

RESEARCH

Open Access



Towards a folksonomy graph-based context-aware recommender system of annotated books

Sara Qassimi^{1*} , El Hassan Abdelwahed², Meriem Hafidi² and Aimad Qazdar²

*Correspondence:

sara.qassimi@uca.ac.ma

¹ Laboratory L2IS,
Department of Computer
Science, Faculty of Sciences
and Techniques Guiliz (FSTG),
University of Cady Ayyad
(UCA), Marrakesh, Morocco
Full list of author information
is available at the end of the
article

Abstract

The emergence of collaborative interactions has empowered users by enabling their interactions through tagging practices that create a folksonomy, also called, classification of the shared resources, any identifiable thing or item on the system. In education, tagging is considered a powerful meta-cognitive strategy that successfully engages learners in the learning process. Besides, the collaborative tagging gathers learners' opinions, thus, provides more comprehensible recommendations. Still, the abundant shared contents are mostly unorganized which makes it hard for users to select and discover the appropriate items of their interests. Thus, the use of recommender systems overcomes the distressing search problem by assisting users in their searching and exploring experience, and suggesting relevant items matching their preferences. In this regard, this article presents a folksonomy graphs based context-aware recommender system (CARS) of annotated books. The generated graphs express the semantic relatedness between these resources, i.e. books, by effectively modeling the folksonomy relationship between user-resource-tag and integrating contextual information within a multi-layer graph referring to a Knowledge Graph (KG). To put our proposal into shape, we model a real-world application of Goodbooks-10k dataset to recommend books. The proposed approach incorporates spectral clustering to deal with the graph partitioning problem. The experimental evaluation shows relevant performance results of graph-based book recommendations.

Keywords: Collaborative Tagging, Folksonomy, Knowledge Graph, Emergent Semantic, Spectral Clustering, Recommender System, Context-awareness

Introduction

The popularity of the social information systems (SocIS) has enabled social interaction leading to tremendous growth in the volume of shared resources. A resource is any identifiable things stored in the system (e.g. books, images, museums, selling items, electronic health records, and other documents). Social information systems SocIS are information systems based on social technologies and open collaboration [1]. The rise of SocIS allows users to easily communicate, learn and coordinate collaboratively. The SocIS platforms use the features of folksonomy to commonly contribute to the management of the shared resources. More contemporary and popular

techniques of SocIS include social network analysis, collaborative filtering and social tagging [2]. The arrival of the social web has enabled users to add metadata using their keywords, called tags, to the shared content. Users collaboratively use tags to annotate the various types of resources such as books, places, blogs, images, videos, audios, and so forth. The collaborative tagging enables a collaborative classification, folksonomy, of the shared resources. Folksonomy, unlike a taxonomy, includes terms without a hierarchy [3]. It is a collection of tags from uncontrolled vocabulary. Folksonomy, named also collaborative tagging and social indexing, derives a classification system from the collaboratively generated tags.

Collaborative tagging techniques are useful for enhancing the recommendation of learning resources [4]. More comprehensible learning resources' recommendations are provided by gathering learners' opinions. Indeed, the collaborative tagging technique is used for automatic analysis of the user's preferences to improve recommender system quality [5]. Folksonomy adds semantics to the shared learning resources without the support of manual indexers or automated generated keywords. Tagging is demonstrated to be an adequate meta-cognitive strategy that successfully engages learners in the learning process [6].

The recommender system (RS) overcomes the distressing search problem for the users, it is not only used in the e-commerce domain to suggest products on Amazon, videos on YouTube, movies on Netflix, friends on Facebook, and so on. RS is also helpful in the education domain. It plays the role of supporting teaching, learning [7] and expanding the knowledge through enhancing the information retrieval and suggesting additional information that has not been yet founded through the user's searching.

Active research in the field of recommender system has been focusing on measuring the similarities between different items. Many Recommender Systems (RSs) utilize traditional filtering techniques to infer similarities between items. Specific designs of graph-based filtering systems unable flexibility in computing the similarity (using co-occurrence, co-rating, co-tagging, co-describing, and so on) among different entities. Besides, for better-personalized user recommendations, RSs leverage the contextual information in their process of recommendation, called context-aware recommendation system (CARS). This paper will mainly respond to the following Research Questions:

- RQ1: How to model the relational information between resources, in our case study books, by effectively analyzing the folksonomy relationships?
- RQ2: How to establish a folksonomy graph-based Context-Aware Recommender System CARS?
- RQ3: How to perform recommendations based on a Knowledge Graph (KG) database?

To put our proposal into shape, we model a real-world application of Goodbooks-10k dataset [8]. The great volume of its relational data provides tagged and rated books, which we aim to model and represent within a multi-layers graph. Where each layer concerns a graph of homogeneous type of nodes related with their weighted edges,

i.e., a graph of tags, graph of books, and graph of users. The combination of the three graphs forms a multi-relational or multi-layer graph, where nodes represent entities and edges represent relationships. The resulting multi-relational graph is referred to as a Knowledge Graph (KG).

Our proposal aims to (i) analyze and model the folksonomy relationship to generate graphs of books and tags (ii) integrate the emergent semantic of books within the CARS by establishing a semantic graph-based recommender system methodology (iii) exploring the generated multi-layer graph that incorporates the three graphs of users, books and tags modeled within a knowledge graph database to perform diversity in the recommendation. Besides, This paper presents the use of spectral clustering as a pre-processing step before constructing the semantic graph-based recommendation algorithm to suggest books.

The rest of the paper is organized as follows: Section "[Background and related work](#)" presents the background and related works of recent graph-based recommender systems approaches. Section "[Proposed approach](#)" depicts the proposed approach of folksonomy graphs based context-aware recommender system using spectral clustering. Section "[Evaluation and results](#)" describes the evaluation and experimental results using the Goodbooks-10k [8] dataset for book recommendations. Finally, the conclusion and future directions are delineated in Sect. "[Conclusion and perspectives](#)".

Background and related work

Recommender system

The recommender system is a branch of information retrieval and artificial intelligence, a subclass of information filtering systems that aims to assist the user's search behavior by suggesting items that best meet their interests and preferences. The main commonly distinguished filtering algorithms of recommender systems are collaborative filtering (CF), content-based (CB) and hybrid filtering recommendations approach [9], which rely on the exploitation of items-items similarities and users-items interactions. These approaches come across some challenges that affect the precision of recommendations [10], such as sparsity, cold start, and scalability issues. Besides, the challenges of recommending in real-time within the scalable systems have called the need to enhance the recommendation method. Given the focus of this article, this literature review will mainly be related to graph-related recommendation algorithms.

Recommender system to support learners

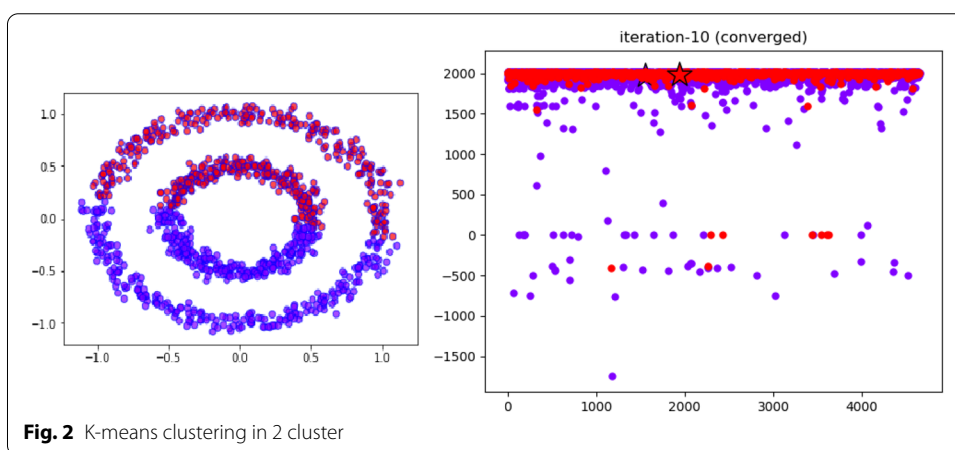
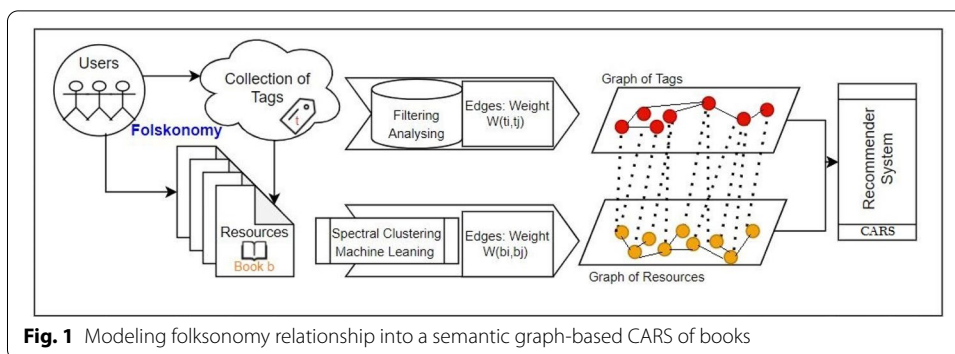
The purpose of recommender systems is to help users conduct searches by suggesting resources (items) that best match their interests and preferences [11, 12]. The use of recommender systems in Computer Environments for Human Learning (CEHL) aims to support learners in the learning process to achieve their learning goals [13]. On such platforms, the pedagogical resources are regularly organized and diffused [14] which involves the learners' concern about which resource to consult. Recommendation systems provide support to learners by recommending content that might be of their interest and which they may not have discovered before.

Graph-based recommender system

Recent works have shown the effectiveness of using graph modeling to enhance the performance of recommender systems. The authors [15] propose three methods for making KG-based recommendations using a general-purpose probabilistic logic system. Linked Open Data has been used as an external knowledge graph adopted within a hybrid graph data model [16]. In [17], they propose a heterogeneous information network-based music recommendation system that uses a graph-based algorithm to generate recommendations. To improve social trust and influence, the authors [18] propose a graph-based model for the social recommendation. To reduce the dimensionality of the recommendation problem, the authors [19] propose a graph-based recommendation system that learns and exploits the geometry of the user space to create clusters in the user domain. In [20], they overcome the major challenges of sparsity and scalability by leveraging a directed graph of jobs connected by multi-edges to recommend jobs. Although these approaches use graphs-based recommendations to solve specific issues, our work differs significantly as we aim to enhance the recommendation's performance and diversity by capturing the semantic and social interaction through modeling folksonomy relationship into a semantic graph described by weighted graphs then exploring the multi-layer graph so-called KG. Graphs are mathematical structures enabling to encode multiple types of interactions, as we will show in what follows, recommendations can be computed thanks to graphs that create meaningful clusters of entities reducing the dimensionality of the recommendation problem.

Graph database

The use of graph-oriented database technology leverages the building of flexible recommendation engines. A graph database is characterized by its distinct "data model" compared to traditional relational databases [21]. A data model is a set of conceptual tools to represent and manage data. The data model consists of three components [22]: data structure, types, query operators, integrity rules. A graph database stores data and represented it in a graph. Each graph database has its specialized graph query language. For example, Neo4J uses Cypher language, and RDF (Resource Description Framework) databases use SPARQL (SPARQL Protocol and RDF Query Language). The integrity rules in a graph database are based on its graph constraints, rather than an imposed relational schema. The popularity of using graph databases has emerged with the increasing complexity of real-world data and growing needs for graph queries. For example, Neo4J is suited for online transaction processing (OLTP), a widely used open-source graph database, and characterized as an "embedded, disk-based, and fully transactional graph database engine" [23]. Graph databases are gaining a lot of interest with their data modeling tools that provide a closer fit to real-world data. Early adopters of graph technology reimagined their businesses around the value of data relationships. These companies have now become industry leaders: LinkedIn, Google, and Facebook [24]. The graph database model is a set of nodes related with edges described with properties. The graph database represents and stores data using a graph structure. It is based on the NoSQL approach to lunch semantic queries to retrieve data.



Graph databases use a data model that stores the data relationships as edges related to the nodes representing the data. Therefore, the graph data model enables storage, processing, and querying connections between data efficiently. While relational databases compute data relationships through expensive, high-cost, and complex join queries.

Proposed approach

The social web has initiated the use of collaborative tagging over the years. It has called attention to analyzes the inter-connectivity of the user-resource-tag to improve the recommender system.

The proposed approach models the folksonomy characteristics by inferring user-resource-tag in graphs linked to one another to emerge the semantic graph-based recommendations (see Fig. 1). In this paper, the annotated resource are books. Two coherent clusters are obtained using spectral clustering to reduce the scalability issue of the dataset (see Figs. 2 and 3).

The use of spectral clustering will pre-process the construction of the graph of books to reduce the scalability issue. For each graph, we examine the relationship between its entities and identify actionable knowledge. For a community of users $U = \{u_s\}$, they annotate a set of books $B = \{b_k\}$ with a set of tags $T = \{t_i\}$. Where, $1 \leq s \leq l$; $1 \leq k \leq m$; $1 \leq i \leq n$. And l , m and n are respectively the total number of users, books and tags.

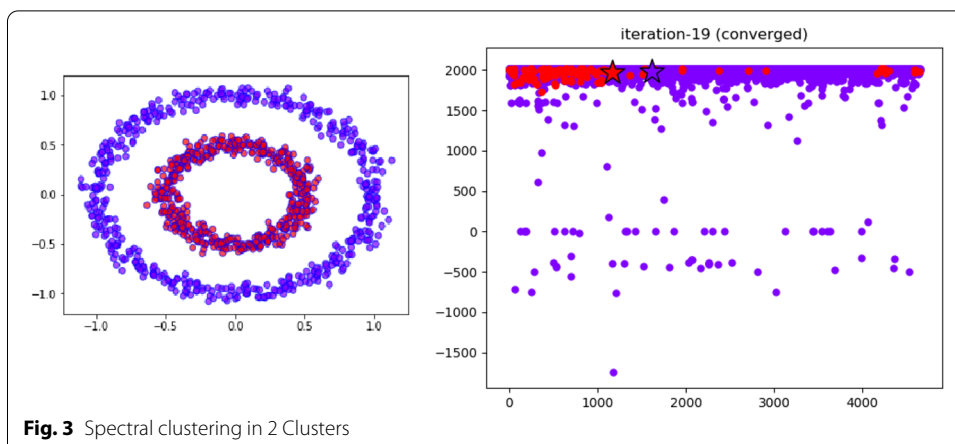


Fig. 3 Spectral clustering in 2 Clusters

Spectral clustering

The use of spectral clustering stages a pre-processing phase before constructing the graph of resources. Spectral clustering deals with the graph partitioning problem. It transforms the current space to bring connected data points close to each other to form clusters. In this context, the data points to be clustered are 10K books.

Clustering

Clustering is one of the most widely used techniques for exploratory data analysis. Its goal is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. Spectral clustering has become increasingly popular due to its promising performance in graph-based clustering. It can be solved efficiently by standard linear algebra software and very often outperforms traditional algorithms such as the k-means algorithm. Spectral clustering does not make assumptions about the shape of clusters. Unlike K-means, which assumes a spherical shape for the resulted clusters. Spectral clustering gives importance to connectivity (within data points) rather than compactness (around cluster centers). The goal of spectral clustering is to cluster data that is connected.

Background

The usefulness of an item depends on the user’s current context and circumstances. Varying the recommendations depends on contextual factors like time, location, mood, user’s actual situation, position, status, and condition. The challenge is to go beyond the representational approach with its predefined and fixed set of observable attributes.

The set of books $B = \{b_i\}$ represents the data points, where the book b_i denotes data entry. Each $b_i \in B^f$, where f is the number of features describing b_i , like spatial, temporal and static contextual features, tags.

Similarity matrix

Given an enumerated set of data points B , the Similarity or Adjacency Matrix is defined as a symmetric matrix A , where $A_{ij} \geq 0$ represents a measure of the similarity between b_i and b_j . $A_{ij} \simeq 1$ when b_i and b_j have the same features. The data points, books, are in the same cluster when there are close, but in different clusters when there are far away.

But data points in the same cluster may also be far away or even farther away than points in different clusters. The goal is to transform the f -dimensional space so that when 2 points b_i and b_j are close, they are always in the same cluster, and when they are far apart, they are in different clusters. A common way to define similarity is by using the Gaussian Kernel A_{ij} .

$$A_{ij} = e^{-\frac{\|b_i - b_j\|^2}{2\sigma^2}}. \quad (1)$$

Unnormalized graph laplacian

The unnormalized Graph Laplacian is a matrix defined as the difference of 2 matrices denoted: $L = D - A$, where D is the diagonal Degree matrix.

$$D_{ii} = \sum_j A_{ij} \quad \text{And} \quad L_{ij} = D_{ii} - A_{ij}. \quad (2)$$

Process of spectral clustering

- 1 Construct the similarity matrix using Gaussian Kernel,
- 2 Compute the first k eigenvectors of its Laplacian matrix to define a feature vector for each object. It embeds the data points, books $\{b_i\}$, in a low-dimensional space in which clusters are more obvious.
- 3 Apply a classical clustering algorithm, like K-means, to partition the books into k classes.

The spectral clustering data entry are books described by their contextual information and most popular tags $f=13$. Each book b_i has its defined vector of contextual information $CI-b_i = \{\text{author}_i, \text{language}_i, \text{year-of-publication}_i, 10\text{-most-popular-tags}\{t_{i_1}, \dots, t_{i_{10}}\}\}$. The spectral clustering defines the 2 clusters of books better than the k-means (see Figs. 2 and 3).

Folksonomy graph-based recommendation:

Graph of books

Each cluster resulting from the spectral clustering has its graph of books. The graph of books emerges from the similarities among books using their descriptive tags. Two books $b_i, b_j \in B$ are semantically similar when their weighted edge $W(b_i, b_j)$ is high. The weight $W(b_i, b_j)$ computes how strongly two books are semantically related (3). The weighted edges are normalized to re-scales values into a range of $[0,1]$.

$$\begin{aligned}
 W(b_i, b_j) &= \sqrt{W_t(b_i, b_j)^2 + W_c(b_i, b_j)^2} \\
 W_t(b_i, b_j) &= \frac{\text{Number of tags describing both books } b_i \text{ and } b_j}{\text{Total Number of tags}} \\
 W_c(b_i, b_j) &= \frac{\sum w_c(b_i, b_j)}{f} \\
 w_c(b_i, b_j) &= \begin{cases} 1 & \text{if Matching - Member}[CI - b_i, CI - b_j] = \text{True} \\ 0 & \text{else.} \end{cases}
 \end{aligned} \tag{3}$$

f = Total Number of books' contextual features.

Matching-Member[CI- b_i , CI- b_j] returns True if an element of CI- b_i matches CI- b_j , and False otherwise.

The contextual information of a book b_i : CI- b_i = {author $_i$, language $_i$, year-of-publication $_i$, 10-most-popular-tags{ t_{i1}, \dots, t_{i10} }}

Therefore, the recommender system will explore the books-books and tags-tags similarity to do suggestions.

Graph of tags

Tags are clustered in an emergent graph of tags. We consider the relevant tags with higher degree of frequency DF (4).

More comprehensive folksonomies emerge from non-expert or novice users' tags than from experts' tags only [25]. Therefore, the proposed approach considers the extraction of tags that are frequently used and understood by many users of the community.

We consider a book, $b \in B$ described by a set of tags from T. The extraction of relevant tags describing this book b is computed by considering the degree of frequency of each tag t_i , denoted by DF(b, t_i).

$$DF(b, t_i) = \sqrt{FT(b, t_i)^2 + FU(b, t_i)^2} \tag{4}$$

where

FT (b, t_i) is the Frequency of the tag t_i annotating the book b ;

FU (b, t_i) is the Frequency of users who use the tag t_i to annotate the book b .

$$FT(b, t_i) = \frac{\text{Number of times the tag } t_i \text{ is used to describe the book } b}{\text{Number of tags used to describe the book } b} \tag{5}$$

$$FU(b, t_i) = \frac{\text{Number of users who use the tag } t_i \text{ to annotate the book } b}{\text{Number of users who annotate the book } b} \tag{6}$$

The relevant tags are those with higher degree of frequency.

The graph of tags $G_T=(V_T, E_T)$ is drawn up by the common weight $W(t_i, t_j)$ relating two tags t_i and t_j . The tags represents the nodes V_T , linked together by the weighted edges $W(t_i, t_j)$. The weight $W(t_i, t_j)$ identifies the semantic relationships among tags t_i and t_j ; it scales how strongly the two tags are semantically related regarding their commonly usage of users' annotation and books' description.

The emergent folks' tags semantic graph enables graph-based reasoning about the relationships between tags attributed to describe different books.

Folksonomy graph-based recommendation algorithm

The recommender system will suggest books based on the emergent graphs of books and tags from the folksonomy.

Algorithm 1 folksonomy graph-based CARS of books

```

 $u_i \in U ; t \in T ; t_{u_i} \in T_{r_{u_i}} ; b_j, b_{u_i} \in B ; Recom_t, Recom_b, RecomReturn_b \subset B$ 
 $Recom_b$  : a set of recommended books based on the graph of books
 $Recom_t$  : a set of recommended books based on the graph of tags
 $k$  : a finite number of recommendations,  $k \in \mathbb{N}$ .
 $T_{r_{u_i}}$  : a finite number of tags describing the book  $b_{u_i}$ ,  $T_{b_{u_i}} \subset T$ .
procedure RECOMMENDATIONGRAPHBOOKS( $u_i, b_{u_i}$ )
  for  $b_j \in B$  do
    if  $W(b_{u_i}, b_j) > 0$  then
       $Recom_b \leftarrow$  list of k-ranked  $b_j$ 
    end if
  end for
  return  $Recom_b$ 
end procedure
procedure RECOMMENDATIONGRAPHTAGS( $u_i, b_{u_i}, t_{u_i}$ )
   $RecomReturn_b \leftarrow$  RecommendationGraphBooks( $u_i, b_{u_i}$ )
  for  $b_{u_i} \in RecomReturn_b$  and  $t_{u_i} \in T_{b_{u_i}}$  do
    if  $W(t_{u_i}, t) > 0$  then
       $Recom_t \leftarrow$  list of k-ranked books annotated with  $t$ 
    end if
  end for
  return  $Recom_t$ 
end procedure

```

Evaluation and results

To evaluate the proposed approach, we use the Goodbooks-10k [8] dataset for book recommendations. The dataset provides ten thousand books tagged and rated by users. The books are described by their metadata like ISBN, authors, year, and title. Besides, the books are rated (from one to five) and tagged by users using 34252 tags. The evaluation of the book’s recommendations is performed using the automatic or offline evaluation that considers the previously rated books (rating ≥ 3) as the ground truth, true positive T_p , interpreted relevant to the user. That is, evaluating how closely the recommendations match the actual preferences of the user. For each user, we select its previously rated book to generate automatic book recommendations based on a graph-based recommender system algorithm. The accuracy metrics, namely precision P, recall R, and F1-measure, are calculated from the number of books that are either rated or not and either recommended or not. Four possible outcomes are shown in the confusion matrix (see Table 1).

The precision P is the probability that the recommended books are relevant to the user. It is defined as the number of recommended relevant books, that were previously rated above 3, divided by the total number of recommended books. While Recall R is defined

Table 1 Confusion matrix accumulating the possible results of books recommendation

	Relevant	Irrelevant
Recommended	True-positive (T_p)	False-positive (Fp)
Not Recommended	False-negative (Fn)	True-negative (Tn)

Table 2 Evaluation: Book Recommendation

Users ID	Rated book	P 1	R 1	F 1	P 2	R 2	F 2
7805	144	0.905	0.132	0.23	0.4	0.027	0.052
41282	178	1	0.129	0.228	0.2	0.011	0.0212
24143	196	1	0.117	0.21	0.3	0.015	0.029
22262	94	1	0.159	0.275	0.1	0.01	0.019
43622	168	0.952	0.119	0.211	0.4	0.024	0.045
22437	4	1	0.5	0.666	0	0	0
12815	167	1	0.125	0.223	0.2	0.012	0.022
35642	111	0.727	0.144	0.24	0.2	0.018	0.033
1	99	0.857	0.06	0.113	0.4	0.04	0.073
6752	7	1	0.286	0.444	0	0	0

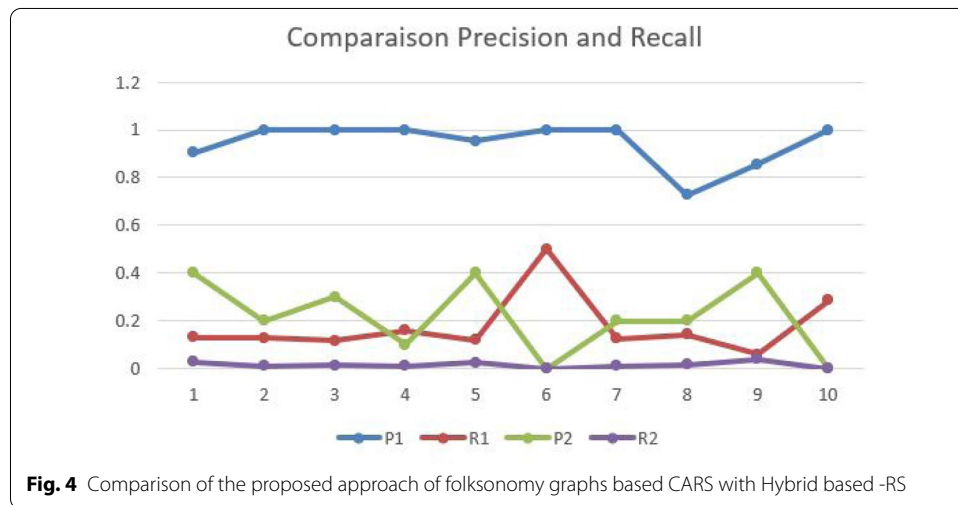


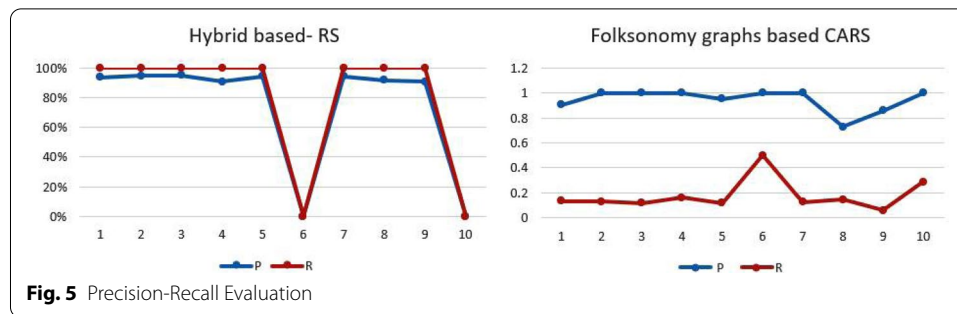
Fig. 4 Comparison of the proposed approach of folksonomy graphs based CARS with Hybrid based -RS

as the number of recommended relevant books divided by the total number of existing relevant books. Recall R is the probability that the user’s previously above 3-rated (relevant) books are recommended. The F1-measure F combines precision and recall.

$$P = \frac{Tp}{Tp + Fp} ; R = \frac{Tp}{Tp + Fn} ; F = \frac{2 \times P \times R}{P + R} \tag{7}$$

We compute Precision P1 for our proposed approach and Precision P2 for hybrid-based recommender system (hybrid based-RS), Recall R1 for our proposed approach, and Recall R2 for hybrid based-RS. The experimental evaluation (Table 2) contains the results of the three metrics evaluating the recommendations presented to randomly selected 10 active users. The accuracy metrics evaluate whether the folksonomy graph-based recommendations algorithm can properly predict the relevant books that were previously well-rated by the users. The high precision results indicate that almost all the recommendations are indeed relevant to the users.

The proposed approach of folksonomy graphs-based recommender system is compared to hybrid recommendations using both filtering approaches CB and CF (Figs. 4 and 5). The algorithm of hybrid based-RS recommends books with similar content to



the 10 active users. Its recommendation process is based also on the similarity of books with users' profiles using collaborative filtering. The precision-recall curve shows that the precision and the recall of the proposed approach "folksonomy graph based CARS" are higher than the hybrid based-RS. Indeed, the comparative figure (see Fig. 4) shows that precision P1 and recall R1 of our proposed approach have higher results than the precision P2 and recall R2 of the hybrid based-RS.

Recommendation based on knowledge graph database

The use of graph database enables: Graph storage structure, Flexible data model, Graph querying, Scalability, and Transaction processing [26]:

Graph storage

Providing native processing capabilities where every node is directly linked to its neighbour node. Therefore, each node acts as an index of other nearby nodes. The graph-based indices enable retrieve graphs quickly from a large database. the graph database is appropriate for local graph queries that need one index lookup to traverse nodes relationships. While in Relational Database Management System (RDMS), it probably needs joining more tables through foreign keys and using additional index lookup.

Query performance

Graph databases come from a flexible data model that enables graph storage in the database strongly tied to graph querying. It generally performs great regardless of the number and depth of connections. In practice, the most frequent queries use the index-free adjacency that includes looking for a node and its neighbours with the shortest path comprising their edges, then retrieve attributes, and so on. A complex query is the subgraph and supergraphs queries that output another graph by ordinarily performing a selection or a projection of the original graph. The current most known graph query language is Cypher native graph query language working with Neo4j database. The declarative query language Cypher is inspired by SQL syntax.

Scalability

Scaling graphs requires the graph partitioning called sharding which may fail under the category of NP-hard problems. The reason is that graph data is connected which avoids distributing its relationships as much as possible called the minimum point-cut problem. The scaling concerns the scaling of large datasets, read and write performance.

In practice, the scalability is not causing a problem in the graph databases area. For instance, Neo4j manages the size of the graph arbitrary on an upper limit on the order of 10^2 which is enough to support most of the real-world graphs. For example, Neo4j deploys on a cluster more than half of Facebook's social graph. Besides, Neo4j has focused on reading performance using two levels of caching to improve scalability in highly concurrent workloads. The scaling for writes can be accomplished by scaling vertically which requires distributing data across multiple machines especially for very heavy write loads. For example, OLTP graph database system optimizes concurrent access and updates for thousands of users. The scaling for write performance still represents a real challenge for graph databases.

Transaction processing

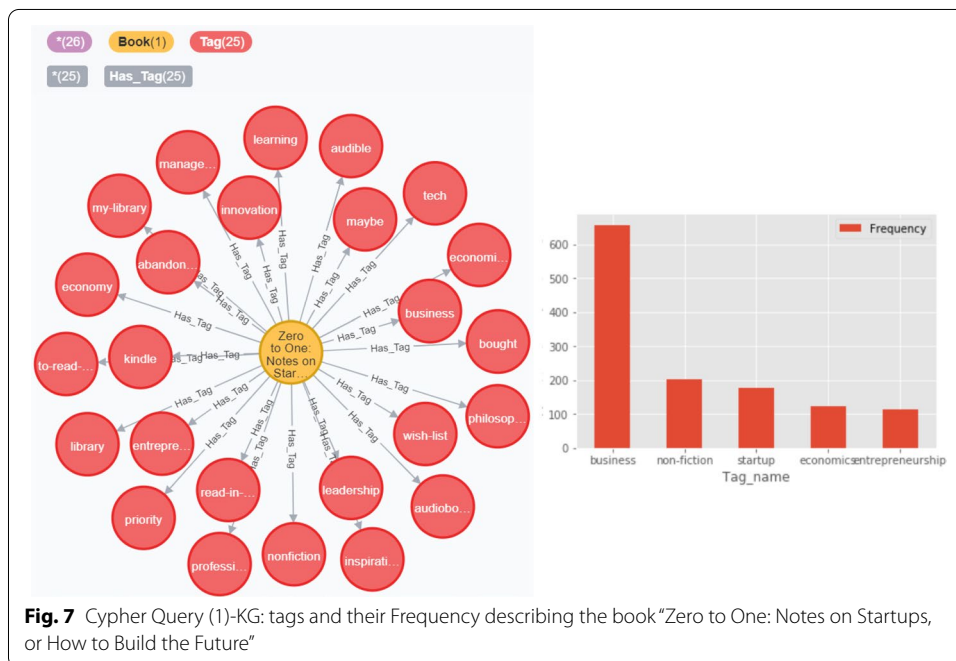
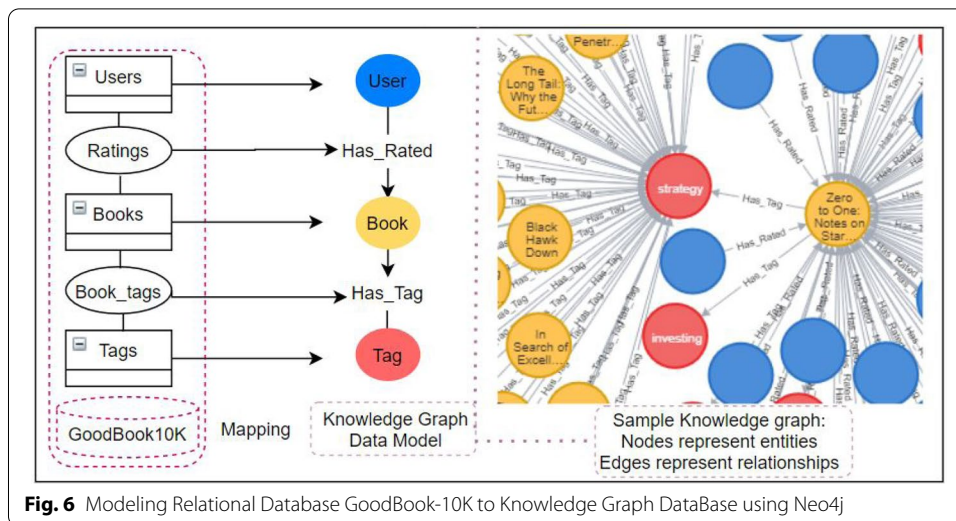
The optimization of Graph databases is often focusing on transaction processing. For example, Neo4j retains ACID transactions. The nature of the graph data structure helps to spread the transactions overhead across the graph. Consequently, the transnational conflicts fade out with the graph expansion and growth. Distributed graph processing requires appropriate partitioning and replication by minimizing the need to ship data between different network nodes. For example, Neo4j uses master-slave replication which means once a machine is a master and the other machines are its slaves. The architecture Neo4j enables that all writes pass through the master first towards any machine, and if the master fails, the cluster will elect a new master automatically. Besides, it contains a bulk loader that operates at a throughput of million records per second. The Graph Model in the Neo4j database has the following components:

- Nodes (equivalent to vertices in graph theory): are the main data elements that are interconnected through relationships. A node can have one or more labels (that describe its role) and properties (i.e. attributes).
- Labels: are used to group nodes, and each node can be assigned multiple labels. Labels are indexed to speed up finding nodes in a graph.
- Properties: are attributes of both nodes and relationships. Neo4j allows for storing data as key-value pairs, which means properties can have any value (string, number, or boolean).

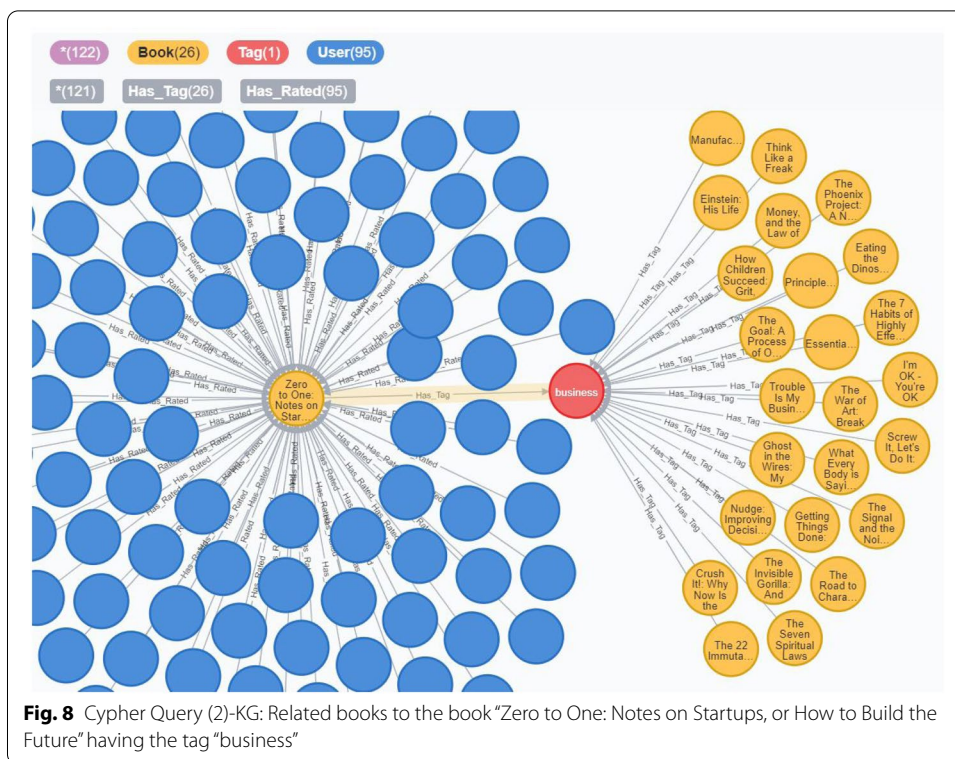
A KG is a heterogeneous network since it contains multiple types of nodes and relations. To illustrate the recommender system based on the KG database, making it simple, we do not consider the weighted edges of every graph of the multi-layer graph (horizontal weighted edges) and focus only on the direct edges (Vertical relations) relating the different type of nodes. The generated Knowledge Graph KG (Fig. 6) provides different ways to organize, manage and retrieve information.

The model chosen for the graph consists in:

- Three kinds of nodes: User, Book and Tag
- Two kind of direct edges: Has_Tag expressing the relationship Book-Tag. Has_Rated relating users to books. The rating $r(b,u)$ given by user u to book b is recorded as a property of the edge.



By focusing on tags, we find movies that have similar tags in the Graph Database. For instance, we executed the following Cypher Queries (Cypher Query (1) and Cypher Query (2)) presented in Fig. 7 and Fig. 8. For a given user that has rated the book titled "Zero to One: Notes on Startups, or How to Build the Future", the Cypher Query (3) recommends books that are described with similar tags (see Fig. 9). For reasoning, this strategy recommends books that were not rated before by the target user.



```
Cypher Query (1)- Neo4j Knowledge Graph :
MATCH p=(b:Book)-[r:Has_Tag]->()
where b.name='Zero to One: Notes on Startups , or How to Build the Future '
RETURN p LIMIT 25
```

```
Cypher Query (2)- Neo4j Knowledge Graph :
MATCH p=(b:Book)-[r:Has_Tag]->(t:Tag)-[:Has_Tag]-(otherBook:Book)
where b.name='Zero to One: Notes on Startups ,
or How to Build the Future '
and t.name='business '
RETURN p LIMIT 25
```

In many domains, it is insufficient to predict how much a user will like an item. The recommender system’s performances need to have a good overview of Novelty, diversity, unexpectedness [27] and other components of utility. Presume the utility and usefulness of the recommendations for the user. The challenge is finding the exact mix of novelty, diversity and familiarity and inclination to recommend items. For a given user that has rated the book titled “Zero to One: Notes on Startups, or How to Build the Future”, the below Cypher Query (3) recommends books that are described with similar tags. For reasoning, this strategy recommends books that were not rated before by the target user.

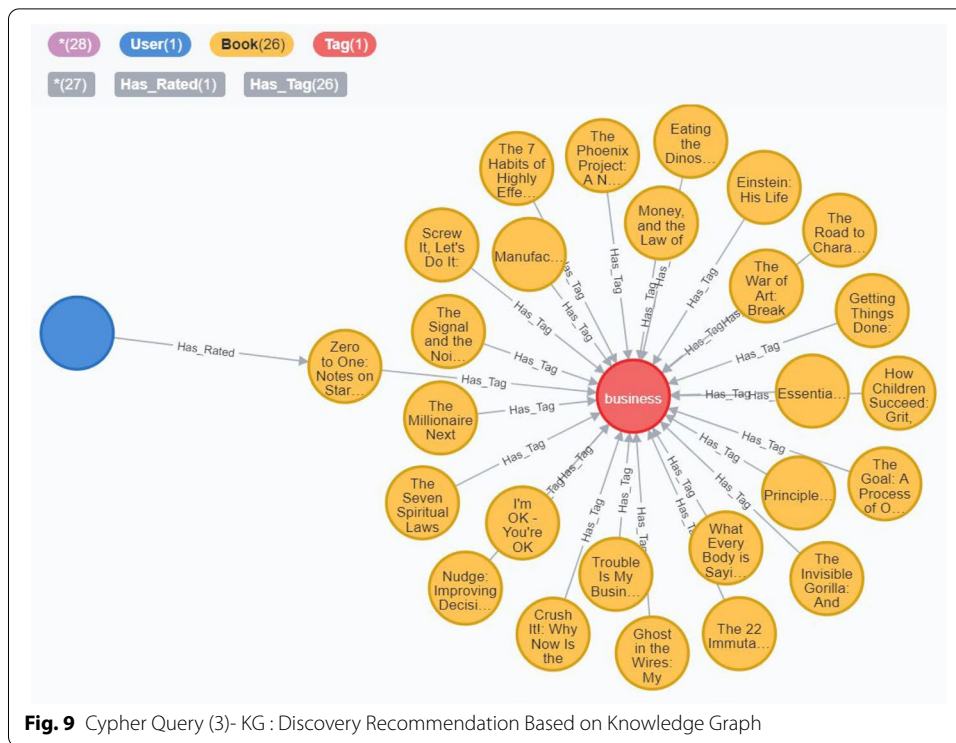
```

Cypher Query (3)- Neo4j Knowledge Graph :
MATCH p=(user:User{id:"6221"})-[:Has_Rated]->(b:Book)-[r:Has_Tag]->
(t:Tag)<-[:Has_Tag]-(otherBook:Book)
WHERE NOT (user)-[:Has_Rated]->(otherBook)
AND b.name='Zero to One: Notes on Startups ,
or How to Build the Future '
AND t.name='business '
RETURN p LIMIT 25

```

Conclusion and perspectives

The emergence of collaborative interactions has scaled up the growth and diversity of shared resources. With the advent of social information systems, users share, annotate and rate resources (e.g, books, movies, articles, etc.) any identifiable thing or item on the system. This abundance of shared resources is mostly unorganized, thus making it tough for users to do their searching and exploring experiences. The arrival of the social web has enabled users to annotate the shared resources with their tags, which creates a collaborative classification or folksonomy. The proposed approach aims to explore the contextual information coming from the application domain as well as analyzing the folksonomy relationship to generate graphs of resources and tags which create the ground of knowledge of the recommender system. the purpose of this article is to enhance the recommendation's performance and diversity by capturing the semantic and social interaction through modeling folksonomy relationship into a semantic graph described by weighted graphs then exploring the multi-layer graph so-called knowledge graph (KG). The recommender system's purpose is to help users in finding and discovering books of their interest. This paper describes a folksonomy graph-based context-aware recommender system of annotated books. Our proposal presents three contributions: (i) analyze and model the folksonomy relationship to generate graphs of books and tags (ii) integrate the emergent semantic of books within the CARS by establishing a semantic graph-based Recommender system methodology (iii) exploring the generated multi-layer graph that incorporates the three graphs of users, books and tags modeled within a knowledge graph database to perform diversity in recommendations. Besides, This paper presents the use of spectral clustering as a pre-processing step before constructing the semantic graph-based recommendation algorithm to suggest books. The Goodbooks-10k dataset has been conducted in this study as a real-world use case. The evaluation has shown the accuracy and effectiveness of the proposed approach. The experimental evaluation has provided higher accuracy measures attesting the relevancy of the proposed folksonomy graphs based CARS algorithm compared to hybrid based RS. Future works will extend the experiment to the online evaluation. Future perspectives will focus on integrating additional contextual information to improve the description of resources, also covering the graph theory and network analysis for generating and adjusting the graph of resources to enhance their recommendations.



Abbreviations

SocIS: Social Information Systems; CARS: Context-Aware Recommender System; KG: Knowledge Graph; RS: Recommender System; RSs: Recommender Systems; RDMS: Relational Database Management System; CB: Content-Based; CF: Collaborative Filtering; CEHL: Computer Environments for Human Learning; OLTP: Online Transactional Processing; ACID: Atomicity, consistency, isolation, durability.

Acknowledgements

Not applicable.

Authors' contributions

All authors contributed to this paper. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Laboratory LZIS, Department of Computer Science, Faculty of Sciences and Techniques Guiliz (FSTG), University of Cady Ayyad (UCA), Marrakesh, Morocco. ²Laboratory LISI, Department of Computer Science, Faculty of Sciences Semailia (FSSM), University of Cady Ayyad (UCA), Marrakesh, Morocco.

Received: 30 December 2020 Accepted: 3 May 2021

Published online: 13 May 2021

References

- Tilly R, Posegga O, Fischbach K, Schoder D. Towards a conceptualization of data and information quality in social information systems. *Bus Inform Syst Eng*. 2017;59(1):3–21. <https://doi.org/10.1007/s12599-016-0459-8>.
- Hafidi M, Abdelwahed EH, Qassimi S. Graph-based tag recommendations using clusters of patients in clinical decision support system. *Concurrency and Computation: Practice and Experience* n/a(n/a), 5624. <https://doi.org/10.1002/cpe.5624>. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.5624>
- Qassimi S, Abdelwahed EH. The role of collaborative tagging and ontologies in emerging semantic of web resources. *Computing*. 2019. <https://doi.org/10.1007/s00607-019-00704-9>
- Klašnja-Miličević A, Ivanović M, Vesin B, Budimac Z. Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques. *Appl Intell*. 2018;48(6):1519–35. <https://doi.org/10.1007/s10489-017-1051-8>.
- Klašnja-Miličević A, Vesin B, Ivanović M, Budimac Z, Jain LC. Folksonomy and tag-based recommender systems in e-learning environments. Cham: Springer. pp. 77–112; 2017. https://doi.org/10.1007/978-3-319-41163-7_7
- Cao Y, Kovachev D, Klamma R, Jarke M, Lau RWH. Tagging diversity in personal learning environments. *J Comp Edu*. 2015;2(1):93–121. <https://doi.org/10.1007/s40692-015-0027-0>.
- Rivera AC, Tapia-Leon M, Lujan-Mora S. Recommendation systems in education: A systematic mapping study. In: Rocha Á, Guarda T (eds) *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*. Cham: Springer; 2018. pp. 937–947.
- Zajac Z. Goodbooks-10k: a new dataset for book recommendations. *FastML* 2017.
- Abbas A, Zhang L, Khan SU. A survey on context-aware recommender systems based on computational intelligence techniques. *Computing*. 2015;97(7):667–90. <https://doi.org/10.1007/s00607-015-0448-7>.
- Khalid O, Khan MUS, Khan SU, Zomaya AY. Omnisuggest: a ubiquitous cloud-based context-aware recommendation system for mobile social networks. *IEEE Trans Services Comput*. 2014;7(3):401–14. <https://doi.org/10.1109/TSC.2013.53>.
- Nassar N, Jafar A, Rahhal Y. Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. *J Big Data*. 2020;7(1):34. <https://doi.org/10.1186/s40537-020-00309-6>.
- Grifi M, Oussous A, Ait Lahcen A, Mouline S. Evaluation of recent advances in recommender systems on Arabic content. *J Big Data*. 2021;8(1):35. <https://doi.org/10.1186/s40537-021-00420-2>.
- Drachsler H, Verbert K, Santos OC, Manouselis N. Panorama of recommender systems to support learning; 2015. pp. 421–451. <https://doi.org/10.1007/978-1-4899-7637-12>
- Manouselis N, Drachsler H, Verbert K, duval e. *Recommender Systems for Learning*; 2012.
- Catherine R, Cohen W. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In: *Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16*, pp. 325–332. Association for Computing Machinery, New York, NY, USA 2016. <https://doi.org/10.1145/2959100.2959131>
- Musto C, Lops P, de Gemmis M, Semeraro G. Semantics-aware recommender systems exploiting linked open data and graph-based features. *Knowl-Based Syst*. 2017;136:1–14. <https://doi.org/10.1016/j.knsys.2017.08.015>
- Wang R, Ma X, Jiang C, Ye Y, Zhang Y. Heterogeneous information network-based music recommendation system in mobile networks. *Comput Commun*. 2020;150:429–37. <https://doi.org/10.1016/j.comcom.2019.12.002>.
- Bathla G, Aggarwal H, Rani R. A graph-based model to improve social trust and influence for social recommendation. *J Supercomput*. 2020;76(6):4057–75.
- Yang K, Toni L. Graph-based recommendation system. In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018; pp.798–802. <https://doi.org/10.1109/GlobalSIP.2018.8646359>
- Shalaby W, AlAila B, Korayem M, Pournajaf L, AlJadda K, Quinn S, Zadrozny W. Help me find a job: A graph-based approach for job recommendation at scale. In: *2017 IEEE International Conference on Big Data (Big Data)*, 2017; pp. 1544–1553. <https://doi.org/10.1109/BigData.2017.8258088>
- Angles R, Gutierrez C. Survey of graph database models. *ACM Comput Surv*. 2008;40(1):1–1139. <https://doi.org/10.1145/1322432.1322433>.
- Codd EF. Data models in database management. *SIGMOD Rec*. 1980;11(2):112–4. <https://doi.org/10.1145/960126.806891>.
- Park Y, Shankar M, Park B-H, Ghosh J. Graph databases for large-scale healthcare systems: A framework for efficient data management and data services; 2014. pp. 12–19. <https://doi.org/10.1109/ICDEW.2014.6818295>
- Unal Y, Oguztuzun H. Migration of data from relational database to graph database. In: *Proceedings of the 8th International Conference on Information Systems and Technologies. ICIST '18*, pp. 6–165. ACM, New York, NY, USA 2018. <https://doi.org/10.1145/3200842.3200852>
- Kang J, Lerman K. Leveraging user diversity to harvest knowledge on the social web. In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*; 2011. pp. 215–222. <https://doi.org/10.1109/PASSAT/SocialCom.2011.106>
- Pokorný J. Graph databases: their power and limitations. In: Saeed K, Homenda W, editors. *Comput Inform Syst Ind Manag*. Cham: Springer; 2015. p. 58–69.
- Castells P, Wang J, Lara R, Zhang D. Introduction to the special issue on diversity and discovery in recommender systems. *ACM Trans Intell Syst Technol*. 2014;5(4):52–1523. <https://doi.org/10.1145/2668113>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.