MSAX: Multivariate symbolic aggregate approximation for time series classification

Manuel Anacleto
Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal

***Abstract*—Time Series (TS) analysis is a central research topic in areas such as finance, bioinformatics, and weather forecasting, where the goal is to extract knowledge through data mining techniques. Symbolic aggregate approximation (SAX) is a state-of-the-art method that performs discretization and dimensionality reduction for univariate TS, which are key steps for TS representation and analysis. In this work, we propose MSAX, an extension of this algorithm to multivariate TS that takes into account the covariance structure of the data. The method is tested in several datasets including the Pen Digits, Character Trajectories, and twelve benchmark files. Depending on the experiment, MSAX exhibits comparable performance with state-of-the-art methods in terms of classification accuracy. Although not superior to 1-NN and DTW, it has interesting characteristics for some classes, and thus enriches the set of methods to analyze multivariate TS.**

***Index Terms*—SAX, Time Series, classification, multivariate analysis**

I. Introduction

The huge quantity of available data nowadays is posing new challenges for knowledge discovery, namely to extract meaningful information such as significant patterns, statistics and regularities. Temporal data, and in particular time series (TS), are now pervasive in many fields, which fully justifies the development of new methods for their analysis. A discrete TS is a series of n observations, each one being measured at a discrete time $t \in \{1, \ldots, T\}$, made sequentially and regularly trough T instances of time. In this case, the i-th TS is given by $\{\boldsymbol{x}^i[t]\}_{t\in\{1,\ldots,T\}}$, where $\boldsymbol{x}^i[t] = (x_1^i[t], \ldots, x_n^i[t])$. When $n = 1$ the TS is said to be univariate; otherwise, when $n > 1$, it is multivariate.

Data representation takes a big focus in TS analyses. A rich wealth of data structures and algorithms for streaming discrete data were developed in recent years, especially by the text processing and bioinformatics communities. In order to make use of these methods, real-values TS need symbolic discretizations. Beside this, representation methods also address the TS dimensionality problem arising from the fact that almost all TS datasets are intrinsically of high dimensionality.

In contrast to univariate TS, Multivariate TS (MTS) are characterized not only by serial correlations (auto-correlation) but also by relationships between the attributes measured at the same time point (intra-correlation). Due to considering the attributes individually their intra-correlations can not be captured, as shown in [4], [6]. In [5], the necessity of different TS representations for MTS classification was discussed and pointed out as desirable the development of methods that consider all attributes simultaneously taking into account the relationships between them.

TABLE I
The look up table used by the MINDIST function. This table is for an alphabet of cardinality of 4, i.e. $a = 4$. The distance between two symbols can be read off by examining the corresponding row and column. For example, $dist(\mathbf{a}, \mathbf{b}) = 0$ and $dist(\mathbf{a}, \mathbf{c}) = 0.67$

	a	**b**	**c**	**d**
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

This work proposes a multivariate extension of the well-known TS representation of Symbolic Aggregate Approximation (SAX) [1]. In the SAX method the TS is normalized to have a temporal mean of zero and a standard deviation of one. A TS normalized in this manner has a Gaussian distribution [2]. If desired, Piecewise Aggregate Approximation (PAA) [3] is then applied, reducing the TS length. This technique divides the TS into w (method parameter) segments of equal length, where each segment is replaced with its average value that is further grouped in a vector representing the TS.

Assuming that the normalized TS has a Gaussian distribution [2], it is possible to divide it into equal size areas under the Gaussian curve trough breakpoints, producing equiprobable symbols. These breakpoints may be determined by a statistical table inspection. After, the discretizing process is done by associating the TS points to a (method parameter that represents the size of the symbolic alphabet) equal area intervals beneath the Gaussian curve associated to the TS to be discretized. An illustrative example of the discretizing process is shown in Fig. 1.

Having a discretized TS, a distance measure between two TS $Q = q_1 q_2 \ldots q_T$ and $C = c_1 c_2 \ldots c_T$, in the new representation space can be defined as:

$$\text{MINDIST}\,(Q, C) = \sqrt{\frac{T}{w}}\sqrt{\sum_{i=1}^{w} dist\,(q[i], c[i])^2}. \qquad (1)$$

The function $dist()$ is implemented using a lookup table as illustrated in Table I.

The function $dist()$ that returns the distance between two symbols is implemented using a lookup table in which the

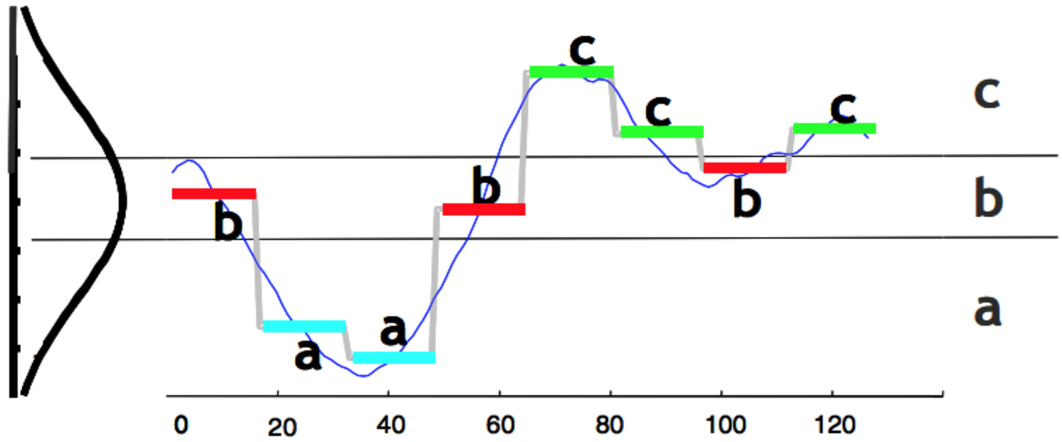


Fig. 1. The TS (in ligth blue) is discretized by first applying the PAA technique and then using predetermined breakpoints to map the PAA coefficients into the symbols. In the example above, with $T = 128$, $w = 8$ and $a = 3$, the TS is mapped to the word BAABBCBC. Figure from [1].

value for entry (r, c) is obtained trough the following function, where β represents the breakpoints values:

$$cell_{r,c} = \begin{cases} 0, & \text{if} \quad |r - c| \leq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & \text{otherwise.} \end{cases} \quad (2)$$

SAX is the only symbolic TS representation, until now, for which the distance measure in the symbolic space lower bounds the distance in the original TS space. This is assumed to be one of the reasons fot its good performance [1], [2]. Nevertheless, SAX only works for univariate TS, paving the way for extending its promising results for MTS. In the literature, SAX has been applied to MTS by dealing with each variable independently, disregarding intra-correlations in the discretization process [7], [8]. We propose to explore these intra-correlations in order to understand the benefits of using these dependencies.

II. MSAX: SAX EXTENSION TO MULTIVARIATE TS

The method proposed in this work, MSAX, expands the SAX algorithm by first performing a multivariate normalization of the MTS. The rationale for this first step is to account for the mean and covariance structure of the data $\boldsymbol{X}[t]$, i.e., $E[\boldsymbol{X}[t]] = \mu$ and $Var[\boldsymbol{X}[t]] = \Sigma_{n \times n}$.

The normalized TS values $\boldsymbol{Z}[t]$ are given by $\boldsymbol{Z}[t] = \Sigma^{-1/2}(\boldsymbol{X}[t] - \boldsymbol{\mu})$, such that the obtained distribution has zero mean and uncorrelated variables. Assuming a Gaussian distribution, we are able to identify the cut points and intervals that define equal volumes, a crucial step to identify the areas associated to symbols used in the discretization.

A. MSAX discretization

After the normalization step, and like in the original method, the PAA procedure is applied to each variable individually, in order to reduce its length dimensionality. First, before the proper discretization of the TS values, the volumes associated to each symbol beneath the multivariate Gaussian curve are defined. With this in consideration the following reasoning is used to define the volumes and correponding cut-points.

Due to the normalization step, the new variables of the MTS are now uncorrelated, i.e., the covariance matrix of the TS is the identity matrix. Since the probability density function of the MTS is equal to the product of the probability density function of each variable when no correlation between the variables exist, a Gaussian distribution of $\mathcal{N}(0, 1)$ is associated to each series variable of the TS, in the same way as in the original method. Then, each Gaussian curve associated to a variable will be split using breakpoints in order to the area of each space split beneath the Gaussian curve to be equal for all the spaces split regions, this is done in the same way the original method in accordance with the a parameter (that indicates the alphabet size per variable).

After, the split regions under the multivariate Gaussian curve are defined trough the breakpoints intersection for each variable, this results in the variable space to be splitted in a grid way with each partition of the grid having the same volume under the multivariate Gaussian curve. Finally, the points of the normalized and PAA processed MTS are mapped to the multivariate splitted space beneath the Gaussian curve associated with multivariate TS. As a result of the entire process a univariate discrete TS is obtained from the multivariate numerical TS.

An example of the full discretization process is given with bivariate TS normalized points X where x_1 and x_2 represent each dimension. Fig. 2 illustrates the Gaussian curve associate to this distribution. If three symbols per variable are used in the discretization process, $a = 3$, the discretization shown in Fig. 2 (right) is obtained, with a total of nine symbols. The final symbol value is obtained by the concatenation of the symbols associated to each each variable (this symbols will be designated by symbol of variable to distinguish from the final symbol). Using the purple partition of the picture as example, its final symbol value is aB, directly obtained by concatenating $x_1 = a$ and $x_2 = B$ (purple partition).

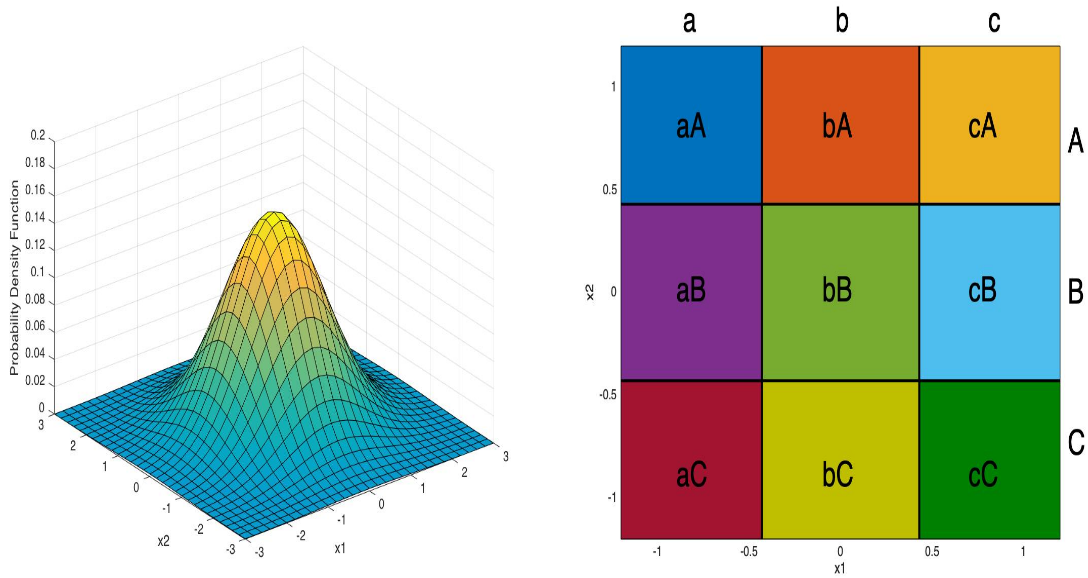


Fig. 2. On the left, plot of the probability density function or Gaussian curve with distribution $\mathcal{N}(0, I)$, for two variables x_1 and x_2. On the right, the areas associated to each symbol on the x_1, x_2 plane, for $a = 3$. Each area with a different color is associated to a symbol. For example, a point situated on the area in orange, the x_1 variable value is associated to b and the x_2 variable value is associated to A. To this x_1, x_2 example point will be associated final symbol of bA.

B. Dissimilarity definition

Having introduced this new representation of MTS, a novel dissimilarity measure should be defined. Two symbolic univariate TS Q and C of the same length T, obtained from a MTS with n attributes, are considered. The distance measure between two TS using the MSAX representation is given by the sum of the distances between each two time points, for all the indexes of the TS length, where the distance between two final symbols of the MSAX is obtained by the sum of the difference between the symbol of variable associated to each variable in this representation:

$$\text{MINDIST\_MSAX}\,(Q, C) = \sqrt{\frac{T}{w}}\sqrt{\sum_{i=0}^{w}\left(\sum_{i=0}^{n} dist\,(q[i], c[i])^2\right)}. \tag{3}$$

The distance between two symbols is calculated based on the univariate representations, and by using the corresponding distance defined originally, i.e., obtained through the same table used in the original SAX distance. This results stems from the fact that the breakpoints are the same due to the Gaussian properties.

In Fig. II-B is illustrated the comparison between an example case TS original representation and its discretization result using MSAX.

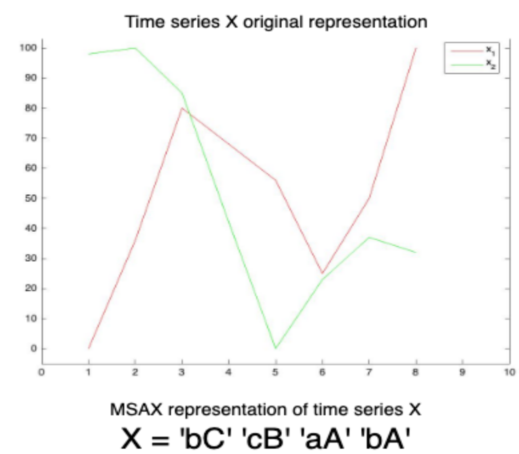
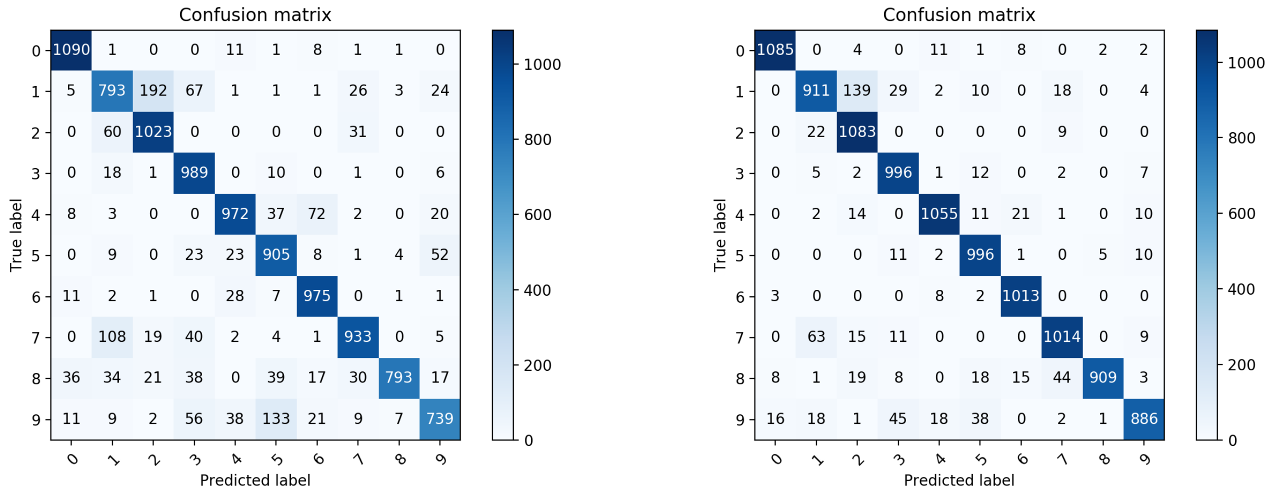


Fig. 3. Illustrative picture of the comparison between an original TS and its symbolic representation obtained trough the application of MSAX. The example case is a bivariate TS with length of 8, the SAX parameters used are $a = 3$ and $w = 4$ achieving a length reduction ratio of $1/2$.

III. **Results**

In this section the MSAX algorithm is tested. As referenced before, in this work the type of task used to assess the method behaviour is classification. This is done through the integration of the MTS representation algorithms with the 1-NN classifier. When setting the classifier as the 1-NN, the only variable in the classification process will be the TS representation method and the associated distance measure, that is exactly what is desired to study. Due to this, in all these studies it is assumed that a classification with a greater accuracy corresponds to a better representation performed by the TS representation algorithm. In these experiments, the following reasoning is applied, MSAX is compared with the direct application of SAX to MTS (SAX_INDP) in order to understand if the proposed method achieves a better representation of MTS. First, the method evaluation is done through the use of two



(a) MSAX (b) SAX_INDP

Fig. 4. Confusion matrices of the classification results of 1-NN with MSAX and SAX_INDP (with an alphabet size of 10 and a TS length reduction ratio of 1) for the PenDigits dataset.

case studies datasets, in these, the algorithms comparison is performed vastly in multiple experiences that study several algorithm features, in turn, this is done through the analysis of metrics like, accuracy and confusion matrix. After, the MSAX and SAX_INDP comparison is expanded to several datasets. In the end, the proposed method is compared with the MTS classifiers studied and tested in the chapter before, where its significance in MTS classification is approached.

As the great difference between MSAX and SAX_INDP lies on the fact that MSAX takes in account the intra-correlation between the MTS variables, MTS with known and high intra-correlations between variables were desirable to understand if MSAX can take advantage of these, translating into a better MTS representation than SAX_INDP. Two case studies datasets were chosen for the first experimental approach of the method. Both were selected owing to their strong and clear to analyse, intra-correlation between the MTS attributes. These are: the *PenDigits* dataset [10] that consists in multiple labeled samples of digit trajectories and the *CharacterTrajectories* dataset [9], representing instead trajectories of characters from the English alphabet.

A. Case studies experiments

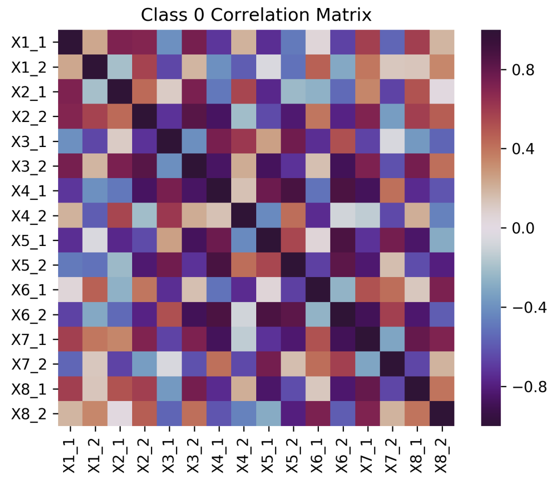
.

The first experiments done were made with the goal of checking if the MSAX could take advantage of the intra-correlations of the MTS in order to achieve a better representation of them. For this the two cases studies datasets were used.

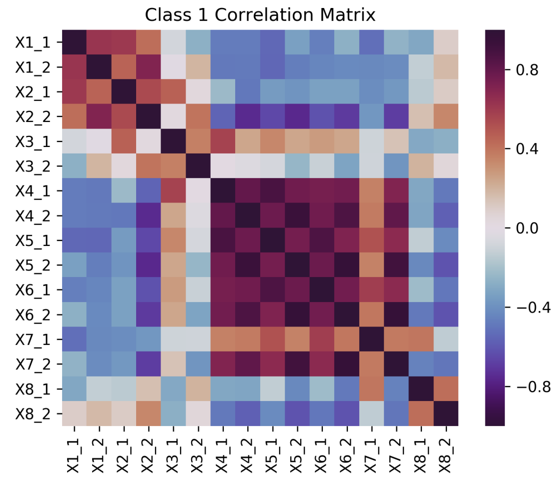
First, MSAX and SAX_INDP were used in a classification task on the PenDigits dataset. Several parameters combination for alphabet size and TS length reduction ratio were tested for both algorithms, but in the end, an alphabet size of 10 and a TS length reduction ratio of 1 were chosen due to being the parameters values that achieve the highest accuracy results for MSAX algorithm. The accuracy obtained for MSAX was 0.861 and for SAX_INDP 0.91. Just by the comparison of these accuracy's MSAX seems not competitive with SAX_INDP, nonetheless the confusion matrices will be analyse in order to understand if for the digit classes with high correlations features MSAX can take advantage of it. After the classification results were obtained, the confusion matrices for each algorithm were draw, these can be seen in Fig. 4.

For a confusion matrix comparison analysis, the correlations matrices for each class were draw with the objective to understand the intra-correlations features of each class. The correlation matrix represents the correlation between all the variables at each time point for all the MTS of a specific class label. Each matrix cell represent the Pearson correlation coefficient between two variable at a certain time point. The Pearson correlation coefficient ranges between -1 and 1, where 1 represents a total positive linear correlation, 0 is no linear correlation and -1 is total negative correlation. The idea behind the use of these matrices is to evaluate each class intra-correlations features through the presence (or not) of cell entries areas with high values (near 1 or -1), which in turn correspond to high correlations features.

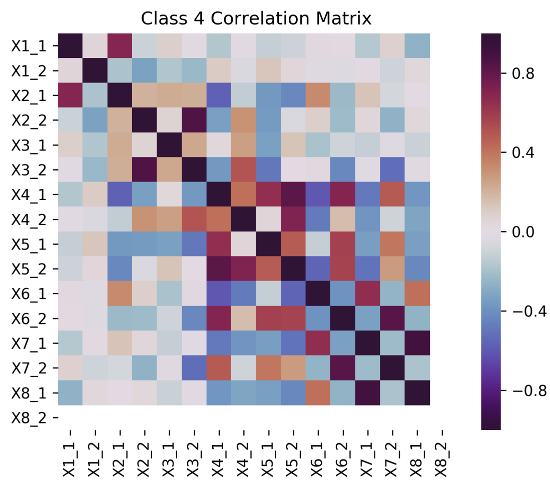
In Fig. 5 the correlations matrices for several classes of the PenDigits dataset are plotted. The axis label reading is done in this way: the PenDigits dataset is composed of bivariate MTS of length 8, so for example, $X4\_2$ represents the second variable at the fourth time point index and $X8\_1$ represents the first variable at the eight time point index. The matrices are represented through heat maps with a cyclic color bar so that for higher (through red) and for lower coefficients values (through blue) the color goes darker, which indicates high linear correlation. The values near 0 are brighter and whiter,



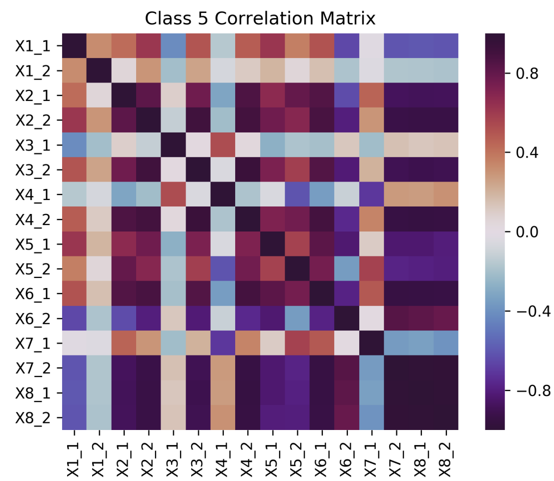
(a) Class 0



(b) Class 1



(c) Class 4



(d) Class 5

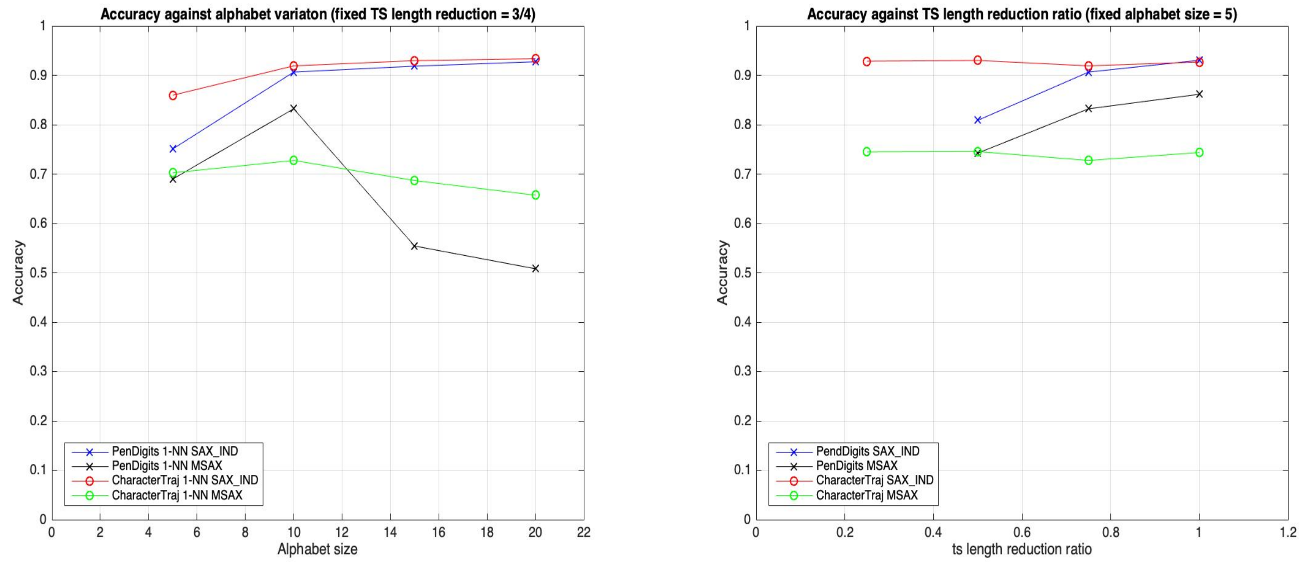
Fig. 5. Correlation matrices of several classes of the PenDigits dataset. The PenDigits dataset is composed of bivariate MTS of length 8, so for example, $X4\_2$ represents the second variable at the fourth time point index.

that in turn indicate low linear correlation.

The first aspect to point in the comparison analysis of the confusions matrices is that for 9 of the 10 classes SAX_INDP achieves a better prediction. Only for the 0 digit class MSAX achieves a better prediction, in Fig. 5 the correlation matrix for this class can be observed. No specific area of correlation can be identified (in comparison to the digit 1 class matrix, where the middle red correlation area is well defined) but from the high presence of dark colors along the matrix, this class can be associated with high intra-correlations features. In contrast with, for example, the digit class 4 where a high presence of whiter colors along the matrix indicates a class with lower intra-correlations features.

Following this reasoning, the digit 5 class that in the same figure presents a very dark correlation matrix would make sense to also achieve a better prediction with the MSAX algorithm. But as it can be seen in the confusion matrices figure this doesn't happen. So from this experiment is difficult to draw a conclusion if MSAX can take advantage from the MTS intra-correlations. Nonetheless, one clear aspect that can be observed from the confusion matrices figure is that, constantly, MSAX prediction enhances the errors of SAX_INDP. For example, in the digit class 2 this can be easily observed, MSAX incorrectly labels some digit 2 class instances but always confusing them to the digits 1 or 7, that is exactly the same confusion observed with SAX_INDP but with a great number of occurrences.

Following, it was desired to study the influence of the algorithms parameters in the representation obtained. In this one the parameters of the algorithms: alphabet size and length



(a) Accuracy against alphabet size

(b) Accuracy against TS length reduction ratio

Fig. 6. For the PenDigits and CharacterTrajectories datasets, the accuracy of 1-NN with MSAX and SAX_INDP is plotted, in the left graph, against the alphabet size values of 5, 10, 15 and 20 for the fixed TS length reduction ratio of $3/4$. In the right plot the accuracy is plotted against the TS length reduction ratio of $1/4$, $1/2$, $3/4$ and 1 for a fixed alphabet size of 5.

of the symbolic TS (associated to the TS length reduction ratio), were varied in order to understand its influence in the TS representation achieved. For the alphabet size influence study, the TS length reduction ratio was fixed as $3/4$ and the alphabet size was ranged from 5 to 20. In the variation of the TS length reduction ratio case, the alphabet size was fixed as 5 and the TS length reduction ratio was ranged from $1/4$ to 1. The obtained values are presented in Fig. 6.

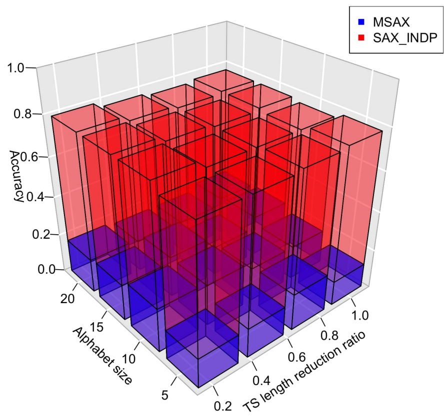
Starting by the analysis of the alphabet size variation plot. Regarding SAX_INDP, in both datasets, as the alphabet size increases the accuracy also increases, stagnating after second a value around a value of 0.92. For MSAX, in both datasets the same increases is seen along the first two values but then the accuracy starts to decrease in the last two points, achieving a better accuracy with low alphabet size values. Comparing both algorithms, SAX_INDP achieves a better accuracy for all a values, in the PenDigits dataset the accuracy achieved besides lower, is still competitive for the first two alphabet size values but in the CharacterTrajetories dataset the difference its significant for all points.

Now regarding the TS length reduction ratio variance plot. First it is relevant to point out that for the PenDigits dataset a TS length reduction ratio of $1/4$ was not used due to the small length of 8 of the MTS. With the usage of a TS length reduction ratio of $1/4$ the MTS is reduced to two temporal points which resulted in very low results in this experiments, which justifies itself due to the significant reduce of information. So due to this, the lowest TS length reduction ratio used for the PenDigits dataset was of $1/2$. In the PenDigits dataset as the TS length reduction ratio is increased the accuracy is also increased, this is clearly seen for both algorithms, nonetheless the SAX_INDP achieves a constant superior accuracy for all points. In the CharacterTrajetories dataset the behaviour of both algorithms is constant as the TS length is varied, regarding to the comparison between the algorithms the same as before is present the, SAX_INDP accuracy proves constantly superior to MSAX.

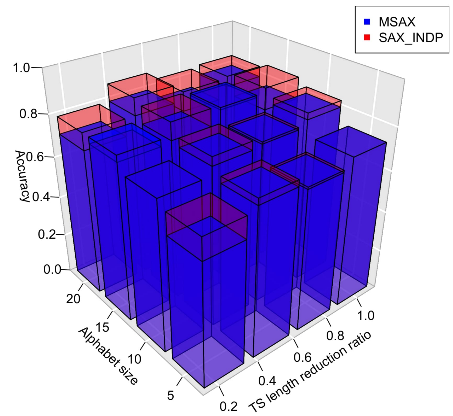
Concluding, from these two experiments the key ideas drawn were: in general SAX_INDP proves a better accuracy than MSAX; for SAX_INDP an increasing alphabet size tends to an higher accuracy, in the MSAX case this increase is only observed for the lowest alphabet size values, starting to decrease in the last two, this decreasing behaviour is not understood because an increase of the alphabet size, which in turn, is an increase of the information carried by the MTS representation, should lead to a better representation and increasing accuracy; regarding to the TS length reduction ratio the behaviour is the same for both algorithms, in one dataset there is an increase in the accuracy as the ratio increases and in the other there is no variation, so due to this no conclusions are withdraw.

B. Experiments on benchmark datasets for MTS classification

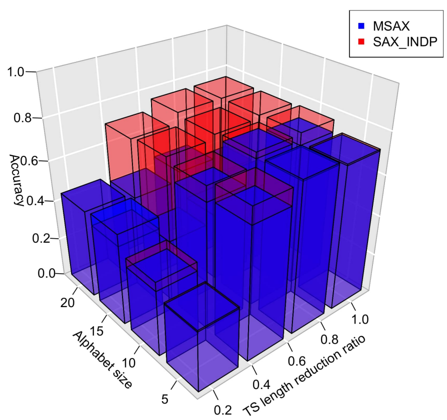
After the analysis of the method through the case studies datasets an analysis with a broader range of 15 datasets datasets was done. These datasets are common benchmark datasets from the MTS classification community TS with different characteristics from a wide range of areas, they can be found in [11], [12]. In these experiences the algorithms were tested with an extensive parameter variation. The alphabet size



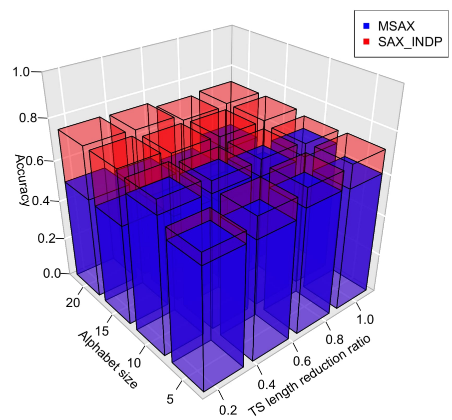
(a) AUSLAN Dataset



(b) ECG Dataset



(c) Epilepsy Dataset



(d) LIBRAS Dataset

Fig. 7. MSAX and SAX_INDP accuracy's comparison for several parameters combinations with: AUSLAN, CharacterTrajectories, ECG, Epilepsy, Libras and PenDigits datasets.

was ranged from 5 to 20 with an interval of 5 and the TS length reduction ratio from $1/4$ to 1 with an interval of $1/4$. Both MSAX and SAX_INDP were ran on each dataset for each possible parameter combination. The accuracy results for the two algorithms are presented in Fig. 7. Due to the extension of datasets, in this figure only some datasets are presented, these were chosen due to presenting more common and significant trends. Following an analysis on some datasets values from Fig. 7 is done.

In the first dataset, **AUSLAN**, the SAX_INDP algorithm is much superior than MSAX for all parameter combinations. In regards to the parameter variation some small variations can be seen, but the accuracy is more or less constant both in SAX_INDP and MSAX. In the **ECG** dataset, MSAX is competitive with SAX_INDP achieving higher accuracy for several parameters combination. In relation to the parameter variation, for both algorithms, an increase alphabet size generates results with a better accuracy and the same is seen for the TS length reduction ratio. Regarding to the **Epilepsy** dataset, for the first alphabet size values MSAX is competitive with SAX_INDP but for the last values the accuracy values decrease achieving SAX_INDP a superior accuracy. This pattern where the MSAX accuracy values for the last alphabet size values tends to decrease, is also recurrent in others datasets. Also in this dataset it can be seen the accuracy of both algorithms increasing as the TS length reduction ratio increases. For the

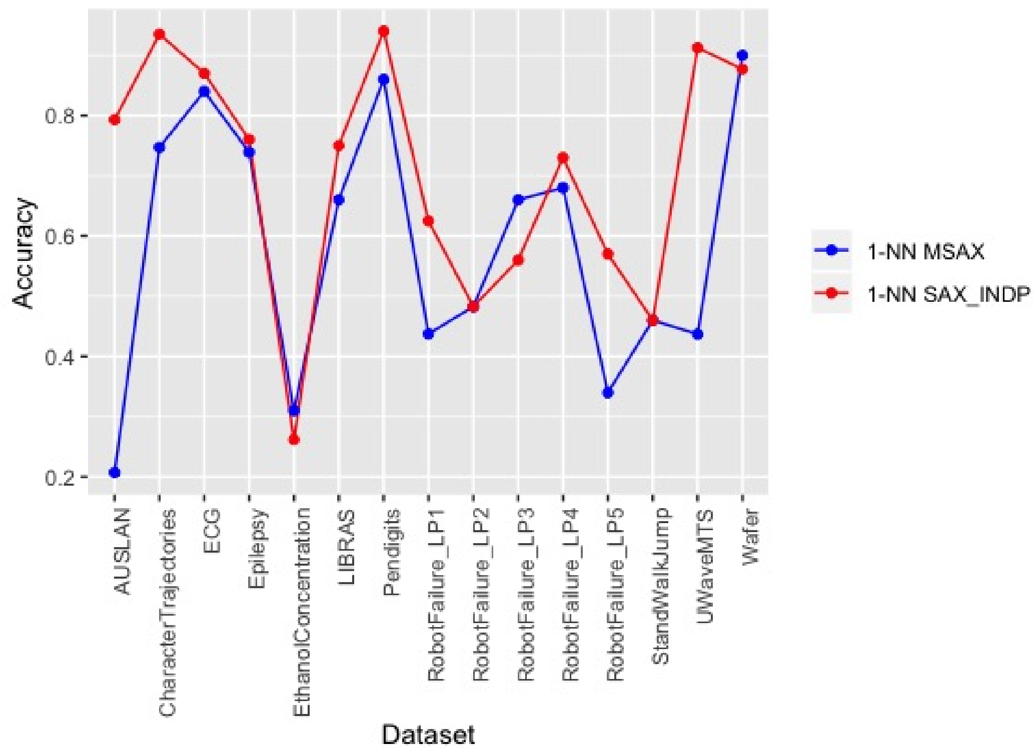


Fig. 8. Accuracy plot of the 1-NN MSAX and 1-NN SAX_INDP values in a MTS classification experiment with 15 datasets.

LIBRAS dataset the SAX_INDP proves superior and both algorithms present a more or less constant accuracy through the parameter variation.

Through an analysis of the data from each dataset some significant trends were observed, the most relevant one is that in general SAX_INDP achieves a higher accuracy than MSAX, tough is some cases MSAX can still be competitive. Also, MSAX accuracy values tend to increase has TS length reduction increases, in regard to the alphabet size parameter is hard to withdraw a clear conclusion due to the variance of the obtained results, in some cases the accuracy tends to increase as the alphabet size increases but its also common in some datasets decreasing for the last two alphabet size values. The same behaviour is not observed in relation to SAX_INDP where for both parameters the accuracy tends to increase as they increase, this behaviour makes sense in the way that more information results in a higher accuracy, which in turn reveals a better MTS representation. Nonetheless is key to refer that these patterns aren't present in all the datasets, the results obtained through out all datasets vary a lot, so even if this trends were found significant, they are not constant.

From the values presented before, for each dataset, the highest accuracy achieved from all the parameters combinations tested was selected from both algorithms and in Fig. 8 theses values are plotted to comparison.

In the 15 datasets tested SAX_INDP achieves a significant higher accuracy in 8 of them, in 4 of them the accuracy's obtained are very competitive and in 3 of them MSAX achieves significant higher accuracy. From these results it is clear to concluded that SAX_INDP, in general performs better than MSAX in MTS classifications tasks, which points to the conclusion that its representation, in general, its preferable than MSAX.

IV. **Method discussion**

In regards to the MSAX behaviour several patterns were clearly extracted from the experiments before, others not so much. It is clear from all experiments that MSAX in general is not competitive with SAX_INDP in classification tasks, thought for some specific cases it can achieve good and competitive results. When translating the results of the experiments to the quality of the MTS representation algorithm it is concluded that SAX_INDP in general achieves a better MTS representation. Regarding to MSAX taking advantage of the MTS intra-correlations, this was not clearly seen in the cases studies datasets experiments where the classification confusion matrices and the classes correlations matrices were studied. A deeper investigation on the causes of this behaviour is pointed out as future investigation guideline. Now, concerning the the algorithm parameters influence on the quality of the TS representation, due to the great variance on the obtained results, the behaviour is not self evident. For the SAX_INDP, a higher alphabet size or TS length reduction ratio tends to a higher accuracy which in turn is associated to a MTS representation of better quality. For MSAX, the increase of accuracy associated to an increasing TS length reduction ratio can be also observed, but not so linearly as in SAX_INDP. Regarding the alphabet size influence, the most significant pattern was that for low alphabet size values, an increase as seen in SAX_INDP is also observed, but for higher values (example: 15 and 20) the accuracy starts to decrease. The cause of this behaviour would also be a relevant point to investigate in a continuous study of MSAX.

V. Conclusion

In this work an extension of SAX for MTS, named MSAX, was proposed. MSAX behaviour was accessed in classifications tasks, comparing it with the SAX_INDP, the original SAX algorithm applied independently to each attribute in the MTS. We concluded that the proposed method is overall not competitive with the SAX_INDP. Nonetheless, the obtained results have utility as benchmark values for SAX-based methods in multivariate classifications tasks. It is also noteworthy that for some datasets and specific cases, MSAX surpasses the other methods.

As future direction, MSAX could be investigated more deeply in order to understand some question that were left unanswered in this work about the MSAX behaviour, referenced before. Also MSAX applications could be studied in other data mining areas, such as clustering or forecasting, to check if it could be useful and achieve competitive performance with state-of-the-art methods.

Acknowledgment

The authors acknowledge funding the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia – FCT) Supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under contracts INESC-ID (UID/CEC/50021/2019) and Instituto de Telecomunicações (UID/EEA/50008/2019), as well as projects PTDC/CCI-CIF/29877/2017, PTDC/EEI-SII/1937/2014 and PTDC/EMS-SIS/0642/2014. Supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) through projects UID/CEC/50021/2019, UID/EMS/50022/2019, UID/EEA/50008/2019, SFRH/BD/97415/2013, PTDC/EMS-SIS/0642/2014 and PTDC/CCI-CIF/29877/2017.

References

[1] Lin, Jessica and Keogh, Eamonn and Lonardi, Stefano and Chiu, Bill. "A symbolic representation of time series, with implications for streaming algorithms". *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2-11, 2003.

[2] Lin, Jessica and Keogh, Eamonn and Wei, Li and Lonardi, Stefano. "Experiencing SAX: a novel symbolic representation of time series". *Data Mining and knowledge discovery*, vol.15, no.2, pp. 107-144, 2007.

[3] Keogh, Eamonn and Chakrabarti, Kaushik and Pazzani, Michael and Mehrotra, Sharad. "Dimensionality reduction for fast similarity search in large time series databases". *Knowledge and information Systems*, vol.3, no.3, pp. 263-286, 2001.

[4] Xiaoqing Weng and Junyi Shen. "Classification of multivariate time series using locality preserving projections". *Knowledge-Based Systems*, vol.21, no.7, pp. 581-587, 2008.

[5] Kadous, Mohammed Waleed and Sammut, Claude. "Classification of Multivariate Time Series and Structured Data Using Constructive Induction". *Machine Learning*, vol.58, no.2, pp. 179-216, 2005.

[6] Zoltán Bankó and János Abonyi. "Correlation based dynamic time warping of multivariate time series". *Expert Systems with Applications*, vol.39, no.17, pp. 12814-12823, 2012.

[7] Esmael, Bilal and Arnaout, Arghad and Fruhwirth, Rudolf and Thonhauser, Gerhard. "Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations". *LNCS*, 2012.

[8] Zhiguang Wang and Wei Song and Lu Liu and Fan Zhang and Junxiao Xue and Yangdong Ye and Ming Fan and Mingliang Xu. "Representation Learning with Deconvolution for Multivariate Time Series Classification and Visualization". *CoRR*, vol. abs/1610.07258, 2016.

[9] Ben H Williams. "UCI Machine Learning Repository". https://archive.ics.uci.edu/ml/datasets/Character+Trajectories. 2008.

[10] F Alimoglu. "Combining multiple classifiers for pen-based handwritten digit recognition". 1996.

[11] Anthony J. Bagnall and Hoang Anh Dau and Jason Lines and Michael Flynn and James Large and Aaron Bostrom and Paul Southam and Eamonn J. Keogh. "The UEA multivariate time series classification archive, 2018". *CoRR*, vol. abs/1811.00075, 2018.

[12] M. Bayodan. "Mustafa Baydogan website". http://www.mustafabaydogan.com. 2019.