Mining Heterogeneous Multivariate Time-Series for Learning Meaningful Patterns: Application to Home Health Telecare

Florence Duchene, Catherine Garbay, Vincent Rialle

Mining Heterogeneous Multivariate Time-Series for Learning Meaningful Patterns: Application to Home Health Telecare

Florence Duchêne[1], *Catherine Garbay*[1], *and Vincent Rialle*[1,2]

[1]Laboratory TIMC-IMAG, Faculté de médecine de Grenoble, France
[2]Department of Medical Informatics (SIIM), Michallon hospital, Grenoble, France

Abstract

For the last years, time-series mining has become a challenging issue for researchers. An important application lies in most monitoring purposes, which require analyzing large sets of time-series for learning usual patterns. Any deviation from this learned profile is then considered as an unexpected situation. Moreover, complex applications may involve the temporal study of several heterogeneous parameters. In that paper, we propose a method for mining heterogeneous multivariate time-series for learning meaningful patterns. The proposed approach allows for mixed time-series – containing both pattern and non-pattern data – such as for imprecise matches, outliers, stretching and global translating of patterns instances in time. We present the early results of our approach in the context of monitoring the health status of a person at home. The purpose is to build a behavioral profile of a person by analyzing the time variations of several quantitative or qualitative parameters recorded through a provision of sensors installed in the home.

Keywords – Time-series mining, Heterogeneous Multivariate Time-series, Temporal Patterns, Unsupervised Learning, Home Health Telecare.

1 Introduction

In the last years, the increasing amount of stored data with possibly high dimensionality has encouraged researchers to take a great interest in discovering new patterns or building models from large datasets, also referred to as knowledge discovery or data mining. Moreover, many business to scientific applications which serve mainly to support diagnosis and predict future behaviors effectively deal with temporal sequences [31], encouraging the development of the related "time-series mining" research field.

In this work we investigate the issue of mining multidimensional and heterogeneous time-series for learning meaningful patterns. This is particularly useful in most monitoring purposes, when dealing with the detection of unusual trends or behaviors of an object or a situation described by the variation of data recorded from several types of sensors or information sources. One application is the monitoring of the health status of a person at home. The aim is to support the caregivers by providing information about unusual trends in the person's behavior observed through the variation of quantitative or qualitative parameters monitored at home. In that context of detecting bad trends in health status, we aim to learn the person's lifestyle to build a sort of profile, which is sensitive to any critical deviation, and then to detect any unusual behavior in comparison with this profile. This approach toward the decision-making issue is required because it is inconceivable to describe all possible critical situations of any nature and level, just as we do not yet have any way of learning the occurrence of such situations (monitoring of persons getting to critical situations and collecting the corresponding data). A learning process is then defined to build a behavioral profile of the person in their activities of daily living, that is to extract and characterize frequent patterns from heterogeneous multivariate time-series recorded in usual conditions of life. The decision-making process must be able to detect unusual behaviors by comparison to this profile. Therefore, the pattern learning process should allow for heterogeneous components defining time-series, as well as for imprecise matches, outliers, stretching, and global translating in time of the sequences corresponding to a same pattern.

Considering our context also justify the choice of extracting multidimensional patterns related to the person's behavior rather than analyzing individually each parameter monitored at home to make a joint decision about their condition of life. Indeed, the observable parameters are selected as a compromise between: (1) being easily observable and non invasive, and (2) gaining a full appreciation of the person's condition, sensitive to any change in the health status. Therefore, all parameters are closely related one to each other, and their joint variations need to be preserved in multidimensional patterns representative of any usual behavior. Our objective is then to build a system performing an unsupervised extraction of this kind of temporal patterns within time-series representative of a person's usual conditions of life. Our contribution lies in extending an algorithm for pattern extraction to both multidimensional and heterogeneous time-series, accounting for the large amount of noise possibly present in patterns' instances. We then also need to introduce a similarity measure appropriate to the comparison of multidimensional and heterogeneous time-series.

The rest of the paper is organized as follows. In section 2 we present works related to time-series mining, and in section 3 our methodology for extracting frequent patterns from heterogeneous multivariate time-series. Then, section 4 defines an appropriate similarity measure, and section 5 details the different steps required for pattern discovery and clustering. Section 6 discusses the early experimental results related to the proposed approach in the context of home health telecare. Finally, section 7 concludes the paper.

2 Related work

Most of the current works dealing with home health telecare are focused either on implementing a generic architecture for the integrated medical information system, on improving the daily life of patients using various automatic devices, specific equipment, and basic alarms, or on providing health care services to patients suffering from specific diseases like asthma, diabetes, cardiac, pulmonary, or Alzheimers. Rialle *et al.* have presented in [30] an overview of projects related to home health telecare. Basic alarms are raised by smart sensors or low layers of a local intelligence unit when a problem occurs at a short temporal scale: either one parameter overpasses a critical value (nocturia, pollakisuria, fall, hypertensive crisis, etc.), or a critical scenario involving the value of possibly more than one parameter is recognized (asthma crisis, etc.). Our focus is on the broadcasting of high level alarms about the persons health status, which concern a larger temporal scale. That issue is solved by first learning the daily living habits of a person to be able to detect later unusual situations. That behavioral profile is built by mining heterogeneous multivariate temporal data collected from sensors installed at home for learning meaningful patterns.

According to Antunes *et al.* [2], temporal sequences are related to series of nominal symbols from a particular alphabet, whereas time-series concern continuous, real-valued elements. In this work we are interested in heterogeneous multivariate sequences of time-varying data, referred to as heterogeneous multivariate either time-series or temporal sequences. Time-series mining is an active field of research (see [2, 31] for overviews of temporal data mining). Discovery algorithms for time-series aim at extracting important patterns such as similarities, trends, or periodicity, in a purpose of description or prediction [27]. Pattern discovery in time-series is useful for temporal sequences synthesis [12], as well as for learning tasks like association rules mining [9, 14], classification [15], unsupervised clustering [32].

By analogy with non sequential domains and because of the exponentially large set of possible subsequences considering temporal sequences, time-series mining used to serve a learning task is sometimes referred to as "feature mining" [18, 20]. Considering non-sequential domains, feature selection corresponds to finding an optimal space of size m from the full d-dimensional feature space, where ideally $m \ll d$. In sequential domains, "feature selection aims to select the best subset of sequential features" [20], that is, the most relevant subsequences regarding the decision purpose. Time-series mining then acts as a preprocessor to construct the best subset of sequential features used to feed into learning algorithms [21]. This is particularly useful to improve learning performances when time-series contain both pattern and non-pattern signals, like in [12]. According to [20], the selection criteria for feature mining include that features should be frequent, distinctive of at least one class, and that feature sets should not contain redundant features, that is subsequences.

Pattern discovery in time-series may be either (1) supervised – that is finding patterns described by empirical knowledge or similar to a given "query sequence" [1, 15, 19] – or (2) unsupervised – that is finding recurrent patterns without any prior knowledge about the regularity of the data under study [7, 13, 12]. Lin *et al.* [22] have introduced the notion of "time-series motifs" considering the unsupervised issue of finding previously unknown, frequently occurring patterns in time-series. These specific patterns are also referred to as "primitive shapes" [9] or "frequent temporal patterns" [14].

The techniques used for time-series mining vary according to the application, regarding the characteristics of both the temporal sequences under study and the expected patterns: degree of variability in the values, allowed transformations between instances of a same pattern, possible stretching in time. For instance, Hong *et al.* have experimented training recurrent neural networks for an unsupervised extraction of multi-temporal sequence patterns [13]. However, this method suffers from noisy data. The use of finite state machines [12] may give out good results, which may however dramatically decrease as the dimensionality increases - that is the number of states. Chiu *et al.* [7] have implemented in the context of times-series motifs extraction an efficient algorithm based on random projections initially proposed by Buhler and Tompa to find motifs in nucleotide sequences [3]. Although they only deal with one-dimensional time-series and then do not address the issue of heterogeneous multivariate time-series, this *projection* algorithm is actually interesting because of the rapid extraction of approximate results and the efficiency even in the presence of noise or "don't care" symbols. However this method, as implemented in [7], does not allow for stretching in time between motifs instances.

Our objective is then to extend feature mining and learning from time-series to the unsupervised extraction of heterogeneous multivariate time-series motifs for learning a behavioral profile. We extend the use of the *projection* algorithm for feature mining in our noisy, heterogeneous and multi-dimensional context. The aim is to extract the most relevant features – that is, subsequences in time-series domain – to feed into a clustering algorithm for motifs identification. As an application, we focus on profiling the daily living habits of

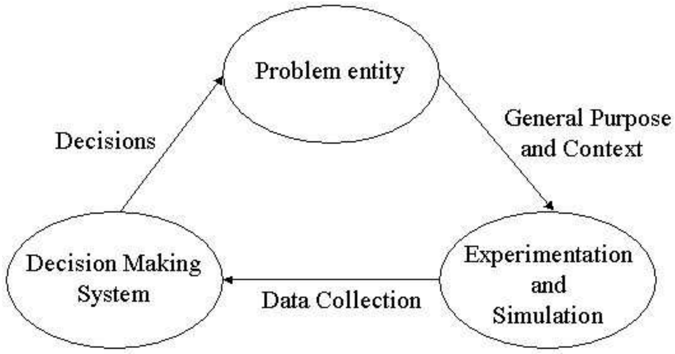


Figure 1: **The recursive steps of a problem solving scheme.**

a person from data recorded using a provision of sensors installed in their home.

3 A methodology for mining heterogeneous multivariate time-series

3.1 Problem entity *Pattern extraction in home health telecare*

Solving any complex decision making system requires to well specify the purpose and context of the decision. Dealing with several levels of details – like different levels of knowledge, data accuracy, decision – particularly requires to carefully set the needs, requirements and constraints of the system, so that the decision making matches the defined purpose at most. Setting up a decision making system – as motifs extraction – should then be considered as part of a problem solving scheme [10], including (see figure 1):

1. **Defining the context and the general purpose of the decision issue.** This aims at narrowing and specifying the space of information and knowledge to consider by answering questions like: what are the relevant observations to set up ? or which level of detail to consider? or what are the performance expected for the problem solving ?

2. **Collecting or generating data related to that context.** This data collection is led by contextual information related to the general purpose and context of the decision issue. Collecting large sets of representative data may be quite challenging in some applications. Setting up a simulation process [10] is then very useful as a first step of setting up a decision system to prevent from the lack of experimental data. Simulation also allows to get a full view of data potentially recorded in the context of study, by varying the parameters of the simulation process, so that the performance of decision making systems are better evaluated.

3. **Testing appropriate methods of decision making to solve the problem.** Data collected from experiments or generated by a simulation process are used as inputs of the decision-making system. The sensitivity and specificity related to these algorithms must match the problem requirements.

Once a decision process has been implemented and experimented, the results of matching between the outputs of the decision making process and the problem requirements may entail to refine one or more steps in order to get better sensitivity and specificity. For instance, more precision in the values of the observed parameters may be required in case of low sensitivity, or conversely, in case of low specificity. The problem solving scheme must also integrate validation at each stage, like face validation, with experts, or mathematical and statistical validation.

Defining the problem entity – that is the context and purpose of the decision – have some consequences on both setting up the experimental context and building the decision making system. Considering the decision making purpose and its complexity, we can specify some key parameters involved at these stages of any problem solving scheme, as follows:

- **Level of knowledge available.** This aims at identifying all knowledge possibly available and relevant to decision making, that is (a) *a priori* knowledge – including intuitive and academic knowledge – and (b) knowledge extracted from experimental data sets. Fusion of several kinds of knowledge is commonly used to deal with complexity and heterogeneity. The lack of knowledge related to a specific issue may also require to rely on various informational sources.

- **Level of details required.** Specifying the level of details required to deal with an issue is crucial to select: (a) appropriate knowledge, and consequently the level of experimental data collection and representation, and (b) relevant algorithms to solve the problem. There is a compromise to be found between the necessity to save the complexity of phenomena, and the restriction to a level of detail relevant to the problem, that is meeting the decision's purpose.

- **Level of performance expected.** The performance is defined using parameters like the sensitivity and specificity of the decision making system, such as an acceptable time to decision.

In our context of home health telecare, the issue of extracting patterns representative of a person's daily behavior at home is a *high level issue.* The ultimate goal is not to interpret precisely a problem that occurred at home, but to set up the context of occurrence of any change in the behavior. Therefore, the pattern extraction aims at identifying recurrent behaviors occurring at the scale of long time intervals, from about thirty minutes to several hours. The "right levels" to deal with that issue are detailed in the following paragraphs:

- **Level of knowledge.** Learning about daily behavioral profiles must be performed on individual basis, since behavioral profiles are specific of a person's physiological status and habits. Consequently, there is only a few *a priori* knowledge related to our decision issue. The decision-making system is then based on a set of data recorded at home and in real-time from a provision of sensors. In order to gain a full appreciation of the person's condition, sensitive to any change in the health status, data may be collected from different classes of sensors: (1) activity (location, posture, etc.), (2) environment (temperature, use of doors, window, lighting, etc.), and (3) physiology (heart rate, blood pressures, weight, etc.). The selection of relevant parameters is also constrained by many ethical, social, and individual issues: respect of privacy, confidentiality of data, ease of use and unobtrusiveness of input devices installed in the home.

- **Level of details.** Dealing with a high level issue, the decision making system may not require a high degree of detail and accuracy as regards the data involved in the process, that is data collected from sensors in real-time. Moreover, the experimental records may require a high level of representation to highlight their global trends, removing minor variations that are insignificant at our observation scale.

- **Level of performance.** The decision system needs to fit the general purpose of monitoring: the detection of all usual patterns (sensitivity) combined with a low rate of false alarms (specificity) – that is identifying patterns that do not correspond to usual behaviors, with an acceptable time to detection.

3.2 Experimental context *Some guidelines for pattern extraction*

An appropriate experimental context is set according to the purpose and requirements of the problem. The characteristics of the data produced in that context induce some guidelines for pattern extraction. Looking for meaningful patterns representative of human behaviors – the activities of daily living of a person at home – from heterogeneous data collected from a provision of sensors, the decision making-system must be able to address the following issues:

- **Multivariate time-series.** Relevance for dealing with time-varying objects or situations described by several parameters.
- **Heterogeneous components.** Capacity of handling qualitative as well as quantitative parameters in a coherent way to describe an object or a situation.
- **Mixed time-series.** Ability to learn from sequences containing both pattern and non-pattern data. Human behaviors captured in daily life indeed contain highly casual as well as regular motions.
- **Imprecise matches.** Capacity to discover "high-level patterns", that is to focus on the global trends embedded in the data despite the strong presence of noise between the instances of expected patterns, especially when considering human behaviors.
- **Outliers.** Capacity to preserve the detection accuracy despite the presence of outliers in subsequences corresponding to frequent patterns. That may be due to anomaly in the sensor or attributed to human failure or disruption.
- **Translation in time.** Ability to detect patterns translated in time: similar behaviors may occur at any time.
- **Stretching in time.** Ability to detect patterns of different lengths: dealing with human behaviors, a same activity does not always last the same duration.

3.3 Decision-making system *Methodology of pattern extraction*

The decision-making system aims at extracting meaningful temporal patterns from heterogeneous multivariate time-series. The patterns should correspond to usual behaviors of a person at home. Given that activities of daily living are specific to a given subject, we need a completely unsupervised learning approach, that may not be driven by prior knowledge about the patterns corresponding to living habits. Unsupervised time-series mining is usually made up of several consecutive steps to perform coarse-to-fine feature extraction [7, 12, 13, 19]. The principle is first to roughly restrict the feature space combining techniques like time-series representation, random projections [7], state-based temporal signal modelling [12], recurrent neural nets training [13], and then to identify more precisely the relevant subsequences for the learning task based for instance on specific constraints or similarity thresholds on the subsequences. As a consequence, this implies to define relevant

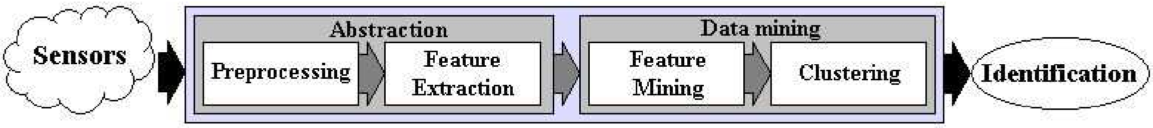


Figure 2: **Pattern recognition system designed in the context of mining *low-level* time-series for identifying *high-level* patterns.**

similarity measures for the given time-series. Antunes *et al.* [2] define temporal data mining as a process including three main steps:

- **Representation of temporal sequences.** Preprocessing, representation, and modelling of the data sequences that need to be applied before actual data mining operations take place (transformation, discretization, generative models building).
- **Similarity measure for sequences.** Definition of an appropriate similarity measure according to the characteristics of the time-series.
- **Mining operations.** Application of models and representations to the actual mining problem (association rules mining, classification, unsupervised clustering, prediction).

Our approach differs from the one of Antunes *et al.* in the way we specify each of these three steps. Considering a complex issue involving a large scale from the level of details embedded in raw data to the decision level, we need to refine the definition of representing and mining temporal sequences: (a) Representation needs then to be defined as a step of *abstraction*, to get from raw data a level of information that better deals with the decision purpose – preprocessing and feature extraction ; and (b) mining operations are divided into two consecutive steps – feature mining and clustering – to progressively focus on the most appropriate features to pattern extraction. We then propose a new general design of pattern recognition systems in the case of dealing with large temporal data sets, as shown on figure 2. The following paragraphs refine in that context the three steps defined by Antunes *et al.*.

Representation

Considering the extraction of high level patterns from time-series (see §3.1), the step of **representation** is not a simple preprocessing of data. Once represented, time-series must fit the level of details required by the decision system. We then define the representation as a step of *abstraction* of raw data to capture a higher level of information most appropriate to the decision's purpose. The aim is to get a synthetic representation of the sequential data, meaningful in terms of identifying the activities of daily living of a person at home. Given than an activity can be described as a succession of elementary *"actions"*, each of them being performed for a certain duration, we aim at representing raw time-series by sequences of meaningful symbols, each symbol representing the person carrying out a given "action" for a certain time. That implies to gather along time successive data records corresponding *a priori* to a same "action", that is whose sequence present no significant temporal variations. Time-series are then abstracted as sequences of symbols – some multidimensional vectors, each of them synthesizing a "stationary" state of the monitored parameters during a certain period of time.

Mining operations

Dealing with mixed time-series, embedding both pattern and non-pattern data subsequences (see §3.2), we propose to divide the definition of **mining operations** into two consecutive steps to progressively better match the decision level:

1. **Feature mining.** Selection of the most meaningful features (that is, subsequences), so called *tentative motifs*, considering the purpose of frequent patterns extraction and classification. These subsequences are used as features to feed into a classification algorithm.

2. **Clustering.** Unsupervised classification of the *tentative motifs* into meaningful classes whose representative sequences are called *time-series motifs*.

In our experimental context, tentative motifs should be representative of a person's repetitive behaviors, and a *motif* is then defined as a meaningful class of tentative motifs, representative of any typical activity of the person. Because we need a completely unsupervised learning approach, both *feature mining* and *clustering* must be unsupervised.

Similarity measure

Considering our level of complexity, a similarity measure is required at the two stages of representing and mining time-series:

1. **Representation.** The purpose of representation is to synthesize in a single symbol subsequences whose successive vectors share similar values, possibly for different durations, and therefore representative of a same "action" performed along the corresponding time. We then need to roughly evaluate the proximity of successive vectors and their relevance to be abstracted in one symbol describing a continuous same type of action performed. For the sake of robustness and efficiency, a discretization step is first of all applied to quantitative parameters. Given the fact that we are looking for global trends, a low approximation of the actual distance between a subsequence of vectors and its corresponding "mean vector" is sufficient. In case these subsequences are roughly similar, we can then assume the "mean vector" well represent the subsequence for the corresponding duration. As a consequence, we propose to use a *discrete minimum distance* for that purpose (see §4.3).

2. **Mining operations.** Once the possible locations of patterns have been identified within the original time-series, a similarity measure between these subsequences is required for (a) *Feature mining*, to decide whether or not they are effectively similar enough to be considered as a possible pattern, and (b) *Clustering*, to classify all these extracted subsequences into meaningful groups in terms of characterizing the activities of daily living. At this stage we need to compute an *actual distance* relevant to heterogeneous multivariate time-series (see §4.2).

In that context, specifying a similarity measure then includes defining the following elements: (1) homogeneous distance for heterogeneous components, (2) *actual distance* between heterogeneous multivariate time-series, and (3) *minimum distance* between time-series.

The next sections details, first, the definition of required similarity measures and, second, the proposed approach for identifying time-series motifs, including (1) representation of time-series, (2) *feature mining*, for tentative motifs discovery, and (3) *clustering*.

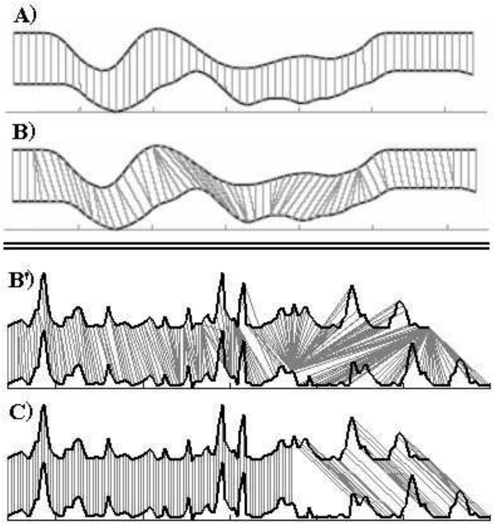


Figure 3: **Pairs of points considered as similar and associated when computing the distance between time-series.**
The graphs represent the results of associations when computing (A) Euclidian distance, (B & B') Dynamic Time Warping (DTW) distance, and (C) distance based on the longest common subsequence (LCSS).

- The comparison of points associations between A) and B) shows the better efficiency of DTW distance over an Euclidian distance to deal with possible distorsions in the time axis.
- The comparison of B') and C) highlights the better efficiency of LCSS over DTW distances to support the presence of outliers.

4 Similarity measure

Various similarity models have been successfully used to compare temporal sequences, as illustrated on figure 3 and detailed below. The simplest approach typically used to define a similarity function is based on the Euclidian distance, or some extensions to support various transformations such as scaling or shifting. Chui *et al.* [7] have used it successfully for extracting one-dimensional time-series motifs in some specific cases. However, this model cannot deal with outliers and is very sensitive to small distorsions in the time axis (see figure 3, case **A)**). Another approach is to use the *Dynamic Time Warping (DTW)* distance which allows stretching in time and comparing time-series of different lengths [16, 17] (see figure 3, case **B)**). However, a great amount of outliers still results in very large distances, even though the difference may be found in only a few points (see figure 3, case **B')**). Non-metric techniques have then been introduced and efficiently used to better deal with noisy data [1, 8, 32]. The idea is to capture the intuitive notion that "two sequences should be considered similar if they have enough non-overlapping time-ordered pairs of subsequences that are similar" [1]. This refers to finding the *Longest Common Subsequence (LCSS)* between two time-series. This approach allows for outliers, different scaling factors, and baselines (see figure 3, case **C)**).

However, the above works mainly deal with low dimensional (from one to three dimensional) time-series and do not address the issue of heterogeneous components (quantitative or qualitative) describing a moving object. Considering heterogeneous multivariate time-series in a particularly noisy context, our objective is then to extend the *LCSS* approach to heterogeneous multivariate time-series. We first need to define a coherent distance between points whatever the type of parameter. Then we detail its integration in computing the *actual distance* between heterogeneous multivariate time-series, which is used for *mining operations.* At last, we use these definitions to extend the approach proposed in [7] for computing a *minimum distance* between time-series, which is used for time-series *representation.*

4.1 Homogeneous distance between heterogeneous components

We would like to allow the description of an object using several parameters of the following possible types:

- Quantitative
- Ordered qualitative
- Unordered qualitative

The simplest way of insuring the coherence of the similarity measure is to make the distances between two values range from 0 to 1 for each type of parameters. Let a and b be two values of a given parameter, and $d(a, b)$ the distance between these two values. In case of a qualitative parameter, let v be the number of variates, the possible values being then the integers from 1 to v. According to the parameter's type, $d(a, b)$ is defined as follows:

$$d(a,b) = |a - b|\,, \tag{1}$$

$$d(a,b) = \frac{|a - b|}{v - 1}, \tag{2}$$

$$d(a,b) = min(|a - b|\,, 1). \tag{3}$$

The equations (2) and (3) are used respectively for ordered and unordered qualitative parameters. In the case (1) of quantitative parameters, getting a distance between 0 and 1 requires a step of normalization so that the possible values range from 0 to 1. We use a min-max normalization, where the minimum and maximum bounds are defined from experts or using statistical analysis of training sets. All values are then restricted to these bounds, lower and upper values being interpreted as noisy or erroneous. Let X_{min} and X_{max} be respectively the minimum and maximum bounds for the values x of a given parameter X. We define the normalized value $norm(x)$ of x as follows:

$$norm(x) = \frac{max\left(0, min\left(x, X_{max}\right) - X_{min}\right)}{X_{max} - X_{min}}$$

4.2 Actual distance between time-series

The similarity function between trajectories is based on the *Longest Common Subsequence (LCSS)*, already used by Vlachos *et al.* [32] in the context of multidimensional (generally two or three dimensional) time-series of quantitative data. Indeed, dealing with noisy data (see 3.2) have proved to be better handled using non-metric [1, 8, 32], based on the *LCSS*, than metric distances, like the Euclidean distance [7], or the *Dynamic Time Warping (DTW)* [16, 17]. Using *LCSS*, the overall idea is to count the number of couple of points from two sequences A and B that matches according to a predefined matching threshold ϵ, and when going through the temporal sequences (see figure 4). One point can never be associated twice to a point of the other sequence, so that the maximum number of associations is the

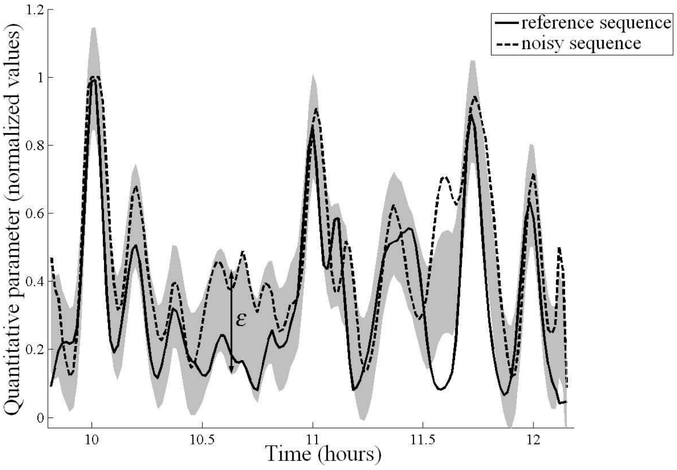


Figure 4: **The notion of the *LCSS* matching within a region of ϵ.** Comparing the trajectories point to point along the time axis, the pairs both within the gray region can be matched.

minimum length of the two sequences. Another constant δ controls how far in time we can go in order to match points from one trajectory to the other one (see figure 5).

We assume objects are points moving in a p-dimensional space $(x_1, \dots, x_p)$. Let A and B be the two trajectories of moving objects with size n and m respectively:

$$A = ((a_{x_1,1}, \dots, a_{x_p,1}), \dots, (a_{x_1,n}, \dots, a_{x_p,n})),$$

$$B = ((b_{x_1,1}, \dots, b_{x_p,1}), \dots, (b_{x_1,m}, \dots, b_{x_p,m})).$$

For a trajectory A, let $Head(A)$ be the sequence:

$$Head(A) = ((a_{x_1,1}, \dots, a_{x_p,1}), \dots, (a_{x_1,n-1}, \dots, a_{x_p,n-1})).$$

Given an integer δ and a real number $0 < \epsilon < 1$, the similarity function $LCSS_{\delta,\epsilon}(A, B)$ is defined using the recurrent algorithm (4) [32]. N and M are the size of the sequences A and B respectively at the first step of the recurrent algorithm.

$$LCSS_{\delta,\epsilon}(A,B) = \begin{cases} 0 \quad \text{if } A \text{ or } B \text{ is empty}, \\ \\ 1 + LCSS_{\delta,\epsilon}(Head(A), Head(B)), \\ \quad \text{if } d\left(a_{x_k,n}, b_{x_k,m}\right) < \epsilon, \ \forall 1 \leq k \leq p, \\ \quad \text{and } |n-m| \leq \delta \text{ and } |N-n-M+m| \leq \delta, \\ \\ max\left(LCSS_{\delta,\epsilon}(Head(A), B), LCSS_{\delta,\epsilon}(A, Head(B))\right) \\ \quad \text{otherwise.} \end{cases} \tag{4}$$

Our similarity measure differs from the one proposed by Vlachos *et al.* [32] in two ways: (1) we have integrated a new temporal constraint on δ to better control how far in time we can go in order to match points, starting from the end of the subsequences – $|N - n - M + m| \leq \delta$; and (2) we have extended the similarity measure to the consideration of heterogeneous parameters. The constraint on values for similarity is based on the distance between points defined for each type of parameters in the previous paragraph 4.1. We have also defined a relevant ϵ threshold on these distances according to the parameter's type, considering that

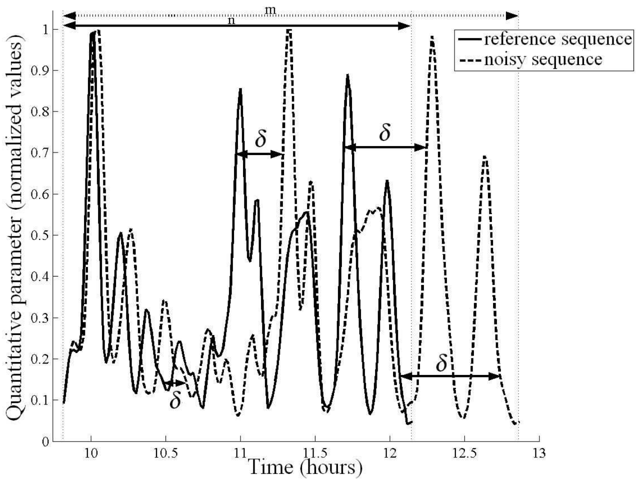


Figure 5: **The notion of the *LCSS* matching within a region of δ.**
The points of two trajectories can be matched if the time interval is under the maximum authorized value for δ.

two values of a qualitative parameter are similar only if they are equal:

- Quantitative $0 < \epsilon < 1$,
- Ordered qualitative $\epsilon = \frac{1}{v-1}$,
- Unordered qualitative $\epsilon = 1$.

The number of matching is normalized by the minimum length of the two trajectories, so that the similarity measure ranges from 0 to 1. Therefore the function $D_{\delta,\epsilon}(A,B)$ between the two trajectories A and B is defined as follows [32]:

$$D_{\delta,\epsilon}(A,B) = 1 - \frac{LCSS_{\delta,\epsilon}(A,B)}{min(n,m)}.$$

$D_{\delta,\epsilon}(A,B)$ verifies the properties of a distance.

4.3 Minimum distance

The minimum distance between time-series is a low approximation of the actual distance, which is interestingly used for getting a rough idea of the similarity between two sequences. Computing a minimum distance of zero between two subsequences means that they can be considered as quite similar. In our context, we use and interpret this information in terms of allowing for temporal aggregation of a sequence of vectors, when close to the corresponding "mean vector sequence" (see 5.1.2). Chiu *et al.* [7] have defined a minimum distance to roughly compare discrete one-dimensional quantitative temporal sequences, in a purpose of classification. In our context, such a distance may also be of great interest to perform temporal aggregation, giving an idea of whether a subsequence can be approximated by its mean vector or not. We then need to extend the minimum distance of Chiu *et al.* [7] to allow for heterogeneous multivariate time-series.

The definition uses the values of breakpoints defining the discrete intervals of values. Let $B = \beta_1, ..., \beta_{a-1}$ be the sorted list of breakpoints for a given quantitative parameter discretized in a symbols α_1,...,α_a (β_0 and β_a are defined as $-\infty$ et $+\infty$ respectively). A sequence $C = c_1, ..., c_n$ of length n can be transformed into a symbolic representation as a word $\hat{C} = \hat{c}_1, ..., \hat{c}_\omega$ where $\hat{c}_i = \alpha_j$ iff $\beta_{j-1} \leq c_i < \beta_j$. Using the principle of Euclidean distance, the minimum distance between the original time-series Q and C of two words $\hat{Q}$

and $\hat{C}$, so called $mindist(\hat{Q}, \hat{C})$, is defined by the following equation [7]:

$$mindist(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{\omega}} \sqrt{\sum_{i=1}^{\omega} (d(\hat{q}_i, \hat{c}_i))^2}.$$

The distance function $d(\alpha_i, \alpha_j)$ between two symbols α_i and α_j of a given ordered alphabet corresponding to discretization intervals, $1 \leq i, j \leq a$, is defined using the values of the corresponding breakpoints, as follows [7]:

$$d(\alpha_i, \alpha_j) = \begin{cases} 0 & \text{if } |i - j| \leq 1 \ , \\ \beta_{max(i,j)-1} - \beta_{min(i,j)} & \text{otherwise.} \end{cases} \tag{5}$$

The implementation of a lookup table used to define the distance between words made up of such symbols is illustrated in table 1.

	α_1	α_2	α_3	α_4
α_1	0	0	0.25	0.57
α_2	0	0	0	0.32
α_3	0.25	0	0	0
α_4	0.57	0.32	0	0

Table 1: **Distance between discrete symbols representing time-series.**
Lookup table used to compute the minimum distance between two words for an alphabet of cardinality 4, $\alpha_1, ...\alpha_4$, defined by discretization using the breakpoints $\beta_1 = 0.12$, $\beta_2 = 0.37$, and $\beta_3 = 0.69$. The distance between two symbols can be read off by examining the corresponding row and column. For instance $d(\alpha_1, \alpha_2) = 0$ and $d(\alpha_1, \alpha_3) = 0.25$.

To extend this notion of minimum distance to multidimensional heterogeneous time-series, we use the distance function between points defined in section 4.1 for qualitative parameters. The implementation of a lookup table used to define the distance between sequence of qualitative symbols is illustrated respectively in tables 2 and 3.

	α_1	α_2	α_3	α_4
α_1	0	1	1	1
α_2	1	0	1	1
α_3	1	1	0	1
α_4	1	1	1	0

Table 2: **Distance between symbols from an unordered alphabet.**
Lookup table used to compute the minimum distance between two words for an unordered alphabet of cardinality 4, $\alpha_1, ...\alpha_4$.

Let $C = ((c_{1,1}, ..., c_{1,p}), ..., (c_{n,1}, ..., c_{n,p}))$ be a p-dimensional heterogeneous time-series represented by the sequence of symbols $\hat{C} = ((\hat{c}_{1,1}, ..., \hat{c}_{1,p}), ..., (\hat{c}_{n,1}, ..., \hat{c}_{n,p}))$. Using the relevant function $d(\hat{q}_{i,j}, \hat{c}_{i,j})$ according to the type of component j – equation (2), (3), or (5) using the normalized quantitative values – the minimum distance between two original p-dimensional time-series Q and C represented as $\hat{Q}$ and $\hat{C}$, so called $mindist(\hat{Q}, \hat{C})$, is then re-defined by the following equation:

$$mindist(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{\omega}} \sqrt{\sum_{i=1}^{\omega} \left(\sum_{j=1}^{p} (d(\hat{q}_{i,j}, \hat{c}_{i,j}))^2 \right)}.$$

	α_1	α_2	α_3
α_1	0	0.5	1
α_2	0.5	0	0.5
α_3	1	0.5	0

Table 3: **Distance between symbols from an ordered alphabet.** Lookup table used to compute the minimum distance between two words for an ordered alphabet of cardinality 3, $\alpha_1 < \alpha_2 < \alpha_3$.

5 Proposed approach for pattern extraction

In that section we describe the proposed approach for recurrent pattern extraction. The schema of figure 6 summarize the successive steps identified in section 3.3 for an unsupervised learning of meaningful "high-level patterns" from heterogeneous multivariate time-series, detailed in the following paragraphs: (§5.1) **Representation** of time-series, (§5.2) **Feature mining** for tentative motifs discovery, and (§5.3) **Clustering** for time-series motifs final identification.

5.1 Representation of time-series

Time-series representation is really important because of the difficulty of directly manipulating continuous, and especially heterogeneous, high-dimensional and possibly noisy data in an efficient way. Defining a suitable representation aims at reducing feature space dimension in order to get an efficient feature mining for running the learning task. Many time-series representation have been introduced, including the Discrete Fourier Transform (DFT), the Discrete Wavelet Transform (DWT), Piecewise Linear and Piecewise Constant models (PAA, APCA), Singular Value Decomposition (SVD) (see [2] for an overview).

Since we are looking for "high-level patterns" (see §3.2) within the time-series – that is, patterns corresponding to usual activities of a person at home – our purpose of representation is to highlight the global trends within the data, while removing minor local variations. Our main concern is in fact robustness rather than accuracy of the extracted patterns. In that aim, we also try to restrict as much as possible the number of parameters involved in the process. Defining the step of representation is then guided by this purpose of getting a long-term, simple, and meaningful view of the time-series, which corresponds in fact to a step of **abstraction**.

We perform time-series abstraction in three steps to get a concise representation of the heterogeneous multivariate time-series under study: (1) preprocessing, (2) discretization, and (3) temporal aggregation. Preprocessing the time-series includes filtering, temporal reduction and alignment . Although well known and usual when analyzing data sets, this step is also really important because it at least partly governs the level of details of the analysis. The next subsections detail the two following steps of discretization and temporal aggregation. The figure 7 illustrates the results of each of these steps from a sample sequence simulated in the contexte of home health telecare.

5.1.1 Discretization

Dealing with heterogeneous time-series, using a symbolic or discrete representation of the lower-level data is interesting to build homogeneous data sets for feature mining. This requires the **discretization** of the continuous components of time-series. Several methods have already been experimented like Piecewise Aggregate Approximation (PAA) to produce

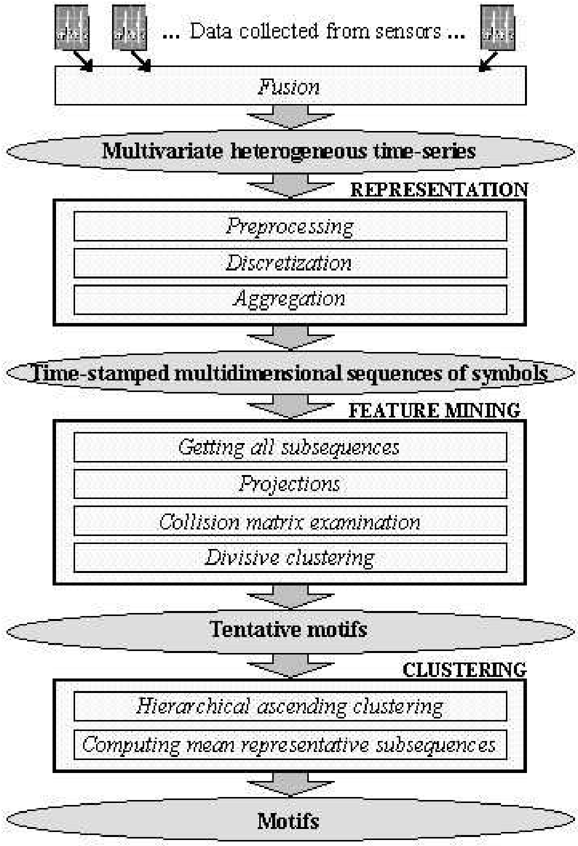

... Data collected from sensors ...
Fusion
Multivariate heterogeneous time-series
REPRESENTATION
Preprocessing
Discretization
Aggregation
Time-stamped multidimensional sequences of symbols
FEATURE MINING
Getting all subsequences
Projections
Collision matrix examination
Divisive clustering
Tentative motifs
CLUSTERING
Hierarchical ascending clustering
Computing mean representative subsequences
Motifs

Figure 6: **Successive steps of motifs extraction from data collected from sensors,** including: (1) Representation, (2) Feature mining, and (3) Clustering.

symbols of equiprobability [7, 22], clustering using k-means [9] or Dynamic Local K-means (DLK) – DLK learns the number of the classes with subject to the constraint that the variance of each class is less than a given sigma-zero [12]. Given that equiprobability in symbols is not necessarily relevant considering monitoring purposes – unusual values need to be distinguishable from usual ones – we use the standard k-means technique on experimental data sets to define the discretization intervals for quantitative parameters.

5.1.2 Temporal aggregation

Once we get discrete time-series, a further step in reducing the feature space dimension is to perform **temporal aggregation**, where aggregate vectors – either called *symbols* – are computed over time-line partitions. The main interest is to get a concise representation of time-series, allowing stretching in time between subsequences represented by a same number of symbols, and possibly similar in terms of this aggregated representation. In general, temporal grouping is done by two types of partitioning [26]: (1) span grouping, based on a defined length in time, and (2) instant grouping, which depends on the data stored. Various techniques have already been proposed and applied to issues where several time-series of same parameters are recorded during overlapping time-intervals [24, 26]. The aim is then to summarize the time variations over only one possibly multidimensional sequence of values.

In our context, the issue is quite simpler because it only aims at partitioning one multidimensional sequence of discrete values into a time-stamped sequence of vectors which

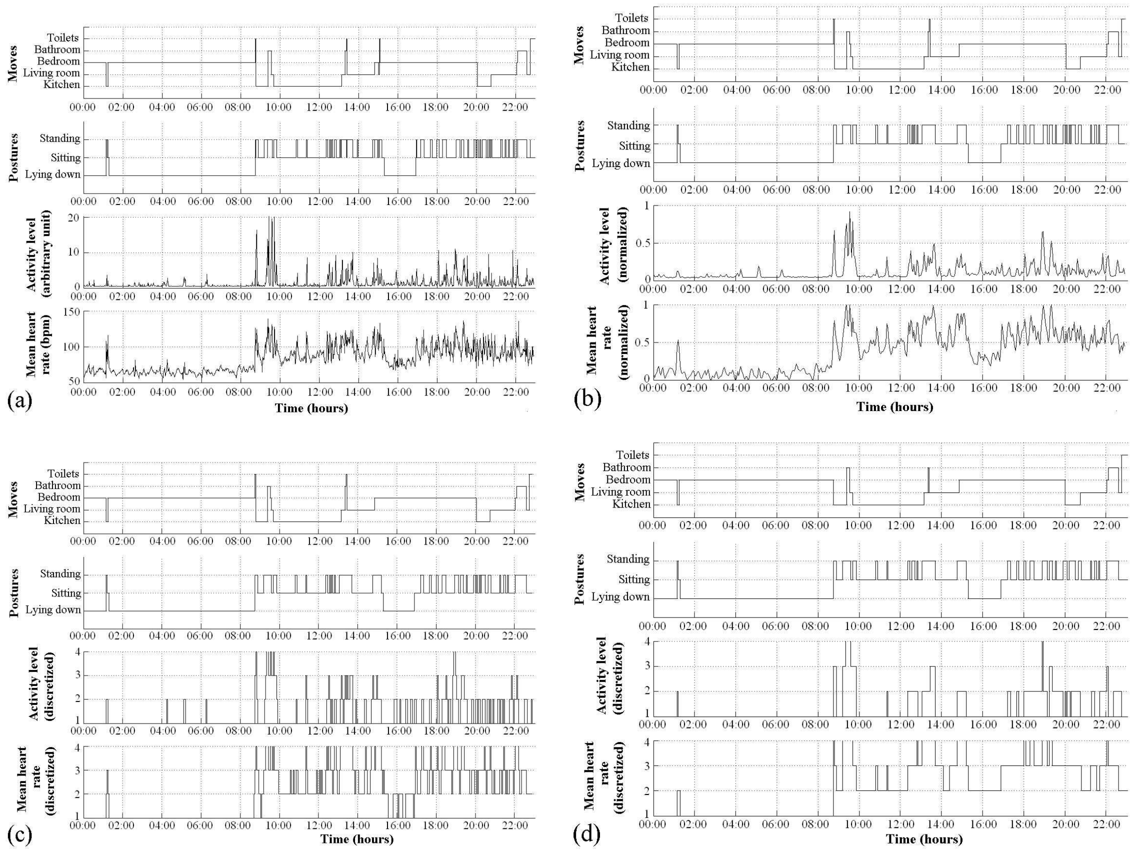


Figure 7: **Abstraction steps performed before analyzing time-series.**
Graphs representing some sequences of 4-dimensional time-series for a person monitored at home over one day, including for each graph the following parameters (from top to bottom): (1) moves, (2) postures, (3) activity levels, and (4) mean heart rate.
From left to right, and from top to bottom, the graphs represent: (a) **Raw data**, produced by a simulation process; (b) **Preprocessed data**, that raw data smoothed by a mean filter, following by temporal reduction, and normalization; (c) **Discretized data**, the discretized intervals being defined by the k-means technique, and (d) **Aggregated data**.

summarizes the global trends of variation. **Span aggregation** may be performed while pre-processing the time-series using sliding windows of fixed-length to mean the data and possibly reduce the sampling rate in the same time. The choice of sampling rate is important to determine the precision and thus the interesting level of details of time-series. A more challenging and also common issue is **instant aggregation**, depending on the variation of the values in time. Since our interest is in observing global trends in the time-series, we need to compute aggregate vectors within time-intervals where there are no significant variations in the multidimensional values (that is, vectors). We use a technique based on a distance threshold using an extension of the minimum distance between time-series proposed in [7]. The great interest in using a minimum distance (see §4.3) is the ease of getting a low approximation of the actual distance, which can be intuitively interpreted in terms of relative global trends of variation between time-series. In order that the aggregation of a sequence into one time-stamped vector is performed only for successive similar vectors, we decide that the minimum distance between the original and aggregated sequences must not be over zero for allowing aggregation. Considering the heterogeneous case, a minimum distance of zero means values of quantitative parameters are similar along time – that is within adjacent discretized intervals – and values of qualitative parameters are the same.

The aggregate vector $\widehat{aggr(C)}$ of a p-dimensional sequence of length n,

$$C = ((c_{1,1}, ..., c_{1,p}), ..., (c_{n,1}, ..., c_{n,p})),$$

is defined as the discretized mean value computed along time for each component:

$$\widehat{aggr(C)} = (mean\,(\widehat{c_{1,1}, ..., c_{n,1}}), ..., mean\,(\widehat{c_{1,p}, ..., c_{n,p}})).$$

The mean value of a symbolic sequence corresponds to the most represented symbol within the whole subsequence. The following equation express the condition required for aggregating the vectors of a subsequence C:

$$mindist(\hat{C}, \widehat{AGGR}(C)) = 0.$$

$\hat{C}$ is the discretized sequence of C; $\widehat{AGGR}(C)$ is a sequence of same length n made up of the repeated aggregate vector,

$$\widehat{AGGR}(C) = (\widehat{aggr(C)}, ..., \widehat{aggr(C)});$$

and $mindist()$ is the minimum distance whose definition is extended from the one proposed by from Chiu *et al.* [7]. Starting from the first point of the sequence, we look for the longest time-intervals where temporal aggregation is allowed according to the previous definitions. At the end, the original time-series is then represented by a sequence of multidimensional vectors, either called symbols, each of them lasting for a specific duration.

5.2 Feature mining: tentative motifs discovery

Feature mining aims at selecting the most relevant features to feed into a learning task, so that we reduce the size of the feature space. In the context of time-series motifs discovery, the purpose is to extract the most relevant subsequences – the *tentative motifs* – used as input for the final identification and classification of frequent patterns – the *motifs*.

Several papers dealing with motifs extraction use a kind of feature mining step to first select the potential location of frequent patterns within the time-series, and then refine the motifs identification. For instance, Hong *et al.* [13] have trained recurrent neural nets to extract temporal patterns candidates. They correspond to subsequences where the trained network can continuously give out correct one-step prediction. In that paper we extend the probabilistic approach experimented in [7] for tentative motifs extraction, and illustrated on figure 8. Time-series motifs candidates – that is *tentative motifs* – are identified from random projections of all the possible subsequences extracted using a sliding window from the original and symbolized time-series. The tentative motifs correspond to subsequences that are often hashed into the same bucket using a mask randomly chosen. Each step of projection increases the counts in a *collision matrix*, a square matrix whose size corresponds to the number of all possible subsequences of a predefined length. A large value of collisions is a strong indicator of two similar subsequences, that is good candidates for motifs. Since time-series are discretized into symbols of constant frequency, this method does not allow for stretching in time between motifs instances. Moreover, Chiu *et al.* have implemented the *projection* algorithm [3] only for one-dimensional real time-series.

Actually, the *projection* algorithm is interesting because of its ability to roughly identify possible instances of motifs in time-series, allowing some noise and imprecision in the discrete sequences representing the original time-series. In our complex issue involving several levels of details from raw data to decision, this algorithm then acts as a stage of feature meaning

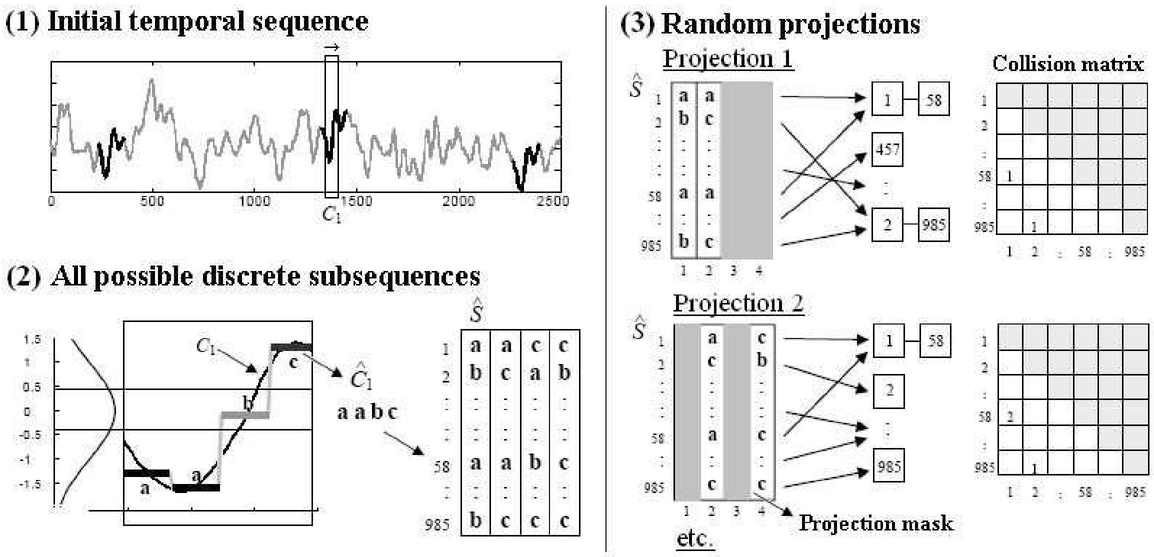


Figure 8: **Principle of using the *projection* algorithm as experimented in [7].** The successive steps are as follows :

(1) A sliding window is defined to extract subsequences C_1 from the initial temporal sequence ;

(2) Each sequence C_1 is converted into its discrete representation $\hat{C}_1$ and placed into matrix $\hat{S}$;

(3) A mask is randomly chosen, so that only part of the discrete values were used to project the matrix $\hat{S}$ into buckets. Collisions are recorded by incrementing the appropriate location in the collision matrix.

to extract the best subsequences that are candidates to motifs from the original time-series represented as a discrete sequence of symbols. We then extend this approach to deal with heterogeneous multivariate time-series. The context of using the *projection* algorithm is also changed so that we can discover motifs whose instances are of different lengths. Using as input discrete sequences obtained with representation techniques including temporal aggregation of symbols (see §5.1.2) addresses this issue.

Using the *projection* algorithm allows to extract subsequences representative of frequent patterns within discrete time-series. This criterion is however not enough to deal with feature mining – that is tentative motifs extraction. According to Lesh *et al.* [20], the criteria for selecting features might depend on the domain and the classifier being use. However, they believe that the following domain- and classifier-independent heuristics are useful for selecting sequences to serve features: (1) Features should be frequent, (2) Features should be distinctive of at least one class, and (3) Feature sets should not contain redundant features.

(1) The first heuristic is clearly insured because subsequences extracted using the *projection* algorithm at least partially matches another subsequence due to their extraction from the collision matrix produced by the successive projections.

(2) The second one cannot be encoded directly from projections because this approach to pattern extraction is rough and unsupervised. Having features distinctive of at least one class is then ensured by three additional steps when examining the collision matrix. First, because a large value of collisions is only a strong indicator of similar subsequences, we go back to the original, preprocessed, time-series to refine the

comparison between pairs of subsequences. Second, because a motifs' instances may be of different lengths regarding the number of corresponding discrete symbols, each instance may be represented by several successive *basic* subsequences. Consequently, we propose to extend matching pairs of subsequences while examining the collision matrix, as long as they are similar, in order to match the whole patterns. Third, we add a constraint on the minimum duration of so extracted subsequences to insure that features are relevant in terms of the person's behavioral profile, and consequently probably in terms of being an instance of a motif.

(3) At last, the third heuristic requires synthesizing the set of subsequences identified from the previous steps in order to get a set of non-overlapping features – that is, subsequences – that are the most representative of each group of overlapping subsequences. The need for synthesizing the results of collision matrix examination has not been put forward in related papers. This stage then ends feature mining, that is the tentative motifs extraction, with respect to the third heuristics of Lesh *et al.* [20].

Feature mining then includes three main steps, detailed in the next subsections:

1. **Time-series random projections**, once represented using discretization and aggregation techniques,
2. **Collision matrix examination** to extract frequent and relevant subsequences in terms of identifying motifs, and
3. **Tentative motifs extraction** by identifying the most relevant, non-overlapping, subsequences from the previous set.

5.2.1 Time-series random projections

Time-series random projections produce a *collisions matrix* recording integers representative of the number of matches between all the possible subsequences extracted from the original sequence. A large value in a cell is not the guarantee of the existence of a corresponding motif, but it is a strong indicator [7]. The diagram of figure 9 sums up random projections principle, according to the following steps:

(1) Preprocessing. Let C be a p-dimensional sequence of n values recorded regularly over time:

$$C = ((c_{1,1}, ..., c_{1,p}), ..., (c_{n,1}, ..., c_{n,p})).$$

(2) Abstraction. Time-series are first represented using the techniques presented in section 5.1 – preprocessing, discretization, and temporal aggregation. Sequence C is then represented by a time-stamped sequence of N symbols – the p-dimensional vectors $(\hat{q}_{i,1}, ..., \hat{q}_{i,p})$, $1 \leq i \leq N$, $N \leq n$:

$$\hat{C} = (((\hat{q}_{1,1}, ..., \hat{q}_{1,p}), t_1), ..., ((\hat{q}_{N,1}, ..., \hat{q}_{N,p})), t_N),$$

where $(t_1, ..., t_N)$ are the ordered instants of symbols occurrence along time.

(3) Random projections

(a) Basic subsequences. Random projections are performed from so called *basic subsequences* of a specified length w, extracted from the original sequence using a sliding window of size w. This produce a matrix $\hat{S}$ of size $(N - w + 1) \times w$.

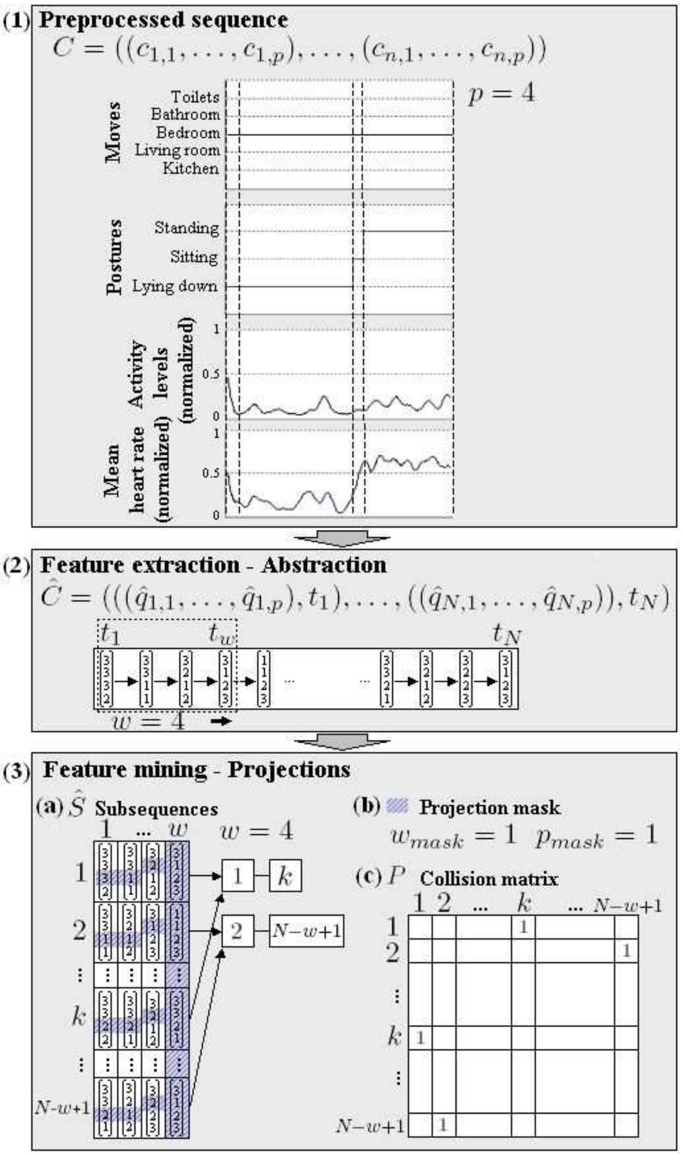

(1) Preprocessed sequence
$C = ((c_{1,1}, \ldots, c_{1,p}), \ldots, (c_{n,1}, \ldots, c_{n,p}))$
$p = 4$
Moves
Toilets
Bathroom
Bedroom
Living room
Kitchen
Postures
Standing
Sitting
Lying down
Activity levels (normalized)
Mean heart rate (normalized)
1
0.5
0
(2) Feature extraction - Abstraction
$\hat{C} = (((\hat{q}_{1,1}, \ldots, \hat{q}_{1,p}), t_1), \ldots, ((\hat{q}_{N,1}, \ldots, \hat{q}_{N,p})), t_N)$
t_1
t_w
t_N
$w = 4$
(3) Feature mining - Projections
(a) $\hat{S}$ Subsequences
1 ... w
$w = 4$
1
2
k
N-w+1
$N-w+1$
(b) Projection mask
$w_{mask} = 1$ $p_{mask} = 1$
(c) P Collision matrix
1 2 ... k ... N-w+1

Figure 9: **Principle of feature mining, once original time-series are represented.**

(b) Projection mask. We randomly select w_{mask} columns of $\hat{S}$ to act as a mask. In a p-dimensional context we need a random mask of size $w_{mask} \times p_{mask}$, where w_{mask} and p_{mask} are integers such as $0 \leq w_{mask} \leq w$ and $0 \leq p_{mask} \leq p$. For example in figure 9, where $w = 4$ and $p = 4$, we have randomly selected column number 4 ($w_{mask} = 1$) to act as a mask on symbols, and parameter number 3 for symbols 1 and 2, such as parameter number 2 for symbol 3 ($p_{mask} = 1$).

(c) Collision matrix. The $(N - w + 1)$ words in the $\hat{S}$ matrix are hashed into buckets based on their non-masked values. In the sample of figure 9, all possible couple of subsequences are compared based on their 1^{nd}, 2^{nd}, and 3^{nd} symbols, considering the 1^{st}, 2^{nd} and 4^{th} parameters for symbols 1 and 2, and the 1^{st}, 3^{rd} and 4^{th} parameters for symbol 3. This produces the *collision matrix* P of size $(N - w + 1) \times (N - w + 1)$, built as follows: if two words corresponding to p-dimensional subsequences i and j are hashed to the same bucket, we increase the count of cell (i, j) in P, previously initialized to all zeros – $P(i, j) = P(i, j) + 1$.

We need to repeat the two last steps (b) and (c) an appropriate number of times so that the collision numbers are statistically significant. The relevance of the collision numbers also depends on the parameters being well selected according to the purpose of motifs extraction.

The key parameters of projection are as follows: (1) number of vectors per subsequence (w), (2) number of vectors defining the first dimension of the projection mask (w_{mask}), (3) number of parameters defining the second dimension of the projection mask (p_{mask}), (4) number of projections performed ($proj$).

5.2.2 Collision matrix examination

Once the collision matrix is significantly filled in, we examine iteratively its values from the largest one to find promising candidates for motifs extraction. We stop the examination when the next value not already examined, and not within the scope of previously reported tentative motifs, is lower than a predefined threshold defining the minimum "large enough" number of collisions, so called the ***collision threshold***. A large value of collisions is only a strong indicator of two similar subsequences, and we go back to the original data to possibly confirm we met a *tentative motif.* The comparison of the original, preprocessed, subsequences corresponding to tentative motifs is performed from the similarity measure defined in section 4 for heterogeneous multivariate time-series. A threshold on this measure, so called the ***distance threshold***, is used to decide whether or not two subsequences can be considered similar enough so that they define tentative motifs.

In order to look for the *whole* and significative patterns, independently of the number w of symbols defining each *basic subsequence* used as input for projections, we perform "pattern growing" – as called in [13] in another methodological context – while examining the collisions matrix. Considering a pair of similar discrete *basic* subsequences in terms of collision number and actual distance, we define "pattern growing" as the consideration of extended subsequences including the basic ones. We try to extend them on their right and left sides – that is before and after the respectively first and last symbols in time – while (1) the numbers of collisions corresponding to the extended area is still large enough – that is over the ***collision threshold***, and (2) the similarity between the extended original subsequences does not overpass the ***distance threshold***.

Moreover, we define some *neighbourhoods of collisions* to allow for noise between recurrent subsequences and imprecision in the abstraction step. Indeed, in that noisy and imprecise context, all basic subsequences that made up a whole motif instance do not generate collision values over the predefined threshold when compared to other instances. An adaptive algorithm is then defined from the observation of a high collisions number, as illustrated on figure 10 : (1) A finest identification of recurrent basic subsequences is performed by finding out the lowest actual distance, under the maximum threshold, between pairs of basic subsequences in a *close collision neighbourhood* that verifies the minimum collisions criteria ; (2) Pattern growing is then performed when the collision criteria is verified in the *neighbourhood of possible extensions* and the actual distance between the extended subsequences is under the maximum threshold.

We then produce a group of subsequences of different lengths in terms of symbols and/ or number of points regarding the original subsequence.

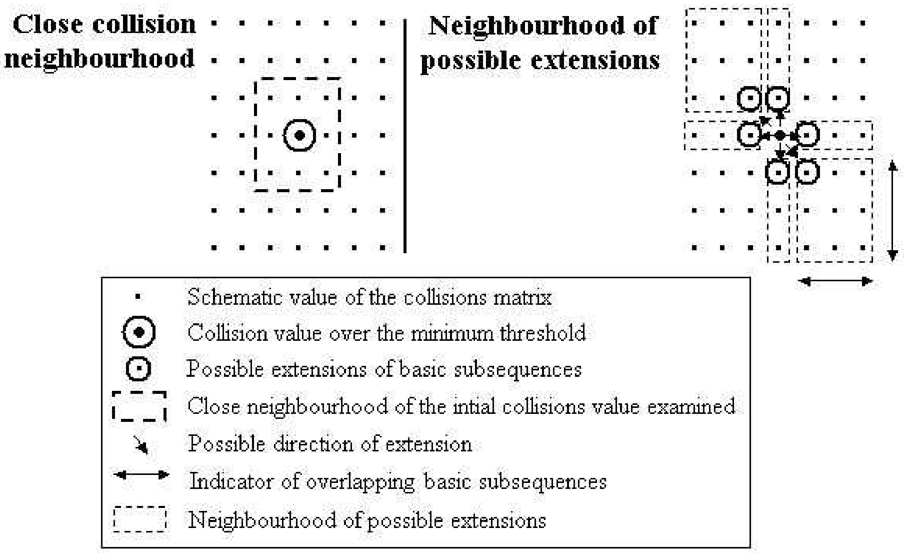

Close collision neighbourhood
Neighbourhood of possible extensions
Schematic value of the collisions matrix
Collision value over the minimum threshold
Possible extensions of basic subsequences
Close neighbourhood of the intial collisions value examined
Possible direction of extension
Indicator of overlapping basic subsequences
Neighbourhood of possible extensions

Figure 10: **Neighbourhoods of collisions used to identify and extend basic recurrent subsequences from the collision matrix.**

5.2.3 Tentative motifs extraction

Feature mining aims at identifying non-redundant features [20], that is a set of non-overlapping subsequences from the whole original sequence that are the most appropriate to motifs extraction. However, the group of subsequences previously extracted from the collision matrix examination may contain overlapping ones because they are extracted by pairs regarding an area of high values within the collisions matrix. We then need to identify relevant groups of subsequences that are well-separated in time, so that we can define at last the tentative motifs, one corresponding to each group.

Considering a group of k subsequences, one overlapping all the other ones of the group, and each of them ranging from indexes $t_{j,1}$ to t_{j,n_j} $(1 \leq j \leq k,\ n_j > 1)$, where $t_{j,1} < t_{j,n_j}$ regarding the original sequence, the *tentative motif* representative of this group is defined by the subsequence ranging from indexes t^i to t^f, where:

$$t^i = min\left(\{t_{j,1}\}_{1 \leq j \leq k}\right) \text{ and } t^f = max\left(\left\{t_{j,n_j}\right\}_{1 \leq j \leq k}\right). \tag{6}$$

The idea is indeed to consider the collision and distance thresholds as restrictive enough so that pattern identification and growing from the collision matrix examination only results in defining significant tentative motifs. However, a *tentative motif* met several times as matching different subsequences may not be extended enough any time because of possible noisy data or imprecision in frequent subsequences identification. The largest subsequence regarding all overlapping subsequences must then be considered as the *tentative motif.*

However, frequent subsequences identified following the collision matrix examination may not be directly divided into well-separated groups of subsequences, where each group contains subsequences overlapping all the other ones. For instance, a subsequence corresponding to a large collision number regarding another subsequence might be hazardously too much extended in reference to the effective location of the corresponding motif. Regarding the group of overlapping subsequences containing this “too long” subsequence, removing this subsequence may result in the corresponding group being in fact possibly divided into two groups of well-separated subsequences. The best way of defining the corresponding *ten-*

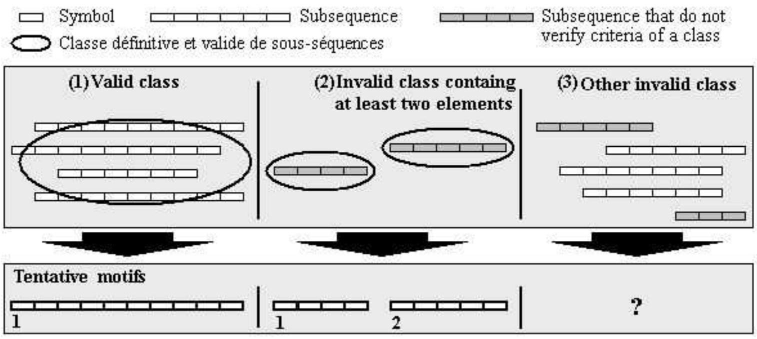

Symbol
Subsequence
Subsequence that do not verify criteria of a class
Classe définitive et valide de sous-séquences
(1)Valid class
(2)Invalid class containg at least two elements
(3)Other invalid class
Tentative motifs
1
1
2
?

Figure 11: **Illustration of possible cases when examining the validity of a class of tentative motifs.**

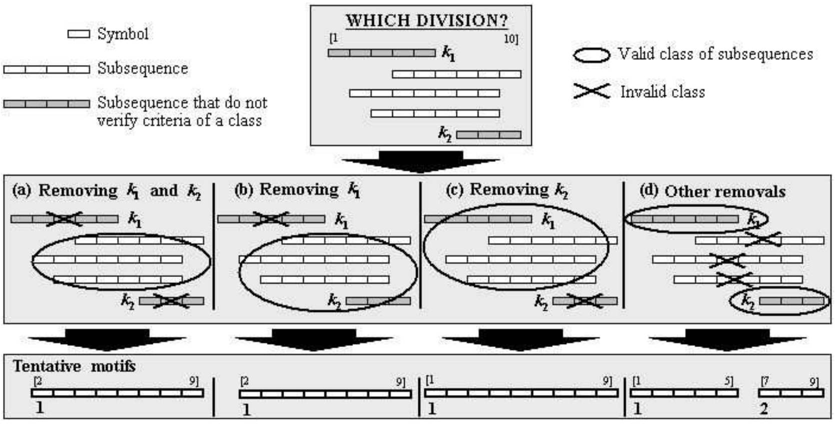


Figure 12: **Illustration of possible divisions of an invalid class containing at least three subsequences.**
A given class is considered as *invalid* when at least two subsequences do not overlapp, so called k_1 and k_2.

tative motif is then to remove the consideration of the longest subsequence and to identify at the end two tentative motifs corresponding to each well-separated group of subsequences. Possible cases met when examining a group of tentative motifs are illustrated on figure 11.

Thus, defining the most relevant tentative motifs considering the results of the collision matrix examination raises an issue of clustering sets of subsequences. Within each set, one subsequence overlaps at least one other subsequence belonging to the same set, and no subsequence from any other set. Clustering is an unsupervised data analysis technique which searches to separate data items, having similar characteristics, in constituent groups. The most common clustering methods are partitioning, hierarchical agglomerative or hierarchical divisive ones [6]. Agglomerative techniques start usually with single member clusters, whereas divisive methods begin with all cases in one large cluster. The divisive algorithm then subdivides it until some tests are satisfied. In theory, these could continue until there are t clusters each containing one object, but in practice they usually stop at an earlier stage. Divisive methods are however more expensive that agglomerative ones.

We use a hierarchical divisive clustering approach to decide about the best tentative motifs to be defined from a group of overlapping subsequences extracted at the end of examining the collision matrix. Criteria available to get clusters are indeed appropriate to that type of iterative algorithm, starting from one cluster that is gradually broken down into smaller and smaller clusters [6]. At each iteration the next class to be divided is chosen, and the process is repeated until a given stop criteria is verified. The crucial elements to be defined are as follows: (1) the criteria to select the next class to be divided, (2) the method for dividing this class, and (3) the criteria to stop the successive divisions. The groups of subsequences considered as to be potentially divided are made up from the set of subsequences extracted when examining the collision matrix, as follows: any subsequence overlaps (a) at least one other subsequence from its group, and (b)no subsequence from any other group. The purpose of clustering the subsequences of each group is: any subsequence overlaps (a) all other subsequences of its group, and (b) no subsequence from any other group. The previous purpose defines the stop criteria for dividing the groups of subsequences. If a group of subsequences does not satisfy these constraints, that means some subsequences are not relevant to be considered and need to be removed from the set of frequent subsequences.

1. **Criteria of selecting the class to be divided.** The selection of the groups to be next divided depends on the subsequences which does not overlap within the initial group. The criteria used to choose the "best" division to be performed – that is also the "best" subsequence to be removed from the group so that all subsequences overlap each other – is to end the divisions with the best representative groups of subsequences in terms of their well-representation of similar trends in the time-series. That is interpreted as removing the lower number of subsequences from the initial set. The "best" division is then determined *a posteriori* considering all possible divisions. In order to prevent the algorithm from getting to an exponential running time, some optimization criteria are used to drive *a priori* the selection of the best division, and to define what is an "acceptable" division – that is an acceptable rate of subsequences removed.

2. **Method of dividing a class.** At each step of dividing a class, three cases are possible, as follows: (1) either every subsequence overlaps all the other subsequences of the class, so that no division is required; (2) or there is only two subsequences that do not overlap within the class, so that the division consists in building two classes, containing one subsequence each; (3) or the class contains at least three subsequences, including at least two subsequences that do not overlap. In that third case, one ore more subsequences need to be removed from the class so that we can get to satisfy the criteria required to stop the divisions. Let k_1 and k_2 be two non-overlapping subsequences of a class. The division can be performed in the following manners (see figure 12): (a) either removing k_1 and k_2, removing other subsequences from the group, (b) or removing k_1, (c) or removing k_2, (d) or keeping k_1 and k_2. Removing the consideration of one or more subsequences might entail that the original group is divided into well-separated groups in time. We then apply the division algorithm on the corresponding new group(s) of subsequences.

3. **Criteria to stop the successive divisions.** We stop this recursive process of division when the purpose of clustering the subsequences of a group is reached, that is: any subsequence overlaps (a) all other subsequences of its group, and (b) no subsequence from any other group.

Some results of identifying tentative motifs from the set of subsequences extracted from the collision matrix are presented on figure 13.

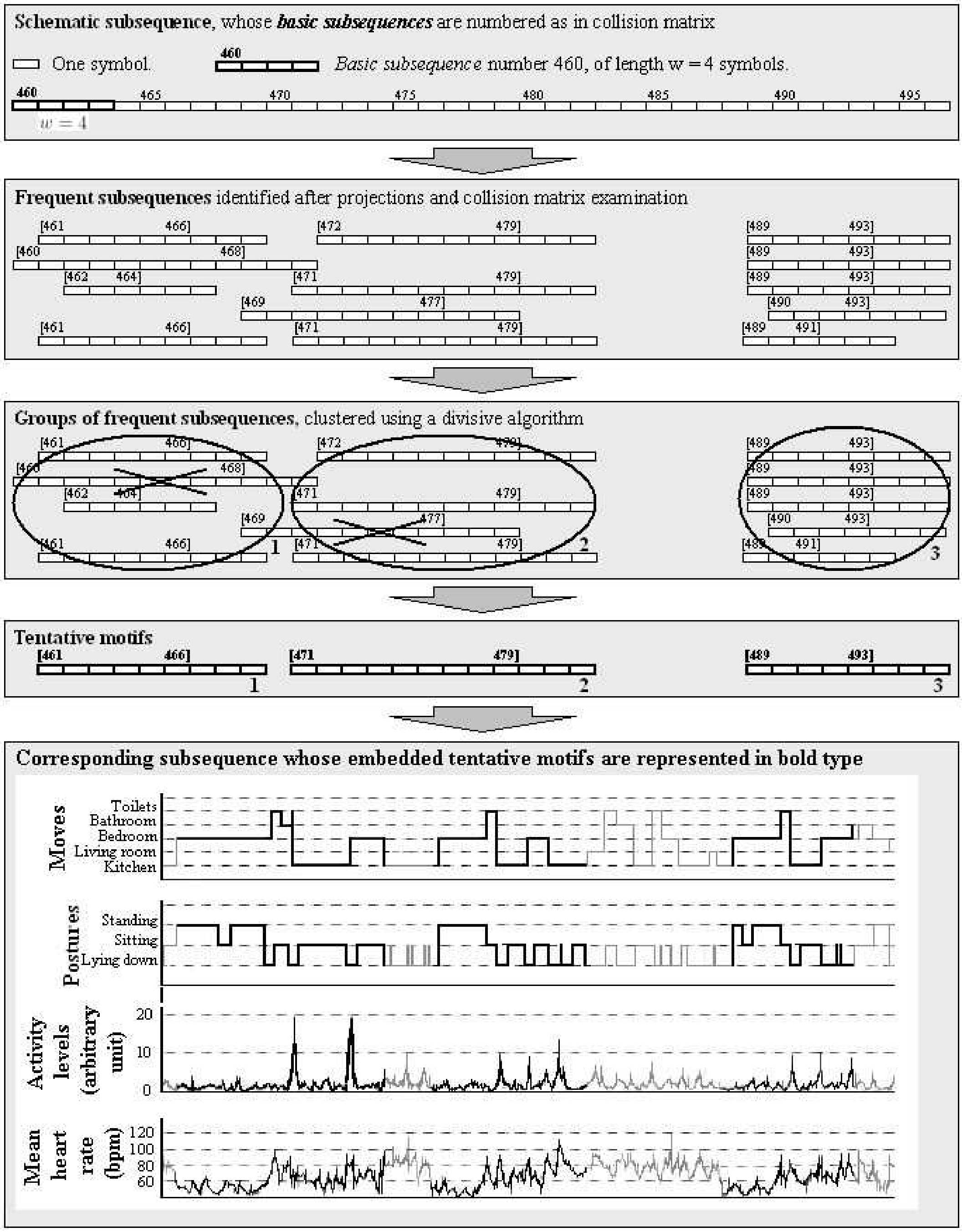
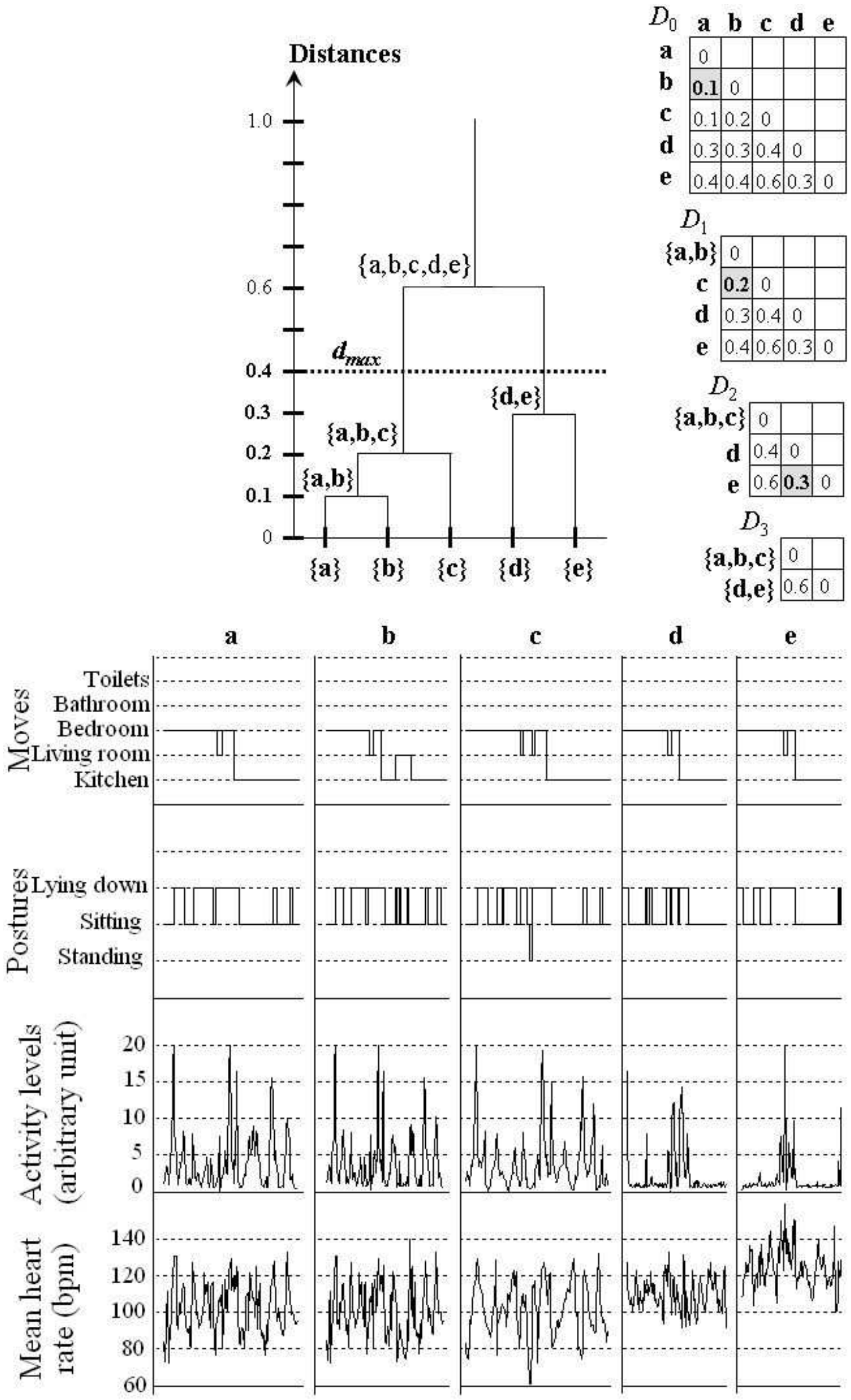

Schematic subsequence, whose basic subsequences are numbered as in collision matrix
One symbol.
460
Basic subsequence number 460, of length w = 4 symbols.
460
465
470
475
480
485
490
495
w = 4
Frequent subsequences identified after projections and collision matrix examination
[461 466]
[460 468]
[462 464]
[469 477]
[461 466]
[472 479]
[471 479]
[471 479]
[489 493]
[489 493]
[489 493]
[490 493]
[489 491]
Groups of frequent subsequences, clustered using a divisive algorithm
1
2
3
Tentative motifs
[461 466]
[471 479]
[489 493]
1
2
3
Corresponding subsequence whose embedded tentative motifs are represented in bold type
Moves
Toilets
Bathroom
Bedroom
Living room
Kitchen
Postures
Standing
Sitting
Lying down
Activity levels (arbitrary unit)
20
10
0
Mean heart rate (bpm)
120
100
80
60

Figure 13: **Tentative motifs extraction.**
Successive steps to identify tentative motifs from the set of frequent subsequences identified after collision matrix examination, using a divisive clustering method.



D_0	a	b	c	d	e
a	0				
b	**0.1**	0			
c	0.1	0.2	0		
d	0.3	0.3	0.4	0	
e	0.4	0.4	0.6	0.3	0

D_1				
{a,b}	0			
c	**0.2**	0		
d	0.3	0.4	0	
e	0.4	0.6	0.3	0

D_2			
{a,b,c}	0		
d	0.4	0	
e	0.6	**0.3**	0

D_3		
{a,b,c}	0	
{d,e}	0.6	0

Figure 14: **Ascending Hierarchical Classification.**

The tentative motifs numbered from (a) to (e) are classified into two classes ({a,b,c} and {d,e}) according to the initial distance matrix D_0 and the distance threshold $d_{max} = 0.4$. The corresponding subsequences are displayed on the bottom graphs. They have been generated by a simulation process in the context of home health telecare, including four parameters : (1) the moves of a monitored person, (2) their postures, (3) the activity level, and (4) the mean heart rate.

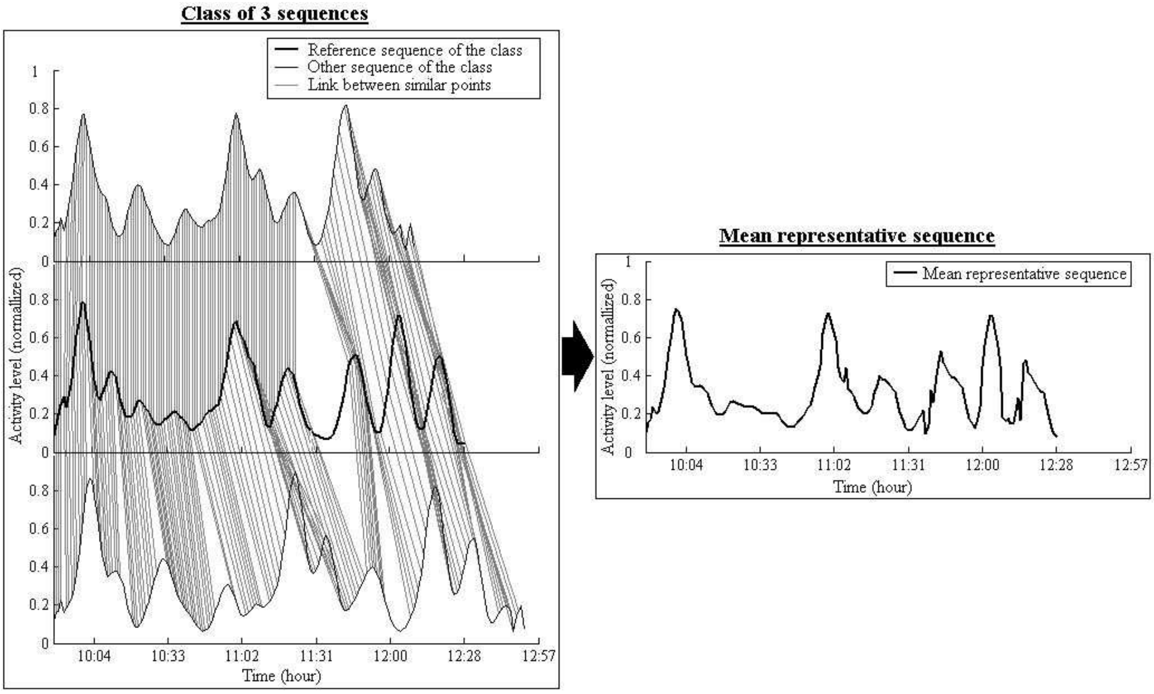


Figure 15: **Computation of the mean representative sequence of a class.** The figure above shows how to get the representant of the three one-dimensional sequences of a given class. The computation is performed from the reference sequence, that is the one whose length is closest to the mean sequences length. Each of its points is meant with similar points from the other sequences of the class.

5.3 Clustering: time-series motifs identification

The last step is the clustering of tentative motifs into classes representative of any typical "behavior". Since the last step, we need a classification method based on an accurate distance measure, that is an actual distance between the original, preprocessed, subsequences corresponding to the identified tentative motifs. We then use the similarity measure defined in section 4 for heterogeneous multivariate time-series. Our purpose is to cluster subsequences in groups whose elements are close to all the other ones belonging to the same group, that means the distances are less than a given *distance threshold*. Then, we use a **hierarchical ascending classification** from the distance table between all the tentative motifs, as illustrated on figure 14. This is an agglomerative technique which starts with single member clusters – the *tentative motifs* – successively gathered into classes according to a distance threshold. For the sake of homogeneity and robustness, we use the same distance threshold than when examining the collision matrix. The distance between two classes is defined as the maximum distance observed between all possible pairs of subsequences, one from each class. This ensures we never gather classes containing subsequences whose distance overpass the distance threshold. An additional constraint on the size of any class of *tentative motif* is also added when clustering tentative motifs to reinforce the relevance of extracted patterns.

Once the tentative motifs are clustered into meaningful groups, a mean representative subsequence corresponding to each class – that is, *motif* – is computed, as illustrated in figure 15. This representative sequence is based on the so called *reference subsequence* whose duration is the closest to the mean duration observed considering all the subsequences of a group. Since based on *LCSS* (see §4.2), computing a similarity measure between any subsequence of a group and the reference one provides sets of similar vectors associated to each vector of the reference subsequence. The representative subsequence is then defined of same length than the reference sequence, replacing each vector by its mean vector considering the set of associated similar ones.

6 Experimental Results

The approach proposed for extracting multidimensional and heterogeneous patterns is experimented in the context of home health telecare. In that section, we first define the experimental context, appropriate to the experimentation of pattern extraction. Testing our approach also requires to define an experimental process, such as relevant measures for evaluating the system's performances. At last, we can discuss the quality of the method and results.

6.1 Experimental context: home health telecare

In the purpose of monitoring a person at home, the aim is to learn the person's lifestyle in order to build a sort of behavioral profile, which is sensitive to any critical deviation. The monitoring system is based on a set of data, recorded at home and in real-time, that may be collected from different classes of sensors: (1) activity (location, position, motion, fall, etc.), (2) environment (temperature, use of doors, window, lighting, etc.), and (3) physiology (blood pressures, weight, etc.). In the definition of these observable parameters, a compromise needs to be found between (a) being easily observable and non invasive, by focusing on the observation of a small set of parameters, and (b) gaining a full appreciation of the person's condition, sensitive to any change in the health status. A deterioration of a person's health status usually entails behavioral disorders whose observable symptoms range from an increase in the risk of falls, slowness in executing simple actions, forgetfulness in daily activities, to a global decrease in the person's ability to perform activities of daily living (ADL). Clinical practice has already widely exploited this correlation by estimating a patient's health status in terms of their ability to perform ADL such as getting washed, dressing, or feeding themselves. The usefulness of monitoring some parameters related to the activity of a person is often underlined as being an essential part of any health evaluation [4, 29], and several projects in home health telecare [28, 33, 34] have already integrated in their concept the assessment of the ADL. Representative of both the activity and the health status, the heart rate is another important and easily observable physiological measure [25].

Thus, we decided to consider in a first step of experimentation four parameters that can be defined from a provision of sensors and that are representative of both the heart rate and activity of a person at home, as follows:

- **Moves:** qualitative, unordered parameter, defining the room occupied by the person at any time. The moves of the person are recorded through infrared motion sensors installed in each room.
- **Postures:** qualitative parameter, ordered according to the effort required by the posture ("lying down", "sitting", and "standing"). The postures are inferred from data provided by a set of accelerometers worn by the person.
- **Activity level:** quantitative parameter, in an arbitrary unit. The activity level is measured by a portable accelerometer worn on the chest and estimated through the body acceleration along the anterior-posterior axis [5].

- **Mean heart rate:** quantitative parameter, in beat per minute. The mean heart rate is computed from the data recorded by an ECG portable recorder.

Thus, we consider a set of heterogeneous parameters for mining meaningful patterns representative of a person's usual behaviors.

6.2 Experimental process

Setting up an experimental process requires to properly define (1) **which data sets** are appropriate to the context and purpose of the decision-making system, (2) **which method** are required to build a full experimentation, and (3) **which performance measures** are the most relevant to objectively evaluate the robustness and efficiency of the system.

6.2.1 Experimental data sets

Collecting experimental sets from a simulation process

The study of any decision-making process requires realistic and accurate data collection. Research projects about home health telecare are as yet only at their first stages of development, and collection of data in realistic environments has just started. Moreover, a full study entails consideration of several profiles of people facing many types of situations. Then, collecting complete and representative sets of data may be a quite hard task, especially to hold data corresponding to rare events. For these reasons, many researchers have turned to simulation as a way to overcome the difficulty of collecting large sets of full experimental data sets. In relation to experimentation, setting up a simulation process enables researchers to have a full and tightly controlled universe of data set, by varying the simulation parameters.

For these reasons, we have set up a simulation process for generating realistic sequences corresponding to the experimental parameters [11]: moves, postures, activity level, and mean heart rate. The simulation process is designed to preserve the problem's complexity, that is especially the joint variations of the parameters. The sequences produced by simulation must also be representative of a person's habits at home, that means they include every day subsequences corresponding to the presence of basic activities of daily living like sleeping, eating three times a day, getting washed. Considering these requirements and relative influences of simulated parameters, the simulation model is defined using a cascade structure, and run in four steps to successively generate time-series corresponding to: (1) the moves of the subject in a given period of time, (2) their successive postures, (3) the sequences of the activity levels, and (4) the values of the mean heart rate [11]. A sample of data produced by the simulation process over one day is shown on figure 16. Our objective is then to identify from these sequences of data "high level patterns" corresponding to usual behaviors of the person at home, especially under the presence of noise.

Defining appropriate sequences for experimenting pattern extraction

In order to validate our approach to pattern extraction, we need to objectively check the relevance of the motifs extracted from time-series. That requires to know *a priori* which motifs are represented within the time-series and the location of their instances, so that we can evaluate the performances of the system. The aim is then to introduce instances of predefined motifs in sequences containing at first no pattern. Due to the simulation method, the way of getting time-series containing no significant patterns while remaining realistic in terms of the joint variations of the parameters is described as follows:

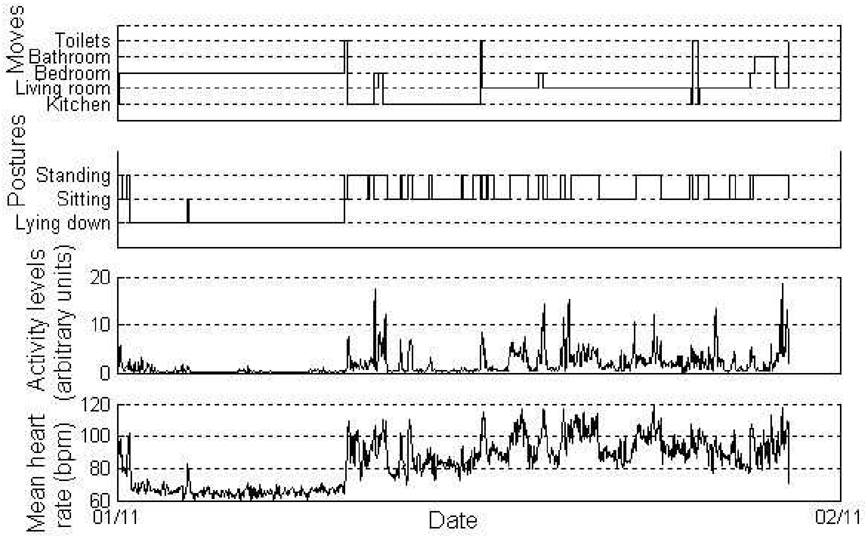


Figure 16: **Sequences produced by the simulation process over one day,** including four parameters that are, from top to bottom: (1) moves, (2) postures, (3) activity levels, (4) mean heart rate. Data are available every minute.

- First, generation of random moves, so that data include no pattern representative of the person's activities of daily living.
- Second, simulation of relevant sequences of values for the other parameters according to these random moves, so that the multidimensional sequences remain realistic anyway.

We also need to identify realistic motifs considering the monitoring of a person at home. This time, the simulation process is used to generate sequences corresponding to living habits, in which we can randomly select subsequences that can be interpreted in terms of the person performing a typical activity for a given time. The length of subsequences is randomly selected in meaningful bounds (from 30 minutes to 2 hours for instance). We also add a constraint on the selected subsequences to ensure they can be interpreted as the person performing a given activity, and not only one or two elementary action. Selected subsequences must then be represented – using the representation step defined for motifs extraction (see §5.1) – by a sequence of at least 4 symbols.

The last step is then to create and introduce some instances of these predefined motifs in the non-pattern sequences previously generated. The introduction of motifs' instances is guided by specific characteristics of patterns identified in our experimental context (see §3.2). We especially need to take account of possible imprecise matches, outliers, translation and stretching in time between instances of a same motif. Therefore, we define several types of noise that can be introduced in a subsequence representing a motif to get instances of the same motif, as follows:

- **Noisy values.** Given that we consider sequences representative of human behaviors, instances of a same motif fit only in their global trends. Then, we may define a high rate of noise in the values.
- **Interruptions.** Any main activity of daily living may be interrupted by a secondary task like answering to the phone or going to the toilets. As much as this additional

task remains of short duration, we would like to recognize the main task anyway. That requires to experiment the algorithm using instances of motifs including short sequences of *outliers*, representing disruptions in the global trend of variation.

- **Stretching in time.** Dealing with human behaviors, a same activity does not always last the same duration. That is the reason why we may introduce instances of different lengths corresponding to a same motif.
- **Translation in time.** Even usual at home, an activity is not always performed exactly at the same time. That is the reason why we introduce instances of a same motif anywhere in the non-pattern time-series.

6.2.2 Experimental method

Setting up an experimental process is relevant for evaluating the system's performance at two levels:

1. **Quality of the method:** that includes to validate each stage of the proposed approach and to define appropriate values for the parameters required. We are especially interested in defining the sensitivity of each parameter, and the way relevant values can be identify and validated.
2. **Quality of the results:** once the system is properly set up, the aim is to evaluate the performances of the system especially under the strong presence of noise. We aim at study the influence on introducing highly noisy instances of given motifs, considering all possible noise: variability in values, outliers, stretching and translation in time.

6.2.3 Performance measures

Setting up an experimental process includes to define appropriate performance measures. In our context, the objective evaluation of the robustness and efficiency of our approach to motifs extraction is performed at two levels: (1) Identification of frequent subsequences within time-series containing both pattern and non-pattern signals, that is the *performance of tentative motifs extraction*; and (2) Classification of these subsequences into motifs, that is the *performance of clustering tentative motifs.* We then define means of evaluating the sensibility and specificity for these two stages of motifs extraction.

Tentative motifs extraction

Defining a measure of sensibility and specificity for this stage aims at evaluating the ability of this algorithm to (a) well identify as tentative motifs subsequences corresponding to instances of motifs (sensibility), and (b) not define as tentative motifs subsequences corresponding actually to non-pattern signals (specificity). Sensibility (Se) and specificity (Sp) are computed from rates of true/false positive/negative (labeled TP, FP, TN, FN), considering one of the two following complementary hypothesis for each point of the original sequence: "the point belong to an instance of a motif" and "the point does not belong to an instance of a motif".

$$Se = \frac{TP}{TP + FN} \text{ and } Sp = \frac{TN}{TN + FP}$$

Clustering into motifs

At this stage, the aim is to evaluate the ability of the algorithm to properly cluster the tentative motifs into motifs, that is (a) to gather all the instances of a same motif in one class (sensibility) and (b) to gather in one class only some instances of a same motif (specificity). Sensibility and specificity of classification algorithms of N vectors are determined using confusion matrix. A confusion matrix C of dimension $m \times n$ represents the results of an algorithm that clusters N vectors, corresponding *a priori* to m motifs, in n classes. Each value c_{ij} represents to the number of vectors belonging to class j that actually correspond to instances of motif i. Considering a confusion matrix, the sum over any row i is the number of instances of motif i, and the sum over any column j is the number of vectors in class j.

Due to the way of extracting the tentative motifs, our context presents some specific features in comparison to defining "simple" performance measures of clustering algorithms.

- The "right" elements to be clustered – that is all and only subsequences corresponding effectively to motifs – may not be available, depending on the performance of tentative motifs extraction. As a consequence, the sum over each row i may be lower than the number of instances of motif i, and the sum over each column j of the confusion matrix lower than the number of elements in class j. Moreover, an instance of a motif may be recognize as more than one tentative motif, so that these sums may also be greater than the usual values expected.
- Because we use unsupervised classification, the number of classes as output of the clustering algorithm might be different from the effective number of motifs, that is $m \neq n$. Analysis of clustering performances reported in the literature does not however usually consider that issue.

Considering these assumptions entails the definition of sensibility and specificity as follows:

- **Sensibility:** *"All the instances of a motif must be gathered in one class, as unsegmented subsequences – that is, an instance of a motif is associated to only one subsequence of the class."*
- **Specificity:** *"All the elements of a class must be representative of a same motif, as unsegmented subsequences – that is, only one element of a class is associated to each instance of the corresponding motif."*

Missing some motifs instances, recognizing some instances as several subsequences, or failing in properly clustering the tentative motifs decrease these performance measures. The proposed definitions of sensibility and specificity are implemented using the concept of *entropy*. A null value of entropy represents the perfect order, which should correspond to a maximum value, 1, of sensibility and specificity: tentative motifs extraction and clustering are perfectly done. Sensibility is related to the well identification and clustering of motifs instances, so that it is defined as a measure of entropy over each row i of the confusion matrix. Specificity is related to the homogeneous composition of each class, so that it is defined as a measure of entropy over each column j of the confusion matrix. These indexes could then be roughly defined as:

$$Se_i = 1 + \frac{1}{log(n)} \cdot \sum_{j=1}^{n} \frac{c_{ij}}{m_i} \cdot log\left(\frac{c_{ij}}{m_i}\right) \text{ and } Sp_j = 1 + \frac{1}{log(m)} \cdot \sum_{i=1}^{m} \frac{c_{ij}}{n_j} \cdot log\left(\frac{c_{ij}}{n_j}\right) \quad (7)$$

where m is the number of motifs expected
n number of classes extracted
m_i number of instances of motif i
n_j number of elements in class j.

The values $\frac{1}{log(n)}$ and $\frac{1}{log(m)}$ correspond to the maximum entropy of a system of n and m states respectively, so that Se_i and Sp_j values are into [0,1].
Using equations (7), the following properties must be verified:

$$\sum_{j=1}^{n} \frac{c_{ij}}{m_i} = 1 \text{ and } \sum_{i=1}^{m} \frac{c_{ij}}{n_j} = 1 \tag{8}$$

However, in our context, we can miss some instances of motifs, or recognize them as divided into several subsequences. Consequently, the values of $\sum_{j=1}^{n} c_{ij}/m_i$ and $\sum_{j=1}^{n} c_{ij}/n_j$ might be either lower – when missing some instances – or greater – when tentative motifs split up – than 1, and we must adapt the formulas (7).

In order to deal with missing instances, we introduce a notion of *"recognition rate"* for each motif i, ρe_i, and each extracted class j, ρp_j defined as:

$$\rho e_i = \frac{\sum_{j=1}^{n} c_{ij}}{m_i} \text{ and } \rho p_j = \frac{\sum_{i=1}^{m} c_{ij}}{n_j} \tag{9}$$

These parameters are used as weights in computing sensibility and specificity, so that the values of sensitivity and specificity decrease in case some instances of a motif are not identified, or some elements of a class are not representative of a motif's instance. Equations (7) then become:

$$Se_i = \rho e_i \cdot \left(1 + \frac{1}{log(n)} \sum_{j=1}^{n} \frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}} \cdot log\left(\frac{c_{ij}}{\sum_{j=1}^{n} c_{ij}}\right)\right), \tag{10}$$

$$Sp_j = \rho p_j \cdot \left(1 + \frac{1}{log(m)} \sum_{i=1}^{m} \frac{c_{ij}}{\sum_{i=1}^{m} c_{ij}} \cdot log\left(\frac{c_{ij}}{\sum_{i=1}^{m} c_{ij}}\right)\right). \tag{11}$$

Furthermore, to get an appropriate measure to deal with motifs' instances possibly discovered as several tentative motifs, we introduce the notion of *"split rate"* for each instance k of motif i, $1/\eta_{ik}$, where η_{ik} is the number of tentative motifs associated to instance k of motif i. When filling in the new confusion matrix $C^{'}$, we then do not consider adding 1 to c_{ij} when instance k of motif i is recognized in class j, but adding $1/\eta_{ik}$ to $c^{'}_{ij}$. Consequently, $\sum_{j=1}^{n} c^{'}_{ij}$ represents the number of instances of motif i that are well identified, even if not well classified, and we have $\sum_{j=1}^{n} c^{'}_{ij} \leq m_i$; and $\sum_{i=1}^{m} c^{'}_{ij}$ represents the number of distinct motifs instances represented in class j. This last value is not necessarily an integer because an element of a class can represent only one part of a motif's instance, the other part(s) being classified in other classes.

Sensibility and specificity are then defined using equations similar to (10) and (11), replacing c_{ij} by $c^{'}_{ij}$. We also re-defined the *"recognition rates"* from (9) to take into account the new definition of the confusion matrix, $C^{'}$, as:

$$\rho e_i = \frac{\sum_{j=1}^{n} c^{'}_{ij}}{\sum_{j=1}^{n} c^{'}_{ij} + m^{'}_i} \text{ and } \rho p_j = \frac{\sum_{i=1}^{m} c^{'}_{ij}}{\sum_{i=1}^{m} c^{'}_{ij} + n^{'}_j},$$

where $m^{'}_i$ is the number of instances of motif i not identified, in any class
$n^{'}_j$ number of elements in class j representative of any motif.

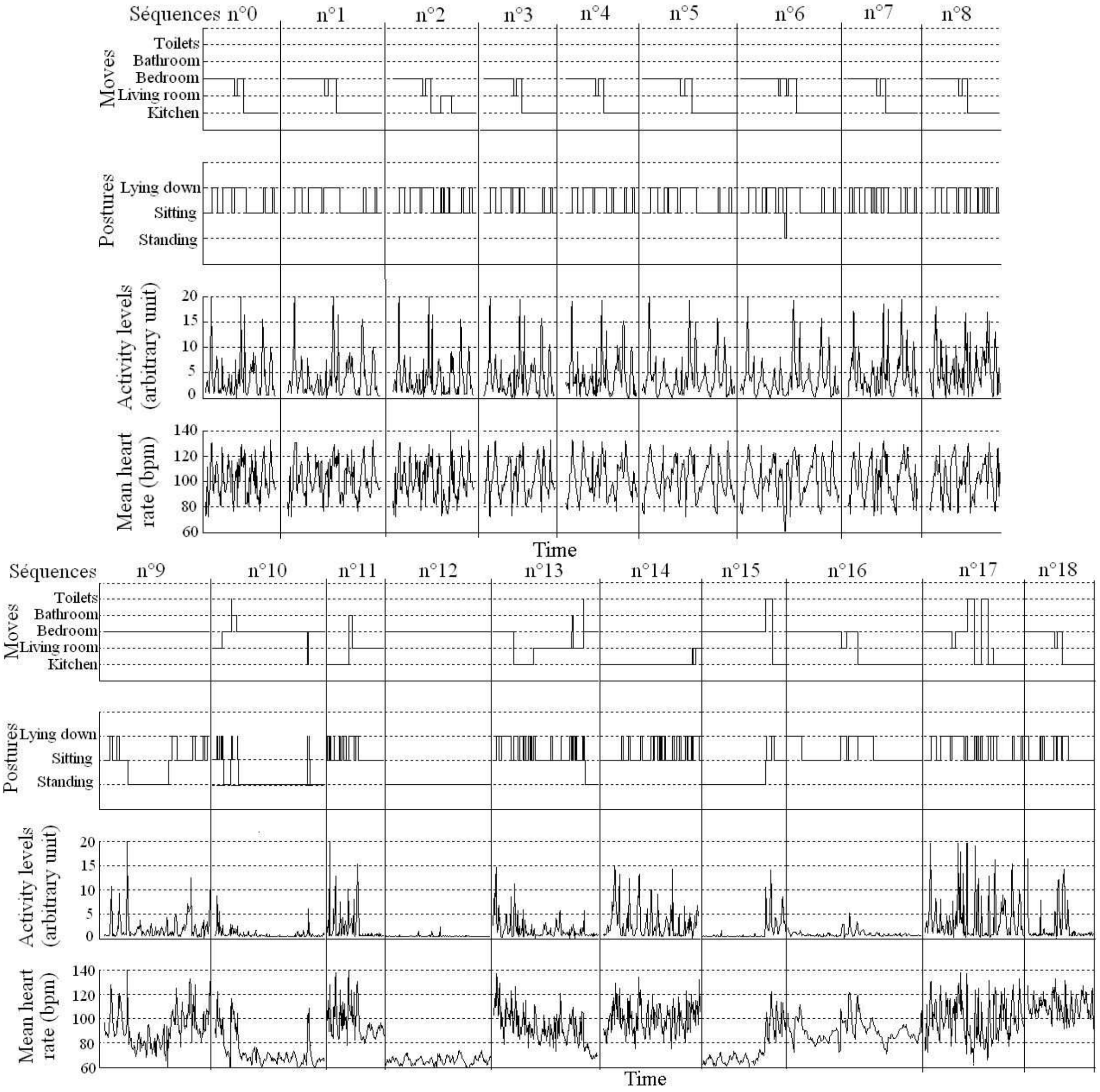


Figure 17: **Experimental sequences.**
From top to bottom, sequences 1 to 8 should be considered close to sequence 0 (class 0), and sequences 9 to 18 far from this reference sequence (class 1).

We also introduce an index of segmentation of motifs recognition, λ, according to equation 12. A maximum value of 1 means that the whole motifs instances are well recognized.

$$\lambda = \frac{1}{n \cdot m} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{c'_{ij}}{c_{ij}} \tag{12}$$

We can at last define mean values of sensibility and specificity, as:

$$Se = \frac{1}{m} \sum_{i=1}^{m} Se_i \text{ and } Sp = \frac{1}{n} \sum_{j=1}^{n} Sp_j.$$

6.3 Quality of the method

Given the high complexity of the problem and the need for several successive steps of analysis, we integrate several levels of validating the proposed approach, that is in consid-

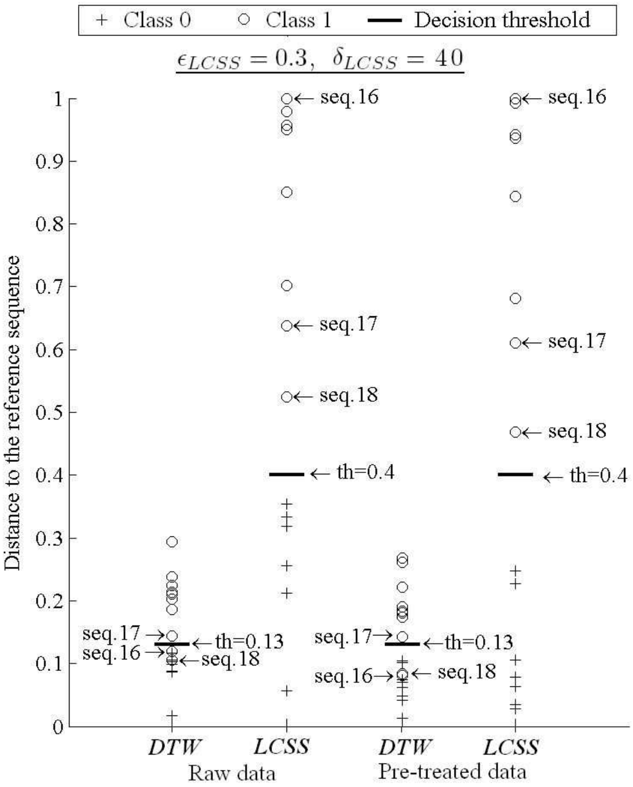


Figure 18: **Distances between sequence 0 and the other experimental sequences,** from either raw or pre-processed data, and using *DTW* and *LCSS* distances. Classes 0 and 1 correspond to the expected classification.

eration to the following steps: (1) Defining a similarity measure, (2) Representation, and (3) Mining operations. Validation is performed from both experts and mathematical and statistical analysis, depending of whether or not appropriate data are available for objective comparisons.

6.3.1 Similarity measure

The approach defined for computing the distance between multivariate and heterogeneous time-series is especially experimented and validated under the strong presence of noise. We compare the performance of our approach, which involves *LCSS*, to the use of *DTW* distances. The distance between time-series is expected to generate low values between sequences corresponding to the realization of a same activity in same conditions, and higher values otherwise. Two experimental sets consist of (see figure 17):

(1) Sequences 0 to 8 representative of a given activity — getting ready in the morning, generated from a reference sequence (sequence 0) by adding noise of three types: stretching in time, variability in values, interruptions (consecutive outliers).

(2) Sequences 9 to 18 representative of other activities like sleeping, having a meal, having a quiet activity, including three sequences (sequences 16 to 18) corresponding to the reference activity (same moves) but carried out in bad conditions, that is (16) slowness, (17) long worrying interruptions, and (18) high values of the mean heart rate.

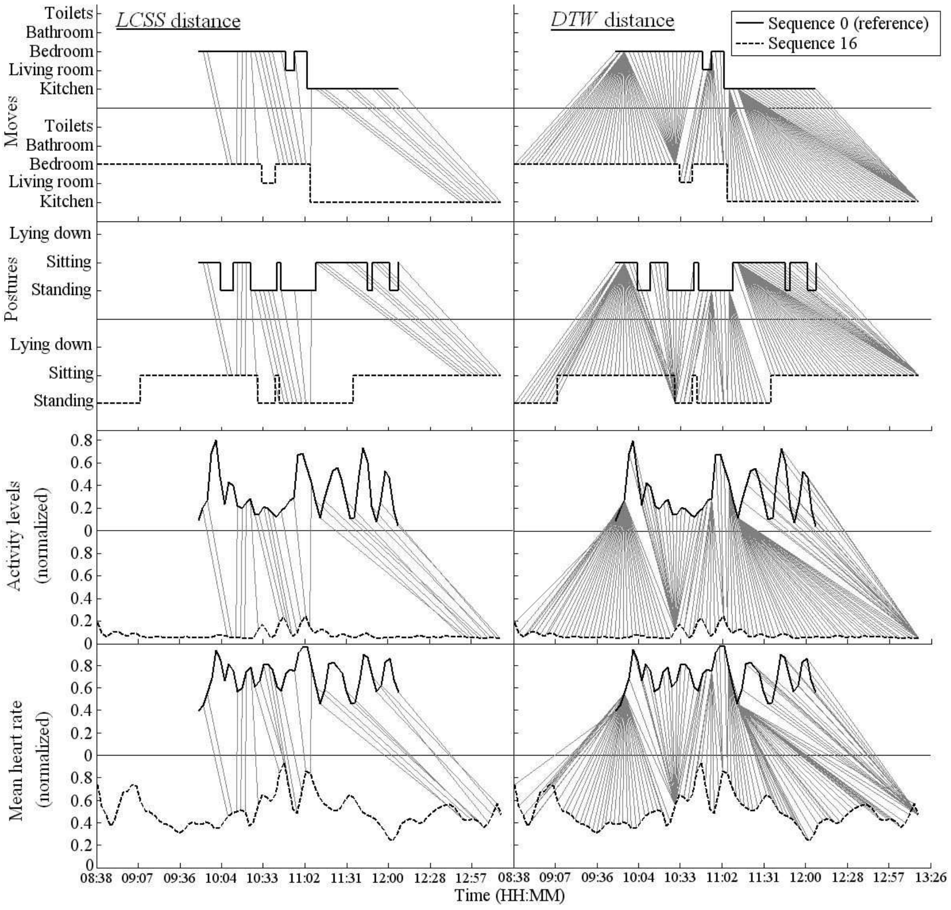


Figure 19: **Pairs of points considered as similar when computing *LCSS* and *DTW* distances.**

> These abnormal behaviors may be detected if sequences 16 to 18 are not considered as representative of sequence 0.

The experimental process aims at classifying these sequences using a threshold on the distance to sequence 0. An appropriate distance may be able to properly discriminate the sequences: 1 to 8 associated to class 0, and 9 to 18 to class 1. We use both DTW and $LCSS$ distances for comparison (see [15] for a clear review of DTW principle), and in each case the distances are computed from both raw and pre-processed data — that is sampling rate reduction to speed-up the computation, and filtering to remove some noise. Preliminary experimentations were required to define relevant values for the $LCSS$ parameters (ϵ,δ) in the context of our application.

The classification results are presented on figure 18. As a general comment, we notice that DTW distances are really lower than $LCSS$ ones, due (1) to different orders of computation — 1 for $LCSS$ and 2 for DTW, and (2) to possible multiple associations of any point using DTW, so that the distance may remain quite low.

The superiority of $LCSS$ over DTW is pointed out by the results matching the expected classification only in the case of using $LCSS$. Using DTW distance fails in properly classifying critical sequences 16 and 18. The behavior of both $LCSS$ — δ set with no restriction in time for associating points, as it is using DTW — and DTW when comparing sequences 0 and 16 is illustrated on figure 19. DTW allows for multiple associations, and all points

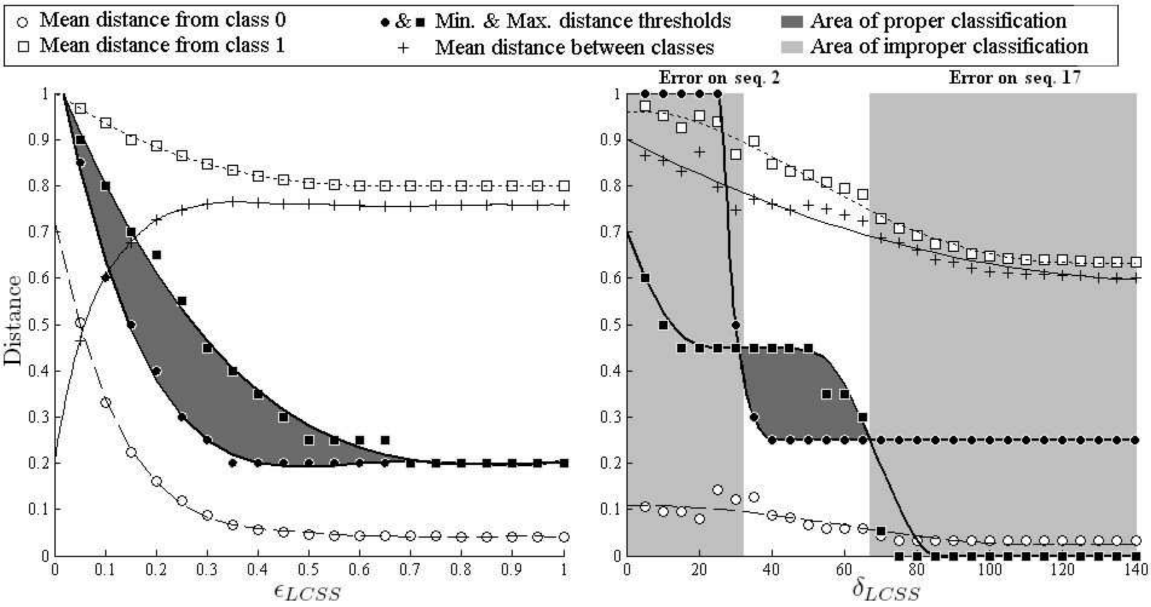


Figure 20: **Distances observed according to selected values of ϵ_{LCSS}.**
These results are obtained when classifying sequences 1 to 18 in class 0 or 1 depending on whether or not close to reference sequence 0 according to a given distance threshold. The graph presents the mean distances observed for class 0 and 1 between each sequence of the class and the reference sequence 0. The minimum and maximum decision thresholds required for a proper classification of sequences in class 0 or 1 is also plotted, such as the mean distance observed between classes. In these samples δ_{LCSS} has been selected as not restrictive.

- The left graph shows the variation of these distances when the similarity threshold on values – ϵ_{LCSS} – is set between 0 to 1, and $\delta_{LCSS} = 40$ minutes.
- The right graph shows these variations for increasing values of the similarity threshold in time – δ_{LCSS} (in minutes) – and $\epsilon_{LCSS} = 0.3$.

must be matched, based on a minimum distance criterion. Then, because the sequences of moves and postures are very close, the poor number of points corresponding to low activity levels and mean heart rate in sequence 0 are associated to the large number of such points in sequence 16, and reciprocally for high values of activity levels and mean heart rate. That results in a low number of pairs corresponding to large distances, so that the distance between the two sequences remains low. The strength of *LCSS* is to base the similarity of points on a threshold criterion, allowing outliers, and excluding overlapping pairs. A higher *LCSS* distance is even obtained for sequence 16 by restricting the value of δ.

We also notice that the two classes are better separated when computing the distances from the preprocessed data. Filtering the sequences indeed results in removing at least part of the variability in the values.
Then, the key parameters in defining the similarity measure are as follows:

- **Maximum temporal difference** (δ_{LCSS}) between two points so that they can be considered as similar.
- **Maximum difference in the values** (ϵ_{LCSS}) of two points so that they can be considered as similar.

Defining δ_{LCSS} and ϵ_{LCSS} restrains the selection of a relevant decision threshold to properly classify sequences, as illustrated on figure 20. For instance, increasing values of ϵ_{LCSS} result in decreasing distance measures, so that an appropriate decision threshold can be defined lower and lower. There is however a stabilization of values as ϵ_{LCSS} ranges over 0.5. Decreasing the value of δ_{LCSS} also results in increasing values of distances.

6.3.2 Representation

The stage of representation is validated at two levels:

1. Checking with experts that the representation step preserves within the sequences the trends of variation they consider as important in identifying the behavioral profile of a person.
2. Analyzing the influence of each step of representation in preserving these fundamental trends according to the purpose of study.

The step of representing temporal sequences includes preprocessing, discretization and aggregation. Validation is performed using intuitive knowledge from experts. The representation steps are anyway highly restrained by the context and purpose of the decision issue, so that there are not many choices in selecting appropriate values for the parameters involved at this stage. Considering several experimentations with the few possible values of parameters, the representation of raw time-series (generated by simulation) is intuitively evaluated according to the purpose of preserving the global trends while removing insignificant variations in terms of studying the activities of daily living of a person. We then define the key parameters as follows:

- **Filter type and length:** we use a mean weighted filter, so that highlights the global trends while preserving the peaks in the values.
- **Rate of temporal reduction:** some analysis show that a rough temporal reduction of initial time-series may remove critical points, especially peaks in the values. Anyway, at the end of the representation step, the aggregation of time-series produces approximately the same number of symbols representing the original sequence, whatever a temporal reduction. We then decide to rely on meaningfully aggregating the successive vectors to reduce the sequences length.
- **Number of discretization intervals:** the meaningful number of discretization intervals for values of quantitative parameters may be approximately defined by experts. Considering the activity level and mean heart rate, the intuitive qualification of possible variations as “resting”, “low”, “moderate”, and “high” guide us to define four intervals of discretization. This rough idea could be refined considering the system’s performances. We use k-means algorithm to determine appropriate bounds of discrete intervals for each monitored person (see §5.1.1). The obtained results are roughly in agreement with related academic knowledge, as illustrated in table 4.
- **Minimum distance threshold:** we use of minimum distance of zero as a threshold for aggregating successive vectors. That means aggregation in one symbol is allowed along subsequences where there is no significant variations of the parameters, so that we can intuitively assume that the person is performing a same “action” all along the corresponding duration. Increasing the minimum distance does not seem appropriate because, especially considering qualitative parameters, a change in a value corresponds

	Academic knowledge		Experimentation	
	Activity level	**Mean heart rate**	**Activity level**	**Mean heart rate**
1	*Rest*	≈ 65	< 1.8	< 75
2	*Very light*	< 75	1.8 to 3.8	75 to 92
3	*Light*	75 to 100	3.8 to 7	92 to 104
4	*Moderate*	100 to 125	> 7	104 to 120

Table 4: **Discretization intervals for the activity level (arbitrary unit) and the mean heart rate (beat per minute) in comparison with empirical boundaries [25].**

to a change in the room occupied or the posture, which represents intuitively a change in the "elementary action" performed.

As a consequence, the number of discretization intervals is the only parameter whose value could be refine by studying its influence on the system's performances, but the possible values are highly constrained anyway. The relevance of considering 4 discrete intervals in our context has been then confirmed by experimentations. For further experimentations, we then decide to use the most intuitively appropriate values of parameters, that is: mean weighted filter, no temporal reduction, 4 intervals of discretization, and a null minimum distance threshold for aggregation.

6.3.3 Mining operations

Mining operations include feature mining – that is, projections, collision matrix examination, and tentative motifs extraction – and clustering. The parameters involved at that stage are differently constrained in their values and have specific influence on the system's performances.

Due to the experimental context, we have identify three highly constrained parameters, so that we can select *a priori* the most appropriate value. Some experimentations can however be performed to possible refine this intuitive choice.

- **Number of symbols** considered to define the length of basic subsequences defined for projections. This length should correspond to the minimum number of symbols defining a motif. Thus, this parameter highly depends on the level of representation of the original sequence in terms of successive aggregated vectors – the symbols. In our context, that corresponds to the minimum number of "actions" successively performed by a person in their activities of daily living, which we have defined intuitively as being 4 symbols. Selecting a lower value, like 3 symbols, could be appropriate, but, reversely, we might miss some motifs' instances.

- **Projection mask** defines the number of symbols, as well as the number of parameters of each symbol, not considered when comparing basic subsequences after projection. That means similar subsequences could differ in these numbers of symbols and components. These parameters are consequently determined according to the rate of noise allowed between two similar subsequences. Given that we consider 4 dimensions for defining symbols, and 4 symbols in basic sequences, the projection mask is defined so that we project 3 symbols and 3 components of these symbols. This choice is validated by the intuitive idea of low collisions numbers between different sequences in the context of observing the mean percentage of collisions between sequences of class 0 or class 1 and the reference sequence 0 (see figure 17), as illustrated on table 5.

Size of projection	**Mean % of collisions**			**Median % of collisions**		
	Class 0	Class 1	Gap	Class 0	Class 1	Gap
3 × 3	**46.1**	**8.5**	**37.6**	**45.5**	**3.5**	**42.0**
2 × 3	56.2	18.3	37.9	61.5	14.5	47.0
3 × 2	58.2	20.3	37.9	64.0	21.0	43.0
2 × 2	67.5	34.5	33.0	78.5	39.5	39.0

Table 5: **Percentage of collisions observed between sequences of class 0 and 1 and the reference sequence 0 according to the size of the projection mask.**
The size of the projection mask correspond to the number of projected symbol × the number of projected parameters per symbol.

- **Number of projections** performed to build the collision matrix. The number of projections should ensure that (1) we do not get hazardously a high number of collisions (specificity), and (2) similar subsequences as defined in our context correspond to a high number of collisions (sensibility). With an increasing number of symbols defining basic subsequences, dimensionality, and rate of possible noise, then should also increase the number of projections performed to get significant results. Given that these influence factors are well defined by the context, we can use a number of projections appropriate in the worst case anyway.

Other key parameters used for mining operations strongly influence the system's performance, but cannot be so easily determined :

- **Minimum collisions threshold.** This parameter defines the number of collisions considered as "significant" in terms of similarity of the corresponding subsequences. This threshold should not be too high so that we miss some tentative motifs. Given that there are next steps to refine the decision about whether a subsequence is representative of a motif, we prefer selecting "too much" candidates at this stage. We should however be careful that the main interest of projections is preserved : not examining the collisions between all possible subsequences all together.

- **Maximum distance threshold.** This parameter set a upper bound on the actual distance between subsequences so that they are considered as similar, and eventually as representative of a same motif. A compromise needs to be found between effectively considering all subsequences representative of a same behavior as similar, even in the presence of noise (sensitivity), and not including *wrong* subsequences as representative of that behavior (specificity).

According to the purpose of motifs extraction, and considering the length of representation of simulated time-series corresponding to typical activities of a person at home, we decide to consider 4 symbols to made up basic subsequences for projections. The mask used for projections is fixed of one unit length for both the number of symbols and the number of components of each symbol.

6.4 Quality of the results

First experimentations are performed using temporal sequences whose structure is known *a priori*. Several instances of a motif randomly selected in simulated sequences are introduced with a moderate amount of noise in a "non-pattern" sequence, that is generated from random moves. Different types of noise are added to motifs' instances when introduced in

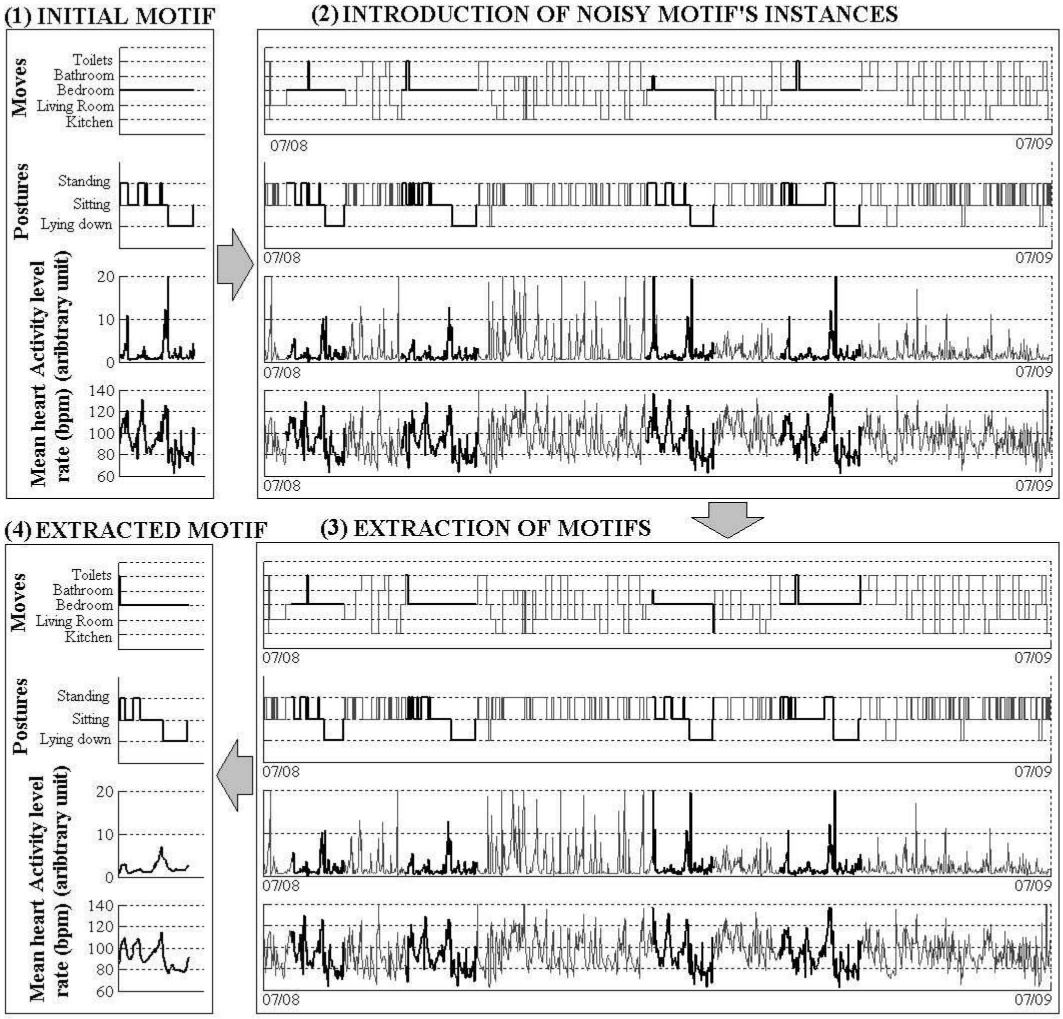


Figure 21: **A sample of pattern recognition from time-series.**
From left to right, and top to bottom, graphs represent (1) a reference motif, (2) introduced several times a day as noisy instances over a "non-pattern" sequence. Then, (3) the next graph represents frequent subsequences as extracted and classified all together from the analysis of previously defined "pattern" and "non-pattern" sequence, and at last (4) the subsequence defined as representative of motif.

non-pattern signals, that is: noisy values, interruptions, and stretching in time (see §6.2.1). A sample of running this experimental process in a noisy context is illustrated through the graphes of figure 21.

After checking the general performance of the system, we especially test its performances in a particularly noisy context (test of sensitivity), such as the results obtained in the presence of abnormaly modified motifs' instances (test of specificity).

6.4.1 General performance results

The first experimental goal is to study the influence of the key parameters on the performances of motifs extraction, evaluated in terms of sensibility and specificity of identifying the tentative motifs and classifying them into motifs. Due to the way of selecting tentative motifs, that is from discrete and aggregated subsequences, the tentative motifs identification cannot be performed really precisely in terms of their indexes of starting and ending (sensitivity and specificity of extraction). Anyway, in our context, the purpose is to identify the occurrence of a motif's instance, without need for exact time and duration. Concerning classification, getting good performances of the system is much more fundamental. Ideally, we need for a "perfect" classification, even if motifs' instances are not precisely identified along time, so that we can recognize all the main activities of a person at home. The system may fail in well classifying the motifs' instances because of bad similarity measures according to the distance threshold. That may be because one of the motif's instance as been too roughly defined, including too many "non-pattern" points for instance and missing too many "pattern" ones, so that the distance increases. Then, improving the precision of tentative motifs identification may be indirectly required to get better performances of clustering.

Critical parameters under study are the collisions and distance threshold, such as thresholds used for computing similarity measure, that is the maximum difference between two similar points in terms of values, ϵ_{LCSS}, and time, δ_{LCSS}. A default system configuration is then defined after many experimentations with varying values of the key parameters, while looking for the best performances in terms of classification results. The graphs of figure 22 present the system's performance according to some possible values of these thresholds. The results highlight the complexity of selecting appropriate values of these parameters, especially because of their relative influence on the system's performances. That is particularly noticed considering the results associated to varying values of **maximum distances in values** (ϵ_{LCSS}) and **time** (δ_{LCSS}) defining the similarity measure.

The **collisions threshold** has not a strong influence on the system's performances, and does not require to be precisely defined. In that specific case, the only interest in defining a collisions threshold lower than the number of projections is to deal with possible imprecision in the representation step. Then, if the minimum collision threshold is too high, we might miss some motifs' instances. Reversely, if too low, we possibly accept "too much" subsequences as potential tentative motifs, and that then requires a lower distance threshold for a proper identification of tentative motifs.

The **distance threshold** is clearly more critical since it is responsible for the end decision about whether or not a frequent subsequence is a motif. Concerning tentative motifs extraction, both true positive and false positive rates increase with the decision threshold : more motifs' instances are well identified, such as more non-pattern subsequences. On the other hand, performances related to the classification task present complex variations of true and false positive rates. Over a certain value, increasing the decision threshold gives better sensibility indexes – more subsequences are considered as similar – but decreases specificity ones – in the same time, some subsequences might be considered as hazardously

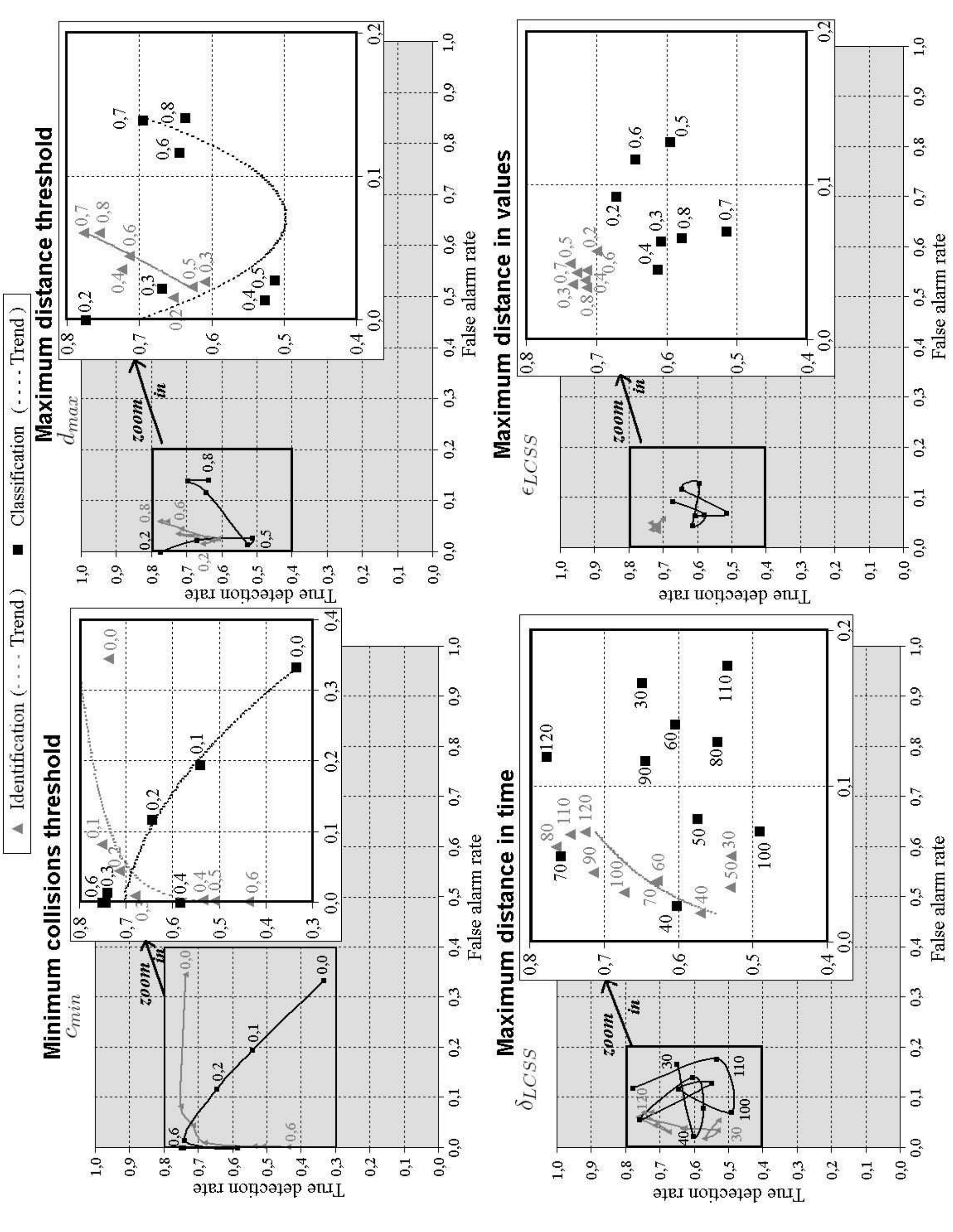


Figure 22: **Mean performance results in terms of true (sensibility) and false positive (1-specificity) rates of the two steps of identifying the tentative motifs and classifying them into motifs, and considering moderate amount of noise between motifs' instances.** We observe individually the influence of each critical parameter in a default configuration of the others.

similar. Considering lower decision thresholds gives however better performances anyway possibly due to higher precision in identifying the motifs' instances, containing few "non pattern" vectors, so that similarity measures are much more relevant for classification. Consequently, we noticed a close relation between the parameters involved in motifs extraction and classification. Then, there is some difficult compromises to be found in defining appropriate values of these parameters, especially in contexts where the system needs to support different possible noise.

Generally, according to the results presented in table 6, the proposed approach for motifs extraction gives good results in terms of sensitivity and specificity of both extraction and classification of motifs. We however notice that the performances indexes are higher for identification than classification, as well as for specificity than sensibility. The system might sometimes fail in well identifying the whole motifs' instances, and some motifs' instances may consequently be missed in the corresponding class. Perfect classification is however possible in some cases. The large variability in the results may partially due to the random selection of motifs for each experimentation. Consequently, corresponding subsequences might not all be representative of a same level of recurrent behavior.

	Identification		**Classification**		**Segmentation**
Indexes	S_e	S_p	S_e	S_p	λ
Mean	**0.71**	**0.92**	**0.66**	**0.79**	**0.89**
Standard deviation	0.18	0.07	0.34	0.26	0.19
Perfect indexes	–	–	35%	60%	70%
Perfect classification			**20%**		

Table 6: **Mean performance results of motifs extraction in a default system configuration and considering moderate amount of noise between motifs' instances.**
The table presents the mean performances in terms of sensibility (S_e) and specificity (S_p) of the two steps of identifying the tentative motifs and classifying them into motifs, as well as the index of segmentation of motifs recognition (λ). All indexes fall into [0,1], the maximum values corresponding to perfect results.

6.4.2 Sensitivity test

The previous study gives an idea of appropriate values for the parameters of motifs extraction, so that we can observe how adding noise influences the system's performances with default values set for extraction parameters. In our experimental context, we should support certain amounts of variability in values, stretching in time, and interruptions.

Analyzing some experimental results, reported on figure 23, show good results for pattern extraction and classification even in the presence of noise. Increasing amounts of the different types of noise effectively degrades sensitiviy indexes. The system as defined using default values of extraction parameters appears to remain especially efficient in the presence of noisy values or stretching in duration, and less resistant to the presence of long interruptions in motif's instances. Anyway, in our experimental context, we do not really know at that stage where is the boundary between "normal" and "abnormal" behaviors.

These results also highlights the complexity of defining appropriate parameters of extraction to deal with all possible types of noise. On one hand, introducing large amount of noise in values or duration mainly requires to increase the values of ϵ_{LCSS} and δ_{LCSS} constraining the similarity between points. High rates of noise in values especially need for higher ϵ_{LCSS}, whereas large possible variations in duration require increasing δ_{LCSS}. On

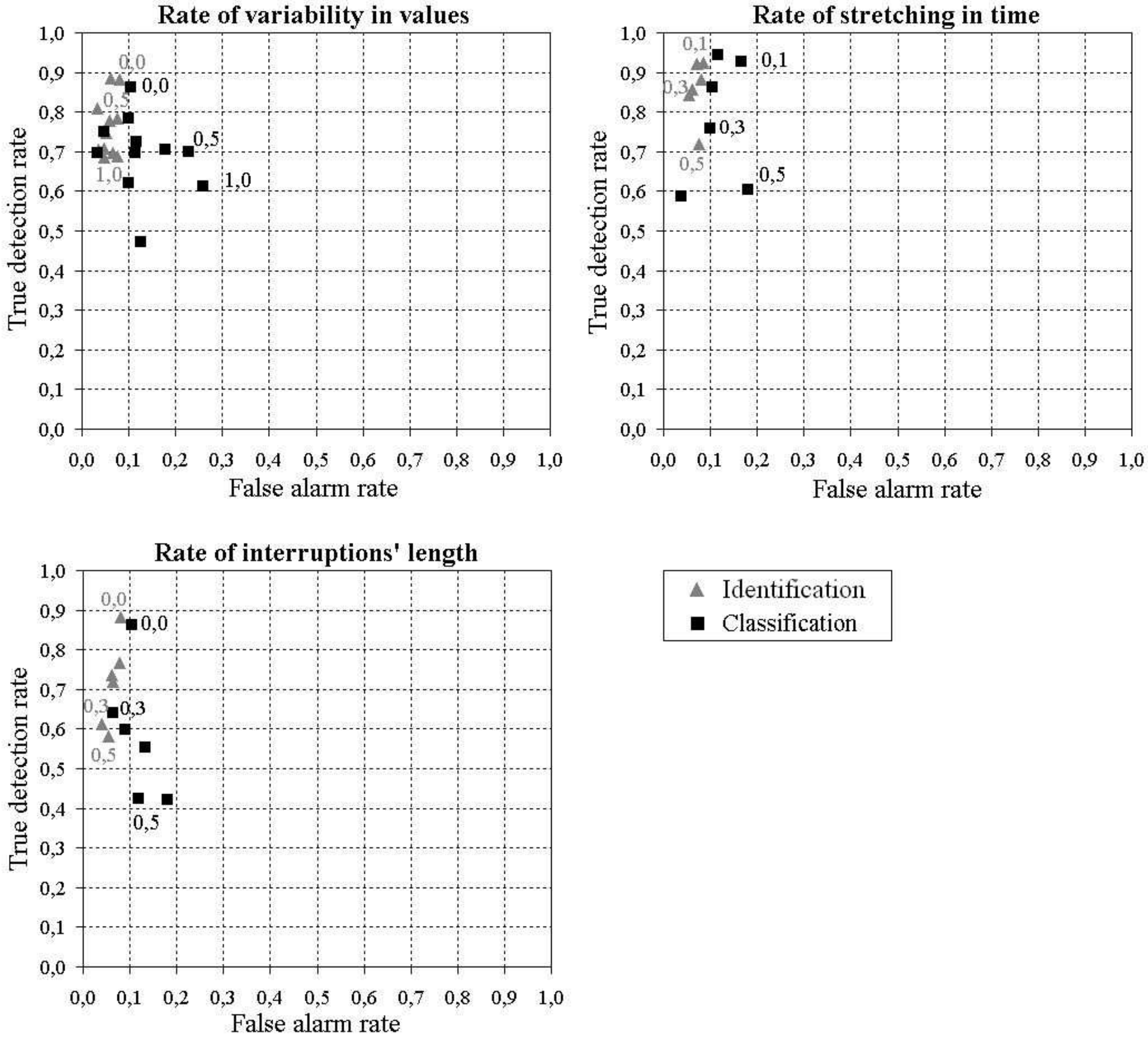


Figure 23: **Mean performance results in terms of true (sensibility) and false positive (1-specificity) rates of the two steps of identifying the tentative motifs and classifying them into motifs, and considering varying rates of noise between motifs' instances.** We observe individually the influence of each type of noise (variability in values, stretching in time, interruptions) on the performance measures in a default configuration of the system.

the other hand, introducing interruptions in motifs' instances mainly influence the selection of appropriate collisions and distance thresholds. That indeed requires to reduce the collisions rate while increasing the distance threshold.

6.4.3 Specificity test

Another test required in the context of critical situation detection is to check the specificity of the system, that is the non-classification of abnormaly modified instances in the class corresponding to "normal" ones. Figure 24 shows a sample of introducing worrying changes in the mean heart rate features, getting to higher values independently of the activity level. Table 7 reports the classification results obtained when introducing successively within a "non-pattern" sequence eight "normal" noisy instances of a given motif followed by four abnormaly modified ones. We especially notice that abnormal instances are rarely associated to the "normal" class. Their recognition rate as "normal" also decreases with increasing worrying change rate in behavior. These results validate that the system might detect critical situations.

Change in behavior	**"Normal"**		**Worrying**		
Worrying change rate	*0%*	*10%*	*20%*	*30%*	*40%*
Recognition rate	**77.5%**	**25%**	**10%**	**10%**	**5%**

Table 7: **Results of classifying tentative motifs from a learning sequence containing both "normal" and abnormaly modified motifs' instances.**
The above table presents the mean percentage of several types of motifs' instances – "normal" and abnormaly modified ones – properly identified and classified in a so called "normal" class.

6.5 Identifying reccurent behaviors from simulated sequences

As a step of validation, we propose to observe the results of extracting motifs from a learning sequence generated over seven days by our simulation process. Simulated sequences might contain subsequences corresponding to recurrent behaviors of the person at home, even if we do not have any a priori knowledge about their explicit features. The results reported on figure 25 highlight that the extracted motifs can be easily interpreted in terms of possible activities of daily living. The frequencies observed are however lower than expected ones, possibly due to some imprecision or incorrectness in the simulation process or to a wrong adjustment of the learning parameters. That however validates the potential good results of the proposed approach for motifs extraction in that experimental context.

7 Conclusion and perspectives

In that paper, we have proposed an approach for mining heterogeneous multivariate time-series to identify meaningful patterns. Generally, some interesting features of the proposed method are the ability to extract motifs from time-series containing both pattern and non-pattern subsequences, in a completely unsupervised way, allowing for noisy values between motifs' instances, and without the need for large amount of learning data sets – the presence of two instances of a frequent pattern is theoretically enough to identify the corresponding motif. Furthermore, the proposed approach presents several advantages compared with several works in finding time-series motifs.

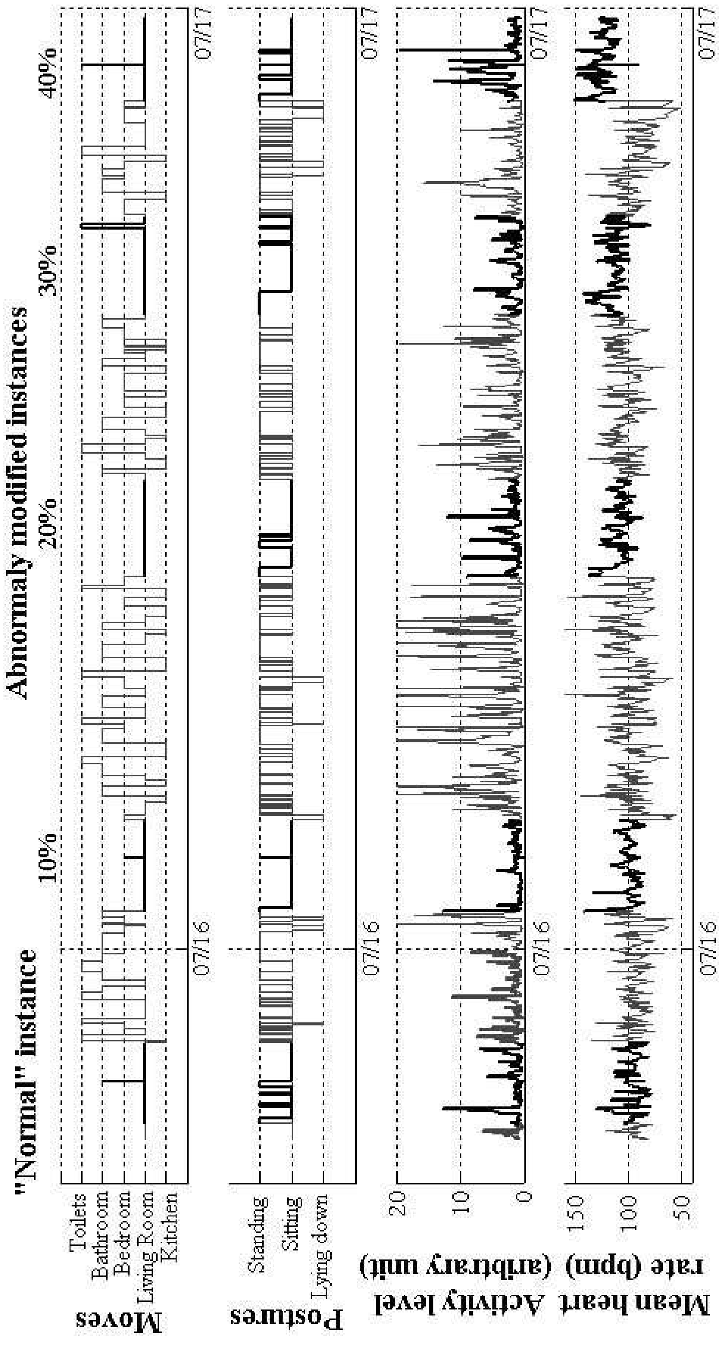
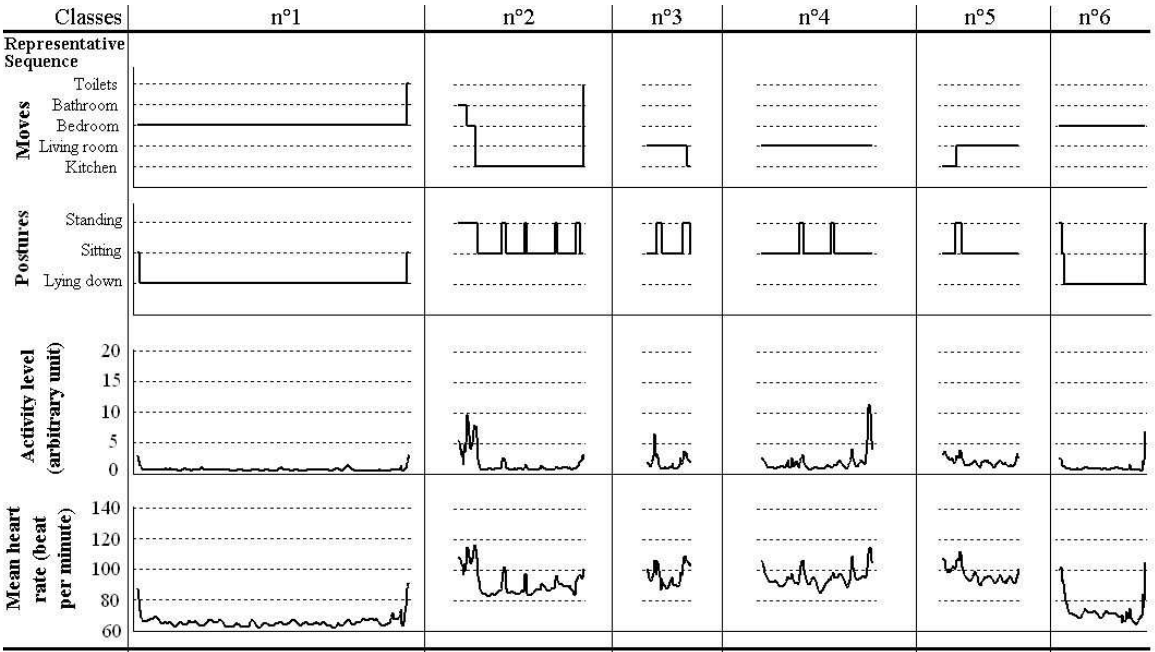


Figure 24: **Sample of introducing worrying changes between motifs' instances.** The motifs' instances are drawn in bold type within a "non-pattern" sequence, including increasing rates of abnormal changes. In that sample, values of the mean heart rate are globally increased, independently of the activity, by 10 to 40%.



Classes	n°1	n°2	n°3	n°4	n°5	n°6
Size	3	5	5	3	10	5
Duration	7.30 (hours)	3.30	1.15	3.00	2.00	2.30
Instant of occurrence	about midnight	about 9.00 am	about 1 pm or 8 pm	about 3 pm	between 4 pm and 7 pm	about 2 pm or 9 pm
Mean observed frequency	every two days	nearly every morning	every two days	every two days	one or two times a day	every two days
Related possible activity	*Night of sleeping*	*Toilet in the morning, light activity and meal in the kitchen*	*Short quiet period in the living room*	*Long quiet period followed by a more heavy work in the living room*	*Quick meal followed by a quiet activity*	*Short resting period*

Figure 25: **Features of recurrent behaviors identified within a learning sequence simulated for a given person over seven successive days.**
A plausible interpretation of each motif is given in terms of possible activities of daily living.

First, we have considered the general case of multivariate and heterogeneous time-series. That particularly implies to define (1) an homogeneous representation of time-series to first roughly identify frequent subsequences, and (2) a similarity measure appropriate to heterogeneous multivariate time-series, in order to compare precisely these frequent subsequences. We have then extended the non-metric distance based on *LCSS* experimented by Vlachos *et al.* [32] to that general context.
Second, we have extended the *projection* algorithm already experimented by Chiu *et al.* [7] for finding time-series motifs to the case of multidimensional symbols defining discrete time-series. This algorithm is indeed particularly suited to strong presence of noise. Our context of using this algorithm also differs from [7] in defining symbols of possibly different lengths, allowing for stretching in time between discovered motifs' instances. We have also proposed a method for extending basic recurrent subsequences to the identification of representative subsequences in terms of observing daily living habits. At the end, a divisive approach to clustering allows to synthesize this set of recurrent subsequences in non-overlapping tentative motifs, then classified into motifs.

Another specificity of our approach lies in a large scale involved from the level of details embedded in raw data to the decision level. That requires to define several levels of extracting relevant information from the original time-series, up to the decision level. However, the decision's purpose considered in our experimental context does not restrict at all the possible use of our approach at different temporal scales or levels of details, or for other applications. First, the same approach can be used a different temporal scale by simply changing the frequency of raw data. Second, appropriate values of parameters can be determined to deal with other context and purpose of decision. The levels of both representation and mining operations can be adapted to a given experimental context. At the representation stage, increasing the filter length, the minimum distance threshold for allowing aggregation, and/ or reducing the number of discrete intervals entails the extraction of "higher level" motifs, and reversely. At the stage of mining operations, changing the values of parameters also results in modifying the sharpness of study. For instance we extract more precise frequent patterns by reducing the maximum distance threshold, or longer motifs are identified by increasing the number of symbols defining *basic subsequences* for projections. Generally, the proposed approach could be appropriate to deal with any application that aims at profiling an usual behavior from the observation of any set of complementary parameters.

First experimentations of the proposed approach for mining heterogeneous multivariate time-series give really promising results. The method allows to well identified a large number of motifs' instances introduced within non-pattern sequences, even in the presence of noise, allowing variability in values, stretching in time, and interruptions. Results also highlight the difficulty of selecting appropriate parameters to a given experimental context. Each possible type of noise influence mainly one type of parameters, which are however closely related one to each other.

Additional experiments may require real data sets to first better define "normality" and "abnormality" in terms of the related temporal features, and then especially to characterize subsequences representative of recurrent behaviors, so that testing the proposed approach may be most relevant. In the context of home health telecare, another step of experimentation is then to validate it is effectively possible to identify similar behaviors in that way from time-series recorded in realistic environments of monitoring a person at home.

References

[1] R. Agrawal, K. Sawhney, K. Shim, "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases," in *Proc. of the . of the 21st International Conference on Very Large Data Bases (VLDB)*, Zurich, Switzerland, September 1995, pp. 490–501.

[2] C.M. Antunes and A.L. Oliveira, "Temporal Data Mining: an overview," in *Proc. of the Workshop on Temporal Data Mining, at the 7th International Conference on Knowledge Discovery and Data Mining (KDD'01)*, San Francisco, CA, August 2001, pp. 1–15.

[3] J. Buhler and M. Tompa, "Finding motifs using random projections," *Journal of Computational Biology*, vol. 9(2), pp. 225–242, 2002.

[4] K. Cameron, K. Hughes, K. Doughty, "Reducing fall incidence in community elders by telecare using predictive systems," in *Proc. of the 19th Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, Chicago, USA, 1997.

[5] S. Charbonnier, J.P. Sich, J.M. Mallion, "Toward a portable blood pressure recorder device equipped with accelerometers," *Medical Engineering and Physics*, vol. 21, pp. 343-352, 1999.

[6] M. Chavent, "A monothetic clustering method," *Pattern Recognition Letters*, vol. 19, pp. 989–996, 1998.

[7] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic Discovery of Time Series Motifs," in *Proc. of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD'03)*, Washington DC., August 2003, pp. 493–498.

[8] G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," *Principles of Data Mining and Knowledge Discovery*, vol. 19, pp. 88–100, 1997.

[9] G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," in *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, New-York, NY, August 27-31, 1998, pp. 16–22.

[10] F. Duchêne, C. Garbay, V. Rialle, "An Hybrid Refinement Methodology for Multivariate Simulation in Home Health Telecare," in *Proc. of the 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (HEALTHCOM)*, Santa Monica, CA, USA, 2003, pp. 101–110.

[11] F. Duchêne, C. Garbay, V. Rialle, "An Hybrid Knowledge-Based Methodology for Multivariate Simulation in Home Health Telecare," in *Proc. of the Joint Workshop Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP) of the 9th Artificial Intelligence in Medicine Europe conference (AIME)*, Cyprus, 19-22 oct. 2003, pp. 87–94.

[12] P. Hong, T.S. Huang, "Learning to extract multi-temporal signal patterns from a temporal signal sequence," in *Proc. of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 3-7, 2000.

[13] P. Hong, S.R. Ray, T.S. Huang, "A new scheme for extracting multi temporal sequence patterns," in *Proc. of the International Joint Conference on Neural Networks*, Washington DC., USA, July 10-16, 1999.

[14] F. Höppner, “Discovery of Temporal Patterns – Learning Rules about the Qualitative Behaviour of Time Series,” in *Proc. of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*, Freiburg, Germany, 2001, pp. 192–203.

[15] E.J. Keogh and M.J. Pazzani, “An enhaced representation of time series which allows fast and accurate classification, clustering and relevance feedback,” *in Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, New-York, NY, August 27-31, 1998, pp. 239-241.

[16] E. Keogh, M. Pazzani, “Scaling up Dynamic Time Warping for Datamining Applications,” in *Proc. of the 21st Int. Conf. on Very Large Databases*, Boston, MA, 2000, pp. 285–289.

[17] J. B. Kruskall & M. Liberman, “The symmetric time warping algorithm: From continuous to discrete,” *Time Warps, String Edits and Macromolecules*, Addison-Wesley, 1983.

[18] D. Kudenko and H. Hirsh, “Feature generation for sequence categorization,” in *Proc. of the 15th Nat'l Conf. Artificial Intelligence (AAAI'98)*, AAAI Press, Menlo Park, California, pp. 733-739.

[19] S.L. Lee, S.J. Chun, D.H. Kim, J.H. Lee, and C.W. Chung, “Similarity Search for Multidimensional Data Sequences,” in *Proc. of the 16th IEEE International Conference on Data Engineering (ICDE'00)*, California, 2000, pp. 599–608.

[20] N. Lesh, M.J. Zaki, M. Ogihara, “Scalable Feature Mining for Sequential Data,” *IEEE Intelligent Systems*, vol. 15(2), pp. 48–56, March/April 2000.

[21] N. Lesh, M.J. Zaki, M. Ogihara, “Mining features for sequence classification,” in *Proc. of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99)*, August 1999, pp. 342–346.

[22] J. Lin, E. Keogh, P. Patel, and S. Lonardi, “Finding motifs in time series,” in *Proc. of the 2nd Workshop on Temporal Data Mining, at the 8th International Conference on Knowledge Discovery and Data Mining (KDD'02)*, Edmonton, Alberta, Canada, July 2002, pp. 53–68.

[23] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, “A symbolic Representation of Time Series, with Implications for Streaming Algorithms,” in *Proc. of the ACM Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03*), San Diego, CA, 2003, pp. 2–11.

[24] N. Meratnia, R.A. de By, “Aggregation and Comparison of Trajectories,” in *Proc. of the 10th ACM International Symposium on Advances in geographic information systems*, McLean, Virginia, USA, November 8-9, 2002, pp. 49–54.

[25] H. Monod et M. Pottier, “Adaptations respiratoires et circulatoires du travail musculaire,” *Prcis de physiologie du travail, Notions d'Ergonomie*, 2nd ed., J. Scherrer et al., Ed. Paris: Masson, pp. 159–204, 1981.

[26] B. Moon, I.F. Vega Lopez, and V. Immanuel, “Efficient Algorithms for Large-Scale Temporal Aggregation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15(3), pp. 744–759, May/June 2003.

[27] A. Nanopoulos, R. Alcock, Y. Manolopoulos, “Feature-based classification of time-series data,” *International Journal of Computer Research, Special Issue: Information processing and technology*, vol. 10(3), pp. 49–61, 2001.

[28] P.H.P. Peeters, “Design criteria for an automatic safety-alarm system for elderly,” *Technol Health Care*, vol. 8, pp. 81–91, 2000.

[29] L.R. Ramos, E.J. Simoes, M.S. Albert, “Dependence in activities of daily living and cognitive impairment strongly predicted mortality in older urban residents in Brazil: a 2-year follow-up,” *J Am Geriatr Soc*, vol. 49(9), pp. 1168–1175, 2001.

[30] V. Rialle, F. Duchêne, N. Noury, L. Bajolle, J. Demongeot, “Health ”Smart” Home: Information Technology for Patients at Home,” *Telemedicine Journal and E-Health*, vol. 8(4), pp. 395–410, Winter 2002.

[31] J.F. Roddick, M. Spiliopoulou, “A survey of Temporal Knowledge Discovery Paradigms and Methods,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 14(4), pp. 750–767, July/August 2002.

[32] M. Vlachos, G. Kollios, ans G. Gunopulos, “Discovering Similar Multidimensional Trajectories,” in *Proc. of the 18th International Conference on Data Engineering (ICDE’02)*, San Jose, CA, February 2002, pp. 673–684.

[33] R. Washburn, K. Smith, A. Jette, C. Janney, “The Physical Activity Scale for the elderly (PASE): development and evaluation,” *J Clin Epidemiol*, vol. 52(4), pp. 153–162, 1993.

[34] G. Williams, K. Doughty, K. Cameron, D.A. Bradley, “A smart fall and activity monitor for telecare applications,” in *Proc. of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Hong-Kong, 1998, Chang and Zhang (eds), pp. 1151–1154.