



Low Complexity Auto-adaptive Algorithm for Finite State Prediction Given Historical Observations

Smal TIGANI

Computer Science Department
 National High School of Electricity and Mechanics
 Casablanca, Morocco

Mohamed OUZZIF

Computer Science Department
 High School of Technology
 Casablanca, Morocco

Rachid SAADANE

Electrical Department.
 Hassania School of Labor Works
 Casablanca, Morocco

ABSTRACT

This work presents an auto-configurable algorithm for finite state prediction. The specificity of this algorithm is the capacity of self-rectification of the prediction strategy before final decision. The auto-rectification mechanism is based on two parallel mathematical models : a Markov chain model for next state prediction rectified with a linear regression model for residues forecasting. For a normal distribution, the interactivity between the two models allows the algorithm to self-optimize its performance and then make better prediction. This work proposes also some statistical key performance indicators in order to prove the efficiency of the approach. Simulation results shows the advantages of the proposed algorithm compared with the traditional one.

General Terms:

Statistical Learning Theory, Prediction Systems.

Keywords:

Auto-configurable Algorithms, Statistical Learning, Stochastic Process, Linear Regression, Performance Analysis.

1. INTRODUCTION

In some systems behaving randomly, the missing of decision information presents, in one hand, some difficulties to maintain good processing going on. In the other hand, usually the manager needs to anticipate decision information to prepare adequate action planning. Prediction patterns so, seem to be the right solution to get necessary information from a random environment. Current prediction technics, based on Markov's theory and statistical analysis, learn just from historical observations and do not include prediction residues as a supplementary experience in order to adjust initial prediction strategy. This paper comes to contribute on intelligent prediction algorithms with strategy self-rectification capability. The work presents a hybrid prediction algorithm using on observations history, as a first experience, and prediction residue history as second supplementary experience. That allows prediction operation to benefit from double experience witch gives it more perfection.

The content of this paper is organized as follows : the section 2 presents some related works, while the third section introduces the mathematical modelling of the prediction rectification mechanism. The algorithm design and computational complexity analysis are reported to the fourth section, while the key performance indicators design and the simulation results are presented in section 5. Finally, Section 6 presents concluding remarks followed by discussion of next works.

2. RELATED WORKS

Several prediction models are discussed by many authors. For one thing, M.M. Mohie El-Din and al (2011) explain in [1] that

one way to predict the future observation is the building of an interval, called prediction interval, which contains the next observation with a fixed probability. For an other, M. Hossain (2013) introduces in [2] prediction model based on Beleaf Bayesian Network for a real-time crash prediction. In order to increase performance aspects, some approaches combine more than one prediction model as described in [3] (2013) by D. Ying Ying and al. That allows us to think about the design of a hybrid algorithm implementing more than one model. C.-J. Cheng and al, in [4], uses Markov chains to predict customer lifetime, however M. Cavers and al used in [5] a Markovian transition matrix to represent states of an Earthquake sequences grouped in zones. In an other side, linear regression is used also for forecasting next value and the trend extraction according to works of D. Ying Ying Sim and al in [7] (2014) and J. Guo-xun and al in [8] (2011).

In this work we propose a finite state prediction algorithm. For this, the use of Markov chains ([3], [4] and [5]) seems to be the best way to predict environment discrete states. In an other side, the auto-rectification mechanism relies on linear regression ([7] and [8]) to extract the residues trend and predict eventual next residue.

3. MATHEMATICAL MODELLING

3.1 Environment Modelling

Let suppose an environment behaving randomly and X the random variable representing the supervised parameter on the environment. This one can be the free memory size of a Web server for example or the available bandwidth of an IP channel. We define X_n as observed value of the random variable X at the instance n . The supervised parameter, at the next instance $n + 1$, changes the previous state X_n by adding an algebraic quantity that we call Δ_{n+1} . We can finally write :

$$X_{n+1} = X_n + \Delta_{n+1} = f(X_n, \Delta_{n+1}) \quad (1)$$

The expression of X_{n+1} in equation 1 confirms that the state sequence $(X_n, n \in \mathbb{N}^*)$ is a Markov chain.

3.2 State Prediction Model

Let consider the Markov chain $(X_n, n \in \mathbb{N}^*)$ having values in a finite state \mathbb{E} . Transition matrix at the instant $n + 1$, denoted $(\mathbb{P}_{ij}^{(n+1)})$, is given by the conditional probability to observe j at the next instant if we know that we have observed i in the past :

$$\mathbb{P}_{ij}^{(n+1)} = p(X_{n+1} = j | X_n = i)$$

The probability to observe a given element j at the instance $n + 1$, considering all observations probabilities, is given by total probability law :

$$p(X_{n+1} = j) = \sum_{i=1}^Q p(X_n = i) \mathbb{P}_{ij}^{(n+1)} \quad (2)$$

Initially, we suppose that all observations are equiprobable, it means that $p(X_0 = i) = \frac{1}{Q}$ for each $i \in \mathbb{E}$ with $Q = \text{Card}(\mathbb{E})$. After each observation, the matrix (\mathbb{P}_{ij}) will be updated and that guarantees learning from experiences. Let X_{n+1}^* be the predicted value using the Markov transition matrix for the next instant $n+1$ is given by the equation 3 :

$$X_{n+1}^* = \arg \max_{j \in \mathbb{E}} p(X_{n+1} = j) \quad (3)$$

3.3 Residue Modelling

Let ε_n be the prediction residue done at the instant n . We define it as the algebraic distance between the observed value and the predicted value for the same instant. The choice of algebraic distance and not the euclidian distance for example is due to the need of the sign, the residue of the prediction should translate the fact that the algorithm had predicted a value more than the necessary or less than it. Formally :

$$\varepsilon_k = X_k^* - X_k \quad (4)$$

Let's $\{\varepsilon_1, \dots, \varepsilon_n\}$ be historical residues set. We represents those data points with a linear model having the form $\varepsilon_k^* = \alpha \cdot k + \beta$, with k represents the instant and ε_k^* the corresponding residue in the linear model. The choice of a linear model is due to the need of extraction of the trend on residues cloud and then predict, approximatively, the next eventual residue. The parameters α in equation 5a and β in 5b are estimated with the least mean square method, See the proof in Appendix A. We write :

$$\alpha = \frac{12}{n^3 - n} \cdot \sum_{k=1}^n \varepsilon_k \left(\frac{2k - n - 1}{2} \right) \quad (5a)$$

$$\beta = \sum_{k=1}^n \varepsilon_k \left(\frac{3(n+1-2k)}{n^2 - n} + \frac{1}{n} \right) \quad (5b)$$

3.4 Rectification Mechanism

Let X_{n+1}^* be the predicted value obtained with the model 3 and ε_{n+1}^* the residue that we can make by deciding that the next value is X_{n+1}^* . This residue is estimated with the linear model described in the previous subsection. **The aim of this work is to develop the mechanism allowing the algorithm to auto-rectify its decision taking in consideration the future residue that it eventually can do.** The main idea is that the algorithm makes double prediction based on two different models : the first prediction is the next state of the environment using the approach in equation 3 and the second prediction is the next residue obtained with the trend of residues. With the two future information we build an interval I centred with X_{n+1}^* and having the radius ε_{n+1}^* .

$$\tilde{X}_{n+1}^* = \arg \min_{u \in \mathbb{E}} |X_{n+1}^* - u| - \varepsilon_{n+1}^* \quad (6)$$

4. ALGORITHM DESIGN

This section presents the main algorithm, named *iNSP* "Intelligent Next State Predictor", who makes the next state prediction. The algorithm 1 implements the equation 6 to rectify the initial decision obtained by the model 3. The total complexity of the algorithm *iNSP* is $\Theta(n)$. See the proof in Appendix B. The next residue serving the rectification process is given by the linear model whose coefficients are computed by 5a and 5b. This rectification comes to find the nearest state from a bound, right or left, of the interval I . See the algorithm :

The table 1 summarises the computational complexity analysis of the algorithm 1 :

Algorithm 1: *iNSP*

Input: $\vec{\varepsilon}$: Residues History

Output: \tilde{X}_{n+1}^* : Rectified Prediction

```

1  $vTmp \leftarrow 0$ 
2 foreach  $u \in \mathbb{E}$  do
3   if  $\pi_{n+1}(u) > vTmp$  then
4      $vTmp \leftarrow \pi_{n+1}(u)$ 
5      $X_{n+1}^* \leftarrow u$ 
6  $\varepsilon_{n+1}^* \leftarrow \alpha(n+1) + \beta$ 
7  $vTmp \leftarrow +\infty$ 
8 foreach  $u \in \mathbb{E}$  do
9   if  $vTmp > |X_{n+1}^* - u| - \varepsilon_{n+1}^*$  then
10     $vTmp \leftarrow |X_{n+1}^* - u| - \varepsilon_{n+1}^*$ 
11     $\tilde{X}_{n+1}^* \leftarrow u$ 
12 return  $\tilde{X}_{n+1}^*$ 

```

Line number	Elementary Operations	Complexity class
2 and 15	2	$\Theta(1)$
3 to 6	$3\text{Card}(\mathbb{E})$	$\Theta(\text{Card}(\mathbb{E}))$
8	$41n - 4$	$\Theta(n)$
10 to 14	$9\text{Card}(\mathbb{E})$	$\Theta(\text{Card}(\mathbb{E}))$

Table 1. : Complexity Analysis of Algorithm 1.

The complexity $\Theta(\text{Card}(\mathbb{E}))$ is a constant complexity class $\Theta(1)$ because \mathbb{E} is a finite states set whatever the observation number. Total complexity $\zeta_T(n)$ is given by the maximum of all structured blocks complexities :

$$\zeta_T(n) = \max(\Theta(1), \Theta(n)) = \Theta(n)$$

5. SIMULATION RESULTS

5.1 Key Performance Indicators

The aim of this section is the proof, by simulation, that proposed algorithm is able to improve the performance by learning from its experience. We present a Monte Carlo simulation in which we simulate the supervised environment by a generator of random states. In order to prove the reliability of the approach, we have designed some key performance indicators (KPIs) to study the performance aspects of proposed approach compared with the traditional one described in literature.

5.1.1 Cumulative Residues Indicator. We define this indicator as the cumulative function of all residues from the beginning of the simulation until its end. This KPI is computed for the traditional approach and the proposed one simultaneously. We define $\Gamma_{ta}(u)$ and $\Gamma_{pa}(u)$ as the cumulative residues of the traditional approach and proposed approach respectively. Formally :

$$\Gamma_{ta}(u) = \sum_{n=1}^u \varepsilon_n \quad (7a)$$

$$\Gamma_{pa}(u) = \sum_{n=1}^u \tilde{\varepsilon}_n \quad (7b)$$

Previous KPI in equations 7a and 7b are computed during a simulation of 2000 observations randomly generated. The figure 1 shows obtained results :

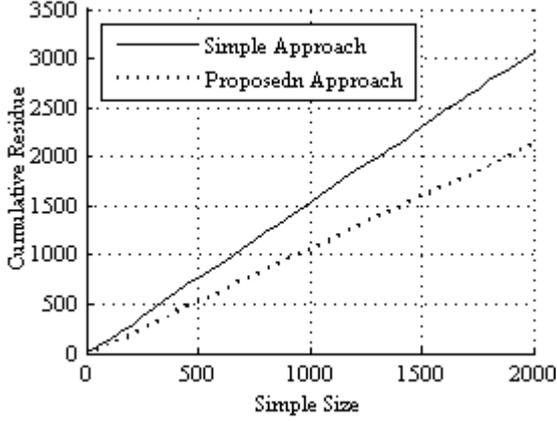


Fig. 1: Cumulative Residues : Comparative Study

The figure 2 shows how residues are distributed for each approach. It is evident, according to the graphic, that small residues, from -1 to 1 are more done by the proposed approach iNSP. However big residues, in range from 1 to 4 , are more done by the traditional approach. See the figure 2 :

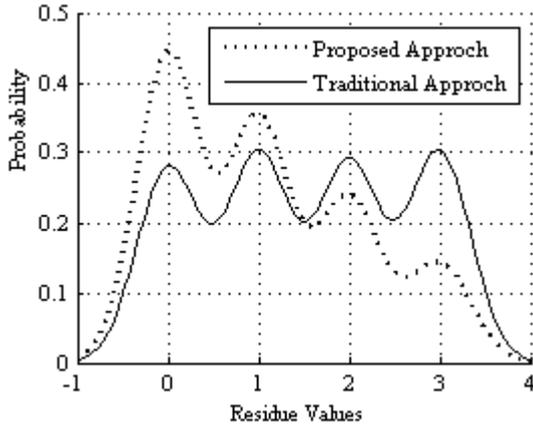


Fig. 2: Residues Distribution : Comparative Study

5.1.2 Decision Indicators. We define the KPI d^+ as the average of all observations having the residue at a given instant less than the residue at the previous instant [9] (2014). The function $\mathbf{1}_{\{\varepsilon_k > \varepsilon_{k+1}\}}$ is equal to 1 if $\varepsilon_k > \varepsilon_{k+1}$ and equals to 0 otherwise.

$$d^+ = \frac{1}{Q} \sum_{n=1}^Q \mathbf{1}_{\{\varepsilon_k > \varepsilon_{k+1}\}} \quad (8a)$$

$$d^- = \frac{1}{Q} \sum_{n=1}^Q \mathbf{1}_{\{\varepsilon_k < \varepsilon_{k+1}\}} \quad (8b)$$

The cumulative residue of all the simulation is given by $\Gamma_{ta}(Q)$ and $\Gamma_{pa}(Q)$ respectively for the approaches with Q is the simulation sample size. Otherwise, let ε^t be the tolerable residue fixed in the simulation at 1. Based on it, we define the tolerable residues average $E(\tilde{\varepsilon}^t)$ for the rectified prediction : proposed approach and $E(\varepsilon^t)$ for the traditional one. Formally :

$$E(\tilde{\varepsilon}^t) = \frac{1}{Q} \sum_{n=1}^Q \tilde{\varepsilon}_n \mathbf{1}_{\{\tilde{\varepsilon}_n \leq \varepsilon^t\}} \quad (9a)$$

$$E(\varepsilon^t) = \frac{1}{Q} \sum_{n=1}^Q \varepsilon_n \mathbf{1}_{\{\varepsilon_n \leq \varepsilon^t\}} \quad (9b)$$

The KPIs in equations 8a, 8b, 9a and 9b are computed during a Monte Carlo simulation and results are shown in the table 2 :

Key Performance Indicator	Value
Good Decision Rate for Traditional Approach	63.982 %
Good Decision Rate for Proposed	64.082 %
Bade Decision Rate Traditional	36.018 %
Bade Decision Rate Proposed	35.918 %
Tolerable Residues Average Traditional	42.174 %
Tolerable Residues Average Proposed	57.825 %
Cumulative Residues Traditional	58.915 %
Cumulative Residues Proposed	41.084 %

Table 2. : KPI Values : Comparative Study.

6. CONCLUSION

This paper presents a finite state prediction algorithm with low computational complexity. The specificity of this algorithm is the self-rectification capacity of the prediction strategy. Designed algorithm is based on Markov chain theory as a main prediction model and linear regression as rectification model. Proposed key performance indicators, computed during the simulation, shows that the proposed algorithm is really able to self-improve performance aspects and we conclude that prediction's perfection benefits from double experience and that makes it increasing.

There is no doubt that with the self-configurable algorithms designed for state prediction, the Big Data Analytic field can benefit considerably [10], [11]. Learning from large volumes of data, stored by sensors for example on exchanged by social networks, loses significance without sensitive algorithms able detects the trend changes. As perspectives, the use of proposed algorithm in financial markets state forecasting is a crucial application field. That can be by building an indicator that gives buy or sell best times or more technically, by building an automatic program, that makes decision it self and sends commands to brokers servers.



Acknowledgment

We are grateful to anonymous reviewers for their constructive comments which improved significantly the quality of the article.

Appendices

Appendix A : Linear Model Proof

Having data $\{(1, \varepsilon_1), \dots, (n, \varepsilon_n)\}$ and the linear model representing those data with the form : $\varepsilon_k^* = \alpha.k + \beta$. Using the Least Mean Square method [8], we find the coefficients α and β minimizing the quantity :

$$e_n = \sum_{k=1}^n [\varepsilon_k - \varepsilon_k^*]^2$$

In order to find the coefficients minimizing the quantity above, we have to solve the equation :

$$\left(\frac{\partial e_n}{\partial \alpha}, \frac{\partial e_n}{\partial \beta} \right) = (0, 0)$$

The calculation of the two derivatives gives the following :

$$\alpha \sum_{k=1}^n k^2 + \beta \sum_{k=1}^n k = \sum_{k=1}^n k \varepsilon_k$$

$$\alpha \sum_{k=1}^n k + n\beta = \sum_{k=1}^n \varepsilon_k$$

In order to simplify, we replace summations with : $\sum_{k=1}^n k = n(n+1)/2$ and $\sum_{k=1}^n k^2 = n(n+1)(2n+1)/6$. After simplifications we have got (5a and 5b) :

$$\alpha = \frac{12}{n^3 - n} \cdot \sum_{k=1}^n \varepsilon_k \left(\frac{2k - n - 1}{2} \right)$$

$$\beta = \sum_{k=1}^n \varepsilon_k \left(\frac{3(n+1-2k)}{n^2 - n} + \frac{1}{n} \right)$$

Biography

Smail TIGANI : Is a network and telecommunication systems engineer and Phd Student in artificial intelligence and systems modelling. Recently, his researches focuses on the application of artificial intelligence on big data and performance analysis and optimization.

Mohammed OUZZIF : Is a Professor of Computer Science at High School of Technology of the Hassan II University. He has prepared his PHD at Mohammed V University in collaborative work field. His research interesting concerns distributed system and Formal description.

Rachid SAADANE : He is currently an Associate Professor in the Electrical Engineering Department at Hassania School of Labor Works of Casablanca, Morocco. His research interests include array of UWB channel measurements modeling and characterization, mobile and wireless communications (GSM, WCDMA, TD/CDMA, LTE and LTE-A) and finally digital signal processing for wireless communications systems. Recently, he is intensively interested to the IR-UWB physical layer for WSN and WBAN. Rachid is an active reviewer of various international conferences and journals.

References

- [1] M.M. Mohie El-Din, Y. Abdel-Aty, A.R. Shafay, *Two sample Bayesian prediction intervals for order statistics based on the inverse exponential-type distributions using right censored sample*, Journal of the Egyptian Mathematical Society, Vol. 19, No. 3, 2011. pp 102–105.
- [2] M. Hossain, Y. Muromachi, *A real-time crash prediction model for the ramp vicinities of urban expressways*, IATSS Research, Vol. 37, No. 1, 2013, pp. 68–79.
- [3] D. Ying Ying Sim, C. SiongTeh, P. Kumar Banerjee, *Prediction model by using Bayesian and cognition-driven techniques: a study in the context of obstructive sleep apnea*, The 9th International Conference on Cognitive Science, 2013, pp. 528–537.
- [4] C.-J. Cheng, S.W. Chiu, C.-B. Cheng, J.-Y. Wu, *Customer lifetime value prediction by a Markov chain based data mining model: Application to an auto repair and maintenance company in Taiwan*, Scientia Iranica, Transactions E: Industrial Engineering, Vol. 19, No. 3, 2012. pp 849–855.
- [5] M. Cavers, K. Vasudevan, *Spatio-Temporal Complex Markov Chain (SCMC) Model Using Directed Graphs: Earthquake Sequencing*, Pure and Applied Geophysics, DOI 10.1007/s00024-014-0850-7, 2014.
- [6] D. Xiong, R. Liu, F. Xiao, and X. Gao, *ProMT: Effective Human Promoter Prediction Using Markov Chain Model Based on DNA Structural Properties*, IEEE Transactions on NanoBioscience, DOI : 10.1109/TNB.2014.2327586, 2013.
- [7] D. Ying Ying Sim, C. SiongTeh, P. Kumar Banerjee, *Wind Speed and Direction Prediction for Wind Farms Using Support Vector Regression*, The fifth International Renewable Energy Congress IREC, 2014, March 25 - 27, Hammamet, TUNISIA.
- [8] J. Guo-xun, X. Sheng-ming, H. Xian-wei, L. Chuang-qi, *Research on the Prediction of Gas Emission Quantity in Coal Mine Based on Grey System and Linear Regression for One Element*, First International Symposium on Mine Safety Science and Engineering, pp 1585–1590, 2011.
- [9] S. Tigani, M. Ouzzif, A. Hasbi, *Monte Carlo Simulation based Algorithm Design for Automatic Learning Machine Performance Analysis*, Fifth International Conference on Next Generation Networks and Services (NGNS) 28-30 May 2014, Casablanca, Morocco.
- [10] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, *Traffic Flow Prediction With Big Data: A Deep Learning Approach*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, Vol. PP, No. 99, 2011. pp 1–9.
- [11] K. Slavakis, Georgios B. Giannakis, and G. Mateos, *Modeling and Optimization for Big Data Analytics*, IEEE SIGNAL PROCESSING MAGAZINE, Vol. PP, No. 99, September 2014. pp 18–31.