Time series data imputation - Comparison of Dynamic Time Warping with Needleman-Wunsch algorithm

Guilherme Moura
Instituto Superior Técnico
Universidade de Lisboa, Portugal
Email: guilherme.reis.moura@hotmail.com
November 2019

***Abstract*—Machine learning algorithms are now being designed and applied to data to help humans in their everyday needs. These algorithms can bring major benefits to many areas and are capable of conducting predictions, clustering and classification on data. They could be used, for example, on medical databases to help in treatments and diagnosis of patients. One major concern that threatens the efficiency of these algorithms are missing values. Many algorithms which are in place today are not prepared to handle these missing values, which means they have to be handled in other ways. In this paper it is aimed to compare two imputation algorithms that could be used in filling these missing values. Both methods use sequence alignment to find matches with which the missing values could then be imputed. One of the algorithms uses dynamic time warping while the other uses Needleman-Wunsch. Both of these algorithms were suitable when it came to data imputation. Imputation done using dynamic time warping was accurate, although it lacked in speed, while the Needleman-Wunsch imputation was faster, but not quite as accurate as the dynamic time warping imputation. The results show that both of these algorithms should be further tested due to their potential in the imputation of values, as well as some suggestions to strengthen the weaknesses of both of these algorithms.**

I. Introduction

Nowadays, there is an increased necessity in handling large volumes of data, particularly when it comes to medical data. This data can be used to extrapolate useful information and, as a consequence, aiding medical professionals [1]. Many algorithms were designed in order to extract useful patterns from medical databases, which would be impossible for a medical team to analyze due to the immensity of the data. This process is called data mining [2, 3] and one of the fields that is used to perform these operations is machine learning [4]. This field is dedicated to the classification, prediction and clustering of data, among others [5]. All of these can be applied to medical databases in various situations requiring data handling, depending on the circumstances. Besides medical databases many other fields benefit from the use of these techniques in many ways [6].

Regarding the usefulness of these algorithms in the medical fields there is a number of ways on which they excel:

- **Diagnosis** - Many classification algorithms have been developed to provide with an answer with haste [7]. By applying these, many lives could be saved, not only because of the speed on which a diagnosis could be achieved, but also on the accuracy, preventing a wrong diagnosis. Of course these can be merely suggestive, being regarded as a second opinion to the medical professionals. Another important tool for diagnosis is image recognition which can lead to a quick diagnostic by analyzing image-related diagnostic tools.On the field of prediction, many options are being explored, one of which is the capability to predict the appearance of a certain disease or an injury, based on data from the patient. Thus, diagnosis is possibly the one with the most direct impact regarding the health of a patient, since these will define the treatment applied.
- **Treatment** - There have been many changes when it comes to treating patients as well, such as making use of patient data, so that professionals can choose a specific treatment for an individual with specific and similar features [8]. Research and drug manufacturing is also being affected with these algorithms, since they are starting to be produced not only based on past results, but also base on predictive data.

As time passes, a lot of applications using these machine learning techniques have encountered some issues regarding data quality. Given the fact that the amount of data collected is enormous (and regarding the data itself, the amount of variables/features captured is also very big), the probability that the collected data is one hundred percent accurate is very low [9]. This effect is even more critical when it comes to medical data, given the sensibility the algorithms have to have with these type of data. Mistakes are bound to happen, whether it is by the hand of man, which is considered to be very common, whether it is done by machine errors. These mistakes can range from non-existing values to simply absurd values that were found because someone misplaced a simple comma. The problem that is faced nowadays is that with the amount of data in existence it is impractical to identify and correct all mistakes by hand. Thus, methods to identify these situations and correctly handle them are needed, or else, whenever these kinds of situations occur, the data would need to be disposed

of, losing valuable information.

Due to the fact that many of the machine learning algorithms require their data intact, a lot of effort is being put into creating accurate imputation algorithms, capable of filling out data with values that could represent the missing value, while still maintaining data coherence [10, 11]. The purpose of this work is to find a suitable way of dealing with missing values in time series data, particularly in medical databases. To do so, a method of imputation will be proposed that will focus on a single feature, which makes this algorithm usable with multivariate and univariate time series, by using an approach based on the similarity measurement of time series using the Dynamic Time Warping (DTW) algorithm [12]. A comparative analysis with the Needleman-Wunsch (NW) algorithm [13, 14] will also be made, due to the similarity in behaviour of both of these algorithms.

II. Background

Nowadays, many steps are done to validate and perfect the data that will be used by many machine learning algorithms to solve many types of issues. One of these important preprocessing steps is to deal with missing data, which can be done in a multitude of ways [15, 16]. The easiest way would be to completely ignore the missing data. However, by doing so, the information regarding this value would be completely lost. The other way of dealing with this problem is by attempting to fill in the missing values. To do so, many methods can be considered, mainly statistical and machine learning methods [17], which use a predictive model based on the available data to infer the missing values. Some of these methods include: k-nearest neighbours [18, 19] and Bayesian approaches [20]. Other simple approaches can be used to tackle this issue, as for example using the mean of the non-missing values to fill in the missing values.

A. Time series analysis

This work will focus on the imputation of missing values in time series, which have a component that differentiate them from the rest of the data - time. Data is to be acquired over time and indexed in accordance to the passage of time, usually with equal time intervals, so that it forms a sequence of time-stamped data. This kind of data is used on many applications nowadays as, for example, in the stock market or medical records, amongst many other fields [21, 22] . Data mining applied to time series share the same concerns as if it were applied to the generality of data: classification, clustering, and forecasting. This last one refers to forecasting future values in time series and is a highly explored field.

A time series is a sequence of data points indexed in time $t \in \{1, \ldots, N\}$ with the result $X = (x_1, x_2, \ldots, x_N)$. Thus, a forecast of a time series, at a further time t, would be x_{N+t}.

Whichever is the goal of the data mining process, there is another procedure that can help filling a time series data. It is called trend analysis [23] and mainly consists of 4 components: trend, seasonal, cyclic and random variations.

- Trend variations refer to the general direction taken over a time interval as, for example, the trend to rise at a certain moment.
- Seasonal variations represent a certain event that re-occurs within a time interval, with a time interval associated with a calendar period, such as monthly or daily routines.
- Cyclic variations are very similar to seasonal variations, being the main difference the interval of time associated - these events use to have a duration longer then. a year
- Random variations, as the name implies, is associated with random events that occur, possibly modifying the previously discussed components.

The existence of these components in time series is very important, since they will enable the use of many algorithms that can analyze the patterns made by them. This is especially important when it comes to matching or aligning time series with one another. Medical databases will benefit greatly from these components because of the big sample of patients they hold, which in turn will increase the diversity of behaviour of many features, making it easier to find connections between the features of some patients.

Time series may need to be handled differently depending on the number of variables, especially when it comes to forecast an event. When a time series is only represented by a single variable, the forecast can only depend on the past and present values. These are called univariate time series. As opposed to this, the multivariate time series are composed by two or more features that are registered along time [24]. An example of this type of time series are medical records, which can hold many variables taken from the variety of exams patients undergo.

B. Dynamic time warping - DTW

The DTW algorithm consists in the alignment of two data series by trying to explain variability in the Y-axis with variability in the X-axis [12]. An example of this can be seen in Figure 1.

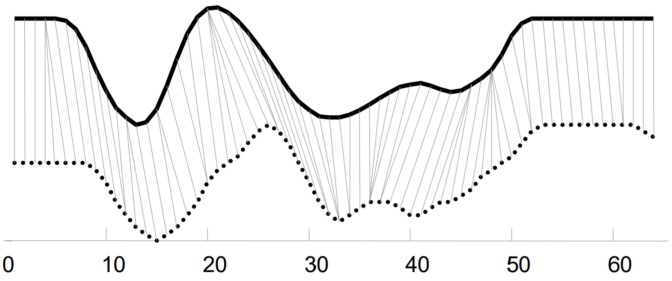


Fig. 1: Example of the dynamic time warping algorithm [25].

This algorithm takes two time series (Equations (1) and (2)), not necessarily with the same size, and builds an m-by-n matrix in order to align both of these sequences. Each of the elements of this matrix will correspond to a distance $\delta(a_i, b_j)$ between two points a_i and b_j, for $i \in \{1, ..., n\}$

and $j \in \{1, ..., m\}$. The goal of this matrix is to help to determine a path in which the distance between both sequences is minimized. This is called a warping path and an example of one is given in Figure 2, and it would be represented as in Equation (3).

$$A = a_1, a_2, ..., a_i, ..., a_n \tag{1}$$

$$B = b_1, b_2, ..., b_j, ..., b_m \tag{2}$$

$$W = w_1, w_2, ..., w_k, ..., w_t \tag{3}$$
$$\max(n, m) \leq t < n + m - 1$$

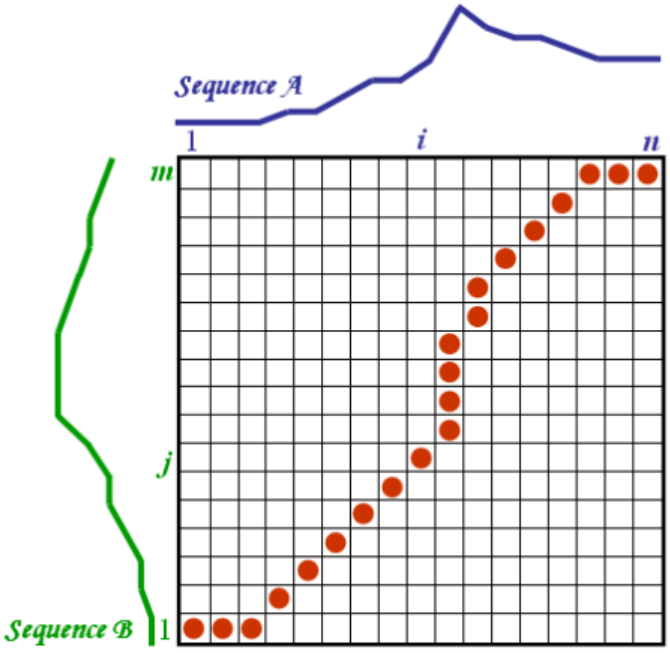


Fig. 2: The alignment of two sequences (Equations (1) and (2)) in a matrix, and the respective warping path (Equation (3)) [26].

The warping path elements w_k represent the alignment of two points of the sequences to be aligned $(i, j)_k$. A warping path must obey certain conditions [12]:

- **Continuity** - There is a limit in the steps that can be given: If $w_k = (i, j)$ and $w_{k-1} = (i', j')$ then we must have $i - i' \leq 1$ and $j - j' \leq 1$. In short terms this means the warping path must only progress in the matrix through adjacent cells (horizontally or diagonally).
- **Monotonicity** - The points of the warping path must be monotonically ordered in time such that for $w_k = (i, j)$ and $w_{k-1} = (i', j')$ we have $i \geq i'$ and $j \geq j'$
- **Boundary Conditions** - The first point of the warping path and the last should refer to the respective first and last points of the sequences to be matched: $w_1 = (1, 1)$ and $w_t = (n, m)$. Although this is one of the conditions, sometimes there are exceptions that are introduced by giving offsets that could be used to initiate/terminate the warping path.

Many paths can be determined using these conditions, however the goal is to minimize the warping path as much as possible, as presented in Equation (4).

$$DTW(A, B) = \min \left(\frac{\sqrt{\sum_{k=1}^{t} w_k}}{Z} \right), \tag{4}$$

Z is a coefficient used to compensate for the difference in size of the warping paths. One of the possibilities for this value is the size of the found warping path $Z = t$. Having w_k as the distance between elements of the time series $w_k = \delta(i, j)$, two commonly used distance measures are presented in Equations (5) and (6).

$$\delta(i, j) = |a_i - b_j| \tag{5}$$

$$\delta(i, j) = (a_i - b_j)^2 \tag{6}$$

The path can then be found by applying dynamic programming to calculate the cumulative distance $\gamma(i, j)$ for each point, which will be the distance for the current points $\delta(i, j)$ added to the minimum of the cumulative distances of the adjacent elements in the matrix (both horizontally and diagonally), as shown in Equation (7).

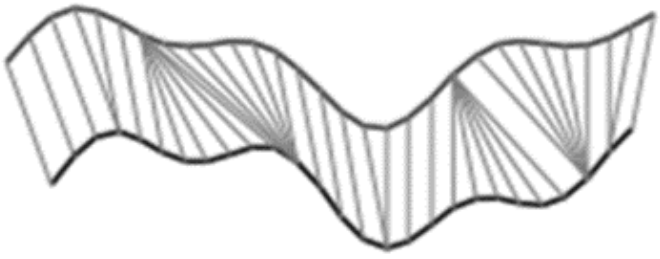
$$\begin{aligned} \gamma(i, j) =& \delta(a_i, b_j) + \\ & min[\gamma(i-1, j), \gamma(i, j-1), \gamma(i-1, j-1)] \end{aligned} \tag{7}$$

When the algorithm is completed, the optimal warping path will be found by tracing backwards through the minimum found values. When the warping path is found a score will be attributed to the match, which will reflect upon the distance of the full warping path to both sequences (in this case reflecting how good the fit was).

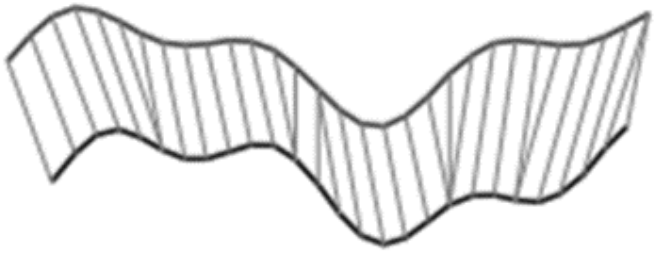
C. Derivative Dynamic Time Series - DDTW

Although DTW is able to align many sequences with great proximity, excelling particularly when it comes to variations in the X-axis (variations in time), there are many sequences in which it will encounter certain issues. One of the problems it will encounter is when it comes to variations in the Y-axis. With the presence of local features, such as peaks and valleys, the alignment will be affected. The DTW algorithm will attempt to justify these local features by making corrections in the X-axis, thus producing singularities which will affect the alignment in the places where these features are located. The issue falls on the alignment being made depending solely on the distance of the sequence points, and not considering the waveform/shape of the sequence. To mitigate this problem, DDTW was created [25]. This algorithm would be using the derivative of the sequences, which will have more significant

results in terms of aligning the waves using their shape. In Figure 3 it is visible the differences when aligning the same sequence with DTW and DDTW.



(a) DTW alignment



(b) DDTW alignment

Fig. 3: The alignment of two sequences by using DTW (a) and by using DDTW (b) [25].

The algorithm will behave in the same way as before, however the alignment will happen on the derivative of the sequences. Usually, for simplicity, the derivative is calculated based on the slopes of the point in question with the point on the left and the one on the right. Thus, the derivative, $D[a_i]$ at any given point, a_i, of the sequence will be as presented in Equation (8).

$$D[a_i] = \frac{(a_i - a_{i-1}) + \frac{(a_{i+1} - a_{i-1})}{2}}{2} \tag{8}$$

By having the derivative calculated as shown in Equation (8), there could be loss of information when it comes to the first and last points of the sequence. Another problem is the exact issue trying to be solved: missing values. Missing values will not only have a direct impact in the point where they are missing, but also in their surroundings, specifically to their immediate right and left. This can have a big impact particularly when it comes to short sequences.

D. Dynamic time warping-based imputation - DTWBI

Having both of the previous sections in mind we can now look at Dynamic time warping-based imputation or DTWBI for short [27]. This method was developed with imputation of gaps (a continuous group of missing values) as a goal. This method was originally applied in univariate time series, and would compare the sequence with the missing gap with other time series of the same dataset, trying to find the most similar sequence to the one with the missing value.

To perform this imputation, the first step would be to extract a sub-sequence, before or after the gap. Then, the DDTW algorithm would be applied in other time series, locally, as a sliding window, to find the best match possible. The final step would be to copy the values associated with the best match, which was found in the previous step, at the relative position of the gap and the sub-sequence used to find the best match. For instance, if there was a gap from positions a_{20} to a_{25} and the sub-sequence a_{14} to a_{19} was used to find a match, then if the best match were to be found (on a different time series) at positions b_7 to b_{12} then the values to be used for the imputation would be from b_{13} to b_{18}. An example of this algorithm can be seen in Figure 4.

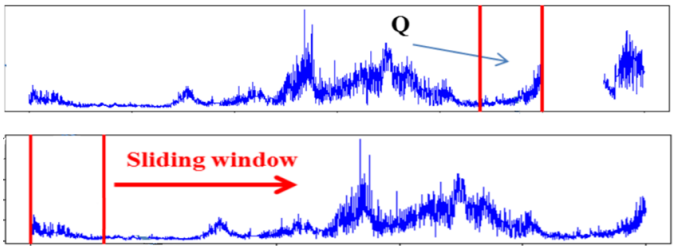


Fig. 4: An example of the DTWBI algorithm using a sliding window to find the optimal match. [27].

E. Needleman-Wunsch algorithm - NW

This algorithm was chosen due to the similarity in terms of behaviour, comparing to DTW algorithm, in the sense that both perform sequence alignment. The NW algorithm [14] was mainly created with the purpose of aligning discrete sequences in the field of bioinformatics such as protein or nucleotide sequences (related with DNA).

It will also behave in a similar way to the DTW algorithm, with the construction of a matrix using both sequences. The difference will be that, in DTW, the distance between data points is used to build the alignment matrix. In NW, a similarity matrix will be used to build the alignment matrix, with given scores between two data points a_i and b_j, for $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$.This similarity matrix must be consulted to define the scores of each cell of the matrix. The similarity matrix can be custom made or it can be built based on the scores attributed to matches, mismatches and gaps. Typically the score for matches is 1, while for mismatches is 0 (this is considered a penalty). Besides these factors there are also two types of gap penalties. These can be gap openings, which refer to when a gap must be introduced to a sequence, and the gap extension penalty which is used when a gap (which was already open) extends. When using a custom similarity matrix certain matches of characters could be assigned a greater score, thus favoring the matching/mismatching with certain characters.

When the matrix is filled with all the values, the traceback phase will occur, in similarity to what happens in DTW. Then, it will generate the alignment by finding the path, starting at the bottom right cell of the matrix and ending at the top left

cell. To find this path the movements from one cell to another can be made upwards, diagonally or to the left. When a move is made diagonally it will correspond to a match/mismatch, and when it moves either to the up or left a gap is to be introduced. In Figure 5 a representation of the Needleman-Wunsch algorithm is shown. On the left the scoring matrix is represented along with the traceback (path in orange), which was made by following the blue lines which were added during the building of the matrix, while on the right side we have the similarity matrix with the scores for the matches and mismatches.

An alternative to this method could be the Smith-Waterman algorithm [28], which is a variation of the NW algorithm. The main difference between these algorithms is that, while the NW algorithm operates globally on a sequence, the Smith-Waterman will act using a local approach. In the traceback phase, instead of aligning both sequences globally it will analyze the best score in the matrix and build the best local path it can find. This means that certain parts of both sequences may not belong to the alignment if they negatively impact the best score.

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

	A	G	C	T	U
A	1	-1	-1	-1	-1
G	-1	1	-1	-1	-1
C	-1	-1	1	-1	-1
T	-1	-1	-1	1	-1
U	-1	-1	-1	-1	1

Fig. 5: Example of the result of the Needleman-Wunsch algorithm applied to align two DNA sequences with a gap value of -1.

III. Proposed Method

This work will consider two main approaches to data imputation, focusing on time series, which will be explained in this section. These methods were chosen due to the similarity in methodology, and will both be compared in a later section.

A. DTW imputation

The first method will be based on DTW to perform imputation of missing values. The algorithm will resemble the DTWBI algorithm explained in the previous chapter, although it will suffer some adjustments.

Firstly, the number of points of the window that will be used to find a match will be chosen, and this number will also have impact in the amount of windows to be analyzed. For instance, by analyzing a window of size 5 we will have a maximum of 6 different windows to analyze placed around the missing value. An example of this is represented in Figure 6. The size of the window should be influenced by the size of the time series analyzed, as well as the amount of missing values in the series. If the missing value is placed in the beginning or ending of a time series the amount of windows to be used will be reduced due to the non-existence of points to analyze surrounding the missing value. If there are other missing values surrounding the analyzed missing value, the amount of windows will also be adapted to whichever number of windows that are able to utilize in the algorithm.

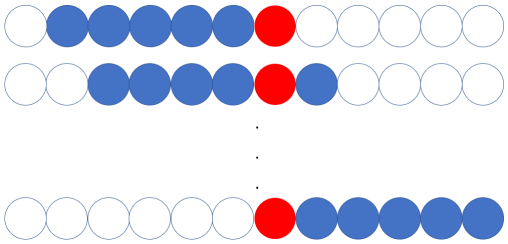


Fig. 6: Example of the result of the windows to be compared with DTW. Blue circles represent the window to be analyzed, red circles represent the missing value, and the white circles are the remaining data points.

After the windows which are to be used for the comparison are chosen, they will be compared against other samples of time series. The objective is to find (regarding the same feature) the most similar sub-sequence in all of the data available against the chosen windows. Needless to say that the more data to be compared, the better due to increased diversity of samples that this could bring. The algorithm must of course not be applied in the presence of missing values in the other data for the alignments to be accurate. Additionally, the position from which the value to fill for the missing value will be retrieved must not contain a missing value.

In this algorithm, the DDTW will be applied, and as such the derivative of the time series must be calculated. One of the problems which was already described would be the loss of information regarding the initial and final points of the time series, as well as loss of information on the points surrounding missing values. To counter this, the derivative was calculated by using two points whenever it was necessary. This was done because, in order for this work to be applied in short time series, the amount of information that would be lost because of the derivative using three points would be impactful in the efficiency of the algorithm.

Regarding the imputation of the actual value, instead of copying the exact values as done in the DTWBI algorithm, the value will be calculated by using (8). The objective will be to calculate a_i, having $D[a_i]$ as the value found (in the same place of the missing value when compared to the window to be searched), and the values of a_{i+1} and a_{i-1} as the values surrounding the missing value to be imputed.

B. Needleman-Wunsch Imputation

The other method to be analyzed will rely on the Needleman-Wunsch algorithm. In order to use this method on the imputation of continuous time series, a pre-processing step

of discretization must be taken. The dataDiscretize function from the R package bnspatial was used. The discretization was made with equal sized classes, being the number of classes to be used based on the number of data points available in the sequences. Another important factor is that the discretization for each individual is made independently, which will be beneficial in terms of creating matches between sequences with similar shapes instead of only considering the values of the data points.

To perform this algorithm on a sequence with missing values, the missing values will be assigned a special character: '?'. Also, when building the similarity matrix, we must consider that the objective of this algorithm is to match the missing values ('?') with another character (a suitable one) in order to discover its value. To do so, the similarity matrix must be built with such objective in mind, which means the matrix should consider the following:

- All matches between characters (except for the matches between '?') will have a score of 1.
- All mismatches between characters (except for the ones involving '?') will have the score of -1.
- The match between '?' characters is unwanted, because it would mean that two missing values are being aligned, which is not the objective of the algorithm. As such, the value assigned to the match of '?' will be the same as a mismatch (-1).
- Mismatches with '?' must be considered beneficial in order to find the values used to replace the missing values. As such, this value must be higher than the normal mismatch value, being the chosen value -0.5.
- The gap introduction penalty was set to -0.8 in order to prioritize finding a mismatch for the missing value instead of opening a gap.
- The gap extension penalty was set to -0.3, so that if a gap opened beforehand a missing value is not mismatched. This means that the missing values will not have a correspondence to a data point inside a gap.

Having the matrix built in such a fashion the algorithm will behave in the usual way, and the best possible match will be the one used to retrieve the missing values. However, there is a possibility that the missing values could align with a gap, which would not be useful to this algorithm. When this occurs the algorithm will have to run again with a feature enabled which will not allow the algorithm to align missing values at the first missing value. It will run as many time as necessary to fill all the missing values.

IV. Results

In this section the results obtained with the previously described methods will be presented. Besides this, some information on the used data will be given as well as a comparison on the performance of both algorithms.

A. Datasets

In this work, three datasets were used: two synthetic datasets, ECG and CMUsubject16 [29] and a real dataset, Epileptic Seizure Recognition [30]. This last dataset is a recording of brain activity in several patients under different situations with the main goal of studying people suffering of epileptic seizures, which is helpful when trying to perform the imputation, due to the variety of data, even though this dataset holds only one feature. This dataset is also the one with the highest range, with values from -1885 to 2047. The ECG dataset contains data on 200 individuals with only two features. Although the time series are short (39 datapoints), there is a wide range of values in the sequences (going from -438 to 430). CMUSubject16 is a database with few individuals but with many features (62). The data has a range from -137.54 to 437.35, although the range is shorter on each characteristic. A brief overview on the features of these datasets is presented in Table I.

Dataset	Individuals	Length	Features
CMUsubject16	58	127	62
ECG	200	39	2
Seizures	11500	178	1

TABLE I: Overview of the used datasets.

B. Single missing values imputation

Firstly, tests were made with a single missing value (picked at random) in a certain individual and feature (also picked at random). To test this, a missing value was inserted in each of the datasets, and once found these would be compared against the original values. In order to be able to compare both procedures, the missing values used for the DTW imputation will be the same used in NW imputation.

As for evaluations measures, the Root Mean Squared Error (RMSE) will be used as a way of measuring precision and accuracy by calculating the deviation between the real values, x and the prediction made x'.

The prediction accuracy (PA) indicator will also be used which is a performance indicator that can be used to evaluate whether the imputation is done correctly or not. It can take values that range from 0 to 1, being a value closer to 1 a better fit than if it were closer to 0.

Finally, the last evaluation measure discussed will be the coefficient of determination (R^2), which is also used to evaluate imputation processes. In similarity with the previous performance indicator, it can take values that range between 0 and 1, being 1, once again, a better fit than 0. This coefficient can be used to assess the variability in the imputed data in relation to the actual values.

The following subsections will discuss the tests made in both approaches.

1) DTW Imputation: The results of the single missing values imputation using DTW is shown on Table II.

By analyzing Table II it is shown that the algorithm worked well overall on all the datasets. The ECG dataset was the one with which this method was least effective, and this could be due to the high range of the data (in the case of ECG data ranges from -438 to 430), as well as the amount of data available to compare in order to search for the missing values. Even

Dataset	RMSE	PA	R^2
CMUsubject16	0.307	0.999	0.766
ECG	13.72	0.993	0.756
Seizures	3.491	0.999	0.875

TABLE II: Evaluation measures of the single imputation using DTW.

so, the obtained result for this dataset was mostly accurate. The seizures dataset also had good results even though the RMSE is higher when compared to the CMUsubject16 dataset. This could be once again caused by the range of the data (which in the case of the seizure dataset ranges from -1885 to 2047). Regardless, considering this dataset contained real data, the algorithm could impute data with good precision. In Figure 7 an example of an alignment can be visualized. The sequence in blue is the one with the missing value, represented by the cross, and the sequence in orange is the best match found in the dataset. This work used a window size of 5 in the algorithm, and because it analyses the surroundings of the missing value (with the window size given) the charts will have the missing value position in the middle, and will also have the 5 prior and following datapoints. Take in consideration also that any of the windows could have been used to find the best match, and that this match is based on the shape of the waves instead of the values.

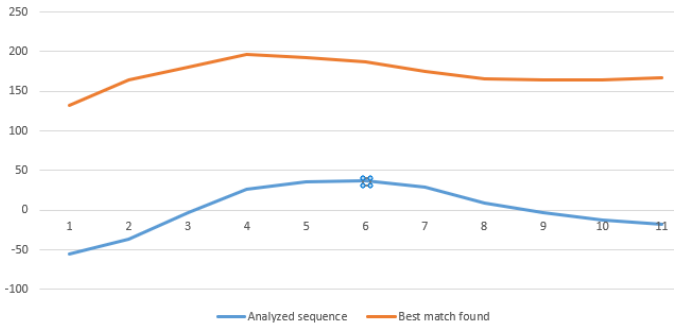


Fig. 7: Example of the result of the DTW algorithm applied to align two sequences of the Seizures dataset.

2) NW Imputation: The results of the single missing values imputation using NW are shown on Table III.

Dataset	RMSE	PA	R^2
CMUsubject16	0.523	0.999	0.765
ECG	39.02	0.946	0.685
Seizures	12.99	0.997	0.872

TABLE III: Evaluation measures of the single imputation using NW.

The NW imputation algorithm shown in Table III shows that for the CMUsubject16 dataset the results were quite good. Regarding ECG and Seizures datasets the results were not as good. Once again, the ECG dataset suffered the worst imputation, which could be associated with the amount of datapoints for each individual. The size of the sequences directly affects the discretization, as was mentioned in the earlier chapter, because the amount of classes created depends directly on the size of the sequences. Because this particular dataset has short sequences and a big range, the boundaries of each class will have a big range, which will make this method more inaccurate (because the values will be imputed with the intermediate value of the range of the class assigned to the missing value).

As for the Seizures dataset, the results could be explained by the same reason. Because this dataset has bigger sequences, the effects of the discretization were not as noticeable as on the ECG dataset, nonetheless the effects of the range of the data of the Seizures dataset and the size of the sequences also affected the performance of the algorithm. The CMUsubject16 dataset had the best imputation out of the three, which can be explained by the size of the sequences (having a size that facilitates a good amount of classes) and the range of the values, which in this case is smaller than the other two datasets (ranges from -137.54 to 437.35, although the ranges for each feature it has are much smaller). Although the results for the NW were the best for this dataset, when examining the alignment of the discrete values, we could observe that some of the matches were not correct, but due to the proximity in class (and having each class a low range of values), the imputation was not as affected by the wrongful alignment of the sequences, as the other two datasets. In Figure 8 an example of the discretized alignment is given. The missing value is represented by a "?".

Analyzed sequence	Best match found
d	d
c	c
b	b
b	b
b	b
c	c
c	c
d	d
?	e
-	e
-	f
g	g
g	g

Fig. 8: Example of the result of the NW algorithm applied to align two sequences of the Seizures dataset.

3) DTW vs NW: Comparing both of these algorithms it can be said that DTW is the most accurate one, although on large sequences with low range of data the results are quite close. Regarding speed, the NW imputation is faster than DTW imputation. It was more noticeable on the Seizure dataset, due to the amount of individuals it had. The NW algorithm also had an issue due to the alignments of missing values with gaps. This was surpassed when forcing the algorithm on finding the best possible alignment which would not align the missing value with gaps.

C. Multiple missing values imputation

On the second part of this work , the imputation of data when multiple missing values are present in a sequence will be analyzed. This will be done by following the same steps as before, although instead of removing only one datapoint at random, an individual and feature will be chosen at random, and from the elected sequence we will have 5%, 10%, 15% and 20% of missing values in randomly picked positions. This will be done in four different runs for each missing percentage, with the exception of the Seizure dataset which will be done in a single run, due to the high computation time of the DTW imputation on this dataset. For effects of comparison the same positions will be applied on DTW and NW imputation.

1) DTW imputation: The results of the DTW imputation with multiple missing values are shown on Table IV.

Dataset	Missing %	RMSE	PA	R^2
CMUsubject16	5%	0.221	0.999	0.934
	10%	1.802	0.997	0.965
	15%	2.08	0.997	0.973
	20%	1.8	0.997	0.978
ECG	5%	24.01	0.979	0.734
	10%	36.24	0.934	0.766
	15%	45.82	0.965	0.856
	20%	42.05	0.948	0.844
Seizures	5%	13.4	0.998	0.787
	10%	53.44	0.997	0.888
	15%	29.66	0.907	0.764
	20%	25.98	0.894	0.757

TABLE IV: Evaluation measures of the multiple imputation using DTW.

The CMUsubject16 had the expected behaviour with the increase in the RMSE, even though for the 20% missing values the RMSE value decreased. Overall the values were in accordance to what was seen in the single imputation for this dataset, having a good accuracy and low errors, even in the presence of many missing values. Regarding the ECG dataset, there were also no surprises, because in similarily to what happened with the single imputation, this was the dataset which had the worst performance. Both of these datasets had a relatively fast execution time, which was not the case for the Seizures dataset, which required a lot of processing time to handle the imputations. Due to these long execution times, only one execution of the algorithm was done, causing the values on Table IV to be not as compliant to the expected when compared to the ECG and CMUsubject16 datasets. Although it was to be expected for the RMSE to be the lowest for the 20% missing values, this was not verified, with the highest RMSE on the 10% missing values. In terms of the prediction accuracy, the values were particularly good in the CMUsubject16 dataset. Both the Seizures and ECG dataset also had good results, although the Seizures dataset struggled with a higher missing percentage. An example of the imputation with 20% missing values is shown on Figure 9.

2) NW imputation: The results of the NW imputation with multiple missing values are shown on Table V.

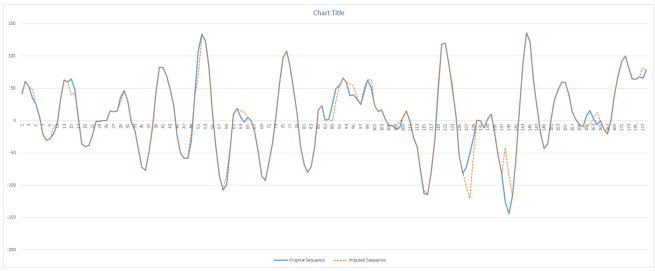


Fig. 9: Alignment of the sequence using DTW with 20% missing values for the Seizures dataset.

Dataset	Missing %	RMSE	PA	R^2
CMUsubject16	5%	1.873	0.998	0.931
	10%	3.784	0.991	0.952
	15%	3.622	0.992	0.963
	20%	3.088	0.994	0.973
ECG	5%	39.8	0.942	0.78
	10%	40.54	0.889	0.727
	15%	45.5	0.925	0.809
	20%	57.81	0.862	0.713
Seizures	5%	39.76	0.998	0.787
	10%	289.2	0.855	0.652
	15%	36.17	0.891	0.627
	20%	55.19	0.502	0.238

TABLE V: Evaluation measures of the multiple imputation using NW.

During the tests, in order to impute all missing values in a sequence, the algorithm had to run multiple times, due to the alignment occurring globally, which means that sometimes a certain amount of missing values would align with gaps. In the following runs the already imputed values would be used and a restriction to find the best matching sequence which could have a match for any missing value would be implemented in order to fill the missing value.

By inspecting Table V we can verify that for the CMUsubject16 dataset the RMSE was generally low. It is also to be noticed the accuracy for the 20% missings was better than the 15% and 10% missings which was not according to expectations. Although this behaviour was not expected, even though this result may have happened due to the low number of tests performed, this could mean the algorithm can still show acceptable results with a large amount of missing values. Regarding the ECG dataset the same could be verified, although with lesser accuracy due to the size of the classes made by the discretization, leading to imputed values with less accuracy. This behaviour also repeats itself in the seizures dataset, although the 10% RMSE results are far from the expected because of the presence of an outlier in the data (and because this algorithm ran only once), and having difficulties with the imputation with the 20% of missing values. An example of the imputation the Seizures dataset is done on Figure 10

3) DTW vs NW: By analyzing the results on Tables IV and V, it is possible to see that both algorithms behave in a similar way (although there are some exceptions such as

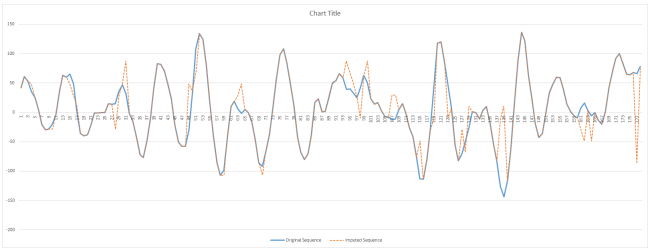


Fig. 10: Alignment of the sequence using NW with 20% missing values for the Seizures dataset.

the results for the Seizures with 10% missing values). The DTW algorithm is mostly the one with the biggest accuracy, although the NW algorithm has shown good accuracy when facing a big percentage of missing values in the ECG and CMUsubject16 datasets. Although DTW had the biggest accuracy it was the slowest when compared to the NW algorithm. With multiple values missing the difference in execution time is more noticeable, because the DTW algorithm analyzes the missing values one by one, while the NW algorithm matches the sequences globally, although more runs of the algorithm may be necessary to fill the missing values aligned with gaps.

While analyzing the imputations in the DTW algorithm it was seen that in the presence of gaps of missing values the imputation accuracy would drop the most, which can be explained by the calculations made for the derivative using only two available points (which is less effective than the derivative using three points). In the NW algorithm these effects were not as noticeable.

D. DTW Discussion

This algorithm gave off the best accuracy overall when compared to the NW algorithm. A few factors could have influenced this such as the fact that this algorithm fills the missing values, and then uses these filled missing values to proceed in finding the next ones. Also, due to the algorithm being applied locally and with several windows, better results were found, which could be overlooked if the algorithm acted on a global scale. That being said, because this algorithm has an exhaustive local analysis, it will also take a long time to complete, especially with multiple missing values on big sequences and a lot of individuals to analyze, which was the case for the Seizures dataset. In order to improve on this aspect a threshold could be established on the scores of the DTW match. Another solution could be comparing all sequences globally in order to determine the ones most likely to succeed in an alignment, and use those on the algorithm. There should also be made attempts to discover a relation between the window size and the size of the sequences being analyzed, as well as the number of windows to be analyzed.

E. NW Discussion

In terms of accuracy, the NW algorithm was lacking when compared to DTW, even though the results were still appropriate when compared to the original values. The positive point is that it takes less time than the DTW algorithm. This is because the algorithm compares sequences globally which makes the process faster. A downside of this is that, when comparing sequences globally, some missing values might only match with gaps, which was countered by adding certain validations, although this means the algorithm will need to process more alignments. Even with this extra step it was still much faster than the DTW algorithm. One of the biggest problems could be the discretization of the sequences. When applying discretization, information will be lost, which could be troublesome when retrieving the final values for the imputation. This issue comes from the fact that the number of classes are chosen based on the size of the sequences being analyzed, which will affect short series with high variability in their data. For example, regarding the ECG and Seizures dataset the size of the sequences and the high variability resulted in a discretization in which classes had a wide range (this happened specially in the ECG dataset).

An additional improvement would be re-designing the similarity matrix. Instead of using the same mismatch value for whenever a mismatch between two characters occur, a score could be assigned to reflect the proximity between the classes. For example, class ‘C’ would have a higher mismatch score with classes ‘B’ and ‘D’ and the mismatch scores would worsen from these on out. This would promote mismatches with the nearest classes, instead of with any class.

Another suggestion of an upgrade for this project is the use of a local version of this algorithm to be compared with DTW which acts locally. To achieve this, the Smith-Waterman algorithm could be used.

V. Conclusion

In this paper, it was intended to analyze the behaviour of two algorithms when applied to input missing values in a continuous time series. This analysis is considered extremely important, given the fact that there are already some predictive algorithms applied to data that have this missing values, and this happening is not being taken in proper consideration, that could be leading to wrong conclusions. Thus, the algorithms implemented were DTW and NW, applied to three different datasets. In addition, some tests were also performed regarding the number of missing values: single missing values imputation and multiple missing value imputation, placing 5%, 10%, 15% and 20% of the dataset as missing values. Results showed both algorithms can be used for imputation, each with their respective strengths - accuracy, when speaking about DTW, or speed, when speaking about NW. Some improvements have been suggested in the previous chapter, which could help lessen the weaknesses of these algorithms. Both of these algorithms should be further tested as well in order to attest them to be used in the imputation of real data.

VI. Acknowledgments

I want to thank my supervisors, Prof. Susana Vinga and Prof. Alexandra Carvalho, for all the support, patience, availability and mainly for providing an incredible experience

during this project. Lastly, I also want to thank to IST for giving me the chance to learn amazing contents and experience great things.

References

[1] Jyoti Soni et al. "Predictive data mining for medical diagnosis: An overview of heart disease prediction". In: *International Journal of Computer Applications* 17.8 (2011), pp. 43–48.

[2] Riccardo Bellazzi, Fulvia Ferrazzi, and Lucia Sacchi. "Predictive data mining in clinical medicine: a focus on selected methods and applications". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.5 (2011), pp. 416–430.

[3] Boris Milovic. "Prediction and decision making in health care using data mining". In: *Kuwait chapter of arabian journal of business and management review* 33.848 (2012), pp. 1–11.

[4] Igor Kononenko. "Machine learning for medical diagnosis: history, state of the art and perspective". In: *Artificial Intelligence in medicine* 23.1 (2001), pp. 89–109.

[5] Rahul C Deo. "Machine learning in medicine". In: *Circulation* 132.20 (2015), pp. 1920–1930.

[6] John S Zdanowicz. "Detecting money laundering and terrorist financing via data mining". In: *Communications of the ACM* 47.5 (2004), pp. 53–55.

[7] A Kusiak et al. "Data mining: medical and engineering case studies". In: *Industrial Engineering Research Conference*. 2000, pp. 1–7.

[8] Mai Shouman, Tim Turner, and Rob Stocker. "Using data mining techniques in heart disease diagnosis and treatment". In: *2012 Japan-Egypt Conference on Electronics, Communications and Computers*. IEEE. 2012, pp. 173–177.

[9] Gustavo EAPA Batista and Maria Carolina Monard. "An analysis of four missing data treatment methods for supervised learning". In: *Applied artificial intelligence* 17.5-6 (2003), pp. 519–533.

[10] Thomas V Perneger and Bernard Burnand. "A simple imputation algorithm reduced missing data in SF-12 health surveys". In: *Journal of clinical epidemiology* 58.2 (2005), pp. 142–149.

[11] Ibrahim Berkan Aydilek and Ahmet Arslan. "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm". In: *Information Sciences* 233 (2013), pp. 25–35.

[12] Donald J Berndt and James Clifford. "Using dynamic time warping to find patterns in time series." In: *KDD workshop*. Vol. 10. 16. Seattle, WA. 1994, pp. 359–370.

[13] Marshall A Beddoe. "Network protocol analysis using bioinformatics algorithms". In: *Toorcon* (2004).

[14] Loris Nanni and Alessandra Lumini. "Generalized Needleman–Wunsch algorithm for the recognition of T-cell epitopes". In: *Expert Systems with Applications* 35.3 (2008), pp. 1463–1467.

[15] Trivellore E Raghunathan et al. "A multivariate technique for multiply imputing missing values using a sequence of regression models". In: *Survey methodology* 27.1 (2001), pp. 85–96.

[16] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

[17] José M Jerez et al. "Missing data imputation using statistical and machine learning methods in a real breast cancer problem". In: *Artificial intelligence in medicine* 50.2 (2010), pp. 105–115.

[18] Evelyn Fix and Joseph Lawson Hodges. "Discriminatory analysis. Nonparametric discrimination: consistency properties". In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247.

[19] Leif E Peterson. "K-nearest neighbor". In: *Scholarpedia* 4.2 (2009), p. 1883.

[20] Marco Di Zio et al. "Bayesian networks for imputation". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 167.2 (2004), pp. 309–322.

[21] Kyoung-jae Kim. "Financial time series forecasting using support vector machines". In: *Neurocomputing* 55.1-2 (2003), pp. 307–319.

[22] Eamonn Keogh et al. "Finding unusual medical time-series subsequences: Algorithms and applications". In: *IEEE Transactions on Information Technology in Biomedicine* 10.3 (2006), pp. 429–439.

[23] Robert M Hirsch, James R Slack, and Richard A Smith. "Techniques of trend analysis for monthly water quality data". In: *Water resources research* 18.1 (1982), pp. 107–121.

[24] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. springer, 2016.

[25] Eamonn J Keogh and Michael J Pazzani. "Derivative dynamic time warping". In: *Proceedings of the 2001 SIAM international conference on data mining*. SIAM. 2001, pp. 1–11.

[26] Computation Biology. *DTW algorithm*. URL: https://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm.

[27] Émilie Poisson Caillault, Alain Lefebvre, André Bigand, et al. "Dynamic time warping-based imputation for univariate time series data". In: *Pattern Recognition Letters* (2017).

[28] Lukasz Ligowski and Witold Rudnicki. "An efficient implementation of Smith Waterman algorithm on GPU using CUDA, for massively parallel scanning of sequence databases". In: *2009 IEEE International Symposium on Parallel & Distributed Processing*. IEEE. 2009, pp. 1–8.

[29] Samuel David Pelaio Arcadinho. "Model-based Learning in Multivariate Time Series". In: (2018).

[30] UCI. *Epileptic Seizure Recognition Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition.