An Efficient Visual Assessment of Cluster Tendency Tool for Large-scale Time Series Data Sets

Timothy B. Iredale, Sarah M. Erfani and Christopher Leckie
Department of Computing and Information Systems, The University of Melbourne, Australia
Email: tiredale@student.unimelb.edu.au, {sarah.erfani,caleckie}@unimelb.edu.au

Abstract—**Data visualization has always been a vital tool to explore and understand underlying data structures and patterns. However, emerging technologies such as the Internet of Things (IoT) have enabled the collection of very large amounts of data over time. The sheer quantity of data available challenges existing time series visualisation methods. In this paper we present an introductory analysis of time series clustering with a focus on a novel shape-based measure of similarity, which is invariant under uniform time shift and uniform amplitude scaling. Based on this measure we develop a Visual Assessment of cluster Tendency (VAT) algorithm to assess large time series data sets and demonstrate its advantages in terms of complexity and propensity for implementation in a distributed computing environment. This algorithm is implemented as a cloud application using Spark where the run-time of the high complexity dissimilarity matrix calculations are reduced by up to 7.0 times in a 16 core computing cluster with even higher speed-up factors expected for larger computing clusters.**

I. Introduction

Time series data are a common form of temporal data being produced in ever increasing quantities driven by the pervasive deployment of devices capable of continuous measurement. The ability to interact with environments remotely and in real time through large quantities of network-connected devices is the essence of the Internet of Things (IoT) paradigm that has seen significant growth in recent years. Time series analysis is an important tool to analyse IoT data streams. Such an analysis capability is a key enabler for optimising the performance of dynamic systems under the control of the IoT.

An important aspect of IoT data analytics is using clustering techniques to characterise the underlying distribution of the observed measurement data. Within the context of IoT applications, cluster estimation and clustering algorithms of time series data present a significant computational challenge. Streaming sensor data from distributed, voluminous sources producing inherently high-dimensional time series presents computational challenges to the calculation of similarity, the estimation of cluster count and the cluster assignment process. For example, pairwise relational similarity matrices of 10000×10000 entries would exceed commercial memory limits, with even more resources required to calculate such a matrix from raw time series. Overcoming this complexity is a significant barrier for time series data streams from many thousands or even millions of individual IoT devices.

To address these massive scales this paper focuses on distributed approaches to aspects of clustering, touching on the choice of similarity metric as well as efficient methods of cluster count estimation and cluster assignment. Techniques based on the Visual Assessment of cluster Tendency (VAT) algorithm [1], [4] are the focus of this paper as they have been demonstrated to produce clear depictions of cluster structure in data sets using only a relational dissimilarity matrix as input. More recent work on VAT has highlighted opportunities for scaling to Big Data sets [3], [9], making them particularly suitable to IoT scale application. However, all previously demonstrated approaches assume that dissimilarity matrices are readily available and centrally located, which are not trivial assumptions for time series Big Data.

To address this issue we propose a distributed algorithm that calculates the dissimilarity matrix from distributed time series data using a novel similarity metric with modest complexity. The algorithm is complemented by a scalable version of the VAT algorithm that visualises a smaller representative sample of the data, which we modify to allow for the distributed selection of a representative sample from the data.

The key contributions of this paper are (i) the incorporation of an efficient shape-based distance for use as a time series dissimilarity metric, (ii) insight into methods of reducing the computation time for relational dissimilarity data in a distributed computing environment, and (iii) a summary of the potential for applying the previous insights to scalable VAT image calculation applications. We quantitatively assess the performance using an Apache Spark application which implements this system. The results obtained demonstrate significant scaling-factor reductions in run time including 7.0 times reduction in dissimilarity matrix calculation for a 16 core computing cluster, relative to single core performance, and reduction in VAT reordering time to negligible quantities. The experiments also expose the impact of an iterative bottleneck in the VAT image generation that ultimately limits the performance improvement obtained by parallelisation.

A brief introduction of the relevant background information precedes a discussion of opportunities for a distributed approach to producing VAT images, followed by a a demonstration of the effectiveness of one such distributed approach based on an empirical analysis of a cloud computing deployment.

II. Background

To motivate the line of investigation presented, this section discusses an initial overview of the core principles with a focus on computational complexity and scalability.

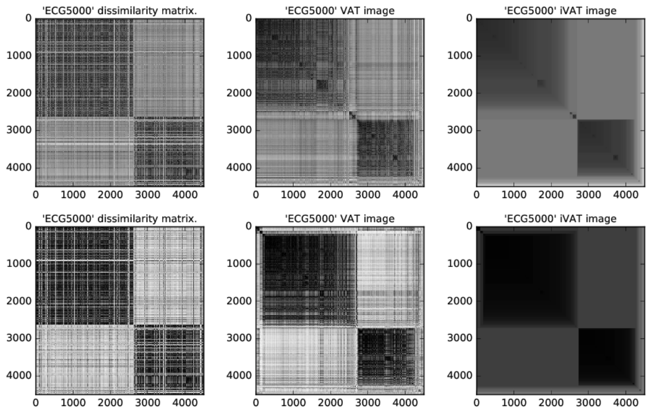

Fig. 1. Generation of iVAT image from time series data set *ECG5000* using Euclidean distance (*top row*) similarity metric and shape-based distance (*bottom row*).

A. (Dis)similarity Metrics for Time Series

This paper focuses on similarity comparisons for time series of equal lengths with uniformly spaced samples of a single dependent variable. 'Similarity' metrics can be used to compute a distance between two time series, the value of which can be interpreted as a measure of dissimilarity, with larger relative values indicating instances that are more dissimilar.

Two fundamental metrics widely referenced in the literature are Euclidean Distance (ED) and Dynamic Time Warping (DTW) [8], [12], [13]. While the former has time complexity $O(m)$ for m-dimensional time series, the latter's complexity is $O(m^2)$, which is a significant hurdle for typically high-dimensional time series. Recent work [10] introduced a shape-based distance (SBD) as a computationally efficient similarity measure that is tolerant to uniform amplitude scaling and uniform time shifting.

Shape-based distance is based on normalised cross-correlation (NCC). A uniform time shift invariant dissimilarity metric can be obtained by using the maximum value of the NCC sequence as:

$$d_{SBD}(x, y) = 1 - \max\left(NCC(x, y)\right) \tag{1}$$

By exploiting the convolution property that equates cross-correlation in the time domain with multiplication in the frequency domain, the implementation complexity is only $O(m \log m)$, for m-dimensional time series, as the iterative NCC calculation is replaced with the recursive Fast Fourier Transform (FFT) algorithm.

This efficient SBD calculation is novel in the clustering literature. Its attractive complexity and invariance to both uniform time shift and uniform amplitude scaling motivates it as a prime candidate for Big Data time series clustering.

B. Visual Assessment of Cluster Tendency

The VAT algorithm was introduced in [1] as a method for reordering a dissimilarity matrix D to produce D^*, then displaying the individual elements of the matrix as grayscale pixels. Visual evidence of clusters is observed as darker 'blocks' on the matrix diagonal. Examples of this are seen in Figure 1. Conceptually this grouping of darker components occurs when similar instances are placed adjacent to one another such that the individual dark pixels, representing relatively low dissimilarity of the two corresponding instances, appear grouped together.

A method for finding a contiguous adjacent ordering of similar values is based on a modified version of Prim's Algorithm, which forms the minimum spanning tree (MST) of a graph by iteratively adding nodes to the partially developed MST with the smallest possible cost. In this context the dissimilarity matrix D is represented as a fully connected graph with nodes representing the data set instances and interconnecting edges weighted by the corresponding value in the dissimilarity matrix. VAT effectiveness is dictated by the ability to discern well-defined dark blocks in the image. Ideal clusters, having large inter-cluster separation and small intra-cluster variance, are said to be compact separated (CS), a desirable property that enhances the contrast of clustering structure in VAT images [5]. Not only are CS clusters easier to discern visually from a VAT image, but the correctly identified cluster count, c, can be trivially used to form a single-linkage (SL) c-partitioning of the data set. It is shown in [6] that SL clustering always produces aligned clusters of a VAT image, so in cases where CS clusters allow the cluster count to be discerned unambiguously, automated SL partitioning, which involves cutting the c largest edges in the MST, is an effective clustering method.

The improved VAT (iVAT) algorithm [4] is an extension to the VAT algorithm, which aims to clarify the image for human interpretation, while using the same reordering identified in the base VAT algorithm. This algorithm performs a transformation on D^*, the VAT reordered dissimilarity matrix, to produce D'^*, where dissimilarity values are transformed to a path-based distance that quantifies the maximum single span cost of the minimum weighted path connecting two nodes. The effect of this additional transformation can be seen in figure 1. A major challenge for both VAT and iVAT is their $O(n^2)$ complexity for $n \times n$ dissimilarity matrices.

III. Distributed Algorithm Approaches

In this section we present our application of VAT-assisted cluster estimation for IoT-scale time series data. To meet the Big Data challenge, we discuss different options for reducing computation time through the distribution of component algorithms. The two main components of our algorithm are (i) the calculation of the dissimilarity matrix, which then provides input for (ii) the calculation of the VAT image. The following sections discuss the distribution of these component algorithms separately.

A. Distribution of Dissimilarity Matrix Calculation

For a set of time series, T, containing n instances of length m, the dissimilarity matrix, D, is formed through the pairwise calculation of dissimilarity for each pair in $T \times T$. Calculating

Algorithm 1: BigDisSimMat

```
Input : $T = \{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_n}\}$ - set of $n$ time series instances.
        $p$ - number of initial partitions in Spark.
        d - choice of time series dissimilarity metric.
Output: $D_n$ - $n \times n$ dissimilarity matrix.

Distribute $T$ as a set of key-value pairs in $p$ partitions.
$T_{RDD} = \texttt{parallelize}(\{(i, \mathbf{t_i}) \;\forall\; \mathbf{t_i} \in T\}, \texttt{partitions} = p)$
Calculate dissimilarity values from the Cartesian product of all key-value pairs.
$T_{\text{cart}} = \texttt{cartesian}(T_{RDD})$    Creates $p^2$ partitions.
$T_{dis} = \{((i, j), d(\mathbf{t_i}, \mathbf{t_j})) \;\forall\; ((i, \mathbf{t_i}), (j, \mathbf{t_j})) \in T_{\text{cart}}\}$
Reduce $T_{dis}$ to $n$ row-indexed arrays using combineByKey.
$T_{rows} = \{(i, [d(\mathbf{t_i}, \mathbf{t_1}), \ldots, d(\mathbf{t_i}, \mathbf{t_n})])\}$
Collect unsorted rows of $T_{rows}$.
Stacks arrays from $T_{rows}$ then sorts by row index $i$ to produce dissimilarity matrix $D_n$.
```

Algorithm 2: Big siVAT

```
Input : $T = \{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_N}\}$ - set of $N$ time series instances.
        d - choice of time series dissimilarity metric.
        $c'$ - overestimate of actual number of clusters.
        $\hat{n}$ - approximate output sample size.
Output: $D_n^*$ - $n \times n$ iVAT image of sampled dissimilarity matrix $D_n$

Distribute $T$ as a set of key-value pairs in $p$ partitions.
$T_{RDD} = \texttt{parallelize}(\{(i, \mathbf{t_i}) \;\forall\; \mathbf{t_i} \in T\})$
Iterative selection of keys, $m_s$, of $c'$ distinguished instances by distributed calculation of corresponding row, $y_{m_s}$, of dissimilarity matrix.
$m_1 = 1, \;\; y_1 = [d(\mathbf{t_1}, \mathbf{t_1}), \ldots, d(\mathbf{t_1}, \mathbf{t_N})]$
for $s \leftarrow 2$ to $c'$ do
    $y_{m_s} = [d(\mathbf{t_{m_s}}, \mathbf{t_1}), \ldots, d(\mathbf{t_{m_s}}, \mathbf{t_N})]$
    $y_{m_s} = [\min\{y_{m_s}[1], y_{m_{s-1}}[1]\}, \ldots, \min\{(y_{m_s}[N], y_{m_{s-1}}[N])\}]$
    $m_s = \arg\max_{1 \le j \le N}\{y_{m_s}[j]\}$
Centralised grouping of time series with their nearest distinguished objects.
$S_1 = S_2 = \ldots = S_{k'} = \emptyset$
for $s \leftarrow 1$ to $N$ do
    $l = \arg\min_{1 \le j \le c'} d(\mathbf{x_{m_j}})$
    $S_l = S_l \cup \{s\}$
Randomly select data near each distinguished object to define $D_n$.
for $s \leftarrow 1$ to $N$ do
    $n_t = \lceil \hat{n}/N \times |S_t| \rceil$
    Randomly select sampled index $\tilde{S}_t$ from $S_t$.
$\tilde{S} = \bigcup_{t=1}^{k} \tilde{S}_t$
Filter $T_{RDD}$ with $\tilde{S}$ to produce $T_{\tilde{S}}$.
Use Algorithm 1 to produce $D_n$ from $T_{\tilde{S}}$.
Apply iVAT to $D_n$ returning $D_n^*$.
```

the similarity metric with complexity $d(m)$ for each ordered pair in the $n \times n$ matrix gives a computational complexity of $O(d(m)n^2)$ for the full dissimilarity matrix calculation. For a subset of T of size $k \le n$, any portion of the $k \times k$ sub-matrix can be calculated independently and trivially recombined with other partitioned sub-matrices to assemble the dissimilarity matrix. This approach allows independent partitions to be executed in a distributed manner, though the complexity of the algorithm still remains $O(d(m)n^2)$.

B. Distribution of VAT Image Calculation

In this section an approximate approach for theoretically reducing this complexity is presented.

Scalable VAT / scalable iVAT (sVAT / siVAT) has been shown in [3], [7]. Rather than distributing the data and computing the VAT image in parallel, this algorithm uses a smaller representative sample of the original data and calculates the VAT image centrally. This method requires two user specified parameters: c' an initial overestimate of the number of clusters, and $\hat{n}$ an approximate size for the sampled data set. The algorithm uses the cluster overestimate to identify c' distinguished instances, i.e., samples from the data set that are guaranteed to represent each of the true clusters [3]. The remaining data points are grouped into proto-clusters based on their nearest distinguished point before a sample is taken from each proto-cluster, which together form the sampled data set. Importantly, the sampling process guarantees that the ratio of samples per proto-cluster in the sampled set is at least the same as the ratio of samples of the complete proto-cluster relative to the full data set. This condition ensures the sample is representative of the original data. Under these conditions, [3] and [7] demonstrate that a true VAT image is produced for the dissimilarity matrix sampled in this way and that this image is representative of the clustering structure in the full data set provided that $c' \ge c$, the true number of clusters. Under these conditions the data excluded from sampling can be mapped to the clusters identified in the resulting VAT image using a Nearest Prototype Rule (NPR), in which each out-of-sample instance is assigned to a cluster in the same way as the least dissimilar instance in the sampled image. It is shown in [3] that this approach scales linearly with the number of instances in the data set where the data set is very large, which is appropriate for the Big Data problem addressed in this paper.

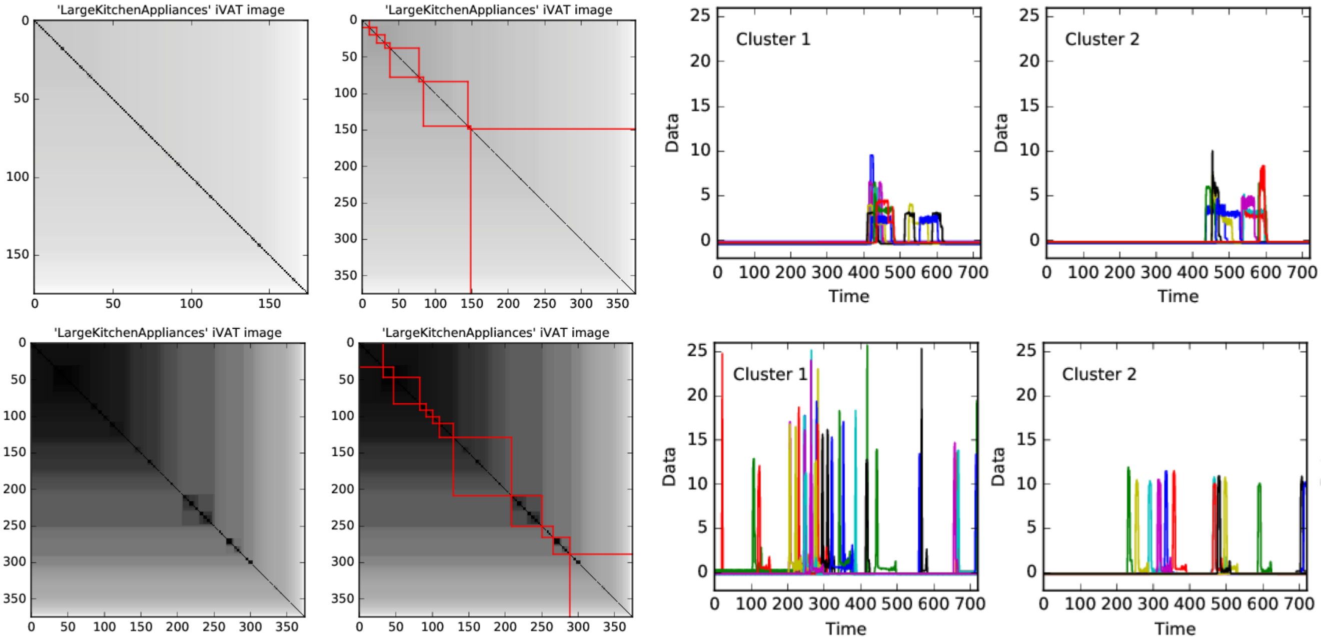

Fig. 2. iVAT image with identified clusters and time series as identified by the first two clusters on the iVAT image diagonal. *Top row:* Clustering using Euclidean distance. *Bottom row:* Clustering using shape-based distance.

In addition to providing a method for cluster tendency assessment, [7] and [9] trivially extend this concept to a single linkage clustering method for Big Data sets in a method similar to that outlined in Section II-B. This approach is shown to be effective even for data sets with non-CS clusters, although in these cases the cluster blocks in the VAT images are more difficult to discern due to reduced contrast in the image. The aspects surrounding VAT effectiveness are reassessed in the context of time series similarity in Section IV-A.

C. A Spark Implementation

This presents the detailed algorithm and challenges involved in mapping the dissimilarity matrix and siVAT approaches covered in Sections III-A and III-B to a distributed computing environment. Ultimately the main contributions of this work are the assessment of the performance improvement of parallelising the dissimilarity matrix calculation and the feasibility of the scalable VAT approach for time series data based on this approach.

Apache Spark was chosen as a platform for implementing this system in a MapReduce environement. While Spark's ability to optimise stage execution is well-developed, design decisions that aim to maximise in-memory operations by reducing the amount of data shuffled between nodes can be used to further tune performance.

The implementation of the dissimilarity matrix calculation outlined in Section III-A is shown in Algorithm 1. Initially the input times series data are distributed as a Resilient Distributed Data set (RDD), Spark's native distributed data structure, of key-value pairs. Spark's `cartesian` function is used to transform this RDD to its Cartesian product allowing pair-wise dissimilarity values to be calculated for the chosen dissimilarity metric d. The initial keys and ordered key pairs formed by the Cartesian product are necessary for correctly ordering dissimilarity values when the dissimilarity matrix is constructed. The n indices of the original time series data form n^2 ordered pairs, interpreted as (row, col) indices that uniquely identify their position in the dissimilarity matrix. Line 4 of Algorithm 1 demonstrates how `combineByKey` uses these ordered pairs to develop individual rows of the dissimilarity matrix, where each value of that row is placed in position col of a row-specific array. The column-sorted rows are then collected by the driver program where they are stacked and sorted by row index to produce the full dissimilarity matrix.

The implementation of siVAT introduced in [3] and summarised in Section III-B is modified to allow for the distributed calculation of dissimilarity matrix rows necessary for determining distinguished objects. This architecture allows for very large time series data, stored in a distributed storage system, to be processed. These modifications are shown in Algorithm 2. Again, the time series data is distributed as an RDD of p partitions, as in Algorithm 1, from which the distributed calculation of rows, y, of the dissimilarity matrix associated with each distinguished point is performed and collected by the driver program. Each successive point is determined by its dissimilarity with the previous point. This necessarily iterative communication between the driver program and the computing cluster creates a performance bottleneck. Once all distinguished points are determined the proto-cluster allocations and subsequent sampling index, $\tilde{S}$, of size t are determined centrally by the driver program. The dissimilarity matrix of the sampled data is then calculated using Algorithm 1. This method of sampling impacts the level of parallelisation as it is dependent on the uniformity of the distribution of sampled instances across the computing cluster. This point is discussed in more detail in Section IV-C.

IV. Experimental Results

In this section a number of experiments on three time series data sets are analysed to assess the suitability of the choice of similarity metric and the performance improvement obtained by distributing the VAT image calculation as proposed in Section III. The time series data is taken from the UCR Time Series Classification Archive [2] which provides sets of labelled time series for the purposes of quantifying the performance of classification and clustering approaches.

In the following experiments up to eight virtual machines were available for building a Spark computing cluster, each defined with two CPU cores, meaning that a maximum of 16 cores could be allocated for parallel use by the Spark application. Three data sets were selected from [2] to demonstrate different aspects of the system design. Firstly the *ECG5000* data set, a very large data set consisting of 4500 instances of 140 dimensions, was chosen to demonstrate the effectiveness of clusters generated by the modified sVAT approach. Second, the *LargeKitchenAppliances* data set, with 375 instances of length 720, was chosen for its predominantly asynchronous time series instances to demonstrate the value of the shift-invariance of the k-shape dissimilarity metric. Finally, the *FaceAll* data set, with 1690 instances of 131 dimensions, was chosen for its intermediate sample size, and hence computation time, that facilitated repeated experiments when assessing the effect of the distributed implementations of VAT and sVAT algorithms on computation time.

A. Time Series Similarity Metrics

In this set of experiments the time series similarity metrics discussed in Section II-A, namely Euclidean distance and the novel shape-based distance, are compared for a number of different time series data sets to qualitatively assess the suitability of the similarity metric choice under different conditions, see Figures 1 and 2.

It can be seen that for the well-synchronised time series data set *ECG5000*, the iVAT images based on ED and SBD are capable of capturing the same major clustering structure, though the ED metric provides a higher contrast image.

A similar comparison is made for the *LargeKitchenAppliances* data set, which has significant uniform time shift variation between instances. It is clear in this case the ED iVAT image is a very poor indicator of clustering tendency with almost no discernible clustering structure visible. With some difficulty, minor clustering structure was extracted, however, analysis of these clusters indicates they reveal less effective groupings. The individual time series from the first two clusters identified are plotted in Figure 2 showing how the ED clustered instances are grouped by a measure of synchronisation. In contrast the SBD iVAT image much more clearly identifies clustering structure, and subsequent cluster analysis indicates effective shape matching regardless of synchronisation. This is a direct result of SBD's invariance to time shifting. The shape-based metric is a more intuitive similarity so SBD would likely be considered the stronger metric for any data set that shows low levels of synchronisation.

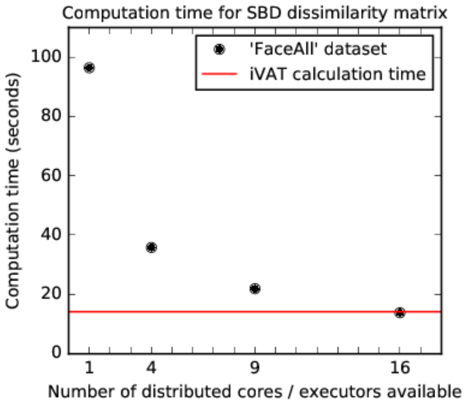

Fig. 3. Computation time of the full NCC dissimilarity matrix for the *FaceAll* data set as a function of computing cluster size.

It is important to note that in all cases the identification of clusters is highly subjective. In many cases rigorous inspection is required to reveal clustering structure. In most cases some subjective cut-off is required as to whether clustering blocks should be further subdivided into finer resolution clusters. The highlighted cluster boundaries in the SBD iVAT image of Figure 2 are good examples of this. Some of the ambiguity that arises is a result of outliers in the data with respect to this dissimilarity metric. These real world data subtleties mean compact separated clusters are unlikely, limiting the applicability of automated single linkage clustering from VAT images, mentioned in Section II-B.

B. Distributed Dissimilarity Matrix

These experiments aimed to determine the validity of the dissimilarity matrices produced by Algorithm 1 as well as measuring the execution time of the program as a function of the size of the Spark computing cluster.

The dissimilarity matrices produced by Algorithm 1 are identical to those produced by other implementations on single machines, which validates the efficacy of the distributed program. What follows is an analysis of the computation time speed-up.

Critical to the analysis of dissimilarity matrix calculation in Spark was the observation that the Cartesian product of a p partition RDD forms a new RDD with p^2 partitions. Each of these partitions is mapped onto one of the computing cluster cores allocated to the Spark application, so for maximum parallelisation it is essential that p^2 cores are available, one to process each partition. If the number of cores available is less than p^2 some partitions will be queued until previous partitions have been processed, resulting in reduced parallelisation. If more than p^2 cores are available only p^2 cores will be allocated

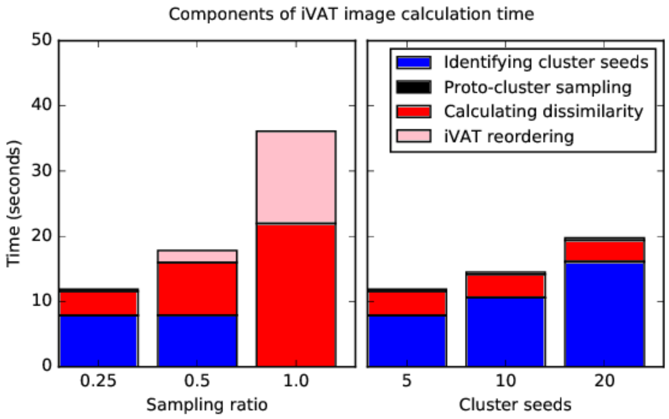

Fig. 4. Impact of siVAT parameters on runtime - r is varied for fixed $c' = 5$, and c' is varied for fixed $r = 0.25$. *'FaceAll'* data set [2] on a nine core computing cluster using SBD metric.

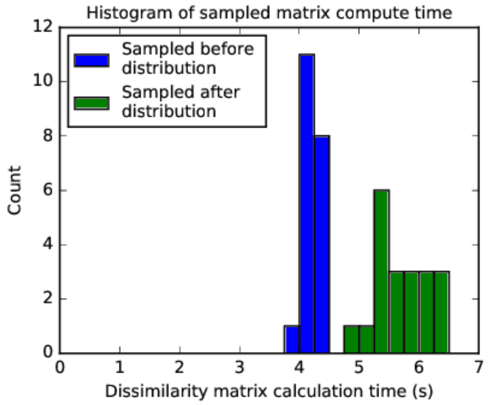

Fig. 5. Impact of sampling after distribution on distributed dissimilarity matrix computation time. Both sets of experiments use sampling ratio of 0.1 on *'FaceAll'* data set [2]. Implementation on nine core computing cluster using SBD metric.

a partition, resulting in under-utilisation of the available CPU resources.

With this in mind, four different computing cluster sizes were analysed for their dissimilarity matrix compute time, as seen in Figure 3. The computing cluster sizes were chosen to match the p^2 partitions for initial $p \in \{1, 2, 3, 4\}$. The results show a significant improvement in computation time, with a speed-up factor of 7.0 times for the 16 core case relative to the 1 core case.

C. Scalable VAT Image Calculation Time

These experiments aimed to investigate how key parameters of the sVAT algorithm influenced the execution time of different components of the program. This broadly aimed to understand if the sVAT image calculation was of comparable complexity to the dissimilarity matrix development in an implementation of Algorithm 2.

Figure 4 shows how sampling ratio, r, and the cluster count seed, c', impact the end-to-end computation time from raw time series data to siVAT image. It is clear that, given a fixed computing cluster configuration, the identification of distinguished points and the final iVAT image calculation are the algorithmic components most sensitive to c' and r.

The sampling ratio, r, reduces the size of the dissimilarity matrix passed to the iVAT algorithm. It is observed that even a relatively small scaling ratio of 0.25 reduces the distributed dissimilarity matrix calculation significantly and the centralised VAT calculation time to an almost negligible amount of the total program execution time.

For the dissimilarity matrix calculation, the relative speed-up achieved by varying r from 1.0 (no sampling) to 0.5, and 0.5 to 0.25 is 2.8 and 2.1 respectively for the example shown, though this varies depending on the sampling index, $\tilde{S}$. In addition, the speed up is observed to be less than a factor of 4.0 expected for a two fold reduction in the size of the data set.

To better understand these observations a comparison was made between the computation time of a sampled dissimilarity matrix, where sampling is performed after distribution of the data as in Algorithm 1, with the computation time of a similar matrix from the same data set where random sampling with the same r was performed before the data was distributed and subsequently passed to Algorithm 2. The results are shown in Figure 5 which highlight that the computation time of a data set sampled after distribution shows a notable increase in the mean and variance of the dissimilarity matrix computation time. These increases stem from the non-uniform distribution of data across the computing cluster resulting from the random sampling process. The importance of carefully specified uniform partitions to maximise parallelisation was observed in Section IV-B, a property that cannot be guaranteed when the data set is sampled after partitioning.

The centralised VAT image calculation time is also impacted significantly by r, with speed up of 7.7 and 5.1 observed when varying r from 1.0 to 0.5, and 0.5 to 0.25, respectively, which exceeds the expected speed-up of 4.0 times. This discrepancy is likely attributable to the consumption of other compute resources, such as memory and cache, for the larger matrices, an effect which is alleviated as the size of the input dissimilarity matrix reduces. These results highlight the need for careful dimensioning of compute resources across the computing cluster to fully capture the benefits of distributed computation.

In the second part of Figure 4 the impact of the number of cluster seeds, c', clearly exposes the iterative bottleneck that arises in Algorithm 2, previously highlighted in Section III-C. For smaller sampling ratios the iterative identification of distinguished instances is significant, forming a lower bound on the total computation time that increases with the number of cluster seeds. For data sets with many clusters this bottleneck will be a key limitation on performance.

D. Inferred Clustering Using Down-sampled iVAT Images

Section IV-C quantifies factors affecting computation time of sVAT for times series, while this section attempts to quali-

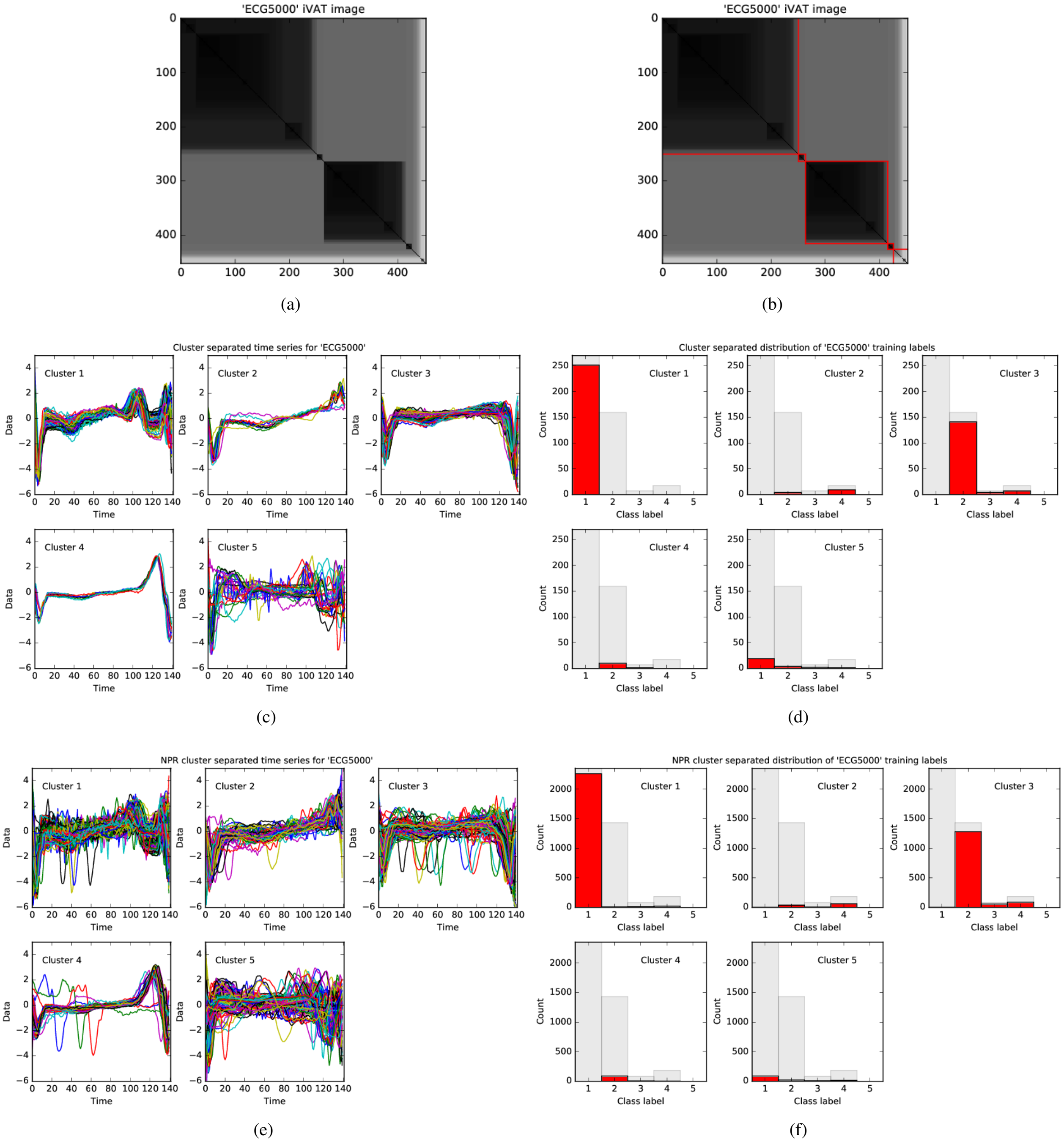

Fig. 6. (a) siVAT image and (b) identified clusters for *ECG5000* data set using shape-based distance similarity metric. (c) Cluster separated time series and (d) distribution of class labels as given in *ECG5000* data set for siVAT image. (e) Cluster separated time series and (f) distribution of class labels as given in *ECG5000* data set for out-of-sample time series instances, with cluster assignment inferred by nearest prototype rule. In (d) and (f) the distribution of class labels in each cluster (red) is shown overlaid with the total class label distribution of the data set (grey).

tatively assess the effectiveness of time series clustering using down-sampled iVAT images. This is achieved by introducing the existing class labels provided with the UCR time series. In many cases these class labels can be effectively mapped to shape-based clusters. Examining distributions of class labels within clusters identified by iVAT images is one way of assessing the effectiveness of the iVAT image.

In Figure 6, the SBD similarity iVAT image for a representative 0.1 sampling of the *ECG5000* data set has been generated and the clusters identified manually. Figures 6c and 6d show the time series and histogram of class labels for each cluster identified. It is clear that clusters 1 and 3 distinctly separate the majority of instances with class label 1 and 2, respectively. It is clear that this is due to the distinct shape of these classes. Cluster 2 and 4 isolate two small, distinct groups of similarly shaped instances from class labels 4 and 2 respectively. Both clusters have a distinct shape from the other instances identified of these classes in cluster 3. Cluster 5 consists of outliers, showing no uniform dark section in the iVAT image nor clear shape similarity.

Figures 6e and 6f are the time series and histogram of class labels of the out-of-sample instances of the *ECG5000* data set, as inferred by the nearest prototype rule. The well separated clusters appear here again, with class labels 1 and 2 effectively isolated by clusters 1 and 3 respectively.

For this data set the siVAT image is a very effective method of identifying the major clusters, though less effective at identifying the remaining, much smaller, classes 3 to 5. Reflecting on Figure 1 it is unlikely that this would have been improved in the full iVAT image as there is clear evidence of outliers that are not associated with the large dark sections in this image as well.

Very similar results are observed for the ED metric on this data set, and examining other data sets it is evident that the NPR rule inferred clustering agrees very well with the original manual clustering of the down-sampled image even in cases with more complex clustering structure. In the interest of brevity these results are not shown here.

V. Conclusion

In this paper we presented a new algorithm for distributing the high complexity calculation and reordering of relational dissimilarity matrices for VAT analysis in a time series data context. The algorithm is suitable for use in Big Data settings where storage and processing of data is performed in a distributed computing system. The algorithm incorporates previous centralised approaches, extending their usefulness to Big Data settings, as well as making use of a flexible, modest complexity, shaped-based metric for time series comparison.

We demonstrated the efficacy of this approach using a Spark implementation which highlighted the need for careful control over partitioning of data sets, as well as an iterative bottleneck that limits distribution potential in some cases where complex clustering structure is being analysed.

Further work would involve a comparison of this partitioning approach with the Cartesian Scheduler proposed in [11] which claims significant performance improvements over standard Spark Cartesian product implementations. We would also like to extend this analysis to real world Big Data and very large compute clusters to better understand the interplay between the dissimilarity matrix calculation, the iterative bottleneck and the down-sampled VAT image generation.

References

[1] Bezdek, J.C., Hathaway, R.J.: Vat: A tool for visual assessment of (cluster) tendency. In: Proc. IJCNN. pp. 2225–2230 (2002)

[2] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The ucr time series classification archive (July 2015), www.cs.ucr.edu/~eamonn/time_series_data/

[3] Hathaway, R.J., Bezdek, J.C., Huband, J.M.: Scalable visual assessment of cluster tendency for large data sets. Pattern Recognition 39, 1315–1324 (2006)

[4] Havens, T.C., Bezdek, J.C.: An efficient formulation of the improved visual assessment of cluster tendency (ivat) algorithm. IEEE Transactions on Knowledge and Data Engineering 24(5), 813–822 (2012)

[5] Havens, T.C., Bezdek, J.C., Keller, J.M., Popescu, M.: Dunn's cluster validity index as a contrast measure of vat images. In: ICPR 2008. 19th International Conference on Pattern Recognition, 2008. pp. 1–4. IEEE (2008)

[6] Havens, T.C., Bezdek, J.C., Keller, J.M., Popescu, M., Huband, J.M.: Is vat really single linkage in disguise? Annals of Mathematics and Artificial Intelligence 55(3-4), 237–251 (2009)

[7] Havens, T.C., Bezdek, J.C., Palaniswami, M.: Scalable single linkage hierarchical clustering for big data. 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing p. 396 (2013)

[8] Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 23(1), 67–72 (1975)

[9] Kumar, D., Bezdek, J.C., Palaniswami, M., Rajasegarar, S., Leckie, C., Havens, T.C.: A hybrid approach to clustering in big data. IEEE Transactions On Cybernetics (2015)

[10] Paparrizos, J., Gravano, L.: k-shape: Efficient and accurate clustering of time series. SIGMOD Record 45(1), 69 (2016)

[11] Phinney, M., Lander, S., Spencer, M., Shyu, C.R.: Cartesian operations on distributed datasets using virtual partitioning. In: 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService). pp. 1–9. IEEE (2016)

[12] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 262–270. ACM (2012)

[13] Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery 26(2), 275–309 (2013)