EGC 2009

9èmes Journées Francophones
Extraction et Gestion de Connaissances

Atelier Fouille de données temporelles
Analyse de flux de données

Organisation : Georges Hébrail, Pascal Poncelet, René Quiniou

Fouille de données temporelles – Analyse de flux de données

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. Les données représentent, par exemple, les valeurs prises par des variables mesurées à intervalles réguliers ou des événements se produisant de manière irrégulière. Les données présentent généralement un caractère temporel qu'il est important 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 suivies étendent les approches classiques de la fouille de données pour prendre en compte la dimension temporelle ou proposent de nouvelles solutions et algorithmes appropriés aux données temporelles. De plus, elles doivent tenir compte de la complexité des algorithmes utilisés et de leur capacité à "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.

Organisateurs

- Georges Hébrail (ENST Paris) hebrail@enst.fr
- Pascal Poncelet (Ecole des Mines d'Alès) Pascal.Poncelet@lirmm.fr
- René Quiniou (IRISA/INRIA Rennes) Rene.Quiniou@irisa.fr

Comité de programme

- Fabrice Clérot (France Telecom R&D, Lannion)
- Michel Dojat (Unité mixte INSERM-UJF U594, Grenoble)
- Alain Dessertaine (EDF R&D, Clamart)
- Joao Gama (Université de Porto, Portugal)
- Catherine Garbay (Laboratoire d'Informatique de Grenoble)
- Thomas Guyet (INRIA Rennes)
- Georges Hébrail (ENST Paris)
- Anne Laurent (LIRMM Montpellier)
- Yves Lechevallier (INRIA Rocquencourt)
- Pierre-François Marteau (Université de Bretagne Sud Vannes)
- Florent Masseglia (INRIA Sophia Antipolis)
- Pascal Poncelet (Ecole des Mines d'Alès)
- René Quiniou (INRIA Rennes)
- Fabrice Rossi (INRIA Rocquencourt)

Detection of Change–Points in the Spectral Density. With Applications to ECG Data

Pierre R. Bertrand, Gilles Teyssière, Gil Boudet, Alain Chamoux

INRIA Saclay, APIS Team
UMR CNRS 6620, Clermont-Ferrand France.
E-mail: Pierre.Bertrand@inria.fr,
,CREATES, Aarhus University
E-mail: stats@gillesteyssiere.net
,Institut de Médecine du travail, UFR Médecine, Univ. Clermont 1
Gil.Boudet@wanadoo.fr
,Institut de Médecine du travail, UFR Médecine, Univ. Clermont 1 ;
CHU Clermont-Ferrand
E-mail: alain.chamoux@u-clermont1.fr

Résumé. Nous proposons une nouvelle méthode pour estimer les ruptures du rythme cardiaque dans les bandes parasympathiques et orthosympathiques basée sur la transformée en ondelettes dans le domaine complexe et l'étude des ruptures des moments des modules de ces transformées en ondelettes. Nous observons des ruptures dans la distribution pour les deux bandes de fréquence.

1 Introduction

ECG signal analysis has a long story after the implementation of the ambulatory monitoring by Holter at the beginning of the fifties. Recent measurement methods, due to the size reduction of the measurement devices, see Chamoux (1984), allow us to record ECG data for healthy people over a long period of time : long distance (marathon) runners, individuals daily (24 hours) records, etc. We then obtain large datasets that allow us to characterize the variations of the heartbeat rate in the two components of the nervous autonomous system : the parasympathetic and the orthosympathetic ones.

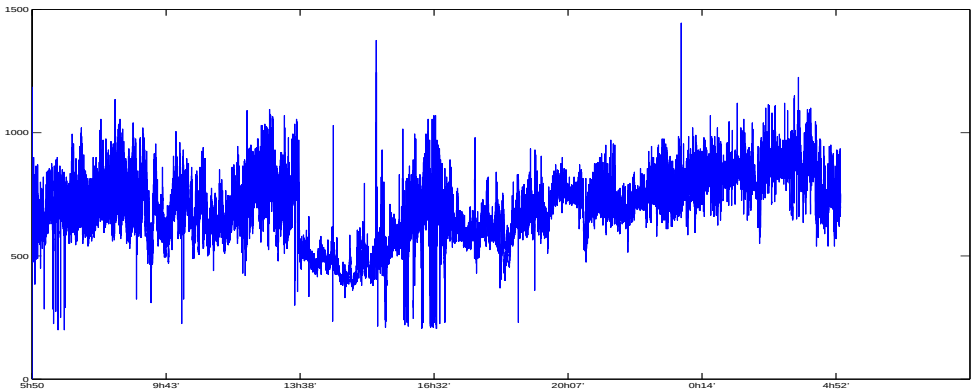


FIG. 1 – *RR interval for a healthy subject during a period of 24 hours*

The data studied in this paper have been recorded the 29^{th} of January 2008, at the Clermont-Ferrand University Hospital (CHU) by G. Boudet. Unlike Boudet *et al.* (2004) and Cottin *et al* (2006), these data are not recorded in the framework of a laboratory experiment, but during the real life of the individual under investigation.

We have daily datasets, over 100,000 observations, and in the near future we will have data over several days, i.e., over that sample size. We are measuring the interval between two RR peaks, i.e., if we denote by $(t_i)_{i=1,\dots,N}$, the sequence of peak times measured with a precision of 1.E-03 second, we consider the series $X(t_i) = \left(t_i - t_{i-1}\right)$ measured in seconds. Several indicators are related to heartbeat data. The most popular is the instantaneous average frequency, i.e., $X(t_i)^{-1}$, which is displayed by runners watches. This quantity is informative on daily activity of observed individual : sleeping and waking up times, physical activity, e.g., sport, manual work, but does not summarize all relevant information. Note that heartbeat data display large variations, clustering, etc, as only individuals with serious disease display a regular heart rate.

Cardiologists are interested in the study of this signal in two frequency bands : the orthosympathetic and parasympathetic bands, i.e., the frequency bands $(0.04\,Hz,\ 0.15\,Hz)$ and $(0.15\,Hz,\ 0.5\,Hz)$ respectively. The definition of these bands is the outcome of research works, see e.g., Task force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996), and is based on the fact that the energy contained inside these bands would be a relevant indicator on the level of the stress of an individual.

Indeed, for the heart rate, the parasympathetic system is often compared to the brake while the orthosympathetic system would be a nice accelerator ; see e.g. Goldberger (2001). At rest there is a permanent braking effect on the heart rate. Any solicitation of the cardiovascular system, any activity initially produces a reduction of parasympathetic brake followed by a gradual involvement of the sympathetic system. These mechanisms are very interesting to watch in many diseases including heart failure, but also rhythm disorders that may fall under one or other of these two effects, monitoring the therapeutic effect of several Medicines including some psychotropic. In the field of physiology such data are crucial for measuring the level of vigilance and particularly the risk of falling asleep while driving a vehicle, the level of stress induced by physical activity or level of perceived stress, which can be considered as a criterion of overtraining in sport.

Fractals models have been used in cardiology after the works by Ivanov *et al.* (1999), who applied the multifractal spectrum analysis advocated by Frisch (1996), for modeling RR series and classifying individuals according to this multifractal spectrum, as this spectrum discriminates between individuals who experienced hearth trouble, and those who did not. However, this tool has some shortcomings as it requires huge samples common in turbulence analysis, and is then inappropriate for studying phenomena occurring at a resolution lower than the daily time interval, such as the variations of the parasympathetic and orthosympathetic systems inside the day.

Wavelets based methods have been used in biostatistics by Diab *et al.* (2007) for uterine EMG signal analysis. However, they consider that the process studied is homogeneous, and used these methods as a classification tool. Another significant difference is the fact that they use discrete wavelet decomposition, i.e., a frequency decomposition on a dyadic wavelet basis, the choice for the frequency bands is made without reference to a biological phenomenon. In our case, the choice for the frequency band is justified by biological considerations, and we fit the wavelets inside these bands.

This is why the continuous analysis of both systems and their quantification is a particularly promising research area. So, the example in Figures 4, 5, 6 shows at the observation 28,220 i.e., 13h40'50", a simultaneous variation of both systems.

2 Mathematical formulation

For biological reasons, the heart rate is within the interval $[20, 250]$ bpm (beats per minute), i.e., $X(t)$ belongs to the RR interval $60/250s. < X(t) < 60/20s.$. This leads to modeling by stationary or locally stationary processes. We wish to decompose this signal in a sequence of homogeneous intervals having the same mean and/or variance. We make the assumption that the series is Gaussian, and then find the optimal segmentation of the process using the same approach as in Lavielle and Teyssière (2006).

We assume that the signal is the sum of a piecewise constant function and a Gaussian process, centered and locally stationary. We then have the following representation :

$$X(t) = \mu(t) + \int_{I\!R} e^{it\xi} f^{1/2}(t,\xi)\, dW(\xi), \quad \text{for all} \quad t \in R, \tag{1}$$

where

- $x \mapsto f(t,x)$ is an even and positive function, called spectral density piecewise constant, i.e., there exists a partition $\tau_1, \ldots, \tau_K$ such that $f(t,x) = f_k(x)$ for $t \in [\tau_i, \tau_{i+1}[$
- the function $t \mapsto \mu(t)$ is also piecewise constant for another partition $\widetilde{\tau}_1, \ldots, \widetilde{\tau}_L$ with $\mu(t) = \mu_\ell$ if $t \in [\widetilde{\tau}_\ell, \widetilde{\tau}_{\ell+1}[$

The wavelet coefficient associated with ψ is

$$W_\psi(b) = \int_{I\!R} \psi(t-b) X(t)\, dt \qquad \text{unit in } second^2. \tag{2}$$

In Bardet and Bertrand (2007) or Bardet, Bertrand and Billat (2008), one can find a theoretical study of the wavelet coefficient for stationary (or with stationary increment) centered Gaussian processes, i.e., for X given by (1) with $\mu(t) = 0$. This approach can be generalized to locally stationary Gaussian processes, this will be detailed in a forthcoming paper.

According to recommendations of Task force of the European Soc. Cardiology and the North American Soc. of Pacing and Electrophysiology (1996), we use the following notations :

- $[\omega_1, \omega_2] = (0.04\, Hz,\ 0.15\, Hz)$ denotes the orthosympathetic frequency band,
- $[\omega_2, \omega_3] = (0.15\, Hz,\ 0.5\, Hz)$ denotes the parasympathetic band

The energy associated with each of these frequency bands and localized around the time b, is measured by the modulus of the complex wavelet coefficients $|W_i(b)|^2$ for $i = 1, 2$, with

$$b \longmapsto W_i(b) \quad = \quad \int_{I\!R} \psi_i(t-b)\, X(t)\, dt, \quad i = 1, 2.$$

In this work, these wavelets coefficients are computed at each second, i.e., the difference between two consecutive values for b is equal to 1 second.

How to choose the wavelets ψ_1 and ψ_2 ?

In the idealistic case, one would use two filters ψ_1 and ψ_2 with compact support, the Fourier transforms of which have support inside the orthosympathetic and parasympathetic bands.

Unfortunately, it does not exist a non null function ψ with compact time domain support and compact frequency support, see for instance Mallat (1998) Th 2.6 p.34. Therefore, the best we can do is to choose between a filter with a compact frequency support and a filter with a compact time domain support. The first choice is well suited for stationary models, see Bardet and Bertrand (2007). But, in this work, we are interested by locally stationary models, thus our specifications are a filter with compact time domain support as a Daubechies wavelet. The price to pay for the compactness of the time domain support is the loss in the compactness of the frequency support. However, the frequency support is *"almost compact"* in the following sense :

Definition 2.1 (ρ pseudo support) *Let $0 < \rho < 1$, be a map $g \in L^2(I\!R)$ that admits the compact interval I as a ρ pseudo support if* $\dfrac{\int_I |g(t)|^2\, dt}{\int_{I\!R} |g(t)|^2\, dt} = \rho.$

Fourier transform of Daubechies wavelet have a reasonably small ρ pseudo support with a ratio ρ close to 1. Moreover, the larger the number of the Daubechies wavelet is, the closer to 1 the ratio ρ is ; see the example below with the Daubechies wavelet D6.

By scaling and modulation, one can adjust the pseudo support inside a specified frequency band as stated by the following proposition

Proposition 2.1 *Let ψ be a filter with compact support $[L_1, L_2]$ and a frequency ρ pseudo support $[\Lambda_1, \Lambda_2]$, for any frequencies band $[\omega_1, \omega_2]$, the map $\psi_1(t) = \mu \times e^{i\eta}\, \psi(\lambda t)$ with*

$$\mu > 0, \qquad \lambda = \frac{\omega_2 - \omega_1}{\Lambda_2 - \Lambda_1} \quad \text{and} \quad \eta = \frac{\omega_1 + \omega_2}{2} - (\omega_2 - \omega_1)\frac{\Lambda_2 + \Lambda_1}{\Lambda_2 - \Lambda_1}$$

has a ρ pseudo support $[\omega_1, \omega_2]$ and a time domain support $\left[\frac{\Lambda_2 - \Lambda_1}{\omega_2 - \omega_1} L_1, \frac{\Lambda_2 - \Lambda_1}{\omega_2 - \omega_1} L_2\right]$.

Proof. Since $\widehat{\psi_1}(\xi) = \mu \times \widehat{\psi}\left(\frac{\xi - \eta}{\lambda}\right)$, one can deduce $\rho\ pseudo\, supp\, \psi_1 = \eta + \lambda \times \rho\ pseudo\ supp\, \psi$ and then the proposition. ■

Different choices for the wavelets ψ_1 and ψ_2

From Proposition 2.1, one can deduce the different possible choices of the filters ψ_1 and ψ_2

• Daubechies wavelet D6 : In this case, we have $\Lambda_1 = 0.08$, $\Lambda_2 = 1.75$, $\rho = 0.9999$ and one can set

$$\psi_1(t) = \mu \times e^{i\eta_1}\, D_6(\lambda_1 t) \quad \text{and} \quad \psi_2(t) = \mu \times e^{i\eta_2}\, D_6(\lambda_2 t)$$
$$\text{with} \quad \eta_1 = -0.0255, \lambda_1 = 0.0659, \eta_2 = -0.0585, \text{ and } \lambda_2 = 0.2096.$$

The length of the time support are $|Supp\, \psi_1|$ and $|Supp\, \psi_2|$. One can see on Fig. 2 (a) that the Fourier transforms $\widehat{\psi}_1(x)$ and $\widehat{\psi}_2(x)$ have almost disjoint supports.

$$\lambda_1 = \frac{\omega_2 - \omega_1}{\Lambda_2 - \Lambda_1} \quad \text{and} \quad \eta_1 = \frac{\omega_1 + \omega_2}{2} - (\omega_2 - \omega_1)\frac{\Lambda_2 + \Lambda_1}{\Lambda_2 - \Lambda_1}$$
$$\lambda_2 = \frac{\omega_3 - \omega_2}{\Lambda_2 - \Lambda_1} \quad \text{and} \quad \eta_2 = \frac{\omega_2 + \omega_3}{2} - (\omega_3 - \omega_2)\frac{\Lambda_2 + \Lambda_1}{\Lambda_2 - \Lambda_1}$$
$$|Supp\, \psi_1| = \frac{\Lambda_2 - \Lambda_1}{\omega_2 - \omega_1} \times |Supp\, D_6| \quad \text{and} \quad |Supp\, \psi_2| = \frac{\Lambda_2 - \Lambda_1}{\omega_3 - \omega_2} \times |Supp\, D_6|$$

• Gabor wavelet : For computational reasons, we will use the Gabor wavelet defined as

$$\psi(t) = e^{i\eta t} g(t), \quad g(t) = \frac{1}{(\sigma^2\pi)^{1/4}}\, e^{-\frac{t^2}{2\sigma^2}} \tag{3}$$

see, e.g., Mallat (1998). This wavelet has the same time and frequency ρ pseudo support $[-L, L] = [-3.5, 3.5]$ with $\rho = 0.9995$. In the spectral domain, we have

$$\widehat{\psi}(t) = \hat{g}(\xi - \eta), \quad \hat{g}(\xi) = (4\pi\sigma^2)^{1/4} e^{\frac{-\sigma^2\xi^2}{2}} \tag{4}$$

We fit the Gabor wavelet inside the orthosympathetic or the parasympathetic frequency bands by using Prop. 2.1 or direct calculations. One obtains the two Gabor wavelets defined by (3) with the following choice for the parameters :

$$\eta_1 = \frac{\omega_1 + \omega_2}{2} \quad \text{and} \quad \sigma_1 = \frac{2L}{\omega_2 - \omega_1}$$
$$\eta_2 = \frac{\omega_2 + \omega_3}{2} \quad \text{and} \quad \sigma_2 = \frac{2L}{\omega_3 - \omega_2}$$
$$\text{moreover} \qquad |\rho\, pseudo\, Supp\, \psi_1| = \frac{4L^2}{\omega_2 - \omega_1} \quad \text{and} \quad |\rho\, pseudo\, Supp\, \psi_2| = \frac{4L^2}{\omega_3 - \omega_2} \quad \text{with } \rho = 0.9995$$

One can see on Fig. 2 that the Fourier transforms $\widehat{\psi}_1(x)$ and $\widehat{\psi}_2(x)$ still have almost disjoint supports.

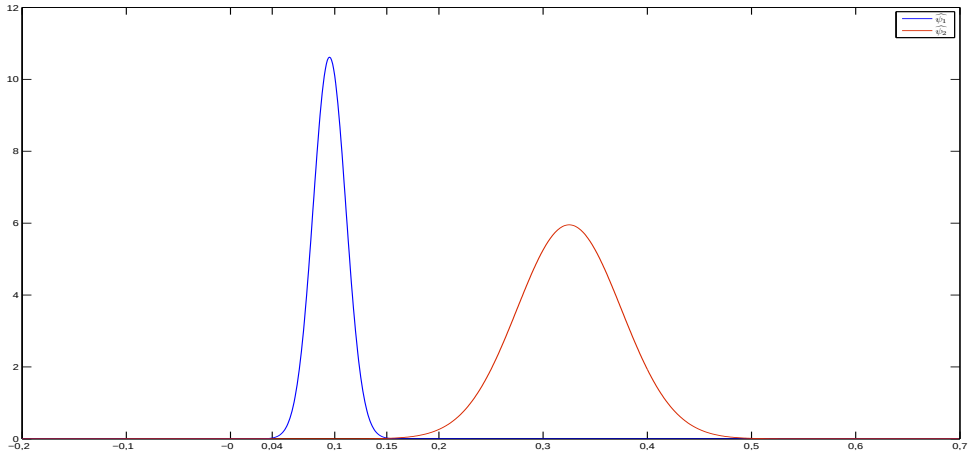


FIG. 2 – *The Fourier transforms* $\widehat{\psi}_1(x)$ *(left) and* $\widehat{\psi}_2(x)$ *(right)*

Using the Gabor wavelet is more efficient in terms of computing time, as it is at least 8 times faster. Figure 3 below displays the Gabor wavelets coefficients in the orthosympathetic and parasympathetic bands respectively for the sample plotted in Figure 1.

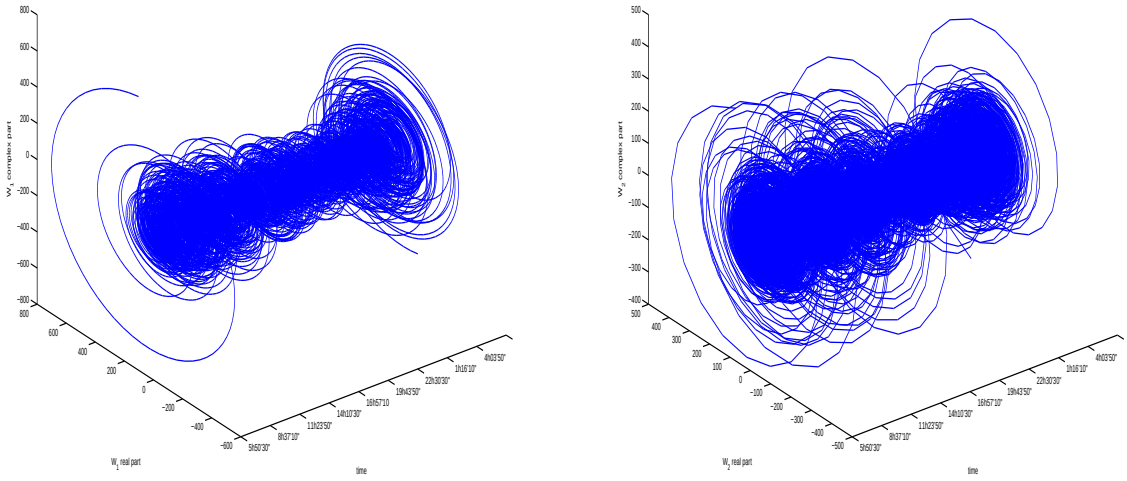


FIG. 3 – *The wavelet coefficients in the orthosympathetic band (left) and in the parasympathetic band (right) of the same healthy subject during a period of 24 hours*

3 Segmentation analysis

We assume that the process $\{X_t\}$ is abruptly changing and is characterized by a parameter $\theta \in \Theta$ that remains constant between two changes. We use this assumption to define our contrast function $J(\boldsymbol{\tau}, \boldsymbol{X})$. Let K be some integer and let

$\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_{K-1}\}$ be an ordered sequence of integers satisfying $0 < \tau_1 < \tau_2 < \ldots < \tau_{K-1} < n$.

For the detection of changes in the mean and variance of a sequence of random variables, i.e., a change in distribution, we use the following contrast function, based on a Gaussian log–likelihood function :

$$J_n(\boldsymbol{\tau}, \boldsymbol{X}) = \frac{1}{n} \sum_{k=1}^{K} \frac{\|X_{\tau_k} - \bar{X}_{\tau_k}\|^2}{\hat{\sigma}_k^2} + n_k \log(\hat{\sigma}_k^2). \tag{5}$$

where $n_k = \tau_k - \tau_{k-1}$ is the length of segment k, $\hat{\sigma}_k^2$ is the empirical variance computed on that segment k, $\hat{\sigma}_k^2 = n_k^{-1} \sum_{i=\tau_{k-1}+1}^{\tau_k} (X_i - \bar{X})^2$, and $\bar{X}$ is the empirical mean of $X_1, \ldots, X_n$.

In fact, we minimize a penalized version of this contrast function, the penalty parameter being selected in an adaptive way so that the obtained segmentation does not depend too much on the penalty parameter ; see also Birgé and Massart (2007) for another choice for the penalty parameter. Although this method has been devised for Gaussian processes, it empirically provides relevant results for non-Gaussian processes, e.g., financial time series, see Lavielle and Teyssière (2006) for further details.

3.1 Orthosympathetic band

For that frequency band, we obtain the following segmentation :

$$\boldsymbol{\tau} = \{28220, 33366, 71048\},$$

i.e., 13h40’50”, 15h06’36”, 1h34’38”.

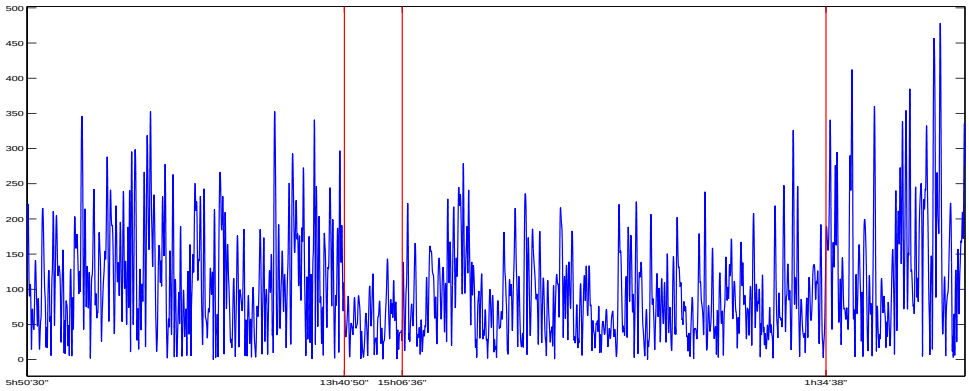


FIG. 4 – *Segmentation in the mean and variance (change in the distribution) of the modulus of the wavelet coefficients in the orthosympathetic band*

3.2 Parasympathetic band

For that frequency band, we obtain the following segmentation :

$$\boldsymbol{\tau} = \{11620, 21912, 28054, 31540, 36022, 40172, 52622, 70550\},$$

i.e., 9h04’10”, 11h45’42”, 13h38’04”, 14h36’10”, 15h50’22”, 17h00’02”, 20h27’32”, 1h26’40”.

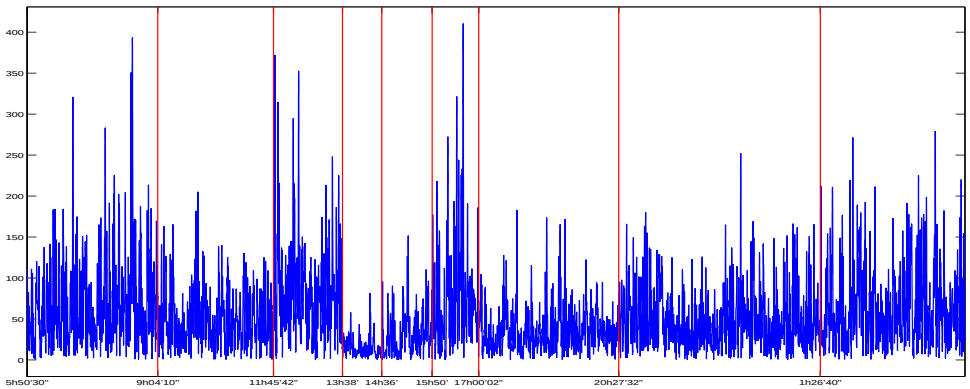


FIG. 5 – *Segmentation in the mean and variance (change in the distribution) of the modulus of the wavelet coefficients in the parasympathetic band*

4 Conclusion

The example illustrated by Figures 4, 5 and 6 shows at around $t = 28,220$ a simultaneous variation of both systems. But, one can also observe change-points in the orthosympathetic and parasympathetic bands occurring at different times. In the future, we will study the existence of a possible causality or sequentiality between these change–points in different bands.

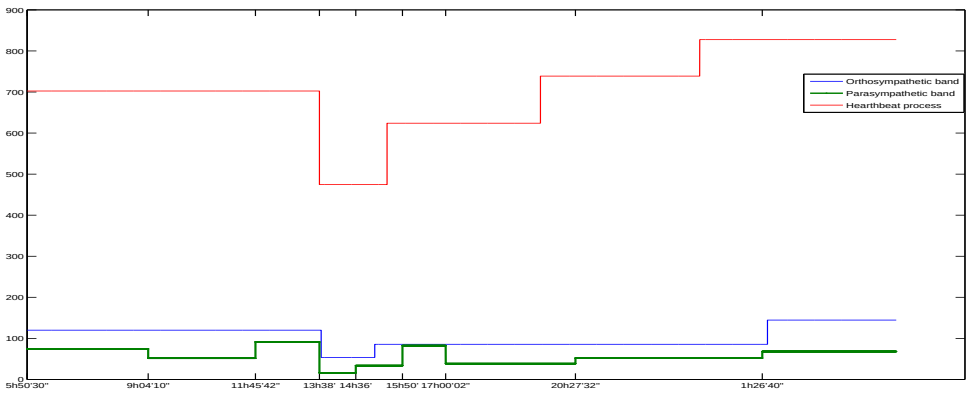


FIG. 6 – *Mean of the modulus if the wavelets coefficients for the orthosympathetic and parasympathetic bands, and of the RR process* $X(t)$

Références

[1] Bardet, J.M. and Bertrand, P.R. (2007). Identification of the multiscale fractional Brownian motion with biomechanical applications. *J. Time Ser. Anal.* **28**, 1–52.

[2] Bardet, J.M., Bertrand, P.R. and Billat, V. (2008). Estimation non-paramétrique de la densité spectrale d'un processus gaussien observé à des instants aléatoires. *Pub. ISUP* **52**, 123–138.

[3] Birgé, L. and Massart, P. (2007). Minimal penalties for Gaussian model selection. *Probab. Theory Related Fields* **138**, 33–73.

[4] Boudet, G., Albuisson, E., Bedu, M. and Chamoux, A. (2004). Heart rate running speed relationships during exhaustive bouts in the laboratory. *Can. J. Appl. Physiol.* **29**(6), 731–742.

[5] Chamoux, A. (1984). Le système Holter en pratique. *Médecine du Sport* **58**, 43-273, 54–284.

[6] Cottin, F., Leprêtre, PM., Lopes, P., Papelier, Y., Médigue, C. and Billat V. (2006). Assessment of ventilatory thresholds from heart rate variability in well-trained subjects during cycling, *Int. J. Sports Med.* **27**, 959–967.

[7] Diab, M.O., Marque, C. and Khalil, M.A. (2007). Classification for uterine EMG Signals : Comparison between AR model and statistical classification method. *Int. J. Computational Cognition* **5**, 8–14,

[8] Frisch, U. (1996). *Turbulence*. Cambridge University Press.

[9] Goldberger, A.L. (2001). Heartbeats, hormones and health : is variability the spice of life ? *Am. J. Crit. Care Med.* **163**, 1289–1290.

[10] Ivanov, P.C., Amaral, L.A.N., Goldberger, A.L., Havlin, S., Rosenblum, M., Struzik, Z, and Stanley, H.E. (1999). Multifractality in human heartbeat dynamics. *Nature* **399**, 461–465.

[11] Lavielle, M. and Moulines, E. (2000). Least-squares estimation of an unknown number of shifts in a time series. *J. of Time Series Anal.* **21**, 33–59.

[12] Lavielle, M. and Teyssière, G. (2006). Detection of multiple change-points in multivariate time series. *Lithuanian Math. J.* **46**, 287–306. [Détection de ruptures multiples dans des séries temporelles multivariées.] (Translation in French) Liet. Mat. Rink. **46**, 351–376.

[13] Mallat, S. (1998). *A wavelet tour of signal processing*. Academic Press.

[14] Task force of the European Soc. Cardiology and the North American Society of Pacing and Electrophysiology (1996). Heart rate variability. Standards of measurement, physiological interpretation, and clinical use. *Circulation* **93**, 1043–1065.

Summary

We propose a new method for estimating the change-points of heart rate in the orthosympathetic and parasympathetic bands, based on the wavelet transform in the complex domain and the study of the change-points in the moments of the modulus of these wavelet transforms. We observe change-points in the distribution for both bands.

Découverte complète et interactive de motifs temporels avec contraintes numériques à partir de séquences d'événements.

Damien Cram*, Béatrice Fuchs*, Yannick Prié*, Alain Mille*

* Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
damien.cram,beatrice.fuchs,yannick.prie,alain.mille@liris.cnrs.fr
http://liris.cnrs.fr

Résumé. Nous proposons une méthode complète d'extraction de motifs temporels fréquents avec contraintes numériques à partir d'une séquence d'événements. Ces motifs, appelés *chroniques*, sont très expressifs par rapport aux motifs séquentiels plus «classiques» tels que les épisodes parallèles ou en série, ce qui implique une grande complexité dans la résolution complète. Un algorithme de résolution permettant l'introduction d'heuristiques et de contraintes utilisateur est proposé. Il est ensuite discuté comment mettre à profit les interactions avec l'utilisateur pour compenser sa grande complexité et le rendre praticable.

1 Introduction : contexte et motivations

[1]De nombreux domaines génèrent des données à caractère temporel et nécessitent des méthodes de fouilles adaptées à cet aspect. C'est notamment le cas du domaine de l'analyse de l'activité, particulièrement lorsque l'activité a été capturée et modélisée en une trace d'interactions racontant l'histoire des actions du sujet (Cram et al., 2008). C'est dans ce cadre que Cram et al. (2008) ont besoin d'une méthode permettant l'abstraction des connaissances sur l'usage d'un système par son utilisateur. Pour cela, il est proposé de trouver dans la trace des motifs temporels pertinents, pouvant être considérés comme la description abstraite d'une situation.

Dans cet article nous proposons une formalisation de ce problème et nous en proposons une solution. La trace d'interactions est considérée comme une séquence d'événements où chaque événement représente une action du sujet. Les méthodes *Winepi* et *Minepi* (Mannila et al., 1997) extraient dans une séquence les épisodes fréquents, en série ou en parallèle. De nombreuses améliorations ont été ajoutées à cette méthode, mais rares sont celles qui ont porté sur la découverte d'épisodes «hybrides», c'est-à-dire imposant un ordre partiel sur les occurrences d'événements. Rares sont également les méthodes qui proposent de quantifier les écarts entre deux événements d'un même épisode.

Ces deux limitations sont très préjudiciables à la bonne représentation de certains motifs de comportement dans l'activité. Par exemple, dans l'activité d'achat en ligne, le motif «renseigner son adresse» serait constitué des événements «cliquer sur le lien `Mon_adresse`», *puis* « remplir le champ `Numéro`» et « remplir le champ `Nom_de_rue`» dans n'importe quel ordre, *puis* de « cliquer sur `Valider`». Sans un formalisme hybride, il serait impossible de représenter le parallélisme des deux événements «remplir» et leur sérialité avec les deux autres. D'autre part, la quantification des écarts entre événements dans le motif permet de savoir si les actions ont été réalisées rapidement ou non, puis par comparaison entre utilisateurs de savoir si le sujet est débutant ou expérimenté, et ainsi d'adapter l'assistance.

Le formalisme de *chronique* (Dousson et al., 1993) permet de représenter ces deux aspects. Une chronique est un épisode permettant de spécifier des écarts minimaux et maximaux entre chaque événement de l'épisode. La question de la découverte de chroniques dans une séquence d'événements a été traitée par Duong (2001), mais la méthode proposée n'est pas complète, car une seule contrainte temporelle n'est extractible pour un couple d'événements donné (cf. section 2). Cette limitation est très restrictive dans la mesure où la grande majorité des chroniques ne seront pas candidates dans le processus de découverte. De plus, dans le cadre de la recherche de motifs dans l'activité, cette limitation nous semble arbitraire.

Afin de ne pas restreindre ainsi *a priori* la découverte de chroniques intéressantes à un sous-ensemble de l'ensemble des chroniques candidates, nous proposons une méthode complète de découverte de chroniques. Évidemment la contrainte de la complétude dans un processus de découverte de motifs très expressifs comme les chroniques fait exploser la complexité, comme nous le verrons par la suite. Mais nous pensons qu'il existe des stratégies pour rendre une telle méthode praticable.

[1]Cette recherche est financée par l'ANR dans le cadre du projet PROCOGEC (`www.procogec.com`).

En effet, la découverte de connaissances est un processus interactif et itératif. Une fois qu'un algorithme de fouille de données est exécuté, ses résultats sont analysés par l'utilisateur qui, selon sa satisfaction, ajustera plus ou moins les paramètres d'entrée de l'algorithme dans l'espoir d'obtenir de meilleurs résultats. Il existe tout de même un certain nombre de travaux qui tentent de bénéficier des interactions avec l'utilisateur pour améliorer le processus de découverte, notamment en ce qui concerne les séquences d'événements. Parthasarathy et al. (1999) et Lin et Lee (2004) proposent de stocker et d'organiser en mémoire, appelée parfois base de connaissances, tout ou partie des résultats des exécutions précédentes de telle sorte que lorsqu'une nouvelle requête est soumise, il soit possible de calculer les nouveaux résultats à partir des informations déjà présentes en mémoire sans avoir à parcourir à nouveau l'ensemble des données. Ces méthodes sont qualifiées d'«interactives», mais l'unique but des interactions est l'optimisation du temps d'exécution du processus au fur et à mesure que les requêtes s'accumulent.

D'autres travaux plus récents visent à tirer profit des interactions avec l'utilisateur pour bâtir et maintenir une base de connaissances que l'utilisateur a sur son domaine. Les algorithmes utilisent ensuite ces connaissances pour mesurer l'intérêt des différents motifs du point de vue de l'utilisateur. Par exemple, Fauré (2007) a mis en place un réseau bayésien pour modéliser les dépendances connues de l'utilisateur sur son domaine. Ce réseau bayésien est mis à jour par les informations annotées par l'utilisateur sur les règles extraites par l'algorithme.

L'interactivité dans le processus de découverte permet notamment l'accélération de l'exécution et l'augmentation de la pertinence des motifs extraits. L'activité fondamentale en fouille de données consiste à élaborer des algorithmes permettant d'explorer le plus grand volume de données possible en un temps le plus acceptable possible. Mais grâce à ces deux aspects de l'interactivité, il est possible de s'attaquer à des problèmes de découverte de plus grande complexité. C'est ce que nous faisons ici. L'objectif principal de cet article n'est pas d'étudier cet aspect interactif, mais de proposer une méthode de fouille répondant au problème posé par notre domaine d'application qui s'avère être très coûteux en calcul. La méthode proposée a été pensée pour offrir des mécanismes d'optimisation par les interactions, dans des développements futurs.

La section 2 pose le problème de la découverte complète de chroniques et une méthode de résolution est proposée en section 3. Les optimisations interactives possibles sont discutées en section 4. La section 5 conclut et dresse les perspectives de ce travail.

2 Formulation du problème

Une *séquence d'événements* est un ensemble noté $\mathcal{S} = \langle (e_1, t_1) \dots (e_l, t_l) \rangle$, où chaque (e_i, t_i) est un *événement* composé d'un *type d'événement* $e_i \in \mathbb{E}$ et d'un entier t_i appelé sa *date*. L'ensemble $\mathbb{E}$ est supposé totalement ordonné et fini. Les motifs temporels recherchés sont les *chroniques*. Une *contrainte temporelle* est un quadruplet (e_g, e_d, I^-, I^+), noté $e_g[I^-, I^+]e_d$, où $(e_g, e_d) \in \mathbb{E}^2$ et I^- et I^+ sont deux entiers vérifiant $I^- \leq I^+$. On dit que deux événements (e_1, t_1) et (e_2, t_2) satisfont la contrainte $e_g[I^-, I^+]e_d$ si et seulement si $e_1 = e_g$ et $e_2 = e_d$ et $t_2 - t_1 \in [I^-, I^+]$, ou si $e_1 = e_d$ et $e_2 = e_g$ et $t_1 - t_2 \in [I^-, I^+]$. Une *chronique* est un couple $(\mathcal{E}, \mathcal{T})$, où :

- $\mathcal{E} = \varepsilon_1 \dots \varepsilon_n$ avec $\forall i, \varepsilon_i \in \mathbb{E}$ et $\forall i < j, \varepsilon_i \leq_{\mathbb{E}} \varepsilon_j$; ($\mathcal{E}$ est la partie *épisode* de la chronique ; n est la *taille* de la chronique)
- $\mathcal{T} = \{\tau_{ij}\}_{1 \leq i < j \leq |\mathcal{E}|}$ est un ensemble de contraintes temporelles sur $\mathcal{E}$ tel que $\forall i < j, \tau_{ij} = \varepsilon_i[\tau_{ij}^-, \tau_{ij}^+]\varepsilon_j$. ($\mathcal{T}$ est la partie *contraintes* de la chronique)

Enfin, une *occurrence* d'une chronique $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ est une liste d'événements de $\mathcal{S}$, notée $\langle (\varepsilon_1, t_1) \dots (\varepsilon_n, t_n) \rangle$ et vérifiant $\forall i < j, t_j - t_i \in [\tau_{ij}^-, \tau_{ij}^+]$. On notera $\mathcal{O}(\mathcal{C}, \mathcal{S})$ l'ensemble des occurrences de $\mathcal{C}$ dans $\mathcal{S}$.

Par exemple, avec $\mathbb{E} = A, B, C$ et $\mathcal{S} = \langle (A,1)(C,2)(B,4)(A,5)(C,5)(B,6) \rangle$, les occurrences de $\mathcal{C}_1 = (ABC, \{A[1,3]B, A[1,1]C, B[-2,-1]C\})$ sont $\langle (A,1)(C,2)(B,4) \rangle$ et $\langle (A,5)(C,5)(B,6) \rangle$ (cf. figure 1).

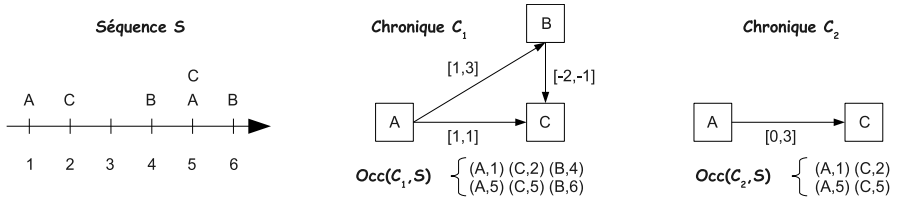
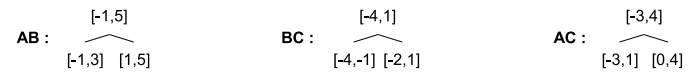


FIG. 1 – *Exemple de séquence d'événements, de chroniques et d'inclusion ($\mathcal{C}_1 \preceq \mathcal{C}_2$).*

Une chronique $\mathcal{C}$ est dite *incluse* dans une chronique $\mathcal{C}'$ ($\mathcal{C}'$ est *plus générale* que $\mathcal{C}$) si $\mathcal{E}'$ est un sous-épisode de $\mathcal{E}$, et si avec h la *fonction d'inclusion* telle que $\mathcal{E}' = \varepsilon'_1 \dots \varepsilon'_{n'} = \varepsilon_{h(1)} \dots \varepsilon_{h(n')}$ on a $\forall i < j, [\tau_{ij}^-, \tau_{ij}^+] \subseteq [\tau_{h(i)h(j)}^-, \tau_{h(i)h(j)}^+]$. Par exemple, sur la figure 1, $\mathcal{C}_1$ est incluse dans $\mathcal{C}_2$.

Étant donné un entier f_{seuil} appelé la *fréquence seuil*, on définit la *fréquence* d'une chronique $\mathcal{C}$ dans une séquence $\mathcal{S}$ comme le nombre de ses occurrences : $f(\mathcal{C}) = |\mathcal{O}(\mathcal{C}, \mathcal{S})|$. Duong (2001) propose de définir $\mathcal{O}(\mathcal{C}, \mathcal{S})$ comme l'ensemble des occurrences de $\mathcal{C}$ reconnues par l'algorithme *CRS* de Dousson et al. (1993). *CRS* ne reconnaît qu'un ensemble d'occurrences $\mathcal{O}(\mathcal{C}, \mathcal{S})_{CRS}$ tel qu'il n'existe pas deux occurrences partageant le même événement. C'est la définition que nous garderons pour la suite, car elle donne un nombre d'occurrences intuitif et présente des propriétés intéressantes pour la découverte, notamment la monotonie par rapport à $\preceq$ (Duong, 2001).

Étant donnés une séquence $\mathcal{S}$ et un minimum de fréquence f_{seuil}, il existe pour chaque épisode $\varepsilon_1\varepsilon_2$ plusieurs contraintes temporelles τ telles que la chronique $(\varepsilon_1\varepsilon_2, \{\tau\})$ soit fréquente (i.e. $f(\varepsilon_1\varepsilon_2, \{\tau\}) \geq f_{seuil}$). Pour trouver toutes ces contraintes τ, on utilise une méthode inspirée de celle proposée par Duong (2001). Reprenons l'exemple de $\varepsilon_1\varepsilon_2 = AB$ avec $f_{seuil} = 3$. On calcule d'abord l'ensemble des amplitudes des occurrences de AB. Cet ensemble est $\{3, 5, -1, 1\}$ et on le tri en $\{-1, 1, 3, 5\}$. À partir de cet ensemble, on bâtit tous les intervalles $[i, j]$ assurant que $f(A[i, j]B) \geq 3$. Les intervalles tels que $f(A[i, j]B) = 3$ sont $[-1, 3]$ et $[1, 5]$, et celui pour $f(A[i, j]B) = 4$ est $[-1, 5]$. On peut organiser ces trois intervalles en un graphe d'inclusions de contraintes pour AB. En appliquant cette méthode pour BC et AC, on obtient ainsi pour $\mathcal{S}$ et $f_{seuil} = 3$ une base de contraintes $\mathcal{D}$ comme suit :



On notera $\mathcal{D}^\top$ l'ensemble des chroniques de taille 2 qui sont construites à partir de la contrainte la plus générale de $\mathcal{D}$. Ici, $\mathcal{D}^\top = \{A[-1, 5]B, B[-4, 1]C, A[-3, 4]C\}$.

Le problème de la découverte complète de chroniques fréquentes ne peut se formuler qu'une fois que $\mathcal{D}$ a été ainsi construite : « trouver toutes les chroniques dont les contraintes temporelles sont dans $\mathcal{D}$ précédemment construit, telles que $f(\mathcal{D}) \geq f_{seuil}$ ».

La méthode non-complète de Duong (2001) propose pour chaque couple d'événements de ne garder dans la base de contraintes qu'une seule contrainte numérique et non pas l'ensemble des contraintes fréquentes. Ce choix est fait par l'intermédiaire d'un mécanisme de notation de chaque contrainte numérique. En conséquence, l'ensemble des chroniques découvertes par la suite étant constitué uniquement de chroniques construites à partir de cette base incomplète de contraintes n'est pas complet.

3 Résolution complète

Étant donnée une base de contraintes $\mathcal{D}$, une chronique $\mathcal{C}'$ est dite *directement incluse* dans une chronique $\mathcal{C}$ si et seulement si $\mathcal{C}$ et $\mathcal{C}'$ sont contruites à partir de $\mathcal{D}$, et $\mathcal{C} \prec \mathcal{C}'$ (inclusion stricte), et il n'existe pas de chroniques $\mathcal{C}''$ construite à partir de $\mathcal{D}$ telle que $\mathcal{C} \prec \mathcal{C}'' \prec \mathcal{C}'$. Ce concept d'*inclusion directe* permet d'introduire la notion de *successeur* d'une chronique, où $\mathcal{C}$ est dite *successeur* de $\mathcal{C}'$ si $\mathcal{C}$ est directement incluse dans $\mathcal{C}'$, relativement à $\mathcal{D}$. C'est cette notion de *successeur* qui est à la base de la résolution complète. En effet, à partir d'un petit ensemble de chroniques très générales, la résolution consiste à générer et tester pour chacune d'elle et en profondeur d'abord de successeurs en successeurs toutes les chroniques candidates, jusqu'à être sûr qu'il n'y ait plus de successeurs qui puissent être fréquents.

Il existe deux types d'opérateur permettant de générer un successeur d'une chronique $\mathcal{C}$. Le premier consiste à ajouter un type d'événement à $\mathcal{C}$, le deuxième consiste à sélectionner dans $\mathcal{C}$ une contrainte $\tau_{ij} = e_i[\tau_{ij}^-, \tau_{ij}^+]e_j$ et à la remplacer par $\sigma_{ij} = e_i[\sigma_{ij}^-, \sigma_{ij}^+]e_j$, où $\sigma_{ij} \in \mathcal{D}$ et σ_{ij} est directement plus stricte que τ_{ij} dans $\mathcal{D}$. Le nombre de successeurs d'une chronique relativement à une base de contraintes est donc fini puisqu'il y a $|\mathbb{E}|$ opérateurs du premier type et en notant $\mathcal{T}$ l'ensemble des contraintes de $\mathcal{C}$, $\sum_{\tau_{ij} \in \mathcal{T}} n_{\tau_{ij}}$ opérateurs du deuxième type, où $n_{\tau_{ij}}$ est le nombre de contraintes directement plus strictes que τ_{ij} dans $\mathcal{D}$.

Pour résoudre le problème de découverte complète, on applique l'algorithme suivant.

Entrées: $\mathcal{S}$; $\mathcal{D}$; f_{seuil}
Sorties: F
1: F$\leftarrow \emptyset$
2: O $\leftarrow \mathcal{D}^\top$
3: **répéter**
4: $\mathcal{C} \leftarrow$ choisir(O)
5: **si** a_plus_gén(nonF,$\mathcal{C}$) **alors**
6: continuer à la ligne 3
7: **si** $\neg$a_plus_stricte(F,$\mathcal{C}$) **alors**
8: $f(\mathcal{C}) \leftarrow$compter($\mathcal{C}, \mathcal{S}$)
9: **si** $f(\mathcal{C}) \geq f_{seuil}$ **alors**
10: ajout_min($\mathcal{C}$,F)
11: **sinon**
12: ajout_max($\mathcal{C}$,nonF)
13: continuer à la ligne 3
14: **sinon**
15: $Succ(\mathcal{C}) \leftarrow$ gén_succ($\mathcal{C}$)
16: **pour** chaque $\mathcal{C}' \in Succ(\mathcal{C})$ **faire**

17: **si** est_acceptable($\mathcal{C}'$) **alors**
18: $\mathtt{O} \leftarrow \mathtt{O} \cup \{\mathcal{C}'\}$
19: **jusqu'à** $\mathtt{O} \neq \emptyset$
20: **retourner** F

L'idée de l'algorithme est de prendre chaque chronique $\mathcal{C}$ construite à partir de $\mathcal{D}$, de calculer sa fréquence dans $\mathcal{S}$ (8) et si on a $f(\mathcal{C}) \geq f_{seuil}$ de traiter de même tous ses successeurs par la suite. Ce processus est effectué dans la boucle 3–19. L'ensemble O qui contient à tout moment l'ensemble des chroniques à traiter dans les itérations futures. L'ensemble F contient l'ensemble des chroniques fréquentes minimales, c'est-à-dire qu'il n'existe pas dans F $\mathcal{C}$ et $\mathcal{C}'$ telles que $\mathcal{C} \prec \mathcal{C}'$. À chaque itération on prend une nouvelle chronique $\mathcal{C}$ de O à traiter (4). Si $\mathcal{C}$ est fréquente, on génère alors ses successeurs et on les ajoute à l'ensemble O (15–18). Le processus se termine lorsqu'il n'y a plus de chroniques à traiter (19) et on retourne alors F (20). Il est initié avec toutes les contraintes les plus générales de $\mathcal{D}$ (2) et on retourne F (20).

Les autres lignes de l'algorithme correspondent à deux optimisations de ce processus. La première optimisation vise à ne compter la chronique que si on n'est pas déjà sûr qu'elle sera fréquente ou non-fréquente (tests des lignes 5 et 7). À cette fin, un ensemble nonF contenant toutes les chroniques maximales non-fréquentes est maintenu tout au long de la découverte. Si nonF contient une chronique plus générale que $\mathcal{C}$, alors cela signifie que sans avoir à compter $\mathcal{C}$, $\mathcal{C}$ n'est pas fréquente car f est monotone. La procédure a_plus_gén renverra alors vrai et la processus reprendra à la ligne 3. De même, si $\mathcal{C}$ est plus générale qu'une chronique déjà fréquente, alors ce n'est la peine ni de la compter, ni de traiter ses successeurs (cf. ligne 7). La deuxième consiste à vérifier que les successeurs sont acceptables (17). Pour cela, la procédure est_acceptable vérifie que le successeur est *consistant*, c'est-à-dire que la chronique n'est pas sur-contrainte (Duong, 2001), et qu'elle vérifie également des contraintes éventuelles spécifiées par l'utilisateur. La procédure est_acceptable vérifie également que le successeur vaut la peine d'être traité, c'est-à-dire qu'on ne sait pas déjà grâce à F et nonF qu'il sera fréquent ou non-fréquent.

4 Analyse de l'agorithme et premiers résultats

4.1 Propriétés de l'algorithme

La complétude de l'algorithme proposé peut être démontré par le raisonnement suivant. Soit $\mathcal{C}$ une chronique construite à partir de $\mathcal{D}$, alors il existe une chronique $\mathcal{C}' \in \mathcal{D}^{\top}$ telle que $\mathcal{C} \preceq \mathcal{C}'$, donc si on initialise le processus à $\mathcal{C}'$ et qu'on traite systématiquement tous ses successeurs, on traitera alors toutes les sous-chroniques de $\mathcal{C}'$ dont $\mathcal{C}$.

La terminaison de l'algorithme est assurée par le raisonnement qui suit. Soit n_{max} la taille de la chronique la plus grande parmi toutes les chroniques fréquentes minimales de $\mathcal{S}$ construites à partir de $\mathcal{D}$. On est alors certain que gén_succ, qui ne s'applique qu'à des chroniques fréquentes, ne générera jamais de chroniques de taille $n_{max}+2$, car alors c'est qu'une chronique de taille $n_{max}+1$ serait fréquente. Soit $N_{\leq}(n_{max}+1)$ le nombre total de chroniques (fréquentes et non-fréquentes) construites à partir de $\mathcal{D}$. Comme le test est_acceptable assure qu'on ne mettra jamais deux fois la même chronique dans l'ensemble O, on est certain qu'à la i^{e} itération de la ligne 3, il reste $N_{\leq}(n_{max}+1)-i$ chroniques potentielles à traiter, donc que l'algorithme aura au pire encore $N_{\leq}(n_{max}+1)-i$ itérations à faire. Le paramètre $N_{\leq}(n_{max}+1)-i$ est donc un paramètre strictement décroissant de l'algorithme.

Au pire des cas, l'algorithme traitera donc $N_{\leq}(n_{max}+1)$ chroniques, où $N_{\leq}(n)$ est le nombre de chroniques de taille inférieure ou égale à n. En notant $N_{=}(n)$ le nom total de chroniques de taille n, on a $N_{\leq}(n_{max}+1) = N_{=}(2) + N_{=}(3) + \ldots + N_{=}(n_{max}+1)$. Si on suppose que $\mathcal{D}$ est constituée de graphes de contraintes étant tous de taille p, on a alors $N_{=}(n) = |\mathbb{E}|^n \times p^{C_n^2}$, et donc $N_{\leq}(n_{max}) = O(p^{n^2_{max}})$. Dans la pratique, l'introduction des tests de non-fréquence permet de ne pas traiter une branche de sous-chroniques dès lors qu'on sait qu'une chronique est non-fréquente. Pour la séquence $\mathcal{S}' = \langle (A,1)(B,3)(A,4)(C,4)(A,7)(B,8)\ (C,9)(B,10)(B,12)\rangle$ avec $f_{seuil} = 2$, on a $N_{\leq} \simeq 10^7$ et pourtant la découverte peut nécessiter moins de 100 itérations selon le choix de l'heuristique (cf. section 4.2).

4.2 Amélioration des performances et de la pertinence par l'interactivité

L'algorithme exposé précédemment est conçu de telle sorte qu'il permette d'y introduire la possibilité de prendre en compte facilement des contraintes spécifiées par l'utilisateur. Chaque contrainte est une condition supplémentaire traitée par est_acceptable lors de l'ajout des successeurs. En procédant de la sorte, le processus de découverte restera complet pour toute contrainte monotone pour l'inclusion de chroniques telle la taille maximale de la chronique, le fait d'être plus général qu'une chronique spécifiée par l'utilisateur, etc. La contrainte d'inclusion dans une chronique est au contraire *antimonotone*. On peut par exemple imaginer que l'utilisateur ne s'intéresse qu'à des chroniques comprenant au moins les événements A et B telles que B succède à A, c'est-à-dire qu'il recherche toutes les chroniques $\mathcal{C}$ vérifiant

$\mathcal{C} \preceq A[0,+\infty]B$. Pour prendre une telle contrainte en compte on pourra initialiser le processus de découverte à $\texttt{O} = A[0,+\infty]B$, réduisant ainsi considérablement l'espace de recherche.

Il est également immédiat d'introduire dans cet algorithme des heuristiques dans le choix effectué par `choisir` de la prochaine chronique à traiter. En effet, si `choisir` est capable de choisir parmi l'ensemble des chroniques à traiter les plus «intéressantes», alors l'ensemble `F` aura tendance à contenir plus rapidement les chroniques qui intéressent l'utilisateur. L'idée du choix de l'heuristique est que dans un temps très court, on puisse extraire une grande partie des chroniques les plus pertinentes même si le pourcentage d'exploration de l'espace des chroniques candidates est encore très faible. On pourra alors se satisfaire du résultat temporaire de l'extraction et couper l'exécution. Pour trouver de telles heuristiques, il est possible de s'inspirer des nombreuses mesures d'intérêt sur les motifs existant dans la littérature (Geng et Hamilton, 2006). Ces heuristiques sont des fonctions visant à mesurer l'intérêt d'un motif sous un autre angle que celui classique de la *fréquence* du motif. On trouve par exemple des mesures visant à spécifier le niveau d'*utilité* pour l'utilisateur, de *nouveauté* ou de *particularité* par rapport aux autres motifs. Ces mesures d'intérêt ont dans la grande majorité été conçues pour l'application à des motifs de type «règle d'association» et très peu à des motifs séquentiels et encore moins au cas particulier des chroniques. Il y a donc un travail important à effectuer d'innovation et d'adaptation de ces mesures pour les données séquentielles.

Le couplage des heuristiques et des contraintes peut mener à un cercle vertueux que nous cherchons à engendrer. En effet, nous imaginons un cycle de découverte interactive dans lequel les premiers résultats retournés par les premières itérations de l'algorithme permettent à l'utilisateur de spécifier des contraintes supplémentaires pour préciser sa recherche. De même, les premiers résultats de la deuxième exécution permettent de peaufiner les contraintes, et ainsi de suite. À chaque interaction avec l'utilisateur, des connaissances sur les chroniques recherchées par l'utilisateur seront ainsi capitalisées et peuvent être exploitées pour élaborer des heuristiques reflétant plus le point de vue de l'utilisateur.

Nous avons appliqué à la séquence $\mathcal{S}'$ trois heuristiques différentes et très simples : *choix aléatoire*, *FIFO* et *LIFO*. Pour *choix aléatoire*, l'algorithme proposé a nécessité 215 itérations et 81 comptages (ligne 8), pour *FIFO* 90442 et 223, et pour *LIFO* 74 et 53. Ces heuristiques, pourtant peu élaborées, mettent en évidence que comme attendu que le choix de l'heuristique a une grande influence sur l'exécution du processus.

5 Conclusion

L'algorithme présenté dans cette article permet l'extraction de motifs temporels à partir d'une séquence d'événements avec expression de contraintes temporelles numériques entre les événements du motifs. La méthode proposée est complète, contrairement à celle qui a été proposée par Duong (2001). La contre-partie de la complétude de la découverte et de la forte expressivité des motifs recherchés est la grande complexité en temps de la méthode. C'est pour cela que le processus de découverte proposé permet d'y introduire facilement des contraintes et des heuristiques permettant respectivement de réduire l'espace de recherche et d'orienter la recherche en priorité vers les chroniques les plus intéressantes pour l'utilisateur. L'autre avantage de l'utilisation d'heuristiques est qu'il est possible pour l'utilisateur, à la manière d'un algorithme anytime, de se satisfaire à un tout moment du processus des solutions partielles découvertes. L'hypothèse faite est que ces solutions partielles donnent assez d'indications à l'utilisateur quant à la pertinence de la requête qu'il a formulée et lui permet ainsi de reformuler une nouvelle requête plus contrainte. Ce processus interactif se poursuit jusqu'à ce que l'ensemble des contraintes accumulées au cours des itérations compensent la grande complexité et permettent de déboucher sur l'ensemble complet des chroniques satisfiant les contraintes en un temps raisonnable.

Ce qui est présenté ici est donc un «cadre de travail» pour la recherche interactive de chroniques, dont les travaux futurs doivent se porter d'une part sur la recherche de telles heuristiques et de contraintes, ainsi que sur l'étude de l'impact de ces heuristiques et contraintes sur l'efficacité du processus de découverte en termes de *rapidité d'exécution* et de *pertinence* des motifs découverts par rapport aux attentes de l'utilisateur.

Nous menons actuellement d'autres recherches sur l'extension du formalisme des chroniques à des événements persistants. Un tel événement a une date de début et une date de fin différentes, et il est à vrai dire plus exact de modéliser les *traces d'interaction* avec des événements persistants. Certains travaux comme ceux de Patel et al. (2008) ont rendu possible la découverte complète de motifs temporels dans des séquences d'événements persistants, mais ils ne permettent pas l'expression de contraintes numériques.

Références

Cram, D., B. Fuchs, Y. Prié, et A. Mille (2008). An approach to User-Centric Context-Aware Assistance based on Interaction Traces. In *MRC2008 : fifth International Workshop on Modeling and Reasoning in Context*.

Dousson, C., P. Gaborit, et M. Ghallab (1993). Situation recognition : Representation and algorithms. In *IJCAI*, pp. 166–174.

Duong, M. T. V. (2001). *Découverte de chroniques à partir de journaux d'alarmes. Application à la supervision de réseaux de télécommunications*. Ph. D. thesis, Institut National Polytechnique de Toulouse.

Fauré, C. (2007). *Découverte de réseau pertinents par l'implémentation d'un réseau bayésien : application à l'industrie aéronautique*. Ph. D. thesis, INSA de Lyon.

Geng, L. et H. J. Hamilton (2006). Interestingness measures for data mining : A survey. *ACM Comput. Surv. 38*(3), 9.

Lin, M.-Y. et S.-Y. Lee (2004). Interactive sequence discovery by incremental mining. *Inf. Sci. Inf. Comput. Sci. 165*(3-4), 187–205.

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.

Parthasarathy, S., M. J. Zaki, M. Ogihara, et S. Dwarkadas (1999). Incremental and interactive sequence mining. In *CIKM*, pp. 251–258.

Patel, D., W. Hsu, et M. L. Lee (2008). Mining relationships among interval-based events for classification. In *SIGMOD '08 : Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, New York, NY, USA, pp. 393–404. ACM.

Summary

This papers proposes a complete method for the discovery of frequent temporal patterns with numerical constraints from a sequence of events. These patterns are called *chronicles*. *Chronicles* are very expressive compared to more "common" sequential patterns like serial and parallel episodes. Consequently, the complete discovery process has a high complexity. We propose an algorithm that solves the complete dicovery problem by a heuristic search and enabling user constraints. It is also discussed the opportunity of taking advantage from user interactions to make this very complex process more efficient and to have an acceptable execution time.

Simulation et détection de l'évolution des données temporelles issues de l'usage du Web

Alzennyr Da Silva*[1], Yves Lechevallier*, Francisco De Carvalho**

* Projet AxIS, INRIA Paris-Rocquencourt
Domaine de Voluceau, Rocquencourt, B.P. 105,78153 Le Chesnay – France
{Alzennyr.Da_Silva, Yves.Lechevallier}@inria.fr
** CIn/UFPE, Caixa Postal 7851, CEP 50732-970, Recife (PE) – Brésil
fatc@cin.ufpe.br

Résumé. Dans le domaine des flux des données, la prise en compte du temps s'avère nécessaire pour l'analyse de ces données car leur distribution sous-jacente peut changer au cours du temps. Un exemple typique concerne les modèles des profils de navigation des internautes. Notre objectif est d'analyser l'évolution de ces profils, celle-ci peut être liée au changement d'effectifs ou aux déplacement de clusters au cours du temps. Afin d'analyser la validité de notre approche, nous mettons en place une méthodologie pour la simulation des données d'usage à partir de laquelle il est possible de contrôler l'occurrence des changements.

1 Introduction

La fouille de données d'usage du Web (*Web Usage Mining*, WUM) désigne l'ensemble de techniques basées sur la fouille de données pour analyser le comportement des utilisateurs d'un site Web (Cooley et al. (1999); Spiliopoulou (1999)). Dans ce contexte, cet article propose dans un premier temps une approche pour la détection de changements liés à l'usage du Web. Puis, une méthodologie de génération des données artificielles pour la simulation de l'usage du Web. Enfin, nous validons notre approche sur trois études de cas de différentes complexités. Les résultats ici présentés sont la suite des travaux déjà exposés dans les deux dernières conférences EGC (cf. Da Silva et Lechevallier (2008) et Da Silva et al. (2007)).

2 Approche de classification non supervisée pour la détection de changements

Notre approche de classification non supervisée pour la détection de changements consiste dans un premier temps à partitioner la période analysée en sous périodes plus petites (fenêtre) dans le but de suivre l'évolution des comportements qu'on manquerait par une analyse globale de toute la période. Considérons une fenêtre initiale W_1, une classification non supervisée est réalisée sur les données contenues à l'intérieur de cette fenêtre. Soit $(g_1, \ldots, g_c, \ldots, g_k)$ l'ensemble de prototypes découverts par la méthode de classification. Ensuite, on fait glisser la fenêtre sur le temps, ce qui nous donne une nouvelle fenêtre W_2. On utilise les prototypes de la fenêtre temporelle précédente pour obtenir une partition P_1 sur les données de la nouvelle fenêtre. Puis, une classification non supervisée est réalisée sur les données de W_2 en définissant une nouvelle partition P_2 sur ces données (cf. figure 1). La détection de changement entre les deux fenêtres sera donc mesurée par la comparaison des deux partitions P_1 et P_2 à l'aide des critères d'évaluation (cf. section 4). Comme méthode de classification, nous utilisons une version de l'algorithme *K-means* (cf. MacQueen (1967)), cependant d'autres algorithmes de partitionnement sont envisageables, mais ils doivent être capables d'affecter de nouvelles observations à une classification existante.

3 Méthodologie pour la génération des données artificielles

Nous proposons dans cet article une méthodologie pour la génération de données artificielles d'usage du Web sous la forme de tableau de contingence $navigations \times catégories\ de\ pages$. Une navigation est l'ensemble de requêtes appartenant au même utilisateur. Notre principale motivation est la possibilité de mesurer l'efficacité de notre approche de détection de changements sur un ensemble de données contenant des changements de comportements pré-établis et sur lesquels nous avons un contrôle total.

[1] L'auter remercie la CAPES-Brésil pour sont soutien à ce travail de recherche.

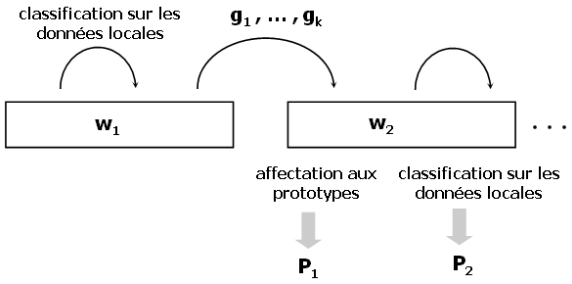


FIG. 1 – *Approche de classification non supervisée pour la détection de changements.*

Paramètre	Signification	Valeurs utilisés lors des simulations
t	nombre de thèmes/catégories de pages	10
$nbWindow$	nombre de fenêtres à générer	2
$nbClicMIN$	nombre minimum de clics dans un thème	10
$nbClicMAX$	nombre maximum de clics dans un thème	50
$totalInd$	nombre total d'individus dans une fenêtre	10.000
$c^{'}$	classe cible	1 (changement d'effectifs) et 2 (déplacement)
μ_1	facteur de rétrécissement	$\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$
μ_2	facteur de grossissement	$\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$
μ_3	facteur de déplacement	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$

TAB. 1 – *Paramètres de l'algorithme de génération des données artificielles.*

Afin de mieux comprendre la génération des données artificielles, considérons $(x_1, \ldots, x_i, \ldots, x_n)$ l'ensemble d'individus artificiels que nous voudrons générer (dans notre cas, un individu correspond à une navigation). Ainsi, un individu i est décrit par un ensemble de t variables : $x_i = \left\{x_i^1, \ldots, x_i^j, \ldots, x_i^t\right\}$. La valeur x_i^j représente la fréquence des clics réalisés par l'individu i sur les pages du thème j. Considérons également $(p_1, \ldots, p_c, \ldots, p_k)$ l'ensemble de profils initiaux de k classes *a priori* et $(\alpha_1, \ldots, \alpha_c, \ldots, \alpha_k)$ le pourcentage d'effectifs de chacune de ces classes, où $\alpha_c \in [0, 1]$ et $\sum_{c=1}^{k} \alpha_c = 1$.

3.1 Changements liés à l'effectif des classes

Pour les changements liés aux effectifs des classes, nous pouvons en avoir de deux types : le rétrécissement ou le grossissement de la classe cible. Afin de maintenir constant le nombre d'individus, le rétrécissement de la classe cible impliquera le grossissement des autres classes dans la fenêtre. De la même manière, le grossissement de la classe cible impliquera le rétrécissement des autres classes dans la fenêtre. L'intensité des changements se fait à l'aide de deux facteurs μ_1 et μ_2 où $\mu_1\mu_2 = 1$. Pour la description de l'algorithme mis en place pour la génération de données artificielles, nous devons tout d'abord connaître ses paramètres d'entrée (discriminés sur le tableau 1). L'algorithme pour la génération des données artificielles est décrit comme suit :

1. Pour chaque classe c
2. Pour chaque individu x_i de la classe c
3. $x_i \leftarrow \{0, ..., 0\}$
4. $nbClic \leftarrow$ random $(nbClicMIN, nbClicMAX)$
5. Pour y dans l'intervalle $[1, nbClic]$
6. $z \leftarrow random(0, 1)$
7. Pour j dans l'intervalle $[1, t]$
8. si $z <= cumul(p_c^j)$
9. $x_i^j \leftarrow x_i^j + 1$
10. Sortir de la boucle la plus interne

Pour le rétrécissement de la classe cible, nous devons multiplier son pourcentage d'effectifs par un facteur de rétrécissement μ_1 où $\mu_1 \in [0, 1]$. Afin de maintenir le même nombre d'individus dans la fenêtre, les autres classes doivent avoir leur effectif multiplié par $\mu_1^{'}$ qui est défini en fonction de μ_1. L'algorithme pour le rétrécissement de la classe cible est décrit comme suit :

1. Pour chaque classe c
2. Tirer au hasard et sans remise $(\alpha_c * totalInd)$ elements de la classe c
3. Pour w dans l'intervalle $[2, nbWindow]$
4. Pour c dans l'intervalle $[1, k]$
5. si $c = c^{'}$
6. $\alpha_c \leftarrow \alpha_c * \mu_1$
7. sinon
8. $\alpha_c \leftarrow \alpha_c * \mu_1^{'}$ où $\mu_1^{'} = \frac{1-\mu_1\alpha_{c^{'}}}{1-\alpha_{c^{'}}}$
9. Tirer au hasard et sans remise $(\alpha_c * totalInd)$ elements de la classe c

Pour le grossissement de la classe cible, nous devons diviser son pourcentage d'effectifs par un facteur de grossissement μ_2 où $\mu_2 \in [0, 1]$. Afin de maintenir le même nombre d'individus dans la fenêtre, les autres classes doivent avoir leur effectif multiplié par $\mu_2^{'}$ qui est défini en fonction de μ_2. L'algorithme pour le grossissement de la classe cible est décrit comme suit :

1. Pour chaque classe c
2. Tirer au hasard et sans remise $(\alpha_c * totalInd)$ elements de la classe c
3. Pour w dans l'intervalle $[2, nbWindow]$
4. Pour c dans l'intervalle $[1, k]$
5. si $c = c^{'}$
6. $\alpha_c \leftarrow \alpha_c / \mu_2$
7. sinon
8. $\alpha_c \leftarrow \alpha_c * \mu_2^{'}$ où $\mu_2^{'} = \frac{1-\frac{\alpha_{c^{'}}}{\mu_2}}{1-\alpha_{c^{'}}}$ avec $\mu_2 > \alpha_{c^{'}}$
9. Tirer au hasard et sans remise $(\alpha_c * totalInd)$ elements de la classe c

3.2 Changements liés au déplacement des classes

Pour effectuer le déplacement d'une classe c vers une classe cible $c^{'}$, nous devons ajouter à chaque valeur j des individus x_i de la classe c, un delta représentant l'écart entre les prototypes g_c et $g_{c'}$ pour la variable j. Un facteur $\mu_3 \in [0, 1]$ a pour but de contrôler l'intensité du déplacement des classes. L'algorithme pour le déplacement des classes est décrit comme suit :

1. Pour chaque classe c
2. Tirer au hasard et sans remise $(\alpha_c * totalInd)$ elements de la classe c
3. Pour w dans l'intervalle $[2, nbWindow]$
4. Pour c dans l'intervalle $[1, k]$
5. si $c \neq c^{'}$
6. Pour chaque j dans l'intervalle $[1, t]$
7. $\Delta^j = (g_c^j - g_{c'}^j) * \mu_3$
8. Pour chaque individu x_i de la classe c
9. $x_i^j \leftarrow x_i^j + \Delta^j$

4 Critères d'évaluation

4.1 F-mesure

La F-mesure (van Rijsbergen (1979)) combine la Précision et le Rappel (cf. équation 1) calculés à partir du tableau de contingence (cf. tableau 2). Cette mesure cherche le meilleur représentant d'un cluster A dans la partition P_1 par un cluster A' dans la partition P_2. Si $R = P = 1$, cela signifie $A = A'$.

$$F(i,j) = \frac{2P(i,j)R(i,j)}{P(i,j)+R(i,j)}, \text{où } P(i,j) = \frac{n_{ij}}{\sum_{i=1}^{m} n_{ij}} \text{ et } R(i,j) = \frac{n_{ij}}{\sum_{j=1}^{k} n_{ij}} \quad (1)$$

clusters de P_1 \ clusters de P_2	1	$\cdots$	j	$\cdots$	k	
1	n_{11}	$\cdots$	n_{1j}	$\cdots$	n_{1k}	$n_{1.}$
$\vdots$						
i	n_{i1}	$\cdots$	n_{ij}	$\cdots$	n_{ik}	$n_{i.}$
$\vdots$						
m	n_{m1}	$\cdots$	n_{mj}	$\cdots$	n_{mk}	$n_{m.}$
	$n_{.1}$		$n_{.j}$		$n_{.k}$	$n_{..} = n$

TAB. 2 – *Tableau de contingence entre deux partitions.*

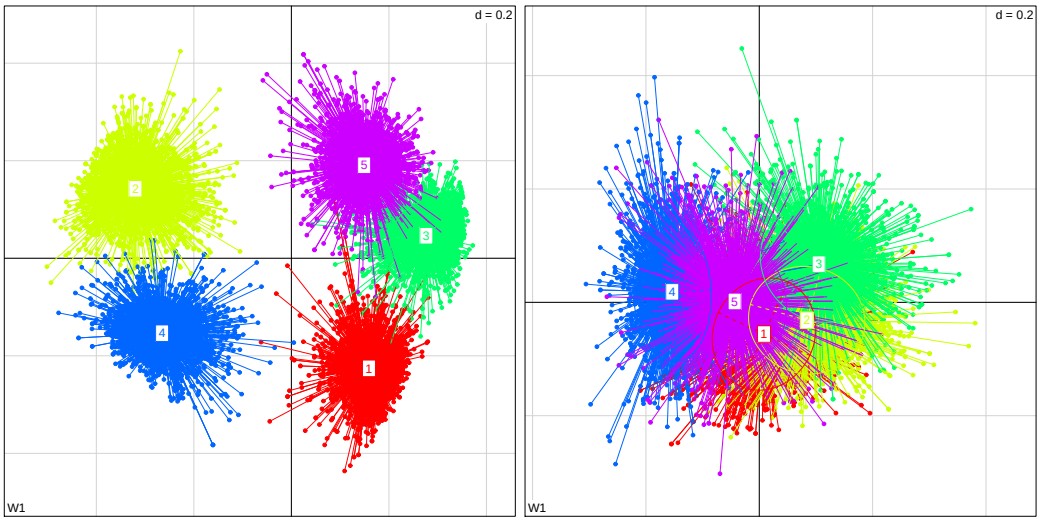


FIG. 2 – *Cinq classes artificielles bien separées (à gauche) et recouvrantes (à droite).*

Au niveau de la partition, la F-mesure est calculée à partir des moyennes (cf. équation 2). La F-mesure assume des valeurs dans le rang $[0, 1]$ et a pour but fournir une évaluation numérique de la similarité entre deux partitions issues d'un même ensemble d'individus.

$$F = \sum_{i=1}^{m} \frac{n_i}{n} max_{j=1,\ldots,k} F(i,j) \tag{2}$$

4.2 Indice corrigé de Rand

Pour une comparaison globale, l'indice corrigé (CR) de Rand (Hubert et Arabie (1985)) évalue le degré de ressemblance entre deux partitions. L'indice corrigé de Rand n'est pas sensible au nombre de clusters dans les partitions ni à la distribution des individus dans les clusters. Le CR est défini selon l'équation 3.

$$\text{CR} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{k}\binom{n_{ij}}{2} - \binom{n}{2}^{-1}\sum_{i=1}^{m}\binom{n_{i.}}{2}\sum_{j=1}^{k}\binom{n_{.j}}{2}}{\frac{1}{2}[\sum_{i=1}^{m}\binom{n_{i.}}{2} + \sum_{j=1}^{C}\binom{n_{.j}}{2}] - \binom{n}{2}^{-1}\sum_{i=1}^{m}\binom{n_{i.}}{2}\sum_{j=1}^{k}\binom{n_{.j}}{2}}, \text{ où } \binom{n}{2} = \frac{n(n-1)}{2} \tag{3}$$

L'indice corrigé de Rand assume des valeurs contenues dans l'intervalle $[-1, +1]$ et ainsi comme la F-mesure, les valeurs proches de 1 ou -1 correspondent à des partitions très semblables, alors que les valeurs proches de 0 correspondent à des partitions très différentes. En résumé, la F-mesure est plus facile à interpréter car son analyse peut être faite classe par classe, alors que l'indice corrigé de Rand fournit une mesure globale basée sur tout l'ensemble de clusters dans les partitions.

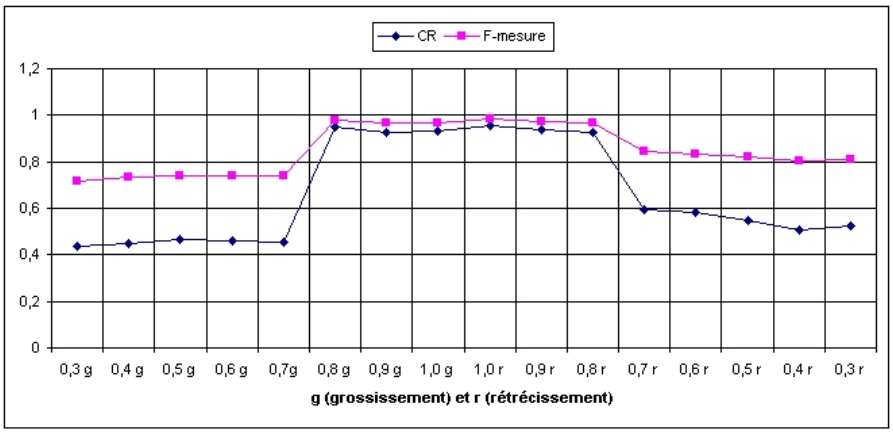


FIG. 3 – *Valeurs de la F-mesure et de l'indice corrigé de Rand pour les classes bien separées.*

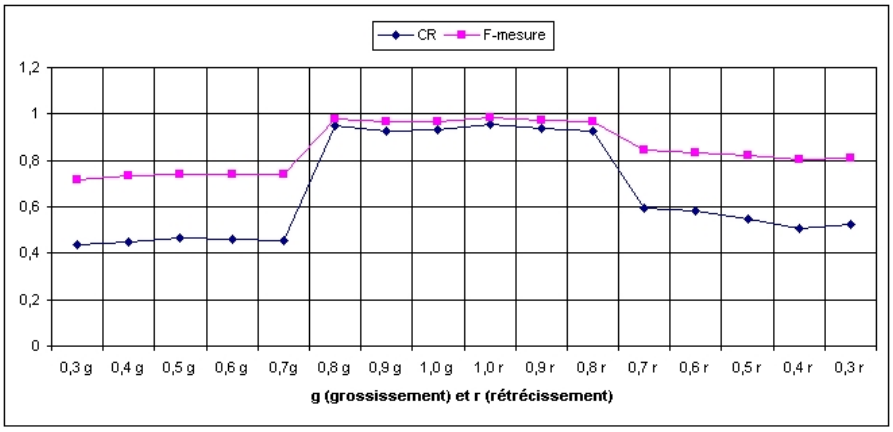


FIG. 4 – *Valeurs de la F-mesure et de l'indice corrigé de Rand pour les classes recouvrantes.*

5 Analyse des résultats

5.1 Premier cas : les classes sont bien séparées

Au départ, les effectifs des classes sont équiprobables et égales à 0.2, ce qui donne un total de 2.000 individus dans chacune des classes et un total de 10.000 individus dans la fenêtre. Une projection sur le premier plan factoriel des individus de toutes les classes est illustrée dans la figure 2. La courbe d'évolution des deux indices d'évaluation est détaillée dans la figure 3. Les indices obtenus varient selon la valeur des facteurs de rétrécissement et grossissement. Notons que plus petite est la valeur des facteurs, plus petites sont les valeurs obtenues par les indices d'évaluation. En d'autres termes, les changements plus importants sont provoqués par les valeurs les plus petites des facteurs de changement. Notons également que l'indice corrigé de Rand est plus sensible aux changements ayant toujours des valeurs inférieures à celles de la F-mesure. Ceci est dû au fait que l'indice corrigé de Rand utilise une analyse globale des deux partitions comparées, alors que la F-mesure réalise une analyse classe par classe. Les changements deviennent plus remarquables quand les facteurs de changement assument des valeurs inférieures à 0.8. Un autre fait important à remarquer est que le grossissement de la classe cible implique plus de changement au niveau de la partition que le rétrécissement de la même classe.

5.2 Deuxième cas : les classes sont recouvrantes

Pour la génération de cinq classes recouvrantes, nous avons utilisé des distributions uniformes comme profils initiaux des classes *a priori*. L'aperçu des individus est donné par la figure 2. Les valeurs obtenues par les indices d'évaluation son également illustrées dans la figure 4. Nous pouvons remarquer que le niveau des changements obtenus par les deux indices d'évaluation est moins important que celui issu de l'analyse sur des classes bien séparées. Cela est dû au fait que les classes

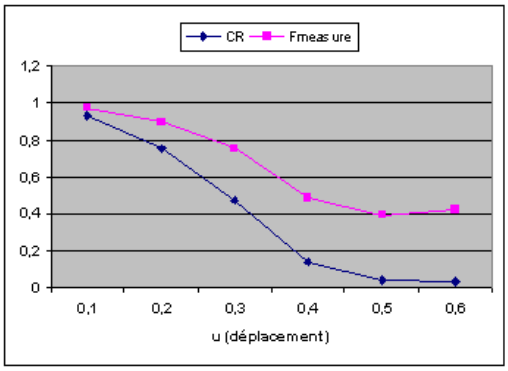


FIG. 5 – *Valeurs de la F-mesure et de l'indice corrigé de Rand pour les classes déplaçantes.*

ont beaucoup de zones de chevauchement. Pour le rétrécissement de la classe cible, les changements sont remarqués pour des valeurs de μ_1 plus petites que 1.0, alors que les changements au niveau du grossissement sont importants pour des valeurs de μ_2 plus petites que 0.5.

5.3 Troisième cas : les classes se déplacent

Pour l'analyse de l'efficacité de notre approche sur la détection de déplacement des classes artificielles, nous avons utilisé les mêmes classes *a priori* bien séparées (cf. figure 2). En variant les valeurs du facteur de déplacement μ_3. Les valeurs obtenues par les indices d'évaluation sont montrées sur la figure 5. Comme nous pouvons nettement remarquer, les valeurs obtenues par les deux indices décroissent quand les classes se rapprochent de la classe cible, car le niveau de déplacement devient plus accentué au regard du positionnement initiaux des classes.

6 Conclusion et perspectives

Dans cet article nous avons présenté une méthodologie pour la génération des données artificielles ayant pour but la simulation de l'usage du Web. La motivation majeure pour la mise en place de cette stratégie est la pénurie de *benchmarks* dans le domaine de la fouille d'usage du Web. Notre proposition présente un algorithme de création de données artificielles ainsi que la simulation de changements liés à l'effectif et au déplacement des classes artificielles. Une autre contribution de cet article est la description d'une approche pour la détection des changements sur les données d'usage. A partir de trois études de cas sur des données simulées, nous avons pu constater l'efficacité de cette approche. Comme travaux futurs, nous pouvons citer l'analyse plus approfondie de l'approche proposée (temps d'exécution, complexité, etc.).

Références

Cooley, R., B. Mobasher, et J. Srivastava (1999). Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems 1*(1), 5–32.

Da Silva, A. et Y. Lechevallier (2008). Stratégies de classification non supervisée sur fenêtres superposées : application aux données d'usage du web. In *Actes des 8ème journées Extraction et Gestion des Connaissances (EGC 2008), Revue des Nouvelles Technologies de l'Information (RNTI)*, Volume I, pp. 219–220. cépaduès.

Da Silva, A., Y. Lechevallier, F. Rossi, et F. De Carvalho (2007). Construction et analyse de résumés de données évolutives : application aux données d'usage du web. In *Actes des 8ème journées Extraction et Gestion des Connaissances (EGC 2007), Revue des Nouvelles Technologies de l'Information (RNTI)*, pp. 539–544. cépaduès.

Hubert, L. et P. Arabie (1985). Comparing partitions. *Journal of Classification 2*, 193–218.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics and Probability*, Volume 1, pp. 281–297.

Spiliopoulou, M. (1999). Data mining for the web. *Workshop on Machine Learning in User Modelling of the ACAI99*, 588–589.

van Rijsbergen, C. J. (1979). *Information Retrieval* (second ed.). London : Butterworths.

Summary

In the data stream domain, taking into account the time factor has become a necessity since the subjacent distribution of the data can evolve over time. A typical example concerns the Web surfer usage profiles. Our aim is to analyze the evolution of these profiles which can be related to the change in the cluster elements or to the displacement of clusters over time. In order to validate the changing indicators, we set up a methodology to simulate usage data which makes it possible to control the occurrence of the changes.

Structure statistique de récupération de données multiphase : Application possible aux flux de données issues des compteurs électriques communicants

Alain Dessertaine

EDF R&D - ICAME
1 avenue du Général De Gaulle
92141 CLAMART Cedex
alain.dessertaine@edf.fr

Résumé. Cette communication a pour vocation de faire un état de l'art des outils statistiques issus de la théorie des sondages en vue de leur application et adaptation dans le cadre de la récupération de données temporelles issues des flux de données en provenance des futurs 34 millions de compteurs communicants installés par EDF auprès de tous ses clients. Une structure statistique de récupération de données multiphase est proposée, basée sur des plans de sondages à plusieurs phases, afin de permettre l'utilisation adaptée des techniques récentes d'échantillonnage équilibré et de redressements à base de calage sur marges ou sur fonction de répartition.

1 Introduction – Pourquoi une stratégie multiphasée ?

Il est prévu l'installation d'ici 2016 d'environ 34 millions de compteurs communicants permettant, entre autre, de récupérer des courbes de consommation électrique sur l'ensemble des clients de EDF avec des granularités temporelles potentiellement de l'ordre de la minute, voire de la seconde. Des travaux de recherche sont en cours afin d'élaborer un système d'information permettant de gérer et d'utiliser au mieux cette information. Ceux-ci abordent une utilisation conjointe de techniques de résumés et d'analyses de flux de données et de techniques et outils d'échantillonnage et d'estimation abordés et développés dans le cadre de la théorie des sondages.

La grande volumétrie des données que nous aborderons impliquera des contraintes très fortes en terme de transmission et de stockage des données. Des réflexions sont donc nécessaires pour définir nos besoins en terme de données à exploiter pour élaborer des systèmes d'information de la consommation pertinents.

Des travaux sont en cours afin de proposer des stratégies d'échantillonnage spatio-temporelles des flux utilisés ; nous allons nous baser sur ces premiers travaux pour prendre en compte certaines contraintes préalables à nos futurs traitements.

Aussi, nous pouvons au jour d'aujourd'hui dire que nous ne pourrons pas conserver (ni récupérer) l'ensemble des courbes de consommation au pas seconde ou au pas minute sur l'ensemble de nos clients. Nous ferons comme hypothèse de travail que nous gèrerons une sorte de panel avec un grand nombre de clients, et que nous récupèrerons des résumés spécifiques de leur(s) courbe(s) de consommation, en fonction de la richesse même de leur propre process de consommation.

Nous pouvons aussi faire comme hypothèse que les besoins des utilisateurs nécessiteront des degrés différents de finesse des données ; ainsi, nous pouvons imaginer une structure permettant l'élaboration de différents flux de sorties d'information avec des résumés pouvant se limiter à de simples requêtes jusqu'à l'élaboration et la gestion d'échantillons de courbes. Cette structure peut se résumer par ce schéma :

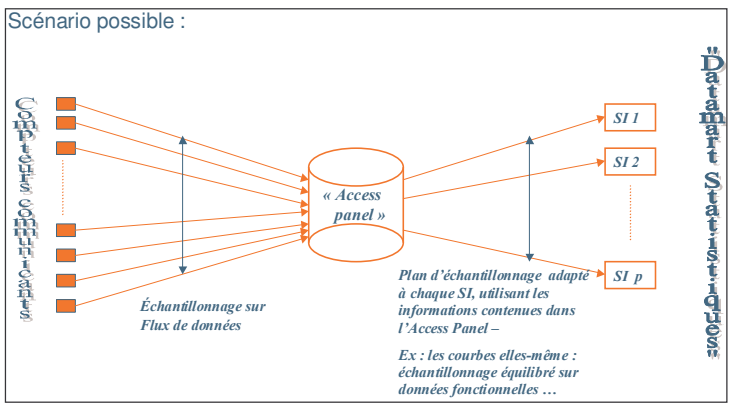


FIG. 1 – *Schéma de récupération de flux de données à plusieurs phases.*

Ainsi, certains flux pourront être échantillonnés en amont d'une base de courbes de charge (base éventuellement temporaire) pour limiter les transmissions (via une gestion anticiper des dates de transmission des données, par exemple), voire pour limiter le stockage éventuel (surtout dans le cadre de données à granularité temporelle très fine). Puis, en aval, ou à partir de cette base de courbes de consommation, nommé « Access panel » sur le schéma de la figure 1, des résumés spécifiques à partir de requêtes ou d'échantillonnage spatio-temporel de courbes permettront d'alimenter des sorties, que nous nommerons par défaut de langage « Datamart Statistiques ». Parmi ceux-ci, nous pourrons avoir, par exemple, un Datamart spécifiquement élaborés pour la prévision, utilisant le principe de la prévision par agrégation/désagégation présenté dans Dessertaine (2008) ; dans ce cadre, il sera nécessaire de conserver des échantillons de courbes avec des historiques suffisamment long afin de construire des modèles de prévisions de consommation électrique sur des classes de clients spécifiquement élaborées à partir de leur courbe de consommation.

Dans le schéma de la figure 1, nous voyons qu'il sera nécessaire d'effectuer des échantillonnages pertinents de manière successive. Cette succession d'échantillonnage est bien connue du monde des sondeurs, sous le nom de plan de sondage à plusieurs phases.

Aussi, dans cette communication, nous allons recenser un certain nombre d'outils issus de la théorie des sondages qui pourront nous être utiles pour traiter les flux de données issus des compteurs. Nous proposons de regarder plus particulièrement les plans sondages à plusieurs phases, les plans à probabilités inégales et les plans équilibrés. Nous illustrerons ces techniques de leurs applications probables dans le cadre de notre problématique.

Voici les notations que nous allons utiliser dans la suite de ce document. Pour chaque individu i d'une population U (la population des clients de EDF par exemple), chaque capteur (compteur communicant) mesure à chaque instant t la valeur instantanée d'une variable C, qui prendra alors la valeur $c_{i,t}$. Nous pourrons aussi nous intéresser à une variable de type courbe mesurée sur un intervalle [T, T+P] qui prendra alors la forme fonctionnelle $c_i(t)$. Nous définissons alors un plan d'échantillonnage (ou un plan de sondage) permettant d'échantillonner un échantillon S de n d'individus i parmi les N de la population U avec une probabilité de sélection π_i déterminée au préalable par le statisticien, avec un algorithme d'échantillonnage respectant ces probabilités et une certaine stratégie d'échantillonnage utilisée dans l'espoir de maîtriser « au mieux » les erreurs d'échantillonnage, comme les plans stratifiées, les plans à plusieurs degrés, les plans à probabilités inégales par exemple – pour une vision large de ces algorithmes, voir le livre de Yves Tillé (2006). Cette stratégie génère une probabilité d'inclusion double, dans la majorité des cas incalculable, mais utilisée dans l'expression de la variance d'échantillonnage des estimateurs utilisés. Parmi ceux-là, l'estimateur d'Horvitz-Thomson est l'estimateur de base utilisé en sondages. Voici, par exemple, l'estimation du total C_t de la variable C à l'instant t avec l'échantillon S :

$$\hat{C}_t = \sum_{i \in S} \frac{C_{i,t}}{\pi_i} \text{ de variance, } Var\left(\hat{C}_t\right) = \sum_{i \in P} \sum_{j \in P} \frac{c_{i,t}}{\pi_i} \frac{c_{j,t}}{\pi_j} \left(\pi_{ij} - \pi_i \pi_j\right) \text{ estimée par :}$$

$$\hat{V}\left(\hat{C}_t\right) = \sum_{i \in P} \sum_{j \in P} \frac{c_{i,t}}{\pi_i} \frac{c_{j,t}}{\pi_j} \frac{\left(\pi_{ij} - \pi_i \pi_j\right)}{\pi_{ij}}$$

Dans le cadre de l'estimation d'une estimation d'un objet courbe avec un échantillon de courbes, il suffit de remplacer dans les expressions ci-dessus $c_{i,t}$ par $c_i(t)$:

$$\hat{C}(t) = \sum_{i \in S} \frac{C_i(t)}{\pi_i}$$

Remarquons d'ores et déjà que nous n'aborderons pas dans cette communication le cas où l'intervalle [T, T+P] est suffisamment grand pour prendre en compte le caractère potentiellement variable de la population, ou les mises à jours de l'échantillon, impliquant un poids variable dans le temps.

Nous appellerons variables auxiliaires toutes variables X connues sur l'échantillon S et, conjointement, sur la population soit pour chaque individu de la population (utiles pour l'élaboration de plan de sondages performants ou de certains redressements) ou des résumés globaux sur la population (comme les totaux utiles pour les calages sur marges). L'utilisation de ces informations supplémentaires sera d'autant plus pertinente que ces variables seront bien corrélées avec notre variable d'intérêt C.

2 Plan de sondage à plusieurs phases :

La stratégie multiphase présentée en introduction de ce document nous oblige à présenter de manière plus détaillée les plans à plusieurs phases. Pour faciliter la compréhension, nous nous limiterons aux plans à deux phases. Après une présentation formelle de ce plan et son utilisation dans la pratique générale, nous décrirons un peu plus en détail son utilité pour notre stratégie d'échantillonnage.

2.1 Définition

Le principe est simple. A partir d'un échantillon S_I (les individus i ayant été échantillonnés avec une certaine probabilité d'inclusion π_{iI}, suivant une certaine stratégie d'échantillonnage), nous échantillonnons de manière « indépendante » un sous échantillon avec une stratégie propre et une nouvelle probabilité d'inclusion : $\pi_{iII} = \pi_{i|S_I}$. Dans ce cas, l'estimation de la somme des consommations C_t s'écrit :

$$\hat{C}_t = \sum_{i \in S_{II}} \frac{C_{i,t}}{\pi_{iI}\pi_{i|S_I}}$$

A ce niveau, il ne faut pas confondre $\pi_{i|S_I}$ et la probabilité $\Pr(i \in S_{II} \mid i \in S_I)$. Le premier terme correspond à la probabilité d'échantillonner i sachant que l'échantillon S_I a été tiré, et le deuxième correspond à la probabilité d'avoir tiré i dans le deuxième échantillon sachant qu'il a été tiré dans le premier. Il est de ce fait important de noter que l'estimateur présenté ci-dessus ne s'agit pas de l'estimateur Horvitz-Thompson, mais nous pouvons démontrer qu'il est sans biais; pour plus de détail, consulter Särndal et al (1992) et Ardilly (2006).

Les stratégies d'échantillonnage à deux (ou plusieurs) phases sont généralement utilisées dans le cas ou nous n'avons pas (ou peu) d'informations auxiliaires pour échantillonner directement avec un plan performant. Ainsi, l'échantillon de 1ère phase (généralement de grande taille) permet de récupérer des informations permettant d'élaborer une stratégie d'échantillonnage performante de deuxième phase[1]. Le « prix à payer » étant l'erreur due à l'échantillonnage de 1ère phase. Il faut donc trouver un compromis entre la perte de précision due à la première phase, et le gain obtenu par l'intermédiaire de la deuxième phase.

2.2 Application aux données de compteurs

Dans le cas de la récupération de flux de données issus des compteurs communicants, plusieurs scénarios peuvent être envisagés :

- Les contraintes techniques liées à la transmission de données et à leur stockage seront sans doute fédérateurs de récupération partielle de celles-ci, surtout si les données récupérables sont de granularités temporelles très fines (une donnée de puissance appelée toute les minutes, voire toutes les secondes). Un échantillonnage de type spatio-temporel pourra alors à cette phase être utile. De tels échantillonnages sont évoqués dans Chiky et al (2008), et dans Dessertaine (2007) pour les impacts sur les estimateurs.
- Certains algorithmes, analyses ou modèles utilisés sur les datamarts statistiques définis plus haut pourront être particulièrement lourds à mettre en place sur l'ensemble des données (ou sur un échantillon trop conséquent). Aussi, un bon échantillon de faible taille pourra être préféré. Il sera « bon » si il brasse suffisamment l'ensemble de la population de manière aléatoire, et si il est performant en terme de précision. A ce niveau, nous pourrons appliquer des échantillons équilibrés, tel que présenté dans Deville et al (2004) ou Chauvet et al (2006), en deuxième phase d'un échantillon de type spatio-temporel (voir ci-après).
- Dans le cas d'estimation en temps réel de certains indicateurs (ou en temps quasi-réel), nous pourrons gérer de manière anticiper les transmissions de données pour ne pas engorger le système. Ainsi, même si nous travaillons sur un échantillon de compteurs, nous pourrons sur cet échantillon élaborer un sous échantillon permettant de récupérer prioritairement des données et estimer avec une bonne qualité la consommation globale de notre clientèle en temps réel.

[1] Ainsi, certains instituts de sondages proposent à leurs clients des enquêtes à partir d'échantillons choisis sur un panel beaucoup plus large, enquêté régulièrement et, de ce fait, bien connue. Cette stratégie permet de réduire sensiblement les coûts d'observations en travaillant sur des échantillons de « petites » tailles, mais généralement performants car précis.

3 Echantillons à probabilités inégales et échantillons équilibrés

Les deux derniers scénarios présentés ci-dessus pourront utiliser deux techniques déjà développés et largement utilisés dans la pratique des sondages, à savoir les échantillons à probabilités inégales et les échantillons équilibrés. Nous allons succinctement les présenter, et nous regarderons plus précisément comment les utiliser dans notre cadre d'application.

3.1 Echantillon à probabilités inégales

Les échantillons à probabilités inégales sont le prolongement des approches par stratification avec allocation de Neyman. Intuitivement, nous sentons qu'il serait souhaitable d'observer presque sûrement les individus qui contribuent fortement à la variance d'échantillonnage par le fait qu'ils possèdent une forte valeur pour la variable d'intérêt. Cette intuition est vérifiée uniquement dans le cas où la taille de l'échantillon n est fixe, et non aléatoire. En effet, dans ce cas, nous pouvons démontrer que la variance de notre estimateur devient :

$$Var\left(\hat{C}_t\right) = -\frac{1}{2}\sum_{i\in P}\sum_{j\in P}\left(\frac{c_{i,t}}{\pi_i} - \frac{c_{j,t}}{\pi_j}\right)^2 \left(\pi_{ij} - \pi_i\pi_j\right)$$

$$\text{estimée par : } \hat{V}\left(\hat{C}_t\right) = -\frac{1}{2}\sum_{i\in P}\sum_{j\in P}\left(\frac{c_{i,t}}{\pi_i} - \frac{c_{j,t}}{\pi_j}\right)^2 \frac{\left(\pi_{ij} - \pi_i\pi_j\right)}{\pi_{ij}}$$

Ainsi, il « suffit » d'élaborer un plan de sondage de taille fixe de manière à ce que les probabilités d'inclusion π_i soit quasiment proportionnelles aux valeurs $c_{t,i}$ pour s'assurer une faible variance d'estimation.

Dans le cadre de la gestion anticiper des transmissions des données des compteurs, nous pourrons construire une probabilité d'inclusion proportionnelle à la moyenne des consommations passées. Ces moyennes seront mises à jour compteur par compteur, en utilisant les flux de données. Remarquons que nous pouvons de fait mettre à jour une fonction de répartition de la variable décrivant la consommation moyenne des clients de EDF. Nous verrons dans la conclusion de ce document que cette fonction de répartition pourra être utilisée pour redresser notre échantillon dans le cas de panne ou de retard engendrant la non-disponibilité des données nécessaires pour les estimations en temps réel de la consommation globale.

3.2 Echantillons équilibrés

Un échantillon à probabilités inégales (ou non) est dit équilibré sur une variable X, si l'estimateur d'horvitz-Thompson du total de X sur la population U est strictement égal au vrai total connu sur U! Ainsi, nous avons les égalités suivantes :

$$\hat{X} = \sum_{k\in s}\frac{x_k}{\pi_k} = \sum_{i\in U} x_i \quad \text{et surtout} \quad Var(\hat{X}) = Var(\sum_{k\in s}\frac{x_k}{\pi_k}) = Var(\sum_{i\in U} x_i) = 0$$

Les échantillons équilibrés sont des échantillons sous contraintes. Seuls les échantillons respectant ces contraintes peuvent être tirés. Dans l'espace des échantillons (de dimension N), les échantillons vérifiant la contrainte d'équilibrage appartiennent à l'hyperplan des contraintes. Cette propriété géométrique a permis l'élaboration d'un algorithme spécifique, nommé la méthode du cube (Chauvet et al, 2004) qui permet de sélectionner un échantillon parfaitement équilibré si c'est possible, sinon de sélectionner un échantillon le plus équilibré possible si l'équilibre parfait est impossible, et dans tous les cas respecter les probabilités d'inclusion qui sont fixées « à priori » par le statisticien.

Un tel échantillon peut s'appliquer dans le cas d'un échantillon de seconde phase (à partir de « l'access panel » défini plus haut). En effet, dans la majorité des datamarts statistiques à développer, l'équilibrage sur les phénomènes temporels de la consommation électrique observés sera souhaitable pour conserver un échantillon pertinent. Ainsi, nous pourrons élaborer un échantillon de manière à ce que l'estimation de la courbe C(t) vérifie :

$$\forall t \in [T, T+P], \hat{C}(t) = \sum_{i\in S_{II}} \frac{C_i(t)}{\pi_{iI}\pi_{i|S_I}} \approx \sum_{i\in S_I} \frac{C_i(t)}{\pi_{iI}}$$

Un travail présenté dans Dessertaine (2007) a montré l'intérêt de transformer en ondelettes les courbes $C_i(t)$ et la courbe C(t) de manière à équilibrer notre échantillon sur les coefficients d'approximation de ces transformations. Dans cette communication, un exemple permettait de juger de la pertinence de cette approche. Ainsi à partir d'une popula-

tion élaborée sur un jeu d'essai de 2300 courbes mesurées demi-heure par demi-heure sur une période de 1 mois autour des congés de noël 2000, les précisions des échantillons équilibrés étaient toujours meilleures que celles obtenues avec un échantillon à probabilité inégale de même taille :

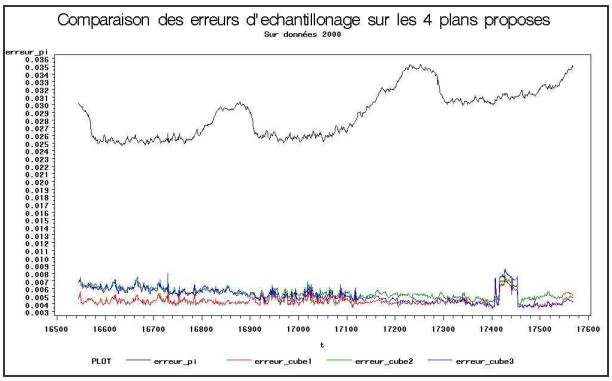


FIG. 2 – *Comparaison de plusieurs stratégies d'échantillonnage à probabilités inégale et échantillons équilibrés*

Les gains observés ici sont particulièrement importants sur l'ensemble de la période d'estimation ; ainsi, les erreurs d'échantillonnage sur le plan à probabilités inégales varient autour de 2,9% (sur une plage allant de 2,5% à 3,4%) alors que la stratégie nommée « Cube1 », correspondant à un équilibrage sur les 64 coefficients d'approximations de niveau 5, suite à une décomposition en ondelettes de Haar, varient autour de 0,44% (variant de 0,4 à 0,5%), soit une diminution de l'ordre de 80% de l'erreur d'échantillonnage.

4 Conclusion

Ce document a permis de présenter un certain nombre de techniques utilisées dans la pratique des sondages aléatoires, et a montré l'intérêt de leur utilisation potentielle dans le cadre d'une structure statistique multiphasée de récupération de données issues des futurs 34 millions de compteurs communicants installés chez tous les clients de EDF.

Il fait l'impasse sur des techniques de redressements pourtant essentielles pour les raisons suivantes :

- la présence de non-réponse partielle ou totale pour certains individus.
  - la non-réponse totale se traitant par des techniques de repondération, comme le calage, développé dans (Deville et al, 1992) et le calage sur fonction de répartition (Ren, 2000).
  - la non-réponse partielle s'abordant soit par des techniques de repondération, mais plus généralement par des techniques d'imputations (remplacement de la valeur manquante par une valeur la plus probable compte-tenu du contexte)
- La présence de valeurs remarquables ou aberrantes perturbant grandement la robustesse de nos estimations
- L'enrichissement de données auxiliaires après la phase d'échantillonnage

Mais le cadre statistique présenté dans ce papier permettra d'adapter ces techniques à notre système. Ainsi, par exemple, nous pourrons utiliser un calage sur la fonction de répartition (dont la construction a été succinctement évoquée dans le paragraphe 3.1) pour redresser notre échantillon dans le cas de panne ou de retard engendrant la non-disponibilité des données nécessaires pour les estimations en temps réel de la consommation globale. Des calages sur résumés en ondelettes de courbes mesurées sur des périodes précédentes pourront aussi être utilisés dans ce cas, comme dans Dessertaine (2006).

Références

Ardilly, P. (2006). *Les techniques de sondage*, Paris : Editions Technip

Chauvet, G. and Tillé, Y. (2006). *A fast algorithm of balanced sampling*, Journal of Computational Statistics.

Chiky, R., Cubillé, J., Dessertaine, A., Hebrail, G. et Picard, ML. (2008), *Gestion de panel dans un environnement de flux de données : Etat de l'art*, dans la Revue des Nouvelles Technologies de l'Information – actes EGC 2008

Dessertaine, A. (2006). *Sondages et séries temporelles : une application pour la prévision de la consommation électrique*, Actes des journées Françaises de Statistique 2006

Dessertaine A. (2007), *Sampling and Data-Stream: Some ideas to built balanced sampling using auxiliary hilbertian informations*, 56th International Statistical Institute Conference : IPM56 - New methods of sampling – 22-29 Août 2007 Lisbonne (Portugal).

Dessertaine, A. (2008). *Non-parametric load forecasting by «aggregation / disaggregation»*, communication à l'International Symposium of Forecasting, Nice, 22-25 Juin, 2008.

Dessertaine, A. (2008). *Echantillonnage et Flux de données : Estimation de courbes de consommation électrique à partir de données fonctionnelles*, publié dans «Méthodes de sondage : applications aux enquêtes longitudinales, à la santé, aux enquêtes électorales et aux enquêtes dans les pays en développement», Dunod

Deville, JC. et Särndal, CE. (1992). *Calibration estimators in survey sampling*, Journal of the American Statistical Association; 87 : 376-382

Deville, JC. et Tillé, Y. (2004). *Efficient balanced sampling : the cube method*, Biometrika, 91, 893-912.

Misiti, M., Misiti, Y., Oppenheim, G. et Poggi, JM. (1998). *Méthodes d'ondelettes en statistique : introduction et exemples*, Journal de la Société Française de Statistique Vol. 139, 4, 3-29.

Ren, R. (2000).*Utilisation d'information auxiliaire par calage sur function de repartition*, these de doctorat – Université Paris-Dauphine

Särndal, CE., Swensson, B. et Wretman, J. (1992). *Model Assistd survey Sampling*, New-York : Springer-Verlag

Tillé, Y. (2006). *Sampling Alorithms*, New-York : Springer-Verlag

Contrôle des observations pour la gestion des systèmes de flux de données.

Christophe Dousson*, Pierre Le Maigat*, Fabrice Clérot*

*Orange Labs – 2, avenue Pierre Marzin – 22300 Lannion
{christophe.dousson, pierre.lemaigat, fabrice.clerot}@orange-ftgroup.com

Résumé. Les systèmes d'analyse de flux de données prennent de plus en plus d'importance dans un contexte où les données circulant sur les réseaux sont de plus en plus volumineuses et où la volonté de réagir au plus vite, en temps réel, devient un besoin nécessaire. Afin de permettre des analyses aussi rapides et efficaces que possible, il convient de pouvoir contrôler les flots de données et de focaliser les traitements sur les données pertinentes. Le protocole présenté dans ce papier donne au module de traitement des capacités d'action et de contrôle sur les observations remontantes en fonction de l'état de l'analyse. La diminution des flux résultant de telles focalisations permet des traitements beaucoup plus efficaces, plus pertinents et moins consommateurs de ressources.

1 Introduction

Les modèles de traitement à la volée s'appuyant sur des flux de données prennent une place de plus en plus large dans l'analyse des données. La communauté des bases de données montre un réel engouement[1] pour ces approches à base de DSMS (Data Stream Management System). Pour autant, si tout le monde s'accorde sur les difficultés dues aux énormes volumes de données à analyser et donc sur la nécessité de maîtriser de tels débits, peu de réflexions sur le contrôle même de ces flux sont mises en avant. Citons toutefois (Portet et al., 2006) qui proposent de piloter des modules de traitement du signal afin de générer uniquement les événements utiles à l'analyse en cours (en l'occurrence, la détection d'arythmies cardiaques). Et, dans un autre registre, (Guillou et al., 2008) met également en place un contrôle des observations dans son architecture de supervision hiérarchique de Web Services : le premier niveau de superviseurs n'envoie ses informations que sur requête du diagnostiqueur central.

Nous proposons donc ici un protocole permettant de propager des informations de contrôle du plus haut-niveau de l'analyse jusqu'aux sources d'événements. L'architecture mise en œuvre, baptisée TESS (pour *Timestamped Event Stream System*) est de type « workflow » où les événements transitent de module en module par des « liens ». Ces liens sont orientés d'une interface dite « Producteur » vers une interface dite « Consommateur » (voir figure 1).

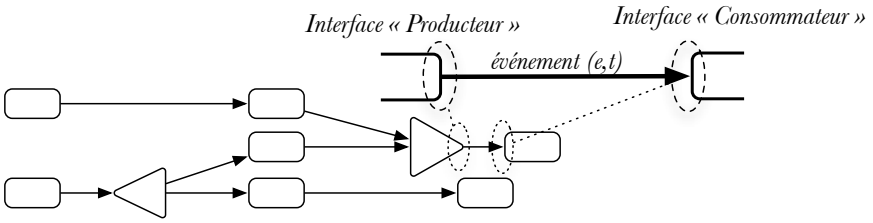


FIG. 1 – *Architecture générale*

La majorité des modules de TESS met en œuvre les deux types d'interfaces ce qui permet d'enchaîner les traitements simplement en établissant des connexions entre celles-ci. Les composants d'entrée (n'offrant que l'interface de producteur) sont appelés des *sources* ; ils peuvent représenter des capteurs, d'autres applications de traitement amont voire des bases de données. Les composants de sortie (avec la seule interface de consommateur) sont des *puits* ; ils peuvent correspondre à des bases de données, à des rapports vers un opérateur, etc.

Les données à analyser par cette structure sont véhiculées dans des événements qui traverseront donc différents modules, en suivant les liens. Cette architecture s'appuie sur les hypothèses suivantes :

– les événements sont tous instantanés (les informations avec durée pourront être modélisées avec un événement de début et un autre de fin),
– les événements sont tous datés (avec une date ponctuelle) et seront donc notés (e, t),

[1] Le lecteur pourra se référer au site web du (CEP Group).

– les connexions entre producteur et consommateur sont de type FIFO (en revanche, il n'y a pas de contrainte sur le fonctionnement interne d'un module),
– les envois de messages et d'événements sont tous *asynchrones*.

2 Protocole de contrôle

Un message de contrôle du flux d'événements est un *triplet* constitué *i)* d'un *type* qui correspond à l'action que l'on veut avoir sur le flux, *ii)* d'une *fenêtre temporelle* limitant l'action aux seuls événements dont la date d'occurrence est contenue dans celle-ci et *iii)* d'une *condition atemporelle* limitant l'action aux seuls événements dont les données (atemporelles) vérifient cette condition.

Par exemple, le message `<TYPE>([10,90], ipsrc=10.12.3.*)` s'appliquera aux événements dont la date est comprise entre 10 et 90 et pour lesquels le champ `ipsrc` contient une adresse IP comprise entre `10.12.3.0` et `10.12.3.255`. De la même façon, le message `<TYPE>` $(]-\infty,+\infty[, \top)$ s'appliquera à tous les événements du flux ($\top$ correspond à la condition toujours vérifiée).

2.1 Du consommateur vers le producteur

Par défaut, la fréquence et le moment de diffusion des événements relèvent de la décision du producteur. Ses critères de choix de diffusion peuvent être divers : le caractère d'urgence ou d'importance d'un événement (s'il existe), la fréquence des mesures, la charge CPU, etc.

Les messages de contrôle en provenance d'un consommateur existent pour permettre à un consommateur de prendre la main sur cette diffusion des événements qui le concernent. Ainsi, deux types de messages sont définis ; ceux alertant le producteur de l'urgence de certains messages et ceux prévenant de l'inutilité des messages.

Message FOCUS(tw, C). Un message de type *FOCUS* signifie que tous les événements *présents ou à venir* vérifiant la portée du message (temporelle et atemporelle) doivent être transmis dès que possible au consommateur. Le producteur doit donc tout mettre en œuvre pour remonter ces informations dès que possible (éventuellement, son composant devra transmettre aussi un *FOCUS* vers l'amont).

Message DISCARD(tw, C). Les messages de type *DISCARD* avertissent un producteur que certains événements (présents ou à venir) n'ont plus d'utilité pour la suite de la chaîne de traitement. Ces événements peuvent donc être supprimés sans être transmis.

Message REMOVE_FOCUS(tw, C) (resp. REMOVE_DISCARD(tw,C)). Les messages *REMOVE_FOCUS* (resp. *REMOVE_DISCARD*) permettent de lever l'urgence (resp. le filtre) sur les événements correspondants. À noter qu'un événement de type *DISCARD* (resp. *FOCUS*) supprime également l'effet d'un *FOCUS* (resp. *DISCARD*) antérieur.

La figure 2 de la page ci-contre illustre les effets de l'envoi successifs de messages des types présentés (afin de simplifier le schéma, les conditions des messages de contrôle sont toutes supposées égales à $\top$).

2.2 Du producteur vers le consommateur

En sus des événements remontés par le producteur, des messages de contrôle transitent par le même canal et permettent de renseigner le consommateur sur l'état du flux.

Message NO_MORE_EVENT(tw, C). Ce message est un message de clôture du flux : il signifie que pour la période de temps considérée, toutes les informations ont été transmises. Lorsqu'un producteur envoie un tel message, il s'engage à *ne plus transmettre aucun événement* dans la fenêtre temporelle spécifiée et correspondant à la condition précisée. Ce message permet aussi de simuler l'avancement d'une horloge (émission régulière de `NO_MORE_EVENT` $(]-\infty,t], \top)$; ce cas d'utilisation sera détaillé dans la section 3.

Message MISSING_EVENT(tw, C). L'utilisation d'un tel message permet de prévenir les consommateurs qu'il est possible que certains événements aient été perdus (volontairement ou non) par ou en amont du producteur. *A contrario*, l'absence de messages de ce type signifie qu'il n'y a eu aucune perte sur les observations. Typiquement, ce message pourra être produit par le superviseur d'un capteur peu fiable ; il pourra aussi être émis par un producteur qui aura supprimé des événements (par exemple, un buffer saturé – voir section 4.3).

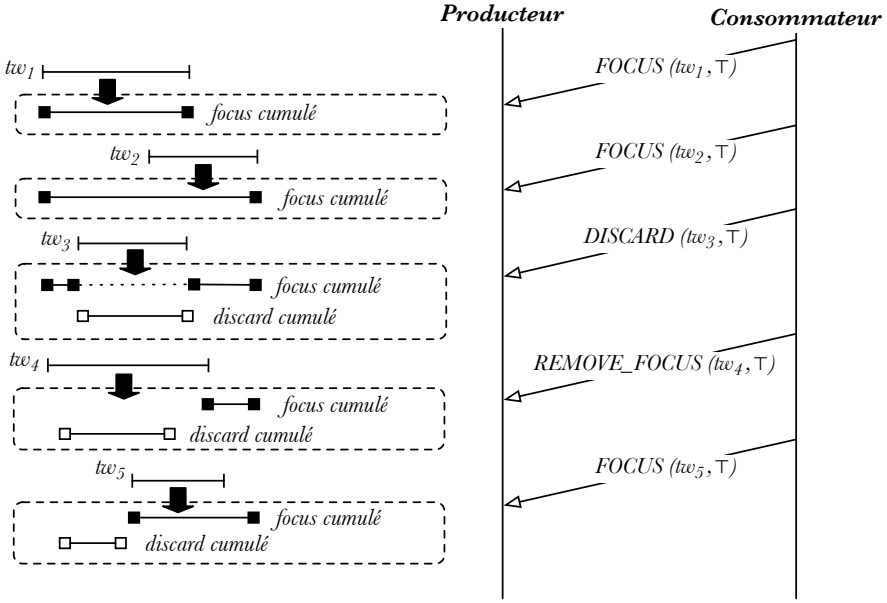


FIG. 2 – *Résultats de l'accumulation de messages de contrôle.*

Message RULE_ON(tw, C). Ce message est émis par un producteur qui souhaite statuer sur certains événements (par exemple, pour s'en débarrasser afin de vider partiellement ses files d'attente). Plutôt que de les envoyer d'office ou de les supprimer et d'envoyer un *MISSING_EVENT*, ce message permet de prévenir les consommateurs que le producteur souhaite légiférer sur la période mise en exergue. En réponse à ce message, le consommateur peut alors décider de recevoir ces messages ou, au contraire, qu'ils soient supprimés sans être transmis (cette réponse s'appuiera sur les messages *DISCARD* ou *FOCUS* détaillés par la suite).

3 La gestion des horloges

La difficulté de gestion des horloges et donc de synchronisation apparaît dès que le système supervisé est réparti : les arrivées des événements ne respectent alors que rarement l'ordre chronologique ce qui nécessitent d'adapter l'algorithmique. Par exemple, (Li et al., 2007) propose l'introduction de piles d'événements dont le but principal est de réordonner les messages de façon chronologique ce qui a, peu ou prou, pour conséquence de faire subir le retard maximum à l'ensemble des événements. À l'opposé, (Srivastava et Widom, 2004) refuse de perturber le flux d'événements et propose d'utiliser des « tics » d'horloge dont la fréquence est déterminée en fonction de caractéristiques du flux évaluées dynamiquement. La difficulté est alors de synchroniser l'ensemble des modules lors d'un traitement distribué du flux.

L'architecture proposée ici ne gère pas d'horloge interne. L'avancement du temps dans le système se traduit par l'insertion de messages de type *NO_MORE_EVENT* dans le flux. L'avantage immédiat est qu'il n'y a aucune synchronisation globale à assurer entre les événements du flux et l'horloge interne du système : en corollaire, le protocole ne fait aucunement l'hypothèse d'un ordonnancement chronologique des événements tout en permettant un pilotage fin de leur arrivée.

Afin d'illustrer l'emploi de ces messages pour la simulation d'horloge, les sections suivantes exposeront successivement le cas où les événements arrivent dans l'ordre chronologique (mode FIFO), puis le cas où les événements peuvent souffrir d'un retard (borné) susceptible de rompre l'ordre chronologique et, enfin, le cas particulier de l'échantillonnage où on connaît le rythme d'arrivée des événements.

Arrivée chronologique des événements (FIFO). Afin de synchroniser l'ensemble du flux sur l'arrivée d'événements en mode FIFO (et chronologique), on intercale dans le flux un composant chargé d'émettre des messages de type *NO_MORE_EVENT* au fur et à mesure de la transmission d'événements.

Algorithme 1 Transmission d'un événement en mode FIFO

Déclencheur : réception d'un événement (e,t)
envoyer l'événement (e, t)
envoyer un message NO_MORE_EVENT($]-\infty, t[, \top$))

Arrivée non chronologique des événements. Supposons que le système de collecte des événements soit réparti et permet ainsi que certains événements se croisent. Avec l'hypothèse que la gigue entre les événements soit bornée par Δ, on peut adapter l'algorithme précédent de la façon suivante :

Algorithme 2 Transmission d'un événement avec gigue bornée par Δ

Déclencheur : réception d'un événement (e,t)
  envoyer l'événement (e, t)
  envoyer un message NO_MORE_EVENT$(]-\infty, t-\Delta[, \top)$)

Échantillonage. On modélise cette fois un processus qui échantillonne régulièrement une donnée (Δ étant l'intervalle de temps d'échantillonnage). Lorsqu'un événement est émis à t, on peut d'ores et déjà informer le flux qu'il n'y aura pas d'autres événements avant $t + \Delta$. Contrairement aux exemples précédents, on ne synchronise pas une horloge mais on se projette dans le futur.

Algorithme 3 Transmission d'un événement échantillonné (Δ étant le pas d'échantillonnage)

Déclencheur : réception d'un événement (e,t)
  envoyer l'événement (e, t)
  envoyer un message NO_MORE_EVENT$(]-\infty, t+\Delta[, \top)$)

Synchronisation multi-sources. Dans la pratique, il est parfois très difficile d'arriver à trouver une borne convenable pour la gigue : si elle est trop juste, des événements vont arriver avec trop de retard et vont donc être en violation du message *NO_MORE_EVENT* précédemment transmis. À l'opposé, si, par précaution, elle est choisi trop lâche, le traitement du flux sera très pénalisé du fait que l'information de clôture du flux arrivera tardivement.

Par exemple, si on a besoin de compter les événements minute par minute, il faudra attendre la durée de la borne après le dernier événement reçu pour être certain de ne pas en avoir manqué : l'information sur le total sera donc émise avec un retard équivalent à cette borne. Si cette valeur de retard est trop prudente et n'est atteinte en réalité que très rarement, le principe de précaution mis en œuvre fera de ce délai théorique une réalité.

En revanche, il est souvent possible d'isoler les sources des événements (les capteurs) et l'hypothèse qu'un capteur envoie ses informations de façon chronologique (FIFO) est rarement mise en défaut. Pour un système de N capteurs K_i, on peut alors utiliser l'algorithme 1 de façon distincte pour chaque capteur (voir algorithme 4).

Algorithme 4 Transmission d'événements issus de capteurs différents (mode FIFO).

Déclencheur : réception d'un événement (e,t) en provenance de K_i
  envoyer l'événement (e, t)
  envoyer un message NO_MORE_EVENT$(]-\infty, t[$, capteur$= K_i)$)

4 Composants d'architecture

Afin d'alléger la présentation, les algorithmes présentés par la suite ne font pas intervenir les conditions de portée des messages (elles sont toutes supposées égales à $\top$). Dans l'implémentation complète, pour chaque i, on gère un ensemble de fenêtres associées chacune à une condition. Ces fenêtres sont mises à jour en respectant les règles d'absorption des sous-conditions : une condition plus générale aura un impact sur toutes les fenêtres de ses sous-conditions, la condition $\top$ étant la plus générale.

4.1 Diffusion de flux

La diffusion d'événements est aisée à mettre en œuvre : lorsqu'un événement arrive en entrée du composant de diffusion, cet événement est transmis sur chacune des sorties (cf. figure 3 de gauche). Ce même mécanisme s'applique également pour la diffusion des messages de contrôle de type *NO_MORE_EVENT*, *RULE_ON* ou *MISSING_EVENT*.

En revanche, pour les messages de contrôle remontant des consommateurs, on distinguera deux familles de messages : ceux nécessitant d'être synchronisés (les « synchrones ») et les autres (les « asynchrones »).

Les « asynchrones » sont les messages de contrôle qui peuvent transiter par le composant de diffusion comme si ce composant n'existait pas : cela signifie que la prise de contrôle du flux par le consommateur ne sera pas préjudiciable aux autres consommateurs partageant le flux par cette diffusion. Seuls les messages *FOCUS* et *REMOVE_DISCARD* sont de ce type car l'effet de ces messages est uniquement l'envoi éventuel de plus d'événements que nécessaire aux autres

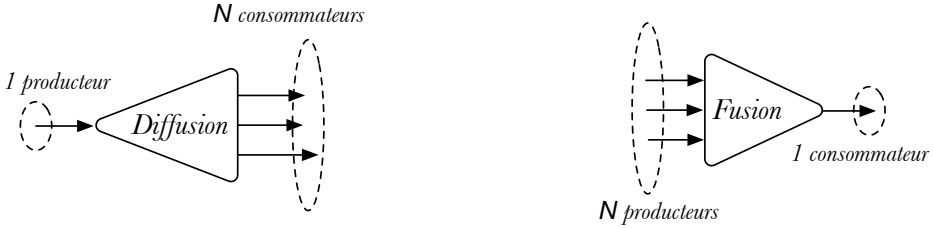


FIG. 3 – *Composants de diffusion et fusion de flux*

consommateurs ; autrement dit, ils ne transgressent pas les messages de contrôle précédemment envoyés par les autres consommateurs et respectent donc la sémantique du protocole.

À l'opposé, les messages « synchrones » que sont *DISCARD* et *REMOVE_FOCUS* doivent être traités de façon plus fine. En effet, si un consommateur envoie un *DISCARD* et que celui-ci remonte en amont du composant de diffusion sans précaution, l'effet potentiel sera la suppression d'événements pour *tous* les consommateurs de la diffusion, y compris ceux toujours en attente de ces événements. Au niveau de chaque sortie i de la diffusion, il faut donc maintenir à jour des fenêtres temporelles D_i (resp RF_i) d'accumulation des *DISCARD* (resp. des *REMOVE_FOCUS*) non encore transmis vers l'amont (cf. algorithme 5, de la présente page).

Algorithme 5 Diffusion de flux du producteur P vers n consommateurs C_i

Déclencheur : réception d'un FOCUS(tw, $\top$) en provenance de C_i
  $RF_i \leftarrow RF_i \setminus tw$
  envoyer le message FOCUS(tw, $\top$) vers P

Déclencheur : réception d'un REMOVE_FOCUS(tw, $\top$) en provenance de C_i
  $RF_i \leftarrow RF_i \cup tw$
  si $\bigcap_{j=0}^{n} RF_j \neq \emptyset$ **alors**
    envoyer un message REMOVE_FOCUS($\bigcap_{j=0}^{n} RF_j, \top$) vers P
    pour tout $k \in [1, n]$, **faire** $RF_k \leftarrow RF_k \setminus \left(\bigcap_{j=0}^{n} RF_j\right)$
  fin si

Déclencheur : réception d'un DISCARD(tw, $\top$) en provenance du C_i
  $D_i \leftarrow D_i \cup tw$, $RF_i \leftarrow RF_i \cup tw$
  si $\bigcap_{j=0}^{n} D_j \neq \emptyset$ **alors**
    envoyer un message DISCARD($\bigcap_{j=0}^{n} D_j, \top$) vers P
    pour tout $k \in [1, n]$, **faire** $D_k \leftarrow D_k \setminus \left(\bigcap_{j=0}^{n} D_j\right)$, $RF_k \leftarrow RF_k \setminus \left(\bigcap_{j=0}^{n} D_j\right)$
  fin si
  si $\bigcap_{j=0}^{n} RF_j \neq \emptyset$ **alors**
    envoyer un message REMOVE_FOCUS($\bigcap_{j=0}^{n} RF_j, \top$) vers P
    pour tout $k \in [1, n]$, **faire** $RF_k \leftarrow RF_k \setminus \left(\bigcap_{j=0}^{n} RF_j\right)$
  fin si

Déclencheur : réception d'un REMOVE_DISCARD(tw, $\top$) en provenance de C_i
  $D_i \leftarrow D_i \setminus tw$
  envoyer le message FOCUS(tw, $\top$) vers P

Déclencheur : réception d'un NO_MORE_EVENT(tw, $\top$) en provenance de P
  pour tout $i \in [1, N]$ **faire**
    $D_i \leftarrow D_i \setminus tw$, $RF_i \leftarrow RF_i \setminus tw$
    envoyer le message NO_MORE_EVENT(tw, $\top$) vers C_i
  fin pour

Déclencheur : réception d'un MISSING_EVENT(tw, $\top$) (resp. RULE_ON) en provenance de P
  pour tout $i \in [1, n]$, **faire** envoyer le message MISSING_EVENT(tw, $\top$) (resp. RULE_ON) vers C_i

Déclencheur : réception d'un événement (e,t) en provenance de P
  pour tout $i \in [1, n]$ **faire si** $t \notin D_i$, **alors** envoyer l'événement (e,t) vers C_i

4.2 Fusion de flux

La fusion de flux est l'opération symétrique de la diffusion (cf. figure 3 de droite), à savoir :

- les messages *DISCARD*, *FOCUS*, *REMOVE_DISCARD* et *REMOVE_FOCUS* remontent immédiatement vers chacun des producteurs en entrée,
- tout événement arrivant sur une entrée est transmis en sortie (i.e. comme un message de contrôle *asynchrone*),
- les messages de contrôle *asynchrones* (*MISSING_EVENT* et *RULE_ON*) sont transmis immédiatement, et
- les messages de type *NO_MORE_EVENT* sont *synchrones* (il faut que chaque producteur informe de la clôture de son flux pour la clôture soit effective en aval) ; une fenêtre temporelle d'accumulation pour les *NO_MORE_EVENT* sera donc maintenue à jour pour les n producteurs de l'entrée de la fusion.

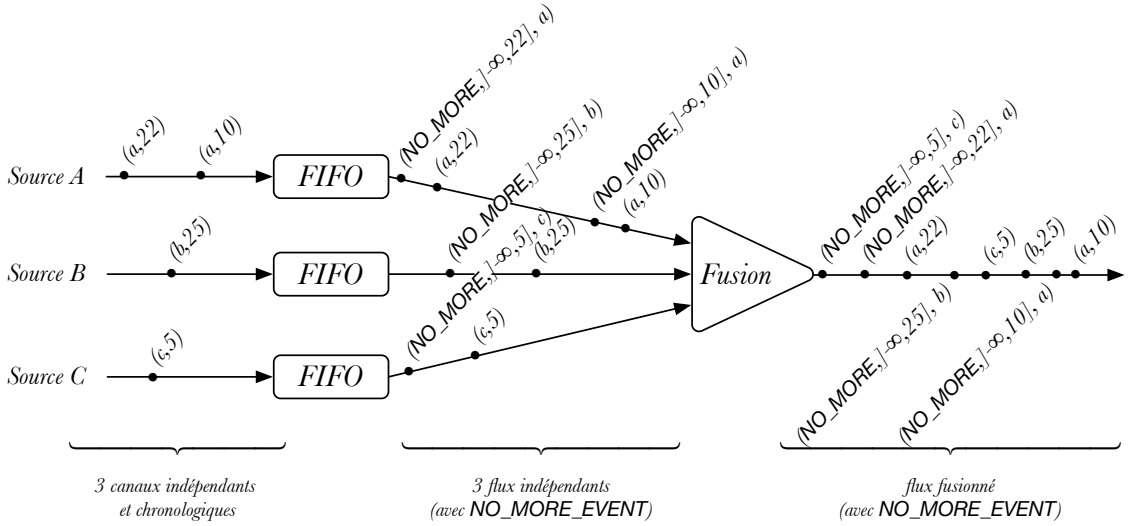


FIG. 4 – *Fusion non-chronologique de plusieurs sources.*

Avec un composant permettant la fusion et plusieurs composant intégrant le mode FIFO (*cf.* l'algorithme 1), on peut construire l'architecture de la figure 4 qui correspond à une gestion distincte des horloges de chaque source. Plutôt qu'une gigue globale, cette architecture évite que le retard des événements d'une source pénalise les autres sources (une gigue globale retarderait l'émission des messages *NO_MORE_EVENT* pour toutes les sources).

L'exemple de la figure 4 illustre très bien le parallèle qui peut-être fait entre l'émission de « tics d'horloge » et les messages *NO_MORE_EVENT*. Mais dès qu'on s'autorise des fenêtres temporelles d'une autre forme que $]-\infty, t]$, ces messages permettent d'autres applications dont une gestion très fine des événements complexes (i.e. des événements créés à partir d'autres), ce qui sera illustré à la section 5.

4.3 Le buffer d'événements

Le buffer d'événements se comporte de la façon suivante : il retient tous les événements qu'il reçoit tant qu'on ne lui réclame pas (par un message de type *FOCUS*). Il convient toutefois de doter un tel composant d'un mécanisme de nettoyage automatique lorsque l'accumulation d'événements devient trop importante.

Par exemple, lorsque le buffer atteint un certain taux de remplissage, il peut avoir une des actions suivantes :

- il transmet d'office certains événements afin de libérer de la place (l'inconvénient est que, si l'objectif de l'utilisation d'un tel buffer était de limiter le débit en sortie, ce comportement peut mettre en péril la suite du traitement),
- il supprime certains événements (les plus vieux, par exemple) quitte à transmettre un message *MISSING_EVENT* si jamais un message *FOCUS* lui parvenait ensuite avec une fenêtre temporelle chevauchant la période des événements supprimés (l'inconvénient ici est la perte d'événements),
- il envoie un message de type *RULE_ON* afin que les consommateurs réagissent par l'envoi de *FOCUS* ou *DISCARD* et lui permette de se nettoyer correctement [2].

5 Génération d'événements complexes

Les traitements prévus dans les architectures de flux visent en partie à générer des événements complexes (i.e. un événement généré à partir de plusieurs autres). Nous allons montrer, ici, comment le protocole mis en avant peut être utilisé lors de cette génération : les messages de contrôle permettent non seulement de synchroniser de façon fiable les différents flux mais aussi d'exploiter des informations supplémentaires sur l'absence d'événements déduits.

[2]Les détails d'une telle implémentation figurent dans (Dousson et Le Maigat, 2007)

5.1 Synchronisation et événements complexes

Comme la génération d'un événement complexe dépend de l'arrivée de plusieurs autres événements, et que, d'autre part, elle peut nécessiter un certain temps de calcul, la date d'émission d'un événement complexe est sujette aux variations de la charge CPU. Lorsque ces mêmes événements complexes doivent servir à d'autres fusion ou jointure, il devient alors compliqué d'écrire des règles de fusion capable de prendre en compte de telles variations.

Même lorsqu'il est possible de pallier ces problèmes (en synchronisant les flux, par exemple, à l'aide de tics d'horloges), certains cycles de déduction peuvent rester inextricables si on ne s'affranchit pas de l'ordre chronologique sur les déductions.

Supposons que nous cherchions à construire l'événement complexe C composé des événements A et B sachant que $t_B - t_A \in [90, 120]$; C étant daté avec la date de B[3]. Supposons que, dans un premier temps, le module de corrélation reçoive alors $(A, 10)$, potentiellement (si un B convenable arrive), cela produira un événement C avec une date d'occurrence incluse dans $\{10\} + [90, 120] = [100, 130]$. Puis, dans un second temps, l'horloge avance jusque $t = 50$ sans autre événement (i.e. réception d'un *NO_MORE_EVENT(*$]-\infty, 50], \top$*)*). B est toujours attendu donc C ne peut pas être généré. Malgré tout, on peut tout de même exploiter l'avancement d'horloge de la façon suivante : on sait qu'il n'est plus possible de générer de C dans $]-\infty, 50] + 90$ hormis s'il résulte de $(A, 10)$ (i.e. hormis $[100, 130]$). On peut donc générer un message de contrôle *NO_MORE_EVENT(*tw, C*)* avec $tw = (]-\infty, 50] + 90) \setminus [100, 130] =]-\infty, 100[\cup]130, 140]$.

La partie du système concernée par ce *NO_MORE_EVENT* pourra alors d'ores et déjà statuer sur tw et en particulier faire lui-même des déductions sur le *futur* ([130,140]) et cela sans attendre qu'un éventuel C soit généré. Ceci peut donner lieu à d'impressionnants gains de performances comme l'illustre (Dousson et Le Maigat, 2007).

5.2 Jointure de flux avec focalisation

On s'intéresse ici à la supervision du trafic du cœur de réseau IP de France Télécom, avec une très grosse volumétrie et des débits conséquents (plusieurs dizaines de milliers d'événements par seconde). L'exemple de jointure de flux choisi correspond à la reconnaissance d'une signature d'une attaque informatique sur des serveurs.

Le mécanisme d'attaque se décompose en deux temps : des serveurs dits « naïfs » sont infectés et envoient des requêtes erronées au serveur cible de l'attaque ; la multiplication des serveurs naïfs provoque un écroulement du serveur cible. L'objectif n'est pas de détecter l'occurrence d'une attaque mais de trouver les serveurs naïfs responsables (à leur insu) de celle-ci ; pour cela, les événements utilisés sont des informations de trafic entre deux adresses IP (*ipsrc* pour la source et *ipdst* pour la destination) : les événements de trafic soutenu mais peu élevés (TWL pour *Traffic Warning Low*) et ceux de trafic important (TWH pour *Traffic Warning High*). Une attaque est détectée par la présence de deux TWH sur le serveur devancés de quelques secondes d'au moins quatre TWL sur les serveurs naïfs.[4]

Pour la suite de la présentation, nous simplifierons cette règle à la présence d'un seul TWL sur le serveur cible précédé d'un seul TWH sur le serveur naïf dans moins de δ unités de temps. Il faut donc corréler un TWH et un TWL de façon à ce que TWL.ipdst = TWH.ipsrc et que, d'un point de vue temporel, TWH.date - TWL.date $\in [0, \delta]$. Notons que comme plusieurs serveurs naïfs participent à une même attaque, il peut y avoir plusieurs TWL à corréler au même TWH.

La difficulté provient de la concomitance de deux facteurs : d'une part, le TWL est très fréquent (même en trafic normal) par rapport au TWH (plus de mille fois plus rare) et, d'autre part, le TWL survient avant le TWH. Faire la jointure des flux TWL et TWH au fur et à mesure conduit à une explosion combinatoire.

Les possibilités de contrôle du flux permettent alors de traiter les événements de façon non chronologique ; en insérant un buffer tel que décrit en section 4.3, les événements TWL vont être retardés jusqu'à être réclamés lors de l'occurrence d'un TWH, et ce dans un intervalle de temps compatible avec la date du TWH : on parle de *focalisation temporelle* dont les tenants et aboutissants son décrits dans (Dousson et Le Maigat, 2007). La récupération de l'adresse *ipsrc* du TWH permettra une focalisation supplémentaire (atemporelle) aux seuls TWL potentiellement utiles (i.e. avec la bonne adresse IP). La figure 5 montre les différents messages de contrôle générés pour parvenir à la corrélation attendue.

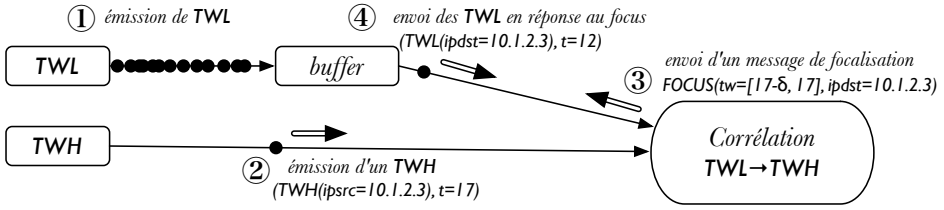


FIG. 5 – *Corrélation avec focalisation.*

[3]Il est possible de dater C par rapport à A (voire de le décaler à $t_A - 10$, par exemple) sans aucunement changer le mécanisme présenté ici.

[4]La pertinence de cette règle tout comme la méthode de génération des TWL et TWH ne sont pas discutées ici.

Le module de corrélation reçoit également des messages *NO_MORE_EVENT* sur son canal *TWH* ; ces messages permettent un nettoyage du buffer dans la mesure où, pour chaque message *NO_MORE_EVENT(tw, TWH)* reçu, il est possible d'envoyer vers le buffer un message *DISCARD(tw*$\ominus[0,\delta]$*, TWL)*[5]. En pratique, pour limiter le nombre de tels messages, ils ne sont envoyés que sur réception d'un message de type *RULE_ON* en provenance du buffer.

Des expérimentations ont été menées sur des journaux d'alarmes de plusieurs millions d'événements, collectés sur le réseau de cœur IP de France Télécom. Dans ces fichiers ont été insérées quelques occurrences d'attaques vers un serveur cible de test. Pour ces cas particuliers d'application, l'accroissement des performances par la focalisation a permis de traiter des débits de plus de trente mille événements par seconde (contre quelques centaines auparavant).

6 Conclusion et perspectives

Le protocole présenté ici permet une grande latitude dans le contrôle des flux de données. Pour la remontée des événements, il permet une grande souplesse dans la gestion de la synchronisation des flux et l'abandon de mécanisme de gestion des horloges. À l'opposé, pour la sélection des données pertinentes, les possibilités de filtrage (avec *DISCARD*) ou de remontée en urgence (avec *FOCUS*) permettent une focalisation du flux sur les données les plus pertinentes pour l'analyse en cours. Une amélioration serait que ce mécanisme soit auto-adaptif : un buffer s'activant sur les événements les plus fréquents et devenant « passant » lorsque la fréquence redevient faible. Dans le même esprit, la gestion de l'avancement du temps et des horloges (présentée en section 3) pourrait bénéficier d'algorithmes statistiques ou d'apprentissage afin d'adapter dynamiquement la borne de retard maximum des événements servant à la génération des *NO_MORE_EVENT*.

L'existence de composants élémentaires comme le buffer tout comme ceux assurant la diffusion des événements et la fusion de flux ou encore ceux responsables de la synchronisation temporelle (avec ou sans gigue) permet à toute nouvelle application de se focaliser juste sur les composants de traitements spécifiques à celle-ci.

Enfin, les premières expérimentations réelles montrent un accroissement net des capacités de traitement lorsque l'analyse se soucie de contrôler le flux des informations qu'elle reçoit. Ces résultats seront à confirmer par la mise en œuvre de ce protocole dans les autres prototypes de supervision et de sécurité développés dans notre laboratoire.

Références

CEP Group. Complex events processing. *http://complexevents.com*.

Dousson, C. et P. Le Maigat (2007). Chronicle recognition improvement using temporal focusing and hierarchization. *Proceedings of the* 20^{th} *IJCAI*, 324–329.

Guillou, X. L., M.-O. Cordier, S. Robin, et L. Rozé (2008). Chronicles for on-line diagnosis of distributed systems. 18^{th} *European Conference on Artificial Intelligence (ECAI'08)*.

Li, M., M. Liu, L. Ding, E. A. Rudensteiner, et M. Mani (2007). Event stream processing with out-of-order data arrival. 27^{th} *International Conference on Distributed Computing Systems Workshops*, 67–74. IEEE Computer Society.

Portet, F., G. Carrault, M. Cordier, et R. Quiniou (2006). Pilotage d'algorithmes pour un diagnostic médical robuste en cardiologie. *RFIA 2006*. Presse universitaires François-Rabelais.

Site, E. W. Event stream intelligence. *http://www.espertech.com*.

Srivastava, U. et J. Widom (2004). Flexible time management in data stream systems. 23^{rd} *Symposium on Principles of Database Systems (PODS'04)*.

Summary

Event Stream Processing (ESP) and Data Stream Management System (DSMS) become more and more popular in a world where huge amount of data flood the networks and where reactivity should as fast as possible. In order to be able to process efficient and quick on-line analysis, it is necessary to allow some control on the event streams and to focus the processing on relevant data. The protocol introduced in this paper is devoted to give control capabilities to the analysis module, depending on its current needs. The reduction of the flow rate induced by this allows more efficient and less resource consuming processing.

[5] L'opérateur de propagation des intervalles est défini comme suit : $[a,b] \ominus [c,d] = [a-d, b-c]$ si $a-d \leq b-c$ et $\emptyset$ sinon.

Extraction et exploitation de données temporelles pour un portail d'e-tourisme

Jérôme Fortin*, Olivier Carloni **, Michel Leclère*, Stéphanie Weiser***

*LIRMM, 161 rue Ada 34392 Montpellier Cedex 5 - France
{fortin,leclere}@lirmm.fr
**Mondeca, 3, cité Nollez 75018 Paris France
olivier.carloni@mondeca.com
***MoDyCo, 200, avenue de la République 92001 Nanterre Cedex - France
sweiser@u-paris10.fr

Résumé. Dans le cadre d'un portail sémantique d'e-tourisme, des données sont collectées depuis des pages web par un processus d'acquisition automatique afin de renseigner une base de connaissances touristique. Une partie cruciale des connaissances à acquérir concerne des informations temporelles sur les dates et horaires d'ouvertures des ressources touristiques. Ces informations d'ouverture sont souvent données de manière incomplète, vague, ambiguë et leur interprétation fait appel à des connaissances implicites du domaine. Dans cet article, nous présentons le processus d'acquisition mis en œuvre en expliquant les problèmes posés par ces informations temporelles. Nous proposons alors d'utiliser une modélisation possibiliste de ces connaissances afin de tenir compte, d'une part, de l'imprécision des connaissances extraites, et d'autre part, de permettre souplesse et flexibilité dans leur exploitation. Nous discutons finalement des raisonnements envisagés, en montrant comment exploiter les informations temporelles ainsi modélisées.

1 Introduction

Le projet Eiffel[1] a pour ambition le développement de portails sémantiques d'e-tourisme permettant à un territoire donné (i.e. un acteur institutionnel d'un bassin touristique) de valoriser son offre touristique en proposant un accès unifié et recomposé aux diverses offres des prestataires de tourisme dépendant de ce territoire. La solution développée se décompose en deux phases (cf. Noël et al. (2008)) : l'une dédiée à l'acquisition de l'offre touristique du territoire dont l'objectif est l'élaboration d'une base de connaissances tourisme/territoire, l'autre dédiée à la présentation de cette offre via un portail sémantique exploitant cette base de connaissances.

La base de connaissances s'appuie sur une ontologie tourisme/territoire adaptée au territoire visé. La phase d'acquisition dispose de diverses modalités de saisie pour "peupler" cette ontologie : saisie manuelle par les acteurs du territoire (par exemple les offices de tourisme), importation automatique de catalogues de grands prestataires touristiques, mais aussi acquisition automatique par exploration du web et construction automatique d'annotations. C'est à cette dernière modalité que nous nous intéressons dans cet article. Elle présente l'intérêt de capter les nombreuses offres temporaires en particulier toutes celles touchant à l'événementiel : concerts, expositions, festivals, foires... En effet, ces offres ne sont en général pas décrites dans les catalogues "classiques" et/ou sont très mal mises à jour. Cette modalité d'acquisition automatique est réalisée en 3 étapes : (1) le web est exploré à la recherche de pages présentant des offres pertinentes pour le territoire visé, (2) une phase d'extraction automatique des caractéristiques de l'offre est alors effectuée sur chaque page identifiée et permet d'obtenir une première annotation "brute" de l'offre touristique, (3) ces annotations sont ensuite analysées, enrichies et validées avant d'être intégrées à la base de connaissances.

Deux caractéristiques sont essentielles lors de la description d'une offre touristique : sa localisation et sa période de validité. Dans cet article, nous nous focalisons sur l'acquisition des caractéristiques temporelles de l'offre. Une première analyse des données extraites des pages web a montré que les expressions linguistiques utilisées pour décrire ces caractéristiques temporelles sont variées, incomplètes, vagues et parfois ambiguës. De plus, l'interprétation de ces expressions nécessite la mobilisation de connaissances implicites liées au domaine (comme par exemple, la notion de haute-saison, les dates des vacances scolaires, les horaires "classiques" d'ouverture de certains types de ressources (les discothèques sont ouvertes la nuit, les musées le jour...)). L'objectif de ce premier travail est de permettre de déterminer automatiquement les périodes d'ouverture d'une offre touristique à partir d'une description textuelle récupérée sur une page web.

Après avoir expliqué les principes de fonctionnement de la phase d'analyse, nous proposons d'utiliser un modèle possibiliste pour les données temporelles afin de prendre en compte l'imprécision des données. Nous montrons alors comment une telle modélisation peut être mise en œuvre dans le cadre du projet Eiffel.

[1]Ce travail a été financé par le projet RNTL-ANR Eiffel (cf. http ://www.projet-eiffel.org).

Dans la Section 2, nous détaillons les caractéristiques des données à analyser et proposons un modèle d'annotation pour les connaissances brutes extraites des expressions linguistiques. En Section 3, nous expliquons les principes de fonctionnement de la phase de construction automatique de ces annotations. La Section 4 présente les objectifs de la phase d'analyse des annotations en exhibant les résultats attendus et en introduisant la méthodologie utilisée. Enfin, la Section 5 propose une modélisation possibiliste des données et discute des avantages de cette approche prometteuse.

2 Caractéristiques des données à traiter

Les expressions temporelles que l'on souhaite repérer et annoter ont une visée informative et pratique. Il ne s'agit pas de dates historiques ou d'expressions descriptives du type *la nuit d'avant* mais d'informations pratiques dans le domaine du tourisme. Il peut ainsi s'agir d'horaires d'ouverture, de dates, de périodes, etc. On peut classer ces expressions en deux catégories principales (cf. Weiser et al. (2008)) : les informations temporelles qui concernent un événement particulier et les informations temporelles répétitives. La première comprend des dates (*concert le 1er octobre*), des périodes (*festival de mai à Juin*), des heures (*le concert commence à 8h*). La seconde comprend des horaires (*le musée ouvre à 10h*), des périodes (*le restaurant est ouvert du lundi au samedi*) et des exceptions (*le camping est ouvert toute l'année sauf en janvier*). Des exemples d'une complexité plus grande peuvent également prendre place dans cette classification comme *de mai à Juin, ouvert tous les jours sauf le mardi*.

Les expressions temporelles touristiques peuvent être génériques ou propres à un type d'objet en particulier. Par exemple une date comme *le 31 octobre 2008* peut aussi bien convenir à un concert, qu'à une représentation de théâtre ou à l'ouverture d'une patinoire. En revanche, une expression comme *ouvert midi et soir sauf le lundi* peut difficilement s'appliquer à autre chose qu'à un restaurant. Ces exemples montrent que la complexité des expressions temporelles varie énormément : certaines expressions sont très simples, d'autres peuvent devenir complexes, jusqu'au point d'être floues ou ambiguës. Par exemple dans *vendredi et samedi soir*, doit-on comprendre qu'il s'agit des deux soirées ou de la journée de vendredi et de la soirée de samedi ? Le contexte est nécessaire pour lever ce type d'ambiguïtés. Pour cet exemple, s'il s'agit d'un concert, la première interprétation sera probablement la bonne.

Certaines expressions présentent donc des difficultés d'interprétation : ambiguïtés virtuelles ou même réelles[2], dates imprécises, etc. Dans d'autres cas, si les expressions sont naturellement interprétées par un internaute, leur interprétation a recours à des connaissances du monde ou nécessite des inférences (par exemple, lorsque l'on a des horaires de fermeture et que l'on peut naturellement en déduire des horaires d'ouverture). Voici un inventaire plus précis de ces difficultés :

- *Fermé le mardi.* Pour une telle expression, le but est de déduire les jours d'ouverture (donc à priori lundi, mercredi, jeudi, vendredi, samedi et dimanche). Si on dispose d'informations complémentaires concernant le type d'objet, on peut éventuellement ajouter des parties de journées : si c'est une boîte de nuit, considérer pour chaque jour d'ouverture qu'il s'agit de la nuit ou s'il s'agit d'un restaurant, déduire des parties de journées comme midi et soir.
- *Visites de Juin à septembre les après-midi, sauf lundi et mardi.* Cette expression est ambiguë : il n'est pas possible de savoir si le lundi et le mardi la ressource est fermée toute la journée ou si, au contraire elle est ouverte le matin. De plus, elle est également floue au niveau des dates : il peut s'agir du 1er Juin ou du premier week-end de Juin ou autre. Et pour septembre, il peut s'agir du début ou de la fin du mois.
- *Fermé durant les vacances scolaires de février. Fermeture hebdomadaire non déterminée.* Avec cette expression, on ne peut déduire aucun jour d'ouverture ou de fermeture. Pour interpréter les périodes de fermeture (et ensuite en déduire les périodes d'ouverture par complémentarité), il faut faire appel à une connaissance du monde : les dates de vacances scolaires. Il est possible d'utiliser pour chaque année un calendrier comprenant les vacances.

Ces différents exemples donnent donc un petit panorama des calculs qui doivent être effectués sur les expressions effectivement repérées dans les pages web de ressources touristiques.

3 Construction de l'annotation

En amont de la phase de construction de l'annotation, une première étape de "crawling" (effectuée par l'un de nos partenaires[3]) se charge de collecter des pages web à caractère touristique (sites d'hôtels, de restaurants, d'événements ponctuels, spectacles, concerts, etc.). Ces pages, au format HTML, sont alors transformées en documents XML, format plus adéquat pour un traitement automatique, et "nettoyées". C'est-à-dire que seules les informations utiles à notre analyse ont été conservées. De nombreuses balises ont donc été supprimées. Il faut noter que, une fois à l'entrée de l'étape d'annotation automatique, chaque page est indépendante : il se peut que plusieurs pages soient issues du même site mais,

[2] On parle d'ambiguïté réelle quand l'expression présente plusieurs interprétations possibles, aussi bien pour un locuteur natif que pour une machine. Une expression virtuellement ambiguë n'est ambiguë que pour une machine.

[3] La société Antidot, partenaire du projet, développe AFS (http ://www.antidot.com) un moteur de recherche adapté à cette tâche.

une fois l'aspiration effectuée, elles ne sont plus liées. Ces pages web sont ensuite analysées de façon à y détecter et à annoter automatiquement les informations utiles à la base de connaissances. Nous nous focalisons ici sur le repérage et l'annotation des informations d'ouverture et de fermeture.

L'objectif de cette première étape est de construire une annotation contenant uniquement les informations explicites contenues dans la page. C'est-à-dire que si la page contient l'expression *"ouvert du lundi au jeudi"*, c'est celle-ci qui sera annotée sans interprétation du type *"ouvert le lundi, le mardi, le mercredi, le jeudi"*. De la même manière, si l'on a *"fermé le mardi"*, cela ne sera pas converti en jours d'ouverture, mais cela sera annoté comme un jour de fermeture. De plus, certaines informations n'ont pas besoin d'être annotées. En effet, seules celles qui peuvent être utiles à l'utilisateur et qui peuvent prendre place dans la base de connaissances doivent être prises en compte. Pour cela, nous nous basons sur l'ontologie tourisme/territoire élaborée pour les besoins du projet.

Comme nous l'avons vu dans la partie précédente, les informations temporelles à annoter sont des périodes d'ouverture ou de fermeture qui peuvent être constituées : d'une date de début et d'une date de fin, d'une date seule, d'une heure de début et d'une heure de fin, de jours. Afin de faciliter ensuite l'intégration des données annotées à la base de connaissances, une DTD[4] définissant le format d'annotation a été établie (une première version est décrite dans Weiser (2008)). Les expressions temporelles peuvent être annotées avec les balises : `période-ouverture`, `période-fermeture`, `exception` et `incertitude`. La balise `exception` permet d'annoter la chaîne textuelle décrivant une exception (comme *sauf le mardi*) et ainsi de garder l'information telle quelle afin de la fournir textuellement à l'utilisateur. La balise `incertitude` permet d'indiquer que le résultat n'est pas fiable : il est flou ou comprend une ambiguïté.

En ce qui concerne les balises `période-ouverture` et `période-fermeture`, elles permettent de définir plus précisément l'expression repérée et peuvent inclure les balises `date`, `date-début`, `date-fin`, `jour`, `heure-début`, `heure-fin` et `partie-de-journée`. La balise `date` sert à annoter les dates seules ; dans la base de connaissances, on considèrera que la date de fin est alors la même que la date de début. La balise `jour` permet d'annoter les jours de la semaine tandis que `heure-début` et `heure-fin` annotent les heures et que `partie-de-journée` annote les informations du types *matin* et *après-midi*. À terme, dans la base de connaissances, toutes les informations d'horaires seront converties en parties de journées.

Pour repérer et annoter les informations temporelles, notre système est basé sur une approche symbolique, reposant sur des patrons linguistiques. Pour chaque type d'information à repérer (informations temporelles, informations spatiales, informations sur le type de la ressource), un module de transducteurs a été développé à l'aide de l'outil Unitex[5]. Cet outil permet de traiter des corpus en utilisant des dictionnaires (basés sur les tables du LADL[6]) ; et ce au niveau du lexique, de la syntaxe ou de la morphologie. Il permet de repérer et de baliser des structures correspondant à des expressions régulières, représentées par des graphes à états finis. A titre indicatif, le module de repérage et d'annotation temporel regroupe 24 graphes comprenant 88 marqueurs de surface (comme *"ouvert"*, *"mardi"*, etc.) et 22 marqueurs généraux (*"le", "du"*, etc.). La sortie d'Unitex est stockée dans un fichier texte dans lequel des balises d'annotations, au format XML, ont été ajoutées pour marquer les données identifiées. Ainsi, pour l'expression *Ouvert Juillet Août, sauf jours fériés du Mardi au Dimanche*, on obtiendra l'annotation suivante :

```
<Periode-Ouverture>
        <Date-Debut> Juillet </Date-Debut>
        <Date-Fin> Août </Date-Fin>
        <Incertitude/>
        <Exception> sauf jours fériés </Exception>
        <Jour>du mardi au dimanche<Jour>
</Periode-Ouverture>
```

4 Enrichissement de l'annotation

L'étape de construction a permis de recueillir les périodes de fermeture et d'ouverture qui sont explicitement précisées dans la page web de la ressource. L'objectif de cette deuxième étape est de prendre en compte les connaissances implicites afin de les rendre explicites dans les informations intégrées à la base. D'une part, une ressource peut présenter uniquement des périodes de fermeture, uniquement des périodes d'ouverture ou bien combiner ces deux types d'informations. On est par exemple confronté à une information du type *"la discothèque l'Oiseau Noir est fermée les dimanche, lundi, mardi et mercredi"* qui signifie d'une façon naturelle qu'elle est ouverte les autres jours de la semaine. Il est donc nécessaire

[4] Une DTD (Document Type Definition) permet de décrire un modèle de document XML.

[5] Unitex : http ://www-igm.univ-mlv.fr/ unitex

[6] Laboratoire d'Automatique Documentaire et Linguistique - les tables ont été créées au LADL par Maurice Gross et contiennent des unités lexicales classées selon des propriétés syntaxiques et distributionnelles.

d'envisager de calculer des périodes d'ouverture à partir de périodes de fermeture. D'autre part, certaines périodes de fermeture peuvent ne pas être explicitement spécifiées car elles sont intuitivement déductibles du contexte associé à la ressource. Par exemple, si la ressource touristique se situe en France, on en déduit qu'elle est fermée les jours fériés. Dans ce cas, on se base sur les propriétés géographiques de la ressource pour déterminer la période de fermeture. Ou encore, si la ressource est une discothèque, elle sera fermée en journée. Dans ce cas, on s'est basé sur la nature de la ressource. De manière générale pour ces deux cas, on s'appuie sur des connaissances qui relèvent du domaine pour déduire une période de fermeture.

L'enrichissement des annotations est réalisée en trois temps :

1. Dans un premier temps, l'annotation subit un traitement au cours duquel toute période est traduite en une donnée facilement exploitable par le mécanisme décrit dans la suite. Ce traitement permet de *désambiguïser* et de *discrétiser* les informations temporelles. Par exemple, *"début janvier"* sera traduit en *"01/01"*, ou bien *"début de la semaine"* en *"lundi"*. Une période est discrétisée en des moments de la journée pour chaque jour de la période (matin, midi, après-midi, soir, nuit). Dans le cadre d'un portail sémantique tourisme/territoire, il parait inutile d'être plus précis.
2. Dans un second temps, on se base sur les propriétés de la ressource pour déterminer ses périodes de fermeture par défaut. Pour ce faire, un ensemble de règles spécifiant les connaissances implicites de domaine est intégré à l'ontologie. Ces règles associent une période de fermeture à un profil de ressource particulier (par défaut une ressource est présumée ouverte tout le temps). Ainsi on peut exprimer des règles du type : *si une ressource est une discothèque alors on en déduit qu'elle est fermée les matins, les midis et les après-midis de tous les jours de la semaine*, ou encore, *si une ressource est située en France alors elle est fermée les jours fériés (ces jours étant explicitement énumérés)*. Ces règles sont appliquées sur les ressources par un service de raisonnement développé dans le cadre d'Eiffel (ce service est décrit dans Carloni et al. (2007)). Ces règles pouvant générer des contradictions, une fois leur application terminée, le service de raisonnement vérifie la cohérence de l'annotation *enrichie* : si une période de fermeture inférée chevauche une période d'ouverture initialement présente dans l'annotation, alors le contenu inféré de l'annotation n'est pas conservé. En cas de contradiction, les connaissances de la page web sont donc considérées comme prioritaires par rapport à celles qui ont été inférées.
3. Dans un troisième temps, on homogénéise le contenu de la base en ramenant toutes les connaissances à des périodes d'ouverture. En effet, l'exploitation de la base de connaissances portera uniquement sur l'ouverture et non sur les périodes de fermeture. Pour ce faire, on s'appuie sur les périodes de fermeture que présente la ressource : on détermine la période de référence que cette période de fermeture impacte, et on en déduit la période d'ouverture comme étant le complémentaire de la période de fermeture par rapport à cette période de référence. La période de référence est la période d'ouverture de la ressource lorsqu'elle en présente une, sinon on choisit celle qui convient le mieux et qui englobe la période de fermeture parmi une liste de périodes de référence par défaut. Par exemple, si la phase d'acquisition a permis d'extraire *la discothèque l'Oiseau Noir qui est fermée les dimanche, lundi, mardi et mercredi* ; lors de l'application des règles, on a déduit que cette discothèque était aussi fermée le matin, le midi et l'après-midi de tous les jours de la semaine. A l'étape d'homogénéisation, comme la période de fermeture est exprimée à la fois en jours de la semaine et en moments de la journée et qu'aucune période d'ouverture n'est spécifiée, on considère que la période de référence est la semaine découpée en moments de la journée. La période d'ouverture obtenue concerne tous ces moments à l'exclusion de ceux appartenant à la période de fermeture : soit jeudi soir/nuit, vendredi soir/nuit et samedi soir/nuit.

L'approche mise en œuvre présente le désavantage de "gommer" certaines nuances en établissant des choix arbitraires qui mériteraient une analyse plus fine. Par exemple, supposons que le processus d'acquisition extrait un restaurant fermé les lundi/mardi et ouvert le jeudi. L'approche décrite ici détermine qu'il est ouvert du mercredi au dimanche ce qui conduit à éliminer l'ambiguïté existant dans l'énoncé initial sur les mercredi, vendredi, samedi et dimanche. Il pourrait être intéressant de considérer cette ambiguïté comme une information à part entière sur laquelle peut s'appuyer le système d'interrogation. On peut aussi souhaiter introduire plus de nuances dans la définition d'une règle de déduction de périodes de fermeture. Par exemple, dire qu'*il y a de fortes chances pour qu'un restaurant soit fermé le lundi* est sans doute préférable à *un restaurant est toujours fermé le lundi*. Ce type d'ambiguïtés et de degré de certitude dans l'expression d'une connaissance est difficilement exprimable dans l'approche actuelle. La section suivante présente une seconde approche qui permet de mieux répondre à ce genre de besoins.

5 Proposition de modélisation possibiliste d'expressions temporelles

Au lieu de fixer de manière arbitraire les périodes d'ouverture d'une ressource touristique au coût d'une interprétation optimiste ou pessimiste (par exemple en interprétant *Ouvert de fin Janvier à fin Février* par *ouvert dès le 30 janvier* ou *ouvert à partir du 2 février*), on peut opter pour l'utilisation de la théorie des possibilités Dubois et Prade (1988) qui,

d'une part, permet de définir une certaine gradualité dans la définition d'une date mal connue d'ouverture ou fermeture et, d'autre part, permet de modéliser un manque d'information de manière réaliste.

Le principe est d'affecter à une ressource pour chaque date, un degré de possibilité d'ouverture μ_O et un degré de possibilité de fermeture μ_F, dont les valeurs peuvent aller de 0 à 1 selon qu'il est respectivement impossible ou possible que la ressource touristique soit ouverte ou fermée. Avec $0,8$ par exemple, on dit que la connaissance est possible à un degré $0,8$. Pour garantir une consistance dans nos données, il faut que l'ouverture ou la fermeture d'un établissement à une date donnée soit complètement possible (i.e. pour chaque date l'un des degrés est à 1), sinon, les informations sont considérées incohérentes. Ceci correspond à une distribution de possibilité normée, dans laquelle au moins une des valeurs envisagées est complètement possible (ici nous avons deux valeurs envisageables qui sont *ouvert* et *fermé*). Notons au passage qu'en cas de manipulation d'informations incohérentes, il est toujours possible de normaliser les degrés de possibilités avant l'utilisation de ces données (par exemple, en ramenant à 1 le degré le plus grand).

Si on suppose qu'une ressource est fermée un jour, mais qu'on imagine tout de même possible qu'elle soit en réalité ouverte (par exemple un restaurant est supposé fermé le lundi soir, mais on sait qu'il n'est pas impossible qu'un restaurant soit tout de même ouvert), on peut modéliser notre connaissance par les degrés de possibilité $\mu_O = 0.5$ et $\mu_F = 1$.

Il est alors intéressant de définir un degré de certitude pour notre information (appelé plus souvent mesure de nécessité). Dans nos exemples, le degré de certitude concernant l'ouverture d'une ressource est défini par 1 moins la possibilité qu'elle soit fermés. Dans l'exemple précédent, on ne peut donc pas être sûr que le restaurant en question soit ouvert. En revanche on est sûr à un degré 0.5 qu'il est fermé. Voici par exemple pour un établissement et un jour donnés, certaines valeurs que peuvent prendre les deux degrés, et les conclusions que l'on peut en tirer :

- $\mu_O = 1$ et $\mu_F = 0$: on est alors sûr que l'établissement est ouvert ce jour la.
- $\mu_O = 0$ et $\mu_F = 1$: on est alors sûr que l'établissement est fermé ce jour la.
- $\mu_O = 1$ et $\mu_F = 1$: on est en présence d'une incertitude totale, on ne peut rien conclure.

5.1 Exemple de modélisation d'une connaissance vague

Prenons l'exemple de l'expression *Fermeture annuelle de fin Janvier à fin Février*. Plutôt que de déterminer "fin janvier" arbitrairement comme le "31 janvier", on peut modéliser l'ouverture de l'établissement par les distributions de possibilités représentées sur la Figure 1. On définit formellement pour chaque date une distribution de possibilité normée sur l'ensemble des éléments $\{ouverture, fermeture\}$, mais pour des questions de lisibilité nous représentons ces informations sous la forme de deux fonctions $\mu_O(.)$ et $\mu_F(.)$ qui donnent les degrés de possibilité des éléments *ouvert* et *fermé* à une date donnée. Du 25 janvier au 31 janvier, il est de de moins en moins pensable que l'établissement soit ouvert, le degré de possibilité d'ouverture diminue donc progressivement de 1 à 0 entre le 25 et le 31 janvier. Inversement, du 20 au 25 janvier, il est de plus en plus pensable que l'établissement soit fermé.

L'établissement en question est-il ouvert le 28 janvier ? On déduit de la distribution de possibilités d'ouverture qu'il est possible à un degré 0.3 (donc très moyennement possible) que l'établissement soit ouvert à cette date là. En revanche, on peut avancer que l'établissement est fermé avec un degré de certitude de 0.7.

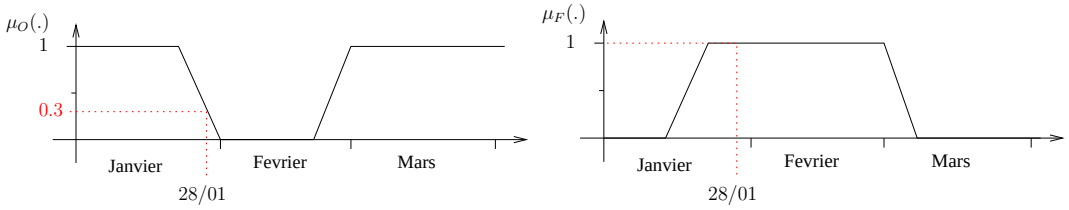


FIG. 1 – *Fermeture annuelle de fin janvier à fin février*

5.2 Exemple de modélisation d'une connaissance ambiguë

Soit l'expression *Ouvert les après-midi sauf lundi et mardi*. Dans cette expression, il est certain que l'établissement est ouvert les après-midis du mercredi au dimanche (degrés de possibilité d'ouverture = 1 et de fermeture = 0). On est sûr qu'il est fermé les lundi après-midi et mardi après-midi (degré de possibilité d'ouverture = 0 et de fermeture = 1), et l'ouverture est incertaine pour le reste du temps puisqu'assujettie à l'interprétation de l'expression. On peut modéliser cette incertitude en affectant un degré de possibilité d'ouverture 0,5 aux matinées (avec un degré de possibilité de fermeture de 1). Ceci nous donne les distributions de possibilités d'ouverture et de fermeture résumées dans la Table 1.

Ainsi à la requête "l'établissement est-t-il ouvert toute la journée du mercredi ?", la réponse serait oui avec un degré de certitude de 0. À la requête "l'établissement est-t-il ouvert à un moment de la journée du mercredi ?", la réponse serait oui avec un degré de certitude de 1. À la requête "l'établissement est-t-il ouvert le lundi ?", la réponse serait oui avec un degré de possibilité de 0.5. Et enfin à la requête "l'établissement est-t-il fermé le lundi ?", la réponse serait oui avec un degré de possibilité de 1 et un degré de certitude de 0.5.

		Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
μ_O	matin	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	soir	0	0	1	1	1	1	1
μ_F	matin	1	1	1	1	1	1	1
	soir	1	1	0	0	0	0	0

TAB. 1 – *Ouvert les après-midi sauf lundi et mardi*

5.3 Exemple de modélisation de connaissances contextuelles

Les connaissances contextuelles peuvent permettre d'obtenir le contour des distributions de possibilités de nos connaissances. Par exemple, *"en général, les musées sont fermés le soir"*, *"les restaurants sont fermés les lundi soir"*.

On peut affecter un degré de possibilité d'ouverture (resp. fermeture) de 1 lorsqu'il est contextuellement supposé qu'un établissement est ouvert, et un degré de possibilité de l'événement contraire fermeture (resp. ouverture) de 0.5. Sans autre source d'information que le contexte, la connaissance est donc incertaine. Pour un restaurant, cela donne les connaissances contextuelles définies dans la Table 2.

S'il est précisé sur le site internet d'un restaurant *ouvert le lundi soir*, il faut alors changer la possibilité d'ouverture du lundi par 1, et la possibilité de fermeture par 0 (on ne pourra affirmer que l'établissement est ouvert le mercredi qu'avec un degré de certitude de 0.5). Si on trouve sur le site *"ouvert tous les jours"*, il faudra alors donner une possibilité d'ouverture de 1 à tous les jours, et diminuer le degré de possibilité de fermeture à 0 pour chaque jour. L'intérêt de ce genre d'information contextuelle est avant tout de pouvoir manipuler une connaissance supposée par défaut. En pratique ce doit être l'information la moins prioritaire des informations dont on dispose.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
μ_O	0.5	1	1	1	1	1	1
μ_F	1	0.5	0.5	0.5	0.5	0.5	0.5

TAB. 2 – *Connaissances contextuelles relatives à un restaurant*

5.4 Exemple complet

Dans cette partie, nous allons essayer de modéliser notre connaissance sur l'ouverture d'un restaurant. Nous connaissons les moments de la semaine où les restaurants sont habituellement ouverts (ces moments sont résumés dans la Table 2). Sur le site internet du restaurant que l'on considère ici, il est écrit : *Ouvert Juillet Août, sauf jours fériés du Mardi au Dimanche*. Nous allons traiter successivement chaque information, de la plus générale à la plus spécifique.

Dans notre exemple, nous allons commencer par discuter des jours d'ouverture possibles de la semaine. Le restaurant doit être ouvert *"du Mardi au Dimanche"*, ce qui nous permet de modifier notre information contextuelle supposée, de manière à obtenir les possibilités d'ouverture et de fermeture la Table 3.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
μ_O	0	1	1	1	1	1	1
μ_F	1	0	0	0	0	0	0

TAB. 3 – *Ouvert du mardi au dimanche*

Ensuite, attachons-nous à la période *"Ouvert Juillet Août"*. Comme il n'est pas précisé explicitement ouvert *à partir du 1er* Juillet, on se doit de supposer que la date d'ouverture se situe aux alentours du 1er Juillet, ce qui donne les possibilités

d'ouverture/fermeture représentées en haut de la Figure 2, (on restreint pour plus de clarté les illustrations aux mois de Juin et Juillet). Contrairement à la Figure 1, le temps est discrétisé. Dans un souci de lisibilité nous ne considérons qu'un seul moment de la journée : le soir.

Nous pouvons maintenant agréger nos connaissances relatives aux jours de la semaine, et à la période d'ouverture. Pour cela nous avons besoin des connaissances relatives à l'année considérée. Plaçons nous dans l'année 2009, les 22 et 29 Juin 2009 ainsi que les 6, 13, 20 et 27 Juillet tombent un lundi. Enfin, nous pouvons utiliser l'information *"sauf jours fériés"* qui nous semble être la plus spécifique. Pour cela nous avons encore besoin des connaissances relatives à l'année considérée pour savoir que le 14 Juillet et le 15 août sont fériés (cela pourrait éventuellement faire partie d'un savoir valide chaque année). L'agrégation de toutes nos connaissances doit donc donner les possibilités représentées en bas de la Figure 2.

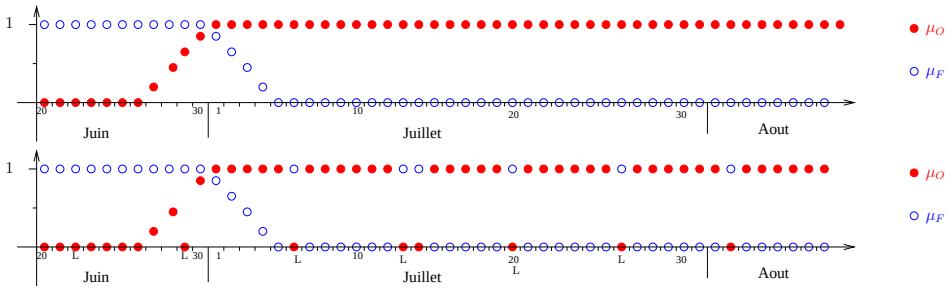


FIG. 2 – *Ouvert Juillet Août*

5.5 Exploitation des connaissances temporelles

Le schéma général d'interrogation des données temporelles utilisée dans le cadre du projet Eiffel est du type : *chercher un créneau de "n jours successifs" sur une période donnée* ou *chercher un créneau de "n jours éparpillés" sur une période donnée*. Par exemple : *"ce restaurant est-il ouvert le <une date> ?", "je cherche un hôtel pour la semaine du <samedi j> au <samedi j+7> ?", "je veux réserver une semaine au mois de Juillet", "je veux profiter de mon séjour pour visiter tel musée ?"...* Il faut également penser à la combinaison de ces requêtes temporelles. Par exemple : *"Je souhaite partir 2 semaines cet été : une semaine dans un village de vacances à la montagne où je puisse faire 2 jours de ski d'été suivie d'une semaine dans un camping de bord de mer pendant un festival d'été."*

Lors de la présentation des résultats à l'utilisateur du portail, il n'est pas envisageable de présenter deux scores, le premier étant le degré de possibilité π et le second le degré de certitude N. D'autant plus que si le degré de certitude est non nul, cela signifie que le degré de possibilité vaut 1. Inversement, si le degré de possibilité est différent de 1 cela implique que le degré de certitude vaut zéro. On peut donc agréger ces deux degrés d sur une échelle unique graduée de 0 à 100 ayant pour valeur $d = \frac{\pi+N}{2} \times 100$. Pour $d = 100$, la réponse est certaine, tandis que pour $d = 0$, la solution associée est impossible. Entre les deux, la réponse est de moins en moins probable et il est recommandé de la vérifier (par exemple en téléphonant aux établissements concernés). Cette échelle pourra donc permettre de classer les différentes réponses d'une requête. En pratique on pourra éventuellement se contenter d'afficher, s'il y en a, les réponses certaines ($d = 100$), et n'afficher les réponses incertaines que s'il n'existe pas de réponse sûre.

Supposons par exemple que la Figure 1 représente l'ouverture d'un hôtel A, et la Figure 3 l'ouverture d'un hôtel B. On pose la question *"peut-on se loger dans l'hôtel A ou B entre le 20 et le 28 février ?"*. Ici la question sous entend qu'on

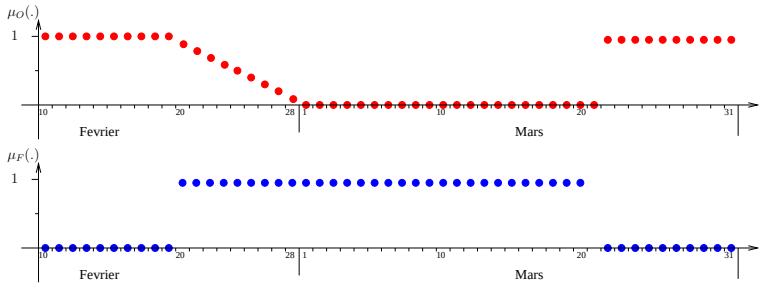


FIG. 3 – *Fermé de fin février au 21 Mars*

veut se loger tous les jours de la période en étant prêt à changer d'hôtel en cours de séjour. La réponse sera alors oui avec

un degré de possibilité $0,5$ et de certitude de 0. Ce degré de possibilité est calculé de la façon suivante : c'est le minimum du maximum des degrés de possibilité d'ouverture des établissements (i.e. le minimum de la fonction supérieure de la Figure 4 sur l'intervalle des dates recherchées). Le score présenté à l'utilisateur sera donc de 25, sur l'échelle graduée jusqu'à 100.

En revanche si l'on pose la question "peut-on se loger une nuit dans l'hôtel A ou B entre le 20 et le 28 février ?", la réponse sera oui avec un score de 100 (réponse certaine).

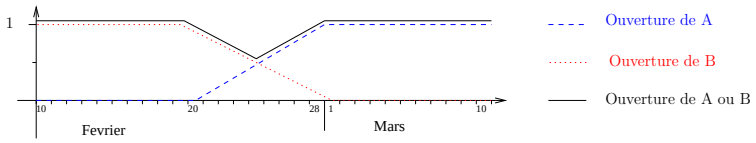


FIG. 4 – *Ouverture de A ou B*

6 Conclusion

Dans le cadre d'un projet concret d'acquisition et gestion de connaissances touristiques, nous proposons d'utiliser une modélisation possibiliste des données temporelles extraites automatiquement afin de mieux prendre en compte l'imprécision de ces données. Pour chaque objet touristique et à chaque chaque date considérée, deux degrés de possibilité sont calculés. Un degré d'ouverture et un degré de fermeture. Ainsi, le système sera en mesure de répondre aux requêtes en proposant les solutions les plus sûres, mais pourra également proposer des solutions envisageable mais non certaine sur la période demandée. Dans ce dernier cas, un degré de fiabilité de la réponse peut être calculé.

Pour l'implémentation de ce modèle possibiliste, nous étudions deux approches. La première consiste en la création et le stockage de tables représentant les distributions de possibilité d'ouverture et de fermeture. Cette opération pourra alors être effectuée dès la fin de l'analyse lexicale. Pour exécuter une requête, il suffira alors de faire un accès aux tables représentant nos connaissances. Le principal inconvénient de cette solution est l'espace mémoire consommé. Une autre solution consiste à ne garder en mémoire que le résultat des annotations brutes extraites. Il faut alors calculer à la volée pour chaque requête les degrés de possibilité d'ouverture des établissements susceptibles d'être concerné par la requête.

Références

Carloni, O., M. Leclère, et M. Mugnier (2007). Introducing reasoning into an industrial knowledge management tool. *Applied Intelligence*.

Dubois, D. et H. Prade (1988). *Possibility Theory. An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York.

Noël, L., O. Carloni, N. Moreau, et S. Weiser (2008). Designing a knowledge-based tourism information system. In *Int. J. of Digital Culture and Electronic Tourisme, Special Issue on National Tourism Organisations and Exploitation of Information Technologies*. Inderscience Publishers Ltd.

Weiser, S. (2008). Informations spatio-temporelles et objets touristiques dans des pages web : repérage et annotation. In *Actes de Recital*, Avignon.

Weiser, S., P. Laublet, et J.-L. Minel (2008). Automatic identification of temporal information in tourism web pages. In E. L. R. A. (ELRA) (Ed.), *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Summary

In the framework of a semantic e-tourism portal, data is automatically collected from web pages in order to fill a knowledge base. Amongst other types of information, temporal information is essential, concerning dates and opening times of tourism resources. These opening times are often incomplete, vague or ambiguous and their interpretation requires implicit knowledge about the tourism field. In this paper, we focus on the acquisition process and we shed light on the problems revealed by this temporal information. We suggest a possibilist modelisation of this extracted knowledge in order to take into account the vagueness of such knowledge and allow flexibility for its use.