R| R |S |T

Researchberg Review of Science and Technology

Vol. 3 No. 1
2020

Research Article

Graph Neural Network for Service Recommender System in Digital Service Marketplace

Sandeepkumar Racherla

https://orcid.org/0000-0002-8155-5643

Abstract

The emergence of the platform economy has resulted in the decline of many traditional forms of doing business. Freelance work makes use of a platform to connect businesses or people with other businesses or persons in order to solve particular issues or deliver specific services in return for payment. The pairing process involves a buyer that needs work done, a platform that handles the algorithm, and a worker who is willing to do the job via the platform. This research argues that by efficiently pairing the talents of workers to the requirements of buyers, the platforms have the ability to expedite business operations for buyers, empower platform workers, and significantly improve the overall customer experience. Graph Convolutional Networks (GCNs) are inspired by CNNs and aim to expand the convolution operation from grid records to graph records, which in turn facilitates advances in the graph domain. In order to develop reliable and accurate embeddings for digital service recommendation, we employed a graph-based technique on a freelance platform dataset using the graph linkages of services and buyer data. We employed an aggregation-based inductive graph convolution network, namely, Graph SAmple and aggreGatE (GraphSAGE). It is a generalized inductive architecture that learns to construct embeddings for previously unknown data by sampling and combining attributes from a node's immediate neighborhood. We also applied PinSage, a stochastic Graph Convolutional Network (GCN) that can learn node embeddings in platform networks with many digital services. When a robust recommender system is used in digital service marketplace, it can offer promising results that may increase users' satisfaction with the service and boost the platform's ability to increase revenue.

Keywords: Digital service, Graph Convolutional Networks, GraphSage, Recommendation system, PinSage

Introduction

The *Platform Economy* has an influence on the workforce, the types of available employment, and the form of those occupations, besides having an impact on retail, consumer behaviors, and sales methods [1]. Many jobs have been digitized or completely overtaken by technology. Freelance and other digital marketplaces are being spurred forward by three forces: accessible network connectivity with rising mobile usage, reputation systems that allow trust among scattered people, and access to low-cost technology with information and tools to collect and manage interactions [2], [3].

With the proliferation of information being delivered as online content content, online shopping, and streaming sites in recent years, discovering and accessing necessary information in an ideal manner has become critical for many organizations and professional institutions. With various platform participants continually vying for users' interests, one of the internet's most popular features is the relevant data supplied by recommendation systems [4]. They are also beneficial for attracting consumers to a company's online products and services.

Recommender systems attempt to forecast users' interests and are able to propose digital services that are likely to be of interest to them. They are some of the most advanced machine learning algorithms used by businesses to increase sales. In order to provide individualized suggestions, the algorithm must learn something about each user. Any recommendation systems must create and preserve a user profiles or user preferences, which includes information such as the user 's interests. The suggestion mechanism is mostly based on feedback data. Platforms can collect information about consumer preferences for a variety of digital services (e.g., logo making, content creation, SEO service, web development, and so on), which may be gathered either explicitly or implicitly. Explicit data is gathered via direct user input, such as digital service reviews and feedback. Implicit information is obtained from the user's search activity, count of visits, or text message information, etc. Another sort of signals is the usage of other types of data such as demographic and/or social data provided by users.

A recommender system's principal purpose would be to increase digital service sales [5], [6]. After all, platforms can use recommender systems to boost their profits. Recommendation system carry relevant things to the notice of consumers by suggesting carefully picked digital services. This enhances the platforms' sales volume and earnings. Although the fundamental purpose of a recommender systems is to improve income for the platforms, this is sometimes accomplished in less apparent ways than may seem at first glance. The following are the typical operational and technological aims of recommender systems in order to accomplish the wider business-centric goal of boosting revenue in a digital marketplace platform:

a) Relevance: A recommender system's most apparent operational purpose can be to recommend digital services that are of interest to the user at present [7]. Digital services that users find intriguing are much more likely to be purchased. Although relevance is a recommender system's principal operational aim, it is insufficient on its own. As a result, there are various secondary objectives that are not as crucial as relevancy still are nonetheless substantial enough to have an influence.

b) Innovativeness: Recommendation systems systems can be very useful when the suggested digital service is something the viewer has never seen before [8]. Popular services in a user's favored services, for example, are unlikely to be unique to the user. Recommending popular digital services repeatedly might likewise reduce sales variety.

c) Surprise: A similar concept is surprise, in which the digital services suggested can be slightly unexpected, and so there is a little element of chance discovery, as compared to obvious suggestions. Surprise differs from innovation in that the suggestions are actually unexpected by the user, instead of merely something they were unaware of before [9]. It is not uncommon for a given user to exclusively consume digital services of a single kind, despite a latent desire in things of other sorts that the user may find unexpected. In contrast to novelty, serendipitous approaches are concerned with finding such suggestions. Surprises have the advantage of expanding sales variety or starting a new pattern of attention in the user. Because of the prospect of uncovering whole new areas of interest, increasing surprise typically offers long-term and strategic advantages for the business. Algorithms that make spontaneous suggestions, in contrast, can propose unrelated digital services. Many times, the long-term and strategic advantages of unexpected tactics exceed the short-term drawbacks.

d) Increasing the variety of recommendations: Typically, recommender systems will provide a list of the top-k things. When all of the suggested things are highly similar, the user is likely to dislike all of them. When the suggested list, in contrast, comprises things of several sorts, the user is more likely to prefer at at-least one of these digital services [10]. The advantage of diversity is that it prevents the consumer from being bored with repeated recommendations of similar services.

A recommender, which is often disregarded as a method of market analysis, may be used to identify users' interests and see what individuals are often interested in. Businesses may guarantee they provide comparable digital services by using ratings and reviews and the counts of consumers viewing a program.

Recommendation system for digital services

While the concept of recommender systems may be traced back to the field of information retrieval, it has developed into its own topic of study since the mid-1990s. Traditionally, recommender systems have gained a lot of focus for applying artificial intelligence approaches to information filtering, such as recommending web sites or filtering and ranking news articles. In reality, recommendation strategies like case-based or rule-based procedures find their origins in the 1980s expert systems [11], [12].

Links can give enormous insight and knowledge of customers to recommender systems in the context of digital service marketplaces. There are three sorts that may occur. The first is digital service-user relationship. When certain consumers have an interest or desire for particular digital services that they need, a user-digital service connection develops. A small business owner, for example, may have a preference for small business-related services, so the platforms sites will create a user-digital service connection of small business owner.

The second is the link between different digital services. When objects are related in nature, whether by look or description, they form service-service associations. Blogpost writing services of the similar topics, thumbnail designing services of the books or magazines, or graphic designs on a certain event are some examples. The third type of link is the link between users. When many clients have similar tastes in terms of a certain digital service or service, user-user connections form. Mutual friends, same backgrounds, comparable ages, and so on are all examples.

The recommendation issue may be expressed in a variety of ways. Below we discuss the two major models in the context of recommending a digital service [13], [14]:

1. Problem estimation: The first strategy is to forecast the rating for a buyer-service pair. It is expected that training data reflecting user preferences for digital services is available. This relates to an incomplete m by n matrix for m individuals and n digital services, where the provided (or reported) values are utilized for training. This training model is used to forecast missing (or unseen) variables. Since we have an imperfectly described matrix of values, the additional values are anticipated by the learning process, this issue is also known as the matrix completeness problem.

2. Problem ranking version: In fact, it is not essential to forecast user ratings for individual things in order to create user recommendations. Rather, a platform may want to propose the top-k digital services for a certain user or identify the top-k individuals to target for a specific service. Although the approaches in the two are completely comparable, determining the top-k things is more frequent than determining the top-k users. Different recommender systems can be employed in the context of digital service marketplace.

1. Collaborative filtering in recommending digital services.

The main assumption behind these algorithms would be that if users had similar interests in the past, such as seeing or purchasing the same digital services, they would have similar preferences in the future [15]. So, if user X and user Y have a buying pattern that substantially overlaps, and user X has just purchased a a digital service that Y has not yet viewed, the fundamental logic is to recommend this service to Y as well. This approach is also known as collaborative filtering since it requires screening the most promising digital services from a big collection and because the customers implicitly cooperate with one another.

Pure collaborative filtering techniques do not make use of or need knowledge of the things themselves. Continuing with the digital service, the recommendation systems are not required to know what a service is about, what category it belongs to, or its price. The clear benefit of this method is that such data do not need to be put into or maintained in the system. On the contrary, utilizing such qualities to recommend digital services that are similar to ones the user has previously enjoyed may be more beneficial.

Collaborative filtering has some disadvantages to in the context of digital service recommendations. This technique cannot work with fresh digital services. The model's prediction for a given (person, service) combination is the dot product of digital service of their associated embeddings. As a result, if a service is not encountered during train process, the system cannot generate an embedding and cannot access the model with it. This is sometimes referred to as the cold-start issue in the recommendation literature.

2. Content-based recommendation in recommending digital services

Generally, recommender systems may fulfill two functions. On the one hand, they may be used to entice consumers to do things like purchase a social media marketing service or buy a translation service. Recommender systems, on the other hand, might also be seen as instruments for coping with information overload since they seek to choose the most interesting digital services from a wider collection. As a result, recommender system research is deeply based in data acquisition and information filtering. However, in these areas, the emphasis is mostly on the difficulty of distinguishing between important and irrelevant materials. Many of the tools developed in these domains rate documents using information drawn from their contents.

The presence of (manually written or electronically extracted) digital service details and a service provider profile that gives priority to these features are at the heart of content-based recommendation [16]. For example, a freelance writing service-attributes can contain the category, the particular subject, or the language of writing. User profiles, like descriptions or bio may be automatically produced and "learned" by evaluating user activity and feedback or by directly inquiring about interests.

When opposed to the above-mentioned alternatives, content-based recommendation offers two benefits. First, vast user clusters are not required to obtain decent suggestion accuracy. Furthermore, once service qualities are accessible, additional things may be promptly suggested. Such service descriptions may be automatically derived (for example, from textual information) or are already accessible in a digital catalog. However, in many fields, more subjective attributes of an item, such as "easy of use" or "sophistication of design," may be valuable in the recommender system. These qualities, however, are difficult to gather automatically, implying that such data must be manually input into the program in a potentially costly and error-prone procedure.

3. Knowledge recommendation in recommending digital services

When we look at some digital service categories, we observe that there are a lot of one-time customers [17]. This implies we can not depend on the presence of a past purchases, which is required for content-based or collaborative filtering systems [18]. However, more thorough and organized material, containing quality and technical elements, may be offered.

A recommendation system for digital services like purchasing a logo, or a business website design, which should assist the end user in locating a service seller that meets his or her specific needs. Because most businesses only purchase a new logo or a website design services every few years, the recommendation systems system cannot create a user profile or suggest services that others bought, which would result in just suggesting top-selling digital services.

As a result, a system that uses extra and means-end information to provide suggestions is required. The recommender system in such knowledge-based techniques often takes use of extra, sometimes manually given, data about both the present user and the accessible digital services.

4. Hybrid Recommendation in recommending digital services

As already discussed, various techniques presented so far have benefits and downsides depending on the issue scenario. One apparent option is to combine several methodologies to provide better or more exact suggestions [19], [20]. If, for example, there is community knowledge and comprehensive information on particular digital services, a recommender system might be improved by combining various techniques.

Graph data has earned a significant amount of interest in the big data age [21], from social networks to biological systems to recommender system. Graphs, which define pairwise relationships between things, are critical abstractions for actual data from a wide range of fields. Over the last several decades, the area of graph representation has advanced significantly, and it can be loosely split into three generations: conventional graph embedding, contemporary graph embedding, and, finally, the graph deep learning [22]. Given the enormous success of deep learning methods in representation learning, attempts have been made to extend them to graphs, which has opened a new phase in graph representation learning, namely deep learning on graphs.

More and more data suggest that the latest version of graph representation learning, particularly graph neural networks (GNNs), has significantly aided computational activities on graphs. GNNs' groundbreaking advancements have also greatly aided the richness and scope of graph representation learning adoption in real-world applications [23]. GNNs achieve state-of-the-art performance and push the boundaries of graph representation learning in traditional application areas such as recommendation systems systems.

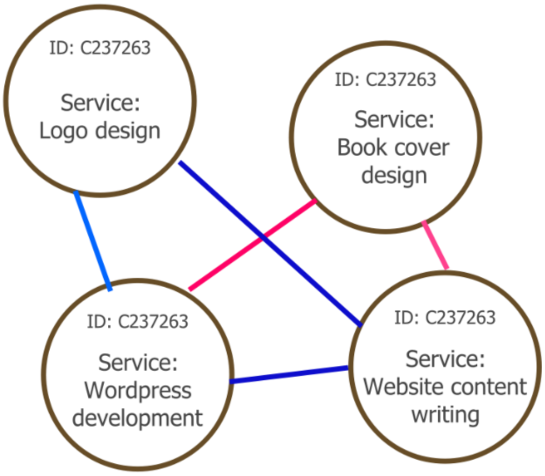
Nodes within graphs are naturally related, implying that nodes really are not dispersed independently and equally. As a result, typical machine learning approaches cannot be applied successfully to graph computing problems.

Methods

Graph neural networks may be thought of as a representation learning system on graphs. GNN research began in the early 21st century, when the very first GNN architecture was suggested [21], [24]. Deep learning approaches acquired significant appeal in many fields, including natural language processing and computer vision, and academics began to devote more resources to this area of study.

GraphSAGE expands GCN to the goal of being inductive supervised methods for new dataset, in which each node is defined by the neighborhood aggregate [25]. As a result, even if a previously unknown node arises in the network during training, it may still be accurately represented by its neighbors [25]. On the one hand, unlike GCN, GraphSAGE selects local neighbors for every node and then learns how to combine features extracted from these selected neighbor nodes in a mini batch manner, that is inductive and useful for large networks. GraphSAGE, on the other hand, expands GCN's aggregator and presents a number of alternative operators.

Figure 1. digital service Graph for GraphSAGE



GraphSAGE evenly samples a fixed-size collection of neighbours at each cycle. Because the present GCN's input is a fixed size entire graph, the constants of GCN must be modified with the gradient of all training dataset in each epoch, a process known as complete batch learning. Because the entire graph is frequently enormous in reality, GraphSAGE selects a fixed-size collection of neighbours for every node in mini batch and computes gradients to modify parameters.

GraphSAGE investigates the fundamental features of an aggregator: In contrast to machine learning over phrases and pictures, a node's neighbors do not have a natural ordering; hence, the aggregator algorithms must work over an unsorted collection of vectors. An aggregator procedure should ideally be symmetrical (i.e., uncorrelated with variations of its values) while being easy to train and having a large representational power.

GraphSAGE Algorithm [25] : GraphSAGE embedding generation

Input: Graph $G(V,\varepsilon)$; input features $\{x_v, \forall_v \in v\}$; depth K; weight matrices $W_k, \forall k \in \{1,...k\}$; non-linearity σ; differentiable aggregator functions $AGGREGATE_k, \forall k \in \{1,...k\}$; neighborhood function $N : v \rightarrow 2^v$

Output: Vector representations z_v for all $v \in V$

$h_v^0 \leftarrow x_v, \forall_v \in V;$

for K=1…k **do**

  for $v \in V$ **do**

    $h_{N(v)}^k \leftarrow AGGREGATE_K(\{h_u^{k-1}\}, \forall u \in N(v)\});$

    $h_v^k \leftarrow \sigma(W^k.CONCAT(h_v^{k-1}, h_{N(v)}^k))$

  end

  $h_v^k \leftarrow \frac{h_v^k}{\| h_v^k \|_2}, \forall v \in V$

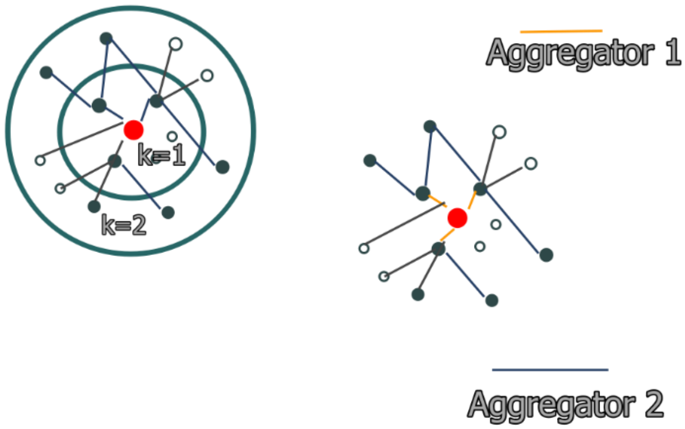
End

$z_v \leftarrow h_v^k, \forall v \in V$

Then, GraphSAGE suggests a number of suitable aggregator functions, including the LSTM, Mean, and Pooling aggregators [25], [26].

This study also applied PinSage. PinSage is a random-walk GCNN that can learn node embeddings in web networks with many items [27]. Pinterest, an online pinboard firm, created the network to provide users with topical visual suggestions. PinSage generates robust embeddings from pins using bipartite graphs of pins and boards, which are then used to promote aesthetically comparable material to users. Unlike classic GCN algorithms, which conduct convolutions on features matrices and the whole graph, PinSage samples neighboring nodes and conducts more effective local convolutions using dynamic graph creation [27]. Convolutions on a node's whole neighborhood will generate a huge computational graph. Conventional GCN techniques adjust a node's presentation by collecting input from its k-hop region to decrease resource needs. PinSage mimics a random walk to choose frequently accessed material as the core neighborhood and then builds a convolution around it.

Figure 2. GraphSAGE Algorithm.



Data preprocessing and results

The digital services in the digital service data file are present by their unique identification numbers. We started by removing all entries with missing data. Next, we filtered our data to preserve only services with associated "also bought services", which displays all ID of services purchased alongside that exact service. These "also bought" digital services characterize our edge list's target nodes. Following that, we deleted any "also bought services" that were not in our dataset. The category column is then manipulated to another column to obtain its sub-category or selected niche in order to have a label for the digital service nodes. Since IDs are strings, we can't utilize them in our model, thus we assigned each ID in our dataset a unique integer. Then we created an edge list. Then, using a TF-IDF Vectorizer, we vectorized the digital service's name text features to serve as the only digital-service node feature. We started generating our graph after collecting node labels (niche area), node characteristics (digital service name), and edge list, and then fed the graph and graph attributes into our model.

We then constructed a graph of all edges and nodes and separated the test and training sets according to the edges. We used a 80-20 train-test split, with 20% of the edges in the testing graph and 80% of the edges in the train graph set. This was done so that our model could learn feature connections based on comparable but not identical ones.

In the review's dataset, digital services are recognized by their unique IDs, and users are identified by their unique ID, followed by the removal of all rows with missing value. We compute the helpful column via dividing the helpful by the total number of votes cast, yielding a helpfulness ratio of 0.0 if no total votes were cast. We feed in the feature representation dataset using the digital service unique IDs discovered in our review's dataset, preserving just the digital service unique IDs that exist in our reviews dataframe, generating a final list of services with both reviews and picture features. The digital service metadata is then read in, and the item metadata and reviews are filtered using the final service ID list. Missing digital service prices are calculated by taking the median price, multiplying it by 100, and converting it to an integer. The dataset is divided into training, validation, and testing datasets. The model has been trained on earlier review actions to forecast 'future' review actions, using reviewing time as the index. The validation and test sets are transformed into user-digital service adjacency matrices, while the training set is transformed into a graph.

We utilized a statistic named hit rate for top-S suggested digital services with S = 500 to assess the digital service suggestions of our algorithms. Given an unknown digital service, the GraphSAGE algorithm recommends the top-S nearest embedded digital service nodes. It is deemed a hit if at minimum one of the suggested digital services is linked to the unknown digital service. The hit rate at K is determined by the amount of hits divided by the quantity of unseen test digital services. Similarly, for each user, the PinSage algorithm provides the top-S nearest embedded digital service nodes. It is deemed a hit if the user reviews at least one of the suggested digital services in the test matrix. The hit rate at S is calculated as the volume of hits divided by the number of visitors.

Table 1. Model Performances

Method	Hit Ratio	Loss
GraphSage	0.59	0.14
PinSage	0.16	0.31

Table 2. GraphSage recommendations

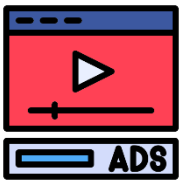
Test service	Recommended services by Graph Neural Network (GraphSage)		
 **logo design for business or product**	 YouTube thumbnail designs.	 Business cards design	 Facebook and Instagram ads design
 SEO keyword research	 website on-page SEO,	 Instagram and Facebook page promotion	 set up Facebook ads campaigns
 promotional video ad making	 animated explainer video of product	 product video making for e-commerce	 motion graphics animated logo

Table 3. PinSage recommendations

Services users reviewed			Recommended services by Graph Neural Network (PineSage)		
product video making for e-commerce platform	animated explainer video of product	promotional video ad making	commercial brand video	facebook video ads	instagram ads
WordPress website development	WordPress landing page building	3-D Logo design	ecommerce store development	responsive WordPress website design	Yoast SEO WordPress
write SEO website content or blog posts	write blog posts in German	writing diet blog posts, articles and eBooks	Translation service from German to English and French	Writing about "us page" and team	Writing real estate blog post

Conclusion

The platform economy has transformed the way individuals work in recent years and is becoming a significant field of academic research. It is desirable to provide such extra value to consumers by integrating suggestions in systems and digital services. Furthermore, it enables businesses to position themselves ahead of their competition and, as a result, enhance their revenues. Recommender systems are gaining popularity in academia, business, and commerce. It is a crucial component of the digital service marketplace environment since it reduces a big quantity of data to a modest number and suggests it to the service buyers according to his preferences, wants, and actions. We argued that digital service marketplace companies and users benefit from recommender systems. They lower the transaction costs associated with locating and choosing digital service in an online service-buying platform. The importance of using precise and effective recommendation strategies inside a system that provides consumers with relevant and trustworthy suggestions cannot be overstated. Recommender systems can provide new avenues for obtaining tailored knowledge on the Internet. It also aids in the alleviation of the issue of informational overload, which is a regular occurrence with information retrieval systems, and allows users access to the digital services that are not immediately accessible to users on the system. Graph neural network-based recommendation system is presently one of the most innovative methodologies, and it consistently outperforms other types of recommender systems.

References

[1] M. Kenney and J. Zysman, “The rise of the platform economy,” *Issues Sci. Technol.*, vol. 32, no. 3, p. 61, 2016.

[2] D. Farrell and F. Greig, “The Online Platform Economy: Has Growth Peaked?,” *Available at SSRN 2911194*, 15-Nov-2017.

[3] D. Farrell, F. Greig, and A. Hamoudi, “The Online Platform Economy in 2018: Drivers, Workers, Sellers, and Lessors,” *JPMorgan Chase Institute*, 24-Sep-2018.

[4] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, Nov. 2015.

[5] R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” *User Model. User-adapt Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[6] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, “A literature review and classification of recommender systems research,” *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10059–10072, Sep. 2012.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 109–132, Jul. 2013.

[8] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 73–105.

[9] G. Shani and A. Gunawardana, “Evaluating Recommendation Systems,” in *Recommender Systems Handbook,* F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 257–297.

[10] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[11] P. Pattaraintakorn, G. M. Zaverucha, and N. Cercone, “Web Based Health Recommender System Using Rough Sets, Survival Analysis and Rule-Based Expert Systems,” in *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, 2007, pp. 491–499.

[12] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, “A hybrid recommender system using artificial neural networks,” *Expert Syst. Appl.*, vol. 83, pp. 300–313, Oct. 2017.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst. Secur.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.

[14] J. Lu, Q. Zhang, and G. Zhang, “Recommender Systems,” 2012. [Online]. Available: https://www.worldscientific.com/worldscibooks/10.1142/11947.

[15] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2010.

[16] R. Winkels, A. Boer, B. Vredebregt, and A. Van Someren, “Towards a Legal Recommender System,” in *JURIX*, 2014, pp. 169–178.

[17] N. Manouselis, H. Drachsler, K. Verbert, and O. C. Santos, *Recommender systems for technology enhanced learning: Research trends and applications*, 2014th ed. New York, NY: Springer, 2014.

[18] G. Kembellec, G. Chartron, and I. Saleh, “Recommender systems,” 2014.

[19] G. Lekakos and P. Caravelas, “A hybrid approach for movie recommendation,” *Multimed. Tools Appl.*, vol. 36, no. 1, pp. 55–70, Jan. 2008.

[20] P. B. Thorat, R. M. Goudar, and S. Barve, “Survey on collaborative filtering, content-based filtering and hybrid recommendation system,” *Int. J. Comput. Appl. Technol.*, vol. 110, no. 4, pp. 31–36, 2015.

[21] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[22] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?,” *arXiv [cs.LG]*, 01-Oct-2018.

[23] J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[24] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in *Proceedings. 2005 IEEE international joint conference on neural networks*, 2005, vol. 2, pp. 729–734.

[25] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[26] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, “Representation Learning on Graphs with Jumping Knowledge Networks,” in *Proceedings of the 35th International Conference on Machine Learning*, 10--15 Jul 2018, vol. 80, pp. 5453–5462.

[27] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph Convolutional Neural Networks for Web-Scale Recommender Systems,” in *Proceedings of*

the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, 2018, pp. 974–983.