Current Trends in Extended Classifier System

Zeshan Murtza
Faculty of Computer Science and Mathematics
University of Passau
Passau, Germany
murtza01@gw.uni-passau.de

***Abstract*—Learning is a way which improves our ability to solve problems related to the environment surrounding us. Extended Classifier System (XCS) is a learning classifier system that use reinforcement learning mechanism to solve complex problems with robust performance. It is an accuracy-based system that works by observing environment, taking input from it and applying suitable actions. Every action of XCS gets a feedback in return from the environment which is used to improve its performance. It also has ability to apply genetic algorithm (GA) on existing classifiers and create new ones by taking cross-over and mutation which have better performance. XCS handles single step and multi-step problems by using different methods like Q-learning mechanism.**

The ultimate challenge of XCS is to design an implementation which arrange multiple components in a unique way to produce compact and comprehensive solution in a least amount of time. Real time implementation requires flexibility for modifications and uniqueness to cover all aspects. XCS has recently been modified for real input values and a memory management system is also introduced which enhance its ability in different kind of applications like data mining, control stock exchange. In this article, there will be a brief discussion about the parameter and components of XCS. Main part of this article will cover the extended versions of XCS with further improvements and focus on applications, usage in real environment and relationship with organic computing.

***Keywords*—LCS, XCS, organic computing, Q learning, genetic algorithm, mutation, crossover, XCSR, XCSF, XCSM.**

I. Introduction

Learning classifier system (LCS) combines evolutionary computation with learning methods to solve complex supervised and reinforcement learning problems. Initial Learning Classifier System (LCS) [14] was introduced by John H. Holland in 1975. The Basic idea behind this model is to specify that how can computers be programmed so that problems are solved by specifying "What is to be done?" rather than "How to do it?". Based on the initial approach by Holland, Wilson proposed a simplified and more efficient classifier system in 1995 called Extended Classifier System" (XCS).

Extended classifier system uses condition-action-prediction rules to maintain a population called classifier to identify the current knowledge about the problem to be solved [2]. Each classifier with binary problem has a condition as a string which is combination of ternary alphabets $\{0, 1, \#\}$. The symbol $\#$ is called don't care operator which indicates that the corresponding position can either match 0 or 1 that identify a portion of the problem and the action of classifier represents a part of the solution identified by the condition.

XCS applies reinforcement learning mechanism to evaluate classifier prediction and genetic algorithm to discover better classifiers by selecting, recombining, and mutating existing ones.

The goal of this work is to discuss about XCS, its component, modified versions with applications in different areas and focusing on how it works and interacts with the environment. In first section, there is a general overview about XCS. The next section will focus on further improvement and modifications in XCS. In third section, some applications and usage of XCS will be highlighted. Fourth section will cover the link of this system with organic computing and at the end there will be a conclusion and summery of this topic.

II. Overview of Extended classifier system

XCS is one of the most popular LCS which uses $Q-learning$ technique for updating its parameters i.e. prediction, prediction error and fitness value. In XCS, there is population set [P] which consists of classifiers. Each classifier has further parts i.e. condition, action, prediction, prediction error and fitness value. The fig. 1 below is representing the parts of classifier.

There are three main components of extended classifier system:

- Performance components (Matching, payoff prediction, action selection)
- Reinforcement components (attribute update, deferred credit assignment)

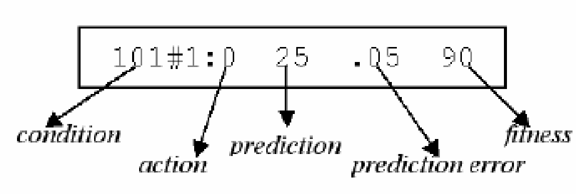

Fig. 1. An example of XCS classifier [13].

- Discovery components (covering of non-explored niches, refinement of poorly explored niches)

Every time XCS takes an input from observed environment, compares it with the set of classifiers in the population set and creates a new matching set [M] as displayed in fig. 2. Among all matching classifiers, prediction array (PA) calculates the most promising classifiers for action set [A]. For every possible action a_i, prediction $P(a_i)$ is computed to analyze the expected payoff. Action selection can be done either by choosing actions with highest prediction (deterministic) or with certain probability to the other actions which do not have null prediction (probabilistic). These actions are accomplished on environment which in return delivers a payoff or reward r. This reward r is used to update the parameters of classifier in action set [A] corresponding to the previous time stamp by using $Q{-}learning$ technique [2].

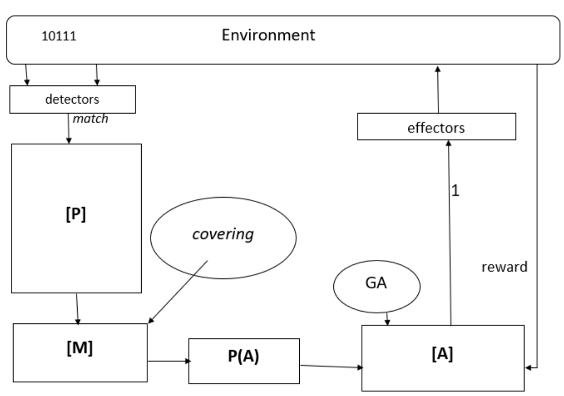

Fig. 2. XCS classifier for single-step problem [13].

In some cases, If the given input does not match with the condition in the population set [P] or the matching set [M] shows empty value then the condition is fulfilled by covering mechanism. Covering is a process which produces novel classifiers which are general enough to match with current input. The values like p, ϵ and F in new classifier are set to be predefined initial values (typically 10:0, 0:0, and 0:01) respectively.

Genetic algorithm (GA) is also a most important part in XCS which triggers when the average time of all classifiers in action set [A] is greater than θ_{GA} (often set to 50). The GA selects two parental classifiers from action set [A] using roulette-wheel selection [16] or tournament selection [17] who have higher fitness value and copied them to generate two new offspring classifiers cl_{off}. GA uses different mechanisms i.e. mutation and crossover in which values in condition part of parent classifiers are shuffled with probability to produce new off springs.

III. XCS Classifier System for Multi-objective Reinforcement Learning Problems

Most of complex real-world problems involve multiple objectives and different evolutionary techniques are used to solve these problems. Extended classifier system uses reinforcement learning technique to tackle real world multi-step issues e.g. traffic control. Unlike classifier with single-step problem which gets reward immediately by applying actions on environment, classifier with multi-step problem uses output of previous iteration to predict next iteration as follows:

$$P = r_{t-1} + \gamma * maxP(a)$$

Where r denotes the reward, t denotes the time and combination of r_{t-1} denotes the reward of previous iteration, γ is a learning factor and $P(a)$ is a system prediction. At every iteration, classifier adds the reward and makes system prediction more accurate by applying reinforcement learning.

IV. Improvements in Extended classifier system

A. XCSI: XCS Modified for Integer Inputs

Wilson proposed a model XCSI which deals with integer input values composed with range 0 to 9 [3]. The modifications are done in input interface, the mutation operator, covering, and subsumption of XCS for handling this format. In condition part, interval predicates l_i, u_i are used, where l_i (lower) and u_i (upper) are integers. Every time classifier matches input x if and only if it satisfies the condition $l_i \leq x_i \leq u_i$ for all x_i. Mutation is done by adding a value with amount $rand(m_0)$, where m_0 is a fixed integer with value range $[0, m_0]$. If a new value of l_i is less than the minimum possible input value, then the new value is set to 1 and If the new value is greater than u_i then it sets equal to u_i.

If set of classifiers do not have matching value with input parameters then the condition is fulfilled by ”covering” in which classifier manage components l_0,u_0,...l_n ,u_n where each $l_i = x_i - rand_1(r)$, but limited to the minimum possible input value and $h_i = x_i + rand_1(r)$, limited to the maximum possible input value; $rand_1$ picks a random integer from $[0, r]$, with r is a fixed integer. Subsumption of one classifier by another occurs if every interval predicate in the first classifiers condition subsumes the corresponding predicate in the second classifiers condition.

B. XCSF: XCS for real-valued function approximation system

XCSF is one of modified version of XCS, which presented by Wilson in 2002 [4]. It evolves classifiers to represent piece wise linear approximations of reward that is commonly the problem solution or the function value [5]. It is different from the original XCS at three points i.e. (1) It uses integer values just like XCSI instead of binary, (2) It uses weight vector for prediction and (3) the weight is updated instead of prediction. For every input value, XCSF uses two values for fulfilling condition. The lower limit is represented by l_i and the upper limit is represented by u_i in each interval. Weight vector is also used in condition part to make prediction by calculating it using formula given below:

$$h(x) = \omega x'$$

Where ω is the weight vector $(\omega_0, \omega_1, \omega_2, ...\omega_n)$ and x' is the input vector x augmented by a constant $(x_0, x_1, x_2, ...x_n)$. XCSF is able to approximate multi-dimensional, real-valued function surfaces from samples by locally weighted, usually linear, models [12]. It also overcome many problems like covering, solution sustenance and learning time challenge by estimating number of required mutations and the learning time and deletion strategy.

C. XCSM: Extended classifier system with internal memory

Extended classifier system uses sensor to interact with the environment for getting perception. The original XCS has no memory management system which means it performs better only in those situations when the environment is completely observable. There are some scenarios when the sensors can get complete knowledge about the environment which means the environment is completely observable and system makes predictions accurately. But in some cases, sensors only get partial observation and do not perform optimal strategy on the basis of current inputs because of incomplete information about environment.
When facing an environment with partial observation, a memory mechanism was introduced in extended classifier system [11] which consists of internal register, extending classifier with internal memory and internal action to deal with the incomplete information deriving from the sensors. Now agents take current situation from the environment and make optimal strategy by accessing past states from internal memory. XCSM is applied on partially observable environments like Woods101 and Woods102 and it learns an optimal policy in these environments.

V. APPLICATIONS OF EXTENDED CLASSIFIER SYSTEM

Extended classifier system with modified versions are used in various fields because of its higher accuracy and prediction power. In this article, Three main applications of XCS are discussed.

A. Data mining

Data mining [7] is a process of finding out patterns in data. These patterns are represented by applying rules on a given data and predict desired outcomes. Similarly, XCS [6], a learning classifier system, evolves rules (classifiers) through which the system gradually improves its ability to obtain environmental reward. Wilson applied XCSI (Extended classifier system for Integer values) to the Wisconsin Breast Cancer Database in paper [7] and made two experiments, the first was on two-dimensional data set and the other one with nine dimensions. In these experiments, Wilson gave a conclusion that XCSI has a potential to handle data with integer values ranging from 0 to 9. For a wide range of problems, XCS based systems are a very good choice for high performance combined with clear pattern discovery.

XCS is also used for distributed data mining for real-valued inputs in dynamic environments [8]. It has ability to adopt continuous environmental changes and reuse the information from the past. In distributed environment, transmission of data depends on the performance of system at client side as the client connects with the server.

B. Supply Chain Management Sales

XCSR (Extended classifier system for real values) is used to manage supply chain management [9]. TicTACtoe approach solves the TAC SCM problem with three modules, Purchase, Production and Sales.

The graphical representation of these modules are given below in Fig. 3:

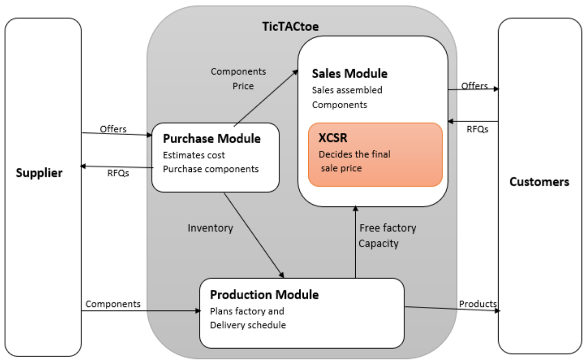

Fig. 3. TicTACtoe Architecture [9].

Each module gets information from the environment and manage a sub problem by taking its own decision and with coordination of other modules. The final decision of the product price in supply chain management is conducted by XCSR. The product is managed in such a way that it has as low price as it can win the order and at the same time as maximum as it gives profit to the customer. This decision is taken inside the Sales Module by accessing the XCSR library through two methods. The first one introduces the current state of the environment, finds the match set and the action that should take effect. Moreover, it associates the action set to the RFQ, in order to reward it later. The second one rewards the action set and saves the error information to compute further population statistics [9].

C. Identifying Trade Entry and Exit Timing

Extended classifier system (XCS) uses mathematical technical indicators to identify the timing of the market exit and financial time series forecasting. These indicators define the algorithm in such a way that the signals are free of errors and can easily be tested on large amount of data. HXCS (Hierarchical configuration of XCS agents) discovered by Gershoff [10] takes input from technical indicators and learn profitable rules for trading market data. Micro agents within HXCS view the state and produce signals in the form of 0 or 1 (buy or sale) and send them to the Meta agent. This Meta agent receive majority voting set of signals and confidence indicators to produce an action signal for executing on environment. The feedback of the action decides whether the price of following day will close above or below the current day. With wrong action, the feedback will be zero and if action is correct then the agents receive non zero value as payoff.

VI. Relation of XCS with Organic Computing

Organic systems have autonomous subsystem consists of sensors and actuators. These sensors interact with the environment and take input from it. This input is then provided to the control mechanism where sub components interact with each other and produce an optimal solution which will then hand over to actuators to apply on environment. Global control is not necessary in organic system because this system is capable to perform certain actions i.e. self-healing, self-adaptation, self-motivation, self-organization and it is robust enough to recover itself from disturbances. Organic system changes design time decision to run-time as the components prepare themselves to interact autonomously with unknown situations and produce outcomes in a unique pattern by displaying emergence.

XCS is used in organic system with two modifications. On one hand, the steady-state niche GA is removed from the conventional main-loop and replaced by an evolution strategy [18] for creating approximately optimal reactions offline, i.e. not directly affecting the system under observation and control. On other hand, The covering operator is modified in a way that it does not create conditions and actions in a probabilistic manner, but considers classifiers in the proximity and selects the nearest neighbours condition to be widened until the uncovered situation is encompassed [19]. Classifiers are now produced on demand instead of periodically and the covering mechanism now stresses the classifiers to generalize.

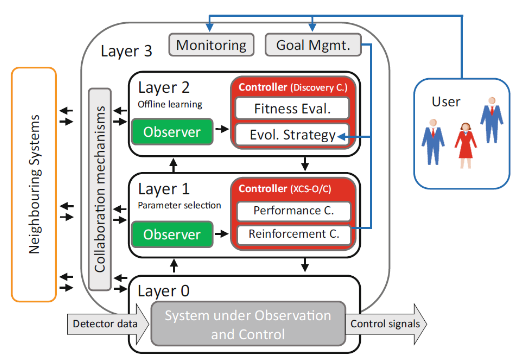

Fig. 4. Multi-layer Observer/Controller architecture for XCS-O/C [19].

Fig. 4 shows the respective parts due to XCS modification. Layer 0 represents productive system which provides access to observation and control interface. Layer 1 covers parameter selection and online learning. It gets observations from layer 0 and generates situation description by changing parameters, structure and techniques to current condition. Layer 2 covers offline learning and generate new rules when the values are missing in layer 1. Layer 3 makes collaboration by communication, negotiation, self explanation and goal modification.

VII. Conclusion and future work

This article has presented the concept and different research directions of extended classifier system with some modifications. Reinforcement learning technique is used in XCS for adaptation of different behaviors and improving itself for incoming tasks. The article began with an overview of Extended classifier system, its different components, working and interaction with environment. Paper then discussed extended classifier system for multi-input values as in real world scenarios, most of the problems have multiple objectives. To improve its performance, XCS is extended for real valued input and a memory mechanism is also applied on it to tackle those situations when the environment is partially observable. With all these modifications, XCS is used in different fields i.e, data mining, supply chain management and identifying trade entry and exit timings in stock market. In addition, this article made a relation of XCS with organic computing and discussed how it will be modified to fit in the working of organic computing. The research field of organic computing systems still generate a variety of questions that have to be answered to satisfy the rising requirements of computing systems to appear in a future not that far away.

References

[1] Larry Bull. A Brief History of Learning Classifier Systems: From CS-1 to XCS and its Variants. Evolutionary Intelligence 8(2) (2015): 55-70.

[2] Wilson, S.W.: Classifier Fitness Based on Accuracy. Evolutionary Computation 3(2), 149175 (1995)

[3] Wilson S.W. Mining oblique data with XCS. In Lanzi, P. L., Stolzmann, W., Wilson, S. W. (Eds.), Advances in learning classifier systems: Third international workshop, IWLCS 2000, LNAI 1996. Berlin Heidelberg: Springer-Verlag, 158 174, 2001.

[4] Wilson, S.W., Classifiers that approximate functions. Natural Computing, 2002. 1(2-3): p. 211-234.

[5] Lanzi, P.L., Learning classifier systems: then and now. Evolutionary Intelligence, 2008. 1(1): p. 63-82

[6] Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors. Learning Classifier Systems: From Foundations to Applications, volume 1813 of LNAI. Springer-Verlag, Berlin, 2000.

[7] Wilson S.W. Mining oblique data with XCS. In Lanzi, P. L., Stolzmann, W., Wilson, S. W. (Eds.), Advances in learning classifier systems: Third international workshop, IWLCS 2000, LNAI 1996. Berlin Heidelberg: Springer-Verlag, 158 174, 2001.

[8] Dam H.H., Abbass H.A., Lokan Ch. DXCS: an XCS System For Distributed Data Mining. In Proceedings of GECCO, 2005.

[9] Pier Luca Lanzi, Learning classifier systems: a gentle introduction Martin Butz , Stewart W. Wilson, An Algorithmic Description of XCS, Revised Papers on Genetic and evolutionary computation, July 07-11, 2010, Portland, Oregon, USA, p.147-165.

[10] Dooley, M., Schaffer, J.: Analysis of Short-Run Exchange Rate Behavior: March 1973 to November 1981. In: Bigman, D., Taya, T. (eds.) Floating Exchange Rates and State of World Trade and Payments, pp. 4370. Ballinger Publishing Company, Cambridge (1983)

[11] Lanzi, Pier Luca (1998). Adding Memory to XCS. In: To appear in the Proceedings of the IEEE Conference on Evolutionary Computation. IEEE Press.

[12] Stalph P.O., Butz M.V. (2010) Current XCSF Capabilities and Challenges. In: Bacardit J., Browne W., Drugowitsch J., Bernad-Mansilla E., Butz M.V. (eds) Learning Classifier Systems. IWLCS 2009, IWLCS 2008. Lecture Notes in Computer Science, vol 6471. Springer, Berlin, Heidelberg

[13] Wasielewska, Katarzyna Seredyski, Franciszek. (2019). Learning Classifier Systems: a way of reinforcement learning based on evolutionary techniques.

[14] Bacardit, Jaume Bernad-Mansilla, Ester Butz, Martin. (2007). Learning Classifier Systems: Looking Back and Glimpsing Ahead. 4998. 1-21. 10.1007/978-3-540-88138-4-1.

[15] Orriols-Puig, Albert Casillas, Jorge Bernad-Mansilla, Ester. (2007). Evolving Fuzzy Rules with UCS: Preliminary Results. 4998. 57-76. 10.1007/978-3-540-88138-4-4.

[16] Stewart W. Wilson. Classifier fitness based on accuracy. Evolutionary Computation, 3(2):149175, 1995.

[17] Martin V. Butz, Kumara Sastry, and David E. Goldberg. Tournament selection in XCS. Proceedings of the Fifth Genetic and Evolutionary Computation Conference (GECCO-2003), pages 18571869, 2003.

[18] H.-G. Beyer, H.-P. Schwefel, Evolution strategies A comprehensive introduction. Nat. Comput. 1(1), 352 (2002). ISSN: 1572-9796

[19] Anthony Stein, Reaction Learning, Christian Mller-Schloer, Sven Tomforde, Organic Computing Technical Systems for Survival in the Real World, 2017, pp.320-328