Eng. & Tech. Journal ,Vol.27, No.16,2009

University Admission System using Machine Learning

Lubna.Z.Bashir*

Received on:25/11/2008
Accepted on:2/7/2009

Abstract

This work examines the entrance procedure for a student seeking admission to a college institute. The system ***ASD (Artificial Student Decision)*** is a simple classifier system, which learns from the performance of the previous batches. This experience coupled with information about his aptitude enables the expert to guide the student towards the branch best suited for him. Genetics Based Machine Learning (GBML) forms our choice, as it is more human like, speculative, seeking better alternatives through the juxtaposition of hunches, inductive, using deductive procedures. Apportionment of credits involved in the evaluation of aptitude is carried out using the famous Bucket Brigade Algorithm. The tripartite process of Genetic Algorithm has been applied to make the system robust. This work addressed an important issue in student education requirement, compares and contrasts what is involved in human learning with what is involved in machine learning. The results shows In the long run for big knowledge based systems, learning will turn out to be more efficient than programming. Development the LCS by using two wildcards this increase the performance of the system.

Keywords: Genetic Based Machine Learning , Learning Classifier System , Artificial Inelegance, Apportionment of Credits , Bucket Brigade Algorithm.

نظام قبول في الجامعة باستخدام أنظمة تعليم المكننة

الخلاصة

في هذا العمل اختبرنا المدخل لطالب يبحث القرار لدخول الجامعة.النظام ***ASD (Artificial Student Decision)*** هو نظام تصنيفي بسيط ، يتعلم من الأداء السابق هــذه الخبــرة تــدمج مــع المعلومات المتوفرة عن ذكاء الطالب لتمكن الخبير من توجيه الطالب إلى أفضل كلية ملائمـــة لـــه. اخترنا أنظمة تعليم المكننة المعتمد على الخوارزمية الجينية لأنها الأكثر شبها بسلوك الإنسان، وهي تخمينية، تبحث عن أفضل البدائل و تستخدم إجراءات استنتاجية. توزيع الاعتمادية يقيم ذكاء الطالـــب باستخدام خوارزمية (Bucket Brigade)، الخوارزمية الجينية طبقت لجعل النظام أكثر كفـــاءة.هـــذا العمل وضح نتائج مهمة في قبول الطلبة، وتمت المقارنة بين ما تحقق في تعليم الإنسان مع ما تحقـــق في تعليم المكننة. النتائج بينت عند تنفيذ أنظمة تضم معلومات كثيرة يكون التعليم اكثـــر كفـــاءة مـــن

***Building and Construction Engineering Department, University of Technology / Baghdad**

https://doi.org/10.30684/etj.27.16.12
2412-0758/University of Technology-Iraq, Baghdad, Iraq

البرمجة. طورت أنظمة التعليم التصنيفية باستخدام اثنان من رموز عدم الاهتمام أدى إلى ازدياد كفاءة النظام.

Introduction

The ability to learn is one of the most fundamental attributes of intelligent behavior. The study and computer modeling of learning processes in their multiple manifestations constitute the subject matter of machine learning. These technologies derived from artificial intelligence are developing fast in today's turbulent environment of increased specialization and competition, where understanding the human thought process is of prime importance. Learning is a multifaceted phenomenon, which includes the acquisition of new declarative knowledge, the development of motor and cognitive skills and the discovery of new facts and theories through observations and experimentation.

The more we look into the way people do things, the more it appears that intelligence is not based on having general models of the world, but rather is the result of skillfully collecting and applying large quantities of anecdotal information, that is to say heuristics, This purpose can be achieved by expert systems, integrating machine learning based on artificial intelligence techniques.

When a student clears the entrance test for admission to an college he his unaware of his own strengths and weaknesses and is easily influenced by the general perspective and is misguided into taking admission to a branch for which he may not possess the right aptitude. This being the most important decision for his career, it can play havoc with his life if not guided properly. Extensive work has been done for improving entry-level education aiming to alleviate the problems of undergraduate teaching, and improvement in teaching through a Computer–aided approach. Our aim is to propose improvements in the admissions at the entry level of undergraduate studies.

Our proposed system uses a Sample Aptitude Test (SAT), which tests a ***user*** (student) for different subjects, which are required to judge his aptitude. These subjects are stored in a ***database***, which serves as an evaluating tool of the learning system, which uses it to decide the best possible branch at the admission level. These decisions are taken by the system based on certain ground rules. People while taking decisions use their intelligence at the cognitive and the sub-cognitive level. The cognitive level intelligence is obtained from the domain experts and can be easily laid down in the form of certain rules. However the sub-cognitive level intelligence is the more difficult to deal with as the human uses it at a subconscious level and these cannot be easily listed. However these can be deduced from past experience and the machine learning system of the inference engine takes care of this aspect of the expert system. In what follows we explore the structure of the machine learning system for education in a sequential manner.

Genetic Based Machine Learning / Learning Classifier Systems

GAs have been used extensively in genetic based machine learning systems (often referred to as learning classifier systems or LCS). Learning classifier systems are simple event driven rule-based systems which use a GA to evolve their rules, the aim being to evolve better and better rules over time. The GA is applied at specific times to the rules (which become members of the current population). The new rule set is then generated using crossover and mutation operators. A classifier system also consists of a rule and message system, a reward mechanism (for successful rules) and some way of obtaining input and generating outputs. The way that a classifier system works is essentially as follows: **[Goldberg,1989/ Jakobsen. ,2004]**:

```
Generate initial rule set
Until total time elapsed do
Until epoch time elapsed do
Get inputs and place on message list
Select and fire rules
Effect output indicated by rules fired
Reward rules
End until
Apply GA to evolve strongest rules
End until
```

Classifier systems have been successfully used in a number of *event-response* problems. For example, from controlling blast furnaces to stock and shares etc. The GA in an LCS is thus searching for fitter and fitter classifiers. However, unlike in a traditional GA system the "fitness value" of a rule is not simply determined by presenting the rule to a fitness function and obtaining a value. Instead in a LCS a "competitive economy" model is used to assign credit to classifiers. LCS contains three components as illustrated in **Fig.1 [Zhou. and Purvis,2004/Hunt2002].**

1. ***Performance system (Rule and message system).***
2. ***Apportionment of credit system (Bucket Brigade algorithm).***
3. ***A rule discovery (the genetic algorithm).***

1. Rule and Message System

The basic structure of the rule and message system is presented in **Fig.2**.

To see how this rule and message system works we will assume a binary internal representation in which:

message ::= {0, 1}

condition ::= {0, 1, #} (# = wild card)

classifier ::= condition : message

The whole process is initiated by an event occurring and being detected by the “Detectors”. These translate the event into an internal representation which will be used to determine which classifier (or rule) should be fired. Next each of the classifiers in the classifier list is compared with the translated event. Any classifiers whose conditions match the event may be fired. In the above example, the first classifiers’ condition matches the event. This is because 10# matches any event in which the first two elements are 1 and 0. Thus it can match 101 and 100.

Once a classifiers condition is matched, that classifier becomes a candidate to post its message to the message list at the next time step. Whether the classifier does post its message to the message list depends on the outcome of the activation action. The activation auction depends on the classifiers value (or weighting). This value is based on the credit which has been assigned to it. There are two approaches to determining the assignment of the credit used to increase a classifiers value. These approaches are the Michigan approach (or Buck Brigade) or the Pitt approach. **[Hunt,2002/Zhou. and Purvis,2004]**

2. Apportionment of Credit / The Bucket Brigade Algorithm(BBA)

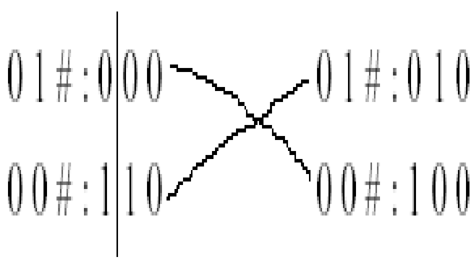
Apportionment of credit via competition and rule discovery using genetic algorithms form a reasonable basis for constructing a machine learning system atop the computationally convenient and complete framework of classifiers. The BBA service economy contains two components: an *auction* and a *clearinghouse*. When classifiers are matched, they do not directly post their messages. Instead, having its tradition matched qualifies a classifier to participate in an activation auction in which it maintains a record of its net worth, called its strength. Each matched classifier makes a *Bid* proportional to its strength in this way; rules that are highly fit (have accumulated a large net worth) are given preference over other rules. The auction permits appropriate classifiers to be selected to post their messages. The selected classifier must clear its payment through the clearinghouse, paying its bid to other classifiers for matching message rendered. A matched and an activated classifier send its bid B to those classifiers responsible for sending the messages that matched the bidding classifier’s condition. The bid payment is divided in some manner among the matching classifiers. This division of payoff among contributing classifiers helps ensure the formation of an appropriately sized sub population Thus different types of rules can cover different types of behavioral requirements without undue inter species competition**[Booker, 1982 / Odetayo,1990]**.

3. Rule Discovery

A situation may arise when the environmental message string may not find any matching classifier. The system should be robust enough to deal with such situations. This is dealt by unleashing the power of Genetic Algorithms. The tripartite process of reproduction, crossover and mutation is used to produce temporary classifiers. The fitness function for these classifiers is in accordance with the message sent.

- **Crossover**

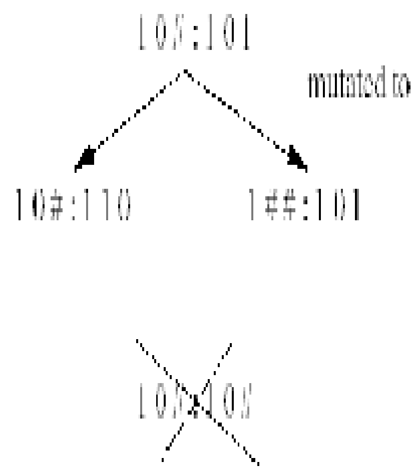
The GA crosses the classifiers in the normal way. For example:



Classifiers to be used in crossover are selected by Roulette Wheel Selection.

- **Mutation**

Mutation must now randomly select 1 of {0, 1, #} for the condition part of the classifier and 1 of {0, 1} for the action part. The action part is not allowed to have a wild card in it. For example:



- **Selection of next Generation**

now searching, not for the single best rule (classifier), but for a well-adapted set of rules. therefore use the “crowding replacement” algorithm to choose the classifiers that should die to make room for new offspring. (this implies combining the best of the parents and offspring.) Crowding replacement aims to replace a low performing classifier with a similar (potentially better classifier) **[Bull,2004/Hnt,2002]**. The crowding algorithm is:

```
for i= 1 to crowding factor do
x:=find worst of a random set
if this is not more similar to offspring
then paste x then
set worst more similar to x
end if
end for
```

replace worst most similar with offspring

Artificial Student Decision (Asd): Case Study

The system has been designed for a student who is seeking admission in a college institute. After clearing the entrance examination, the student is advised by ASD, as to which branch would be most suited for him. This purpose is achieved by measuring his aptitude through a test, which evaluates him for various subjects such as Arabic, English, Mathematics, Physics, Chemistry and Biology subjects.

Algorithm for Branch Selection

The seats available in every branch ‘B’ (N_B) are calculated and a list of the available branches are formed. Then it calculates Branch Aptitude Total (BAT) for the listed branches, which can be mathematically expressed as BAT for a branch ‘B’,

$$T_B = \frac{1}{X_{BT}} \sum_{S.} X_{BS} * S_{BT} \quad \text{............ (a)}$$

Where

B = Subscript for a particular branch.

S = Subscript for a particular subject.

X_{BT} = Total of the weighted subjects.

S_{BT} = Score obtained by student in a particular subject.

X_{BS} = Weight of subject 'S' for branch 'B'.

- Check availability of branches to a student on the basis of number of seats available.
- Read subject score of the student from the Aptitude Test Database.
- Calculate BAT in accordance with the formula given for the available branches.
- Arrange these branches, after comparison, in descending order of BAT by calling the function SORT (T_B) and display it for the user to choose from.

The **Table.1** lists the various subjects such as:, Arabic , English, Mathematics, Physics , Chemistry and Biology subjects. Along with their weight (X_{BS}), where subscripts ‘B’ and ‘S’ stand for branch and subject respectively. E.g., X_{BS} for Arabic in engineering college would be X_{ENAR} for each branch. These variable subjects are automatically updated with experience by ASD. The relevance of each subject to each branch was rated on a scale of 1-100.

For instance, an engineering college student must possess very good mathematics ability, just as a student studying medicine should have exceptional biology. This particular conclusion can be derived using genetic based machine learning.

Generation of Messages from the Environment

Table.2 shows the weights (synthetic) required for each subject in

Engineering College. To ensure that the system learns from experience, a string is generated by the following algorithm, which acts as message for Bucket Brigade Algorithm.

- Find the average of each subject from the list of passed out students of a particular branch. In our example, it is shown in **Table.3.**
- Compare the value of existing weights with that of the average performance of students.

•If (Xij >= Avg (Xij))
then S[k] = 1
else S[k] = 0
where i = branch (= Engineering in our case)
j = subject
S[k] = element of the binary string representing the message
k= 1,2,3…..l, where l = length of string.

Generation of Classifier String (Student String (St[l]))

To illustrate how ASD generates the student string we consider the score of five students of engineering collage as given in **Table.3.** If the performance of the student has been exceptionally good then we can easily introduce a 1 in the binary string, similarly a 0 can be introduced if his performance has been poor. Now the system reflects fuzziness when their performance is near to the average or there have been many students performing at the same level. So even if his performance is good it cannot be said that he is exceptional. Also, if the student's performance is below average but not poor it cannot be said that his performance can be represented by a 0. To take care of this fuzziness in the system we introduce two variables # and $.

'#' represents the performance region between average and exceptional performance and
'$' , the region between poor and the average performance. Mathematically,

S[k] = 1, when Xij > Avg.(Xij)+δj
S[k] = 0, when Xij < Avg (Xij)- δj
S[k] = #, when Avg (Xij) < Xij < Avg (Xij)+ δj
S[k] = $, when Avg (Xij)- δj < Xij < Avg (Xij)

where δj defines the fuzzy region as shown in **Fig.3**

δj can be calculated by considering the Normal Distribution(ND) or the Bell Shaped curve which is plotted from the student database taking performance as the independent axis and the number of students as the dependent axis for a particular parameter. We have taken $\delta=\sigma/2$ where σ=standard deviation.**Fig.**3 showing the ND has some useful properties. The area under the curve represents the total Number of students distributed in a performance region. The curve has its maximum height at the average performance of the students. The distribution curve runs from 0 to 100 on the performance axis. The fuzzy region on the axis ranges from (avg.X)-

δ to (avg.X)+δ where δ is proportional to the standard deviation of the sample data of student weight for a particular subject.

Generation Of Messages From The Classifier

The messages generated from the activated classifier which are responsible for sending the messages that match the bidding classifier's condition are generated by the following procedure,

The classifier string elements contain 1 or 0 or # or $ to its behavioral requirements. These classifiers send message according to its behavior as follows,

1 =1, 0 = 0 , # = 1 , $= 0, for example, 1$#0 = 1010

These messages further activate other classifiers.

Performance System of "ASD"

Performance system of the System ***ASD*** consists of a message list and classifier store. The classifier stores of ***ASD*** contain a set of rules called classifiers, which represents the knowledge and controller of the system at execution time. Condition part of classifier consists of **(6 bit),** and action part consists of **(6 bit)**. The size of classifier store for ***ASD*** will be **(4^6=4096)** Rules and all classifiers have the same strength value at the beginning.

Implementation of BBA to "ASD"

The synthetic data provided in **Table.2** represents the sample weights of subjects in Engineering College according to its requirements before learning mechanism is applied. The aim of ASD is to regularly update these weights with experiences derived from a database of passed out students. A sample database of scores for five students is given in Table.3. To generate the strings we use the algorithm explained above the procedure is given in **Table.4.**

The weights of the various subjects of Engineering College are now replaced by the weights of the winning student S_2 as illustrate in **Table.5.**

Executing of System "ASD"

Executing the ASD code, the system responds by presenting the initial and last snapshot report display in **Appendix. A** for LCS – ASD .The classifier system run for 1000 iterations, the correct rules have achieved high strength values, by contrast the bad rules have strength and bid value near zero. The classifier system eliminates the bad rule quickly there by achieving near perfect performance.

Conclusions

- A clear candidate for a cognitive invariant in humans is the learning mechanism-the innate ability to acquire facts, skills and more abstract concepts. Therefore, understanding human learning well enough to reproduce aspects of that learning behavior in a computer system is, in itself, a worthy scientific goal.

- In the long run for big knowledge based systems, learning will turn out to be more efficient than programming.
- This work compares and contrasts what is involved in human learning with what is involved in machine learning.

  1. The first obvious fact about human learning is their tediousness.
  2. Secondly, there is no copy process. In contrast, once you get a debugged program in computer, you can have as many copies as you want. When one computer has learnt it, they have all learnt it in principle.
- This work addressed an important issue in student education requirement. Apart from this, our endeavor opens a plethora of opportunities for application in various fields such as study of human nature, the effects of the changes in his environment and his response towards the same.
- Development the Learning Classifier Systems by using two wildcards # and $ this increase the performance of the system.

References

[1][Booker,L B,1982] , "Inelegance behavior as an adaptation to the task environment". Doctoral dissertation, technical report no243.Ann arbor :university of Michigan, logic of computers group. dissertations abstracts international,43(2),4698.university microfilms no.8214966.

[2][Bull.Larry,2004], "Learning Classifier Systems: A Brief Introduction", Faculty of Computing, Engineering & Mathematical Sciences University of the West of England, Bristol BS16 1QY, U.K. Larry.

[3][Goldberg, David E.,1989], "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley Longman, International Student Edition .

[4][HuntJohn,2002],"learningclassifier systems"JaydeeTechnologyLtd,HarthamPark,Corsham,Wiltshire,SN13ORP.

[5][Jakobsen. Troels,2004],"Classifier System Abstracts "Aarhus school of business ,Denmark.

[6][Odetayo Michael O,1990], " Machine Learning Using a Genetic-Based Approach". School of Mathematical and Information Sciences, Coventry University, Priory Street, Coventry CV1 5FB, UK.

[7][Zhou. Qing Qing and Purvis. Martin,2004] "A Market-Based Rule Learning System" aGuangDong Data Communication Bureau China Telecom 1 Dongyuanheng Rd., Yuexiunan, Guangzhou 510110, China, Department of Information Science, University of Otago, PO Box 56, Dunedin, New Zealand and/or improving the comprehensibility of the rules.

APPENDIX - A

```
==========================
=========
ARTIFICIAL STIDENT DECISION
(ASD)
==========================
=========
population parameters
-----------------------------
number of classifiers      =   20
numberof positions         =    6
number of action           =    6
bid coefficient          =  0.1000
bid spread               =  0.0750
bidding tax              =  0.0100
existence tax            =  0.0200
generality probability   =  0.5000
bid specificity base     =  0.2500
bid specificity mult.    =  0.1250
edid specificity base    =  0.2500
ebid specificity mult.   =  0.1250
Environmental parameters
----------------------------------
Total number of signal   =    6
Apportionment of credit parameters
---------------------------------
Bucket brigade flag  =   false
Reinforcement parameters
-----------------------
 Reinforcement reward  =   10.0
Timekeeper parameters
--------------------
Initial iteration          =     0
Initial block              =     0
Report period              =   200
Console report period      =   200
Plot report period         =   200
Genetic algorithm period   =    10

Genetic Algorithm Parameters
----------------------------
Proportion to select/gen    =  0.4000
Number to select            =       4
Mutation probability        =  0.0200
Crossover probability       =  1.0000
Crowding factor             =       3
Crowding subpopulation      =       3
Snapshot report
---------------------
[block: iteration]  -  [0:0]
current status
------------------
signal    =    000000
Decoded signal     =     0
desired output     =     0
classifier output  =     0
Environmental message:     000000
no.  strength     bid     ebid   M
classifiers
---------------------------------------------
----------------
   1     10.00     0.00     0.00
#$0000:[100000]
   2     10.00     0.00     0.00
0#$$01:[010001]
   3     10.00     0.00     0.00
$$#010:[001010]
   4     10.00     0.00     0.00
000#$$:[000100]
   5     10.00     0.00     0.00
$$$1#0:[000110]
   6     10.00     0.00     0.00
$0010#:[000101]
   7     10.00     0.00     0.00
0$01#0:[000110]
```

8 10.00 0.00 0.00
00$#11:[000111]
9 10.00 0.00 0.00
0##00$:[011000]
10 10.00 0.00 0.00
#01$$1:[101001]
11 10.00 0.00 0.00
1$####:[101111]
12 10.00 0.00 0.00
1###$$:[111100]
13 10.00 0.00 0.00
1##$$$:[111000]
14 10.00 0.00 0.00
00$$##:[000011]
15 10.00 0.00 0.00
00##$$:[001100]
16 10.00 0.00 0.00
00$###:[000111]
17 10.00 0.00 0.00
101##$:[101110]
18 10.00 0.00 0.00
101#$#:[101101]
19 10.00 0.00 0.00
101$##:[101011]
20 10.00 0.00 0.00
$010##:[001011]
new winner[1] : old winner[1]

Initial report for ASD

System[block: iteration] - [0:1000]
current status
----
signal = 100101
Decoded signal = 37
desired output = 37
classifier output = 37
environmental message: 100101
no. strength bid ebid M
classifiers
----
1 70.23 5.43 5.48 x
100#0#:[100101]
2 70.23 5.43 5.39 x
100#0#:[000101]
3 54.99 0.00 0.00
000#0#:[100101]
4 18.49 0.00 0.00
100#1#:[100101]
5 32.68 2.95 2.97 x
10010#:[101101]
6 32.68 2.53 2.58 x
100#0#:[100101]
7 10.78 0.97 1.02 x
10010#:[100101]
8 27.17 0.00 0.00
100#1#:[100101]
9 95.24 7.14 7.16 x
100#0#:[100101]
10 32.16 0.00 0.00
000#0#:[100101]
11 9.75 0.00 0.00
0#0#0#:[100100]
12 31.69 2.45 2.35 x
100#0#:[100101]
13 70.23 5.43 5.43 x
100#0#:[100101]
14 13.54 0.00 0.00
10011#:[100101]
15 77.82 0.00 0.00
100#00:[100101]
16 52.49 0.00 0.00
100#1#:[100101]
17 32.26 2.49 2.47 x
100#0#:[100101]
18 49.63 3.84 3.94 x
100#0#:[100101]

19 10.78 0.97 0.92 x New winner [9]: old winner [9]
10010#:[101101]
20 47.38 3.66 3.62 x Last report for ASD System
100#0#:[100101]

Table(1)Variable weights for subjects for different branches

	Medicine college	Engineering college	Science college	Administration &Economics college	Agriculture college	Sociology college
Arabic	X_{MEAR}	X_{ENAR}	X_{SCAR}	X_{AEAR}	X_{AGAR}	X_{SOAR}
English	X_{MEEN}	X_{ENEN}	X_{SCEN}	X_{AEEN}	X_{AGEN}	X_{SOEN}
Mathematics	X_{MEMA}	X_{ENMA}	X_{SCMA}	X_{AEMA}	X_{AGMA}	X_{SOMA}
Physics	X_{MEPH}	X_{ENPH}	X_{SCPH}	X_{AEPH}	X_{AGPH}	X_{SOPH}
Chemistry	X_{MECH}	X_{ENCH}	X_{SCCH}	X_{AECH}	X_{AGCH}	X_{SOCH}
Biology	X_{MEBI}	X_{ENBI}	X_{SCBI}	X_{AEBI}	X_{AGBI}	X_{SOBI}

Table (2)Sample weights of subjects in Engineering College

	Arabic	English	Mathematics	Physics	Chemistry	Biology
Engineering College	85	85	92	80	86	100

Table (3) Sample Scores of students in Engineering College

Students	Arabic		English		Mathematics		Physics		Chemistry		Biology	
S_1	90		93		91		86		93		98	
S_2	83		84		91		82		85		95	
S_3	89		81		90		84		86		89	
S_4	93		87		94		90		82		82	
S_5	99		91		84		95		87		73	
Average	90.8		87.2		90		87.4		86.6		87.4	
σ	5.848		4.919		3.674		5.176		4.037		10.114	
δ= σ / 2	2.924		2.459		1.837		2.588		2.018		5.057	
Avg -/+ δ	87.87	93.72	84.74	89.65	88.16	91.87	84.81	89.98	84.58	88.61	82.34	92.45

Table (4) Implementation of BBA

		T=0				T=1					
index	**classifier**	**strength**	**messages**	**match**	**bid**	**strength**	**messages**	**match**	**bid**	**strength**	**payoff**
S_1	\$1#\$11:111011	100				100				100	25
S_2	00#0\$1:001001	100		E	10	90	001001			135	
S_3	\$0#0\$#:001001	100		E	10	90	001001			90	
S_4	#\$1100:101100	100				100				100	
S_5	1101#0:110110	100				100				100	
Environment		0	001001			20					

Table (5)Comparison of weights of subjects in Engineering College

Engineering College	Arabic	English	Mathematics	Physics	Chemistry	Biology
Old weights	85	85	92	80	86	100
New weights	83	84	91	82	85	95

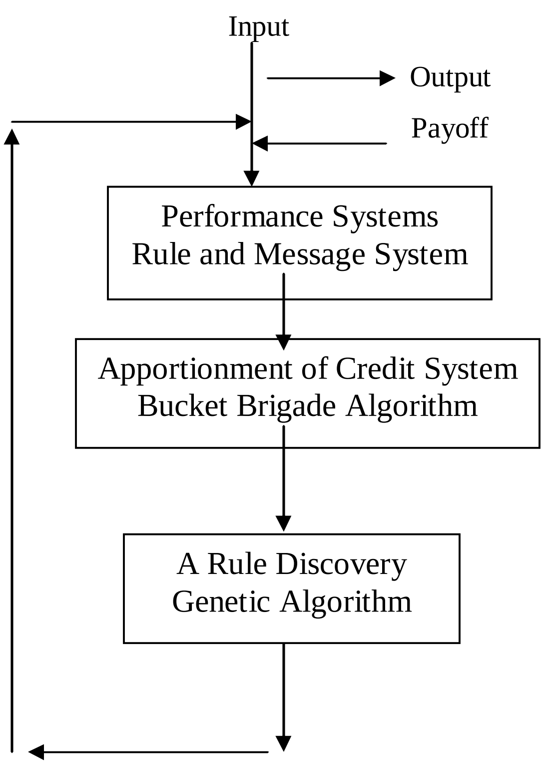


Figure (1) : The learning classifier system.

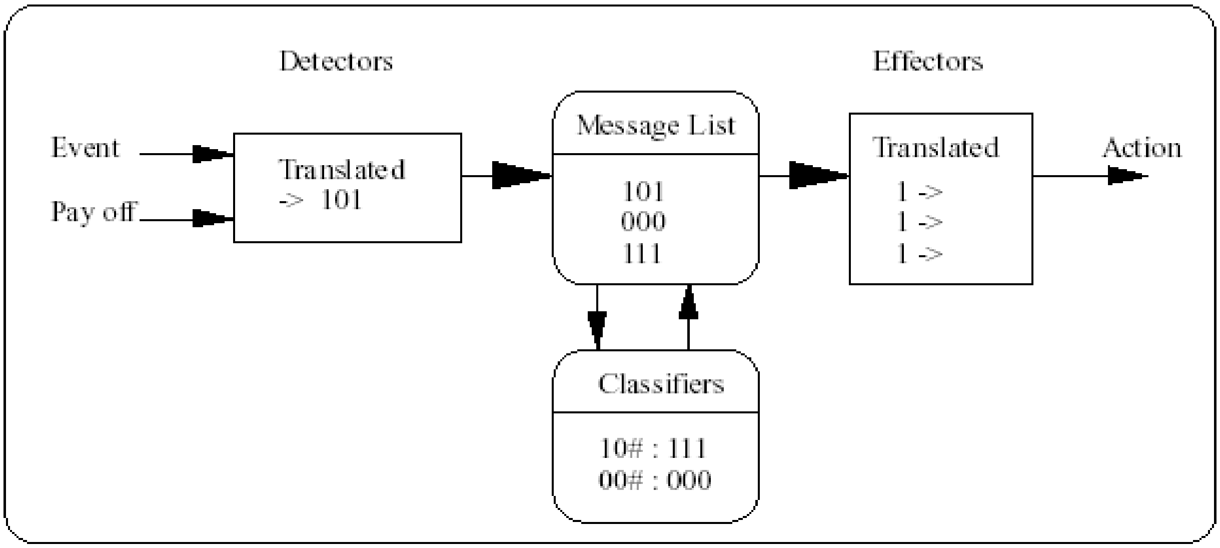


Figure (2) :Rules and Message system

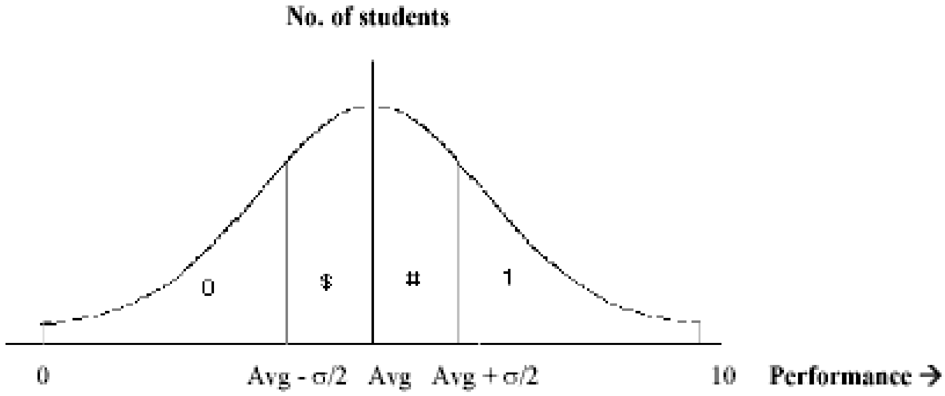


Figure (3) Student performance distribution