RecWalk: Nearly Uncoupled Random Walks for Top-N Recommendation

paper

code : julia 15+ stars

code : python 3+ stars

benchmark : movielens, yahoo! R2Music, pintrest

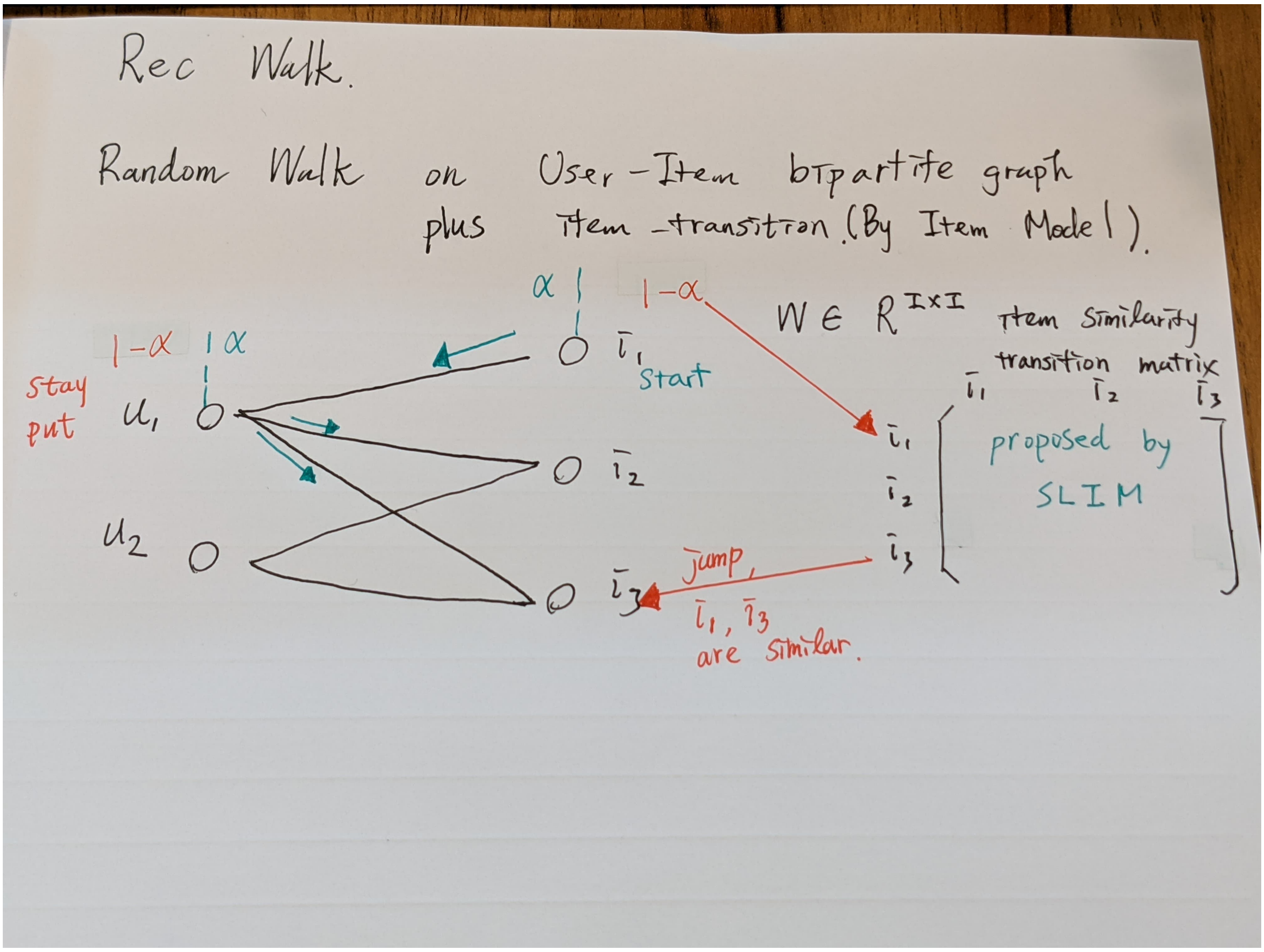
Abstract

1. claim : RecWalk leverages the spectral properties of *nearly uncoupled Markov chains* to provably lift this limitation and prolong the infliuence of user's past preferneces on the seccessive steps of the walk.

2. We do actually achieve a significant improvement on 3 dataset above!

3. Randomwalk + ItemModel(proposed in SLIM)

Idea



Algorithm 2 RecWalk$^{\text{K-step}}$

Input: RecWalk model $\mathbf{P}$, user $u \in \mathcal{U}$.
Output: Recommendation vector $\boldsymbol{\pi}_u$.
$\boldsymbol{\pi}_u^\top \leftarrow \mathbf{e}_u^\top$
for $k \in 1, \ldots, K$ **do**
    $\boldsymbol{\pi}_u^\top \leftarrow \boldsymbol{\pi}_u^\top \mathbf{P}$
end for

There are K-steps and PR methods.

K-steps : you can run any times of random walk jumping.

PR methods : theorically coverage(although we still need to set a tolerance)

Notations :

1. $A_{G} \in \R^{U+I \times U+I}$: adjacency matrix.

We define $\mathcal{G} \triangleq (\{\mathcal{U}, \mathcal{I}\}, \mathcal{E})$ to be the *user-item bipartite network*; i.e the network with adjacency matrix $\mathbf{A}_{\mathcal{G}} \in \Re^{(U+I)\times(U+I)}$ given by

$$\mathbf{A}_{\mathcal{G}} \triangleq \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^{\top} & \mathbf{0} \end{pmatrix}. \tag{3}$$

2. $P \in R^{U+I \times U+I}$: mixed transition matrix of simple random walk and inter-item transition probability

2.3 The Transition Probability Matrix

The transition probability matrix $\mathbf{P}$ that governs the behavior of our random walker can be usefully expressed as a weighted sum of two stochastic matrices $\mathbf{H}$ and $\mathbf{M}$ as

$$\mathbf{P} \triangleq \alpha\mathbf{H} + (1-\alpha)\mathbf{M} \tag{4}$$

3. $H \in R^{U+I \times U+I}$: transition matrix(maybe normalized by outdegree)
4. $M in \R^{U+I \times U+I}$: inter-item transition probablity matrix

$$\mathbf{H} \triangleq \mathrm{Diag}(\mathbf{A}_{\mathcal{G}}\mathbf{1})^{-1}\mathbf{A}_{\mathcal{G}}. \tag{5}$$

Matrix $\mathbf{M}$, is defined as

$$\mathbf{M} \triangleq \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathcal{I}} \end{pmatrix} \tag{6}$$

$$\mathbf{M}_{\mathcal{I}} \triangleq \frac{1}{\|\mathbf{W}\|_{\infty}}\mathbf{W} + \mathrm{Diag}(\mathbf{1} - \frac{1}{\|\mathbf{W}\|_{\infty}}\mathbf{W}\mathbf{1}). \tag{7}$$

5. $e_{u} \in R^{U+I}$: start vector contains the element 1 on the position that corresponds to user u and zeros elsewhere.
6. π_{u} recommendation vector of user u

Result

Other Discussion

From source code python 3+ stars

we know the training phase (actually no training ..., it's need a item model trained from slim)

then build matrix.