)). Julisch (2003) agrège les alarmes contenues dans une table relationnelle selon leur type. Les attributs des alarmes d'un type donné sont généralisés selon des taxonomies fournies par des opérateurs tant que le nombre d'occurrences de l'alarme en cours de généralisation n'atteint pas un certain seuil. Morin (2004) systématise cette approche en traitant la corrélation d'alarmes comme un processus de recherche d'information : un treillis de concepts (Ganter et al. (2005)) représentant les alarmes contenues dans le journal est construit de manière incrémentale. Un concept de ce treillis est une méta-alarme synthétisant un sous-ensemble d'alarmes.

Dans notre approche, une alarme est associée à une transaction si elle possède un certain nombre d'attributs pouvant être mis en relation avec des attributs des alarmes de la transaction dans une fenêtre temporelle limitée. L'égalité des valeurs est généralement utilisée mais une notion de similarité étendue, par exemple basée sur une taxonomie telle que celle employée par Julisch, pourrait être introduite, ce qui nous rapprocherait de la méthode de Cuppens (2001). L'explicitation de relations logiques entre les attributs permettent, à notre avis, d'expliquer plus clairement les liens entre les alarmes à l'opérateur que des statistiques sur l'occurrence des attributs des alarmes.

L'étape d'agrégation est généralement suivie d'une étape de corrélation qui met en évidence des épisodes ou chroniques, encore appelés plans d'intrusion, constitués d'une ou plusieurs séquence d'alarmes élémentaires ou de méta-alarmes. Klemettinen et al. (1999) utilise en guise de corrélation d'alarmes la méthode proposée par Mannila et al. (1997) en fouille de données temporelles pour extraire des règles d'association et des épisodes sur la base de motifs temporels fréquents. Qin et Lee (2003) utilisent le test de causalité de Granger (1969) pour regrouper des méta-alarmes sans connaissances *a priori* des propriétés des alarmes. Cette approche paraît séduisante mais nécessite des données volumineuses contenant de nombreuses méta-alertes pour être utilisable. Dans notre approche, à toutes les transactions sont associées des statistiques sur leur occurrence mais c'est l'opérateur qui décide si une transaction doit être retenue ou non. Cuppens et Miège (2002) utilisent une notion de corrélation logique. Une alarme A est corrélée à l'alarme B si les conséquences de A rendent l'alarme B exécutable ce qui signifie qu'un ou plusieurs attributs des deux alarmes sont similaires. Cuppens et Miège prévoient une génération automatique des règles de corrélation d'alarmes mais le processus n'est pas détaillé. Nous utilisons également une

[4] http ://xml.coverpages.org/idmef.html

notion de corrélation *relationnelle* mais basée sur une relation de causalité s'appuyant uniquement sur la séquentialité temporelle, plus simple à mettre en œuvre de manière semi-automatique.

6 Conclusion

Ce document décrit une préparation assistée d'un opérateur humain d'un journal d'alarmes issues d'un concentrateur VPN. La principale difficulté est le manque de structure explicite et d'informations sur les attributs des alarmes. L'opérateur interagit avec les différents modules afin d'élaborer une synthèse du journal et d'acquérir de nouvelles connaissances sur celui-ci.

L'analyse effectuée n'est pas propre aux journaux de concentrateur VPN mais est généralisable à d'autres journaux faiblement structurés : syslog, trace de programme, etc. Les techniques utilisées, l'extraction des attributs à partir de leur forme, l'agrégation en transactions primitives composées d'alarmes corrélées relationnellement et la construction des modèles de transactions, sont toutes généralisables à d'autres problèmes.

Références

Cuppens, F. (2001). Managing alerts in a multi-intrusion detection environment. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*.

Cuppens, F. et A. Miège (2002). Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*.

Dain, O. et R. Cunningham (2001). Fusing a heterogeneous alert stream into scenarios. In *Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications*.

Ganter, B., G. Stumme, et R. Wille (2005). *Formal Concept Analysis, Foundations and Applications*. Springer Verlag.

Granger, C. (1969). Investigating causal relations by econometric methods and cross-spectral methods. *Econometrica 34*, 424–438.

Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security 6*(4).

Klemettinen, M., H. Mannila, et H. Toivonen (1999). Rule discovery in telecommunication alarm data. *Journal of Network and Systems Management 7*(4).

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

Morin, B. (2004). *Corrélation d'alertes issues d'outils de détection d'intrusions avec prise en compte d'informations sur le système surveillé*. Ph. D. thesis, INSA de Rennes.

Plotkin, G. (1970). A note on inductive generalization. In *Machine Intelligence*, Volume 5, pp. 153–163. Edinburgh University Press.

Qin, X. et W. Lee (2003). Statistical causality analysis of infosec alert data. In *RAID 2003*, Volume 2820 of *LNCS*, pp. 73–93. Springer Verlag.

Valdes, A. et K. Skinner (2001). Probabilistic alert correlation. In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, Volume 2212 of *LNCS*. Springer Verlag.

Vitanyi, P. M. B. (2005). *Algorithmic statistics and Kolmogorov's Structure Functions*, pp. 151–174. MIT Press.

Analyse factorielle d'opérateurs pour l'étude du mouvement: Application à la danse

Sullivan Hidot, Jean-Yves Lafaye, Christophe Saint-Jean

{shidot,jylafaye,csaintje}@univ-lr.fr

Université de La Rochelle
Pôle Sciences et Technologie
17042 La Rochelle Cedex 1

Résumé. Nous présentons une méthode relevant de l'analyse factorielle, qui permet d'une part d'analyser à des fins descriptives chacun des mouvements types de la danse classique, et d'autre part d'aider à reconnaître automatiquement le type d'un mouvement quelconque dont on ignore a priori la nature. Nous illustrons notre propos par des mouvements de danse issus de l'observation du Ballet Atlantique Régine Chopinot (BARC). Les techniques utilisées sont principalement l'analyse en composantes principales de la covariance relationnelle et l'analyse factorielle d'opérateurs, menant naturellement à une méthode de classification automatique.

1 Introduction

L'analyse du mouvement humain assisté par ordinateur est devenu un sujet de recherche récurrent et l'on peut citer les trois axes applicatifs majeurs (Moeslund et Granum, 2001) : la surveillance, le contrôle et l'analyse. Cet article s'inscrit dans cette dernière optique et s'intéresse plus particulièrement à l'étude du mouvement dansé. Nous proposons une méthode basée sur l'analyse factorielle d'opérateurs (AFO) (Foucart, 1984) et dérivée de l'analyse en composantes principales (ACP) (Joliffe, 1986) permettant la description, la reconnaissance et le classement de données multi-dimensionnelles et temporelles. Les données que nous étudions sont issues de figures de Danse Classique. Nous disposons pour cela de tableaux constitués d'un échantillon de réalisations d'un vecteur aléatoire réel. Les variables aléatoires sont les coordonnées spatiales x, y ou z des capteurs placés sur un danseur exécutant une figure chorégraphique. Afin d'appliquer notre méthode, chaque mouvement est repéré par une signature spécifique : l'opérateur de covariance relationnelle (Lebart et al., 2000) entre les variables. Comme on le verra par la suite, les avantages de cette signature sont multiples. Elle permet entre autres la prise en compte naturelle de l'ordre des captures ainsi que des liens inter-capteurs. Pour notre cas, le principe de l'AFO est de rechercher les plans principaux sur lesquelles les signatures des mouvements sont projetés afin d'en dégager des caractéristiques propres à chacun d'eux.
Un état de l'art sur les diverses approches concernant le clustering de données séquentielles est proposée dans (Dietterich, 2002). Dans le cas de trajectoires, les méthodes classiques dites

vectorielles sont inadaptées (Gaffney et Smyth, 1999). L'article (Gaffney et Smyth, 1999) présente une technique de clustering de séries temporelles basée sur l'algorithme EM. Dans (Zeng et Garcia-Frias, 2005), les auteurs proposent d'utiliser les modèles de Markov cachés en tenant compte du caractère dynamique des données. En outre (Oates et al., 2001) présente des résultats plus probants sur le clustering de séries temporelles en utilisant le Dynamic Time Warping. Malgré sa dimension artistique et culturelle, il n'y a que peu de travaux significatifs liant l'informatique et la danse. Parmi eux, on peut citer Thullier *et al.* (Thullier et Moufti, 1995) comparant la figure "rond de jambe" exécutée par des danseurs de différents niveaux. Dans la thèse (Chenevière, 2004), Cheneviere *et al.* s'intéressent à la classification des mouvements dansés par modèles de Markov cachées.

La suite de l'article s'organise comme suit. La section 2 est consacrée au formalisme de la méthode. Nous définissons les principaux outils mathématiques nécessaires : opérateurs de covariance relationnelle, Laplacien, métrique de Hilbert-Schmidt et matrice de Gramm. Nous explicitons également le sens concret de la métrique choisie. La section 3 est dédiée aux résultats et aux expérimentations de notre méthode appliquée au mouvement dansé. Dans la section 4, nous formulons quelques remarques sur la méthode proposée.

2 Théorie de la méthode proposée

2.1 Signature des mouvements

Soit $\mathbb{X} = (X_1, \ldots, X_p)$ un vecteur aléatoire constitué de p v.a. définies sur un espace probabilisé $(\Omega, \mathcal{A}, P)$ pour lequel on observe également la variable $\mathcal{T}$, date de la capture. Les variables X_i sont supposées centrées et de carré sommable. On appelle *individu* le vecteur de $\mathbb{R}^p$ constitué des $(X_i(\omega))_{i=1}^p$ pour $\omega \in \Omega$ fixé. Sur l'espace vectoriel engendré par ces individus, on définit une métrique M, symétrique définie positive. Pour calculer la covariance entre 2 v.a. X_i et X_j, on utilise la formule suivante :

$$Cov(X_i, X_j) = \frac{1}{2} \sum_{(\omega,\omega')\in\Omega\times\Omega} (X_i(\omega) - X_i(\omega'))(X_j(\omega) - X_j(\omega'))p_\omega p_{\omega'} \tag{1}$$

où $(p_\omega)_{\omega\in\Omega}$ est l'ensemble des poids des individus.

Définition 1 *Si Ω est fini de taille T, l'expression matricielle de la covariance standard est donnée par la relation :*

$$V = {}^t\mathbb{X}D\mathbb{X} \tag{2}$$

où D est la matrice diagonale d'ordre T constituée des poids p_ω des individus (i.e. métrique standard dans l'espace des variables).

Dans la réalité, on dispose d'informations complémentaires sur la proximité temporelle des individus et donc sur les éléments de Ω. Une relation binaire est une forme générale pour représenter ce genre d'information. Soit g_θ une variable aléatoire définie sur $\Omega \times \Omega$ par la relation :

$$g_\theta(\omega, \omega') = \begin{cases} 1 & \text{si } |\mathcal{T}(\omega) - \mathcal{T}(\omega')| \leq \theta \\ 0 & \text{sinon} \end{cases} \tag{3}$$

Plus θ est grand, et plus il y aura d'individus en relation pour g_θ. L'équation (3) induit un graphe noté G_θ et on a l'équivalence :

$$g_\theta(\omega, \omega') = 1 \Longleftrightarrow (\omega, \omega') \in G_\theta \tag{4}$$

Le graphe G_θ est symétrique par définition de g_θ et l'arête (ω, ω') est affectée du poids $p_\omega . p_{\omega'}$. On en déduit la formule de la covariance relationnelle par (1) en restreignant la double somme aux individus connectés pour G_θ :

$$Cov_\theta(X_i, X_j) = \frac{1}{2K_\theta} \sum_{(\omega,\omega') \in G_\theta} (X_i(\omega) - X_i(\omega'))(X_j(\omega) - X_j(\omega'))p_\omega p_{\omega'} \tag{5}$$

où K_θ est le facteur de normalisation, appelé *connectivité du graphe* :

$$K_\theta = \sum_{(\omega,\omega') \in \Omega \times \Omega} p_\omega p_{\omega'} g_\theta(\omega, \omega') \tag{6}$$

Définition 2 *Une expression matricielle simple de la covariance relationnelle est obtenue à partir de la matrice Δ_θ du Laplacien du graphe (Chartrand et al., 1991) :*

$$V_\theta = \frac{1}{K_\theta} {}^t\mathbb{X}\Delta_\theta\mathbb{X} \tag{7}$$

Contrairement à la métrique standard D, Δ_θ tient compte de la contrainte temporelle. En effet, par définition de g_θ, les individus éloignés avec un retard supérieur à θ n'auront pas d'influence pour le calcul de la covariance. On remarque aussi que si on choisit θ maximal (i.e. $\theta = T$), alors on retrouve la covariance standard définie en (2).
La covariance relationnelle est invariante par translation des individus. Δ_θ est une matrice symétrique semi-définie positive d'ordre T et de rang $T-1$ si le graphe est connexe (Bollodas, 1998). En général, si le graphe a k composantes connexes, alors le rang du Laplacien du graphe sera $T - k$. De plus, quelque soit θ, Δ_θ a pour vecteur propre $\mathbf{1} = {}^t(1, 1, ..., 1)$ associée à la valeur propre 0. L'espérance d'une variable étant un vecteur du sous-espace $< \mathbf{1} >$, il est inutile de centrer préalablement les variables pour déterminer la matrice de covariance relationnelle. Le nuage des individus sera translaté mais la disposition mutuelle entre chacun d'eux restera inchangée.

Définition 3 *Un mouvement $\mathbb{X}_k$ est identifié par la signature suivante :*

$$\Gamma_k = V_\theta^k M \tag{8}$$

où V_θ^k est la matrice de covariance relationnelle du mouvement $\mathbb{X}_k$ et M la métrique des individus sur $\mathbb{R}^p$.

Γ_k est un opérateur semi-défini positif auto-adjoint pour $< ., . >_M$ sur $\mathbb{R}^p$. Le choix de cette signature a été motivé par le fait que l'on s'affranchit de plusieurs difficultés liées à la capture et l'exécution d'un mouvement. Nous pouvons en effet formuler trois contraintes gérées par ce choix. Il s'agit de caractéristiques pouvant varier même pour deux mouvements similaires. Premièrement, la fréquence d'acquisition des captures peut ne pas être la même. Deuxièmement, la vitesse d'exécution peut différer suivant le niveau de maîtrise du danseur. Enfin, puisque la vitesse varie, le nombre de captures peut également différer. Les mouvements sont identifiés par des opérateurs de même taille quelques soient les caractéristiques formulées ci-dessus et deviennent comparables car appartenant dorénavant au même espace.

2.2 Choix de la métrique entre signatures

On considère un ensemble de K mouvements caractérisé par le vecteur suivant :

$$\Gamma = (\Gamma_1, ..., \Gamma_K) \tag{9}$$

Par la nature même de la signature matricielle, les mouvements sont plongés dans un espace vectoriel de dimension p^2. Chacun d'eux est identifié par un point Γ_k dans cet espace et leur similarité est mesurée en utilisant la métrique de Hilbert-Schmidt (*HS*). Pour deux mouvements identifiés par Γ_i et Γ_j, opérateurs auto-adjoints pour $< ., . >_M$, le produit scalaire de Hilbert-Schmidt est donnée par :

$$< \Gamma_i, \Gamma_j >_{HS} = trace(\Gamma_i^*.\Gamma_j) = trace(\Gamma_i \circ \Gamma_j) \tag{10}$$

de norme associée :

$$\|\Gamma_k\|_{HS} = \sqrt{trace(\Gamma_k^2)} \tag{11}$$

et de distance associée :

$$d(\Gamma_i, \Gamma_j) = \|\Gamma_i - \Gamma_j\|_{HS} \tag{12}$$

Sachant que pour $M = I_p$, le produit scalaire HS sur les opérateurs de taille p est équivalent au produit scalaire canonique sur $\mathbb{R}^{p^2}$ en considérant la représentation vectorielle des opérateurs, il est facile de donner une interprétation concrète des quantités (11) et (12). La distance entre deux opérateurs est faible lorsque les covariances entre les variables sont mutuellement similaires pour les deux mouvements. Plus la norme d'un opérateur est grande et plus la dépendance linéaire entre les variables (i.e. coordonnées de capteurs) est forte. Enfin, le cosinus entre deux opérateurs (RV d'Escoufier (Lavit et al., 1994)) rend compte des proportionnalités entre les signatures.
L'AFO s'apparente à l'ACP car elle consiste à rechercher les sous-espaces préservant au mieux la distance entre les opérateurs. Ces sous-espaces, qualifiés de *principaux*, sont engendrés par des opérateurs *principaux* obtenus par combinaison linéaire des éléments de Γ. L'espace potentiellement utilisé est donc de dimension $d = min(K, p^2)$. Dans le cadre applicatif de cet article, on a $K \ll p^2$. Du fait des contraintes physiques et morphologiques, la possibilité d'exécution d'un mouvement par un être humain est naturellement limitée. La dimension de ces *sous-espaces principaux* sera donc très inférieure à d.
Soit H la matrice des produits scalaires HS entre les éléments de Γ :

$$H(i,j) = < \Gamma_i, \Gamma_j >_{HS} \tag{13}$$

Pour déterminer les opérateurs principaux, on calcule les éléments propres (Λ, U) de H. Λ est la matrice diagonale des valeurs propres classées dans l'ordre décroissant et U est la matrice des vecteurs propres (en colonne) associés. On a donc :

$$HU = U\Lambda \tag{14}$$

Les vecteurs propres U sont supposés normés ($^tUU = I_K$). On en déduit les opérateurs principaux, notés $O = (O_1, ..., O_K)$, par l'équation :

$$O = \Gamma U \tag{15}$$

et par conséquent :

$$< O_i, O_j >_{HS} = \delta_{i,j}\Lambda(i,j) \tag{16}$$

On note $O' = (O'_1, ..., O'_K)$ les opérateurs principaux après normalisation :

$$O' = O\Lambda^{-\frac{1}{2}} \tag{17}$$

Pour visualiser les liaisons entre les mouvements, chaque opérateur Γ_k est projeté sur les plans principaux engendrés par les éléments de O. Pour le plan $i \times j$, on a :

$$P_{<O'_i,O'_j>}(\Gamma_k) = < \Gamma_k, O'_i >_{HS} O'_i + < \Gamma_k, O'_j >_{HS} O'_j \tag{18}$$

Sur ces plans, la distance entre deux éléments projetés s'interprète comme celle définie en (12). Nous verrons dans la prochaine section quel sens on peut donner aux opérateurs principaux.

3 Expérimentations

3.1 Acquisition des données

Les mouvements ont été enregistrés au Cyberdome et exécutés par des danseurs du Ballet Atlantique Régine Chopinot. L'enregistrement d'un mouvement est obtenu de la manière suivante. On place 15 capteurs sur diverses parties du corps du danseur : bras, tête, poignets, mains, torse, pelvis, hanches, jambes et pieds. Un échantillon du mouvement est capturé lors de son exécution. Les coordonnées spatiales de chaque capteur sont enregistrées avec une fréquence d'acquisition de 25 Hz (25 captures par seconde). Nous obtenons donc un ensemble de 45 variables rangées dans un tableau $\mathbb{X}$ de taille $T \times 45$ pour un mouvement de T captures.
Nous disposons de 14 mouvements types[1] : la Marche (ma), l'Enveloppé (en), la Chute Pliée (cp), l'Enroulé Tête (et), l'Enroulé Bassin (eb), le Grand Plié (gp), la Marche Glissée (mg), le Saut Attitude (sa), la Chute Repliée (cr), la Chute Allongée (ca), le Jetté (jt), le Tour (tr), la Suspension (su) et enfin le Saut Temps Levé (st).
Pour l'étude qui va suivre, chaque mouvement type a été réalisé à quatre reprises par des danseurs. Nous avons donc un ensemble constitué de 4 Marches, 4 Tours,... soit $14 \times 4 = 56$ mouvements. Une fois cet ensemble créé, la méthode décrite plus haut nous permet d'étudier la typologie des mouvements ; c'est à dire analyser les ressemblances et les dissemblances. Le but étant de décrire, analyser et reconnaître des mouvements de Danse Classique.

3.2 Résultats expérimentaux

Pour tous les mouvements, nous prenons $M = I_{45}$ et $\theta = 10$. Ainsi, les variables sont équipondérées et deux captures éloignées d'un retard supérieur à 10 n'auront pas d'influence pour le calcul de la covariance. De plus, les individus sont supposés équipondérés ($\forall \omega,\ p_\omega = 1/T$). Tous les opérateurs sont normalisés à l'unité pour éviter la présence de deux groupes de mouvements prédominants : celui où le danseur se déplace (e.g. Marche) ou non (e.g. Enveloppé). Les projections sur les plans principaux sont donc contenues dans la sphère unité.
La figure 1 montre les valeurs propres de la matrice H par ordre décroissant. Les 10 premiers

[1] Les abréviations mentionnées entre parenthèses sont celles utilisées pour les figures.

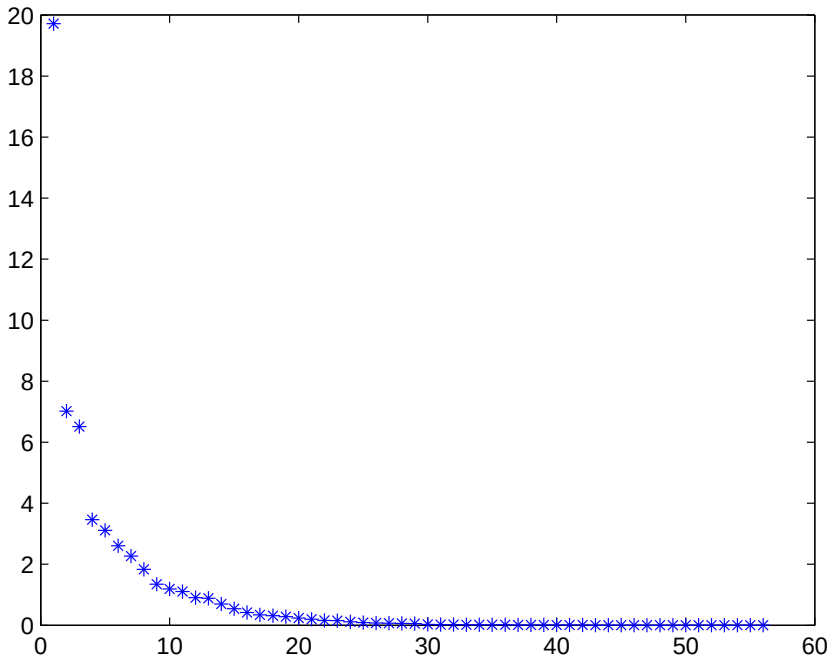


FIG. 1 – *Valeurs propres de H (produit scalaire HS entre opérateurs de covariance normalisés)*

opérateurs principaux apportent plus de 87% de l'inertie totale, O_1 en représente approximativement 35%. Ainsi, en vue des pourcentages d'information, les 10 premiers axes nous permettent d'extraire les principales caractéristiques des mouvements et donc de pouvoir les différencier. O_1 est l'opérateur *compromis* qui caractérise les informations communes à tous les mouvements. Il ne permet pas de discriminer les mouvements autrement que par leur norme et ne constitue donc pas a priori un bon critère de différenciation. Par exemple, cet opérateur peut être relié à des caractéristiques d'ordre morphologique (e.g. le "squelette").
La figure 2 montre la projection des 56 signatures sur le plan principal engendré par le deuxième et le troisième opérateur principal. Tous les mouvements ont été regroupés suivant leur mouvement type (courbe sur le graphique). Ce plan semble être suffisamment discriminant pour la plupart des mouvements. Les types de mouvements Marche, Sauts, Suspension, Chutes forment chacun des groupes isolés. Le nuage des mouvements Jetté est très étendu car il existe différentes façons d'exécuter cette figure : Jetté devant, Jetté derrière et Jetté de côté.
La disposition des points sur les plans principaux permet de donner une caractérisation des opérateurs O_k. En ce qui concerne O_2, nous pourrions l'interpréter comme l'axe qui caractérise le "sens" du mouvement. Les éléments se situant à gauche de l'axe correspondent aux

mouvements verticaux et ceux se situant à droite sont les mouvements dits horizontaux. Considérons par exemple les trois mouvements types suivants : la Marche, le Saut Temps Levé et le Saut Attitude. La Marche, qui est un mouvement plutôt horizontal, se situe à droite du graphique. Le Saut Attitude, qui lui est vertical se trouve à gauche de l'axe. Enfin, le Saut Temps Levé est le mouvement intermédiaire aux deux autres et se situe au milieu. Le Saut Temps Levé, plus complexe, consiste à prendre de l'élan (sens horizontal) puis d'effectuer un saut (sens vertical). Pour O_3, les mouvements qui se situent en bas par rapport à l'origine de cet axe sont les mouvements où la position relative au sol du centre de gravité du danseur varie peu. A contrario, les mouvements se situant plutôt à gauche sont ceux où il y a une grande mobilité du centre de gravité par rapport au sol. Il existe également d'autres opérateurs dont le rôle est

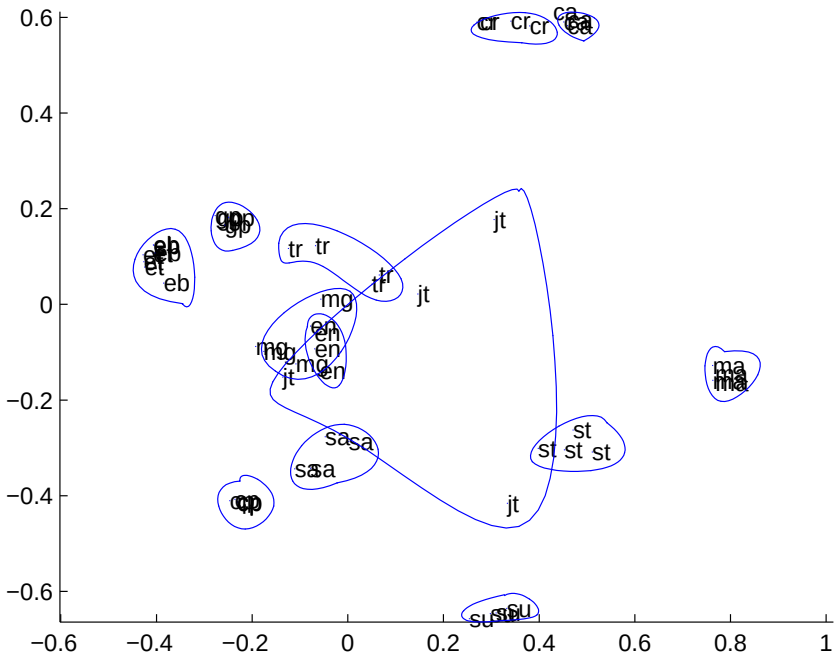


FIG. 2 – *Plan principal 2x3 (inertie* $\approx 25\%$*)*

facilement interprétable. Le graphe de la figure 3 montre le plan 2×5 avec les opérateurs projetés. On constate que le Tour, mouvement mal discriminé sur le plan 2×3, se distingue clairement des autres mouvements. L'opérateur O_5 permet de mieux identifier ce mouvement et cet axe pourrait donc caractériser la circularité des mouvements.

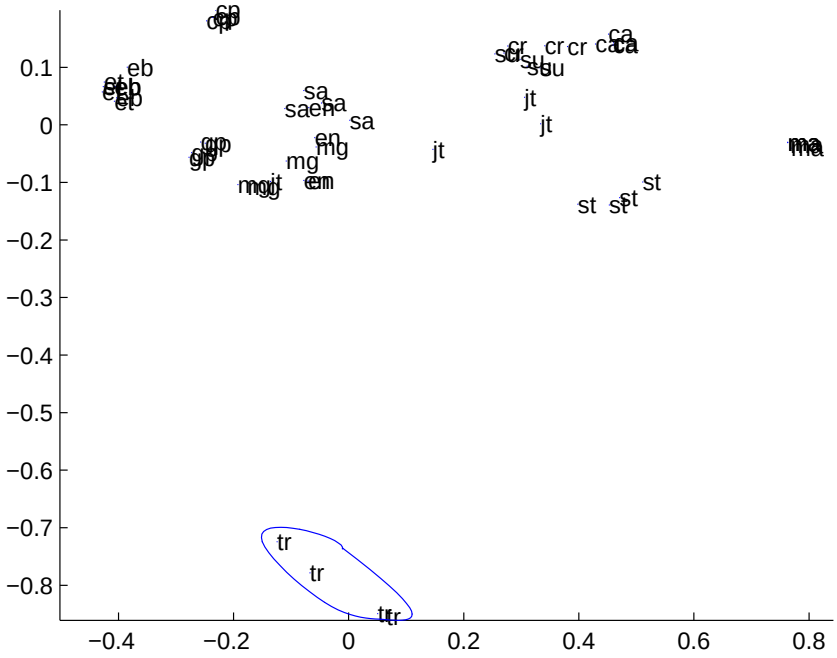


FIG. 3 – *Mouvement Tour isolé sur le plan principal 2x5*

4 Discussion

Outre les différents avantages de la signature choisie, il est important de préciser que nous utilisons la covariance et non la corrélation entre variables. Le système d'acquisition étant susceptible de fournir des données bruitées, normer préalablement les variables a pour effet d'accroître ce bruit pour le calcul de la signature. Par ailleurs, l'utilisation de la corrélation implique que toutes les variables ont la même importance ; cela accentue donc le rôle de certaines variables moins significatives.

Un autre point est l'influence de la position initiale du danseur sur la méthode proposée. Supposons par exemple que le danseur exécute une marche le long de la bissectrice du plan xOz[2] puis le long de l'axe x. Dans le premier cas, les variables en x et z sont fortement corrélées et dans le deuxième, ces corrélations seront inexistantes puisque les coordonnées en z des capteurs sont quasi-nulles. Les signatures pour ces deux mêmes mouvements sont donc significativement différentes. Cependant, pour pallier à ce problème, il existe plusieurs alternatives comme par exemple considérer non pas le tableau des coordonnées brutes des capteurs mais celui de leur distance mutuelle, invariante quelque soit la position initiale du danseur.

[2]Le trièdre (x, y, z) est direct mais c'est l'axe y qui détermine la hauteur.

5 Conclusion

Dans cet article nous avons présenté une méthode de reconnaissance et de classification basée sur l'analyse factorielle d'opérateurs traitant des données multi-dimensionnelles et temporelles. Nous l'avons appliqué à l'étude de mouvements dansés obtenues par captures successives des coordonnées spatiales de 15 marqueurs situés sur les principales articulations du danseur. Malgré le fait que leur fréquence d'acquisition, leur vitesse d'exécution et leur nombre de captures peuvent varier, il est possible de comparer les mouvements en les identifiant par leur opérateur de covariance relationnelle. Cette signature tient compte des liens inter-capteurs et de la contrainte d'ordre temporel. Notre méthode est une variante de l'analyse en composantes principales car elle consiste à projeter les signatures sur des espaces de dimension réduite contenant le maximum d'information afin d'étudier les ressemblances ou non des mouvements. Sans être expert en chorégraphie, cette méthode autorise une analyse descriptive et une reconnaissance automatique de mouvement.
Actuellement, nous travaillons sur une technique de segmentation comportementale appliquée aux trajectoires. Les exemples traités dans cet article sont très structurés puisque nous n'avons pas étudier le cas où des mouvements seraient constitués d'une succession de figures distinctes. En présence de tels mouvements, notre but est d'en identifier les comportements hétérogènes par une technique basée sur l'analyse factorielle et les modèles de Markov cachés.

Références

Bollodas, B. (1998). *Graph Theory*. Springer Verlag.

Chartrand, G., O. Oellermann, et A. Schwenk (1991). Graph theory, combinatorics and applications. *2*, 871–898.

Chenevière, F. (2004). *Contribution à l'analyse du mouvement dansé*. Ph. D. thesis, Université de La Rochelle.

Dietterich, T. (2002). Machine learning for sequential data : A review. In Springer (Ed.), *Lectures Notes in Computer Science*, Volume 2396, pp. 15–30.

Foucart, T. (1984). *Analyse factorielle de tableaux multiples*. Paris.

Gaffney, S. et P. Smyth (1999). Trajectory clustering with mixtures of regression models. Technical Report 99-15, Department of Information and Computer Science, University of Californian Irvine.

Joliffe, I. (1986). *Principal Component Analysis*. Springer Verlag.

Lavit, C., Y. Escoufier, R. Sabatier, et P. Traissac (1994). The act statis method. *Computational Statistics and Data Analysis 18*, 97–119.

Lebart, L., A. Morineau, et M. Piron (2000). *Statistique exploratoire multidimensionnelle*. Dunod.

Moeslund, T. et E. Granum (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding 81*(3), 231–268.

Oates, T., L. Firoiu, et P. R. Cohen (2001). Using dynamic time warping to bootstrap hmm-based clustering of time series. In *Sequence Learning - Paradigms, Algorithms, and Appli-*

cations, London, UK, pp. 35–52. Springer-Verlag.

Thullier, F. et H. Moufti (1995). Multi-joint coordination in ballet dancers. *Neuroscience Letters 369*, 80–84.

Zeng, Y. et J. Garcia-Frias (2005). A novel hmm-based clustering algorithm for the analysis of gene expression time-course data. *Computational Statistics and Data Analysis*.

Summary

We propose a method derived from the factorial analysis, which, on the one hand, describe types of movements and on the other hand automatically recognize them. We experiment our method on motion-captured movements realized by several dancers from the 'Ballet Atlantique Régine Chopinot (BARC)'. Presented techniques are mainly the analysis in principal components of relational covariance and the factorial analysis of operators, which can be easily extended for automatic recognition of movements.

Réduction de la dimension temporelle de séries multivariées par extraction des tendances locales

Ahlame Chouakria-Douzal*

*Timc-Imag, Université Joseph Fourier, Grenoble 1
domaine de La-Merci F-38706, France
Ahlame.Douzal@imag.fr

Résumé. L'application des méthodes d'analyse de données à des séries temporelles se trouve vite limitée par le nombre souvent très élevé des observations composant les séries. Ce travail propose une nouvelle technique de réduction de la dimension temporelle d'une série multivariée préservant la structure de variance-covariance, élément de base pour de nombreuses méthodes d'analyse de données. Cette nouvelle technique est fondée sur l'extraction des principales tendances locales composant la série et permet la conservation de la dimension temporelle des séries réduites. Les résultats de cette approche sont illustrés sur une application médicale en anesthésie-réanimation.

1 Introduction

La multiplication d'applications industrielles (télésurveillance, télécommunication, satellite,...), médicales ou scientifiques portant sur des bases de données temporelles exprime un besoin grandissant autour des problématiques d'analyse et d'extraction automatique de connaissance à partir de données temporelles. Par ailleurs, ces base de données temporelles souvent caractérisées par des volumes très importants rendent inapplicables toutes méthodes classiques d'analyse. Une première approche consiste à réduire la dimension temporelle des séries avant toute analyse. Un des premiers travaux de réduction de la dimension temporelle au sein des base de données temporelles fondé sur la transformée de Fourier discrète (DFT) fut introduit par AGRAWAL ((Agrawal et al., 1993)). D'autres travaux suivirent, basés sur la décomposition en valeurs singulières (SVD) (Wu et al. (1996)) ou sur la transformée par ondelettes discrètes (DWT) (Chan et Fu (1999)). Ces trois techniques de réduction de dimension ont la caractéristique d'exprimer les séries temporelles dans un nouvel espace de dimension réduite dont les dimensions sont définies par les valeurs singulières (cas d'une SVD) ou par les fréquences (cas d'une DFT ou d'une DWT). L'objectif de ce travail est de proposer une nouvelles techniques de réduction de la dimension temporelle, fondée sur l'extraction des tendances locales, préservant l'information de variance-covariance introduite par les descripteurs et conservant la dimension temporelle des séries réduites. Cette nouvelle approche est illustrée sur des données issues d'une application en anesthésie-réanimation. Le reste de l'article est organisé comme suit. La section 2 introduit la notion de variance-covariance locale, notion introduite par J. VON NEUMANN (Von Neumann (1942)). Les résultats de la variance-covariance locale sont ensuite développés dans le cas du temporel. La section 3, présente les propriétés et les définitions liées à la variance-covariance temporelle. La section 4, présente la nouvelle technique

de réduction de dimension temporelle. Avant de conclure en section 6, la section 5 illustre la technique proposée sur des données médicales en anesthésie-réanimation.

2 Structure de contiguïté locale vs. temporelle

La notion de variance temporelle a été introduite initialement par J. VON NEUMANN (Von Neumann (1942)). Il montre que le calcul de la variance classique sur des données décrivant une tendance dans le temps est une surestimation de la variance réelle introduite par ces mêmes données. Une généralisation de ces travaux à une structure de contiguïté en général (spatial, temporel, intéraction, ...) a été proposée par R.C. GEARY (Geary (1954)) puis par L. LEBART (Lebart (1969))par l'introduction de la notion de variance locale.

Considérons N observations $w_1, ..., w_N$ dotées d'une structure de contiguïté connue a priori. Chaque observation donne la description d'un ensemble de variable $X_1, ..., X_p$. On note x_{ij} la valeur prise par la variable X_j lors de l'observation w_i. La structure de contiguïté associe à chaque observation w_i son voisinage $E_i \subset \{1, ..., N\}$ défini comme suit : $\forall k \in E_i$ w_k contiguë à w_i. On note $V_L(v_{jl}^L)$ la matrice de variance-covariance locale de dimension p associée aux N observations et tenant compte de la structure de contiguïté :

$$v_{jl}^L = \frac{1}{2\,m} \sum_{i=1}^{N} \sum_{i' \in E_i} (x_{ij} - x_{i'j})(x_{il} - x_{i'l}) \tag{1}$$

où $m = \sum_{i=1}^{N} Card(E_i)$, $Card(E_i)$ étant la cardinalité du voisinage E_i. Particulièrement, quand toutes les observations sont contiguës, et où la structure de contiguïté est représentée par un graphe complet, la variance-covariance locale converge vers la variance-covariance classique :

$$v_{jl}^L = \frac{1}{2\,N\,(N-1)} \sum_{i=1}^{N} \sum_{i'=1}^{N} (x_{ij} - x_{i'j})(x_{il} - x_{i'l})$$

La variance-covariance locale rapportée à la variance-covariance classique permet de mesurer de combien les données observées sont dépendantes de la structure de contiguïté *a priori*. Dans le cas de données indépendantes de la structure de contiguïté, la variance-covariance locale converge vers la variance-covariance classique. Par contre, dans le cas de données dépendantes de la structure de contiguïté, la variance-covariance classique surestime la variance-covariance locale. Ainsi GEARY propose un indice compris entre 0 et 1 mesurant le rapport antre la variance-covariance locale et la variance-covariance classique. Une valeur de cet indice proche de 0 implique une situation de dépendance forte de la structure *a priori*, et une valeur proche de 1 indique une situation d'indépendance forte de la structure de contiguïté.

Soit S une une série temporelle multivariée composée de N observations $w_1, ..., w_N$ effectuées respectivement aux instants $t_1, ..., t_N$ et donnant la description de p variables $X_1, ..., X_p$:

$$S = \begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix} = \begin{array}{c} \begin{matrix} X_1 & \ldots & X_p \end{matrix} \\ \begin{pmatrix} x_{11} & \ldots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{N1} & \ldots & x_{Np} \end{pmatrix} \end{array} \quad (2)$$

Soit w_i une observation effectuée à l'instant t_i. On définit dans ce qui suit le voisinage temporel d'ordre $k \geq 1$ de w_i comme étant l'ensemble des observations effectuées dans la fenêtre temporelle $[t_{i-k}, t_{i+k}]$. On note $E_i \subset \{1, ..., N\}$ l'ensemble référençant le voisinage de w_i et vérifiant :

$$\forall i' \in E_i \quad \Rightarrow \quad max(1, i-k) \leq i' \leq min(N, i+k)$$

On considère dans ce qui suit un voisinage temporel d'ordre $k = 1$. L'expression de la matrice de variance-covariance temporelle associée à S et notée $V_T(S)$ de terme général $v_{jl}^T(S)$ est :

$$v_{jl}^T(S) = \frac{1}{4\,(N-1)} \sum_{i=1}^{N-1} (x_{(i+1)j} - x_{ij})(x_{(i+1)l} - x_{il}) \quad (3)$$

L'indice de GEARY, dans le cas de la contiguïté temporelle, permet de mesurer combien les données observées sont dépendantes du temps. Plus cet indice est faible et plus les données expriment une tendance dans le temps ; réciproquement, plus fort est cet indice (proche de 1) moins les observations expriment une forme ou une tendance dans le temps (exemple : cas d'un processus aléatoire).

3 Définitions et propriétés

Cette section introduit des opérateurs définis sur l'ensemble des séries temporelles et annonce quelques propriétés.

3.1 Définitions

Décomposition et concaténation de séries temporelles Soit $S = (w_1, ..., w_N)$ une série temporelle multivariée définie par les observations $w_1, ..., w_N$. On considère la décomposition de S en k sous-séries temporelles $S_1 = (w_1, ..., w_{n_1})$, $S_2 = (w_{n_1}, ..., w_{n_2})$,...,$S_k = (w_{n_{k-1}}, ..., w_N)$ où $n_i \in \{1, .., N\}$. On dénote $L_1 = n_1, L_2 = n_2 - n_1 + 1, ..., L_k = N - n_{k-1} + 1$ les longueurs respectives de $S_1, S_2, ..., S_k$. On dénote dans ce qui suit une telle décomposition par $S = S_1 \oplus S_2 \oplus ..., \oplus S_k$. On remarque que deux sous-séries temporelles S_i, S_{i+1} successives ont une observation commune w_{n_i}.

Opérateur d'inclusion Soit S une série temporelle multivariée, $P(S)$ l'ensemble de toutes les sous-séries temporelles extraites de S. On définit l'opérateur d'inclusion dans $P(S)$ par :

$$\forall\, S_i, S_j \in P(S) \quad S_i \subset S_j \quad \Rightarrow \quad \exists\, S_k \in P(S) \text{ tel que } S_j = S_i \oplus S_k.$$

Relation d'ordre dans $P(S)$ On définit une relation d'ordre $\leq$ dans P(S) comme suit :

$$\forall\ S_i, S_j \in P(S) \quad S_i \leq S_j \quad \Leftrightarrow S_i \subset S_j \tag{4}$$

3.2 Propriétés

Contribution à la variance-covariance temporelle Soit S une série temporelle multivariée de longueur N. On décompose S en k sous-séries temporelles S_1,..., S_k de longueurs respectives L_1, ..., L_k, avec $S = S_1\oplus, ..., \oplus S_k$. La matrice de variance-covariance temporelle $V_T(S)$ vérifie la propriété suivante :

$$V_T(S) = V_T(S_1\oplus, ..., \oplus S_k) \quad = \quad \sum_{i=1}^{k} \frac{L_i - 1}{N - 1} V_T(S_i) \tag{5}$$

On définit C_S la fonction mesurant la contribution des sous-séries temporelles S_i de $P(S)$ à la variance-covariance temporelle de S :

$$C_S : P(S) \quad \longrightarrow \quad R^+$$

$$S_i \quad \longrightarrow \quad C_S(S_i) = \frac{L_i - 1}{N - 1} Tr(V_T(S_i) \tag{6}$$

On vérifie aisément que $C_S(S) = Tr(V_T(S))$.

Propriété d'additivité de C_S Soit $S = S_1\oplus, ..., \oplus S_k$ une série temporelle multivariée de longueur N décomposée en k sous-séries temporelles, la fonction C_S vérifie la propriété d'additivité :

$$C_S(S) = C_S(S_1\oplus, ..., \oplus S_k) \quad = \quad \sum_{i=1}^{k} C_S(S_i) \tag{7}$$

Propriété de monotonicité et de croissance de C_S Soit S une série temporelle multivariée de longueur N. Soit S_i, S_j deux séries temporelles de $P(S)$ avec $S_i \subset S_j$. La mesure des contributions C_S à la variance-covariance de S définit une fonction monotone croissante :

$$\forall\ S_i,\ S_j \in P(S)\ \ S_i \leq S_j \quad \Rightarrow \quad C_S(S_i) \leq C_S(S_j)$$

4 Nouvelle technique de réduction de la dimension temporelle

Toute méthode de réduction de dimension temporelle représentant une série temporelle par un nombre d'observation plus faible, engendre une perte d'information. On s'intéresse dans ce qui suit à l'information de variance-covariance introduite par les descripteurs de la série. Nous proposons une méthode de réduction s'appuyant sur un seul paramètre en entrée,

le taux maximal de perte d'information suite à la réduction de dimension. Soit $\alpha_p \in [0,1]$ ce paramètre. On Note $\tau_{rd} \in [0,1]$ le taux de réduction réalisé :

$$\tau_{rd} \quad = \quad 1 - \frac{Nombre\ d'observation\ final}{Nombre\ d'observation\ initial}$$

Une réduction au seuil $\alpha_p \simeq 0$ (aucune perte d'information) fournit une série temporelle résultat identique à la série initiale avec un taux $\tau_{RD} \simeq 0$. A l'opposé, une réduction au seuil $\alpha_p \simeq 1$ (une perte d'information jusqu'à 100%) fournit une série temporelle résultat représentée par deux observations avec un taux $\tau_{RD} \simeq 1$. Statistiquement, afin de capturer les liens caractéristiques entre les descripteurs il est important qu'une réduction de dimension préserve un minimum de 70% ($\alpha_p \leq 0.3$) des corrélations issues des données initiales. Etant donné une valeur du seuil α_p, la méthode de réduction proposée peut être résumée en trois principales étapes. La première consiste à segmenter la série temporelle S en plusieurs sous-séries temporelles formant une partition sur l'ensemble des observations. Chaque sous-série temporelle est extraite sous condition d'inclure une contribution minimale à la variance-covariance temporelle totale α_{C_S} de S. Chaque sous-série temporelle ainsi extraite capture une forme ou une tendance représentative de la série temporelle. Vient ensuite l'étape de régression où chaque sous-série temporelle, issue de la segmentation, est représentée par un segment en procédant à une régression linéaire multiple sur l'ensemble des observations la composant. La série temporelle de dimension réduite est alors représentée par la succession de segments correspondants aux sous-séries temporelles extraites. Finalement dans la troisième étape, on évalue la perte d'information (corrélations) engendrée par la réduction de dimension. Pour cela on mesure l'écart moyen entre les corrélations issues de la série temporelle avant et après réduction. En fonction de cet écart, on réajuste (incrémente / décrémente) le seuil α_{C_S}. On réitère les trois étapes précédentes avec le nouveau seuil α_{C_S} jusqu'à l'obtention d'une réduction de dimension optimale maximisant τ_{RD} et préservant les corrélations à un taux supérieur ou égal à $1 - \alpha_p$. Les principales étapes de l'algorithme sont détaillées dans la section suivante.

4.1 Initialisation

Cette étape consiste à initialiser trois paramètres : - la valeur du seuil $\alpha_p \in [0,1]$. Une valeur typique de $\alpha_p = 0.1$ (taux de 10%) correspond à un taux de conservation de l'information de 90% suite à la réduction de la dimension ; - la valeur initiale du seuil $\alpha_{C_S} \in [0,1]$. On choisit typiquement une valeur initiale de $\alpha_{C_S} = 0.01$; celle-ci sera réajustée itérativement en fonction des données composant S. α_{C_S} correspond au taux minimal de contribution à la variance-covariance temporelle de S que doit vérifier une sous-série temporelle S_i ($C_S(S_i) \geq \alpha_{C_S}$) pour être extraite lors de la segmentation. Sémantiquement, cette condition assure une tendance minimale au sein de chaque sous-série temporelle extraite ; - enfin à calculer $V_T(S)$ la matrice de variance-covariance de S, de terme général $v_{jl}^T(S)$. Le calcul de $V_T(S)$ se fait de manière incrémentale (propriété d'additivité).

4.2 Segmentation

Extraire à partir de $S = (w_1, ..., w_N)$ et dans l'ordre les sous-séries temporelles $S_1 = (w_1, ..., w_{n_1}), ..., S_k = (w_{n_{k-1}}, ..., w_N)$ de longueur respectivement $L_1, ..., L_k$ avec $k, L_1, ..., L_k$

non connus *a priori*. On considère S_i^r la sous-série temporelle portant sur les r observations $w_{n_{i-1}}, ..., w_{n_{i-1}+r}$. De part la propriété de monotonicité et de croissance de C_S, on a :

$$\forall r \in [0, N] \;\; S_i^{r-1} \subset S_i^r \quad \Rightarrow \quad C_S(S_i^{r-1}) \leq C_S(S_i^r)$$

le processus d'extraction de la sous-série temporelle S_i consiste à rechercher la première valeur de r vérifiant la condition de coupure :

$$C_S(S_i^r) \leq \quad \alpha_{C_S} C_S(S) \quad < C_S(S_i^{r+1})$$

Après l'extraction de la sous-série temporelle S_i de longueur $L_i = r$ on procède à l'extraction de S_{i+1}, S_{i+2}..., jusqu' à la segmentation totale de S.

4.3 Régression

Soit $S_i = (w_{n_{i-1}}, .., w_{n_i})$ la ième sous-série temporelle extraite de longueur L_i, on note $t_{n_{i-1}}, ..., t_{n_i}$ les temps correspondants aux observations $w_{n_{i-1}}, .., w_{n_i}$:

$$\begin{array}{cccc} & X_1^{S_i} & \dots & X_p^{S_i} \end{array}$$

$$S_i \quad = \quad \begin{pmatrix} x_{n_{i-1}1} & \dots & x_{n_{i-1}p} \\ \vdots & \ddots & \vdots \\ x_{n_i 1} & \dots & x_{n_i p} \end{pmatrix}$$

où $X_j^{S_i}$ représente les valeurs prises par la variable X_j pour la série temporelle S_i. Pour chaque sous-série temporelle S_i ainsi obtenue on effectue une régression linéaire multiple afin de déterminer la surface de réponse approximant (au sens des moindres carrés) au mieux les L_i observations. Le modèle linéaire de régression utilisé est :

$$X_j^{S_i} \quad = \quad \alpha_0 \mathbf{1}_{L_i} + \alpha_1 T + \sum_{s=2}^{p} \alpha_s X_{l_s}^{S_i}$$

où, $l_s \in \{1, ..., p\} \setminus \{j\}$ et $X_j^{S_i}$ est la variable à estimer de dimension $(L_i \times 1)$, 1_{L_i} le vecteur unitaire de dimension $(L_i \times 1)$ et $\alpha_0, ..., \alpha_p$ les coefficients de régression associés aux autres variables du modèle. Les valeurs des coefficients de régression sont obtenues par la résolution du système d'équations suivant :

$$\begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_p \end{pmatrix} \quad = \quad \begin{pmatrix} 1 & t_{n_{i-1}} & x_{n_{i-1}1} & ... & x_{n_{i-1}p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_{n_i} & x_{n_i 1} & ... & x_{n_i p} \end{pmatrix}^{-1} \begin{pmatrix} x_{n_{i-1}j} \\ \vdots \\ x_{n_i j} \end{pmatrix}$$

Chaque sous-série temporelle S_i sera représentée par uniquement deux observations correspondant aux estimations aux instants $t_{n_{i-1}}$ et t_{n_i}.

4.4 Estimation de l'erreur

On note S^* la série temporelle obtenue suite à la réduction de la dimension temporelle de S. On note $Cor(S^*)$ de terme général $c_{jl}(S^*)$, $Cor(S)$ de terme général $c_{jl}(S)$, respectivement, les matrices des corrélations associées à S^*, à S. On propose d'estimer l'erreur moyenne dûe à la réduction de dimension comme suit :

$$e(S, S^*) \quad = \quad \frac{2}{p(p-1)} \sum_{j=1}^{p} \sum_{l=j+1}^{p} \mid c_{jl}(S) - c_{jl}(S^*) \mid$$

avec $e(S, S^*) \in [0, 1]$. On évalue la perte d'information (des corrélations) suite à la réduction de dimension en comparant $e(S, S^*)$ au seuil α_p. Deux cas de figure se présentent :
Cas 1 $e(S, S^*) \leq \alpha_p$ **:** L'objectif ici est de rechercher la valeur maximale de τ_{RD} préservant les corrélations au seuil défini *a priori*. Pour ce faire, on incrémente α_{C_S} d'un pas défini comme suit :

$$\alpha_{C_S} \quad = \quad min(\alpha_{C_S}, 2/3)\, 3/2$$

On note S_l^* la série temporelle réduite obtenue à l'itération l. On réitère les trois étapes avec la nouvelle valeur du seuil α_{C_S} et ce jusqu'à la vérification de la condition d'arrêt suivante :

$$e(S, S_l^*) \leq \quad \alpha_p \quad < e(S, S_{l+1}^*)$$

où S_l^* est la série temporelle réduite maximisant τ_{RD} et préservant les corrélations à taux de $1 - \alpha_p$. Remarquons que l'augmentation du seuil α_{C_S} a pour effet d'augmenter la variance minimale au sein de chaque sous-série temporelle, ce qui a pour consérie temporelle de réduire le nombre des sous-séries temporelles extraites donc d'augmenter le taux τ_{RD}. **Cas 2** $e(S, S_1^*) > \alpha_p$ *:* L'objectif ici est de rechercher la sous-série temporelle S^* préservant les corrélations au seuil défini *a priori*. Pour ce faire, on décrémente α_{C_S} d'un pas défini comme suit :

$$\alpha_{C_S} \quad = \quad \alpha_{C_S}\, max(0.5, e(S, S^*))$$

On réitère les trois étapes avec le nouveau seuil α_{C_S} et ce jusqu'à l'itération l vérifiant la contrainte suivante :

$$e(S, S_l^*) \leq \quad \alpha_p \quad < e(S, S_{l-1}^*)$$

où S_l^* est la série temporelle préservant les corrélations au taux de $1 - \alpha_p$.

5 Applications et résultats

On dispose de données issues du monitoring composé de 11 capteurs physiologiques observant l'état d'un patient en anesthésie réanimation. Ces données constituent une série temporelle multivariée donnant la description de 5269 observations d'un patient par 11 variables descriptives quantitatives. Les observations sont effectuées à des intervalles de temps réguliers. Pour des soucis de bonne visibilité, la figure 1 donne la représentation en deux dimensions (temps

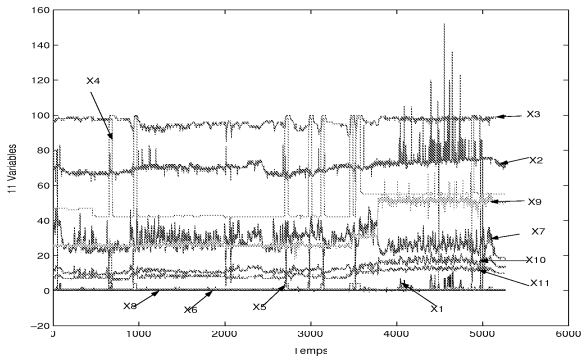


FIG. 1 – *Représentation plane des courbes associées aux 11 descripteurs*

$\times$ descripteur) des courbes associées aux 11 descripteurs. Pour illustrer le fonctionnement de l'algorithme, on propose d'appliquer la méthode de réduction à différents niveaux de perte d'information (de corrélations) allant de 10% à 70%. Chaque ligne du tableau 1 résume les résultats obtenus suite à l'application de la réduction de dimension avec un taux α_p de perte maximale de corrélations. Par exemple, pour une perte maximale de 10%, on obtient une série

α_p (%)	Nb observation	τ_{RD} (%)	$e(S, S^*)$ (%)	Temps (s)	α^*_{Cs}	Nb itération
10	388	92.63	9.3	515.46	0.0013	3
20	158	97.00	18.38	132.40	0.005	2
30	32	99.39	26.19	492.59	0.0338	4
40	18	99.65	34.75	396.14	0.0759	6
50	12	99.77	44.37	594.98	0.1139	7
60	6	99.88	55.05	858.42	0.2563	9
70	4	99.92	47.22	543.53	0.5767	11

TAB. 1 – *Réduction de dimension de la série à différents taux* α_p

temporelle réduite représentée par uniquement 388 observations, soit un taux de réduction de 92.63%, la perte effective d'information est de 9.3% soit une conservation de 90.7% des corrélations initiales, le temps d'exécution total est de 515.46 secondes. L'algorithme converge en trois itérations avec une valeur finale de α^*_{Cs} de 0.0013. La figure 2 représente les séries réduites suite à une réduction de dimension aux seuils de perte de corrélation de 10% et 20%. Remarquons que plus le taux de perte d'informations fixé *a priori* α_p est élevé, plus on "agrège" d'observations au sein de chaque sous-série temporelle extraite et plus fort est le taux de réduction τ_{RD}. La figure 3 représente les rapports taux de réduction, perte d'information effectives $\frac{\tau_{RD}}{e(S,S^*)}$ associés aux différentes valeurs de α_p. On remarque que la réduction de dimension de la série temporelle multivariée qui maximise ce rapport est obtenue à un seuil de perte d'information de 10%, soit à un taux de réduction de dimension de 92.63% et à taux de préservation de corrélations de 90.7%.

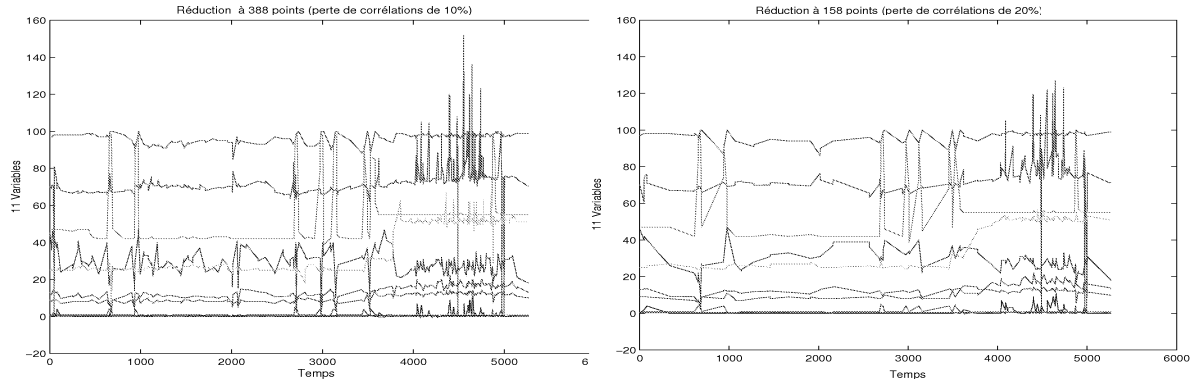


FIG. 2 – $\alpha_p = 10\%$ $\alpha_p = 20\%$

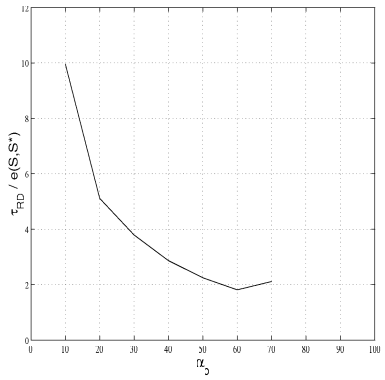


FIG. 3 – *L'évolution des paramètres taux de réduction et perte de variance/covariance après réduction*

6 Conclusion

Ce travail propose une nouvelle technique de réduction de la dimension temporelle de séries multivariées préservant la structure de variance/covariance introduite par les données. L'approche proposée est fondée sur l'extraction des principales tendances locales caractérisant les séries. Cette technique à l'avantage de ne dépendre en entrée que du seuil de perte d'information (variance/covariance) maximale autorisé suite au processus de réduction.

Références

Agrawal, r., C. Faloutsos, et a. Swami (1993). Efficient similarity search in sequence databases. In *Proc. of the* 4^{th} *Conference on Foundations of Data Organization and Algorithms*.

Chan, k. et w. Fu (1999). Efficient time series matching by wavelets. In *Proc. of the* 15^{th} *IEEE International Conference on Data Engineering*.

Geary, R. (1954). The contiguity ratio and statsitical mapping. *The Incorporated Statistician 5,3*, 115–145.

Lebart, L. (1969). Analyse statistique de la contiguité. *Publication de l'ISUP 18*, 81–112.

Von Neumann, j. (1942). The mean square successive difference. *Annals of Mathematical Statistics*, 153–162.

Wu, D., r. Agrawal, a. El Abbadi, a. Singh, et t. Smith (1996). Efficient retrieval for browsing large image databases. In *Proc. of the* 5^{th} *International Conference on Knowledge Information*, pp. 11–18.

Summary

The application of Data Analysis methods to time series is limited because of the inherent high dimension of temporal data. One promising solution consists in performing dimensionality reduction before the time series analysis. We propose a new technique which reduces the temporal dimension of multivariate time series while preserving variance/covariance structure. We illustrate the efficiency of the proposed technique through an intensive care unit application.

Echantillonnage adaptatif et classification supervisée de séries temporelles

Pierre-François Marteau, Marwan Fuad, Gildas Ménier

Valoria, Université de Bretagne Sud, BP573, 56017 Vannes France
{pierre-francois.marteau,marwan.fuad, gildas.menier}@univ-ubs.fr
http://www-valoria.univ-ubs.fr/Pierre-Francois.Marteau/

Résumé. Nous développons dans cet article une première étude expérimentale dans le but d'évaluer l'impact de la réduction de la dimension de l'espace de représentation des séries temporelles sur une tâche de classification supervisée. La réduction de la dimension de l'espace de représentation est abordée sous l'angle d'une technique d'échantillonnage adaptatif des séries temporelles. L'étude expérimentale concerne des données de synthèses bruitées se répartissant en trois classes (fonctions Cylinder, Bell, Funel). Pour différents taux de compression et différentes distances ou fonctions de similarités, les résultats de classification sont calculés et comparés. Cette étude montre que, pour les données traitées, l'échantillonnage adaptatif proposé ne dégrade pas les résultats de classification (des améliorations de performances sont mêmes possibles) pour les taux de compression faible et moyens : elle ne les dégrade que très faiblement pour les taux relativement élevés.

1 Introduction

De plus en plus d'applications nécessitent la recherche et l'extraction de séries temporelles similaires. Parmi les nombreux exemples, on peut citer l'analyse de données financières et la prédiction de marchés (Agrawal et al.., 1993, 1995), (Faloutsos et al. 1994) la détermination de trajectoire pour des objets en mouvements (Zhu et al. 2003) et la recherche de morceaux musicaux (Chen et al.2003) . Les études dans ces domaines tournent autour de deux questions clés : le choix d'une métrique (distance ou similarité) appropriée au domaine traité, et les mécanismes pour accélérer l'efficacité de la recherche. Concernant la première question, un grand nombre de métrique ont été proposées dans la littérature, en particulier les normes Lp marchés (Agrawal et al.., 1993, 1995), (Faloutsos et al. 1994), la comparaison dynamique (Dynamic Time Warping, DTW) (Sakoe & Chiba 1978) (Yi 1998) (Kim, 2001) (Keogh 2002), la plus longue sous séquence commune (Longest Common Sub Sequence, LCSS) (Boreczky & Rowe 1996) et la distance d'édition pour des représentation symbolique ou réelles (Symbolic Edit Distance SED, Edit Distance on Real EDR) (Levenshtein, 1966) (Chen 2003). Concernant la deuxième question, les normes Lp sont facilement calculables (en complexité *O(n)*) par contre, elle ne peuvent prendre en compte une flexibilité temporelle (étirement ou compression), une propriété souvent déterminante dans de nombreuses applications pour améliorer la recherche de similarité entre les séries temporelles. DTW, LCSS et EDR ont été proposées précisément pour gérer des déformations temporelles. Ce ne sont

malheureusement pas des distances puisque celles-ci ne respectent pas l'inégalité triangulaire. Cette limite peut s'avérer pénalisante puisque bon nombre d'optimisations basées sur des principes de réduction de l'espace de recherche nécessitent l'exploitation de l'inégalité triangulaire. Pour répondre à ce manque, certains travaux proposent des approches qui visent à marier les propriétés des distances d'édition avec les propriétés des DTW : ainsi la distance d'édition à base de pénalités réelles (Edit Distance with Real Penaly, ERP) (Chen 2004) conjugue la propriété « d'élasticité temporelle » tout en vérifiant l'inégalité triangulaire. Quoiqu'il en soit, la complexité des métriques ou fonctions de similarité « élastiques » reposent sur des algorithmes en complexité $O(n^2)$, ce qui constitue la principale difficulté lors du passage à l'échelle des applications. L'ampleur des bases de données temporelles ou séquentielles exploitées aujourd'hui justifie le nombre croissant de travaux qui ciblent l'accélération des processus mis en œuvre lors de la recherche par similarité (Agrawal 1995, Keogh2002, etc).

Les travaux portant sur la réduction de la dimension de l'espace de représentation des séries temporelles apportent des solutions particulièrement adaptées à cette question. L'utilisation de mécanismes de recherche dans l'espace à dimension réduite est ensuite mise en œuvre.
Ce principe a été introduit en (Agrawal et al. 1993, 1995). Le travail original d'Agrawal et al. utilise la Transformée de Fourier Discrète (Discrete Fourier Transform, DFT) pour réduire la dimension de l'espace de représentation . D'autres techniques ont été suggérées, en particulier la décomposition en valeurs singulières, Singular Value Decomposition, (SVD) (Keogh 2001), approximation adaptative en segments constants (APCA : Adaptive Piecewise Constant Approximation) (Keogh et al., 2001), et la transformé en ondelettes discrète (Discrete Wavelet Transform, DWT) (Chan 1999).

Nous proposons dans cet article d'aborder la question de la réduction de la dimension de l'espace de représentation des séries temporelles sous l'angle de principes d'échantillonnage adaptatifs (Marteau et al., 2005). Pour évaluer l'intérêt de cette approche, nous proposons un cadre expérimental permettant d'évaluer l'impact de la réduction d'information que celle-ci engendre sur une tâche de classification supervisée. Nous comparons également la méthode par rapport à la technique de réduction basée sur les coefficients de Fourrier. Enfin, nous analysons l'intérêt d'utiliser des fonctions de similarité « élastique » sur l'axe temporelle par rapport à des métriques de type norme Lp, sur la tâche considérée.

2 Echantillonnage adaptatif de courbes multidimensionnelles

Nous proposons une approche orientée modèle au problème de l'échantillonnage adaptatif. Plus précisément, nous recherchons une approximation $X_{\hat{\theta}}$ de $X(n)$ telle que :

$$\hat{\theta} = \underset{\theta}{ArgMin}\left(E(X, X_\theta)\right)$$

où E est l'erreur quadratique entre X et le modèle X_θ .

Nous nous limitons ici, à la famille $\{ X_\theta(n) \}$ des fonctions linéaires par morceaux. De nombreuses méthodes ont été proposées pour approcher des courbes multidimensionnelles en utilisant des simplifications linéaires par morceaux via des algorithmes à base de programmation dynamique en complexité $O(kn^2)$, où k est le nombre de segments (Perez et al. 1994). Quelques algorithmes (Goodrich, 1994) en complexité $O(nlog(n))$ (Agarwal, 2002) ont été développés pour les courbes planes, mais cette complexité n'a pas été atteinte pour le cas général dans R^d. Il existe par ailleurs des méthodes sous optimales, mais en complexité linéaire, basées sur la sélection de points importants (Fink & Pratt, 2003), ou sur l'utilisation du test du CUSUM pour l'extraction de tendance et la segmentation des séries temporelles.

Dans ce contexte, nous proposons un algorithme de complexité intermédiaire qui contraint la recherche des segments linéaires en imposant que leurs extrémités soient sur la trajectoire $X(t)$. Ainsi, θ est l'ensemble des instants discrets $\{n_i\}$ qui correspondent aux extrémités des segments. Puisque la fin d'un segment correspond au début du segment suivant, deux segments successifs partagent un n_i commun à leur interface. La sélection de l'ensemble optimal des paramètres $\hat{\theta} = \{\hat{n}_i\}$ est obtenue par une approche classique de programmation (Bellman, 1957) comme décrit ci-dessous.

Nous définissons en premier lieu le taux de compression ρ d'une approximation linéaire par morceaux par :

$$\rho = 1 - \frac{|\{n_i\}|}{|\{X(n)\}|} \times \frac{p+1}{p}$$

où $|A|$ désigne le cardinal de l'ensemble A et $X(n) \in \Re^p, \forall n$

Etant donné une valeur pour ρ et la taille de la fenêtre temporelle sur laquelle nous souhaitons sous échantillonner la trajectoire multidimensionnelle $w=|\{X(n)\}_{n\in\{1,..,w\}}|$, le nombre $N=|\{n_i\}|-1$ de segments linéaires est connu.

Si $\theta(k)$ représente l'ensemble des paramètres associés à une approximation linéaire par morceaux contenant k segments, et si $\delta(k,i)$ désigne l'erreur minimale pour la meilleure approximation contenant k segments et recouvrant la fenêtre temporelle discrète $\{1,..,i\}$, alors $\delta(k,i)$ s'exprime de la manière suivante :

$$\delta(k,i) = \underset{\theta(k)}{Min}\left\{ \sum_{n=1}^{i} \left\| X_{\theta(k)}(n) - X(n) \right\|^2 \right\}$$

D'après le principe d'optimalité de Bellman, $\delta(k,i)$ peut être décomposé comme suit :

$$\delta(k,i) = \underset{k-1 \le n_k \le i}{Min} \left\{ d(n_k, i) + \delta(k-1, n_k) \right\}$$

où $d(n_k, i) = \sum_{n=n_k}^{i} \left\| Y_{k,i}(n) - X(n) \right\|^2$ et $Y_{k,i} = (X(i) - X(n_k)).\frac{n - n_k}{i - n_k} + X(n_k)$

est le segment linéaire entre $X(i)$ et $X(n_k)$.

L'initialisation de la récurrence est obtenue en observant que :

$$\forall k,\ \forall i < k,\ \delta(k,i) = 0$$

L'approximation est obtenue à la fin de la récurrence grâce à la détermination de l'ensemble des instants d'échantillonnage correspondant aux extrémités des segments linéaires :

$$\hat{\theta}(k) = \underset{\theta(k)}{ArgMin}\left\{ \sum_{n=1}^{w} \left\| X_{\theta(k)} - X(n) \right\|^2 \right\}$$

Ce modèle optimal conduit à l'erreur minimale :

$$\delta(k,w) = \sum_{n=1}^{w} \left\| X_{\hat{\theta}(k)}(n) - X(n) \right\|^2$$

La complexité de cet algorithme est en $O(k.w^2)$. Afin de réduire cette complexité, la fenêtre de recherche des extrémités de segments peut être diminuée en utilisant une borne inférieure de recherche à chaque étape i: $lb=max\{i-band,0\}$, où $band$ est un paramètre fixé par l'utilisateur :

$$\delta(k,i) = \underset{lb \le n_k \le i}{Min}\left\{ d(n_k, i) + \delta(k-1, n_k) \right\}$$

En pratique, nous utilisons $band = 2*w/k$, ce qui conduit à une complexité en $O(w^2/k)$.

Un exemple de sous échantillonnage adaptatif est présenté en Figures *FIG. 1* et *FIG.2* pour l'attracteur de Lorentz (Lorentz, 1963).

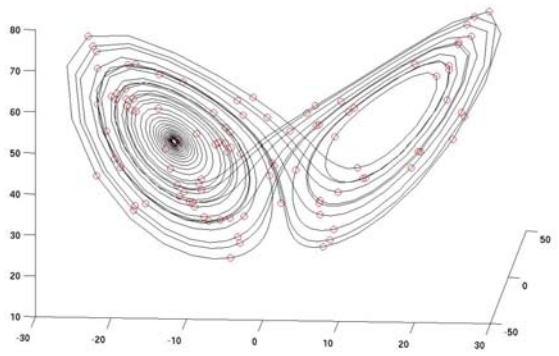


FIG. 1 – *L'attracteur de Lorentz en dimension trois et la localisation des points de sous échantillonnage avec un taux de compression de 90%.*

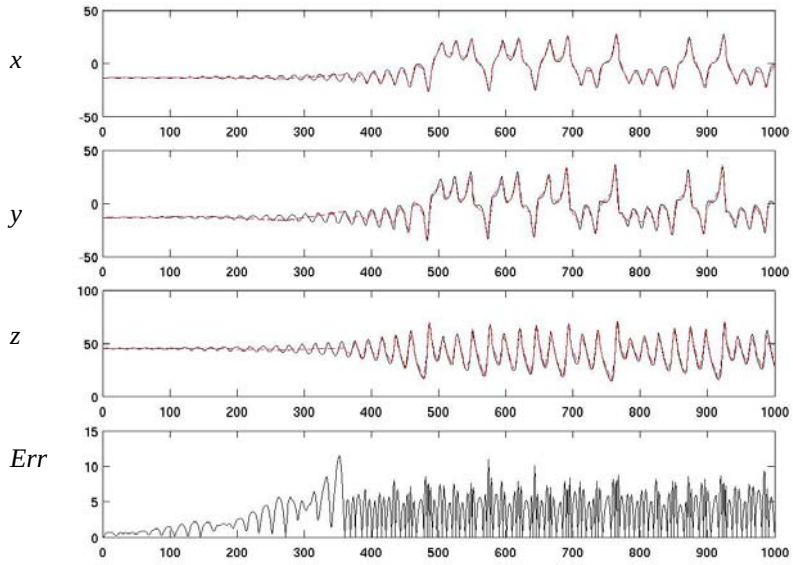


FIG. 2 – *Les trois variables x,y,z de l'attracteur de Lorentz pour un taux de compression de 90% ; les signaux non compressés sont représentés en traits pleins (noirs), les signaux compressés reconstruits sont représentés en traits pointillés (rouges).Err représente l'erreur quadratique entre la trajectoire non compressée et la trajectoire compressée reconstruite par segments linéaires.*

3 Expérimentations

3.1 Les données

La tâche artificielle “cylinder-bell-funnel” a été proposée initialement par Saito (Saito, 1994), puis ensuite exploitée par Manganaris (Manganaris, 1997) et plus récemment par Keogh et al. (Keogh et al. 2002). La tâche consiste à classer un ensemble de séries temporelles en trois classes, « cylinder (*c*), bell (*b*) ou funnel (*f*) ».

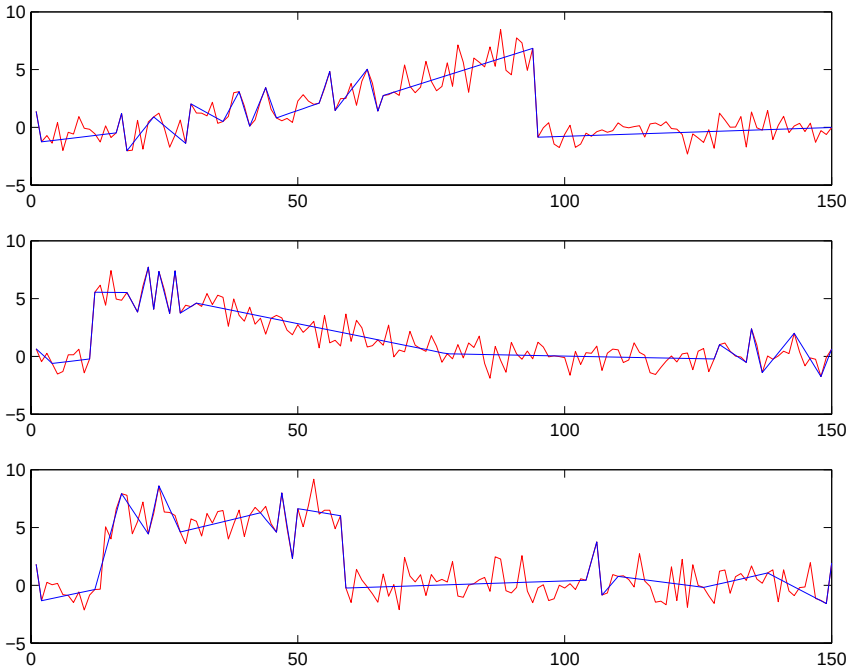


FIG. 3 – *Exemples de fonctions «cylinder (en bas), bell (en haut) et funnel (au milieu)» présentées avec leur approximation obtenues pour un taux de compression de 87%.*

Les séries temporelles sont synthétisées comme suit :

$$c(t) = (6+\eta).\chi_{[a,b]}(t) + \varepsilon(t)$$

$$b(t) = (6+\eta).\chi_{[a,b]}(t).\frac{t-a}{b-a} + \varepsilon(t)$$

$$f(t) = (6+\eta).\chi_{[a,b]}(t).\frac{b-t}{b-a} + \varepsilon(t)$$

avec : $\chi_{[a,b]} = \begin{cases} 0 \text{ si } t < a \\ 1 \text{ si a} \le t \le b \\ 0 \text{ si } t > b \end{cases}$

η et ε(t) tirés aléatoirement à partir de la distribution normale N(0,1)

a est un entier tiré uniformément dans l'intervalle $[16, 32]$

et (b - a) est tiré uniformément dans l'intervalle $[32, 96]$

La figure *FIG.3* montre la forme de ces fonctions ainsi que leurs approximations respectives obtenues par échantillonnage adaptatif.

3.2 Protocole expérimental

20 séries de 9 courbes sont engendrées. Pour chaque série, 3 courbes appartiennent à la classe *c(t)*, 3 à la classe *b(t)* et 3 à la classe *f(t)*. Sur chaque série, on effectue une classification selon l'approche du laissé pour compte : à tour de rôle, l'une des courbes est comparée aux 8 autres. Si le plus proche voisin ne relève pas de la même classe, une erreur est comptabilisée. Pour chaque méthode expérimentée, la moyenne et l'écart type de l'erreur évaluée sur les 20 séries sont calculés.

3.3 Résultats

La table *TAB.1* présente de manière comparée les approches par échantillonnage adaptatif par PLA et par DFT. Pour la PLA, la distance euclidienne ED (méthode ED-PLA) ainsi que la comparaison dynamique DTW (méthode DTW-PLA) sont exploitées.

Ces résultats montrent que la comparaison dynamique est particulièrement adaptée à la tâche de classification considérée (DTW). Les gains apportés par cette méthode sont de 57% vis-à-vis de la méthode exploitant une distance euclidienne (ED) et de 30% vis-à-vis de la méthode par coefficient de fourrier (DFT). L'échantillonnage adaptatif associé à une comparaison dynamique (DTW-PLA) ne dégrade que légèrement les performances de classification comparativement à l'approche sans échantillonnage adaptatif exploitant une comparaison dynamique. On peut observer que pour un taux de compression inférieur ou égal à 25%, les résultats sont comparables. Pour des taux de compression de l'ordre de 75%, la perte est seulement de l'ordre de 5% en moyenne. La dégradation est significative pour des taux de compression de l'ordre de 90%. La méthode de compression par DFT simple semble peu

sensible au taux de compression et conduit à des taux d'erreur moyen qui varient entre 30% et 45%. Lorsque celle-ci est associée à l'échantillonnage adaptatif, les résultats se dégradent.

On peut noter enfin, à titre de comparaison, qu'un classifieur aléatoire qui consiste à affecter à la série temporelle de test la classe d'appartenance d'une série aléatoire tirée aléatoirement parmi les 8 restantes conduit à un taux d'erreur de 75%. Toutes les méthodes présentées font donc mieux qu'un classifieur aléatoire.

Méthode	**Taux d'erreur moyen**	**Ecart type**
DTW ρ = 0%	0.0056	0.0248
DTW-PLA ρ = 5%	0.0056	0.0248
DTW-PLA ρ = 10%	0.017	0.041
DTW-PLA ρ = 25%	**0.00**	**0.00**
DTW-PLA ρ = 50%	0.028	0.049
DTW-PLA ρ = 75%	0.05	0.092
DTW-PLA ρ = 90%	0.328	0.175
ED ρ = 0%	0.578	0.133
ED-PLA ρ = 5%	0.161	0.132
ED-PLA ρ = 10%	0.228	0.117
ED-PLA ρ = 25%	0.156	0.115
ED-PLA ρ = 50%	0.205	0.166
ED-PLA ρ = 75%	0.333	0.184
ED-PLA ρ = 90%	0.583	0.213
DFT ρ = 0%	0.311	0.152
DFT ρ = 5%	0.372	0.158
DFT ρ = 10%	0.433	0.129
DFT ρ = 25%	0.406	0.181
DFT ρ = 50%	0.317	0.174
DFT ρ = 75%	0.322	0.200
DFT ρ = 90%	0.300	0.191
DFT-PLA ρ = 5%	0.478	0.188
DFT-PLA ρ = 10%	0.422	0.164
DFT-PLA ρ = 25%	0.450	0.175
DFT-PLA ρ = 50%	0.511	0.174
DFT-PLA ρ = 75%	0.533	0.203
DFT-PLA ρ = 90%	0.638	0.210

TAB. 1 – *Résultats présentés sous la forme de taux d'erreur moyen et écarts types associés pour les approches PLA-DTW, PLA-ED et DFT .*

4 Conclusion

Les résultats de cette étude préliminaire montrent que, sur les données synthétiques traitées, des taux de compression important peuvent être obtenus par des techniques d'échantillonnage adaptatif sans pour autant dégrader les résultats de classification de manière trop critique. Ces résultats sont obtenus lorsque la comparaison dynamique est utilisée pour évaluer la similarité des séries temporelles. Pour quelques valeurs faibles du taux de compression, le nombre erreurs de classification est même inférieur à la situation sans compression d'information lorsque l'on exploite la comparaison dynamique. Cette approche, lorsque comparée à la technique de réduction de dimension par coefficients de Fourrier, offre des gains important de performances de classification (de 15 à 20 %). Ces gains sont toutefois à mettre en correspondance avec les complexités des algorithmes mis en œuvre. La méthode d'échantillonnage adaptatif proposée permet de gagner un facteur 16 (1 ordre de grandeur) pour un taux de compression de 75% par rapport à une comparaison dynamique effectuée dans l'espace originel non réduit, tout en préservant un gain de performance de l'ordre de 31% par rapport à l'approche par DFT. Un compromis « complexité/qualité » régit, comme on pouvait s'y attendre, le comportement de cette méthode.

Ces premiers résultats encouragent à explorer plus avant ces techniques de réduction de dimensionnalité à base d'échantillonnage adaptatif susceptibles d'être exploitées dans le cadre d'application de datamining à grande échelle. Dans ce but, des ouvertures vers les travaux permettant de réduire la complexité des algorithmes présentés devront être explorées.

Références

R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In Proc. 4th Int. Conf. of Foundations of Data Organization and Algorithms, pages 69–84, 1993.

R. Agrawal, G. Psaila, E. L. Wimmers, and M. Za¨ıt. Querying shapes of histories. In Proc. 21th Int. Conf. on Very Large Data Bases, pages 502–514, 1995.

J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In Proc. 8th Int. Symp. on Storage and Retrieval for Image and Video Databases, pages 170–179, 1996.

L. Chen, R. Ng, "On the Marriage of Lp-Norm and Edit Distance", In *Proceedings of 30th International Conference on Very Large Data Base*, Toronto, Canada, August 2004, pages 792-803.

C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In Proc. ACM SIGMOD Int. Conf. on Management of Data, pages 419–429, 1994.

E. Fink and B. Pratt, "Indexing of compressed time series," Data Mining in Time Series Databases, pp.51-78, 2003.

L. Chen, M. T. ¨ Ozsu, and V. Oria. Robust and efficient similarity search for moving object trajectories. In CS Tech. Report. CS-2003-30, School of Computer

Chan, K. & Fu, A. W. (1999). Efficient time series matching by wavelets. In proceedings of the 15th IEEE Int'l Conference on Data Engineering. Sydney, Australia, Mar 23-26. pp 126-133. Science, University of Waterloo.

Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, S. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. In proceedings of ACM SIGMOD Conference on Management of Data. Santa Barbara, CA, May 21-24. pp 151-162.

E. J. Keogh, K. Chakrabarti, S. Mehrotra, M. J. Pazzani: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. SIGMOD Conference 2001

E. Keogh. Exact indexing of dynamic time warping. In Proc. 28th Int. Conf. on Very Large Data Bases, pages 406–417, 2002.

S Kim, S. Park, and W. Chu. An indexed-based approach for similarity search supporting time warping in large sequence databases. In Proc. 17th Int. Conf. on Data Engineering, pages 607–614, 2001.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, 6:707-710, 1966

E. N. Lorenz, "Deterministic Non-periodic Flow," *J. Atmos. Sci.*, 20, 130-141, 1963.

P.-F. Marteau and S. Gibet. Adaptive Sampling of Motion Trajectories for Discrete Task-Based Analysis and Synthesis of Gesture. In *LNCS/LNAI Proc. of Int. Gesture Workshop*, volume vol. of series, page pages, Vannes, France, month 2005. Publisher.

E. Fink and B. Pratt, "Indexing of compressed time series," *Data Mining in Time Series Databases*, pp.51-78, 2003.

Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoustics Speech Signal Process ASSP 26:43–49

B-K Yi, H. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In Proc. 14th Int. Conf. on Data Engineering, pages 23–27, 1998.

Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In Proc. ACM SIGMOD Int. Conf. on Management of Data, pages 181–192, 2003

Summary

We develop in this paper an experimental study in order to analyze the impact of dimensionality reduction techniques commonly used in time series Information Retrieval. We characterize this impact while considering a supervised classification task. The dimension reduction is tackled by means of adaptive sampling techniques based on a piecewise linear approximation of time series. This technique is compared with a DFT approximation scheme. The experimental study is based on a synthetic dataset, namely the so called “Cylinder, Bell, Funel” functions, that define a three class noisy problem. For various compression rates, classification error rates are evaluated. This preliminary study shows that adaptive sampling do not affect drastically the classification rates, even for high compression rates.

Analyse de trajectoires hospitalières de patients atteints d'un infarctus du myocarde

Myriam Touati*, Mohamed Cherif Rahal*
Catherine Quantin**, Gwénaël Le Teuff**, Mehdi Limam*,
Filipe Afonso*,Giampaolo Bataglia*
*CEREMADE - UMR 7534 Université de Paris-Dauphine
75775 PARIS CEDEX 16 FRANCE
nom@ceremade.dauphine.fr,
**Service de Biostatistique et Informatique Médicale,
Centre Hospitalier du Bocage, BP 77908,
21079 DIJON CEDEX
catherine.quantin@chu-dijon.fr
dim@chu-dijon.fr

Résumé : Ce travail consiste à développer une méthodologie d'analyse du parcours des patients au sein de la succession des différents services hospitaliers, par l'intermédiaire de l'analyse de données symboliques (A.D.S.). L'objectif est de définir et de construire une trajectoire pour chaque patient, cette trajectoire devenant le concept (objet symbolique) à analyser, puis de les agréger en classes de trajectoires en vue de les interpréter par l'A.D.S. Pour ce faire, il a été proposé d'utiliser un panel de méthodes diversifiées (formes fortes, recherche de motifs fréquents pour données symboliques, méthode hiérarchique, arbre de décision symbolique,...). Une fois les trajectoires et les classes de trajectoires construites, et validées par les experts médecins, ces concepts ont été analysés de manière à déterminer si la trajectoire d'un patient atteint d'un Infarctus Aigu du Myocarde (I.D.M) est liée à sa survie.

1 Introduction

L'étude réalisée porte sur le projet hospitalier de recherche clinique de cardiologie (PHRC) qui a pour objectifs de savoir dans quelle mesure on peut évaluer l'impact de l'hospitalisation d'un patient souffrant d'un infarctus du myocarde lorsque celui-ci rentre ou non dans un service de cardiologie. En effet, il est important de pouvoir déterminer si le risque des patients atteints d'IDM aigu varie en fonction de l'hospitalisation de première intention dans un service cardiologique/non cardiologique. Ce risque est évalué en fonction du suivi dans le temps de ces individus, plus particulièrement s'ils ont décédé dans un délai de 30 jours à une année après hospitalisation, ou s'ils ont été réadmis en service d'hospitalisation dans l'année qui suit (pour toutes causes dont celles cardiovasculaires). Sont à distinguer en fonction de la gravité de l'infarctus aigu du myocarde : les patients qui ont intégré en premier lieu un service de cardiologie et ceux qui n'ont pas intégré en premier lieu un service de cardiologie. En outre, il sera nécessaire de distinguer différentes catégories de la population, par analyse des caractéristiques épidémiologiques, comme les caractéristiques démographiques (âge, sexe, zone de résidence) ainsi que les facteurs de risque (tabac, diabète, hérédité, obésité),

les antécédents cardio-vasculaires, les caractéristiques cliniques de l'IDM (localisation de l'IDM, choc cardiogénique...), les éléments de revascularisation (angioplastie, pontage...) ainsi que les éventuelles complications qui pourraient survenir (rythmiques, AVC, récidive d'IDM, ischémie récurrente). L'objectif serait de pouvoir distinguer dans quelle mesure les trajectoires des patients décrits par leurs caractéristiques épidémiologiques pourraient être liés à leur survie.

2 Les données

Les individus intervenant dans l'étude sont des patients souffrant d'un Infarctus du myocarde en phase aiguë, qui ont été hospitalisés dans tout service de l'un des cinq établissements de Côte d'Or (4 du secteur public, 1 du secteur privé) du 3 Mai 2001 au 28 Décembre 2002. Il est important de rappeler que les individus composant l'étude peuvent être admis dans une succession de services hospitaliers qui forme alors une trajectoire. Chaque phase de la séquence formée est constituée, en termes de codification :

- du code du service dans lequel il est admis
- du code de l'hôpital correspondant au service

En effet, il est possible d'avoir le même code de service pour deux hôpitaux différents. Il est donc nécessaire de préciser quel hôpital est concerné. La source des données est issue du PMSI -Programme de Médicalisation des Systèmes d'Information –dont l'objectif est de permettre aux établissements hospitaliers de coder les diagnostics et actes médicaux ou chirurgicaux du séjour dans une unité fonctionnelle d'un service donné (c'est-à-dire le service hospitalier concerné). Grâce au PMSI, il a été possible de recevoir, deux bases de données en vue d'étudier le risque de survie des patients en fonction des trajectoires réalisées. Une première base (base1.xls) représente l'ensemble des séjours hospitaliers que le patient a réalisé à l'issue de l'infarctus aigu du myocarde. Les informations contenues dans cette base peuvent être considérées comme mobiles ou dynamiques dans la mesure où chaque ligne représente le passage d'un patient au sein d'une Unité Fonctionnelle d'un hôpital, avec le Résumé d'Unité Médicale (RUM) correspondant. Ainsi, on retrouvera plusieurs résumés d'unité médicale d'un même patient si celui-ci a fréquenté plusieurs services ou unités fonctionnelles d'un ou plusieurs hôpitaux. Une seconde base (RICO Complet.xls) représente les informations biologiques de chaque patient récoltées lors de la première admission en services d'hospitalisation. Ces données ne sont constituées qu'une seule fois, et sont donc considérées comme statiques dans la mesure où elles ne génèrent pas particulièrement de modifications en fonction des différents passages des patients dans les Unités Fonctionnelles. Les deux bases possèdent toutes deux la clé Norico, unique pour chaque patient, et peuvent donc être jointes pour représenter la totalité de l'information.

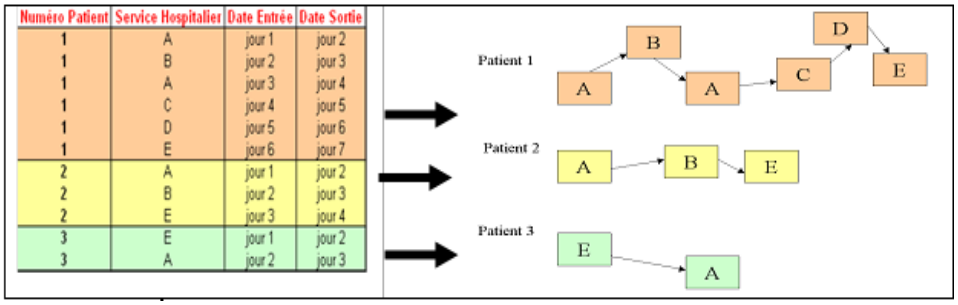


FIG. 1 – *Passage des données initiales aux trajectoires par patient*

3 Construction des trajectoires-patient et des trajectoires-type

3.1 Définition

Au sein du domaine hospitalier, la trajectoire des patients que nous avons réalisée suit les analyses de Dart et al (2003) et de Leduff et al. (2001) dans lesquelles une importance particulière a été accordée au type d'évènement lié à l'hospitalisation, ses caractéristiques et son aspect temporel.

Ainsi, la définition de la trajectoire hospitalière est basée sur 3 types d'information :

- le type d'hôpital dans lequel le patient est affecté
- le type du service hospitaliser dans lequel le patient est affecté : on l'appelle Unité Fonctionnelle
- l'ordre chronologique d'admission dans chaque unité de soin.

3.2 Caractéristiques des trajectoires

Un patient qui souffre d'infarctus aigu du myocarde (IDM) peut être accueilli :

- dans une unité fonctionnelle (UF) relative à un service cardiologique puis soit :
  a) être transféré dans le service de cardiologie d'un autre hôpital
  b) ne pas subir de mutation ou de transfert au court du même séjour
- dans une UF n'étant pas relative à un service de cardiologie, puis soit :
  a) subir une mutation dans un service de cardiologie du même établissement (au cours du même séjour)
  b) être transféré dans un service de cardiologie d'un établissement différent
  c) n'être ni transféré, ni muté

Ainsi, à chaque patient correspond une et une seule trajectoire alors qu'une trajectoire peut correspondre à plusieurs patients.

N° Patient (RICO)	Unité Fonctionnelle	Date Entrée	Date Sortie	Etablis	Diagnostic Principal	Diagnostic 2	Acte 1	Acte 2	Age	Sexe
1	UF1	15/05/2001	18/05/2001	729	[...]	[...]	[...]	[...]	23	H
1	UF4	18/05/2001	22/05/2001	710	[...]	[...]	[...]	[...]	23	H
1	UF7	22/05/2001	23/05/2001	710	[...]	[...]	[...]	[...]	23	H
[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]	[...]
5	UF7	[...]	[...]	[...]	[...]	[...]	[...]	[...]	45	F

FIG. 2 – *Chaque ligne est un « RUM »*

Patient	Order_1	Order_2	Order_3	Order_4	[...]	Order_9
1	UF:3301_Beaune	UF:3001_H:Beaune	UF:3001_H:Beaune		[...]	
2	UF:3301_Beaune	UF:3001_H:Beaune	UF:2091_H:Dijon	UF:3001_H:Beaune	[...]	
3	UF:2092_Dijon	UF:2095_H:Dijon	UF:2092_H:Dijon	UF:2095_H:Dijon	[...]	UF:3001_H:Beaune

FIG. 3 – *Chaque ligne est un des 1095 patient décrit par sa trajectoire*

3.3 Construction des trajectoires

Les informations issues de la base contenant les Résumés d'Unités Médicales ont été réorganisées en agrégeant les patients de sorte à obtenir une ligne d'information par numéro RICO, donc par patient.

Etape 1 : Données Initiales :

Au départ, les informations relatives aux patients sont présentes dans une seule base (RUM), dans laquelle chaque ligne représente 1 passage d'un patient dans une Unité Fonctionnelle à un moment donné (voir la figure 2).

Etape 2 : Trajectoire de chaque patient :

On va tenter « d'agréger » l'information des différentes lignes concernant le même patient sur une seule. Pour ce faire :

- On regroupe les différentes Unités Fonctionnelles d'un même patient par ordre chronologique (de la date la plus ancienne à la plus récente).
- On affecte à l'UF par lequel est passé le patient à son arrivée la nomination d'UF d'ORDRE_1, la suivante étant d'ordre_2, etc.. . .

La succession de tous les ordres pour le patient i représente la trajectoire complète du patient i (Figure 3).

Etape 3 : Ajout de la bio-information aux 1095 patients décrits par leur trajectoire

A ce niveau, on agrège à la trajectoire de chaque patient, des variables « dynamiques » telles que ses diagnostics et ses actes et des variables « statiques » telles que ses variables biologiques, ces dernières étant extraites de la seconde base (appelée « RICO complet »). La nouvelle table obtenue (voir la figure 4) sera agrégée par la méthode DB2SO (Data Base to Symbolic Object)[1] du logiciel d'Analyse de Données Symboliques

[1] DB2SO A partir d'une base de données relationnelle, le logiciel SODAS élabore un répertoire de données symboliques, pouvant comporter un ensemble de règles de taxonomies.

N_Patients	Trajectoire complète	SEXE	Age	IDM	Diabete	Angine _Poitrine	[...]	Tabagisme
5	3301xBeaune;3001xBeaune;2092xDijon;3001xBeaune;x;x ;	H	45	0	Non	Oui	[...]	Oui
...	...	..	..	..	..	...	...	...
17	3301xBeaune;3001xBeaune;x;x;x;x;x;x	H	75	0	Oui	Oui	[...]	Non
18	3301xBeaune;3001xBeaune;x;x;x;x;x;x	F	81	0	Oui	Non	[...]	Non

FIG. 4 – *Trajectoire et bio-information agrégées pour les 1095 patients*

SODAS, qui prendra, comme concept : les patients.

Etape 4 : Construction des 204 trajectoires-type représentant un tableau de données symbolique

Or, parmi les 1095 patients décrits dans la figure 4, plusieurs ont réalisé à une date différente mais dans le même ordre, la même trajectoire. Il s'agit de trajectoires-type parcourues par l'ensemble des patients hospitalisés. La construction de ces trajectoires a été effectuée par la méthode des « Formes Fortes » qui en a fait émerger 204 (voir paragraphe d).

Sachant qu'une trajectoire peut correspondre à plusieurs patients, alors chaque trajectoire peut correspondre à :

- un intervalle d'âge, de poids, de tailles spécifiques
- une proportion de modalités variées (sexe, présence d'antécédents Oui/Non, diabète,..)

Ainsi, chaque trajectoire est caractérisée par l'agrégation des caractéristiques biologiques de l'ensemble des patients ayant parcouru cette trajectoire, à l'aide de la méthode DB2SO (Data Base to Symbolic Object) du logiciel SODAS, qui prendra, comme concept : les trajectoires. La nouvelle table obtenue à l'issue de cette étape est illustrée dans la figure 4.

Remarque : Les informations relatives aux patients, aussi bien quantitatives (poids, taille, fréquence cardiaque,...) que qualitatives (insuffisance cardiaque, antécédents IDM, diabète sucré, tabagisme ancien ou passif...) sont, dans certains cas, manquantes. Afin de remédier au problème des données manquantes, nous avons utilisé la méthode d'imputation employant l'algorithme de type Nuées Dynamiques Symbolique appelé SCLUST[2] *dans le Logiciel SODAS sur la base entière. En étendant l'algorithme des K-Means au cas de données symboliques, il est possible d'obtenir une partition de la population dont chaque classe est représentée par un prototype représentatif sous forme d'une conjonction de propriétés. En outre, cet algorithme fonctionne en cas de données manquantes en utilisant uniquement les données présentes. Il est, de plus, nécessaire que le nombre de classes ne soit pas trop élevé afin de s'assurer que les variables ont au moins une variable à valeur présente dans chaque classe.*

[2] *SCLUST : Symbolic dynamic CLUSTering : Méthode de classification symbolique de type Nuées Dynamiques du logiciel SODAS*

TRAJECTOIRE	ANGINE DE POITRINE	DUREE D'HOSPITALISATION	REHOSPITALISATION	...	SEXE
2095*710;2092*710;x;x;x;x;x;x;x;x;	NON(0.706),OUI(0.294)	[4.00;19.00]	NON(0.765),OUI(0.235)		M(0.588),F(0.412)
.....	.....	.....	.....	....	.....
6*144;x;x;x;x;x;x;x;	NON(0.767),OUI(0.183),NA(0.050)	[0.00;0.00]	NON(0.917),OUI(0.083)		M(0.783),F(0.217)

FIG. 5 – *Les 204 trajectoires et leur description biologique*

3.4 Méthode des « Formes Fortes » pour la construction des trajectoires-type (204)

Afin de pouvoir réaliser des analyses pertinentes du type de trajectoire des patients en fonction des caractéristiques de survie, nous avons utilisé la méthodologie des Formes Fortes pour : traiter la base de données initiale afin de faire émerger les trajectoires et les agencer afin de traiter les motifs qui apparaissent le plus souvent en première UF de la trajectoire nouvellement formée (ce qui correspond à l'entrée du patient en structure hospitalière). L'algorithme consiste à rechercher des séquences de trajectoires-type de patients en privilégiant l'ordre d'arrivée « du début vers la fin », la consécution des Unités Fonctionnelles au sein des différents hôpitaux, l'ordre de fréquence d'apparition (décroissant) et l'élimination des services de fréquence d'apparition (dans la base) inférieure à un seuil epsilon. La méthode des Formes Fortes permet d'établir une répartition des trajectoires en termes de taille (longueur maximale de la trajectoire, qui équivaut au nombre maximal de visites dans les différentes unités fonctionnelles de l'hôpital) ainsi qu'en termes de pourcentage de la population couverte. Au final, on remarque que la base est constituée de 204 formes fortes, qui représentent la totalité des individus (au total 1095 patients). Sur ces 204 trajectoires, 14 d'entre elles (soient 6.73%) représentent 796 individus (soient 71.91% de la population totale). Les 190 trajectoires restantes (93.27% de l'ensemble) représentent pour le 311 patients (soient 28.1%). En termes de longueur des séquences, on remarque que la majorité des trajectoires sont de taille 1 et 2. On note que la proportion de trajectoires augmente très rapidement jusqu'à la taille 4, puis se stabilise, le nombre maximal de visites effectuées dans toute la base étant de neuf Unités Fonctionnelles. En considérant, en terme de nombre de patients couverts, les 14 premières trajectoires distinctes (voir la figure 6) - avec une valeur de l'epsilon fixée à 7 patients - on remarque qu'elles sont toutes de taille maximale égale à 4 ordres (au 5ème ordre, tous les patients quittent les Unités Fonctionnelles). Notons que 3 trajectoires (en orange) ne possèdent pas en 1er ordre un service cardiologique, et parmi celles-ci, une seule n'est pas suivie d'un service cardiologique. En définitive, cette méthode nous a permis de mettre en avant l'ensemble des trajectoires complètes, en terme de « formes fortes », ainsi que celles qui sont les plus fréquentes.

Ordre 1	Ordre 2	Ordre 3	Ordre 4	Ordre 5	Somme
S: Cardio.II ; H: Dijon	x	X	x	x	211
S: Cardio.II ; H: Dijon	S: Cardio.II ; H: Dijon	X	x	x	197
S: Cardio.II ; H: Dijon	S: Cardio.I ; H: Dijon	X	x	x	112
S:Soins int.cardio; H: Fontaine	x	X	x	x	109
S:Coro; H: Fontaine	x	X	x	x	67
S:USIC; H: Semur	S: Cardio.II ; H: Dijon	S:Med.Cardio.; H: Semur	x	x	20
S:USIC; H: Semur	S:Med.Cardio.; H: Semur	X	x	x	13
S:Medecine 1; H: Beaune	S: Cardio.II ; H: Dijon	S:Medecine 1; H: Beaune	x	x	12
S: Cardio.II ; H: Dijon	S: Cardio.II ; H: Dijon	S: Cardio.II ; H: Dijon	x	x	11
S:Réanimation; H: Beaune	S: Cardio.II ; H: Dijon	S:Medecine 1; H: Beaune	x	x	10
S: Cardio.II ; H: Dijon	S: Rea. SIC ; H: Dijon	S: Chir.CardioVasculaire ; H: Dijon	x	x	9
S:USIC; H: Semur	S:Med.Cardio.; H: Semur	S: Cardio.II ; H: Dijon	S:Med.Cardio.; H: Semur	x	9
S:USIC; H: Semur	S: Cardio.II ; H: Dijon	S:USIC; H: Semur	S:Med.Cardio.; H: Semur	x	8
S:Réanimation; H: Beaune	S:Medecine 1; H: Beaune	X	x	x	8

FIG. 6 – *Les trajectoires les plus fréquentes obtenues par la méthode des "formes fortes" avec un seuil de 8*

4 Classification de trajectoires

A ces trajectoires, il est possible de passer à un niveau d'agrégation supérieur en formant des classes regroupant les trajectoires qui pourraient se ressembler (ou, au moins, avoir des motifs identiques au sein de leur séquence). En fonction des méthodes spécifiques employées (Recherche des motifs fréquents, classification hiérarchique, Expert Médecin), les classes de trajectoires obtenues ne seront pas les mêmes. Nous avons intégré ces orientations en concevant nos classes de trajectoires

- par regroupement symbolique avec le programme de recherche d'associations S_APRIORI
- par « regroupement expert » par l'intermédiaire du groupe médical de Dijon.

4.1 Recherche de profils fréquents par la méthode S_APRIORI (Symbolic Apriori)

La méthode de recherche de règles d'associations et de motifs fréquents, décrite pour la première fois dans Agrawal et al. (1995) a pour objectif d'analyser une série de données afin d'en extraire des règles et des motifs fréquents (ie. les séquences, consécutives ou non, les plus fréquentes) qui en découlent. Cette démarche a été utilisée pour le regroupement de trajectoires Xing et al. (2004), dans le cadre de visites d'internautes sur un site Web (web usage mining). Après avoir retranscrit toutes les pages visitées par consultation (fichiers .log), les auteurs ont tenté de déterminer des classes de trajectoires par arbre de décision en vue d'optimiser le site en question. Les recherches dans Afonso (2004) se situent dans le prolongement de cette méthode : la recherche de règles d'association et de motifs fréquents est développée, en parvenant à prendre en compte, en plus, l'approche symbolique.

On peut donc réaliser une recherche non seulement par rapport aux individus mais aussi avec les concepts : en prenant en entrée un ensemble d'informations recensées dans le temps, il est possible de trouver l'ensemble des séquences présentant un support supérieur à S0, paramètre de la méthode. Ainsi, les séquences trouvées sont appelées

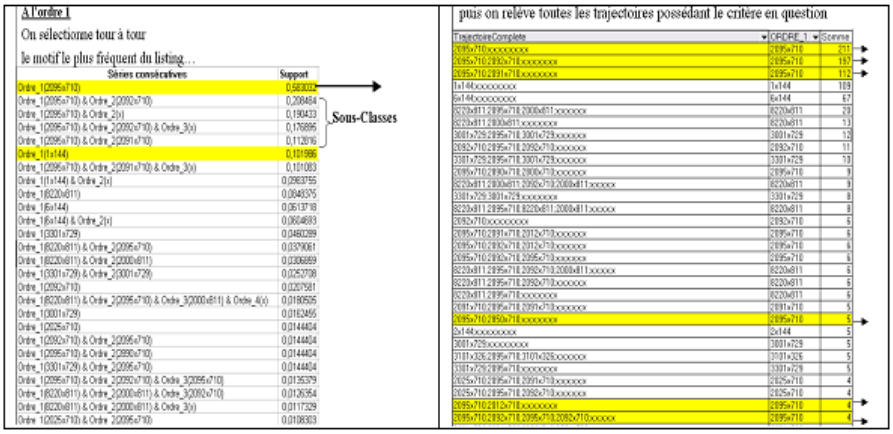


FIG. 7 – *Classification des motifs selon leur fréquence*

Classes sélectionnées	Nb_Individus
Ordre_1(1x144)	110
Ordre_1(2025x710)	20
Ordre_1(2092x710)	22
Ordre_1(2095x710)	616
Ordre_1(3001x729)	29
Ordre_1(3301x729)	56
Ordre_1(6x144)	68
Ordre_1(8220x811)	94
Ordre_2(2095x710)	52
Ordre_2(8220x811)	18
Ordre_3(2000x811)	1
Ordre_3(2095x710)	5
Ordre_3(3001x729)	1
Ordre_3(8220x811)	2
Ordre_1(2x144)	5
Ordre_1(8102x710)	1
Ordre_1(2115x710)	1
Ordre_1(2360x710)	1
Ordre_1(4x144)	1
Ordre_2(1x144)	2
Ordre_4(2095x710)	2

FIG. 8 – *21 Classes de trajectoires fréquentes*

séquences fréquentes. Ici, par exemple, il peut sembler particulièrement intéressant de faire émerger les services hospitaliers les plus fréquents, de façon à 1) élaborer des classes de trajectoires pertinentes ; 2) donner des éléments nouveaux et particulièrement pertinents aux experts métiers afin qu'il conçoivent, eux mêmes, des classes de trajectoires. Plusieurs approches et méthodes alternatives (dérivées de SAPRIORI) ont été réalisées pour obtenir des classes recouvrant tous les patients. L'approche que nous avons retenue est celle qui privilégie l'ordre initial des Unités Fonctionnelles, puis les fréquences d'apparition des motifs obtenus (en utilisant SAPRIORI). A partir des ordres de passage des patients dans les différents hôpitaux et UF, il est nécessaire de prendre les classes les plus fréquentes en observant d'abord les UF x Hôpitaux d'ordre 1, puis 2, puis 3, etc.... On va donc « trier » les ordres obtenus en fonction des rangs. A l'ordre 1, on sélectionne le motif (c'est l'Unité Fonctionnelle) le plus fréquent de la base, et on relève alors toutes les trajectoires possédant le critère en question (Figure 7) puis on réalise de même, pour tous les critères d'Ordre 1, en sélectionnant dans le tableau les motifs fréquents, puis en « flaguant » les trajectoires correspondantes dans la base (Figure 7). Pour les ordres suivants, on sélectionne, tour à tour, le motif le plus fréquent du listing tout en faisant attention à ne pas sélectionner des trajectoires ayant déjà été prises en compte. On réitère pour tous les ordres, de façon à ce que la totalité des trajectoires de la base soit prise en compte. Notons que les classes choisies dans la figure 8 possèdent, comme intitulé l'ordre qui correspond au rang sélectionné ainsi que la valeur de code de l'Unité Fonctionnelle, croisée avec le code de l'hôpital. Ainsi, on peut remarquer que la classe qui recouvre le plus grand nombre de patients est « ordre_1(2095x710) » signifiant qu'ont été sélectionnées toutes les trajectoires pour lesquelles les patients ont visité, en premier lieu (rang 1), l'Unité Fonctionnelle 2095 de l'hôpital 710. On peut remarquer que les catégories en gris clair, de rang 1, ont été privilégiées pour choisir les trajectoires par rapport à celles en gris pâle, de rang 2, elles mêmes privilégiées sur celles en gris foncé, de rang 3. Les catégories en gris moyen représentent les trajectoires qui sont considérées comme «orphelines» dans le sens qu'aucune nouvelle séquence fréquente ne les avait mises en valeur.

4.2 Le regroupement expert-médecin

En définitive, les différentes classifications automatiques effectuées ont permis de donner une idée d'ensemble de la répartition des trajectoires aux experts médicaux. En effet, ceux-ci se sont basés sur l'importance toute particulière à accorder aux deux premières Unités Fonctionnelles pour réaliser une nouvelle répartition des trajectoires en 8 catégories distinctes (voir figures 9 et 10).

5 Analyse des trajectoires-type et des classes de trajectoires

Une fois les trajectoires et les classes de trajectoires obtenues, on se demande si la survie des patients est influencée par leur trajectoire. Pour cela, on cherche les variables qui expliquent le mieux la survie dans le cas où les unités sont les patients puis quand

CLASSE_DIJON	Explication	NB_Trajectoires	NB Patients
Groupe_1	Mono unité; UF1=cardio	8	403
Groupe_2	Mono unité; UF1=non cardio	2	2
Groupe_3&4	Mutation: Mono établissement_Multi unité; UF1=cardio et UF2=non cardio et cardio	58	441
Groupe_5	Mutation : Mono établissement _ Multi unité; UF1=non cardio et UF2=non cardio	5	5
Groupe_6	Mutation : Mono établissement _ Multi unité; UF1=non cardio et UF2=cardio	32	43
Groupe_7	Transfert : Multi établissement _ Multi unité; UF1=cardio et UF2=non cardio	2	3
Groupe_8	Transfert : Multi établissement_ Multi unité; UF1=cardio et UF2=cardio	78	177
Groupe_9&10	Transfert : Multi établissement _ Multi unité; UF1=non cardio et UF2=non cardio et cardio	19	21
Total		204	1095

FIG. 9 – *Classification des trajectoires retenue par les experts médecin*

CLASS	EXPLANATION	SEX	AGE	AMI	.....	DIABETE	Nicotisme
Group_8	Transfer : multi_etablissement-multi_unite UF1=cardio & UF2=cardio	M(0.75),F(0.25)	[29.00 :93.00]	0(0.89), 1(0.10), NA(0.01)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.62), 0(0.36), NA(0.02)
Group_3&4	Mutation: mono_etablissement-multi_unite UF1=cardio & UF2=cardio or no cardio	M(0.70),F(0.30)	[25.00 :97.00]	0(0.84), 1(0.15), NA(0.01)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.64), 0(0.35), NA(0.01)
Group_1	mono unité UF1=cardio	M(0.71),F(0.29)	[21.00 :94.00]	0(0.88), 1(0.10), NA(0.02)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.61), 0(0.35), NA(0.04)
0Group_6	Mutation : mono_etablissement-multi_unite) UF1=no cardio & UF2=cardio	M(0.53),F(0.47)	[36.00 :91.00]	0(0.84), 1(0.16)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.44), 0(0.56)
Group_5	Mutation : mono_etablissement-multi_unite) UF1&UF2=no cardio	M(0.60),F(0.40)	[71.00 :89.00]	0(1.00)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.60), 0(0.40)
Group_2	mono unité UF1=no cardio	M(0.50),F(0.50)	[50.00 :70.00]	0(0.50), 1(0.50)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.50), 0(0.50)
Group_9&10	Transfer : multi_etablissement-multi_unite UF1=no cardio & UF2=no cardio or cardio	M(0.67),F(0.33)	[41.00:91.00]	0(0.80), 1(0.10), NA(0.10)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.48), 0(0.52)
Groupe_7	Transfer multi_etablissement-multi_unite UF1=cardio & UF2=no cardio	M(0.33),F(0.67)	[72.00:101.00]	0(0.67), 1(0.33)	.....	0(0.89), 1(0.10), NA(0.01)	1(0.33), 0(0.67)

FIG. 10 – *Description symbolique des classes de trajectoires retenues par les experts médecin*

se sont les trajectoires en utilisant deux méthodes : les arbres de décision symboliques et les étoiles.

5.1 Arbres de décision symbolique (SYCLAD)

Le programme SYCLAD de Limam (2005) est l'implémentation d'une méthode descendante hiérarchique divisant la population en deux classes en respectant à la fois l'homogénéité et la discrimination, mesurées par le paramètre α et en tenant compte d'une variable à expliquer appelée aussi « variable classe ou partition à priori». Le paramètre α varie entre 0 et 1 : il vaut 0 quand seule la discrimination est considérée et 1 quand seule l'homogénéité est considérée. Le programme calcule la valeur de α qui optimise à la fois l'homogénéité et la discrimination. Dans cette étude, nous nous sommes particulièrement intéressés aux résultats les plus significatifs, concernant la variable à expliquer DECES (booléenne oui/non), dont la répartition au sein de la population est décrite dans la figure 11. Notons que la répartition des experts médecin a été retenue pour réaliser les classifications par arbre de décision (SYCLAD). Deux classifications par arbre de décision ont été effectuées, la première portant sur les patients décrits par leurs trajectoires et leurs caractéristiques biologiques (Age, sexe, antécédents ... etc.), la seconde portant sur les trajectoires-types (204 trajectoires).

Décès	Nbr_Ind	%
Non (0)	928	84,75
Oui (1)	167	15,25
Total	*1095*	*100*

FIG. 11 – *Fréquence des décès au sein de la population (1095 patients)*

5.1.1 Analyse 1 : Les unités statistiques sont les patients

Deux analyses, portant sur les 1095 patients, ont été effectuées, autour de la variable à expliquer Décès Oui/Non, la valeur du α (critère d'homogénéité) est l'enjeu de ces études dans la mesure où l'on favorise un peu plus l'homogénéité dans la première analyse (α=0.3) que dans la deuxième (α=0). Pour la première analyse, on a la classe des experts métier comme principal critère de coupure (2 coupures successives). Les 31 patients discriminés par la première coupure (Nœud 1) font partie des classes expertes médecin n°9&10 (Transfert : Multi établissement Multi unité ; UF1=non cardio et UF2=non cardio et cardio), 2 (Mono unité ; UF1=non cardio), 5 (Mutation : Mono établissement _ Multi unité ; UF1=non cardio et UF2=non cardio), 7 (Transfert : Multi établissement _ Multi unité ; UF1=cardio et UF2=non cardio). Ils possèdent un taux de décès élevé par rapport au taux moyen de la population (29% contre les 15,25%). Sur les 1064 patients restants (Nœud 2), l'algorithme de classification réalise des coupures successives sur les 43 patients (Nœud 3) de la classe Dijon n°6 (Mutation : Mono établissement _ Multi unité ; UF1=non cardio et UF2=cardio), alors que les 1021 patients (Nœud 4) des classes 1(Mono unité ; UF1=cardio), 3&4 (Mutation : Mono établissement Multi unité ; UF1=cardio et UF2=non cardio et cardio) et 8 (Transfert : Multi établissement Multi unités ; UF1=cardio et UF2=cardio) sont réunis, avec une proportion de décès dans « la moyenne », à 14%. Nous remarquons ici que la première coupure discrimine de façon presque totale les classes experts médecin dont le premier UF est CARDIO des classes dont le premier UF est NON-CARDIO, ces dernières ayant un taux de décès de 30%, taux deux fois supérieur à la moyenne de 14% constatée pour la population totale.

Ceci nous montre donc l'influence de la trajectoire sur la survie du patient. Les autres variables les plus discriminantes sont le tabagisme et la classe d'age (>75 ans et < 75 ans) où l'on trouve que les patients qui ont plus de 75 ans ayant fumé représentent un taux de décès très élevé (30%).

Dans la deuxième analyse, on retrouve encore une fois la classe de l'expert médecin comme principal critère de coupe (avec 3 coupes successives). Les 1064 patients discriminés par la première coupe font partie des classes de l'expert métier au Nœud 2, numéro 1(Mono unité ; UF1=cardio), 3&4 (Mutation : Mono établissement_Multi unité ; UF1=cardio et UF2=non cardio et cardio), 6 (Mutation : Mono établissement _ Multi unité ; UF1=non cardio et UF2=cardio) et 8 (Transfert : Multi établissement_ Multi unité ; UF1=cardio et UF2=cardio). Ils possèdent un taux de décès égal à la moyenne pour la population totale (15%).

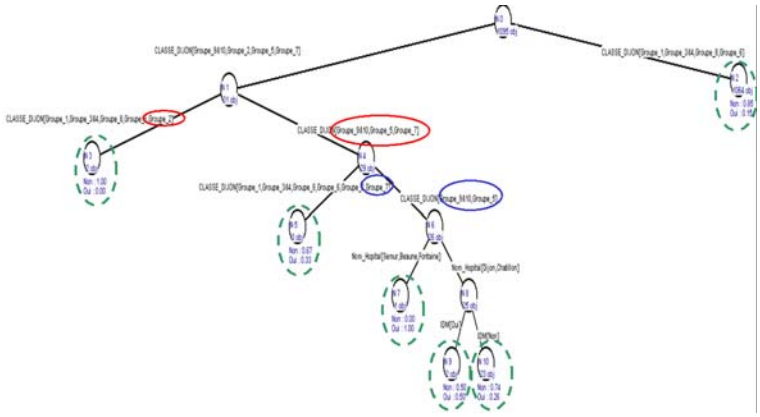


FIG. 12 – *Arbre de décision appliqué sur les 1095 patients avec α =0*

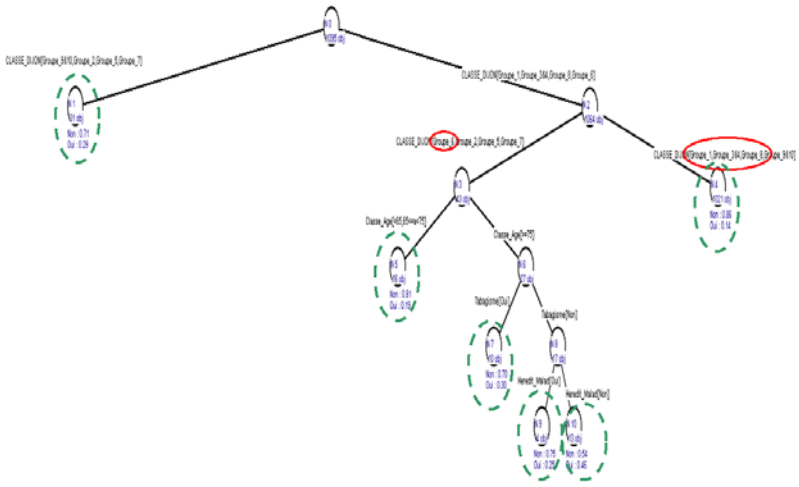
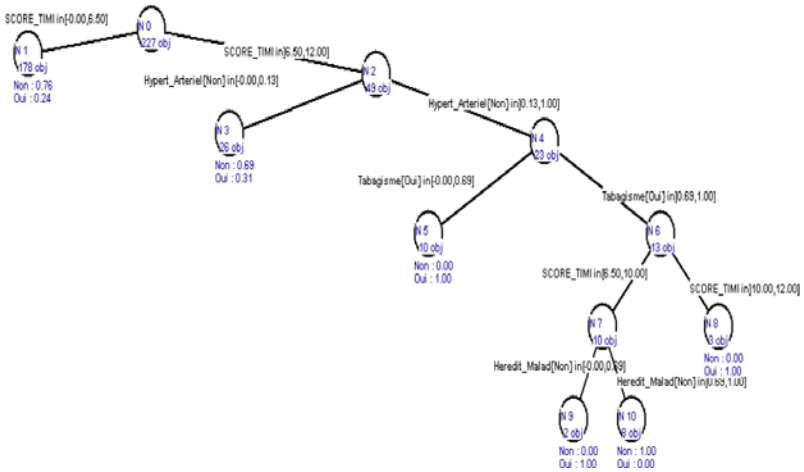


FIG. 13 – *Arbre de décision appliqué sur les 1095 patients avec α =0.3*

Ici, on remarque que le principal critère de coupure des patients est la variable « classes expertes médecin », qui réalise un découpage peu équilibré des patients (2.8% contre 97.2%). Donc la variable « classes expertes médecin », qui privilégie le type de la première et de la seconde UF (cardio ou non-cardio) explique le décès en premier. Les autres variables intervenant comme discriminantes sont : L'hôpital de la 1ère UF et les antécédents d'IDM.

5.1.2 Analyse 2 : Les unités statistiques sont les trajectoires

Nous présentons dans ce qui suit une analyse effectuée sur les trajectoires afin d'expliquer au mieux la variable décès. On étudie les 204 trajectoires croisées avec la variable DECES Oui/Non. En effet, l'une des particularités de la classification par arbre de décision (particulièrement la méthode SYCLAD) porte sur la nécessité de posséder des variables à expliquer « pures ».



Arbre de décision appliqué sur les trajectoires

Or, l'agrégation des patients en termes de trajectoires amène à former des concepts dont le décès est alors représenté par un histogramme. En solution, il a été choisi de croiser, au niveau du concept, les trajectoires avec la variable de décès, afin de conserver la pureté de celle-ci. Nous avons obtenu au final 227 objets autour de la variable à expliquer Décès Oui/Non. Les variables explicatives sont celles employées usuellement (Angine_Poitrine, Sexe, IDM, SusDecalage_IDM, Hypert_Arteriel, Hypercholestérolémie, Diabete, Heredit_Malad, Tabagisme, Nom_Hopital). A la demande des experts-médecin, on a rajouté une autre variable appelée Score TIMI (scores de gravité de l'IDM) réalisé par le CHU de Dijon. On observe que le Score TIMI apparaît comme premier critère de coupure. Il départage les 227 trajectoires croisées avec le décès en deux classes : la première de 178 trajectoires (qui représentent 809 patients), avec une proportion de décès pour 24% des trajectoires (et pour seulement 9.77% des patients et la seconde de 49 trajectoires (soit 286 patients) dont les moyennes des Scores TIMI sont

Nœud	NB_Traj	deces	NB_Patients	Proportion
N_1	178_traj.	Non	730	90,23
N_1	178_traj.	Oui	79	9,77
N_10	8_traj.	Non	175	100,00
N_3	26_traj.	Non	23	74,19
N_3	26_traj.	Oui	8	25,81
N_5	10_traj.	Oui	73	100,00
N_8	3_traj.	Oui	3	100,00
N_9	2_traj.	Oui	4	100,00

FIG. 14 – *Description des nœuds de l'arbre de décision*

plus élevées. On remarque également que les variables discriminantes des trajectoires sont le Score TIMI, l'antécédent d'hypertension artérielle (en particulier la modalité « Non »), l'antécédent de Tabagisme ancien ou actif (en particulier la modalité « Oui »), l'antécédent de Maladie Héréditaire (en particulier la modalité « Non »). En outre, le taux de décès est élevé avec un Score_Timi élevé et une proportion de la modalité « Non » très basse, pour l'hypertension artérielle (Nœud n°3) et avec en plus, une proportion de la modalité « Oui » très basse, pour le tabagisme (Nœud 5). Les proportions de décès sont d'autant plus importantes que le Score_TIMI est élevé (Nœud 8). Quand on utilise les même variables explicatives sur les unités statistiques « patients », seule la variable Score_TIMI apparait à tous les niveaux de l'arbre, ce qui nous montre l'influence importante de la trajectoire sur la survie du patient.

5.2 Méthode de représentation graphique (SOE-VSTAR)

La méthode SOE-VSTAR pour Symbolic Objects Editor de M. Noirhomme et Al. Bock et Diday (2000), est une méthode de représentation graphique par étoiles pour représenter : les données multivariées, les variables qualitatives et quantitatives (quantitatives : données intervalles, qualitatives : valeurs pondérées ou multivariées), munies éventuellement des taxonomies hiérarchiques et de dépendances logiques. Elle permet d'éditer, et donc d'observer les objets symboliques – les concepts – à partir d'une interface graphique, notamment par un affichage en étoile en deux et même trois dimensions. Cette représentation est fort utile dans la mesure où elle permet de visualiser en une fois tout le « profil » du concept, par rapport à toutes les variables sélectionnées, ou même, à la guise de l'utilisateur, de sélectionner juste une partie des variables décrivant l'objet symbolique en question. Il est possible en outre de pouvoir réaliser des modifications sur l'affichage des variables observées, en terme de changement de libellé, du nom de la variable, angle de vue de l'étoile, passage 2D/3D. En deux dimensions (voir fig.7), les variables continues apparaissent comme des intervalles alors que les variables présentant des modalités sont des points dont la grosseur dépend du pourcentage de distribution du mode en question. En trois dimensions (voir fig. 7), les variables continues apparaissent toujours comme des intervalles, alors que celles présentant des modalités deviennent des histogrammes 3D, dont la hauteur correspond au pourcentage de la distribution du mode. Il est important de noter que l'affichage en étoile apporte une information judicieuse vis-à-vis de l'échelle de chaque branche : en

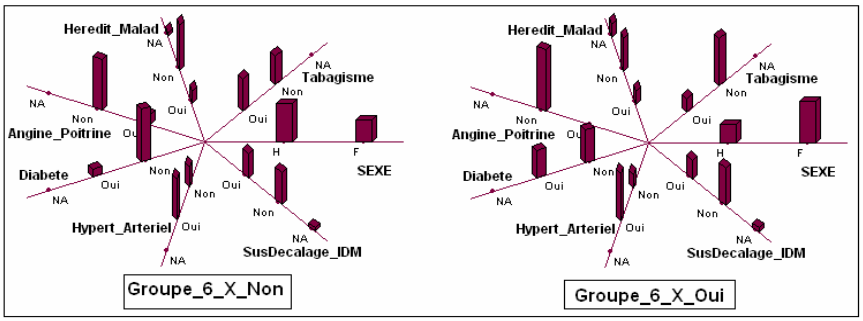


FIG. 15 – *Représentation des sous-classes Deces et Non_ Deces de la classe6*

effet, la mesure choisie par SOE_VSTAR dépend du maximum possible de la variable en question, sur l'ensemble de tous les concepts. En d'autres termes, la comparaison de deux concepts selon les mêmes variables nous permet de jauger les écarts qui contribuent à qualifier un profil particulier. Grâce à cette méthode, il est possible d'éditer rapidement les objets symboliques de l'étude, comme : les trajectoires elles-mêmes, et leurs caractéristiques en terme d'agrégation des particularités de différents patients et les différentes classes de trajectoires, en terme d'agrégation de caractéristiques symboliques de trajectoires. Ces étoiles nous permettent de comparer visuellement, en un seul coup d'œil, les sous-classes DECES et NON-DECES d'une même classe (ici, la classe 6) de trajectoire et permettent de voir quelles sont les variables qui distinguent la sous-classe des DECES de la sous-classe des NON-DECES. Nous comparons également avec les fréquences observées sur la population totale, ce qui permet de voir la significativité de ces variables. Description de la classe 6 : Mutation : Mono-établissement_Multi-unité ; UF1=non cardio et UF2=cardio ; 43 Patients ; 32 trajectoires. On remarque une proportion très élevée de femmes dans la sous-classe des décès alors qu'elles ne forment que 30% de la population totale (DC 70% / NDC 37% / TOT 30%). Il s'agit donc d'une classe de trajectoires plus mortelles pour les femmes. Par contre le tabagisme (NDC 53% / DC 23% / TOT 63%) et le sus-décalage IDM (NDC 47% / DC 38% / TOT 66%) sont beaucoup plus forts dans la sous-classe des non-décès mais nettement moins élevé que dans la population totale ; Ceci s'explique certainement par le fait que la sous-classe des non-décès a une forte proportion d'hommes, ceux-ci ayant une forte consommation de tabac. Par contre, le diabète (DC 46% / NDC 13% / TOT 25%) et l'hypertension artérielle (DC 77% / NDC 66% / TOT 50%) sont beaucoup plus forts dans la sous-classe des décès et nettement plus élevés que leur répartition sur la population totale.

6 Conclusion

Les résultats fournis par les formes fortes puis les motifs fréquents ont permis de donner une base de réflexion aux experts médecins afin d'élaborer des classes de trajec-

toire « expertes ». Ces trajectoires décrites sous forme de données symboliques grâce au module DB2SO Hébrail et al (2000), dans Bock et Diday(2000) ont ensuite permis de voir qu'elles avaient un impact sur la survie d'une part, par l'apparition des classes expertes médecin qui discriminent en fonction de la nature CARDIO ou NON-CARDIO du premier service (UF) fréquenté et d'autre part, par comparaison de l'arbre de décision obtenu selon que les unités statistiques sont les patients ou les trajectoires puisque les variables explicatives les plus explicatives de la survie diffèrent.

Bibliographie

Afonso.F (2004), Extension de l'algorithme Apriori et des règles d'association au cas des données symboliques diagrammes et sélection des meilleures règles par la régression linéaire symbolique, *Revue RNTI, Classification et Fouille de données*, D.A Zighed., G. Venturini, Eds, 2004, Cepadues, p. 1 – 12.

Agrawal.R, Srikan.R (1995), Mining Sequential Pattern, *11th Int'l Conference on Data Engineering (DE'95),* Taipei, Taiwan, 1995.

Bock H.H, Diday.E (2000), *Analysis of Symbolic Data : Exploratory methods for extracting statistical information from complex data*, Springer Verlag, Heidelberg, 2000.

Dart.T, Cui.Y, Chatellier.G, Degoulet.P (2003), Analysis of hospitalised patient flows using data-mining, 2003, *Santé Publique et Informatique Médicale (SPIM)*, Université Paris VI

Diday.E, (1992) Analyse des données et classification automatique numérique et symbolique. *EUSTAT*. Zarautz, 1992.

Diday.E (1971) La méthode des données dynamiques. *Revue Statistique Appliquée, Vol 19-2.* 1971, p. 19 – 34.

Leduff.F, Muntean.C, Cuggia.M, Mabo.P (2004), *Predicting survival Causes after out of Hospital Cardiac Arrest Using Data Mining Methods*, Medical Informatics Laboratory, Rennes, 2004

Limam. M (2005), *Méthode de description de classes combinant classification et discrimination*. Thèse de Doctorat. Université Paris Dauphine, 2005.

Xing.D, Shen.J (2004), Efficient data mining for web navigation patterns. *Information and Software technology, n° 46 :* 2004, p. 55 – 63.

Séries Temporelles : Vers une mesure de distance optimale

Rémi Gaudin, Nicolas Nicoloyannis

LABORATOIRE ERIC 3038 Université Lumière - Lyon2
5 av. Pierre Mendès-France 69676 BRON cedex FRANCE
Remi.Gaudin@univ-lyon2.fr, Nicolas.Nicoloyannis@univ-lyon2.fr

Résumé. De très nombreuses méthodes d'apprentissage et de classification appliquées aux séries temporelles nécessitent de définir une mesure de distance adaptée à ce type de données. En dehors de la distance Euclidienne, plusieurs autres mesures de distance ont été mises au point, la plus populaire étant le *Dynamic Time Warping*. Nous proposons dans cet article une nouvelle distance, l'*Optimal Time Warping*, qui englobe et généralise toutes les mesures de distances précédentes. La définition de cette distance permet l'application d'un algorithme génétique qui l'adaptera de manière optimale, en fonction du problème de classification que l'on a à résoudre. Le processus d'apprentissage que nous proposons est donc basé directement sur la mesure de distance, et non sur les séries elles-mêmes, ce qui constitue toute l'originalité de ce travail.

1 Introduction

Dans le domaine de l'apprentissage – ou classification – automatique, il est souvent nécessaire de définir une mesure de distance entre les différents individus du jeu de données. Lorsque le jeu de données comprend des individus classiques (i.e. non temporelle : chaque individu possède *m* variables *non ordonnées* entre elles), la mesure de distance de pose en général pas de problème, la distance Euclidienne étant le plus souvent utilisée. Par contre, si le jeu de donnée est composé de séries temporelles (i.e. chaque individu possède une variable qui évolue *m* fois dans le temps, i.e. *m* variables *ordonnées*), la distance Euclidienne devient alors trop rudimentaire pour effectuer une classification pertinente dans la plupart des cas.

De nouvelles mesures de distances ont apparues, ce sont les distances de type *Time Warping* (Faloutsos et al., 1997). Contrairement à la distance Euclidienne, elles se permettent de comparer un point d'une série avec n points d'une autre série, ceux-ci n'intervenant pas nécessairement au même instant *t*. La plus connue de ces distances est le *Dynamic Time Warping*. Malgré le degré de raffinement élevé que présentent ces mesures de distance, aucune d'entre elles ne peut être considérée comme optimale pour tous les problèmes de classification de séries temporelles. Il est souvent indispensable de les adapter de manière *ad hoc*, ce qui nécessite des connaissances *a priori* du domaine et fait perdre toute l'utilité des algorithmes de classification ou d'apprentissage automatique.

Nous avançons ici l'idée d'une mesure de distance plus générale, qui engloberai toutes les distances de type *Time Warping* possibles, et qui serait optimisable à l'aide d'un processus d'apprentissage. Ce processus d'apprentissage sera applicable à tout problème de

classification de séries temporelles et permettra ainsi d'adapter automatiquement la mesure de distance en fonction du problème à résoudre. Nous proposons donc une nouvelle mesure de distance, appelée *Optimal Time Warping* (*OTW*), ainsi qu'un algorithme d'apprentissage simple permettant d'optimiser cette distance. Ce travail étant en cours de réalisation, nous n'exposerons ici que les algorithmes, ainsi que les difficultés qu'il nous reste à résoudre.

Nous allons tout d'abord présenter les mesures de distances existantes, en particulier les distances de type *Time Warping* (section 2). Nous présentons ensuite la définition du *OTW* dans la section 3. Dans la section 4, nous expliquons pourquoi cette nouvelle mesure de distance est nécessaire pour résoudre de nombreux problèmes de classification puis dans la section 5 nous présentons l'algorithme d'apprentissage qui lui est associé, celui-ci s'inspirant du mécanisme des Algorithmes Génétiques. Enfin nous concluons dans la section 6 en exposant les principaux problèmes qu'il nous reste à résoudre avant la finalisation de ce travail.

2 Les principales mesures de distance

2.1 Distance Euclidienne

Cette mesure de distance est couramment utilisée car elle a le mérite d'être simple à mettre en œuvre. Son principe est de comparer chaque point de la première série avec un (et un seul) point de la seconde : celui qui intervient au même instant t (figure 1). On définit donc la distance $Eucl(Q,C)$ entre les séries $Q = q_1, q_2, \ldots, q_m$ et $C = c_1, c_2, \ldots, c_m$ de la manière suivante :

$$Eucl(Q,C) = \sum_{i=1}^{m} |q_i - c_i| \qquad (1)$$

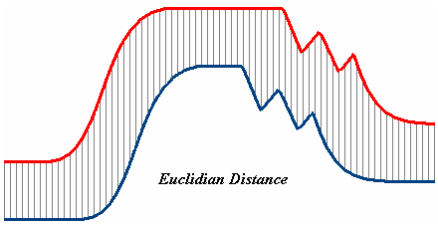


FIG. 1 – *Comparaison de deux séries temporelles à l'aide de la distance Euclidienne.*

L'inconvénient de cette distance est son incapacité à gérer les décalages temporels d'une série à l'autre. Deux séries ayant exactement les mêmes formes peuvent avoir une distance Euclidienne très élevée si elles n'interviennent pas au même instant *t*. Ce résultat peu intuitif peut considérablement altérer la qualité de la classification (Keogh et Pazzani, 2000).

2.2 Dynamic Time Warping (*DTW*)

La mesure de distance Dynamic Time Warping (*DTW*) a la capacité de gérer les décalages temporels qui peuvent éventuellement exister entre deux séries (Berndt et Clifford, 1994, Sakoe et Chiba, 1978). Au lieu de comparer chaque point d'une série avec celui de l'autre série qui intervient au même instant *t*, on permet à la mesure de comparer chaque point d'une série avec un ou plusieurs points de l'autre série, ceux-ci pouvant être décalés dans le temps (figure 2).

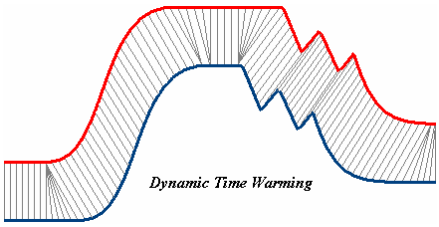


FIG. 2 – *Comparaison de deux séries temporelles à l'aide du Dynamic Time Warping.*

Le *DTW* possède une définition récursive qui calcule la similarité entre les séries $Q = q_1, q_2, \ldots, q_m$ et $C = c_1, c_2, \ldots, c_n$ de la manière suivante :

$$\begin{aligned} &\gamma(q_i, c_j) = \left|q_i - c_j\right| + \min\left\{\gamma(q_{i-1}, c_j), \gamma(q_i, c_{j-1}), \gamma(q_{i-1}, c_{j-1})\right\} \\ &DTW(Q, C) = \gamma(q_m, c_n) \end{aligned} \tag{2}$$

On peut aussi paramétrer l'écart temporel maximum permis à la mesure pour comparer deux points. On définit ainsi une fenêtre temporelle (appelée δ) que l'on fera "glisser" sur chacune des séries à comparer. Par exemple, si on fixe la taille de la fenêtre δ = 3, chaque point d'une série qui intervient à un instant t ne pourra être comparé qu'avec les points de l'autre série qui interviennent aux instants $t \in [t-3, t+3]$. Le choix de la taille de cette fenêtre peut avoir une influence sur le résultat et il convient de la fixer avec soin, une fenêtre temporelle trop large ou trop fine pouvant nuire à la qualité de la classification (Ratanamahatana et Keogh, 2004a).

2.3 Distances de type *Time Warping*

Le *DTW* n'est en fait qu'une mesure parmi toutes les mesures permettant la comparaison de points n'intervenant pas au même instant *t*. Toutes ces distances ont un objectif commun : il s'agit de trouver la meilleur série de comparaison entre les points de deux séries, appelée *best warping path*. Un *warping path* est une séquence de couples d'indices de la forme :

$$W = w_1, w_2, \ldots, w_T \quad avec\ w_t = (i, j)_t$$

i et *j* sont des indices compris entre 1 et la taille de chacune des deux séries *Q* et *C*. *T* est égal au nombre de comparaisons total que l'on effectue entre les deux séries. Par exemple, si *Q* et *C* sont de la même taille *m* et que l'on applique la distance Euclidienne, le *warping path* correspondant est égal à $W = (1, 1), (2, 2),\ldots, (m\text{-}1, m\text{-}1), (m, m)$, i.e. $i = j$ pour tout w_t. Si on y applique une mesure de distance de type *Time Warping*, alors on peut avoir $i \neq j$. En règle générale, on considère qu'un *warping path* raisonnable doit respecter une contrainte de monotonie croissante, i.e. $i_{t+1} \geq i_t$. Par contre, il n'est pas nécessairement continu, un point d'une série peut être totalement ignoré lors du calcul de la distance (Faloutsos et al., 1997, Keogh et Pazzani, 2001).

En plus de la contrainte sur la monotonie, une autre contrainte courante est celle de la fenêtre temporelle δ qui interdit de comparer deux points trop éloignées dans le temps, cette fenêtre peut avoir une taille fixe ou variable au cours du temps (Ratanamahatana et Keogh, 2004a).

Toutes les distances de type *Time Warping* peuvent être définies sous forme d'algorithme récursif. Mais la complexité des algorithmes récursifs étant trop élevé, il est nécessaire d'avoir recours à une méthode d'implémentation appelée *programmation dynamique* (Bellman, 1957, Berndt et Clifford, 1996). Cette méthode consiste à créer une matrice de taille $m \times n$. A l'intérieur de chaque cellule (i, j) de la matrice, nous allons stocker la valeur de la distance entre le $i^{ème}$ point de la série Q et le $j^{ème}$ point de la série C. Une fois la matrice remplie, nous pouvons ensuite chercher le meilleur *warping path* qui part de la cellule (1, 1) et qui arrive à la cellule (m, n). La figure 3 présente un exemple de *warping path* entre deux séries temporelles de taille $m = n = 8$.

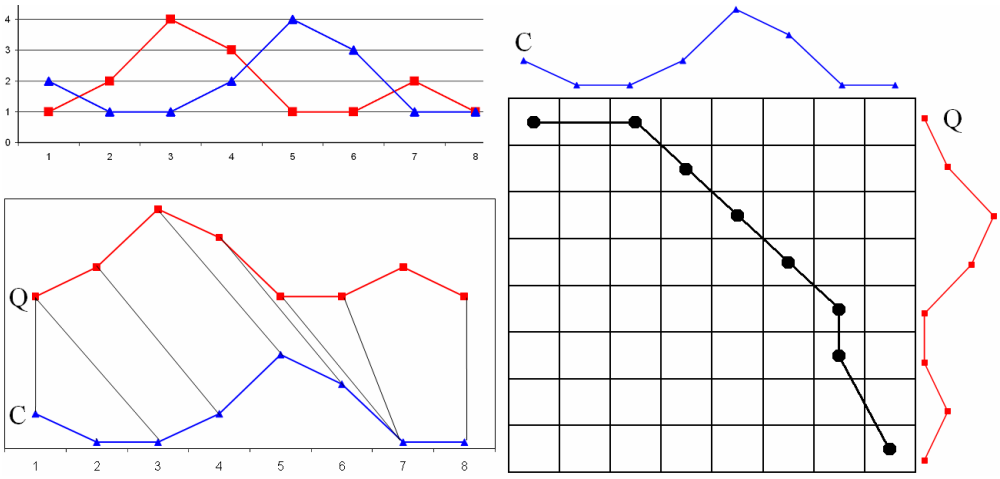


FIG. 3 – *Exemple de calcul de la distance entre deux séries Q et C de tailles m = n = 8. Le warping path trouvé ici est égal à W = (1,1), (1,3), (2,4), (3,5), (4,6), (5,7), (6,7), (8,8).*

Nous avons constaté que toutes les distances et variantes de distances de type *Time Warping* peuvent être définies à l'aide d'une seule définition que nous avons appelé *Optimal Time Warping* (*OTW*) et que nous présentons dans la section suivante.

3 Optimal Time Warping (*OTW*)

Nous allons présenter ici une nouvelle définition de mesure de distance entre deux séries temporelles. Cette définition est construite de telle manière à ce que les distances Euclidiennes, *DTW* et, en règle général, toute distance de type *Time Warping* ne soient que des cas particuliers de celle-ci. L'équation 3 est la définition de cette distance :

$$\begin{aligned} &\gamma(q_i, c_j, \pi_{ij}) = \left|q_i - c_j\right| \times \pi_{ij} + \min\left\{\gamma(q_{i-1}, c_j, \pi_{(i-1)j}), \gamma(q_i, c_{j-1}, \pi_{i(j-1)}), \gamma(q_{i-1}, c_{j-1}, \pi_{(i-1)(j-1)})\right\} \\ &OTW(Q, C, \Pi) = \gamma(q_m, c_n, \pi_{mn}) \end{aligned} \quad (3)$$

Π est une matrice de taille $m \times n$. π_{ij} correspond au poids que l'on applique lors de la comparaison entre les points q_i et c_j. C'est un réel. Si l'on souhaite comparer notre distance généralisée à la distance Euclidienne, il est facile de démontrer que :

$$\begin{aligned} &si\ \forall \pi_{ij} \in \Pi_{Eucl},\ \pi_{ij} = \begin{cases} 1 & si\ \ i = j \\ \infty & sinon \end{cases} \\ &alors\ OTW(Q, C, \Pi_{Eucl}) = Eucl(Q, C) \end{aligned} \quad (4)$$

De la même manière, dans le cas du *DTW* avec une fenêtre temporelle δ, on démontre que :

$$\begin{aligned} &si\ \forall \pi_{ij} \in \Pi_{DTW},\ \pi_{ij} = \begin{cases} 1 & si\ \ \left|i - j\right| \le \delta \\ \infty & sinon \end{cases} \\ &alors\ OTW(Q, C, \Pi_{DTW}) = DTW(Q, C) \end{aligned} \quad (5)$$

Plus généralement, pour toute distance *D* de type *Time Warping,* il existe une matrice Π_D correspondante permettant d'obtenir l'égalité $OTW(Q, C, \Pi_D) = D(Q,C)$. Ainsi pour tout problème de classification donné, si on possède la matrice Π optimale correspondant à ce problème, on a la certitude que *OTW* obtiendra des résultats au moins aussi bons, sinon meilleurs, que toutes les distances de type *Time Warping*.

4 A quoi sert l'*OTW* ?

Afin de bien expliciter l'utilité de l'*OTW*, nous allons illustrer notre propos à l'aide d'un exemple inspiré du jeu de donnée « GunX » (Ratanamahatana et E. Keogh, 2004a). Ce jeu de donnée est composé de séries temporelles issues de vidéos montrant des acteurs en train de dégainer une arme (pour simplifier, seules les séries représentant le mouvement sur l'axe en abscisse – X – sont conservées). Soit ils tiennent réellement une arme dans leur main (classe A), soit ils ne possèdent pas d'armes et la simulent avec leur main (classe B). A l'intérieur de chacune de ces deux classes, les acteurs peuvent être soit des femmes (sous classes A_1 et B_1), soit des hommes (sous classes A_2 et B_2). La figure 4 présente un exemple pour chacune des

quatre classes (ces exemples ont été simplifiés et lissés pour mieux faire comprendre le raisonnement sous-jacent).

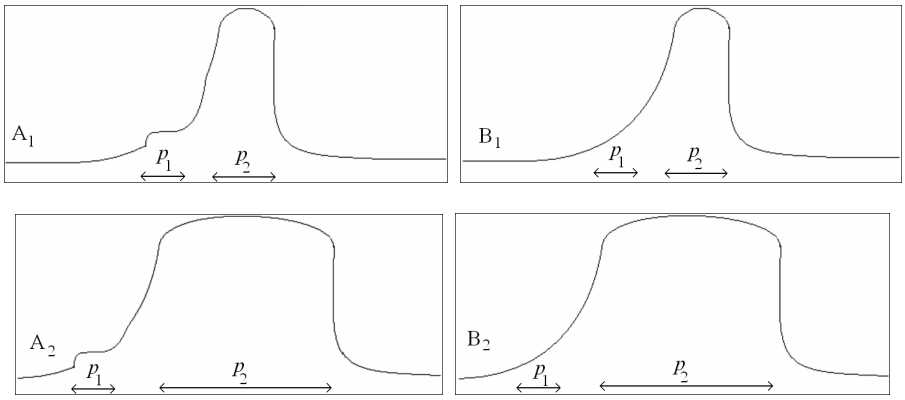


FIG. 4 – *Ces quatre archétypes représentant chacune des quatre classes du jeu GunX. Les zones p_1 et p_2 sont discriminantes. Elles seules permettent de décider à quelle classe appartient une série donnée.*

Dans cet exemple, la zone p_1 discrimine la classe A de la classe B, et la taille de la zone p_2 discrimine les sous-classes A_1 et B_1 des classes A_2 et B_2. On suppose que l'on ne connaît pas ces deux zones à priori (c'est au processus d'apprentissage de les découvrir). Si on utilise la méthode des k plus proche voisins avec pour mesure de distance le *DTW* et que l'on souhaite dissocier les quatre classes entre elles, deux cas de figures apparaissent :

1. Si on fixe un δ trop faible, le *DTW* a tendance à mélanger A_1 avec B_1, et A_2 avec B_2 (principalement à cause des nombreuses variations ou bruits présents dans les zones non discriminantes).
2. Si on fixe un δ trop élevé, le *DTW* a tendance à mélanger A_1 avec A_2, et B_1 avec B_2 (car une fenêtre temporelle trop élevée ne considère plus de différence entre les deux zones p_2 de différentes tailles).

Cet exemple illustre bien les deux principales faiblesses des mesures de distance de type *Time Warping* qui sont l'impossibilité de fixer des pondérations plus ou moins élevés sur les différentes zones d'une série temporelle (les zones les plus discriminantes – dans notre exemple la zone p_1 – auront un poids plus élevé), et l'incapacité de gérer les décalages temporelles de manière progressive et non binaire (par exemple on pourrait considérer que deux zones p_2 de tailles différentes peuvent être entièrement comparées entre elles mais que plus l'écart de taille est important plus la distance est élevée, en y associant une pondération qui croit selon une loi gaussienne). La distance *OTW* peut intégrer ces caractéristiques si elles sont nécessaires à une bonne classification. Il suffit par exemple d'imposer à la matrice Π les contraintes adéquats. L'équation 6 est un exemple de contrainte permettant de mettre en valeur une zone discriminante p_1 et l'équation 7 est un exemple de gestion de décalages temporelles dans la zone p_2.

$$\forall \pi_{ij} \in \Pi, \pi_{ij} = \begin{cases} a_{ij} & si\ i\ et\ j \in p_1 \\ b_{ij} & sinon \end{cases} \quad avec\ a_{ij} > b_{kl}\ \forall i,j,k,l \tag{6}$$

$$\forall i,j \in p_2, \pi_{i(j+1)} = \pi_{(i+1)j} = a_{ij} \times \pi_{ij}\ \ avec\ a_{ij} > 1 \tag{7}$$

5 Etablir la distance optimale pour un problème donné

A tout problème de classification donné, il existe une matrice Π optimale, i.e. la matrice qui permet d'obtenir le meilleur taux de classification pour le problème en question. Nous avons mis au point un algorithme d'apprentissage qui permettra de découvrir la matrice optimale. C'est un algorithme d'optimisation de type Algorithme Génétique (Spears et al., 1993, Kwong et al., 1996), dont la fonction d'évaluation serait le taux de bonne classification obtenu à l'aide de la distance *OTW* et de la méthode des *k* plus proches voisins :

```
procedure OptimiserΠ;
{
   t = 0 ;
   initialiser_population P(t) ;
   evaluer P(t)
   tant que ( !ConditionFin)
   {
      t = t + 1 ;
      selection_parents P(t) ;
      reconbiner_parents P(t) ;
      muter_enfants P(t) ;
      evaluer P(t) ;
      survivre P(t) ;
   }
}
```

```
fonction evaluer P(t)
{
   pour tout Π_j de P(t) faire
   {
      soit X = {x_1, …,x_n} le jeu à classer
      nbBienclassé(Π_j) = 0 ;
      pour i = 1 à n faire
      {
         si kpp(x_i , OTW, Π_j) → bonne classification
            alors nbBienclassé(Π_j) = nbBienclassé(Π_j) + 1 ;
      }
      nbBienclassé(Π_j) = nbBienclassé(Π_j) / n ;
   }
}
```

P est la population d'individus composée de q matrice candidates Π_j. Chaque matrice Π_j est un individu possédant m^2 variables (chaque cellule de la matrice est une variable, m est la taille de la plus longue série du jeu de donnée). On cherche à optimiser ses variables afin d'obtenir la meilleur classification possible.

La fonction *initialiser_population* initialise les q matrices Π_j, par exemple en affectant une valeur aléatoire à chacune des m^2 cellules. Ensuite la fonction *evaluer* calcule, pour chacune des q matrices, le taux de bonne classification que l'on obtient grâce à elle. Pour cela il lui suffit de classer l'ensemble des séries du jeu de donnée à l'aide de la méthode des k plus proches voisins et avec pour mesure de distance l'*OTW* associé à la matrice Π_j. La fonction *selection_parents* choisit ensuite les q' individus Π_j qui ont obtenus la meilleure évaluation ($q' << q$), puis on les recombine entre elles afin de générer q'' matrices enfants (par exemple en calculant $\pi_{ij}(\text{enfant}) = (\pi_{ij}(\text{parent}_1) + \pi_{ij}(\text{parent}_2)) / 2$). Afin d'éviter que l'algorithme ne stagne à cause d'un mauvais optimum local, la fonction *muter_enfant* s'occupe de modifier aléatoirement un certain nombre de variables dans une certain nombre d'individus Π_j. Enfin, la fonction *survivre* choisit parmi l'ensemble des parents et enfants les q meilleurs individus par le biais de la fonction *evaluer*. On répète ce processus jusqu'à l'obtention d'une matrice Π_j satisfaisante, i.e. qui permet d'obtenir une bonne classification.

Le surapprentissage peut être naturellement évité en appliquant des méthodes de validation croisée (on divise le jeu de donnée en n sous jeux : n-1 sous jeux d'apprentissage que l'on cherche à bien classer et un sous jeux de test qui permet à la fonction *evaluer* de calculer le taux de bonne classification. On cherche à maximiser la moyenne de tous les taux obtenus pour toutes les combinaisons de sous jeux possible).

Le nombre de variable à optimiser est égal à m^2, ce qui peut générer des temps de calcul très élevé dans le cas de séries temporelles longues. On peut diminuer ce nombre en fixant arbitrairement une fenêtre temporelle δ maximum : Le nombre de variables à optimiser devient alors inférieur à $m \times (2\delta+1)$. Il est démontré empiriquement que dans la très grande majorité des problèmes de classification de séries temporelles, il n'est pas nécessaire que δ dépasse 10 % de la valeur de m (Ratanamahatana et E. Keogh, 2004b).

6 Conclusions

Nous proposons dans cet article une mesure de distance entre séries temporelles, l'*Optimal Time Warping*, qui généralise toutes les distances de type *Time Warping*. Cette distance a le mérite de pouvoir être adaptée et optimisée en fonction du problème de classification que l'on a à résoudre. Nous proposons pour cela une méthode d'apprentissage et d'optimisation basée sur un Algorithme Génétique. Les points qu'il nous reste à résoudre avant la finalisation définitive de ce travail sont :

- Définir précisément le fonctionnement des procédures de sélection, de recombinaison, de mutation et de survie dans le cas des matrices Π.
- Evaluer la complexité réelle de l'algorithme totale, et si besoin mettre en place des heuristiques permettant de réduire cette complexité.
- Tester notre démarche à grande échelle, sur de nombreux jeux de données.

Références

R. Bellman (1957). *Dynamic Programming*, Princeton University Press, New Jersey.

D.J. Berndt et J. Clifford (1994). Using Dynamic Time Warping to Find Patterns in Time Series. *In Proceedings of the KDD Workshop, pages 359--370, Seattle, WA*.

D.J. Berndt et J. Clifford (1996). Finding Patterns in Time Series: A Dynamic Programming Approach. *Advances in Knowledge Discovery and Data Mining*, pp 229-248.

C. Faloutsos, H. Jagadish, A. Mendelzon, et T. Milo (1997). A signature technique for similarity based queries. *Proc. International Conference on Compression and Complexity of Sequences,* PositanoSalerno, Italy.

E. Keogh et M. Pazzani (2000). Scaling Up Dynamic Time Warping for Data Mining Applications. *In proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Aug. 20-23. Boston, MA, USA. pp. 285-289.

E. Keogh et M. Pazzani (2001). Derivative dynamic time warping. *In First SIAM International Conference on Data Mining (SDM'2001),* Chicago, USA.

S. Kwong, Q. He, et K. Man (1996). Genetic time warping for isolated word recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(7) :849–865.

C.A. Ratanamahatana et E. Keogh (2004a). Making Time-series Classification More Accurate Using Learned Constraints, *In proceedings of SIAM International Conference on Data Mining (SDM '04)*, Lake Buena Vista, Florida, pp. 11-22.

C. A. Ratanamahatana and E. Keogh (2004b). Everything you know about dynamic time warping is wrong. *In Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2004)* - Seattle, WA.

H. Sakoe et S. Chiba (1978). Dynamic programming algorithm optimisation for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-26(1):43-49*.

W.M. Spears, K.A. De Jong, T. Bäck, D.B. Fogel, et H. de Garis (1993). An overview of evolutionary computation. *In Proceedings of the 1993 European Conference on Machine Learning*.

Summary

Numerous classification algorithms for time series need a suitable distance measure. In addition to Euclidean distance, several other distance measures exist and the most famous is *Dynamic Time Warping*. In this paper we propose a new distance measure, called *Optimal Time Warping*, which generalizes all previous time series distances. The definition of this distance allows applying a Genetic Algorithm that adapts it in an optimal way, according to the current classification problem we have to resolve. The originality of this work is that we propose a learning process directly based on the distance measure, and not on time series themselves.

Représentations symboliques adaptatives de séries temporelles : principes et algorithmes de construction

Bernard Hugueney*

* Université PARIS-DAUPHINE
LAMSADE
Place du Maréchal de Lattre de Tassigny
75775 PARIS CEDEX 16
bernard.hugueney@lamsade.dauphine.fr,
http://www.lamsade.dauphine.fr/~hugueney

Résumé. Les séries temporelles constituent un domaine très important de la fouille de données. En effet, les très gros volumes de données numériques généralement entreposés ne se prêtent pas à une analyse directe. Dans un but à la fois de réduction de la dimensionnalité et d'extraction d'information, la fouille de données de séries temporelles donne généralement lieu à un changement de représentation des séries temporelles. Dans un objectif d'intelligibilité de l'information extraite lors du changement de représentation, on peut avoir recours à des représentations symboliques de séries temporelles. Nous proposons un cadre général de représentation de séries temporelles, ainsi que deux représentations particulières (Clustering-Based Symbolic Representations: CBSR et Segmentation-Based Symbolic Representation with Linear models of 0th order : SBSR-L0) s'inscrivant dans ce cadre général.

1 Introduction

Les séries temporelles constituent un domaine très actif de la fouille de données. En effet, les bases de données de séries temporelles sont caractérisées non seulement par leur très grand volume, mais aussi par le fait que les informations intéressantes (tendances, corrélations,...) ne sont pas directement accessibles à partir des données brutes. Pour cette raison, des changements de représentation sont alors mis en œuvre. Nous nous intéressons particulièrement à des représentations symboliques, plutôt que numériques, car elles peuvent être interprétées par les utilisateurs. SAX (Symbolic Aggregate approXimation) est une représentation symbolique de séries temporelles classiquement utilisée. Nous présentons un cadre général permettant de formuler une très large gamme de représentations symbolique. Parmi celles-ci, nous en proposons deux qui peuvent être considérées comme des extensions adaptatives de SAX : CBSR (Clustering-Based Symbolic Representations) et SBSR-L0 (Segmentation-Based Symbolic Representation with Linear models of 0th order). Pour chacune de ces représentations symboliques, nous proposons un algorithme de construction. En conclusion, nous suggérons d'autre représentations symboliques à partir du même cadre général, ainsi que diverse applications possibles de ces représentations symboliques.

Il n'est plus rare d'avoir des bases de données constituées de centaines de séries temporelles, elles-mêmes constituées de dizaines de milliers de points. En effet, les séries temporelles sont parmi les données qu'il est le plus facile d'accumuler en très grande quantité car il suffit de disposer de capteurs numériques pour remplir les bases de données, à une vitesse qui n'est limitée que par la fréquence d'échantillonnage des capteurs. On peut donc très facilement se retrouver dans le cas typique de la fouille de données, à savoir que des masses de données ont été recueillies à un niveau de détail sans rapport avec les utilisations qui en seront faites. La figure 1 montre par exemple un extrait de série temporelle issu de la base que nous allons utiliser pour valider notre approche. On y voit un extrait d'une série temporelle sur une période d'une semaine alors que la base contient 420 séries temporelles enregistrées pendant un an.

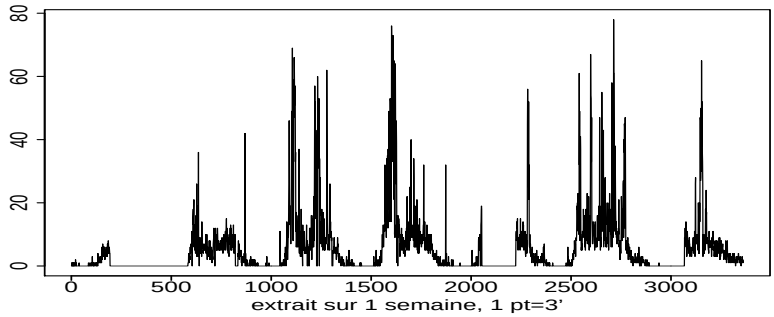


FIG. 1 – *Extrait d'une série temporelle de la base.*

Les changements de représentation peuvent être effectués pour deux raisons distinctes. Ces raisons ne sont pas non exclusives l'une de l'autre mais induisent des variations sur les critères d'évaluation des représentations.

La motivation première est souvent la simple réduction de dimensionnalité. Le volume des données en jeu dans l'exemple vu plus haut ne permet pas de manipuler directement celles-ci, soit parce qu'elles ne tiennent pas en mémoire centrale, soit que le temps de calcul de n'importe quel algorithme non trivial est prohibitif. On cherche alors à construire des représentations préservant au maximum l'information présente dans les données, mais sans avoir de connaissances sur ce qui constitue justement cette information. On en est alors réduit à utiliser des représentations qui permettent de reconstruire des séries temporelles et à chercher les paramètres de ces représentations de façon à minimiser l'erreur de modélisation, par exemple au sens des moindres carrés. Par exemple, la représentation APCA (Adaptive Piecewise Constant Approximation) présentée par Keogh et al. (2001a) permet une plus faible erreur quadratique de modélisation (à complexité de représentation égale) que la représentation PAA (Piecewise Aggregate Approximation Keogh et al. (2001b) , Yi et Faloutsos (2000)). Elle permet aussi une indexation basée sur une distance L^p plus efficace que cette dernière.

La finalité de l'analyse de données étant la présentation d'informations aux utilisateurs, un autre objectif des changements de représentation peut être la construction des éléments d'information de plus haut niveau qui seront pertinents. L'évaluation de telles représentations devra alors prendre en compte une sémantique adaptée à la problématique du domaine.

2 Représentations symboliques de séries temporelles

Les utilisateurs finaux d'outils d'analyse de données réfléchissent à propos de celles-ci en des termes symboliques. Les outils d'analyse de données ayant intérêt à mettre en œuvre un "univers du discours" associé à la sémantique du domaine, cela amène naturellement à considérer des représentations symboliques des données à analyser et non simplement des "résultats numériques". Diverses représentations symboliques de séries temporelles ont déjà été proposées. Nous présentons un cadre général permettant d'unifier celles-ci ainsi que deux représentations symboliques particulières s'incrivant dans ce cadre général : l'une (CBSR) basée sur la classification d'extraits de séries temporelles, l'autre (SBSR-L0) basée sur la segmentation par modèles linéaires d'ordre 0.

2.1 Représentation symbolique existante : Symbolic Aggregate approXimation (SAX)

Dans un objectif de réduction de la dimensionnalité, il est important de définir des unités temporelles permettant de regrouper les points des séries temporelles. On définit généralement des épisodes comme des intervalles du domaine de définition temporel des séries temporelles à représenter. Très généralement, en raison du coût minime d'acquisition et de stockage des données, les séries temporelles sont enregistrées dans les bases de données sous la forme la plus détaillée possible, indépendamment de l'échelle de temps à laquelle se développent les comportements à identifier. On pourra alors regrouper les points en épisodes sans perdre d'information essentielle. C'est le principe de la représentation symbolique SAX, présentée par Lin et al. (2003).

SAX est une représentation symbolique de séries temporelles univariées centrées réduites qui n'est pas adaptative :

- le domaine temporel est divisé en épisodes de même taille
- les classes d'équivalence des valeurs prises par les séries temporelles sont fixées a priori en fonction du nombre de symboles à utiliser, de façon à obtenir un découpage en classes de même effectif sous réserve que la distribution centrée et réduite des valeurs soit normale.

La non adaptativité de la représentation et la référence à la distance euclidienne donne à SAX les avantages suivants :

- la construction des représentations est extrêmement efficace en temps de calcul ($O(N)$) pour une représentation d'une série temporelle de N points.
- toutes les représentations (basées sur un même nombre de symboles et des épisodes de même taille) sont trivialement commensurables.

En revanche, cette représentation souffre des inconvénients qui sont intrinsèquement liés :

- l'erreur de modélisation, pour une réduction donnée de la dimensionnalité, n'est pas minimale puisque le modèle n'est pas localement adapté aux données
- les classes d'équivalence auxquelles les symboles sont associés ne sont pas forcément pertinentes car elles ne sont pas adaptées aux données.

Si l'on dispose des ressources nécessaires, il peut donc être intéressant de construire des représentations symboliques adaptatives. De nombreuses représentations symboliques peuvent être envisagées, non seulement en fonction des données à analyser, mais aussi en fonction

des tâches d'analyse à effectuer. Nous présentons un cadre générique permettant d'unifer ces différentes représentations symboliques, avant de détailler deux types de représentations symboliques qui peuvent être considérés comme des extensions adaptatives des principes qui sous-tendent SAX.L'un est basé sur des classifications automatiques d'extraits de séries temporelles, l'autre sur des segmentations de séries temporelles,

2.2 Cadre général pour les représentations symboliques de séries temporelles

Soit une série temporelle ST définie par $\mathrm{ST} = \{(d_i, v_i)\}_{i\in\{1\cdots N\}}$, avec $d_i \in D$ et $v_i \in V$, avec D le domaine de définition temporel et V l'espace numérique des valeurs de la série temporelle, éventuellement muni de l'élément NA indiquant une valeur manquante.

Nous proposons de définir une représentation symbolique de ST par :

- un découpage en P épisodes $E = \{e_j = [d_{j_{\mathrm{debut}}}, d_{j_{\mathrm{fin}}}[\}_{j\in\{1\cdots P\}}$, avec$(d_{j_{\mathrm{debut}}}, d_{j_{\mathrm{fin}}}) \in D^2$ et $d_{j_{\mathrm{debut}}} < d_{j_{\mathrm{fin}}}$,
- un alphabet de K symboles $\Lambda = \{s_m\}_{m\in\{1\cdots K\}}$,
- la représentation symbolique proprement dite $\mathrm{RS} : E \to \Lambda, e_j \mapsto \mathrm{RS}(e_j)$ qui est alors l'application associant chacun des épisodes de E à un symbole de Λ.

Cette définition ne spécifie absolument pas la sémantique associée aux éléments de l'alphabet symbolique.Ceci lui permet d'être indépendante des domaines d'application spécifiques. En revanche, afin de permettre une évaluation des représentations symboliques, nous imposons qu'elles permettent de reconstruire une série temporelle numérique sur le domaine de définition temporel D de la série d'origine.

Une représentation symbolique sera pertinente si elle satisfait au mieux les impératifs contradictoires suivants :

- concision maximale de la représentation : avec des cardinalités de E et Λ minimales et des symboles aussi simples que possible
- fidélité maximale : permettant une reconstruction aussi proche que possible de la série d'origine, au sens d'une distance qui peut être liée au domaine d'application. En l'absence de distance propre au domaine, on pourra utiliser une distance point à point et minimiser par exemple la somme des erreurs quadratiques.

Bien sûr l'ensemble des représentations symboliques qu'il est possible de définir est infini et dépend principalement de la sémantique associée à l'alphabet symbolique. Pour une sémantique donnée, l'espace des représentations symboliques possibles d'une série temporelle de N points avec P épisodes associés à des symboles d'un alphabet de cardinalité K est $K^P C_N^P$. Il est donc évident qu'il sera hors de question d'énumérer les solutions possibles. Les représentations symboliques devront donc être choisies de façon à permettre leur construction de façon efficace. Les représentations symboliques que nous proposons ci-après atteignent cet objectif en se basant respectivement sur des algorithmes de classification et sur des algorithmes de segmentation de séries temporelles.

Dès lors que les représentations des séries temporelles d'une base de données sont adaptées aux données, se pose la question de décider si les paramètres (découpage en épisodes et interprétation des symboles de l'alphabet) doivent être adaptés spécifiquement pour chacune des séries temporelles. En effet, il est aussi possible de choisir une adaptation globale afin de déterminer un seul découpage et un seul ensemble d'interprétations de symboles communs à

l'ensemble des séries. Afin de faciliter la comparaison avec SAX, nous présentons ici les résultats correspondants à une adaptation globale pour l'ensemble des séries de la base. Néanmoins, nous reviendrons sur la question en 3.1.

2.3 Représentations symboliques adaptatives basées sur des classifications automatiques d'extraits de séries temporelles

2.3.1 Principes et définitions

SAX effectue des regroupements des points des séries temporelles représentées suivant des épisodes de même taille. Or les séries temporelles que nous étudions sont issues de la mesure de phénomènes en rapport avec des activités humaines et présentent des périodicités liées aux cycles d'activité (journalier et hebdomadaire). Un découpage pertinent en épisodes de même taille de telles séries pourrait tirer parti de ces périodicités.

SAX représente les épisodes par un symbole associé à un intervalle de valeurs. La représentation symbolique obtenue sera d'autant plus fidèle que les points d'un épisode donné appartiennent bien à un même intervalle de valeurs. Une telle interprétation ne permet donc pas de représenter correctement des épisodes d'une journée ou une semaine au cours desquels les séries temporelles ne sont pas constantes.

Pour cette raison, nous proposons d'associer à ces épisodes de même taille des formes que nous appellerons *formes prototypiques* car chacune sera associée à un ensemble d'extraits de séries temporelles. Ces extraits représentés par un même symbole sont donc conceptuellement réunis au sein d'une classe d'équivalence. Afin de déterminer automatiquement ces classes, nous utilisons un algorithme de classification automatique numérique non-supervisée (par exemple les k-moyennes). Comme dans le cas de la compression vectorielle, il est possible de reconstruire une série temporelle à l'aide des formes prototypiques. Ainsi que vu en 1, il est possible de chercher à minimiser soit l'erreur en reconstruction (dans une optique de réduction de la dimensionnalité), soit l'erreur de classification (dans une optique d'extraction d'information). Afin de permettre une évaluation des résultats sans avoir recours à une interprétation experte sur la pertinence des classes, nous privilégions ici le critère d'erreur en reconstruction.

2.3.2 Algorithme de construction

Pour construire une représentation CBSR, il suffit d'effectuer une classification numérique non-supervisée des extraits de séries temporelles après avoir déterminé les deux paramètres :

- d la durée des épisodes
- $o \in \{1, \cdots, d\}$, indice, que nous appelons *offset*, de début du premier épisode (qui ne correspond donc pas forcément à 1, indice du premier instant de la série temporelle à modéliser).

La durée des épisodes est généralement donnée par des connaissances a priori sur les séries temporelles à modéliser, par exemple une journée ou une semaine. Lorsque cela n'est pas le cas, il est possible d'utiliser la densité spectrale des séries temporelles afin d'estimer des périodicités intéressantes. Deux méthodes sont couramment utilisées à cet effet :

- la transformée de Fourier de la fonction d'auto-corrélation (théorème de Wiener-Kinchine) : cette méthode a une bonne résolution spectrale, mais le calcul de la fonction d'auto-corrélation est assez coûteux en temps de calcul.

– le périodogramme moyenné : il faut moyenner le périodogramme car celui-ci représente le spectre instantané. La précision de la localisation des pics de puissance qui nous intéressent est affectée par le lissage, mais le calcul du périodogramme par transformée de Fourier rapide (FFT) est efficace en temps de calcul.

Pour des raisons d'efficacités algorithmique, on utilise généralement le périodogramme moyenné.

Une fois la durée des épisodes déterminée, plusieurs approches (détaillées dans Hugueney (2003)) peuvent être utilisées pour déterminer un offset pertinent. Nous avons ici utilisé la plus simple, à savoir celle qui place les points de coupure entre épisodes aux endroits de la série temporelle où les variations locales sont les plus faibles, ceci de façon à essayer d'éviter de couper un comportement (lié à des variations de la série temporelle) entre deux épisodes.

2.3.3 Illustration des résultats

Le contexte de l'activité humaine à laquelle sont liées nos séries temporelles nous a amené à choisir des épisodes d'une journée. La figure 2 montre les formes prototypiques obtenues pour un alphabet de cinq symboles. Bien qu'il soit toujours possible d'approximer aussi fidèlement que nécessaire des séries temporelles quelconques à l'aide de modèles constants par morceaux, les profils observés montrent clairement qu'il faudrait alors un certain nombre d'épisodes pour modéliser correctement une journée. Au contraire, le fait que les séries temporelles de la base de données soient associées à des activités humaines, donc périodiques et répétitives, permet une réprésentation fidèle par CBSR avec un seul épisode par jour.

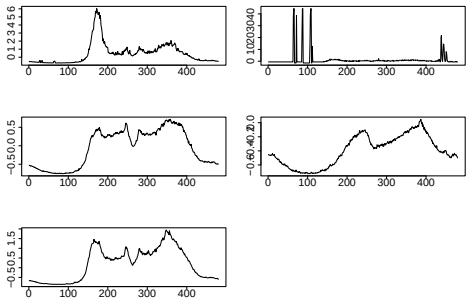


FIG. 2 – *Formes prototypiques associées à des épisodes d'une journée.*

2.4 Représentations symboliques adaptatives basées sur des segmentations linéaires d'ordre 0 (SBSR-L0)

2.4.1 Principes et définitions

Une représentation symbolique telle que définie en 2.2 est caractérisée par l'interprétation donnée à l'alphabet symbolique. Nous proposons une représentation symbolique de séries temporelles univariées dont les symboles correspondent à des niveaux constants. Les séries temporelles reconstruites à partir de telles représentations symboliques seront donc constantes par morceaux. Alors que dans le cas d'une segmentation classique par modèles linéaires d'ordre 0, chaque épisode est associé à un niveau qui est optimisé localement de façon à représenter au

mieux la série sur cet épisode, dans le cas d'une représentation SBSR-L0, chaque niveau est associé à un symbole de l'alphabet dont la cardinalité est $K \ll P$. Chaque niveau représente ainsi un ensemble d'épisodes et nous parlons donc de niveaux *prototypiques*. Pour simplifier, nous utiliserons l'association biunivoque entre les symboles et leur interprétation par un niveau prototypique pour parler d'un alphabet de niveaux prototypiques Λ plutôt que d'un ensemble de niveaux associés aux symboles de l'alphabet.

Par rapport à SAX, qui est la représentation symbolique la plus proche, les différences sont les suivantes :

- les symboles sont associés à des niveaux et non pas un intervalle. On associe un épisode au symbole correspondant au niveau le plus proche de la moyenne des valeurs prises par la série sur cet épisode plutôt qu'à un symbole correspondant à l'intervalle contenant cette valeur moyenne.
- Les niveaux associés aux symboles sont adaptés aux valeurs prises par les séries temporelles à représenter, alors que SAX utilise un partitionnement prédéfini de façon à obtenir une distribution équi-répartie si la distribution des valeurs est normale.
- Le découpage en épisodes est adapté aux valeurs prises par les séries temporelles à représenter, alors que SAX utilise un découpage en épisodes de même taille.

Il faut bien sûr décider d'un critère pour adapter l'ensemble des niveaux et le découpage en épisodes. Conformément à l'objectif de fidélité aux données présenté en 2.2, nous chercherons à minimiser la somme des erreurs quadratiques entre les séries à représenter et les reconstructions permises par leur représentation symbolique.

Formellement, la définition de cette représentation symbolique est constituée à partir du cadre général présenté en 2.2 en précisant la définition de l'alphabet de niveaux prototypiques associé à l'alphabet symbolique $\Lambda = \{l_m\}_{m\in\{1\cdots K\}}$ avec $l_m \in \mathbb{R}$.

Puisque le découpage en épisodes et l'alphabet symbolique sont optimisés en fonction des données, deux séries temporelles dont on calcule les représentations symboliques n'auront a priori pas les mêmes ensembles d'épisodes et de niveaux prototypiques. Afin de simplifier les calculs des distances entre représentations symboliques de séries temporelles et d'avoir une comparaison plus immédiate avec SAX, nous avons décidé d'optimiser E et Λ sur l'ensemble des séries temporelles de la base. Cela est rendu possible par le fait que toutes nos séries temporelles sont définies sur le même domaine temporel D, et que leurs valeurs numériques sont commensurables. Il serait cependant tout à fait envisageable de calculer des découpages et/ou alphabets de niveaux symboliques propres à chaque série en définissant un calcul de distance adapté, par exemple sur le modèle de celui défini pour une autre représentation adaptative comme celle présentée par Keogh et Pazanni (1998).

2.4.2 Algorithme de construction

Avec les définitions que nous venons de voir en 2.4.1, l'erreur de reconstruction que nous cherchons à minimiser est $\mathrm{SSE}(\mathrm{ST}, R(\mathrm{RS}(\mathrm{ST}, P, K)))$, avec SSE la somme des erreurs quadratiques, ST la série temporelle à représenter et $R(\mathrm{RS}(\mathrm{ST}, P, K))$ la reconstruction obtenue à partir de la représentation symbolique de ST avec un découpage du domaine temporel en P épisodes et un alphabet de K niveaux prototypiques. Cette erreur de modélisation vaut $\sum_{e_j\in E}\sum_{i\in e_j}\left(v_i - \mathrm{DS}_\Lambda(e_j)\right)^2$ qui exprime que l'erreur à minimiser est la somme des erreurs quadratiques sur tous les épisodes de E et que l'erreur quadratique sur un épisode e_j est calculée entre les valeurs v_i de la série à modéliser et la description symbolique par

un niveau prototypique de Λ de cet épisode. Pour minimiser cette erreur, il faut aussi avoir $\mathrm{DS}_\Lambda(e_j) = \mathrm{argmin}_{l \in \Lambda}(\sum_{i \in e_j} (v_i - \mathrm{DS}_\Lambda(e_j))^2)$.

Calculer une SBSR-L0 optimale, revient donc à trouver les $P-1$ points de rupture entre épisodes qui définissent E et les K niveaux qui définissent Λ de façon à minimiser ce coût.

Pour des raisons évidentes de complexité algorithmique, il n'est pas envisageable de calculer simultanément E et Λ car la complexité algorithmique serait alors rédhibitoire, comme vu en 2.2. Il est néanmoins possible de construire une solution en procédant de façon itérative en optimisant alternativement E et Λ :

1. Étape initiale : calcul d'un découpage en épisodes E. En l'absence d'un alphabet de niveaux, on effectue une segmentation classique avec modèle linéaire d'ordre 0.
2. Calcul d'un ensemble de K niveaux permettant de minimiser l'erreur en reconstruction si chaque épisode de E est associé à l'un de ces niveaux. Cet ensemble définit l'interprétation de Λ.
3. Calcul d'un ensemble de $P-1$ points délimitant les P épisodes qui définissent E de façon à minimiser l'erreur en reconstruction si chaque épisode est associé à l'un des K niveaux trouvés en 2.
4. Itérer en retournant à l'étape 2 jusqu'à convergence de l'erreur en reconstruction. Cette convergence est garantie par le fait que les étapes 2 et 3 diminuent l'erreur en reconstruction et que celle-ci ne peut être négative.

Cet algorithme ne peut garantir qu'une optimalité locale de la solution trouvée, mais le volume des données à traiter est tel qu'il impose d'énormes contraintes sur la complexité algorithmique acceptable.

2.4.3 Illustration des résultats

Sur la gauche de la figure 3, nous montrons un extrait de la distribution des valeurs prises par les séries temporelles, centrées et réduites, de notre base de données, ainsi que les intervalles correspondants aux cinq symboles d'un alphabet d'une représentation SAX en traits pleins. Il ne s'agit que d'un extrait de la distribution car il y a dans la base quelques valeurs atypiques supérieures à 20. On constate que l'hypothèse d'une distribution normale qui sous-tend SAX n'est pas respectée. En effet, il n'y a pas de valeurs inférieures à zéro dans notre base, car nous étudions les mesures de quantités positives ou nulles, ce qui induit un "déséquilibre" de la distribution que l'on retrouve après avoir centré et réduit les séries temporelles. De plus, l'information sur les valeurs atypiques est complètement perdue par la représentation SAX. Sur cette même figure de gauche, les traits en pointillés et mixtes représentent les valeurs prototypiques que l'on peut associer respectivement aux symboles SAX et aux symboles SBSR-L0. Seulement quatre des cinq valeurs prototypiques de SBSR-L0 sont visibles sur la figure car la cinquième valeur, qui correspond aux valeurs atypiques, est située hors de l'extrait de distribution représenté. Sur la figure de droite, on a représenté le découpage en épisodes obtenu par SBSR-L0 sur un extrait d'une série temporelle. On remarque que celui-ci permet effectivement de mieux modéliser la série en question alors même qu'il est issu d'un découpage adapté globalement à l'ensemble de toutes les séries de la base.

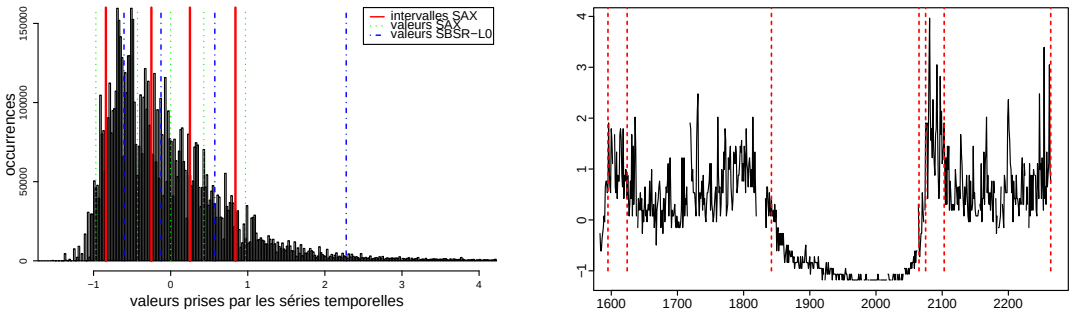


FIG. 3 – *Distribution des valeurs des séries temporelles et valeurs prototypiques. Découpage adapté aux données.*

3 Conclusions et recherches en cours

Au sein du cadre très général que nous avons présenté en 2.2, de nombreuses représentations symboliques sont envisageables.

3.1 Découpage et alphabets adaptés à des classes de séries temporelles

Pour permettre une comparaison équitable avec l'existant, nous avons étudié ici le cas de découpages et alphabets de symboles adaptés à l'ensemble des séries temporelles d'une base de données. Il serait bien sûr possible de calculer des découpages et/ou des alphabets adaptés à chacune des séries temporelles. Néanmoins, une telle multiplication des découpages et alphabets peut s'avérer préjudiciable lorsqu'il s'agit de comparer les séries temporelles entre elles. Pour cette raison, on aura intérêt à utiliser lorsque cela est possible des découpages et alphabets communs. A cette fin, on adaptera les découpages et alphabets à des classes homogènes de séries temporelles au sein de la base de données.

3.2 Généralisation à d'autres représentations symboliques basées sur des segmentations

SBSR-L0 est une représentation symbolique basée sur des segmentations par modèles linéaires d'ordre 0. Bien évidemment, le principe est généralisable à tous types de segmentations basés sur d'autres modèles dont on peut classifier automatiquement tout ou partie des paramètres. Nous avons par exemple étudié des représentations symboliques basées sur des segmentations linéaires d'ordre 1 associant les symboles à des classes de pentes.

3.3 Traitements en ligne de flux de données numériques

L'inconvénient majeur des représentations symboliques adaptatives est la nécessité de disposer de l'ensemble des séries temporelles à traiter, ainsi que le temps de calcul nécessaire aux algorithmes de construction de ces représentations. Ceci interdit a priori leur usage pour du traitement en ligne de flux de données. Néanmoins, dans la plupart des cas, il sera possible d'apprendre un alphabet hors-ligne sur un historique des données et d'utiliser celui-ci

pour réaliser en-ligne la représentation symbolique d'un flux de données numériques. Il faudra cependant veiller à s'assurer que l'alphabet précalculé reste pertinent pour représenter les nouvelles données, et éventuellement procéder aux réactualisations nécessaires.

Références

Hugueney, B. (2003). *"Représentations symboliques de longues séries temporelles"*. Ph. D. thesis, LIP6.

Keogh, E., K. Chakrabarti, M. Pazzani, et S. Mehrotra (2001a). Locally adaptive dimensionality reduction for indexing large time series databases. *SIGMOD Record (ACM Special Interest Group on Management of Data) 30*(2), 151–162.

Keogh, E. et M. J. Pazanni (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In D. Heckerman, H. Mannila, D. Pregibon, et R. Uthurusamy (Eds.), *Proceedings of the Forth International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI Press.

Keogh, E. J., K. Chakrabarti, M. J. Pazzani, et S. Mehrotra (2001b). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems 3*(3), 263–286.

Lin, J., E. Keogh, S. Lonardi, et B. Chiu (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11. ACM Press.

Yi, B.-K. et C. Faloutsos (2000). Fast time sequence indexing for arbitrary L_p norms. In A. El Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, et K.-Y. Whang (Eds.), *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10–14, 2000, Cairo, Egypt*, Los Altos, CA 94022, USA, pp. 385–394. Morgan Kaufmann Publishers.

Summary

Time series analysis is a very important topic for the data-mining field of study. Huge amounts of numerical data can be collected very easily, but their analysis is not straightforward. A common prerequisite to most data-mining tasks is changing data representation, both for dimensionality reduction and information extraction. In order to have meaningful representations, one can build symbolic representations of time series. We present a new generic framework for symbolic representations of time series and two kinds of such representations : CBSR (Clustering-Based Symbolic Representations) and SBSR-L0 (Segmentation-Based Symbolic Representation with Linear models of 0th order).

Algorithme d'apprentissage de scénarios à partir de séries symboliques temporelles

Thomas Guyet*, Catherine Garbay*
Michel Dojat**

*Laboratoire TIMC Thomas.Guyet@imag.fr,
http://www-timc.imag.fr/Thomas.Guyet/
**INSERM/UJF U594
michel.dojat@ujf-grenoble.fr

Résumé. Dans cet article, on présente un algorithme d'apprentissage de motifs fréquents (scénarios) à partir de séries symboliques temporelles exemples. La difficulté de ce problème réside dans le fait de prendre en compte la dimension temporelle des évènements symboliques et de pouvoir la retranscrire dans le motif appris aussi bien sous la forme de contraintes numériques (date, durée ou temps de séparation entre évènements) que de contraintes qualitatives (avant, après, pendant, ...). Après avoir introduit l'objet de la publication comme une problématique au domaine d'application très large et assez ouvert, on présentera la représentation des séries symboliques temporelles comme des hyper-rectangles que nous utilisons pour l'algorithme. Cette représentation permettra de conserver l'information quantitative et de mettre en évidence les contraintes qualitatives temporelles simples. Ensuite, on présentera notre algorithme, adaptation de l'algorithme `A Priori`, qui permet d'apprendre les scénarios.

1 Introduction

Dans cette article on présente un algorithme pour permettre un apprentissage de scénarios à partir de séries symboliques temporelles et découvrant aussi bien la structure symbolique des scénarios que les relations temporelles qui les spécifient. Après avoir précisé les notions qui nous sont utiles puis avoir fait un bref état de l'art du domaine, nous décrivons le principe de cet algorithme pour lequel, à defaut de proposer des expérimentations, nous évaluons sa complexité.

1.1 Séries symboliques temporelles

On appelle **série symbolique temporelle** un ensemble de symboles datés et qualifiés par une durée. En pratique, il s'agit d'une suite d'évènements, où un évènement est réprésenté par un symbole, qui a été extraite du fonctionnement d'un système. Même si un système est monitoré par plusieurs capteurs, une séquence de son fonctionnement pourra être réduite à

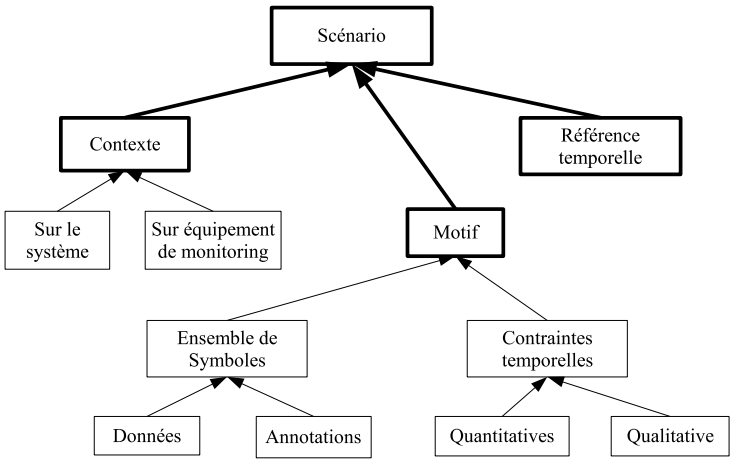


FIG. 1 – *Méta-modèle de scénario.*

une simple série temporelle symbolique (pour laquelle un symbole sera un couple *(IdSensor,IdEvent)* [1]).

Les séries symboliques temporelles sont des objets très courants car les systèmes monitorés sont très variés. Mais elles sont considérées, de par l'information temporelle (aussi bien quantitative que qualitative) qui y est contenue, comme des données complexes et de ce fait, les méthodes permettant d'en extraire des scénarios, en limitant le biais d'apprentissage, sont difficiles à mettre en œuvre.

1.2 Scénarios

Définissons maintenant ce que nous appelons des scénarios. On trouve parfois le terme de chronique (Dousson et Duong (1999)) qui recouvre la même idée. Les **scénarios** permettent de représenter des enchaînements d'évènements typiques du système. La figure 1 illustre un méta-modèle de scénario.

La partie principale d'un scénario est l'ensemble de symboles contraints par des relations temporelles qu'on appelera **motif**. Les symboles peuvent provenir soit directement de l'observation du système, soit provenant d'annotations faites pour informer de façon experte le fonctionnement du système (considérées comme une donnée et non comme un biais d'apprentissage). Les symboles sont associés à un ensemble de contraintes temporelles quantitatives (les dates, les durées, les temps entre évènements, ...) et/ou qualitatives (avant, pendant, ...) qui s'appliquent aux symboles pour les positionner dans le temps les uns par rapports aux autres. L'ensemble de ces contraintes doivent être cohérentes (ne pas se contredire entre elles) mais pas nécessairement positionner de façon déterministe chaque symbole. Lorsque des contraintes sont exprimées en terme de date, il est nécessaire que le scénario possède une référence temporelle pouvant correspondre au début ou à la fin (théorique) du scénario.

[1] *IdSensor* est l'identifiant du capteur, et *IdEvent* est l'identifiant d'un évenement qui peut être détecté avec ce capteur.

Le contexte de scénario permet d'informer les conditions générales du système qui correspond à l'observation de ce type de scénarios. Si on prend l'exemple de patients monitorés, les scénarios provenant de patients ayant subit un traumatisme est, en théorie ou pour étude, à distinguer du cas de patients sains. Le contexte peut aussi provenir des contraintes de récupération des données (disponibilités des capteurs, ...). Les informations contextuelles ne sont pas soumises à des contraintes temporelles, elles sont appliquées sur l'ensemble de celui-ci.

1.3 Apprentissage des motifs d'un scénarios

L'apprentissage de scénarios doit permettre de construire des scénarios à partir des séries temporelles symboliques qui ont été recueillies sur le système auquel on s'intéresse. On cherche ainsi à apprendre et comprendre le fonctionnement de celui-ci. Pour l'apprentissage de scénarios, le contexte est donné et commun à chaque exemple d'apprentissage et la référence temporelle doit permettre d'aligner les dates par rapport à cette référence commune. L'algorithme doit donc se focaliser sur la construction d'ensembles de symboles ainsi que les contraintes qui constituent des motifs communs aux exemples. Lorsqu'on parle d'apprentissage de scénario, il faut donc entendre apprentissage du motif d'un scénario.

Le motif d'un scénario doit être significatif soit parce qu'il représente des co-occurences d'évènements fréquentes (Sur l'observation d'un carrefour : à chaque fois qu'il y a un accident sur la voie, il y a un bouchon pendant au minimum 30 min), soit parce qu'il intéresse un expert du système qui cherche à comprendre l'occurence d'évènements (généralement critiques). Dans le premier cas, la recherche exhaustive des scénarios est très difficile car elle doit utiliser la globalité des séries temporelles et ne permet pas, dans le second cas, à partir du critère de fréquence, de retrouver des scénarios rares mais intéressants en pratique. En s'intéressant à la compréhension d'un évènement donné par l'expert, on peut se focaliser sur des parties des séries temporelles : celles qui sont proches[2] des occurences d'un évènement (*i.e.* un symbole S) intéressant. On constitue ainsi une base naturelle d'exemples à partir de laquelle on va chercher à construire un (des) motif(s) faisant intervenir S.

Dans la suite du papier, on se place dans le cadre d'un apprentissage de motifs à partir d'exemples. Les méthodes existantes d'apprentissage de scénarios présentées dans la section suivante ne permettent pas de prendre en compte toute l'information temporelle : soit parce cette information est traitée comme un paramètre à raffiner à partir d'une structure, soit parce que la représentation choisie ne rend pas compte de toute la richesse des scénarios. On propose ici (section 4) un algorithme permettant d'extraire, à partir des motifs exemples, les motifs les plus fréquents en extrayant les symboles *et* les contraintes temporelles quantitatives qui les composent.

2 Etat de l'art des méthodes d'apprentissage de motifs

Dans ce bref état de l'art non exhaustif, on s'intéresse dans un premier temps aux la représentations du temps, puis aux formalismes qui les utilisent pour représenter les motifs. Enfin, on donne des méthodes d'apprentissage de motifs.

[2]Les experts doivent alors définir quel est la limite de proximité temporelle entre deux évènements qui peut être considérée.

2.1 Représentations du temps

L'information temporelle peut se décrire à travers différentes représentations plus ou moins expressives. La logique de McDermott (1982) est une logique de points qui permet de conserver la structure linéaire du temps. Les prédicats peuvent être associés à des dates numériques. C'est une logique de description très expressive, mais en pratique les méthodes d'induction logique sur cette logique ne sont pas applicables (*i.e.* elles ne permettent pas d'apprendre à partir d'exemples). Au mieux elle permet de faire de la vérification de satisfaction de contraintes. On retrouve les mêmes difficultés dans l'utilisation de logiques modales.

Il faut alors pourvoir représenter le temps à l'aide de propositions. La méthode plus connue est certainement la logique d'Allen (1984) qui est une logique d'intervalles à base de sept prédicats décrivant les relations temporelles qualitatives élémentaires (Egal, avant, pendant, rencontre, recouvre, commence, fini). Dans Quiniou et al. (2003), l'abstraction du temps est faite sous la forme de prédicats (`court,normal,long`), mais dans les deux cas, il a fallu opérer à une réduction drastique de l'information temporelle. Les logiques floues permettent d'enrichir cette information en la valuant de manière plus fine.

2.2 Formalismes pour les motifs

Les graphes permettent de traduire de façon naturelle la notion de motif avec ses dimensions symbolique et temporelle. Les états permettent de représenter les symboles et les transitions contiennent une information basée sur une logique temporelle. Les transitions d'un graphe orienté informent naturellement sur le séquencement et parfois le parallélisme temporel des symboles. On distingue ici trois types de graphes. Les graphes temporels Dojat et al. (1998)), les réseaux de Petri temporisés (Bulitko et Wilkins (2005)) et les réseaux bayésiens (Boyen et al. (1999)). Ces derniers permettent représenter des séquences d'évènements à temps discret. Si on veut exprimer des durées d'évènement, on pourra aisément échantillonner à l'aide de la plus petite durée d'évènement et au besoin répéter un évènement pour exprimer la durée. L'expressivité est alors limitée par l'impossibilité de différencier "deux fois l'évènement A de durée t" et l'occurence de "l'évènement A de durée de $2t$". Ce problème n'existe pas pour les précédentes représentations. Pour les deux premiers formalismes, les transitions sont enrichies par une information temporelle qui permet de résoudre ce problème.

2.3 Méthodes d'apprentissage

La difficulté de l'apprentissage de motifs est de pouvoir extraire des exemples aussi bien l'information symbolique que l'information temporelle. Les algorithmes présentés ont des difficultés à faire intervenir les deux dimensions de manière équitable.

Certaines méthodes privilégient l'apprentissage de la dimension temporelle en partant d'une structure de motif préétablie. C'est le cas des réseaux bayésiens pour lesquels la structure est généralement donnée et les distributions des transitions sont apprises à l'aide des méthodes de HMM (*Hidden Markov Model*). Dans Bulitko et Wilkins (2005), les auteurs présentent des méthodes d'apprentissage à partir de réseaux de Petri pour lesquelles ils limitent l'apprentissage de la dimension symbolique du motif. En partant d'un pré-motif, l'algorithme permet d'apprendre les durées des transitions et de supprimer également certaines transitions et états qui ne sont pas signifiants.

A l'inverse, d'autres méthodes privilégient la dimension symbolique et la dimension temporelle est réduite à un raffinement de contraintes. Dans Quiniou et al. (2003), l'*Inductive Logic Programming* (Blockeel et Raedt (1994)) est utilisée pour apprendre des motifs sous la forme de propositions logiques. Elle fonctionne bien avec une représentation du temps par des propositions qui ont nécessité beaucoup de connaissances a priori et introduit un biais important dans l'apprentissage. La méthode `A Priori` permet à Agrawal et Srikant (1994) de construire les règles d'association où les séquences les plus grandes et de fréquence minimum fixée (f). Le principe étant basé sur le théorème que l'on énonce dans la section 4.2. L'algorithme consiste à construire des motifs de plus en plus grands en élaguant ceux qui ne sont pas de fréquence suffisament grande. L'espace de version parcouru a alors la forme d'un diamant : le nombre de motifs fréquents commence avec la taille du motif par grandir puis contraint par la nécessité d'être suffisament fréquent, ce nombre décroît ensuite en fonction de sa taille. De nombreuses heuristiques existent pour effectuer un parcours efficace de l'espace de version afin d'aller au plus vite rejoindre la limite recherchée (en un point ou en tout point de la limite). L'approche de C. Dousson (Dousson et Duong (1999)) est d'ajouter une dimension temporelle à cet algorithme pour faire l'apprentissage de séquences avec construction de contraintes temporelles. D'autres approches similaires par le fait d'utiliser les séquences d'évènements ont été proposées par Höppner (2002) ou encore Mannila et al. (1995).

L'algorithme qui est proposé, dans la suite, s'appuie sur une représentation géométrique d'un motif qui concerve une information temporelle riche, sans biais d'apprentissage, sous la forme d'intervalle définie par des valeurs numériques. On a raisonné sur cette représentation pour contruire un algorithme qui permet d'extraire les motifs, sur le principe de l'algorithme `A Priori`, en découvrant la structure symbolique et les contraintes temporelles.

3 Représentation d'un motif par un hyper-rectangle

Les exemples dont on se sert pour notre algorithme sont des motifs. A chaque symbole, y compris s'il y a plusieurs occurrences de symboles de même nature dans un motif, est associé une dimension (échelle des temps) sur laquelle on délimite la position dans le temps du symbole. Un motif comportant N symboles, dit de taille N, sera représenté dans un espace à N dimensions. En intersectant l'ensemble des contraintes quantitatives, on définit ainsi un hypercube unique pour chaque motif. Cet hypercube est en effet la représentation géométrique du motif. Dans Chouakria (1998), on pourra trouver des détails sur ce type de représentation qui est appliquée, plus généralement, à des données de type intervalle. On pourra trouver des généralisations des méthodes statistiques à ce type de données chez Diday et Billard (2003). D'autres représentations géométriques de motifs sont discutées dans Kosara et Miksch (2001). Si dans la suite on ne s'intéresse qu'aux contraintes quantitatives, on peut donner des interprétations de relations qualitatives temporelles à l'aide de la position des sommets par rapport aux hyperplans diagonaux.

Afin de mieux faire comprendre ces idées, on présente cette représentation pour les motifs de tailles 3. Le motif comportant 3 symboles A, B et C tel que le montre la figure 2 (à gauche) peut être représenté dans un espace à 3 dimensions par le cube (un *3*-hyperrectangle) de la figure 2 (à droite). Chaque dimension représente le temps pour un évènement.

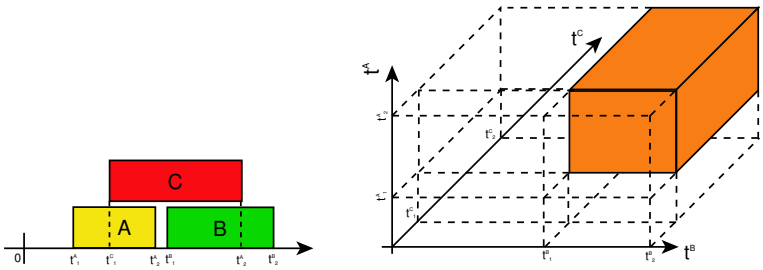


FIG. 2 – *Motif de taille 3 : à gauche le motif, et à droite sa représentation en 3 dimensions.*

4 Algorithme d'apprentissage

L'algorithme construit à partir des motifs exemples par agrégation progressive de symboles le plus grand motif commun à ces exemples et de fréquence supérieure à f (f est un paramètre de l'algorithme) reprenant ainsi l'idée de l'algorithme `A Priori`. Mais dans notre représentation, l'agrégation d'un nouveau symbole signifie augmenter les motifs en cours de construction d'une dimension.

L'algorithme, dont le pseudo-code est donné en annexe, est la répétition de successions de deux étapes, à savoir la sélection des motifs de dimension N suffisamment fréquents, et la construction des candidats-motifs de dimension $N+1$ à partir de ceux de dimension N, qui sont présentées en détails dans les paragraphes suivants. L'arrêt se fait lorsque qu'il n'y a plus de candidats possibles à construire ou que ceux-ci ne sont plus suffisamment fréquents.

4.1 Sélection des motifs de dimension N communs aux exemples

Soit k le nombre de symboles différents constituant les séries symboliques. Dans le cas le plus général, il y a au plus $k * N$ types de motifs de dimension N possibles, par la suite on limitera ce nombre. On associe à chacun un hyper-histogramme : histogramme en dimension $N+1$. Pour chaque exemple, on extrait les motifs de dimension N et on empile l'$(N+1)$-hypercube (de base l'hyperrectangle défini par le motif et de hauteur 1) sur l'hyperhistogramme correspondant (*i.e.* celui où la base de l'hyperhistogramme est l'espace de représentation du motif). La figure 3 donne l'exemple de l'empilement d'hypercube de dimension 2, pour former un histogramme en dimension 3 (c'est le seul cas visualisable).

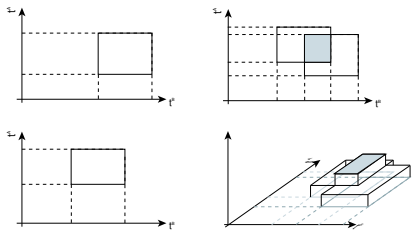


FIG. 3 – *Empilement de motifs de taille 2.*

Cet histogramme peut être vu comme une approximation (modulo la loi des grands nombres) de la probabilité du motif de dimension N. L'utilisation d'un algorithme de recherche de mixture de gaussiennes (Algorithme EM ou dérivé pour lequel on prend des exemples tirés au hasard avec des probabilités définies par l'histogramme) permet alors de définir ce motif comme une gaussienne à $2N$ paramètres : N moyennes et N variances pour la date moyenne et la durée moyenne de chaque symbole dans le motif. Si on permettait d'avoir des covariances non nulle, pouvant être traduit par l'interdépendance des durées moyennes des motifs, alors il faudrait N paramètres de moyenne et une matrice de covariance de dimension N : pour simplifier dans cette première version de l'algorithme les calculs et l'explication, on ne considère pas ces co-variances. Dans Chouakria (1998), d'autres méthodes d'analyses factorielles permettent de faire cette même étape.

4.2 Passage à la dimension supérieure

On cherche maintenant à construire des candidats-motifs de dimension $N+1$ à partir des motifs de dimension N. L'idée de l'algorithme `A Priori` qui est reprise ici peut être traduite par un théorème dont la preuve est triviale par l'absurde.

Théorème : Un motif de dimension $N+1$ ne peut être de fréquence supérieure à f si tous les sous-motifs qui le composent (en particulier ceux de dimension N) ne sont pas au moins de fréquence f.

On va donc construire les candidats à partir des motifs de dimension N, ils sont moins nombreux et on peut ainsi réduire l'espace de recherche au fur et à mesure que la dimension augmente. De plus, on sait que *a posteriori*, les motifs de dimension N correspondent à des projections suivant la dimension complémentaire associée à l'un des symboles possibles du motif. La figure 4 illustre cette construction.

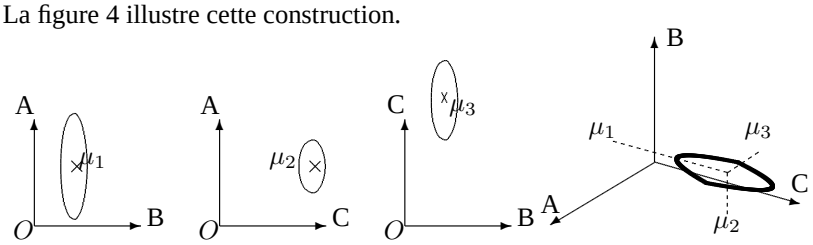


FIG. 4 – *Illustration de la construction de motifs de taille 3 à partir de motifs de tailles 2.*

Par conséquent, deux conditions sont nécessairement à vérifier par les candidats-motifs. Si le motif est constitué des symboles $(S_1, S_2, ..., S_{N+1})$, il faut :

1. (Condition du théorème) $\forall n \in [1, N+1]$, il existe un motif de dimension N de fréquence supérieur à f avec les symboles :
$$(S_1, S_2, ..., S_{n-1}, S_{n+1}, ..., S_{N+1}).$$
2. Les $N+1$ motifs de dimension N doivent être consistants, *i.e.* qu'ils ne doivent pas violer la condition *a posteriori* de projection.

En pratique, on peut vérifier ces conditions (sur la figure 4) en calculant la distance entre μ_3 et la projection du point en 3 dimensions construit à partir de μ_1 et μ_2 sur le plan BC. Une

fois que ces conditions ont été vérifiées, on passe a une nouvelle étape de sélection des motifs de taille $N + 1$ maintenant. On considère ainsi des motifs de tailles de plus en plus grandes élaguant ceux qui ne nous conviennent pas.

4.3 Complexité et limite

4.3.1 Fléau de la dimension

Le fléau (ou malédiction) de la dimension de Bellman (1961) est un phénomène statistique qui fait intervenir le *phénomène de l'espace vide* et le *peaking phenomenon* (Robineau (2004)). Retenons simplement de ces deux phénomènes, que d'une part il est très difficile de connaître le nombre d'exemples nécessaire pour faire un apprentissage et que d'autre part pour l'approximation d'une distribution normale multivariée le résultat de Silverman (1989) montre que le jeu d'exemples doit croître exponentiellement.

Le problème se pose dans notre algorithme lorsqu'il faut estimer à l'aide de l'algorithme EM la mixture de Gaussienne qui permet de générer les candidats. Le nombre d'exemples à tirer à partir de l'histogramme doit croître exponentiellement. Par conséquent, la taille des motifs à apprendre sera limitée rapidement par cette méthode.

4.3.2 Estimation de la complexité

Soient M le nombre de symboles, P le nombre d'exemples et N le taille des motifs finaux. En supposant que jusqu'au dernier rang toutes les combinaisons de symboles restent suffisamment fréquentes et simples (pas de motifs différenciés par les contraintes temporelles uniquement), alors le calcul donne la complexité de l'algorithme de l'ordre de :

$$\frac{1 - M^{N+1}}{1 - M} (CoutEM + P + M)$$

Comme dans l'algorithme `A Priori`, le coût est exponentiel en fonction de la taille du motif appris, sans même se poser la question du coût de l'algorithme EM qui augmente exponentiellement également car on doit augmenter exponentiellement le nombre d'exemples pour limiter le fléau de la dimension.

La complexité de l'algorithme ainsi que malédiction de la dimension limiteront la taille des motifs apprenables. Seule l'expérimentation permettra d'estimer la taille des motifs qui pourra être apprise, en testant différentes heuristiques.

5 Conclusion

Dans ce papier, après avoir proposé un cadre général pour basée sur une représentation des scénarios sous la forme d'un hyper-rectangle. La méthode proposée permet d'intégrer la dimension temporelle et symbolique dans la construction des plus grands motifs communs à des exemples de séries symboliques temporelles et de fréquence minimum f. Elle présente une complexité algorithmique importante et laisse envisager des difficultés pour l'apprentissage de scénarios de grande taille. Cependant, la perspective est d'implémenter à court terme celui-ci pour pouvoir en faire une évaluation tant sur ses capacités à tirer de l'information temporelle et symbolique que sur son efficacité.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, et C. Zaniolo (Eds.), *Proceedings of the 20th Internationnal Conférence in Very Large Databases*, pp. 487–499. Morgan Kaufmann.

Allen, J. (1984). Towards a general theory of action and time. *Artificial Intelligence 23*, 123–154.

Bellman, R. E. (1961). *Adaptative Control Processes : A Guided Tour*. New Jersey, U.S.A. : Princeton University Press.

Blockeel, H. et L. D. Raedt (1994). Inductive Logic Programming : Theory and Methods. *The Journal of Logic Programming 19&20*, 629–680.

Boyen, X., N. Friedman, et D. Koller (1999). Discovering the hidden structure of complex dynamic systems. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 91–100. San Francisco : Morgan Kaufmann.

Bulitko, V. et D. C. Wilkins (2005). Machine Learning for Time Interval Petri Nets. In *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Sydney, Australia, pp. 959–965. Springer-Verlag.

Chouakria, A. (1998). *Extension des méthodes d'analyse factorielle à des données de type intervalle*. Ph. D. thesis, University Paris IX Dauphine.

Diday, E. et L. Billard (2003). From the statistics of data to the statistic of knowledge : Symbolic data analysis. *Journal of the American Statistical Association 98*(462).

Dojat, M., N. Ramaux, et D. Fontaine (1998). Scenario recognition for temporal reasoning in medical domains. *Artificial Intelligence in Medicine 14*, 139–155.

Dousson, C. et T. V. Duong (1999). Discovering Chronicles with Numerical Time Constraints from Alarm Logs for Monitoring Dynamic Systems. In T. Dean (Ed.), *IJCAI*, pp. 620–626. Morgan Kaufmann.

Höppner, F. (2002). Learning Dependencies in Multivariate Time Series. In *Proceedings of the ECAI'02 Workshop on Knowledge Discovery in (Spatio-) Temporal Data*, pp. 25–31.

Kosara, R. et S. Miksch (2001). Visualizing Complex Notions of Time. In *Proceedings of the Conference on Medical Informatics*, pp. 211–215.

Mannila, H., H. Toivonen, et A. Verkamo (1995). Discovering frequent episodes in sequences. In *Proceedings of 1st International Conference on Knowledge Discovery and Data Mining (KDD'95)*, Montreal, Canada, pp. 210–215.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive Science 6*, 101–155.

Quiniou, R., G. Carrault, M.-O. Cordier, et F. Wang (2003). Temporal abstraction and inductive logic programming for arrythmia recognition from electrocardiograms. *AIME 28*, 231–263.

Robineau, J.-F. (2004). *Méthode de sélections de variables (parmi un grand nombre) dans un cadre de discrimation*. Ph. D. thesis, Université Joseph Fourier.

Silverman, B. W. (1989). *Density Estimation for Statistics and Data Analysis*. London : Chapman & Hall.

Annexe : Pseudo-code de l'algorithme

procédure A Priori **(**f : fréquence min, E : les exemples**)**
début
    $L_2 \leftarrow$ {Motifs de dimension 2, fréquents}
    $k \leftarrow 2$
    Tant que $L_k \neq \emptyset$ **faire**
        Pour tout $m \in L_k$ **faire**
            $h \leftarrow$ CreerHistogramme(m,E)
            $M_m \leftarrow$ CreerMotifs(h,f)
        Fin pour tout
        $L_{k+1} \leftarrow$ GenererCandidats(M)
        $k \leftarrow k + 1$
    Fin tant que
fin

procédure CreerHistogramme **(**m : Espace support, E : Series Exemples**)**
début
    $H \leftarrow$ Histogramme plat de base m
    Pour tout $ex \in E$ **faire**
        $H(ex) \leftarrow H(ex) + 1$
    Fin pour tout
    retourner H
fin

procédure CreerMotifs **(**H : Histogramme, f : Fréquence minimum**)**
début
    $M \leftarrow$ Résultat de l'algorithme *EM*
    $M \leftarrow$ Suppression de M des Gaussiennes trop 'faibles' pour f
    retourner M
fin

procédure GenererCandidats **(**M : Liste de motifs de dimension k**)**
début
    $C \leftarrow$ Liste vide de motifs de dimension $k + 1$
    $C \leftarrow$ Ensemble des motifs de dimension $k + 1$ vérifiant les 2 contraintes : A Priori et A Posteriori
    retourner C
fin

Summary

This article presents an algorithm to learn frequent patterns (motifs) from examples provided as symbolic times series. The main difficulties are 1) to take into account of the temporal dimension of the symbolics events and 2) to find quantitative constraints (time stamp, durations, ...) and qualitative constraints (before, during, ...) between them. After introducing the issue as a very large and open problem, we propose to represent symbolic time series as hyper-rectangles. This representation allows to keep quantitative information and to easily extract simple qualitative temporal information. Then, we present the algorithm, based on the `A Priori` algorithm, which allows us to build the biggest and the most frequent pattern.

Partitionnement de données paramétriques en zones de comportement homogène dans un but explicatif

Manoranjan Muniandi*, Christophe Demko**, Bertrand Vachon***

L3I, Université de La Rochelle, PST
Avenue Michel Crépeau - 17042 LA ROCHELLE Cedex 1

*manoranjan.muniandi@univ-lr.fr
**christophe.demko@univ-lr.fr
***bertrand.vachon@univ-lr.fr

http://www.univ-lr.fr/labo/l3i/

Résumé. Afin d'étudier le comportement d'un système évalué par la mesure de données, l'étude de l'évolution des paramètres par rapport à d'autres et/ou par rapport au temps est indispensable. La méthode de *simulation qualitative* est un processus d'inférence capable d'exprimer la connaissance implicite et de dériver l'ensemble de tous les comportements possibles malgré l'imperfection du modèle. Des informations qualitatives, tels que les signes des dérivées, sont étudiés pour le partitionnement des données en zone de comportement homogène. Des règles de contraintes qualitatives sont extraites pour exprimer la dynamique du comportement du système en utilisant des représentations discrètes: états, actions, événements et changements d'états. L'évolution des paramètres est analysée sous une forme symbolique permettant de représenter les données sous une forme compréhensible tels que les graphes causaux. Le *bagging* est employé comme méthode ensembliste pour éliminer le bruit des données.

1 Introduction

Les systèmes réels sont généralement complexes et souvent chaotiques. Il y a, très souvent, des changements modifiant l'état du système à chaque instant (Forbus, 1984; De Kleer et Brown, 1984). Mais le raisonnement humain s'en accomode de façon très satisfaisante et arrive souvent à une conclusion malgré la connaissance incomplète et l'imperfection du modèle (Kuipers, 1986).

Dans plusieurs domaines du traitement des données, i.e. la fouille de données, l'apprentissage, les systèmes experts, les systèmes de filtrage d'information, ou les systèmes de reconnaissance des formes paramétriques, les données sont souvent ou toujours incertaines, imprécises ou incomplètes et représentent l'imperfection du modèle.

Le raisonnement causal qualitatif, la *simulation qualitative* en particulier, est une technique d'inférence capable d'exprimer la nature de la connaissance partielle et de dériver l'ensemble de tous les comportements possibles malgré la complexité et l'imperfection du système (Kuipers, 2001). Fondé sur les mathématiques qualitatives, telles que les équations différentielles

qualitatives, la *simulation qualitative* représente symboliquement les changements continuels d'un système en utilisant des valeurs qualitatives, des variables qualitatives et des contraintes qualitatives sur l'espace des quantités (Kuipers, 1994).

Dans notre étude, nous utilisons l'estimation des dérivées afin d'étudier leurs signes. Cette information qualitative est alors employée pour partitionner l'espace des quantités en zone de comportement homogène. Les valeurs qualitatives retenues pour un système pourrait être par exemple :

- strictement négative ;
- négative ;
- zéro
- positive ;
- strictement positive.

Toutes les lois possibles et nécessaires sont extraites en explorant les données pour détecter les contraintes qualitatives sous-jacentes permettant de réaliser la simulation qualitative. En initialisant le processus à partir d'un état initial, la *simulation qualitative* prédit toutes les descriptions qualitatives des états qui peuvent vraisemblablement être des successeurs directs de l'état actuel. La répétition de ce processus crée un graphe de comportement qui exprime les dynamiques de l'évolution des données en termes discrets. Explorer ce graphe à partir de l'état initial permet de comprendre les comportements sous-jacents.

Le *bagging* (Breiman, 1994) (*bootstrap aggregating*), est une méthode ensembliste employée pour éliminer le bruit. C'est une procédure permettant de construire un estimateur en utilisant des techniques de rééchantillonnage et de combinaison (Grandvalet, 2000). Nous concentrons actuellement nos efforts pour l'utilisation d'une telle méthode pour le filtrage des données parasites.

2 Raisonnement causal qualitatif

Un système expert est souvent considéré comme *un modèle peu profond*, dans le sens où toutes les conclusions sont inférées des observations de la situation actuelle (Kuipers, 1986). Pour construire *un modèle profond* qui peut exprimer la connaissance incomplète où l'état d'une situation n'est peut-être pas directement observable, de nombreuses recherches sont effectuées dans le domaine de raisonnement causal qualitatif :

- La physique qualitative ou la physique naïve (De Kleer et Brown, 1984) ;
- La théorie des processus qualitatifs (Forbus, 1984) ;
- La simulation qualitative (Kuipers, 1986).

Quels que soient les domaines étudiés ou les approches utilisées, l'inférence principale est la simulation qualitative : une description du comportement est dérivée des équations de contraintes qualitatives.

Simulation Qualitative

La *simulation qualitative* est fondée sur le modèle des équations différentielles qualitatives qui est une abstraction des équations différentielles ordinaires. Les valeurs des variables sont décrites en termes de relations ordinales par rapport à des valeurs symboliques. Les relations

fonctionnelles entre les variables sont décrites en termes de fonctions monotones sur des régions particulières. Ces descriptions qualitatives peuvent être étendues avec des connaissances semi-quantitatives, ou floue (Sugeno et Yasukawa, 1993).

Le modèle qualitatif est représenté par l'ensemble des variables reliées à des contraintes. La portée de chaque variable, y compris la variable indépendante *temps*, est décrite qualitativement par un espace de quantité limité et contenant des valeurs symboliques nommées bornes. Celles-ci représentent les valeurs qualitativement importantes (initialement : $-\infty$, 0 et $+\infty$) pour lesquelles des événements intéressants surviennent. Les contraintes liées aux variables peuvent être algébriques, différentielles, ou fonctionnelles.

2.1 Contraintes

2.1.1 Contraintes algébriques

Elles expriment des relations simples d'évolution conjointes de paramètres. Parmi la multitude de contraintes possibles, l'attention est principalement portée sur 3 d'entre elles :

$$\begin{aligned} \text{add}(x,y,z) &\equiv x(t) + y(t) = z(t) \\ \text{mult}(x,y,z) &\equiv x(t) * y(t) = z(t) \\ \text{minus}(x,y) &\equiv y(t) = -x(t) \end{aligned}$$

2.1.2 Contraintes différentielles

Les contraintes différentielles expriment l'évolution d'un paramètre par rapport au signe d'un autre.

$$\begin{aligned} \text{deriv}(x,y) &\equiv \tfrac{d}{dt}x(t) = y(t) \\ \text{const}(x) &\equiv \tfrac{d}{dt}x(t) = 0 \end{aligned}$$

2.1.3 Contraintes fonctionnelles

Les contraintes fonctionnelles M^+ et M^- utilisent des fonctions anonymes dont les formes explicites sont inconnues mais qui sont monotones sur l'intervalle considéré :

$$\begin{aligned} \text{M+}(x,y) &\equiv y(t) = f(x(t)), f \in M^+ \\ \text{M-}(x,y) &\equiv y(t) = -f(x(t)), f \in M^- \end{aligned}$$

2.2 Valeur qualitative

La magnitude d'une variable est qualitativement décrite par une borne ou par un intervalle entre deux bornes dans l'espace des quantités. La valeur qualitative d'une variable $\text{qval}(v)$ est donc décrite par le signe de sa dérivée et sa magnitude qualitative $(\text{qdir}(v), \text{qmag}(v))$. $\text{qdir}(v)$ prend initalement ses valeurs dans le domaine $\{dec, std, inc\}$ (*en diminution*, *stable* ou en *augmentation*). Cet ensemble de valeurs admises pour le signe de la dérivée peut être

complété par d'autres modes exprimant l'évolution de la variable (*en nette diminution*, *en nette augmentation*, etc).

La simulation commence à partir d'une condition initiale (l'état initial) et prédit tous les états possibles, tout en respectant les contraintes qualitatives qui expriment des comportements homogènes d'évolution. En répétant ce processus la *simulation qualitative* crée un graphe de comportement qui exprime l'évolution du système.

L'explosion combinatoire du nombre de prédictions des comportements peut générer un graphe des états qualitatifs assez volumineux. Explorer les comportements dans ce graphe nécessite des techniques d'abstraction pour obtenir une conclusion. Les méthodes de vérification et de surveillance de processus doivent être prise en compte pour éliminer l'existence des prédictions de comportements factices et de comportements cycliques.

3 Données évolutives

Le but de ce travail est de représenter graphiquement l'évolution de données paramètriques par rapport à d'autres. Pour ce faire, nous nous efforçons de rechercher des algorithmes efficaces permettant de trouver des lois de contraintes qualitatives afin de partitionner l'espace des données paramétriques en zone de comportement homogène. Cette information est ensuite utilisée pour effectuer la *simulation qualitative* qui crée une représentation symbolique de l'evolution des données, tout en respectant les lois de contraintes qualitatives extraites.

Soit X *l'univers des données* représenté par p paramètres ($X = (x_{ik})_{i \in 1..p, k \in 1..n}$). Si les données intègrent le temps t et sont analysées selon t, le comportement de chaque paramètre P_i est étudié selon t. Si les données sont indépendantes du temps t, les étapes suivantes sont effectuées :

- Un des paramètre intéressant P_i est choisi pour exprimer le temps t
- X est trié selon $t : Y = \text{sort}(X, t)$
- Le comportement de chaque paramètre P_i est étudié selon t
- Y est ensuite partitionné en zones de comportement homogène selon t

Afin d'obtenir les évolutions de chaque paramètre il est nécessaire d'effectuer l'estimation des dérivées.

3.1 Estimation des dérivées

Les changements intéressants marquant les comportements de chaque paramètre P_i par rapport à t sont exprimés en fonction de l'étude des dérivées selon t. Seules les informations qualitatives, tels que les signes (dec, std, inc) des dérivées sont étudiées. Elles sont ensuite utilisées pour décomposer l'espace des quantités en bornes et en intervalles, où chaque variable suit un comportement d'évolution homogène. L'estimation des dérivées de y_{ik} selon t est décrite par la matrice suivante :

$$\begin{bmatrix} \widehat{y'_{1k}} & \widehat{y'_{2k}} & \cdots & \widehat{y'_{ik}} & \cdots & \widehat{y'_{pk}} \\ \widehat{y''_{1k}} & \widehat{y''_{2k}} & \cdots & \widehat{y''_{ik}} & \cdots & \widehat{y''_{pk}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \widehat{y^{(m)}_{1k}} & \widehat{y^{(m)}_{2k}} & \cdots & \widehat{y^{(m)}_{ik}} & \cdots & \widehat{y^{(m)}_{pk}} \end{bmatrix}$$

où, $\widehat{y_{ik}^{(s)}}$ représente l'estimation de la dérivée s-ième de l'individu k pour le paramètre i.

Des techniques de *bagging* sont actuellement en cours de test pour estimer cette matrice. L'élimination des individus parasites est en effet primordiale pour obtenir de bons résultats qualitatifs.

3.2 Expressions des relations

Afin d'effectuer au mieux la *simulation qualitative*, chaque paramètre doit être contraint. Les relations entre les paramètres doivent être expressives et significatives par rapport au temps *t*. Puisque on ne considère, pour l'étude des dérivées, que les informations qualitatives tels que leurs signes, $\text{sgn}(y^{(s)}) \in \{dec, std, inc\}$, les expressions qualitatives entre les paramètres peuvent être exprimées par les relations suivantes :

Relations unaires

$$\begin{array}{lcl} \text{sgn}(y_i^{(s)}) = - & \Rightarrow & \text{qdir}(y_i^{(s-1)}) = dec \\ \text{sgn}(y_i^{(s)}) = 0 & \Rightarrow & \text{qdir}(y_i^{(s-1)}) = std \\ \text{sgn}(y_i^{(s)}) = + & \Rightarrow & \text{qdir}(y_i^{(s-1)}) = inc \end{array}$$

Relations binaires

$$\begin{array}{lcl} \text{sgn}(y_i^{(s)}) = \text{sgn}(y_j^{(s)}) & \Rightarrow & M^+(y_i^{(s-1)}, y_j^{(s-1)}) \\ \text{sgn}(y_i^{(s)}) = \text{opp}(\text{sgn}(y_j^{(s)})) & \Rightarrow & M^-(y_i^{(s-1)}, y_j^{(s-1)}) \\ \text{sgn}(y_i^{(s+1)}) = \text{sgn}(y_j^{(s)}) = 0 \,\&\, \text{sgn}(y_i^{(s)}) = \text{sgn}(y_j^{(s-1)}) & \Rightarrow & \text{deriv}(y_i^{(s-1)}, y_j^{(s-1)}) \\ \text{sgn}(y_i^{(s)}) = \text{opp}(\text{sgn}(y_j^{(s)})) & \Rightarrow & \text{minus}(y_i^{(s)}, y_j^{(s)}) \end{array}$$

Relations ternaires

$$\begin{array}{ll} \text{sgn}(y_i^{(s)}) = \text{sgn}(y_j^{(s)}) = \text{sgn}(y_k^{(s)}) & \Rightarrow \text{add}(y_i^{(s)}, y_j^{(s)}, y_k^{(s)}) \\ \text{sgn}(y_i^{(s)}) = \text{sgn}(y_k^{(s)}) \,\&\, \text{sgn}(y_j^{(s)}) = 0 & \Rightarrow \text{add}(y_i^{(s)}, y_j^{(s)}, y_k^{(s)}) \\ \text{sgn}(y_j^{(s)}) = \text{sgn}(y_k^{(s)}) \,\&\, \text{sgn}(y_i^{(s)}) = 0 & \Rightarrow \text{add}(y_i^{(s)}, y_j^{(s)}, y_k^{(s)}) \end{array}$$

Où, y_i, y_j et $y_k \in P_i$.

4 Conclusion

Dans ce travail, nous proposons une méthode pour extraire des informations de comportement qualitatif en utilisant une estimation des dérivées. Nous en sommes encore au stade expérimental et nous cherchons actuellement à mettre au point l'estimation des dérivées qui est le processus critique de notre système. Nous espérons que cette phase de test achevée, nous pourrons effectivement extraire des graphes causaux exprimant les lois d'évolutions des paramètres. L'analyse du graphe d'évolution du système permettra à terme de générer des règles de système expert permettant de prédire l'évolution de données en partant d'un état initial.

Partition...

Références

Breiman, L. (1994). Bagging Predictors. Technical Report 421, Department of Statistics, University of California at Berkeley, Department of Statistics, University of California at Berkeley, California 94720.

De Kleer, J. et J. S. Brown (1984). A Qualitative Physics Based on Confluences. *Artificial Intelligence 24*, 7–83.

Forbus, K. D. (1984). Qualitative Process Theory. *Artificial Intelligence 24*, 85–168.

Grandvalet, Y. (2000). Bagging Down-Weights Leverage Points. *IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00) 4*, 505–510.

Kuipers, B. J. (1986). Qualitative Silmulation. *Artificial Intelligence 29*, 289–338.

Kuipers, B. J. (1994). *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press.

Kuipers, B. J. (2001). Qualitative Simulation. In Robert A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology* (Third Edition ed.)., pp. 14. Academic Press.

Sugeno, M. et T. Yasukawa (1993). A Fuzzy-Logic-Based Approach to Qualitative Modelling. *IEEE Transactions on Fuzzy Systems 1*(1), 7–31.

Summary

In this work we investigate the evolution of parameters with respect to time and/or with other parameters in order to find the behavior of a model given in the form of data. Qualitative simulation method as a key process for representing and reasoning with incomplete knowledge is employed to predict all possible behavior from data derivatives for such model. Signs of data derivatives are studied and this qualitative information is then used to partition data into zones where behaviors are homogeneous. Qualitative constraint rules are extracted for qualitative simulation, where the dynamics of behavior in discrete terms, such as states, actions, events and events-triggered state-changes are explored. Thus evolution is described symbolically, in understandable form like finite-state causal graphs. Bagging (bootstrap aggregating) as an ensemble method for noise elimination is applied to data so that chatter in variables can be reduced.

FIDS : Extraction efficace d'itemsets dans des flots de données

Chedy Raïssi[*,**], Pascal Poncelet[**] et Maguelonne Teisseire[*]

[*]LIRMM UMR CNRS 5506, 161 Rue Ada, 34392 Montpellier cedex 5 - France
{raissi,teisseire}@lirmm.fr
[**]EMA-LGI2P/Site EERIE, Parc Scientifique Georges Besse, 30035 Nîmes Cedex, France
{Chedy.Raissi,Pascal.Poncelet}@ema.fr

Résumé. La recherche d'itemsets dans des flots de données (*data stream*) est un nouveau problème auquel se trouve confrontée la communauté Fouille de Données. Dans cet article, nous proposons une nouvelle approche pour extraire des itemsets maximaux de ces flots. Elle possède les avantages suivants : une nouvelle représentation des items et une nouvelle structure de données pour maintenir les itemsets fréquents couplée avec une stratégie d'élagage efficace. A tout moment, les utilisateurs peuvent ainsi connaître quels sont les itemsets fréquents pour un intervalle de temps donné.

1 Introduction

Pour de nouvelles applications émergentes (transactions financières, navigation sur le Web, téléphonie mobile, ...), il s'avère que les techniques classiques de fouille de données ne sont plus adaptées. En effet, dans ce contexte, les données manipulées sont des séquences d'items obtenues en temps réel, de manière continue et ordonnée (*data streams*). Ainsi, nous nous trouvons confrontés à des flux très importants, potentiellement infinis, de données (paquets TCP/IP, transactions, clickstreams, capteurs physiques, ...) et l'accès rapide à l'intégralité des données devient impossible. Un tel contexte impose alors que l'extraction de connaissances se fasse de manière très rapide (si possible en temps réel) car les résultats doivent pouvoir être utilisés pour réagir rapidement (adaptation en temps réel à des utilisateurs dans le cas de clickstreams, détection de fraude, supervision de processus, ...). Récemment, de nouveaux travaux de recherche, appelés "*Data Stream Mining*", s'intéressent à la définition d'algorithmes de fouille pour prendre en compte ces nouveaux besoins [9, 4, 3, 8, 12]. Comme les algorithmes traditionnels nécessitent plusieurs passes ou bien que le contenu entier de la base soit accessible ils ne sont plus adaptés et l'un des challenges important est de trouver et maintenir les itemsets fréquents des flots [6, 7]. Dans cet article, nous proposons une nouvelle approche appelée FIDS (*Frequent Itemsets mining on Data Streams*) pour extraire les itemsets fréquents dans des flots continus de données. L'une des originalités de notre approche est d'utiliser une nouvelle structure de données, couplée à une stratégie d'élagage rapide, pour maintenir les itemsets. Cette dernière offre la possibilité de réduire l'espace de recherche efficacement dans la mesure où nous sommes à même de retrouver rapidement quel est l'ensemble minimal d'itemsets à considérer lorsque de nouvelles données arrivent dans le flot. Nous reconsidérons également la

problématique de manipulation des itemsets en proposant une nouvelle représentation de ces derniers par des nombres premiers. L'avantage de cette représentation est de traduire les itemsets en une seule valeur par une multiplication de nombres premiers et d'utiliser certaines propriétés arithmétiques associées pour optimiser les traitements. Enfin, nous intégrons la notion de tilted-time windows table, précédemment introduite dans [4], pour permettre à l'utilisateur final de connaître quels sont les itemsets fréquents pour n'importe quelle période de temps. En effet, étant donné les contraintes existantes sur les flots, il n'est plus possible de donner une réponse exacte. Nous pouvons toutefois garantir que notre approche propose une réponse *approximative*, la plus proche possible du réel, par égard à un seuil d'erreur fixé, aux requêtes de l'utilisateur.

Le reste de l'article est organisé de la manière suivante. La Section 2 présente plus formellement le problème. Dans la Section 3, nous proposons un survol des travaux liés et nos motivations pour une nouvelle approche. La Section 4 présente notre solution. Les expérimentations sont décrites dans la Section 5 et une conclusion est proposée dans la Section 6.

2 Problématique

La problématique de la recherche d'itemsets fréquents a initialement été définie par [1] : Soit $I = \{i_1, i_2, \ldots i_m\}$ un ensemble d'items. Soit une base de données DB de transactions, où chaque transaction t_r identifiée de manière unique est un ensemble d'items tel que $t_r \subseteq I$. Un ensemble $X \subseteq I$ est aussi appelé un *k-itemset* et est représenté par $(x_1, x_2, \ldots x_k)$ avec $x_1 < x_2 < \ldots < x_k$. Le support d'un itemset X, noté $supp(X)$, correspond au nombre de transactions dans lesquelles l'itemset apparaît. Un itemset est appelé *fréquent* si $supp(X) \geq \sigma \times |DB|$ où $\sigma \in (0, 1)$ correspond au support minimal spécifié par l'utilisateur et $|DB|$ à la taille de la base de données. Le problème de la recherche des itemsets fréquents consiste à rechercher tous les itemsets dont le support est supérieur ou égal à $\sigma \times |DB|$ dans DB.

Les définitions précédentes considèrent que la base de données est statique. Considérons, à présent, que les données arrivent de manière séquentielle sous la forme d'un flot continu. Soit *data stream* $DS = B_0^1, B_1^2$, ..., B_{n-1}^n, ..., un ensemble infini de batches où chaque batch est estampillé selon un intervalle de temps $[t, t+1]$, i.e. B_t^{t+1}, et n est l'identifiant du dernier batch B_{n-1}^n. Chaque batch B_a^b correspond à un ensemble de transactions : $B_a^b = [T_1, T_2, T_3,$ ..., $T_k]$. Nous supposons également que les batches n'ont pas nécessairement la même taille. La longueur L du data stream est définie par $L = |B_0^1| + |B_1^2| + ... + |B_{n-1}^n|$ où $|B_a^b|$ correspond à la cardinalité de l'ensemble B_a^b.

B_0^1	T_a T_b	(1 2 3 4 5) (8 9)
B_1^2	T_c	(1 2)
B_2^3	T_d T_e	(1 2 3) (1 2 8 9)

FIG. 1 – *Les ensembles de batches B_0^1, B_1^2 et B_2^3*

Le support d'un itemset X à un instant donné t est défini comme étant le ratio du nombre de clients qui possèdent X dans leur fenêtre de temps sur le nombre total de clients. Ainsi, étant donné un support minimal spécifié par l'utilisateur, le problème de la recherche des itemsets dans un flot de données consiste à rechercher tous les itemsets fréquents X, i.e. vérifiant $\sum_{t=0}^{L} \text{support}_t(X) \geq \sigma \times L$, dans le flot.

Exemple 1 *La Figure 1 illustre l'ensemble de tous les batches. Supposons que le support minimal soit fixé à 50%. A partir des transactions de B_0^1, i.e. pour le premier intervalle de temps [0-1], nous pouvons extraire les deux itemsets maximaux suivants : (1 2 3 4 5) et (8 9). Si nous considérons, à présent, l'intervalle de temps [0-2], i.e. batches B_0^1 et B_1^2, l'ensemble des itemsets maximaux est réduit à : (1 2). Finalement, en considérant tous les batches, i.e. pour l'intervalle [0-3], nous obtenons l'ensemble d'itemsets suivant : (1 2), (1) et (2).*

3 Travaux liés

La première approche d'extraction d'itemsets dans des flots de données a été proposée par [9] avec un algorithme, en une passe, basé sur la propriété d'antimonotonie du support. Pour cela, les auteurs utilisent une structure à base d'arbres pour représenter les itemsets. L'autre originalité de cette approche est de proposer des résultats approximatifs en introduisant le concept de ϵ-*deficient function*. En effet, comme nous l'avons précisé en introduction, étant donné la quantité de données et la vitesse à laquelle celles-ci arrivent, il n'est plus possible d'offrir à l'utilisateur une réponse exacte. Une telle réponse, si elle était possible, nécessiterait d'une part un trop grand espace de stockage et d'autre part de parcourir très rapidement un grand nombre de données. Li et al. [8] proposent d'extraire les itemsets fréquents en partant des plus grands aux plus petits et utilisent une extension d'une représentation basée sur des arbres préfixés. Dans [4], les auteurs proposent d'utiliser une structure de type FP-tree [5] pour rechercher des itemsets fréquents à différents niveaux de granularité temporelle. Pour cela, ils mettent en place une nouvelle technique appelée *tilted-time windows table*. Cette notion a initialement été introduite dans [2] et est basée sur l'intuition suivante : les utilisateurs s'intéressent plus souvent aux récents changements avec une granularité fine et aux changements à long terme avec une granularité plus large.

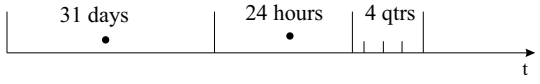


FIG. 2 – *Structure de Tilted-Time Window Naturelle*

La figure 2, extraite de [4], illustre une table de tilted-time windows : les 4 quarts d'heure plus récents, puis les dernières 24 heures et enfin 31 jours. A partir de ce modèle, nous pouvons représenter les informations apparues pendant la dernière heure avec une précision d'un quart d'heure, le dernier jour avec une précision d'une heure, etc. Dans le cas de notre problématique, nous pouvons associer à chaque itemset une tilted time windows table, il est possible de détecter et de connaître leurs supports avec les intervalles de temps associés. Dans [4], les

auteurs étendent ces fenêtre en *tilted-time windows table logarithmique* pour offrir une représentation temporelle plus compacte en espace mémoire tout en introduisant en contrepartie un mécanisme d'approximation : soient $\tau_0, \tau_1, \ldots, \tau_n$ les unités de temps déjà passées. Soit $n \geq b, a \geq 0$ et $b > a$, la fenêtre de temps T_a^b correspond à l'intervalle de temps $\tau_a, \ldots, \tau_b$. Une *table de fenêtres de temps logarithmique* (ou tilted-time windows table logarithmique) est une partition des unités temporelles passées. Chacune de ces partitions représente un niveau de granularité et consiste en une collection de fenêtres de temps consécutives. Chaque niveau contient au plus $max_L \geq 2$ fenêtres. Par exemple, le premier niveau de taille $max_L = 2$ dans une table de fenêtres de temps logarithmique est :

$$\underbrace{T_{a_n}^{b_n}, T_{a_n-1}^{b_n-1}}_{niveau0}, \ldots, T_{a_0}^{b_0}$$

La mise à jour des fenêtres temporelles au sein de la table se fait grâce a une opération de décalage et de compactage sur chacun des niveaux de granularités en commençant par la granularité la plus fine. Le processus s'arrête lorsqu'il n'est plus possible de décaler de nouvelles fenêtres.

Comme nous venons de le constater, l'extraction d'itemsets dans des flots est difficile car de nombreuses contraintes doivent être considérées. La notion de tilted-time windows table offre une solution efficace pour permettre de poser des requêtes sur n'importe quel intervalle de temps et en garantissant une réponse "approximative" la plus proche possible du réel, par égard à un seuil d'erreur fixé. Néanmoins, dans un contexte aussi dynamique, deux problèmes persistent :

(a) Comment retrouver efficacement les itemsets fréquents précédents afin de mettre à jour leur tilted-time windows ? Idéalement, nous souhaiterions éviter de naviguer dans tous les itemsets stockés ou, en d'autres termes, nous souhaiterions réduire l'espace de recherche aux seuls itemsets "intéressants". Dans [10], nous avons proposé une solution basée sur une notion de région qui s'avère bien adaptée pour les motifs séquentiels. Dans ce cadre, la question devient : est-ce qu'une solution basée sur des régions est adaptée aux itemsets ?

(b) Comment vérifier efficacement si un itemset est un sous ensemble d'un autre itemset ? Plus précisément, pouvons nous trouver une nouvelle représentation pour les itemsets nous permettant de vérifier l'inclusion de manière efficace ?

Nous répondons à ces différentes questions dans la section suivante.

4 L'approche FIDS

Dans cette section, nous proposons l'approche FIDS pour extraire des itemsets dans des flots de données. Tout d'abord nous présentons un survol de l'approche. Nous examinons ensuite une nouvelle représentation des itemsets. Finalement, nous décrivons les algorithmes.

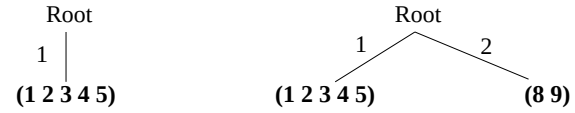
4.1 Aperçu de l'approche

De manière à illustrer le fonctionnement de FIDS, reprenons l'exemple de la Figure 1. Tout d'abord, nous traitons la première transaction T_a dans B_0^1 en stockant T_a dans un treillis

Items	Tilted-T W.	Reg.	Root$_{Reg}$
1	$[t_0^1 = 1]$	1	(1 2 3 4 5)
2	$[t_0^1 = 1]$	1	(1 2 3 4 5)
3	$[t_0^1 = 1]$	1	(1 2 3 4 5)
4	$[t_0^1 = 1]$	1	(1 2 3 4 5)
5	$[t_0^1 = 1]$	1	(1 2 3 4 5)

Itemsets	Taille	Tilted-T W.
$(1\ 2\ 3\ 4\ 5)$	5	$[t_0^1 = 1]$

FIG. 3 – *Les items (gauche) et les itemsets (droite) après la transaction* T_a



(Batch 1 − Ta) *(Batch 2 − Tb)*

FIG. 4 – $LatticeReg$ *après le premier batch*

($LatticeReg$). Ce treillis possède les caractéristiques suivantes : chaque chemin dans le treillis est apparenté à une *région*, les itemsets correspondent aux nœuds et les itemsets stockés sont ordonnées en respectant la propriété d'inclusion. Par construction, tous les sous-ensembles d'un itemset appartiennent à la même région. Ce principe de région est utilisé afin réduire l'espace de recherche lors des étapes de comparaison et d'élagage.

Nous ne stockons que les "itemsets maximaux" dans $LatticeReg$. Ces derniers sont en fait : soit les itemsets, i.e. les transactions, directement extraits du batch, soit leurs sous-ensembles maximaux. Un *sous-ensemble maximal*, dans notre contexte, correspond au plus grand sous-ensemble de la transaction qui peut être construit à partir des items appartenant à une même région. En ne stockant que ces itemsets, notre but est de stocker le minimum d'information pour pouvoir répondre aux requêtes des utilisateurs. A la fin de l'analyse de T_a, nous disposons donc d'un ensemble d'items {1,2,3,4,5}, d'un itemset **(1 2 3 4 5)** et de $LatticeReg$ mis à jour. Les items sont stockés comme illustré Figure 3 (gauche). L'attribut "*Tilted-T W*" correspond au nombre d'occurrences de l'item dans le batch dans l'intervalle de temps considéré. L'attribut "$Root_{Reg}$" est un pointeur vers la racine de région dans $LatticeReg$. Une *racine de région* correspond à l'itemset maximal pour une région donnée. Remarquons que pour une région, il ne peut exister qu'une racine et qu'un item peut apparaître dans plusieurs régions. L'attribut *Val.* correspond à un entier identifiant la région. Pour les itemsets (voir Figure 3 (droite)), nous stockons à la fois les tailles et les tilted-time windows tables associées. Cette dernière information sera utilisée lors de l'étape d'élagage. La partie gauche de la Figure 4 illustre comment le treillis $LatticeReg$ est mis à jour lorsqu'on considère T_a.

Examinons à présent la seconde transaction T_b du batch B_0^1. Comme T_b n'est pas incluse dans T_a, elle est directement insérée dans $LatticeReg$ dans une nouvelle région (voir sous-arbre $(8\ 9)$ dans la Figure 4).

Le batch B_1^2 est simplement réduit à la transaction T_c. T_c étant un itemset maximal nous devons l'inclure dans $LatticeReg$ (C.f. Figure 6). Cependant, l'itemset $(1\ 2)$ est inclus dans

Itemsets	Taille	Tilted-T W.
$(1\ 2\ 3\ 4\ 5)$	5	$[t_0^1 = 1]$
$(8\ 9)$	2	$[t_0^1 = 1]$
$(1\ 2)$	2	$[t_0^1 = 1]$, $[t_1^2 = 1]$

Itemsets	Taille	Tilted-T W.
$(1\ 2\ 3\ 4\ 5)$	5	$[t_0^1, 1]$
$(8\ 9)$	2	$[t_0^1 = 1]$
$(1\ 2)$	2	$[t_0^1 = 1]$, $[t_1^2 = 1]$, $[t_2 = 1]$
$(1\ 2\ 3)$	3	$[t_2^3 = 1]$

FIG. 5 – *Les itemsets mis à jour après B_1^2 (gauche) et après T_d de B_2^3 (droite)*

$(1\ 2\ 3\ 4\ 5)$, i.e. T_a, ce qui veut dire que quand T_a est intervenu dans les batches précédent, il en est de même pour T_c. Les tilted-time windows tables associées doivent donc être mises à jour (voir Figure 5 (gauche)).

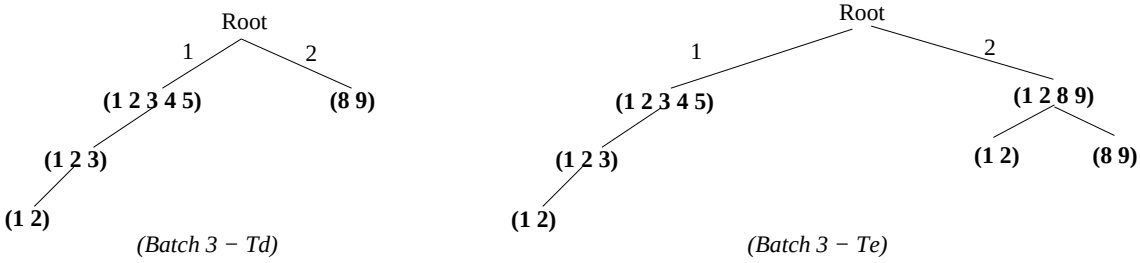


FIG. 6 – *Le treillis des régions après le dernier batch*

La transaction T_d est considérée de la même manière que T_c. Les modifications sur les structures sont décrites Figure 6 et Figure 5 (droite). Considérons à présent la transaction T_e. Cet itemset n'est pas inclus dans l'ensemble de tous les itemsets. Nous remarquons que les items 1 et 2 sont dans la région de valeur 1 alors que 8 et 9 appartiennent à la région de valeur 2. Nous remarquons également que l'itemset $(8\ 9)$ existe déjà et est inclus dans T_e. Nous incluons donc T_e comme racine de région pour la valeur 2.

Examinons à présent comment les itemsets stockés peuvent être effacés en fonction d'une valeur de support minimale spécifiée par l'utilisateur. En prenant une valeur de σ =50%, pour les transactions incluses dans le batch B_0^1 un itemset est fréquent si nous avons : $support_{B_0^1}(X) \geq \sigma|B_0^1|$. Pour le batch B_1^2, en examinant l'ensemble de tous les itemsets, nous trouvons que seul **(1 2)** vérifie la contrainte de support minimal. Tous les autres itemsets non fréquents sont élagués du treillis en navigant dans $LatticeReg$ en fonction des régions des items.

Ce processus est répété après chaque nouveau batch de manière à n'utiliser que le minimum de mémoire.

4.2 Une représentation efficace des items

Comme nous l'avons précisé dans la section concernant les travaux liés, l'un des problèmes important est de pouvoir calculer rapidement l'inclusion entre deux itemsets. Cette opération coûteuse peut toutefois être améliorée avec une nouvelle représentation des items dans les transactions. Ansi, nous représentons les items à l'aide de nombres premiers (C.f. Figure 7).

Items	**1**	**2**	**3**	**4**	**5**	...	**8**	**9**	...
Nombres premiers	2	3	5	7	11	...	19	23	...

FIG. 7 – *Transformation en nombres premiers*

Remarquons qu'une représentation similaire mais dans un autre contexte a également été proposée de manière indépendante dans [11]. Chaque itemset peut maintenant être représenté par le produit de tous les nombres premiers correspondant aux items inclus. Par définition, le produit de nombres premiers étant décomposable d'une manière unique, nous pouvons rapidement vérifier l'inclusion de deux itemsets (e.g. $X \preceq Y$) en réalisant une opération de modulo sur chacun des itemsets ($Y \ mod \ X$). Si le reste est 0, nous savons que $X \preceq Y$, autrement X n'est pas inclus dans Y. Par exemple, Figure 8, $T_c \prec T_a$, puisque $T_a \ mod \ T_c = 0$.

T_a	(1 2 3 4 5)	$2 \times 3 \times 5 \times 7 \times 11$	2310
T_b	(8 9)	19×23	437
T_c	(1 2)	2×3	6
T_d	(1 2 3)	$2 \times 3 \times 5$	30
T_e	(1 2 8 9)	$2 \times 3 \times 19 \times 23$	2622

FIG. 8 – *Transactions transformées*

4.3 L'algorithme FIDS

Algorithm 1: L'algorithme FIDS

Data: Un ensemble infini de batches B=B_0^1, B_1^2, ... B_{n-1}^n ... ; σ le support minimum défini par l'utilisateur ; ϵ le seuil d'erreur maximal défini par l'utilisateur.

Result: L'ensemble des itemsets maximaux sur le flot.

$REGION \leftarrow \emptyset$; $ITEMS \leftarrow \emptyset$; $ISETS \leftarrow \emptyset$;$region \leftarrow 1$;

repeat

  foreach $B_a^b \in B$ **do**

    UPDATE(B_a^b, $ITEMS$, $ISETS$);

    PRUNE($ITEMS$, $ISETS$, σ, ε);

until *plus de batches*;

Tant que des batches sont disponibles, nous examinons les itemsets pour mettre à jour nos structures (UPDATE). Nous élaguons alors les itemsets non fréquents pour conserver les structures en mémoire (PRUNE). Par la suite, nous considérons que nous disposons de trois structures. Chaque valeur d'$ITEMS$ est un tuple ($labelitem$, $\{time, occ\}$, $\{regions, Root_{Reg}\}$) où $labelitem$ est l'identifiant de l'item, $\{time, occ\}$ est utilisé pour stocker le nombre d'occurrences de l'item au cours des différents batches, et pour chaque région dans $\{regions\}$ nous stockons l'itemset racine de région ($Root_{Reg}$) de la structure $LatticeReg$. La structure

$ISETS$ est utilisée pour stocker les itemsets. Chaque valeur de $ISETS$ est un tuple ($itemset$, $size(itemset)$, $\{time, occ\}$). Finalement, la structure $LatticeReg$ est un treillis où chaque nœud est un itemset stocké dans $ISETS$ et où les arcs correspondent aux régions associées.

Algorithm 2: L'algorithme UPDATE

Data: Un batch $B_a^b = [T_1, T_2, T_3, \ldots, \$T_k]$;
Result: $ITEMS$ et $ISETS$ mis-à-jour.
foreach $transaction\ T\ \in B_a^b$ **do**
  $LatticeMerge \leftarrow \emptyset$; $DelayedInsert \leftarrow \emptyset$; $Candidates \leftarrow$ GETREGIONS(T);
  if $Candidates = \emptyset$ **then**
    INSERT(T,Nouvelle région) ;
  else
    foreach $region\ reg\ \in Candidates$ **do**
      $FirstItemset \leftarrow$ GETROOT$_{Reg}$(Val) ; // On récupère la racine de la région candidate à l'insertion du nouvel itemset
      $NewIts \leftarrow$ GCD(Seq,$FirstItemset$) ; // On calcule l'itemset maximal commun (avec notre représentation, un simple gcd)
      if $(NewIts == T)||(NewIts == FirstItemset)$ **then**
        $LatticeMerge \leftarrow Val$;
      else
        // Un nouvel itemset doit être introduit
        INSERT($NewIts, reg$) ; UPDATETTW($NewIts$) ;
        $DelayedInsert \leftarrow NewIts$;
    // Si l'itemset à introduire ne peut pas être introduit dans une région
    //déjà existante, on en crée une nouvelle
    if $|LatticeMerge| = 0$ **then**
      INSERT(T,Nouvelle région) ; UPDATETTW(T);
    else
      if $|LatticeMerge| = 1$ **then**
        INSERT($T, LatticeMerge[0]$) ; UPDATETTW(T);
      else
        // Un itemset maximal va fusionner 2 ou plusieurs régions
        MERGE($LatticeMerge, T$);

Examinons à présent l'algorithme Update (voir Algorithme 2). Nous considérons chaque transaction T incluse dans le batch et cherchons tout d'abord les régions où tous ses items apparaissent(GETREGIONS). Si les items n'ont pas encore été examinés nous insérons la transaction T dans $LatticeReg$ avec une nouvelle région. Autrement, nous examinons l'ensemble des différentes valeurs de région associées aux items de T. Pour chaque région la fonction GETROOT retourne la racine de région, $FirstItemset$, associée à $Root_{Reg}$, i.e. l'itemset maximal de la région reg, qui sera utilisé dans les étapes suivantes. Comme nous représen-

tons les items par des nombres premiers, la recherche des itemsets maximaux est simplement réalisée par le calcul du plus grand commun dénominateur (PGCD) entre T et $FirstItemset$ (fonction GCD). Cette fonction retourne l'ensemble vide lorsqu'il n'y a pas d'itemsets maximaux ou si les itemsets sont réduits à un seul item. S'il n'existe qu'un seul itemset, i.e. la cardinalité de NewIts est 1, nous savons que celui-ci est soit racine de région, soit la transaction T elle même. Nous la stockons alors dans un tableau temporaire (*LatticeMerge*) utilisé par la suite pour éviter de créer une nouvelle région inutile. Autrement, nous savons que nous avons un nouvel itemset que nous insérons dans $LatticeReg$ (INSERT) et nous effectuons les mises à jour dans les tilted-time windows tables associées (UPDATETTW). Les itemsets sont également stockées dans un tableau temporaire (*DelayedInsert*). S'il existe plus d'un seul sous-itemsets (venant de GCD), nous les insérons dans les branches de $LatticeReg$ correspondantes aux régions concernées. Nous les stockons également avec T dans *DelayedInsert* de manière à retarder leur insertion en tant que nouvelle région. Si *LatticeMerge* est vide, nous savons qu'il n'existe aucun itemset de T qui soit inclus dans des itemsets de $LatticeReg$ et nous pouvons donc inclure T dans $LatticeReg$ dans une nouvelle région. Si la cardinalité de $LatticeMerge$ est supérieure à 1, nous avons forcément un itemset qui sera une nouvelle racine de région et l'insérons.

Par manque de place, nous ne détaillons pas la fonction PRUNE. Son principe est le suivant. Tout d'abord les tilted-time windows des itemsets sont mis à jour en utilisant la même approche que [4]. Les itemsets ne vérifiant pas la contrainte de support sont stockés dans un tableau temporaire ($ToPruned$). Nous considérons alors les items de $ITEMS$. Si un item n'est plus fréquent nous naviguons dans $LatticeReg$ pour supprimer cet item des itemsets et pour supprimer les itemsets qui apparaissent dans $ToPruned$. Cette fonction tire avantage de la propriété d'antimonotonie ainsi que de l'ordre des itemsets dans $LatticeReg$.

Par manque de place, nous ne présentons pas la complexité de l'algorithme FIDS. Toutefois, dans [10], nous calculons la complexité d'une approche basée sur des régions pour rechercher des motifs séquentiels.

5 Experimentations

Les flots de données utilisés sont générés à partir de ceux du ECML/PKDD Challenge 2005. A partir des données initiales pré-traitées, nous avons considéré une durée de batches de 30 secondes et un taux d'erreur de 0.1 (soit 10% d'erreur), les batches contenant à peu près 5000 itemsets. Toutes ces transactions ont été considérées comme entrée standard pour notre programme. Ce dernier a été développé en Java. Toutes les expérimentations ont été réalisées sur un Pentium 3 (1200Mhz) fonctionnant sous Linux avec 512 Mb de RAM.

A chaque traitement des batches, les informations suivantes ont été analysées : la taille de la structure $LatticeReg$ en Bytes et le temps de traitement pour tous les itemsets. L'axe des x représente le numéro du batch. Les Figures 9 illustrent le temps de traitement pour les itemsets. Tous les deux batches ($max_L = 2$ dans notre implémentation), l'algorithme a besoin d'un temps supplémentaire pour calculer les itemsets. Ce temps est en fait lié a l'opération de "compactage" des tilted-time windows tables, réalisé dans nos expérimentations tous les deux batches, dans la structure $LatticeReg$. Deux conditions se posent pour l'extraction d'itemsets sur les flots :

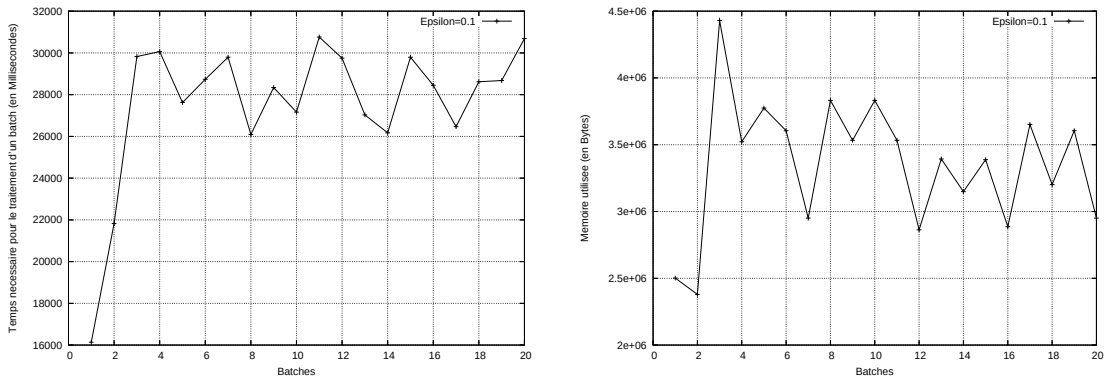


FIG. 9 – *Temps de traitement et mémoire nécessaire pour Fids*

1. Borner la mémoire utilisée. Cette condition est respectée avec cette implémentation de FIDS, puisque la mémoire nécessaire pour la gestion des structures ne dépasse pas les 4,5Mo.
2. Borner le temps de calcul entre chaque batch. Cette condition est aussi respectée, puisque pour chaque étape de calcul, l'implémentation actuelle de FIDS ne dépasse pas les 30 secondes (qui est aussi la taille du batch)

6 Conclusion

Dans cet article, nous avons abordé la problématique de la recherche d'itemsets dans des flots de données. Nos contributions principales sont les suivantes. Premièrement, en utilisant des nombres premiers pour représenter les items du flots nous améliorons la vérification des inclusions d'itemsets et donc améliorons le processus global (cette représentation nécessite quand même de borner le nombre d'items possibles). Deuxièmement, nous avons montré qu'une nouvelle structure de représentation basée sur la notion de région était non seulement adaptée à la recherche de motifs [10] mais également aux itemsets car elle permet de rapidement trouver les itemsets stockés soit pour en extraire les itemsets inclus soit pour élaguer la structure, ce qui évite, grâce à l'indexage, de parcourir des branches ou des parties entières du treillis contenant les itemsets maximaux. Finalement, en stockant seulement un nombre minimal d'itemsets (i.e. les itemsets maximaux) couplée avec la notion de tilted time windows table nous pouvons fournir une réponse (avec un taux d'erreur borné) à l'utilisateur tout en respectant les contraintes de support et de temps fixées. Avec FIDS, l'utilisateur final peut, à tout moment, connaître les itemsets fréquents sur un intervalle de temps ou détecter les changements de fréquences.

Références

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors,

SIGMOD Conference, pages 207–216. ACM Press, 1993.

[2] Yixin Chen, Guozhu Dong, Jiawei Han, Benjamin W. Wah, and Jianyong Wang. Multi-dimensional regression analysis of time-series data streams. In *VLDB*, pages 323–334, 2002.

[3] Y. Chi, H. Wang, P.S. Yu, and R.R. Muntz. Moment : Maintaining closed frequent itemsets over a stream sliding window. In *Proc. of ICDM'04 Conference*, 2004.

[4] C. Giannella, J. Han, J. Pei, X. Yan, and P. Yu. Mining frequent patterns in data streams at multiple time granularities, 2002.

[5] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Mei-chun Hsu. Freespan : frequent pattern-projected sequential pattern mining. In *KDD*, pages 355–359, 2000.

[6] C. Jin, W. Qian, C. Sha, J.-X. Yu, and A. Zhou. Dynamically maintaining frequent items over a data stream. In *Proc. of CIKM'04 Conference*, 2003.

[7] R.-M. Karp, S. Shenker, and C.-H. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems*, 28(1) :51–55, 2003.

[8] H.-F. Li, S.Y. Lee, and M.-K. Shan. An efficient algorithm for mining frequent itemsets over the entire history of data streams. In *Proc. of the 1st Intl. Workshop on Knowledge Discovery in Data Streams*, 2004.

[9] G. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proc. of VLDB'02 Conference*, 2002.

[10] C. Raissi, P. Poncelet, and M. Teisseire. Need for speed : Mining sequential patterns in data streams. In *Proc. of the Bases de Données Avancées Conference (BDA 2005)*, 2005.

[11] S.N. Sivanandam, D. Sumathi, T. Hamsapriya, and K. Babu. Parallel buddy prima - a hybrid parallel frequent itemset mining algorithm for very large databases. In *www.acadjournal.com, Vol.13*, 2004.

[12] W.-G. Teng, M.-S. Chen, and P.S. Yu. A regression-based temporal patterns mining schema for data streams. In *Proc. of VLDB'03 Conference*, 2003.