Random Walking With A Purpose

Trenton W. Ford

CSE 60742

Graph Kernel: Random Walk

What is a random walk?

Given a graph $G(V, E)$ where V is a set of vertices, and E is a set of edges.

1. Select an arbitrary starting node u in G.
2. Randomly select a neighbor of u, say v.
3. Move across the edge (u, v), let $u < - v$ and repeat steps 2 and 3... or stop.

Interesting Fact: If we walk long enough, the probability of being at a particular vertex and any given time is proportional to the degree of that vertex.

Random Walk Applications

- Modeling diffusion (Brownian, epidemics, mosquitos, etc.)
- Sampling from large networks (embeddings, grammars, etc.)
- Fun games (Conway's Game of Life… others?)

Random Walk: “Embarrassingly Parallel”

Naive Approach: Given a graph G

1. Create k copies of G
2. Give a copy to each of k random walkers
3. When they are done walking, merge their results

This process works even when the traversal probabilities aren’t uniformly distributed.

Random Walk: Complexity

- Determining complexity requires a stopping criteria. When is the walk over?
  - Vertex Coverage (seen it all)
  - Commute Time (there and back again)
  - Mixture Rate (convergence towards vertex degree)
  - Many more

Random Walk: Vertex Cover Complexity

- Depends on the connectivity of the graph, and the transition probabilities.
  - Example: Given a complete and undirected graph G, of order n compute the algorithmic complexity.

Graph Kernel: Random Walk

Imagine you’re drunk…

You want to go home…

But you aren’t sure how…

Random Walking + Dynamic Weights

- Again… You're drunk…
- Imagine making wrong turns and for a short while you'll remember not to do it again.
- Naive Parallel Algorithm Doesn't Share Knowledge
- Yell to the other drunken random walkers about which roads to avoid

Random Walking + Dynamic Weights

Pseudocode:

1. Learn unweighted walk statistics for the entire graph. (shape, degree distribution, centrality measures, maybe some new ones.)

2. Use statistics to generate minimum crosstalk partitions that are of similar size. (maybe min-cut like). Dynamic granularity would be nice as well.

3. Partition nodes and adjacent edges to different random walker processes.

4. Allow them to communicate with each other only when it is important to another walker.

5. Combine their results, which each has reached the stopping criteria.

Random Walking + Dynamic Weights: Complexity

Complexity analysis: It depends on the purpose(stopping criteria) of the walk.

- Random walker crosstalk cost will be an important metric of partitioning success.

Random Walking + Dynamic Weights: Implementation Plans

Language: Parallel Boost

Data: Any dataset with interesting networks. Multiple types of networks should be used to strengthen partitioning criteria.

Questions?

Random Walking: Brownian Particle Diffusion

- **Brownian motion**, also called **Brownian movement**, any of various physical phenomena in which some quantity is constantly undergoing small, random fluctuations. It was named for the Scottish botanist Robert Brown, the first to study such fluctuations (1827).

Random Walking + Dynamic Weights: Implementation Plans

Why not matrices Satyaki?

Because there is value is a more easily understood implementation.