ABSTRACT
Among the many dimensionality reduction methods for time-series data, Symbolic Aggregate approXimation (SAX) is perhaps the most popular due to its simplicity and uniqueness. With SAX, time-series data can be represented as string sequences which enables the utilization of methods found in text mining and bioinformatics to enhance data mining tasks. We propose an application of L-tuples to improve clustering SAX-represented time-series. Using the L-tuple frequency distributions of sequences, we compute dissimilarity based on maximum Kullback-Leibler divergence. We compare our new approach and dissimilarity measure with existing SAX measures and show that our dissimilarity measure with L-tuples is able to enhance the quality of clustering of time-series.

1. INTRODUCTION
Time-series data can come from any domain in which temporal measurements are recorded, such as meteorology, healthcare, finance and environmental science. As such, the data mining literature on time-series is very rich. One particular topic of interest is clustering time-series data because it is used as a subroutine in more complex data mining algorithms, such as anomaly detection indexing, and classification [1]. Specifically, we are interested in clustering whole time-series which entails grouping similar time-series objects based on how similar they are with one another without any prior knowledge of each group. Whole time-series clustering requires us to assume a proper representation, defining a similarity/dissimilarity measure and using an appropriate clustering algorithm. Each of these steps in of itself is a research area ([2] provides a very comprehensive review) so we will specifically focus on developing a novel distance measure that improves the clustering result for SAX-represented sequences. In this paper, we will propose a new dissimilarity measure for SAX [3] representation of time-series, one of the most popular time-series representation.

Our dissimilarity measure is based on an information-theoretic approach commonly used in the bioinformatics for the multiple sequence alignment problem using word-based distribution vectors. We will compare our dissimilarity measure with other SAX-based measures and show that our clustering results is better than the pre-existing results. To the best of our knowledge, this paper is the first application of the bioinformatics sequence alignment problem to time-series data mining, specifically to the task of clustering.

The paper is organized as follows. In Section 2, we will discuss the challenges of using bioinformatics methodology and developing a dissimilarity measure to for SAX sequences and also discuss the contribution of our paper. In Section 3, we will discuss previous works which have been applied to SAX sequences and alignment-free dissimilarity measures commonly used in the bioinformatics literature. In Section 4, we will review the mathematical preliminaries relevant to the paper. We will then introduce our L-tuple approach and dissimilarity measure in Section 5. Our experimentation and its results relative to other approaches for SAX sequences is discussed in Section 6. Lastly, we will conclude and provide possible future work in Section 7.

2. PROBLEM STATEMENT
In the original SAX paper [3], the authors concluded that using SAX as a representation scheme allows for the utilization of methods found in other research areas, such as text mining and bioinformatics, which might enhance the data mining results. Since then, there have been only a few works which utilized methods and measures found in other disciplines. To the best of our knowledge, there are no works in which applies bioinformatic methods to the clustering of SAX-represented time-series. This is most likely due to two challenges: (1) methods are often base-specific and (2) rely on domain- or problem-specific information.

First, many dissimilarity measures found in the bioinformatics literature use L-tuples or some variation of it. However, these approaches are often base-specific. For example, DNA is base-4 so if we set $r = 4$, we can use DNA sequence alignment methods. Amino acids is base-20 so if we set $r = 20$, we can utilize methods found in protein sequence alignment methods. Because the size of the collections are dependent on the alphabet size, it is not clear on how we can apply measures found in bioinformatics to the symbolic time-series since the size of the alphabet is
user-defined. Many of the methods and algorithms found in the bioinformatics literature are inapplicable in the general context due to this reason.

The second challenge is that many bioinformatics methods rely on domain- or problem-specific information to enhance the data mining task. For example, the authors in [1] used the fact that nucleotide bases are divided into three base-pair categories to formulate a feature vector space to improve clustering. The work of [2] exploits information contained in base triplets to extract information from sequences for comparison. As a time-series may come from a variety of different sources, in general, we cannot use methods which utilize specific information structure about the underlying data.

In this paper, we will propose a new dissimilarity measure based on alignment-free sequencing using $L$-tuples and entropy. We will show that our dissimilarity measure improves the SAX clustering result found in previous works.

3. RELATED WORK

3.1 Previous dissimilarity measures for SAX

Let us discuss several dissimilarity measures that have been applied to the SAX representation. A commonly used dissimilarity measure in text mining is the edit distance (ED) which assigns a cost of insertion, deletion or substitution for each symbol. The main drawback of the edit distance is that is requires sequence alignment which is computationally inefficient given a large database of strings to cluster. Another problem with using the edit distance for sequence clustering is assume equal penalty in operations depending on the symbols frequency. The ordering of the SAX symbols is important as it is based on alignment-free sequencing using $L$-tuples and entropy. Let us discuss several dissimilarity measures that have been applied to the SAX representation. A commonly used dissimilarity measure in text mining is the edit distance (ED) which assigns a cost of insertion, deletion or substitution for each symbol. The main drawback of the edit distance is that it requires sequence alignment which is computationally inefficient given a large database of strings to cluster. Another problem with using the edit distance for sequence clustering is that it assumes equal penalty in operations depending on the symbols frequency. The ordering of the SAX symbols is important as it is based on alignment-free sequencing using $L$-tuples and entropy.

An attempt at this was provided in the work of [3] where a penalty function of different weights for the sequence operations depending on the symbols frequency.

$$EED(\hat{X}^{(1)}, \hat{X}^{(2)}) \triangleq ED(\hat{X}^{(1)}, \hat{X}^{(2)}) + \lambda \left[ |\hat{X}^{(1)}| + |\hat{X}^{(2)}| - 2 \sum_{i} \min(f_i^{\hat{X}^{(1)}}, f_i^{\hat{X}^{(2)}}) \right]$$ (1)

where $f_i^{\hat{X}^{(i)}}$ is the frequency of symbol $i$ in sequence $\hat{X}^{(i)}$, $|\cdot|$ is the length of the sequence and $\lambda \geq 0$ is a frequency factor. The problem with EED is that the parameter $\lambda$ is unintuitive. And although it has been found to improve on the edit distance, albeit only slightly, it still suffers from scalability if we have a large alphabet.

The distance measure originally proposed for SAX in [3] is given by Equation (2)

$$MINDIST(\hat{X}^{(1)}, \hat{X}^{(2)}) = \sqrt{\frac{n}{N} \sum_{i=1}^{N} \left( \text{dist}(\hat{x}_i^{(1)}, \hat{x}_i^{(2)}) \right)^2}$$ (3)

where $\text{dist}(\cdot)$ is a mapping function corresponding to a lookup table in which cells are calculated by

$$\text{cell}_{ij} = \begin{cases} 0 & |i-j| \leq 1 \\ \beta_{\max(i,j)} - \beta_{\min(i,j)} & \text{otherwise} \end{cases}$$ (4)

It was shown that SAX with the MINDIST measure provides a guaranteed lower bounding of the Euclidean distance. The drawbacks with MINDIST is that it depends on sequence alignment in constructing the lookup table. Although MINDIST is appropriate and was developed for indexing and classification problem, this distance measure is not accurate enough to for clustering as the distance between two symbols in neighboring region is considered zero and the maximal and minimal points of the time-series are ignored. In [4], a measure called APXDIST was proposed to consider the global max/min of time-series and non-zero distance between neighbor symbols for SAX sequences. Their proposed dissimilarity measure, APXDIST, considers the distance between SAX sequences to be the distance between indicators of breakpoint regions. An indicator indicates the closest closeness of PAA coefficients (in the region) to the indicator in that region. In defining the indicator, the arithmetic mean of each area (minimum and maximum) is defined as the indicator and the best estimate of the region:

$$Ind_i = \frac{\beta_{i-1} + \beta_i}{2}$$ (5)

where $\beta_0$ is the global minimum and $\beta_n$ is the global maximum. The APXDIST between a pair of symbolized time-series is defined as

$$APXDIST(\hat{X}_1, \hat{X}_2) = \sqrt{\frac{n}{w} \sum_{i=1}^{w} \text{dist}(Ind_{12}, Ind_{21})^2}$$ (6)

where

$$\text{dist}(Ind_i, Ind_j) = \begin{cases} 0, & i = j \\ \frac{\beta_{i+1} + \beta_{i}}{2} - (\beta_{i} + \beta_{i-1}), & \text{otherwise} \end{cases}$$ (7)

The integration of APXDIST with $k$-modes was found to have more accurate and higher quality results for SAX sequence clustering.

3.2 Alignment-free Measures

In the context of sequence alignment, alignment-free methods are often preferred to alignment-based methods because, among many other benefits, alignment-free methods do not assume contiguity of sequence data, are less sensitive to sequence variation, and algorithms are more computationally efficient. The most commonly used alignment-free method for sequence dissimilarity is the word-based feature representation called $L$-tuples which is analogous to n-grams in computational linguistic [8, 9, 10].

The framework of $L$-tuple methods is to segment a DNA sequence via a sliding window of length $L$. This essentially represents the sequence as a feature vector of combinatorial subsequences (words). For example, for DNA sequences the feature vector is of size $4^L$ and can contain combinations of 1-, 2-, 3- or 4-nucleotide combinations. Then, using the feature vector, the distance between DNA sequences can be computed based on the words’ frequency [11, 12]. However, frequency alone does not completely describe the entirety of information of a DNA sequence and so many
modified $L$-tuple algorithms have been proposed. The authors in [13] considered the mean and variance of each word’s position distribution to compute the distance between sequences. The consideration of more information comes with the trade-off of computational efficiency as the feature vector became three times as large as $4^L$. Reinert, et al. [14] proposed a variant of the $D_2$ statistics for $L$-tuples that compares sequences based on differences between words probability distribution.

Information theoretic approach to $L$-tuple have also been applied in practice and theory [15]. One of the earliest application is found in [16] who uses $L$-tuple frequencies to estimate a sequence’s divergence. Sadovsky [17] utilized relative entropy to compare sequences based on their frequency dictionaries. More recently, the DMk measure, proposed in [18] considers the entropy of $L$-tuple internucleotide distances as a measure of dissimilarity of DNA. DMk considers all L-mer subsequences’ positional distribution which amounts to $4^L$ subsequences. The work of [4] improves on DMk by considering nucleotide pairings to reduce the subsequences total to $2^L$.

It has been found that long $L$-tuple lengths ($L \geq 30$) outperforms short $L$-tuple models and better separate biological sequences with high dissimilarity [19]. As such, given an arbitrary alphabet of size $r$, it is very computationally expensive to compute intersymbol distances using DMk. Let us propose another $L$-tuple dissimilarity measure based on that considers order and location of symbols.

4. PRELIMINARIES

In this section, we will discuss some background information relevant to our proposed research. We will review the SAX representation of time-series data and the $k$-tuple representation widely used in the bioinformatics literature for the sequence alignment problem.

A time-series is a sequence of observations obtained successively from a continuous time interval. For example, a company’s stock price is continuous recorded over the period of the time the exchange it is listed on is open. We denote a time-series $X$ of length $n$ to be $X = \{x_1, ..., x_n\}$ where each $x_i$ represents a data point.

Given a collection of time-series, $C = \{X^{(1)}, X^{(2)}, ..., X^{(k)}\}$, we are interested in forming groups which consists of similar time-series. For example, let $C$ be a collection of time-series of companies’ stock prices. We may be interested in clustering companies based on trends of their stock prices. Clustering will enable us to identify companies from the same industry, companies who share complimentary products or the maximum number of symbols which we do not know yet.

4.1 SAX Representation

Time-series are inherently high-dimensional and so there are numerous representation techniques available as a means for dimensionality reduction (we refer to [2] for a comprehensive list). One of the most popular representation method is SAX which allows us to represent time-series as a concatenation of symbols. The authors in [3] showed that SAX is very efficient (its complexity is $O(n)$) and, with an appropriate distance measure, can provide a lower bound to the Euclidean distance. Let us briefly review the SAX representation.

SAX consists of two-steps: (1) piecewise aggregate approximation (PAA) and (2) discretization of the PAA feature vectors into symbols. The idea behind PAA is to segment the time series into equal parts and store the average value for each part. The average values is stored in a vector which becomes the data-reduced representation. Formally, PAA approximates $X$ into a vector $\bar{x} = (\bar{x}_1, ..., \bar{x}_M)$ of length $M$, where $\bar{x}_i$ is calculated as follows:

$$\bar{x}_i = \frac{1}{n} \sum_{j=\frac{iM}{n}}^{\frac{(i+1)M}{n}} x_j$$

(8)

This vector is Z-normalized and we assume the transformed data follows a standard Gaussian distribution. This standard Gaussian distribution is divided into $k + 1$ equiprobable sections. The sections are divided by a collection of values called breakpoints. Formally, breakpoints is a sorted list of numbers $B = \{\beta_1, ..., \beta_{k+1}\}$ such that the area under a $N(0, 1)$ Gaussian density from $\beta_i$ to $\beta_{i+1} = \frac{1}{n}$.

With this notation, the parameter $k$ specifies the number of symbols. The maximum number of symbols for SAX is 20. A symbol for each section based on where the PAA points lies relative to the equiprobable sections is assigned. The output of SAX is an ordered sequence of symbols which we will notate as $\bar{X}$.

Figure [11] illustrates the SAX representation of the time-series. The red line is the PAA approximation of the time-series and the colored horizontal lines are the breakpoints.

4.2 Shannon Entropy

In information theory, entropy is a measure of uncertainty of information for a probabilistic experiment. Given a discrete random variable $X$ with values $\{x_1, x_2, ..., x_M\}$ with corresponding probabilities $\{p_1, p_2, ..., p_M\}$, the Shannon entropy of $X$ is given by

$$H(X) = E[-\ln(p(X))] = -\sum_{i=1}^{M} p_i \log(p_i)$$

(9)

Equation (9) tells us the minimum number of binary questions necessary to determine the output of one observation of $X$, on average. Entropy is maximized if $p_1 = p_2 = ... = p_M = \frac{1}{M}$ which implies we are maximally uncertain about the outcome of a random sample.

For two distributions, $X_1$ and $X_2$, the distance between them is refered to as the Kullback-Leibler (KL) divergence (or relative entropy) is defined by the following:

$$H(X_1||X_2) = \sum_i p_i \log \left( \frac{p_i}{q_i} \right),$$

(10)

where $q_i$ is the probabilities for discrete random variable $X_2$. Relative entropy have long been used as a distance measure for text dissimilarity where $X_1$ may represent the true distribution of data and $X_2$ represents the theoretical distribution. From Equation (10) we can see that

$$H(X_1||X_2) = \sum_i p_i \log \left( \frac{p_i}{q_i} \right)$$

(11)

$$= -\sum_i p_i \ln(q_i) + \sum_i p_i \ln(p_i)$$

(12)

$$= H(X_1) - H(X_1)$$

(13)

where $H(X_1, X_2)$ is referred to the cross entropy which measures how close the distributions are to one another (i.e., if
$X_1 = X_2$, then the cross entropy is simply the entropy of $X_1$. The closer the cross entropy is to the entropy, the better $X_2$ is an approximation of $X_1$.

5. METHODOLOGY

5.1 L-Tuples

A common representation in bioinformatics is called L-tuple or (L-gram). Given a sequence of symbols, $X$ of size $N$, from an alphabet $A$ of length $k$, a L-tuple is a segment of $L$ symbols with $L \leq N$. Let $W^L_X = \{w_{1}, ..., w_{J}\}$ be a collection of all possible L-tuples from $X$. The size of $W_L$ is given by $J = r^L$.

We can count the occurrences by taking a sliding window of size $k$ through the sequence from position 1 to $N - L + 1$. Let us denote the collection of counts for sequence $X$ to be $c^L_X = \{c^L_{1}, c^L_{2}, ..., c^L_{J}\}$. Using the count, we can compute the L-tuple frequencies by

$$f^L_X = \frac{c^L_X}{N - L + 1}.$$  \hspace{1cm} (14)

Let us use an example to better illustrate these definitions. Suppose we have a DNA sequence $X = ATTAACAATG$ of size $N = 10$. The alphabet is the nucleotide base $A = \{A, T, C, G\}$ so $r = 4$. Nucleotide bases can be paired based on their chemical properties (e.g., \{A, G\} is the purine group, etc.) so let us consider $L = 2$. We have the sliding window to be from 1 to $N - L + 1 = 10 - 2 + 1 = 9$. Then, we have the following:

$$W_2 = \{AT, AA, AC, GA...\}$$  \hspace{1cm} (15)

$$c^L_{X} = \{2, 2, 1, 0, ...\}$$  \hspace{1cm} (16)

$$f^L_{X} = \{\frac{2}{9}, \frac{2}{9}, \frac{1}{9}, 0, ...\}$$  \hspace{1cm} (17)

Note that the $W_2, c^L_{X}$ and $f^L_{X}$ has $J = 4^2 = 16$ elements.

Essentially, we are moving a window of size $L$ through the sequence $X$ and counting for the tuples from the alphabet set. Let us now discuss how to use these L-tuple frequency counts to compare sequences.

5.2 KL-Divergence of L-tuples

The Shannon entropy of an L-tuple reflects the degree of importance of position and order in a sequence. It gives us the degree of variability of L-tuples in a sequence which can be used to compare between different sequences. In [20], the KL-divergence between sequences $X_1$ and $X_2$ is given by

$$d^{KL}_{L}(f^L_{X_1} || f^L_{X_2}) = \sum_{i=1}^{K} f^L_{X_1 i} \log \left( \frac{f^L_{X_1 i}}{f^L_{X_2 i}} \right).$$  \hspace{1cm} (18)

There are two issues with directly applying this distance to the SAX sequences. First, as noted by the authors, if $f^L_{X} = 0$, then the distance will be infinite. As the SAX alphabet size could be arbitrary large, we would expect a majority of the frequencies to be 0. To resolve this issue, we can adapt the optimal scaling of the frequencies from [21]:

$$d^{KL}_{L}(f^L_{X_1} || f^L_{X_2}) = \frac{1}{(N-L+1) + 0.5} \sum_{i=1}^{K} \left( \frac{\hat{c}^L_{X_1 i} + \epsilon_L}{\hat{c}^L_{X_2 i} + \epsilon_L} \right) \log \left( \frac{\hat{c}^L_{X_1 i} + \epsilon_L}{\hat{c}^L_{X_2 i} + \epsilon_L} \right)$$

where $\epsilon_L = 0.5 \cdot r^{-L}$.

The second issue is that Equation [18] is non-negative and asymmetric. Because of asymmetry, $X_1$ is considered to be the “true” or referenced sequence and so KL divergence is not a distance metric. One intuitive approach to symmetrize KL-divergence is compute the average of $d^{KL}_{L}(f^L_{X_1} || f^L_{X_2})$ and $d^{KL}_{L}(f^L_{X_2} || f^L_{X_1})$ as in [21]. However, this does not give a precise measure as it might discard lower and higher frequency values. Since our SAX symbols cor-
respects to location in the breakpoints, we cannot assume 
an average else we suffer from loss of information. A more 
robust measure is to consider the maximum KL-divergence 
(KLMAX):

\[
KLMAXDIST(\mathcal{L}_1||\mathcal{L}_2) = \max\{d_{KL}(\mathcal{L}_1||\mathcal{L}_2), d_{KL}(\mathcal{L}_2||\mathcal{L}_1)\}
\]

which was recently applied in [22] to the task of detection 
changes between data stream densities. We will use this 
measure to compare the SAX sequence L-tuples.

6. EXPERIMENTATION

Let us compare the clustering results of MINDIST, ED, 
APXDIST and L-tuple with KLMAX. We will not consider 
EED as it gives only slightly better results than ED. We 
refer the reader to the original paper for EED [6] for its 
comparison with ED and MINDIST. As with the previous 
works on SAX, we will consider two clustering algorithms, 
Hierarchical clustering and partition clustering. Hierarchical 
clustering is slow (quadratic in time) and is subjective as 
we judge which measure create the most natural groupings. 
Despite its bad scalability, if we know the data labels in 
advance, we can make objective statements about the 
clustering results and so hierarchical is useful in such regards. 
For partition clustering, we will use k-medoids algorithm.

6.1 Data

We examine two datasets: (1) Organisation for Economic 
Co-operation and Development (OECD) interest rates [23], 
and (2) UCR time-series archive [24].

The OECD dataset contains monthly 10-year bond interest 
rates for 17 different countries and the European Union 
from January 1995 to 2012. The choice of this dataset is for 
the hierarchical clustering as we can easily see the groupings 
between companies using the different measures.

The UCR time-series archive is a database consisting a large collection of synthetic time-series of different sizes, shapes and densities. The time-series also contains noises and drifts. Due to the scope of this course, we will not test over the entire database. We select four datasets from the archives labeled “TRAIN” and using the columns as se-
quences), each with different sizes and noise level. Table I 
summarizes number of observations and sequences from 
each of the datasets selected. We aim to examine whether 
higher number of observations yield more accurate clustering 
results. Also, how will the accuracy be affected by the 
number of sequences.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Observations</th>
<th>Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>50words</td>
<td>450</td>
<td>270</td>
</tr>
<tr>
<td>Adiac</td>
<td>390</td>
<td>176</td>
</tr>
<tr>
<td>Car</td>
<td>60</td>
<td>577</td>
</tr>
<tr>
<td>Control</td>
<td>300</td>
<td>60</td>
</tr>
<tr>
<td>Wine</td>
<td>235</td>
<td>57</td>
</tr>
</tbody>
</table>

6.2 Analysis using Hierarchical Clustering

In Figure 6.2, we plot the time-series sequences to formu-
late a general understanding of the data. At first, it seems 
as if several time-series to a similar trends such as EMU,

France, Finland, Spain, Netherlands, and Italy. However, 
starting from year 2010, these time-series have very differ-
te tails. As such, we would expect these time-series to be 
in the same branch but not necessarily be in the same leaf 
in a dendrogram or phylogeny tree. There are also a few 
time-series such as USA and Japan which are highly volatile 
and exhibit no similar trends to the other time-series. We 
would expect these two countries to be in their own clusters.

We use complete hierarchical clustering with these four 
measures. The clustering results are shown is phylogeny 
trees in Figure 6.2 We use an alphabet size of r = 12, 
w = 26 PAA frames (27 breakpoints) and L = 2 for KL-
MAXDIST. We see that MINDIST has France, Germany, 
Netherlands and Denmark is considered the same as well as 
Finland and Australia. This is not very accurate upon close 
inspection of the raw time-series. The other three measures 
were better able to differentiate the subtle trends and shifts 
from the SAX sequences. Another interesting observation 
is that EDITDIST, MINDIST and APXDIST all consider 
UK and USA to be closer than KLMAXDIST. From the 
raw time-series it looks like both countries follow the same 
trends but the USA’s interest rates is more volatile which 
indicates higher variability for the SAX symbols.

This initial analysis tells us that KLMAXDIST is more 
sensitive to highly variable SAX sequences than the other 
measures. We will need to consider a more rigorous evalu-
ation in order to determine whether is this beneficial or 
not.

6.3 Evaluation of Clustering Quality

The previous section tells us that KLMAXDIST is able 
to pick up certain subtleties which the other measures were 
not able to. However, while hierarchical clustering can give 
us some good initial insights, it does not give an objective 
measure of cluster quality. Let us verify this more rigorously 
by examining the clustering quality.

There are a variety of evaluation criteria to compare clus-
ters, each with its own benefit. We will assume the set-
matching F-measure found in [25]. Let C be the clusters 
generated from using SAX with one of the measures dis-
cussed and G be the ground truth. Then, we define the 
F-measure between C and G. Define the precision of C 
given G be

\[
Precision(C,G) = \frac{|C \cap G|}{|C|}
\]

Then, we can define the F-measure of cluster quality to be

\[
F = \sum_{i} \frac{|G_i|}{n} \max_{j} \{F(G_i,C_j)\}
\]

where

\[
F(G_i,C_j) = \frac{2 \cdot \text{Recall}(G_i,C_j) \cdot \text{Precision}(G_i,C_j)}{\text{Recall}(G_i,C_j) + \text{Precision}(G_i,C_j)}
\]

and

\[
\text{Recall}(G,C) = \text{Precision}(C,G)
\]
Table 2: PAM Clustering Quality for Selected UCR Archive Datasets ($r = 12, w = 26$)

<table>
<thead>
<tr>
<th>MEASURES</th>
<th>50words</th>
<th>Adiac</th>
<th>Car</th>
<th>Control</th>
<th>Wine</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAW-DTW</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>EDITDIST</td>
<td>0.42</td>
<td>0.59</td>
<td>0.13</td>
<td>0.31</td>
<td>0.38</td>
</tr>
<tr>
<td>MINDIST</td>
<td>0.51</td>
<td>0.54</td>
<td>0.25</td>
<td>0.33</td>
<td>0.34</td>
</tr>
<tr>
<td>APXDIST</td>
<td>0.56</td>
<td>0.57</td>
<td>0.27</td>
<td>0.45</td>
<td>0.38</td>
</tr>
<tr>
<td>KLMAXDIST ($L=2$)</td>
<td>0.63</td>
<td>0.64</td>
<td>0.24</td>
<td>0.36</td>
<td>0.37</td>
</tr>
<tr>
<td>KLMAXDIST ($L=3$)</td>
<td>0.65</td>
<td>0.63</td>
<td>0.27</td>
<td>0.34</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Figure 4: Cluster Quality Comparison

around medoids (PAM) with $k = 5$. We will examine the KLMAXDIST results with $L = 2$ and $L = 3$. Table 2 summarizes the clustering quality of these measures using PAM and Figure 4 provides a graphical illustration. We see with the 50words, Adiac and Car dataset KLMAXDIST has a better clustering quality than the other measures. We can conclude that KLMAXDIST gives better quality clustering results if the number of observations is large. This is to be expected as more observations means higher variability in the symbols. And since we are using $L$-tuple, this means we have more information for comparison. If we increased the alphabet size, it would also increase variability in the symbols which also gives us better results.

An interesting thing to note is that when we increase $L$ from 2 to 3, we obtained a better result for the 50words and Car dataset and not the Adiac and Control dataset. The number of sequences we cluster could be the reason for this phenomenon but it remains unclear until we can formulate a more rigorous experiment.

7. CONCLUSIONS

For this research, we proposed using $L$-tuples with max KL-divergence dissimilarity to improve clustering for SAX-represented time-series. We’ve showed that through experiment that our approach was better at capturing trends and shifts for the underlying data by considering the position and ordering of symbols than MINDIST. We’ve also shown that with SAX sequences with high variability stemming from a large alphabet or a high number of observations, KLMAXDIST provides better clustering results than the other measures. There are several possible extensions to consider.

First, we have only experimented with hierarchical clustering and PAM. We can consider different clustering algorithms to see if such choice have an impact on our results. It is common in time-series clustering to use partitional methods and so we can consider $k$-modes and other methods. It would also be interesting to see if we can devise a density-based algorithm for $L$-tuples. A second possible extension is testing how the results are affect when we vary the parameters $r, w, L$ and the number of clusters $k$. As mentioned we would expect the increase in the alphabet and sequence length to increase variability which should result in better clustering results from KLMAXDIST. The quality evaluations indicates that there are certain cases where a higher $L$ value is beneficial. However, as with $n$-grams, increasing the size of the window comes with the tradeoff of efficiency as it increases the size of the frequency vectors exponentially. So, thirdly, as we did not evaluate efficiency, this itself is something we need to investigate in the future. Lastly, another extension which we are doing in the future is using other clustering quality measure such as entropy and tightness.

8. REFERENCES


[8] B. E. Blaisdell. A measure of the similarity of sets of


Figure 2: OECD Dataset of 18 Time-Series
Figure 3: Hierarchical Clustering of OECD Data with Four Different Measures