Discovering Semantics from Multiple Correlated Time Series Stream

Zhi Qiao[1,2], Guangyan Huang[1], Jing He[1], Peng Zhang[2], Li Guo[2], Jie Cao[3], and Yanchun Zhang[1]

[1] Victoria University, Melbourne, Australia
[2] Institute of Information Engineering, Chinese Academy of Science, Beijing, China
[3] Nanjing University of Finance and Economics, China
zhiqiao.ict@gmail.com, {zhangpeng,guoli}@iie.ac.cn,
{Guangyan.Huang,Jing.He,yanchun.zhang}@vu.edu.au, caojie690929@163.com

Abstract. In this paper, we study a challenging problem of mining data generating rules and state transforming rules (i.e., semantics) underneath multiple correlated time series streams. A novel Correlation field-based Semantics Learning Framework (CfSLF) is proposed to learn the semantic. In the framework, we use Hidden Markov Random Field (HMRF) method to model relationship between latent states and observations in multiple correlated time series to learn data generating rules. The transforming rules are learned from corresponding latent state sequence of multiple time series based on Markov chain character. The reusable semantics learned by CfSLF can be fed into various analysis tools, such as prediction or anomaly detection. Moreover, we present two algorithms based on the semantics, which can later be applied to next-n step prediction and anomaly detection. Experiments on real world data sets demonstrate the efficiency and effectiveness of the proposed method.

Keywords: Semantics, correlated time series streams, prediction, anomaly detection.

1 Introduction

Time series data have emerged in a wide range of applications from almost every domain. Examples include economic index data in stock markets, patient medical observation data, experimental biological data, to name a few. As a result, it is of utmost importance to find inherent semantics from time series data. Take the medical examination for example, doctors often estimate the physical status of a patient by monitoring and collecting multiple correlated time series data from electrocardiograms (ECG), electroencephalograms (EEG), heart beat rate (HR), and blood pressure observations. Obviously, it is very hard to make accurate estimation by only relying on a single time series data (e.g., HR increases does not mean the patient is suffering from severe illness, perhaps she/he just had exercises). Hence, it is necessary to combine all these time series data for estimation.

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 509–520, 2013.

Many methods have been proposed to analyze multiple time series in [3][5][11]. In addition, there are some popular time series models for time series forecasting, such as the Vector Auto-regression (VAR) and Linear-regression (LR) models. These approaches focus on frequent patterns in the time series, which cannot explain observations using the internal dynamics of systems. The dynamics of a system can be considered as a mechanism of system-work as equivalent to semantics. Here, the system is unseen and unknown, which determines observed time series. It can help us to know more about observation generating rules and state transforming rules underneath data by learning the mechanism. A previous work [12] also studies the semantics detection problem from time series data. The difference is that it uses the pattern-based Hidden Markov Model (pHMM) to describe the univariate time series data where a line segmentation method is used to obtain significant segment patterns. In our work, the patterns are irregularly summarized from multiple time series, and we use the Hidden Markov Random Field (MRF) as the solution.

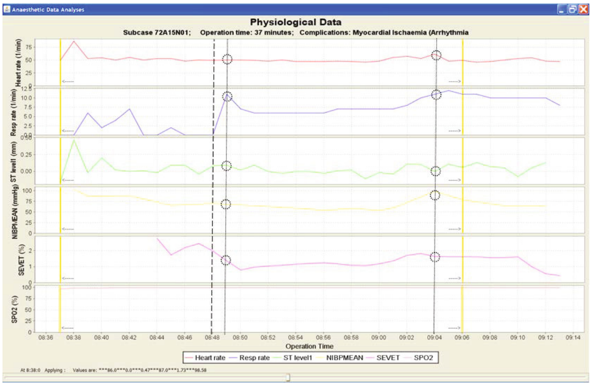

Fig. 1. Real Medical Case (All signals are sampled in a minute unit)

Supposing all time series are observed synchronously at each time point, observations from all time series compose of a tuple at each moment and each tuple is regarded as an output generated by a certain latent state. Each state demonstrates a certain pattern of fluctuation. In this paper, the pattern of fluctuation is taken as a generating rule, conforming with generated observations. Semantics learning basically learns both observation value generating rules and transformation rules among latent states. It is widely admitted that the Hidden Markov Model (HMM) can be used to learn semantics. In HMM, state assignment is mainly determined by aligning observation value production and state transmission. However, a single observation tuple value contains little information. We illustrate this by a real medical example in Figure 1. Figure 1 demonstrates 6 vital body signals of a patient in an operating theatre. At the t_0 time point, the operation starts. Apparently, observation values at t_1 are very similar with observation values at t_2. Actually, they represent different situations of the patient. At t_1, the observation value represents a natural reaction of the patient to an outside emergency which can be regarded as the operation. At t_2, the observation value represents physical state of patient during the operation. If we only consider observation values of one tuple, we cannot obtain significant semantics for a practical problem. Actually, what is different from other common data is that time series data has natural temporal ordering. The temporal ordering of observation data is not directly considered in HMM. According to the temporal ordering of time series, each observation tuple is strongly correlated with the previous tuple. In Figure 1, it is obvious that correlation between t_1 tuple and $t_1 - 1$ tuple is different from correlation between t_2 tuple and $t_2 - 1$ tuple.

Hence, we propose Correlation field-based Semantics Learning Framework (CfSLF) to learn semantics underneath multiple correlated time series in this paper. In the framework, generating rules and state transforming rules will be learned. In order to comprehensively considering both temporal ordering and observation value of tuple, we use hidden Markov Random Field (HMRF) to obtain approximately optimal latent state assignment to learn generating rules. Then, state transforming rules are learned from label sequence in the framework.

The rest of the paper is structured as follows. Section 2 introduces a mathematical description of the problem. Section 3 describes detailed proposed CfSLF. Section 4 introduces two main applications: Observation Value Prediction & Anomaly Detection. Section 5 reports experimental results to show the advantage of our CfSLF model compared with some other algorithms. Section 6 introduces related works. We conclude the paper in Section 7.

2 Problem Setting

In this paper, we propose a Correlation field-based Semantics Learning Framework (CfSLF) to learn latent semantics underlying multiple time series which represent a mechanism of system work. It contains two parts: generating rules learning and state transforming rules learning.

Given multiple time series $X = \{X_1, ..., X_n\}$, $X_i = \{x_{i1}, ..., x_{im}\}$. X_i represents an observed tuple from the m time series at the i^{th} moment and x_{ij} represents observed data from the j_{th} time series at i_{th} moment. Assume that there exists a state set $S = \{s_1,..., s_k\}$. Each tuple is produced by one state of S. $Z = \{z_1, z_2, ..., z_n\}$ is a latent label set. z_i is a label variable, which is discrete and represents latent state of the i_{th} tuple. The value of the variable is in the range from 1 to k. Assume z_i=j, which represents that the latent state of the i_{th} tuple is the j_{th} state s_j. In order to learn data generating rules, we need to obtain optimal latent state assignment and the corresponding generating rules to maximize production probability of observation tuples. The problem can be described as follow:

$$\hat{Z} = argmax_Z P(X, Z) = argmax_Z P(X|Z)P(Z) \tag{1}$$

which is maximized to obtain state assignment. According to the continuity of tuples sequence, the label variable set constituted of finite states can be seen as a latent state sequence. After we obtain optimal state assignment, transforming rules among latent states are learned from the state sequence based on Markov chain characteristics.

3 CfSLF Learning

In this section, we discuss how to learn semantics by our proposed CfSLF.

3.1 Construction of Dependence Relationship

In time series, temporal ordering can be seen as the relationship between the current and previous tuples. In this paper, we simply consider temporal ordering as changing the trend to represent the relationship. We introduce a definition of a local trend as follow,

Definition 1. *Local Trend: In time series, we consider the direction and volume of changing from the last node to the current node as the local trend along time axis.*

Suppose that the current tuple is x_t with observation value o_t. We directly obtain local trend of by $T_{d_t} = o_t - o_{t-1}$. For the local trend of the tuple, we apply the attribute to represent temporal ordering of the current tuple. Then, we simply use the cosine distance to measure the similarity among local trends of different tuples. Intuitively, continuity of time series can be considered as integrating temporal ordering of all tuples. We use the local trend of tuple to represent temporal ordering, which can be seen as an independent attribute of each tuple. In doing so, each tuple has two attributes: its observation value and its local trend. For each tuple, the observation value represents its individual character and local trend represents sequence character. As a result, the tuple series with size n is divided into n independent observation tuples

3.2 Latent State Assignment

In the procedure of state assignment, tuples with similar local trends and observation values are more likely to have the same latent state. Therefore, we use HMRF to learn the rules. A correlation field is built based on local trend similarity. In the correlation field, the assignment of labels depends on corresponding brotherhood set. In this paper, the similarity matrix is considered as a correlation network describing dependency relationship among latent labels of tuples in the correlation field. In the network, similarity is seen as a weight of dependency relationship.

Suppose that we have a correlation network denoted as M. Here, M is the symmetric $n \times n$ matrix, where w_{ij} is the link weight between labels z_i and z_j.

The links in M induce dependence relationships among latent labels, with the rationale that if the link weight is higher between labels z_i and z_j, then they are more likely to have the same value equivalent to the same state.

We define a brotherhood set of the i_{th} label as a label set consisting of labels which have the same latent state value, B_i=$\{z_j$,$i \neq j$ & $z_i = z_j\}$. The random field defined over hidden label variable Z is a Markov random field, where the Markov property is satisfied by $p(z_i|z_{B_i})$. It indicates the probability of z_i depending on z_i's brotherhood set. By introducing the HMRF model, latent labels of tuples are mapped to a correlation field, where assignment of labels depends on a corresponding brotherhood set without considering tuple value.

Because the observation value of tuple is generated by the corresponding state, it is irrelevant to other states and only depends on its latent state. Thus, the values of tuples are conditional independent given their labels.

$$P(X|Z) = \prod_{i=1}^{n} p(x_i = o_i|z_i) \tag{2}$$

We assume that the observation value of the i_{th} tuple generated by the k_{th} latent state is characterized by a set of parameters θ_k, i.e., as we mainly consider multiple time series as real data, we propose to model observation data by Gaussian distribution, because of its flexibility in approximating a wide range of continuous distributions. Therefore, we use the parameter mean vector μ and variance matrix $\sum$ to describe the kth latent state, $\theta_k = (\mu_k, \Sigma_k)$.

We first assume model parameters λ=$\{$ Θ_i, i from 1 to k$\}$ are known a prior. In order to obtain approximately optimal assignment of latent variables for each observation tuple, we transform to find the optimal configuration that maximizes the posterior distribution given λ.

As discussed in Eq. (1), the probability distribution of Z is given by $P(Z) = exp(\gamma \sum \omega_{ij} \delta(z_i - z_j))/H$.

In the above equation, H is a constant value, which can be neglected. We use the Iterated Conditional Modes (ICM) algorithm [5] to estimate the maximum a posteriori probability (MAP). The greedy algorithm can be used by calculating local minimization iteratively, which converges after a few iterations. The basic idea is to sequentially update the label of each object, keeping the labels of the other objects fixed. At each step, the algorithm updates z_i given x_i and the label by maximizing the conditional posterior probability, $p(z_i|x_i = o_i, Z_{I-\{i\}})$.

$$p(z_i|x_i = o_i, Z_{I-\{i\}}) = p(x_i|z_i = s) \times \frac{exp(\gamma \sum_{z_i=s} \omega_{ij} \delta(z_i - z_j))}{H_2} \tag{3}$$

Actually, H_2 can be considered as a constant variable. In doing so, we take the logarithm of the posterior probability, and transform the MAP estimation problem into the minimization of the conditional posterior energy function as shown in the following equation,

$$U_i(k) = -ln(p(x_i|z_i = k)) - \gamma \sum_{j \in B_i} \omega_{ij} \delta(z_i - z_j) \tag{4}$$

where γ is a predefined parameter that represents the importance of the temporal ordering correlation. $\gamma > 0$ represents the confidence of the temporal ordering correlation network. To minimize $U_i(k)$, we find the latent state k of the tuple i by $k = argmax_k U_i(k)$.

3.3 Parameter Estimation

In this part, we consider the problem of estimating unknown λ in order to iteratively learn optimal state assignment. λ describes the pattern conformed with the time stamp that x is generated. We first seek to find λ to maximize $P(X|\lambda)$, which can be considered as the maximal likelihood estimation for λ. However, since both the hidden label and the parameter are unknown and inter-dependent, it is intractable to directly maximize the data likelihood. We view it as an "incomplete-data" estimation problem, and use the Expectation-Maximization (EM) algorithm as the solution.

The basic procedure is as follows. We start with an initial estimate λ_0. Assume that there exist k latent states, where λ_0 is obtained by a simple K-Means algorithm. In the E-step, we calculate the conditional expectation $Q(\Theta|\Theta(t))$,

$$Q(\lambda|\lambda_t) = ElnP(X, Z|\lambda_t) = \sum_Z \{P(Z|X, \lambda_t) \times ln(X, Z|\lambda_t)\} \tag{5}$$

Next, in the M-step, we find λ_{t+1} by computing the derivation of the maximizing function $Q(\lambda|\lambda_t)$.

$$\mu_j^{t+1} = \frac{\sum_Z \sum_{i=1}^n \{p(z_i = s_j)|x_i, \lambda_t \times x_i\}}{\sum_Z \sum_{i=1}^n \{p(z_i = s_j|x_i, \lambda_t)} \tag{6}$$

$$\Sigma_j^{t+1} = \frac{\sum_Z \sum_{i=1}^n \{p(z_i = s_j)|x_i, \lambda_t \times (x_i - \mu_j^{t+1})(x_i - \mu_j^{t+1})^T\}}{\sum_Z \sum_{i=1}^n \{p(z_i = s_j|x_i, \lambda_t)} \tag{7}$$

In each iteration, we have obtained the optimal latent state assignment from the last step. Assume the latent state of tuple i is state j, given λ_t. Thus, $p(z_i|x_i, \lambda_t)$ is 1 when z_i is at state j, else 0. As a result, the E-step and M-step can be recursively computed until $Q(\lambda|\lambda_t)$ converges to a local optimal solution.

3.4 State Transforming Learning

After we obtain the optimal latent state assignment, each observation tuple is assigned a label corresponding to the latent state set of tuples. The labels can be seen as a sequence consisting of limit states along the time axis. We then model correlation among states to reveal system dynamics. Here we regard correlation as the transforming probability $p(s_i|s_j)$ representing the probability from the state j to state i. A Markov chain model can be used to approximately estimate the transforming probability among states by using $p(s_j|s_i) = N(s_i s_j)/N(s_i)$, where $N(s_j)$ represents the amount of labels with s_j value in label series and $N(s_j s_i)$ represents the amount of adjacent labels with s_j and s_i values in label series.

4 Applications of the Model

Observation Value Prediction. Time series semantics can be used to make the following value prediction. We first introduce the next 1-step value prediction. Assume that we have learned the semantics from the training data, we then have λ and the state transformation rule. In test step, we consider time series X, as we care about the local trend of tuples, let $x_{t-1}$and x_t are current tuples. Our task is to predict x_{t+1}. We first compute the current latent state of x_t. When we assign a label to the current tuple, we need to predict the label of the next tuple according to the Markov chain characteristics. Assume the current label z_t is s_c, the next label z_{t+1} can be obtained by maximizing $p(z_{t+1} = i|z_t = s_c)$. Additionally, according to time series continuity, we can estimate that the next tuple close to the current tuple with high probability. Therefore, it is safe to say that the next state maintains the continuity with high probability. That is, next state can be predicted by,

$$\hat{Z}_{t+1} = argmax_i p(\dot{x}_{t+1} = x_t) \times p(z_{t+1} = i|z_t = s_c) \tag{8}$$

Because observation tuples produced by a state have similar values and similar trends, we approximately predict observation values of the state by,

$$\hat{x}_{t+1} = argmax_x |x - E_{z_{t+1}}| + |x - x_t| \times |\frac{x - x_t}{|x - x_t|} - \frac{V_{z_{t+1}}}{|V_{z_{t+1}}|}| \tag{9}$$

where $E_{Z_{t+1}}$ represents the Expectation of the observation value of the predicted state, and $V_{Z_{t+1}}$ represents the exception of the local trend of the predicted state. We use the Euclidean distance to measure similarity. By computing the derivative of function, the prediction value x can be obtained. Then we extend next 1-step value prediction to the next n-step value prediction. We iteratively apply the predicted value as the new observation value to forecast the next value until n steps have been performed.

Anomaly Detection. Our proposed model also can be used to detect data anomaly. In time series, there is no apparent and definite label to represent which observation is normal or abnormal. So, it is not a classification problem. Generally speaking, we only know that anomaly occurs in a certain period. Take a finance application for example. The worldwide economical recessions have occurred several times in history. The recession always lasts for a period of time, which is regarded as recession date. It impacts all business activities. Compared with economic affairs in other periods, economical affairs in recessions can be seen as anomaly. Hence, we propose the method based on a semantics model only qualitatively to indirectly reflect anomalies.

According to time series continuity and semantics rules, we know that the current tuple is similar to the last tuple with high probability, and the current state has high transformation probability from the last state. Considering both rules, we can measure the probability with which the current tuple normally is generated by the following equation: $f(x_t) = p(x_t|z_{t-1}) \times p(z_t|z_{t-1})$. A logarithm function is generally used to obtain a degree of the energy. Thus, we compute the

Table 1. Runtime and Accuracy Comparison

N	0.05	0.1	0.2	0.5	0.8	1
H-Runtime	2.223	4.727	5.82	13.413	30.568	49.377
H-Error	0.0259	0.0152	0.0129	0.0113	0.0112	0.0111
V-Runtime	1.915	4.9590	3.785	10.65	19.315	26.024
V-Error	0.115	0.0117	0.0092	0.0078	0.0064	0.0059
T-Runtime	0.845	16.433	25.635	73.180	136.077	180.367
T-Error	0.0389	0.0265	0.0211	0.0169	0.0163	0.0152
H(Humid data) V(Volt data) and T(Temp data)						

anomaly score of the current tuple by $-logf(x_t)$, which measures the significance the current tuple deviates from that emanating from the normal producing rule. Intuitively, high scores indicate anomalous data with high probability.

5 Empirical Evaluation

In this section, we present extensive experiments on real-world multiple time series data to validate performance of our proposed approach. All experiments are conducted on a 3.0GHZ CPU with 2 GB RAM. The experimental environment is windows XP with Matlab.

Benchmark Data. We adopt four multiple time series data as our test-bed: 1) Price data[1] 2) Mote data sets[2] 3) Medical data[3] 4) Financial data[4]

Baseline Methods. As discussed above, our proposed semantics learning method can be applied to value prediction. We compare our method with following value prediction algorithms: (1) Multivariate Autoregression Model. (2) Hidden Markov Model (HMM).

5.1 Experiment Results

We first consider the time cost of model learning and hyper parameter λ sensitivity for data. Then we compare our proposed model with benchmark method on multiple step prediction accuracy. Additionally, our proposed model can be used to indirectly reflect latent anomalies, which are hard to see from the original data. Here we mainly analyze financial data to discover financial mark depression.

[1] The data set consists of the Reference Price Data (RPD) for APX Power UK Spot market, which can be downloaded from Website `http://www.apxendex.com/index.php?id=466`.

[2] Mote data sets are collected using Berkeley Mote sensors, at several different locations in a lab, over a period of a month. For each category of data, we just select four time series.

[3] It consists of 11 medical time series of patient from an Australian Hospital.

[4] It is constituted of 11 economical time series GS1, DTB3, TB3MS, WTB3MS, GS5, GS10, MPRIME, WPRIME, FEDFUNDS, AAA and BAA respectively, which are obtained from the Website of the Federal Reserve Bank of St. Louis.

We use relative error to measure accuracy of prediction. We compute the relative error by $|\hat{x}_i - x_i|/|x_i|$ where $|x_i|$ is the estimated value, and x_i is the real value. Thus, the lower relative error is, the higher the accuracy is.

Time Complexity. Suppose the number of tuples is N in multiple correlated time series. In M-step,the time complexity is O(N). In E-step, the time complexity is O(N_2) because of aggregating the effect of the labels of brotherhood set of v_i to compute $P(Z)$. Actually, semantics rules seen as latent pattern repeatedly exist in the multiple time series. Hence, we do not need to learn a model based on entire training set. We can approximately obtain the semantics from part of the data set. In the experiment, we conduct our proposed model on all of benchmark data set, and compare the runtime and next 1-step value prediction accuracy under different N, the selection ratio of boundary points. For example, assume size of train-ing is 2000, and 0.05 means we choose 0.05*2000 =100 time points to train the model. The results are shown in Table 1. In Table 1, the error is average relative error for applying learned CfSLF to 200 testing tuples sampled from testing set. It can be seen that runtime gets longer and accuracy gets higher when ratio N gets bigger. We can see that in every data set, when the percentage is equal to or larger than a certain value, the accuracy is not affected much.

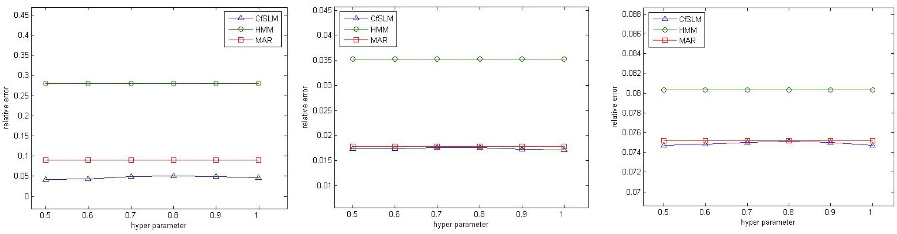

Fig. 2. Hyper parameter λ sensitive

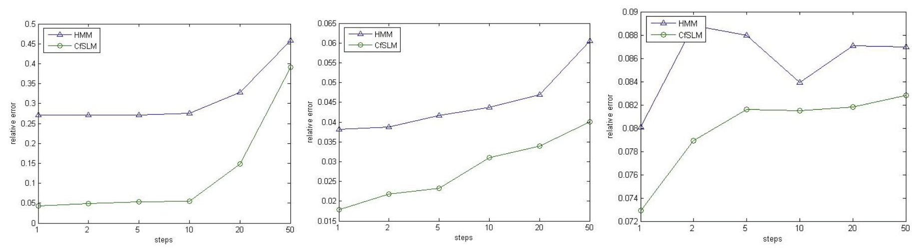

Fig. 3. Next n-step prediction comparison

Hyper Parameter Sensitivity. In our proposed model, parameter λ represents the confidence of the temporal ordering correlation network. Different values of λ determine different effects of the corre-lation field. We separately conduct experiments on all of the data sets to demonstrate hyper parameter λ setting and effect. In each experiment, we vary λ from 0.1 to 2 separately, and compute the corresponding relative error of the next 1-step value prediction under the predefined λ. HMM is used as baseline method to compare with our proposed method. The outcome is shown in Figure 2. It can be seen that there are slight changes in performance when parameters are varied and CfSLF has better performance than the HMM and Multivariate Autoregression models.

Application of the Model. In the former section, we have tested performance on the next 1-step value prediction. In the following, we will discuss additional next n-step value prediction and anomaly detection.

Next n-step Value Prediction. In this experiment, we test the accuracy of CfSLF for the next n-step value prediction. The experiments are conducted for all data sets. We select 10 points randomly. For each selected point, we predict the values after 1, 2, 5, 10, 20, and 50 steps respectively. The CfSLF is compared with a Hidden Markov Model (denoted by HMM). The HMM can be used to learn system-work mechanisms underlying time series. In the experiment, we suppose that each tuple is produced by a latent state and the producing procedure conforms to Gaussian distribution. We first find latent state of the tuple at a selected time point, then make state predictions at 6 future time points. Prediction value is corresponding expectation of predicted state. We use relative error as the measurement. The results are shown in Figure 3. It can be seen that on all data sets, CfSLF is more accurate than HMM.

Anomaly Detection. In the experiment, we use CfSLF on financial data to verify its performance for anomaly detection. The experimental results indicate that the proposed method detected deviations from that emanating from the normal producing rule as anomalies and these corresponded to actual economic events. The degree of anomaly in these time series is shown in Figure 4. In Figure 4, we see that two apparently peak deviated from other scores. Each peak corresponded to big economic events occurring in corresponding month. The first peak appeared on January 2008, where the Federal Reserve lowered its federal funds rate, which impacts how much consumers pay on credit card debt, home equity lines of credit and auto loans, to 3.5 percent from 4.25 percent, which was the biggest rate cut by the Fed since October 1984. The second peak appeared on September 2008, where Lehman Brothers announced its bankruptcy. The second peak indicates that the proposed method detected the depression whictarted in September 2008, as anomalies.

6 Related Works

There are many works on analysing time series, such as summary learning, time series segmentation, forecasting and so on, which have always been popular topics [8][9][11][12][13][14][15][16]. However, they just can be used to analyse single time series. Additionally, pattern learning from time series based on sliding windows has attracted more and more attention [1][2][3][4][5][6][7][17]. However, these methods cannot reveal global system-work rules. In recent years, semantics mining has been always a popular topic. In time series analysis, semantics can

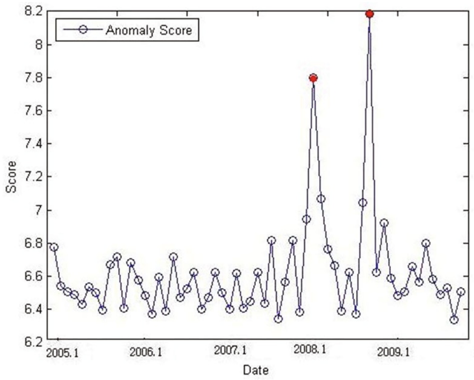

Fig. 4. Anomaly score time series of multiple correlated financial time series data

mainly be seen as system-work mechanisms. While, there are few studies on it. In [12], pHMM is proposed to learn time series semantics. However, it is just used to analyze a single time series. Generally, a Hidden Markov Model can be used to learn the semantics rules. Some other improved methods based on HMM are applied to learn latent system rules [16]. However, in multiple time series, a single observation value contains little information. Compared with these methods, our proposed model introduces local trends to extend information of tuple, and learn semantics from multiple time series based on both observation values and local trend correlation.

7 Conclusions

In this paper, we present a new Correlation field-based Semantics Learning Framework (CfSLF) to model multiple correlated time series. Our model aims to find semantics underneath multiple time series, by detecting data generating rules and transforming rules. Experiments have demonstrated the utility of the proposed method. The contribution of the study is three folder: (1) The Hidden Markov Random Field (HMRF) is used to model the data observations and corresponding states, by which the irregular patterns can be summarized from multiple correlated time series. (2) A value prediction method is presented based on semantics learned by CfSLF. (3) An anomaly detection method is proposed based on data semantics.

Acknowledgments. This research was supported by the ARC Linkage Project (LP100200682) named "Real-time and Self- Adaptive Stream Data Analyser for Intensive Care Management"; Internation S&T Cooperation Program of China No.2011DFA12910; National Science Foundation of China (NSFC) Grant (71072172; 61103229; 61272480; 61003167); the "Strategic Priority Research Program" of the Chinese Academy of Sciences Grant (XDA06030200); 863 Program Grant (2011AA01A103).

References

1. Zhang, C., Weng, N., Chang, J., Zhou, A.: Detecting Abnormal Trend Evolution over Multiple Data Streams. In: Chen, L., Liu, C., Zhang, X., Wang, S., Strasunskas, D., Tomassen, S.L., Rao, J., Li, W.-S., Candan, K.S., Chiu, D.K.W., Zhuang, Y., Ellis, C.A., Kim, K.-H. (eds.) WCMT 2009. LNCS, vol. 5731, pp. 285–296. Springer, Heidelberg (2009)
2. Zhang, P., Gao, B.J., Liu, P., Shi, Y., Guo, L.: A framework for application-driven classification of data streams. Neurocomputing 92, 170–182 (2012)
3. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming Pattern Discovery in Multiple Time-Series. In: Proceedings of VLDB 2005 (2005)
4. Chan, P.K., Mahoney, M.V.: Modeling Multiple Time Series for Anomaly Detection. In: Proceedings of ICDM
5. Hirose, S., Yamanishi, K., Nakata, T., Fujimaki, R.:]Network Anomaly Detection based on Eigen Equation Compression. In: Proceedings of SIGKDD 2009 (2009)
6. Qiao, Z., He, J., Cao, J., Huang, G., Zhang, P.: Multiple Time Series Anomaly Detection Based on Compression and Correlation Analysis: A Medical Surveillance Case Study. In: Sheng, Q.Z., Wang, G., Jensen, C.S., Xu, G. (eds.) APWeb 2012. LNCS, vol. 7235, pp. 294–305. Springer, Heidelberg (2012)
7. Fujimaki, R., Nakata, T., Tsukahara, H., Sato, A., Yamanishi, K.: Mining Abnormal Patterns from Heterogeneous Time-Series with Irrelevant Features for Fault Event Detection. Statistical Analysis and Data Mining 2 (2009)
8. Zhang, P., Gao, B.J., Zhu, X., Guo, L.: Enabling Fast Lazy Learning for Data Streams. In: Proceedings of ICDM (2011)
9. Zhang, P., Zhu, X., Shi, Y., Guo, L., Wu, X.: Robust ensemble learning for mining noisy data streams. Decision Support Systems 50(2), 469–479 (2011)
10. Stock, J.H., Watson, M.W.: Vector Autoregressions. Journal of Economic Perspectives 15(4), 101–115
11. Yves, N.: Total Least Squares: State-of-the-Art Regression in Numerical Analysis. SIAM Review 36 (2), 258–264
12. Wang, P., Wang, H., Wang, W.: Finding Semantics in Time Series. In: Proceedings of SIGMOD 2011 (2011)
13. Duncan, G., Gorr, W., Szczypula, J.: Forecasting Analogous Time Series, pp. 15213–13890. Carnegie Mellon University, Pittsburgh
14. Pang, C., Zhang, Q., Hansen, D.P., Maeder, A.J.: Unrestricted wavelet synopses under maximum error bound. In: Proceedings of EDBT 2009 (2009)
15. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings of ICDM 2001 (2001)
16. Wang, Y., Zhou, L.: Mining complex time-series data by learning the temporal structure using bayesian techniques and markovian models. In: Proceedings of ICDM 2006 (2006)
17. Zhang, P., Li, J., Wang, P., Gao, B., Zhu, X., Guo, L.: Enabling Fast Prediction for Ensemble Models on Data Streams. In: Proceedings of SIGKDD 2011 (2011)