Graph Model based Recommendation Architecture for E-commerce Applications

SONAL TUTEJA[1]
RAJEEV KUMAR[2]

School of Computer & Systems Sciences
Jawaharlal Nehru University
New Delhi-110067, India
[1]sonalt9@gmail.com
[2]rajeevkumar.cse@gmail.com

Abstract. It is very challenging to provide relevant data to users almost instantaneously due to a large amount of data present in an application. The role of recommendations is to provide relevant data to users considering relationships among data and users. Graph models are enriched in relationships; therefore, we propose an architecture for recommendations based on a graph model in e-commerce. The proposed architecture consists of two phases: offline phase for graph creation and recommendation phase for results generation. In the offline phase, different data sources are unified into a recommendation graph which is utilised by different recommendation algorithms to generate results. We also design algorithms for content-based and collaborative recommendations based on the generated graph. We implement a prototype of the proposed architecture in e-commerce and analyse and compare its performance with the relational model. We also verify the improved performance of the proposed graph model asymptotically. The graph model outperformed the relational model for content-based and collaborative recommendations. Thus, our architecture can be used in various applications for recommendations.

Keywords: Graph Model, Relational Model, Recommendation, E-Commerce.

(Received Oct. 27, 2021 / Accepted Nov. 20, 2021)

1 Introduction

The abundance of information in an application makes it difficult to find the relevant data for the customers. Considering the scenario of an e-commerce application, millions of products listed in the application make it difficult to find the appropriate products for the customers. Recommendation plays a vital role in providing relevant data to users considering the relationships among users and/or data (Figure 1). Recommendation techniques can be broadly classified into two types: (i) Content-based recommendation and (ii) Collaborative recommendation [10]. In content-based recommendation techniques, the contents similar to those accessed by users in the past are recommended. In collaborative recommendation, the contents accessed by similar users in the past are recommended. Therefore, the relationships among contents and/or users are considered to provide recommendations.

The relational model has been widely considered appropriate for traditional applications having concrete schema and lesser changing requirements [3]. Due to the inability of the relational model to satisfy the needs of several applications, various other NoSQL models for data representation have been proposed, including key-value, document base, and graph model [2, 16]. Among all data models like relational, NoSQL, etc., the graph model is the most appropriate for relationship-centric applications. The graph provides a natural way of modelling, querying, and updating data of relationship-centric applications [13, 17]. The model

represents data as nodes and edges, where nodes represent entities and edges represent relationships among them [2]. Each node and relationship can have properties associated with them in the form of key-value pairs. The model helps in faster access of the relationships as nodes are directly connected, called index-free adjacency.

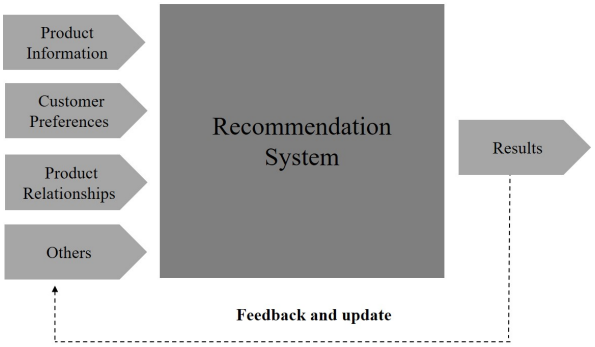

Figure 1: Overview of Recommendation System

The graph model has been used widely for modelling e-commerce data into a graph. Huang et al. [8] utilised a graph model to create relationships among products based on their similarity. Ding et al. [7] incorporated a graph model to represent customer preferences focussed on improving the search results. Cao et al. [5] included a knowledge graph consisting of customers' preferences for the recommendation in e-commerce which helped improve the recommendation results. Various e-commerce companies have also adopted the graph model for real-time recommendation and analysis. Different algorithms have been proposed for the recommendation based on the graph model [9, 21].

In this work, we propose an architecture for the recommendation based on a graph model. The motivation behind providing a graph-based architecture for the recommendation is manifold. First, the graph model can represent relationships among nodes and access them in $O(1)$ complexity which has a significant advantage in providing faster recommendations to the users [13]. Secondly, the graph model offers different types of mappings, which can be used to create different kinds of relationships among nodes [6]. The main contributions of the research work are:

- To propose a graph-based architecture for recommendation in e-commerce applications with multiple data sources.
- To verify the effectiveness of proposed architecture using empirical and asymptotic techniques for e-commerce application data.

The proposed architecture for recommendation consists of two phases: offline phase and recommendation phase. The offline phase maps the data into graph models and creates different types of mapping, which help in the faster recommendation of data. The recommendation phase is used to provide real-time recommendations based on the graph model created by the offline phase. We propose algorithms for the content-based and collaborative recommendations based on graph traversal. To verify the proposed architecture, the performance of the graph model with the underlying relational model is compared. We also asymptotically verify and compare the performance of the graph model with the relational model.

The paper is organised as follows. The literature review is summarised in Section 2. The proposed work is discussed in Section 3. In Section 4, experimental details is mentioned, followed by the analysis of results in Section 5. Finally, Section 5 concludes the paper.

2 Literature Review

2.1 Graph Data Modelling

Graph models have been utilised in various relationship-centric applications for data modelling. Costa et al. discussed various mapping techniques such as communication mapping, co-existence mapping, co-relation mapping, etc., which have been incorporated for mapping data into graph models in different applications [6]. Petermann et al. proposed a graph-based framework in which unified metadata and instance graphs were created from various data sources [15]. The queries related to pattern matching and aggregation were performed on the instance graph; however, performance analysis was not performed.

Yoon et al. proposed a technique for the integration of heterogeneous biological data into the graph model and compared its performance with the relational model [22]. The authors concluded the out-performance of the graph model over the relational model for various queries. Tuteja and Kumar proposed an architecture to transform application data into a graph that can be utilised by different subsystems of the application [18]. The architecture was also verified for various software engineering properties like maintainability, robustness, trustworthiness, etc. Abulaish et al. utilised the graph model for representing tweets in which relationships based on co-relation mapping were created among tweets, and clustering of tweets was applied for event detection [1].

2.2 Graph based Recommendation and Analysis

Researchers have utilised the graph model for recommendation and compared its performance with other data models. Vicknair et al. analysed and compared the performance, scalability, and space requirements of the graph model with the relational model [20]. The graph model was found to outperform the relational model for structural and data queries. Kumar designed different graph models for the representation and analysis of election tweets. The author focussed on designing an appropriate graph model based on the queries to be addressed [11]. Noel et al. utilised the graph model for representing the relationships among network entities in the cybersecurity domain [14]. The graph model was utilised to predict critical vulnerabilities as well as attack paths.

Ding et al. incorporated the graph model to represent customers' preferences to provide recommendations to them, which improved the search results than other comparative algorithms [7]. The generated graph model was also utilised for the clustering of users based on their preferences. Huang et al. applied the graph model to create a network of products by creating relationships among similar products [8]. The network was analysed to find the patterns in product sales. Cao et al. incorporated a knowledge graph consisting of customers' preferences for the recommendation in e-commerce, which helped improve the recommendation results [5]. Li et al. included a knowledge graph to represent user interests, product information, and their relationships in e-commerce applications [12]. The knowledge graph was found suitable for data representation, searching, as well as recommendation. Tuteja and Kumar proposed architecture to map e-commerce data into a graph and analyse its performance for various e-commerce subsystems such as searching, data analysis, and recommendation [19]. The graph model outperformed the relational model for different search queries.

The comparison of graph-based operations with the relational model has been performed by Ma et al. [13]. The author justified how the traversal operation in the graph model performs better than the join operation in the relational model. Based on this, we analyse and evaluate the proposed recommendation algorithms using the graph model with the relational model that is not much explored.

3 Proposed Framework

The proposed framework for the graph-based recommendation in e-commerce consists of two phases: offline phase for graph generation from e-commerce data sources and recommendation phase for result generation using recommendation algorithms. We also design the algorithms for content-based and collaborative recommendations using the generated graph model (section 3.2).

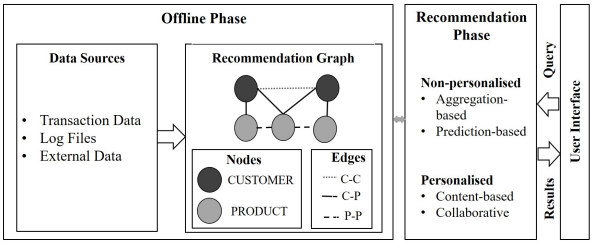

Figure 2: A System Architecture for Recommendation

3.1 Offline Phase

The offline phase in the proposed architecture is focused on generating the graph model from different data sources, which are utilised to generate appropriate recommendation results in an e-commerce application. For example, product similarity data can be utilised to provide content-based recommendation results. The customers' similarity information can be used to recommend the products purchased by similar customers. Also, transactional data can help in understanding the purchasing patterns of customers. All these data sources, if unified, can help in the generation of faster search results. In this phase, these data sources are unified for creating the recommendation graph.

The data sources considered for the unification are: transactional data, log files, and external data sources. Based on these data sources, the recommendation graph is generated with customers and products nodes and their relationships. The $CUSTOMER$ and $PRODUCT$ node types in the recommendation graph are represented as C and P, respectively (Figure 2). Also, three types of relationships are created in the recommendation graph as:

- *Customer-Product relationship:* The relationships between node types $CUSTOMER$ and $PRODUCT$ are represented as $C - P$ in the recommendation graph (Figure 2), representing the product purchased by the customer.

- *Product-Product relationship:* The relationships among product nodes are represented as $P - P$ in

the recommendation graph (Figure 2), which represents the similarities among products. The relationship can be utilised to find similar products for content-based recommendations.

- *Customer-Customer relationship:* The relationships among customer nodes are represented as $C-C$ in the recommendation graph, representing customer similarities. The relationship can be utilised to find similar customers for collaborative recommendations.

The graph generated in this phase is utilised in the recommendation phase for providing the results using various recommendation algorithms.

3.2 Recommendation Phase

In this phase, the results are generated for the recommendation queries. The unified graph generated in the offline phase is utilised to find the results using different algorithms.

3.2.1 Content-based Recommendation

The products similar to the products purchased by a customer are recommended using the content-based recommendation algorithm [4]. For a given customer c, products purchased by c can be retrieved by traversing the relationships of type $C - P$. Then, traversing the relationships of type $P - P$ in the recommendation graph, similar products are retrieved.

Algorithm 1 Content-based Recommendation Algorithm using Graph Traversal

Input: Recommendation Graph: RG & Customer: c
Output: Product Set: PS

1: $PS \leftarrow \phi$
2: **for** all p_i in $c.ORDERS$ **do**
3: **for** all p_j in $p_i.IS\_SIM\_P$ **do**
4: $PS.add(p_j)$
5: **end for**
6: **end for**

The algorithm for the content-based recommendation has been represented in Algorithm 1. It takes recommendation graph RG and customer c as input and generates a set of products PS to be recommended to c. The recommendation graph consists of nodes: $CUSTOMER$, $ORDER$, and relationships: $ORDERS$, IS_SIM_C, IS_SIM_P (Figure 4). Initially, PS is assigned ϕ as there are no products in the recommendation list (line 1). For each product p_i retrieved from traversing the ORDERS relationship, the products similar to p_i are added to PS and recommended to c (lines 2–6).

3.2.2 Collaborative Recommendation

The products purchased by customers similar to the given customer are recommended using collaborative recommendation techniques. For a given customer c, similar customers can be retrieved by traversing the relationships of type $C - C$. After that, traversing the relationships of type $C-P$ in the recommendation graph, products are returned.

Algorithm 2 Collaborative Recommendation Algorithm using Graph Traversal

Input: Recommendation Graph: RG & Customer: c
Output: Product Set: PS

1: $PS \leftarrow \phi$
2: **for** all c_i in $c.IS_SIM_C$ **do**
3: **for** all p_j in $c_i.ORDERS$ **do**
4: $PS.add(p_j)$
5: **end for**
6: **end for**

The algorithm for the collaborative recommendation has been represented in Algorithm 2. It takes recommendation graph RG and customer c as input and generates a set of products PS to be recommended to c. The product set PS is initialised to ϕ (line 1). Similar customers are retrieved from the c by traversing the IS_SIM_C relationships (line 2). For each customer c_i similar to c, the products ordered by c_i are added to PS and recommended to c (lines 2–6).

Therefore, the proposed framework can be utilised to speed up the recommendation by unifying data sources into the graph model.

Table 1: Datasets and their size

Dataset	Size (in MB)
D1	0.5
D2	2.5
D3	33.5

4 Experimental Details

4.1 Dataset

The dataset used for the experimentation has been crawled from Kaggle consisting of approximately 400K e-commerce transactions. Using these transactions, the

data is pre-processed, and product similarity and customer similarities are computed using different mapping techniques [6]. The pre-processed data consists of five relations: (i) $CUSTOMER$ for storing customer details, (ii) IS_SIM_C for storing customers similar to given customers, (iii) $ORDERS$ for storing products ordered by customers, (iv) IS_SIM_P for storing products similar to given products, and (v) $PRODUCT$ for storing product details. The relationships among these relations are shown in Figure 3.

The corresponding graph model generated using the proposed architecture consists of two nodes: (i) $CUSTOMER$, (ii) $PRODUCT$, and three relationships: (i) $ORDERS$, (ii) IS_SIM_C, (iii) IS_SIM_P (Figure 4). Different subsets (D1, D2) of the original dataset (D3) are created to check the proposed architecture's scalability. The datasets and their size (in MB) are shown in Table 1.

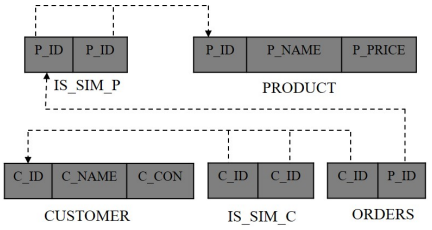

Figure 3: Data representation using Relational Model

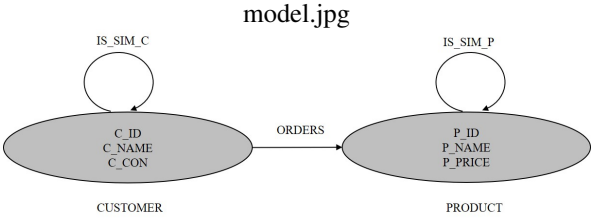

Figure 4: Data representation using Graph Model

4.2 Measures

For the performance comparison of the proposed graph model with the relational model, we consider average execution time, which is calculated as an average of the execution times of queries run ten times.

4.3 Tools

- *MySQL:*

  MySQL: For the implementation of the relational model, Oracle MySQL Community Server version 5.7 is chosen. The features for the consideration of MySQL are object-oriented, open-source availability, and community support.

- *Neo4j:* For the implementation of the graph model, Neo4j Community Edition version 3.3.0 is selected, managed supported by Neo4j Inc. for the implementation of the graph model. Various features such as open-source availability, maturity, and support, are considered to select Neo4j for graph model implementation.

4.4 Setup

The experiments have been run on a system running on Windows 10, with an Intel Core i5 processor and 8 GB RAM. No other program except system programs were running during the experimentation. The data from datasets D1, D2, and D3 (section 4.1) are loaded into MySQL and Neo4j for the performance analysis of various recommendation algorithms.

5 Results and Discussions

To analyse the performance of the proposed architecture, the recommendation algorithms (section 3) have been executed using the generated graph model (Figure 4) and compared with the underlying relational model (Figure 3) for datasets D1, D2, and D3 (section 4.1). Also, the asymptotic analysis of the recommendation algorithms using the proposed graph model has been performed and compared with the relational model.

5.1 Content-based Recommendation

To analyse the performance of the content-based recommendation, Algorithm 1 (section 3.2.1) has been executed for the randomly generated customers on the graph model (Figure 4) as well as the underlying relational model (Figure 3). The average execution time is computed by averaging the execution times (in milliseconds) of queries run ten times. As observed from Table 2, the average execution time of the graph model is lesser than that of relational models for all datasets. As the size of the dataset increases, the reduction in average execution time is more significant. The slope of the average execution time using the graph model is lesser than that of the relational model. It signifies the slower growth of execution time of the graph model with dataset size (Figure 5). Therefore, the graph model performs much better than the relational model for larger datasets.

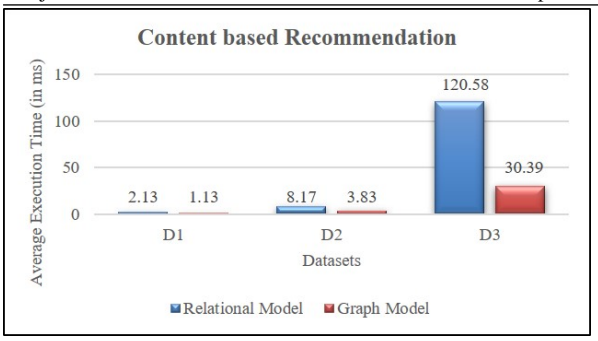

Figure 5: Performance Comparison of graph model with relational model for content based recommendation

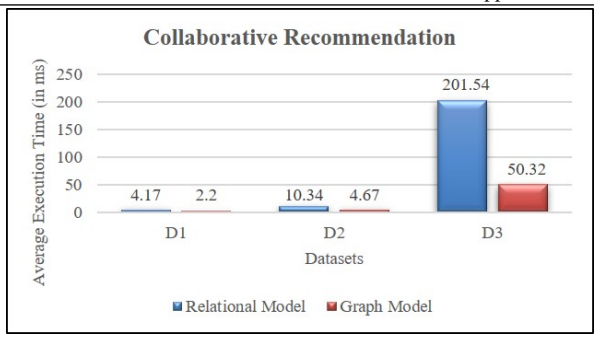

Figure 6: Performance Comparison of graph model with relational model for collaborative recommendation

Table 2: Average execution times of different recommendation algorithms for different datasets

Dataset	*Algo.*	*Ave.Exe.Time*		*Reduction*
		Relational	*Graph*	in (%)
D1	CB	2.13	1.13	46.94
D2		8.17	3.82	53.24
D3		120.58	30.39	74.79
D1	CL	4.17	2.2	53.19
D2		10.34	4.67	54.83
D3		201.54	50.32	75.03

CB: Content-based; CL: Collaborative

5.2 Collaborative Recommendation

To analyse the performance of collaborative recommendation, Algorithm 2 (Section 3.2.1) has been executed for the randomly generated customers on graph model (Figure 4) as well as the underlying relational model (Figure 3). The average execution time of the graph model is found to be lesser than that of the relational model for datasets D1, D2, D3 (Table 2). For dataset D3, the percentage reduction in average execution time using the graph model is 75.03% which is more significant than D1 and D2. As observed from Figure 6, the slope of average execution time using relational model is higher than that of graph model, signifying the slower growth of execution time of graph model with dataset size.

5.3 Asymptotic Analysis

To justify the applicability of the proposed framework, complexities of content-based and collaborative recommendation algorithms using the graph model and the underlying relational model have been analysed. Using the relational model (Figure 3) and graph model (Figure 4) discussed in section 4, an example has been illustrated to show the asymptotic complexity of the relational model and graph model as shown in Figure 7.

Content-based Recommendation To find the content-based recommendation results for the customer with C_NAME as $'ROY'$ using the relational model, its C_ID (C1) is fetched from the $CUSTOMER$ relation. The products ($P\_ID : P1$) ordered by $'C1'$ can be retrieved from the $ORDER$ relation. Subsequently, the product ($P\_ID : P2$) similar to $P1$ are fetched from the IS_SIM_P relation. The attributes of similar products ($P2$) are fetched from the PRODUCT relation (Figure 7 (a)). To achieve the same using the graph model, C_ID of $'ROY'$ can be fetched using node/index-based searching on C_NAME on the nodes of type CUSTOMER. Traversing the ORDERS relationship from the CUSTOMER node, nodes of type PRODUCT ($P\_ID : P1$) ordered by $'ROY'$ is retrieved. Subsequently, traversing the IS_SIM_P relationship from $P1$, similar product nodes ($P\_ID : P2$) are retrieved (Figure 7 (b)).

Assuming n, p, q, r, s be the number of rows in the CUSTOMER, IS_SIM_C, $ORDERS$, IS_SIM_P, and $PRODUCT$ relations, respectively. The complexity of content-based recommendation using relational model is: $O(logn) + O(logq) + O(logr) + O(logs)$. Whereas, the complexity of content-based recommendation using graph model is $O(logn + c)$ in which $O(logn)$ represents the time for accessing required CUSTOMER node and $O(c)$ represents the time for accessing the $ORDERS$ and IS_SIM_P relationships, which is independent of the count of the $ORDERS$ and IS_SIM_P relationships.

Collaborative Recommendation To find the collaborative recommendation for the customer with C_NAME as $'ROY'$ using relational model, its

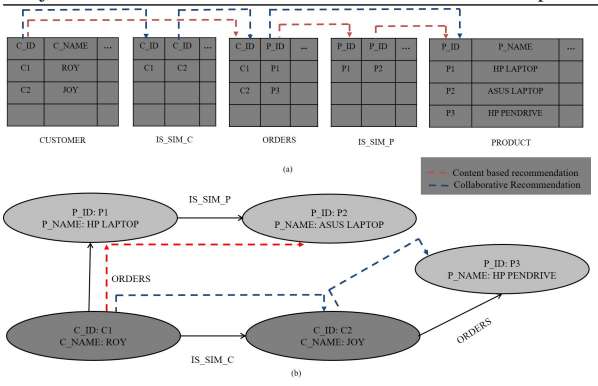

Figure 7: Asymptotic analysis of Content-based and Collaborative recommendation using (a) relational model, and (b) graph model

$C\_ID(C1)$ is fetched from the CUSTOMER relation. The customers ($C\_ID : C2$) similar to $C1$ are fetched from the IS_SIM_C relation. Subsequently, the products ($P\_ID : P3$) ordered by $C2$ are fetched from the $ORDERS$ relationship. The attributes of retrieved products ($P3$) are fetched from the PRODUCT relation (7 (a)). To achieve the same using the graph model, C_ID of $'ROY'$ can be fetched using node/index-based searching on C_NAME on the nodes of type $CUSTOMER$. Traversing the IS_SIM_C relationship from the $CUSTOMER$ node, similar customer nodes ($C\_ID : C2$) are retrieved. Subsequently, traversing the $ORDERS$ relationship from $C2$, product ($P\_ID : P2$) ordered by $C2$ is retrieved (7 (b)).

For the collaborative recommendation, the complexity using relational model is: $O(logn)+O(logp)+O(logq)+O(logs)$, where n, p, q, r, s be the number of rows in the $CUSTOMER$, IS_SIM_C, $ORDERS$, IS_SIM_P, and $PRODUCT$ relations, respectively. On the other hand, the complexity using graph database is $O(logn + k)$ where $O(logn)$ represents the time for accessing the $CUSTOMER$ node using label/index-based searching, and $O(k)$ represents the time for accessing the IS_SIM_C and $ORDERS$ relationships from the $CUSTOMER$ node.

Therefore, the complexity of the graph model is not affected by the number of products, orders, the similarity between customers, the similarity between products, etc. With the increase in the number of customers, products, orders, similarities, etc., the recommendation complexity increases faster for the relational model than the graph model. Therefore, the proposed graph model is beneficial for recommendation algorithms which are also verified using empirical and asymptotic techniques.

6 Conclusion

The graph model is revolutionizing in relationship-centric applications due to its flexibility and index-free adjacency. The recommendation is an intrinsic task supported by e-commerce applications to provide customised results to the customers. To improve the execution time of the recommendation, an architecture has been proposed to map the data into a graph. The recommendation algorithms based on graph traversal have also been proposed for the proposed architecture. To justify the applicability of the proposed framework, the performance of generated graph model with the underlying relational model has been compared for different recommendation algorithms. The graph model outperformed the relational model for the content-based as well as collaborative recommendation algorithms. The performance of the graph model using asymptotic analysis has also been verified and compared with the relational model. Thus, the proposed framework can be used in e-commerce as well as other applications for the recommendation. In the future, the proposed framework can also be applied in other applications for the recommendation.

References

[1] Abulaish, M., Sharma, S., and Fazil, M. A multi-attributed graph-based approach for text data modeling and event detection in Twitter. In *Proc. 11th Int. Conf. Communication Systems Networks*, pages 703–708. IEEE, 2019.

[2] Angles, R. A comparison of current graph database models. In *Proc. IEEE 28th Int. Data Engineering Workshops*, pages 171 – 177. IEEE Computer Society, 2012.

[3] Atzeni, P., Jensen, C. S., Orsi, G., Ram, S., Tanca, L., and Torlone, R. The relational model is dead, SQL is dead, and I don't feel so good myself. *SIGMOD Record*, 42(2):64–68, July 2013.

[4] Balabanović, M. and Shoham, Y. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[5] Cao, Y., Wang, X., He, X., Hu, Z., and Chua, T.-S. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proc. The World Wide Web (WWW) Conf.*, pages 151 – 161. ACM Press, 2019.

[6] Costa, L. d. F., Oliveira Jr, O. N., Travieso, G., Rodrigues, F. A., Villas Boas, P. R., Antiqueira,

L., Viana, M. P., and Correa Rocha, L. E. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329 – 412, 2011.

[7] Ding, L., Han, B., Wang, S., Li, X., and Song, B. User-centered recommendation using US-ELM based on dynamic graph model in E-commerce. *Int. Journal Machine Learning & Cybernetics*, 10(4):693 – 703, 2019.

[8] Huang, H. J., Yang, J., and Zheng, B. Demand effects of product similarity network in e-commerce platform. *Electronic Commerce Research*, pages 1 – 31, 2019.

[9] Huang, Z. *Graph-based analysis for e-commerce recommendation*. PhD thesis, The University of Arizona, 2005.

[10] Huang, Z., Zeng, D., and Chen, H. A comparative study of recommendation algorithms in e-commerce applications. *IEEE Intelligent Systems*, 22(5):68 – 78, 2007.

[11] Kumar, P. Graph data modeling for political communication on Twitter. Master's thesis, Dept. of Computer Science, Iowa State University, 2016.

[12] Li, F.-L., Chen, H., Xu, G., Qiu, T., Ji, F., Zhang, J., and Chen, H. AliMeKG: Domain knowledge graph construction and application in e-commerce. In *Proc. 29th ACM Int. Conf. on Information & Knowledge Management (CIKM)*, pages 2581 – 2588. ACM Press, 2020.

[13] Ma, S., Li, J., Hu, C., Lin, X., and Huai, J. Big graph search: challenges and techniques. *Frontiers of Computer Science*, 10:387 – 398, 2016.

[14] Noel, S., Harley, E., Tam, K., Limiero, M., and Share, M. CyGraph: graph-based analytics and visualization for cybersecurity. In *Cognitive Computing: Theory & Applications*, volume 35 of *Handbook of Statistics*, pages 117–167. Elsevier, 2016.

[15] Petermann, A., Junghanns, M., Müller, R., and Rahm, E. Graph-based data integration and business intelligence with BIIIG. *Proc. VLDB Endow.*, 7(13):1577 – 1580, 2014.

[16] Sharma, S. An extended classification and comparison of nosql big data models. *CoRR*, abs/1509.08035, 2015.

[17] Srinivasa, S. *Data, Storage and Index Models for Graph Databases*, pages 47–70. IGI Global, 2012.

[18] Tuteja, S. and Kumar, R. A system architecture for mapping application data into complex graph. In Kaushik, S., Gupta, D., Kharb, L., and Chahal, D., editors, *Information, Communication and Computing Technology*, pages 148–155, Singapore, 2017. Springer Singapore.

[19] Tuteja, S. and Kumar, R. An Architecture for Data Unification in E-commerce using Graph. In *Strategic System Assurance and Business Analytics*, pages 407 – 417. Springer, 2020.

[20] Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., and Wilkins, D. A comparison of a graph database and a relational database: A data provenance perspective. In *Proc. 48th Annual Southeast Regional Conf.*, pages 1 – 6. ACM Press, 2010.

[21] Yang, K. and Toni, L. Graph-based recommendation system. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 798–802, 2018.

[22] Yoon, B.-H., Kim, S.-K., and Kim, S.-Y. Use of graph database for the integration of heterogeneous biological data. *Genomics & informatics*, 15(1):19 – 27, 2017.