## Games of Chance

Introduction to
Artificial Intelligence
COS302
Michael L. Littman
Fall 2001

## Administration

Rush hour (10/22).

Today not part of midterm (10/24), just final.

## Uncertainty in Search

We've assumed everything is known: starting state, neighbors, goals, etc.

Often need to make decisions even though some things are uncertain.

Complicates things…

## Types of Uncertainty

Opponent: What will other player do?
- Minimax

Outcome: Which neighbor get?
- Model via probability distribution

State: Where are we now?
- Hidden information

Transition: What are the rules?
- Need to use learning to find out

## Nim-Rand

Pile of sticks.
- Lose if take last stick.
- On your turn, take 1 or 2.
- Flip a coin. If H, take 1 more.

Which type of uncertainty?

## Value of a Game

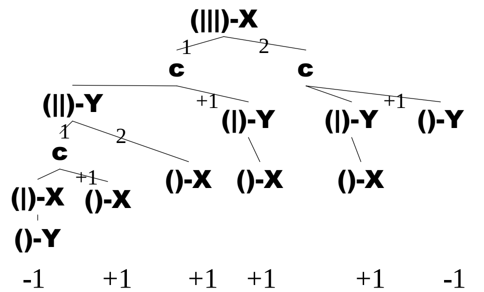
Without randomness: maximize your winnings in the worst case.

With randomness: maximize your *expected* winnings in the worst case.

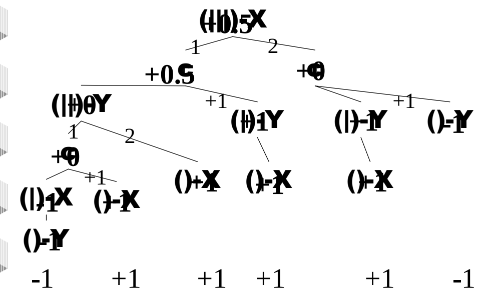
Want to do well on average.

What games are like this?

## Nim-Rand Tree




## Nim-Rand Values




## Search Model

- States, terminal states (G), values for terminal states (V).
- X states (maximizer), Y states (minimizer), Z states (chance)
- For all s in Z, for all s' in N(s)
  - P(s'|s) is the probability of reaching s' from s.

## Game Value (no loops)

Gameval(s) = {<br>
If (G(s)) return V(s)<br>
Else if s in X<br>
  return $\max_{s' \text{ in } N(s)}$ Gameval(s')<br>
Else if s in Y<br>
  return $\min_{s' \text{ in } N(s)}$ Gameval(s')<br>
Else<br>
  return $\text{sum}_{s' \text{ in } N(s)}$ P(s'|s) Gameval(s')<br>
}

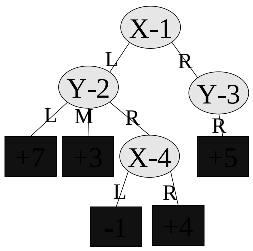
## Games with Loops

- No known poly time algorithm.
- Approximated by *value iteration*:
- For all s, if G(s), L(s) = V(s), else 0
- Repeat until changes are small:
  - for all s, L(s) =
    - max, min, avg L(s'), s' in N(s)
- depending on s in X, Y, or Z.

## Hidden Information

- Games like Poker, 2-player bridge, Scrabble ™, Diplomacy, Stratego
- Don't fit game tree model, even when chance nodes included.

## Pure Strategies




**X:** I: **1=L, 4=L**
II: **1=L, 4=R**
III: **1=R, 4=L**
IV: **1=R, 4=R**

**Y:** I: **2=L, 3=R**
II: **2=M, 3=R**
III: **2=R, 3=R**

## Matrix Form

**Summarizes all decisions in one for each, chosen simultaneously**

| | **X-**I | **X-**II | **X-**III | **X-**IV |
|---|---|---|---|---|
| **Y-**I | **7** | **7** | **2** | **2** |
| **Y-**II | **3** | **3** | **2** | **2** |
| **Y-**III | **-1** | **4** | **2** | **2** |

## Value of Matrix Game

**X picks column with largest min**

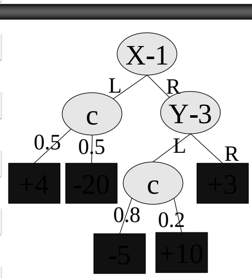
**Y picks row with smallest max**

| | **X-**I | **X-**II | **X-**III | **X-**IV |
|---|---|---|---|---|
| **Y-**I | **7** | **7** | **2** | **2** |
| **Y-**II | **3** | **3** | **2** | **2** |
| **Y-**III | **-1** | **4** | **2** | **2** |

## Minimax

- **Von Neumann proved zero-sum matrix game, minimax=maximin.**
- **Given perfect information (no state uncertainty), there exists optimal pure strategy for each player.**

## Game w/ Chance Nodes




**Use expected values**

| | **X-**I (L) | **X-**II (R) |
|---|---|---|
| **Y-**I (L) | **-8** | **-2** |
| **Y-**II (R) | **-8** | **+3** |

## More General Matrices

- **What game tree leads to this matrix?**
- **Does von Neumann's theorem still hold?**

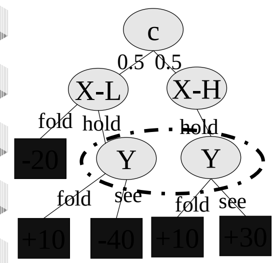
| | **X-**I (L) | **X-**II (R) |
|---|---|---|
| **Y-**I (L) | **1** | **0** |
| **Y-**II (R) | **0** | **1** |

## Hidden Info. Matrices

- X picks L or R, keeping the choice hidden from Y.
- Y makes a choice.
- X's choice is revealed and game ends.

| | **X-**I (L) | **X-**II (R) |
|---|---|---|
| **Y-**I (L) | **1** | **0** |
| **Y-**II (R) | **0** | **1** |

## Micro Poker




- X is dealt high or low card, holds/folds.
- Y folds/sees.
- High card wins
- Y can't see X's card.

## Matrix Form

| | **X-**I (fold) | **X-**II (hold) |
|---|---|---|
| **Y-**I (fold) | **-5** | **+10** |
| **Y-**II (see) | **+5** | **-5** |

- Player X can guarantee itself +1 on average. How?
- It can even announce its strategy.

## Mixed Strategies

- Pick a number p.
- X: With prob. p, fold; else hold.
- Since Y doesn't know what's coming, the response will sometimes work, sometimes not.

## Guess a Probability

- X announces p=1/3.
- Y's pick?

| | **X-**I (fold) | **X-**II (hold) |
|---|---|---|
| **Y-**I (fold) | **-5** | **+10** |
| **Y-**II (see) | **+5** | **-5** |

**Fold: +5**
**See: -1 2/3**
**see**

## Guess a Probability

- X announces p=2/3.
- Y's pick?

| | **X-**I (fold) | **X-**II (hold) |
|---|---|---|
| **Y-**I (fold) | **-5** | **+10** |
| **Y-**II (see) | **+5** | **-5** |

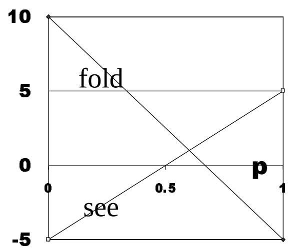
**Fold: +0**
**See: +1 2/3**
**fold**

## All Strategies

What should X pick for p to maximize its worst case?

p=0.6

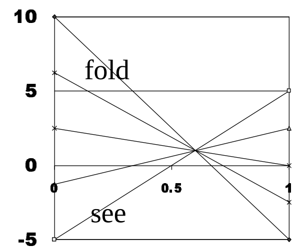
Payoff +1



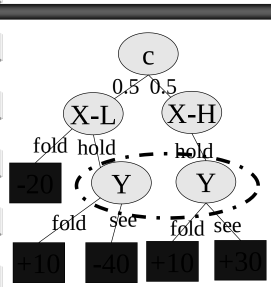

## Randomizing Y

If Y random, answer is the same.

No matter what, X can guarantee itself +1.




## Bluffing




X: On a low card, bluff with prob. 0.4.

Y: On hold, fold with prob. 0.4.

## Solving 2x2 Game

X-I with prob. p

X's expected gain vs. Y-I :

$m_{11}p+m_{12}(1-p)$

vs. Y-II :

$m_{21}p+m_{22}(1-p)$

Maximize the minimum.

|  | X-I | X-II |
|---|---|---|
| Y-I | $m_{11}$ | $m_{12}$ |
| Y-II | $m_{21}$ | $m_{22}$ |

Try p=0, p=1, where lines meet.

## Solving General mxn

Linear program: $p_1,\ldots,p_n$.

$p_1+\ldots+p_n = 1$, $p_i \geq 0$

Maximize X's gain, g

vs Y-I: $m_{11}\ p_1 + \ldots + m_{n1}\ p_n \geq g$

vs Y-II: $m_{12}\ p_1 + \ldots + m_{n2}\ p_n \geq g$

...

Against all Y strategies.

## Issues

Can we solve poker?

- More than 2 players
- Not zero sum (collude)
- Huge state space

Poker: Opponent modeling

Bridge: Use simulation to approximate

## What to Learn

**Minimax value in games of chance and the DFS algorithm for computing it.**

**Converting games to matrix form.**

**Solve 2x2 game.**

## Homework 5 (due 11/7)

1. **The value iteration algorithm from the *Games of Chance* lecture can be applied to deterministic games with loops. Argue that it produces the same answer as the "Loopy" algorithm from the *Game Tree* lecture.**
2. **Write the matrix form of the game tree below.**

## Game Tree