Time Series Analysis Based on Enhanced NLCS

Dacheng Nie
Department of Software
University of Electronic Science and Technology of China
Chengdu, China
e-mail: niedc@163.com

Yan Fu
Department of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: fuyan@uestc.edu.cn

Junlin Zhou
Department of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: jlzhou@uestc.edu.cn

Yuke Fang
Department of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: fangyuke@uestc.edu.cn

Hu Xia
Department of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
e-mail: xiahu@uestc.edu.cn

***Abstract*—Similarity analysis plays a key role in clustering of time series. Normalized longest common subsequence (NLCS) is a similarity measurement widely used in comparing character sequences. In this paper, we developed the NLCS and present a novel algorithm to precisely calculate the similarity of time series. The algorithm used the sum of all common subsequence instead of longest common subsequence which can not represent the similarity of sequences accurately. The experiments based on synthetic and real-life datasets shown that the proposed algorithm performed better in comparing the similarity of time series. Comparing with Euclidean distance on four cluster validity indices, the results lead to a better performance by k-means or self-organize map.**

Keywords- Time series analysis; Normalized longest common subsequence; Clustering

I. INTRODUCTION

The longest common subsequence [1] has important applications in the field of data mining, such as speech recognition, anomaly detection [2] and biological sequences [3], they are all using longest common subsequence algorithm. But the normalized longest common subsequences neglect the influence of other common subsequences and lead to depressed measurement of similarity.

To obtain precisely similarity of sequence, this paper perfect the distance of sequences which represent the similarity of sequences. And use the sum of all common subsequences to express the distance of sequences. For continuous time series analysis, we transform them into symbol sequences, and then use the longest common subsequence to analyze. Thus we can use the NLCS to calculate similarity of the subsequences and adopt the new distance of sequences in the cluster method.

In the subsequent part of this paper we will discuss the NLCS algorithm and our improved algorithm of NLCS. Comparisons and experiments show that our algorithm has more precise than the NLCS.

We evaluate our algorithm with unsupervised clustering quality measurements. The clustering quality measure is divided into external validity and internal validity. The external validity indices were employed when true class labels are known and the internal validity indices were employed when true class labels are unknown. In our experiments, Silhouette, Davies-Bouldin, Homogeneity and Separation indices were employed since all the true labels were unknown to us.

II. METHODOLOGY

To analyze the similarity of time series, we need to preprocess the time series, and then calculate the similarity of sequence. In this paper, we symbolized the time series if it is not a character string. After processing symmetrical, the time series data is transformed character string, and then use the sum of normalized common subsequence to calculate the distance and the similarity of sequence. Clustering the character string, and finding groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups.

A Symbolization

We adopt the method that rule-based fixed window symbolize the time series [4]. We can fragment the time series' shaft and value, and homogeneity fragmenting or according to different needs fragment difference, and then assign a character in terms of different time slot's value. Thus we can obtain a character string to express the time series.

B Sum of the common sequence

The normalized longest common subsequence distance was used for comparing sequences similarity. Assume we have two sequences denoted as X and Y, the length of sequence is m and n respectively. The NLCS [2] is calculated as following:

$$NLCS = \frac{length(LCS)}{\sqrt{m \cdot n}} \quad (1)$$

For long time series, longest common subsequence only indicates a small fraction of information to describe the similarity. To precisely present the distance we employed normalized longest common subsequence on the subsequence iteratively after removing the longest common subsequence. Then, we calculate the sum normalized longest common subsequence (sNLCS) by the formula:

$$sNLCS = \frac{\sum length(LCS)}{\sqrt{m \cdot n}} \quad (2)$$

Given two sequences X and Y, we find every sequential common subsequence and sum of their length. Using this method we could express the similarity of sequences perfectly.

Algorithm 2.1: sNLCS algorithm.

Input: The time series data sets X and Y.

Output: The similarity of two sequences.

Step 1: Transform the time series X and Y into character string Z and W.

Step 2: Calculate the length of sequence Z and W. For each character strings in sequences Z and W, find the longest common subsequence. The parameter *m* and *n* represent the length of character string Z and W. The matrix ***numb*** results stored of the compared character. And matrix ***num*** is used to results stored every diagonal climbing element of matrix ***numb*** added 1 except the first line and row. Thus, we can find the maximum form matrix ***num*** easily. The ***Maxvalue*** represent the maximum of matrix ***num***.

```
for i = 1 to m
      for j = 1 to n
            if Z(i) = = W(j)
                  numb(j, i) = 1
            end
            if Z(i) = = W(j) but i != 1, j != 1
                  num(j, i) = num(j-1, i-1) + 1
            else
                  num(j, i) = 1
            end
   end
end
f = Maxvalue
```

Step 3: Sum the other longest common subsequence.

```
Repeat
      numb(x – Maxvalue + 1 to x, 1 to n) = 0
      numb(1 to m, y – Maxvalue + 1 to y) = 0
      numb(x + 1 to n, 1 to y – Maxvalue + 1) = 0
      numb(1 to x – Maxvalue + 1, y+1 to m) = 0
      num = numb
for i1 = 2 to m
      for j1 = 2 to n
            if num(j1, i1) != 0 and num(j1 – 1, i1 - 1) != 0
            num(j1, i1) = num(j1 – 1,i1 - 1) +1;
            end
   end
end
   Find the maximum value Maxvalue and position x and
y of new matrix num.
if Maxvalue != 0
f = f + Maxvalue
   Until the maximum value equals zero of matrix num or
reach the iteration.
sNLCS = f / sqrt()
```

Thus, we can find the maximum value *Maxvalue* and position x and y of matrix ***num***. If the two sequences have same two or more longest common subsequences, we use the first longest common subsequence.

And we can according to our different needs choose iteration or calculate the matrix ***num*** until all the elements of matrix num are zero. Not only increase the precision of similarity, but also reduce the efficiency unobvious.

The normalized longest common subsequence algorithm use $O(n^2)$ times to calculate the similarity of time series, and our algorithm use $O(kn^2)$ times to compute the similarity of time series, the value k represent iteration that calculate the maximum of matrix num times. But we spend a little more time improving the precision of our algorithm.

III. EXPERIMENTS

Our experiments are performed on several synthetic and real datasets. The real datasets came from Google Trends. With Google Trends, you can compare the world's interest in your favorite topics. Google Trends analyzes a portion of Google web searches to compute how many searches have been done for the terms you enter. The experiments have been performed according to the following steps:

A Experiments with symbolization

For real life dataset, we selected 157 pop songs then searched them with the title of them and name of the related singers in Google Trends. The Google Trends dataset is form Jan.2004 to Jan.2009, in total 255 weeks. And the time unit is one week. We use time sample interval is 5, and the time series' interval of value is 20. And then we transformed the time series into symbol sequences. For example, we choose 4 time series (Figure 1) to explain our method. Figure 2 presents character strings from Figure 1.

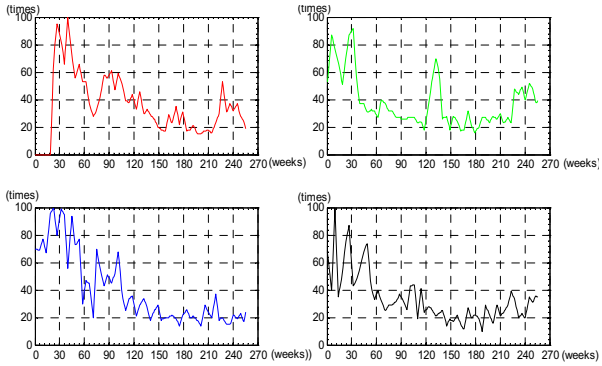

Figure1. Time series curves on real dataset

```
S=
aaaaeededcdccbbccccccbbbbbbbbaabbbbbaaaaaaabcbbbbba
dddcdedbbbbbbbbbbbbbbbbbcddbbbbbabbabbbbbbbbbccbccb
ddeeeeeceddbcbdccccdcbbbbbbabbaabaabbaaabbbbaaabab
cebcedccddbbbbbbbbbbcabbbbbbaaaaabaabaabbbbbbbabbb
```

Figure2. Transformed time series into character strings

B Experiments with compared the enhance NLCS and NLCS.

We use several synthetic datasets in order to explain our modification is perfect to calculate the similarity precisely of the sequence.

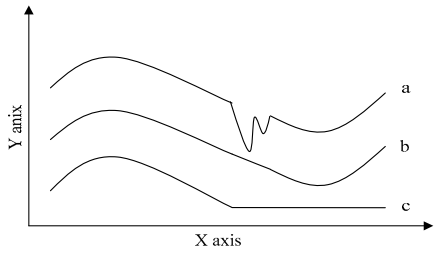

Figure3. Time series on several synthetic dataset

From the three series we can see directly that they have the same longest common fragment, but the most similarity pair is curve 'a' and 'b'. This is why we consider the other common subsequence. Assume that we have three character strings to express the time series from Figure 3:

s=
abcdefabcedcba
abcdefcbaedcba
abcdefgggggggg

Figure4. The character string expressed Figure 3

We use synthetic datasets character array s from Figure 3, and calculate the similarity of them.

TABLEI. THE SIMILARITY OF USE NLCS ALGORITHM

curve	a	b	c
a	1	0.4286	0.4286
b	0.4286	1	0.4286
c	0.4286	0.4286	1

TABLEII. THE SIMILARITY OF USE MODIFY NLCS ALGORITHM

curve	a	b	c
a	1	0.8571	0.4286
b	0.8571	1	0.4286
c	0.4286	0.4286	1

Comparing the Table 1 and Table 2, we find that if we use the normalized longest common subsequence to calculate the similarity, the three sequences will have the same similarity. Using the algorithm NLCS calculate the similarity is 0.4286 among the three sequences, but using the sum of the normalized common subsequence calculate the similarity between the first and second character string is 0.8571, and the first and third character string is 0.4286, the second and third character string is 0.4286. From this value we can see that the first and second character strings are the most similar, and in fact, it is true. It is declared that the modify NLCS have more precise similarity than NLCS.

We use the real datasets in order to explain our experiments' availability, and we use real datasets that transformed the time series into character array s from Figure 1. The similarity among character array s is:

TABLEIII. THE SIMILARITY BASED ON NLCS ALGORITHM

curve	s1	s2	s3	s4
s1	1	0.1569	0.1373	0.1961
s2	0.1569	1	0.1765	0.2157
s3	0.1373	0.1765	1	0.1765
s4	0.1961	0.2157	0.1765	1

TABLEIV. THE SIMILARITY BASED ON IMPROVED NLCS ALGORITHM

curve	s1	s2	s3	s4
s1	1	0.4510	0.6078	0.6471
s2	0.4510	1	0.4902	0.6863
s3	0.6078	0.4902	1	0.3529
s4	0.6471	0.6863	0.3529	1

Compared the Table 3 and Table 4, we can get a result that our modification of normalized longest common subsequence have more precisely similarity than NLCS.

C Clustering quality evaluation

The Silhouettes S(i) [5] is computed as follows:

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{3}$$

Where a (i) is maximum of average dissimilarity of i-object to all other objects in the same cluster; b (i) is the minimum of average dissimilarity of i-object to all objects in other cluster.

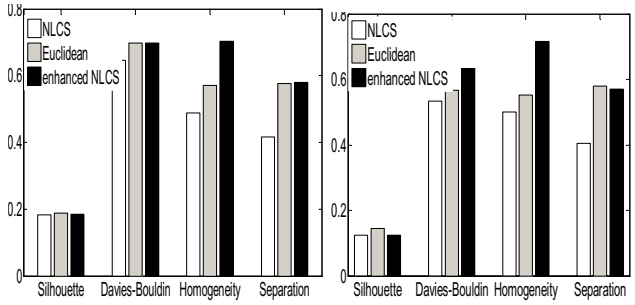

Figure5. Clustering validity by k-means and SOM

Davies-Bouldin Validity Index [6] is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. We calculate the DB by the formula:

$$DB=\frac{1}{n}\sum_{i=1}^{n}\max_{i\neq j}\left\{\frac{s_n(Q_i)+s_n(Q_j)}{S(Q_i,Q_j)}\right\} \tag{4}$$

Where n is number of cluster, s_n is average distance of all objects from the cluster to their cluster centre, $S(Q_i,Q_j)$ is

distance between clusters centers. Hence the ratio is small if the clusters are compact and far from each other.

The cluster separation [7] of a clustering system's output is defined by:

$$Sep = \frac{1}{c(c-1)} \sum_{i=1}^{c} \sum_{j=1, j \neq i}^{c} \exp\left(- \frac{d^2(x_{c_i}, x_{c_j})}{2\sigma^2} \right) \quad (5)$$

Where σ is a Gaussian constant, C is the number of clusters, x_{ci} is the centroid of the cluster C_i and $d(x_{ci}, x_{cj})$ is the distance between the centroid of C_i and the centroid of C_j.

From the Figure 5, compared the evaluation validity indices of clustering quality, we can find our proposed method is better than using NLCS and Euclidean. Bigger value is better.

IV. Conclusion

Similarity analysis of time series is an important technology in data mining, and transforming the time series into character strings can help us analyze the similarity of time series. We enhanced NLCS algorithm by using sum of lengths of all common subsequences to calculate the similarity of sequences. Experimental results on synthetic and real-life datasets indicate that our algorithm achieves better precise similarity in time series analysis. The future work will be focus on improving the algorithm's efficiency.

V. Acknowledgements

This work was supported by the National Science Foundation of China under Grant No. 60973120 and No. 60903073, by the National High-Tech Research and Development Plan of China under Grant No. 2007AA01Z440, by Research and Development Project on Application Technology of Sichuan under Grant No. 2008GZ0009.

References

[1] M. Paterson and V. Dancik. Longest common subsequence. In Mathematical Foundations of Computer Science, 19th International Symposium, volume 841 of LNCS. Pages 127-142, 1994.

[2] Budalakoti, S., Srivastava, A., Akella, R., and Turkov, E. Anomaly detection in large sets of high-dimensional symbol sequence Tech. Rep. NASA TM-2006-214553, NASA Ames.

[3] S. Bereg and B. Zhu. RNA multiple structural alignment with longest common subsequences. Proc. 11th Intl. Ann. Comput. and Combinatorics (COCOON'05), LNCS 3595, pp .32-41,2005

[4] G. Das, K.I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In proc. of the 4th International Conference of Knowledge Discovery and Data Mining, pages 16-22. AAAI Press,1998

[5] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. 1987. Journal of Computational and Applied Mathematics. 20. 53-65.

[6] D.L. Davies, D.W. Bouldin. A cluster separation measure. 1979. IEEE Trans. Pattern Anal. Machine Intell. 1 (4). 224-227.

[7] J. He, A.-H. Tan, C. L. Tan, and S. Y. Sung. Clustering and Information Retrieval, chapter On Quantitative Evaluation of Clustering Systems, pages105-134. Kluwer, 2003.