18734 Recitation

Course Project

Audit Games

Course Project

- Teams finalized?
- 10 teams on the doc.

- Project Idea
- Related readings

Project Proposal

- Pdf document (1-2 pages):
  - Team members
  - Motivation & Problem Statement
  - Approach
  - Deliverables & Timeline

- In-class presentation by members

Game Theory

Game Theory

- Developed to explain the optimal strategy in two-person interactions.

An example:
Big Monkey and Little Monkey

One coconut per tree.
A Coconut yields 10 Calories

Big Monkey expends 2 Calories climbing the tree.
Little Monkey expends 0 Calories climbing the tree.

An example:
Big Monkey and Little Monkey

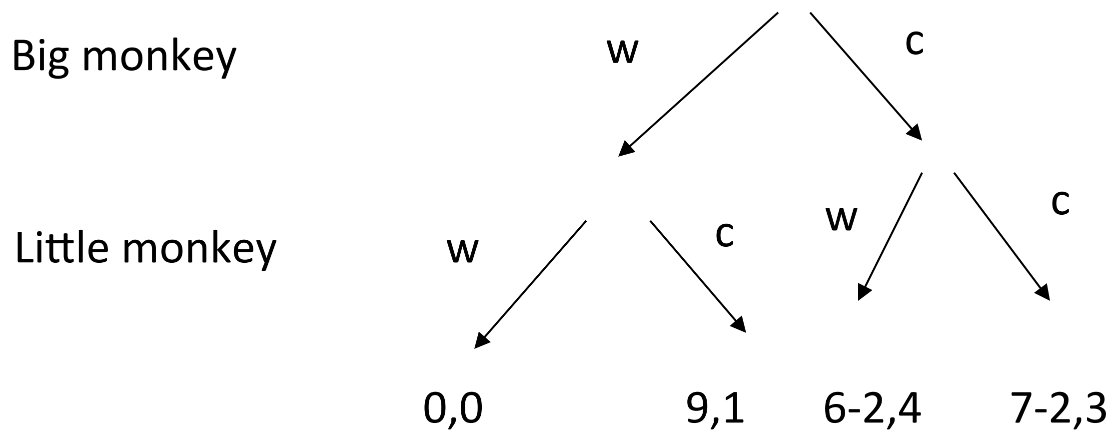
- If BM climbs the tree
  - BM gets 6 C, LM gets 4 C
  - LM eats some before BM gets down
- If LM climbs the tree
  - BM gets 9 C, LM gets 1 C
  - BM eats almost all before LM gets down
- If both climb the tree
  - BM gets 7 C, LM gets 3 C
  - BM hogs coconut
- How should the monkeys *each* act so as to maximize *their own* calorie gain?

An example:
Big Monkey and Little Monkey

- Assume BM decides first
  - Two choices: wait or climb

- LM has four choices:
  - Always wait (ww), always climb (cc), same as BM (wc), opposite of BM (cw).
  - The first letter in indicates Little Monkey's move if Big Monkey waits, and the second is Little Monkey's move if Big Monkey climbs.

An example: Big Monkey and Little Monkey



What should Big Monkey do?

- If BM waits, LM will climb – BM gets 9
- If BM climbs, LM will wait – BM gets 4
- BM should wait.
- What about LM?
- Opposite of BM (even though we' ll never get to the right side of the tree)

An example: Big Monkey and Little Monkey

Normal Form:

Little Monkey

Big Monkey		cc	cw	wc	ww
	w	9,1	9,1	0,0	0,0
	c	5,3	4,4	5,3	4,4

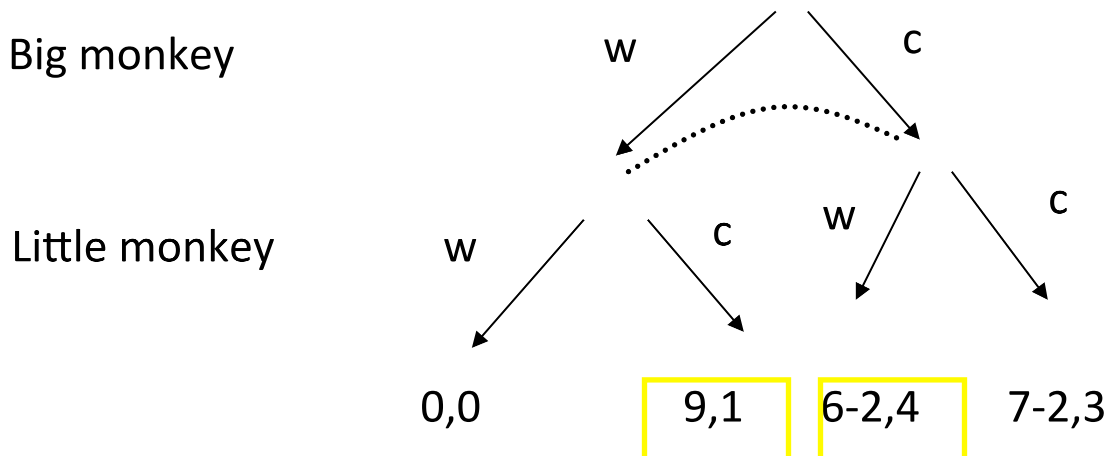
An example:
Big Monkey and Little Monkey

- These strategies (w and cw) are called *best responses.*
  - Given what the other guy is doing, this is the best thing to do.

- A solution where everyone is playing a best response is called a *Nash equilibrium.*
  - No one can unilaterally change and improve things.

An example:
Big Monkey and Little Monkey

- What if the monkeys have to decide simultaneously?



Now Little Monkey has to choose before he sees Big Monkey move
Two Nash equilibria (c,w), (w,c)

An example:
Big Monkey and Little Monkey

- What if the monkeys have to decide simultaneously?

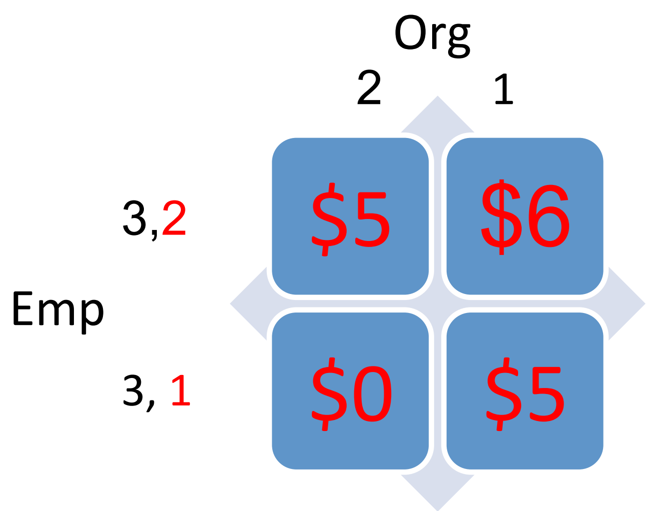
Little Monkey

Big Monkey

	c	w
c	5,3	4,4
w	9,1	0,0

Regret Minimization

Regret by Example



Strategy: outputs an action for every round

$$Total\ Regret(s, s_1) = -5 - (-6) = 1$$
$$regret(s, s_1) = 1/2$$

Players	Round 1	Round 2	Total Payoff
Emp	3,2	3, 1	Unknown
Org: s	1 ($6)	2 ($0)	$6
Org : s_1	2 ($5)	2 ($0)	$5

Audit Algorithm Choices

Only 30 inspections

Consider 4 possible allocations of the available 30 inspections

Sandra Bullock

Sandra Bullock	0	10	20	30
Other	30	20	10	0
Weights	1.0	1.0	1.0	1.0

Choose allocation probabilistically based on weights

Audit Algorithm Run

No. of Access	Actual Violation
30	2
70	4

0	10	20	30
30	20	10	0

Observed Loss

Estimated Loss

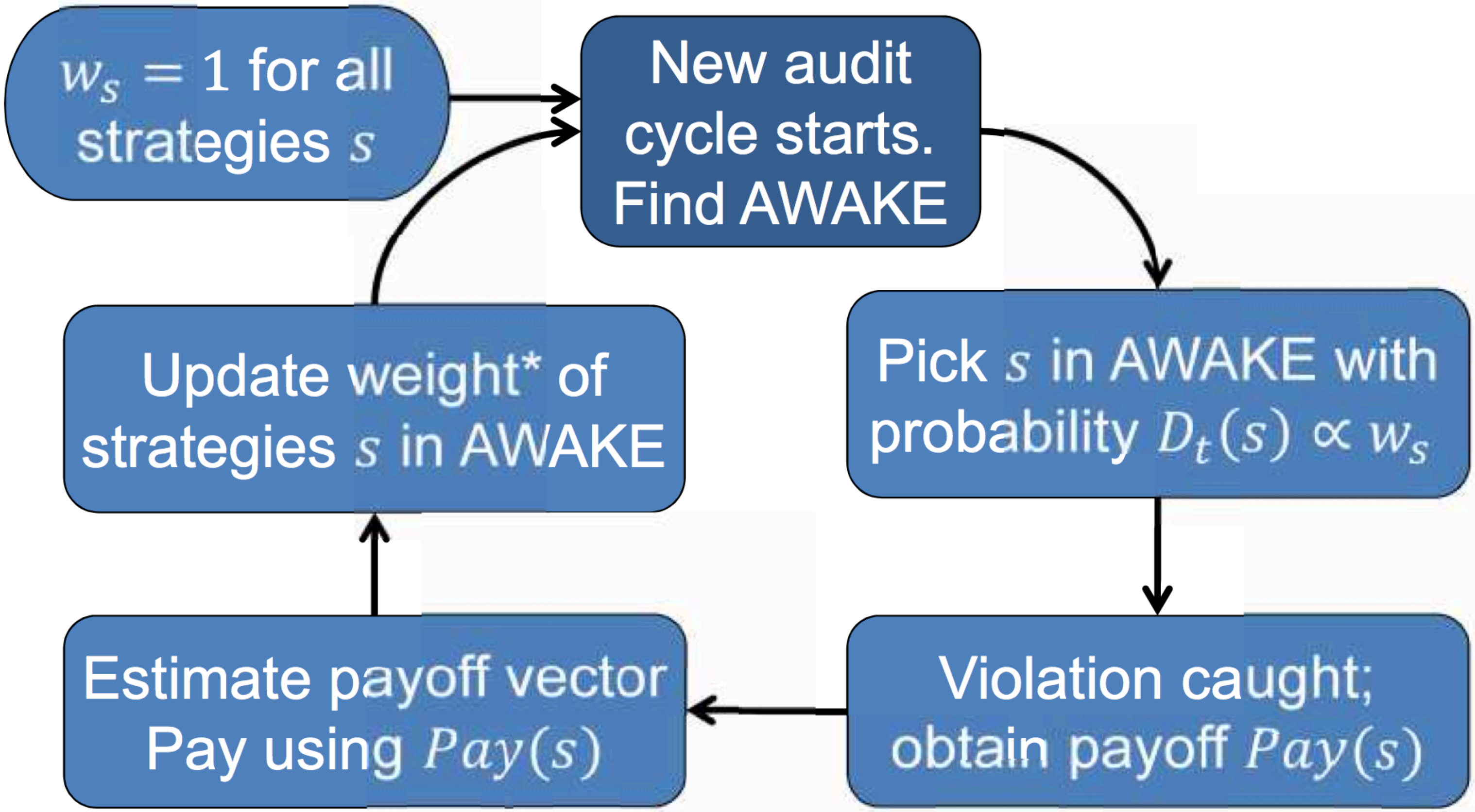
Int. Caught	Ext. Caught
1	1
2	1

\$2000	\$1500	\$1000	\$1000
\$750	\$1000	\$1250	\$1500

Updated weights	0.5	1.0	2.0	1.0

Learn from observed and estimated loss

Regret Minimizing Algorithm

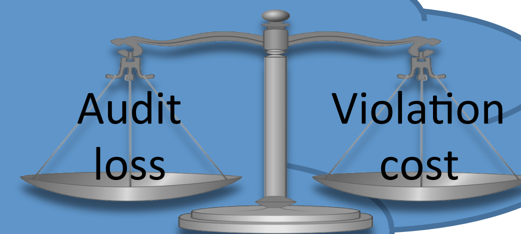


$$^* \; w_s \leftarrow w_s \cdot \gamma^{-Pay(s)+\gamma \sum_{s'} D_t(s') Pay(s')}$$

Model/Algorithm by Example

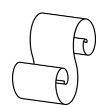
Auditor

Auditing budget: $3000/ cycle
Cost for one inspection: $100
Only 30 inspections per cycle
Employee incentives unknown



Access divided into 2 types

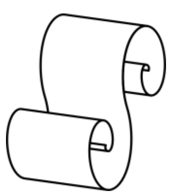
Loss from 1 violation (internal, external)



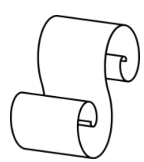
30 accesses

Sandra Bullock

$500, $1000



100 accesses



70 accesses

$250, $500

Utilities

$$\boldsymbol{U}(\vec{s}, \overrightarrow{\boldsymbol{O}}) = \underbrace{\sum_k U_1(s_k)}_{\text{Audit Cost}} + \underbrace{\sum_k U_2(\boldsymbol{O}_k)}_{\text{Violation Cost}}$$

Average utility over T rounds

$$= \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{U}(\vec{s}^{\,t}, \overrightarrow{\boldsymbol{O}}^{t})$$

Adversary utility unknown