An Introduction to Anticipatory Classifier Systems

Wolfgang Stolzmann

2000, Leaning Classifier System: From Foundation to Application, book chapter

- Stolzmann, W., "Learning Classifier Systems using the Cognitive Mechanism of Anticipatory Behavioral Control", *First European Workshop on Cognitive Modeling. CM'96*

- Stolzmann, W., "Anticipatory Classifier Systems", *GP-98*

- Stolzmann, W., "An Introduction to Anticipatory Classifier System", *Learning Classifier System: From Foundation to Applications, 2000*, book chapter

Stimulus-Response

- S-R(Stimulus-Response) units.
- Modifiable strengths between S-R units.
- Stimulus-response connections are the basis for all behavior.
- Connectionistic models

14-unit T-maze Experiment

1930 Tolman & Honzik "Introduction and removal of reward, and maze performance in rats", University of California Publications in Psychology 4, pp.257-275.

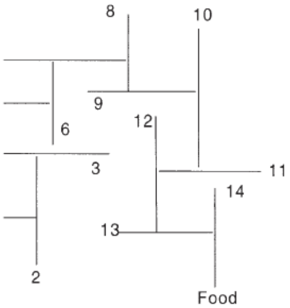
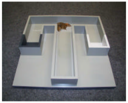


Figure 10. T maze used in Tolman and Honzik (1930). The numbers represent blinds in the maze. When a rat chose the wrong direction and entered a blind, it needed to turn around and go in a different direction to go to the next T junction.

Emergence of S-R-S Units

1932 Tolman "Purposive behavior in animals and men"

- Postulated that S-R-S units instead of S-R units.



1949 Seward "An Experimental Analysis of Latent Learning", Journal of Experimental Psychology 39, pp.177-186.

- Simple T-maze experiment.
- Confirms Tolman's theory.

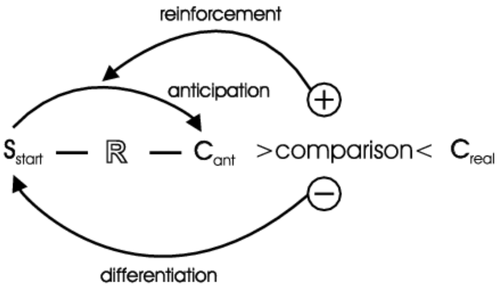
Further Question. . .

Tolman and Seward left an important question unanswered:

- How these S-R-S units are formed?
- The learning mechanism?

Anticipatory Behavioral Control

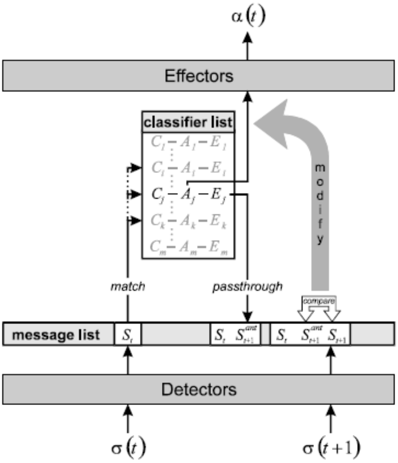
1993 Hoffmann: theory of anticipatory behavioral contorl



Basic Components of ACS

Four basic components of ACS:

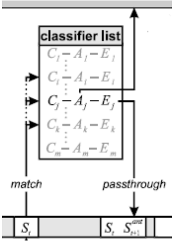
1. input interface
2. output interface
3. classifier list
4. message list



ACS Classifier

Three parts of ACS classifier:

1. Condition part C
   - $C \in \{0, 1, \#\}^n$
   - #: DON'T CARE
2. Action part A
3. Expectation part E
   - $E \in \{0, 1, \#\}^n$
   - #: PASS-THROUGH



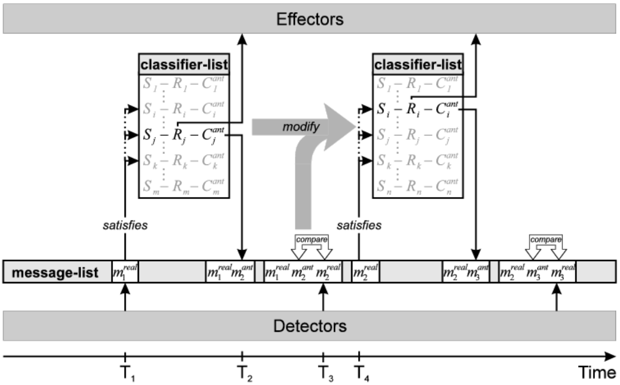
Ex. C-A-E: 0#0#-10-###1

- By 0101 $\Rightarrow$ 0101-10-0101
- By 0000 $\Rightarrow$ 0000-10-0001

Two strength values:

1. q: accuracy of anticipation
2. r: reward from environment

Behavioral Acts



Anticipation-Learning

Basic Idea:

If S_{t+1} was anticipated correctly (S_{t+1} matches S_{t+1}^{ant}), then the quality q should be increased.

If the anticipation was wrong, then generate a new classifier that anticipates S_{t+1} correctly.

If it is not possible to generate such a classifier, then the quality q should be decreased.

In detail four cases are distinguished. . .

Anticipation-Learning: S_{t+1} matches S_{t+1}^{ant}

1. *useless case:*
   - If the behavioral act does not change anything $(S_t{=}S_{t+1})$.
   - q becomes smaller: $q_c(t{+}1) = (1\text{-}b_q){\cdot}q_c(t)$
2. *expected case:*
   - If the behavioral act changes something $(S_t{\neq}S_{t+1})$.
   - q becomes larger: $q_c(t{+}1) = (1\text{-}b_q){\cdot}q_c(t){+}b_q$

$*$ $b_q \in [0,1]$: bid ratio for update the quality q

Anticipation-Learning: S_{t+1} doesn't matches S_{t+1}^{ant}

Let the differences of S_{t+1} and S_{t+1}^{ant} be represented as

$$D(i) = \begin{cases} 0, & \text{if } S_{t+1}(i) = S_{t+1}^{ant}(i); \\ 1, & \text{if } S_{t+1}(i) \neq S_{t+1}^{ant}(i). \end{cases} \quad \text{and } i \in \{1, \ldots, \text{n}\},$$

which means

$$D(i) = \begin{cases} 0, & \text{if the } i\text{th components are the same;} \\ 1, & \text{if the } i\text{th components are different.} \end{cases}$$

Anticipation-Learning: S_{t+1} doesn't matches S^{ant}_{t+1}

3 *correctable case:*

If ...

- the behavioral act changes something ($S_t \neq S_{t+1}$), and
- for all i such that $D(i) = 1$, $C(i) = E(i) = \#$

then create a new classifier $c_{new} = C_{new}$-A_{new}-E_{new} with ...

- $A_{new} = A$
- $C_{new}(i) = \begin{cases} S_t(i), & \text{if } D(i) = 1; \\ C(i), & \text{if } D(i) = 0. \end{cases}$
- $E_{new}(i) = \begin{cases} S_{t+1}(i), & \text{if } D(i) = 1; \\ E(i), & \text{if } D(i) = 0. \end{cases}$

Anticipation-Learning: S_{t+1} doesn't matches S_{t+1}^{ant}

4 *not correctable case:*

- If not possible to correct the classifier.
- q becomes smaller: $q_c(t+1) = (1\text{-}b_q) \cdot q_c(t)$

$*$ $b_q \in [0,1]$: bid ratio for update the quality q

Reward-Learning: Bucket Brigade Algorithm

Let c_t be the active classifier at time t and c_{t+1} be the active classifier at time $t{+}1$,

- If $\rho(t{+}1) \neq 0$, then $r_{c_t}(t{+}1) = (1 - b_r) \cdot r_{c_t}(t) + b_r \cdot \rho(t{+}1)$
- If $\rho(t{+}1) = 0$, then $r_{c_t}(t{+}1) = (1 - b_r) \cdot r_{c_t}(t) + b_r \cdot r_{c_{t+1}}(t)$

where $\rho(t)$ is the environmental reward at time t and b_r is the bid ratio for update the reward r.

Tasks of ACS

In each environment an ACS has 2 tasks:

1. To learn the shortest path to the goal state.
2. To learn an internal model of the environment.

Woods Environment & Performance Measure

1. Average steps to food:

   Optimal: 1.8

   (1+1+2+2+3)÷5 = 1.8

2. Achieved knowledge:

   Reliable classifiers: $q_c \geq \theta_r$

   Reliable anticipations.

T	T	T	T
T	$_2$	F$_1$	T
T	$_3$	T	T
T	$_4$	$_5$	T
T	$_6$	T	T
T	T	T	T

MazeF1

Classifier Encoding

Classifier: C - A - E

- C: $\{$t(tree), b(blank), f(food), $\#\}^8$
- A: $\{\leftarrow, \rightarrow, \uparrow, \downarrow, \nearrow, \searrow, \nwarrow, \swarrow\}$
- E: $\{$t(tree), b(blank), f(food), $\#\}^8$

Initial Classifier List

#########	↑	#########
#########	→	#########
#########	↓	#########
#########	←	#########
#########	↗	#########
#########	↘	#########
#########	↙	#########
#########	↖	#########

T	T	T	T
T	2	F_1	T
T	3	T	T
T	4	5	T
T	6	T	T
T	T	T	T

MazeF1

In position 2 of MazeF1, apply c: $\begin{array}{ccc} \# & \# & \# \\ \# & & \# \\ \# & \# & \# \end{array} \downarrow \begin{array}{ccc} \# & \# & \# \\ \# & & \# \\ \# & \# & \# \end{array}$

$$S_t: \begin{array}{ccc} t & t & t \\ t & & f \\ t & b & t \end{array} \quad S_{t+1}^{ant}: \begin{array}{ccc} t & t & t \\ t & & f \\ t & b & t \end{array} \quad S_{t+1}: \begin{array}{ccc} t & b & f \\ t & & t \\ t & b & b \end{array}$$

Using case 3(correctable), get c_{new}: $\begin{array}{ccc} \# & t & t \\ \# & & f \\ \# & \# & t \end{array} \downarrow \begin{array}{ccc} \# & b & f \\ \# & & t \\ \# & \# & b \end{array}$

Performance with Different Rewards

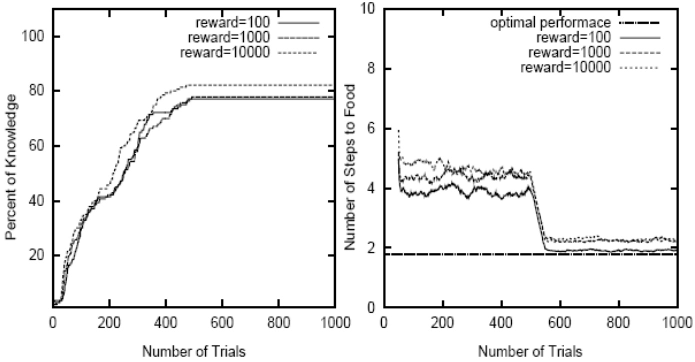


Fig. 5. Average of 10 learning experiments in MazeF1: $b_q = 0.05$, $b_r = 0.1$, $p_x = 0.5$, $\theta_r = 0.9$, $\theta_i = 0.1$, MaxSteps = 100, end_explor = 500

Performance with Different b_r's

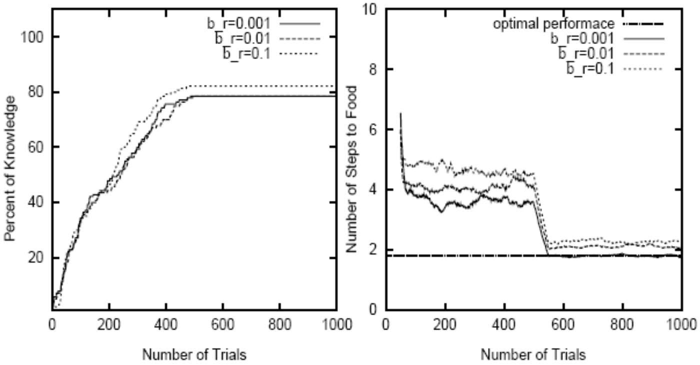


Fig. 6. Average of 10 learning experiments in MazeF1: $\rho = 10000$ in state F, $b_q = 0.05$, $p_x = 0.5$, $\theta_r = 0.9$, $\theta_i = 0.1$, MaxSteps = 100, end_explor = 500

Performance with Different end_explor's and No Reward

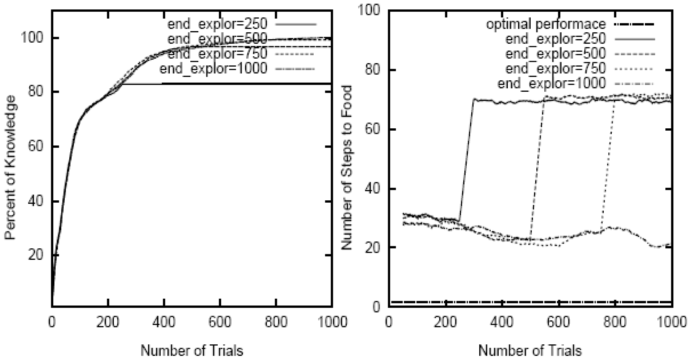


Fig. 7. Average of 100 learning experiments in MazeF1: $\rho = 0$ in state F, $b_q = 0.05$, $b_r = 0.0$, $p_x = 0.5$, $\theta_r = 0.9$, $\theta_i = 0.1$, MaxSteps = 100

Performance with Different b_q's and No Reward

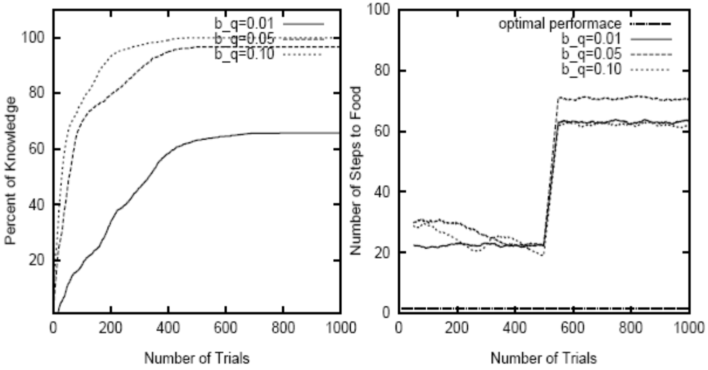


Fig. 8. Average of 100 learning experiments in MazeF1: $\rho = 0$ in state F, $b_r = 0.0$, $p_x = 0.5$, $\theta_r = 0.9$, $\theta_i = 0.1$, MaxSteps = 100, end_explor = 500

Differences between ACS and LCS

Feature	LCS	ACS
Idea	Economics	Psychology
Rule Structure	Two parts	Three parts
Rule Generation	GA	ABC
Direction	?	General to Specific
Strength	One type	Two types
Latent Learning	No	Yes