Study on Representation of Time Series Based on Subsection Polynomial Fitting

Daqi Li [1, 2] , Junyi Shen [1] , Jianfeng Xie [2]
(1 College of Electronic and Information, Xi'an Jiaotong University, Xi'an Shanxi 710049, China)
(2 Beijing Aerospace Control Center, Beijing 100094, China)
Li_daqi@sina.com

Abstract

This paper studied the representation of time series. The signal and the noise separated by wavelet analysis, the whole data sampling was divided into many continuous intervals, in which the signal was monotone. Each interval was fitted by n-degree polynomial and its eigenvector was made up of the coefficients of the polynomial, its width and Signal Noise Ratio (SNR). The eigenvectors of continuous intervals constituted an eigenvector sequence, which could represent the whole sampling. We analyzed respiratory intensity slice of Chinese astronaut, built the eigenvector sequence and finally found the similar intervals by means of cosine distance of eigenvectors.

Key words: *Time Series, Polynomial Fitting, Eigenvector, Similarity Comparison*

1. Introduction

Time series was a sequence of real members, representing the measurement of a real variable at equal intervals that are widely used in various domains. The recent study on time series focused on Temporal Pattern Identification [1], Dynamic Time Warping Distances for Multivariate Time Series, Prediction of Time Series [2], Mining Partial Periodic Patterns [3], Probabilistic Discovery of Time Series Motifs [4], Novelty Detection, Feature–based Classification of Time-series Data [5], and so on.

The problem of comparability comparison of time series was firstly presented by Agrawal in IBM on 1993，which was described as To a certain time series, to find the most similar ones in a large time series databases. The simplest algorithm was to compare the temporal polynomials of two time series to calculate the degree of comparability. From then on, a number of comparability representations of time series were presented, for example, the DFT by Agrawal[6], the Haar wavelet by Chan et al[7,8], the SVD(Signal Value Decomposition) by Korn[9], the PAA(Piecewise Aggregate Approximation)[10], the APCA(Adaptive Piecewise Constant Approximation), the PLR(Piecewise Linear Representation) by Keogh et al, and so on.

The traditional time series modeling technique was based on holistic modal. In many fields, people were interested in partial characteristic of a certain signal and tried to find the expected or alike pieces of the signal.

In our previous research, we studied a presentation of time series based on partition-linear eigenvector [11], in which the eigenvector included the approximate slope, width, SNR of a monotone interval. To some extend, we could find the difference among vary intervals but could not exactly describe the inherent traits of them. So we presented a presentation of time series based on subsection polynomial fitting. We divided the whole sampling into many continuous monotone intervals. In a certain interval, we fit the sampling with n-degree polynomial. The interval could be represented with a new eigenvector which was made up of the coefficients of the n-degree polynomial, the width and the SNR of the interval. Such eigenvector could exactly describe the shape and the variation trait of the interval. The eigenvectors of continuous intervals composed an eigenvector sequence which could represent the whole sampling. According to such presentation, we could exactly compare one interval with another using cosine distance between the two intervals' eigenvectors.

2. Polynomial Fitting

Theorem 1: If $f(x)$ is continuous in $[a,b]$, for any $\varepsilon > 0$, polynomial $p(x) = \sum_{k=0}^{n} p_k x^k$ exists and satisfies $|f(x) - P(x)| < \varepsilon, x \in [a,b]$.

It denoted that, in $[a,b]$, the polynomial fitting of a certain sampling was existed and fits the data to any precision.

Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)
0-7695-2874-0/07 $25.00 © 2007 IEEE

Theorem 2: If $f(x)\in C[a,b]$, $\prod_n$ is the aggregate of polynomial with the degree is no more than n, then the only polynomial $P_n^*(x)$ exists and satisfies that $\max_{a\le x\le b}|f(x)-P_n^*(x)|$ is minimal.

$$\max_{a\le x\le b}|f(x)-P_n^*(x)|=\min_{P_n\in\prod_n}\max_{a\le x\le b}|f(x)-P_n(x)|.$$

$P_n^*(x)$ is the optimum n-degree polynomial fitting of $f(x)$ in $[a,b]$.

If $f(x)=\sum_{k=0}^{\infty}p_k x^k, x\in[a,b]$, then $S_n(x)=\sum_{k=0}^{n}p_k x^k$ is the n-degree polynomial fitting of $f(x)$.Commonly, for certain sampling $f(x)=\{x_i\}_0^m$, $\{p_i\}_0^n$ satisfies equation (1).

$$\begin{bmatrix} m & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}=\begin{bmatrix} \sum f_i \\ \sum x_i f_i \\ \sum x_i^2 f_i \\ \vdots \\ \sum x_i^n f_i \end{bmatrix} \quad (1)$$

For example, let's fit sampling in Table.1. The fitting result $(n=1,2,3)$ was in Table.2. The fitting map $(n=1,2,3)$ was in Figure.1.

Table.1 Sampling

i	0	1	2	3	4	5	6	7	8	9
x_i	0	0.16	0.32	0.48	0.64	0.80	0.96	1.12	1.28	1.44
f_i	0	-0.23	-0.44	-0.54	-0.66	-0.74	-0.79	-0.78	-0.87	-094

Table.2 Fitting result $(n=1,2,3)$

n	p_0	p_1	p_2	p_3
1	-0.1761	-0.5883	———	———
2	-0.0409	-1.2217	0.4399	———
3	0.0061	-1.7593	1.4238	-0.4555

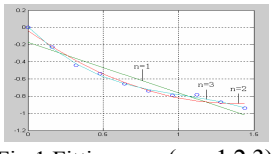

Fig.1 Fitting map $(n=1,2,3)$

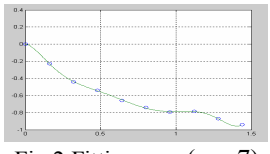

Fig.2 Fitting map $(n=7)$

○ represented the sampling — represented the fitting curve

When $n=7$, $S_7(x)=15.7786x^7-78.5430x^6+153.2819x^5-148.1925x^4+73.2051x^3-16.1010x^2-0.2152x+0.0001$.

The fitting map $(n=7)$ was in Fig.2. $S_7(x)$ fit the sampling accurately.

So far, we faced another problem, which is how to make sure the degree of n.

From above result, we could see that, in a way, the higher was n degree the more exactly $S_n(x)$ fit $f(x)$.

The ε mentioned in theorem 1 could be a method to ascertain n. But unsuitability ε could lead that n was too low and $S_n(x)$ could not describe the variation trait of the sampling, or that n was too high and the computational complexity was too high also.

To get the tidy polynomial $S_n(x)$, we introduced a method as follows.

In theorem 2, $\max_{a\le x\le b}|f(x)-P_n^*(x)|$ was mentioned. When n increased from 1, $\max_{a\le x\le b}|f(x)-P_n^*(x)|$ was evidently degressive. When n reached one certain N, $\max_{a\le x\le b}|f(x)-P_N^*(x)|-\max_{a\le x\le b}|f(x)-P_{N+1}^*(x)|$ is nearly equal to 0. N was regarded as appropriate n for polynomial fitting.

The result of $\delta_i=\max_{a\le x\le b}|f(x)-p_i^*(x)|$ was in Table.3.

Table.3 The result of δ_i

i	1	2	3	4	5	6	7	8
δ_i	0.2558	0.0977	0.0614	0.0522	0.0453	0.0448	0.0445	0.0444

The result of $\Delta\delta_i=\delta_i-\delta_{i+1}$ was in Table.4.

Table.4 The result of $\Delta\delta_i=\delta_i-\delta_{i+1}$

i	1	2	3	4	5	6	7
$\Delta\delta_i$	0.1581	0.0363	0.0092	0.0069	0.0005	0.0003	0.0001

$\Delta\delta_7=0.0001$ showed that δ_7 was close to δ_8. So we regarded 7 was the appropriate n for polynomial fitting.

3. Eigenvector's Calculation and Application

Definition 1: If $f(x)$ was continuous and monotone in $[a,b]$ then $[a,b]$ was defined an interval of $f(x)$.

Definition 2: If $[a,b]$ was an interval of $f(x)$ then $W=b-a$ was defined the width of $[a,b]$.

Definition 3: If $f(x),x\in[a,b]$ was decomposed by wavelet as $f(x)=f'(x)+e(x)$ then $R=\lg\left(\frac{\sum_{t=t_n}^{t_{n+1}}|f'(t)|}{\sum_{t=t_n}^{t_{n+1}}|e(t)|}\right)$

was defined SNR of $[a,b]$.

Definition 4: If $s_n(x)=\sum_{k=0}^{n} p_k x^k, x\in[a,b]$ was the n-degree polynomial fitting for $f(x), x\in[a,b]$ then $V=(p_n,p_{n-1},\cdots,p_0,W_n,R_n)$ was defined the eigenvector of $f(x), x\in[a,b]$.

Definition 5: If $V=\prod_{i=1}^{m} V_i$, V was defined eigenvector sequence of $f(x)$ in m intervals.

Definition 6: The cosine distance between two intervals' eigenvector was defined the similarity degree between the two intervals.

The procedure of eigenvector sequence calculation was in Fig.3.

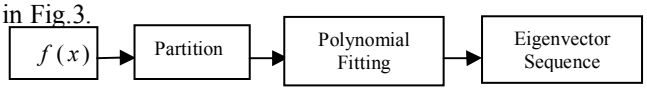

Fig.3 The creating procedure of eigenvector sequence

The calculation of SNR and subsection of the signal, which had been discussed in references [6], was omitted.

In order to implement the comparison of intervals, we had to perform the same n-degree polynomial fitting on each interval. So ascertaining appropriate n was very important. We introduced a fore-fitting procedure: To chose several random intervals and find appropriate n for each intervals, to regard the maximum of those n as the appropriate n of $f(x)$.

To calculate the eigenvector sequence, we introduced two algorithms: PolyFit and VectorCreating.

PolyFit algorithm, described as follows, performed n-degree polynomial-fitting for a single interval.

The PolyFit algorithm:

Input: $[a,b], n, m, \{x_i\}_1^m, \{f(x_i)\}_1^m$

Output: p

step 1: **for** $i=1,2,\cdots,m$

$x_i = x_i - x_0$;

end for

step 2: $AA=\begin{bmatrix} m & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{bmatrix}$;

step 3: $BB = AA'$;

step 4: $DD = BB * AA$;

step 5: $EE = DD^{-1}$;

step 6: $y = \{f(x_i)\}_1^m$;

step 7: **for** $i=1,2,\cdots,m$

$f(x_i) = f(x_i) - f(x_1)$;

end for

step 8: $p = EE * BB * y'$;

step 9: output p;

Stop.

VectorCreating algorithm, described as follows, performed n-degree polynomial-fitting for all intervals.

The VectorCreating algorithm:

Input: $\{f(x_i)\}_1^{m\times M}, \{x_i\}_1^{m\times M}, \{a_i,b_i\}_1^M, \{R_i\}_1^M, \{W_i\}_1^M, \varepsilon, M_1, k_s, k_e$

Output: $\{V_i\}_1^M$

//fore-fitting procedure started

Step 1: **set** $n=k_s$; $\{a'_i, b'_i\}_1^{M_1} \in \{a_i,b_i\}_1^M$;

step 2: **for** $i=1,2,\cdots,M_1$

$p_i = \text{PolyFit}([a'_i, b'_i], n, m, \{x_i\}_1^m, \{f(x_i)\}_1^m)$;

end for

step 3: $n = n+1$;

step 4: **set** $\{mark\}_1^{M_1} = 0$;

step 5: **while** $n < k_e$ **or** $\{mark\}_1^{M_1} = 1$

for $i=1,2,\cdots,M_1$

if $mark_i = 0$

$p'_i = \text{PolyFit}([a'_i, b'_i], n, m, \{x_i\}_1^m, \{f(x_i)\}_1^m)$;

$\Delta\delta_i = \max_{a\le x\le b}|f(x)-P_i(x)| - \max_{a\le x\le b}|f(x)-P'_{i+1}(x)|$;

if $\Delta\delta_i < \varepsilon$ $mark_i = 1$;

end if

end if

$n = n+1$;

$p_i = p'_i$;

end for

end while

//fore-fitting procedure ended

//creating the eigenvector sequence started

step 6: **for** $i=1,2,\cdots,M$

$p_i = \text{PolyFit}([a_i, b_i], n, m, \{x_i\}_1^m, \{f(x_i)\}_1^m)$;

$V_i = (p_i, R_i, W_i)$;

end for

step 7: 输出 $\{V_i\}_1^M$

Stop.

Eigenvector sequence represented the whole sampling. Each eigenvector exactly described the corresponding interval's inherent characteristic. Similarity comparison of intervals was feasible. In this paper, the similarity of interval was measured by eigenvector's cosine distance. The cosine distance between two eigenvectors was a value in $[-1,1]$. 1 means absolute similarity of two intervals and -1 means the contrary. It was no doubt that the distance between an interval and itself was 1.0. The distance threshold had to be preestablished. Furthermore, the continuous intervals' similarity comparison can be measured by the product of cosine distance of continuous eigenvectors.

We applied the algorithms on searching similar segment of astronaut's respiratory intensity. Reference [6] provided the sampling (see in Fig.4) of astronaut's respiratory intensity. Reference [6] also provided the result of the interval and SNR calculated.

In fore-fitting procedure, we ascertained 7-degree polynomial fitting for the whole sampling. The result of eigenvector sequence was listed in Table.5.

The comparison of sampling and 7-degree polynomial fitting was showed in Fig.5. The polynomial exactly fit the signal.

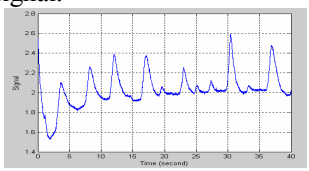

Fig.4 Sampling

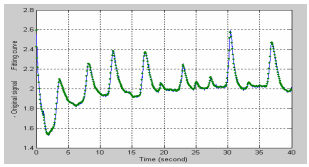

Fig.5 Sampling and fitting

According to the result in Table.5, we could calculate the cosine distance between any two intervals. The partial result among $V_5 \sim V_9$ was listed in Table.6.

The cosine distance between V_6 and V_8 was 0.993279. If we preestablished 0.95 as the threshold of distance then the intervals $[a_6, b_6]$ and $[a_8, b_8]$ were two similar intervals.

The comparison of sampling and 7-degree polynomial fitting in $[a_6, b_6]$ and $[a_8, b_8]$ was showed in Fig.6.

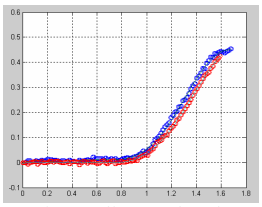

Fig.6 The comparison of sampling and 7-degree polynomial fitting in $[a_6, b_6]$ and $[a_8, b_8]$

Table.6 The result of cosine distance among $V_5 \sim V_9$

	V_5	V_6	V_7	V_8	V_9
V_5	1.000000	0.056865	0.677247	-0.047028	0.242628
V_6	0.056865	1.000000	0.668064	0.993279	0.921864
V_7	0.677247	0.668064	1.000000	0.616168	0.854006
V_8	-0.047028	0.993279	0.616168	1.000000	0.911836
V_9	0.242628	0.921864	0.854006	0.911836	1.000000

4. Conclusion

In this paper, we introduce a presentation of time series based on subsection polynomial fitting. The n-degree polynomial effectively fits the sampling. Any intersected interval has its own eigenvector which is made up of the coefficients of the n-degree polynomial, the width and the SNR of the interval. The eigenvector describes the shape and inherent characteristic of the interval. The eigenvectors of continuous intervals constitute a sequence which denotes the whole sampling. The cosine distance between any two intervals' eigenvector exactly calculates the similarity of the two intervals and helps us find the similar intervals. Analysis on astronaut**'s** respiratory intensity supports our research.

This paper emphasizes the creating and application of eigenvector but primary research on how to ascertain the optimum degree of the polynomial fitting effectively. Our farther study will be focused on this.

References:

[1] Povinelli R J, Feng X, *A New Temporal Pattern Identification Method for Characterization and Prediction of Complex Time Series Events*, IEEE Trans on Knowledge and Data Engineering, 2003, 15(2): 339-352

[2] Povinelli R J, Feng X, *Characterization and Prediction of Welding Droplet Release Using Time Series Data Mining*, Proc of the Int'l Conf on Artificial Neural Networks in Engineering, 2000: 857-862

[3] Walid G. Aref, Mohamed G. Elfeky, Ahmed K. Elmagarmid, *Incremental, Online, and Merge Mining of Partial Periodic Patterns in Time-Series Databases*, IEEE

Transactions on Knowledge and Data Engineering (March 2004)

[4] Bill Chiu, Eamonn Keogh, Stefano Lonardi, *Probabilistic Discovery of Time Series Motifs*, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (August 2003)

[5] Alex Nanopoulos, Rob Alcock, Yannis Manolopoulos, *Feature-based Classification of Time-series Data*, Information processing and technology (January, 2001)

[6] R Agrawal, C Faloutsos, A Swami, *Efficient Similarity Search in Sequence Databases*. In: D Lomet ed. Proceedings of the 4th International Conference of Foundations of Data Organization and algorithms (FODO), 1993: 69-84

[7] K P Chan, A W Fu, *Efficient Time Series Matching by Wavelets*. In: Proceedings of the 15th IEEE International conference on Data Engineering.1999:126-133

[8] Z Struzik, A Siebes, *The Haar Wavelet Transform in The Time Series Similarity Paradigm*. In: Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases.1999: 12-22

[9] P Korn, N Sidiropoulos, C Faloutsos et al, *Fast Nearest-neighbor Search in Medical Image Databases*, In: Proceedings of 22th International Conference on Very Large Data Bases, Bombay, India, 1996:215-226

[10] E Keogh, K Chakrabarti, M Pazzani et al, *Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases*, In: Proceedings of the ACM SIGMOD International Conference on Management of Data, 2001:151-162

[11] Daqi Li，Junyi Shen，*Research on Approach for Sequential Pattern's Representation with Interval's Eigenvector Sequences Extracted by Wavelet*, Computer Science, Vol.9A, 2006:232-235

Table.5 The result of eigenvector sequence of respiratory intensity sampling of Chinese astronaut

$[a_i, b_i]$	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	w_n	R_n
1	−0. 8437	5. 8266	−15. 7488.	21. 2410	−15. 4194	6. 6379	−2. 5267	0. 0445	1.856	3.358219
2	0. 1417	−0. 7240	0. 9429	0. 4188	−1. 3861	0. 7616	−0. 0397	0. 0006	1.760	3.180943
3	0. 0230	−0. 1966	0. 6548	−1. 0826	0. 9582	−0. 4208	−0. 1109	0. 0075	2.592	3.138912
4	0. 1689	−1. 3440	3. 9773	−5. 5084	3. 7774	−1. 2345	0. 2105	−0. 0076	1.920	3.094653
5	−0. 0310	0. 2366	−0. 6526	0. 6646	0. 1743	−0. 7059	0. 1039	0. 0067	2.240	3.198165
6	0. 6260	−3. 7638	8. 4093	−8. 7205	4. 5239	−1. 1750	0. 1418	−0. 0049	1.600	3.199014
7	0. 0159	−0. 1905	0. 9134	−2. 2483	2. 9965	−1. 9716	0. 1996	0. 0204	3.392	3.202145
8	1. 1128	−6. 6614	15. 0438	−16. 1441	8. 7831	−2. 3566	0. 2808	−0. 0040	1.696	3.225291
9	0. 1793	−1. 3370	4. 0368	−6. 36628	5. 66458	−2. 7422	0. 2851	−0. 0028	1.984	3.226417
10	−333. 2941	473. 1421	−302. 6714	124. 0316	−33. 7835	5. 5075	−0. 3403	0. 0004	0.488	3.51081
11	6. 2202	−18. 2280	18. 9831	−7. 7821	0. 5518	0. 3152	−0. 1345	0. 0017	0.920	3.325312
12	1666. 4999	−2807. 2902	1901. 4510.	−661. 4764	124. 8052	−12. 3016	0. 5438	0. 0014	0.488	3.269903
13	−882. 1985	1323. 3498	−758. 3891	209. 4597	−29. 4955	2. 1581	−0. 1177	0. 0014	0.504	3.282602
14	957. 9929	−1716. 6752	1198. 1212	−412. 2435	72. 6842	−6. 0456	0. 1788	0. 0020	0.544	3.154226
15	−34. 9807	122. 2788	−168. 3485	115. 3186	−40. 4658	6. 8635	−0. 4394	0. 0036	0.992	3.253655
16	0. 7073	−4. 3066	10. 7836	−14. 3755	10. 9743	−4. 6127	0. 6590	0. 0007	1.568	3.487642
17	1386. 5408	−2436. 7108	1651. 2552	−540. 1594	86. 7414	−5. 5596	−0. 0032	−0. 0013	0.512	3.307216
18	−11. 9016	46. 2525	−70. 0437	51. 4459	−18. 1654	2. 4154	−0. 0617	0. 0049	1.120	3.266521
19	−1. 1372	2. 40082	−0. 4661	−1. 8809	1. 4557	−0. 3355	0. 0160	0. 0017	1.120	3.314198
20	0. 1664	−1. 3162	4. 2326	−7. 0708	6. 4425	−2. 9645	0. 4340	−0. 0014	2.016	3.255456
21	7. 7341	−19. 4725	12. 1436	5. 2082	−7. 3443	2. 3455	−0. 2382	0. 0032	1.088	2.992995
22	0. 4339	−3. 4207	11. 03767	−18. 72397	17. 69787	−8. 74897	1. 3314	0. 0081	2.080	3.128018
23	81. 5112	−202. 0832	192. 2454	−85. 4601	16. 6778	−0. 7220	−0. 1034	−0. 0003	0.736	3.19672
24	−0. 1586	1. 9266	−5. 4114	6. 0100	−2. 5086	−0. 0657	0. 1937	−0. 0042	1.088	3.17519
25	−1. 0899	18. 3687	−35. 8916	28. 8578	−11. 3902	2. 1718	−0. 1638	0. 0045	0.672	3.288278
26	0. 9332	−3. 0173	3. 4432	−1. 6288	0. 2882	−0. 0146	−0. 0105	0. 0053	0.992	3.070666
27	−127. 7634	3. 69. 1300.	−423. 4180	244. 4972	−74. 3530	12. 0304	−0. 7540	0. 0053	0.832	3.420647
28	0. 0016	−0. 0175	0. 09811	−0. 3692	0. 8649	−0. 9935	0. 1371	0. 0014	2.464	3.23121