A Summary of Some Topics: Learning Automata

B. John Oommen
Chancellor’s Professor
Carleton University, Canada
Life Fellow: IEEE; Fellow: IAPR
Learning Automata

• **Learning Problem:**
  - Acquisition and utilization of relevant knowledge
  - Improve the performance of a system.

• **Learning Automaton:** Model of computer learning used to solve the learning problem
  - **Models** - Biological learning systems
  - **Goal** - Determine the optimal action from a set
  - **Optimal Action** - Has *Minimum Penalty Probability*
  - **Learns** - Process responses from random *Environment*
Learning Automaton - Learning Loop

\[ \alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\} \] - \( r \) actions

\[ \{c_1, c_2, ..., c_r\} \] - action penalty probabilities

\[ \beta = \{0,1\} \] - response from the Environment: Reward and Penalty
**Learning Automata - Learning Loop**

LA chooses one from set of actions \(\{\alpha_1, \ldots, \alpha_r\}\) offered by Environment RE

RE's response is Input to LA
Then chooses next action

RE rewards or penalizes LA
Based on penalty probabilities

Chosen action \(\alpha(t)\) is Input to the RE
Norms of Behavior

**Expedient:** Automaton better than pure-chance machine:
\[
\lim_{t \to \infty} E[M(t)] < M_0
\]

**Absolutely Expedient:**
\[
E[M(t+1) | P(t)] < M(t)
\]

**Optimal:**
\[
\lim_{t \to \infty} p_b(t) \to 1 \quad \text{with probability 1.}
\]

Note: There are no optimal learning automata.

**\(\varepsilon\)-Optimal:** A LA is said to be \(\varepsilon\)-optimal if
For any \(\varepsilon > 0\) and \(\delta > 0\), there exists \(t_0 > \infty\) and \(\lambda_0 > 0\) such that
\[
\Pr \left[ |p_b(t) - 1| < \varepsilon \right] > 1 - \delta
\]
Norms of Behavior (II)

Absolute Expediency \[ \Rightarrow \] \[ \varepsilon \]-Optimality

In all stationary random environments
Categories of Learning Automata

- **Deterministic** – Transition/Output Matrices deterministic

- **Stochastic** - Transition or output matrices are stochastic

  - **Fixed Structure Stochastic Automata (FSSA):** Transition and output matrices are *time invariant*

  - **Variable Structure Stochastic Automata (VSSA):** Transition or output matrices *change with time*
Deterministic Automata

- **Tsetlin Automaton**
- **Krinsky Automaton**

Are deterministic automata with $2N$ states and 2 actions.
Tsetlin Automaton

Favorable Response $\beta=0$

Unfavorable Response $\beta=1$

- $\epsilon$-Optimal when $\min\{c_1, c_2\} \leq 0.5$
- Ergodic: (Type of Markov Chain; Don’t worry about it)
Krinsky Automaton

Favorable Response  \( \beta=0 \)

Unfavorable Response  \( \beta=1 \)

- \( \varepsilon \)-Optimal in all stationary random environments
- Ergodic (Type of Markov Chain; Don’t worry about it)
Krylov Automaton

Action $\alpha_1$

Favorable Response $\beta=0$

Unfavorable Response $\beta=1$

- FSSA automaton with $2N$ states and 2 actions
- $\varepsilon$-Optimal in all stationary random environments
Variable Structure Stochastic Automata

- State transition probabilities or action selecting probabilities are updated with time

- Defined in terms of Action Probability Updating Schemes

- Operates on the action probability vector

\[ P(t) = \begin{pmatrix} p_1(t) \\ p_2(t) \\ \vdots \\ p_r(t) \end{pmatrix} \]

- Updates the action probability vector \( P(t+1) \) based on
  - \( P(t) \) - Previous value of the action probability vector
  - \( \beta(t) \) - Response of the Environment
Categories of VSSA

Classification based on the learning paradigm

- Reward-Penalty schemes
- Reward-Inaction schemes
- Inaction-Penalty schemes

Classification based on the properties of probability space [0,1]

- Continuous schemes
- Discrete schemes
Categories of VSSA (II)

- **Ergodic Schemes**
  - Learning Automaton does not lock in any action
  - Limiting distribution: independent of initial distribution
  - Used for Non-Stationary Random Environments
  - Example: Linear Reward-Penalty ($L_{RP}$) scheme

- **Absorbing Schemes**
  - Learning Automaton gets locked into its final action
  - Limiting distribution: dependent of initial distribution
  - Used for Stationary Random Environments
  - Example: Linear Reward-Inaction ($L_{RI}$) scheme
Exp. of Continuous Scheme: $L_{RI}(II)$

- **Action Probability Updating Scheme:**

\[
\begin{align*}
    p_1(t+1) &= p_1(t) + \lambda (1-p_1(t)) \\
    p_1(t+1) &= (1-\lambda)p_1(t) \\
    p_2(t+1) &= 1-p_1(t)
\end{align*}
\]

- if $\alpha_1$ is rewarded or $\alpha_2$ is penalized
- if $\alpha_1$ is penalized or $\alpha_2$ is rewarded
Exp. of Continuous Scheme: $L_{RI}$

- Example:

$$P(t) = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.1 \\ 0.2 \end{pmatrix}$$

$\alpha_2$ chosen and rewarded
$\lambda = 0.2$

- $p_2$ increased
- $p_1$, $p_3$, $p_4$ decreased

If $\alpha_1$ is the best action:

$$P(t+1) = \begin{pmatrix} 0.32 \\ 0.44 \\ 0.08 \\ 0.16 \end{pmatrix}$$

$$\begin{pmatrix} p_1(\infty) \\ p_2(\infty) \\ p_3(\infty) \\ p_4(\infty) \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
Thanks!!!

Thank You Very Much!!