Online Recommendation Systems: A special case of information filtering systems

Whitepaper by Vibha Bhagchandani, AVP – AI & Cognitive Solutions, Mphasis NEXT Labs |
Narayan Mishra, Senior Associate Business Analyst, Mphasis NEXT Labs

Contents

1.
The Backdrop to Online Recommendation Systems

Internet-based business such as e-commerce, streaming services, online portals, etc., store enormous volumes of data. This includes user data, web traffic and transactional data that can help companies make crucial decisions regarding strategy, marketing and product roadmap, to name a few. Online platforms such as Netflix, Amazon, Flipkart, Facebook, Disney+, LinkedIn, etc., use recommendation systems to make relevant suggestions to users. With burgeoning data providing too many options on various platforms that confuse users (and cause them to abandon their search and leave the websites), a good recommendation system plays an important role in resolving their dilemma of 'choice overload' and retaining customers. Recommendation systems also improve user experience and customer satisfaction, leading to higher sales and growth – a true win-win situation for both users and businesses.

2.
The Versatility of Online Recommendation Systems

In general, recommendation systems are a sub-class of information filtering systems that manage information overload. They remove redundant data and reveal only relevant information to users. For example, web-content (music, webpages, films, etc.), recommendation systems find items that are interesting and relevant to users, rank them on multiple factors, and present them based on such rankings. This drives better user engagement, especially when presented at different levels – in home pages, product pages, search pages, out-of-stock pages and cart pages.

Home page recommendations are made at the beginning of the user journey. They can be further divided into two categories based on whether the user is logged in or not. If the user is not logged into the website, recommendations can be provided based on known factors such as location, time of the day, etc. When the user is logged in, more precise recommendations can be targeted by using collaborative filtering methods (we will describe them later in the paper). These include suggesting items for purchase based on what was purchased by similar users, where user similarities can be evaluated through their buying patterns and user specific features.

Product page recommendations, on the other hand, are provided when the user navigates to a specific product page. At this point, the user is still searching for potential products based on his/her needs and purchasing power. E-commerce businesses see this situation as an opportunity to upsell and cross-sell other products to increase cart value and revenue.

The cross-sell approach refers to recommending items that are bought together while upsell approach refers to recommending higher priced products with increased functionalities.

Search page recommendations are provided to further improve user experience and share other relevant information. These recommendations take care of incorrect search terms and provide smart recommendations such as ‘best-selling’, ‘most-viewed’, ‘most-liked’, ‘trending’, etc., according to the selected search criteria.

Out-of-stock page recommendations are provided to compensate the inconvenience faced by the user on not finding the item or brand of interest. These recommendations include items from best-selling category, similar products from other brands and also from the browsing history of the user.

Cart level recommendations are provided between the intermediate and final stages of the purchase journey – say, when the potential customer has selected a few items and added them to the purchase cart. At this stage, the system has gained more knowledge of the user’s interest, and hence can suggest related items with greater relevance and precision.

3.
Goals and Approaches of Recommendation Systems

Recommendation systems must be designed to benefit business and customers alike, and their goals should encompass the following:

Relevance

This is the primary aspect of any recommendation system, and critical to keep users engaged. The suggested items must spike users’ interest based on their historical choices, interactions or profile.

Novelty

Recommendations must have the novelty factor to ensure sustained interest (and hence sustained revenue). Repetitive recommendations can blunt users’ interest and cause decline in revenues.

Diversity

Providing users a curated variety of choices, personalized to their persona is very important to retain their interest and ensure high conversion rates.

Recommendation systems in retail industry works in three ways: cross-sell, upsell or a hybrid of cross-sell and upsell.

The Cross-sell Approach

This recommends items that are generally bought or used together – for example, an add-on, or complementary or supplementary items using item-association or frequent pattern analysis. Recommending a mouse with a computer or laptop, bread with milk or hotel accommodation with flight bookings are examples of cross-sell.

The Upsell Approach

This recommends higher-priced products with increased functionalities based on past customer journeys or successful customer transitions. Recommending a higher RAM laptop for a user of a lower RAM laptop, or a higher vehicle variant, are examples of upsell strategy.

CROSS-SELL

UPSELL

4.
Three Stages of Implementing Recommendation Systems

Let us look at the three phases in the implementation of a recommendation system – information collection, learning, prediction and recommendation.

In the *information collection phase,* user attributes and behaviors (demographic data, age, gender, education and information related to items such as domain, company, industry, business, purpose, etc.) are collated. The quality and extent of recommendations varies, based on the information collected. Explicit feedback is the most effective and convenient information for building recommendations. Some recommendation systems also use hybrid feedback systems (a combination of explicit and implicit feedback) to collect information.

Type of feedback	Properties	Requires users' effort?
Explicit	• Collected by using system prompts • Forces users to provide ratings for items of interest • Provides more accuracy	Yes
Implicit	• Collected automatically by monitoring users' actions (time spent on webpages, links clicked, purchase and navigation history, etc.) • Does not force users to provide ratings for items of interest • Delivers less accuracy	No
Hybrid	• Combines the pros of explicit and implicit feedback systems and reduces the cons • Users can or opt out of providing explicit feedback	Optional

The *learning phase* involves the training of algorithms on the data collected in the information collection phase to derive insights and learning. Generally, unsupervised ML algorithms like clustering methods, feature pattern mining, association rule mining, etc., are leveraged to extract insights.

The *prediction and recommendation phase* leverages data, insights and the 'user-item' relationship pattern (that were extracted in the learning phase) to recommend products to customers and potential customers. For example, similar genre movies to viewers, based on their preferences of movies watched in the past, can be recommended on future visits.

5.
Methods of Building Recommendation Systems

Based on the information available about users, transactions and scenarios, different methods can be deployed to build a recommendation engine.

Content-based Recommendation Systems

This method suggests items to a user based on a combination of his/her profile and item attributes. User profiles can be built based on information provided directly by the user (demographics, rating preferences, etc.) or generated implicitly by the system (user interactions, browsing history, etc.). The item's attributes are known features such as movie genre, language, region or book language, publisher and author.

Collaborative Filtering-based Recommendation Systems

The main idea behind collaborative filtering systems is that similar users like similar items.

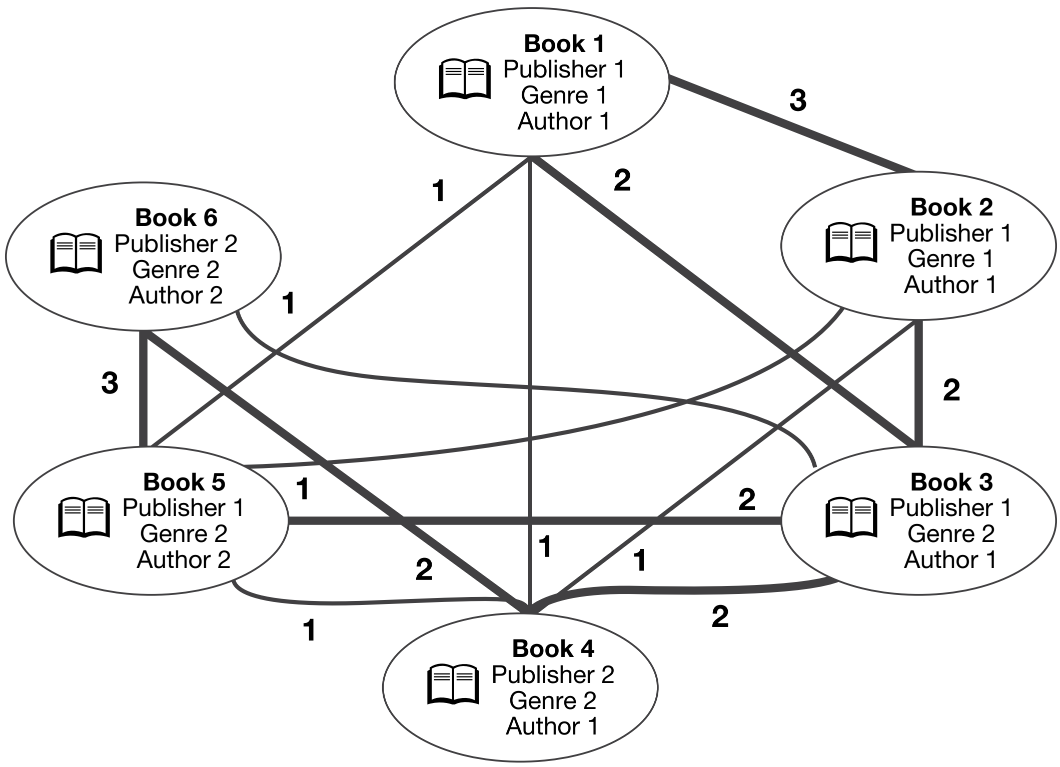
- *Memory-based methods* work on the premise that similar users have similar way of rating items and similar items receive similar ratings. However, in real-world scenarios, all users do not interact with all items – and this leads to data sparsity, which in turn, increases the computation complexity of pattern recognition. Such cases can be resolved using graph-based recommendation systems.

User-based collaborative filtering	**Item-based collaborative filtering**
The neighboring users of the target user are identified, and the ratings for target items are predicted using the ratings of neighboring users.	The neighboring items of the target item is identified, and the ratings for the target items are predicted using the target user's ratings on neighboring items.
Neighborhoods are defined by similarities among users.	Neighborhoods are defined by similarities among items.

- *Model-based methods* use ML and data mining techniques such as decision trees, Bayesian methods and latent factor models. These models are trained on historical data and are used to predict items that customers are more likely to buy in the future.
- *Graph-based recommendation systems* provide recommendations using various network navigation algorithms like random walk, shortest path, etc. The ratings matrices can be represented in user-item graphs, user-user graphs and item-item graphs.

Let us briefly look at some terms before moving to understand how graph structures apply to recommendation systems. **Nodes** represent different items or products – for example, specific types of book or a particular SKU in retail industry. Two products which are purchased together, share a common link known as the **edge.** The **edge weight** represents the similarity of the products joined by the edge based on the categories of the product. For example, two fiction books share a strong edge while two different genre books share a weak edge.

Now, taking the example of an item-item graph, the first step in making recommendations is to get the product code and the metadata associated with the product. The node is then identified according to the product code (say, SKU). Once this is done, edges with lower edge weight are trimmed. Finally, nodes with remaining edges (and connected to the originally selected node) are arranged in the decreasing order of edge weights. The product codes associated with final list of nodes form the list of recommendations.



In the above item-item graph, the item is a book, and the thickness of edges shows the similarity between books. As we can see, Book 1 and Book 2 have the same author, genre and publisher - and that is why the edge weight between Book 1 and Book 2 is the highest (with a similarity value of 3). Book 1 and Book 6 have nothing in common, and so they do not have an edge between them.

Knowledge-based Recommendation Systems

Some items such as real estate, premium automobiles, expensive luxury goods, etc., do not have high demand and hence there is very minimal transactional data available. In such cases, recommendations can be provided based on the domain knowledge about these items. In case of cars, domain knowledge covers knowledge of engine variations, emission rules, manual or automatic gear type, cruise control, insurance policies, etc.

Approach	Conceptual goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + Community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + Item attributes
Knowledge-based	Give me recommendations based on my obvious (explicit) interests about the product (attributes).	User specifications + Item attributes + Domain knowledge
Hybrid	Give me recommendations based on combination of actions of my peers and my past actions.	User ratings + Community ratings + Item attributes

** Aggarwal, C. C. (2016)*

6.

Key Considerations for Implementing Recommendation Systems

Customers are the center of any business. Once we explore the business use-case, a feasible level of customer engagement (e.g., home page, product page, cart-page) should be identified before implementing a recommendation system.

The next important consideration is the identification of unique levels of products or a group of products – especially as businesses deal in multiple product levels. In the learning phase of recommendation systems, patterns that are relevant for the end customer should be extracted at the unique level of product category. For example, a laptop contains many hardware components (screen, processor, hard-drive, touchpad, keyboard etc.). All these elements can be identified uniquely the potential purchaser is interested in the final combination of these elements. Hence, recommendations should be provided at a full product level of the laptop, rather than at independent component levels.

Identifying seasonality effect in transaction data is important and can be done in two dimensions - one, the identification of seasonal products and the other, seasonality in the user's buying behavior. For example, winter clothing is a seasonal product useful in winter. On the other hand, some consumers opt to buy a lot in one part of the year, say, during holiday and sale season – this represents seasonality in purchase behavior.

Another factor to be considered is to experiment with an exhaustive list of feasible recommendation algorithms. While exploring historical transaction data, a feasible set of recommendation algorithms should be shortlisted, which will also be effective, reliable and applicable. For example, the frequent item-sets in market basket analysis can be explored by applying Apriori algorithm, feature pattern algorithm, etc.

Earlier, data scientists used to manually code algorithms which made the development cycle long and time-consuming. However, today with the amount and pace of contributions from the open-source community, we have the privilege of leveraging pre-built open-source packages and libraries that reduce development time drastically. One still needs to factor in the time to explore right packages and libraries for a specific use case.

With the exponential surge in variety of products and numbers of clients, huge volumes of incoming transactional data impact the performance of recommendation system in terms of response time and generalization of recommendations. Scalability of recommendation rules at timely intervals should therefore be factored in designing an implementation approach.

Constant learning and updating through a feedback loop are critical for the accuracy of recommendations. Measuring and monitoring system performance to analyze drift in data over time is an important activity to ensure that recommendations remain relevant and form the basis of identifying customer behavior patterns over time.

7.

Key Challenges in Implementing Effective Recommendation Systems

Sparsity of data is the main challenge. In general, most transactions do not contain all the items, and some items are present in only a few transactions. This makes it difficult to find some items in transactions, leading to slower extraction of information to build recommendation systems.

Imperfect labeling of either items or users leads to items being associated, but not grouped – giving rise to latent association. The item label here refers to the features such as product category, unique product codes, etc., and the user label refers to user features such as demographics, gender, age, educational qualification, etc. For example, when any of the two books belonging to same genre, is labeled incorrectly as a different genre, a latent relation or association is created between the two books.

Memory-based recommendation systems store data in matrices and calculate similarity scores between users and items to provide recommendations. They use user-item interaction, user features and item features matrices to do so. As the number of items and users increase over time, so does code complexity and it can become difficult to scale and maintain in real-time.

Recommendations can be presented to the users at different stages of engagement. The challenge is to understand the customer need at different phases of the user journey through all webpage engagements and provide relevant recommendations of cross-sell products, up-sell products, hybrid of cross-sell and upsell, sponsored products and newly added products.

8.
Conclusion

Well-designed recommendation systems are powerful customer engagement and experience tools and deliver the following business benefits:

- **High conversion rates:** A good recommendation system can drive better customer engagement, higher conversion figures and lower dropout rates. More visitors stay involved on websites and casual browsers are converted to customers.
- **Increased revenues:** The presence of relevant cross-sell and up-sell products result in higher customer conversion rates and enhanced revenue over time.
- **High customer satisfaction:** Building strong relevance into item searches, creating easily navigable content and comparative analysis between items lead to more engaged and satisfied customers.
- **Powerful personalization:** People believe in reviews and value recommendations of other users. Recommendation systems built on such parameters enable targeted and customized suggestions during a buyer's journey.
- **Discovery of new genius patterns:** Sometimes, recommendation systems help discover new genius patterns. For example, in hurricane seasons, it has been observed that people buy a lot of DVDs and just not essential items like milk and bread.
- **Improved user experience:** A good recommendation system discards useless information and provides only useful information to the user. Sometimes, it also helps users find new items. All this helps elevate user's experience in looking for the right items.
- **Better designed marketing campaigns:** Once slow-moving products are identified, targeted campaigns can be designed to identify users of interest and drive customer engagement.

In this paper, we have dived into memory-based methods, model-based methods, and graph-based methods of recommendation systems. In our study, we found that memory-based methods, though simple in approach, are difficult to scale and maintain in the production environment, as the data grows over time. Model-based methods are suitable for the production environment, but are complex to interpret by business. Graph-based methods effectively address both, scalability and interpretation. They have an added advantage of discovering hidden correlations and patterns through network navigation methods.

In the next part of this series, we will deep dive into graph-based recommendation systems to discuss the feasibility of implementation and its complexities and benefits.

9. References

- Aggarwal, C. C. (2016). Recommender systems (Vol. 1). Cham: Springer International Publishing.
- Fleder, D. M., & Hosanagar, K. (2007, June). Recommender systems and their impact on sales diversity. In Proceedings of the 8th ACM conference on Electronic commerce (pp. 192-199). diversity. ACM Conference on Electronic Commerce, pp. 192–199, 2007.
- Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. User modeling and user-adapted interaction, 11(3), 203-259.
- Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal, 16(3), 261-273.
- Pu, P., Chen, L., & Hu, R. (2011, October). A user-centric evaluation framework for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems (pp. 157-164).
- Recommendation Systems, Developers Course, Google. https://developers.google.com/machine-learning/recommendation
- https://github.com/agrawal-rohit/Graph-based-Recommendation-System
- https://www.nosto.com/blog/product-recommendations-examples/
- https://econsultancy.com/six-different-approaches-to-online-product-recommendations/
- https://en.wikipedia.org/wiki/Information_filtering_system
- 5 Advantages Recommendation Engines can Offer to Businesses | by Maruti Techlabs | Towards Data Science https://towardsdatascience.com/5-advantages-recommendation-engines-can-offer-to-businesses-10b663977673
- Graph-based real-time recommendation systems | by Matteo Cimini | Quantyca | Medium https://medium.com/quantyca/graph-based-real-time-recommendation-systems-8a6b3909b603
- Deep Learning for Collaborative Filtering (using FastAI) | by Matteo Cimini | Quantyca | Medium https://medium.com/quantyca/deep-learning-for-collaborative-filtering-using-fastai-b28e197ccd59

Authors

Vibha Bhagchandani

AVP – AI & Cognitive Solutions, Mphasis NEXT Labs

Vibha has 13+ years of experience in the technology industry and works as an Assistant VP - AI & Cognitive Solutions with Mphasis NEXT Labs. She is responsible for solutioning and building IP components as well as Proof of Concepts for client-focused cognitive projects. Vibha led the efforts for onboarding Mphasis proprietary ML algorithms on various cloud provider Marketplaces like AWS. She has been actively involved in the research, development and delivery of Natural Language Processing (NLP) focused product development initiatives for some of Mphasis' premier clients. She has done her B.E. in Computer Science and a Post-Graduate in Management from the Indian School of Business (ISB), Hyderabad.

Narayan Mishra

Senior Associate Business Analyst, Mphasis NEXT Labs

Narayan Mishra is a Senior Associate Business Analyst at Mphasis. He completed his Master of Technology from IIT Kanpur in Industrial & Management Engineering. He is proficient in Operations Research and Statistical Modeling. He has interest in various Machine Learning & Deep Learning areas like product recommendations, natural language processing and computer vision. He has also been actively involved in building production scale recommendation system for a large organization. He has keen interest in Quantum Computing and has been actively exploring the same.

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM} = 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. To know more, please visit www.mphasis.com

For more information, contact: marketinginfo.m@mphasis.com

USA
460 Park Avenue South
Suite #1101
New York, NY 10016, USA
Tel.: +1 212 686 6655

UK
1 Ropemaker Street, London
EC2Y 9HT, United Kingdom
T : +44 020 7153 1327

INDIA
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village
Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

www.mphasis.com

NR 22/01/21 US LETTER BASIL6633