# RecWalk: Nearly Uncoupled Random Walks for Top-N Recommendation

Athanasios N. Nikolakopoulos
Digital Technology Center, University of Minnesota
anikolak@umn.edu

George Karypis
Computer Science & Engineering, University of Minnesota
karypis@umn.edu

## ABSTRACT

*Random walks* can provide a powerful tool for harvesting the rich network of interactions captured within *item-based models* for top-$n$ recommendation. They can exploit indirect relations between the items, mitigate the effects of sparsity, ensure wider itemspace coverage, as well as increase the diversity of recommendation lists. Their potential however, is hindered by the tendency of the walks to rapidly concentrate towards the central nodes of the graph, thereby significantly restricting the range of $K$-step distributions that can be exploited for personalized recommendations. In this work we introduce **RecWalk**; a novel random walk-based method that leverages the spectral properties of *nearly uncoupled Markov chains* to provably lift this limitation and prolong the influence of users' past preferences on the successive steps of the walk—allowing the walker to explore the underlying network more fruitfully. A comprehensive set of experiments on real-world datasets verify the theoretically predicted properties of the proposed approach and indicate that they are directly linked to significant improvements in top-$n$ recommendation accuracy. They also highlight RecWalk's potential in providing a framework for boosting the performance of item-based models. RecWalk achieves state-of-the-art top-$n$ recommendation quality outperforming several competing approaches, including recently proposed methods that rely on deep neural networks.

**ACM Reference Format:**
Athanasios N. Nikolakopoulos and George Karypis. 2019. RecWalk: Nearly Uncoupled Random Walks for Top-N Recommendation. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291016

## 1 INTRODUCTION

Top-$n$ recommendation algorithms provide ranked lists of items tailored to the particular tastes of the users, as depicted by their past interactions within the system. Over the past years they have become an indispensable component of most e-commerce applications as well as content delivery platforms.

*Item-based methods* are among the most popular approaches for top-$n$ recommendation [11, 33, 38]. Such methods work by building a model that captures the relations between the items, which is then used to recommend new items that are "close" to the ones each user has consumed in the past. Item-based models have been shown to achieve high top-$n$ recommendation accuracy [33, 38] while being scalable and easy to interpret [11]. The fact, however, that they typically consider only direct inter-item relations can impose fundamental limitations to their quality and make them brittle to the presence of sparsity—leading to poor itemspace coverage and significant decay in performance [3]. *Random-walk-based methods* are particularly well-suited for alleviating such problems. Having the innate ability to relate items that are not directly connected by propagating information along the edges of the underlying graph, random walk methods are more robust to the effects of sparsity and they can afford better coverage of the itemspace. However their effectiveness in terms of top-$n$ recommendation can be limited by the tendency of the walks to concentrate probability mass towards the central nodes of the graph—thus disproportionately boosting the recommendation scores of popular items in the system. This means that in order to produce high quality recommendations random-walk-based techniques are restricted to exploit just the first few steps of the walk that are still influenced by the personalized starting distribution. This is in accordance to the mathematical properties of random walks and it has also been empirically verified that when applied to real-world networks, short-length random walks typically work best [7, 8, 12].

In this work we introduce **RecWalk**; a novel framework for top-$n$ recommendations that aims to combine the potential of item-based models to discern meaningful relations between the items, with the inherent ability of random walks to diffuse these relations across the itemspace and exploit the rich network of interactions they shape. RecWalk produces recommendations based on a *random walk with node-dependent restarts* designed to prolong the influence of the personalized initialization on the successive $K$-step landing probabilities of the walk—thereby eliminating the need of clipping the walks early. Intuitively, this gives the walker "more time" to harvest the information captured within the item model before succumbing to the "pull" of central nodes. The advocated random walk construction leverages the spectral properties of *nearly uncoupled Markov chains* [9] in order to enforce a *time-scale dissociation* of the stochastic dynamics of the walk towards equilibrium—thus increasing the number of successive landing probabilities that carry personalized information useful for top-$n$ recommendation. The properties of our model are backed by rigorous theoretical analysis of the mixing characteristics of the walk which we empirically verify that are indeed intertwined with top-$n$ recommendation accuracy. A comprehensive set of experiments on real-world datasets showcase the potential of the proposed methodology in providing a framework for boosting the performance of item models. RecWalk achieves


*WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia*

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3291016

high recommendation quality outperforming state-of-the-art competing approaches, including recently proposed methods relying on deep neural networks. Open source implementation of the method is available at: https://github.com/nikolakopoulos/RecWalk.

## 2 PROPOSED METHOD

**Definitions.** Let $\mathcal{U} = \{1, \dots, U\}$ be a set of *users* and $\mathcal{I} = \{1, \dots, I\}$ a set of *items*. Let $\mathbf{R} \in \Re^{U \times I}$ be the *user-item interaction matrix*; i.e. the matrix whose $ui$-th element is 1 if user $u$ has interacted with item $i$, and 0 otherwise. Each user $u \in \mathcal{U}$ is modeled by a vector $\mathbf{r}_u^\top \in \Re^I$ which coincides with the corresponding row of the user-item interaction matrix $\mathbf{R}$; similarly, each item $i \in \mathcal{I}$ will be modeled by a vector $\mathbf{r}_i \in \Re^U$ which coincides with the corresponding column of matrix $\mathbf{R}$. The rows and columns of $\mathbf{R}$ are assumed to be non-zero; i.e. every user has interacted with at least one item, and for every item there exists at least one user who has interacted with it. Finally, we use the term *item model* to refer to a matrix $\mathbf{W} \in \Re^{I \times I}$ the $ij$-th element of which gives a measure of *proximity* or *similarity* between items $i$ and $j$.

### 2.1 Random Walks and Item Models

The fundamental premise of the present work is that combining random walks and item models allows for more effective utilization of the information captured in the item model; considering direct as well as transitive relations between the items, and also alleviating sparsity related problems. However directly applying random walks on item models can lead to a number of problems that arise from their inherent mathematical properties and the way these relate to the underlying top-$n$ recommendation task.

Imagine of a random walker "jumping" from node to node on an item-to-item graph with transition probabilities proportional to the inter-item proximity scores depicted by an item model. If the starting distribution of this walker reflects the items consumed by a particular user $u$ in the past, the probability the walker "lands" on different nodes after $K$ steps provide an intuitive measure of proximity that can be used to rank the nodes and recommend items to user $u$ accordingly.

Specifically, if $\mathbf{W}$ denotes the item model and $\mathbf{S} \triangleq \mathrm{Diag}(\mathbf{W}\mathbf{1})^{-1}\mathbf{W}$ the transition probability matrix of the walk, personalized recommendations for each user $u$ can be produced e.g. by utilizing the *$K$-step landing probability distributions* of a walk rooted on the items consumed by $u$:

$$\boldsymbol{\pi}_u^\top \triangleq \boldsymbol{\phi}_u^\top \mathbf{S}^K, \qquad \boldsymbol{\phi}_u^\top \triangleq \frac{\mathbf{r}_u^\top}{\|\mathbf{r}_u^\top\|_1} \tag{1}$$

or by computing the stationary distribution of a *random walk with restarts* on $\mathbf{S}$, using $\boldsymbol{\phi}_u^\top$ as the restarting distribution. The latter approach is the well-known *personalized PageRank* model [34] with *teleportation vector* $\boldsymbol{\phi}_u^\top$ and *damping factor* $p$, and its stationary distribution can be expressed [22] as

$$\boldsymbol{\pi}_u^\top \triangleq \boldsymbol{\phi}_u^\top \sum_{k=0}^{\infty} (1-p)p^k \mathbf{S}^k. \tag{2}$$

Clearly, both schemes harvest the information captured in the $K$-step landing probabilities $\{\boldsymbol{\phi}_u^\top \mathbf{S}^k\}_{k=0,1,\dots}$. In the former case, the recommendations are produced by using a fixed $K$; in the latter case they are computed as a weighted sum of all landing probabilities, with the $K$-step's contribution weighted by $(1-p)p^k$. But, how do these landing probabilities change as the number of steps $K$ increases? For how long will they still be significantly influenced by user's prior history as depicted in $\boldsymbol{\phi}_u^\top$?

When $\mathbf{S}$ is irreducible and aperiodic—which is typically the case in practice—the landing probabilities will converge to a *unique* limiting distribution irrespectively of the initialization of the walk [15]. This means that for large enough $K$ the $K$-step landing probabilities will no longer be "personalized" in the sense that they will become independent of the user-specific starting vector $\boldsymbol{\phi}_u^\top$. Furthermore, long before reaching equilibrium, the usefulness of these vectors in terms of recommendation will start to decay as more and more probability mass gets concentrated to the central nodes of the graph—thereby restricting the number of landing probability distributions that are helpful for personalized recommendation. This imposes a fundamental limitation to the ability of the walk to properly exploit the information encoded in the item model.

Motivated by this, here we propose **RecWalk**; a novel random-walk model designed to give control over the stochastic dynamics of the walk towards equilibrium; provably, and irrespectively of the dataset or the specific item model onto which it is applied. In RecWalk the item model is incorporated as a direct inter-item transition component of a walk on the user-item bipartite network. This component is followed by the random walker with a fixed probability determined by a model parameter that controls the *spectral characteristics* of the underlying walk. This allows for effective exploration of the item model while the influence of the personalized initialization on the successive landing probability distributions remains strong. Incorporating the item model in a walk on the user-item graph (instead of the item graph alone) is crucial in providing control over the mixing properties; and as we will see in the experimental section of this work such mixing properties are intimately linked with top-$n$ recommendation accuracy.

### 2.2 The RecWalk Stochastic Process

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

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

Consider a random walker jumping from node to node on $\mathcal{G}$. Suppose the walker currently occupies node $c \in \mathcal{U} \cup \mathcal{I}$. In order to determine her next step transition she flips a biased coin that yields heads with probability $\alpha$ and tails with probability $(1-\alpha)$:

1. If it turns *heads*, then:
   (a) if $c \in \mathcal{U}$, the walker jumps to one of the items rated by the user corresponding to node $c$ uniformly at random;
   (b) if $c \in \mathcal{I}$, the walker jumps to one of the users that have rated the current item uniformly at random;
2. If it turns *tails*, then:
   (a) if $c \in \mathcal{U}$, the walker stays put;
   (b) if $c \in \mathcal{I}$, the walker jumps to a related item abiding by an *inter-item transition probability matrix* (to be explicitly defined in the following section).

The stochastic process that describes this random walk is defined to be a homogeneous discrete time Markov chain with state space

$\mathcal{U} \cup \mathcal{I}$; i.e. the transition probabilities from any given node $c$ to the other nodes are fixed and independent of the nodes visited by the random walker before reaching $c$.

### 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

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

where $0 < \alpha < 1$, is a parameter that controls the involvement of these two components in the final model. Matrix $\mathbf{H}$ can be thought as the transition probability matrix of a simple random walk on the user-item bipartite network. Since every row and column of matrix $\mathbf{R}$ are non zero, matrix $\mathbf{H}$ is well-defined and it can be expressed as

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

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

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

where $\mathbf{I} \in \Re^{U \times U}$ the identity matrix and $\mathbf{M}_{\mathcal{I}} \in \Re^{I \times I}$ an inter-item transition probability matrix designed to capture relations between the items. In particular, given an item model with non-negative weights $\mathbf{W}$ we define this matrix using the following stochasticity adjustment strategy:

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

The first term divides all the elements by the maximum row-sum of $\mathbf{W}$ and the second enforces stochasticity by adding residuals to the diagonal, appropriately. The motivation behind this definition is to retain the information captured by the relative differences of the inter-item relations in $\mathbf{W}$, ensuring that $[\mathbf{W}]_{ij} \geq [\mathbf{W}]_{i'j'} \Rightarrow [\mathbf{M}_{\mathcal{I}}]_{ij} \geq [\mathbf{M}_{\mathcal{I}}]_{i'j'}$ for all $i \neq j, i' \neq j'$. This prevents items that are loosely related to the rest of the itemspace to disproportionately influence the inter-item transitions and introduce noise to the model.

#### 2.3.1 *Choice of the core Item-model.*

The construction of matrix $\mathbf{W}$ itself can be approached in several ways depending on the available information, the characteristics of the underlying recommendation problem, the properties of matrix $\mathbf{R}$, etc. The fact that random walk methods can achieve naturally item-space coverage allows us to define this component in a way that *enforces locality* in the relations between the items, having also the advantage to be easy to compute. In particular, we propose the use of a locally restricted variant of the well-known SLIM method [33] that is forced to consider only fixed-size neighborhoods when learning relations between the items. Concretely, for any given item $i$ we find the set of its $C$ closest neighbors (in terms of cosine similarity between their vector representations) and we form a matrix $\mathbf{N}_i \in \Re^{I \times C}$, by selecting the corresponding columns of the initial matrix $\mathbf{R}$. We then solve for each item the optimization problem

$$\begin{aligned} &\underset{\mathbf{x} \in \Re^C}{\text{minimize}} && \tfrac{1}{2}\|\mathbf{r}_i - \mathbf{N}_i\mathbf{x}\|_2^2 + \gamma_1\|\mathbf{x}\|_1 + \tfrac{1}{2}\gamma_2\|\mathbf{x}\|_2^2 \\ &\text{subject to} && \mathbf{x} \geq 0 \end{aligned} \tag{8}$$

and we fill the corresponding elements in the $i$-th column of matrix $\mathbf{W}$. The complete procedure for building the RecWalk model is given in Algorithm 1.

### 2.4 Recommendation Strategies

Having defined the RecWalk transition probability matrix we can produce recommendations by exploiting the information captured in the successive landing probability distributions of a walk initialized in a user-specific way. Here, we will consider two recommendation strategies; namely

**RecWalk$^{\text{K-step}}$:** The recommendation score of user $u$ for item $i$ is defined to be the probability the random walker lands on node $i$ after $K$ steps, given that she started on node $u$. Therefore the recommendation score for item $i$ is given by the corresponding elements of

$$\boldsymbol{\pi}_u^\top \triangleq \mathbf{e}_u^\top\mathbf{P}^K \tag{9}$$

where $\mathbf{e}_u \in \Re^{U+I}$ is a vector that contains the element 1 on the position that corresponds to user $u$ and zeros elsewhere. The computation of the recommendations is presented in Algorithm 2 and it entails $\Theta(K\,\mathrm{nnz}(\mathbf{P}))$ operations, where $\mathrm{nnz}(\mathbf{P})$ is the number of nonzero elements in $\mathbf{P}$.

**RecWalk$^{\text{PR}}$:** The recommendation score of user $u$ for item $i$ is defined to be the element that corresponds to item $i$ in the limiting distribution of a random walk with restarts on $\mathbf{P}$, with damping factor $\eta$ and teleportation distribution $\mathbf{e}_u$:

$$\boldsymbol{\pi}_u^\top \triangleq \lim_{K\to\infty} \mathbf{e}_u^\top(\eta\mathbf{P} + (1-\eta)\mathbf{1}\mathbf{e}_u^\top)^K. \tag{10}$$

This can be computed based on the power method as in Algorithm 3. Producing recommendations for a user involves $\Theta((\log\epsilon/\log\eta)\,\mathrm{nnz}(\mathbf{P}))$ floating point operations for convergence up to a tolerance $\epsilon$ [28].

### 2.5 Theoretical Properties

As we will show in the rest of this section, a key property of the RecWalk model is that for small values of parameter $\alpha$ the RecWalk chain is going to be *nearly uncoupled* into a large number of blocks, thereby allowing the random walk process dynamics towards equilibrium to disentangle into a *slow-mixing* and a *fast-mixing* component. This implies personalized landing probabilities even when the number of steps gets large.

#### 2.5.1 *Nearly Uncoupled Markov Chains.*

A nearly uncoupled Markov chain is a discrete time chain whose transition probability matrix is almost block diagonal [9, 41]. Formally, let $\mathbf{Z} \in \Re^{n \times n}$ be the transition probability matrix of an irreducible and aperiodic Markov chain. Matrix $\mathbf{Z}$ can always be written as $\mathbf{Z} = \mathbf{Z}^\star + \varepsilon\mathbf{C}$, where $\mathbf{Z}^\star$ is a block-diagonal matrix of order $n$, given by $\mathbf{Z}^\star = \mathrm{Diag}(\mathbf{Z}^\star_{11}, \mathbf{Z}^\star_{22}, \ldots, \mathbf{Z}^\star_{LL})$; and matrices $\mathbf{Z}^\star_{II}$ are irreducible stochastic matrices of order $n(I)$. Hence, $n = \sum_{I=1}^{L} n(I)$ , and because both $\mathbf{Z}$ and $\mathbf{Z}^\star$ are stochastic, the row-sums of $\mathbf{C}$ are zero.

**Algorithm 1** RecWalkModel

**Input:** Implicit feedback matrix $\mathbf{R}$, *parameters*: $\alpha, \gamma_1, \gamma_2, C$.
**Output:** RecWalk transition probability matrix $\mathbf{P}$.
**parfor** $i \in \mathcal{I}$ **do**
  Find the $C$ nearest neighbors of item $i$ and form $\mathbf{N}_i$

$$\begin{array}{ll} \underset{\mathbf{x}\in\Re^C}{\text{minimize}} & \frac{1}{2}\|\mathbf{r}_i - \mathbf{N}_i\mathbf{x}\|_2^2 + \gamma_1\|\mathbf{x}\|_1 + \frac{1}{2}\gamma_2\|\mathbf{x}\|_2^2 \\ \text{subject to} & \mathbf{x} \geq 0 \end{array}$$

  Fill the corresponding elements in the $i$–th column of $\mathbf{W}$.
**end parfor**
$\mathbf{M}_{\mathcal{I}} \leftarrow \frac{1}{\|\mathbf{W}\|_\infty}\mathbf{W} + \text{Diag}(\mathbf{1} - \frac{1}{\|\mathbf{W}\|_\infty}\mathbf{W}\mathbf{1})$
$\mathbf{A}_{\mathcal{G}} \leftarrow \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{pmatrix}$
$\mathbf{P} \leftarrow \alpha\,\text{Diag}(\mathbf{A}_{\mathcal{G}}\mathbf{1})^{-1}\mathbf{A}_{\mathcal{G}} + (1-\alpha)\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathcal{I}} \end{pmatrix}$

Matrix $\mathbf{C}$ and the non-negative real number $\varepsilon$ are chosen such that for all rows

$$\varepsilon = \max_{m_I} \sum_{J\neq I} \sum_{l=1}^{n(J)} Z_{m_I l_J} \tag{11}$$

$$\varepsilon \sum_{J\neq I} \sum_{l=1}^{n(J)} C_{m_I l_J} = \sum_{J\neq I} \sum_{l=1}^{n(J)} Z_{m_I l_J} \tag{12}$$

where we use $Z_{m_I l_J}$ to denote the element at the intersection of the $m$-th row and $l$-th column of the $\mathbf{Z}_{IJ}$ submatrix of $\mathbf{Z}$.

Parameter $\varepsilon$ is referred to as the *maximum degree of coupling* between the blocks. When $\varepsilon$ is sufficiently small, the Markov chain with transition probability matrix $\mathbf{Z}$ is called *nearly uncoupled* into $L$ blocks [9].

*2.5.2 Mixing properties of RecWalk.* Since the discrete time Markov chain defined by $\mathbf{P}$ is finite and aperiodic,[1] as the number of steps $K$ increases the landing probabilities of RecWalk will converge to a limiting distribution. It is well known (see e.g.[42]) that the rate of convergence to this distribution depends on the modulus of the subdominant eigenvalue of the transition probability matrix of the walk. In particular, the asymptotic rate of convergence to the limiting distribution is the rate at which $|\lambda_2(\mathbf{P})|^k \to 0$. Intuitively, the smaller $|\lambda_2(\mathbf{P})|$ is, the sooner the landing probability distributions will start yielding recommendation vectors that are "unpersonalized," in the sense that they are the same for all users irrespectively of the items with which they have interacted.

The following theorem sheds more light to the spectral properties of matrix $\mathbf{P}$.

Theorem 2.1. *Let* $\mathbf{P}$ *be the RecWalk transition probability matrix with* $\alpha \in (0,1)$ *defined over a connected user-item network* $\mathcal{G}$*, and also let* $\lambda(\mathbf{P})$ *be the set of the eigenvalues of* $\mathbf{P}$*. Irrespectively of the item model used to define matrix* $\mathbf{M}_{\mathcal{I}}$ *it holds*

(a) $1 - 2\alpha \in \lambda(\mathbf{P})$
(b) *when* $\alpha$ *is small enough the Markov chain with transition probability matrix* $\mathbf{P}$ *will be nearly uncoupled into at least* $U + 1$ *blocks.*

[1] Note that aperiodicity is ensured by the nonzero diagonal of $\mathbf{M}$ (see e.g. [27]).

**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**

**Algorithm 3** RecWalk$^{\text{PR}}$

**Input:** RecWalk model $\mathbf{P}$, user $u \in \mathcal{U}$, damping factor $\eta$.
**Output:** Recommendation vector $\boldsymbol{\pi}_u$.
$\mathbf{x}_{(0)}^\top \leftarrow \mathbf{e}_u^\top$
$k \leftarrow 0$
**repeat**
  $k \leftarrow k + 1$
  $\mathbf{x}_{(k)}^\top \leftarrow \eta\mathbf{x}_{(k-1)}^\top\mathbf{P} + (1-\eta)\mathbf{e}_u^\top$
  Normalize $\mathbf{x}_{(k)}^\top$
**until** $\|\mathbf{x}_{(k)}^\top - \mathbf{x}_{(k-1)}^\top\|_1 < tol$
$\boldsymbol{\pi}_u \leftarrow \mathbf{x}_{(k)}$

Proof. When $\mathcal{G}$ is connected, matrix $\mathbf{H}$ is irreducible [22]. Furthermore, since the graph is bipartite a simple random walk on $\mathcal{G}$ results in a periodic Markov chain with period $d = 2$. Therefore, from the Perron-Frobenius theorem [15] we get that

$$\lambda_1(\mathbf{H}) = 1, \quad \text{and} \quad \lambda_2(\mathbf{H}) = e^{2i\pi/d} = e^{i\pi} = -1.$$

The so-called Perron eigenvalue $\lambda_1(\mathbf{H})$ is associated with the right eigenvector $\mathbf{1}$; whereas eigenvalue $\lambda_2(\mathbf{H})$ with a right eigenvector which we denote $\mathbf{v}$.

The special structure of $\mathbf{H}$ makes it easy to guess the form of the eigenvector $\mathbf{v}$ as well as to verify that it actually denotes an eigenvector of matrix $\mathbf{M}$ too. In particular, we have

$$\mathbf{v} \triangleq [\overbrace{1 \quad 1 \quad 1 \quad \cdots \quad 1}^{|\mathcal{U}|\text{ user nodes}} \quad \overbrace{-1 \quad -1 \quad \cdots \quad -1}^{|\mathcal{I}|\text{ item nodes}}].$$

It is easy to see that $\mathbf{v}$ is indeed an eigenvector of both matrices $\mathbf{H}$ and $\mathbf{M}$. In particular, we have

$$\mathbf{H}\mathbf{v} = \begin{pmatrix} \mathbf{0} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{1}_U \\ -\mathbf{1}_I \end{pmatrix} = \begin{pmatrix} -\mathbf{1}_U \\ \mathbf{1}_I \end{pmatrix} = -\mathbf{v} \tag{13}$$

from which we get that $(-1, \mathbf{v})$ is an eigenpair of matrix $\mathbf{H}$; and

$$\mathbf{M}\mathbf{v} = \begin{pmatrix} \mathbf{1}_U \\ -\mathbf{M}_{\mathcal{I}}\mathbf{1}_I \end{pmatrix} = \mathbf{v} \tag{14}$$

which implies that $(1, \mathbf{v})$ is an eigenpair of matrix $\mathbf{M}$.

Now consider a non-singular matrix, $\mathbf{Q} \triangleq \begin{pmatrix} \mathbf{1} & \mathbf{v} & \mathbf{X} \end{pmatrix}$, which contains in its first two columns the eigenvectors $\mathbf{1}$ and $\mathbf{v}$. Also let

$$\mathbf{Q}^{-1} \triangleq \begin{pmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \mathbf{Y}^\top \end{pmatrix}. \tag{15}$$

By definition it holds $\mathbf{Q}^{-1}\mathbf{Q} = \mathbf{I}$, which can be usefully written as

$$\begin{pmatrix} \mathbf{y}_1^\top \mathbf{1} & \mathbf{y}_1^\top \mathbf{v} & \mathbf{y}_1^\top \mathbf{X} \\ \mathbf{y}_2^\top \mathbf{1} & \mathbf{y}_2^\top \mathbf{v} & \mathbf{y}_2^\top \mathbf{X} \\ \mathbf{Y}^\top \mathbf{1} & \mathbf{Y}^\top \mathbf{v} & \mathbf{Y}^\top \mathbf{X} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}. \tag{16}$$

Now, if we consider the similarity transformation of the RecWalk transition probability matrix, $\mathbf{Q}^{-1}\mathbf{P}\mathbf{Q}$, also taking into consideration (13), (14) and the identities (16), we have

$$\begin{aligned} \mathbf{Q}^{-1}\mathbf{P}\mathbf{Q} &= \mathbf{Q}^{-1}\left(\alpha\mathbf{H} + (1-\alpha)\mathbf{M}\right)\mathbf{Q} = \cdots = \\ &= \begin{pmatrix} 1 & 0 & \alpha\mathbf{y}_1^\top\mathbf{H}\mathbf{X} + (1-\alpha)\mathbf{y}_1^\top\mathbf{M}\mathbf{X} \\ 0 & 1-2\alpha & \alpha\mathbf{y}_2^\top\mathbf{H}\mathbf{X} + (1-\alpha)\mathbf{y}_2^\top\mathbf{M}\mathbf{X} \\ \mathbf{0} & \mathbf{0} & \alpha\mathbf{Y}^\top\mathbf{H}\mathbf{X} + (1-\alpha)\mathbf{Y}^\top\mathbf{M}\mathbf{X} \end{pmatrix} \end{aligned} \tag{17}$$

from which we directly establish that $1 - 2\alpha$ is an eigenvalue of matrix $\mathbf{P}$, and the first part of the theorem is proved.

To prove the second part it suffices to show that there exists a partition of the state space into blocks such that the maximum probability of leaving a block upon a single transition is upper-bounded by $\alpha$ [41]. In particular, consider the partition

$$\mathcal{A} \triangleq \{\{u_1\}, \{u_2\}, \ldots, \{u_U\}, \mathcal{I}\}. \tag{18}$$

By definition, in the RecWalk model the probability of leaving $u$ is equal to $\alpha$, for all $u \in \mathcal{U}$. Concretely,

$$\Pr\{\text{jump from } u \in \mathcal{U} \text{ to any } j \neq u\} = \sum_{j \neq u} P_{uj} = \sum_{j \neq u} \alpha H_{uj} = \alpha.$$

Similarly, the probability of leaving block $\mathcal{I}$ upon a transition is

$$\Pr\{\text{jump from } i \in \mathcal{I} \text{ to any } \ell \notin \mathcal{I}\} = \sum_{\ell \notin \mathcal{I}} P_{i\ell} = \sum_{\ell \notin \mathcal{I}} \alpha H_{i\ell} = \alpha.$$

Therefore, the RecWalk chain can always be decomposed according to $\mathcal{A}$ such that the maximum degree of coupling between the involved blocks is exactly equal to $\alpha$. Hence, choosing $\alpha$ to be sufficiently small ensures that the chain will be nearly uncoupled into (at least) $U + 1$ blocks. □

Theorem 2.1 asserts that the proposed random walk construction ensures the existence of an eigenvalue equal to $1 - 2\alpha$. This means that the modulus of the eigenvalue that determines the rate of convergence to the limiting distribution will be at least $1 - 2\alpha$. Hence, choosing $\alpha$ allows us to ensure that the RecWalk process will converge as slow as we need to increase the number of landing probability distributions that can still serve as personalized recommendation vectors in our model—irrespectively of the particular user-item network or the chosen item model upon which it is built. Moreover note that the spectral “fingerprint” of nearly uncoupled Markov chains is the existence of a set of subdominand eigenvalues that are relatively close (but not equal) to 1 [9]. In our case, for small values of $\alpha$ these eigenvalues are expected to be clustered close to the value $1 - \alpha$ (cf. (17)). The number of these eigenvalues depicts the number of blocks of states into which the chain is nearly uncoupled. Therefore, subject to $\alpha$ being small the RecWalk chain will have at least $U + 1$ eigenvalues “clustered” around the value 1, which means that matrix $\mathbf{P}$ can be expressed as

$$\mathbf{P} = \mathbf{1}\boldsymbol{\pi}^\top + \mathbf{T}_{\text{slow}} + \mathbf{T}_{\text{fast}} \tag{19}$$

where $\boldsymbol{\pi}^\top$ is the stationary distribution of the walk, $\mathbf{T}_{\text{slow}}$ is a *slow transient* component, and $\mathbf{T}_{\text{fast}}$ is a *fast transient* component (see [41] for details). As $K$ gets large the fast transient term will diminish while the elements of the slow transient term will remain large enough to ensure that the recommendation vectors are not completely dominated by $\mathbf{1}\boldsymbol{\pi}^\top$. Of course, as $K$ gets larger and larger the relative influence of the first term will become stronger and stronger, up to the point where each user is assigned the exact same recommendation vector $\boldsymbol{\pi}^\top$; however this outcome will be delayed by the existence of the slow transient term $\mathbf{T}_{\text{slow}}$. Note that in a simple random walk on $\mathbf{W}$ such time-scale dissociation of the stochastic dynamics of the walk is typically absent; and certainly it cannot be guaranteed in advance. On the contrary, the proposed random walk construction in RecWalk provides a clear mechanism to ensure such property, and as we will see in the experimental section of this paper this property alone can lead to significant improvements in top-$n$ recommendation quality compared to what one would get by using the item model directly.

## 3 RELATED WORK

Over the past decade a vast number of algorithms have been proposed to tackle the top-$n$ recommendation task. These include neighborhood-based methods [11, 33]; latent-space methods [10, 18, 21, 31, 43]; graph-based methods [7, 8, 20]; and more recently methods relying on deep neural networks [17, 23, 44]. Recommendation methods relying on random walks have also been deployed in several large-scale industrial settings with remarkable success [12, 14, 16]. RecWalk combines item-models with random walks, and therefore lies at the intersection between neighborhood- and graph-based methods; the inter-item transition component captures the neighborhoods of the items which are then incorporated on a random walk framework to produce recommendations.

The construction of RecWalk is inspired by the properties of nearly uncoupled Markov chains. The analysis of nearly uncoupled systems—also referred to as *nearly decomposable* systems—has been pioneered by Simon [40], who reported on state aggregation in linear models of economic systems. However, the universality of Simon's ideas has permitted the theory to be used with significant success in the analysis of complex systems arising in social sciences and economics [4, 35, 37], evolutionary biology [39], cognitive science [24, 25], administrative sciences and management [45, 46], etc. The introduction of these ideas in the fields of computer science and engineering can be traced back to Courtois [9] who applied Simon's theory in the performance analysis of computer systems. More recently near decomposability has been recognized as a property of the Web [19] and it has inspired the development of algorithms for faster computation of PageRank [6, 47] (building on a large body of related research in the field of numerical linear algebra; see e.g. [26, 41, 42]) as well as the development of new network centrality measures [13, 29, 32]. In the field of recommender systems the notion of decomposability has inspired the development of methods for incorporating meta-information about the items [30] with the blocks chosen to highlight known structural/organizational properties of the underlying itemspace. Here, on the contrary, we exploit decomposability in the “time-domain” with the blocks defined to separate the short-term from the long-term temporal dynamics of the walk in order to effect the desired mixing properties that can lead to improved recommendation performance.

# 4 EXPERIMENTAL SETTING

## 4.1 Datasets

Our experimental evaluation is based on three real-world publicly available datasets; namely (i) the `movielens` dataset (obtained from [1]) which contains the ratings of 6,040 users for 3,706 movies and it has been used extensively for the evaluation of top-$n$ recommendation methods; (ii) the `yahoo` dataset (obtained from [31]) which is a subset of the Yahoo!R2Music dataset [2] containing the ratings of 7,307 users for 3,312 songs; and (iii) the `pinterest` dataset (obtained from [17]) which captures the interactions of 55,187 users regarding 9,916 images where each interaction denotes whether the user has "pinned" the image to her own board.

## 4.2 Evaluation Methodology and Metrics

To evaluate the top-$n$ recommendation performance, we adopted the widely used *leave-one-out* evaluation protocol [17, 21, 31, 33, 36]. In particular, for each user we randomly select one item she liked[2] and we create a test set $\mathcal{T}$. The rest of the dataset is used for training the models. For model selection we repeat the same procedure on the training data and we create a validation set $\mathcal{V}$; and for each method considered we explore the hyperparameter space to find the model that yields the best performance in recommending the items in $\mathcal{V}$, and then we evaluate its out-of-sample performance based on the held-out items in $\mathcal{T}$. For the evaluation we consider for each user her corresponding test item alongside 999 randomly sampled unseen items and we rank the 1000 item lists based on the recommendation scores produced by each method. During training of all competing methods we consider only binary feedback.

The evaluation of the top-$n$ recommendation performance is based on three widely used ranking-based metrics; namely the *hit ratio* (HR@$n$), the *average reciprocal hit rank* (ARHR@$n$), and the truncated *normalized discounted cumulative gain* (NDCG@$n$) over the set of users (due to space constraints we refer the reader to [23, 31] for a detailed definition). For each user, all metrics compare the predicted rank of the held-out item with the ideal recommendation vector which ranks the held-out item first among the items in each user's test-set list. For all competing methods we get the predicted rank by sorting the recommendation scores that correspond to the items included in each user's test-set list. While HR@$n$ gives a perfect score if the held-out item is ranked within the first $n$, ARHR@$n$ and NDCG@$n$ use a monotonically increasing discount to emphasize the importance of the actual position of the held-out item in the top-$n$ recommendation list.

# 5 EXPERIMENTAL RESULTS

## 5.1 Effect of Parameter $\alpha$

The theoretical analysis of our method suggests that parameter $\alpha$ controls the convergence properties of the RecWalk process and it can be chosen to enable a time-scale dissociation of the stochastic dynamics towards equilibrium that ensures a larger number of personalized landing distributions. Here we verify experimentally[3] the predicted properties and we evaluate their effect on the recommendation quality of the $K$-step landing probabilities.

We build the item model $\mathbf{W}$ that yields the best performance on the validation set ($\gamma_1 = 10, \gamma_2 = 10, C = 0.1|\mathcal{I}|$) and we use it to create matrix $\mathbf{M}_{\mathcal{I}}$ as in (7). We then build the RecWalk model, we run it for different values of $\alpha$ ranging from 0.005 to 0.5, and we report: (i) the performance in terms of average NDCG@$n$ (for $n = 10$) across all users for values of steps $K$ up to 30 (Fig.1-A); (ii) the spectra of the corresponding transition probability matrices $\mathbf{P}$ (Fig.1-B); (iii) the peak performance per parameter $\alpha$ along with the step for which it was achieved (Fig.1-C); and (iv) the performance of RecWalk with respect to using the base model $\mathbf{W}$ directly (Fig.1-D).

We find that as the value of $\alpha$ gets smaller the top-$n$ recommendation quality increases and stabilizes for $\alpha < 0.01$. Similarly the number of steps that yield the best performance increase (see Fig.1-C). The spectrum of the corresponding transition probability matrices reflects the theoretically predicted properties of the model. Indeed for very small values of $\alpha$ we observe that the subdominant eigenvalues cluster near the value 1, thus forming the slowly varying component of matrix $\mathbf{P}$, which ensures that the successive landing probabilities of the random walk are influenced by the initial state for longer. Furthermore, we find that the proposed methodology entails a significant increase in performance with respect to what one would get by using the proposed base item-model directly. In particular the recommendation performance of the $K$-step landing probabilities of RecWalk overpasses the performance of the base model (see Fig.1-D) for a wide range of steps up to a maximum increase of 39.09% for $K = 18$ steps.

Our results suggest that the mixing properties of the RecWalk chain are indeed intertwined with the top-$n$ recommendation quality and can lead to a significant boost in performance. This justifies the intuition that motivated the particular design of the walk.

## 5.2 RecWalk as a Framework

In the definition of the inter-item transition probability matrix we proposed a particular strategy for constructing matrix $\mathbf{W}$ that was designed to promote locality on the direct inter-item transitions while being easy to compute. Instead of this particular matrix $\mathbf{W}$ one could use any model that captures inter-item relations. But does our approach offer any benefit with respect to performing simple random walks on the corresponding item-model or to simply using the item model directly?

Here we explore this question by empirically evaluating two commonly used item models. Namely:

(1) a cosine similarity model $\mathbf{W}_{\text{COS}}$ defined such that its $ij$-th element is given by $\mathbf{r}_i^\top \mathbf{r}_j/(\|\mathbf{r}_i\|\|\mathbf{r}_j\|)$;
(2) a `SLIM` model which learns a matrix $\mathbf{W}_{\text{SLIM}}$ by solving an $\ell_1, \ell_2$ regularized optimization problem (see [33] for details).

We consider the respective base models alongside six approaches based on random walks; namely (i) `SRW`, which recommends using the $K$-step distribution of a simple random walk on $\mathbf{W}$ with transition probability matrix $\mathbf{S}$ initialized with $\boldsymbol{\phi}_u^\top$ as in (1); (ii) `PR`, which produces recommendations based on $\mathbf{S}$ as in (2); (iii-iv)

[2] When rating information is available in the original data, the per-user target item is randomly sampled among the highest rated items of each particular user in order to ensure that it indeed denotes an item that the user liked. Such approach is in accordance with the methodology described in the seminal papers [10, 21]; in our setting, however, all users are equally represented in the testset.

[3] Due to space constrains we report results on the `movielens` dataset; similar patterns arise using the other datasets as well.

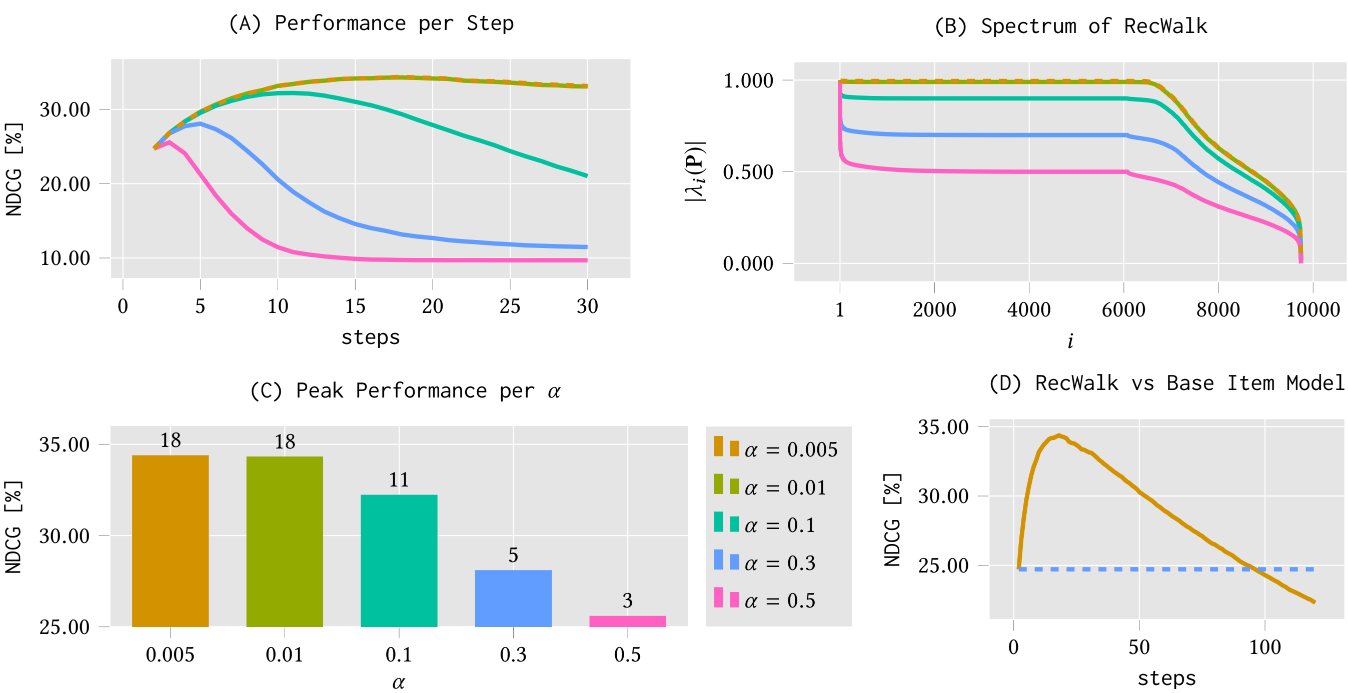


**Figure 1: Fig (A) reports the performance of RecWalk in terms of NDCG@10 as a function of the steps for different values of the parameter $\alpha$. Fig (B) plots the spectrum of the RecWalk transition probability matrix for different values of $\alpha$. Fig (C) reports the peak performance for the different values of $\alpha$, as well as the number of steps for which it is achieved (on top of each bar). Fig (D) reports RecWalk performance (with $\alpha$ fixed at 0.005) for different number of steps compared to the performance one would get using the item model directly (blue dashed line).**

RecWalk$^{\text{K-step}}$ and RecWalk$^{\text{PR}}$ with transition matrix $\mathbf{P}$ constructed using the respective item models for the definition of the inter-item transition probability component; (v) RecWalk$[\mathbf{M}_{\mathcal{I}}]^{\text{K-step}}$ which produces recommendations as in (1) but using the RecWalk inter-item transition probability matrix introduced in (7) instead of S; and (vi) RecWalk$[\mathbf{M}_{\mathcal{I}}]^{\text{PR}}$, which produces recommendations as in (2) using RecWalk's $\mathbf{M}_{\mathcal{I}}$ instead of $\mathbf{S}$.

**Table 1: Top-n Recommendation Quality under Different Random Walk Constructions**

| Method | COS | SLIM |
|---|---|---|
| Base model | 17.61 | 27.28 |
| SRW | 17.82 | 25.37 |
| PR | 18.11 | 25.37 |
| RecWalk$^{\text{K-step}}$ | 20.52 | 31.87 |
| RecWalk$^{\text{PR}}$ | 20.33 | 31.80 |
| RecWalk$[\mathbf{M}_{\mathcal{I}}]^{\text{K-step}}$ | 17.85 | 31.41 |
| RecWalk$[\mathbf{M}_{\mathcal{I}}]^{\text{PR}}$ | 20.27 | 31.78 |

Hyperparameters: SRW: K$\in \{1, \ldots, 50\}$; PR: $p \in \{0.1, \ldots, 0.9\}$; SLIM: $\lambda, \beta \in \{0.1, 0.5, 1, 3, 5, 10, 20\}$; RecWalk: $\alpha = 0.005$ and $S \in \{1, \ldots, 30\}$ for RecWalk$^{\text{K-step}}$ and $\eta \in \{0.1, 0.2, \ldots, 0.9\}$ for RecWalk$^{\text{PR}}$.

We run all models on the movielens dataset and in Table 1 we report their performance on the NDCG@$n$ metric. We see that RecWalk$^{\text{K-step}}$ and RecWalk$^{\text{PR}}$ were able to boost the performance of both item models (up to +16.52% for COS and +16.82% for SLIM) with the performance difference between the two variants being insignificant. Applying simple random walks (SRW) or random walks with restarts (PR) directly to the row-normalized version of the item graph does not perform well (+2.84% in case of COS and -7% in case of SLIM). In particular in the case of SRW we observed a rapid decay in performance (after the first step for SLIM and after the first few steps for COS); similarly in case of PR the best performance was obtained for very small values of $p$—enforcing essentially the $K$-step landing probabilities after the first few steps to contribute negligibly to the production of the recommendation scores (cf. (2)). Using RecWalk's inter-item transition probability matrix alone on the other hand performed very well especially when we use the SLIM model as a base.

To gain more insight of the observed differences in performance of the walks on the item graphs, we also plot the spectra of the transition probability matrices $\mathbf{M}_{\mathcal{I}}$, alongside the spectra of the respective matrices $\mathbf{S}$ (Fig. 2). We see that in case of $\mathbf{S}$ the magnitude of the eigenvalues cause the walks to mix very quickly. In case of matrix $\mathbf{M}_{\mathcal{I}}$ on the other hand, the eigenvalues decay more gracefully and on the SLIM graph in particular, there appears to be a large number of eigenvalues near 1, which delay the convergence of the landing probabilities distributions towards equilibrium. This effect is not as pronounced in case of COS which is reflected in the small increase in performance in case of RecWalk$[\mathbf{M}_{\mathcal{I}}]^{\text{K-step}}$.

**Table 2: Top-n Recommendation Quality of the Competing Approaches in Terms of HR, ARHR and NDCG**

| | movielens | | | yahoo | | | pinterest | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | HR [%] | ARHR [%] | NDCG [%] | HR [%] | ARHR [%] | NDCG [%] | HR [%] | ARHR [%] | NDCG [%] |
| EigenRec | 45.21 | 20.44 | 26.35 | 48.12 | 23.30 | 29.23 | 33.81 | 13.51 | 18.41 |
| PureSVD | 44.14 | 19.33 | 25.36 | 38.68 | 18.30 | 22.62 | 30.97 | 11.85 | 16.30 |
| RP3b | 34.87 | 15.02 | 19.66 | 41.51 | 17.82 | 22.94 | 27.01 | 8.07 | 12.45 |
| SLIM | 46.34 | 21.39 | 27.28 | 52.44 | 26.15 | 32.35 | 34.17 | 13.63 | 18.57 |
| Mult-DAE | 44.06 | 18.97 | 24.83 | 45.37 | 21.46 | 27.07 | 35.03 | 13.79 | 18.77 |
| Mult-VAE | 44.35 | 19.50 | 25.31 | 45.09 | 21.22 | 26.80 | 35.13 | 13.73 | 18.71 |
| RecWalk$^{\text{K-step}}$ | 50.28 | 27.20 | 33.13 | **55.02** | **28.94** | **35.10** | **35.38** | **14.07** | 18.95 |
| RecWalk$^{\text{PR}}$ | **52.47** | **27.29** | **34.09** | 54.92 | 28.74 | 34.91 | 35.29 | **14.07** | **19.00** |

Hyperparameters: RecWalk: (fixed) $\alpha = 0.005$, $C \in \{0.1|\mathcal{I}|, 0.25|\mathcal{I}|\}$ $\gamma_1, \gamma_2 \in \{1, \dots, 15\}$ and $K \in \{1, \dots, 30\}$ for RecWalk$^{\text{K-step}}$ and $\eta = \{0.1, \dots, 0.9\}$ for RecWalk$^{\text{PR}}$; EigenRec: $f \in \{5, 10, \dots, 300\}$, $d \in \{-2, -1.95, \dots, 2\}$; RP3b: $b \in \{0, 0.05, \dots, 1\}$; SLIM: $\lambda, \beta \in \{0.1, 0.5, 1, 3, 5, 10, 20\}$; Mult-VAE$^{\text{pr}}$-Mult-DAE: we used the hyperparameter tuning approach provided by the authors in their publicly available implementation; for each model we considered both architectures proposed in [23]; namely $[I - 200 - I]$ and $[I - 600 - 200 - 600 - I]$.

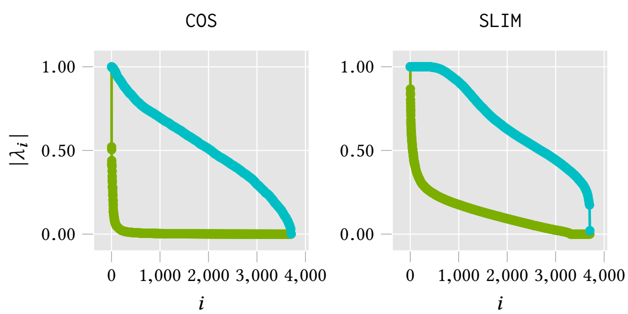


**Figure 2: The figure plots the spectra of the transition probability matrices S (green line) and $\mathbf{M}_{\mathcal{I}}$ (blue line) defined using the corresponding item models.**

Again our experiments reveal a clear connection between the mixing properties of the walks and their potential in achieving good recommendation quality. Note also that the stochasticity adjustment strategy proposed in (7) seems to promote slow mixing in itself. However, using $\mathbf{M}_{\mathcal{I}}$ alone cannot in general guarantee this property irrespectively of the underlying item model whereas using the complete RecWalk model can give absolute control over convergence (as Theorem 2.1 predicted).

## 5.3 Performance against Competing Approaches

We evaluate the top-$n$ recommendation accuracy of RecWalk$^{\text{K-step}}$ and RecWalk$^{\text{PR}}$ against competing approaches.

*5.3.1 Competing Baselines.* We compare against six state-of-the-art baselines; namely (i) the well-known item-based method SLIM [33] which builds a sparse item model by solving an $\ell_1, \ell_2$-regularized optimization problem; (ii) the random-walk approach RP3b [7], which recommends based on a short-length walk on the user-item graph after rescaling the landing probabilities to compensate for the bias towards popular items; (iii) the well-known PureSVD method [10], which produces recommendations based on the truncated SVD of $\mathbf{R}$; (iv) the EigenRec method [31], which builds a factored item model based on a scaled cosine similarity matrix; as well as the recently proposed deep learning methods (v-vi) Mult-VAE and Mult-DAE [23] which extend variational and denoising autoencoders to collaborative filtering using a multinomial likelihood and were shown to achieve state-of-the-art recommendation quality, outperforming several other deep-network-based approaches.

*5.3.2 Results.* Table 2 reports the top-$n$ recommendation performance of the competing approaches. The performance was measured in terms of HR@$n$, ARHR@$n$ and NDCG@$n$, focusing on the $n = 10$. Model selection was performed for each dataset and metric following the procedure detailed in Section 4.2 and considering for each method the hyperparameters reported on Table 2. We see that both variants of RecWalk outperform all other methods considered on every metric and for all datasets. The results indicate the potential of the proposed methodology in achieving high quality top-$n$ recommendations.

# 6 CONCLUSIONS AND FUTURE DIRECTIONS

Combining random walks with item models has the potential of exploiting better the information encoded in the inter-item relations which can lead to increased top-$n$ recommendation accuracy. To gain this benefit, however, one needs to define judiciously the transition probabilities of the walks in order to counterbalance their tendency to rapidly concentrate towards the central nodes of the graph. To this end we introduced RecWalk; a novel random walk framework for top-$n$ recommendations that can provably provide control over convergence allowing the walk to harvest more effectively the rich network of interactions encoded within the item model on top of which it is built. Our experiments reveal that the mixing properties of the walks are indeed intertwined with top-$n$ recommendation performance. A very interesting direction we are currently pursuing involves the exploration of methods for statistical learning in the space of landing probabilities [5] produced by RecWalk. Here we proposed two simple recommendation strategies to exploit these landing probabilities that were able to provide high top-$n$ recommendation accuracy, outperforming several state-of-the-art competing approaches. Our findings showcase the value of combining item-models with graph-based techniques.

## ACKNOWLEDGMENTS

This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251), Army Research Office (W911NF1810344), Intel Corp, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

## REFERENCES

[1] [n. d.]. GroupLens. http://grouplens.org/.
[2] [n. d.]. Yahoo Webscope Program. https://webscope.sandbox.yahoo.com/.
[3] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), 734–749.
[4] Albert Ando, Franklin M Fisher, and Herbert Alexander Simon. 1963. *Essays on the structure of social science models.* Mit Press Cambridge.
[5] Dimitris Berberidis, Athanasios N Nikolakopoulos, and Georgios B Giannakis. 2018. Adaptive Diffusions for Scalable Learning over Graphs. *arXiv preprint arXiv:1804.02081* (2018).
[6] A. Cevahir, C. Aykanat, A. Turk, and B.B. Cambazoglu. 2011. Site-Based Partitioning and Repartitioning Techniques for Parallel PageRank Computation. *IEEE Trans. Parallel Distrib. Syst.*, 22, 5 (2011), 786–802.
[7] Fabian Christoffel, Bibek Paudel, Chris Newell, and Abraham Bernstein. 2015. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *Proceedings of the 9th ACM Conference on Recommender Systems.* ACM, 163–170.
[8] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random Walks in Recommender Systems: Exact Computation and Simulations. In *Proceedings of the 23rd International Conference on World Wide Web.* ACM, New York, NY, USA, 811–816. https://doi.org/10.1145/2567948.2579244
[9] Pierre-Jacque Courtois. 1977. *Decomposability: Queueing and Computer System Applications.* Academic Press. http://books.google.gr/books?id=jD5RAAAAMAAJ
[10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems (RecSys '10).* ACM, 39–46. https://doi.org/10.1145/1864708.1864721
[11] Christian Desrosiers and George Karypis. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, 107–144. https://doi.org/10.1007/978-0-387-85820-3_4
[12] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time. In *Proceedings of the 2018 World Wide Web Conference (WWW '18).* 1775–1784. https://doi.org/10.1145/3178876.3186183
[13] Guang Feng, Tie-Yan Liu, Ying Wang, Ying Bao, Zhiming Ma, Xu-Dong Zhang, and Wei-Ying Ma. 2006. AggregateRank: bringing order to web sites. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06).* ACM, New York, NY, USA, 75–82. https://doi.org/10.1145/1148170.1148187
[14] Ashish Goel, Pankaj Gupta, John Sirois, Dong Wang, Aneesh Sharma, and Siva Gurumurthy. 2015. The Who-To-Follow System at Twitter: Strategy, Algorithms, and Revenue Impact. *Interfaces* 45, 1 (Feb. 2015), 98–107. https://doi.org/10.1287/inte.2014.0784
[15] Geoffrey Grimmett and David Stirzaker. 2001. *Probability and random processes.* Oxford university press.
[16] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. WTF: The Who to Follow Service at Twitter. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13).* ACM, New York, NY, USA, 505–514. https://doi.org/10.1145/2488388.2488433
[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 173–182. https://doi.org/10.1145/3038912.3052569
[18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.* Ieee, 263–272.
[19] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. 2003. Exploiting the Block Structure of the Web for Computing PageRank. In *Stanford University Technical Report.*
[20] Zhao Kang, Chong Peng, Ming Yang, and Qiang Cheng. 2016. Top-n recommendation on graphs. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* ACM, 2101–2106.
[21] Yehuda Koren. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans on Knowledge Discovery from Data (TKDD)* 4, 1 (2010), 1.
[22] Amy N. Langville and Carl D. Meyer. 2011. *Google's PageRank and beyond: The science of search engine rankings.* Princeton University Press.
[23] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference (WWW '18).* 689–698. https://doi.org/10.1145/3178876.3186150
[24] David Meunier, Renaud Lambiotte, and Edward T Bullmore. 2010. Modular and hierarchically modular organization of brain networks. *Front Neurosci* 4 (2010).
[25] D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore. 2009. Hierarchical modularity in human brain functional networks. *Front Neuroinform.* 3 (2009).
[26] Carl D. Meyer. 1989. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Rev.* 31, 2 (June 1989), 240–272. https://doi.org/10.1137/1031050
[27] Carl D. Meyer. 2000. *Matrix analysis and applied linear algebra.* Vol. 2. Siam.
[28] Carl D. Meyer and Charles D. Wessell. 2012. Stochastic Data Clustering. *SIAM J. Matrix Anal. Appl.* 33, 4 (2012), 1214–1236.
[29] Athanasios N. Nikolakopoulos and John D. Garofalakis. 2013. NCDawareRank: a novel ranking method that exploits the decomposable structure of the web. In *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM '13).* ACM, New York, NY, USA, 143–152. https://doi.org/10.1145/2433396.2433415
[30] Athanasios N. Nikolakopoulos and John D. Garofalakis. 2014. NCDREC: A Decomposability Inspired Framework for Top-N Recommendation. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, Vol. 1. 183–190. https://doi.org/10.1109/WI-IAT.2014.32
[31] Athanasios N. Nikolakopoulos, Vassilis Kalantzis, Efstratios Gallopoulos, and John D. Garofalakis. 2018. EigenRec: generalizing PureSVD for effective and efficient top-N recommendations. *Knowledge and Information Systems* (03 May 2018). https://doi.org/10.1007/s10115-018-1197-7
[32] A. N. Nikolakopoulos, A. Korba, and J. D. Garofalakis. 2016. Random surfing on multipartite graphs. In *2016 IEEE International Conference on Big Data (Big Data).* 736–745. https://doi.org/10.1109/BigData.2016.7840666
[33] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *2011 11th IEEE International Conference on Data Mining.* IEEE, 497–506.
[34] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/
[35] Andreas Reinstaller. 2007. The division of labor in the firm: Agency, near-decomposability and the Babbage principle. *Journal of Institutional Economics* 3 (12 2007), 293–322. Issue 03. https://doi.org/10.1017/S1744137407000732
[36] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 452–461.
[37] Saras D Sarasvathy. 2003. Entrepreneurship as a science of the artificial. *Journal of Economic Psychology* 24, 2 (2003), 203 – 220. https://doi.org/10.1016/S0167-4870(02)00203-9 The Economic Psychology of Herbert A. Simon.
[38] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web.* ACM, 285–295.
[39] Max Shpak, Peter Stadler, Gunter P Wagner, and Lee Altenberg. 2004. Simon-Ando decomposability and fitness landscapes. *Theory Biosci.* 123, 2 (2004), 139–180.
[40] Herbert A. Simon and Albert Ando. 1961. Aggregation of variables in dynamic systems. *Econometrica: Journal of The Econometric Society* (1961), 111–138.
[41] GW Stewart. 1991. On the sensitivity of nearly uncoupled Markov chains. *Numerical solution of Markov chains. Probability: pure and applied* 8 (1991), 105–119.
[42] William J. Stewart. 1994. *Introduction to the numerical solution of Markov chains.* Princeton University Press. http://books.google.gr/books?id=h6NuKuZPKugC
[43] Jason Weston, Ron J Weiss, and Hector Yee. 2013. Nonlinear latent factorization by embedding multiple user interests. In *Proceedings of the 7th ACM conference on Recommender systems.* ACM, 65–68.
[44] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining.* ACM, 153–162.
[45] Ramsin Yakob and Fredrik Tell. 2007. Managing near decomposability in complex platform development projects. *International Journal of Technology Intelligence and Planning* 3, 4 (2007), 387–407.
[46] Sai Yayavaram and Gautam Ahuja. 2008. Decomposability in knowledge structures and its impact on the usefulness of inventions and knowledge-base malleability. *Administrative Science Quarterly* 53, 2 (2008), 333–362.
[47] Yangbo Zhu, Shaozhi Ye, and Xing Li. 2005. Distributed PageRank computation based on iterative aggregation-disaggregation methods. In *Proceedings of the 14th ACM international conference on Information and knowledge management.* ACM, 578–585.