Parameter Learning from Stochastic Teachers and Stochastic Compulsive Liars

B. John Oommen

SCS, Carleton University

Plenary Talk : PRIS'04 (Porto)

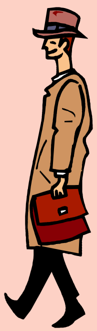
A Joint Work with

G. Raghunath and B. Kuipers

Problem Statement

We are learning from:

a Stochastic Teacher or a Stochastic Liar



Do I go … ?

Left *Right*

LIFE *DEATH*

Problem Statement

Teacher / Liar : Identity Unknown

Tells Lies — Tells Truth

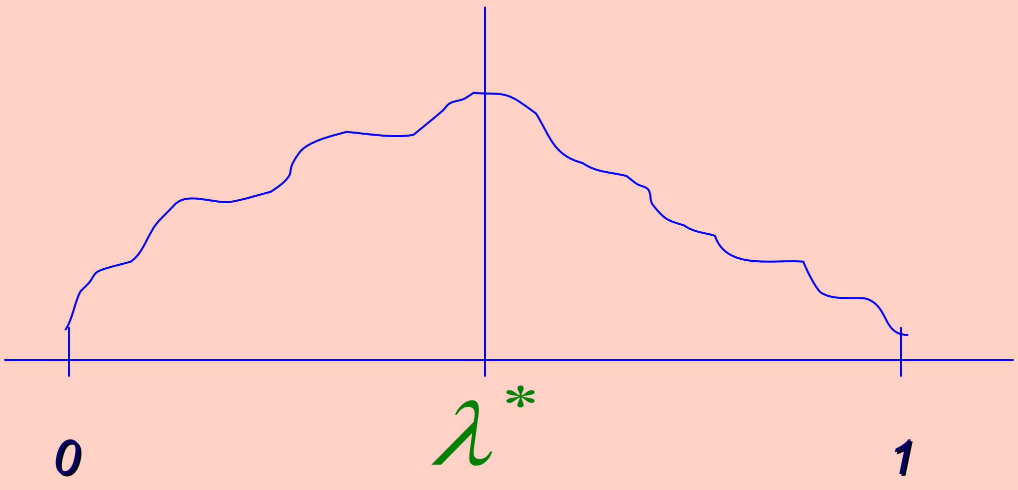
Life Gate — Death Gate

- We have to ask only ONE question,
- Go through the Gate to LIFE.

General Version of this Problem

We have :

- *A Stochastic Teacher or a Liar*
- *The Answer in [0,1] ; Any point in Interval*



General Version of this Problem

Question:

Shall we go Left or Right ?

Teacher

Go Right with prob. p

Go Left with prob. 1 - p , *where $p > 0.5$*

Liar

Do the same *with* $p < 0.5$

$$\lambda(n) \longrightarrow \lambda^*$$

Stochastic Teacher

We are on a Line Searching for a Point

Don't know how far we are from the point

We interact with a Deterministic Teacher

Charges us with how far we are from point

If Points are Integers :

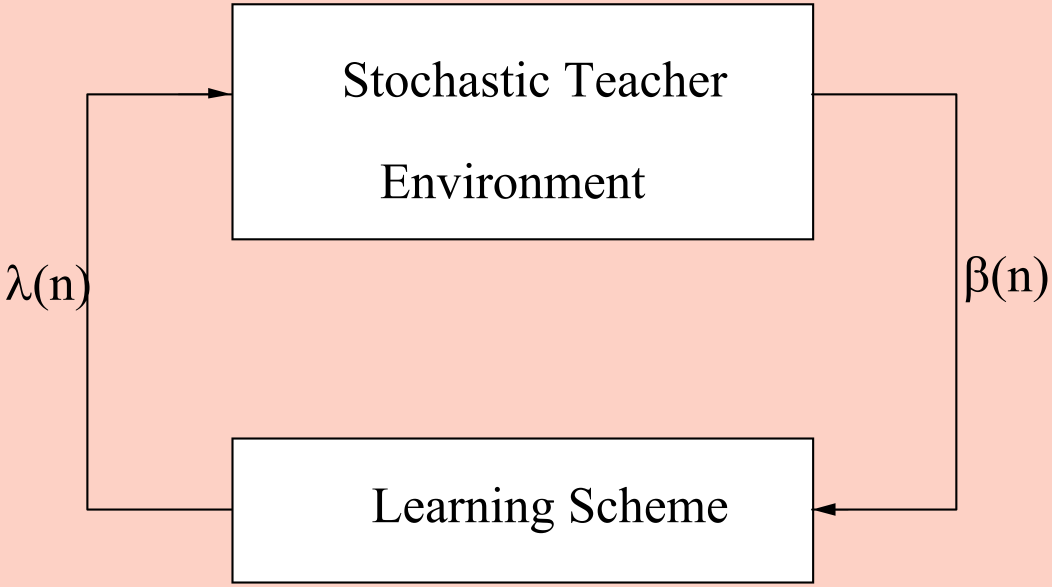
Problem can be solved in O(N) steps

Question :

What shall we do if the Teacher is Stochastic

Model of Computation

We are searching for a point $\lambda^* \in [0,1]$
We interact with a Stochastic Teacher



$\beta(n)$: Response from the Environment
- *Move Left / Right*
- *Stochastic -- Erroneous*

Model of Computation

Suppose $\lambda^* = 0.8725$

If the Current choice for λ is 0.6345

We get a response

Go Right with prob. p

Go Left with prob. 1- p

Fortunately,The Environment is Informative

i.e., $p > 0.5$

Question : Can we still learn Best parameter ?

Numerous applications:NONLINEAR OPTIMIZATION

Application to Optimization

Aim in Optimization :

Minimize (maximize) a criterion function

Generally speaking :

The algorithm works from
a "current" solution
towards the Optimal (???) solution

Based on information it currently has.

Crucial Issue:

What is the Parameter the
Optimization Algorithm
should use.

Application to Optimization

If the parameter is Too Small

the convergence is Sluggish.

If it is Too Large

Erroneous Convergence or Oscillations.

In many cases the parameter

related to the second derivative

analogous to a "Newton's" method.

Application to Optimization

First Relax Assumptions on λ

Normalize if bounds on parameter known :

$$\lambda = (\mu - \mu_{min})/(\mu_{max} - \mu_{min})$$

Thus $\lambda^* \in [0,1]$

In the case of Neural Networks Functions range of parameter varies from 10^{-3} *to* 10^{3}

Use a monotonic one-to-one mapping

$\lambda := A. \, Log_b \mu$ *for some b.*

Application to Optimization

Since λ *Converges Arbitrarily Close to* λ^*
μ *converges Arbitrarily Close to* μ^*.

Can Learn the Best Parameter in NN...

Proposed Solution

Work in a Discretized space

Discretize λ to be element of a finite set

$$\{0, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{N-1}{N}, 1\}$$

Makes steps from one value of λ to the next

Based on Response from the Environment

Advantages of Discretizing in Learning

Advantages of Discretizing in Learning

(i) Practical considerations

Random Number Generator

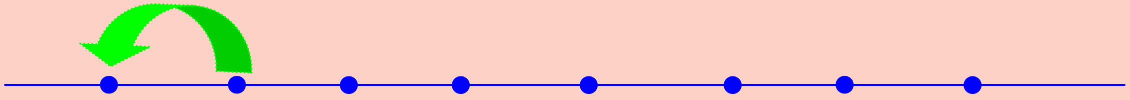
Typically finite accuracy

Action probability not any real number

(ii) Probability Changes

Jumps and not continuously.

Convergence in "finite" time possible



(iii) Proofs ε-optimal different

Discrete State Markov Chain

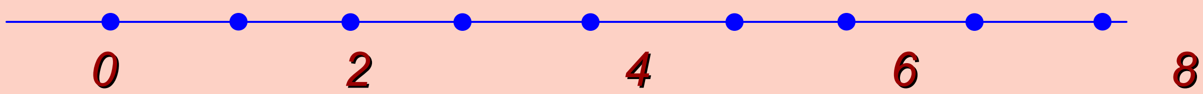
Advantages of Discretizing in Learning

(iv) Rate of convergence

Faster than continuous schemes

Increase probability to unity directly

rather than asymptotically.



(v) Reduces the time per iteration

Addition is quicker than Multiplication

Don't need floating point numbers

Generally : Discrete algorithms are superior

In terms of both time and space.

The Learning Algorithm

Assume Current Value for λ is $\lambda(n)$. Then :

In Internal State (i.e., $0 < \lambda(n) < 1$) :

If E suggests Increasing λ

$$\lambda(n+1) := \lambda(n) + 1/N$$

Else {If E suggests Decreasing λ }

$$\lambda(n+1) := \lambda(n) - 1/N$$

The Learning Algorithm

At End States :

If $\lambda(n) = 1$

If E suggests Increasing λ

$\lambda(n+1) := \lambda(n)$

Else {If E suggests decreasing λ }

$\lambda(n+1) := \lambda(n) - 1/N$

If $\lambda(n) = 0$

If E suggests Decreasing λ

$\lambda(n+1) := \lambda(n)$

Else {If E suggests decreasing λ }

$\lambda(n+1) := \lambda(n) + 1/N$

The Learning Algorithm

Note :

Rules are Deterministic
State Transitions are stochastic
Because "Environment" is stochastic.

Properties of this Scheme

Theorem I

The learning algorithm is ε-optimal.

Sketch of Proof :

We can prove that as N is increased :

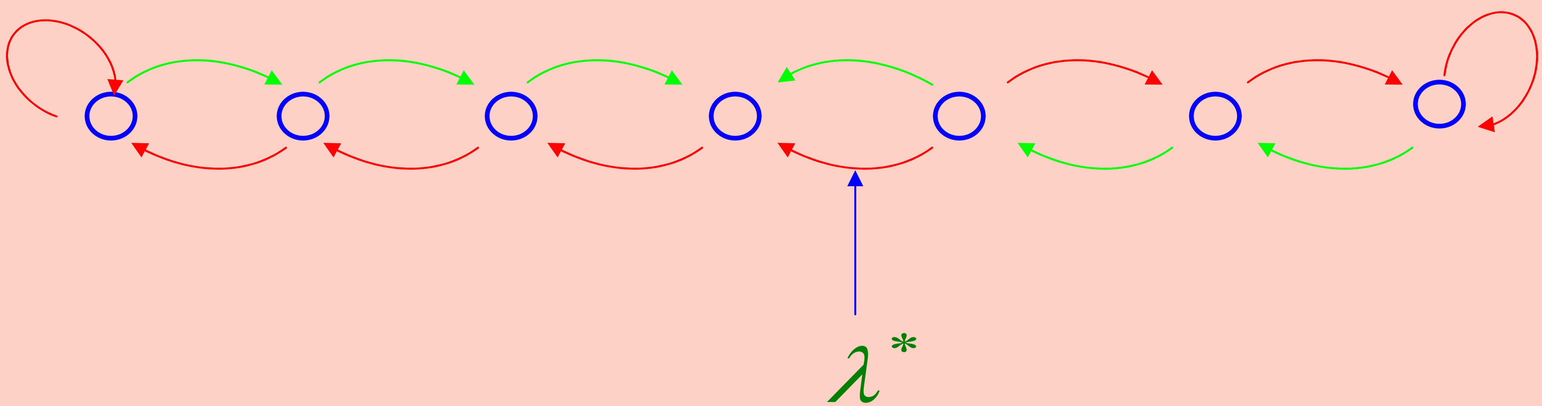
$$\lim_{N \to \infty} \lim_{n \to \infty} E[\lambda(n)] \to \lambda^*$$

The Learning Algorithm

The States of the Markov Chain

Integers {0,1,2,..., N}

State 'i' refers to the value i/N.



Good Advice - w.p. p

Wrong advice - w.p. q = 1 - p

The Learning Algorithm

Let Z be the index for which

$$Z/N < \lambda^* < (Z+1)/N.$$

Then if q = 1-p, the Transition Matrix T is :

➔ $T_{i,i+1}$	=	p	if $0 \leq i \leq Z$
➔	=	q	if $Z < i \leq N-1$.
➔ $T_{i,i-1}$	=	q	if $1 < i \leq Z$
➔	=	p	if $Z < i \leq N$.

For the self loops :

$$T_{0,0} = q$$
$$T_{N,N} = q$$

The Markov Matrix

$$\begin{bmatrix} q & p & 0 & . & . & . & 0 & 0 & . & . & 0 & 0 \\ q & 0 & p & . & . & . & 0 & 0 & . & . & 0 & 0 \\ 0 & q & 0 & p & . & . & 0 & 0 & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & p & . & . & . & . & . \\ 0 & 0 & 0 & . & . & q & 0 & p & . & . & . & . \\ 0 & 0 & 0 & . & . & 0 & p & 0 & q & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & p & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & . & . & . & . & . & . & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & 0 & q \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & q \end{bmatrix}^{T} \begin{bmatrix} \Pi_0 \\ \Pi_1 \\ \Pi_2 \\ . \\ . \\ . \\ \Pi_{Z-1} \\ \Pi_Z \\ \Pi_{Z+1} \\ . \\ \Pi_{N-1} \\ \Pi_N \end{bmatrix} = \begin{bmatrix} \Pi_0 \\ \Pi_1 \\ \Pi_2 \\ . \\ . \\ . \\ \Pi_{Z-1} \\ \Pi_Z \\ \Pi_{Z+1} \\ . \\ \Pi_{N-1} \\ \Pi_N \end{bmatrix}$$

By a lengthy induction it can be proved that :

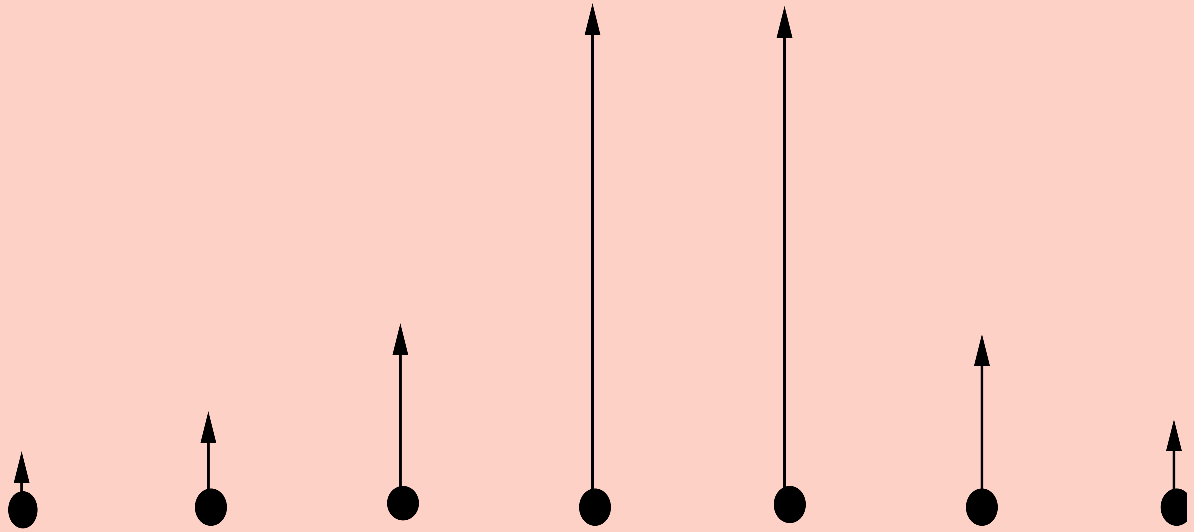
$\pi_i = e.\pi_{i-1}$ ***whenever*** $i \leq Z.$

$\pi_i = \pi_{i-1} / e$ ***whenever*** $i > Z$, ***and,***

$\pi_{Z+1} = \pi_Z.$

where $e = p/q < 1.$

The Learning Algorithm



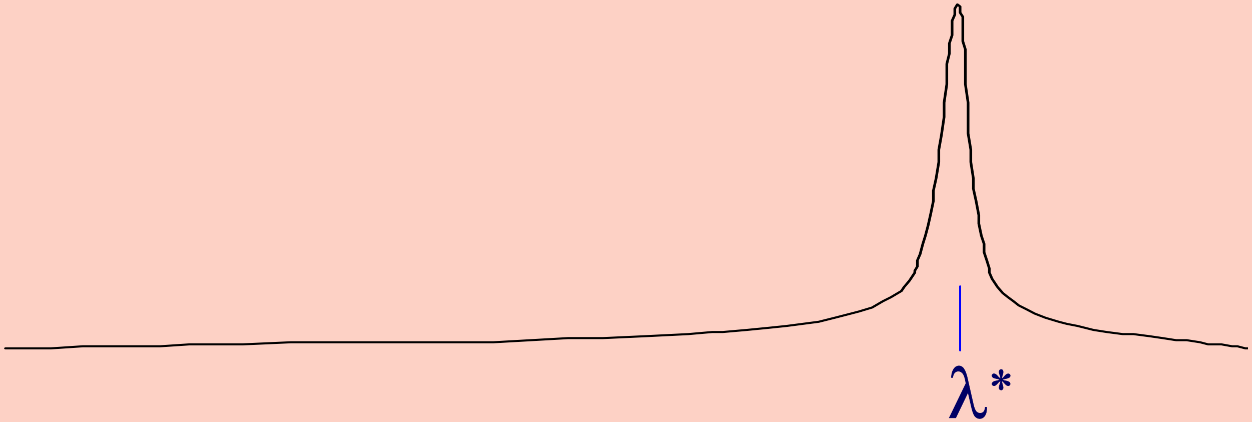
To Conclude the proof compute :

$\mathbf{E}[\lambda(\infty)]$ *as* $\mathbf{N} \rightarrow \infty$.

INCREASE / DECREASE : GEOMETRIC …...

The Learning Algorithm

As $N \to \infty$*, the Prob. Mass looks like this :*



An Increasing Geometric series

0 till Z*. Most of the mass near* Z*.*

A Decreasing Geometric series

$Z+1$ *till* N*. Most of the mass near* Z*.*

Indeed, $E[\lambda(\infty)]$ *is arbitrarily close to* λ^**.*

Example

Partition interval [0,1] into eight intervals

{0, 1/8 , 2/8 , … , 7/8 , 1) }

Suppose λ^* *is* **0.65.**

(i) All transitions for {0, 1/8, 2/8 , 3/8, 4/8, 5/8 }

Increased with prob. p

Decreased with prob. q

(ii) All transitions for {6/8, 7/8 , 1} *are :*

Decreased with prob. p

Increased with prob. p

Example

In this case, If e := p/q :

$\pi_o \leftarrow K$; $\pi_i \leftarrow K.e$; $\pi_2 \leftarrow K.e^2$

$\pi_3 \leftarrow K.e^3$; $\pi_4 \leftarrow K.e^4$;

$\pi_5 \leftarrow K.e^5$ $\pi_6 \leftarrow K.e^5$;

$\pi_7 \leftarrow K.e^4$; $\pi_8 \leftarrow K.e^3$

GEOMETRIC

Experimental Results

Table I : True value of $E[\lambda(\infty)]$ *for various* p *and Various Resolutions,* N. λ^* **is 0.9123.**

Log_2N	p = 0.70	p = 0.85	p = 0.95
2	0.7470205	0.8341304	0.8644939
3	0.8615779	0.9167632	0.9322447
4	0.885711	0.9035668	0.9060284
5	0.9164335	0.9215552	0.9218668
6	0.9137105	0.914061	0.9140624
7	0.9101543	0.9101563	0.9101561
8	0.9121094	0.9121095	0.9121093
9	0.9130859	0.9130861	0.9130858
10	0.9125975	0.9125977	0.9125975
11	0.9123535	0.9123536	0.9123535
12	0.9122314	0.9122314	0.9122314

Experimental Results

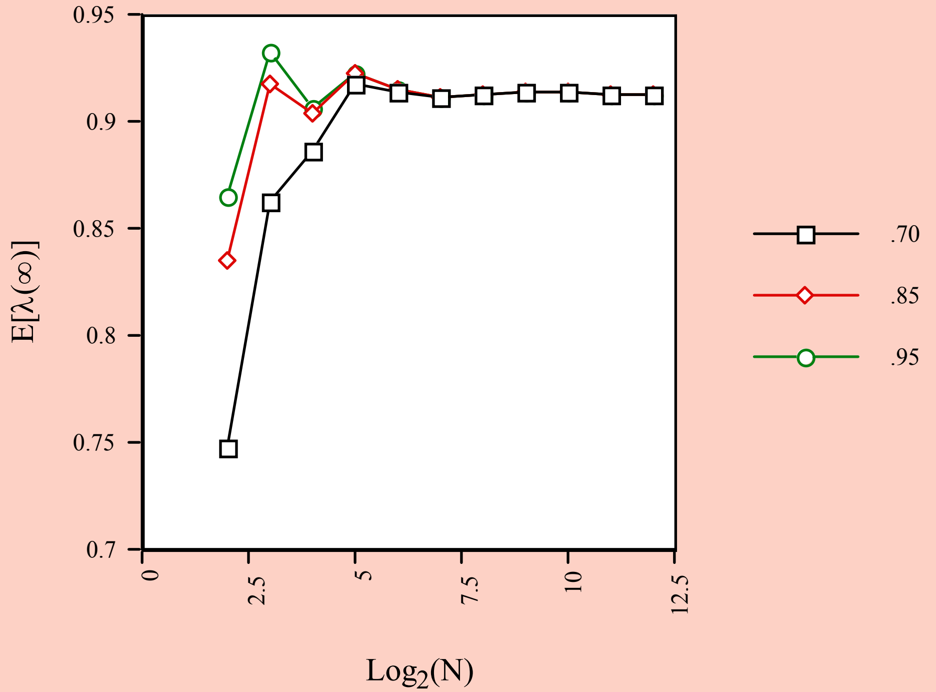


Figure I : Plot of $E[\lambda(\infty)]$ *with* N.

Experimental Results

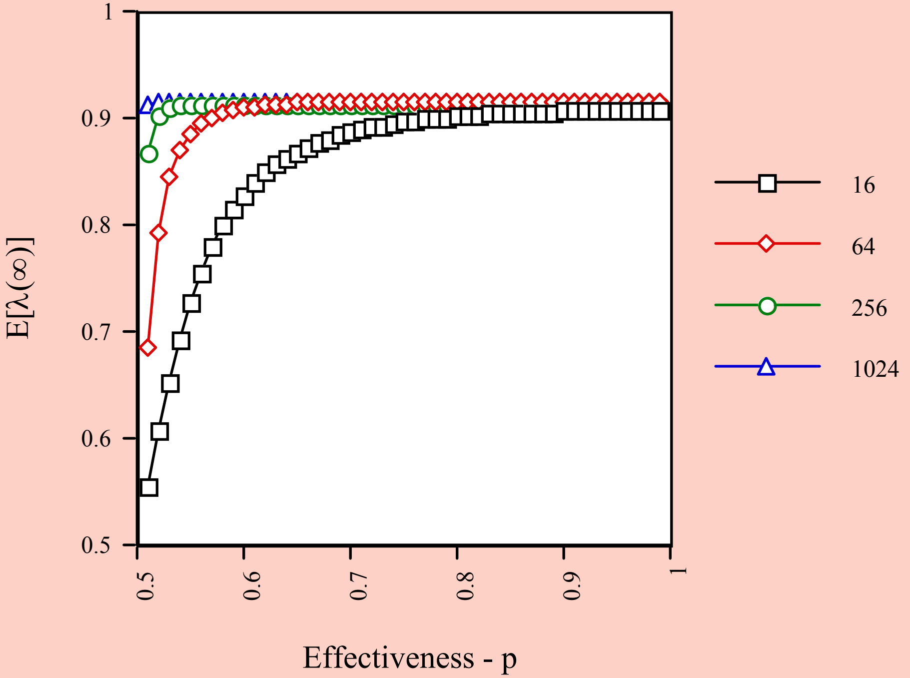


Figure II : Plot of $E[\lambda(\infty)]$ *with* p.

Continuous Solution : L_{RI} Scheme

$$\begin{bmatrix} p_1(n) \\ p_2(n) \\ p_3(n) \\ p_4(n) \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.1 \\ 0.2 \end{bmatrix}$$

If α_2 Chosen & Rewarded.

→ p_2 increased;

→ p_1, p_3, p_4 decreased linearly.

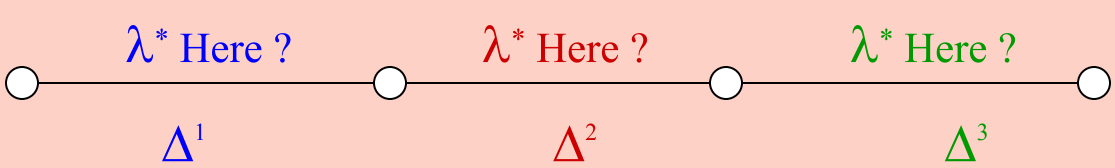
$$\begin{bmatrix} p_1(n+1) \\ p_2(n+1) \\ p_3(n+1) \\ p_4(n+1) \end{bmatrix} = \begin{bmatrix} 0.36 \\ 1-0.36-0.9-0.18 \\ 0.09 \\ 0.18 \end{bmatrix} = \begin{bmatrix} 0.36 \\ 0.37 \\ 0.09 \\ 0.18 \end{bmatrix}$$

If α_1 is the best action:

$$\begin{bmatrix} p_1(\infty) \\ p_2(\infty) \\ p_3(\infty) \\ p_4(\infty) \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

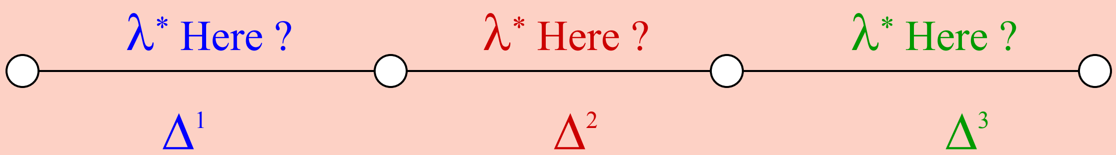
Continuous Solution

- *Systematically Explore the Given Interval*
- *Use mid-point as the Initial Guess*
- *Partition the interval into 3 sub-intervals*
- *Use ε-optimal learning in each Sub-interval*



Continuous Solution

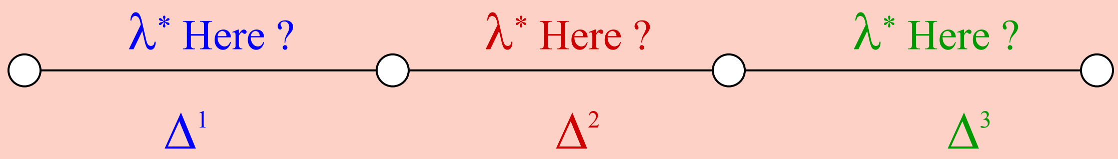
- *Is* λ^* Left, Inside *or* Right of *sub-interval??*



- *Intelligently Eliminate sub-intervals*
- *Recursively Search the remaining sub-interval(s) until the width is small enough*
- *The mid point of this interval is the result.*

Adaptive Tertiary Search

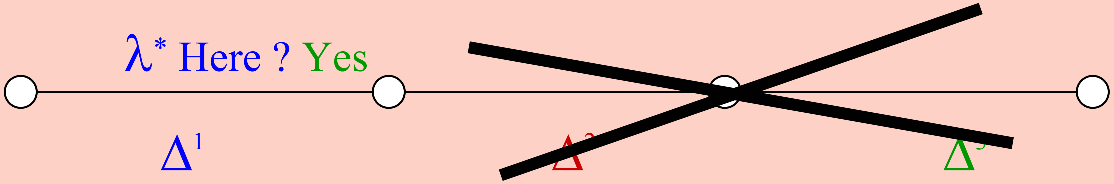
- $\Delta = [\sigma,\gamma)$: Current interval containing λ^*.



- It is Partitioned into $\Delta^{j=1,2,3}$ sub-intervals.
- λ^* is in <u>exactly</u> one of sub-intervals $\Delta^{j=1,2,3}$.
- ε-optimal Automata decides Relative Position of Δ^j with respect to λ^*.
- Since points on the real interval are Monotonically Increasing, $\Delta^1 < \Delta^2 < \Delta^3$

Adaptive Tertiary Search

- *Pruned Interval* $\Delta \supset \Delta'$ *so that:*
  - λ^* *in* Δ'
  - Δ' *is one of* $\{\Delta^1, \Delta^2, \Delta^3, \Delta^1 \cup \Delta^2, \Delta^2 \cup \Delta^3\}$



Adaptive Tertiary Search

- *Because of monotonicity of intervals*
- *Because of ε-optimality of the schemes*
- *The above two constraints indicate that*
  - *the Search converges*
  - *Search converges monotonically*

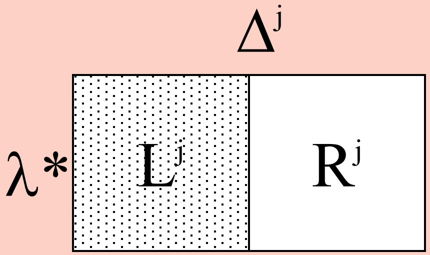
Learning Automata for Δ^j

- **Each Automaton :**
  - *Two possible actions : Left Half / Right Half*

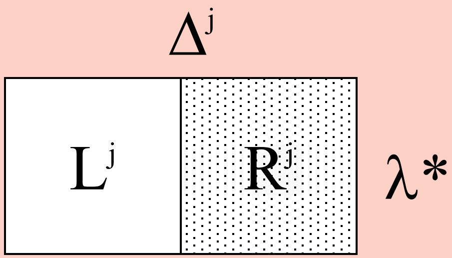
- **Uses Input from Teacher : L_{RI} Scheme**

- **Converges after One Epoch to**
  - *Left End* $\rightarrow$ $[1, \ 0]^T$
  - *Right End* $\rightarrow$ $[0, \ 1]^T$
  - *Cannot Decde (Inside)* $\rightarrow$ $[0.78, \ 0.22]^T$ (e.g.).

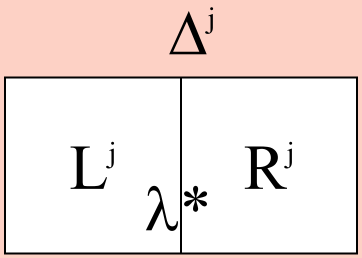
Location of Δ^j Convergence of L_{RI}



(a)



(b)



(e)

Results on Convergence

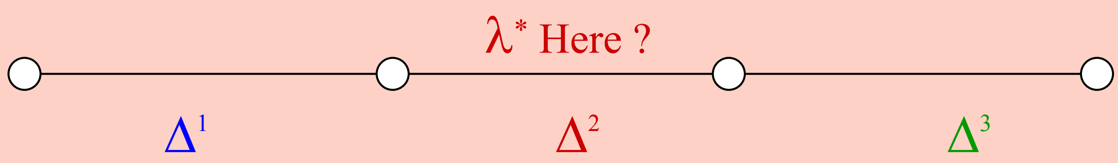
- For an Stochastic Teacher &The L_{RI}

If $(\lambda^*$ Left of $\Delta^j)$ Then $Pr(\Omega^j = \text{Left}) \rightarrow 1$

If $(\lambda^*$ Right of $\Delta^j)$ Then $Pr(\Omega^j = \text{Right}) \rightarrow 1$

If $(\lambda^*$ Inside $\Delta^j)$ Then

$Pr(\Omega^j = \text{Left, Right, Inside}) \rightarrow 1$



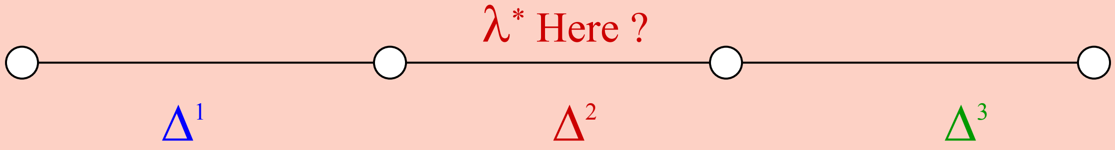
Results on Convergence

- Conversely,

  If $[\Omega^j = \textit{Left}]$ Then $Pr[\lambda^* \textit{ Left of } \Delta^j] \to 1$

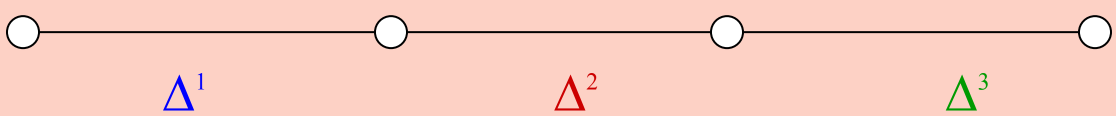
  If $[\Omega^j = \textit{Right}]$ Then $Pr[\lambda^* \textit{ Right of } \Delta^j] \to 1$

  If $[\Omega^j = \textit{Inside}]$ Then $Pr[\lambda^* \textit{ Inside } \Delta^j] \to 1$



Decision Table to Prune Δ^j

Output Ω^1 for Δ^1	Output Ω^2 for Δ^2	Output Ω^3 for Δ^3	New sub-interval Δ'
Left	Left	Left	Δ^1
Inside	Left	Left	Δ^1
Right	Left	Left	Δ^1 U Δ^2
Right	Inside	Left	Δ^2
Right	Right	Left	Δ^2 U Δ^3
Right	Right	Inside	Δ^3
Right	Right	Right	Δ^3



Convergence : Stochastic Teachers

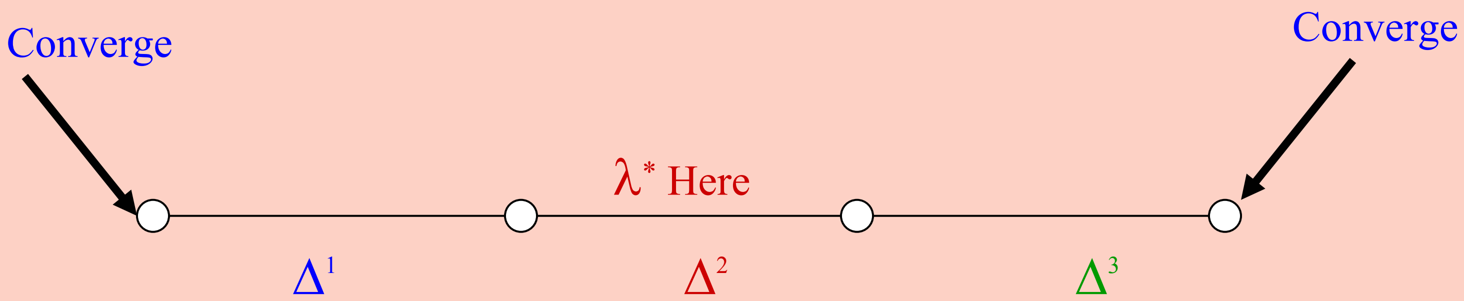
- *In the Previous Decision Table*
- *Only 7 out of 27 combinations are shown.*
- *The Rest Impossible*
- *The Decision Table to prune is Complete*
- *Pr[λ^* in the Pruned Interval] $\rightarrow 1$*

Consequences of Convergence

- Consequence : Dual Problem
  - *If E : Environment w.p. p then, E' has p' = 1-p*
  - *Dual of an Stochastic Teacher (p > 0.5) is a Stochastic Liar (p' < 0.5)*

Decision Table for Stoch. Liar

- ***How will the Stoch. Liar Teach ?***
- ***Left Machine : Converge to Left End***
- ***Right Machine : Converge to Right End***



Learning from Stochastic Liar

- Start with an Extended Search Interval

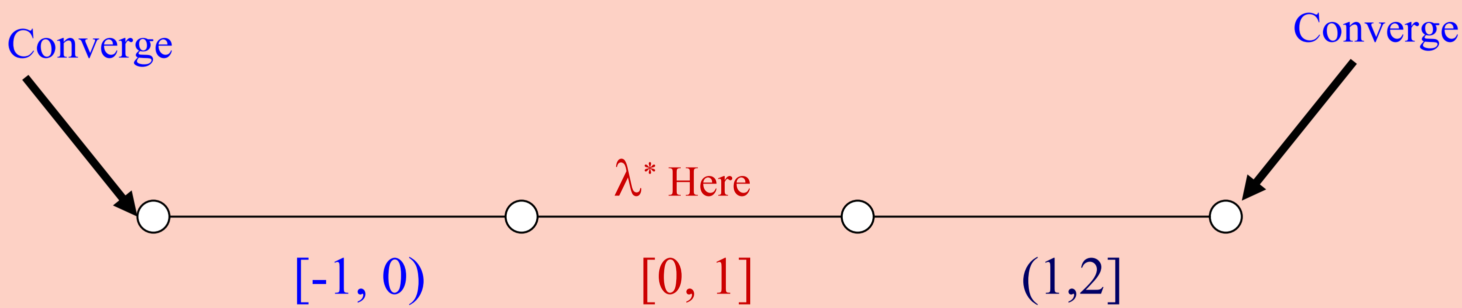
  $\Delta' = [-1,2)$ where as $\Delta = [0,1)$

- After ONE Epoch

  Liar Will Force [-1,0) or (0,2] with Prob. 1.

- **GOTCHA RED HANDED !!!!!**

- We Know Environment is Deceptive



Learning from Stochastic Liar

- KNOW Environment is Deceptive
- Use the Original interval $\Delta = [0,1)$

Treat Go Left as Go Right

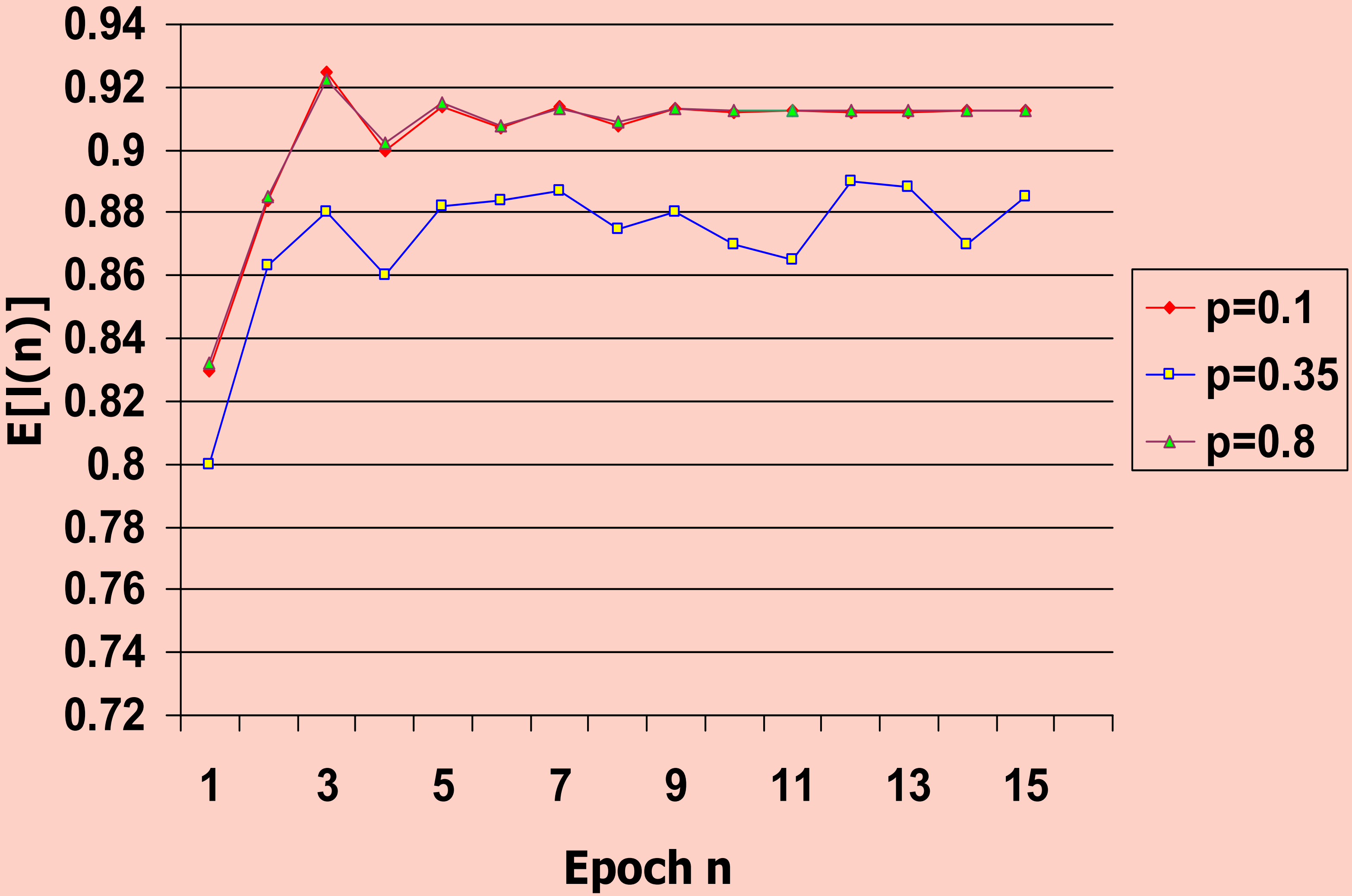
Go Right as Go Left !!!!

Experimental Results

	p	θ=0.8	θ=0.85	θ=0.9
Deceptive Environment	0.10	0.912298	0.912273	0.912194
	0.15	0.912312	0.912298	0.912222
	0.20	0.912193	0.912299	0.912236
Informative Environment	0.80	0.912317	0.912284	0.912234
	0.85	0.912299	0.912275	0.912202
	0.90	0.912302	0.912275	0.912202

Note: $\lambda^* = 0.9123$, $N_\infty = 250$, $\varepsilon = 0.005$

Convergence of CPL-ATS



Observations on Results

- Convergence : p=0.1 & p=0.8 - almost identical The former is highly deceptive environment
- Even in the first epoch ONLY 9% error
- In two more epochs the error is within 1.5%
- For p nearer 0.5 the convergence is Sluggish

Conclusions

- *Can use a combination of L_{RI} and Pruning*
- *Scheme is ε-optimal.*
- *Can be applied to Stochastic Teachers and Liars*
- *Can detect the nature of Unknown Environment*
- *Can learn the parameter correctly w. p. $\rightarrow 1$*

THANK YOU

VERY MUCH

How to Simulate Environment

Our idea : analogous to the RPROP network.

$$
\begin{aligned}
\Delta_{ij}(t) &= -\Delta_{ij}(t-1)\cdot\eta_{+} && \textit{If } {*}> 0,\\
&= -\Delta_{ij}(t-1)\cdot\eta_{-} && \textit{If } {*}< 0,\\
&= \Delta_{ij}(t-1) && \textit{Otherwise.}
\end{aligned}
$$

where η_{+} *and* η_{-} *are parameters of the scheme.*

Increments are influenced by
the sign of two succeeding derivatives

Experimental Results

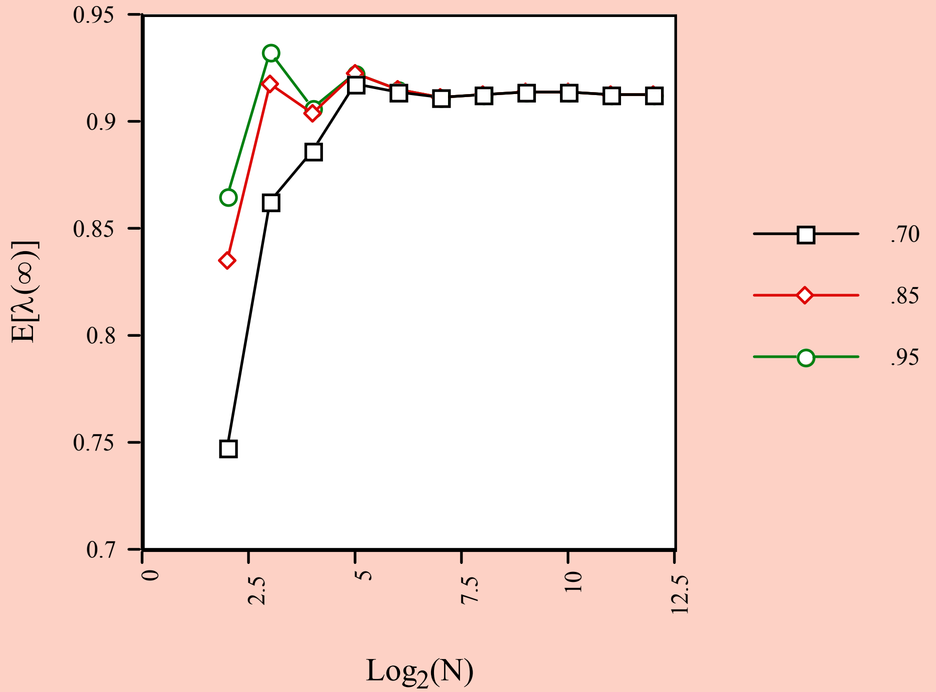
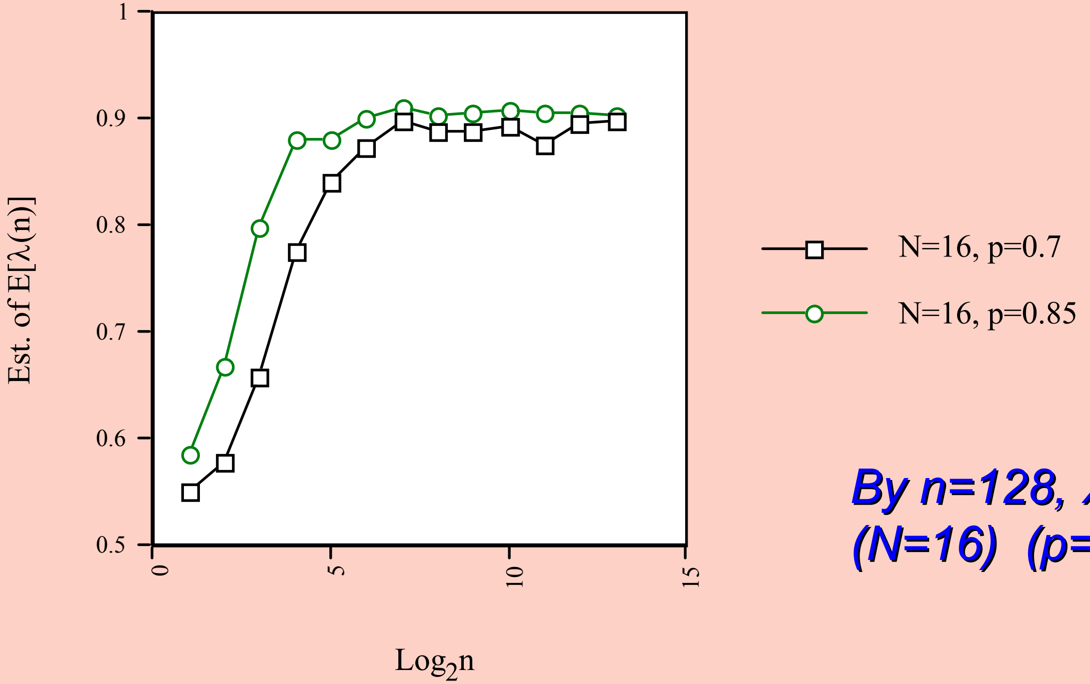


Figure I : Plot of E[λ(∞)] *with* N.

Type II Results

Simulations

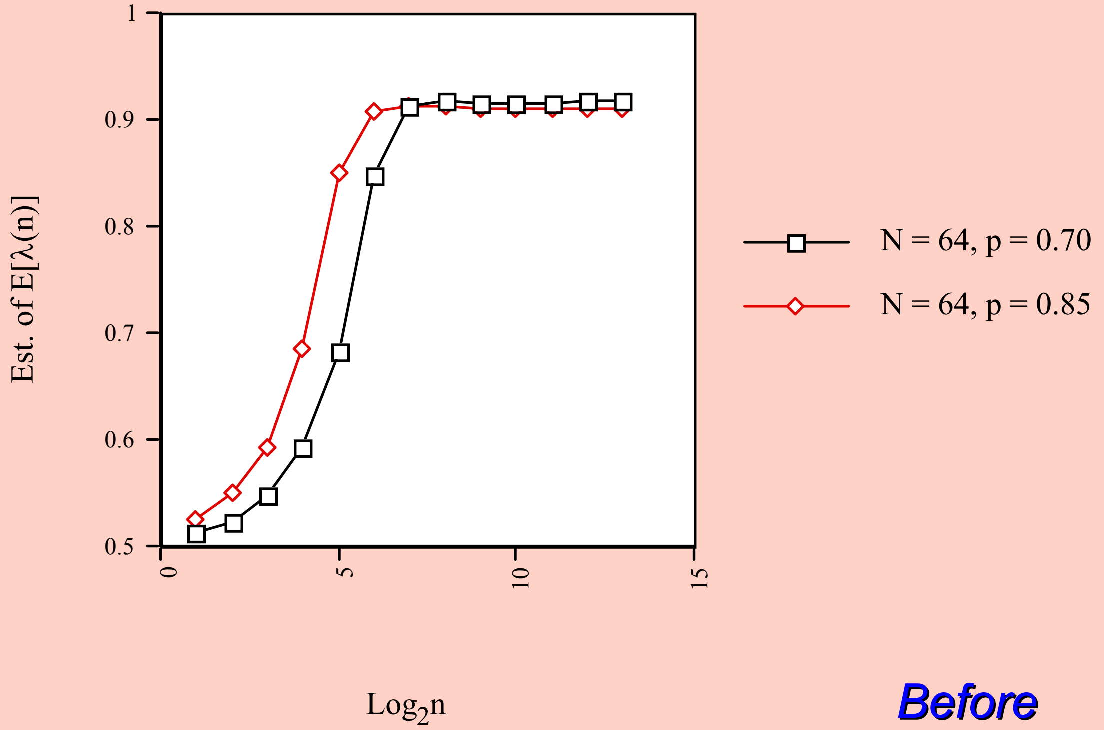
- *Hundred parallel experiments for various values of* p
- *Record ensemble average of the results*



By n=128, $\hat{\lambda}(n)$=0.910 (N=16) (p=0.7)

Figure III : ***Plot of est. of*** $E[\lambda(n)]$ ***with time,*** n, ***for*** $N = 16$. λ^* ***is*** 0.9123.

Type II Results



Before $\hat{\lambda}(n)=0.908$
(N=64) (p=0.85)

Figure IV : Plot of est. of E[λ(n)] ***with time***, n, ***for*** N = 64.
λ* ***is*** 0.9123.

Learning Automata for Δ^j (Cont'd)

- *Rule for updating action probabilities:*

  If α_k^j was rewarded,

  $P_{1-k}^j(n+1) := \theta . P_{1-k}^j(n)$

  $P_k^j(n+1) := (1-\theta) . P_{1-k}^j(n)$

  where, θ is the L_{RI} reward parameter

- *The decision output Ω for at the end of N_∞ iterations:*

  Left **if $P_0^j(N_\infty) >= 1 - \varepsilon$**

  Right **if $P_1^j(N_\infty) >= 1 - \varepsilon$**

  Inside **otherwise.**

How to Simulate the Environment

Typically, if E is the Criterion Function

Question :

When is $\delta E/\delta \mathbf{x}$ zero?

Simple Linear rule

Moves '$\mathbf{x}$' in the direction of the solution.

Second Derivative information tells

How much to move.

How to Simulate Environment

Whenever :

Partial Derivative changes sign

- ➔ *Last update was too big*
- ➔ *Jumped over a local minimum*
- ➔ *Increment is decreased by η_{-}.*

If the Derivative Retains its sign

increment is slightly increased

- ➔ *Converge faster in shallow regions*

How to Simulate environment

Same philosophy for designing E

If the Partial Derivative changes sign

Decrement value of λ

Otherwise Increment it.

Effectively attempting to "simulate"

Newton's Rule

Without Evaluating the Second Derivatives.