Data-driven symbolic representations for high-level time series analysis

Konstantinos Bountrogiannis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Prof. *Panagiotis Tsakalides*
Thesis Supervisor: Dr. *George Tzagkarakis*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been funded by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

Data-driven symbolic representations for high-level time series analysis

Thesis submitted by
Konstantinos Bountrogiannis
in partial fulfillment of the requirements for the
Master's of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Konstantinos Bountrogiannis

Committee approvals: _____
Panagiotis Tsakalides
Professor, Thesis Advisor

_____
George Tzagkarakis
Principal Researcher, Committee Member

_____
George N. Karystinos
Professor, Committee Member

_____
Yannis Tzitzikas
Assoc. Professor, Committee Member

Departmental approval: _____
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, July 2020

Data-driven symbolic representations for high-level time series analysis

Abstract

The systematic collection of data has become an intrinsic process of all aspects in modern life. From industrial to healthcare machines and wearable sensors, an unprecedented amount of data is becoming available for mining and information retrieval. The ever-increasing volume and complexity of time series data necessitate efficient dimensionality reduction for facilitating data mining tasks. Symbolic representations, especially the family of symbolic aggregate approximations (SAX), have proven very effective in compacting the information content of time series while exploiting the wealth of search algorithms used in bioinformatics and text mining communities. However, typical SAX-based techniques rely on a Gaussian assumption for the underlying data statistics, which often deteriorates their performance in practical scenarios. To overcome this limitation, this thesis introduces a method that negates any assumption on the probability distribution of time series, by means of kernel density estimation (KDE) and Lloyd-Max quantization. Experimental evaluation on real-world datasets demonstrates the superiority of the proposed method, when compared against the conventional SAX and an alternative data-adaptive SAX-based method. Finally, in the present thesis, the proposed dimensionality reduction method is utilized to provide compact representations of time series for the purposes of anomaly detection. To this end, a computationally efficient, yet highly accurate, framework for anomaly detection of streaming data in lower-dimensional spaces is developed, whereas alternative quantization schemes are explored and utilized for more accurate statistical inference.

Συμβολικές αναπαραστάσεις βάσει δεδομένων για ανάλυση χρονοσειρών σε υψηλό επίπεδο

Περίληψη

Η συστηματική συλλογή δεδομένων είναι πλέον μια εγγενής διαδικασία όλων των πτυχών της σύγχρονης ζωής. Από βιομηχανικά μηχανήματα έως μηχανήματα υγειονομικής περίθαλψης και φορητούς αισθητήρες, μια άνευ προηγουμένου ποσότητα δεδομένων διατίθεται για εξόρυξη και ανάκτηση πληροφοριών. Ο συνεχώς αυξανόμενος όγκος και η πολυπλοκότητα των δεδομένων χρονοσειρών απαιτούν αποτελεσματική μείωση των διαστάσεων των δεδομένων για τη διευκόλυνση των εργασιών εξόρυξης δεδομένων. Οι συμβολικές αναπαραστάσεις, ειδικότερα η οικογένεια των συμβολικών συναθροιστικών προσεγγίσεων (SAX), έχουν αποδειχθεί πολύ αποτελεσματικές για τη συμπίεση της πληροφορίας που περιέχεται στις χρονοσειρές, ενώ εκμεταλλεύονται τον πλούτο των αλγορίθμων αναζήτησης που χρησιμοποιούνται στις κοινότητες της βιοπληροφορικής και της εξόρυξης κειμένου. Ωστόσο, οι τυπικές τεχνικές που βασίζονται στην SAX υποθέτουν ότι τα υποκείμενα στατιστικά χαρακτηριστικά των δεδομένων είναι Γκαουσιανά, με αποτέλεσμα συχνά να επιδεινώνεται η απόδοσή τους σε πρακτικές εφαρμογές. Για να ξεπεραστεί αυτός ο περιορισμός, η διατριβή αυτή παρουσιάζει μια μέθοδο που αναιρεί οποιαδήποτε υπόθεση σχετικά με την κατανομή πιθανότητας των χρονοσειρών, μέσω εκτίμησης πυκνότητας με πυρήνα (KDE) και Lloyd-Max κβάντισης. Η πειραματική αξιολόγηση σε πραγματικά δεδομένα καταδεικνύει την ανωτερότητα της προτεινόμενης μεθόδου, σε σύγκριση με τη συμβατική SAX και μια εναλλακτική μέθοδο βασιζόμενη στη SAX, που λειτουργεί με απευθείας προσαρμογή στα δεδομένα. Τέλος, στην παρούσα διατριβή, η προτεινόμενη μέθοδος μείωσης διαστάσεων αξιοποιείται για να παρέχει συμπαγείς αναπαραστάσεις χρονοσειρών με στόχο την ανίχνευση ανωμαλιών. Για το σκοπό αυτό, αναπτύσσεται ένα υπολογιστικά αποτελεσματικό, αλλά πολύ ακριβές, πλαίσιο για ανίχνευση ανωμαλιών σε ροές δεδομένων σε χώρους λιγότερων διαστάσεων, ενώ εναλλακτικά σχήματα κβαντισμού διερευνώνται και χρησιμοποιούνται για πιο ακριβή άντληση στατιστικών συμπερασμάτων.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Time series representations for data mining

Representing and interpreting complex time-varying phenomena is a challenging task in several application domains. Such issues become even more demanding in view of the large volumes of time series data, emerging thanks to the advances of computing technologies. From industrial to healthcare machines and wearable sensors, an unprecedented amount of data is becoming available. Such examples, which are characterized by their temporal nature, belong to the class of time series data.

Formally, a time series is a collection of observations made chronologically. Adjacent points of time series data are typically highly correlated and hence many conventional statistical methods which traditionally dependent on the assumption that data samples are independent and identically distributed, are inapplicable.

Efficiently mining this data deluge necessitates the extraction of descriptive motifs in appropriate lower-dimensional spaces, which provide a meaningful, yet compact, representation of the original inherent information to be further employed for executing high-level tasks, such as event detection and classification.

One of the strongest benefits that data mining methods can gain from representation methods is dimensionality reduction. Here, dimensionality refers to the cardinality of the representation space of a data object, i.e. the number of values that describe the object. Indeed, current data objects contain large amount of information, either due to their time resolution (e.g. sequences from rapidly sampled data sources), or due to the multi-dimensionality of the data source itself (e.g. standard electrocardiographs collect 12 concurrent values per sample).

Moreover, the definition of appropriate similarity measure between time series

based on their representation is necessary for time series mining tasks. More precisely, the transformation of data objects to a lower-dimensional subspace should be, ideally, distance-preserving. This property would allow data mining tasks to perform equally well on the lower-dimensional space as in the high-dimensional. However, except when all data objects are equal when projected to some dimension, this is impossible. Due to this limitation, similarity measures defined on lower-dimensional subspaces can only approximately preserve distances. The degree of this approximation is the central property of similarity measures.

A milestone for the definition and utilization of such measures has been the GEneric Multimedia INdexIng (GEMINI) framework, introduced in [11]. This framework dictates a condition for defining similarity measures that are appropriate for similarity searching in databases. According to GEMINI, the similarity measure in the lower-dimensional space must lower-bound the objects' distance in the high-dimensional space. When this property is satisfied, it is proved that approximate queries (i.e. queries that return all objects with up to a maximum raw distance from the query object), in the lower-dimensional space, return no false dismissals. In advance, the tighter this bound is, the less false alarms are returned. Notice that the standard distance measure in the raw space is the Euclidean distance, i.e. the L^2 norm.

1.1.1 Numerical representations

The simplest form of dimensionality reduction is sub-sampling. Unfortunately, this method distorts the shape of the signal, without keeping any information for the lost samples. An improvement over simple sub-sampling is achieved by averaging the segments in-between the sampling time steps. The averaged segments preserve more accurately the signal's shape and are used to represent the time series. This method, introduced in [46], is encountered either as Segmented Means or Piecewise Aggregate Approximation (PAA) in the literature. PAA, in accordance with the GEMINI framework, lower bounds the distance in higher-dimensional spaces for all L^p norm distances. APCA [15] is an extented version of PAA, where the segment size is not constant but adapts to the signal temporal variation. The Clipped representation, a single-bit representation, is proposed in [36], in which each sample is represented by a bit depending on whether it is higher or lower than the mean value of the series. A distance measure that lower bounds the Euclidean distance is defined, as well.

The aforementioned methods either merge or reduce the time series samples in a piecewise manner. A different approach is to keep only the points that are the most

important for the shape of the series. The identification of these points is not a given. Perteptually Important Points (PIP) [5], a bottom-up method, suggests that the important points is those that maximize the absolute difference of the adjacent values. Using this methodology, only the local minima and maxima are considered as potential important points. PIP is mostly used in financial data, where indeed the temporal variation is the most informative characteristic. A related method is the Landmark model [34]. In this model, all points where the n-th order derivative is 0 are considered important points. This includes all local minima and maxima, where the first order derivative is 0. The decision about which derivative orders n to use is based on the nature of the data and the tradeoff between representation accuracy and dimensionality reduction.

1.1.2 Symbolic representations

Another class of representations is the symbolic representations, which transform the time series into a compact sequence of symbols. Among them, the symbolic aggregate approximation (SAX) [19, 20] has been one of the most commonly used time series representations. In particular, SAX first computes the PAA representation of the time series. Then, assuming that the time series follows the standard Gaussian distribution, the segmented means are quantized by mapping them to equiprobable intervals, where each interval is represented by a unique symbol. The final output is a sequence of symbols. In addition, a lower-bounding distance measure is defined, which is shown to bound the Euclidean distance tighter than most of the best performing dimensionality reduction methods.

Two other symbolic representations are the Piecewise Vector Quantized Approximation (PVQA) [27] and its multi-resolution extension Multiresolution Vector Quantized (MVQ) [28] approximation. These methods, instead of computing the mean values of the segments and then quantizing, they directly quantize the segments with the LGB vector quantization algorithm. Then, the time series is represented by the histogram (probability distribution) of the vector-codewords in the series. However, the histogram-based distance metric they utilize does not lower-bound some distance measure in the raw space.

1.1.3 Representations based on basis decomposition

All of the above methods represent the time series with yet another time series of reduced dimensionality. That is, the representation itself is a sequence of objects in chronological order. A completely different approach is to decompose the time

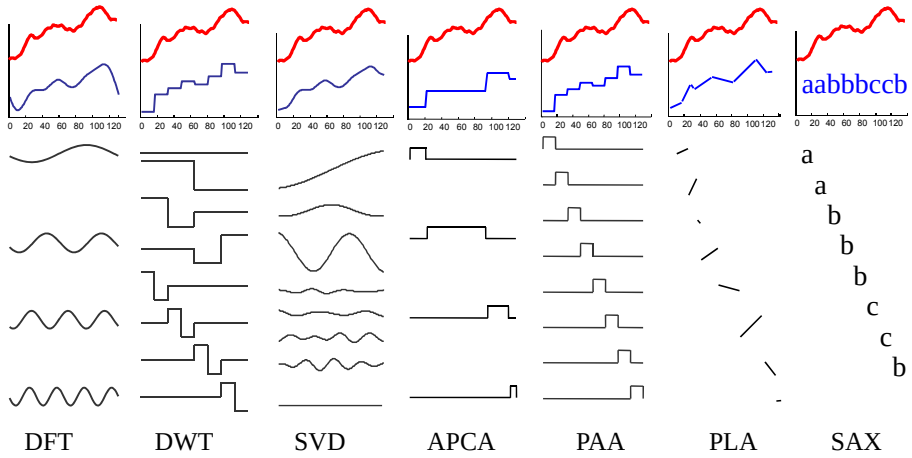


Figure 1.1: Common time series representations. Reprinted from Eamonn Keogh's tutorial "Machine Learning in Time Series Databases", AAAI 2011, retrieved from `https://www.cs.ucr.edu/~eamonn`. Reprinted with permission.

series over a basis. Essentially, the time series is decomposed into a set of coefficients, which reproduce the time series when used as coefficients for the linear combination of the basis vectors. These coefficients are used to represent the time series, whereas dimensionality reduction is imposed by nullifying the smallest of them.

Commonly used bases are those of the Discrete Fourier Transform (DFT) and the Discrete Wavelet Transform (DWT) [16], while more complex decompositions such as Singular Value Decomposition (SVD) [16] and Principal Component Analysis (PCA) [45] have been used but suffer highly from computational and memory complexity. When the chosen basis is orthogonal (which holds for the Fourier and most of the Wavelets bases), due to Parseval's theorem, the L^2 norm in the basis domain lower-bounds the L^2 norm in the time domain. SVD and PCA have the lower-bounding property as well.

1.2 Anomaly detection

Anomaly detection refers to the problem of finding abnormal patterns in data. The characteristics of normal behaviour can be either inferred by other data which are known to be normal, or deduced by the common behaviour of the the available data. Other names for anomaly are outlier, surprise, novelty and contaminant, where each of these names is used more often than the others in specific application domains.

Due to its prominent role in monitoring and predicting critical processes and phenomena, anomaly detection is performed in a plethora of distinct application scenarios employing both non-streaming and streaming data, such as network intrusion detection, fraud detection, detection of data abnormalities or instrumentation errors in the medical domain, novelty detection in textual data [4].

The fundamental problem of anomaly detection is defining the characteristics of the normal behaviour and where exactly are the boundaries between normal and abnormal. Many setbacks render this problem remarkably hard. For example, knowing all the normal cases is rarely possible. In advance, the boundary between normal and anomalous behaviour is often not precise and small fluctuations can either be random or indicate anomalous activity. Last but not least, anomalies must be characterized with respect to their neighbourhood in the data domain (temporal, spatial or other dimensions) they come up in. For example, in time series data, anomalies are subject to the specific time period they come up. Obviously, what is normal for every single point in the data domain cannot be specified precisely.

Anomaly detection methods, in accordance to machine learning algorithms in general, can be categorized as supervised, semi-supervised, or unsupervised. Supervised methods learn to detect anomalies based on a collection of data with exact knowledge of normal and anomalous instances. Semi-supervised learning is based on a collection of data with only normal instances. In unsupervised learning, there is no previous knowledge of what normal or anomalous data instances are. These methods assume that normal instances are far more frequent than anomalies in the data and thus can learn the characteristics of normal instances by the general behaviour of the data.

Several approaches have been adopted in developing anomaly detection methods. Classification-based techniques learn a strict boundary between normal and anomalous data in some feature space and classify accordingly. Nearest-neighbour techniques assume that anomalous data instances occur far from their neighbouring data, or in other words, their neighbourhood is sparse. Statistical methods model the data generator process as a stochastic process. With this approach, anomalies are instances of data with low probability to occur.

Statistical methods can be either parametric or non-parametric. Parametric methods presuppose a statistical model and learn the parameters from the data. On the other hand, non-parametric methods make no assumptions regarding the statistical structure of the data and attempt to determine the statistical characteristics in whole from the data itself.

1.3 Problem statement and Contribution

Focusing on the case of streaming data arriving in nearly real-time, necessitates the design of fast data mining algorithms, whereas edge processing applications impose additional computational constraints due to the limited power and memory resources available on-board small sensing devices. As a use-case example, anomaly detection plays a key role in a wide range of applications, and has been studied extensively. However, many anomaly detection methods are unsuitable in practical scenarios, due to the high data rate and limited devices resources.

The family of symbolic aggregate approximation (SAX) methods [19] has a prominent role among the several existing motif discovery techniques. Due to its conceptual simplicity and computational tractability, SAX has been widely used in monitoring, processing and mining data of numerous sources, including physiological data [32], smart grids [43], building systems [30], and stock market [1].

However, SAX and its variants [38, 23, 40, 21] rely on a Gaussian assumption for the underlying data statistics, which often deteriorates their performance in practical scenarios. Indeed, although typical SAX-based techniques can lead to high-precision results in the case of data characterized by Gaussian statistics, however, their performance may degrade dramatically in more general cases. In practical scenarios, where the underlying probability distribution of a time series deviates significantly from a Gaussian, or when the distribution changes across time, then, the previous SAX-based techniques are not capable of adapting to the time-evolving statistics. As a result, their low-dimensional representation and motif interpretation power diminishes.

This thesis considers the problem of developing a time series representation which is highly efficient, highly adaptive and being easily adopted by existing data mining applications. To this end, a novel symbolic representation is developed, built upon the framework of SAX. The novel method, contrary to conventional SAX methods, is non-parametric, and is shown to exhibit comparatively superior performance. In advance, the representation method preserves the statistical characteristics of the raw data and hence it is compatible with existing statistical anomaly detection (and other data mining) methods. In order to demonstrate its performance, an unsupervised anomaly detection method, with focus on streaming data, is incorporated. Due to the reduced dimensionality of the symbolic representation, power and memory requirements are decreased, whilst the increased accuracy of the representation retains the detector's precision in the lower-dimensional space.

Chapter 2

Background

Fundamentally, a time series is a sequence of randomly generated values. As such, time series analysis is conceptually analysis of stochastic processes, as will be discussed here. The first section of this chapter provides the necessary definitions and methods that are used throughout the text. The second section covers the symbolic aggregate approximation of a time series in details, which will be used later to transform a continuous time series into a discrete one.

2.1 Preliminaries

In the context of experiments, a random variable is a real-valued function of the experimental outcome. The set of all possible outcomes is called the sample space of the random variable. A random variable is called discrete if its sample space is finite or countably infinite, whereas it is called continuous if it is uncountably infinite [3, Sec. 2].

A discrete random variable X has an associated probability mass function (pmf) $P_X(x)$, which gives the probability of each value x that X can take. A continuous random variable Y has an associated probability density function (pdf) $f_Y(y)$, of which the integral within a specific interval of y values gives the probability that Y will take a value inside this interval. For every (discrete or continuous) random variable X, a cumulative distribution function (cdf) $F_X(x)$ can be defined, which gives the probability that X will take a value lower than or equal to x.

A stochastic process is a mathematical model of a probabilistic experiment that evolves in time and generates a sequence of numerical values. Each numerical value in the sequence is modeled by a random variable, so a stochastic process is simply a (finite or infinite) sequence of random variables [3, Sec. 6].

It is easy to see the relation between stochastic processes and time series. A

stochastic process is a sequence of random variables and a time series is a sequence of randomly generated values. That is, a time series is *an implementation of a stochastic process*, or, *a snapshot of an ongoing stochastic process.*

This is the reason that tools and definitions from stochastic processes (e.g. stationarity, ergodicity, certain models such as Markov chains) are widely used and have proven particularly useful in time series analysis. An in-depth review of this concept is out of the scope of this work. At this point however, it suffices to infer that analysis and processing of time series with tools and methods from probability theory is, mathematically, a valid approach. Now let us consider some concepts from the field of information theory.

2.1.1 Elements of Information theory

The majority of the content in this subsection is taken from [7], in which the proofs of the following lemmas and theorems can be found.

Let X be a discrete random variable with probability mass function $P_X(x)$, $x \in \mathcal{X}$, where $\mathcal{X}$ is the sample space, i.e. the set of all possible outcomes, of the random variable X. The sample space is also named the alphabet of the random variable and its cardinality $|\mathcal{X}|$ is the alphabet size.

Definition. The entropy $H(X)$ of a discrete random variable is defined by

$$H(X) = -\sum_{x \in X} P_X(x) \log P_X(x) \ , \tag{2.1}$$

with the convention that $0 \log 0 = 0$.

When the base of log is 2, entropy is expressed in bits, when the base is e, entropy is expressed in nats, and when the base is 10, entropy is expressed in bans.

Entropy is a measure of uncertainty. It is a measure of the amount of information (e.g. bits, when log is to the base of 2) required on average to describe the random variable. It is easy to prove the range and the distribution that maximizes entropy:

Lemma 2.1.1.

$$0 \leq H(X) \leq \log |\mathcal{X}| \tag{2.2}$$

Lemma 2.1.2. *The entropy $H(X)$ is equal to $\log |\mathcal{X}|$ if and only if X has a uniform distribution.*

Statistical properties, such as the probability distribution, are useful for comparing two time series. The fundamental notion is that when the time series are

similar in the probability domain, then the time series should be similar in the time domain, too. Quantifying the similarity of distributions, however, is not straightforward. For this purpose, the f-divergences are used.

Definition. Let P_X and P_Q be two discrete probability distributions, defined on the same sample space $\mathcal{X}$. Additionally, let f be a convex function with $f(1) = 0$. Then, the discrete f-divergence of P_X and P_Q is defined as

$$D_f(X \parallel Q) = \sum_{x \in \mathcal{X}} P_Q(x) f\left(\frac{P_X(x)}{P_Q(x)}\right) . \tag{2.3}$$

Notice that an f-divergence is not necessarily symmetric, that is, the relation $D_f(P \parallel Q) = D_f(Q \parallel P)$ does not always hold, nor does the triangular inequality. This means that f-divergences are not proper distances and should be carefully handled accordingly.

Particular functions f are used for different applications, taking advantage on their unique properties. The most common f-divergence, which originates from the field of telecommunications but is widely used in machine learning as well, is the Kullback-Leibler (KL) divergence, also known as the *relative entropy*.

Definition. Let P_X and P_Q be two discrete probability distributions, defined on the same sample space $\mathcal{X}$. The Kullback-Leibler (KL) divergence is defined as

$$D_{KL}(X \parallel Q) = \sum_{x \in \mathcal{X}} P_X(x) \log \frac{P_X(x)}{P_Q(x)} , \tag{2.4}$$

with the conventions that $0 \log \frac{0}{0} = 0$, $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$.

It is easy to notice that the KL divergence equals the expected value of $\log \frac{P_X(x)}{P_Q(x)}$ in terms of $P_X(x)$:

$$D_{KL}(X \parallel Q) = \mathbb{E}_{x \sim X}\left[\frac{P_X(x)}{P_Q(x)}\right] \tag{2.5}$$

A symmetric version of KL divergence is Jeffrey's divergence.

Definition. Let P_X and P_Q be two discrete probability distributions, defined on the same sample space $\mathcal{X}$. Jeffrey's divergence is defined as

$$\begin{aligned} D_J(X \parallel Q) &= D_{KL}(X \parallel Q) + D_{KL}(Q \parallel X) \\ &= \sum_{x \in \mathcal{X}} (P_X(x) - P_Q(x)) \log \frac{P_X(x)}{P_Q(x)} . \end{aligned} \tag{2.6}$$

The following properties hold for the above f-divergences.

Lemma 2.1.3.

$$D_{KL}(X \parallel Q) \geq 0 \tag{2.7}$$

$$D_J(X \parallel Q) \geq 0 \tag{2.8}$$

with equalities in (2.7) *and* (2.8) *if and only if* $P_X(x) = P_Q(x) \ \ \forall\ x \in \mathcal{X}$.

In information theory, the application of the law of large numbers results to the asymptotic equipartition property (AEP).

Theorem 2.1.1 (AEP)**.** *If* $X_1, \ldots, X_n$ *are i.i.d.* $\sim P_X(x)$*, then*

$$-\frac{1}{n} \log p(X_1, X_2, \ldots, X_n) \to H(X) \ , \tag{2.9}$$

where the arrow denotes convergence in probability.

The equation (2.9) can be re-written as:

$$p(X_1, X_2, \ldots, X_n) \to 2^{-nH(X)} \ , \tag{2.10}$$

having assumed that entropy is computed with base-2 logarithms.

AEP leads us to the definition of the typical set, which contains the sequences with sample entropy close to the true entropy. Then, the sequences in the typical set are most likely to appear by sampling a random variable.

Definition. The *typical set* $A_\epsilon^{(n)}$ with respect to $P_X(x)$ is the set of sequences $(X_1, X_2, \ldots, X_n) \in \mathcal{X}^n$ with the property

$$2^{-n(H(X)+\epsilon)} \leq p(X_1, X_2, \ldots, X_n) \leq 2^{-n(H(X)-\epsilon)} \ . \tag{2.11}$$

The following properties of the typical set hold due to AEP:

Theorem 2.1.2. *For any given* $\epsilon > 0$*, there exists a sufficient large* n *such that,*

- *If* $(X_1, X_2, \ldots, X_n) \in A_\epsilon^{(n)}$*, then* $H(X) - \epsilon \leq -\frac{1}{n} \log p(X_1, X_2, \ldots, X_n) \leq H(X) + \epsilon$.
- $Pr\left\{A_\epsilon^{(n)}\right\} > 1 - \epsilon$.
- $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$.
- $|A_\epsilon^{(n)}| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$.

Theorem 2.1.2 implies that, for sufficiently large number of samples, all elements in the typical set are nearly equiprobable (property 1), the typical set has

probability close to 1 (property 2), and the cardinality of the typical set is nearly $2^{nH(X)}$ (properties 3-4).

One significant consequence of AEP is related with data compression. Data compression can be achieved by assigning short descriptions to the most frequent outcomes of the data source, and necessarily longer descriptions to the less frequent outcomes [7].

Definition. A *source code* C for a random variable X is a mapping from its sample space $\mathcal{X}$ to D^*, the set of finite-length strings of symbols from a D-ary alphabet. Let $C(x)$ denote the codeword corresponding to x and let $l(x)$ denote the length of $C(x)$.

The relation between the distribution $P_X(x)$ and the expected length $L(C)$ of a source code C of X is given by

$$L(C) = \mathbb{E}_{x\sim X} l(x) = \sum_{x\in\mathcal{X}} P_X(x) l(x) \ . \tag{2.12}$$

A source code is *optimal* if its expected length is the minimum possible for the random variable.

Since there are $\leq 2^{n(H(X)+\epsilon)}$ sequences in the typical set, they can be indexed using no more than $\lceil n(H(X)+\epsilon)\rceil \leq n(H(X)+\epsilon)+1$ bits. Similarly, the other sequences can be indexed using no more than $n\log|\mathcal{X}|+1$ bits which is longer than the index of the typical set because $H(X)\leq \log|\mathcal{X}|$. Since the fraction of the sequences that are not in the typical set diminishes as n grows large, the following theorem can be proved:

Theorem 2.1.3. *Let X^n be i.i.d. $\sim P_X(x)$. Let $\epsilon > 0$. Then there exists a code that maps the sequences X^n into binary strings such as the mapping is invertible and*

$$E\left[l(X^n)\right] \leq nH(X) + \epsilon \tag{2.13}$$

for n sufficiently large.

Theorem 2.1.3 implies that the sequences X^n can be coded using $nH(X)$ bits on average and is the principal theoretical result for data compression.

The next theorem relates the KL divergence with the information loss that is induced when assuming an approximate distribution $P_{\hat{X}}(x)$ instead of $P_X(x)$ when source coding a random variable.

Theorem 2.1.4. *The expected description length of the random variable X by a D-ary alphabet, assuming optimal coding under the probability mass function $P_{\hat{X}}(x)$, satisfies*

$$H(X) + D_{KL}(X \parallel \hat{X}) \leq L(C) < H(X) + D_{KL}(X \parallel \hat{X}) + 1 \ , \tag{2.14}$$

where the logarithm bases are D.

Theorem 2.1.4 states that the lack of knowledge of the true pmf of a random variable costs additional storage space for the description of its samples which is quantified by the KL divergence. It is a loss of information that needs to be recovered with additional symbols. By combining Lemma 2.1.3 and Theorem 2.1.4, the inequalities change as follows when using the true pmf.

Theorem 2.1.5. *The expected description length of the random variable X by a D-ary alphabet, assuming optimal coding under the probability mass function $P_X(x)$, satisfies*

$$H(X) \leq L(C) < H(X) + 1 \ , \tag{2.15}$$

where the logarithm bases are D.

2.1.2 Kernel density estimation

Several data processing techniques incorporate the probability distribution of the data source. However, the distribution is rarely known. Rather, only a set of samples is given à priori (in other words, a training set), or the collection of data to be processed itself. In this situation, the distribution must be estimated.

This work considers real-valued continuous time series, with samples either from the set of real numbers or from a specific interval of it. Hence, the distribution of the data is given by a probability density function (pdf). The problem we address here, i.e. the estimation of a pdf, is called density estimation. The simplest method of density estimation is the histogram, which is described next.

Let a random variable X with pdf $f_X(x),\ x \in \mathcal{X}$, where $\mathcal{X}$ is the sample space of X. Without loss of generality, let $\mathcal{X}$ be the interval $[a, b),\ a, b \in \mathbb{R}$. Now, let $x_1, \ldots, x_n$ be n observations of X.

The first step to compute the histogram is to partition the sample space into the equi-length bins $[a, a+h), [a+h, a+2h), \ldots, [b-h, b)$ of length h. Hence, any point in the sample space belongs to one bin in the partitioned space. Then, the histogram is an estimation $\hat{f}_{\mathrm{H}}(x)$ derived from the observations $x_1, \ldots, x_n$, defined as

Definition.

$$\hat{f}_{\mathrm{H}}(x_0) = \frac{1}{nh} \sum_i \mathbb{1}_{x_i \in \text{ bin of } x_0} \ . \tag{2.16}$$

In the general case where the sample space is not bounded, as we assumed above, the bounds are set empirically in order to properly define the bins.

The histogram is easy and fast to compute and is extremely useful for the presentation of data. However, it has two main drawbacks. Firstly, the choice of the empirical bounds of the sample space might severely alter the final estimation. Secondly, when the estimated distribution is needed by other processing methods, the discontinuity of the histogram causes difficulties.

For the reasons mentioned above, an improved density estimation technique is widely used, known as kernel density estimation (KDE) [33]. The histogram counts the neighbours of each sample in the same bin. This means that all neighbours inside the bin are given the same weight, and any other sample is disregarded. In the case of KDE, all samples affect the estimated distribution for any point in the sample space, according to their relative position.

Essentially, the estimated distribution $\hat{f}_{\mathrm{KDE}}$ is the summation of a kernel function centered at each observed sample, as defined below.

Definition.

$$\hat{f}_{\mathrm{KDE}}(x_0) = \frac{1}{nh} \sum_i K\left(\frac{x_i - x_0}{h}\right) \ , \tag{2.17}$$

where h is the smoothness parameter, which controls the width of the kernel around x_0 and K is the kernel function, which controls the weight given to the points in the neighbourhood of x_0.

A visual comparison of the two density estimation methods is available in Fig. 2.1.

The KDE method requires the assignment of a kernel function K and a value for the smoothness parameter h. From these two parameters, it has been observed that the choice of the smoothness parameter has a bigger impact on the final result. The choice of kernel function is significant when the number of available samples is small, but as it get bigger, most kernels perform similarly.

In the domain of estimation theory, two particular metrics evaluate the performance of an estimator, such as KDE, for a specific target parameter. The first is the bias $B(\hat{\theta})$, which is the difference of the estimator's expected estimation $\hat{\theta}$ and the true parameter θ being estimated. The second metric is the variance $\mathrm{Var}(\hat{\theta})$ of the estimation.

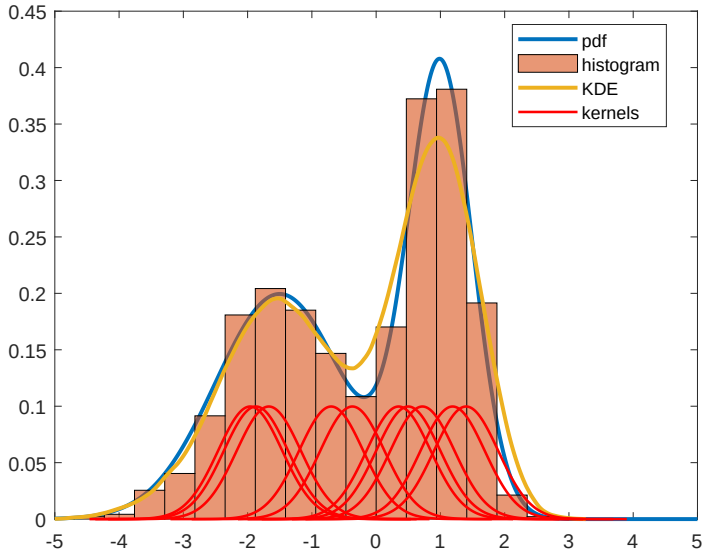


Figure 2.1: Estimation of a mixture of two Gaussian distributions from 1000 observed samples. The blue line is the target pdf, the orange bars are the histogram and the yellow line is the KDE estimated distribution using Gaussian kernels. The kernels are shown in red above the observed values (here only 10 of them are depicted). The histogram is not continuous and overfits to the observed samples. On the other hand, KDE is continuous and approximates better the target pdf.

Definition. The bias and the variance of an estimator for the target parameter θ are defined as follows.

$$B(\hat{\theta}) = \mathbb{E}[\hat{\theta}|\theta] - \theta \tag{2.18}$$

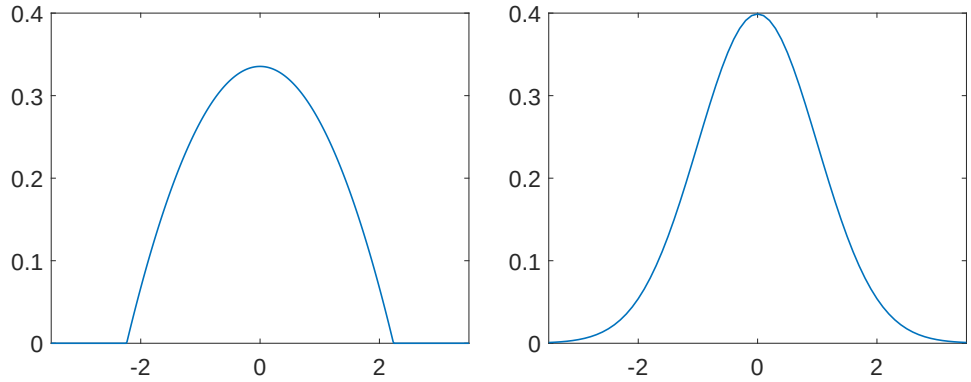
$$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2|\theta] \tag{2.19}$$

The estimator's mean square error (MSE) of the parameter θ can be expressed in terms of its bias and variance as:

$$\text{MSE}(\hat{\theta}) = (B(\hat{\theta}))^2 + \text{Var}(\hat{\theta}) \ . \tag{2.20}$$

For the case of KDE, under some assumptions which are true for the majority of kernel functions, the asymptotic MSE (AMSE), namely the MSE as the number of available samples n grows large, is approximated as [33]:

$$\text{AMSE}(\hat{f}_{KDE}(x)) = \frac{1}{nh} f(x) \int_{-\infty}^{+\infty} K^2(y)dy + h^4 \frac{k_2^2}{4} (f''(x))^2 + o(h^4) + o(\frac{1}{nh}) \ , \tag{2.21}$$



(a) Epanechnikov kernel (b) Normal kernel

where f is the target pdf and k_2 is a constant that depends on the kernel function K.

Integrating the AMSE over the sample space yields the asymptotic mean integrated squared error (AMISE):

$$\text{AMISE}(\hat{f}_{KDE}) = \frac{1}{nh}\int_{-\infty}^{+\infty} K^2(y)dy + h^4\frac{k_2^2}{4}\int_{-\infty}^{+\infty}(f''(x))^2dx \ . \tag{2.22}$$

The optimal kernel function, in terms of AMISE, is proved to be the Epanechnikov kernel [10] (Fig. 2.2a). Another frequently used kernel is the Gaussian kernel (Fig. 2.2b), which is not optimal in the general case, but performs better when the underlying distribution is close to a mixture of Gaussians.

Definition.

$$\text{Epanechnikov kernel: } K(x) = \begin{cases} \frac{3}{4\sqrt{5}}(1-\frac{x^2}{5}) & , \text{ for } |x| \le \sqrt{5} \\ 0 & , \text{ for } |x| > \sqrt{5} \end{cases} \tag{2.23}$$

$$\text{Gaussian kernel: } K(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2} \tag{2.24}$$

A key difference between those two kernels is that the Epanechnikov kernel has a finite support, from $-\sqrt{5}$ to $+\sqrt{5}$, whereas the Gaussian has an infinite support.

As already mentioned, the smoothness parameter h has a greater impact on the estimator's performance. A very low smoothness parameter results to a distribution with narrow spikes above the observed samples, in other words it overfits to the observed samples. A very high smoothness parameter results to an over-smoothed distribution that does not represent accurately the observed samples, nor the target

pdf. The optimal smoothness parameter h_{opt}, in terms of minimum AMISE, can be easily derived from (2.22):

$$h_{opt} = \left[\frac{\int_{-\infty}^{+\infty} K^2(y)dy}{nk_2^2 \int_{-\infty}^{+\infty} (f''(x))^2 dx} \right]^{1/5} \tag{2.25}$$

The equation in (2.25) would provide the optimal smoothness parameter easily, but unfortunately it depends on the target pdf itself. Because in actual implementations of an estimator, the knowledge of the target pdf is not possible, in [39] it is suggested that a known family of distributions can be used to assign a value for the term $f''(x)$. The most frequent candidate for distribution assignment is the Gaussian. If the underlying process indicates that its distribution is similar to some other known distribution, then it will give better results if it is used instead. In the case of the Gaussian assumption, the following approximation is derived:

$$h_{opt}^{\mathcal{N}} = \hat{\sigma} \cdot \left[\frac{8\pi^{1/2} \int_{-\infty}^{+\infty} K^2(y)dy}{3nk_2^2} \right]^{1/5}, \tag{2.26}$$

where $\hat{\sigma}$ is an approximation of the standard deviation and can be computed as the standard deviation of the given samples, or with some other more robust method.

For the two kernels introduced previously, the following calculations can be used in the nominator of (2.26).

$$\text{Gaussian kernel:} \quad \int_{-\infty}^{+\infty} K^2(y)dy = \frac{1}{2\sqrt{\pi}} \quad , \quad k_2 = 1 \tag{2.27}$$

$$\text{Epanechnikov kernel:} \quad \int_{-\infty}^{+\infty} K^2(y)dy = \frac{3}{5} \quad , \quad k_2 = \frac{1}{5} \tag{2.28}$$

Plugging the values from (2.27)-(2.28) in (2.26) provides the following values for optimal smoothness parameters:

$$\text{Gaussian kernel:} \quad h_{opt}^{\mathcal{N}} = 1.0592 \cdot \hat{\sigma} n^{-1/5} \tag{2.29}$$

$$\text{Epanechnikov kernel:} \quad h_{opt}^{\mathcal{N}} = 2.3449 \cdot \hat{\sigma} n^{-1/5} \tag{2.30}$$

It is important to note that the above values should be handled only as a good starting point for further adjustment, because every data source differs.

Apart from the accuracy of the estimation, KDE's performance is high in terms of convergence speed, too, given that the target distribution is smooth [37]. This

is important when the computational resources are low or when the number of available samples is not large.

2.1.3 Lloyd-Max quantization

Quantization refers to a mapping from a set with many, possibly infinite, elements to another set with fewer elements. This process is essential for many applications that involve computing systems, due to their digital nature. Typical examples are analog-to-digital converters, communication systems and lossy data compression. Quantization methods must specify both the mapping and the values of the elements in the smaller set, which are used to represent the values of the input set.

The expected distance between the input and output values is the distortion. Sufficient criteria for a quantizer to have minimum distortion were derived independently in [25] and [22] by Max and Lloyd, respectively. Because the criteria have no analytical solutions, they are generally solved through minimization algorithms. The simplest algorithm, derived by the authors of [25] and [22], is called the Lloyd-Max quantizer, which is basically an alternating minimization algorithm. Below, the basic principles of Lloyd-Max quantization are explained.

Let a partition $\{X_1, \ldots, X_k\}$ of the input class $\mathcal{X}$, where X_i's are disjoint subsets with arbitrary number of input elements from $\mathcal{X}$. Let a set of codewords (called *codebook* or *alphabet*) $\{c_1, \ldots, c_k\}$. Exactly one codeword is assigned to all elements of exactly one subset of the input partition with the quantization mapping $Q(\cdot)$,

$$Q(x_j) = c_i \ , \quad \forall \ x_j \in X_i \ . \tag{2.31}$$

The number k of quantization intervals is a pre-defined parameter of the quantization process.

Define a distance measure $d(x - Q(x))$ between an input element and the codeword assigned to it. Denote as $f(x)$ the underlying probability density function of the input class $\mathcal{X}$ and denote as b_i and b_{i+1} the bounds of the subset X_i. The distortion D is defined as the expected value of d:

$$D = \mathbb{E}_f[d(x - Q(x))] = \sum_{i=1}^{k} \int_{b_i}^{b_{i+1}} d(x - c_i) f(x) dx \tag{2.32}$$

The necessary criteria for minimizing the distortion function D are derived by calculating the partial derivatives with respect to b_i's and c_i's and setting them to zero. With simple calculus, the necessary criteria are given by the following equations:

Algorithm 1 Lloyd-Max Quantizer

1: Initialize all codewords c_i, $i = 1, \ldots, k$
2: Initialize sample space: $b_1 \leftarrow -\infty$, $b_{k+1} \leftarrow +\infty$
3: **while** stopping criteria are not met **do**
4: $\quad b_i \leftarrow (c_i + c_{i-1})/2$, $\quad$ for $i = 2, \ldots, k$
5: $\quad c_i \leftarrow \frac{\int_{b_i}^{b_{i+1}} x f(x) dx}{\int_{b_i}^{b_{i+1}} f(x) dx}$, $\quad$ for $i = 1, \ldots, k$

$$d(b_i - c_{i-1}) = d(b_i - c_i) \ , \quad i = 2, \ldots, k \tag{2.33}$$

$$\int_{b_i}^{b_{i+1}} d'(x - c_i) f(x) dx = 0 \ , \quad i = 1, \ldots, k \tag{2.34}$$

The criteria in (2.33) and (2.34) are necessary but not sufficient. In order to be sufficient, the Hessian matrix of the distortion function must be positive definite.

For the special case where the distance measure is the Euclidean distance, $d(x - c_i) = (x - c_i)^2$, the equations (2.33) and (2.34) become

$$b_i = (c_i + c_{i-1})/2 \ , \quad i = 2, \ldots, k \tag{2.35}$$

$$c_i = \frac{\int_{b_i}^{b_{i+1}} x f(x) dx}{\int_{b_i}^{b_{i+1}} f(x) dx} \ , \quad i = 1, \ldots, n \tag{2.36}$$

That is, the bounds must be exactly in the middle between two adjacent codewords and the codewords must be the centroids of the intervals between two adjacent bounds.

Due to the mutual relationship of the equations in (2.35)-(2.36), their solution is not easy. For this reason, the critical points are iteratively approximated by alternating between the necessary criteria. The process is summarized in Alg. 1.

The Lloyd-Max quantizer is closely related with the k-means algorithm[1], which is a standard method for clustering. In particular, k-means is the equivalent of Lloyd-Max for the scenarios where, instead of the probability density function, a set of observed samples is given à priori. In analogy to Lloyd-Max, the necessary optimality criteria and the overall k-means algorithm can be seen in Alg.2, where the observed samples are denoted as s_j, $j = 1, \ldots, m$. It is easy to see that k-means is asymptotically equal to Lloyd-Max, as the number of observed samples increases.

Because the approximated critical point is not always the global minimum

[1]In the machine learning community, the names k-means and Lloyd-Max (or more often, Lloyd's) algorithm are used interchangeably. We emphasize that both Lloyd and Max published their method in the form that is presented here.

Algorithm 2 k-means

1: Initialize all codewords c_i, $i = 1, \ldots, k$
2: Initialize sample range: $b_1 \leftarrow s_1$, $b_{k+1} \leftarrow s_m$
3: **while** stopping criteria are not met **do**
4: $\quad b_i \leftarrow (c_i + c_{i-1})/2$, $\quad$ for $i = 2, \ldots, k$
5: $\quad c_i \leftarrow \frac{1}{|Q_i|} \sum_{b_i \leq s_j \leq b_{i+1}} s_j$, $\quad$ for $i = 1, \ldots, k$

Algorithm 3 k-means++

1: Choose c_1 uniformly at random in $[s_1, s_m]$.
2: **for** $i = 2, \ldots, k$ **do**
3: $\quad$ For each sample s_j, compute d_j, the distance to the closest centroid of those already chosen.
4: $\quad$ Choose c_i at random over the set of samples, with probability for each sample s_j equal to $\frac{(d_j)^2}{\sum_{l=1}^{m}(d_l)^2}$.
5: Proceed to the standard k-means algorithm.

point, the initialization affects the attained distortion. In fact, the final distortion is generally very sensitive with the starting centroids. The initialization also greatly affects the convergence rate. The simplest initialization technique is choosing the starting codewords completely at random, but has no guarantees for the results. A better technique is the k-means++ [2] (ref. Alg. 3), which has the advantage of provably bounding in the mean value the optimal distortion D_{opt}. More precisely, the following result, proved in [2], holds for the final distortion D_{++} when using the k-means++ initialization method:

Theorem 2.1.6. *For any set of data samples,* $\mathbb{E}[D_{++}] \leq 8(\ln k + 2)D_{opt}$.

In fact, this bound holds even without proceeding to the standard k-means after initialization.

2.1.4 Statistical hypothesis testing

A statistical hypothesis test is a method of statistical inference. A hypothesis is a certain statement regarding the statistical characteristics of a family of events. An event is defined as a collection of data samples, including time series (sub)sequences. An event under investigation is tested whether it confirms or rejects a pre-defined hypothesis by using an appropriate metric. In particular, for statistical inference, two hypotheses must be defined:

Definition. The *null hypothesis* H_0 is the statement that is assumed to hold true for the questioned event. The *alternative hypothesis* H_1 is the complementary

hypothesis of H_0. That is, H_1 is confirmed when H_0 is rejected and vice versa.

The relevant statistical characteristics of the event are summarized with a test statistic:

Definition. A *test statistic* T is a function of an event, defined in such a way as to quantify the important statistical characteristics that would distinguish the null from the alternative hypothesis.

Because the event is modeled as a random variable, the function T is a random variable, too. That is, when sampling an event, the test statistic is sampled at the same time.

The hypothesis test is performed by comparing the test statistic with a threshold, called significance level.

Definition. The *significance level*, denoted by γ, is a boundary in the probability density domain of the null hypothesis H_0, which separates the *normal* and the *critical* region. When the statistic lies in the normal region, then the null hypothesis H_0 is confirmed. Otherwise, H_0 is rejected and the alternative hypothesis H_1 is confirmed.

In other words, the significance level is the density of the critical region. Additionally, it is the probability of rejecting the null hypothesis while it is true, in which case the decision is false and an error occurs.

Indeed, due to the random nature of the event being tested, the hypothesis that is confirmed might be false. In hypothesis testing, two types of error are distinguished, named Type I and Type II errors.

Definition. A *Type I error* occurs when the null hypothesis is true, but is rejected by the test. A *Type II error* occurs when the null hypothesis is false, but is confirmed by the test.

Statistical hypothesis testing can be seen a form of binary classification. In this context, rejecting the null hypothesis corresponds to a *positive* result, while confirming the null hypothesis corresponds to a *negative* result. Thus, Type I errors are equivalent to *false positives*, whereas Type II errors are equivalent to *false negatives*.

The performance of a hypothesis test is evaluated in terms of Type I and Type II error rates. Denote with $P(\hat{\theta_1}|H_0)$ the probability of Type I errors and with $P(\hat{\theta_0}|H_1)$ the probability of Type II errors, where $\hat{\theta_j}$ denotes that the test confirmed the H_j hypothesis. Then,

$$\text{False Positive / Type I Error rate} : \quad P(\hat{\theta_1}|H_0) = \gamma \ , \tag{2.37}$$

$$\text{False Negative / Type II Error rate} : \quad P(\hat{\theta_0}|H_1) = \beta \ , \tag{2.38}$$

where γ is the significance level of the test and β is related to the *power* of a test, which is defined as

$$\text{True Positive / Power} : \quad P(\hat{\theta_1}|H_1) = 1 - \beta \ . \tag{2.39}$$

The formulation of a test is complicated due to the trade-off between Type I and Type II errors. That is, while minimizing one type of error, the other tends to increase and vice versa. In practice, a test statistic is designed by minimizing one type of error with the constraint of fixing the other. In the specific case of minimizing β, while fixing γ, the test is said to be *powerful*.

One particular kind of hypothesis tests is the goodness of fit. A goodness of fit test assesses the likelihood that a statistical model (a pmf or a pdf) fits an event. In this case, the null hypothesis is that the generator process of the event is described from the defined probability density function.

There are three distinct classic methods for performing a goodness of fit test: the likelihood ratio test, the Lagrange multiplier test and the Wald test. For the scope of this work, we will cover the likelihood ratio test and note some results.

Definition. Let X be a random variable following a parametric statistical model f. Denote with $\mathbf{x}$ an event which is a collection of samples from X. Let a subset of f's parameters be denoted with the vector $\boldsymbol{\theta}$. Then,

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{x}) = f(\mathbf{x}|\boldsymbol{\theta}) \tag{2.40}$$

is the *likelihood function* of θ.

Suppose that $\boldsymbol{\theta}$ is an unknown vector and $\boldsymbol{\theta_0}$, $\boldsymbol{\theta_1}$ be two candidates of $\boldsymbol{\theta}$. The likelihood ratio test (LRT) assesses whether $\boldsymbol{\theta}$ is more likely to equal $\boldsymbol{\theta_0}$ than $\boldsymbol{\theta_1}$.

Definition. Define the null hypothesis $H_0 : \ \boldsymbol{\theta} = \boldsymbol{\theta_0}$ and the alternative hypothesis $H_1 : \ \boldsymbol{\theta} = \boldsymbol{\theta_1}$. The LRT statistic is defined as

$$\Lambda(\mathbf{x}) = \frac{\mathcal{L}(\boldsymbol{\theta_0}|\mathbf{x})}{\mathcal{L}(\boldsymbol{\theta_1}|\mathbf{x})} \ . \tag{2.41}$$

The null hypothesis is rejected when the following inequality is true:

$$\Lambda(\mathbf{x}) \leq \eta \ , \tag{2.42}$$

where η is chosen so that the significance level is

$$\gamma = P(\Lambda(\mathbf{x}) \leq \eta | H_0) \ . \tag{2.43}$$

LRT is a simple vs. simple test. That is, under both null and alternative hypotheses, the unknown parameters are simple points and thus, together with the known parameters, define completely the statistical model. The opposite of a simple hypothesis is a composite hypothesis, under which a range of values is considered for the unknown parameters. In the fundamental paper of Neyman and Pearson [31], a Lemma is proved which states that the LRT is the most powerful (i.e. with the highest power (2.39)) simple vs. simple test at a given significance level γ, for any pair of $(\boldsymbol{\theta_0}, \boldsymbol{\theta_1})$.

The composite vs. composite version of LRT is the generalized likelihood ratio test (GLRT). Simple vs. composite tests are also a subset of GLRT.

Definition. Denote with Θ the parametric space of the unknown parameters $\boldsymbol{\theta}$ of the statistical model f. Define the null hypothesis $H_0 : \ \boldsymbol{\theta} \in \Theta_0$ and the alternative hypothesis $H_1 : \ \boldsymbol{\theta} \in \Theta_0^c$, where $\Theta_0 \in \Theta$ and $\Theta_0^c = \Theta \setminus \Theta_0$. The GLRT statistic is defined as

$$\Lambda(\mathbf{x}) = -2 \ln \frac{\sup_{\boldsymbol{\theta} \in \Theta_0} \mathcal{L}(\boldsymbol{\theta}|\mathbf{x})}{\sup_{\boldsymbol{\theta} \in \Theta_0^c} \mathcal{L}(\boldsymbol{\theta}|\mathbf{x})} \ . \tag{2.44}$$

The operator $-2\ln(\cdot)$ seems redundant, but facilitates the operations with the logarithms. The null hypothesis is rejected when the following inequality is true:

$$\Lambda(\mathbf{x}) \geq \eta \ , \tag{2.45}$$

where η is chosen so that the significance level is

$$\gamma = P(\Lambda(\mathbf{x}) \geq \eta | H_0) \ . \tag{2.46}$$

The Neyman-Pearson Lemma does not hold for GLRT. In fact, GLRT is not the most powerful test in the general case.

A convenient result for GLRT is Wilks' Theorem [44], which describes the asymptotic distribution of the GLRT statistic (2.44). Basically, Wilks' Theorem categorizes the GLRT to the class of tests that are called chi-squared tests.

Theorem 2.1.7 (Wilks' Theorem)**.** *Let X be a random variable following a statistical model f with parameters $\theta_1, \ldots, \theta_h$. Denote with $\mathbf{x}$ an event which is a collection of N samples from X. Define the null hypothesis $H_0 : \theta_{m+1} = \theta_{0(m+1)}, \ \ldots \ , \theta_h = \theta_{0(h)}$. Then, under suitable restrictions on f, when H_0 is true the distribution of*

$\Lambda(\mathbf{x})$ *approaches uniformly, as* $N \to \infty$*, the* χ^2 *distribution with* $h - m$ *degrees of freedom.*

In light of Wilks' Theorem, the problem of parameter estimation from a sufficiently large collection of samples can be performed as follows. Suppose that the distribution function f depends on h parameters and we want to estimate $m \leq h$ of them with means of GLRT. Define the desired significance level γ (usually between 0.01 and 0.05). Then, the threshold η is computed as

$$\begin{aligned} \gamma &= P(\Lambda(\mathbf{x}) \geq \eta | H_0) \\ \overset{\text{Wilks' Th.}}{\Longrightarrow} \gamma &= F_{\chi^2}(\eta) \\ \Rightarrow \eta &= F_{\chi^2}^{-1}(\gamma) \ , \end{aligned} \tag{2.47}$$

where $F_{\chi^2}^{-1}$ is the inverse cdf (sometimes called quantile function) of the χ^2 distribution with $h-m$ degrees of freedom, which has no closed-form representation but can be sufficiently approximated with various methods.

With (2.47), a rule for determining optimal values for the statistic threshold η is derived. Following this procedure, optimal parameters of a statistical model that best fit an event of N samples from the random variable X can be estimated. It is very often the case that a continuous data source is discretized (quantized), in order to provide a basis for efficient statistical inference. The statistical model the describes a discrete random variable X is a mass function. As shown later in Sec. 2.2.1, the number of parameters required to define a pmf, when no other information is given, is equal to $|\mathcal{X}| - 1$, where $|\mathcal{X}|$ is the cardinality of the sample space of the discretized data source. That is, $|\mathcal{X}|$ equals the number of quantization intervals.

In that case, what is the best number of quantization intervals? If we had an infinite number of samples, then the largest this number, the better would be the estimated model. For relatively few samples, however, a very large number of quantization intervals would result to poor modeling, as the samples would not be enough to estimate accurately so many parameters. On the other hand, a very small number of quantization intervals would provide a good estimation of the parameters but also an overly simple statistical model, that does not represent the data source adequately.

A rule for deriving the number of quantization intervals for chi-squared tests, in terms of guaranteeing that the power (2.39) of the test is always greater than or equal to 1/2, has been derived by Mann and Wald in [24]. In their result, the number of quantization intervals K_N is given by the formula

$$K_N = 4 \cdot \left[\frac{2(N-1)^2}{c^2} \right]^{1/5} , \tag{2.48}$$

where c is determined from the significance level γ so that

$$\gamma = \frac{1}{\sqrt{2\pi}} \int_c^\infty e^{-x^2/2} dx \ . \tag{2.49}$$

The formula (2.48) is based on the assumptions that the sample size N is large and the quantization intervals are equiprobable. In any case, it can be used as a good starting point for further optimization.

2.2 Anomaly detection based on statistical hypothesis testing

Due to the heterogeneity of different data sources and the broad range of cases where anomaly detection applies, and also due to the divergent sources of possible anomalous activity, it is impossible to implement an anomaly detection method that works universally well. Indeed, a perfect universal anomaly detector would work in a maximum likelihood estimation fashion, i.e. it would model exactly all conditional probabilities of all temporal combinations of data points and would flag anomalous events according to their probability of appearance. This is however impossible in any realistic scenario. On the other hand, a more restrictive definition of what anomaly is, enables the implementation of feasible anomaly detection algorithms that work adequately well.

One such approach is to compare the distribution of a block of adjacent data points with the other blocks near the given block, or even everywhere in the data samples. A very dissimilar distribution indicates abnormality in the block. We denote this type of anomaly as *anomaly in distribution*. Here, adjacency can be considered across any dimension of the data. For example, it may be time in time series and/or space in graphs. This includes two wide classes of anomalies that appear frequently. In advance, distribution similarity has been studied excessively in many scientific domains (ref. f-divergences in Sec. 2.1.1 and likelihood ratio tests in Sec. 2.1.4). The theoretical advancements in these domains can be incorporated for the detection of anomalies in distribution.

The major downside of this approach is that the correlation between the data points inside a block is disregarded. Illustratively, any permutation of a block has exactly the same distribution.In fact, a stronger definition of anomaly would take into account the conditional probabilities inside the separate blocks, which would

in turn require a certain statistical model to be used. Approaches of this kind have been studied recently [6, 47, 41], which are handled with the means of statistical hypothesis testing.

In the present work, an anomaly detection method is developed, that is based upon the method proposed in [42]. This method is essentially a method for detecting anomalies in distribution and relies on the framework of likelihood ratio tests and more specifically Wilks' Theorem. The method is particularly attractive because it assumes no specific model of the data, a property that renders it, with minor modifications, a universal method for many application domains. Also, its simplicity and computational efficiency is very important in scenarios with limited resources. The following subsection presents the method in [42] and its theoretical background.

2.2.1 Anomaly detection via goodness of fit

The method proposed in [42] performs a goodness of fit test which involves the Kullback-Leibler divergence (2.4). Hereafter, we denote this method by "KL GoF". The method is based on the following corollary, whose proof is also provided below.

Corollary. *Let X and $\hat{X}$ be two discrete random variables defined on the same sample space $\mathcal{X}$, where the probability distribution of $\hat{X}$ is the empirical distribution of X, estimated by drawing N samples from X. Then, the distribution of $(2N \cdot D_{KL}(\hat{X} \parallel X))$ approaches uniformly, as $N \to \infty$, the χ^2 distribution with $|\mathcal{X}| - 1$ degrees of freedom.*

Proof. Consider the discrete random variables X and $\hat{X}$, defined on the same sample space $\mathcal{X} = (x_1, \ldots, x_s)$, where $s = |\mathcal{X}|$. The probability mass function of X is given by

$$P_X(x_i) = \begin{cases} p_1 & , \text{ if } i = 1 \\ \vdots & \\ p_{s-1} & , \text{ if } i = s-1 \\ 1 - \sum_{j=1}^{s-1} p_j & , \text{ if } i = s \end{cases} \tag{2.50}$$

Hence, P_X can be expressed in terms of the parameter vector $\boldsymbol{\theta} = (p_1, p_2, \ldots, p_{s-1})$. Similarly, the parameter vector of $P_{\hat{X}}$, denoted by $\hat{\boldsymbol{\theta}}$, is a vector of length $s - 1$. Suppose that $\boldsymbol{\theta}$ is a known fixed vector $\boldsymbol{\theta} = \boldsymbol{\theta_0}$ and that $\hat{\boldsymbol{\theta}} \in \Theta$ is an unknown vector that lies in the parametric space Θ.

Next, let $\mathbf{x} = (x_1, \ldots, x_N)$ be a vector of N samples drawn from $\hat{X}$. Define the null hypothesis $H_0 : \hat{\boldsymbol{\theta}} = \boldsymbol{\theta_0}$ and the alternative hypothesis $H_1 : \hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta_0}$. Then, the GLRT statistic (2.44) is given by

$$\begin{aligned}
\Lambda(\mathbf{x}) &= -2 \ln \frac{\mathcal{L}(\boldsymbol{\theta_0}|\mathbf{x})}{\sup_{\hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta_0}} \mathcal{L}(\hat{\boldsymbol{\theta}}|\mathbf{x})} \\
&= -2 \ln \frac{\mathcal{L}(\boldsymbol{\theta_0}|\mathbf{x})}{\mathcal{L}(\boldsymbol{\theta_1}|\mathbf{x})} \\
&= -2 \ln \frac{\prod_{i=1}^{N} P_X(x_i)}{\prod_{i=1}^{N} P_{\hat{X}}(x_i)} \\
&= -2 \ln \prod_{i=1}^{N} \frac{P_X(x_i)}{P_{\hat{X}}(x_i)} \\
&= -2 \sum_{i=1}^{N} \ln \frac{P_X(x_i)}{P_{\hat{X}}(x_i)} \\
&= -2N \frac{1}{N} \sum_{i=1}^{N} \ln \frac{P_X(x_i)}{P_{\hat{X}}(x_i)} \\
&\xrightarrow[L.L.N.]{a.s.} -2N \cdot \mathbb{E}_{x \sim \hat{X}} \ln \frac{P_X(x)}{P_{\hat{X}}(x)} \\
&= 2N \cdot \mathbb{E}_{x \sim \hat{X}} \ln \frac{P_{\hat{X}}(x)}{P_X(x)} \\
&= 2N \cdot D_{KL}(\hat{X} \parallel X) \ ,
\end{aligned} \tag{2.51}$$

where $\boldsymbol{\theta_1}$ denotes the empirical distribution of $\mathbf{x}$, the Law of Large Numbers is used in line 4 and (2.5) is used in the last line. Using Wilks' Theorem in (2.51) completes the proof.

□

The KL GoF method exploits this corollary to test whether the most recent block of N samples is distributed similarly with the past data. Specifically, the null hypothesis is a composite hypothesis that consists of the union of the empirical distributions of the past N-length blocks. The null hypothesis is then partitioned into multiple simple hypotheses which are tested separately by following the procedure in (2.47). When the null hypothesis is rejected, the current block is flagged as anomalous.

It should be noticed that the above corollary is proved for i.i.d. samples. Because this is not the case for time series, where the samples are highly correlated, the result holds only approximately.

An important feature of the method is that the data needs to be discrete. This

is the price of not having knowledge of the statistical model that underlies the generator process. In the case the model was known, we would still be able to employ a GLRT for the classification of its temporal parameters. Conventionally, the KL GoF method discretizes the time series data samples via uniform quantization, where the quantization intervals are of equal size regardless of the density of the samples inside each of them.

Following the discretization of the current sample, the empirical distribution of the N most recent samples is estimated in a sliding window fashion. The empirical distributions need to be saved, so they can be compared with future blocks. Because only the past distributions are considered in the hypothesis test, the method is causal and thus is appropriate for streaming data.

2.3 The symbolic aggregate approximation (SAX)

This section introduces the conventional SAX [19, 20] method. The core of a SAX consists of a two-step transformation that reduces the dimensionality of the data, coupled with the definition of an appropriate distance measure in the lower-dimensional space, which lower bounds the Euclidean distance in the original space.

2.3.1 The SAX transformation

In the following, $\mathcal{T}^N$ denotes the set of time series of length N, $\mathcal{Y}^M$ is the set of real vectors of length M, and $\mathcal{C}_A^M$ the set of all vectors of M codewords belonging to an alphabet A of size $|A| = \alpha$. Let $U = (u_1, u_2, \ldots, u_N) \in \mathcal{T}^N$ be a discrete time series of N samples, where u_i is the ith sample. Without loss of generality, U is first Z-normalized, as follows.

$$\hat{U} = \frac{U - \mu}{\sigma} \,, \tag{2.52}$$

where μ and σ are the mean and standard deviation of U. This ensures zero mean and unit variance of the time series, regardless of the underlying distribution. For ease of notation, hereafter U is assumed to be Z-normalized à priori.

The first step of SAX implements a piecewise aggregate approximation (PAA) $\mathcal{T}^N \to \mathcal{Y}^M$, which transforms a given time series $U \in \mathcal{T}^N$ into a vector $Y = (y_1, \ldots, y_M) \in \mathcal{Y}^M$, with $M < N$. For this, U is divided into M equal size segments and the average value is calculated for each segment. The ratio M/N determines the degree of dimensionality reduction.

In the second step, a discretization $\mathcal{Y}^M \to \mathcal{C}_A^M$ is applied to Y, which maps the averages into a predefined set of symbols. More precisely, the Z-normalized

time series is assumed to follow a standard Gaussian distribution. Under this assumption, the M averages in Y are quantized within α equiprobable intervals under the standard Gaussian pdf curve. Each quantization interval is bounded by two cutlines and is assigned a codeword from the alphabet A. The two-step transformation $\mathcal{T}^N \to \mathcal{Y}^M \to \mathcal{C}_A^M$ produces the SAX representation of length M from the alphabet A. Figure 2.3 provides a visualization of the two-step process.

2.3.2 Lower-bounding distance measure

Given two distinct time series U, $S \in \mathcal{T}^N$, their Euclidean distance is defined by

$$d(U,S) = \sqrt{\sum_{i=1}^{N} (u_i - s_i)^2}\ , \tag{2.53}$$

which is the L^2 norm of their difference.

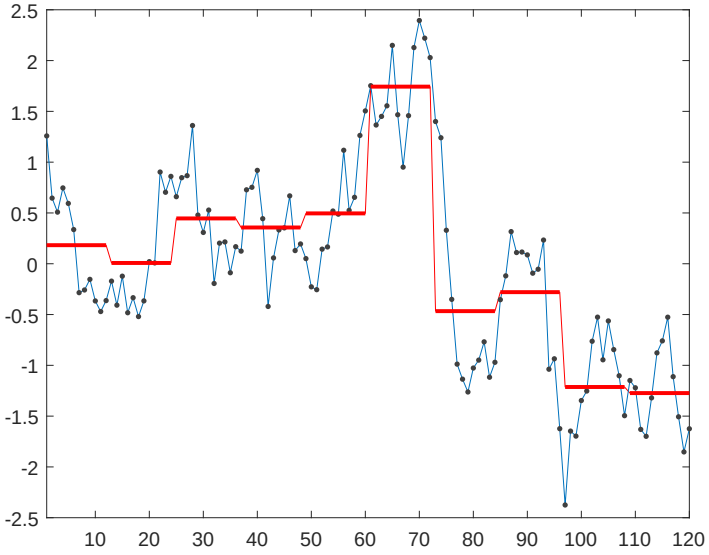
As discussed in Sec. 1.1, the GEMINI framework [11] states that in order to guarantee the absence of false dismissals when performing high-level tasks, such as similarity searching, it suffices to define a distance measure in the lower-dimensional space $\mathcal{C}_A^M$ that lower bounds the Euclidean distance in the original space $\mathcal{T}^N$. Let C, $Q \in \mathcal{C}_A^M$ be the symbolic representations of the time series U and S, respectively. Then, a distance measure in the quantized space of alphabet symbols, which lower bounds the Euclidean distance in the original time series space is defined as follows,

$$mindist(C,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \left(dist(c_i, q_i)\right)^2}\ , \tag{2.54}$$

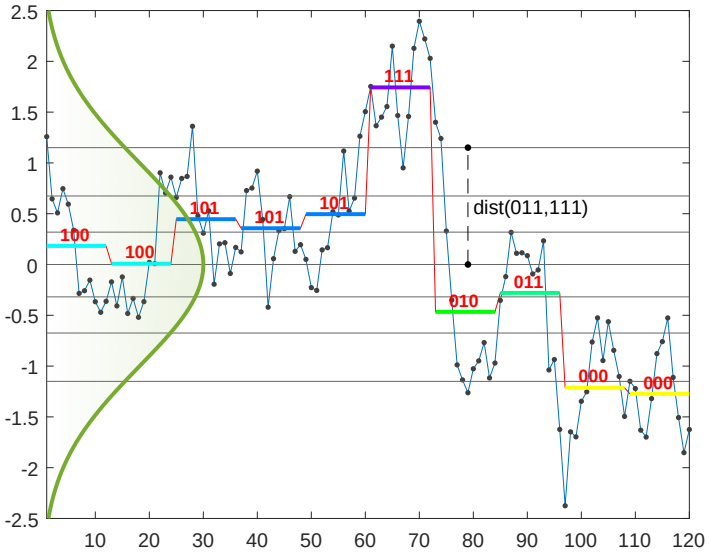
where $dist(c_i, q_i)$ is the absolute difference of the two closest cutlines that respectively bound c_i and q_i (refer to bottom plot in Fig. 2.3 for an example). Furthermore, if $Y \in \mathcal{Y}^M$ is the PAA of U and $Q \in \mathcal{C}_A^M$ is the SAX representation of S, a tighter lower bounding distance measure can be defined by

$$mindist\_PAA(Y,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \begin{cases} (\beta_{L_i} - y_i)^2 & \text{if } \beta_{L_i} > y_i \\ (\beta_{U_i} - y_i)^2 & \text{if } \beta_{U_i} < y_i \\ 0 & \text{otherwise} \end{cases}}\ , \tag{2.55}$$

where β_{L_i} and β_{U_i} are the lower and upper cutlines of codeword q_i. By combining (2.53) and (2.55), the tightness of lower bound (TLB) measure is defined as



(a) Piecewise aggregate approximation (PAA)



(b) Discretization of PAA

Figure 2.3: SAX representation of a time series. A series of $N = 120$ samples is first transformed into its PAA representation by segmenting and averaging the series into $M = 12$ pieces. Then, each segment is assigned a codeword (shown in red above the segments), subject to which of the $\alpha = 8$ equiprobable intervals of the standard Gaussian pdf it falls in. Here, the codewords are the binary representations of $1, 2, \ldots, 8$.

$$TLB(U,S) = \frac{mindist\_PAA(Y,Q)}{d(U,S)} \ . \tag{2.56}$$

TLB ranges in $[0,1]$ and is the fraction of how close is $mindist\_PAA(Y,Q)$ to the true Euclidean distance of the data. It is a metric which determines the rate of false positives (2.37) (Type I errors) in similarity searching.

Chapter 3

Data-driven SAX-based Representation of Time Series

3.1 Optimal quantization for SAX

As noted in Section 2.3.1, the piecewise aggregate approximation $\mathcal{T}^N \to \mathcal{Y}^M$ of the time series $U \in \mathcal{T}^N$ is followed by the discretization $\mathcal{Y}^M \to \mathcal{C}_A^M$, which quantizes the PAA segments into α equiprobable intervals under the standard Gaussian pdf. Assuming standard Gaussian distribution of $\mathcal{Y}^M$, the output is a random sequence that follows the uniform distribution in A. Furthermore, the Gaussian assumption is adopted due to the widely accepted observation that time series from various sources, very often, follow approximate Gaussian statistics.

Although not stated clearly in the introductory paper, the choice of employing equiprobable intervals is usually based upon the observation that a sequence of samples drawn from a uniform distribution maximizes the entropy of the output sequences (Lemma 2.1.2). In turn, this leads to a larger typical set (Theorem 2.1.2) and hence to a larger number of distinct symbolic representations after the transformation of a large number of time series. This implies an increased distinctability of the time series in the lower-dimensional space. This is an application of the Principle of Maximum Entropy [14].

In summary, the approach of discretization in SAX is based on the following arguments: i) Z-normalized time series approximately follow the standard Gaussian distribution, ii) statistical inference augments with entropy. In this section, we controvert both of those arguments on the ground of time series representations.

3.1.1 On the Gaussian assumption

Gaussian statistics might appear in a range of data sources, however they are very often invalid. As such, a mismatch in the distribution yields an information loss that is specified by the Kullback-Leibler divergence (Theorem 2.1.4 for discrete values). When the underlying distribution of the data is not close to Gaussian, the information loss can severely decrease the quality of SAX. In such cases, in order to produce computationally tractable processing methods, it is necessary to fit the data to specific models (e.g. [12, 9]).

In the following we quantify the information lost from a quantized data source, induced by assuming standard Gaussian distribution, as SAX does, whilst the true underlying distribution is given by another probability density function. In order to achieve this, we first introduce the notion of differential entropy and the Kullback-Leibler divergence of continuous random variables.

Definition. The *differential entropy* $h(X)$ of a continuous random variable X with pdf $f(x)$ is defined as

$$h(X) = -\int f(x) \log f(x) dx, \tag{3.1}$$

where the integration is performed in the areas of the sample space where $f(x) > 0$.

Notice that the differential entropy (3.1) can be negative, which is counter-intuitive. Translating and scaling a random variable affects the differential entropy as follows:

Lemma 3.1.1.

$$h(X + c) = h(X) \ , \tag{3.2}$$

$$h(aX) = h(X) + \log |a| \ . \tag{3.3}$$

Definition. The *Kullback-Leibler* divergence $D_{KL}(f \parallel g)$ of the densities f and g is defined by

$$D_{KL}(f \parallel g) = \int f(x) \log \frac{f(x)}{g(x)} dx \ , \tag{3.4}$$

with the convention that $0 \log \frac{0}{0} = 0$.

It turns that the Asymptotic Equipartition Property and the properties of the continuous analog of the typical set hold similar to the discrete case [7, Sec. 8.2]. Also, the Kullback-Leibler divergence has the following properties.

Lemma 3.1.2.

$$D_{KL}(f \parallel g) \geq 0 \tag{3.5}$$

with equality iff f=g almost everywhere.

Lemma 3.1.3.

$$D_{KL}(f \parallel g) = h(f,g) - h(f) \ , \tag{3.6}$$

where $h(f,g) = \int f(x) \log g(x) dx$ *is the continuous cross-entropy.*

Now, let us consider a random variable X with density $f(x)$. Suppose that the sample space of X is divided into intervals of equal length Δ. Assuming that the density is continuous within the bins, there exists a value x_i within each bin such that

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)dx \ . \tag{3.7}$$

Then, consider the quantized version of X, denoted with X^Δ and defined by

$$X^\Delta = x_i \quad , \text{if } i\Delta \leq X < (i+1)\Delta \ , \tag{3.8}$$

which implies that the pmf of X^Δ is

$$P(x_i) = f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)dx \ . \tag{3.9}$$

We assumed quantization intervals of equal length to simplify our computations. However, because our result holds asymptotically, this assumption does not cancel its validity. This is correct because, for any quantization scheme, as the number of the quantization intervals approaches infinity, they become equally infinitesimal.

The next Theorem is the continuous analog of Theorem 2.1.4, which quantifies the information loss of assuming a wrong pdf $q(x)$ instead of $f(x)$ when source coding the quantized version X^Δ of X.

Theorem 3.1.1. *The expected description length* $L(C)$ *of the random variable* X^Δ *by a D-ary alphabet, assuming optimal coding under the probability density function* $g(x)$*, satisfies*

$$h(f) + D_{KL}(f \parallel g) + \log\frac{1}{\Delta} \leq L(C) < h(f) + D_{KL}(f \parallel g) + \log\frac{1}{\Delta} + 1 \ , \tag{3.10}$$

where the logarithm bases are D.

Proof is written in the Appendix, A.1.

In accordance to the discrete case, combining Lemma 3.1.2 and Theorem 3.1.1, we derive the following theorem.

Theorem 3.1.2. *The expected description length of the random variable X^{Δ} by a D-ary alphabet, assuming optimal coding under the probability density function $f(x)$, satisfies*

$$h(f) + \log \frac{1}{\Delta} \leq L(C) < h(f) + \log \frac{1}{\Delta} + 1 \ , \tag{3.11}$$

where the logarithm bases are D.

Theorems 3.1.1 and 3.1.2 together imply that the information lost by assuming a wrong pdf when source coding a quantized sampled random variable is the Kullback-Leibler divergence of the correct and the wrong pdf. This is similar to the result derived in the discrete case.

Next, consider the Gaussian random variable $G \sim N(\mu_g, \sigma_g^2)$ and its density $g(x)$.

Proposition 3.1.1. *Let X be an arbitrary continuous random variable with density $f(x)$, mean μ_x and variance σ_x^2, and $G \sim N(\mu_g, \sigma_g^2)$ with $\mu_x = \mu_g$, then*

$$D_{KL}(f \parallel g) = \ln \sigma_g \sqrt{2\pi} + \frac{1}{2}\frac{\sigma_x^2}{\sigma_g^2} - h(f) \ , \tag{3.12}$$

measured in nats.

Proof is written in the Appendix, A.2.

Consider, for example, that $f(x)$ follows a Laplace distribution $L(\mu, b)$, which is defined by

Definition (Laplace density function)**.**

$$f(x) = \frac{1}{2b} e^{-\left(\frac{|x-\mu|}{b}\right)} \ , \quad b > 0 \tag{3.13}$$

The mean value and variance of the Laplace distribution are equal to μ and $2b^2$, respectively. After a Z-normalization, it holds that $\mu = 0$, $b = \frac{1}{\sqrt{2}}$. Then,

$$\begin{aligned} D_{KL}(f \parallel g) &= \ln \sigma_g \sqrt{2\pi} + \frac{1}{2}\frac{\sigma_x^2}{\sigma_g^2} - h(f) \\ &= \ln \sigma_g \sqrt{2\pi} + \frac{1}{2}\frac{2b^2}{\sigma_g^2} - \ln 2be \\ &= \ln \sqrt{2\pi} + \frac{1}{2} - \ln \sqrt{2e} \\ &\approx 0.07 \text{ nats} \\ &\approx 0.1 \text{ bits} \ , \end{aligned} \tag{3.14}$$

which is a small penalty, due to the close relationship between the Laplace and Gaussian distributions.

As a second example, consider a mixture of Gaussians. More precisely, assume that $f(x)$ is the density function of a mixture of two Gaussians with equal variance σ^2 and opposite mean values μ and $-\mu$. The entropy of this distribution has been approximated by the authors of [29]. Let us name this distribution as split-Gaussian. The density function of this distribution is defined by

Definition (Split-Gaussian density function)**.**

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \left[e^{-\frac{x-\mu}{2\sigma^2}} + e^{-\frac{x+\mu}{2\sigma^2}} \right] . \tag{3.15}$$

The split-Gaussian distribution has zero mean and variance given by $\sigma^2_{mg} = \sigma^2 + \mu^2$. Consider that $\mu = 2$ and $\sigma = 1$, which results to a mixture of two standard Gaussian distributions centered around -2 and $+2$, respectively, and an ultimate variance of $\sigma^2_{mg} = 5$. After Z-normalization, the relative entropy

$$\begin{aligned} D_{KL}(f \parallel g) &= \ln \sigma_g \sqrt{2\pi} + \frac{1}{2}\frac{\sigma_x^2}{\sigma_g^2} - h(f) \\ &= \ln \sqrt{2\pi} + \frac{1}{2} - \left(\ln \sqrt{2\pi e} + 0.633 + \ln \frac{1}{\sqrt{5}} \right) \\ &\approx 0.17 \text{ nats} \\ &\approx 0.25 \text{ bits} , \end{aligned} \tag{3.16}$$

which means that when source coding a sequence of values drawn from $f(x)$ assuming that it is a Gaussian distribution, for every 4 symbols, 1 bit is redundant.

3.1.2 On Z-Normalization in SAX

Z-normalization is performed in the input data of SAX in order to "standardize" their underlying distribution, such that their mean value and variance are zero and one, respectively. This helps in the discretization step, where the PAA segments (segmented means) are quantized, to choose the Gaussian curve with the standard parameters.

Nevertheless, even when the Gaussian assumption holds, Z-normalizing the input data does not guarantee standard distribution in the PAA segments. To verify, consider the case where the time series $U \in \mathcal{T}^N$ is a sequence of correlated and jointly Gaussian-distributed random variables. Without loss of generality, consider the case where $M = N/2$, i.e. PAA takes pairs of adjacent samples and computes their averages. Consider a specific pair of those samples (u_j, u_{j+1}) that

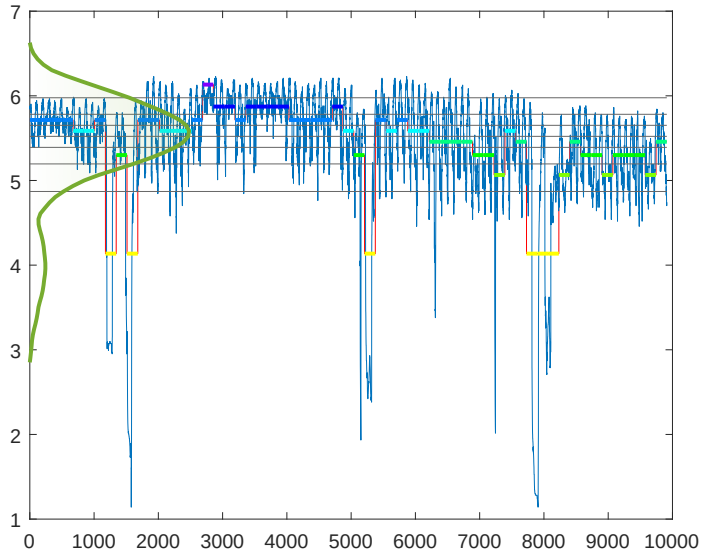


Figure 3.1: SAX with equiprobable intervals quantization

are Z-normalized and thus are distributed $u_j \sim N(0,1)$, $u_{j+1} \sim N(0,1)$. Denote with ρ the correlation coefficient of the pair.

Then, it can be proved that $Y = (u_j + u_{j+1})/2$ is distributed $Y \sim N(\mu_Y, \sigma_Y^2)$, where

$$\begin{aligned} \mu_Y &= 0 \ , \\ \sigma_Y^2 &= \frac{1+\rho}{2} \ . \end{aligned} \tag{3.17}$$

That is, the PAA segment Y follows the standard Gaussian distribution only when $\rho = 1$.

A simple, yet effective, fix to this behaviour is to apply Z-normalization on the PAA segments, rather than the raw data samples. This guarantees standard Gaussian distribution of the PAA segments.

3.1.3 On quantization with equiprobable intervals

As discussed in the introduction of this section, the choice of equiprobable quantization intervals is attractive due to the increased distinctability it offers to the quantized sequences by maximizing their entropy. When no other side information is available, this approach is intuitively correct. However, there are applications where this is not true.

Consider for example that an application benefits when assorting the range

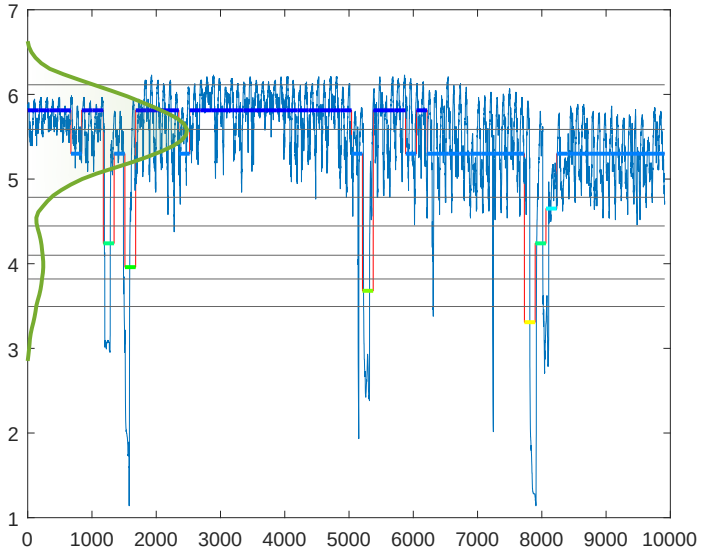


Figure 3.2: SAX with mode-bounding Lloyd-Max quantization

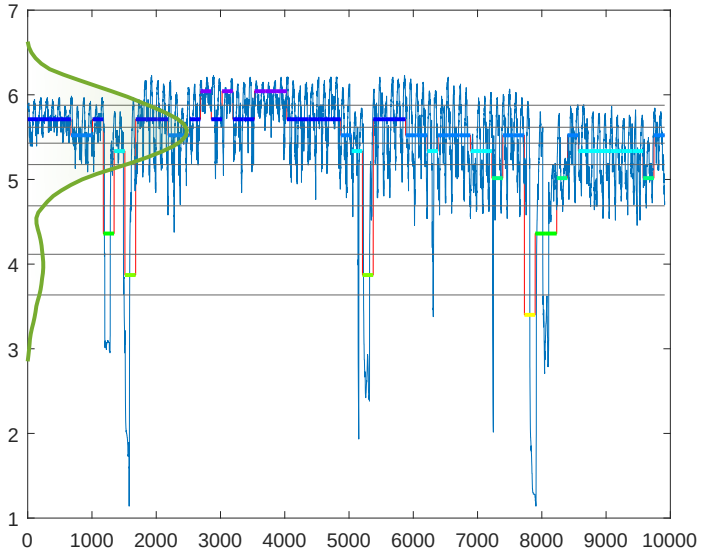


Figure 3.3: SAX with conventional Lloyd-Max quantization

of values into intervals that distinguishes adequately the extreme values from the other. This is the case with most of anomaly detection scenarios. An illustrative example is given in Figures 3.1 and 3.2, where equiprobable interval quantization is compared with a non-equiprobable method (mode-bounding Lloyd-Max), described in Sec. 4.1. As a second example, consider that the quantization method aims to generate a sequence that is as close as possible to the original sequence. This quantization method is the Lloyd-Max quantizer, described in Sec. 2.1.3, which does not output equiprobable codewords (Fig. 3.3). For all of the examples given in this subsection, the underlying distribution of the PAA segments is estimated via KDE (ref. Sec. 2.1.2).

Notice that indeed, with equiprobable intervals, the symbolic representation has maximum entropy (visually noticed by the highly fluctuating behaviour), whereas the Lloyd-Max quantizer, which minimizes the MSE, does not produce equiprobable intervals. On the other hand, the mode-bounding Lloyd-Max quantizer, represents better the high-level behaviour of the time series.

3.2 Data-driven kernel-based probabilistic SAX

Following the remarks in the previous subsection, a SAX-based symbolic representation method is developed within this thesis, hereafter referred to as probabilistic SAX (pSAX), which negates any assumptions regarding the probability distribution of a given time series. That is, the mapping between the time series space and the space of symbols adapts directly to the data statistics.

To this end, the proposed method applies a kernel density estimator (KDE) (Sec. 2.1.2) directly after the PAA to approximate accurately the underlying probability density function of the segmented means, without any prior probabilistic assumption. To further enhance the generalization capability of the proposed method, the KDE-based step is coupled with a Lloyd-Max quantizer (Sec. 2.1.3, Alg. 1) to estimate the optimal quantization boundaries, which are further used to define the map between the time series samples and the alphabet's symbols. Fig. 3.4 illustrates the overall pipeline of our method, whilst the bottom plot in Fig. 3.5 visualizes the outcome of the proposed symbolic representation, as opposed to the conventional SAX (top plot).

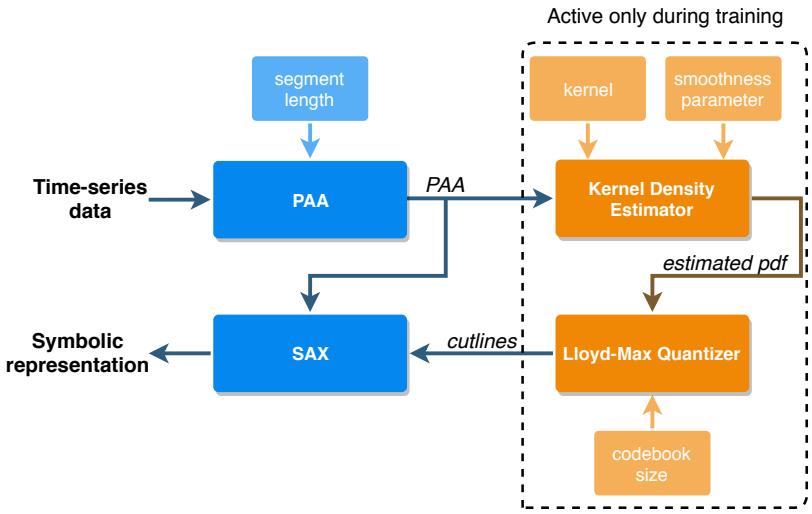


Figure 3.4: Overview of pSAX pipeline. Notice that KDE and Lloyd-Max are active only during training. After training, the boundaries are fixed and used unchanged for future inputs.

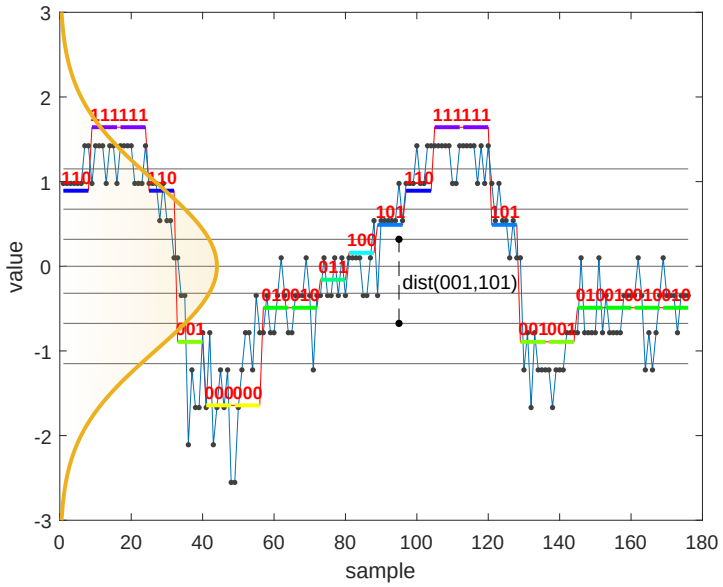
3.2.1 Relation to prior work

The method presented here takes advantage of the model-free data-adaptive nature of KDEs, along with the optimality (in the MSE sense) of Lloyd-Max quantization, for the samples-to-symbols assignment. Previous works in time series representations also perform data-driven discretization, by employing the k-means algorithm [8, 26], self-organizing maps [17], or other clustering methods [13]. A modified version of SAX (a.k.a. aSAX) [35] employs the k-means algorithm (Alg. 2) for the discretization procedure. The experimental results exhibit a better generalization capability of pSAX compared to aSAX, as shown in the Section 3.3.

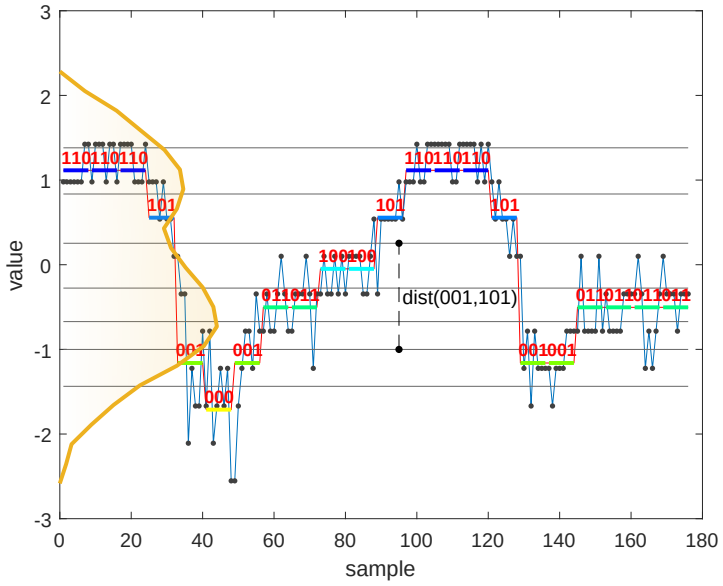
While the present study is related to prior works in data-driven discretization, our methodology capitalizes on the fact that Lloyd-Max quantization partitions the data according to the underlying probability distribution estimated directly from the data source. This was not considered in earlier studies, where the k-means and the other clustering methods mentioned above partition the data according to the observed samples.

3.2.2 Implementation details

In our pSAX method, the Lloyd-Max quantizer is chosen on the ground that it will aid in the precision of generic data mining tasks. Other quantization schemes might replace Lloyd-Max in specific tasks, as we demonstrate in Section 4.2 in the



(a) SAX



(b) pSAX

Figure 3.5: SAX representation of a time series with different quantization schemes. Top plot: Equiprobable intervals under standard Gaussian distribution; Bottom plot: Lloyd-Max quantization under KDE-estimated distribution.

Method	**Time (ms)**			
	$\alpha = 16$	$\alpha = 32$	$\alpha = 64$	$\alpha = 128$
pSAX	15.206	20.539	29.546	62.407
aSAX	2.859	4.644	6.402	9.587

Table 3.1: Training time of pSAX and aSAX on 10 sequences from the Koski ECG dataset ($M = 40$, $\alpha \in \{16, 32, 64, 128\}$, CPU: Intel i7-6700@3.8GHz).

case of anomaly detection.

In our implementation, we initialize the quantizer with the k-means++ algorithm (Sec. 2.1.3, Alg. 3), which uses the raw PAA samples to quickly compute a good starting point for the boundaries before optimizing upon the estimated density function. This yields an improved convergence to a local minimum, in terms of MSE and convergence speed.

Regarding the KDE, we employ the Epanechnikov kernel (2.23). This choice is motivated by (i) the asymptotic optimality (ref. Sec. 2.1.2) and (ii) the compact support of this kernel, which yields an increased computational efficiency. In advance, the smoothness parameter was chosen according to Silverman's rule of thumb (2.30). Although, after experimenting with the smoothness parameter, we concluded that reducing Silverman's approximation to its half resulted to better results.

Notice that a training phase is required for our pSAX method. Specifically, the KDE module for is trained first with a sufficient number of PAA segments. Then, the Lloyd-Max quantization intervals are calculated based on the estimated distribution. Nevertheless, this is done only once during initialization. As such, the running time of the training phase does not contribute to the running time of the subsequent dimensionality reduction process. Besides, our algorithm is trained efficiently using a highly reduced set of training sequences (at the order of 10 in this study). Table 3.1 shows the training times of pSAX and aSAX. As expected, the training of pSAX takes longer than its aSAX counterpart, which is due to the KDE step and the numerical integrations in the Lloyd-Max step.

3.2.3 A novel distance measure

The pSAX method uniquely defines a distance measure in the lower-dimensional space. Let us first repeat the basic distance measures defined in Sec. 2.3.2 and their properties. Let C, $Q \in \mathcal{C}_A^M$ be the symbolic representations of the time series U and S, respectively. The *mindist* measure lower bounds the Euclidean distance in the raw data space, a property that holds regardless of the chosen quantization

intervals and hence does in pSAX, too.

$$mindist(C,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \left(dist(c_i, q_i)\right)^2} \ , \tag{3.18}$$

where $dist(c_i, q_i)$ is the absolute difference of the two closest boundaries that respectively bound c_i and q_i.

Let $Y \in \mathcal{Y}^M$ be the PAA of U and $Q \in \mathcal{C}_A^M$ the SAX representation of S. A tighter than $mindist$ lower bounding distance measure is

$$mindist\_PAA(Y,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \begin{cases} (\beta_{L_i} - y_i)^2 & \text{if } \beta_{L_i} > y_i \\ (\beta_{U_i} - y_i)^2 & \text{if } \beta_{U_i} < y_i \\ 0 & \text{otherwise} \end{cases}} \ , \tag{3.19}$$

where β_{L_i} and β_{U_i} are the lower and upper boundaries of codeword q_i.

The tightness of lower bound (TLB) is defined as

$$TLB(U,S) = \frac{mindist\_PAA(Y,Q)}{d(U,S)} \ , \tag{3.20}$$

where $d(U,S)$ is the Euclidean distance of U and S.

Up to this point, only lower-bounding distance measures have been defined. Specifically, the distances in (3.18)-(3.19) employ the interval boundaries to lower-bound the true Euclidean distance. Notably, the conventional SAX does not define arithmetic values for the symbols in the lower-dimensional space.

On the other hand, the Lloyd-Max quantizer does provide arithmetic values for the codewords, apart for the boundaries. As discussed in Sec. 2.1.3, these values are the centroids of the bounded intervals. This feature can be further exploited to define a new distance measure between two symbolic sequences Q, $C \in \mathcal{C}_A^M$, as follows,

$$d_s(Q,C) = \sqrt{\frac{N}{M} \sum_{i=1}^{M} \left(q_i - c_i\right)^2} \ . \tag{3.21}$$

Although this measure does not lower bound the Euclidean distance, however, it is the closest to the true Euclidean distance in the MSE sense, up to a distortion caused by the inefficiency of KDE to estimate exactly the true distribution and of Lloyd-Max to minimize globally the MSE. Accordingly, a distance measure between a time series and a symbolic sequence can be derived by utilizing the computed

centroids. Specifically, given $U \in \mathcal{T}^N$ and $C \in \mathcal{C}_A^M$, their distance is defined by

$$d_e(U,C) = \sqrt{\frac{1}{N}\sum_{i=1}^{M}\left(\sum_{j=(N/M)(i-1)+1}^{(N/M)i}(u_j - c_i)^2\right)}. \tag{3.22}$$

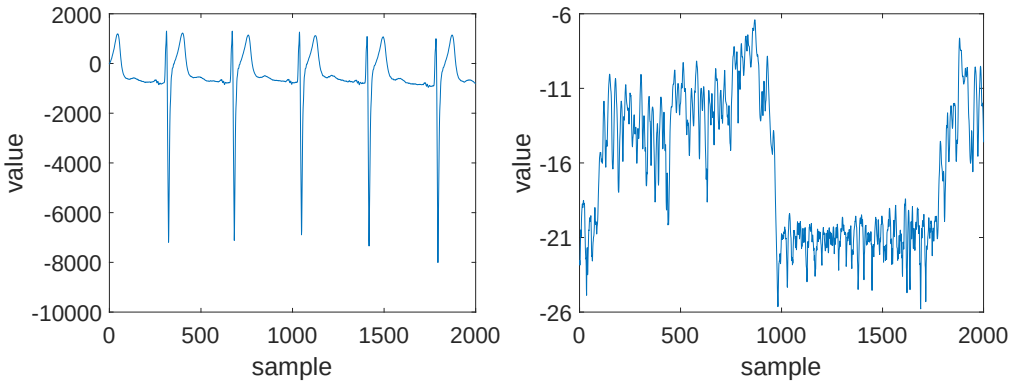
In the special case when C is the symbolic representation of U, then d_e is the root mean squared error (RMSE) between U and its reconstruction from C.

3.3 Experimental evaluation

This section evaluates the performance of pSAX and compares against aSAX and the conventional SAX, with respect to the achieved TLB (3.20) and RMSE (3.22) values. The methods are compared for a varying symbolic sequence length $M \in \{32, 48, 64, 80\}$ (the lower this number, the higher the dimensionality reduction), alphabet size $\alpha \in \{8, 16, 32, 64, 128\}$ and time series subsequence length $N \in \{480, 1920\}$ (short and long). Four distinct datasets are employed (Koski ECG, Muscle Activation, Rittweger EOG, and Respiration)[1], which are characterized by both structured and complex behaviors. Figure 3.6 illustrates the datasets. For each dataset, the results are averaged over 8000 Monte Carlo iterations, each one corresponding to a randomly selected segment from the associated time series. The length of the segments N and the parameters M and α are chosen in compliance with the experimental sections in [19], [35]. In order to simulate streaming scenarios, the training samples are taken from the beginning of each dataset. We note that, although our method does not require a Z-normalization of the time series, however, we also Z-normalize the given data for a fair comparison with the other methods.

As a first experiment, we investigate the effect of the symbolic sequence length M on the performance of our method. In particular, Fig. 3.7 shows the average TLB and RMSE values for the Respiration dataset (characterized by dense spikes). As it can be seen, pSAX achieves a tighter lower bound, yielding a more accurate lower-bounding distance (2.55) in the lower-dimensional space. Additionally, pSAX achieves a more accurate reconstruction (lower RMSE) against both SAX and aSAX. Furthermore, the superiority of pSAX is more prominent as M increases. Table 3.2 shows the average TLB and RMSE values for two additional datasets, namely, the Rittweger EOG (a waveform with varying frequency) and Koski ECG (characterized by a repetitive pattern). As it can be seen, similar results are obtained, demonstrating the superiority of our proposed method.

[1]Datasets available at `www.cs.ucr.edu/~eamonn/iSAX/iSAX.html`



(a) Koski ECG (b) Muscle Activation

(c) Rittweger EOG (d) Respiration

Figure 3.6: The datasets employed in the experimental evaluation of the proposed pSAX method. Here, a segment of 2000 samples from each dataset is visualized.

Next, we study the effect of the alphabet size α on the performance of pSAX. To this end, Fig. 3.8 shows the average TLB and RMSE values for the Muscle Activation dataset (a noisy periodic signal), as a function of α. The experiments show that pSAX achieves a tighter lower bound, along with a better reconstruction quality, when compared against SAX. The same holds against aSAX most of the times, except for a few datasets, when the alphabet size is $\alpha \leq 8$. As expected, the larger the alphabet size the more improved the performance of all the three methods (i.e., higher TLB and lower RMSE). Similar results are shown in Table 3.3 for the Respiration and Muscle Activation datasets, demonstrating the efficiency of pSAX in adapting to distinct data generating processes. Note that, for the calculation of RMSE (3.22), the pSAX codewords are computed using the Lloyd-Max algorithm, whilst for aSAX and SAX they are computed using the k-means and the line 5 in Algorithm 1, respectively.

Overall, we conclude that, under all the experimental parameters settings tested herein, pSAX achieves a tighter lower bound (i.e., closer to 1) and a smaller RMSE when compared with SAX. Moreover, in the vast majority of the settings, pSAX also outperforms aSAX, except for a few cases when the alphabet size is very small. Finally, note that, when SAX-based methods are used for data indexing purposes, the achieved speedup is generally nonlinear with respect to TLB [38]. Thus, we expect that our pSAX method will require a reduced number of disk accesses, when compared against aSAX and SAX, due to its higher TLB. However, the design of a pSAX-based indexing technique is left as a future thorough study.

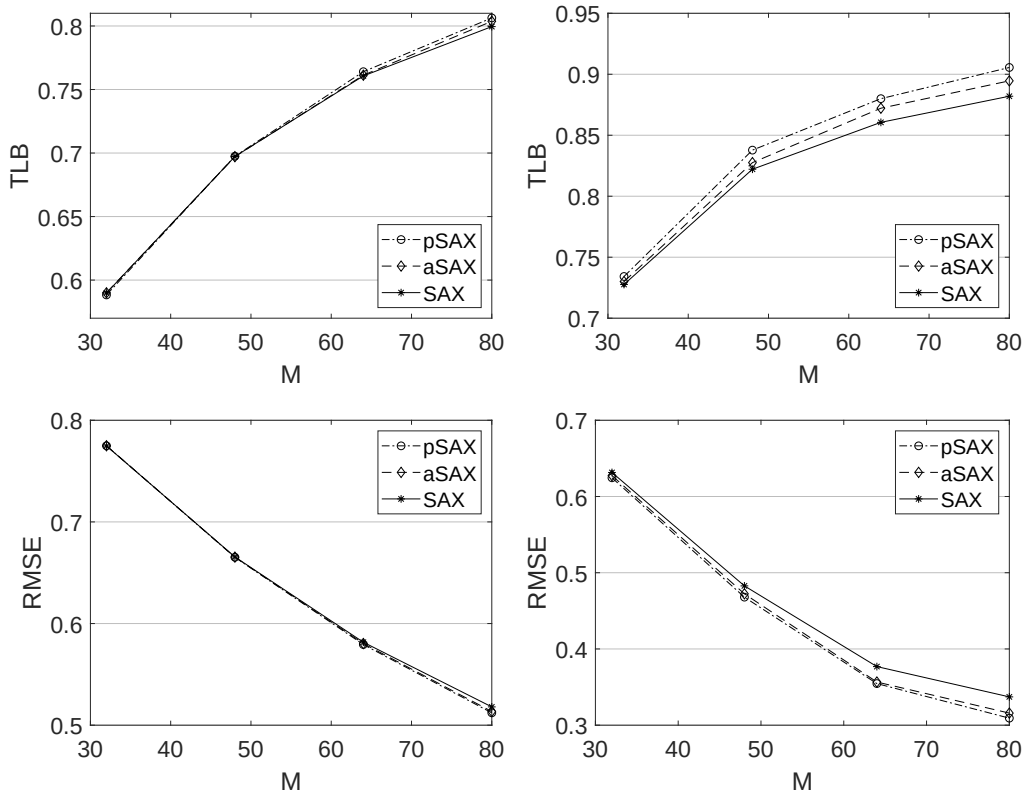


Figure 3.7: Tightness of lower bound (top) and reconstruction error (bottom) vs. M for the Respiration dataset ($\alpha = 32$). Left: N = 1920 – Right: N = 480

		$M = 32$	$M = 64$	$M = 80$
Dataset	**Method**	**TLB**		
Rittweger EOG	pSAX	**0.8954**	**0.9481**	**0.9570**
	aSAX	0.8930	0.9416	0.9500
	SAX	0.8936	0.9388	0.9463
Koski ECG	pSAX	**0.8205**	**0.9200**	**0.9380**
	aSAX	0.8084	0.8984	0.9219
	SAX	0.7719	0.8292	0.8406
Dataset	**Method**	**RMSE**		
Rittweger EOG	pSAX	**0.3614**	**0.1858**	**0.1606**
	aSAX	0.3638	0.1900	0.1663
	SAX	0.3626	0.2010	0.1782
Koski ECG	pSAX	**0.5025**	**0.2917**	**0.2535**
	aSAX	0.5037	0.2956	0.2574
	SAX	0.5415	0.4443	0.4310

Table 3.2: Average TLB and RMSE vs. M, for the Rittweger EOG and Koski ECG datasets ($N = 480$, $\alpha = 64$).

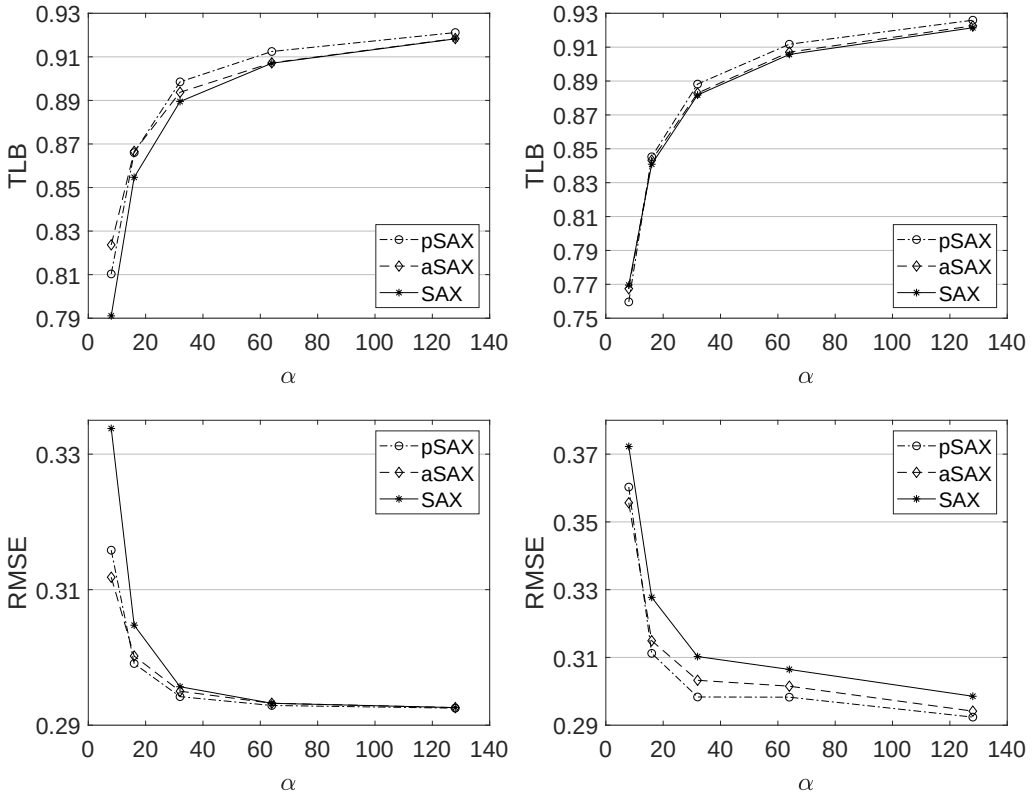


Figure 3.8: Tightness of lower bound (top) and reconstruction error (bottom) vs. α for the Muscle Activation dataset ($M = 80$). Left: N = 1920 – Right: N = 480

		$\alpha = 8$	$\alpha = 32$	$\alpha = 128$
Dataset	**Method**	**TLB**		
Respiration	pSAX	**0.5006**	**0.5834**	**0.6042**
	aSAX	0.4954	0.5794	0.6032
	SAX	0.4942	0.5664	0.5917
Muscle Activation	pSAX	0.8102	**0.8968**	**0.9223**
	aSAX	**0.8231**	0.8920	0.9196
	SAX	0.7911	0.8878	0.9194
Dataset	**Method**	**RMSE**		
Respiration	pSAX	**0.7894**	**0.7727**	**0.7740**
	aSAX	0.7910	0.7742	0.7741
	SAX	0.8110	0.7952	0.7912
Muscle Activation	pSAX	0.3160	**0.2941**	**0.2924**
	aSAX	**0.3120**	0.2949	0.2925
	SAX	0.3341	0.2955	0.2925

Table 3.3: Average TLB and RMSE vs. α, for the Respiration and Muscle Activation datasets ($N = 1920$, $M = 80$).

Chapter 4

Fast Anomaly Detection of Time Series

In this section, an unsupervised, non-parametric method for anomaly detection is developed, characterized by low power and memory demands. Due to these properties, and because of the causal nature of the method, the detector is an excellent candidate for processing streaming data. In particular, fast processing of the received streaming data is enabled by first applying a dimensionality reduction step. To this end, we rely on a variant of pSAX (Sec. 3.2), where the quantizer is a modified version of Lloyd-Max. The choice of pSAX is further motivated by the fact that symbolic representations can be coupled effectively with the KL GoF method (Sec. 2.2.1) in order to evaluate the time-evolving distribution of the generated symbols. The efficiency of the proposed method is evaluated by employing the Numenta Anomaly Benchmark (NAB) [18], a highly comparative scoring system. Below, the proposed online anomaly detection method, which we name "SAX-KL", is described in details.

4.1 Mode-bounding Lloyd-Max

As noted above, the proposed anomaly detector first performs dimensionality reduction on the streaming data via a variant of pSAX. The modification that distinguishes the present pSAX from the one that is described in Section 3.2 lies in the quantization step. Particularly, it can be observed that Lloyd-Max is not suitable for clustering data into clusters that correspond to distinct states of the time series, as would be desirable for the purposes of anomaly detection. Specifically, as seen in Figure 4.1a, the calculated intervals often split the true clusters (i.e.,

Algorithm 4 Mode-bounding Lloyd-Max

1: Inputs: α, k
2: Compute $B = k \cdot \alpha$ quantization intervals with bounds $\mathbf{M} = [m_1, \ldots, m_{B+1}]$ via Lloyd-Max.
3: **while** $|\mathbf{M}| > \alpha$ **do**
4: remove m_{j^*} from $\mathbf{M}$, with $j^* = \arg\min_j (m_j - m_{j-1})$

the intervals around the modes) of the source's pdf. This is undesirable, since, although data falling in the same mode are assumed to be similar, however, splitting a mode may yield a misinterpretation as of the data belonging to distinct subclusters being significantly different. From a statistical viewpoint, this is a problem of bounding the modes of a probability density function. To overcome this drawback, a modification of the Lloyd-Max quantizer is proposed below.

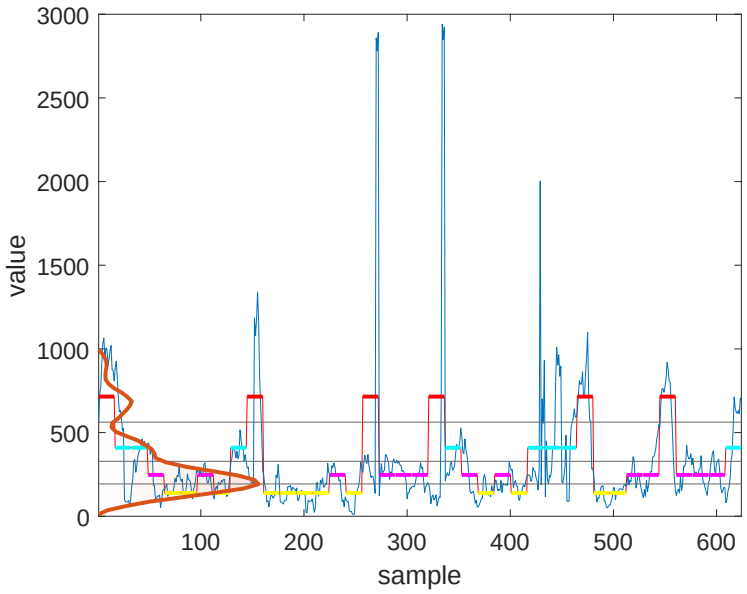
The proposed quantization method is a simple modification of the conventional Lloyd-Max quantizer, aiming at better detecting the modes rather than the quantiles of a probability density function. Specifically, let α be a predefined number of quantization intervals. The mode-bounding Lloyd-Max quantizer first estimates a number of quantization intervals $k \cdot \alpha$, $k \in \mathcal{N}$. Then, a merging of the smallest intervals with their neighbours is carried out iteratively until the α largest intervals are left. This process is summarized in Algorithm 4 and illustrated in Fig. 4.1b, in contrast with the conventional Lloyd-Max quantizer in Fig. 4.1a.

The idea behind the proposed quantizer is that a finer quantization will fine-slice the peaks of the modes in the pdf, as the density around the peaks is high. Subsequently, merging the smallest intervals implicitly merges the intervals around the peaks, leaving the boundaries of the modes intact. The effect is visualized in Fig. 4.2.

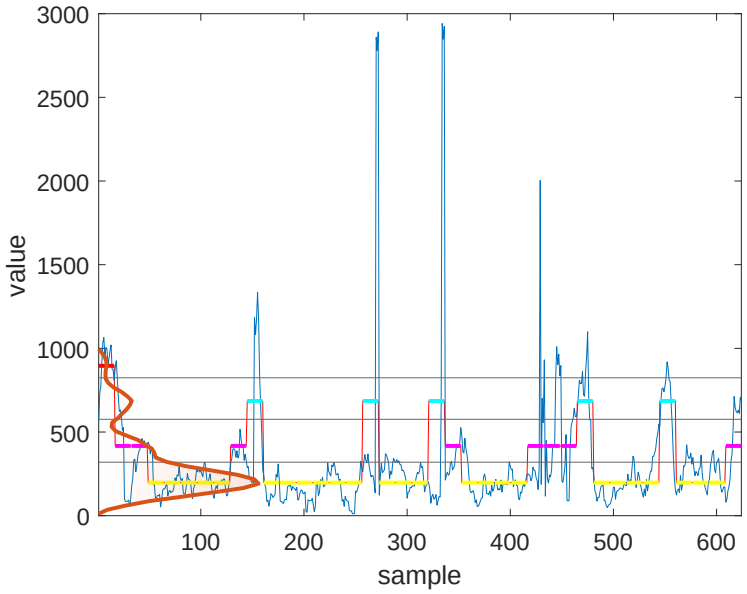
4.2 SAX-KL

The proposed anomaly detection method is applied in a lower-dimensional space by incorporating the pSAX variant described above, in conjunction with the KL GoF test overviewed in Sec. 2.2.1.

In particular, working in a sliding window fashion, given the alphabet size α and the dimensionality reduction ratio M/N, the current window of length N_w is first transformed into a symbolic sequence S of length $M_w = \frac{M}{N} \cdot N_w$. The transformation is carried out by pSAX integrated with the mode-bounding Lloyd-Max quantizer described in Sec. 4.1, while alternative quantization methods are compared in the experimental section 4.3.2.

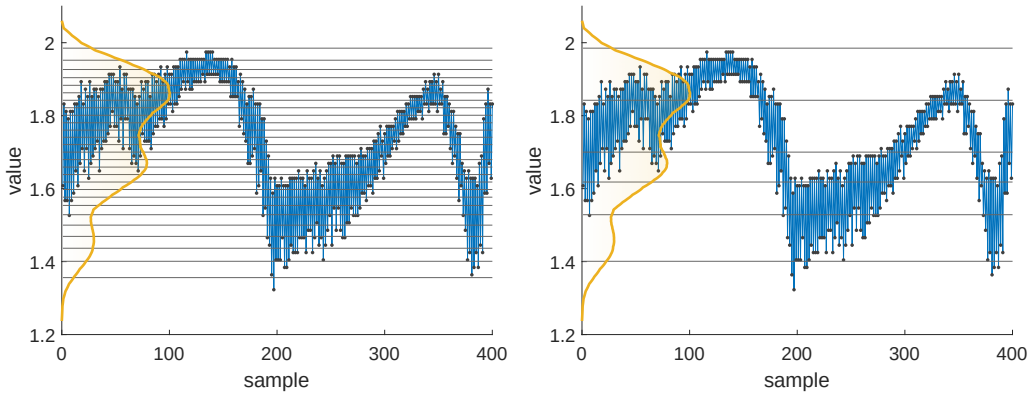


(a) pSAX with Lloyd-Max quantization



(b) pSAX with Mode-bounding Lloyd-Max quantization

Figure 4.1: Output of pSAX employing two different quantization options. Note that the dominant mode is splitted in two intervals by the conventional Lloyd-Max quantizer, while a more accurate bounding is achieved by the mode-bounding Lloyd-Max.



(a) Mode-bounding Lloyd-Max before interval merging

(b) Mode-bounding Lloyd-Max after interval merging

Figure 4.2: Illustration of the Mode-bounding effect. The output alphabet size is $\alpha = 7$, resulted after merging $4 \cdot 7 = 28$ initial intervals.

Having generated the symbolic sequence of length M for the current window, the probability distribution of the α alphabet symbols is calculated for the M-sized sequence. Then, the goodness-of-fit test described in Sec. 2.2.1 is applied to classify the window as anomalous or not. Here, the cardinality of the sample space of the symbols in S, which is required in Wilks' Theorem (2.47) for the definition of the chi-squared distribution, is equal to the alphabet size α.

Note that both α and M/N determine the degree of compressibility achieved by the symbolic sequence, and hence the computational and memory savings of the overall anomaly detection system. However, α and M/N affect the detector's efficiency in a different way. In the optimal case, the alphabet size should match the number of "states" in the given time series. For instance, a binary source with additive noise can be represented efficiently with a binary alphabet. Likewise, a CPU activity log may be represented adequately with an alphabet size equal to the expected number of activity states. On the other hand, the dimensionality reduction ratio should preserve the raw data patterns.

We emphasize again that the proposed online anomaly detection method is distribution-free, by not relying on any prior assumption on the underlying data distribution. Furthermore, it does not require access to past data, but only to the probability distributions of the past (symbolic) windows. Memory-wise, this is more efficient, since a window of length N_w is represented by only $\alpha \ll N_w$ numbers, i.e., the probability of appearance of each symbol.

4.3 Results

4.3.1 Performance metrics

In this section, the anomaly detection accuracy of the proposed method is evaluated. On the whole, the performance of anomaly detection methods is evaluated in terms of the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) rates. Remember that, in the context of hypothesis testing, a positive occurs when the null hypothesis is rejected, whereas a negative occurs when the null hypothesis is confirmed. Also, FP's and FN's are equivalent to Type I (2.37) and Type II (2.38) errors, respectively.

Standard performance metrics for anomaly detectors (classifiers in general) include the following information retrieval measures:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} , \tag{4.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} , \tag{4.2}$$

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} . \tag{4.3}$$

Precision measures the correctness of positives, whereas Recall measures the success in detecting anomalies. F-score is the harmonic mean of Precision and Recall, which provides an overall measurement of the performance.

The above metrics are suitable for anomaly detection over batches of data or subsequences over a time window of predefined length (e.g. packets in communication networks, daily stock market prices, traffic in rush hours, etc.). On the other hand, the accuracy of anomaly detection data which do not form predefined batches, such as real-time streaming data, cannot be evaluated directly with the above metrics. To alleviate this issue, the authors in [18] propose a benchmark algorithm (NAB) to tackle these limitations. The outcome is a scoring system tailored to streaming anomaly detectors, which contains a total of 58 synthetic and real-world streaming datasets with labeled anomalies. Moreover, the algorithm calculates three different scores by weighing Type I and Type II errors differently: (i) favoring fewer Type I errors, (ii) favoring fewer Type II errors, and (iii) a "standardized" score, that balances both Type I and Type II errors. The scores range between 0 and 100 (the higher the better).

A core concept of NAB is the definition of anomalous windows, i.e., windows centered on anomaly points, with which a true positive is scored according to how early or late within the window it is located (the earlier the better). Naturally,

the points that are classified as anomalous within an anomalous window are jointly accounted for as a single true positive. The length of the windows is set heuristically and separately for each dataset.

The concept of anomalous windows in NAB lead us to the idea of splitting the streaming data, following the same heuristics as NAB does, into equal-sized windows, either anomalous or anomaly-free, into which the detected anomalies are merged. Adopting this approach allows us to exploit the commonly used performance metrics defined by (4.1)-(4.3) by merging the positives and negatives within the imaginary windows.

In the following subsections, the performance of SAX-KL is evaluated for varying dimensionality reduction ratios. To this end, the NAB scores and the metrics in (4.1)-(4.3) (which are calculated according to the methodology described above) are used, as well as a collection of time series from pressure sensors in supplying water pipes. Doing so, we provide a complete assessment of the detector's performance.

4.3.2 Experimental evaluation: NAB

In this subsection, the collection of datasets provided by NAB are employed to evaluate the performance of the proposed method. The results are averaged over 100 Monte Carlo iterations, although the deviation across the iterations was not significant. The NAB scores achieved by the proposed method are compared with some of the currently best performing anomaly detection algorithms, whose scores are obtained directly from the online repository[1] maintained by the authors of [18].

Regarding KL GoF and SAX-KL, the following parameters setting is used for all datasets: window length $N_w = 48$, alphabet size $\alpha = 7$, and quantiles multiplier for the mode-bounding Lloyd-Max (Alg. 4) $k = 4$. In order to set the significance level γ, we experimented with (2.48) by fixing $\alpha = 7$. In general, we ended up rising the initial computations of significance levels, especially when the PAA length was in the higher end. Note that the aforementioned online repository lists the values of the KL GoF under the name "Relative Entropy", and with scores significantly lower than those reported herein. The reason is that the anomaly detector they employed is executed for $\alpha = 5$ and $\gamma = 0.01$, whereas we found out that our setting achieves higher scores.

As it can be seen in Table 4.1, when no dimensionality reduction is applied, the proposed method has improved performance compared to the conventional KL GoF, whilst it clearly competes most of the currently best performing detectors.

[1] `https://github.com/numenta/NAB/#scoreboard`

Detector	Standard	Low FP	Low FN	F-score
Numenta HTM	70.1	63.1	74.3	0.5908
CAD OSE	69.9	67.0	73.2	0.6970
SAX-KL (mode-bounding)	66.3	61.2	70.1	0.5542
KL GoF (uniform)	62.0	57.4	65.6	0.5574
SAX-KL (L-M)	60.4	51.4	65.9	0.3944
earthgecko Skyline	58.2	46.2	63.9	0.4153
KNN CAD	58.0	43.4	64.8	0.3758
SAX-KL (k-means)	57.2	47.9	62.8	0.3685
Random Cut Forest	51.7	38.4	59.7	0.4728
Twitter ADVec v1.0.0	47.1	33.6	53.5	0.4151
SAX-KL (Gaussian equipr.)	43.1	36.5	47.1	0.3150

Table 4.1: NAB scores. The proposed SAX-KL method has been set with $M/N = 1.0$ (i.e., no dimensionality reduction). The parameters of SAX-KL and KL GoF are the same and optimized for NAB's datasets: $N_w = 48$, $\alpha = 7$, $\gamma = 0.002$. In parentheses, the quantization scheme, where applied.

The scores also highlight the superiority of the mode-bounding variation of Lloyd-Max (L-M) for the purpose of statistical inference against both the conventional L-M and the k-means (used in aSAX), whereas the assumption of Gaussian statistics (i.e. as in conventional SAX) reduces the efficacy of the detector tremendously.

More importantly, the proposed method uniquely enables anomaly detection in a lower-dimensional space, due to pSAX. The second experiment investigates the effect of the degree of dimensionality reduction on the performance of SAX-KL. Table 4.2 presents the anomaly detection performance of SAX-KL by varying the dimensionality reduction ratio $M/N \in [1/48, 1/1]$ (from large to low dimensionality reduction), in terms of the NAB scores, as well as the Precision and Recall. Furthermore, Fig. 4.3 shows the respective average F-score as a function of M/N.

Notably, according to the NAB scores in Table 4.2, the performance of the proposed method in the lower-dimensional space is still as good as the best performing detectors (Table 4.1), even for large dimensionality reduction. An interesting observation is that the performance of the detector does not decrease monotonically with the dimensionality reduction. A more thorough study of the effect of the dimensionality reduction ratio on the detector's performance is left as a future work.

				NAB Scores				
M/N	**N_w**	**γ**	**α**	**Stand.**	**Low FP**	**Low FN**	**Precis.**	**Recall**
1/1	48	0.002	7	66.3	61.2	70.1	0.4280	0.7863
1/2	48	0.022	7	65.0	59.5	68.9	0.4023	0.7766
1/3	48	0.052	7	64.0	58.3	68.2	0.3933	0.7743
1/4	48	0.082	7	65.0	59.3	69.4	0.4029	0.7891
1/6	48	0.134	7	63.5	57.8	67.8	0.3666	0.7295
1/8	48	0.174	7	62.6	56.9	66.9	0.3875	0.7643
1/12	48	0.234	7	60.4	55.4	64.3	0.3865	0.7296
1/16	48	0.276	7	59.9	54.9	64.0	0.3723	0.7309
1/24	48	0.337	7	59.8	55.4	63.9	0.3835	0.7297
1/32	64	0.337	7	58.1	53.6	62.3	0.3644	0.7136
1/48	96	0.337	7	54.5	50.0	58.6	0.3235	0.6767

Table 4.2: Performance of the proposed SAX-KL method vs. dimensionality reduction ratio (M/N).

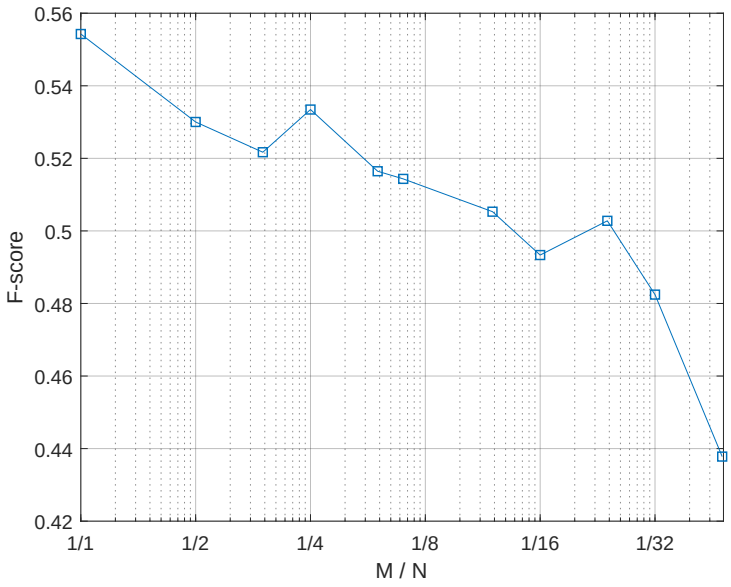


Figure 4.3: Average F-score vs. dimensionality reduction ratio.

Algorithm Part	Space Complexity	Time Complexity
quantization intervals	$O(\alpha - 1)$	
PAA	$O(M \log \alpha)$	$O(N)$
distributions	$O(M\alpha)$	$O(M)$
KL divergence	$O(\alpha)$	$O(M\alpha)$

Table 4.3: SAX-KL Complexity

Detector	Running time (ms)
Numenta HTM	123.76
CAD OSE	24.48
SAX-KL ($k = 4$), $M/N = 1/1$	4.10
SAX-KL ($k = 4$), $M/N = 1/2$	1.95
SAX-KL ($k = 4$), $M/N = 1/4$	1.06
SAX-KL ($k = 4$), $M/N = 1/8$	0.68
KL GoF (uniform)	2.54
earthgecko Skyline	93.05

Table 4.4: Running time for the "machine_temperature_system_failure" dataset

4.3.3 Computational and space requirements

The running time of SAX-KL and KL GoF method is exactly the same when $M/N = 1.0$, since the training phase of the KDE step is carried out only once during initialization, and thus can be disregarded in the subsequent application of the method on the streaming data. Practically, during the initial application of the detector, a better approximation of the true distribution function can be computed as samples come in and update the quantization intervals.

Overall, the method requires small memory and computationally benefits from the dimensionality reduction. The space and time complexity of SAX-KL can be seen in Table 4.3. Notice that the efficiency of SAX-KL increases as M decreases and that when $M = N$, the efficiency of SAX-KL reduces to that of KL GoF.

As an example, we ran the top 5 detectors from Table 4.1 on the real-world dataset "machine_temperature_system_failure" from NAB's collection, which contains 22695 samples, on an Intel i7-6700@3.8GHz. The results can be seen in Table 4.4.

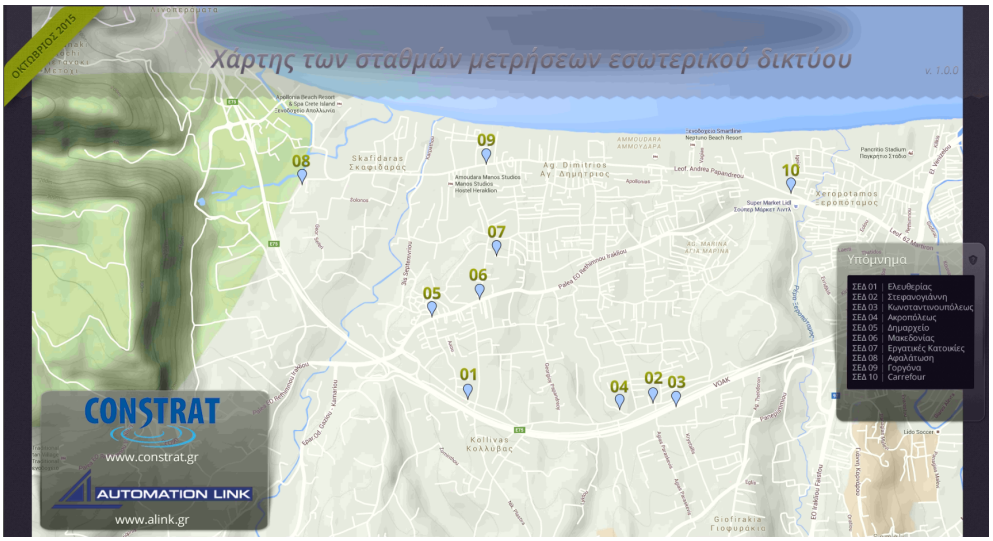


Figure 4.4: Topology of the regions where the pressure of the water inside the input and output pipelines is recorded. Picture provided by CONSTRAT Ltd.

4.3.4 Experimental evaluation: Water data

This subsection presents the results of SAX-KL on a collection of datasets of water pressure time series collected from sensors inside supplying water pipes in Heraklion, Crete. The sensors are grouped into pairs, where each pair consists of an input and an output pressure reading to, and from, a specific region. Also, some of the sensors are located in different regions but are connected in series, thus reading correlated signals. As such, the datasets are good candidates for testing our method to multi-dimensional data. To this end, we transformed each time series independently via pSAX and then applied our anomaly detector on the multi-dimensional symbolic sequence output. The symbols of the multi-dimensional sequence are formed by the combination of the concurrent individual, uni-dimensional, codewords. The topology of the sensors is depicted in Fig. 4.4, where the regions corresponding to each pair of input and output sensors are marked.

Unfortunately, there is no ground truth available for anomaly events in our data. Nevertheless, these can easily be observed by sudden upward or downward spikes on the data. An upwards spike indicates that the water flow is obstructed by some foreign material. On the other hand, a downward spike indicates a leak. We emphasize that the proposed method is context-unaware and as such, only implicitly can classify sudden spikes as anomalous. Our results show that it succeeds to do so.

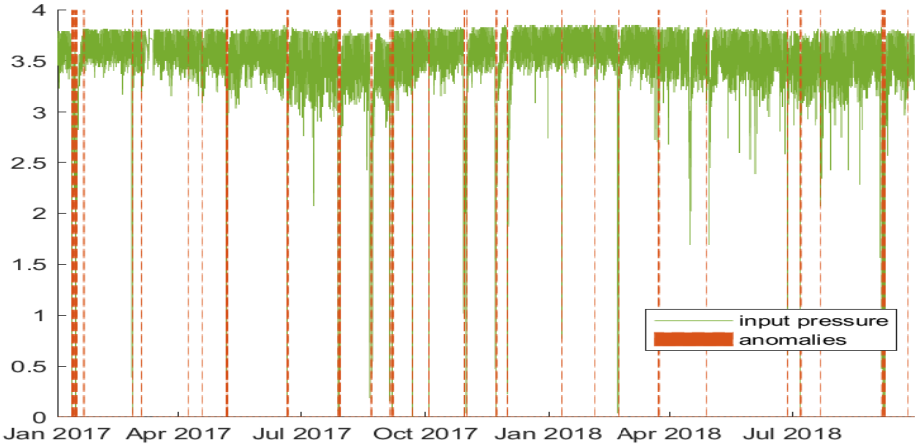
As of the parameter setting, we kept the same settings as in the previous subsection, i.e. $N_w = 48, \alpha = 7, \gamma = 0.002$. Here, the window length $N_w = 48$ is equivalent to 12 hours. Moreover, the KDE was allowed to train on the first full year of recorded data, simulating the scenario of training from historic data. Lastly, the null hypothesis is defined to be the distributions of the windows of only the past year from the current window. This has the following benefits: i) it prevents the method from over-fitting to previous anomalies, ii) allows the method to adapt faster to more recent data, iii) reduces the space requirements by deleting the old distributions from the memory.

At first, pSAX is set with $M/N = 1/1$, i.e. with no dimensionality reduction. Because the sampling rate is low (1 sample per 15 minutes), it might not be necessary to reduce the dimensionality whatsoever, but we provide results with dimensionality reduction at the end of this subsection. Nevertheless, SAX-KL benefits power- and memory-wise from the quantization step.

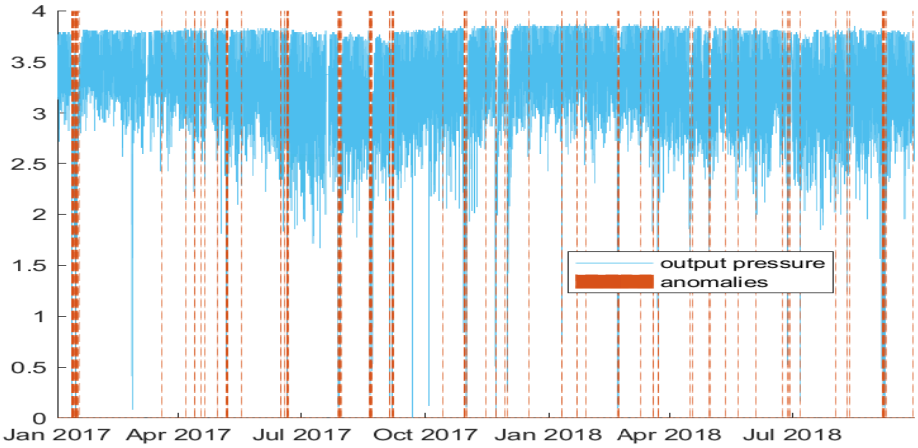
Figure 4.5 depicts the pressure and the SAX-KL results for the regions #5 and #7, numbered as indicated in Fig. 4.4. It can be seen that most of the sudden spikes we discussed are indeed flagged as anomalous events. In addition, processing jointly the input and output data increases the accuracy of the detector in the general case (fewer false positives, while the true positives are marginally increased).

One problem that arises in water monitoring is to infer whether anomalies affect neighbouring regions, other than the one where the anomaly initiated. In that case, the problem concerns one of the major pipelines and might affect other places that are not being monitored. To answer this question, we experimented with neighboring regions, in our example with the regions #1 and #5, of which the pipelines are interconnected and thus, anomalies might be propagated. To infer whether anomalies concur in both regions, we computed the element-wise product of the pSAX-symbolic sequences of all four (two input and two output) pressure time series from the regions and ran the SAX-KL on it. Here, in order to allow the product calculation, the codewords for each of the symbolic sequences are the integers $0, 1 \ldots, \alpha - 1$. Anomalies on this product denote anomalies that affect the whole neighbour of the two regions. Because it is natural to have delays in the anomalies between the two regions, we allowed our pSAX to reduce the dimensionality of the raw time series with dimensionality reduction ratio $M/N = 1/4$, thus merging 4 adjacent samples, which is one hour up to the current sample. In addition, due to the fact that the task here is to find more coarse anomalies in our data, we slightly reduced the alphabet size of pSAX (to 6 from 7), which resulted to much fewer false positives. The results are shown in Fig. 4.6.

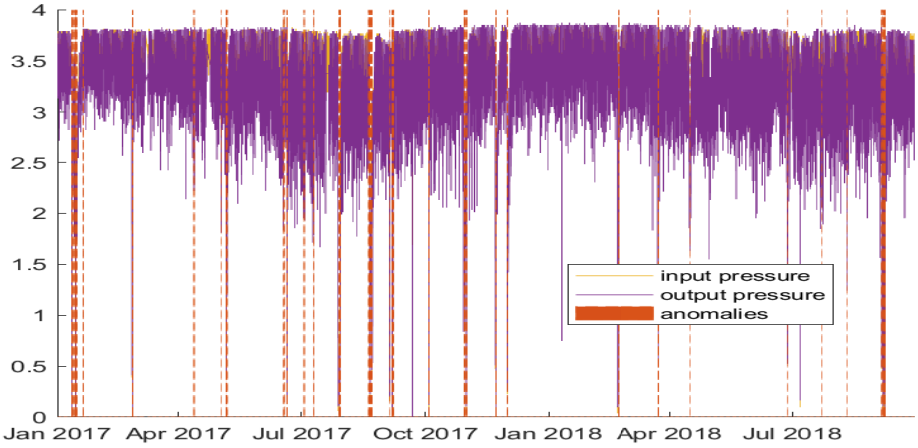
Lastly, we experimented with the dimensionality reduction ratio. The reference



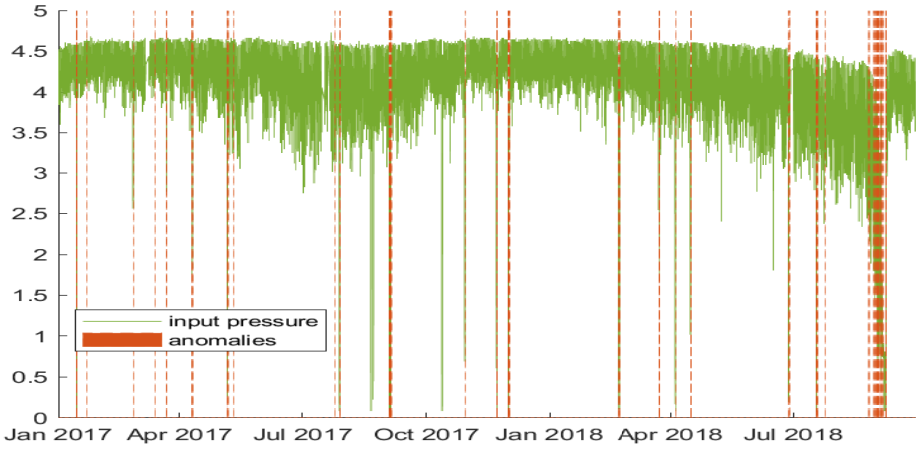
(a) Input of region #5



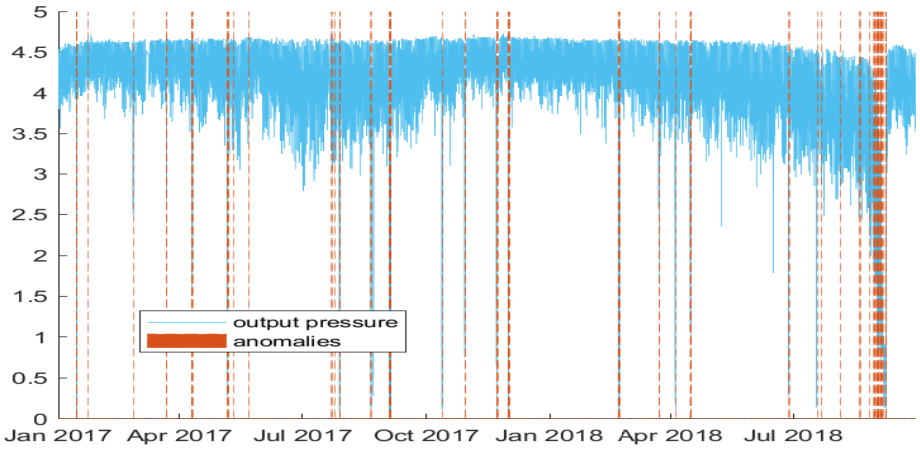
(b) Output of region #5



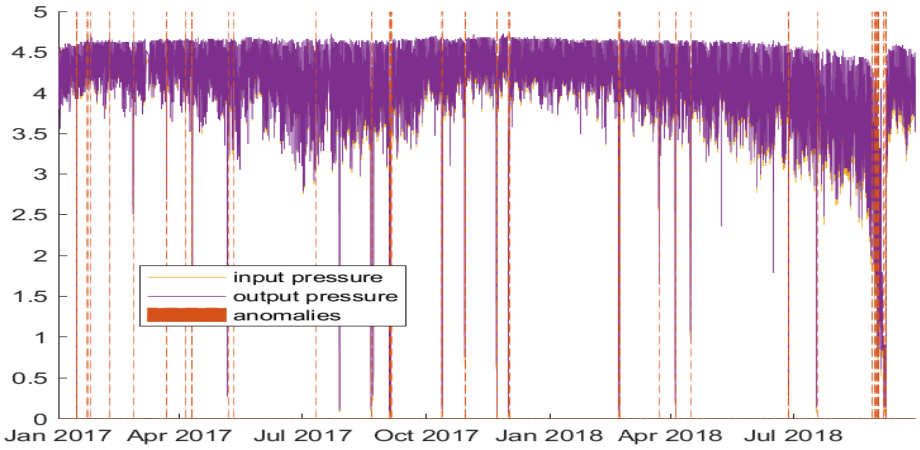
(c) Input and output of region #5



(d) Input of region #7

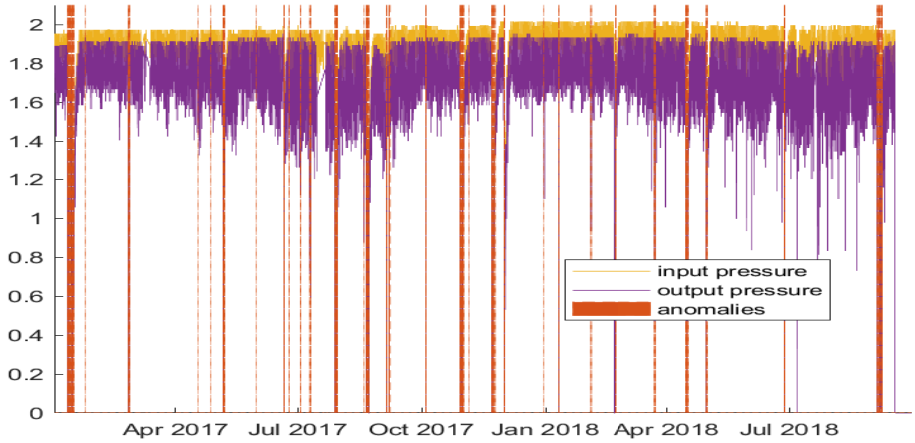


(e) Output of region #7

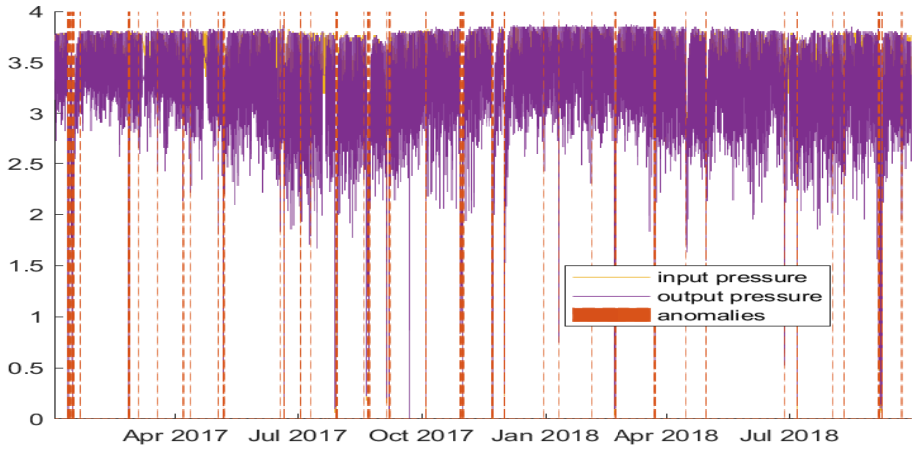


(f) Input and output of region #7

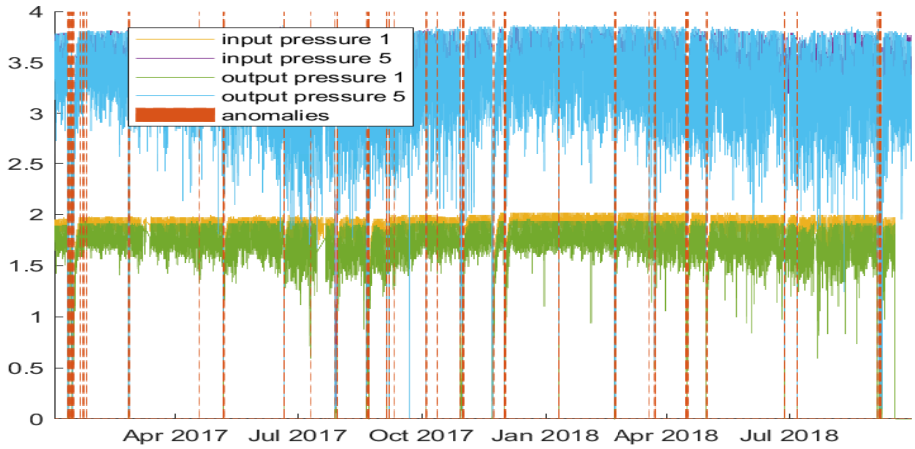
Figure 4.5: SAX-KL results in uni- and two-dimensional water pressure data.



(a) Input and output of region #1



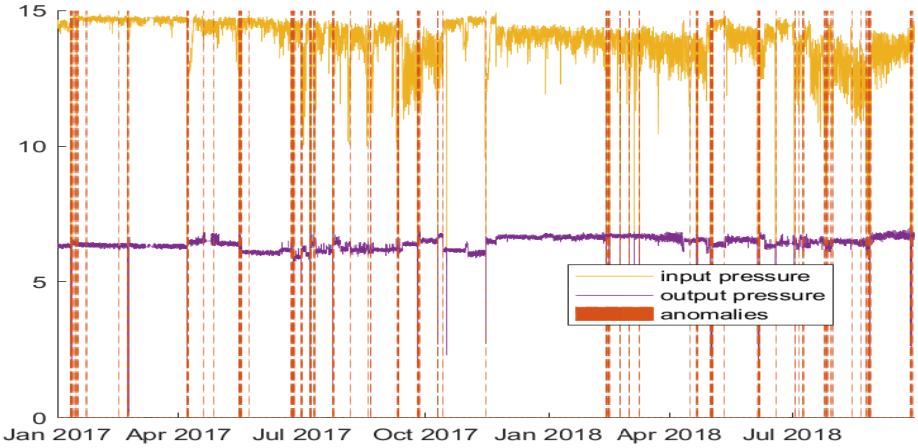
(b) Input and output of region #5



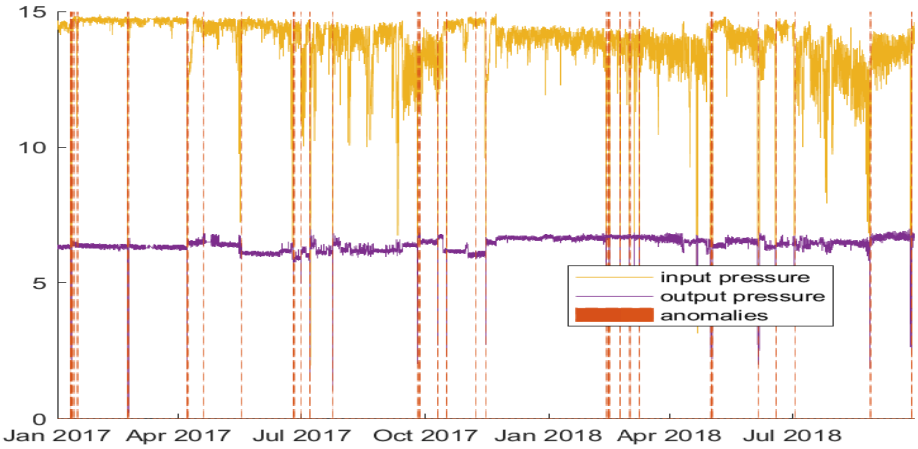
(c) Input and output of regions #1 and #5

Figure 4.6: SAX-KL results for concurrent anomalies detection.

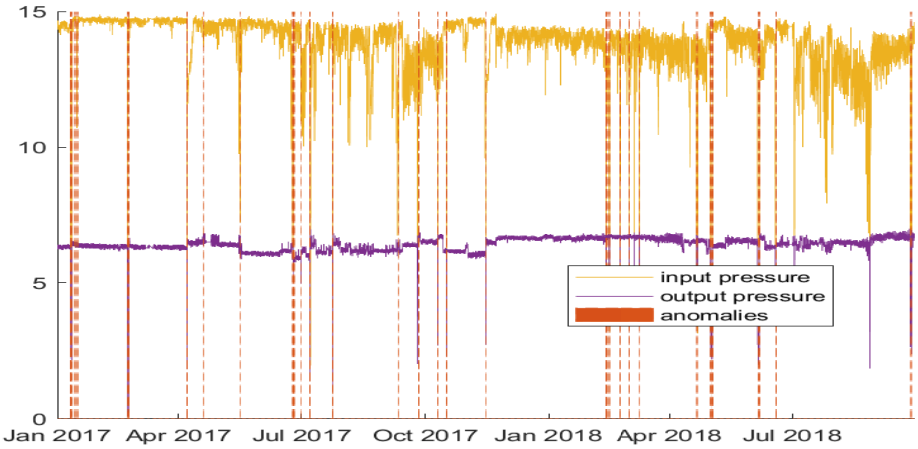
dataset in this experiment is from the region #8. Figure 4.7 illustrates the results for $M/N \in \{1/1, 1/2, 1/4\}$ (from no to moderate dimensionality reduction). It can be seen that most of the major changes in water pressure are captured from the proposed SAX-KL method, regardless of the reduced dimensionality.



(a) $M/N = 1/1$



(b) $M/N = 1/2$



(c) $M/N = 1/4$

Figure 4.7: SAX-KL results for region #8 for different settings of M/N.

Chapter 5

Conclusion and future work

This work proposes a new symbolic representation method, called pSAX, which generalizes previous SAX-based techniques by adapting directly to the underlying probability distribution of a given time series, without any prior model assumption for the data generating process. To this end, the proposed pSAX method exploits the power of KDEs for accurate density estimation by a restricted amount of training samples, with the efficiency of Lloyd-Max quantization for optimizing the intervals and associated codewords. Furthermore, two novel distance measure in the lower-dimensional space of symbolic sequences are introduced, which can be employed in data mining tasks. Our experiments revealed the superiority of pSAX, compared to alternative SAX-based techniques, in terms of representation accuracy. Notably, the proposed methodology can be coupled with other variants of SAX in a straightforward way, to improve their performance in the case of non-Gaussian data.

Furthermore, the proposed dimensionality reduction method is employed to speed-up a statistical anomaly detector. While reducing the dimensionality enabled faster processing of the time series, the conventional Lloyd-Max quantizer was not able to provide a good basis for statistical inference. To tackle this issue, an alternative quantization scheme is proposed, based on a simple heuristic variation of the Lloyd-Max, which clusters the data according to the modes of the density function.

The proposed anomaly detector, named by SAX-KL, is evaluated by means of the Numenta Anomaly Benchmark, an anomaly detection benchmark tailored to streaming data. In addition, the proposed method is also tested on a collection of water pipelines pressure time series. In the latter case, SAX-KL is also tested for two-dimensional time series, while a method is shown to quickly find anomalies in multiple correlated time series. Overall, the proposed anomaly detector achieves

similar performance, or even it outperforms, the best performing methods available for streaming data. Most importantly, this is also the case even for large dimensionality reduction ratios (i.e., highly compressed data). It is also demonstrated to handle efficiently multi-dimensional data and data from correlated sources.

Currently, time series and their statistics are considered in a static framework by our methods. As a further generalization of pSAX, an online extension over sliding windows is under investigation for real-time applications. To this end, we are interested in tracking the evolution of data statistics across time, while also applying a dynamic quantization scheme, under execution time constraints. Finally, the efficacy of our proposed distance measure will also be evaluated in data indexing scenarios, which is feasible due to the lower bounding property.

Appendix A

Proofs

A.1 Theorem 3.1.1

Proof. Denote with $G(x_i)$ the PMF of the quantized X under the wrong distribution $g(x)$. Assuming optimal coding under $G(x_i)$, the expected code length is

$$\begin{aligned}
\mathbb{E}[l(X^\Delta)] &= \sum_{x_i} P(x_i) \lceil \log \frac{1}{G(x_i)} \rceil \\
&< \sum_{x_i} P(x_i) \left(\log \frac{1}{G(x_i)} + 1 \right) \\
&= \sum_{x_i} P(x_i) \left(\log \frac{P(x_i)}{G(x_i)} \frac{1}{P(x_i)} + 1 \right) \\
&= \sum_{x_i} P(x_i) \log \frac{P(x_i)}{G(x_i)} + \sum_{x_i} P(x_i) \left(\log \frac{1}{P(x_i)} + 1 \right) \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)\Delta}{g(x_i)\Delta} + \sum_{x_i} f(x_i)\Delta \left(\log \frac{1}{f(x_i)\Delta} + 1 \right) \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)}{g(x_i)} - \sum_{x_i} f(x_i)\Delta \left(\log(f(x_i)\Delta) - 1 \right) \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)}{g(x_i)} - \sum_{x_i} f(x_i)\Delta \log f(x_i) - \sum_{x_i} f(x_i)\Delta (\log \Delta - 1) \\
&\longrightarrow D(f \parallel g) + h(X) + \log \frac{1}{\Delta} + 1 \ , \quad \text{as } \Delta \to 0 \ ,
\end{aligned} \tag{A.1}$$

where, in the last line, Riemann integrability is assumed.

Similarly,

$$\begin{aligned}
\mathbb{E}[l(X^{\Delta})] &= \sum_{x_i} P(x_i) \lceil \log \frac{1}{G(x_i)} \rceil \\
&\geq \sum_{x_i} P(x_i) \log \frac{1}{G(x_i)} \\
&= \sum_{x_i} P(x_i) \log \frac{P(x_i)}{G(x_i)} \frac{1}{P(x_i)} \\
&= \sum_{x_i} P(x_i) \log \frac{P(x_i)}{G(x_i)} + \sum_{x_i} P(x_i) \log \frac{1}{P(x_i)} \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)\Delta}{g(x_i)\Delta} + \sum_{x_i} f(x_i)\Delta \log \frac{1}{f(x_i)\Delta} \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)}{g(x_i)} - \sum_{x_i} f(x_i)\Delta \log(f(x_i)\Delta) \\
&= \sum_{x_i} f(x_i)\Delta \log \frac{f(x_i)}{g(x_i)} - \sum_{x_i} f(x_i)\Delta \log f(x_i) - \sum_{x_i} f(x_i)\Delta \log \Delta \\
&\longrightarrow D(f \parallel g) + h(X) + \log \frac{1}{\Delta} , \quad \text{as } \Delta \to 0 .
\end{aligned} \tag{A.2}$$

□

A.2 Proposition 3.1.1

Proof. From Lemma 3.1.3, we have that $D_{KL}(f \parallel g) = h(f,g) - h(f)$. Also,

$$\begin{aligned}
h(f,g) &= -\int f(x) \log g(x) dx \\
&= -\int f(x) \log \frac{1}{\sqrt{2\pi\sigma_g^2}} e^{-(x-\mu_g)^2/(2\sigma_g^2)} dx \\
&= -\int f(x) \left(\log \frac{1}{\sqrt{2\pi\sigma_g^2}} - \frac{(x-\mu_g)^2}{2\sigma_g^2} \right) dx \\
&= -\int f(x) \log \frac{1}{\sqrt{2\pi\sigma_g^2}} dx + \int f(x) \frac{(x-\mu_g)^2}{2\sigma_g^2} dx
\end{aligned} \tag{A.3}$$

Then, if $\mu_x = \mu_g = \mu$,

$$\begin{aligned} h(f,g) &= \log \sigma_g \sqrt{2\pi} + \frac{1}{2\sigma_g^2}\sigma_x^2 \\ &= \log \sigma_g \sqrt{2\pi} + \frac{1}{2}\frac{\sigma_x^2}{\sigma_g^2} \ , \end{aligned} \tag{A.4}$$

where in the second term of the first line of (A.4), the definition of the variance is used. □

Bibliography

[1] Saeed Aghabozorgi and Ying Wah Teh. Stock market co-movement assessment using a three-phase clustering method. *Expert Systems with Applications*, 41(4, Part 1):1301 – 1314, 2014.

[2] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms*, volume 8, pages 1027–1035, 01 2007.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability, 2nd Edition.* Athena Scientific, 07 2008.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41, 07 2009.

[5] F.L. Chung, Tak-Chung Fu, Robert Luk, and Vincent Ng. Flexible time series pattern matching based on perceptually important points. *Intl' Joint Conf. on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*, pages 1–7, 01 2001.

[6] K. Cohen and Q. Zhao. Quickest anomaly detection: A case of active hypothesis testing. In *2014 Information Theory and Applications Workshop (ITA)*, pages 1–5, 2014.

[7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*, chapter 5, pages 103–158. John Wiley & Sons, Ltd, 2005.

[8] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. *KDD Proc*, 98, 07 1998.

[9] J. Durbin and S. J. Koopman. Time series analysis of non-gaussian observations based on state space models from both classical and bayesian perspectives. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(1):3–56, 2000.

[10] V. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability and its Applications*, 14(1):153–158, 1969.

[11] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *SIGMOD Rec.*, 23(2):419–429, 1994.

[12] Gary Grunwald, Rob Hyndman, Leanna Tedesco, and Richard Tweedie. Theory & methods: Non-gaussian conditional linear ar(1) models. *Australian & New Zealand Journal of Statistics*, 42:479 – 495, 12 2000.

[13] Y. Huang and P. S. Yu. Adaptive query processing for time-series data. In *Proc. Fifth ACM SIGKDD Intl' Conf. on Knowledge Discovery and Data Mining*, KDD '99, pages 282–286, New York, NY, USA, 1999. ACM.

[14] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957.

[15] Eamonn Keogh, Kaushik Chakrabarti, Sharad Mehrotra, and Michael Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *ACM SIGMOD Record*, volume 30, pages 151–162, 06 2001.

[16] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3, 01 2002.

[17] T. Kohonen and P. Somervuo. Self-organizing maps of symbol strings. *Neurocomputing*, 21(1):19 – 30, 1998.

[18] A. Lavin and S. Ahmad. Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark. In *2015 IEEE 14th Intl' Conf. on Machine Learning and Applications (ICMLA)*, pages 38–44, Dec 2015.

[19] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, pages 2–11, 01 2003.

[20] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15:107–144, 08 2007.

[21] Battuguldur Lkhagva, Yu Suzuki, and Kyoji Kawagoe. Extended SAX: extension of symbolic aggregate approximation for financial time series data representation. *Proc. of IEICE the 17th Data Engineering Workshop*, 01 2006.

[22] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. on Information Theory*, 28(2):129–137, 1982.

[23] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. 1d-SAX: A novel symbolic representation for time series. In Allan Tucker, Frank

Höppner, Arno Siebes, and Stephen Swift, editors, *Advances in Intelligent Data Analysis XII*, pages 273–284, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[24] H. B. Mann and A. Wald. On the choice of the number of class intervals in the application of the chi square test. *Annals of Mathematical Statistics*, 13(3):306–317, 09 1942.

[25] J. Max. Quantizing for minimum distortion. *IRE Trans. on Information Theory*, 6(1):7–12, 1960.

[26] V. Megalooikonomou, Q. Wang, G. Li, and C. Faloutsos. A multiresolution symbolic representation of time series. In *21st Intl' Conf. on Data Engineering (ICDE'05)*, pages 668–679, 2005.

[27] Vasileios Megalooikonomou, Guo Li, and Qiang Wang. A dimensionality reduction technique for efficient similarity analysis of time series databases. In *Proc. of the 13th ACM Intl' Conf. on Information and Knowledge Management*, CIKM '04, page 160–161, New York, NY, USA, 2004. Association for Computing Machinery.

[28] Vasileios Megalooikonomou, Qiang Wang, Guo Li, and Christos Faloutsos. A multiresolution symbolic representation of time series. In *Proc. of the 21st Intl' Conf. on Data Engineering*, ICDE '05, page 668–679, USA, 2005. IEEE Computer Society.

[29] Joseph Michalowicz, Jonathan Nichols, and Frank Bucholtz. Calculation of differential entropy for a mixed gaussian distribution. *Entropy*, 10, 09 2008.

[30] Clayton Miller, Zoltán Nagy, and Arno Schlueter. Automated daily pattern filtering of measured building performance data. *Automation in Construction*, 49:1 – 17, 2015.

[31] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[32] Patricia Ordóñez, Tom Armstrong, Tim Oates, and James C Fackler. Using modified multivariate bag-of-words models to classify physiological data. In *Proc. - IEEE Intl' Conf. on Data Mining, ICDM*, pages 534–539, 2011.

[33] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[34] Chang-Shing Perng, Haixun Wang, Sylvia Zhang, and Douglas Jr. Landmarks: a new model for similarity-based pattern querying in time series databases. In *Proc. - Intl' Conf. on Data Engineering*, pages 33–42, 01 2000.

[35] Ninh D. Pham, Quang Loc Le, and Tran Khanh Dang. Two novel adaptive symbolic representations for similarity search in time series databases. *2010 12th Intl' Asia-Pacific Web Conf.*, pages 181–187, 2010.

[36] Chotirat Ratanamahatana, Eamonn Keogh, Anthony J. Bagnall, and Stefano Lonardi. A novel bit level time series representation with implication of similarity search and clustering. In Tu Bao Ho, David Cheung, and Huan Liu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 771–777, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[37] Philippe Rigolett and Regis Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.

[38] Jin Shieh and Eamonn Keogh. iSAX: Indexing and mining terabyte sized time series. *Proc. of the ACM SIGKDD Intl' Conf. on Knowledge Discovery and Data Mining*, pages 623–631, 08 2008.

[39] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[40] Youqiang Sun, Jiuyong Li, Jixue Liu, Bingyu Sun, and Christopher Chow. An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing*, 138:189–198, 08 2014.

[41] G. Thatte, U. Mitra, and J. Heidemann. Parametric methods for anomaly detection in aggregate traffic. *IEEE/ACM Trans. on Networking*, 19(2):512–525, 2011.

[42] C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, and K. Schwan. Statistical techniques for online anomaly detection in data centers. In *12th IFIP/IEEE Intl' Symp. on Integrated Network Management (IM 2011) and Workshops*, pages 385–392, 2011.

[43] Yi Wang, Qixin Chen, Chongqing Kang, and Qing Xia. Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE Trans. on Smart Grid*, 7:2437– 2447, 09 2016.

[44] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics*, 9(1):60–62, 03 1938.

[45] Kiyoung Yang and Cyrus Shahabi. A pca-based similarity measure for multivariate time series. In *Proc. of the 2nd ACM Intl' Workshop on Multimedia Databases*, MMDB '04, page 65–74, New York, NY, USA, 2004. Association for Computing Machinery.

[46] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. *Proc. of the 26th Intl' Conf. on Very Large Data Bases, VLDB'00*, pages 385–394, 01 2000.

[47] J. Zhang and I. C. Paschalidis. Statistical anomaly detection via composite hypothesis testing for markov models. *IEEE Trans. on Signal Processing*, 66(3):589–602, 2018.