1. Exploratory Random Walk

We start with a couple definitions.

Definition 1 (Bipartite Graph Instance)**.** Let $RBG(V, W, \mathbf{p})$ be a random bipartite graph, we call an instance of $RBG(V, W, \mathbf{p})$ for a given realization of the family $\xi = \{\xi_{v,w}\}_{v \in V, w \in W}$, a *bipartite instance graph* $\mathcal{B}(V, W, \xi)$.

Definition 2 (Remainder Graphs)**.** Let $\mathcal{B}(V, W, \xi)$ be a bipartite instance graph, and let $I \subset W$. The *remainder graph*

$$\mathcal{B}|_I = \mathcal{B}(V|_I, W \setminus I, \xi|_I)$$

is the bipartite instance graph on:

- $W \setminus I$
- $V|_I = V \setminus \left(\bigcup_{i \in I} \{v \in V : \xi_{v,i} = 1\} \right)$
- $\xi|_I = \begin{cases} \xi_{u,v} & \text{if } i \in W \setminus I, u \in V|_I \\ 0 & \text{otherwise} \end{cases}$

As is canon in the theory of random graphs[**?**, **?**, **?**], we will now define a procedure for revealing the connected component to which a given node belongs.

Our process will explore the bipartite graph whose projection yields the dual random intersection graph. We define the procedure recursively using the remainder graphs defined above. Given a bipartite graph between users and items, we explore the connected components of the dual random intersection graph by exploring a recursive sequence of bipartite graphs using a generational breadth first search. Our graph exploration process will therefore yield a sequence of graphs, a queue of *pre-actives*, and sequences of *active*, and *removed* sets. The pre-active queue enforces the generational exploration of the component. We start by choosing an item, and finding its neighbors via the users that pick it. We push that set of neighbors into our pre-active queue, and produce the remainder graph obtained by removing the chosen item from the bipartite graph. This is now our new bipartite graph on which we will perform the next step of our exploration. We finalize our initial step, by adding our initially chosen item to the set of removed. In all subsequent steps, we are in one of the following 3 cases:

stop: : we do this if both the active set and pre-active queue is empty;
switch generation: : we do this if the set of active is empty by popping a set out of the queue of pre-actives;
uncover an item: : this corresponds to picking the next node to explore from the actives.

Definition 3 (Graph Exploration Process)**.** At step $t = 0$, $\mathcal{B}_0 = \mathcal{B}(V, W, \xi)$ and:

- j_0: is the node whose connected component we want to explore;
- U: is the set of unexplored items, i.e. $W \setminus \{j_0\}$;
- $V(j_0)$: is the set of users who picked j_0, i.e. $V(j_0) = \{v \in V : \xi_{v,j_0} = 1\}$;
- $\mathcal{N}$: is the set of items, not including j_0, that the above users picked, i.e. $\mathcal{N} = \bigcup_{u \in V(j_0)} \{j \in U : \xi_{u,j} = 1\}$;
- P_0: is an empty FIFO queue of sets, to which we push $\mathcal{N}$;
- R_0: is the set of removed nodes, i.e. $R_0 = \{j_0\}$;
- A_0: is the empty set of active nodes.

At step $t + 1$, $\mathcal{B}_{t+1} = \mathcal{B}_t|_{\{j_t\}}$:

- we update the set of actives and pre-actives if needed in order to pick j_{t+1}:

$$(A_t, P_t) = \begin{cases} A_t & \text{if } A_t \neq \emptyset \\ pop(P_t) & \text{otherwise} \end{cases}$$

- we pick j_{t+1} from the set of actives A_t and set $A_{t+1} = A_t \setminus \{j_{t+1}\}$;
- U: is the set of unexplored items, i.e. $(W \setminus R_t) \setminus \{j_{t+1}\}$;
- $V(j_{t+1})$: is the set of users who picked j_{t+1} in the remainder graph $\mathcal{B}_{t+1}$, i.e. $V(j_{t+1}) = \{v \in V|_{R_t} : \xi_{v,j_{t+1}} = 1\}$;
- $\mathcal{N}$: is the set of items, not including j_{t+1}, that the above users picked, i.e. $\mathcal{N} = \bigcup_{u \in V(j_{t+1})} \{j \in U : \xi_{u,j} = 1\}$;
- P_{t+1}: is the P_t to which we push $\mathcal{N}$ only if $\mathcal{N} \neq \emptyset$;

- $R_{t+1} = R_t \cup \{j_{t+1}\}$.

The process stops whenever both A_t and P_t are empty.

References