Generating Reference Models for Structurally Complex Data

Application to the Stabilometry Medical Domain

F. Alonso[1]; J. A. Lara[2]; L. Martinez[1]; A. Pérez[1]; J. P. Valente[1]

[1]CETTICO Research Group, Departamento de Lenguajes y Sistemas Informáticos e Ingenieria del Software, Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain;
[2]Facultad de Enseñanzas Técnicas, Universidad a Distancia de Madrid, Madrid, Spain

Summary

Objectives: We present a framework specially designed to deal with structurally complex data, where all individuals have the same structure, as is the case in many medical domains. A structurally complex individual may be composed of any type of single-valued or multivalued attributes, including time series, for example. These attributes are structured according to domain-dependent hierarchies. Our aim is to generate reference models of population groups. These models represent the population archetype and are very useful for supporting such important tasks as diagnosis, detecting fraud, analyzing patient evolution, identifying control groups, etc.

Methods: We have developed a conceptual model to represent structurally complex data hierarchically. Additionally, we have devised a method that uses the similarity tree concept to measure how similar two structurally complex individuals are, plus an outlier detection and filtering method. These methods provide the groundwork for the method that we have designed for generating reference models of a set of structurally complex individuals. A key idea of this method is to use event-based analysis for modeling time series.

Results: The proposed framework has been applied to the medical field of stabilometry. To validate the outlier detection method we used 142 individuals, and there was a match between the outlier ratings by the experts and by the system for 139 individuals (97.8%). To validate the reference model generation method, we applied k-fold cross validation (k = 5) with 60 athletes (basketball players and ice-skaters), and the system correctly classified 55 (91.7%). We then added 30 non-athletes as a control group, and the method output the correct result in a very high percentage of cases (96.6%).

Conclusions: We have achieved very satisfactory results for the tests on data from such a complex domain as stabilometry and for the comparison of the reference model generation method with other methods. This supports the validity of this framework.

Correspondence to:
Fernando Alonso
Facultad de Informática, Universidad Politécnica de Madrid Campus de Montegancedo
28660 Boadilla del Monte
Madrid
Spain
E-mail: falonso@fi.upm.es

1. Introduction

Knowledge discovery in databases (KDD) and data mining (DM) have proven to be successful in the field of medicine [1–7]. But a limitation of many data mining techniques is that they have been designed to deal with individuals represented as a record where each field contains simple data types. So, in the field of medicine, patient sets are typically modeled as a table with simple attributes such as age, gender, blood pressure and so on.

However, the current trend towards the digitalization of medical tests has led to more complex data being associated with patients. A person's medical history now contains simple and also more complex data such as time series (i.e. electroencephalograms). These time series can even be multidimensional, where each timestamp contains several data. This results in individuals with structurally complex data.

Without KDD and DM techniques that are able to deal with such structurally complex data, patient medical histories cannot be fully analyzed to discover relevant knowledge [8, 9]. In this paper we define a conceptual model for managing structurally complex data and propose methods for detecting outliers and for creating reference models of structurally complex data sets. Building reference models from a set of individuals is a very important issue, as they are useful in a wide range of tasks such as diagnostics, decision support, fraud detection, etc.

The proposed framework has been applied to the medical domain of stabilometry (also called posturography, statokinesemetry or posturometry) that is concerned with patients with balance disorders or vertigo. Stabilometry involves measuring stability of stance or postural equilibrium in human beings. It transforms the mechanical oscillations of a human being's physiologic gravity center into electric signals, and then amplifies, records and analyzes these signals [10]. In this research, we have worked with the Health and Sports Area of

the Spanish Council for Sports. They use stabilometry to assess elite athletes for functional disorders.

This paper is structured as follows. Section 2 outlines the main goals of our research. Section 3 describes the stabilometry domain and how individuals in this domain can be modeled. Section 4 describes the data mining methods proposed for outlier detection and for reference model generation. Section 5 then shows the results of applying these methods in the stabilometry domain. Finally, the paper provides concluding remarks.

2. Objectives

This article is part of research aiming to define a general-purpose framework for reference model discovery designed to operate in domains where individuals are structurally complex (data are hard to represent as tuples in relational tables) and composed of different types of attributes that can be organized hierarchically.

The proposed method is able to output reference models for particular population groups: the model will represent the group and act as an archetype for this group. This is the basis for generating models for normal populations (control groups) or populations with specific features (e.g. a particular disease, joint disorder, etc.). These models can be used to classify new individuals in a group, as a decision-making aid for physicians, etc.

To do this, our framework includes:

- A conceptual model for modeling structurally complex individuals. Each individual will be represented by a series of hierarchically related entities that contain different data types, including continuous or discrete single-valued data, multidimensional time series, etc.
- A method for comparing individuals represented according to this conceptual model. Calculating the similarity between two individuals is a fundamental task for the application of data mining techniques in any domain. Solutions like the Euclidean distance between attribute vectors, which are sufficient for other data types, are not applicable in this case because of the complexity of the individuals.
- An outlier detection method for identifying individuals that deviate from the norm. Taking into account that these individuals generally have more negative than positive effects on the representativeness of the resulting model, outlier detection is an essential task that should be performed before models are built.
- The method for generating reference models, which uses the above mechanisms to output the reference model for a particular population group.

In this paper we describe the proposed framework and its application to the stabilometry domain. The framework is summarized in ▶Figure 1, where all arrows represent data flows and section numbers have been added to link the processes to the sections in this paper. The first step is to define a conceptual model to represent individuals based on expert knowledge and domain information. Then the expert can define population groups for which the system will generate reference models. This process relies on the methods for comparing individuals and detecting outliers.

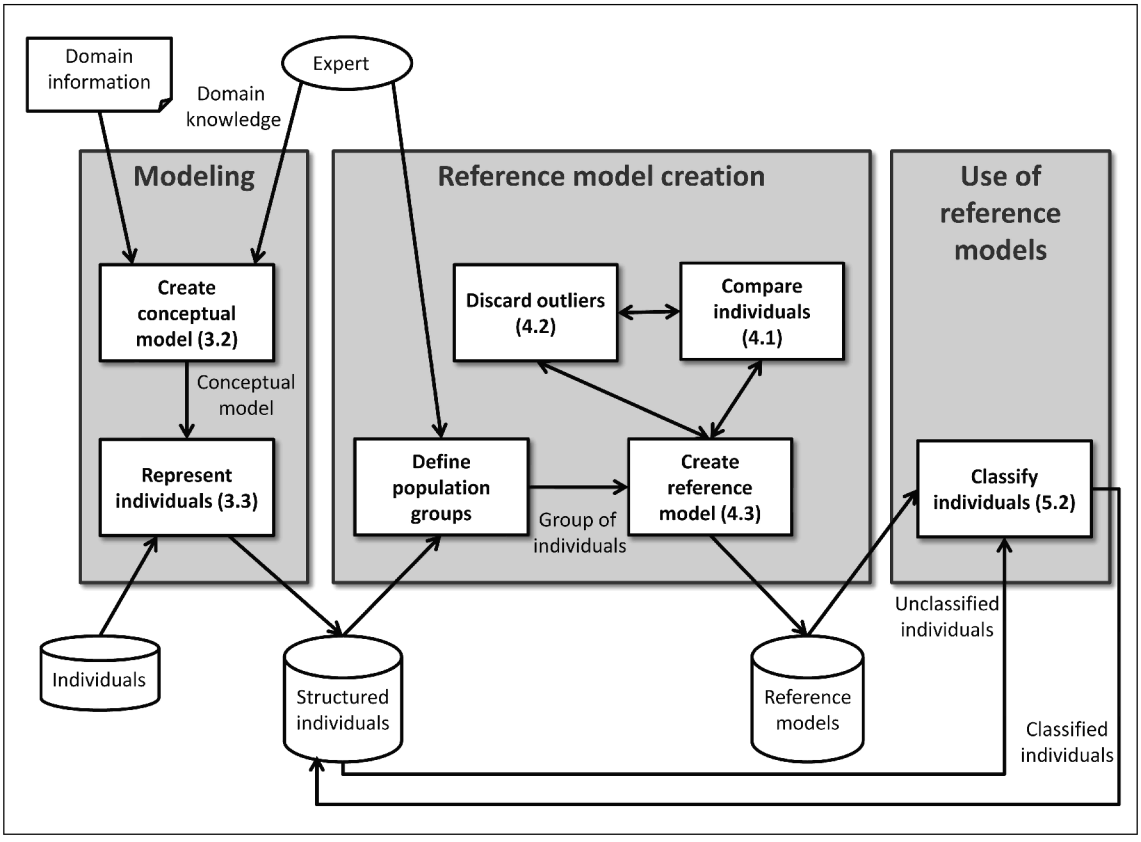

Figure 1
Reference model generation and use for structurally complex data

Finally, the generated reference models can be used to classify new individuals.

3. The Stabilometric Domain and Conceptual Model

This section describes the stabilometric tests, the modeling approach used and the resulting conceptual model for individuals containing stabilometric data.

3.1 Stabilometric Tests

We have worked on data generated by a stabilometric device known as a posturograph. This device consists of a platform on which a person stands. The platform has four pressure sensors in the four corners: left front (LF), left rear (LR), right front (RF) and right rear (RR). These sensors record the pressure exerted by the patient with a 10 ms sampling interval, generating a multidimensional time series.

The posturograph can be used to run a wide range of tests according to predefined protocols. The resulting set of tests is called a stabilometric examination. We have focused on three tests that output most useful information for domain experts. These tests are called Limits of Stability, Unilateral Stance and Rhythmic Weight Shift:

a) *Limits of Stability (LOS).* The goal of this test is to measure patients' ability to voluntarily move their center of gravity towards a specific position in space (called target) with both feet on the platform and to hold this position for a while without losing balance. This test is composed of eight parts, each one corresponding to a different target. ►Figure 2 shows the center of gravity paths of a patient who is trying to move towards the different targets (squares). The position of the center of gravity at each timestamp is calculated using the time series values generated by the pressure sensors.

b) *Unilateral Stance (UNI).* This test aims to measure patients' ability to keep their balance when standing on one leg with both eyes either open or closed. The ideal result for this test would be for patients not to wobble at all but to keep a steady stance throughout the test. What typically happens is that patient balance is constantly shifting, and, in some cases, patients have to put the lifted foot down on the platform (this is called fall). ►Figure 3 shows an example of this test performed by a patient who fell twice.

c) *Rhythmic Weight Shift (RWS).* The aim of this test is to measure patients' ability to rhythmically move their center of gravity horizontally (from left to right and from right to left) and vertically (from front to rear and rear to front) at different speeds. The appearance of the resulting time series is as shown in ►Figure 4.

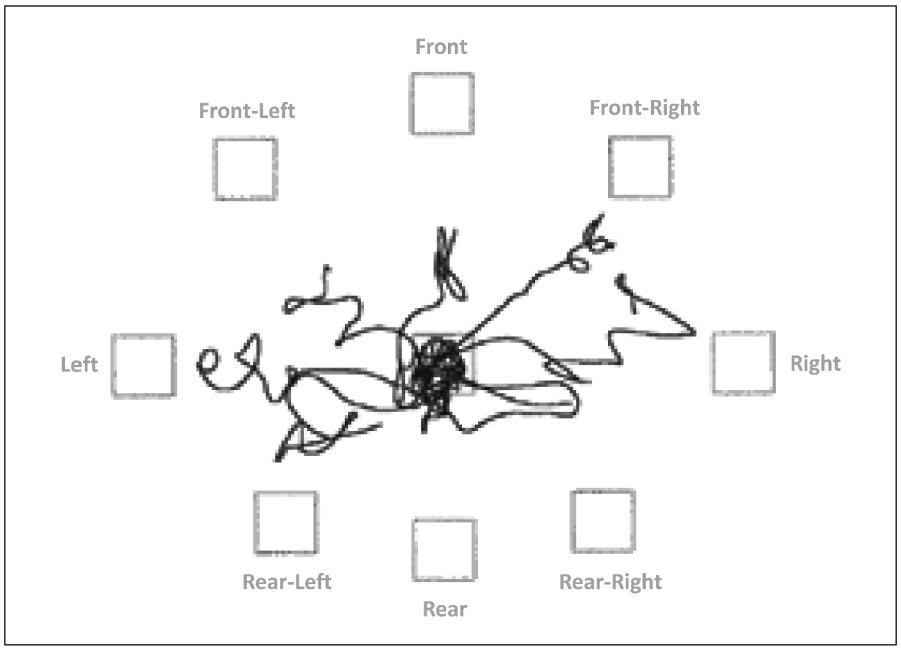

Figure 2 Visualization of LOS test showing patient movements

3.2 Conceptual Model

The conceptual model in our framework represents structurally complex data in which each individual is defined by means of a number of hierarchically related entities that contain different attribute types, like time series or single-valued attributes.

Unlike other conceptual modeling proposals reported in the state of the art [11], we propose a hierarchical model of the individuals designed for the execution of the data mining techniques that are to be used (comparison of individuals, reference model generation). The conceptual model is capable of defining multidimensional time series attributes, as well as the type of each attribute (continuous and quantitative, ordinal and qualitative, etc.) and so on. This will be very useful for applying the data mining techniques discussed later.

The elements of our conceptual model are:

- Entities. They can contain qualitative data (Entity_QL) only, quantitative data (Entity_QU) only or mixed data (Entity_QL_QU). We have also defined a special entity type for time series (Entity_Series_QL and Entity_Series_QU). These entities may contain a number of qualitative or quantitative time series dimensions.
- Attributes. They are the leaf nodes of the conceptual model and store all the data on individuals. The possible attributes are:
  - Single-valued attributes. They are defined by their name, type and description. There are several possible types: continuous and quantitative, discrete and quantitative, nominal and qualitative, and ordinal and qualitative attributes.
  - Time series dimensions. They account for each dimension of the time series, which can be qualitative

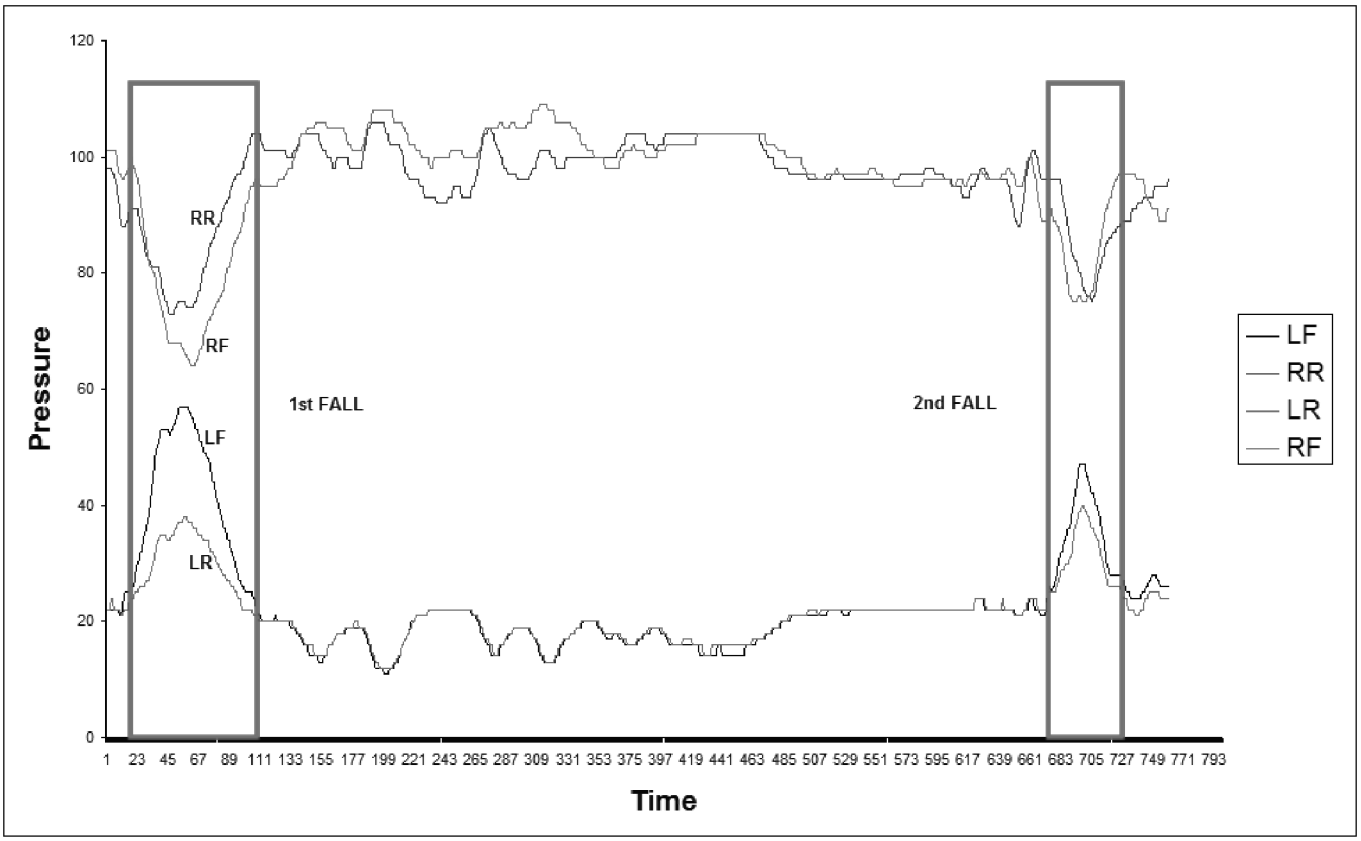

Figure 3 UNI test time series, highlighting two falls

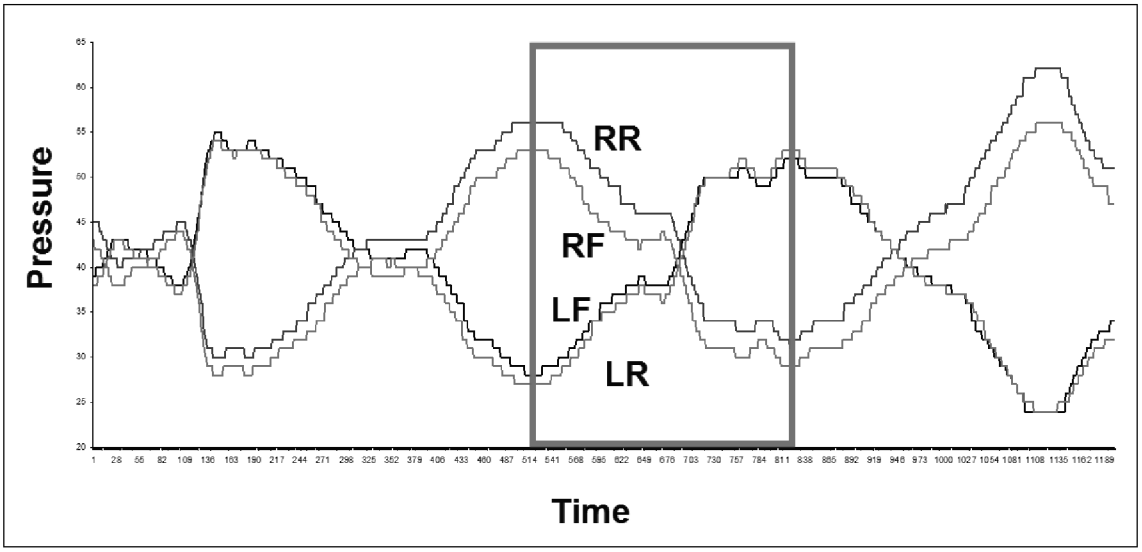

Figure 4
RWS time series with one highlighted pause in the rhythmic movement

(Dimension_QL) or quantitative (Dimension_QU).

- Relationships. Relationships between the above elements (entities, single-valued attributes and time series) may indicate that an element contains another element (Contains_S) or contains several elements (Contains_M). These associations are used to model the hierarchical structure of the data from the root to the leaf nodes.

This conceptual modeling system has been defined as an extension of UML [12–14] (unified modeling language) in conformance with the graphical notation illustrated in ▶Figure 5.

3.3 Stabilometric Conceptual Model

The individuals to which the data mining algorithms are applied in the stabilometric domain are stabilometric examinations completed by patients. ▶Figure 6 shows a fragment of an individual's hierarchical conceptual model. The root of this hierarchy (*stabilometric examination*) contains only one occurrence of each first-level entity (each *stabilometric test*). On the other hand, the UNI test has four trials (one for each leg with eyes first open then shut). Each trial contains three occurrences (which match the three repetitions of each trial according to the predefined protocol). For time series, we record the number of timestamps and model their dimensions using the appropriate icon depending on whether they contain quantitative or qualitative data. ▶Figure 6 shows the modeling of the Left_Leg-Eyes_Closed time series, which contains 1000 timestamps and is composed of four quantitative dimensions (LF, RF, LR, RR).

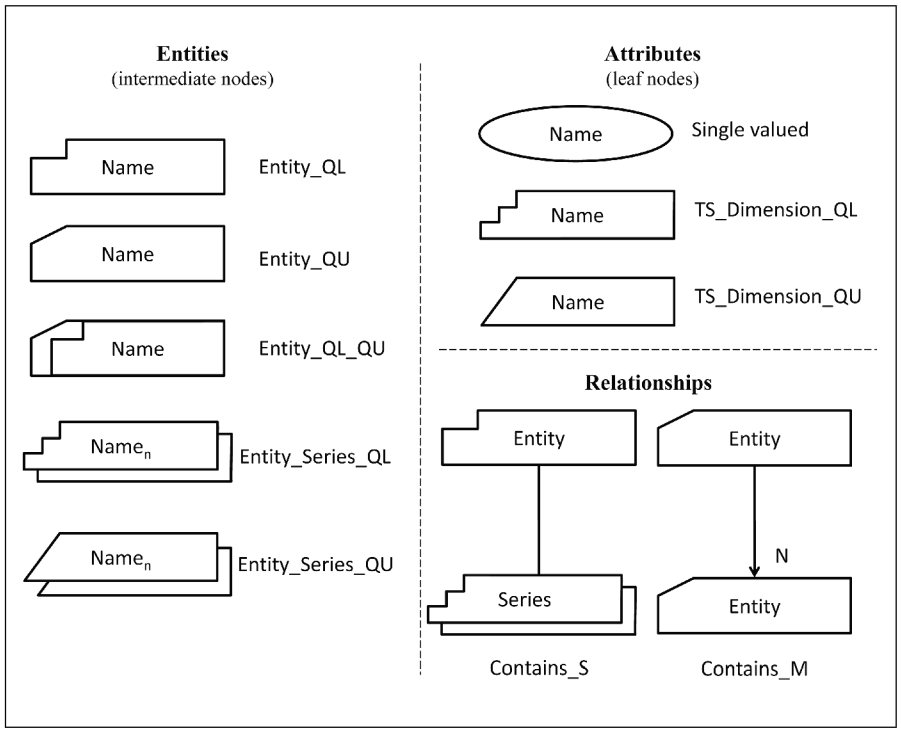

Figure 5 Graphical representation for the proposed conceptual modeling language

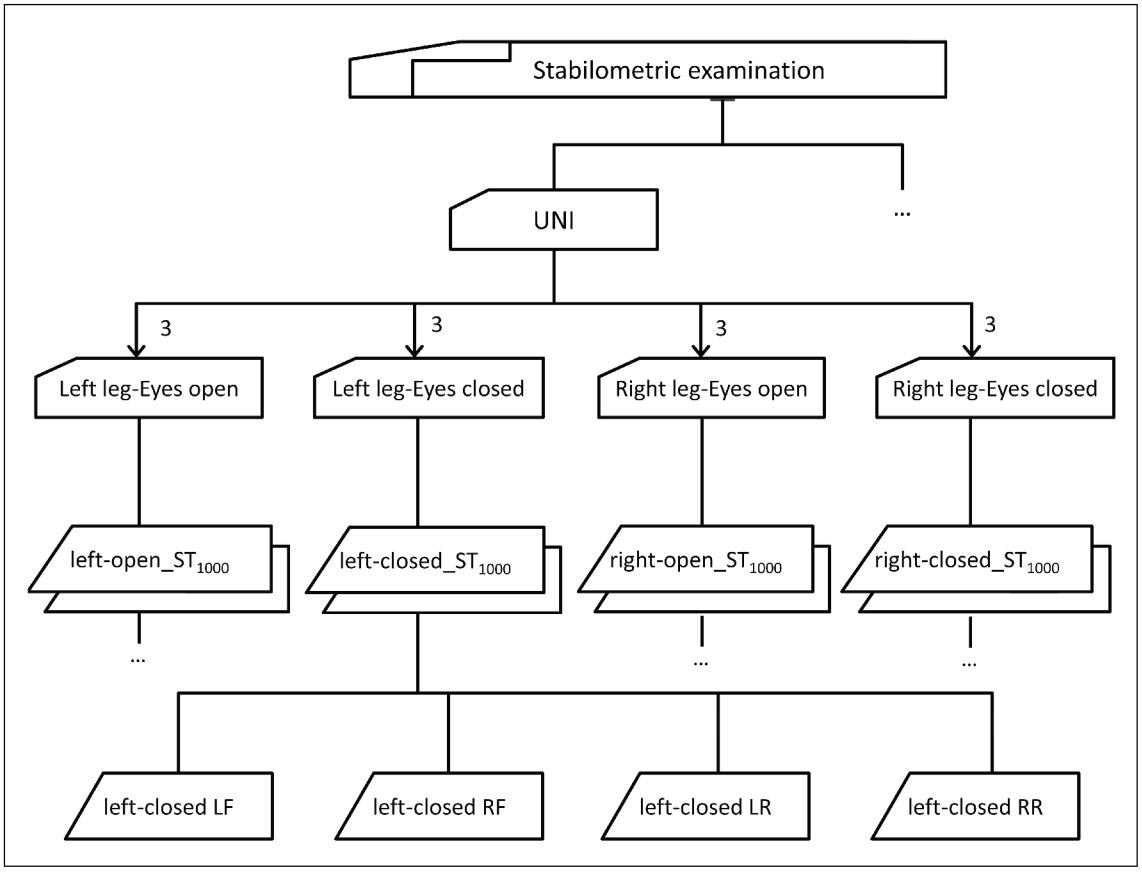

Figure 6
Fragment of the model of an individual in the stabilometric domain

4. Methods

In this section, we describe the methods used to generate reference models. First we describe the method that determines how similar individuals are. Then we explain the method for detecting outliers and finally we describe the reference model generation method.

4.1 Comparing Two Complex Individuals

This method determines a measure of similarity between two hierarchically organized structurally complex individuals. The method receives two individuals as input and outputs a real number in the interval (0, 1) that indicates how similar they are.

Methods for comparing hierarchical structures have been reported in the literature. Most of these methods [15, 16] compare tree structures (linearizing trees and using edit distances) and are capable of managing issues like the number of children in each node, node labeling, etc. In our case, the structure of all the individuals that we are comparing is identical because the data are taken from medical examinations conducted according to a strict protocol, as applies in many medical and other domains. Therefore, all the individuals have the same attributes, have performed the same repetitions of each trial of each test, etc. Additionally, our tree nodes store the attribute values (attributes may be single-valued or time series), and these are the target values of the comparison. Therefore, a simple comparison of whether or not the node label is equal is of no use.

As the proposed conceptual model represents individuals as a tree, we have introduced the similarity tree concept to determine the similarity value between two individuals. A similarity tree is a data structure with the same format as the individuals that are to be compared. This data structure is used to calculate the partial similarity values between the two individuals at each node of the tree. The comparison starts at the leaf nodes and calculates the similarity value of each leaf node. Once the similarity value of each leaf node is known, the similarity of each parent node is computed as the weighted mean of the similarity values of its child nodes. At the end of this process, the root node will contain the similarity value between the two individuals. ▶Figure 7 illustrates an example of a comparison between two individuals (top) with the resulting similarity tree (bottom).

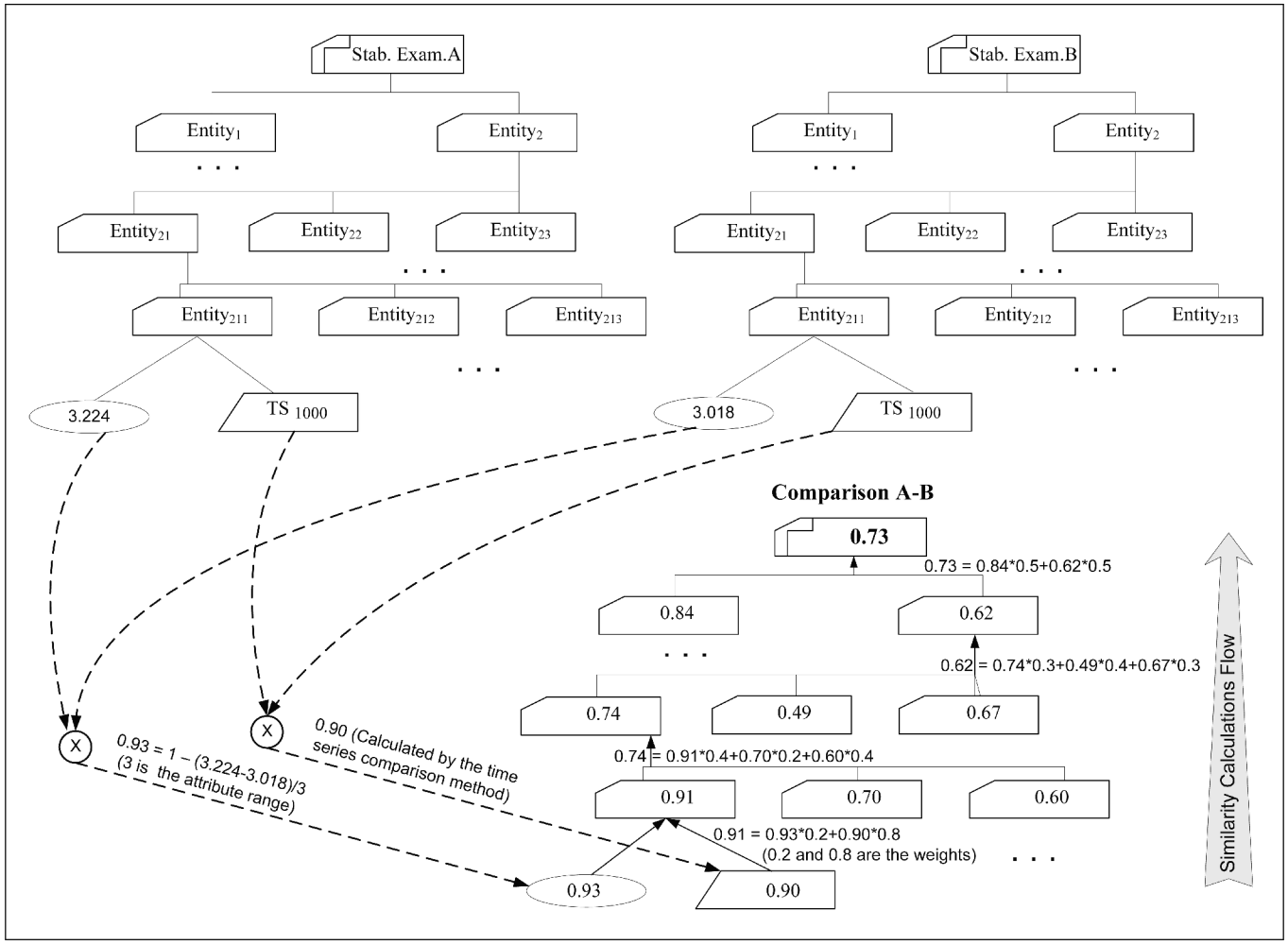

Figure 7 Example illustrating how the method for comparing structurally complex individuals works

According to the conceptual model, the leaf nodes can be single-valued attributes or time series dimensions. For the comparison of single-valued attributes, the values of the numerical attributes are subtracted and normalized depending on the possible attribute value range (as shown in ►Figure 7). For the categorical attributes, our approach relies on the expert to identify the comparison values between single-valued attributes, as comparison is mostly domain dependent. To do this, experts use a graphical tool to straightforwardly establish the similarity values between each pair of attribute values (for more details, see [17]).

Comparing time series is a tougher problem, for which many different approaches have been proposed in the literature, ranging from more classical approaches based on Fourier transforms [18] or wavelets [19], through methods that transform and approximate the time series to another series with which it is being compared [20], methods based on the identification of breakpoints in time series [21], methods that compare time series in terms of how many subsequences they have in common [22], etc. Most of these techniques work with whole time series. In many domains, however, parts of the time series do not contain relevant information for the domain and should therefore be omitted [23]. To deal with this problem, we have developed a method [24] to compare two time series. This method takes into account only the relevant parts of the time series, called events.

The final result of the comparison is a similarity tree (with the same structure as the compared individuals). This tree provides not only a global measure of the similarity between two individuals (value of the root node) but also the similarity value at any tree level. This outputs partial similarity values, which would be equivalent to individual medical tests, trials, etc. in a medical domain. This provides very useful detailed information for medical specialists.

4.2 Outlier Detection Method

Before building a reference model that correctly represents a particular population group, outliers have to be identified and excluded from the population that is to be taken into account to build the model because, if included, they would degrade the model [25, 26].

There are many different techniques for outlier detection. A large subset of outlier detection techniques is based on a previous clustering process [27–30]. The output clusters are then analyzed to determine which elements should be considered outliers. In most of these techniques [31–33], users are required to establish the parameter values that are used in the outlier detection process. Very often, outlier detection is a small part of a much broader process, and users may not have accurate enough knowledge about the parameters to assign the proper values.

In this paper, we propose an outlier detection algorithm that requires the least possible user intervention while at the same time providing a reliable and true measure of when an individual is an outlier. Individuals have to be clustered before the outlier detection algorithm is applied. To do this, a bottom-up hierarchical clustering method is applied. The outlier detection algorithm is as follows:

Let $D = \{I_j, 1 \le j \le n\}$ be the set of individuals (n is the number of individuals) and $C = \{C_i, 1 \le i \le k\}$ be the set of clusters (k is the number of clusters across which n individuals are distributed).

1. Calculate the outlier factor $OF \in (0, 1)$ for each individual I_j that is a member of cluster C_i according to ►Equation 1.

$$\mathrm{OF}(\mathrm{I_j}) = \frac{\mathrm{OF_{\#NEIGB}}(\mathrm{I_j}) + \mathrm{OF_{LOC}}(\mathrm{I_j})}{2} \quad (1)$$

The factor $OF_{\#NEIGB}(I_j)$ (►Equation 2) increases the likelihood of an individual I_j being an outlier as the number of elements in cluster C_i of which it is a member decreases. The factor $OF_{LOC}(I_j)$ (►Equation 3) is calculated depending on whether or not there is a single cluster C_r that contains most domain individuals. If there is, this factor increases the likelihood of individual I_j being an outlier as it moves away from cluster C_r. If there is not, the individual will be more of an outlier the further away it is on average from other clusters.

$$\mathrm{OF_{\#NEIGB}}(\mathrm{I_j}) = 1 - \frac{|\mathrm{C_i}|}{|\mathrm{D}|} \quad (2)$$

$$\mathrm{OF_{LOC}}(\mathrm{I_j}) = \quad (3)$$

$$\begin{cases} 1 - \mathrm{Sim}(\mathrm{I_j}, \mathrm{C_r}) & \text{if } \exists \mathrm{C_r} \\ 1 - \dfrac{\sum_s \mathrm{Sim}(\mathrm{I_j}, \mathrm{C_s})}{\mathrm{k}-1} & \text{otherwise} \end{cases}$$

2. Calculate the mean (mOF) and standard deviation (sOF) of the outlier factors of all the individuals $I_j \in D$.
3. Establish the outlier threshold OT according to ►Equation 4.

$$\mathrm{OT} = \mu_{\mathrm{OF}} + (1 + 2\mathrm{d}^2)\,\sigma_{\mathrm{OF}} \quad (4)$$

In order to be able to regulate the likelihood of an individual being an outlier depending on the domain dispersion, we introduce the dispersion factor $d \in [0, 1]$. In domains where individuals are naturally far apart from each other, we will define a dispersion factor close to 1. This increases the outlier threshold OT making it quite a lot harder for an individual to be considered an outlier.

4. Return as outliers those individuals whose OF is greater than OT:

$$\mathrm{OUTL} = \{\mathrm{I_j} \in \mathrm{D} \,|\, \mathrm{OF}(\mathrm{I_j}) > \mathrm{OT}, 1 \le \mathrm{j} \le \mathrm{n}\} \quad (5)$$

4.3 Reference Model Generation Method

In the context of this research, a reference model is an individual built from a group of individuals with the aim of representing that group. Based on a set of individuals from the same domain that have the same data structure (and can contain both single-valued data and time series), then we aim to generate an individual that acts as an archetype capable of representing the whole set. The method for generating reference models is as follows:

INPUT: A set of individuals represented using the hierarchical conceptual model

OUTPUT: The reference model for this set

METHOD:

1. For each leaf node
   1.1. Apply the outlier detection algorithm to identify and ignore the nodes whose attributes have outlier values
   1.2. Use the remaining nodes to calculate the typical value of the leaf node attribute depending on the attribute type (numeric and single-valued, symbolic and single-valued or time series)
2. Generate the reference model: create an individual that has the same structure as the domain individuals and assign to its leaf nodes the typical values calculated in step 1.2

Step 1.2 is a fundamental part of the process. Finding the typical value of a single-valued attribute is a relatively simple problem that can be satisfactorily solved using appropriate statistical indicators. Depending on the attribute type, some statistics work better than others. Thus, the indicators used for the different types of single-valued attributes to represent their typical values within a population group are shown in ▶Table 1.

The problem is a lot more complex for attributes whose values are time series which may be continuous, discrete or symbolic, where either the whole time series or just a few segments of the time series contain relevant information. For this case, we have designed a new algorithm to determine which would be the typical time series that best represents the population group. This time series will be the value that this attribute takes in the reference model for the population group.

The typical time series, which might also be referred to as reference model for a set of time series, is very useful in many domains. However, there are relatively few proposals for model generation from time series compared with the number of techniques developed in other fields of time series analysis. In one approach, Chan and Mahoney [34] model a set of time series incrementally. They start with one time series and create the succession of smallest boxes enclosing each pair of successive time-stamps of this time series. They then add the other series, expanding the boxes to enclose each one. The succession of boxes is the model of the set of time series. In another approach, Rombo and Terracina [35] mention representative models, but their concept of model is very different to the notion defined here. They really search for subsequences that are repeated within one and the same time series. To generalize the procedure, they add wild cards to these subsequences. The most frequent subsequences in the time series are their models.

But neither of these papers proposes a model of the set of series, defined in the same format as each element of the set which can be used as a typical representative of the set. Nor do they build the models using only the relevant information in the time series, which is concentrated within certain regions of interest for the domain expert (or events). This is a very common situation in many domains such as seismology, medicine, industry, etc.

In this paper, an event is a fragment of the series that satisfies conditions specified by the domain expert. We specify which event attributes will be stored for each event type, implicitly defining a data structure for the event. These attributes cover what the expert considers to be the key features for each type of domain event. For example, falls (one of the events of interest in the stabilometry domain) are characterized by the following attributes:

a) Region in which the lifted leg falls.
b) Intensity of the pressure exerted by the falling patient's foot on the platform and drop in the intensity of pressure of the standing leg sensors.
c) Time from when the patient starts to lose balance until he or she falls.
d) Time from when the patient falls to when he or she recovers.

To do this, we have defined an event specification language (ESL)[36]. ESL is used to define which time series behaviors are indicative of an event. We have also built an ESL compiler that translates the specifications to a computer program called TSEI (Time Series Event Identifier), capable of analyzing the time series and identifying the events that meet the expert specifications. The TSEI runs through the time series identifying the subsequences that meet the expert's event specification, calculates the values of the event attributes (as indicated above for falls) and outputs the data structures for such events.

As the event identifier (TSEI) is automatically generated from the event specification, the entire process (specification of each event type, TSEI generation and TSEI execution for event identification) can be iterated until it outputs the best event specification for the target objectives in the application domain.

The time series reference model generation method receives a set of time series $S = \{S_1, S_2, \ldots, S_n\}$, each containing a number of events, and generates a reference model M that represents this set of time series. The key idea of the algorithm is to build the model M on the basis of the most characteristic events, that is, events that appear in a higher number of time series in S. This method is composed of the following steps:

1. **Initialize the model**, that is, $M = \emptyset$.
2. **Identify events**. Use TSEI to scan the series in S extracting all the events E_v with their respective attributes.
3. **Determine the typical number of events *m*** that will make up the model. We have chosen the mode (m) of the number of events in the time series in S to assure that the model that represents the set has the same number of events as most of the series in S. If the distribu-

Table 1 Statistic indicators used for single-valued attributes in the reference model

Attribute type	Typical value	Complementary data gathered
Quantitative continuous	Mean	Minimum and maximum values and standard deviation
Quantitative discrete	Mode	Minimum and maximum values
Qualitative ordinal	Mode	Minimum and maximum values
Qualitative nominal	Mode	–

tion of the typical number of events in the time series in S is not unimodal, take the value closest to the mean of the number of events.

4. **Cluster events**. Cluster all the events extracted in step 2. As there is no a priori information for specifying the optimum number of clusters in each domain, use bottom-up hierarchical clustering, as the number of clusters does not have to be specified beforehand using this technique.

Repeat steps 5 to 9 *m* times

5. **Select the most significant cluster C_k.** Cluster significance is given by the number of time series that have events in that cluster over the total number of time series n (►Equation 6).

$$SIGNF(C_k) = \frac{\#\,TS\,(C_k)}{n} \qquad (6)$$

A cluster may contain not just one but several events from one time series. For this reason, a cluster selected as being the most significant is not discarded in later iterations.

6. **Extract the event E_c that best represents the cluster C_k,** that is, the event E_c that minimizes the distance to the other events in the cluster. City-block distance is used for this purpose. Let S_j be the time series in which event E_c was found.
7. **Add event E_c to the model**, that is, $M = M \cup E_c$.
8. **Discard event E_c**. This event is discarded as it should not be taken into account again in later iterations.
9. **Discard events E_p,** which are the events most like E_c within C_k. For each time series $S_i \neq S_j$ in cluster C_k, discard the event $E_p \in S_i$ that is closest to the representative event (E_c) output in step 6. Each E_p will be represented in the model by the event E_c and, therefore, these events E_p are discarded to assure that they are not considered in later iterations.
10. **Return *M* as a model of the set *S*.**

The overall structure of the proposed method is shown on the left of ►Figure 8. On the right, ►Figure 8.7 shows an example of the application of the proposed method to a set of time series $S = \{S_1, S_2, S_3, S_4\}$ where the method outputs the model M composed of two events.

5. Results. Analysis and Validation

This section reports the results of applying the outlier detection and reference model generation methods to the domain of stabilometry.

5.1 Outlier Detection

We have used a group of 127 stabilometric tests completed by elite athletes to validate the proposed outlier detection method. These athletes were of different sex, ages and practiced different sports, and they all had very good balance. To this group we added another 15 tests by non-athletes with very wide-ranging, but generally much higher levels of instability than the group of athletes. We selected one of the tests (Uni-

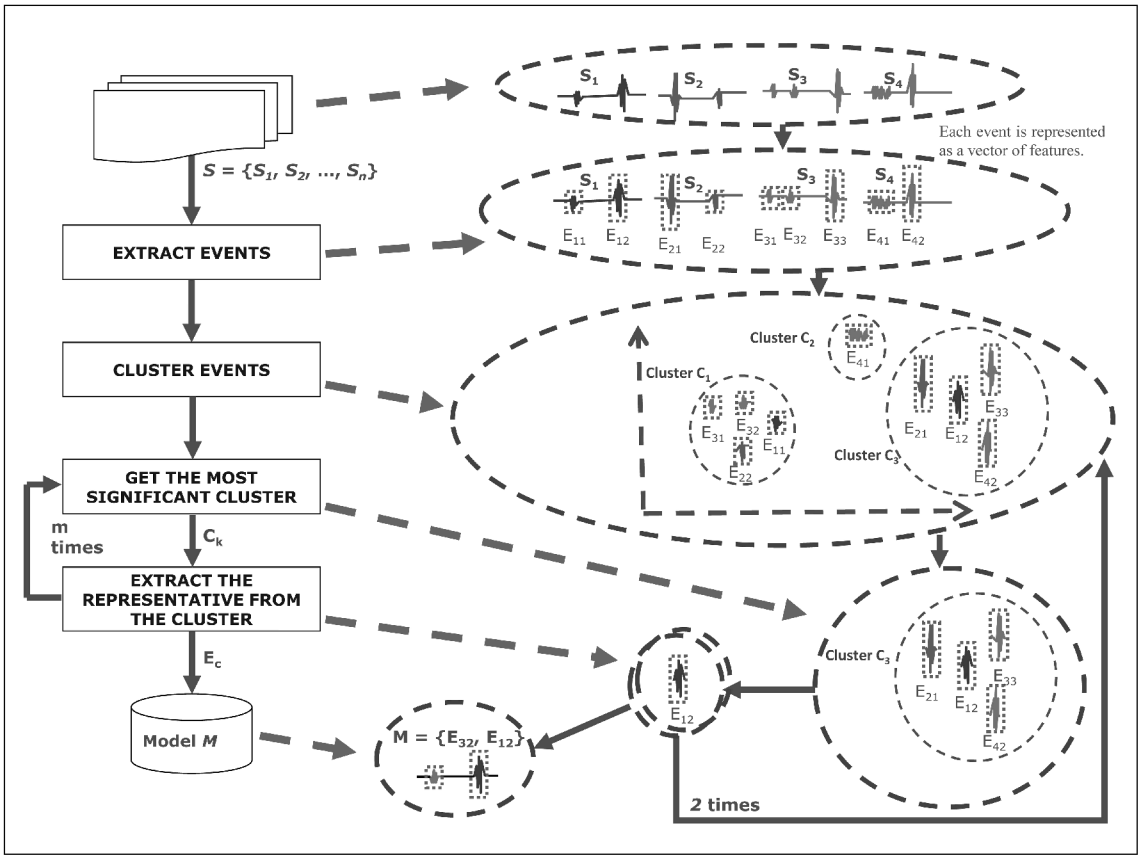

Figure 8
Overall structure and example of the proposed method for time series model generation

Table 2 Outlier detection (unstable and very unstable) by experts

	Stable	Unstable	Very unstable
Experts	S1–S127, N10, N11, N14	N2, N5–N9, N12, N13, N15	N1, N3, N4

Table 3 Results of the application of the outlier detection method adjusting the value of the dispersion parameter *d*. Patients defined as unstable by the expert are shown in italics.

d	S1–S6, S8–S50, S52–S127	N10, N11, N14, *N15*	S7, S51, *N2, N5–N9, N12, N13*	*N1, N3, N4*
0–0.7	Stable	Outliers		
0.8–0.9	Stable		Outliers	
1	Stable			Outliers

Table 4 Recall and precision for the outlier detection process performed to adjust the value of the dispersion parameter *d*

d	Stable detection		Outlier detection	
	recall	precision	recall	precision
0–0.7	96.1% (125/130)	100% (125/125)	100% (12/12)	70.5% (12/17)
0.8–0.9	98.4% (128/130)	99.2% (128/129)	91.6% (11/12)	84.6% (11/13)
1	100% (130/130)	93.5% (130/139)	25% (3/12)	100% (3/3)

lateral Stance – UNI) that are part of the stabilometric examination. These input data have been called S1, S2, …, S127 (for athletes) and N1, N2, …, N15 (for non-athletes).

Then the experts classified these 142 tests according to the level of instability of stance that they indicated. All the tests completed by the elite athletes and tests N10, N11 and N14 were assigned to the same group, as they recorded hardly any instability (a total of 130 stable cases). The other tests were considered unstable (12 cases), and the experts specifically highlighted that tests N1, N3 and N4 were very unstable (▶Table 2).

We then applied the proposed outlier detection method varying the d factor from 0 to 1, with 0.1 increments. This factor represents domain dispersion. The results showed that the system considers tests S7, S51 and N1 to N15 to be outliers within the interval of d from 0 to 0.7. By increasing the value of *d*, tests N10, N11, N14 and N15 are no longer considered outliers within the interval 0.8 to 0.9. If *d* is increased from 0.9 to 1, only tests N1, N3 and N4 are considered outliers (▶Table 3).

▶Table 4 shows a more thorough analysis of the effect of the evolution of parameter *d*, stating the precision and recall values[a] for the detection of outliers (a total of 12 unstable and very unstable patients according to ▶Table 2) and non-outliers (130 stable patients according to ▶Table 2).

From the tests run, we find that, for this domain, a factor of dispersion of 0.8 is the best. For this value, the system correctly classified 139 of the total of 142 tests, that is, 97.8%. For this dispersion factor value, the recall and precision percentages are high for both stable and unstable patients.

[a] $$precision = \frac{truepositives}{truepositives + falsepositives}$$

$$recall = \frac{truepositives}{truepositives + falsenegatives}$$

The proposed method proves to be flexible and is able to reliably define the value of *d*, enabling users to run the tests that are best suited for their experimentation domain.

5.2 Reference Model Generation

We have worked with a set of data composed of 30 stabilometric examinations completed by professional basketball players and 30 elite ice-skaters, to validate the reference model generation method. This is a reasonable number of individuals taking into account that there is a relatively small number of elite athletes. Considering that each individual (stabilometric examination) contains two to three megabytes of information including different types of data, such as single-valued data and multidimensional time series, the size of the data set is rather large, specially taking into account that the comparison of individuals is a far from straightforward task.

We have used stratified k-fold cross validation, with k = 5 (because of the size of the data set). We have run five iterations to generate the reference models for both sports (basketball and ice-skating). Each model was built from 24 individuals for the respective sport. Thus, this process outputs two reference models in each iteration, one for each sport; the 6 + 6 remaining sport cases are classified according to their similarity with each of the two generated reference models (using the method for comparing two data sets described in Section 4.1). Based on previous experiments, the experts determined that classification (individual → model) is successful provided that the similarity value used to compare a test against a model was greater than or equal to 0.9.

Thus, each test case (6 + 6) is compared with each of the two reference models created by the method: if the similarity value between the test case and a model is greater than or equal to 0.9, this case is assigned to the respective class for that model (and the case may be assigned to one, both or neither of the classes).

For the purposes of the experiment, the test cases of the basketball players were labeled B1 to B30 and the ice-skaters S1 to S30. The results (▶Table 5, left side) show

Table 5
Results of the tests run with the stabilometric reference models

Stratified 5-fold cross validation	Classified as basketball players	Classified as ice-skaters	Unclassified	Non-athletes	
				Correctly unclassified as expected	Incorrectly classified
Iteration 1	B1, B2, B3, B4, B5, B6	S1, S2, S3, S4, S5, S6		C1–C30	
Iteration 2	B7, B8, B9, B10, B12	S7, S8, S9, S11, S12	B11, S10	C1–C9 , C11–C30	C10
Iteration 3	B13, B14, B15, B16, B18	S13, S14, S15, S16, S17, S18	B17	C1–C9, C11–C24, C26–C30	C10, C25
Iteration 4	B19, B20, B21 B22, B23, B24	S19, S20, S21 S22, S23, S24		C1–C30	
Iteration 5	B25, B26, B27, B28, B30	S25, S27, S28, S29, S30	B29, S26	C1–C9 , C11–C30	C10, C25
Summary	Training sets: 5 x (24 + 24) Test sets: 5 x (6 + 6) Correctly classified tests: 55 (91.7%)			Test sets: 5 x 30 Correctly unclassified tests: 145 (96.7%)	

that 55 out of a total of 60 tests were correctly classified (91.7%).

For a more rigorous evaluation process, we added a third group of non-athletes (labeled as C1 to C30), which is used as a control group at the Spanish Council for Sports. No reference model is generated for this group, as it includes individuals with very different features; all we do is classify the group members (at the end of each cross validation iteration), that is, see whether or not they fit any of the two sports models. The expected result is that the system should not classify individuals in this control group as members of either sports group (referred to here as "correctly unclassified"), as the individuals will not generally be similar enough to the model of either sport. The results (▶Table 5, right side) show that 145 out of a total of 150 tests (96.7%) were correctly unclassified as expected. On this ground, we consider that these results confirm the goodness of the reference model generation method. This is also endorsed by the experts using the application routinely. They state that the models generated for different sets of individuals are providing satisfactory results.

The times series model generation method has been compared with other approaches on electroencephalographic (EEG) time series data [37], using publicly available datasets (described in [38]).

The complete data set consists of five sets (denoted *A*–*E*) composed of time series generated by EEG devices, each containing 100 single-channel (100 electrodes) EEG recordings of the five patient classes (*A*–*E*). For this study, we focused on sets labeled *A* (healthy patients) and *E* (epileptic seizure session recordings).

The ultimate aim of the evaluation is to measure how good the model generation method is. To do this, it has been compared with other methods, namely an adaptive fuzzy inference neural network system (AFINN) and a classic multilayer perceptron neural network (MLP). We have used a 10-fold cross validation approach in order to evaluate the proposed method. In each iteration, two models were created for each class ($M_{healthy}$ and $M_{epileptic}$). The first model ($M_{healthy}$) was created from a training set composed of 90 of the 100 healthy patients (set *A*). The other 10 patients constituted the test set. The second model ($M_{epileptic}$) was generated from a training set composed of 90 of the 100 epileptic patients (set *E*). The other 10 patients were used as the test set. The patients in the test set were chosen at random.

The generated models were evaluated by checking whether the $M_{healthy}$ model properly represents the group of healthy patients and whether the $M_{epileptic}$ model is representative of the group of epileptic patients. To do this, we classified the 20 individuals in the test group according to their similarity to the two models.

The training data set was used to train the AFINN model to classify the two classes of EEG signals. The proposed system was trained and tested with the extracted features using the discrete wavelet transform of the EEG signals. The simulation results reveal a perfect performance compared to a classic MLP neural network.

▶Table 6 shows the results of the proposed classifier, using two different training sets. These results are very satisfactory, especially for the critical class of patients with epilepsy (class *E*). This shows that our framework provides comparable results to other more specific methods. Note that the input set used in this research contains only EEG time series data. While this is useful for testing the operation of our reference model generation method for time series, it does not take full advantage of the potential of the framework presented here, which is capable of operating with structurally complex individuals composed of single-valued and time series data.

6. Conclusions

In this paper, we have presented a framework for discovering knowledge from hierarchically organized structurally complex data. This framework includes methods for representing and comparing individuals,

Table 6 Comparison of the three methods

Class	Reference Model	AFINN	MLP
A	92%	98.12%	94.98%
E	96%	97.96%	95.86%

detecting outliers and building reference models.

First, UML was extended to represent any set of hierarchically organized structurally complex data with single-valued attributes and both one-dimensional and multidimensional time series. This notation is useful for automating tasks, such as the comparison of individuals and the generation of reference models for this type of data.

Second, we have designed a method for comparing individuals represented by this type of hierarchically organized structurally complex data. Taking the conceptual model of the data as a structural guide, the method calculates the similarity between individuals at different levels of the hierarchy until it obtains a final value at the root node.

Third, we have developed an outlier detection and filtering method that will be used as a previous step to the generation of reference models.

Finally, we have developed a method for generating reference models of a set of structurally complex individuals. The primary component of this method is a submethod for building reference models of a set of time series based on the analysis of the clusters of the events in those series. This submethod includes an algorithm for comparing time series in which important information is confined to certain regions of the time series and the remainder of the series provides hardly any information.

The described framework has been applied with satisfactory results to structurally complex data from the field of stabilometry, a discipline that studies balance in human beings. Also, we are currently applying this framework to traffic flow forecasting [39]. In this research, we use the California Department of Transportation PEMS-SF data set (http://pems.dot.ca.gov/) downloaded from the UCI Machine Learning Repository. This data set contains 15 months' worth of daily traffic flow data (from January 1, 2008 to March 30, 2009) and is larger than 400 Mb. The data contains the occupancy rate of car lanes of San Francisco Bay Area freeways. We have defined the reference model for the weekend and working day traffic flow with good forecasting results. We are now refining these models to be able to identify each particular day of the week. This research is currently under development.

References

1. Bellazzi R, Diomidous M, Sarkar IN, Takabayashi K, Ziegler A, McCray AT. Data Analysis and Data Mining: Current Issues in Biomedical Informatics. Methods Inf Med 2001; 50 (6): 536–544.
2. Jouhet V, Defossez G, Burgun A, Le Beux P, Levillain P, Ingrand P, et al. Automated Classification of Free-text Pathology Reports for Registration of Incident Cases of Cancer. Methods Inf Med 2012; 51 (3): 242–251.
3. Rantner LJ, Stühlinger MC, Nowak CN, Spuller K, Etsadashvili K, Stühlinger X, et al. Localizing the Accessory Pathway in Ventricular Preexcitation Patients Using a Score Based Algorithm. Methods Inf Med 2012; 51 (1): 3–12.
4. Harle CA, Downs JS, Padman R. A Clustering Approach to Segmenting Users of Internet-based Risk Calculators. Methods Inform Med 2011; 50 (3): 244–252.
5. Paoin W. Lessons Learned from Data Mining of WHO Mortality Database. Methods Inf Med 2011; 50 (4): 380–385.
6. Bethel CL, Hall LO, Goldgof D. Mining for Implications in Medical Data. In: Tang YY, Wang SP, Lorette G, Yeung DS, Han T, editors. Proceedings of the 18th International Conference on Pattern Recognition; Aug 20–24, 2006; Hong Kong, China. Washington, DC: IEEE Computer Society; pp 1212–1215.
7. Lama E, Mello P, Nanetti A, Riguzzi F, Storari S, Valastro G. Artificial Intelligence Techniques for Monitoring Dangerous Infections. IEEE T Inf Technol B 2006; 10 (1): 143–155.
8. Cios KJ, editor. Medical Data Mining and Knowledge Discovery. Heidelberg: Springer; 2001.
9. Clarke R, Ressom HW, Wang A, Xuan J, Liu MC, Gehan EA. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nat Rev Cancer 2008; 8: 37–49.
10. Terekhov Y. Stabilometry as a diagnostic tool in clinical medicine. Can Med Assoc J 1976; 115 (7): 631–633.
11. Embley DW, Thalheim B, editors. Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges. Berlin: Springer; 2011.
12. Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide. 2nd ed. Reading, MA: Addison-Wesley; 2005.
13. OMG Unified Modeling Language (OMG UML). Infrastructure specification. Version 2.4.1. 2011 Aug (cited 2011). Available from: http://www.omg.org/spec/UML/2.4.1/Infrastructure
14. OMG Unified Modeling Language (OMG UML). Superstructure specification. Version 2.4.1. 2011 Aug (cited 2011). Available from: http://www.omg.org/spec/UML/2.4.1/Superstructure
15. Yang R, Kalnis P, Tung AKH. Similarity evaluation on tree-structured data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, June 14–16, 2005, Baltimore, MA. New York, NY: ACM; pp 754–765.
16. Li G, Liu X, Feng J, Zhou L. Efficient Similarity Search for Tree-Structured Data. In: Ludäscher B, Mamoulis N, editors. Proceedings of 20th International Conference, SSDBM; July 9–11, 2008; Hong Kong, China. Lecture Notes in Computer Science 5069. Berlin/Heidelberg: Springer-Verlag; 2008. pp 131–149.
17. Alonso F, Martínez L, Pérez A, Santamaría A, Valente JP. Modelling Medical Time Series Using Grammar-Guided Genetic Programming. In: Perner P, editor. ICDM 2008: Proceedings of the Industrial Conference on Advances in Data Mining: Medical Applications, E-Commerce, Marketing and Theoretical Aspects; July 16–18, 2008; Leipzig, Germany. Lecture Notes in Computer Science 5077. Berlin/Heidelberg: Springer-Verlag; 2008. pp 32–46.
18. Agrawal R, Faloutsos C, Swami A. Efficient Similarity Search in Sequence Databases. FODO Conference; Oct 13–15, 19935; Evanston, IL.
19. Chan K-P, Fu AW-C. Efficient Time Series Matching by Wavelets. In: Kitsuregawa M, Papazoglou MP, Pu C, editors. Proceedings of the 15th International Conference on Data Engineering; March 23–26, 1999; Sydney, Australia. Washington, DC: IEEE Computer Society; 1999. pp 126–133.
20. Kahveci T, Singh AK, Gürel A. An Efficient Index Structure for Shift and Scale Invariant Search of Multi-Attribute Time Sequences. In: Agrawal R, Dittrich KR, editors. Proceedings of the 18th International Conference on Data Engineering; Feb 26–March 1, 2002 ; San Jose, CA. Washington DC: IEEE Computer Society; 2002. p 266.
21. Perng C-S, Wang H, Zhang SR, Parker DS. Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases. In: Lomet DB, Weikum G, editors. Proceedings of the 16th International Conference on Data Engineering; Feb 28–March 3, 2000; San Diego, USA. Washington, DC: IEEE Computer Society. pp 33–44.
22. Negi T, Bansal V. Time Series: Similarity Search and its Applications. In: Proceedings of the International Conference on Systemics, Cybernetics and Informatics; Jan 7–9, 2005; Hyderabad, India. Hyderabad, India: Pentagram Research Centre Pvt. Ltd; 2005. pp 528–533.
23. Rakthanmanon T, Keogh EJ, Lonardi S, Evans S. Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data. In: Cook D, Pei J, Wang W, Zaïane O, Wu X, editors. ICDM 2011: Proceedings of the 11th IEEE International Conference on Data Mining; Dec 11–14, 2011; Vancouver, Canada. Washington, DC: IEEE Computer Society. pp 547–556.
24. Lara JA, Pérez A, Valente JP, López-Illescas A. Comparing time series through event clustering. In: Corchado JM, De Paz JF, Rocha MP, Fernández Riverola F, editors. IWPACBB'08: Proceedings of the 2nd International Workshop on Practical Applications of Computational Biology & Bioinformatics; Oct 22–24, 2008; University of Salamanca, Spain. Berlin: Springer; 2009. pp 1–9.
25. Kuhnt S, Griefahn B. Annoyance from multiple transportation noise: Statistical models and outlier detection. Methods Inf Med 2004; 43 (5): 510–515.
26. Krusinska E, Mathiesen UL, Franzen L, Bodemar G, Wigertz O. Influence of Outliers on the Association between Laboratory Data and Histopatho-

logical Findings in Liver-Biopsy. Methods Inf Med 1993; 32 (5): 388–395.
27. Stefatos G, Hamza AB. Cluster PCA for Outliers Detection in High-Dimensional Data. Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics; Oct 7–10, 2007; Montréal, Canada. IEEE; 2007. pp 3961–3966.
28. Wang J-S, Chiang, J-C. A Cluster Validity Measure with Outlier Detection for Support Vector Clustering. IEEE T Syst Man Cy B 2008; 38 (1): 78–89.
29. Yang P, Huang B. A Spectral Clustering Algorithm for Outlier Detection. In: Tan H, editor. FITMW'08: Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering; Nov 20, 2008. Washington, DC: IEEE Computer Society; 2008. pp 33–36.
30. Zhang T, Ramakhrishnan R, Livny M. Birch: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS, editors. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data; June 4–6, 1996; Montréal, Canada. ACM Press; 1996. pp 103–114.
31. Li XY, Ye N. A supervised clustering algorithm for computer intrusion detection. Knowl Inform Syst 2005; 8 (4): 498–509.
32. Yoon K-A, Kwon O-S, Bae D-H. An Approach to Outlier Detection of Software Measurement Data using the K-means Clustering Method. In: ESEM '07: Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement; Sep 20–21, 2007; Madrid, Spain. Washington, DC: IEEE Computer Society; 2007. pp 443–445.
33. Torgo L, Soares C. Resource-bounded Outlier Detection using Clustering Methods. Proceedings of the 2010 Conference on Data Mining for Business Applications; 2010. The Netherlands: IOS Press Amsterdam. pp 84–98.
34. Chan PK, Mahoney MV. Modeling Multiple Time Series for Anomaly Detection. In: Proceedings of the 5th IEEE International Conference on Data Mining; Nov 27–30, 2005; Houston, TX. Washington, DC: IEEE Computer Society; 2005. pp 90 – 97.
35. Rombo S, Terracina G. Discovering Representative Models in Large Time Series Databases. In: Christiansen H, Hacid M-S, Andreasen T, Larsen HL, editors. Proceedings of the 6th International Conference on Flexible Query Answering Systems; June 24–26, 2004; Lyon, France. Lecture Notes in Computer Science 2004; 3055. Berlin: Springer; pp 84–97.
36. Lara JA, López-Illescas A, Pérez A, Valente JP. A Language for Defining Events in Multi-Dimensional Time Series: Application to a Medical Domain. In: Troncoso A, Arias M, editors. Proceedings of the 1st International Workshop on Mining of Non-Conventional Data; Nov 13, 2009; Seville, Spain. Seville: Universidad de Sevilla; 2009.
37. Jahankhani P, Lara JA, Pérez A, Valente JP. Two Different Approaches of Feature Extraction for Classifying the EEG Signals. In: Iliadis L, Jayne C, (editors). Engineering Applications of Neural Networks: Proceedings of the 12th INNS EANN-SIG International Conference (EANN 2011) and 7th IFIP WG 12.5 International Conference (AIAI 2011); Sep 15–18, 2011; Corfu, Greece. IFIP Advances in Information and Communication Technology Series; 363. Berlin: Springer; 2011. pp 229–239.
38. Jahankhani P, Revett K, Kodogiannis V. EEG Signal Classification Using Wavelet Feature Extraction and Neural Networks. In: IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing; Oct 3–6, 2006; Sofia, Bulgaria. pp 120 – 125.
39. Bo Zhu. Applying Event-based Data Mining to Traffic Flow Forecasting. Master Thesis, Universidad Politécnica de Madrid, Julio 2012.