CS664 Lecture #5: Markov chains

Some source material taken from:

- **Joseph Chang**

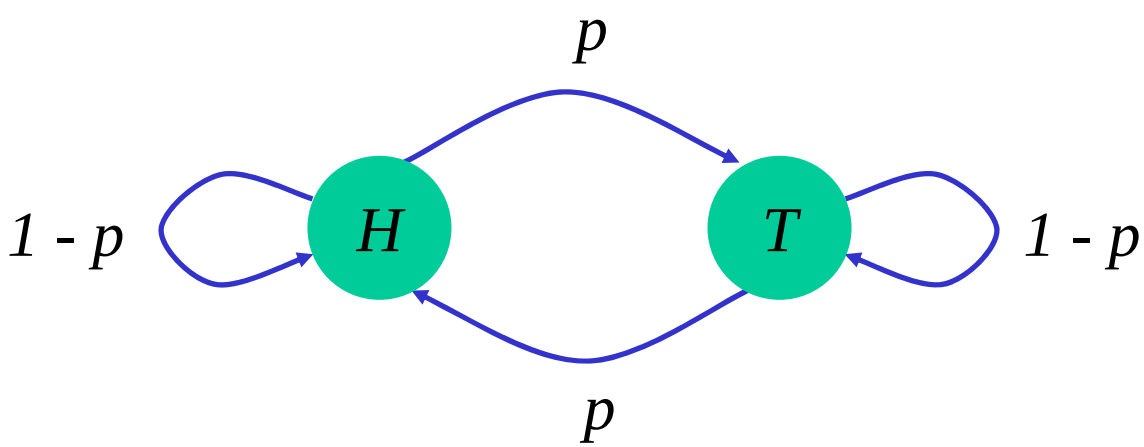
http://www.stat.yale.edu/~jtc5/jtc.html

Coins with memory

- Suppose that the coin acts the way that gamblers think
  - Look back at last result
  - Produce the opposite answer (probability p) or the same answer (probability 1-p)
- At p = .5, what percentage of heads do we expect in the limit?
  - What about at p = .1? (“Stubborn” coin)
  - What about at p = .9? (“Flighty” coin)
  - What about at p = 0? (“Stuck” coin)

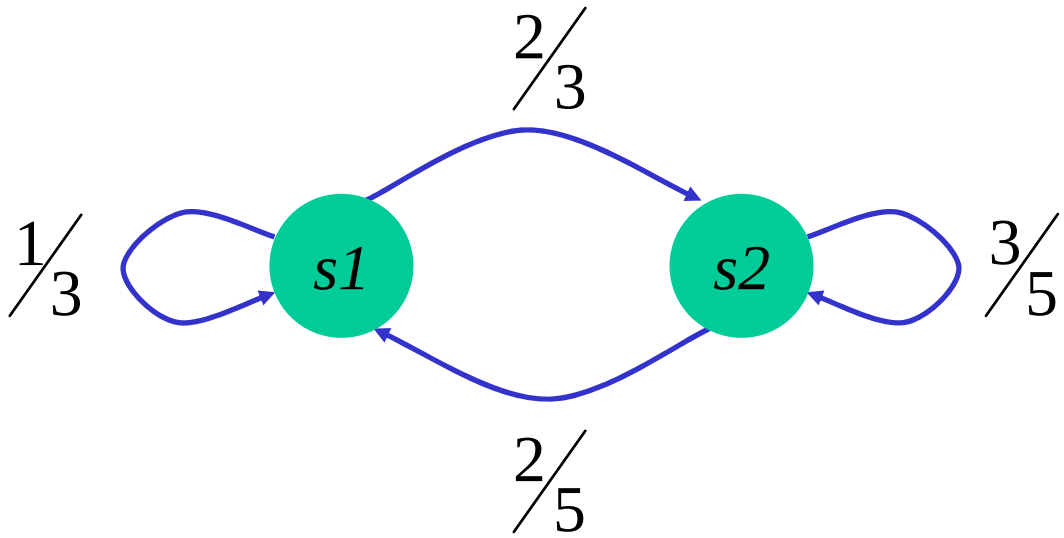
Markov chains

- Generalization of a finite automaton
- Probabilistic transitions (edge weights)

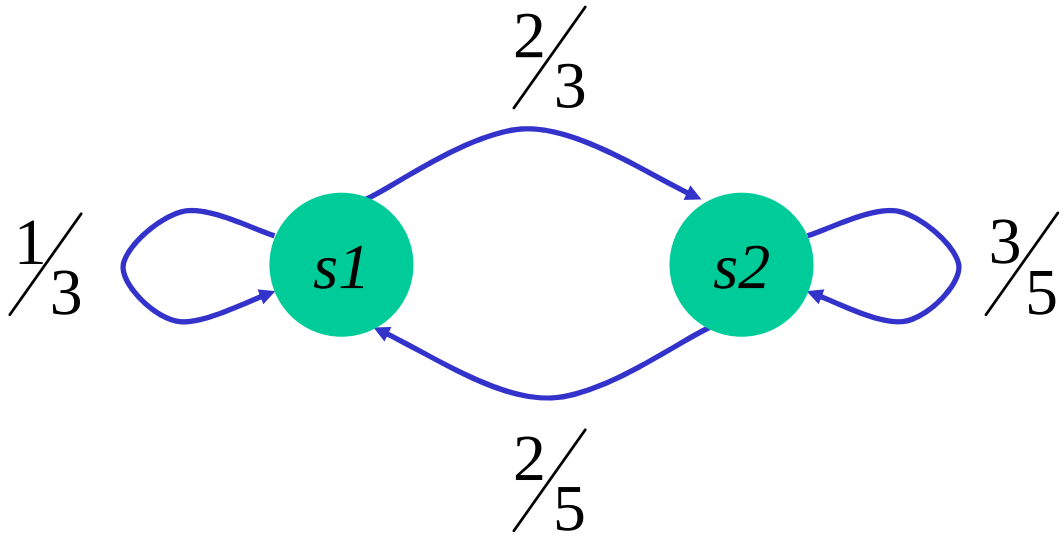


Markov chain evolution

- Distribution over states at a given time
- Taking a step updates the distribution
  - According to the edge weights
  - Consider the “Markov frog”



Example in action



$$\begin{vmatrix} 1 \\ 0 \end{vmatrix} \quad \begin{vmatrix} \frac{1}{3} \\ \frac{2}{3} \end{vmatrix} \quad \begin{vmatrix} \frac{1}{3}\frac{1}{3} + \frac{2}{5}\frac{2}{3} \\ \frac{2}{3}\frac{1}{3} + \frac{3}{5}\frac{2}{3} \end{vmatrix}$$

$$P_1' = \frac{1}{3} \cdot P_1 + \frac{2}{5} \cdot P_2$$

$$P_2' = \frac{2}{3} \cdot P_1 + \frac{3}{5} \cdot P_2$$

Transition matrix

- The “probability mass” moved to a state is a linear combination of the masses at adjacent states
  - Coefficients are the edge weights

$$\begin{vmatrix} P'_1 \\ P'_2 \end{vmatrix} = \begin{vmatrix} \frac{1}{3} & \frac{2}{5} \\ \frac{2}{3} & \frac{3}{5} \end{vmatrix} \begin{vmatrix} P_1 \\ P_2 \end{vmatrix}$$

Some notation

- Stochastic vector π has non-negative elements that sum to 1
  - Stochastic matrix K has stochastic columns
  - π_n is the distribution after n steps

$$\pi_n = K\pi_{n-1} = K(K\pi_{n-1}) = \cdots = K^n\pi_0$$

Stationary distributions

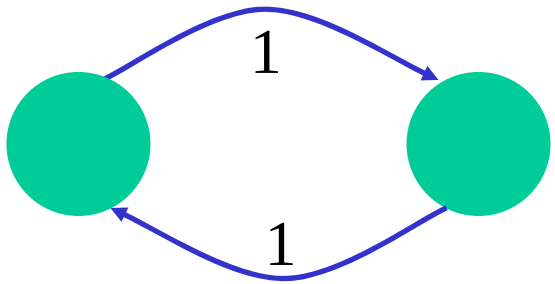
$$\left\| \begin{array}{cc|c} \frac{1}{3} & \frac{2}{5} & \frac{3}{8} \\ & & \\ \frac{2}{3} & \frac{3}{5} & \frac{5}{8} \end{array} \right\|$$

π^* is *stationary* if $\pi^* = K\pi^*$

K *converges to* π^* if $\forall \pi \quad \lim_{n \to \infty} K^n \pi = \pi^*$

Perron-Frobenius theorem

- If a Markov chain is strongly connected and has self-loops, it converges to a unique stationary distribution
  - No matter what the starting distribution
- Multiple self-loops are not required
  - Need to avoid “oscillating” cases



Convergence rates

- Nothing in this theorem about rate!
- There are some complicated theorems on this topic
  - Nothing that guarantees fast convergence for the cases of interest

Markov coin revisited

- Transition matrix is given by

$$\begin{vmatrix} \Pr(HH) & \Pr(TH) \\ \Pr(HT) & \Pr(TT) \end{vmatrix}$$

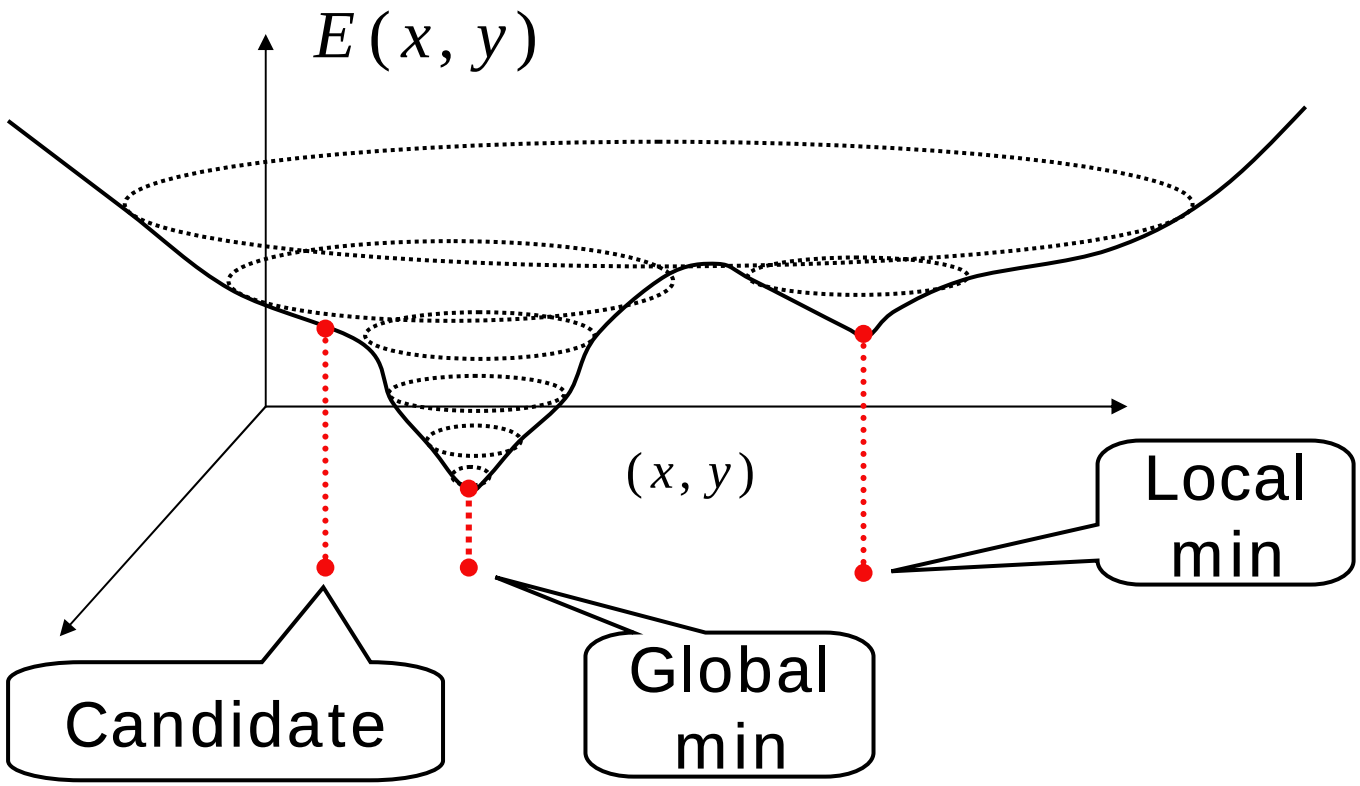
$$\begin{vmatrix} 1-p & q \\ p & 1-q \end{vmatrix}$$

  - What about $p=q=0$?

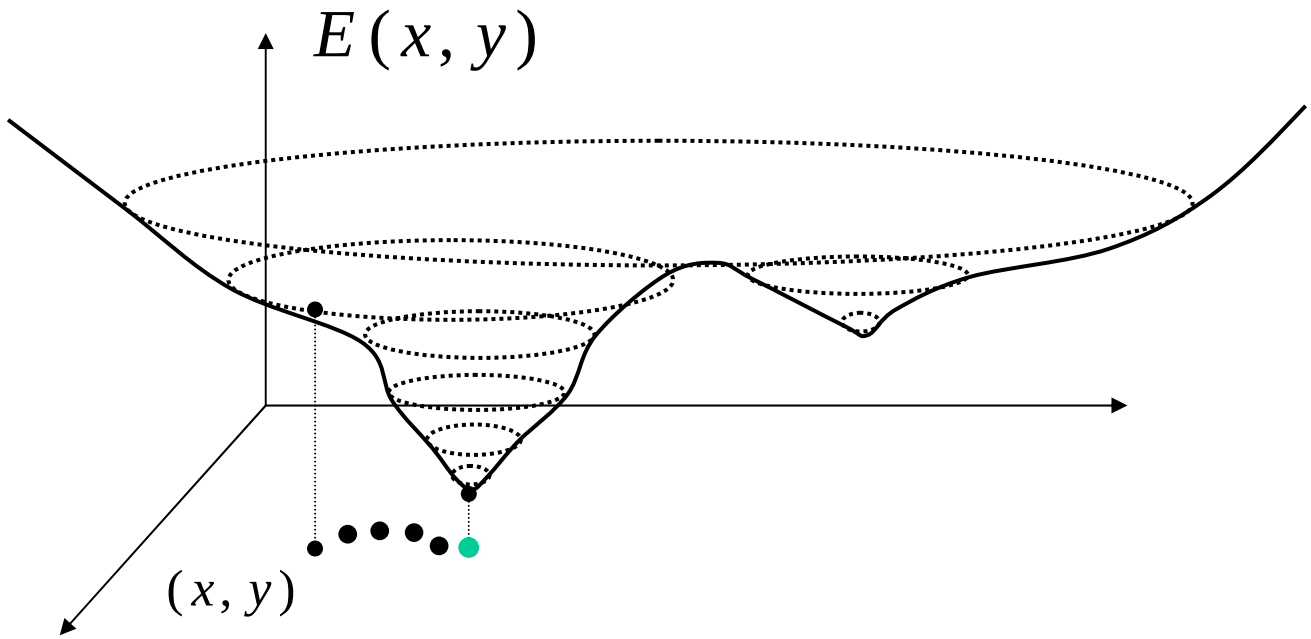
Markov chains in vision

- Vital tool for many vision problems
  - Basis for trigrams, hence Efros & Leung
    - Images have “local” structure
- Major application: sampling
  - Generating answers from a distribution
- Major application: energy minimization
  - Also known as optimization
  - Elegant way to formulate most vision problems
  - Lots of interesting and powerful algorithms

Energy function



Gradient descent



Properties of E

- Local versus global minimum
- If there is a unique minimum, E is said to be convex
  - Issue becomes convergence speed
  - In vision, we're rarely so lucky
- We can compute global min sometimes
  - Other times compute a "strong" local min

Tradeoffs of optimization

- Advantages
  - Clean separation between what you want to compute and how you compute it
  - Easy to add new constraints (terms)
  - Simple to explain
- Disadvantages
  - Optimization is often difficult
  - Separation of what and how can hurt you

Complexity

- In complete generality, computing global min requires exhaustive search
- Consider 2 energy functions
  - Uniform (flat everywhere)
  - Uniform with a well somewhere
- True even if P=NP

Consequences

- Consider an optimization method that can find the global min of an arbitrary E
  - Must require exponential time
  - Asymptotically same as exhaustive search
- Might work for a particular problem
- Strong methods have limited E
  - You need to understand and exploit the structure of the problem

General-purpose methods

- Example: genetic algorithms
  - Not a method taken seriously by reputable academics, in vision or elsewhere
- Population of candidate solutions
  - Representation is key
- Create new population
  - Crossovers, mutations
  - Replace the worst (highest E) candidates

Gradient descent alternatives?

- If E is convex we can just roll downhill
  - Risk is being stuck in local min
- What if we sometimes move uphill?

Metropolis(E,T)

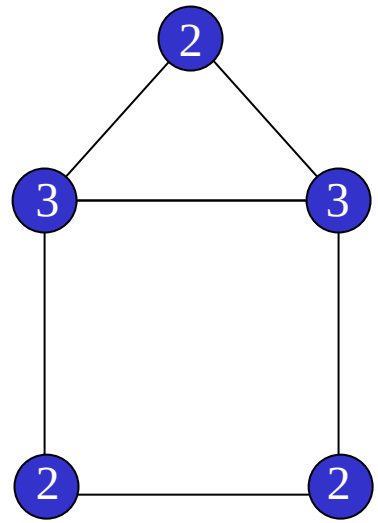
1. Generate random change (“sampling”)
2. If the energy is lower, go there
3. If the energy is higher
    1. Go there with probability $\propto \exp(-\Delta E/T)$
    2. Otherwise, stay at old candidate

Metropolis properties

- We can do nothing (step 3.2)
- Gradient descent at low T, random search at high T
- Randomized algorithm
  - Output is a distribution over candidates
  - Hence, distribution over energy

Random walks on graphs

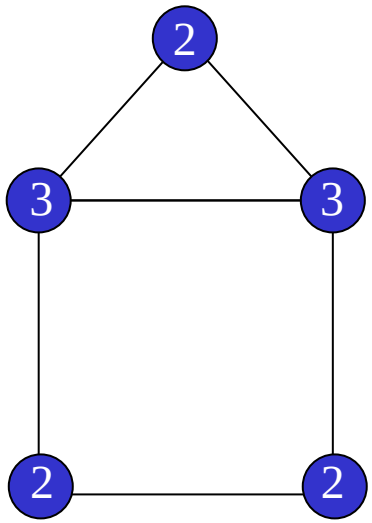
- Suppose we pick an edge uniformly at random from the outgoing edges
  - Undirected graph with self-loops
  - What is the stationary distribution?
    - More likely to end up at a node with many (incoming) edges, i.e. high degree



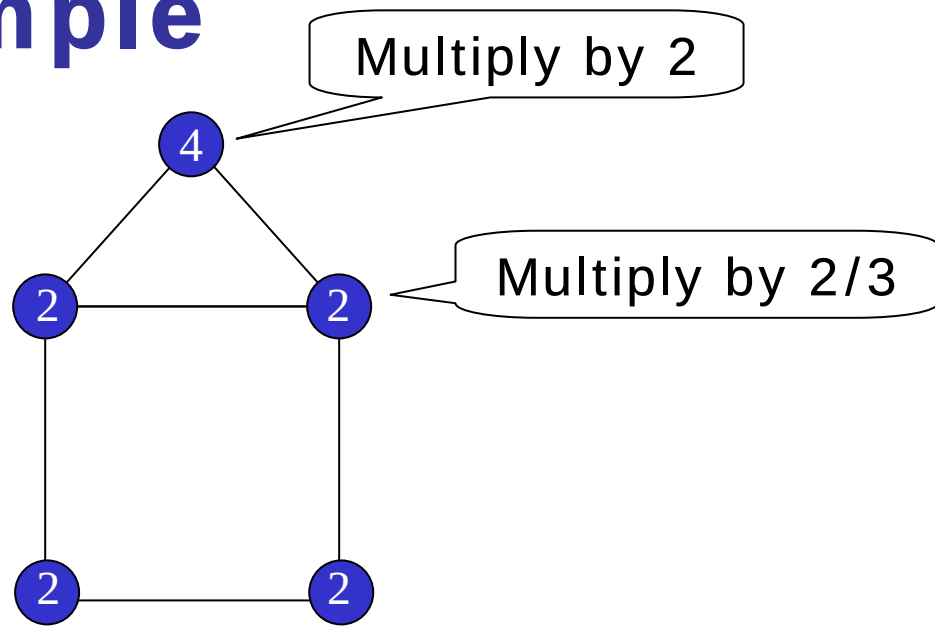
Biased random walks

- What if we want a different stationary distribution?
  - E.g., high degree node should be “unpopular”
  - Solution: change transition probabilities
    - I.e., don’t pick an outgoing edge uniformly
- Weight the outgoing edges by their relative popularity in desired distribution

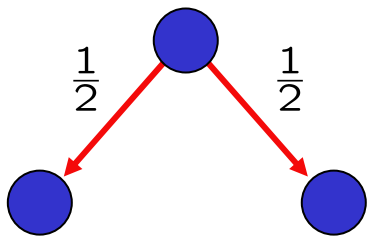
Example

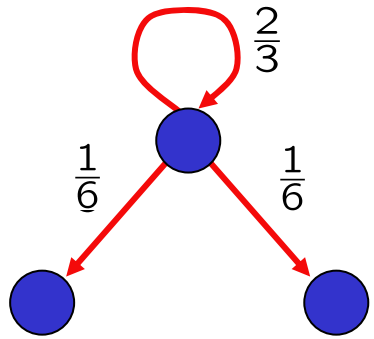


Stationary distribution



Desired distribution





$$\frac{1}{6} = \frac{1}{2} \cdot \frac{2}{3}$$

$$\frac{2}{3} = 1 - \frac{1}{6} - \frac{1}{6}$$