FAST PATTERN MATCHING IN STREAM TIME SERIES DATA FOR ELECTRICAL APPLICATIONS

K.Manikandan[1] , Er.S.Bhuvaneswari[2]

[1]M.Tech Student, Dr.Mgr Educational and Research Institute University, Chennai,Tamilnadu,India.

[2]Assistant Professor, Dr.Mgr Educational and Research Institute University,Chennai,Tamilnadu,India.

Abstract— *Wide usage in many applications, such as sweep frequency response analysis, Coal mine surveillance, Thermal power stations (used in control and and instrumentation) and multimedia data retrieval. The original task is to find the stream time series similar to a pattern (query) time-series data. In this work, we address the problem of matching both static and dynamic patterns over stream time-series data. we propose a novel multi step filtering mechanism, over the time series data representation. This mechanism can greatly prune the search space and thus offer fast response.*

Key words —Similarity pattern match, stream time series, multi scale segment median of Image approximation.

1. INTRODUCTION

Stream time-series data management has become a hot research topic due to its wide range of applications such as sweep frequency response analysis, Coal mine surveillance, Thermal power stations (used in control and instrumentation) and multimedia data retrieval which require continuously monitoring stream time series .Compared to achieved data, stream time series have their own characteristic:

1. Data are frequently updated in stream time series.
2. Due to the frequent updates, it is very difficult to store all the data in memory or on disk, thus, data summarization and one pass algorithms are usually required to achieve a fast time response.

In this work we propose a novel approach to efficiently perform the pattern matching over stream time series.In order to save the computational cost and offer a fast response, we present a novel multi scale representation for time series, namely multi scale segment median of image (MSMI).

Most importantly, we propose a multi step filtering approach, with respect to the MSMI representation, to prune false candidates before computing the real distances between patterns and stream time series.

2. RELATED WORK:

This section overviews previous work on similarity search over archived time-series data and monitoring over stream time series.

In the research literature, many approaches have been proposed for the similarity search on the archived time series. The pioneering work by Agrawal[4] proposed the whole matching, which finds data sequences of the same length that are similar to a query sequence. Later, Faloutsos[5] extended this work to allow the subsequence matching, which finds subsequences in the archived time series that are similar to a given query series. In these two works, Euclidean distance is used to measure the similarity between (sub) sequences. In order to perform an efficient similarity search, the GEMINI framework [5] is proposed to index time series and answer similarity queries without false dismissals. Since the dimensionality of time series are usually high (e.g., 1,024), the similarity search over high-dimensional index usually encounters a serious problem, known as the "curse of dimensionality." That is, the query performance of the similarity search over indexes degrades dramatically with the increasing dimensionality. In order to break such curse, various dimensionality reduction techniques have been proposed to reduce the dimensionality of time series before indexing them. But only DFT and DWT have been used in the scenario of stream time series, however, with the limitation that only one pattern is considered. **Similarity measures.** In addition to Euclidean distance (L2-norm)[4] [5] several other distance functions have been proposed to measure the similarity between two time series in different applications such as Dynamic Time Warping, longest Common Subsequence

, and Edit Distance with Real Penalty Specifically, Euclidean distance requires the time series to have the same length, which may restrict its applications. On the other hand, DTW can handle sequences with different lengths and local time shifting; however, it does not follow the triangle inequality, which is one of the most important properties of a metric distance function. A recent work makes DTW index able by approximating time series with bounding envelopes. The resulting R-tree index[6] is clearly inefficient for query processing on stream time series, in terms of both update and search cost. ERP can support local time shifting and is a metric distance function. LCSS[7] is proposed to handle noise in data; however, it ignores various gaps in between similar subsequences, which leads to inaccuracy. In contrast, our work focuses on Lp-norm, which covers a wide range of applications. Formally, the Lp-norm distance between two series X(X[0],X[1],....X[n-1]) and Y(Y[0],Y[1],...,Y[n-1]) of length n is defined as

$$Lp(X,Y)=\sqrt{(\sum_{i=0}^{n-1}|X[i]-Y[i]^{p})}$$

where p>= 1. Note that L1-norm is also called Manhattan distance, whereas L2-norm is Euclidean distance.

2.1 Monitoring in Stream Time Series:

Not much previous work has been published for monitoring stream time-series data. Zhu and Shasha[9] proposed a method to monitor the correlation among any pair of stream time series within a sliding window, in which DFT was used as a summary of the data. Later, they introduced a shift wavelet tree (SWT) based on DWT to monitor bursts over stream time-series data. Bulut and Singh[8] improved the technique by using multi scale DWT trees to represent data. Gao and Wang[3] proposed a prediction model to save the computational cost during the matching between a single stream time series and multiple static patterns. Recently, Papadimitriou[10] proposed a method for capturing correlations among multiple stream time-series data with the help of an incremental PCA computation method. With respect to data estimation, estimated the current values of a co-evolving time series through a multivariate linear regression. These approaches, however, are different from our similarity match problem, in the sense that they do not assume any patterns available. Instead, they aim at detecting either patterns in a stream time series or changes in correlation pattern over multiple stream time series. Moreover, Wu[11] proposed an online matching algorithm for detecting subsequences of financial data over a dynamic database. However, their segmentation and pruning methods are designed for financial data only and cannot be applied to detect general patterns in stream time series.

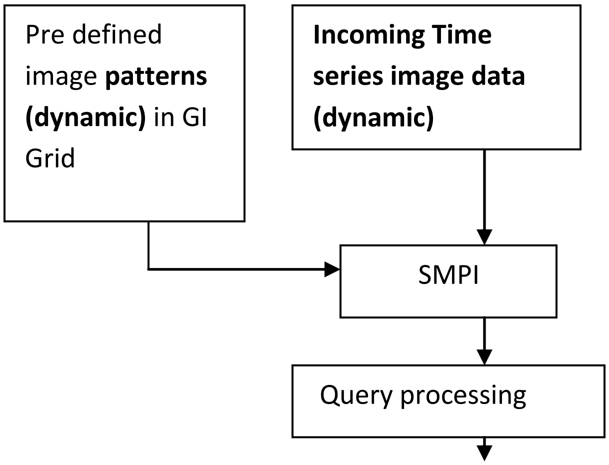
3. OVERVIEW OF THE APPROACH

A stream time series is an ordered sequence, $S = (s_1, s_2 . . . s_i, . . . ; s_t)$ where each s_i is a real value arriving at a specific time i, and index t is the current timestamp. In this work, we focus on the sliding window model due to its popularity and generality. A sliding window is denoted as

$W_i = (s_i,s_{i+1},....S_{i+w-1})$ where

i = 1 . . . (t _ w + 1) and w is the predefined window size.

We have a set of predefined time-series patterns, P = {p1,p2,....pm}, where the length of each pattern pj is equal to w. We formally define the pattern matching problem over stream time series below. Specifically, we want to retrieve the similar match between every window Wi, for i = {1 . . . ,t-w+1}, in stream time series S and patterns pj € P. Here, a sliding window Wi is similar to a pattern pj if dist(wi,pj)<=ε where dist is an Lp-norm distance function . Moreover, ε is specified by users depending on different applications (or the length, w, of patterns).


Pre defined image patterns (dynamic) in GI Grid
Incoming Time series image data (dynamic)
SMPI
Query processing

A set of matching pair (Wi,pt)

Fig 1 Illustration of pattern matching

The pattern matching between dynamic patterns stored in grid index GI and incoming time series image data is performed in an efficient manner using SMPI and query processing algorithms. Fig 1 illustrates the above pattern matching technique.

4.MULTI SCALE SEGMENT MEDIAN OF IMAGE REPRESENTATION :

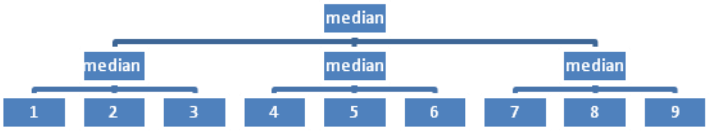


Fig 2 Illustration of the MSMI approximation

In this section, we propose a novel multi scale segment median approximation of time series, which is based on the segment median. Specifically, suppose we have a time series W of length w, where $w = 3^l$ for a nonnegative l (note: here we assume the length w of W is a multiple of 3; otherwise, a sequence of 0 can be appended). We construct multi scale segment median approximation of W by computing the segment median representations from (the coarsest) level 1 to (the finest) level l. In particular, on each level j, there are totally 3^{j-1} disjoint segments of equal size 3^{l-j+1}, each of which is represented by the median of all values within itself. Fig. 2 illustrates a simple example of MSMI representation of a time series W with length w =9 and thus l = 2. On the approximation level 2, there are in total three segments, each of which contains the median of three values within itself.

5.FILTERING METHOD

Next, we present an effective pruning algorithm, namely SMPI to check whether or not a sliding window W of size w ($=2^l$) matches with any pattern pt in a pattern set P. Specifically, since the size of the pattern set P can be large, it is not efficient to compare every pattern pt in P with W during the matching process. Thus, we propose a d-dimensional grid index GI to store patterns, where d ($= 2^{l_{min}-1}$) is typically low (e.g., 1 or 2) to provide low space and accessing costs. In particular, we split the data space into cells with equal-sized bins along each dimension. For each pattern pt , we compute its l_{min}th level MSMI approximation $A_{lmin}(pt)$ and store pt (with pattern identifiers and MSMI approximations) in a cell of GI that contains $A_{lmin}(pt)$. Then, during the stream processing (SMPI),given a sliding window W, we can efficiently access those patterns in cells of GI that are close to $A_{lmin}(W)$.

Multi scale segment median approximations of two time series W and W' on any level j can define a lower bound of their real distance, which can be used to fast pattern matching in stream time series. However, different levels of approximation have different computational cost and pruning power.

Algorithm 1. SMPI

```
Input: a pattern set P, a sliding window W of length w (= $3^l$), a similarity threshold ε, and a d-dimensional grid GI on P
Output: a candidate pattern set P' matching with W.
1. let T contain all ids for patterns in P that may match with W through the grid GI
2. j = $l_{min}$ + 1
3. while (j <= l) and (T <> Ø) and Ej do // Ej is the early stop condition
4. T' ← Ø
5. for each pt € T do
6. if Lp(Aj(pt),Aj(W))<= $ε/2^{l+1-j}$ then
   //SS pruning scheme
7. T'← T'U'{pt} // candidate patterns to //be pruned in next level
8. end if
9. end for
10. T ←T'
11. j = j + 1
12. end while
13. let P' contain all the patterns with ids in T'
14. return P'
```

Algorithm 1 illustrates the details of SMPI, which performs the pattern matching when a new sliding window W from a stream time series is obtained. Basically, we first retrieve those candidate patterns in cells of GI that are within ε distances from $A_{lmin}(W)$, and then filter out false alarms by MSMIs on different levels. Given a pattern set P, a similarity threshold ε, and a d-dimensional grid index GI built on P,SMPI returns a candidate pattern set P0 (subset of P) in which patterns cannot be pruned by our pruning method (i.e., they may match with the sliding window W).Without loss of generality, assume that we have pre calculated MSMI approximations Aj(pt) for all the patterns pt € P on levels j (l_{min} < j <=l), and store them in grid index GI. Each step of Algorithm 1 is given as follows: First, we access grid index GI to obtain all the pattern identifiers in the cells that have their minimum distances from $A_{lmin(}W)$ smaller than or equal to ε (line 1). Then, we initialize a set T containing identifiers of the result (line 1) and set j to l_{min} + 1 (line 2). SMPI prunes patterns in P for sliding window W in a while-statement using MSMIs from level $l_{min+}1$ up to l at most (lines 3-12). The distance computation with higher level MSMI approximations is more costly, however, achieving more pruning power. Thus, we need an early stop condition, denoted as Ej, to decide up to which level we should terminate the pruning procedure (i.e., while statement) such that the total cost can be minimized.

6. SIMILARITY MATCHING

Based on MSMI , Algorithm 2 (Similarity_Match) illustrates the query procedure to retrieve the pattern matches between patterns and a stream time series S.

Algorithm 2 Similarity_Match

```
Input: a time-series stream S, a set P of patterns and ε, and grid index GI on P
Output: a set of matching pairs
1. while a new data item new_d arrives do
2. let Wi be a sliding window with the w most recent values containing new_d
3. P' ← SMPI (P,Wi, ε,GI)
4. for each pt € P' do
5. if Lp(Wi, pt)<= ε, then
6. output the similar pair (Wi, pt)
7. end if
8. end for
9. end while
```

6.1 Discussion:

When a new data item new_d arrives at S, we obtain the latest sliding window Wi that contains this new data (lines 1 and 2). Then, we invoke Algorithm 1, SMP, to find all the candidate patterns (i.e., P') that may potentially match with W_i (line 3). Then, for each candidate pattern pt € P', we calculate the real distance from pt to Wi, and finally output the pair (Wi; pt) if they are similar (lines 4-8).

6.2 Experimental evaluation:

The CPU time of DWT is always greater than that of MSMI under various Lp- norms. Although DWT under L3- and L1-norms performs better than that in the static case, the space for the array is large. It is illustrated in Fig. 3.

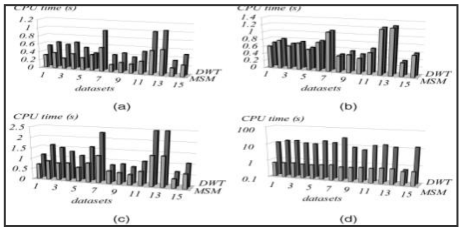


Fig. 3. Comparison between MSMI and DWT (15 stock data sets).(a) L1-norm. (b) L2-norm. (c) L3-norm. (d) L1-norm.

7. CONCLUSIONS

In this paper, we have proposed a novel MSMI representation together with a multi step filtering scheme, which facilitates detecting both static and dynamic patterns over time-series stream efficiently. The MSMI representation can be incrementally computed with a low cost and enables us to prune out false alarms with no false dismissals. Compared to another popular multi scale representation, DWT, we prove that MSMI has similar computational cost to DWT under L2-norm. However, it is an order of magnitude better under other Lp-norms, since MSMI is applicable to arbitrary Lp-norm, whereas it is not possible for DWT unless a very loose lower bound is used. Extensive experiments show that the multi step filtering on MSMI offers an efficient and scalable methodology to detect patterns over stream time series image data. In the future, it would be interesting to apply our MSMI approximation as well as multi step filtering techniques to answer pattern matching queries over batch of data arrived at same time, which is very useful in applications like coal mine surveillance to report dangerous events, sweep frequency response analysis and multimedia data retrieval.

REFERENCES

[1] Xiang Lian, Lei Chen, Jeffrey Xu Yut Guoren Wang Ge Yu, Similarity Match Over High Speed Time-Series Streams IEEE 2007.

[2] Multiscale Representations for Fast Pattern Matching in Stream Time Series Xiang Lian, Student Member, IEEE, Lei Chen, Member, IEEE, Jeffrey Xu Yu, Senior Member, IEEE, Jinsong Han, Member, IEEE, and Jian Ma.

[3] L. Gao and X.S. Wang, "Continually Evaluating Similarity-Based Pattern Queries on a Streaming Time Series," Proc. ACM SIGMOD, 2002.

[4] R. Agrawal, C. Faloutsos, and A.N. Swami, "Efficient Similarity Search in Sequence Databases," Proc. Fourth Int'l Conf. Foundations of Data Organization and Algorithms (FODO), 1993.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proc. ACM SIGMOD, 1994.

[6] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD, 1984.

[7] J.S. Boreczky and L.A. Rowe, "Comparison of Video Shot Boundary Detection Techniques," Proc. Eighth Int'l Symp. Storage and Retrieval for Image and Video Databases, 1996.

[8] A. Bulut and A.K. Singh, "A Unified Framework for Monitoring Data Streams in Real Time," Proc. 21st Int'l Conf. Data Eng. (ICDE), 2005.

[9] Y. Zhu and D. Shasha, "Warping Indexes with Envelope Transforms for Query by Humming," Proc. ACM SIGMOD, 2003.

[10] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming Pattern Discovery in Multiple Time-Series," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB), 2005.

[11] H. Wu, B. Salzberg, and D. Zhang, "Online Event Driven Subsequence Matching over Financial Data Streams," Proc. ACM SIGMOD, 2004.

[12] "Time series analysis" www.dreg.com.