Random Walks; Walking Matrices and Diffusion Problems

Adi Khera

October 14, 2023

Overview

- A process that describes a sequential path over random steps within a graph

Overview

- A process that describes a sequential path over random steps within a graph
  - The diffusion of a gas over a space
  - Search path of a predator
  - The price of a fluctuating stock

Overview

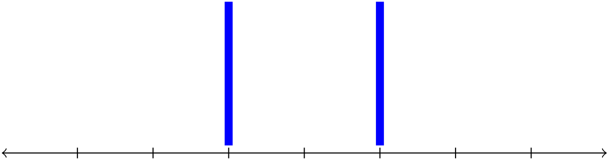
- A process that describes a sequential path over random steps within a graph
  - The diffusion of a gas over a space
  - Search path of a predator
  - The price of a fluctuating stock
- Can help with understanding social networks and convergence problems, toy example to follow

Unweighted 1-Dimensional

- Imagine a number line centered at zero
- For every flip of a fair-sided coin $+1$ or -1

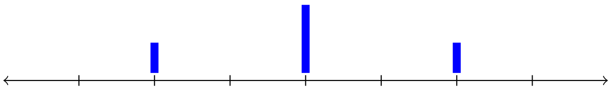
Unweighted 1-Dimensional

- Imagine a number line centered at zero
- For every flip of a fair-sided coin +1 or -1
- What does the path look like after **one** flip?



Unweighted 1-Dimensional

- Imagine a number line centered at zero
- For every flip of a fair-sided coin +1 or -1
- What does the path look like after **two** flips?



Unweighted 1-Dimensional

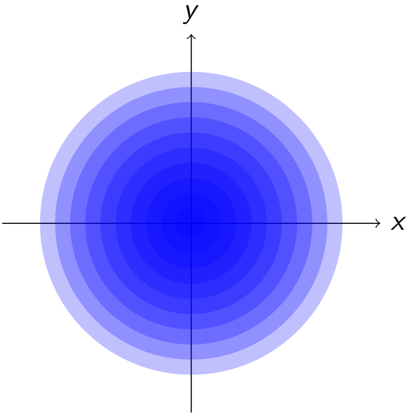
- Imagine a number line centered at zero
- For every flip of a fair-sided coin +1 or -1
- What does the path look like after **n** flips?

Unweighted 2-Dimensional

- Imagine a Cartesian plane, subject can move 4 directions
- Overtime, there is a convergence around the origin

Unweighted 2-Dimensional

- Imagine a Cartesian plane, subject can move 4 directions
- Overtime, there is a convergence around the origin



Unweighted 2-Dimensional

- Imagine a Cartesian plane, subject can move 4 directions
- Overtime, there is a convergence around the origin

A random walk is recurrent if it **will** return to its starting position

A transient walk is recurrent if it **will not** return to its starting position

drunk man will find his way home, but a drunk bird may get lost forever!

Recap and Definitions

- Let Graph $G = (V, E, w)$ be a weighted undirected graph
  - M is the adjacency matrix of Graph G
  - A is the *normalized* adjacency matrix of Graph G
  - D is the diagonal matrix of the degrees of Graph G
- Random walk represents the probability distribution over vertices after a certain number of steps/time

Probabilistic Setup

- When G is unweighted, there is a uniform probability of movement to neighbor u
- When G *is* weighted, probability is proportional to the weight of the corresponding edge (u, v)

Probabilistic Setup

- When G is unweighted, there is a uniform probability of movement to neighbor u
- When G *is* weighted, probability is proportional to the weight of the corresponding edge (u, v)
- $p_t(u)$ represents probability of being at vertex u at time t
- Probability vector p satisfies $p(u) \geq 0$ for all $u \in V$ and $\sum_u p(u) = 1$

Probabilistic Setup

- Initial probability distribution is centered around one vertex v, so $p_0(v) = 1$
- To derive the probability of existing at vertex u at time $t+1$,

$$p_{t+1}(u) = \sum_{v:(u,v)\in E} \frac{w(u,v)}{d(v)} p_t(v)$$

- Pretty intuitive, quasi-bayesian formula where d(v) is weighted degree of vertex

Lazy Walk

- Now imagine a walk with half chance of remaining at current vertex, half chance of moving to neighbor
- Probability function now becomes,

$$p_{t+1}(u) = (1/2)p_t(u) + (1/2) \sum_{v:(u,v)\in E} \frac{w(u,v)}{d(v)} p_t(v)$$

- Extremely powerful in diffusion or chemical real-world problems

Matrix Setup

- We can rewrite the random walk and lazy random walk in matrix form as follows

$$p_{t+1}(u) = \sum_{v:(u,v)\in E} \frac{w(u,v)}{d(v)} p_t(v)$$

$$W = (A_G D_G^{-1})$$

$$p_{t+1}(u) = (1/2)p_t(u) + (1/2) \sum_{v:(u,v)\in E} \frac{w(u,v)}{d(v)} p_t(v)$$

$$\widetilde{W} = (1/2)(I + A_G D_G^{-1})$$

Matrix Setup

- Walking Matrix is **not symmetric**, but is **similar** to symmetric matrices
- Define our normalized adjacency matrix as

$$A := D^{-1/2} W D^{1/2} = D^{-1/2} M D^{-1/2}$$

Matrix Setup

- Walking Matrix is **not symmetric**, but is **similar** to symmetric matrices
- Define our normalized adjacency matrix as

$$A := D^{-1/2} W D^{1/2} = D^{-1/2} M D^{-1/2}$$

- Thus, vector ψ is an eigenvector of A of eigenvalue ω if and only if $D^{1/2}\psi$ is an eigenvector of W of eigenvalue ω.
- W and $\widetilde{W}$ must share eigenvalues so both are similar to **symmetric** matrices

Matrix Set Up

- Using **Perron–Frobenius** theorem, we have a unique largest eigenvalue ω_1 and the eigenvalues of W are between 1 and 0
- We can also assert that

$$1 = \omega_1 \geq \omega_2 \geq \cdots \omega_n \geq 0$$

- The proof of this is saved for times sake

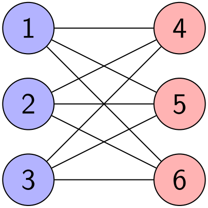
Stable Distribution

- Regardless of initial distribution, a lazy random walk on a connected graph always converges to one distribution - the *stable distribution*
- The vector encoding this distribution is π

$$\pi(i) = \frac{d(i)}{\sum_j d(j)}$$

Commentary on Stable Distribution

- Recall $d(v)$ is the weighted degree of vector v
- Certain graphs would not converge in a non-lazy random path
  - Bipartite graphs would bounce between sets with parity t without convergence



Proof of Stable Distribution

$$D^{-1/2}p_0 = \sum_i c_i v_i$$

$$c_1 = v_1^T(D^{-1/2}p_0) = \frac{(d^{1/2})^T}{||d^{1/2}||}(D^{-1/2}p_0) = 1^T p_0 = \frac{1}{||d^{1/2}||}$$

Proof of Stable Distribution

$$p_t = W^t p_0$$

$$= (D^{1/2}(I/2 + M/2)D^{-1/2})^t p_0$$

$$= (D^{1/2}(I/2 + M/2)^t D^{-1/2})p_0$$

$$= D^{1/2}(I/2 + M/2)^t \sum_i c_i v_i$$

$$= D^{1/2} \sum_i \omega_i^t c_i v_i$$

$$= D^{1/2} c_1 v_1 + D^{1/2} \sum_{i \geq 2} \omega_i^t c_i v_i$$

since $\omega_i < 1$ for all $i \geq 2$, the right hand goes to zero over time

Proof of Stable Distribution

$$= D^{1/2} c_1 v_1 + D^{1/2} \sum_{i \geq 2} \omega_i^t c_i v_i$$

$$D^{1/2} c_1 v_1 = D^{1/2} \left(\frac{1}{||d^{1/2}||} \right) \frac{d^{1/2}}{||d^{1/2}||^2}$$

$$= \frac{d}{||d^{1/2}||^2} = \frac{d}{\sum_j d(j)} = \pi$$

Rate of Convergence

Assuming a starting point on $a \in V$ for every vertex b the random walk is bound by

$$|p_t(b) - \pi(b)| \leq \sqrt{\frac{d(b)}{d(a)}}\omega_2^t$$

Meaning as time t progresses we will converge upon the stable distribution at a rate $\propto \sqrt{\frac{d(b)}{d(a)}}\omega_2^t$

Commentary on Rate of Convergence

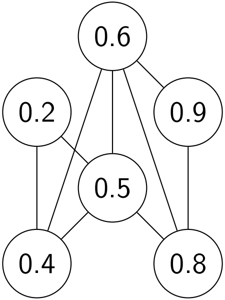
As previously discussed, the ratio of the largest to second largest eigenvalues determined how quickly we would expect the graph to converge to the top eigenvector (if $\lambda_1 \sim \lambda_2$, then we need more iterations)

The dynamics of a well connected-graph follow

- They will converge quickly
- The second smallest eigenvalue of the Laplacian will be large

Social Networks and Opinion Mining

- Assume we have a social network of 6 friends with different views on masking in groceries stores
- After each week the opinions of each person becomes the average of their friends, how quickly will the opinions converge



Social Networks and Opinion Mining

- If we use matrices to solve this problem, the equation becomes $v_{t+1} = D^1 A v_t$
- This would be cumbersome to solve, and require constant updating, if we didn't recognize this as an eigenvector problem!
- If v converges it must satisfy $v = D^1 A v$, so v must be eigenvector of $D^1 A$ with eigenvalue 1

Circular Table Example