Towards optimal symbolization for time series comparisons

Gavin Smith
Horizon Digital Economy Research
The University of Nottingham, UK
gavin.smith@nottingham.ac.uk

James Goulding
Horizon Digital Economy Research
The University of Nottingham, UK
james.goulding@nottingham.ac.uk

Duncan Barrack
Horizon Digital Economy Research
The University of Nottingham, UK
duncan.barrack@nottingham.ac.uk

***Abstract*—The abundance and value of mining large time series data sets has long been acknowledged. Ubiquitous in fields ranging from astronomy, biology and web science the size and number of these datasets continues to increase, a situation exacerbated by the exponential growth of our digital footprints. The prevalence and potential utility of this data has led to a vast number of time-series data mining techniques, many of which require symbolization of the raw time series as a pre-processing step[1] for which a number of well used, pre-existing approaches from the literature are typically employed. In this work we note that these standard approaches are sub-optimal in (at least) the broad application area of time series comparison leading to unnecessary data corruption and potential performance loss before any real data mining takes place. Addressing this we present a novel quantizer based upon optimization of comparison fidelity and a computationally tractable algorithm for its implementation on big datasets. We demonstrate empirically that our new approach provides a statistically significant reduction in the amount of error introduced by the symbolization process compared to current state-of-the-art. The approach therefore provides a more accurate input for the vast number of data mining techniques in the literature, providing the potential of increased real world performance across a wide range of existing data mining algorithms and applications.**

I. Introduction

Time series are an exceptionally common form of big data, fuelled by the increasing proportions of our daily lives that are logged and recorded. Research into how best to extract value from this data has produced an exceptional array of data mining techniques for classifying, clustering and predicting. However, a common pre-processing step in many of these is to convert continuous time series into symbolic representations. This is implemented for a variety of reasons: for big data it is often crucial due to computational and storage constraints, but it is also often used to reduce noise, enhance interpretation, or simply to allow application of algorithms designed specifically for discrete domains [1]–[3]. Due in part to the extensive use of symbolization, it is often thought of as a solved problem. We will show in this paper that this is far from the case and that current methods of symbolization are often not optimal for the task they are being used for leading to information loss and performance degradation in many machine learning algorithms.

The symbolization (or *quantization*) of time series is certainly a well-studied problem within information theory, and for many the process equates to reducing *reconstruction error* which is optimized by minimizing the mean squared error (MSE) between the original time series and its quantized form[2]. But this only considers a single application - signal reconstruction - and while MSE is optimal for that case, it is *sub-optimal* for a broad range of other applications. In particular it is sub-optimal for the most common use of symbolized time series within data mining: *time series similarity search*. This seems an important oversight: similarity search is the cornerstone of many important subproblems such as time series clustering, classification, motif discovery and anomaly detection. We argue therefore that it is not *reconstruction* fidelity of times series which should be of most concern, but *comparison* fidelity. We develop this notion, resulting in a novel quantization technique for data mining which we refer to as an *Independent Comparison Error* (ICE) quantizer.

We show through empirical experiments that our implementation of ICE outperforms state-of-the-art alternatives (to statistically significant levels and over a range of time series lengths and symbol set cardinalities) - despite not being strictly optimal due to simplifications invoked for purposes computational tractability. This approach provides the main contribution of our paper, covering a wide range of real world applications and providing a theoretical underpinning upon which additional algorithms may be derived in the future.

II. Problem statement

This work considers the task of learning an optimal quantizer for univariate time series where the quantized data is then used for time series comparisons. Univariate time series are an extremely common digital commodity with examples including measures of usage (e.g. household energy or hourly website traffic), measures of communication (e.g. daily email traffic) and movement data (encoded, for example, as relative displacement) [6]. Note that the task of learning an optimal quantizer is distinct from the process of actually quantizing that time series with it - once a quantizer has been learnt it is

[1] for example over 90 recent data mining publications used Symbolic Aggregation Approximation (SAX) (see http://www.cs.ucr.edu/∼eamonn/SAX.htm.)

[2] There are of course other proposed notions of *good* quantization including the maximisation of symbol entropy (e.g. the often used SAX method [1] and it's subsequent refinements [4], [5]) or the optimisation of the end result of classifiers for signal detection. These are further discussed in section III.

Published in the Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW 2014).
IEEE Xplore DOI: http://dx.doi.org/10.1109/ICDMW.2013.59

trivial (i.e. exhibits constant time complexity) to apply, since a quantizer is merely a non-linear mapping from a domain of high cardinality (typically continuous) to discrete domain of significantly lower cardinality.

The specific quantization problem addressed is that of finding an $m-$level scalar quantizer $Q(x)$, where $Q(x)$ is a zero-memory nonlinear mapping that takes a real valued scalar input, x, and maps it to one of m values based on which of the m quantization intervals contains the input. Formally:

$$Q(x) = \begin{cases} q_1 & B_0 < x \leq B_1 \\ q_2 & B_1 < x \leq B_2 \\ \vdots & \\ q_m & B_{m-1} < x \leq B_m \end{cases} \tag{1}$$

where $B_0 = -\infty$ and $B_m = \infty$.

Consider a set of N univariate time series $\mathcal{T} = \langle T_1, T_2, \ldots, T_N \rangle$ where the i^{th} time series $T_i = \langle T_{i1}, T_{i2}, \ldots \rangle$ is indexed by the natural numbers. The quantization of that time series involves the repeated application of the quantizer to each point in the time series, $\hat{Q}(T_i) = \langle Q(T_{i1}), Q(T_{i2}), \ldots \rangle$. The problem is then, given some data and an objective function, to find the symbol values (the set of q_i's) and boundary values (the set of B_i's) such that the objective function is minimal, and hence represents the optimal quantizer. Specifically, to solve the optimisation problem:

$$\operatorname*{argmin}_{Q} \mathcal{E}(\mathcal{T}, Q) \tag{2}$$

where $\mathcal{E}(\mathcal{T}, Q)$ is some error/objective function, identifying the divergence between the original time series ($\mathcal{T}$) with their quantized forms, $\mathcal{Q} = \langle \hat{Q}(T_1), \hat{Q}(T_2), \ldots, \hat{Q}(T_N) \rangle$.

We have chosen to focus on the quantization of univariate time series via a zero-memory quantizer, rather than the more complex cases of multivariate time series and/or quantizers with memory due to two reasons. First, the use of zero-memory quantizers are exceptionally common in practice. Second, and more importantly, the more complex cases can generally be viewed and implemented as extensions to the zero-memory univariate case, which therefore provides an ideal theoretical basis and the obvious starting point to underpin the development and evaluation of a novel quantizer. As further motivation for this decision, we point to the fact that extensions utilising *temporal dependencies* in time series commonly apply linear pre-processing followed by zero-memory scalar quantization [7, p.66] - and as such can also be directly be applied to this work.

A. Defining a "good" quantization: Time series comparisons

The generalized quantization problem is instantiated within an application domain by an objective function which quantifies the notion of a *good* quantization. Note, however, there are large classes of applications for which a single objective function can be justified. Specifically, here we consider the large class of applications which are based on time series comparisons. For this class *good* translates to minimizing the comparison error between any two time series, leading to the following, previously unconsidered, definition of $\mathcal{E}$:

$$\mathcal{E}(\mathcal{T}, Q) = \sum_{\forall T_a, T_b \in \mathcal{T}} |\delta(T_a, T_b) - \delta(\hat{Q}(T_a), \hat{Q}(T_b))| \tag{3}$$

where $\delta(T_a, T_b)$ is the distance between time series T_a and T_b and $\delta(\hat{Q}(T_a), \hat{Q}(T_b))$ is the distance between the two quantized time series $\hat{Q}(T_a), \hat{Q}(T_b)$. While any distance measure is possible, in this work we consider the commonly chosen Euclidean distance as the measure of distance between time series - and hence we intend to maintain, after quantization, as good as an approximate as possible of the original Euclidian distances between time series. Armed with this new objective function, the question therefore arises as to whether the current methods, which are optimal for their original goals (such as reconstruction), are also optimal for time series comparisons.

III. The suboptimality of existing quantizers for time series comparison

As previously noted, although a number of methods for quantizing time series have been proposed in the literature, none have addressed the end goal of maintaining the fidelity of time series comparisons. This is surprising given, first, the substantial literature on quantization for signal reconstruction, classification or maximization of human perception (see [7], [8], [9]–[11] and [12] respectively for examples) and, second, the huge amount of quantized time series data that is being regularly indexed and retrieved via computational comparisons.

For instance, the often cited SAX method for indexing time series makes no consideration of the end goal of time series comparison. This is despite its frequent use as part of k-nearest neighbour or range queries. Specifically, the quantizer embedded within SAX (and additionally within iSAX [13], iSAX 2.0 [5] and variants) utilises a restricted maximum entropy quantizer and does not consider data point values, but rather only their frequencies. As will be shown empirically later in section V, this leads to the approach to symbolization performing sub-optimally for time series comparisons, where evaluations occur based upon these values. This is not a specific criticism of SAX, for which quantization is only one component, rather it is a criticism of the lack of research into the effect of quantization for the broad application of time series comparison. Note that even in the recent work presented in [14] (which providing valuable insights through the comparison of different representation methods for time series data) there is no consideration of the effect (or alternatives) of the type of quantization performed within the representations.

We provide below a concise review of current quantization techniques. Under the sub-section *Example of failure*, we highlight the sub-optimality for comparison based applications of each of these techniques via a counter-example, giving an example where the method does not provide the best quantisation with respect to maintaining the distance between two time series. When applicable, we use an illustrative toy example $T_a = [10, 10, 4]$ and $T_b = [10, 0, 4]$ where $T_{ai}, T_{bi} \in [0, 4, 10]$

to highlight sub-optimality. Under optimal *comparison quantization* of these time-series into a binary alphabet, T_a and T_b become $[10, 10, 10]$ and $[10, 0, 10]$ respectively because such an encoding best preserves the Euclidean distance between the two original time series of 10 (minimizing equation 3). Once more, it is worth emphasizing that we are not proposing that there are inherent short comings with current techniques themselves - all provide optimal solutions for some application. Rather the issue is in their erroneous application to data mining tasks which rely on time series comparisons, where a quantizer specifically designed for that purpose should have been preferred.

A. Uniform quantization

Uniform quantization is by far the simplest form of time series quantization and is simply the partitioning of all potential values into m equal regions. No information regarding the distribution, values or end use are taken into account. The centre point of each region is then assigned to any points in the time series that fall into that region. The non-optimal nature of such an approach with respect to post-quantization time series comparisons is easily seen by numerous toy examples.

Example of failure: Under uniform quantization T_a, T_b are quantized to [7.5,7.5,2.5], [7.5,2.5,2.5] respectively resulting in a distance of 5. This is only half the distance of 10, captured by the optimal encoding of $[10, 10, 10]$ and $[10, 0, 10]$.

B. Minimal reconstruction error (MSE)

A mean squared error objective function minimizes the reconstruction error between the original and quantized time series. Specifically, for a given set of time series the quantization levels are selected so that the following error function $\mathcal{E}$ is minimized:

$$\mathcal{E}(\mathcal{T}, Q) = \sum_{T_a \in \mathcal{T}} \sum_{i=1}^{n} (T_{ai} - Q(T_{ai}))^2 \tag{4}$$

While intuitive, MSE does not take into account how often comparisons are made, nor adjust its quantization accordingly, as is shown in the example case:

Example of failure: Under MSE, T_a, T_b are quantized to $[10, 10, 3]$, $[10, 3, 3]$, resulting in a separation distance of 7. While better than the uniform quantizer, again the optimal solution, 10, is not achieved.

C. Maximum Output Entropy (MOE) quantization

MOE quantization aims to maximise the average mutual information between the input and the output. If Y is the number output levels in the resulting quantizer and m is the number of levels in the quantizer, then the average mutual information is maximized between the input and output of the quantizer when $P(Y_k) = 1/m$ [15]. Despite ensuring each symbol will convey an optimal amount of *shannon information*, an MOE quantization is not optimal when comparing post-quantized time series under distance measures that penalize changes in the amplitude. This is because the distance measure values large changes in magnitude far more than small changes. And since an MOE quantizer's only goal is only to maximise the amount of per symbol information, and not the importance of that information to a specific task, it can significantly underperform when quantizing with respect to maintaining the fidelity of a distance metric.

Example of failure: Here we use a slightly more involved illustrative example. Consider series the time series: [0,0,0,0,-1,-1,-1,-1,50,60,70,80] and [-1,-1,-1,-1,0,0,0,0,80,70,60,50]. Under MOE with 3 symbols these quantize to [0,0,0,0,-1,-1,-1,-1,65,65,65,65] and [-1,-1,-1,-1,0,0,0,0,65,65,65,65] respectively. Calculating the distance between the quantized series produces a results of 2.83, which is far below the actual Euclidean distance of 44.81 (whereas the best possible result obtainable using a three level quantizer is in fact, the far closer value of 42.43).

D. SAX, iSAX, iSAX 2.0 and variants:

Symbolic Aggregation Approximation is a time series representation that actually supports an arbitrary underlying quantizer [5, pg. 59] (as denoted by Q in equation 1). However, in their research SAX's authors have chosen to use a quantizer based on MOE, but with the added assumption of a normal distribution [1]. It is this quantizer variant used within SAX that we consider, henceforth denoted as *qSAX*. Note we only address the performance of this specific part of the SAX representation, and not the other aspects such as temporal quantization or the efficient indexing of the symbolic representation addressed in subsequent publications (e.g. extended SAX [4], iSAX [13] or iSAX 2.0 [5], which continue to use qSAX as the underlying quantizer). Importantly, all of the quantizers evaluated in this work could also be used in any of the overall SAX frameworks, allowing any performance improvements we report to also benefit these more involved approaches to working with time series data.

Example of failure: Consider the same series as detailed for MOE. Under qSAX with 3 symbols *both* of the subject time-series quantize to [-1,-1,-1,-1,-1,-1,-1,-1,65,65,65,65]. This surprising result is due to the assumption of a normal prior, and results in a reported distance of 0. This is clearly suboptimal when the actual Euclidean distance is, as before, 44.81.

E. Other quantizers

Other quantizers proposed in the literature seek to minimize objective functions specific to their individual problem spaces. For instance perceptual distance quantizers aim to maximise the ability to *discriminate* in the context of a binary decision where two hypotheses and associated conditional probabilities (giving the probability of the input assuming a pre-specified hypothesis is true) are known in advance [16]. Assuming additional, application specific knowledge, these approaches are not applicable to the general class of time-series comparison problems we consider. Other proposed quantizers have included perceptual distance quantizers, which seek to quantize such that as much perceptual information is retained [17], and the *Persist algorithm* [18] which aims to quantize such that the symbols are persistent temporally with a focus on the human interpretability of the states. While approaching their specific problems from a similar angle to ourselves, the end use, the problems addressed, and the subsequent developments of a custom quantizer, are very different to that presented here.

IV. Learning Improved quantizers for time series comparison

In section III we discussed the fact that current state-of-the-art solutions are not optimal with respect to the goal of comparing time series. That this is the case is of no real surprise since this prior work makes no claim to be optimal in this sense, nor do they even attempt to strive towards this type of optimality. However, these techniques have been used (perhaps deleteriously) as part of data mining processes due to their prevalence and availability. In theory, therefore, such approaches to learning quantizers should be avoided in favour of those directly attempting to find the optimal quantization with respect to minimizing time series comparison error. Unfortunately, enumerating all possible quantizers and selecting the one with the lowest comparison error is intractable since *simply checking a solution* has a runtime complexity quadratic in the number of time series multiplied by the time series length.

Addressing this issue we now present a novel alternative approach which we have named Independent Comparison Error quantization (ICE). Our ICE implementation is based upon simulated annealing, for which we provide an exceptionally cheap to compute error function based on equation 3. Specifically, our error function enables a solution to be checked in $O(m^2)$ time[3] and is independent of the number and length of the time series making it particularly applicable to big datasets. This tractability is gained through the introduction of two approximations/assumptions, coupled with a specific reformulation of the problem and an integrated caching strategy.

Recall from equation 2 that our goal in learning a quantizer is to minimize comparison error, $\operatorname{argmin}_{Q(\cdot)} \mathcal{E}(\mathcal{T}, Q)$. If we were trying to optimize with respect to a reconstruction error function, then we could have used one of a number of algorithms that provide a computationally tractable deterministic solution [19] (in practice, the stochastic Max-Lloyd algorithm [8] still predominates, due to its computational efficiency and in spite of the fact that it may return only a locally optimal solution). We do not possess a tractable deterministic solution to optimizing comparison error (i.e. equation 3) so by necessity (rather than choice) we use a global optimisation method (in similar fashion to [20] who minimized reconstruction error in the case of vector quantization). Specifically, our optimization approach is based upon a modification of the algorithm detailed in [21], which implements a basic simulated annealing algorithm in a parallel fashion on the GPU, thus enabling a far greater search space to be considered.

In implementing a simulated annealing approach, a state within the system is defined as a specific instance of a quantizer Q as defined in equation 1. The cost function is an error function ($\mathcal{E}(\mathcal{T}, Q)$) of the form previously discussed, with a function producing a randomised set of valid permutations of boundary and symbol values being used as the *next neighbour* function. In order to ensure that this approach is tractable, the cost of checking each potential solution must be very low due to the vast size of the space that must be iterated through. Therefore, checking equation 3 directly is simply not an option due to its quadratic complexity in the number of time series and their length, leading to computational intractability. As such, we present a solution to this problem via a re-formulation of the error function which is able to reduce the cost of checking a solution to being quadratic only in the number of symbols quantized to, m. Since m is typically small the quadratic nature is not a concern. Importantly, we are now *independent* of both the number and length of the time series.

Note that when the assumptions we present hold exactly, the learnt quantizer is guaranteed to be optimal. Although situations where they do hold perfectly are unlikely to occur within real world datasets, we show empirically in section V via real world data that even when these assumptions are moderately violated the learnt quantizer still performs extremely well.

A. A novel quantizer for time series comparisons

Consider using the error function from equation 3 as the cost function in the simulated annealing, instantiated with the standard Euclidean distance as previously motivated (letting n and N denote the length of an individual time series and the number of time series respectively):

$$\mathcal{E}(\mathcal{T}, Q)=\sum_{\forall T_a, T_b \in \mathcal{T}} \left| \sqrt{\sum_{i=1}^{n} [T_{ai}-T_{bi}]^2} - \sqrt{\sum_{i=1}^{n} [Q(T_{ai})-Q(T_{bi})]^2} \right|$$

Note that the terminology *error function* and *cost function* is interchangeable here since they have the same functional form. Specifically, just like $\mathcal{E}(\mathcal{T}, Q)$ the simulated annealing cost function has available the time series set and evaluates a fixed instantiation of a quantizer, Q, in this case corresponding to a state within the simulating annealing algorithm.

As previously discussed, using this equation directly as the cost function within the simulated annealing process via a brute force computation is intractable. In order to achieve tractability we reconsider our choice of the Euclidean distance as the distance function δ, substituting it with the Manhattan distance. The rational for utilising the Manhattan distance is purely pragmatic[4], allowing the subsequent reformulations presented in this work (when considering applications which are inextricably tied to L2 metrics this change can be considered a practical approximation). Having selected the Manhattan distance the error (cost) function then becomes:

$$\mathcal{E}(\mathcal{T}, Q) = \sum_{\forall T_a, T_b \in \mathcal{T}} \left| \sum_{i=1}^{n} |T_{ai} - T_{bi}| - \sum_{i=1}^{n} |Q(T_{ai}) - Q(T_{bi})| \right|$$

[3] recall that m is the number of output symbols produced by the quantizer, and is assumed to be relatively small.

[4] Although we note that the Manhattan distance is an equally good choice in end use tasks such as classification [22] and hence is worth optimising for in its own right.

Since the order of the elementwise comparisons considered is not of a concern within the distance measure each member of $\mathcal{T}$ can be modelled as a random variable, and assuming the time series are all of the same length, the above equation can be rewritten as:

$$\mathcal{E}(\mathcal{T},Q) = \sum_{\forall T_a,T_b\in\mathcal{T}} \Bigg| n \iint_{-\infty}^{\infty} P(T_a = x, T_b = y)|x-y|\;dx\;dy \\ -n \iint_{-\infty}^{\infty} P(T_a = x, T_b = y)\,|Q(x)-Q(y)|\;\;dx\;dy \Bigg|$$

which can be rearranged to:

$$n \sum_{\substack{\forall T_a,T_b \\ \in\mathcal{T}}} \Bigg| \iint_{-\infty}^{\infty} P(T_a{=}x, T_b{=}y)\,[|x-y| - |Q(x)-Q(y)|]\,dx\,dy \Bigg|$$

where $P(T_a = x, T_b = y)$ is the empirical probability that a particular value in T_a will be compared with a particular value in T_b when the time series are examined against each other. Unfortunately, even in this form we still need to calculate the empirical probabilities between each pair of time series, and are facing a time complexity quadratic in the total number of time series within our dataset. Therefore, we take the further step of bringing the outer modulus within the integration itself, as this allows us to make the logical interpretation of our overall error function as being the sum of the magnitude of the per time series element comparison errors. The validity of making this adjustment is sound for time series of low length (it is identical for time series of length one), but the adjustment becomes more tenuous as these lengths grow and the likelihood increases that sign changes within component parts of the integration will occur, producing unpredictable interaction effects. The assumption here, therefore, is that these sign changes do not occur. While this is highly unlikely to be true in general, our empirical results indicate that this formulation still provides a good approximation of the desired objective function in practice. At the same time we divide by the constant $n \times N^2$ (the manipulation of the equation via constants will not change the resultant optimisation). The result is our proposed error (cost) function:

$$ICE(\mathcal{T},Q) =$$

$$\frac{1}{N^2} \sum_{\substack{\forall T_a,T_b \\ \in\mathcal{T}}} \iint_{-\infty}^{\infty} P(T_a{=}x, T_b{=}y)\,||x-y| - |Q(x)-Q(y)||\;\;dx\;dy$$

which can be rearranged to:

$$\iint_{-\infty}^{\infty} \frac{1}{N^2} \sum_{\substack{\forall T_a,T_b \\ \in\mathcal{T}}} P(T_a{=}x, T_b{=}y)\,||x-y| - |Q(x)-Q(y)||\;\;dx\,dy$$

Defining $P(x,y)$ as the probability of a comparison between values x,y over all comparisons within pairwise time series comparisons in $\mathcal{T}$ we get:

$$ICE(\mathcal{T},Q) {=} \iint_{-\infty}^{\infty} P(x,y)\,||x-y| - |Q(x)-Q(y)||\,dx\,dy \quad (5)$$

It is this specific re-arrangement of the proposed function that we primarily refer to as Independent[5] Comparison Error (ICE). We now provide a tractable, $O(m^2)$, algorithm for computing equation 5 for use within the simulated annealing algorithm. Recall that the error (cost) function is evaluating a fixed quantizer (Q) which maps values, via boundary points (B_i) to a set of symbols with values q_i. By noting that equation 5 can (1) be re-interpreted as a double summation over the comparison of possible symbol mappings and (2) that within a single comparison between two symbol mappings that $|Q(x) - Q(y)| = q_{ij}$ is a constant we get:

$$ICE(\mathcal{T},Q) {=} \iint_{-\infty}^{\infty} P(x,y)\,||x-y| - |Q(x)-Q(y)||\;\;dx\;dy \\ {=} \sum_{i=1}^{m}\sum_{j=1}^{m} \int_{B_{i-1}}^{B_i} \int_{B_{j-1}}^{B_j} P(x,y)\,||x-y| - q_{ij}|\,dx\,dy \quad (6)$$

From these observations and reformulations the $O(m^2)$ complexity can be achieved by computing the double integral from equation 6 in constant time by first pre-computing $\phi(a,b,q_{ij})$, the joint cumulative functions for x and y for any q_{ij}:

$$\phi(a,b,q_{ij}) = \int_{-\infty}^{a} \int_{-\infty}^{b} P(x,y)\,||x-y| - q_{ij}|\;\;dx\;dy \quad (7)$$

In practice $\phi(a,b,q_{ij})$ is approximated and pre-computed via an arbitrarily fine uniform discretization of the continuous space. Note that for a large number of symbols a uniform discretization introduces very little error [23] and thus it is appropriate in this instance. A fixed double integral instance from equation 6 can then be evaluated in constant time via:

$$\begin{aligned} &\int_{B_{i-1}}^{B_i} \int_{B_{j-1}}^{B_j} P(x,y)\,||x-y| - q_{ij}|\,dx\;dy \\ &\quad = \phi(B_i, B_j, q_{ij}) + \phi(B_{i-1}, B_{j-1}, q_{ij}) \\ &\qquad - \phi(B_{i-1}, B_j, q_{ij}) - \phi(B_i, B_{j-1}, q_{ij}) \end{aligned} \quad (8)$$

This approach has cubic space complexity in the chosen uniform discretization, with one two-dimensional array required for each possible q_{ij} value. This complexity provides a practical limit on how arbitrarily fine-grained the uniform quantization can be. In this work we set this value at 900. This allows the function to be precomputed and stored within a 3GB GPU[6]. In section V empirical evaluations indicate that the choice of 900 is sufficiently high. Note that the amount of memory used here is solely based on this choice, and not the number or lengths of the time series within the data set.

The time complexity to construct this function is also of cubic complexity and additionally dependent on the cost of calculating $P(x,y)$ over the time series data. In both cases this is only done once as a pre-processing step outside of the simulated annealing. With the fine-grained approximation set at 900, the former presents no computational barrier,

[5] Referring to the alteration of the error function to be the sum of the independent, per time series element, comparison errors

[6] Using a 64bit floating point representation. The cumulative function is symmetric so only the upper triangle (including the diagonal) is stored.

contributing negligible time to process. The latter calculation of $P(x,y)$, however, is potentially dependent on the length and number of time series. While computing this within the GPU means a significant number of time series can be processed in reasonable time, this is not possible for truly big data sets. Instead, rather than exhaustively computing $P(x,y)$ one must therefore accurately approximate the joint probability distribution via random sampling of a sufficiently large number of time series[7] based on standard statistical techniques. Once the joint cumulative function has been pre-computed the error function from equation 5 can be computed in $O(m^2)$ enabling its use in simulated annealing algorithms since m, the number of resulting symbols, is small.

In summary, the cost function for use within the simulated annealing algorithm is:

$$\begin{aligned} cost_{ICE}(\phi, Q) = \textstyle\sum_{i=1}^{m}\sum_{j=1}^{m}[\quad & \phi(B_i, \quad B_j, \quad |q_i - q_j|) \\ + \quad & \phi(B_{i-1}, B_{j-1}, |q_i - q_j|) \\ - \quad & \phi(B_{i-1}, B_j, \quad |q_i - q_j|) \\ - \quad & \phi(B_i, \quad B_{j-1}, |q_i - q_j|)] \end{aligned}$$

V. Experimental Evaluation

The novel symbolization method presented within this work is evaluated against five of the most prevalent quantization methods within the literature, with evaluations performed on three datasets, covering both synthetic and real world data. The quantization methods evaluated (including their abbreviations which are used throughout) are:

UNI	- Uniform quantization
aMSE	- Mean Squared Error quantization (sim. annealing)
lMSE	- Mean Squared Error quantization (Lloyd-Max)
MOE	- Maximum output entropy quantization
qSAX	- MOE quantization assuming a normal distribution.
ICE	- Comparison Error minimization (proposed method)

The first five methods represent prominent approaches within the literature as discussed in section III. Of these lMSE and aMSE are both methods for symbolization based on reconstruction error. aMSE denotes the minimization of the MSE objective function via simulated annealing within the GPU based on the fine-grained discretization of the continuous space to 900 levels and utilises the same simulated annealing algorithm used in ICE. This enables a fair comparison with respect to the amount of computational resources used in the optimisation. lMSE denotes the direct optimisation of the reconstruction error in the continuous space based on the commonly used Lloyd-Max algorithm[8]. This serves to provide an indication as to the validity of the chosen value of 900 for the approximation of the continuous space in the GPU methods. While it is not expected that the simulated annealing and Lloyd-Max versions of the algorithm will be *exactly* the same, if the continuous approximation is fine-grained enough it is expected that the simulated annealing approach will perform as well as, if not better than, the Lloyd-Max algorithm in the continuous space.

In order to assess their performances in a practical application within the target domain of this work (the large class of applications based on time series comparisons), we examine the impact of each quantizer on the extremely common task of nearest-neighbour search based on the Euclidean distance. As such the use of the Manhattan distance in the ICE quantizer becomes an approximation. Note that a nearest neighbour search based on the Manhattan distance would be equally valid (since it obtains similar performance in real world tasks [22]) and would likely result in improved performance of the ICE quantizer. However, we chose to use the Euclidean distance in the first instance since it represents the standard baseline. Investigating the performance under other valid applications remains interesting future work.

For our tests the set of time-series data, $\mathcal{T}$, is partitioned into a test set $\mathcal{T}_A$, and a training set, $\mathcal{T}_B$. Given an input time series from $\mathcal{T}_A$, our binary evaluation function will return one if it's nearest neighbour in $\mathcal{T}_B$ is the same under both its symbolized and original form, returning zero if this is not the case. We iterate over each time-series in the test set to find the average rate of successful matches. Let N_A be the number of time series in $\mathcal{T}_A$ then formally:

$$\mathcal{E}_{NN} = \frac{1}{N_A}\sum_{A\in\mathcal{T}_A}\begin{cases} 1 & \text{if } \operatorname{argmin}_{B\in\mathcal{T}_B} L2(A,B) = \\ & \quad \operatorname{argmin}_{B\in\mathcal{T}_B} L2(Q(A),Q(B)) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $L2(\cdot,\cdot)$ is the standard $L2$ norm (Euclidean distance) and in this case $Q(\cdot)$ was learnt on the set $\mathcal{T}_B$ since this reflects the real world case where the data being queried is known.

The error function (equation 9) is used as part of a cross validation procedure in order to evaluate the expected error across varying test and training sets and to provide a statistical significance on the expected error. Specifically we use the procedure motivated and discussed in-depth in [24], correcting the variance to account for the data reuse inherent in cross-validation and perform statistical tests using the *corrected resampled t-test* [24, pg 251]. Since multiple methods are compared the p-values are corrected according to the Holm procedure[8]. For a given training and test set the evaluation measure is the proportion of nearest-neighbours that are correctly identified using the symbolized time series by considering the original time series as the ground truth. The mean of these proportions is the generalised error. Note that corrected t-tests are able to be used since the distribution of the sample proportions are approximately normal by the central limit theorem [25].

[7] The GPU implementation used can easily compute the joint probability distribution from tens of thousands of randomly sampled time series.

[8] The implementation from http://www.r-project.org/ was used.

A. Results: Smart Meter Electricity data

For real world data, data from The Commission for Energy Regulation (CER), Electricity Customer Behaviour Trial[9] was used. The data set consists of over 6435 time series of building energy usage sampled at 30 minute intervals. The average time series length is $24,552$ data points. The distribution of the combined temporal samples was typically log-normal. Time series lengths of 24 (12 hours), 48 (1 day) and 96 (2 days) were considered and symbolization to 8, 16 and 24 symbols was evaluated. The results are shown in Table I (a) - (c).

B. Results: 80 Million Tiny Images

As a second real world dataset we use a subset of the 80 Million Tiny Image dataset as detailed in [26][10]. Following the work of [5] in evaluating time series, we convert each image to a colour histogram with 256 bins. These histograms can be considered as time series with a length of 256 and the same techniques and evaluation applied. For this experiment a dataset of the first $5,000$ images was considered. The evaluation procedure as previously detailed was once again used and the results are shown in Table I (d).

C. Results: Synthetic data

Finally, we consider synthetic random walk time series. We produced test sets of size 5,000 for time series of lengths 24, 48 and 96 using the random walker code from http://www.cs.ucr.edu/~mueen/MK/. The data was evaluated via the previously discussed cross-validation procedure for symbolization to 8, 16 and 24 symbols. The results are shown in Table I (e) - (g).

VI. Discussion

Overall the ICE quantizer consistently provided the best performance out of all the quantizers tested. In the real world datasets, ICE produced the best results for all time series lengths (which were up to 256 elements). For all quantizations that employed more than eight symbols this was to a statistically significant level (to 95% confidence), with ICE showing an increase in performance of 9.58% on average compared to the next best performing method.

In general all quantizers followed the expected trend of monotonically increasing performance with respect to the number of symbols used for quantization and the length of time series tested. Note that this increase in performance is generally expected due to the fact that increasing the number of symbols and/or the time series length provides more detailed time series from which to discriminate. Importantly, the ICE quantizer also followed this trend, showing the validity of approximation of the comparison error used in practice.

While ICE provided the best performance on all experiments using real-world data, its performance was less marked when quantizing with 8 symbols (with MSE and qSAX still providing strong competition at this level). This is likely due

(a) Electricity Dataset: Time series of length 24

Num Syms:	8	16	24
ICE	**0.2295 (0.0160)**	**0.4464 (0.0154)**	**0.5269 (0.0241)**
aMSE	0.1211 (0.0155)	0.3005 (0.0181)	0.4166 (0.0248)
lMSE	**0.1905 (0.0279)**	0.2199 (0.0446)	0.2168 (0.0274)
qSAX	**0.1992 (0.0268)**	0.3504 (0.0208)	0.4531 (0.0246)
UNI	0.0081 (0.0034)	0.0236 (0.0048)	0.0483 (0.0092)
MOE	**0.1928 (0.0147)**	0.3100 (0.0199)	0.4021 (0.0164)

(b) Electricity Dataset: Time series of length 48

Num Syms:	8	16	24
ICE	**0.3343 (0.0168)**	**0.5863 (0.0154)**	**0.6664 (0.0189)**
aMSE	0.2505 (0.0181)	0.4887 (0.0180)	0.5720 (0.0184)
lMSE	**0.3259 (0.0311)**	0.3204 (0.0483)	0.3031 (0.0294)
qSAX	**0.2919 (0.0177)**	0.4724 (0.0186)	0.5818 (0.0198)
UNI	0.0164 (0.0057)	0.0542 (0.0087)	0.1115 (0.0136)
MOE	0.2080 (0.0163)	0.3564 (0.0222)	0.4737 (0.0209)

(c) Electricity Dataset: Time series of length 96

Num Syms:	8	16	24
ICE	**0.3894 (0.0214)**	**0.6229 (0.0221)**	**0.6995 (0.0218)**
aMSE	0.2630 (0.0163)	0.5235 (0.0103)	**0.5841 (0.1208)**
lMSE	**0.3624 (0.0133)**	0.3442 (0.0411)	0.3240 (0.0420)
qSAX	**0.3260 (0.0289)**	0.4993 (0.0303)	0.5976 (0.0214)
UNI	0.0158 (0.0064)	0.0671 (0.0088)	0.1314 (0.0068)
MOE	0.2032 (0.0161)	0.3554 (0.0160)	0.4702 (0.0169)

(d) Image Dataset: Time series of length 256

Num Syms:	8	16	24
ICE	**0.4435 (0.0222)**	**0.6569 (0.0192)**	**0.7468 (0.0225)**
aMSE	0.2660 (0.0180)	0.5643 (0.0199)	0.6716 (0.0160)
lMSE	**0.4195 (0.0216)**	0.5255 (0.0507)	0.3380 (0.0313)
qSAX	**0.3736 (0.0300)**	0.5348 (0.0333)	0.6020 (0.0279)
UNI	0.0175 (0.0057)	0.0404 (0.0103)	0.0591 (0.0079)
MOE	0.2499 (0.0438)	0.3571 (0.0256)	0.4461 (0.0325)

(e) Random Walk Dataset: Time series of length 24

Num Syms:	8	16	24
ICE	**0.2233 (0.0192)**	**0.4796 (0.0197)**	**0.6180 (0.0270)**
aMSE	**0.2035 (0.0109)**	**0.4481 (0.0199)**	**0.5948 (0.0220)**
lMSE	0.0248 (0.0361)	0.1200 (0.0608)	**0.3053 (0.1129)**
qSAX	**0.2499 (0.0186)**	**0.4795 (0.0220)**	**0.5809 (0.0128)**
UNI	0.0769 (0.0124)	0.2168 (0.0154)	0.3555 (0.0208)
MOE	**0.2207 (0.0214)**	**0.4419 (0.0263)**	**0.5659 (0.0185)**

(f) Random Walk Dataset: Time series of length 48

Num Syms:	8	16	24
ICE	**0.3024 (0.0230)**	**0.5723 (0.0196)**	**0.6945 (0.0189)**
aMSE	**0.2780 (0.0177)**	**0.5508 (0.0213)**	**0.6871 (0.0142)**
lMSE	0.0179 (0.0189)	**0.2173 (0.1214)**	**0.3715 (0.1270)**
qSAX	**0.3184 (0.0228)**	**0.5429 (0.0249)**	**0.6555 (0.0220)**
UNI	0.0885 (0.0140)	0.2936 (0.0190)	0.4721 (0.0183)
MOE	**0.2939 (0.0288)**	**0.5209 (0.0244)**	**0.6515 (0.0195)**

(g) Random Walk Dataset: Time series of length 96

Num Syms:	8	16	24
ICE	**0.3761 (0.0174)**	**0.6652 (0.0172)**	**0.7720 (0.0161)**
aMSE	**0.3399 (0.0198)**	**0.6444 (0.0259)**	**0.7612 (0.0165)**
lMSE	0.0225 (0.0204)	0.1172 (0.0814)	0.3616 (0.1271)
qSAX	**0.3688 (0.0253)**	0.5911 (0.0162)	0.6963 (0.0177)
UNI	0.0929 (0.0110)	0.3123 (0.0124)	0.5347 (0.0176)
MOE	**0.3449 (0.0178)**	0.5787 (0.0196)	0.6936 (0.0210)

TABLE I

Experimental Results: Mean (stdev) of the proportion of nearest neighbours correctly found. Bold indicates the best performing method(s) with multiple highlighted if they are not separated by a statistically significant difference from the best ($p < 0.05$) performing method.

[9] Avaliable from the Irish Social Science Data Archive: http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[10] Available from http://horatio.cs.nyu.edu/mit/tiny/data/index.html

to the fact that, for such small symbol sets, all comparisons begin to occur with same frequency and hence the advantages of ICE become less pronounced.

For our synthetic random-walk dataset ICE again generally provided the best performance - but this time the results could not be confirmed to a statistically significant level, even for larger symbol sets. Note that in general the synthetic data provided an easier task, with almost all methods performing better than for the real world energy data when quantizing to the same number of symbols. A potential explanation is that the nature of the random walk means that the time series are more spread out in the space[11], and hence it is easier to discriminate between time series in general. This results in less refined symbolizations (with respect to comparisons) still being able to correctly identify the nearest neighbour and increases their performance closing the gap on the ICE quantizer as shown in the results. Finally, note that in this dataset the symbol distributions for the time series are Gaussian, and this helps promote the effectiveness of qSax.

It is worth also noting that aMSE (which used the same core simulated annealing algorithm as ICE, varying only the cost function) generally matched, and often performed better than, lMSE and this offers evidence that our chosen value of 900 for the approximation of the continuous space in the GPU methods was of an appropriate granularity. A final observation is with regard to the poor performance of the MOE and Uniform quantizers. Their consistent losses, and general inability to identify the majority of nearest neighbours correctly, serve to highlight that the consideration of quantization approach can be vital to the effectiveness of data mining algorithms.

VII. Conclusions

Many quantization techniques which are thought of as optimal, are in fact only optimal within the context of a specific problem domain such as signal reconstruction. In this work we have shown that in the extremely prevalent case where quantized input is used as the basis for time-series comparisons, standard approaches can lead to potential performance loss. To address this issue we have presented a novel quantizer (ICE) based on minimising the comparison error, with adjustments made to provide a computationally tractable implementation effective to large data. Our empirical results based upon three different datasets (and using various time-series lengths and symbol-set cardinalities) have provided initial evidence for the superiority of ICE for comparison-based data mining tasks. Even though one might expect the assumptions that underpins ICE to be violated by many real-world datasets, our results have indicated that even when such violations occur our quantizer can still provide superior performance to current state-of-the-art quantization approaches.

Acknowledgements

This work was funded by the RCUK Horizon Digital Economy Research Hub grant, EP/G065802/1.

[11] The variance in the pairwise distances between the original time series in the random walk dataset was over double that of the electricity dataset.

References

[1] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD)*. ACM, 2003, pp. 2–11.

[2] C. Daw, C. Finney, and E. Tracy, "A review of symbolic analysis of experimental data," *Review of Scientific Instruments*, vol. 74, no. 2, pp. 915–930, 2003.

[3] B. Hu, T. Rakthanmanon, Y. Hao, S. Evans, S. Lonardi, and E. Keogh, "Discovering the intrinsic cardinality and dimensionality of time series using MDL," in *Proc. Int'l Conf. Data Mining*, 2011, pp. 1086–1091.

[4] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation," in *Proc. Int'l Conf. Data Mining Workshops*, 2006.

[5] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh, "iSAX 2.0: Indexing and mining one billion time series," in *Proc. Int'l Conf. Data Mining (ICDM)*. IEEE, 2010, pp. 58–67.

[6] J. Goulding, G. Smith, D. Barrack, S. Preston, and K. Hopcraft, "Neo-demographics: Distributions in the digital shadow," in *Proc. Digital Economy All Hands Conference*, 2012.

[7] I. Bocharova, *Compression for Multimedia*. Cambridge University Press, 2009.

[8] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Information Theory*, vol. 6, no. 1, pp. 7–12, March.

[9] S. Kassam, "Optimum quantization for signal detection," *IEEE Trans. Communications*, vol. 25, no. 5, pp. 479 – 484, May 1977.

[10] R. Kohavi and M. Sahami, "Error-based and entropy-based discretization of continuous features," in *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 114–119.

[11] R. Gupta and A. Hero, "High-rate vector quantization for detection," *IEEE Trans. Information Theory*, vol. 49, no. 8, pp. 1951 – 1969, 2003.

[12] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1385–1422, 1993.

[13] J. Shieh and E. Keogh, "iSAX: disk-aware mining and indexing of massive time series datasets," *Data Mining and Knowledge Discovery*, vol. 19, no. 1, pp. 24–57, 2009.

[14] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[15] D. Messerschmitt, "Quantizing for maximum output entropy," *IEEE Trans. Information Theory*, vol. 17, no. 5, p. 612, Sep 1971.

[16] H. Poor and J. Thomas, "Applications of ali-silvey distance measures in the design of generalized quantizers," *IEEE Trans. Communications*, vol. 25, no. 9, pp. 893–900, 1977.

[17] C. Mota and J. Gomes, "Optimal image quantization, perception and the median cut algorithm," *Anais da Academia Brasileira de Ciłncias*, vol. 73, pp. 303 – 317, Sept. 2001.

[18] F. Mörchen and A. Ultsch, "Optimizing time series discretization for knowledge discovery," in *Proc. Int'l Conf. Knowledge discovery in data mining (KDD)*. ACM, 2005, pp. 660–665.

[19] X. Wu and K. Zhang, "Quantizer monotonicities and globally optimal scalar quantizer design," *IEEE Trans. Information Theory*, vol. 39, no. 3, pp. 1049 –1053, May 1993.

[20] H.-C. Huang, J.-S. Pan, Z.-M. Lu, S.-H. Sun, and H.-M. Hang, "Vector quantization based on genetic simulated annealing," *Signal Processing*, vol. 81, no. 7, pp. 1513 – 1523, 2001.

[21] A. Ferreiro, J. Rodrıguez, J. Salas, and C. Cendón, "Cusimann: An optimized simulated annealing software for GPUs," 2012. [Online]. Available: https://cusimann.googlecode.com/files/CUSIMANN.pdf

[22] D. Dohare and V. Devi, "Combination of similarity measures for time series classification using genetic algorithms," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, pp. 401–408.

[23] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Information Theory*, vol. 18, no. 6, pp. 759–765, 1972.

[24] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, pp. 239–281, 2003.

[25] R. Lomax, *An introduction to statistical concepts*. Routledge, 2007.

[26] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.