Quantile Symbolic Aggregate approXimation: A guaranteed equiprobable SAX

Eduardo Silveira [1], **Joaquim Assunção** [1], **Leonardo Emmendorfer** [1]

[1]Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)
Av. Roraima 1000, Cidade Universitária, Prédio 7, Bairro Camobi
Santa Maria, Brazil. CEP: 97105-900.

{esilveira, joaquim}@inf.ufsm.br, leonardo.emmendorfer@ufsm.br

Abstract. *Time series are broadly present in science and industry. In specific scenarios, it is useful to classify series in order to gain knowledge regarding a specific range of values. In such cases, we often use symbolic representation, as it can reduce the data dimensionality creating representative symbols, making the data discrete and allowing specialized algorithms to be applied to the data. One of the most prominent methods of this type of representation is the Symbolic Aggregate approXimation (SAX), which, in addition to generating the symbolic sequence, also reduces the data dimension. However, one of the problems of SAX is that, in order to guarantee the balance of symbols, it assumes the normality of the distribution, which fails in some distributions and causes the class imbalance problem. We propose a unique seamless approach to guarantee the balance among the classes, which may lead to better performance in classification algorithms.*

1. Introduction

Symbolic Aggregate approXimation (SAX) [Lin et al. 2007] is a method applied to achieve a symbolic representation and dimensional reduction of time series. This type of representation is flexible and fast to work with real-time algorithms and also provides a discrete data representation, from an original continuous, which can be easily used by discrete-oriented approaches, such as genetic programming [Espejo et al. 2010]. Besides, SAX proposes a distance metric defined over the symbolic space which is limited inferiorly. This feature is useful for larger time series, which would require higher processing power or not fit in the main memory.

In order to ensure class balancing, SAX assumes the data distribution to be normal and determines the continuous intervals corresponding to each discrete symbol. However, when the data are not normally distributed, the generated representation might be imbalanced. Class balance is important since machine learning algorithms tend to be biased towards the majority class under imbalanced classes situations [Niaz et al. 2022]. Despite the prevalence of normal distributions, this type of balancing would not work under all circumstances (*i.e.*, different distributions). A few versions of SAX already address this problem [Kloska and Rozinajova 2020, Bountrogiannis et al. 2021, Bountrogiannis et al. 2022]. Among them we can highlight the Distribution-wise Symbolic Aggregate approXimation (dwSAX) [Kloska and Rozinajova 2020] which uses the Kernel Density Function to estimate the underlying distribution leading to a fairer distribution of symbols. However, even versions like dwSAX do not guarantee an even distribution.

In order to address this issue, this work proposes a version of SAX called Quantile Symbolic Aggregate approXimation (qSAX), which ensures class balancing, regardless of the input data distribution.

2. SAX and dwSAX

Symbolic Aggregate approXimation (SAX) [Lin et al. 2007] is used to transform a time series with size n into a sequence with w symbols, without much loss of information in the process. This process can be executed in two steps: initially, Piecewise Aggregate Approximation (PAA) performs a dimensional reduction and generates an intermediate representation with dimension w. Next, a discretization is performed, where a symbol is set to each element of that intermediate representation.

Given a time series C with size n, and an integer $w \mid w < n$, one can transform C into an array $\bar{C}$ with size w as follows. C is split into w equally sized sections, then, the arithmetic mean is computed for each section, making the array $\bar{C}$ to be composed by the w means. Therefore, the i-th element in $\bar{C}$ can be computed as defined in Equation 1.

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{1}$$

Before applying the PAA, the time series must be normalized such that the mean equals zero and the standard deviation equals one. After the normalization, a symbol from an alphabet A is assigned to each element in $\bar{C}$. Since the probability must be uniformly distributed over the symbols in A, a set of breakpoints must be determined, which splits the probability density function of the time series into sections with similar probabilities. Lin *et al.* [Lin et al. 2003] define this set as the ordered list $B = \{\beta_1, \ldots, \beta_{a-1}\}$, with $\beta_0 = -\infty$ and $\beta_a = \infty$, such that the area under a normal distribution $N(0, 1)$ from β_i to β_{i+1} is $\frac{1}{a}$. Given an alphabet size and an assumed normally distributed data, the breakpoints can be determined.

Given a sequence $\bar{C}$, an alphabet A and a set of breakpoints B, one can obtain a symbolic representation for $\bar{C}$, which we denote by $\hat{C}$, attributing a symbol $\alpha_j \in A$ to each $\bar{c}_i \; in \; \bar{C}$ if, and only if, $\beta_{j-1} \leq \bar{c}_i \leq \beta_j$, which guarantees that the probability of each symbol to occur is the same for all symbols in A. Lin *et al.* [Lin et al. 2003] call this representation *word*, which is the output of the SAX algorithm. However, the guarantee only holds true when the data is normally distributed.

The Distribution-Wise Symbolic Aggregate approXimation (dwSAX) is designed to obtain probability distributions from data when Gaussian assumptions are not necessarily met. A Kernel Density Estimator (KDE) is adopted, which fits the actual distribution of data. The KDE works with an independently and identically distributed sample $Y = \{y_1 ... y_n\}$, with an unknown density which is mapped by a function kernel K and a smooth parameter h ($h \in R; h > 0$), the KDE $\hat{f}(y)$, for data y is defined in Equation 2.

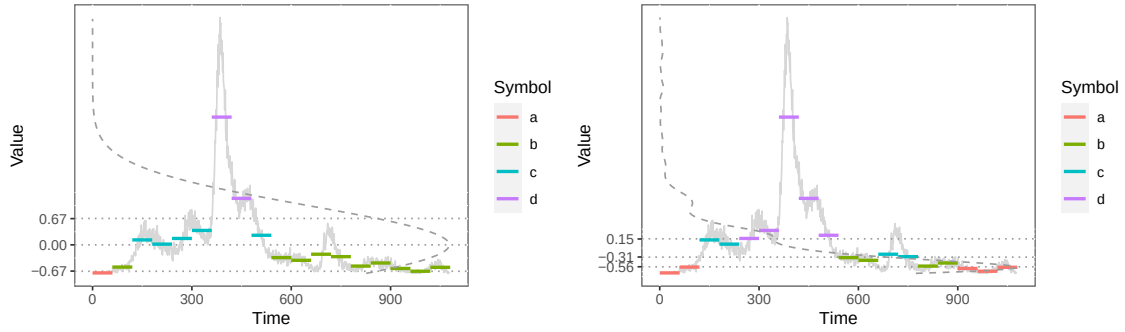
$$\hat{f}(y) = \frac{1}{N} \sum_{n=1}^{N} K(\frac{y - y_n}{h}) \tag{2}$$

However, the precision of $\hat{f}$ highly depends on the h parameter (also known as bandwidth). Thus, the fit is tied to the correct use of h and cannot guarantee equally distributed data output, which can cause problems for mining specific data intervals.

3. Quantile Symbolic Aggregate approXimation (qSAX)

qSAX key change lies on the forced equiprobable output, regarding the number of symbols. In qSAX, to determine the breakpoints, it is necessary to calculate the quantiles of the data. This method ensures that the class balancing will be as accurate as possible, unlike dwSAX which is subject to approximation errors, since it estimates the Probability Density Function of the data.

Figure 1 shows a visual comparizon of SAX (1a) and qSAX (1b). The dotted, grey, line shows the expected distribution. In this example, it is clear that SAX performs poorly for the symbol/class **d**, while qSAX forces a smaller range for **a** to **c** aiming to balance the class distribution[1].



(a) SAX representation. **(b) qSAX representation.**

Figure 1: Example of representation, SAX and qSAX.

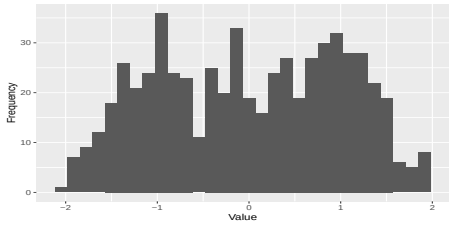
4. Experiments

For these experiments, we used two different datasets. First, the dataset Synthetic Control which contains 60 time series with 600 observations each (with different frequency distributions) and, second, a real World dataset on COVID-19. In the first experiment, we use a non-normal distribution, from the Synthetic Control series, aiming to get the class distribution for each method (SAX, dwSAX and qSAX). In the second experiment, we trained classification models, using the letters of the alphabet as classes. As a visual example of the problem, we used regression trees.

4.1. Experiment I

This experiment aims to analyse the amount of classes generated by SAX, dwSAX, and qSAX when supplied a non-normal time series from the Synthetic Control dataset. Figure 2a shows the series statistical distribution while Figure 2b shows that the only method that was not able to guarantee class balancing was SAX. This was expected, given that SAX breakpoints are retrieved using a normal distribution. Furthermore, we see that qSAX has perfectly distributed the classes. This will always happen if the number of observations is a multiple of the size of the alphabet and there are no value repetitions in the data. Otherwise, qSAX will force the distribution to be as equiprobable as possible.

[1]dwSAX generates an intermediated version of SAX and qSAX.



(a) A histogram of the chosen series.

	a	b	c	d
SAX	194	105	107	194
qSAX	150	150	150	150
dwSAX	156	148	143	153

(b) Number of classes per method.

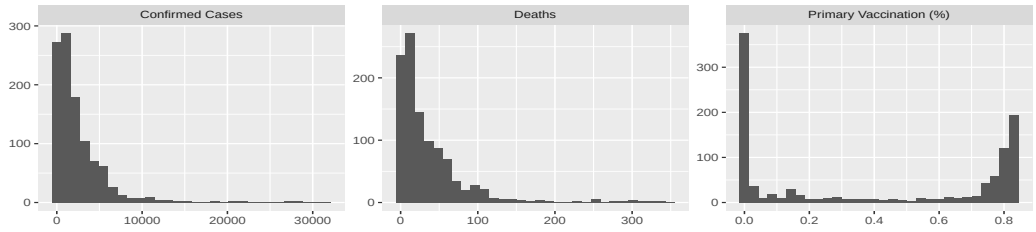
Figure 2: Synthetic Control dataset

4.2. Experiment II

Once we fixed the distribution problem, we used classification via random tree to generate rules for specific classes. This experiment aims to compare SAX, dwSAX and qSAX results in simple classification tree models. As a use case, we used the data of COVID-19 in the State of Rio Grande do Sul [Silveira and Assunção 2023]. The goal is to have an option to classify specific ranges of the series to be able to further mine patterns and anomalies regarding different related variables. Figure 3 shows the distribution of the selected series.

These series shows the number of confirmed cases (left) and deaths (middle) per day, as well as the percentage of people with the complete primary vaccination schedule on each day, *i.e.*, people who took either the single dose, or the first and second doses of the vaccine (right).

Figure 3: Histogram of COVID-19 cases, deaths and vaccination in RS, Brazil, series.



To compare the democratization of SAX, qSAX and dwSAX, we chose the last attribute, *Primary Vacination*, as *target*. This attribute is the most different from a bell curve, thus, it should be the most appropriated to illustrate the application. Furthermore, we specifically want to classify the intermediate values between the moment before the start of the vaccination campaign and the moment when it reached stability.

For the sake of clarity, we used an alphabet of cardinality 4 to generate 4 classes. Furthermore, as the *dataset* is relatively small, we used all the observations. Formally, it is like choosing $w = n$ for the PAA step.

Table 1a show the distribution of symbols (i.e., classes) using SAX. It is clear that the classes **b** and **c** are underrepresented. These are precisely the intermediate classes that we want to classify. Furthermore, we noticed that the **b** class is overrepresented in the attributes *Confirmed Cases* (CC) and *Deaths* (D). Therefore, the discretization by SAX was not able to guarantee class balancing and the models generated from this discretization are likely to be biased by the majority classes.

Tables 1a, 1b and 1c show us that both extensions successfully provide a more balanced output. For the attributes *confirmed cases* (CC) and *deaths* (D), the results of qSAX and dwSAX are statistically similar. However, we see that qSAX has reserved more observations for the intermediate attributes of *Primary Vacination* (PV) than dwSAX, providing a better balance.

(a) SAX

	PV	CC	S
a	456	86	116
b	99	655	608
c	66	200	213
d	456	136	140

(b) qSAX

	PV	CC	S
a	358	271	275
b	182	268	270
c	274	269	274
d	263	269	258

(c) dwSAX

	PV	CC	S
a	381	290	275
b	160	265	282
c	208	250	250
d	328	272	270

Table 1: Number of symbols by attribute.

It is worth noting that the **a** class of *PV* is overrepresented in both discretizations. This occurs because there are a large number of equal values in the first quartile, which are the zeroes for the period in which no one was vaccinated. Indeed, in the case of the qSAX discretization, the **a** class refers only to this period, which is the period of 358 days from the registration of the first cases until the start of vaccination.

After creating the symbolic representations of the time series, 70% of the observations are used for the training stage of the decision tree inference algorithm. For each SAX method studied here, 100 decision trees are obtained and corresponding accuracy values are computed. Table 2 shows the intervals corresponding to the SAX breakpoints and its versions. Decision trees are obtained using the *rpart* R package, with default values.

Table 2: Resulting intervals.

Class	SAX	qSAX	dwSAX
a	$[0, 0.15]$	$[0, 0]$	$[0, 0.03]$
b	$(0.15, 0.40]$	$(0, 0.32]$	$(0.03, 0.38]$
c	$(0.40, 0.54]$	$(0.32, 0.80]$	$(0.38, 0.77]$
d	$(0.54, 0.83]$	$(0.80, 0.83]$	$(0.77, 0.83]$

Tables 3b and 3c, corresponding to the results from qSAX and dwSAX respectively, show that classes **b** and **c** were classified as expected. However, as a result of the qSAX discretization, a higher number of intermediary classes was reached. Besides, it is worth mentioning that these models have low accuracy due to the fact that we have not used time as an attribute. If so, time alone would be sufficient to predict the percentage of people with complete vaccination. Finally, the average accuracy of each model was 0.675875, 0.5643614 and 0.5431056, for SAX, qSAX, and dwSAX, respectively. As we can see, models related to SAX are more accurate. However, this accuracy is erroneous, as it is only considering two out of four classes.

5. Conclusion

This work shows a simple and effective solution to the SAX class imbalance problem when the underlying distribution of the data is not normal. As shown in our experiments, qSAX has a solid capacity to generate near-equal class output. Such a feature is more reliable than some variations that address the same problem because it forces the class output to be equally distributed by narrowing the value range on the generated alphabet.

(a) SAX

	a	b	c	d
a	94.81	16.67	0	14.53
b	0	0	0	0
c	0	0	0	0
d	41.19	12.33	19	121.47

(b) qSAX

	a	b	c	d
a	49.48	3.66	9.21	21.69
b	25.85	43.2	5.16	0
c	27.18	7.14	50.85	18.68
d	4.49	0	16.78	37.63

(c) dwSAX

	a	b	c	d
a	54.06	19	17.11	14.89
b	24.45	24.22	3.81	0
c	16.26	4.46	23.46	9.97
d	19.23	0.32	17.62	73.14

Table 3: Average confusion matrix.

Despite the stable generation of a near-uniform set of symbols, qSAX is not a substitute for SAX in all scenarios. In fact, the algorithm should only be used in cases where the data distribution makes SAX generate an imbalanced set of symbols. Moreover, in some specific cases, variations such as dwSAX can also be used with satisfactory results. As a future work, we shall explore options for dynamic inputs, since sometimes it is desired to classify data using a specific range of values.

Acknowledgement

This work is funded by FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul), process number: 22/2551-0000390-7.

References

Bountrogiannis, K., Tzagkarakis, G., and Tsakalides, P. (2021). Data-driven kernel-based probabilistic sax for time series dimensionality reduction. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 2343–2347.

Bountrogiannis, K., Tzagkarakis, G., and Tsakalides, P. (2022). Distribution agnostic symbolic representations for time series dimensionality reduction and online anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Espejo, P. G., Ventura, S., and Herrera, F. (2010). A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(2):121–144.

Kloska, M. and Rozinajova, V. (2020). Distribution-wise symbolic aggregate approximation (dwsax). In *Intelligent Data Engineering and Automated Learning – IDEAL 2020: 21st International Conference, Guimaraes, Portugal, November 4–6, 2020, Proceedings, Part I*, page 304–315, Berlin, Heidelberg. Springer-Verlag.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, page 2–11.

Lin, J., Keogh, E., Wei, L., and Lonardi, S. (2007). Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144.

Niaz, N. U., Shahariar, K. N., and Patwary, M. J. A. (2022). Class imbalance problems in machine learning: A review of methods and future challenges. In *Proceedings of the 2nd International Conference on Computing Advancements*, ICCA '22, page 485–490, New York, NY, USA. Association for Computing Machinery.

Silveira, E. and Assunção, J. (2023). Coronavirus - Time Series - Vaccination by Attribute - RS, Brazil. Available at: https://doi.org/10.7910/DVN/KM5FOX.