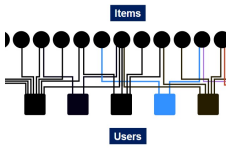
Random Walk with Restarts and Personalized PageRank

Example: Recommendation

A bipartite graph representing user and item interactions (e.g. purchase)

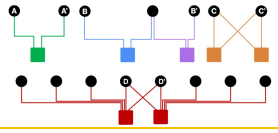


Bipartite User-Item Graph

problem: What items should we recommend to a user who interacts with item Q?

Intuition: if items Q and P are interacted by similar users, recommend P when user interacts with Q

problem: Which is more related A,A' , B,B' or C,C' ?



- Shortest path
- Common Neighbors

Node **proximity** measurement

3 algos

Teleports

- PageRank — any node — same prob — [0.2, 0.2, 0.2, 0.2, 0.2]
- Personalized PageRank (Topic-Specific) — subset S — [0.4, 0, 0.4, 0, 0.2]
- Random walk with restarts (Proximity on graphs) — starting node — [1, 0, 0, 0, 0]

diff

Random Walks

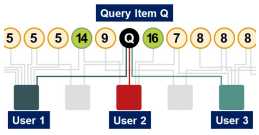
Idea
- Every node has some importance
- Importance gets evenly split among all edges and pushed to the neighbors:

Given a set of **QUERY_NODES**, we simulate a random walk

steps
1. start from query node
2. make a random walk 1 hop
3. add count to the visited node
4. decide whether to continue the walk or teleport back to query node based on prob alpha

```
item = QUERY_NODES.sample_by_weight( )
for i in range( N_STEPS ):
    user = item.get_random_neighbor( )
    item = user.get_random_neighbor( )
    item.visit_count += 1
    if random( ) < ALPHA:
        item = QUERY_NODES.sample.by_weight ( )
```

vis



benefits
- measure proximity by random walks — Number of visits by random walks starting at Q
- simple
- scalable
- ▪ Multiple connections ▪ Multiple paths ▪ Direct and indirect connections ▪ Degree of the node