Clustering of Electrocardiograph Signals in Computer-aided Holter Analysis.

David Cuesta-Frau*, Juan C. Pérez-Cortés, Gabriela Andreu-García

Department of Computer Engineering (DISCA), Polytechnic University of Valencia, EPSA, Plaza Ferrándiz y Carbonell 2, 03801, Alcoi, Spain

Abstract

A number of methods for temporal alignment, feature extraction, and clustering of electrocardiographic signals are proposed. The ultimate aim of the paper is to find a method to automatically reduce the quantity of beats to examine in a long-term electrocardiographic signal, known as Holter signal, without loss of valuable information for the diagnosis. These signals include thousands of beats, and therefore, visual inspection is difficult and cumbersome.

All the elements involved in each stage will be described and a thorough experimental study will be presented. The electrocardiograph signals used in the experiments belong to the well-known MIT database, where many different waveforms can be found. Based on the results of the experiments, a complete process is proposed to obtain the significant beats present within a signal, with a reasonable computational cost. Hence, cardiologists will only have to examine a small but fully representative subset of beats, making this method a very useful tool for medical decision support systems.

Keywords: Electrocardiogram, Holter Register, Medical Decision Support Systems, Signal Clustering, Signal Feature Extraction.

1. Introduction

The importance of electrocardiographic signals for the diagnosis of many cardiac diseases is widely known. Among the different types of electrocardiogram registration and interpretation, there is a special case where an automated aid for inspection is of major importance, since the duration of the signal makes thorough visual inspection almost impossible. Such signals are known as Holter registers [1], and they are very useful in the detection of some diseases like cardiac arrhythmias, transient ischemic episodes and silent myocardial ischemia [2], which are not readily detected in a short-time electrocardiogram.

Holter registers are ambulatory recordings of the heart activity, with a duration ranging from 24 to 48 hours. Record analysis is performed off-line.

Due to its long duration, in a single recording there can be more than 100,000 beats to examine, being very important not to leave out any beat, since the diagnosis can depend on just a few of them.

However, the intra-signal variability along the time axis is often quite low in a typical Holter register. This means that, among the complete set of beats, only a few abnormal periods are usually found, possibly scattered along the signal. By examining only these abnormal periods, the diagnosis could be yielded with a great reduction in effort. In this paper, we try to design automatic methods to extract a small set of representative beats from the whole electrocardiogram.

Two different approaches can be followed in the design of a procedure to extract the representative beats: supervised and unsupervised. In the first case, it is necessary to have a complete corpus of manually labeled beats, and classify each new beat of the signal under analysis into one of c known classes, giving as a result an automatic diagnosis. The main drawback of this technique is derived from the high variability of the electrocardiograph signals obtained from patients with very diverse pathologies, different acquisition equipment, derivations, etc. The automatic diagnosis obtained would certainly not be reliable enough to replace the visual inspection by a trained cardiologist.

On the other hand, unsupervised techniques do not need a labeled set and are more flexible. However, since no manually labeled reference beats are used, an automatic diagnosis cannot be obtained through these techniques. This is the type of method proposed here, based on examining all the beats and using a suitable dissimilarity measure to obtain an unsupervised classification (or clustering) of the set. The different families of beats present in the signal are grouped so that by only examining one from

---

* Corresponding author. Tel: +34 966528459; fax: +34 966528409
E-mail address: dcuesta@disca.upv.es
This work was carried out within the Polytechnic University of Valencia-Hospital of Alcoy Co-operative Program
each family, the diagnosis can be accurate. The goal is that the physician will be able to accurately diagnose any latent disease by only examining a prototype from each subset of homogeneous beats.

There must be a compromise between the number of groups and the size of each group, in order for the method to be useful and practical. Obviously, if the number of families is high enough to be comparable to the number of beats, the time needed to examine all of them will be comparable to the time needed using plain visual inspection. On the other hand, if the quantity of groups is very low, the probability of an abnormal beat remaining hidden in a cluster of normal beats increases.

The main constraint of the task is that no significant beat should be omitted. In this sense, a moderately high number of clusters is acceptable, but a cluster with beats in it that are significantly different from the prototype cannot be allowed.

In the rest of the paper, a number of methods to carry out the task described are discussed, and their results are compared.

2. Method

In this section, the operations performed on the data, starting with acquisition and signal segmentation to the final stage of clustering and visual inspection, are described. The main phases of the process to obtain a suitable experimental set of beats [3] are shown in Fig.1.

![Fig. 1. Phases of the process to obtain an experimental set of beats.](image)

The signals used in the experiments belong to different registers of the MIT database. These signals are also included in the Physionet database [4], and some of them are labeled. In order to deal with as many waveforms as possible, the registers chosen correspond to different diseases.

The reason for using this database among others is due to [5]: easy availability, widespread use in research by groups involved in biomedical engineering, wide range of diseases covered, many registers with duration ranging from a few minutes to several hours, the fact that it is labeled, and very well documented.

The following sections describe all the stages involved in obtaining a proper initial beat set \( P \).

2.1 Signal Segmentation

First of all, once a discrete time electrocardiograph signal \( e_0[n] \) has been acquired by the sampling device, it must be processed in order to reduce the noise, the baseline wandering and other artifacts that can be present within it, obtaining a greater quality signal \( e'[n] \).

Then, the individual beats must be isolated and extracted from the signal. In order to carry out this task, a QRS detector algorithm is applied to find the position of the R waves. Each position is referenced as a time index, obtaining the set:

\[
t_R = \{t_{R_1}, t_{R_2}, \ldots, t_{R_n}\}
\]

The algorithm employed in this research uses the first derivative and a number of non-linear transformations of the signal, as described in [6].

Once the position of each R wave is known, the beats are segmented as follows: The beginning and end of the beat are placed at two given time distances \( i_{R_t}^1 \) and \( i_{R_t}^2 \) before and after the R wave, respectively. The values used in this research are \( i_{R_t}^1 = t_{R_t} - 0.2T_t \), and \( i_{R_t}^2 = 0.8T_t + t_{R_t} \), where \( T_t = t_{R_{t+1}} - t_{R_t} \), is the RR interval at that point, i.e. the distance between two consecutive R waves, as depicted in Fig.2.

![Fig. 2. Location of a beat using the R wave position as a reference.](image)
Some overlapping between consecutive beats is allowed.

2.2. Beat normalization

When comparing two beats, their waveforms \( p_i \) and \( p_j \) are considered equivalent if the following equation holds [7]:

\[
p_i(t) = \alpha_j \cdot f(t) + \tau
\]

where \( \alpha \) is an amplitude scaling factor, \( f \) a time-scale distortion function and \( \tau \) a time delay. Therefore, amplitude differences, time distortions, and delays, must be compensated before comparing two beats, or else, a dissimilarity measure translation and scale-invariant in time and amplitude must be used.

Differences in amplitude are easy to avoid by applying a normalization factor. In this study, the average module of amplitude of each beat is used instead of others such as the maximum value, in order to reduce as much as possible the effect of spurious peaks or outliers. The normalization factor for each beat \( p_i \) is obtained from the expression (in the discrete time domain):

\[
M_i = \frac{\sum_{k=0}^{n_i-1} |p_i[k]|}{n_i},
\]

where \( n_i \) is the number of samples of beat \( p_i \).

The effect of the time delay \( \tau \) could be integrated into function \( f \), but it is presented as an independent value because it has a special meaning as the error of the signal segmentation process. A robust signal segmentation procedure like the one presented in the previous section minimizes the effect of this time shift between beats.

Finally, to modelize the distortions in the time axis, two methods of temporal alignment will be used: linear and non-linear temporal alignment. In general, depending on the cardiac rhythm, the duration of the ECG waves can change very significantly. Additionally, the relation among the durations of some parts of the QRS wave for beat waveforms acquired at different cardiac rates is not linear, suggesting the need of non-linear rather than linear temporal alignment. Anyway, both methods of temporal alignment will be studied in order to compare not only the accuracy, but the computational cost as well.

The temporal alignment will be performed along with the computation of the dissimilarity measure during the clustering process.

2.3 Beat labeling

In each experiment, once all the beats have been extracted from the signal and have been normalized in amplitude, manual labeling is performed so as to later be able to quantify the errors. In order to enable other researchers to perform a comparative analysis, this labeling procedure is that implemented in the MIT database, as described in [4,5]. An expert assigned a class label to each beat, according to its shape and clinical meaning. In the particular case of very noisy beats or incomplete beats, we have added a common label (meaning “noise”) that has been assigned to all the beats of this kind. In Fig.3, a labeling example is shown.

![Fig. 3. Example of labeling. The beats shown correspond to the register e0411 of the MIT Database. Labels on the top of the beats, N and V stand for normal and premature ventricular contraction beat, respectively.](image)

The ideal result would be a set of clusters, each one composed of beats belonging to the same class. The cardiologist would then need to examine just one beat from each cluster to formulate an accurate diagnosis, although the whole set is still available in case further examination was required. To take a beat representative of each cluster, the most centered one is found, in order to avoid outliers and minority kinds.

To measure the goodness of the clustering results, two error criteria will be considered: clustering error and critical error.

The first one, the clustering error, takes into account the number of beats included in a cluster, but belonging to a class different from the dominant class in that cluster. This measure of error is commonly used in clustering and classification techniques.

The second one, the critical error, has been introduced by the clinicians. It measures the number of beats belonging to a class that is not the dominant one in any cluster. The rationale behind this error measure is that, in case some kind of beats does not have its own cluster, no representative of such kind will be examined, and therefore, their clinical meaning will not be taken into account.

In Fig.4, a basic example of these two measures is shown. In cluster I, \( N \) are the majority kind of beats, and \( V \) beats correspond to clustering error.
the contrary, \( V \) beats are the majority kind and \( N \) beats the clustering error in cluster II. \( J \) beats are also hidden among \( V \) beats in cluster II, making its examination very unlikely.

![Fig. 4. Error measures example.](image)

\( N \) and \( V \) beats are considered clustering errors when not placed in the cluster where they are majority. \( J \) beats are considered critical error since no cluster is composed of such beats preponderantly.

### 2.4. Dissimilarity measure

To identify the groups of similar beats that can be represented with a single prototype, it is necessary to define a quantitative similarity (or dissimilarity) measure. The Minkowsky norms \( L_1 \) and \( L_2 \) are widely used as dissimilarity measures in \( \mathbb{R}^n \):

\[
d = \sum_j |x_j - y_j|
\]

\[
d = \sqrt{\sum_j (x_j - y_j)^2}
\]

Nevertheless, beats with different number of samples cannot be compared using these norms. In this case, some kind of time normalization must be used.

In Section 2.2, the need for time-normalization or time-alignment of beats of different duration is addressed. In this study, Dynamic Programming is used to perform this task, with some local and global restrictions. In order to compare the results, linear temporal alignment based on uniform upsampling and downsampling was also used.

The Dynamic Programming algorithmic paradigm can be used to optimize a cost function. In this sense, it has been applied successfully to tasks such as speech recognition, economics, process control, etc [8]. Dynamic Time Warping (DTW) [9] is a particular instance of this paradigm that makes it possible to find an optimal alignment function between two sequences of different length. The cumulative cost of the alignment path obtained can be used as a dissimilarity measure.

Let \( x, y \) be two sequences of length \( n_x \) and \( n_y \), respectively, accounting for two beats under comparison:

\[
x = [x[0], x[1], \ldots, x[n_x - 1]]
\]

\[
y = [y[0], y[1], \ldots, y[n_y - 1]]
\]

The alignment procedure allows us to compare each value \( x[i] \) of the first beat with a value \( y[j] \) of the second beat. The whole set of possible comparisons can be represented as a matrix \( G \), or as a multistage graph, as shown in Fig. 5.

![Fig. 5. Dynamic Programming Matrix \( G \), and associated multistage graph.](image)

The goal is to find a node path of length \( f \),

\[(i_1, j_1)(i_2, j_2), \ldots, (i_f, j_f),\]

such that the final cost along that path is the minimum possible, according to certain constraints.

For each node, the cost involved in the transition from the previous node in the path, \((i_{k-1}, j_{k-1})\),

\[
d[(i_k, j_k) | (i_{k-1}, j_{k-1})]
\]

is computed. The final cost is equal to the cumulative sum:

\[
D = \sum_{k=1}^{f} d[(i_k, j_k) | (i_{k-1}, j_{k-1})]
\]

In each transition, a weighting function \( w \) can be applied, and the whole expression can be normalized in order to get a measure independent of the path length:
\[
D = \frac{\sum_{k=1}^{\xi} d((i_k, j_k) \mid (i_{k-1}, j_{k-1})) \cdot w(a_k, b_k)}{\sum_{k=1}^{\xi} w(a_k, b_k)}
\]

where \( a = i_k - i_{k-1} \), and \( b = j_k - j_{k-1} \).

To perform this computation efficiently, each value \( G[i, j] \), of the dynamic programming matrix is calculated according to the cost accumulated in the previous nodes \( G[i-a, j-b] \), and their associated transition costs:

\[
G[i, j] = \min_{(a,b)} \left( G[i-a, j-b] + d((i, j) \mid (i_{k-1}, j_{k-1})) \cdot w(a_k, b_k) \right)
\]

Several alternatives are possible for the local constraints. In our case, only three transitions between consecutive nodes have been allowed, with the following values for the transition weights:

\[
G[i, j] = \min \left( \begin{array}{c}
G[i-1, j-1] + d((i, j) \mid (i_{k-1}, j_{k-1})) \cdot w(a_k, b_k) \\
G[i-1, j] + 2d((i, j) \mid (i_{k-1}, j_{k-1})) \\
G[i, j-1] + 2d((i, j) \mid (i_{k-1}, j_{k-1}))
\end{array} \right)
\]

Global constraints can also be applied in order to keep the search path inside a region where the alignment is more meaningful from the point of view of the application. To avoid paths or path portions along the boundaries of the dynamic programming matrix, the search is reduced to a centered window, assigning a high cost to the nodes outside that region.

\[
G[i, j] = \infty, \text{if } ((i, j) \not\in \Sigma)
\]

2.5. Feature Extraction

Four feature extraction techniques have been tested, and their error rates and computational costs have been compared.

The simplest features considered were the signal amplitudes obtained from the acquisition system (after standard signal conditioning). These features are taken as a reference for the rest of the feature extraction techniques, aimed at reducing the number of elements in the sequence representing each beat, but without losing distinctive information. In this case, each beat is represented by a discrete time amplitudes sequence of length \( n_x \), being \( x \) the name of the sequence:

\[
x = \{x[0], x[1], \ldots, x[n_x-1]\}
\]

The second feature extraction method tested is based on trace segmentation. Trace segmentation is a technique used in Speech Recognition to normalize the duration of different utterances in order to compare them using standard vector-space norms [10]. In addition to this use, we have also adapted this technique to work as a non-uniform sampling method for feature extraction.

The goal of the process is to obtain only those samples of the signal where the main changes take place.

First of all, the partial accumulated derivative \( \Delta_j \) of the sequence \( x \) under analysis is calculated using equation (6):

\[
\Delta_j = \sum_{i=1}^{\xi} |x[i] - x[i-1]|
\]

And, therefore, the accumulated derivative at the end of the sequence is:

\[
\Delta = \sum_{i=1}^{n_x-1} |x[i] - x[i-1]|
\]

If the number of desired samples in the final set after trace segmentation is \( h + 1 \), the height of each uniform amplitude interval from \( \Delta \) is given by:

\[
L = \frac{\Delta}{h}
\]

To obtain the output sequence \( x_p = \{x_p[0], x_p[1], \ldots, x_p[h]\} \) which includes the \( x \) values where the main changes take place, the parameters described in equations (7) and (8) must be used as follows: Each sample \( x_p[i] \) is taken from the sample \( x[j] \) which exceeds an integer multiple of \( L \), as described in equation (9):

\[
x_p[i] = x[j] \quad j = \arg \min_{0 < i < n} (i \cdot L < \Delta_j)
\]

where \( x_p[L] = x[n_x-1] \) and \( x_p[0] = x[0] \). The process is depicted in Fig.6.
The third feature extraction method tested is based on polygonal approximation. Polygonal approximation of curves is a technique used in many computer vision and signal processing applications [11]. The goal of this method is to obtain a list of vertices \( P_x \), in such a way that the polygon associated to this list, represents the curve or signal \( x \) without exceeding a previously defined error threshold \( E \).

In this paper, the polygonal approximation method described in [11] has been modified to be applied to one-dimensional signals. It consists of finding the furthest point from a previous piecewise linear approximation of a signal interval. In case \( E \) is exceeded at any point within some segment \( i \equiv [(x_{i1}, y_{i1}), (x_{i2}, y_{i2})] \), it is split at the maximum error point and the method iterates with each part until no segment exceeds the threshold. For example, in Fig.8, the polygonal approximation of signal in Fig.7 is shown.

\[
\begin{align*}
P_x &= \{P_x[0], P_x[1], \ldots, P_x[n_{P_x}-1]\} \\
\text{where each component corresponds to a coordinate pair:} \\
P_x[j] &= (Cx_j, Cy_j)
\end{align*}
\]

The list of points is computed as described in the recursive algorithm shown in Fig.9.

\[
P_x[j] = \left\{ 
\begin{array}{ll}
P_x[j] &= (Cx_j, Cy_j) \\
\text{Input:} & \text{x: input sequence} \\
\text{E: Maximum approximation error} \\
\text{Output:} & P_x: list of vertices \\
\text{Method:} & \text{SegmentExamination}(0, x[0]), (n_{P_x}-1, x[n_{P_x}-1]) \\
\text{end function} & \text{SegmentExamination}((x_{i1}, y_{i1}), (x_{i2}, y_{i2})) \\
\text{end function} \\
\end{array}
\right.
\]

Fig. 6. Trace segmentation process. The samples of the sequence \( x_n \) are obtained from the samples of the sequence \( x \) where main changes take place.

Fig. 7. Electrocardiograph signal.

Fig. 8. Polygonal approximation of the electrocardiograph signal shown in Fig.7.

The sequence of vertices computed for an input \( x \) is expressed as:

\[
P_x[j] = \left\{ 
\begin{array}{ll}
P_x[j] &= (Cx_j, Cy_j) \\
\text{Input:} & \text{x: input sequence} \\
\text{E: Maximum approximation error} \\
\text{Output:} & P_x: list of vertices \\
\text{Method:} & \text{SegmentExamination}(0, x[0]), (n_{P_x}-1, x[n_{P_x}-1]) \\
\text{end function} & \text{SegmentExamination}((x_{i1}, y_{i1}), (x_{i2}, y_{i2})) \\
\text{end function} \\
\end{array}
\right.
\]

Finally, the fourth feature extraction method tested is based on the wavelet transform. This transform is being increasingly used in many signal processing applications, such as noise reduction, lossy compression, abrupt changes and signal long term evolution detection, etc. Some of these techniques have been successfully applied to ECG processing [12] in particular.

The continuous wavelet transform (CWT) of a signal \( f(t) \) is defined as:

\[
C(a, b) = \int_{-\infty}^{+\infty} f(t)\psi_{ab}(t)dt 
\]

where the function \( \psi_{ab}(t) \) corresponds to a translated and scaled version of the function \( \psi(t) \), known as mother wavelet.
\[ \psi_{a,b}(t) = a^{-\frac{t}{b}} \psi \left( \frac{t-b}{a} \right) \]  \hspace{1cm} (13)

The coefficients obtained \( C(a,b) \), are considered detail or approximation coefficients, depending on which frequency band they represent, high or low respectively. Approximation coefficients are mainly used in lossy compression applications, since the signal can be reconstructed from such coefficients, with the loss of the information contained in the detail coefficients. This last coefficients are used for signal denoising, principally.

Using the discrete-time formulation, and adding the so-called scaling function \( \varphi \), used to allow multiresolution analysis [13], the expression for the Discrete Wavelet Transform (DWT) is:

\[
DWT(f[n]) = \begin{cases} 
  c_k = \langle f[n] \varphi_k[n] \rangle = \sum_n f[n] \varphi_k[n] \\
  d_{jk} = \langle f[n] \psi_{jk}[n] \rangle = \sum_n f[n] \psi_{jk}[n]
\end{cases}
\]

where \( c_k \) are the approximation coefficients, and \( d_{jk} \) the detail coefficients. Values of \( j \) and \( k \) are related to \( a \) and \( b \) values of the continuous version, by the dyadic grid [13].

The Wavelet-based lossy compression of the ECG can be used as a feature extraction method. The approximation coefficients obtained using a Wavelet decomposition of some level of a signal, correspond to the main features of such signal, and they can be used as the values of a sequence for dissimilarity measure among beats [14], as is the case in this study.

For this purpose, a set of sequences from the approximation coefficients \( c_k \) will be created using the third level Wavelet transform of each beat. In [14], a selection of the Wavelet coefficients is carried out, in order to improve the performance of the classifier. In this case, it is not possible since we are dealing with a non-supervised clustering task. The selection of the third level transformation, is due to a trade-off between the accuracy of the signal representation, and the quantity of coefficients required. If the input signal length is \( n_x \), the output sequence length is given by \( \frac{n_x}{2^3} \), which is comparable to the length obtained using trace segmentation.

Summing up, the feature extraction process is depicted in Fig.10.

2.6 Preclustering

Previously to describing clustering techniques, additional considerations must be taken into account. A Holter signal can have thousands of beats, as has been mentioned previously in the introduction. This means that the computational cost of any method using the complete set of beats, \( P = \{p_1, p_2, ..., p_n\} \), can be very high, specially when dynamic programming is involved. Therefore, it is convenient to perform a reduction in the number of beats prior to the clustering process.

In fact, there is a property in this kind of signal that can be used to reduce the quantity of beats to use in the clustering set. Each electrocardiogram has a low intra-signal variability, namely, all the beats belong to a few classes. Besides, it is highly likely that consecutive beats belong to the same class. So, there are many portions of the electrocardiogram where the basic waveform remains constant.

In order to exploit the last property, a preclustering method has been applied to reduce the number of beats to process, but considering no significant beat can be omitted. This is achieved using a very conservative threshold in a preclustering process. To choose the value of this threshold \( \delta \), a short experimental study has been performed to find out the order of magnitude of the minimal dissimilarity obtained when comparing two equivalent beats. This value is the reference threshold.

It is to be noted that the preclustering process presented can take place simultaneously with the acquisition stage, as its computational cost is moderate. Since any preclustering or clustering method needs a distance calculation, but depending on the feature extraction methods, it is not always possible, we will use the term dissimilarity calculation instead, which is more general.

The outline of the preclustering process is as follows. Each time a complete beat is acquired, the dissimilarity measure \( d \) between this beat and \( N \) previously acquired beats placed in a subset \( R \subset P \) is computed, from the closest to the furthest in time.

![Fig. 10. Ways of obtaining the clustering set using the four methods of feature extraction described in the text.](image-url)
If \( d \) is below the threshold, \( d < \delta \), for some beat in \( R \), it means that this kind of beat is already in \( R \), and so it is discarded. Otherwise, the beat is included in the final set \( R \). The initialization is performed by making \( R = \{ p_1 \} \). Non-linear temporal alignment will be used to calculate the dissimilarity.

Therefore, from the complete set of beats \( P = \{ p_1, p_2, ..., p_m \} \), a representative subset \( R = \{ r_1, r_2, ..., r_m \} \) is obtained, where \( m << n \). The clustering computational cost will be significantly reduced. Moreover, due to using a conservative threshold, no significant beat is dismissed.

2.7 Clustering

The sequences in the preclustering set \( R \), must be grouped in a non-supervised manner (clustering), according to the type of beat represented. In order to perform this task, two clustering algorithms have been used: Max-Min and \( k \)-Means, which are well known, and used in many applications [15].

The objective is to find a partition \( C = \{ C_1, C_2, ..., C_k \} \) of the input set of beats, \( R = \{ r_1, r_2, ..., r_m \} \), after preclustering, in such a way that the intracluster variability is minimized, and the intercluster variability is maximized. Each resulting cluster \( C_j \), will contain a set of equivalent beats, according to the dissimilarity measure defined in section 2.4, and only a representative beat from \( C_j \), the so-called centroid, \( q_i \), will be used in the clinical analysis. Thus, the final set of centroids to be examined is given by \( Q = \{ q_1, q_2, ..., q_k \} \), where \( k << m \). Nonetheless, backtracking can be carried out to examine any other beat if necessary.

The first clustering algorithm to be described, the Max-Min, is a non-parametric partitioning clustering algorithm, based on a greedy approach, but with a lower computational cost.

It starts by taking randomly a centroid \( q_1 \) for the first cluster \( C_1 \). From the rest of beats, \( R - \{ q_1 \} \), a second centroid \( q_2 \) is chosen according to:

\[
q_2 = r_j \left| j = \arg \max_{r \in R - Q} d(r_j, q_1) \right. \tag{14}
\]

At this point, the \( k-2 \) remaining centroids must be established. These are found after examining the rest of beats, looking for that one of which the minimal dissimilarity to any centroid in \( Q \) is maximal, namely:

\[
q_i = r_j \left| j = \arg \max_{r \in R - Q} \left( \min_{q \in \bar{Q}} d(r_j, q) \right) \right. \tag{15}
\]

Finally, the rest of beats are included in that cluster with the nearest centroid, using a nearest-neighbour approach:

\[
r_i \in C_j, j = \arg \min_{1 \leq l \leq k, r \in R - Q} d(r_i, q_l) \tag{16}
\]

The second clustering algorithm, \( k \)-Means, is a non-parametric partitional clustering algorithm as well, but based on centroid recalculation. In these case, a complete initialization of the centroids set is needed. This can be accomplished by random initialization, which in some occasions may produce a bad partition if the centroids are too close, or by using another clustering algorithm to get the partition of the first iteration. This last approach is the one chosen in this study, using the Max-Min algorithm.

For iteration \( t = 1 \), the centroids are defined by:

\[
C_1^1 = \{ q_1^1 \}, C_2^1 = \{ q_2^1 \}, ..., C_k^1 = \{ q_k^1 \} \tag{17}
\]

and, the initial partition, is obtained according to the nearest-neighbour, as described in equation (16). The algorithm is aimed at minimizing the criterion function:

\[
J = \sum_{i=1}^{k} \sum_{t \in r} \left| r - q_i \right|^2 \tag{18}
\]

which corresponds to the case where:

\[
q_i = m_j = \frac{1}{n_j} \sum_{r \in C_j} r \tag{19}
\]

namely, the centroid is the mean of the cluster \( C_j \), being \( n_j \), the number of beats in such cluster.

The process is repeated, calculating the parameters at each iteration \( t \):

\[
C_1^t = \{ q_1^t \}, C_2^t = \{ q_2^t \}, ..., C_k^t = \{ q_k^t \} \tag{20}
\]

\[
r_i \in C_j^t, j = \arg \min_{1 \leq l \leq k, r \in R - Q} d(r_i, q_l^t) \tag{21}
\]

\[
q_i^t = m_j^t = \frac{1}{n_j^t} \sum_{r \in C_j^t} r \tag{22}
\]

until the stability condition \( q_i^{t+1} = q_i^t \) for all the clusters, is reached, or the number of iterations is equal to some predefined threshold. In this study, we have fixed the number of iterations in order to reduce the computational cost.

Nevertheless, the fact that \( r_i \) sequences do not have the same length in general, causes the \( k \)-Means algorithm steps to be modified slightly in order to cope with this non-Euclidean space structure, giving rise to the so-called \( k \)-Medians clustering algorithm. This structure has already been taken into account, when we have used the term dissimilarity, which
includes methods such as dynamic time warping, instead of distance.

Besides, in the k-Means algorithm, it is necessary to determine the mean sequence in a cluster to recalculate the centroid, which is not possible in a non-Euclidean space. Therefore, to recalculate the centroid, the median will be used instead:

$$m_i = \arg \min_{r_j \in C_i} \left( \sum_{r \in C_j} d(r_j, r') \right)$$

(23)

The criterion function is now:

$$J = \sum \sum d(x, m_j)$$

(24)

With these changes, the clustering algorithm k-means turns into the k-Medians algorithm [16].

3. Results

In order to find the best combination from the different possibilities proposed, a thorough experimental study has been carried out, using different temporal alignment, feature extraction methods, and clustering algorithms. The parameters used in the experiments are shown in Table I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registers</td>
<td>27</td>
</tr>
<tr>
<td>Beats</td>
<td>27412</td>
</tr>
<tr>
<td>Beats after preclustering</td>
<td>1026</td>
</tr>
<tr>
<td>Preclustering errors</td>
<td>0</td>
</tr>
<tr>
<td>Clusters</td>
<td>2 to 12</td>
</tr>
</tbody>
</table>

Table I. Parameters of the experiments.

In sections 3.1, 3.2, and 3.3, the graphs of the experiments will be shown. Each percentage has been calculated from the quantity of misplaced beats relative to the number of beats not in the majority class. This criterion gives rise to a 100% error rate when only one cluster is defined, and consequently all the beats are assigned to it.

Although 27 different registers in the experiments have been utilized, the results are calculated as if they belonged to the same register. This can be considered as a difficult test, since in a real register, it is very unlikely that there will be such great variability.

3.1 Temporal alignment comparative results

In this case the results will show which kind of temporal alignment, linear or non-linear, offers the best performance according to the error rate.

The first experiment consists of using sample amplitudes as feature extraction method. In Fig. 11 the clustering algorithm is the Max-Min and in Fig. 12, it is k-Medians.

Fig. 11. Experimental results using amplitude samples, Max-Min clustering algorithm, and both linear (L) and non-linear (NL) temporal alignment. The suffix C stands for critical error, while its absence corresponds to clustering error. Critical error accounts for the clustering error when no cluster is occupied preponderantly by this kind of beats.

Fig. 12. Experimental results using amplitude samples, k-Medians clustering algorithm, and both linear (L) and non-linear (NL) temporal alignment.

Fig. 13. Experimental results using Wavelet coefficients, Max-Min clustering algorithm, and both linear (L) and non-linear (NL) temporal alignment.
The third experiment consists of using trace segmentation as feature extraction method. In Fig. 15 the clustering algorithm is the Max-Min and in Fig. 16, it is \(k\)-Medians.

Polygonal approximation has been used only with non-linear temporal alignment, and so, the experiments using this feature extraction are not included in this section.

According to the results of the experiments, related to time normalization, and presented in this section, non-linear temporal alignment using dynamic programming produces better results than linear temporal alignment. The differences are not very large, probably due to the intrinsic low variability of each register and, consequently, the low percentage of abnormal beats present within the signal.

These lower than expected differences are also related to the preclustering process, as will be justified in Section 4.

### 3.2 Feature extraction comparative results

Once the time normalization methods have been tested, another aspect to take into account is the effect of the four feature extraction methods, which is shown in the following experiments.

Taking the clustering algorithm Max-Min, in Fig.17, the results obtained using the four feature extraction methods are shown.

If the clustering algorithm is the \(k\)-Medians, the results obtained in this case are shown in Fig.18.

From these results, trace segmentation is the feature extraction technique that offers the best performance, although the differences among the methods tested are not very significant. Besides, trace segmentation reduces the samples to 20% of the original length, making the clustering faster, and the memory requirements lower.
3.3 Clustering algorithm comparative results

Finally, the effect of the two clustering algorithms is tested. In Fig. 19-22, using non-linear temporal alignment, and the feature extraction methods, the differences between both clustering algorithms are shown.

Fig. 19. Comparison using amplitude samples as feature extraction, utilizing \( k \)-Medians and Max-Min clustering algorithms.

Fig. 20. Comparison using trace segmentation as feature extraction, utilizing \( k \)-Medians and Max-Min clustering algorithms.

Fig. 21. Comparison using polygonal approximation as feature extraction, utilizing \( k \)-Medians and Max-Min clustering algorithms.

Fig. 22. Comparison using Wavelet transform as feature extraction, utilizing \( k \)-Medians and Max-Min clustering algorithms.

From the previous set of experiments, it can be observed that \( k \)-Medians produces better results than the Max-Min clustering algorithm, although its computational cost is higher.

4. Discussion

First of all, according to the kind of temporal alignment, the results show how the difference in using linear or non-linear temporal alignment is not very significant in some cases. Nevertheless, there is a consideration to take into account: the dissimilarity measure \( d \) used in the preclustering process is indeed based on non-linear DTW alignment, and so, a part of the non-linear variability has already been dealt with.

In order to analyze this preclustering effect, additional experiments have been carried out. For those tests where the differences between linear and non-linear temporal alignment are minimal, the same experiments have been repeated with and without preclustering. In Fig. 23 and 24, two of these results, obtained using trace segmentation, are shown. From these results, it is clear that the preclustering reduced the differences between both techniques of temporal alignment, as expected.

Fig. 23. Results obtained using trace segmentation without preclustering.
Fig. 24. Results obtained using trace segmentation with preclustering. It can be seen that the differences between both curves is smaller than those in Fig. 23.

The results obtained for different feature extraction methods show that the best technique seems to be trace segmentation, which is also the procedure with the lowest computational cost. Wavelet transform and polygonal approximation, although reducing the number of samples, do not improve the error values.

Considering the clustering algorithms, the best results are obtained with the k-Medians algorithm, although the computational cost is higher since the centroid must be recalculated after each iteration.

In general, the critical error is around 1% in the best case. This is a low value for an error measure in many applications but, in this case, it is very important to be at a level as close to zero as possible, since the omission of a single beat might be the difference between an accurate or a wrong diagnosis. However, after a close examination of the critical errors, many of them are due to signal noise or labeling errors. The clustering error reaches a value around 7% in the best case.

Differences between methods are not very large. This is because of the low variability of single electrocardiograph registers, but however small the differences are, it must be taken into account that the misplacing of just a single beat, might cause a diagnostic error.

The computational cost is high when dealing with complete Holter signals, but many of the stages can be implemented in the portable acquisition system. Hence, some calculations will be performed in parallel, and the associated computational cost will, therefore, not translated into longer response times. It is important to note that the primary goal is to find a reliable method to extract accurately significant beats from a register and, in second place, to reduce its computational cost, which is currently under study.

5. Conclusion

In this paper, a method to automatically group beats from a Holter signal is presented, involving the stages of beat segmentation, normalization, feature extraction, preclustering, and clustering.

A complete set of experiments has been performed in order to find the best combination to obtain a set of beat clusters as uniform as possible.

From the experimental results, the method finally proposed consists of non-linear temporal alignment, trace segmentation as feature extraction, and k-Medians as clustering algorithm. A preclustering method has also been used to reduce the computational cost. The effect of such preclustering is to reduce the differences between linear and non-linear temporal alignment, which might permit using the latter in the clustering process. In Fig. 25, this method is shown schematically.

Fig. 25. Method proposed for implementation in a Holter system.

With this method, doctors will only have to examine a member of each cluster to formulate the diagnosis. However, since the critical error value could be too high in some cases, further refinements are needed in order to improve the method.

Those cases where dynamic programming yields a dissimilarity measure not representative of the differences between the beats should be considered. Another method to calculate the dissimilarity using Hidden Markov Models (HMM) is also being studied.

The labeling procedure should also be reviewed since some critical errors can be due to inappropriate beat labels.
In order to find out the suitable number of clusters for each case, a detailed study over real labeled Holter signals should also be carried out. The goal would be to establish a relation between the number of clusters used, and the possibility of getting a null critical error.

Finally, the method proposed or some of its stages, could also be applied to other biological signals, or semiperiodic registers in general.

Acknowledgements

We would like to thank the cardiology department of the Alcoy Hospital for their cooperation and technical advice.

This research has been supported by grant number 20010946 of the Polytechnic University of Valencia, for the development of a complete Holter system based on pattern recognition techniques.

6. References