LEARNING AUTOMATA: An Introduction (2)

By Kumpati S. Narendra
& Mandayam A. L. Thathachar

CH 1.3 & CH 2

October 7th, 2016

Derek Hommel
Master's Program in Linguistics
Computational Linguistics Lab, SNU
http://knlp.snu.ac.kr

Outline

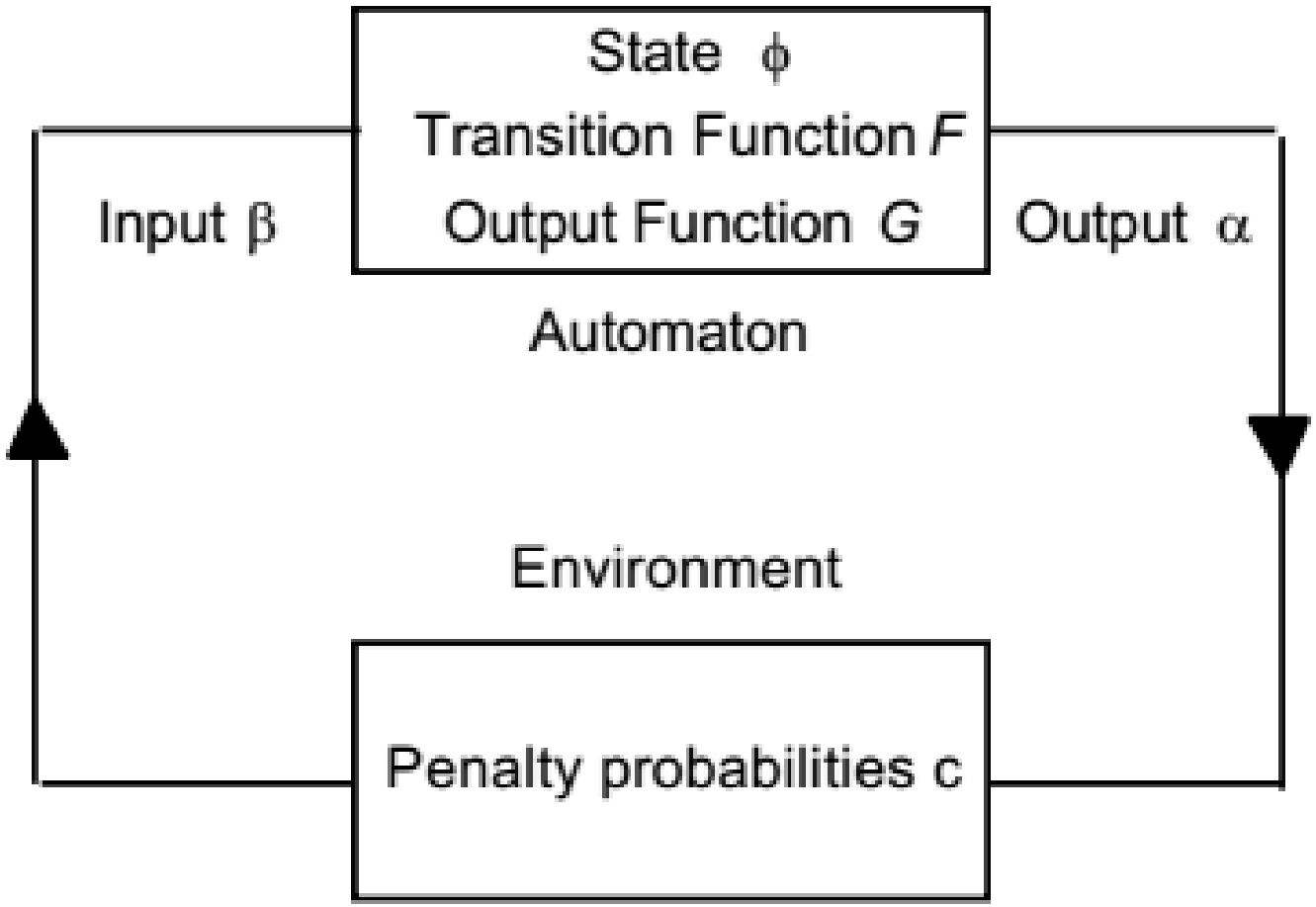
- what is an automaton?
- an analogy
- example: a basic LA
- LA and random environments
- LA and stochastic hill climbing
- LA and inductive inference
- LA and dual control
- other topics in brief
- TL;DR & Discussion Questions

What is a learning automaton?

- "The principal theme of the book is how a sequential decision maker with a finite number of choices in the action set would choose an action at every instant, based on the response of a random environment."

- a finite number of actions can be performed in a random environment.
- when an action is performed, the environment randomly responds either favorably or unfavorably.
- the choice of action should be guided by past actions and responses; its performance should improve over time.
- decisions must be made with very little knowledge concerning the nature of the environment (deterministic, stochastic, **adaptive**).

What is a learning automaton?



- Quintuple {$\underline{\Phi}$, $\underline{\alpha}$, $\underline{\beta}$, F(•,•), H(•,•)}
- states $\underline{\Phi} = \{\phi_1, \phi_3, \phi_3, \ldots, \phi_s\}$
- actions $\underline{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_r\}$ where $\alpha(n)$ is action at instant n
- responses $\underline{\beta}$ from environment:
- P: $\underline{\beta} = \{a,b\}$
- Q: $\underline{\beta}$ = finite set > 2 elements
- S: $\underline{\beta}$ = continuous interval [0,1]
- Transition function F(•,•): $\Phi \times \beta \rightarrow \Phi$
- Output function H(•,•): $\Phi \times \beta \rightarrow \alpha$
- State-Output: G(•): $\Phi \rightarrow \alpha$ (e.g. ID)

From Unsal, Cem, *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach* (1998)
c.f. *Learning Automata: An Introduction* p.52

What is a learning automaton?

- Teacher metaphor (LA p.53):
- Your professor believes in the *reinforcement learning* approach.
- He poses you a question for which you have a finite set of possible answers.
- However, when you give him an answer, he *sometimes* replies contrary to the actual reply (‘no’ instead of ‘good’).
- How can you ascertain what the correct answer is in this uncertain environment?

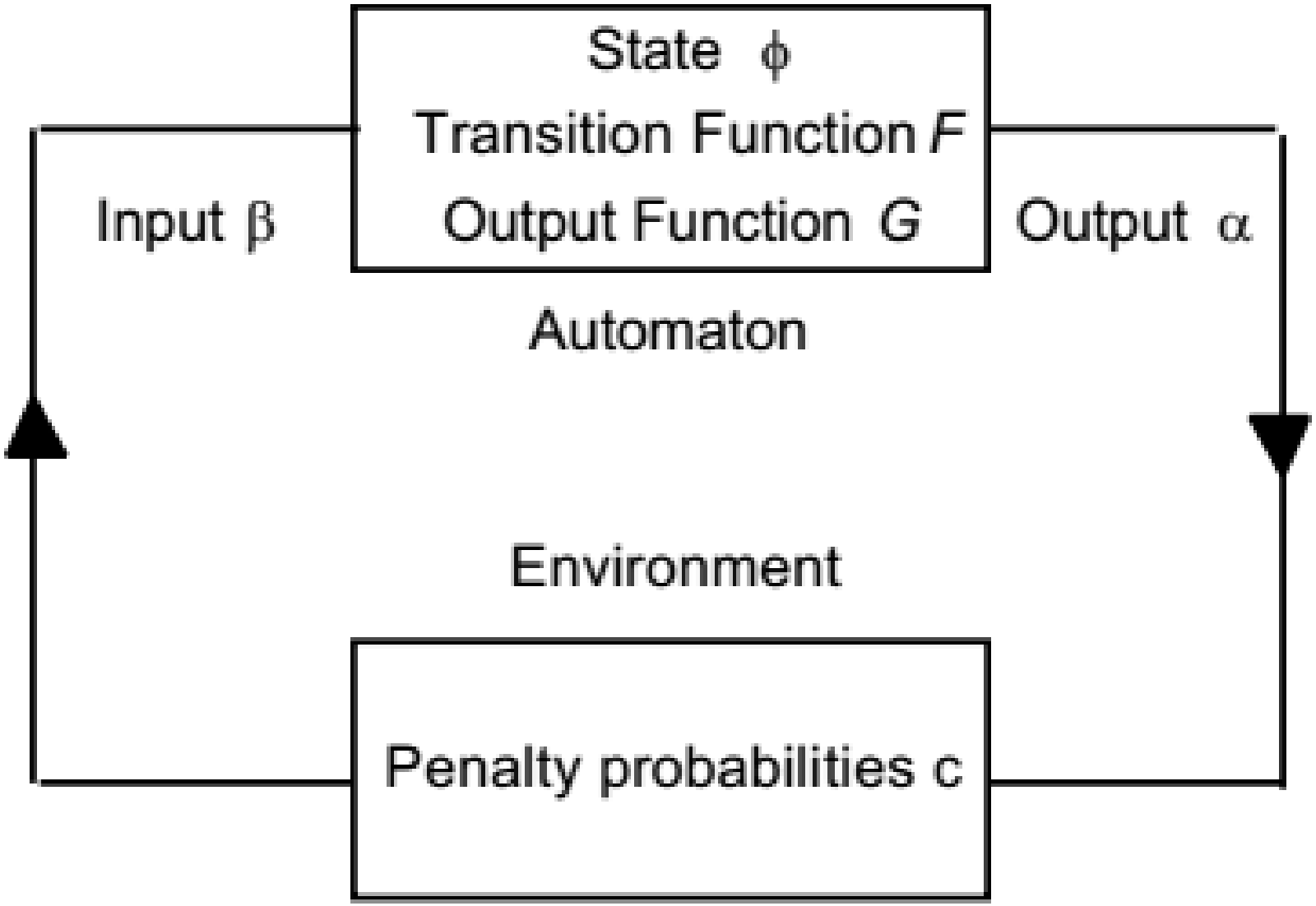
A clearer analogy

Overwatch

Overwatch bot

- Quintuple {Φ, α, β, F(•,•), H(•,•)}
- actions α = { aim, shoot, crouch, jump, run, ult, …}
- states Φ = { aim, shoot, crouch, jump, run, ult, …}
- Input β = { bad outcome, good outcome }
- F(•,•): Φ x β → Φ “given input β when in state i, go to state j”
- H(•,•): Φ x β → α “given input β when in state i, do some action”

Overwatch bot



What is a learning automaton?

- **Terms**:
- **Deterministic:** the transition from any state to another is fixed, and the output given any state is fixed
- **Stochastic:** the transition function and/or the output function has a probabilistic element; e.g. if F is stochastic, the next state is random and F gives the probabilities of moving to each other state
- **Fixed-structure:** transition and output are fixed; so in a *stochastic fixed structure* LA transitions are still random but transition and output probabilities f_{ij} and g_{ij} are fixed
- **variable-structure:** transition and output probabilities are able to be modified given input

A Basic Learning Automaton

- A very simple VSLA that ‘learns’ which item in a list is best <whatever>
  - example: finds which person is the current US President
- **P-model**: environment input is binary {0, 1} corresponding to yes-no
- **state-output**: output not determined by past states, only current one
- **stochastic:** the next state is chosen randomly according to probs in F
- **variable-structure**: this LA will update transition probabilities at each iteration, so that it reflects the current environment (at *n*)

A Basic Learning Automaton

- *In this case* since this is a state-output machine using identity matrix for G (= just output state), then we only need to consider transition function
- Here, the 'best' action does not depend on the last β input, so:

$$F(\beta_1) = F(\beta_2) \equiv F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

- Also, our next state does not depend on the current one, so:

$$F = \left[f_{11} = f_{21} \equiv f_1 \quad f_{12} = f_{22} \equiv f_2 \right] \equiv p$$

- where *p* is a probability vector and p_i is the probability in being in i^{th} state

A Basic Learning Automaton

- choose between *r* items: naïve assumption?
- Continue until done:
- Choose a ‘good’ action
- Do action and see what happens
- Given this new information, adjust my likely course of action

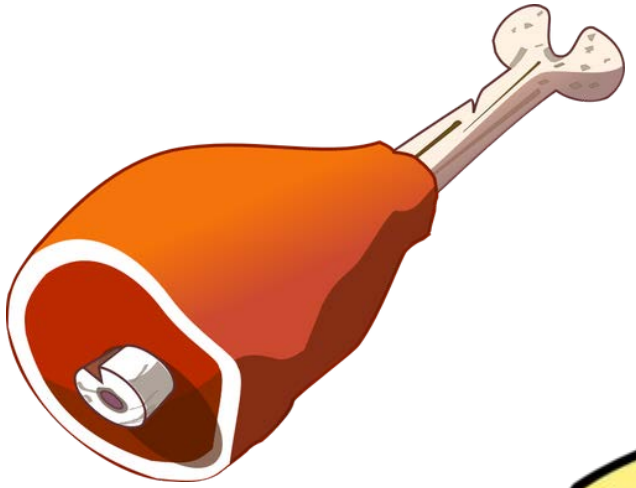
- `p = [1/r, 1/r, 1/r, …, 1/r]`
- `While not done:`
  - `select i from prob p`
  - `do i and observe beta`
  - `Update p by learning scheme`

See: Masoumi and Meybodi, *Learning automata based multi-agent system algorithms for finding optimal policies in Markov games* (2012) and Unsal, Cem, *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach* (1998)

A Basic Learning Automaton

- **General form of linear learning scheme**:
- If $\alpha(n) = a_i$:
  - When $\beta = 0$: $p_j(n+1) = (1 - a) \cdot p_j(n)$ for all $j \neq i$
  - $p_i(n+1) = p_i(n) + a \cdot [1 - p_i(n)]$
  - When $\beta = 0$: $p_j(n+1) = b/(1 - r) + (1 - b) \cdot p_j(n)$ for all $j \neq I$
  - $p_i(n+1) = (1 - b) \cdot p_i(n)$
- L_{R-P} scheme: reward & penalty params equal: a = b

Changing Paradigms



Norms of Behavior

- LA is ***expedient*** if $\lim_{n\to\infty} E[M(n)] < M_0$ where M(n) is avg penalty of α
  - That is, if the LA performs better than choosing at random (=M_0) ('pure-choice')
- LA is ***optimal*** if $\lim_{n\to\infty} E[M(n)] = c_\ell$ where $c_\ell = \min_i\{c_i\}$
  - That is, the LA trends towards choosing the best option 100% of time
- That's difficult to achieve so ***ε-optimal*** if $\lim_{n\to\infty} E[M(n)] = c_\ell + \varepsilon$
  - That is, it converges to action close to to c_ℓ. Good enough!
- LA is ***absolutely expedient*** if $E[M(n+1)|p(n)] < M(n)$ for all n, all $p_i(n) \in (0, 1)$ and for all possible sets $\{c_1, c_2, \ldots, c_r\} \rightarrow E[M(n+1)] < E[M(n)]$
  - That is, the expected average penalty gets better each iteration

Looking back to our issues…

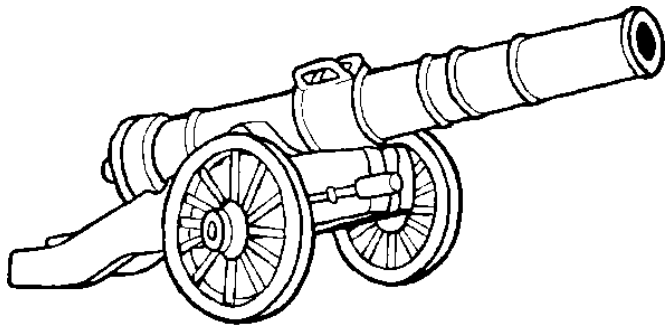
- Random environments
- Stochastic hill climbing
- Inductive inference
- Dual control
- Bayesian learning

LA & Random Environments

- **Problem**: potentially many possible actions to take
- You *could* try every option x times, get average reward/penalty, take max
- But a lot of trials wasted on undesirable actions
- Learning scheme should ensure that probability weights become concentrated on fewer alternatives during learning (inverse-H)
- LA should be able to include new actions and eliminate actions (for example if their probability drops below a certain threshold)

LA & Stochastic Hill Climbing

- **Problem**: is LA a type of hill-climbing (machine learning)?
- Usually, hill-climbing (e.g. gradient descent) is done over the *action space*; the algorithm is trying to reduce some cost function (e.g. mean-square error), essentially trying to 'choose better action' given last action



- In LA, no concept of neighborhood between actions ([?]because discrete)
- But in a (*variable-structure*) LA where output probabilities are updated iteratively, this results in monotonically increasing performance and can be viewed as hill-climbing in *probability space*

LA & Inductive Inference

- **Issue**: getting the expected answer only provides evidence for validity
- That is, we can't be unequivocal about anything found experimentally
- Learning Automata use both inductive and deductive processes:
- Given a set of prior probabilities, the LA deduces what action to take
- Then it observes the results and updates its model inductively
- [?] this iterative inductive-predictive process is similar to EM

LA & Dual Control

- **Problem**: the surgeon's dilemma between testing and operating
- $\lim_{n\to\infty}\hat{f}(n) \approx f(n)$
- We need good model but we can't afford to wait around forever
- = our model needs to get incrementally better
- For Learning Automata, this depends on our learning scheme:
- too many actions to choose from or updates too gradually: too slow
- changes too greatly given one input: may converge to wrong answer

LA & Bayesian Learning

- The learning of learning automata is similar to Bayesian learning but differs in some regards:
- While the inductive part of the LA may roughly parallel Bayesian learning, there is no close parallel to the deductive action selection.
- Various learning schemes exist; the learning scheme is a big factor in the efficacy of the learning automata

Other Topics

- LA & Psychology
  - Learning automata have been used to describe and model learning in organisms
- LA & Pattern Recognition
  - LA may be employed in pattern recognition (which has been called a type of learning), either singularly (action = categorization) or as a team of LA's, each identifying various features of a pattern to aid classification.
- LA & Algorithms, Heuristics
  - Learning schemes (input >> probabilities) are algorithms
  - The choice of learning scheme is heuristic

Notes from CH9 about LA Application

- Best when many automata, each with small number of actions, operate in distributed complex system
- Systems that might benefit from LA approach have these qualities:
  - Sufficiently complex with large uncertainties that preclude mathematical modeling
  - Must be open to distributed control (finite actions at each stage)
  - Feedback must be provided by random performance criterion **at each level**
  - small performance improvements must lead to large economic gains (realistically)
- Domains using LA: routing traffic in communication networks, scheduling computer networks, decision-making in economic networks, image processing and understanding

TL;DR

- Learning Automata model decision-making in a random environment
- Based on reinforcement learning
- Similar to previous (deterministic/stochastic) state-based models but incorporates ML and adaptive model concepts
- Parallels to the shift from Skinnerian behaviorist psychology to cognitive psychology (internal states, internal model of reality [p-vector])

Discussion Questions

- What might future applications of this model be?
- What are the potential weaknesses of this model?
- What of this model's cognitive/psychological reality?

Citations

- Masoumi, B. & Meybodi, M. R. *Learning automata based multi-agent system algorithms for finding optimal policies in Markov Games*. Asian Journal of Control 14 (1), pp.137 – 152. 2012.
- Narendra, K. & Thathachar, M. *Learning Automata: An Introduction*. Dover Publications, 2012.
- Ünsal, Cem. *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach*. Carnegie Mellon University, 1998. (Available online: https://theses.lib.vt.edu/theses/available/etd-5414132139711101/)