Progressive Similarity Search on Time Series Data

[Vision Paper]

Anna Gogolou\textsuperscript{1} Theophanis Tsandilas\textsuperscript{1} Themis Palpanas\textsuperscript{3} Anastasia Bezerianos\textsuperscript{2}

\textsuperscript{1}Inria, Univ. Paris-Sud & CNRS, and Univ. Paris-Saclay, France
\textsuperscript{2}Univ. Paris-Sud & CNRS, Inria and Univ. Paris-Saclay, France
\textsuperscript{3}Univ. Paris-Descartes, France
\{first.last\}@inria.fr \{first.last\}@lri.fr \{first\}@mi.parisdescartes.fr

ABSTRACT

Time series data are increasing at a dramatic rate, yet their analysis remains highly relevant in a wide range of human activities. Due to their volume, existing systems dealing with time series data cannot guarantee interactive response times, even for fundamental tasks such as similarity search. Therefore, in this paper, we present our vision to develop analytic approaches that support exploration and decision making by providing progressive results, before the final and exact ones have been computed. We demonstrate through experiments that providing first approximate and then progressive answers is useful (and necessary) for similarity search queries on very large time series data. Our findings indicate that there is a gap between the time the most similar answer is found and the time when the search algorithm terminates, resulting in inflated waiting times without any improvement. We present preliminary ideas on computing probabilistic estimates of the final results that could help users decide when to stop the search process, i.e., deciding when improvement in the final answer is unlikely, thus eliminating waiting time. Finally, we discuss two additional challenges: how to compute efficiently these probabilistic estimates, and how to communicate them to users.

Keywords
time series, progressive similarity search; progressive visual analytics; progressive error

1. INTRODUCTION

Time series (TS) are sequences of value measurements derived from a wide range of human activities or natural processes, such as temperatures per hour, blood oxygen saturation per day, or electroencephalography (EEG) signals. These sequences are becoming ubiquitous in modern life, requiring their analysis that is becoming increasingly challenging given their sizes \[21\].

Time series analysis involves tasks such as pattern matching, anomaly detection, frequent pattern identification, and time series clustering or classification. These tasks rely on the notion of time series similarity. The data-mining community has proposed several techniques, including many similarity measures (or distance measure algorithms), for calculating the distance between two time series \[9\], as well as corresponding indexing techniques and algorithms \[10\], in order to address the scalability challenges.

Nevertheless, we observe that time series similarity can be domain- and visualization-dependent \[2\] \[14\], and in many situations, analysts depend on time-consuming manual analysis processes. For example, neuroscientists manually inspect the EEG data of their patients, using visual analysis tools, so as to identify patterns of interest \[15\] \[14\]. In such cases, it is of paramount importance to have techniques that can operate within interactive response times \[20\], in order to enable analysts to complete their tasks easily and quickly.

In the past years, several visual analysis tools have combined visualizations with advanced data management and analytics techniques (e.g., \[22\] \[16\]), albeit not targeted to time series similarity search. The focus of the data management community is on the scalability issues related to the processing and analysis of very large datasets. However, the state-of-the-art methods on time series processing are still far from achieving interactive response times \[10\].

To allow for interactive response times when users analyze large time series collections, progressive and iterative visual analytics approaches need to be considered \[1\] \[28\] \[26\]. These approaches provide progressive answers to users’ requests \[12\] \[24\] \[19\], sometimes based on algorithms that return quick approximate answers \[5\] \[8\] \[11\]. They support exploration and decision making by providing progressive results, before the final and exact ones have been computed. Most of these techniques consider approximations of aggregate queries on relational databases, with the exception of Ciaccia and Patella \[5\], who examine progressive similarity search in multi-dimensional metric spaces. Nevertheless, none of these works has considered time series data, which have the additional characteristic of being high-dimensional, i.e., they are hundreds to thousands of points long.

Contributions. We demonstrate the importance of providing progressive whole-matching \[10\] similarity search results on large time series collections. Our preliminary experiments show that there is a gap between the time the 1st Nearest Neighbour (1-NN) is found and the time when the search algorithm terminates. In other words, users often wait without any improvement in their answers. We further show that high-quality approximate answers are found very early, e.g., in less than one second, so they can support highly interactive visual analysis tasks. For our benchmarks we utilize the state-of-the-art Adaptive Data Series (ADS) index \[29\], which is among the fastest techniques for answering k-Nearest Neighbor (k-NN) queries on time series data. Finally, we lay out our vision for developing progressive ap-
proaches for time series exploration and analytics tasks. We
discuss promising directions (and our ongoing work) on how
to estimate probabilistic distance bounds, and how to help
users evaluate the quality of progressive results.

2. STATE OF THE ART

Similarity Measures: Ding et al. [9] discuss measures
calculate similarity between time series. Euclidean Distance
(ED) is the most popular, performing point-by-point
value comparison between two time series. ED can be com-
bined with data normalization, often z-normalization, con-
sidering as similar patterns that may vary in amplitude or
value offset. Based on their analysis, Ding et al. [9] con-
ccluded that there is no superior measure. In our work, we
focus on ED because (i) it is an effective and the
most commonly used measure in the visualization and data-
mining literature; (ii) it leads to efficient solutions for large
datasets. (We plan to examine other measures, e.g., Dy-
namic Time Warping (DTW) in our future work.)

Similarity Search and Interactive Querying: The data-
base community has optimized similarity search methods by
using index structures [6, 27, 4, 29], or by directly optimizing
sequential scans [23]. Recently, Echihabi et al. [10] compared
these methods in terms of efficiency under a single, unified
experimental framework. Her work indicates that there is
no single best method that outperforms all the rest. In our
work, we use the state-of-the-art ADS index [29], which pro-
vides high-quality approximate answers almost immediately,
and then updates that converge fast to the exact answer.

The human-computer interaction community focuses on
the interactive visual exploration and querying of time series.
In particular, they are interested in how to form interactive
similarity search queries. Existing querying approaches top
line chart visualizations [24], rely either on the inter-
active selection of part of an existing time series [3], or on
sketching of patterns to search for [7, 15]. Although we do
not study mechanisms of querying in this paper, this line of
work is orthogonal to our approach, that considers ap-
proximate and progressive results from these queries when
interactive search-times are not possible.

Progressive Visual Analytics: A recent research direc-
tion studies the problem of how we can support interactive,
real-time visual analytics when back-end computations can-
not be performed instantaneously, as is the case of our work.
To this effect we can use progressive and iterative methods
in order to produce fast, but approximate, computational
results and visualizations, that are refined over time with
increasing precision. Fekete and Primet [11] provide a sum-
mary of the features of a progressive system, where here we
focus on how to provide: (i) progressively improved answers;
(ii) feedback about the state and costs of the computation;
and (iii) guarantees of time and error bounds for progressive
and final results. We address these features in Sec. 3 and 4.

The state-of-the-art in big data exploration takes advan-
tage of the power of distributed systems, indexing, and sam-
ping methods, and different works utilize one or more of
these methods in order to provide progressive results for dif-
ferent kinds of queries and data. Moritz et al. [12] used an
existing algorithm [5] which exploits sampling methods and
approximate query processing for incremental, approximate
aggregate database queries. On the other hand, Ciaccia and

Table 1: Experimental datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Cardinality</th>
<th>TS Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>seismic</td>
<td>seismic records</td>
<td>100M</td>
<td>256</td>
</tr>
<tr>
<td>SALD</td>
<td>MRI data</td>
<td>200M</td>
<td>128</td>
</tr>
<tr>
<td>deep1B</td>
<td>image descriptors</td>
<td>267M</td>
<td>96</td>
</tr>
</tbody>
</table>

Patella [5] studied progressive similarity search queries over
multi-dimensional spaces and proposed a probabilistic ap-
proach for computing the uncertainty of partial similarity
search results. However, their approach does not scale to
the dataset sizes and number of dimensions that we target.

We focus on very large collections (i.e., in the order of TBs)
of data series (where the dimensionality of each series
is in the order of hundreds to thousands), and how we can
develop approaches to support progressive visual analysis in
a fully interactive system. Our ultimate goal is to study
how users decide to terminate a search that is progressive
in nature (and thus reduce waiting times) when they are
provided with approximate answers and information about
their uncertainty. In particular, we are interested in the
quality of approximate answers and how to communicate to
users when no improvement is expected to be obtained even
if the search algorithm is still running.

3. PRELIMINARY OBSERVATIONS

We first investigate whether progressive time series simi-
larity search in large datasets is feasible. We examine how
early we can provide approximate answers, and how good
these answers are compared to the exact answers. To this
end, we conducted similarity search experiments on three
real datasets with the state-of-the-art ADS index [29], which
can quickly provide good initial approximate answers and
can potentially support progressive similarity search within
interactive time thresholds.

Scope. We examine an important class of queries, i.e., ap-
proximate and exact 1-NN whole-matching similarity search
queries [10]. (We expect that similar results will hold for
k-NN and r-range queries, as well as subsequence matching.
In this paper, we cover these cases in our future work.)

Environment. We ran all experiments on a Dell T630
Rack Server with two Intel Xeon E5-2643 v4 3.4Ghz CPUs,
512GB of RAM, and 3.6TB (2 x 1.8TB) HDD in RAID0.
The search algorithm is a single-core implementation.

Datasets. We tested real datasets that have also been used
in previous studies [29, 10]. They include a different number
of series and a different number of points (Table 1) but have
the same overall dataset size of 100GB. The IRIS seismic
dataset consists of seismic instrument recordings from sev-
eral stations worldwide and contains 100 million series with
a length of 256 points. The neuroscience dataset, SALD
consists of MRI data and contains 200 million series of 128
points. The image processing dataset, deep1B consists of
vectors extracted from the last layers of a convolutional neu-
ral network and contains 267 million series of size 96.

Whole-matching refers to the situation where the query and
all candidate answers (series) in the dataset have the same
length. We used the ADS index [29, 10] which performs pro-
gressive similarity search within interactive time thresholds.

References:

[2] ED (Euclidean Distance) is the most popular measure.
[6] We tested real datasets that have also been used in previous studies.
[7] They include a different number of series and a different number of points.
[8] The search algorithm is a single-core implementation.
[9] Datasets. We tested real datasets that have also been used in previous studies.
[10] They include a different number of series and a different number of points.
Queries. All our query workloads include 100 query series. We generated the query datasets by extracting random data series from the raw data. For the deep1B dataset, we used a real query workload that came with the original dataset.

Measures. For each similarity query, we recorded its overall completion time, the time for each approximate answer, as well as the time passed until the algorithm finds the exact answer to the query, i.e., the 1-NN. For each approximate and exact answer, we also recorded its Euclidean distance from the query.

Results. Table 2 summarizes our results. For each dataset, we present the average, minimum, and maximum time (in seconds) for the 1-NN query answering algorithm to first encounter the 1-NN answer (marked as 1-NN Time in Table 2), and the corresponding times for the same algorithm to finish execution (marked as Total Time). We observe that the total time waiting for a single query to finish can be long, e.g., up to three minutes, which is beyond acceptable thresholds for interactive data-analysis tasks. Moreover, these delays are orders of magnitude longer than the actual time needed to find the best answer (i.e., first encounter of 1-NN). This means that for most queries, the greatest cost is not locating the 1-NN, but rather confirming that there is no better answer: this is why the query answering algorithm finishes execution long after having retrieved the 1-NN value. This finding is consistent with results by Ciaccia and Patella [5], who report that most of the time spent in an exact NN search is “wasted time, during which no improvement is obtained.”

The time needed to locate the 1-NN was especially fast for the SALD and deep1B datasets, where average times were below 1 sec. However, times varied greatly on the seismic dataset, ranging from a few milliseconds to 48.5 sec. For 28% of the queries, the delay was greater than 10 sec, which is considered as a limit for keeping a user’s attention focused on a dialog [11]. We expect that such delays will further increase in larger datasets, and for k-NN exact search. In such cases, providing early approximate answers will also be crucial.

Fig. 1 presents the progressive results for five example queries on the seismic data. For these queries, the time to locate the 1-NN (right-most point in each curve) is relatively long (> 20 sec), while approximate answers (intermediate points in each curve) appear with various frequencies and trends. For example, for queries q1 and q65, results converge quickly (in the order of hundreds of milliseconds), and then only slightly improve. For other queries, such as q57, convergence is more progressive. In all cases, however, the first approximate answer (left-most point in each curve) becomes available very early (< 50 msec), and approximate results very close to the final answer appear within interaction times (< 1 sec).

Overall, our results indicate that (i) supporting interactive similarity search over large datasets is feasible, and (ii) providing early progressive answers to users could drastically reduce waiting times. The challenge is how to help users to assess the quality of such progressive answers and decide whether to trust these answers, or wait for a better one.

### Table 2: Summary of experimental results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1-NN Time (sec) Avg Min Max</th>
<th>Total Time (sec) Avg Min Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>seismic</td>
<td>8.5 0.017 48.5</td>
<td>92 21.3 111</td>
</tr>
<tr>
<td>SALD</td>
<td>0.4 0.003 5.2</td>
<td>49 0.24 183</td>
</tr>
<tr>
<td>deep1B</td>
<td>0.2 0.001 2.8</td>
<td>76 0.05 189</td>
</tr>
</tbody>
</table>

4. VISION AND CHALLENGES

Previous work on progressive or optimistic visual analytics [12, 24, 19] has focused on the estimation of aggregated functions, such as the mean, over random or carefully selected samples of the data. In this case, providing feedback to users about the uncertainty of progressive results can rely on common statistical methods, e.g., confidence intervals [12, 19], or coarse-grain visualizations of aggregated data [13]. However, such approaches cannot apply to our problem. Although k-NN distances converge over time (see Fig. 1), providing bounds for their error requires a different set of statistical tools.

Our vision is to develop methods for progressive time series similarity search, coupled with appropriate bounds on the errors of the intermediate results. We are inspired by the probabilistic approach of Ciaccia and Patella [5] on approximate search in multi-dimensional spaces. According to this approach, a dataset is considered as a random sample (or instance) drawn from a large multidimensional space. If the distribution of k-NN distances in this space is known for any given query, then we can derive an estimate of the k-NN distance for the sample. Unfortunately, such distributions are unknown, so the challenge is how to approximate them from a given dataset. Ciaccia and Patella [5] use this framework to determine a mix of probability and distance error bounds as stopping conditions for an approximate similarity search. Our goal instead is to provide live estimates of probabilistic distance bounds to users, and let users decide whether to trust the current results and stop, wait for a better answer, or keep the process executing in the background while they continue with a new search.

We briefly discuss promising measures that could help users evaluate the progressive results of a similarity search. Let \( T \) be a space of time series data. For a given query \( Q \in T \), we define the cumulative distribution function \( F_Q(x) = \Pr\{d_Q \leq x\} \) that gives the probability that its distance \( d_Q \) from a random series in \( T \) is lower than or equal to \( x \).
From this function, we can derive the probability distribution of k-NN distances $d_{kQ}^+$ and from this, we can estimate the expected k-NN distance and assess the uncertainty of this value. Furthermore, we can infer probabilistic distance bounds for $d_{Q}^+$. For example, we can estimate a distance bound $d^+$ such that $P_r(d_{kQ}^+ > d^+) \leq 5\%$. The $Q_k$ function can be also used to derive a probability distribution for the number of answers whose distance is better than a given distance $d_{Q}$.

Investigating additional probabilistic measures about expected CPU and I/O costs [6] can also be useful.

We note that the above directions pose two main challenges. First, the $Q_k(x)$ function can be only approximated based on a specific instance of the space, i.e., the given time series dataset. Ciaccia and Patella [5] argue that for high-dimensional spaces, this function is close to the cumulative distribution function $F(x)$ of all pairwise distances, and can be approximated from a sample of distances that are randomly drawn from a given dataset. Unfortunately, reliably estimating the probability distribution of k-NN distances requires large samples of distances. Our early tests have shown that their precomputation can be prohibitively expensive for large datasets, such as the ones in Table I or larger. We are currently working on solutions that reduce such computation costs.

The second challenge is how to communicate to users probabilistic distance bounds and errors, given that similarity distance values do not have a clear interpretation. Our goal is to integrate these distance bounds and errors into the visual representation of a time series, by either using the query itself, or its progressive k-NN answers. Finally, we are interested in experimentally assessing how users perceive and understand such probabilistic measures. We plan to measure whether, and to what extent, the visualization of these probabilistic measures helps them to effectively complete their visual analysis tasks.

5. CONCLUSIONS

In this work, we presented preliminary experiments that demonstrate the usefulness (and need) of progressive answering of similarity search queries on very large time series data. Our findings show that the greatest cost is not locating the 1-NN, but rather waiting for the algorithm to confirm that there is no better answer and finish execution.

We argue that providing progressive answers and estimates of probabilistic distance bounds to users, and letting them decide when to stop the search process, are important research questions. This would eliminate wasted time and reduce user waiting times, in cases where improvement in the final answer is not possible. We have identified two main challenges: (i) how to compute efficiently distance probability distributions for large data series collections; and (ii) how to communicate to users probabilistic distance bounds and errors. Given the increasing popularity of data series analysis tasks, these research directions are both relevant and important, offering exciting research opportunities.

6. REFERENCES


[26] Y. Wang, P. Wang, J. Pei, W. Wang, and S. Huang. A data-adaptive and dynamic segmentation index for whole data series dataset. Ciaccia and Patella [5] argue that for high-dimensional spaces, this function is close to the cumulative distribution function $F(x)$ of all pairwise distances, and can be approximated from a sample of distances that are randomly drawn from a given dataset. Unfortunately, reliably estimating the probability distribution of k-NN distances requires large samples of distances. Our early tests have shown that their precomputation can be prohibitively expensive for large datasets, such as the ones in Table I or larger. We are currently working on solutions that reduce such computation costs.

The second challenge is how to communicate to users probabilistic distance bounds and errors, given that similarity distance values do not have a clear interpretation. Our goal is to integrate these distance bounds and errors into the visual representation of a time series, by either using the query itself, or its progressive k-NN answers. Finally, we are interested in experimentally assessing how users perceive and understand such probabilistic measures. We plan to measure whether, and to what extent, the visualization of these probabilistic measures helps them to effectively complete their visual analysis tasks.

5. CONCLUSIONS

In this work, we presented preliminary experiments that demonstrate the usefulness (and need) of progressive answering of similarity search queries on very large time series data. Our findings show that the greatest cost is not locating the 1-NN, but rather waiting for the algorithm to confirm that there is no better answer and finish execution.

We argue that providing progressive answers and estimates of probabilistic distance bounds to users, and letting them decide when to stop the search process, are important research questions. This would eliminate wasted time and reduce user waiting times, in cases where improvement in the final answer is not possible. We have identified two main challenges: (i) how to compute efficiently distance probability distributions for large data series collections; and (ii) how to communicate to users probabilistic distance bounds and errors. Given the increasing popularity of data series analysis tasks, these research directions are both relevant and important, offering exciting research opportunities.

6. REFERENCES
