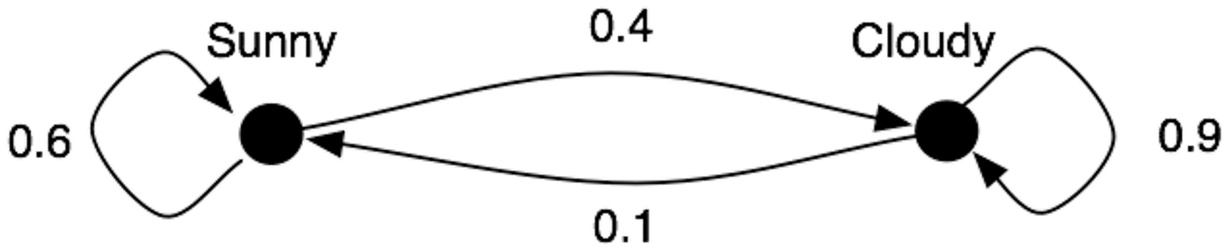
Hidden Markov Models: Viterbi

Markov chains

- Stochastic process model
  - Due to Andrey Markov (1906)
  - e.g.,



Markov chain

- Models a system which is in exactly one state at any time *t*, denoted by random variable Q_t
- A Markov chain model consists of:
  - A discrete set of states $S=\{s_1, \ldots s_N\}$
  - *An initial probability distribution* $P(Q_0)$
  - Transition probability distribution, given by a conditional distribution $P(Q_{t+1} | Q_t)$
- The Markov assumption:
  - The probability of transitioning to each new state depends *only* on the current state (and not on the previous states)
  - More formally,

$$P(Q_{t+1} = q_{t+1} | Q_t = q_t, Q_{t-1} = q_{t-1}, \ldots, Q_0 = q_0) = P(Q_{t+1} = q_{t+1} | Q_t = q_t)$$

Hidden Markov Models (HMMs)

- A Markov Chain, but the system state is *not observable*
  - Instead there is an observable random variable, *O*, whose value probabilistically depends on the current state

- More formally, an HMM consists of:
  - Transition probabilities

$$p_{ij} = P(Q_{t+1} = j | Q_t = i)$$

  - Initial state distribution

$$w_i = P(Q_0 = i)$$

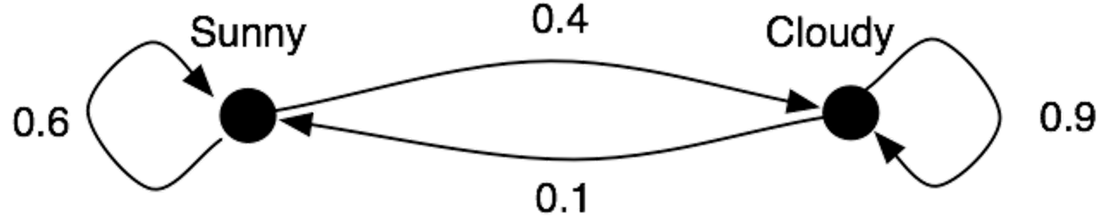
  - Emission probabilities

$$e_i(a) = P(O_t = a | Q_t = i)$$

Example

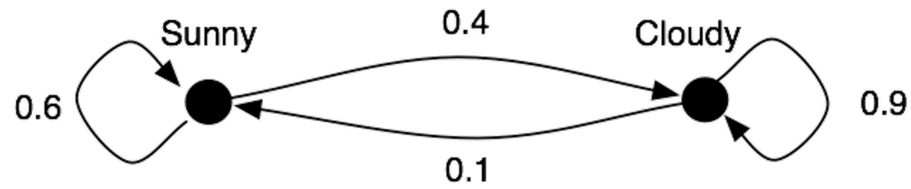
- Mary lives in Seattle, and tells the truth 80% of the time. Every day, she calls you to report the weather in Seattle.
  – It's either Sunny (S) or Cloudy (C)

- You know (based on historical data) that the weather in Seattle follows a Markov chain,



  – Also, the probability of sun on any given day is 0.2

- Mary reports that the following sequence over a 5 day period: SCSCC



- Transition probabilities $p_{SS} = P(Q_{t+1} = S | Q_t = S) = 0.6$

$p_{CS} = 0.1 \qquad p_{CC} = 0.9 \qquad p_{SC} = 0.4$

- Emission probabilities

$e_C(S) = P(O_t = S | Q_t = C) = 0.2 \qquad e_S(C) = P(O_t = C | Q_t = S) = 0.2$

$e_C(C) = P(O_t = C | Q_t = C) = 0.8 \qquad e_S(S) = P(O_t = S | Q_t = S) = 0.8$

- Initial state distribution

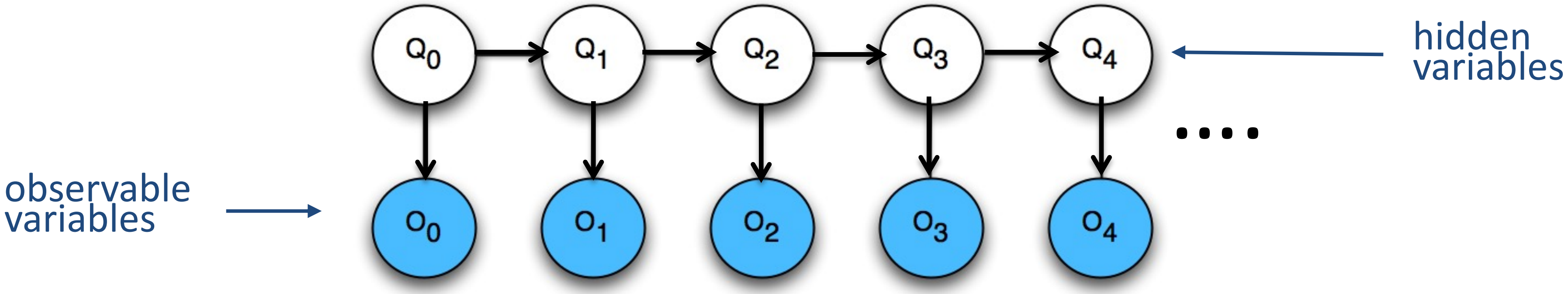
$w_S = P(Q_0 = S) = 0.2 \qquad w_C = P(Q_0 = C) = 0.8$

- Observation sequence

$$O_0 = S, O_1 = C, O_2 = S, O_3 = C, O_4 = C$$

Inference on HMMs

- HMMs are just special cases of Bayes Nets!



- Intuitively, the HMM is balancing two goals:
  - maximizing emission probabilities -- finding a state sequence that agrees with the observations
  - maximizing transition probabilities -- finding a state sequence that has high likelihood according to the Markov chain

Classifying photo streams

3:35pm

Alcatraz, SF bay?
Ellis Island, NYC?

8:03pm

Piazza San Marco, Venice?
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
Geo Wash Bridge, NYC?

Classifying photo streams

3:35pm

Alcatraz, SF bay?
~~Ellis Island, NYC?~~

8:03pm

~~Piazza San Marco, Venice?~~
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
~~Geo Wash Bridge, NYC?~~

Classifying photo streams

3:35pm

Alcatraz, SF bay?
~~Ellis Island, NYC?~~

8:03pm

~~Piazza San Marco, Venice?~~
Sather Tower, Berkeley?

9:27pm

Bay Bridge, SF bay?
~~Geo Wash Bridge, NYC?~~

- Model as a Hidden Markov Model, do fast inference using the Viterbi algorithm

HMM inference

- How do we find the most likely state sequence, given a sequence of observations?
  - Brute force approach: Try all possible state sequences. Find the one that maximizes P(Q|O).
  - Viterbi decoding: Efficient algorithm based on dynamic programming.

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0...O_T|Q_0 = q_0...Q_T = q_T)P(Q_0 = q_0...Q_T = q_T)}{P(O_0...O_T)}$$

(denom depends only on O) $$\propto P(O_0...O_T|Q_0 = q_0...Q_T = q_T)P(Q_0 = q_0...Q_T = q_T)$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T)$$

(Bayes' Law) $$= \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

(denom depends only on O) $$\propto P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)$$

(O_t depends only on Q_t) $$= P(Q_0 = q_0 ... Q_T = q_T) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

Viterbi decoding

- Key idea: the posterior probability of a state sequence, P(Q|O), factors nicely

$$P(Q_0 = q_0, \ldots, Q_T = q_T | O_0 \ldots O_T)$$

(Bayes' Law)
$$= \frac{P(O_0 \ldots O_T | Q_0 = q_0 \ldots Q_T = q_T) P(Q_0 = q_0 \ldots Q_T = q_T)}{P(O_0 \ldots O_T)}$$

(denom depends only on O)
$$\propto P(O_0 \ldots O_T | Q_0 = q_0 \ldots Q_T = q_T) P(Q_0 = q_0 \ldots Q_T = q_T)$$

(O_t depends only on Q_t)
$$= P(Q_0 = q_0 \ldots Q_T = q_T) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

(Markov property: Q_{t+1} depends Only on Q_t)
$$= P(Q_0 = q_0) \prod_{t=0}^{T-1} P(Q_{t+1} = q_{t+1} | Q_t = q_t) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

Viterbi decoding

- Based on dynamic programming
  - Let $v_i(t)$ be the probability of the most probable path ending at state *i* at time *t*,

$$v_i(t) = \max_{q_0 \cdots q_{t-1}} P(Q_0 = q_0, \ldots, Q_{t-1} = q_{t-1}, Q_t = i | O_0, O_1, \ldots, O_t)$$

  - Then we can recursively find the probability of the most probable path ending at state j at time t+1,

$$v_j(t+1) = e_j(O_{t+1}) \max_{1 \le i \le N} v_i(t) p_{ij}$$

Probability of transition from state i to j

Probability that system is in state j at time t+1 (Q_{t+1}=j)

Probability of observing O_{t+1} given that system is in state j at time t+1

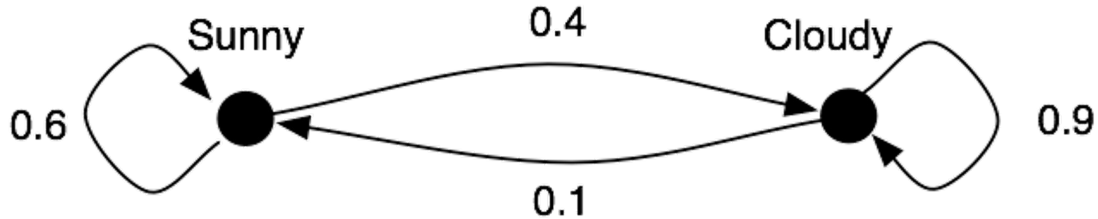
Max over all possible states at time t

Probability that system in state i at time t

Example

- Mary lives in Seattle, and tells the truth 80% of the time. Every day, she calls you to report the weather in Seattle.
  - It's either Sunny (S) or Cloudy (C)

- You know (based on historical data) that the weather in Seattle follows a Markov chain,



  - Also, the probability of sun on any given day is 0.2

- Mary reports that the following sequence over a 5 day period: SCSCC

$$v_j(t+1) = e_j(O_{t+1}) \max_{1 \leq i \leq N} v_i(t) p_{ij}$$

Viterbi decoding

- Takes time $O(N^2T)$
  - N is the number of states
  - T is the length of the sequence

- For many useful state transition probability functions, it's possible to do this faster

Max probability vs Min cost

- Maximizing the probability P(Q|O),

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T) \;=\; P(Q_0 = q_0) \prod_{t=0}^{T-1} P(Q_{t+1} = q_{t+1} | Q_t = q_t) \prod_{t=0}^{T} P(O_t | Q_t = q_t)$$

is equivalent to minimizing a negative log,

$$\begin{aligned} &-\ln P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T) \\ &\quad = -\ln \left[P(Q_0 = q_0) \prod_{t=0}^{T-1} P(Q_{t+1} = q_{t+1} | Q_t = q_t) \prod_{t=0}^{T} P(O_t | Q_t = q_t) \right] \\ &\quad = -\ln P(Q_0 = q_0) + \sum_{t=0}^{T-1} - \ln P(Q_{t+1} = q_{t+1} | Q_t = q_t) + \sum_{t=0}^{T} - \ln P(O_t | Q_t = q_t) \end{aligned}$$

Cost minimization view

- Viterbi finds a state sequence $q_0 \ldots q_T$ that maximizes the posterior probability,

$$P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T) = \frac{P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) P(Q_0 = q_0 ... Q_T = q_T)}{P(O_0 ... O_T)}$$

- Equivalently, we can find a sequence that minimizes,

$$\begin{aligned} -\ln P(Q_0 = q_0, ..., Q_T = q_T | O_0 ... O_T) \quad = \quad & -\ln P(O_0 ... O_T | Q_0 = q_0 ... Q_T = q_T) \\ & -\ln P(Q_0 = q_0 ... Q_T = q_T) + \ln P(O_0 ... O_T) \end{aligned}$$

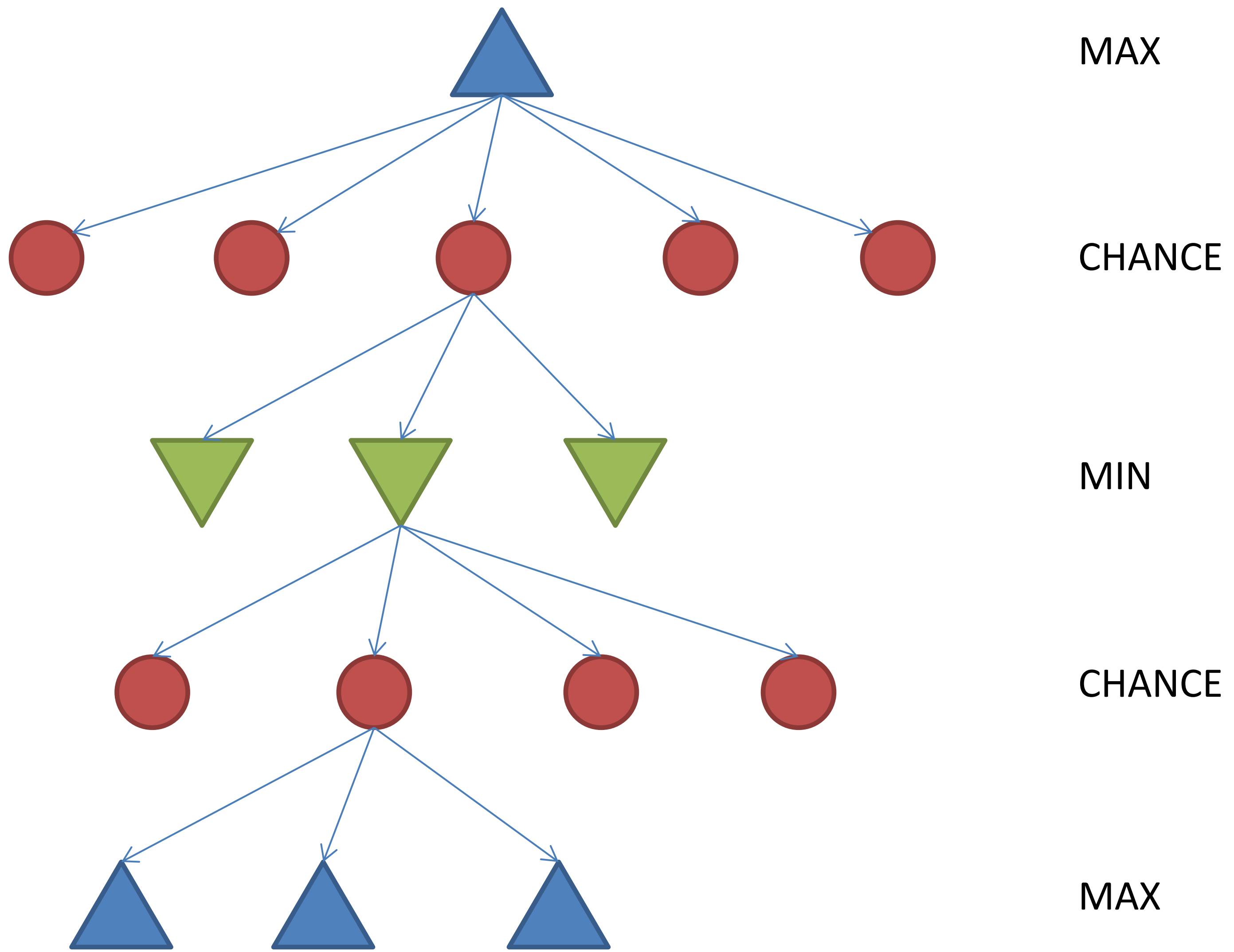
  - View the negative log probabilities as "costs"
  - More convenient computationally (avoids multiplying very small probabilities)

Games of chance, and more Variable Elimination

Games of chance

- Many games have non-determinism
  - E.g. Coin flips, dice, cards drawn from a pile, …

- Other games have state that a player cannot observe
  - E.g., which cards other player holds, how much money has been bet, whether a monster is behind me, …


MAX
CHANCE
MIN
CHANCE
MAX

Expected Values

- The utility of a MAX/MIN node in the game tree is the **max/min** of the utility values of its successors
- The expected utility of a CHANCE node is the **expected value** of the utility values of its successors

ExpectedValue(s) = $\Sigma_{s' \in SUCC(s)}$ ExpectedValue(s') P(s') **CHANCE nodes**

Compare to

MinimaxValue(s) = max $_{s' \in SUCC(s)}$ MinimaxValue(s') **MAX nodes**

MinimaxValue(s) = min $_{s' \in SUCC(s)}$ MinimaxValue(s') **MIN nodes**

Generalizing Minimax Values

- *Utilities* can be continuous numerical values, rather than +1, 0, -1
  - Allows maximizing the amount of “points” (e.g., $) rewarded instead of just achieving a win
- Rewards associated with terminal states
- Costs can be associated with certain decisions at non-terminal states (e.g., placing a bet)

Roulette

- 18 red, 18 black, 2 green spots (American version)
- Bet on color: bet $1, get $2 back.
- Bet on a number: bet $1, get $35 back

http://www.siwallpaperhd.com/wp-content/uploads/2016/07/gambling_roulette_wallpaper_full_hd.jpg

Roulette

- "Game tree" only has depth 2
  - Place a bet
  - Observe the roulette wheel

No bet

Bet: Red, $5

Chance node

Probabilities

18/38

20/38

Red

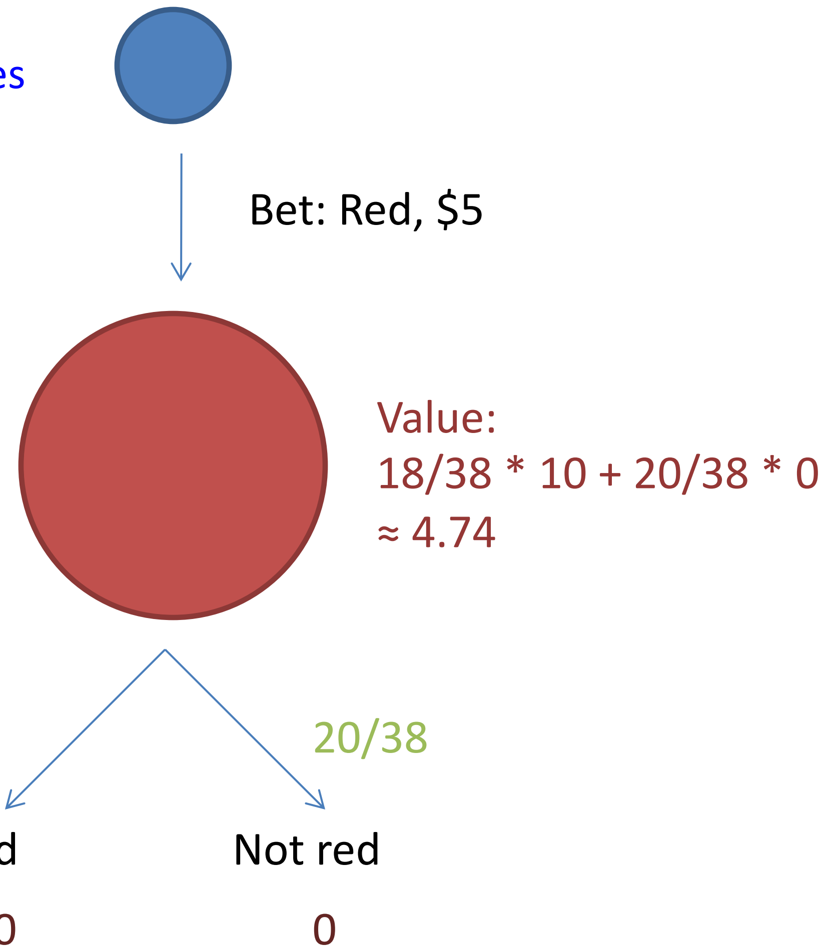
+10

Not red

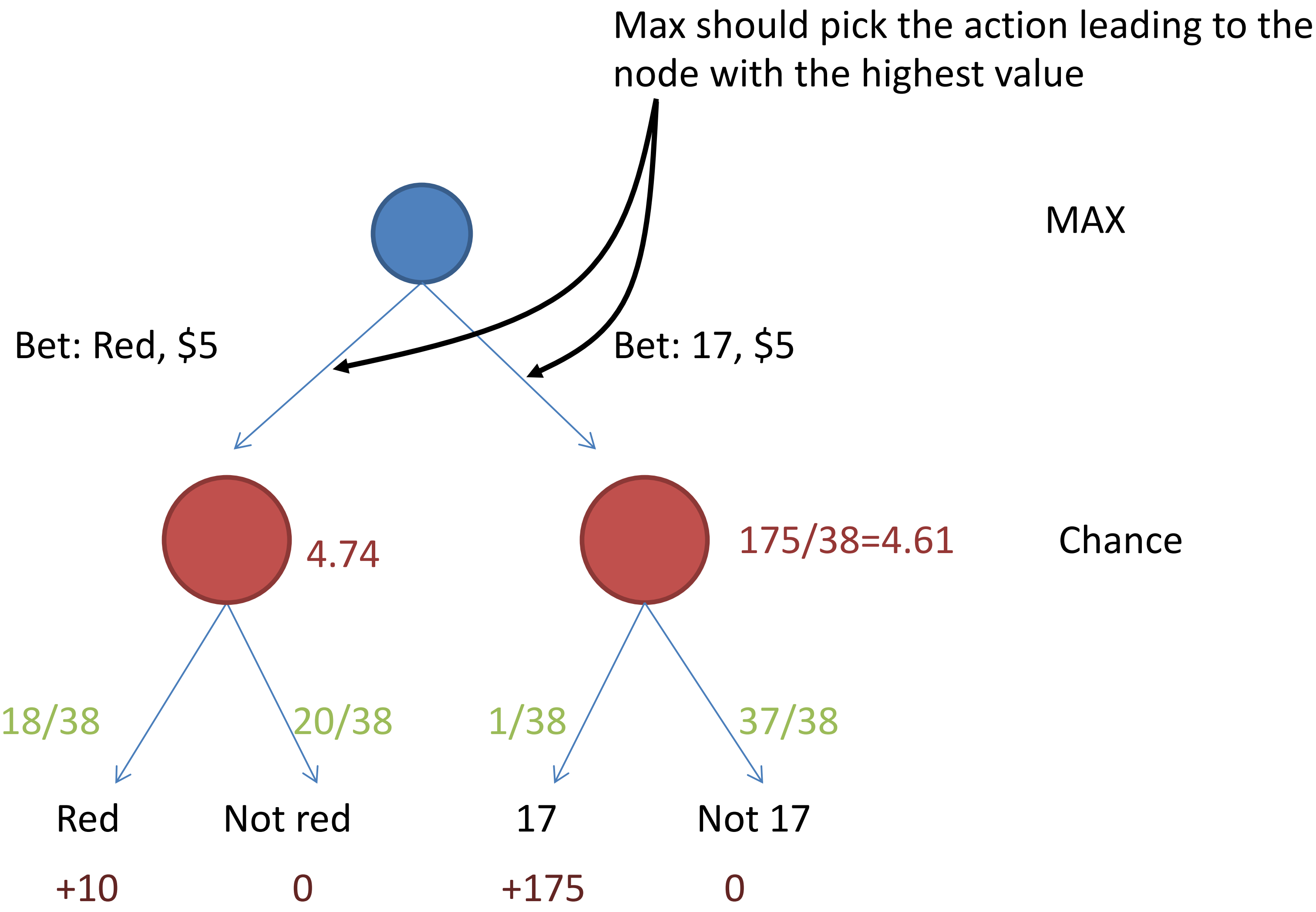
0

Chance Node Backup

- Expected value:
  – For k children, with backed up values $v_1, \ldots, v_k$
  – Chance node value = $p_1 * v_1 + p_2 * v_2 + \ldots + p_k * v_k$



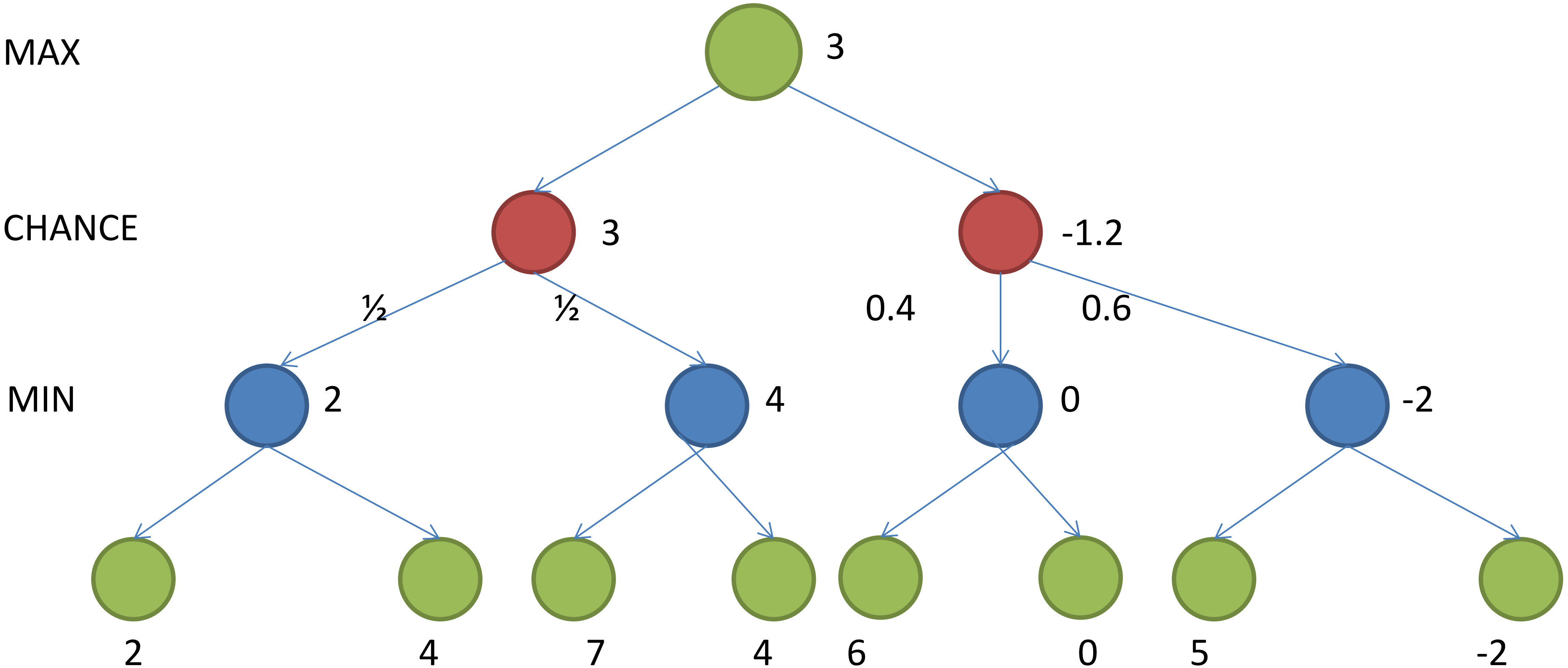
MAX/Chance Nodes



Adversarial Games of Chance

- E.g., Backgammon
- MAX nodes, MIN nodes, CHANCE nodes
- Expectiminimax search
- Backup step:
  - MAX = maximum of children
  - CHANCE = average of children
  - MIN = minimum of children
  - CHANCE = average of children
- 4 levels of the game tree separate each of MAX’s turns!
- Evaluation function? Pruning?

Another example



Card Games

- Blackjack (6-deck), video poker: similar to coin-flipping game
- But in many card games, need to keep track of history of dealt cards in state because it affects future probabilities
  - One-deck blackjack
  - Bridge
  - Poker

Partially Observable Games

- Partial observability
  - Don’t see entire state (e.g., other players’ hands)
  - “Fog of war”

- Examples:
  - Kriegspiel (see R&N)
  - Battleship

Next time

- Sampling and MCMC