Mining Good Sliding Window for Positive Pathogens Prediction in Pathogenic Spectrum Analysis*

Lei Duan[1], Changjie Tang[1], Chi Gou[1], Min Jiang[2], and Jie Zuo[1]

[1] School of Computer Science, Sichuan University,
Chengdu 610065, China
[2] West China School of Public Health, Sichuan University,
Chengdu 610041, China
{leiduan,cjtang}@scu.edu.cn

Abstract. Positive pathogens prediction is the basis of pathogenic spectrum analysis, which is a meaningful work in public health. Gene Expression Programming (GEP) can develop the model without predetermined assumptions, so applying GEP to positive pathogens prediction is desirable. However, traditional time-adjacent sliding window may not be suitable for GEP evolving accurate prediction model. The main contributions of this work include: (1) applying GEP-based prediction method to diarrhea syndrome related pathogens prediction, (2) analyzing the disadvantages of traditional time-adjacent sliding window in GEP prediction, (3) proposing a heuristic method to mine good sliding window for generating training set that is used for GEP evolution, (4) proving the problem of training set selection is NP-hard, (5) giving an experimental study on both real-world and simulated data to demonstrate the effectiveness of the proposed method, and discussing some future studies.

Keywords: Data Mining, Time Series, Sliding Window, Pathogens Prediction.

1 Introduction

Infectious disease prevention and control is an important and urgent issue in daily life. For example, thousands of people lost lives by SARS and A/H1N1. Correspondingly, adopting effective measures to prevent and control infectious diseases is a meaningful and challenging problem for public health research. To implement effective measures for infectious disease prevention and control, it is necessary for scientists to make clear of the infectious agent, that is, the pathogen of the infectious disease. The pathogen is a disease producer such as a virus, bacteria, prion, or fungus that causes disease to its host. For example, SARS is caused by a coronavirus [1].

The pathogenic spectrum of an infectious disease demonstrates the constituent ratio of each pathogen, which is related to the infectious disease. Example 1 gives an

* This work was supported by the National Research Foundation for the Doctoral Program by the Chinese Ministry of Education under grant No.20100181120029, and the Young Faculty Foundation of Sichuan University under grant No. 2009SCU11030.

J. Tang et al. (Eds.): ADMA 2011, Part II, LNAI 7121, pp. 152–165, 2011.

example of calculating the pathogenic spectrum. Pathogenic spectrum analysis is a meaningful work in public health, since the variation of the pathogenic spectrum is the basis of disease break. Specifically, the change rate of the pathogenic spectrum is a significant indicator to evaluate the possibility of infectious disease break. Moreover, predicting the trend of the pathogenic spectrum alternation is helpful for the early warning of infectious disease break.

Example 1. Given an infectious disease, suppose there are four viruses, v_1, v_2, v_3 and v_4, related to it. The virus-test result shows that the numbers of cases that are positive to these four viruses are 20, 50, 60 and 70, respectively. Then the virus pathogenic spectrum of this infectious disease consists of four parts. That is, v_1: 20/(20+50+60+70) = 10%, v_2: 50/(20+50+60+70) = 25%, v_3: 60/(20+50+60+70) = 30%, and v_4: 70/(20+50+60+70) = 35%.

As shown in Example 1, the problem of pathogenic spectrum prediction can be converted into predicting the number of positive cases of each disease-related pathogen. In practice, the public-health researchers apply the virus test to patients, and record the numbers of patients whose test results are positive. This kind of test is carried out in a fixed period, such as one week, one month. As a result, we can see that the positive pathogens prediction is a time series problem.

In public health domain, some traditional time series analysis methods, such as ARMA, ARIMA [2-4], Artificial Neural Networks (ANN) [5-7], which are implemented in SAS or SPSS software, have been widely used in positive pathogens prediction. However, none of these methods works well in all situations. For example, ARIMA is suitable for developing a linear model, while the disadvantages of ANN include "black box" nature, computational burden, proneness to over fitting, and the empirical nature of model building. Traditional methods may fail to develop adequate models due to the nonlinear dynamic behavior of time series, but also due to the lack of adaptation of the methods. This makes the problem is suitable for using heuristic methods, like evolutionary computation, which can develop the model without making many assumptions. For example, Genetic Programming has been widely performed for time series forecasting [8, 9].

The diarrhea syndrome monitoring data records the numbers of positive pathogens that are related to diarrhea syndrome every month since 2009 in China mainland. In this study, we apply GEP (Gene Expression Programming, GEP), the newest development of Genetic Programming [10, 11], to positive pathogens prediction in diarrhea syndrome monitoring data analysis.

We choose GEP as the prediction method, since it has following advantages:

- GEP can learn the fittest model from the data automatically without any predefined assumption. It has a powerful numeric calculation capability to evolve accurate model.
- Previous studies on applying GEP to time series analysis get desirable results.

The basic idea of applying GEP to time series mining is a sliding window prediction method. In training stage, once the size of sliding window is determined, the training

set can be generated by the sliding window. For data in sliding windows, GEP takes them as the independent variables and evolves a model to fit the target values. Moreover, the sliding window is always time-adjacent prior to the target value. For example, given a dataset $D = \{d_i \mid 1 \le i \le n\}$, suppose the sliding window size is 3. Then, for target value d_i, the dataset in sliding window is $\{d_{i\text{-}3}, d_{i\text{-}2}, d_{i\text{-}1}\}$. However, Example 2 demonstrates that this kind of sliding window may not be suitable to predict positive pathogens in diarrhea syndrome monitoring data.

Example 2. Let dataset $D = \{d_i \mid 1 \le i \le 24\}$ be the numbers of positive cases of a diarrhea syndrome related pathogen every month since 2005 to 2006. Each data in D is list in table below.

	Jan	*Feb*	*Mar*	*Apr*	*May*	*Jun*	*Jul*	*Aug*	*Sep*	*Oct*	*Nov*	*Dec*
2005	47	29	32	38	19	21	37	11	23	38	22	33
index	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
2006	35	19	24	32	14	17	34	8	21	36	20	31
index	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*

Suppose the sliding window size is 3. If we apply GEP to find the relationship between d_i and (d_i-1, d_i-2, d_i-3), $4 \le i \le 24$, GEP fails to find accurate relationship. However, if we apply GEP to find the relationship between d_i and (d_i-1, d_i-12, d_i-13), $14 \le i \le 24$, GEP can find that $d_i = d_{i\text{-}12} + (d_{i\text{-}1} - d_{i\text{-}13}) * 0.8$.

Though Example 2 is a simple synthetic example, it reveals the fact that traditional time-adjacent sliding window is not suitable for predicting positive pathogens, which are related with diarrhea syndrome. The reasons include:

- Firstly, in the diarrhea syndrome monitoring data analysis, the number of positive pathogens is related to environment factors, such as season, temperature. For example, it is unreasonable to predict the number of positive pathogens in autumn by the numbers in summer.
- Secondly, besides the numbers of positive pathogens in previous months, the numbers in the same of months of last year is important while predicting the positive pathogens of current month.

Additionally, Example 2 shows that sliding window is important for GEP. Since GEP takes the data in sliding window as independent variables, it cannot evolve the accurate prediction model from incorrect dataset.

To the best of our knowledge, there is no previous work on mining sliding window for GEP prediction. The main contributions of this work include: (1) applying GEP-based prediction method to diarrhea syndrome related pathogens prediction, (2) analyzing the disadvantages of traditional time-adjacent sliding window in GEP prediction, (3) proposing a heuristic method to mine good sliding window for generating training set that is used for GEP evolution, (4) proving the problem of training set selection is NP-hard, (5) giving an experimental study on both real-world

and simulated data to demonstrate the effectiveness of the proposed method, and discussing some future studies.

The rest of this paper is organized as follows. Section 2 introduces related works. Section 3 presents the main ideas used by our methods and the implementation of the algorithm. Section 4 reports an experimental study on both real-world diarrhea syndrome monitoring data and synthetic data. Section 5 discusses future works, and concluding remarks.

2 Related Works

2.1 Traditional Time Series Prediction Methods

Time series study is distinct from other data analysis problems, since time series data have a natural temporal ordering. By time series study, scientists can extract meaningful statistics and other characteristics of the data, and use the model to forecast future events based on known past events. Specifically, the model developed by time series prediction from the past data is used to predict data points before they are measured. Time series study has been widely applied in many domains, such as econometrics, meteorology, astronomy.

The model for time series data represents stochastic process. Based on the model form, time series prediction methods can be classified into three types: linear model, such as ARMA, ARIMA [12], non-linear, such as ARCH, GARCH [13, 14], and model-free, such as some wavelet transform based methods [15].

Traditional time series modeling methods have been widely applied to many infectious disease prevention and control studies [2-7]. The authors in [3] used ARIMA to predict the number of beds occupied during a SARS outbreak in a Singapore's tertiary hospital. In [4], ARIMA is used to predict the incidence of pulmonary tuberculosis. ANN can overcome the linear-modeling limitation of ARIMA, so it has been applied to many disease incidence predictions, such as cancer and hepatitis [6, 7].

2.2 GEP-Based Time Series Prediction

GEP is a new development of Genetic Algorithms (GA) and Genetic Programming (GP). The basic steps of using GEP to seek the optimal solution are the same as those of GA and GP. However, compared with GA or GP, the coding of individuals (candidate solutions) in GEP is more flexible and efficient [10, 11].

The most characteristic players in GEP are the chromosomes and the expression trees, the latter consisting of the expression tree of the genetic information encoded in the former. The chromosome is a linear, symbolic string of fixed length. One or more genes compose a chromosome by using linking function. Each gene is divided into a head and a tail. The head contains symbols that represent both functions and terminals, whereas the tail contains only terminals [10]. For each problem, the length of the head h is chosen by the user, whereas the length of the tail t is a function of h and the number of arguments of the function with more arguments n, and is evaluated

by the equation: $t = h\ (n - 1) + 1$. Consider a gene for which the set of functions F = {+, -, *, /}. In this case the maximum number of arguments of the element in F is 2, then $n = 2$.

In GEP, the length of a gene and the number of genes composed in a chromosome are fixed. Despite its fixed length, each gene has the potential to code for expression trees of different sizes and shapes, the simplest being composed of only one node and the biggest composed of as many nodes as the length of the gene [11].

Through parsing the expression tree in the hierarchy way, the algebraic expression part of GEP genes can be obtained. The structural organization of GEP genes guarantees that any genic change in the chromosome always generates a valid expression tree [10]. That is, all candidate solutions evolved by GEP are syntactically correct. The chromosome is called as the individual's genotype, while the expression tree is called as the individual's phenotype [11].

Figure 1 shows a gene is encoded as a linear string and its expression in expression tree. The valid part of gene is shown in bold in Figure 1.

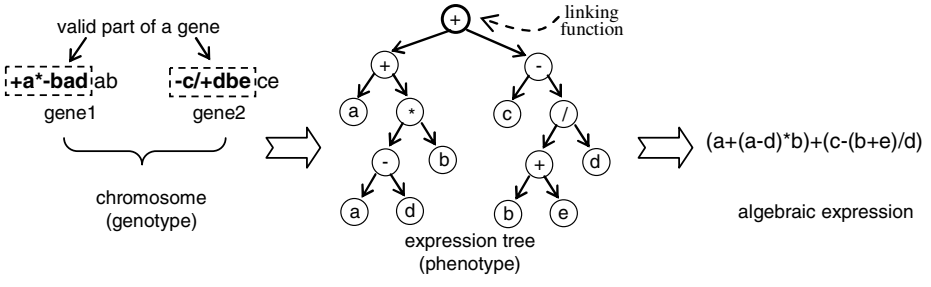

Fig. 1. The genotype, phenotype and algebraic expression of a GEP individual

The GEP algorithm begins with generating of the initial population, composed of a set of chromosomes, in a random way. Each chromosome is a candidate solution. Then the chromosomes (genotype) are expressed as expression trees (phenotype) and the fitness of each individual is evaluated by the predetermined fitness function. There are many kinds of measurements can be used as the fitness function, such as relative error and absolute error. The individuals are then selected according to fitness to reproduce with modification, generating new individuals with new traits. The individuals of the new generation are subjected to the same evolutionary process: expression of the genomes, selection by the fitness, and new individual generation. This procedure is repeated until a satisfactory solution is found, or a predetermined stop condition is reached. Then the evolution stops and the best-so-far solution, evolved by GEP, is returned [10].

GEP creates necessary genetic diversity for the selected individuals for keeping the evolutionary power in the long run. In nature, several genetic modifications, such as mutation, deletion, and insertion, are performed during the replication of the genomes. In basic GEP algorithm, the genetic operators perform in an orderly fashion, starting with replication and continuing with mutation, transposition, and recombination. The details of GEP implementation can be referred in [11].

Since GEP can evolve accurate mathematic model, it is used to build the prediction model that fits the time series data as well as possible. C. Ferreira gives a basic sliding window based method of applying GEP to time series study in [11]. This method consists of two steps.

- The first step is deciding the size of sliding window, that is, how many previous data points are used in predicting current data point. Suppose the window size is s. then the model predicts the value at a moment t, d_t, using the previous s values in the sample, denoted as d_{t-1}, d_{t-2},…, d_{t-s}. Based on the sliding window, the data set is participated into several training samples.
- Then GEP evolves a function f that predicts the values of a time series data as accurately as possible. Formally, let $f(d_{t-1}, d_{t-2},\ldots, d_{t-s}) = d'_t$, the function f that has the smallest error between d_t and d'_t is the best model, which is to be used for further prediction.

GEP has been used successfully to solve various time series problems so far. Besides the work in [11], the authors in [16] designed a GEP-based method, called as Differential by Microscope Interpolation, for sunspot series prediction. In [17], the authors applied an adaptive GEP-based method to predict the precipitation and temperatures in a region of Romania.

3 Sliding Window Mining

3.1 Sub-sliding Windows Enumeration

As stated above, the first step of applying GEP to time series prediction is determining the sliding window. In the basic GEP-based method for time series prediction, if the size of sliding window is s, for the value to be predicted at a moment t, d_t, the data in sliding window are previous s values to d_t. However, this kind of time-adjacent window may not be suitable for diarrhea syndrome monitoring data analysis as shown in Example 2.

Let the number of observed data be n. We select s data to compose the sliding window. Then, there will be C_n^s different sliding windows for selection. In general, the size of sliding window, s, is no greater than half of all observed data, $n/2$. We can get following:

$$C_n^s = \frac{n!}{(n-s)!s!} \geq \frac{(2s)!}{(2s-s)!s!} = \frac{(2s)!}{(s!)^2} \geq 2^s \tag{1}$$

From equation presented above, we can see that the search of selecting s data from all observed data is in exponential space. As a result, a polynomial time algorithm cannot enumerate all sliding windows that consist of s data.

From the diarrhea syndrome monthly monitoring data, we get two observations as follows. Firstly, the periodicity exists in the monthly positive pathogens. Intuitively, it is worthwhile to consider the pathogenic spectrum in June 2009, when predict the one in June 2010. Secondly, it is unreasonable to select much data, which are in the same

observation time period but the intervals to the predicted data are large in the sliding window. For example, compared with Nov. 2010, the data in Feb. 2010 is not helpful to improve the prediction accuracy of pathogenic spectrum in Dec. 2010.

According to the characteristics of monthly positive pathogens prediction, we design a heuristic method to enumerate candidate sliding windows based on following two principles:

- Considering data in previous time periods while predicting the current data.
- Paying more attention to the recent than to the past in prediction.

Given a dataset, D, contains all observed data. Let $d_t \in D$ be the value to be predicted at moment t, the time period of the observed data be T. Then we divide D into time-partitions, P, from d_t backward. For each $p_i \in P$, $p_i = \{d_r \mid 1 \le t - i \cdot T < r \le t\text{-}(i\text{-}1)\cdot T\}$. So, p_i with smaller index is closer to d_t. For example, suppose $T = 4$ and $t = 15$, then $p_1 = \{d_{12}, d_{13}, d_{14}, d_{15}\}$ as well as $p_2 = \{d_8, d_9, d_{10}, d_{11}\}$.

Definition 1 (sub-sliding window). Given a sliding window W, w_i is a sub-sliding window of W, iff w_i satisfies following conditions: i) $W = \bigcup_{i=1}^{k} w_i$; ii) $w_i \cap w_j = \varnothing$, $i \neq j$.

Definition 1 shows that $|W| = |w_1| + |w_2| + \ldots + |w_k|$, and $0 \le |w_i| \le |W|$. In this study, the value of k is determined by the user, and each sub-sliding sliding window is a subset of time-partition. That is, for each $w_i \in W$, $w_i \subseteq p_i$. As a result, the value of k is not greater than $|P|$. It is worthwhile to note that as w_i can be null ($\varnothing$), k is the maximal number of sub-sliding windows. The data in each sub-sliding window satisfy following constraints.

Constraint (i) The data in w_i are those in the rightmost side of p_i. But d_t is excluded from w_1, since it is the value to be predicted.

Constraint (ii) $|w_{i+1}| - |w_i| < \delta$, where δ is a predefined small positive integer.

In this work, we set δ is 1, since we prefer to pay more weight to the recent than to the past in prediction. Alternatively, the relationship between $|w_i|$ and $|w_{i+1}|$ can be a ratio. Note that, there is no limitation of how greater $|w_i|$ than $|w_{i+1}|$ is. Constraints (i) and (ii) satisfy the two principles stated above.

Example 3. Given a sliding window $W = \{w_1, w_2, w_3\}$, $|W| = 7$. Suppose the time period is 5 and d_{15} is the value to be predicted. Figure 2 illustrates the sub-sliding windows, when $|w_1| = 3$, $|w_2| = 2$, and $|w_3| = 2$.

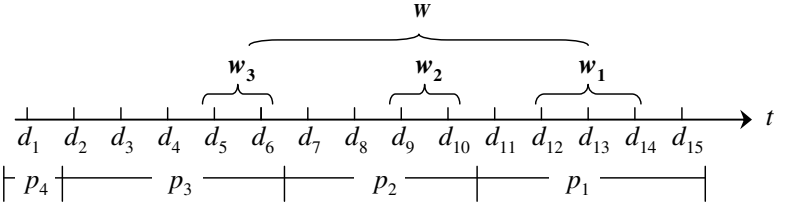

Fig. 2. An Example of a sliding window consists of 3 sub-sliding windows

Sub-sliding windows compose the sliding window for prediction. Given the size of sliding window and the maximal number of sub-sliding windows, there are different sub-sliding window combinations. Take w_1, w_2, and w_3 in Example 3 as an example, the lengths of them, denoted as ($|w_1|$, $|w_2|$ $|w_3|$), can be (7, 0, 0), (6, 1, 0), (6, 0, 1), (5, 2, 0), (5, 1, 1), (4, 3, 0), (4, 2, 1), (4, 1, 2), (3, 4, 0), (3, 3, 1), (2, 3, 2), (2, 2, 3), besides (3, 2, 2). The data in sub-sliding window are determined as soon as the size of the sub-sliding window is determined.

3.2 Finding the Best Sliding Window

Once the size of sliding window and the maximal number of sub-sliding windows are determined, an available sub-sliding window combination can be generated. Each sub-sliding window combination constructs a candidate sliding window. We find all candidate sliding windows that satisfied Constraint (i) and (ii), and make use of these candidate sliding windows to generate training sets. Afterwards, we apply GEP to training sets to evolve the best model as well as good sliding window. Algorithm 1 describes the pseudo code of finding the most accurate model for prediction.

Algorithm 1: Prediction_Model_Mine (D, T, w, k)
Input: (1) observed dataset: D; (2) the time period: T; (3) the size of sliding window: w; (4) the maximal number of sub-sliding windows: k.
Output: prediction model: gepModel.

```
begin
    1. subwinSet ← subWinGenerate(w,k)
    2. dataSubSet ← DataSplit(D,T)
    3. For each subwin in subwinSet
    4.      trainingSet ← Select(subwin, dataSubSet)
    5.      TrainSets ← TrainSets + trainingSet
    6. For each trainset in TrainSets
    7.      gepScore ← gepPrediction(trainset)
    8. gepModel ← the model with the highest gepScore
    9. return gepModel
end.
```

In Algorithm 1, Function *subWinGenerate*(*w*,*k*) in Step 1 generates all candidate sliding windows, which satisfy Constraint (i) and (ii), based on the size of sliding window (*w*) and the maximal number of sub-sliding windows (*k*). For each data to be taken as a target value in training set, Function *DataSplit*(*D*,*T*) in Step 2 divides the data before it into several time-partitions. From Step 3 to Step5, for each candidate sliding window, Function *Select*(*subwin*, *dataSubSet*) generates the training samples by selecting data from *dataSubSet* based on sliding window *subwin*. Each generated training set is evaluated by GEP in Step 7. Then the most accurate model (good sliding window) evolved by GEP will be used for prediction.

Proposition 1. Given a sliding window $W = \{w_1, w_2, w_3, \ldots, w_k\}$, $k > 1$ and $|W| = M$. Let $NC(W)$ be the number of sub-sliding window combinations satisfying the Constraints (i) and (ii). Then $NC(W) < (M+2)^{k-1}$.

Proof. We prove Proposition 1 by induction.

Basis: When $k = 2$, $W = \{w_1, w_2\}$, $|w_2| = M - |w_1|$. As $|w_1| \in \{0, 1, 2, ..., M\}$, $NC(W) = (M+1) < (M+2)$ as desired.

Inductive steps: Assume $NC(W) < (M+2)^{n-1}$, when $k = n$. That is, $W = \{w_1, w_2, \ldots, w_n\}$. For $k = n + 1$, let $W = W' \cup w_{n+1}$, where $W' = \{w_1, w_2, w_3, \ldots, w_n\}$. As $|W'| = M - |w_{n+1}|$, $NC(W') < (M - |w_{n+1}| + 2)^n < (M + 2)^n$. The number of combinations between W' and w_1 is $(M + 1)$. Thus, $NC(W) < (M + 2)^n \cdot (M + 1) < (M + 2)^{(n+1)} = (M + 2)^k$.

Thus, it holds for $k = (n + 1)$ and this completes the proof.

Proposition 1 shows that the number of sub-sliding window combinations is increased in polynomial space. Thus, the calculation-step in Algorithm 1 generates training sets for prediction in polynomial time.

Algorithm 1 describes the process of generating the training sets followed by applying GEP method to evolve the best model for prediction. In this work, we call this process as *Training set selection problem*.

Intuitively, the training set selection problem is more difficult than finding the minimal attribute reduction of decision table, which is a NP-hard problem proved by Wong S K M and Ziarko W [18]. From Reference [19], we have following lemma.

Lemma 1. The problem of subset sum is NP-complete.

Theorem 1. The problem of training set selection is NP-hard.

Proof. The basic idea of proof is proving that the subset sum problem is polynomial time Turing-reducible to training set selection problem.

Given an integer set $C = \{c_1, c_2, \ldots, c_n\}$, construct a training set D_T as follows.

$$D_T = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \text{ where } a_{ij} \in \{c_1, c_2, \ldots, c_n\} \cup \{0\}, \text{ and } m \geq 2^n.$$

Suppose the mother function:

$$f(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n - d_t$$

where $x \in [1 - \varepsilon, 1 + \varepsilon]$, ε is a predefined small positive number, d_t is the target value in prediction. Without loss of generality, we assume both training data and d_t are integers, since we can expand all non-integer values to integers synchronously.

We now apply GEP to train $f(x)$ over D_T to optimize $f(x)$, so that $|f(x)| < 1$. The values of $(b_0, b_1, b_2, \ldots, b_n)$ are fetched from D_T in training process.

When $x = 1$,

$$|f(1)| = |b_0 + b_1 + b_2 + \ldots + b_n - d_t| < 1$$

As all values of b_0, b_1, b_2, …, b_n and d_t are integers, the value of $|f(1)|$ is an integer. However, the only integer that is less than 1 is 0, so $|f(1)| = 0$. Then, $b_0 + b_1 + b_2 + \ldots + b_n = d_t$. This shows that the subset sum problem is reduced to this problem.

By Proposition 1, the size of D_T, generated in our proposed method, is increased in polynomial space. The training set for prediction can be generated in polynomial time. Moreover, based on [9], GEP can evolve $f(x)$ to the optimize-target in polynomial time. Finally, the subset sum problem can be reduced to training set selection problem in polynomial time. The problem of training set selection is NP-hard.

4 Experimental Study

4.1 Real-World Positive Pathogens Prediction

To evaluate the performance of our GEP-based sliding window mining method, we implement all proposed algorithms in Java. The experiments are performed on an Intel Pentium Dual 1.80 GHz (2 Cores) PC with 2G memory running Windows XP operating system. We apply our proposed method to the real-world diarrhea syndrome monitoring data, which is provided by the department of health statistics, Sichuan University. There are four viruses, calicivirus, rotavirus, adenovirus and astrovirus, related to diarrhea syndrome, so these four viruses are pathogens of diarrhea syndrome. The monitoring data contains the numbers of cases that are positive to virus test for these four pathogens in every month of Year 2009 and Year 2010. Since the data is sensitive, we skip over the semantic details and formulate the data formally as follows. Let D be the monitoring data of any related pathogen. Suppose $D = \{d_i \mid 1 \leq i \leq 24\}$, where d_i is the number of positive pathogen cases in one month. In D, d_1 is the data of January in 2009, and d_{24} is the data of December in 2010. In our work, we just consider the effectiveness of the proposed method in the real-world positive pathogens prediction, instead of the meaning of predicted data.

As the observation time period of diarrhea syndrome monitoring data is one year, the monitoring data can be divided into two time-partitions, p_1 and p_2, at most based on Algorithm 1. Take d_{24} as an example, the first time-partition p_1 contains data in Year 2010, and the second time-partition p_2 contains data in Year 2009.

Table 1. Parameters for GEP Evolution

Parameter	value	Parameter	value
Population size	100	One-point recombination rate	0.4
Number of Generations	10000	Two-point recombination rate	0.2
Linking function	+	Gene recombination rate	0.1
Function set	{+, –, *, / }	IS transposition rate	0.1
Number of genes	3	IS elements length	1, 2, 3
Gene head size	8	RIS transposition rate	0.1
Selection operator	tournament	RIS elements length	1, 2, 3
Mutation rate	0.04	Gene transposition rate	0.1

The proposed method is applied to the monitoring data to generate the training sets for GEP prediction. The size of sliding window is set as 6. In considering the requirement of diarrhea syndrome analysis and the total number of monitoring data is small, for each pathogen, we take the last monitoring data, d_{24}, as the test value in the experiments. Let W be the sliding window. There are two sub-sliding windows, w_1 and w_2, satisfying $w_1 \subseteq p_1$ and $w_2 \subseteq p_2$. The proposed method generates different combinations of w_1 and w_2, as well as corresponding training sets. For each training set, we run GEP 10 times independently, and record the average training accuracy and prediction accuracy, which are measured in absolute error. Table 1 lists the GEP related parameters in our experiments.

Table 2 to Table 5 lists the experiment results. As the size of sliding window is set as 6, the available sub-sliding window combinations include ($|w_1|$=6, $|w_2|$=0), ($|w_1|$=5, $|w_2|$=1), ($|w_1|$=4, $|w_2|$=2) and ($|w_1|$=3, $|w_2|$=3). We take the combination ($|w_1|$=6, $|w_2|$=0) as the baseline sliding window, for it is the traditional time-adjacent sliding window. The highest average accuracies of training and test are in bold font. As the main purpose of this experiment is verifying the effectiveness of the method to discover good sliding window for GEP prediction, without loss of generality, we apply the basic GEP method on each training set. We believe that more accurate prediction results can be got by some improved GEP methods, such as the methods in [16, 17].

Table 2. The experimental results on positive calicivirus prediction when $|W| = 6$

	($\|w_1\|$=6, $\|w_2\|$=0)	($\|w_1\|$=5, $\|w_2\|$=1)	($\|w_1\|$=4, $\|w_2\|$=2)	($\|w_1\|$=3, $\|w_2\|$=3)
Training Accu.	96.21	89.72	93.08	**82.60**
Test Accu.	135.67	98.89	104.10	**45.67**

Table 3. The experimental results on positive rotavirus prediction when $|W| = 6$

	($\|w_1\|$=6, $\|w_2\|$=0)	($\|w_1\|$=5, $\|w_2\|$=1)	($\|w_1\|$=4, $\|w_2\|$=2)	($\|w_1\|$=3, $\|w_2\|$=3)
Training Accu.	**102.82**	109.39	116.68	107.22
Test Accu.	**222.89**	357.33	330.75	290.57

Table 4. The experimental results on positive adenovirus prediction when $|W| = 6$

	($\|w_1\|$=6, $\|w_2\|$=0)	($\|w_1\|$=5, $\|w_2\|$=1)	($\|w_1\|$=4, $\|w_2\|$=2)	($\|w_1\|$=3, $\|w_2\|$=3)
Training Accu.	18.34	18.77	17.18	**16.04**
Test Accu.	10.56	12.73	14.75	**5.13**

Table 5. The experimental results on positive astrovirus prediction when $|W| = 6$

	($\|w_1\|$=6, $\|w_2\|$=0)	($\|w_1\|$=5, $\|w_2\|$=1)	($\|w_1\|$=4, $\|w_2\|$=2)	($\|w_1\|$=3, $\|w_2\|$=3)
Training Accu.	14.79	13.69	14.21	**12.85**
Test Accu.	44.78	56.80	58.17	**32.11**

From Table 2 to Table 5, we can see that for predicting the positive pathogens of calicivirus, rotavirus, adenovirus and astrovirus, different sub-sliding window combinations get different training and test accuracies. The highest training accuracy and test accuracy can be got when the combination of sub-sliding windows is ($|w_1|$=3,

$|w_2|$=3), while for predicting the positive pathogens of adenovirus, the highest training accuracy and test accuracy are got when the combination of sub-sliding windows is ($|w_1|$=6, $|w_2|$=0). Thus, better sliding window for prediction, compared with traditional time-adjacent sliding window, can be discovered by our proposed method. Moreover, in the case of time-adjacent sliding window is good for prediction our method also can find it, such as predicting the positive pathogens of rotavirus.

For each training set generated by the good sliding window, we increase the number of evolution generations as 30000, and run GEP 10 times independently. Then more accurate prediction results can be got as list in Table 6.

Table 6. The prediction accuracy of GEP-based method evolving 30000 generations

	calicivirus	rotavirus	adenovirus	astrovirus
Test Accu.	38.17	217.80	3.83	18.50

As shown in above tables (from Table 2 to Table 6), we can see that it is necessary and effective to apply the proposed method to diarrhea syndrome related pathogens prediction to discover good sliding windows, which can improve the prediction accuracy .

4.2 Synthetic Data Prediction

As there are only two years real-world diarrhea syndrome monitoring data available, in order to demonstrate that the proposed sliding window mining method is effective for long-term monitoring data, we copy the calicivirus monitoring data 10 times to simulate the 20-years monitoring data. We apply the proposed method to the simulated data to generate the training sets for GEP prediction. The size of sliding window is set as 7. The maximal number of sub-sliding window (k) is set as 2, since we simulate the data by the 2-years monitoring data. The sliding window includes the data, which equal to the values of the target data, in the case of k equals to 3.

For each dataset generated by sliding windows that enumerated by the proposed method, we keep the last 12 data as the test set, and run GEP on the rest data 10 times independently. The GEP related parameters are kept the same as shown in Table 1. Table 7 lists the average training and prediction accuracies of GEP model per data under each sliding window, which is composed by different sub-sliding windows.

Table 7. The experimental results on simulated data when $|W|$ = 7

	($|w_1|$=7, $|w_2|$=0)	($|w_1|$=6, $|w_2|$=1)	($|w_1|$=5, $|w_2|$=2)	($|w_1|$=4, $|w_2|$=3)	($|w_1|$=3, $|w_2|$=4)
Trai. Accu.	19.14	20.93	18.86	17.89	19.59
Test Accu.	22.80	26.50	22.25	20.40	25.25

Table 7 shows the model with the highest accuracy, evolved by GEP, is got when the combination of sub-sliding windows is ($|w_1|$=4, $|w_2|$=3). Besides combination ($|w_1|$=4, $|w_2|$=3), the model evolved under the combination ($|w_1|$=5, $|w_2|$=2) is more accurate than the one evolved under time-adjacent sliding window, i.e. the

combination ($|w_1|$=7, $|w_2|$=0). The combinations ($|w_1|$=6, $|w_2|$=1) and ($|w_1|$=3, $|w_2|$=4) are not suitable for prediction compared with other combinations.

In addition, compared with Table 2, we can see that the performance on synthetic dataset is better than the one on real dataset. We analyze the reason lies that for synthetic dataset analysis, there are more data are taken as the training set, which improve the accuracy of evolved model. After all, from the experimental results on the synthetic data, we can see that mining good sliding window is helpful for GEP to evolve accurate models.

5 Discussions and Conclusions

Positive pathogens prediction is the basis of pathogenic spectrum analysis, which is a meaningful work in public health. Different form traditional methods that may fail to develop adequate models due to the nonlinear dynamic behavior of time series, or the lack of adaptation of the methods, GEP can develop the model without making many assumptions. As a result, applying GEP to positive pathogens prediction is desirable. However, traditional time-adjacent sliding window may not be suitable for GEP evolving accurate prediction model. Based on analyzing the characteristics of diarrhea syndrome, we propose a heuristic method to mine good sliding window for generating training set, which is used for GEP evolution. Furthermore, we prove the problem of training set selection is NP-hard. The experimental study on real-world positive pathogens prediction shows that our proposed method is necessary and effective for diarrhea syndrome related pathogens prediction.

There are many works worth to be deeply analyzed in the future. For example, how to add the environment factors in good sliding window mining, how to describe the relationships among previous data, and how to evaluate the candidate sliding windows in a fast way. Moreover, we will consider applying the proposed method to other applications in public health, and other domains, such as economics and finance.

Acknowledgments. The authors thank to the faculty of the department of health statistics, Sichuan University, for providing the research data and their helpful comments to this work.

References

1. United Nations World Health Organization, http://www.who.int/mediacentre/releases/2003/pr31/en/
2. Reis, B.Y., Mandl, K.D.: Time Series Modeling for Syndromic Surveillance. BMC Med. Inform. Decis. Mak. 3(1), 2 (2003)
3. Earnest, A., Chen, M.I., Ng, D., Sin, L.Y.: Using Autoregressive Integrated Moving Average (ARIMA) Models to Predict and Monitor the Number of Beds Occupied During a SARS Outbreak in a Tertiary Hospital in Singapore. BMC Health Services Research 5, 5–36 (2005)

4. Meng, Lei, Wang, Yuming: Application of ARIMA Model on Prediction of Pulmonary Tuberculosis Incidence. Chinese Journal of Health Statistics 27(5), 507–509 (2010)
5. Zhang, G.P.: Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing 50, 159–175 (2003)
6. Khan, J., Wei, J.S., Ringnér, M., Saal, L.H., et al.: Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks. Nature Medicine 7, 673–679 (2001)
7. Guan, P., Huang, D.-S., Zhou, B.-S.: Forecasting Model for the Incidence of Hepatitis A based on Artificial Neural Network. World Journal of Gastroenterology 10(24), 3579–3582 (2004)
8. De Falco, Della Cioppa, A., Tarantino, E.: A Genetic Programming System for Time Series Prediction and Its Application to El Niño Forecast. Advances in Soft Computing 32, 151–162 (2005)
9. Barbulescu, A., Bautu, E.: ARIMA Models versus Gene Expression Programming in Precipitation Modeling. In: Proc. of the 10th WSEAS Int'l Conf. on Evolutionary Computing, pp. 112–117 (2009)
10. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems 13(2), 87–129 (2001)
11. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. Angra do Heroismo, Portugal (2002)
12. Brockwell, P., Davies, R.: Introduction to Time Series. Springer, New York (2002)
13. Bollerslev, T.: Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics 31, 307–327 (1986)
14. Hacker, R.S., Hatemi, J.A.: A Test for Multivariate ARCH Effects. Applied Economics Letters 12(7), 411–417 (2005)
15. Chui, C.K.: An Introduction to Wavelets. Academic Press, San Diego (1992)
16. Zuo, J., Tang, C., Li, C., Yuan, C.-A., Chen, A.-l.: Time Series Prediction Based on Gene Expression Programming. In: Li, Q., Wang, G., Feng, L. (eds.) WAIM 2004. LNCS, vol. 3129, pp. 55–64. Springer, Heidelberg (2004)
17. Barbulescu, A., Bautu, E.: Time Series Modeling Using an Adaptive Gene Expression Programming Algorithm. International Journal of Mathematical Models and Methods in Applied Sciences 3(2), 85–93 (2009)
18. Wong, S.K.M., Ziarko, W.: On Optimal Decision Rules in Decision Tables. Bulletin of Polish Academy of Sciences 33(11-12), 693–696 (1985)
19. Sipser, M.: Introduction to the Theory of Computation, 2nd edn., Thomson Learning, Stanford (2005)