Games: Expectimax
Introduction to Utility Theory

Andrea Danyluk
February 22, 2017

Announcements

- Assignment 1
  - Code reviews today and tomorrow
  - Sign up by 4:00 PM today
- Programming Assignment 2 in progress

Today's Lecture

- Expectimax
- Utility Theory

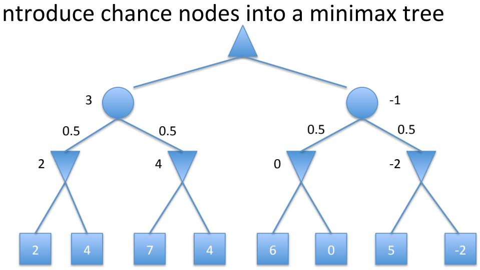
Multi-Player Games

- Evaluation function might return/returns a vector of utilities
- Each player chooses the move that maximizes its utility.

Stochastic Games

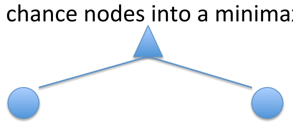
Expectimax

- Introduce chance nodes into a minimax tree



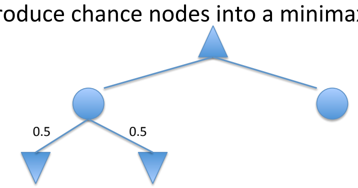
Expectimax

- Introduce chance nodes into a minimax tree



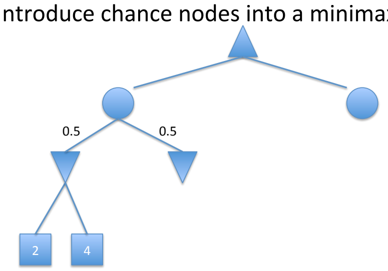
Expectimax

- Introduce chance nodes into a minimax tree



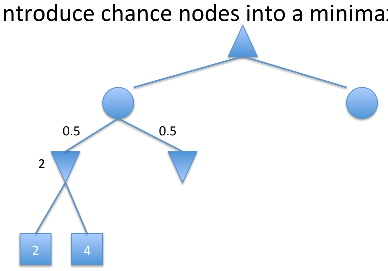
Expectimax

- Introduce chance nodes into a minimax tree



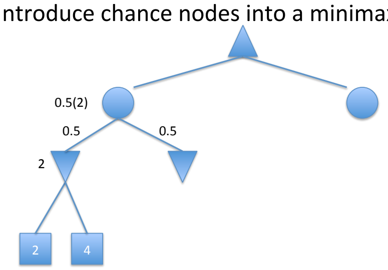
Expectimax

- Introduce chance nodes into a minimax tree



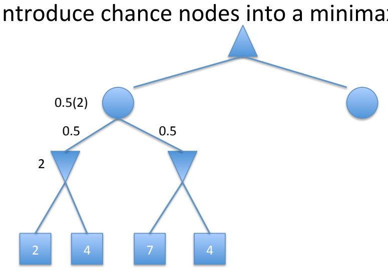
Expectimax

- Introduce chance nodes into a minimax tree



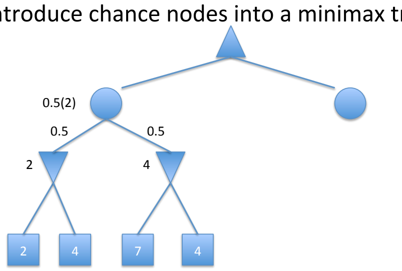
Expectimax

- Introduce chance nodes into a minimax tree



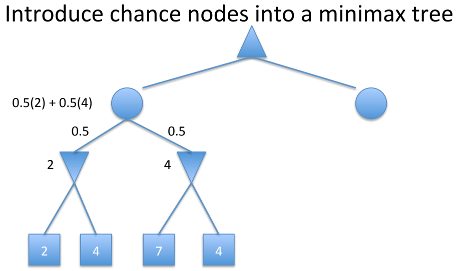
Expectimax

- Introduce chance nodes into a minimax tree



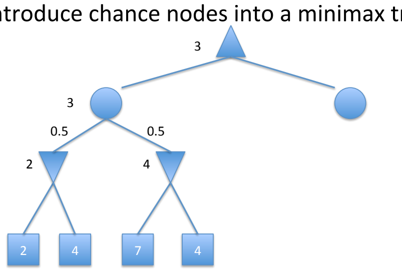
Expectimax

- Introduce chance nodes into a minimax tree



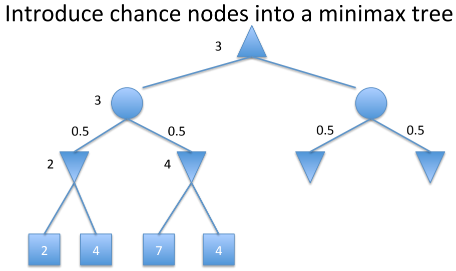
Expectimax

- Introduce chance nodes into a minimax tree



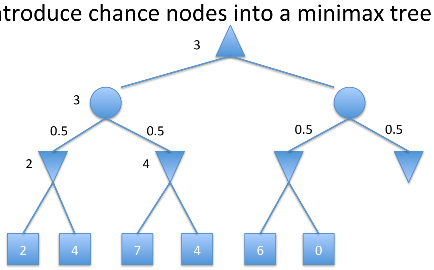
Expectimax

- Introduce chance nodes into a minimax tree



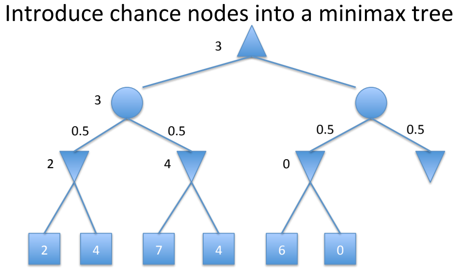
Expectimax

- Introduce chance nodes into a minimax tree



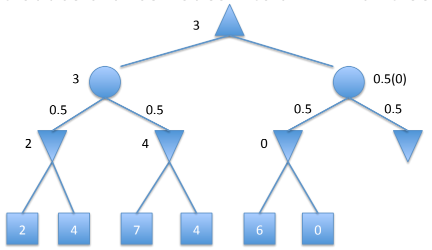
Expectimax

- Introduce chance nodes into a minimax tree



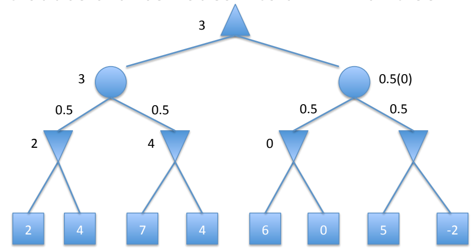
Expectimax

- Introduce chance nodes into a minimax tree



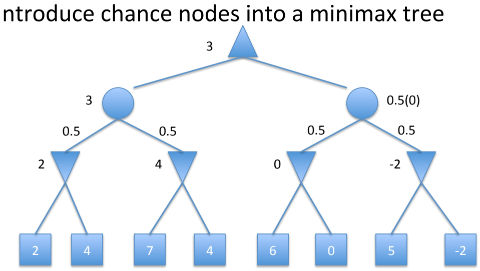
Expectimax

- Introduce chance nodes into a minimax tree



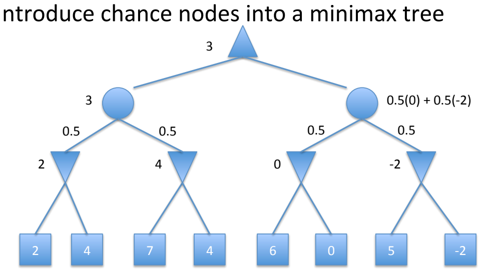
Expectimax

- Introduce chance nodes into a minimax tree



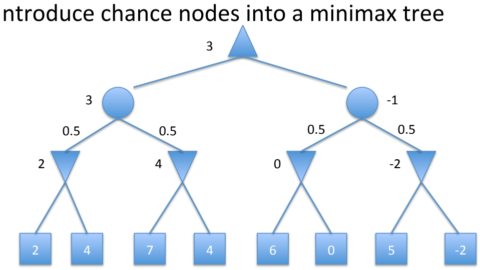
Expectimax

- Introduce chance nodes into a minimax tree



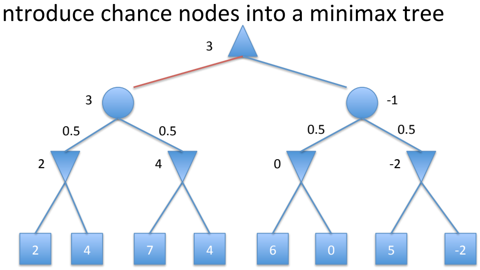
Expectimax

- Introduce chance nodes into a minimax tree



Expectimax

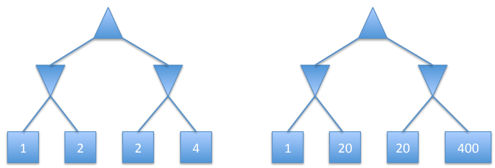
- Introduce chance nodes into a minimax tree



[From Russell]

Evaluation Functions Revisited

In Minimax (with Alpha-Beta Pruning)

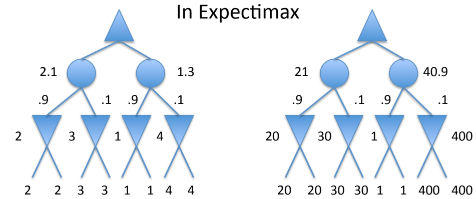


Behavior is preserved under any monotonic transformation of EVAL

[From Russell]

Evaluation Functions Revisited

In Expectimax

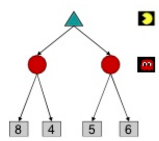


Behavior is preserved only under positive linear transformation of EVAL

[From Russell]

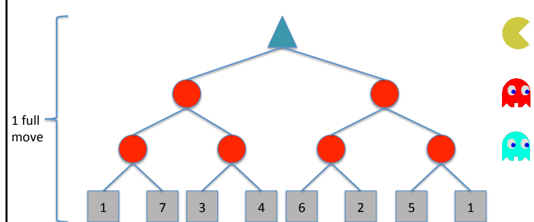
Expectimax Pacman

```
def value(s)
    if s is a max node return maxValue(s)
    if s is an exp node return expValue(s)
    if s is a terminal node return eval(s)
def maxValue(s)
    values = [value(s¹) for s¹ in succ(s)]
    return max(values)
def expValue(s)
    values = [value(s¹) for s¹ in succ(s)]
    weights = [prob(s,s¹) for s¹ in succ(s)]
    return expectation(values, weights)
```



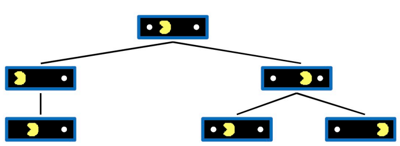
[Verbatim from CS 188]

Depth-limited Expectimax



Don't forget: Magnitudes of the utilities / heuristic evaluations need to be meaningful.

Why Pacman Can Starve



He knows his score can go up by eating now.
He knows his score can go up by eating later.
Within this search window there are no other eating opportunities.

Probabilities

- A **random variable** represents an event whose outcome is unknown
- A **probability distribution** is an assignment of weights to outcomes
- Example: L = type of lunch you'd see me eat today
  - Outcomes: L in (LunchBox, Thai, Subway)
  - Distribution : P(L=LB)=0.85,P(L=T)=0.13,P(L=S)=0.02
- Some laws of probability:
  - Probabilities are always non-negative
  - Probabilities over all possible outcomes sum to 1
- As we get more evidence, probabilities may change
  - P(L=T) = 0.15, P(L=T|I lunch meeting) = .9
  - But let's not worry about conditional probabilities for now

[Adapted from CS 188 Berkeley]

Expectations

- We can define a function f(X) of a random variable X
- The **expected value** of a function is its average value, weighted by the probability distribution over inputs
- How much money will be spent on lunch
  - M(LB) = $2.00, M(T) = $12.00, M(S) = $5.00
  - What is my expected lunch payment?

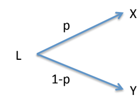
    E(M(L)) = M(LB)*P(LB)+M(T)*P(T)+M(S)*P(S)

    = 2.00(.85)+12.00(.13)+5.00(.02) = $3.36

[Adapted from CS 188 Berkeley]

Preferences (in an uncertain world)

- An agent chooses among prizes (e.g., X, Y) and lotteries (situations with uncertain prizes)
- Lottery L = [p, X; 1-p, Y]



- Notation:

$A \succ B$ — A preferred to B
$A \sim B$ — indifference between A and B
$A \succsim B$ — B not preferred to A

[This and the following either taken or adapted from Russell]

Rational Preferences

- Preferences of a rational agent must obey certain constraints

Constraints:

Orderability
$(A \succ B) \vee (B \succ A) \vee (A \sim B)$

Transitivity
$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$

Continuity
$A \succ B \succ C \Rightarrow \exists p \ [p, A; \ 1-p, C] \sim B$

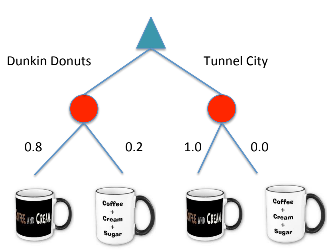
Substitutability
$A \sim B \Rightarrow [p, A; \ 1-p, C] \sim [p, B; 1-p, C]$

Monotonicity
$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; \ 1-p, B] \succsim [q, A; \ 1-q, B])$

Utilities

- A utility function captures an agent's preferences
  - DD coffee with cream only: U(DD/C) = 200
  - DD coffee with cream and sugar: U(DD/C&S) = 1
  - TC coffee with cream only: U(TC/C) = 100
  - TC coffee with cream and sugar: U(TC/C&S) = 1

Utilities: Uncertain Outcomes



0.8(200) + 0.2(1) = 160.2 1.0(100) + 0.0(1) = 100

MEU Principle

Theorem [von Neumann and Morgenstern, 1944]

- Given preferences satisfying the constraints (axioms of utility theory), there exists a real-valued function U such that

$$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$$
$$U([p_1, S_1; \ \ldots \ ; \ p_n, S_n]) = \Sigma_i \ p_i U(S_i)$$

- An agent can act rationally (i.e., consistently with its preferences) only if it chooses an action that maximizes expected utility

Class Exercise
St. Petersburg paradox [Nicolas Bernoulli,1713]

You have the opportunity to play a game in which a fair coin is tossed repeatedly until it comes up heads. If the first heads appears on the nth toss, you win 2^n dollars.

What is the expected monetary value of this game?

How much would you play to play the game?

Paradox Resolved

Nicolas's cousin Daniel Bernoulli resolved the apparent paradox in 1738 by suggesting that the utility of money is measured on a log scale:

$U(S_n) = a \log_2 n + b$, where

S_n is the state of having \$n

What is the expected utility of the game under this assumption?

Paradox Resolved

Nicolas's cousin Daniel Bernoulli resolved the apparent paradox in 1738 by suggesting that the utility of money is measured on a log scale:

$U(S_n) = a \log_2 n + b$, where

S_n is the state of having \$n

What is the expected utility of the game under this assumption?

What is the maximum amount it would be rational to play to play?

Money

- Money does not behave as a utility function, but we can talk about the utility of having money (or being in debt)