Random Walks on Graphs

Robert Y. Lewis

CS 0220 2023

April 26, 2023

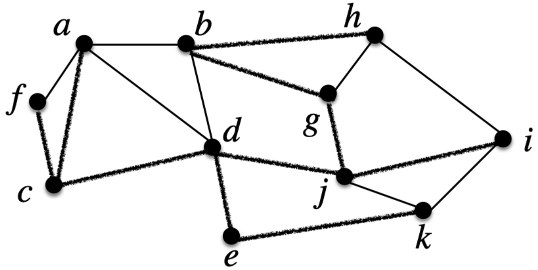
Overview

End-of-class details

- Recitation 10: automatic checkoff, do on your own, released shortly
- Homework 10: two problems + mindbender, released tomorrow, due May 5
- Review sessions before final exam (review material like for midterm)
- Final exam: 9-11am, May 12
  - Last names A-T: Salomon DECI
  - Last names U-Z: Salomon 001
  - Cumulative, similar structure to midterm
  - Review recitations, lecture material, problem sets!

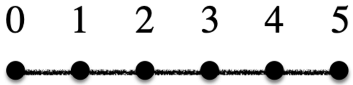
Random walks on graphs

Many problems can be modeled as a random walk on a graph.



- Cars on the road.
- Searchers on the web.
- Packets on a network.
- Spreading of disease.

Random walk problems



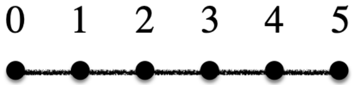
Idea:

- We start at some vertex in a graph.
- On each step, we walk to one of the adjacent vertices, chosen uniformly at random.
- Repeat.

Questions:

- What's the probability of reaching one vertex before another?
- How long until we reach some target vertex?
- What fraction of the time do we spend at each of the vertices?

Chain of resources



Let's associate with each vertex of a chain the amount of money we have. The random walk corresponds to, say, stock market investment or lottery or some other fair bet, with an equal probability of going up or down.

Let's say we start with $4 with the goal of getting out of the game if we reach $5, our target. However, if we drop to zero, we can't invest anymore and we go bust/broke/are ruined.

How likely are we to reach our target?

Linear recurrence

Let p_i be the probability of winning (reaching \$$n$) starting from \$$i$.

We peg $p_0 = 0$ (guaranteed loss) and $p_n = 1$ (guaranteed win).

What about other values of p_i? Note that the probability of winning with \$$i$ is independent of whether we start there or whether we reach that level after a number of successful and/or failed investments.

We can write

$$p_i = \frac{1}{2}p_{i-1} + \frac{1}{2}p_{i+1},$$

because we're equally likely to go up or down, and then proceed to win (or lose!) from there.

Solving the recurrence

Let's try some small cases to get a feel for it.

If $n = 1$, there's just $p_0 = 0$ and $p_1 = 1$.

If $n = 2$, starting from 1 is equally likely to win or ruin in one step, so $p_1 = 1/2$.

If $n = 3$, we have:

$$\begin{aligned} p_1 &= 1/2\ p_2 \\ p_2 &= 1/2\ p_1 + 1/2 \\ 2p_1 &= 1/2\ p_1 + 1/2 \\ 4p_1 &= p_1 + 1 \\ 3p_1 &= 1 \\ p_1 &= 1/3 \\ p_2 &= 2/3 \end{aligned}$$

Up to 4

Starting to see a pattern. Let's do one more ($n = 4$) to increase our confidence.

$$
\begin{aligned}
p_1 &= 1/2\ p_2 \\
p_2 &= 1/2\ p_1 + 1/2\ p_3 \\
p_3 &= 1/2\ p_2 + 1/2 \\
p_2 &= 1/4\ p_2 + 1/2\ p_3 \\
4p_2 &= p_2 + 2p_3 \\
3p_2 &= 2p_3 \\
3p_2 &= p_2 + 1 \\
p_2 &= 1/2 \\
p_1 &= 1/4 \\
p_3 &= 3/4
\end{aligned}
$$

Looking for a pattern

	p_0	p_1	p_2	p_3	p_4
$n = 1$	0	1			
$n = 2$	0	1/2	1		
$n = 3$	0	1/3	2/3	1	
$n = 4$	0	1/4	1/2	3/4	1

Maybe $p_i = i/n$?

Verify solution

$p_0 = 0/n = 0.$
$p_n = n/n = 1.$
p_i
$= 1/2\ p_{i-1} + 1/2\ p_{i+1}$
$= 1/2\ (i-1)/n + 1/2\ (i+1)/n$
$= i/n.$

$p_i = i/n$ works. Linear improvement as we move toward the target.

Note that we showed that we have a solution to the recurrence (kind of like induction), but we didn't show this solution is unique. (It is.)

Time to end

Now, we'll quit when we hit *either* end. How many steps, in expectation, does it take before that happens?

$$\begin{aligned} s_0 &= 0 \\ s_n &= 0 \\ s_i &= 1 + 1/2\ s_{i-1} + 1/2\ s_{i+1} \end{aligned}$$

$n = 1$: $s_0 = 0, s_1 = 0$.
$n = 2$: $s_0 = 0, s_1 = 1, s_2 = 0$.

Four chain

$n = 3$: $s_0 = 0, s_1 = x, s_2 = x, s_3 = 0.$

$$\begin{aligned} x &= 1 + 1/2 \ \cdot 0 + 1/2\, x \\ &= 1 + 1/2\, x \\ 1/2\, x &= 1 \\ x &= 2 \end{aligned}$$

Five chain

$n = 4$: $s_0 = 0, s_1 = x, s_2 = y, s_3 = x, s_4 = 0.$

$$\begin{aligned} x &= 1 + 1/2\, y \\ y &= 1 + x \\ x &= 1 + 1/2(1 + x) \\ x &= 3/2 + 1/2\, x \\ 1/2\, x &= 3/2 \\ x &= 3 \\ y &= 4 \end{aligned}$$

Six chain

$n = 5$: $s_0 = 0, s_1 = x, s_2 = y, s_3 = y, s_4 = x, s_5 = 0.$

$$
\begin{aligned}
x &= 1 + 1/2\, y \\
y &= 1 + 1/2\, x + 1/2\, y \\
2x &= 2 + y \\
2y &= 2 + x + y \\
y &= 2 + x \\
2x &= 2 + 2 + x \\
x &= 4 \\
y &= 6
\end{aligned}
$$

Looking for the pattern

	s_0	s_1	s_2	s_3	s_4	s_5	s_6
$n = 1$	0	0					
$n = 2$	0	1	0				
$n = 3$	0	2	2	0			
$n = 4$	0	3	4	3	0		
$n = 5$	0	4	6	6	4	0	
$n = 6$	0	5	8	9	8	5	0

Algebraic pattern

$s_1 = n - 1$. s_2 needs to be the thing that makes s_1 be the average of s_0 and s_2 plus one. And so on.

$$\begin{aligned} s_0 &= 0 \\ s_1 &= n - 1 \\ s_2 &= 2n - 4 \\ s_3 &= 3n - 9 \\ s_4 &= 4n - 16 \end{aligned}$$

Looks like: $s_i = in - i^2$.

Check it

$s_i = in - i^2$.

$s_{i-1} + s_{i+1}$
$= (i-1)n - (i-1)^2 + (i+1)n - (i+1)^2$
$= 2in - i^2 + 2i - 1 - i^2 - 2i - 1$
$= 2in - 2i^2 - 2$
$= 2(s_i - 1)$

s_i
$= 1 + 1/2\ s_{i-1} + 1/2\ s_{i+1}$

So, that works. Square factor this time. Could view it as $i(i-n)$ or the product of the distance from one end and the distance to the other end.

Long-term distribution

Imagine starting a random walk running on graph G, and then taking a long nap. You wake up. What's the probability that the random walk is currently at vertex i (d_i)?

$$d_i = \sum_{\{i,j\} \in E(G)} d_j \cdot 1/\deg(j),$$

$$\sum_{i \in V(G)} d_i = 1.$$

The idea is that, on each step, the chance of landing in vertex i can be calculated by taking the probability of being at each vertex j, and summing the likelihood of transitioning from j to i. That last quantity is $1/\deg(j)$, since j has one edge out of $\deg(j)$ that goes to i. Balance rule and distribution rule.

Small chain examples

$n = 1, d_0 = 1/2, d_1 = 1/2.$
$n = 2$: $d_0 = 1/4, d_1 = 1/2, d_2 = 1/4.$
$n = 3$:

$$\begin{aligned} d_0 &= 1/2\, d_1 \\ d_1 &= d_0 + 1/2\, d_2 \\ d_2 &= 1/2\, d_1 + d_3 \\ d_3 &= 1/2\, d_2 \end{aligned}$$

$d_1 = d_2, d_0 = d_3, d_1 = 2d_0$. So, $d_0 = d_3 = 1/6, d_2 = d_3 = 1/3.$

General chain of length n

$d_0 = d_n = 1/(2n)$.
$d_i = 1/n$.

$\sum_{i=0}^{n} d_i = 1/(2n) + (n-1)/n + 1/(2n) = 1$.

Note: For all i, $d_i \cdot 1/\deg(i) = 1/(2n)$ because the ends of the chain have degree 1 and $d_i = 1/(2n)$ and the internal vertices have degree 2 and $d_i = 1/n$.

So, for all i, d_i as defined by the balance rule and d_i in the solution match. That's because the ends of the chain are connected to one node (thus get $1/(2n)$) and the internal vertices are connected to two, each of which contributes $1/(2n) + 1/(2n) = 1/n$.

Likelihood is proportional to degree!

General graphs

This result holds for general graphs!

Let $D = \sum_{i \in V(G)} \deg(i)$.

Let $d_i = \deg(i)/D$.

Distribution rule:
$\sum_{i \in V(G)} d_i$
$= \sum_{i \in V(G)} \deg(i)/D = D/D = 1.$

Balance rule: ($d_i = \sum_{\{i,j\} \in E(G)} d_j \cdot 1/\deg(j)$)
For any vertex j, $d_j \cdot 1/\deg(j) = \deg(j)/D \cdot 1/\deg(j) = 1/D$. So, the contribution from each of a vertex's $\deg(i)$ neighbors is $1/D$, for a total of $\deg(i)/D$, exactly our solution for d_i.

Thus, the amount of time a random walk spends at a vertex is proportional to its degree, *independent of the structure of the graph*!

Final thoughts

There are many caveats, which you'd study in a stochastic process class.

If we consider directed graphs, where the distribution over neighbors can be non-uniform, we get a well-studied model called the Markov chain.

If you additionally allow a set of possible distributions at each vertex and the random walker can choose which one to follow, and there's a notion of rewards on vertices that the random walker wants to maximize, then you get a model called the Markov decision process.

It's summertime!

We've seen a lot of topics this semester. Remember why we've done this:

- Vocabulary. Use the languages of logic, combinatorics, probability, … as a shared, precise vocabulary for discussing problems.
- Abstraction. A lot of the problems we've studied will show up in different contexts, in and out of computer science. Remember our abstract solutions and adapt them to reality.
- Team problem solving. CS is collaborative, and hopefully you've gotten practice solving problems with a team.

- Lean: there is a formal concept of what a proof "is." Proving and programming are not totally separate concepts! Making the rules of the proof game explicit can affect (and improve) how we write informal proofs.