User-Centric Product Discovery for Personalized e-Commerce Recommendations

Mustafa Keskin, Enis Teper and Sinan Keçeci
Hepsiburada, Turkey

Keywords: Recommender Systems, Personalization, Graphsage, node2vec, Approximate Nearest Neighbor Search, user2user Similarity.

Abstract: Personalized recommendations in e-commerce platforms often rely on user-item interactions or product similarity. In this work, we explore a user2user recommendation paradigm, where products are recommended based on purchases made by similar users. We investigate three methods for modeling user similarity: binary category vectors with sparse dot-product search, GraphSAGE embeddings trained on a user–product bipartite graph, and behavioral user embeddings obtained by averaging Node2Vec-based product vectors. Recommendations are drawn from complementary or previously browsed categories and ranked using recency-aware, diversity-promoting strategies. Offline experiments using HitRate@K demonstrate that graph and embedding-based methods significantly outperform the category-based baseline, effectively capturing latent user preferences and surfacing relevant, novel items.

1 INTRODUCTION

Modern e-commerce platforms host millions of products and users, creating both an opportunity and a challenge for delivering personalized shopping experiences. Recommender systems have become essential tools for guiding users through large product catalogs by suggesting items that align with their preferences. While item-based and content-based methods are commonly used in production systems, they often rely heavily on product metadata or the user's individual history, limiting their effectiveness in scenarios with sparse data or fast-changing inventories.

In this study, we implement a user2user recommendation pipeline designed specifically for e-commerce applications. The core idea is to identify users who exhibit similar purchase or browsing behaviors and recommend products that these similar users have bought, but which the user has not previously engaged with. Unlike traditional collaborative filtering approaches that focus on item-to-item relationships, our method emphasizes user-level similarity, making it particularly effective for surfacing products from the long tail or handling cold-start users with limited activity.

This approach is inspired by the intuition that "users like you" often discover items you might also be interested in. By leveraging behavioral signals across the user base, such as co-purchases, session data, or category preferences, the user2user method captures latent interests that are difficult to model with item features alone. It also enables the system to generate novel recommendations that are not just similar to what the user has already seen, but reflective of what like-minded users are exploring.

We evaluate the effectiveness of our method using both offline metrics (e.g., precision, recall, coverage) and online simulations. We further discuss implementation challenges in large-scale environments, such as maintaining real-time similarity graphs, handling popularity bias, and ensuring recommendation diversity.

2 RELATED WORKS

Personalized recommendation has been an important study area in information retrieval and recommender systems research, with a variety of approaches ranging from collaborative filtering to deep representation learning. In this section, we review relevant literature across three key dimensions aligned with our proposed framework: user-based collaborative filtering, graph-based recommendation models, and embedding-based user modeling.

Keskin, M., Teper, E. and Keçeci, S.
User-Centric Product Discovery for Personalized e-Commerce Recommendations.
DOI: 10.5220/0014287000004848

In *Proceedings of the 2nd International Conference on Advances in Electrical, Electronics, Energy, and Computer Sciences (ICEEECS 2025)*, pages 25-30
ISBN: 978-989-758-783-2

2.1 User-Based Collaborative Filtering

User-based collaborative filtering (UBCF) is one of the earliest and most intuitive approaches to recommendation, where users receive recommendations based on the preferences of similar users [Sarwar et al., 2001] (Sarwar et al., 2001). Traditional UBCF relies on computing similarity scores between users, typically via cosine similarity or Pearson correlation, based on their interactions (e.g., ratings, purchases, or clicks). Despite its simplicity, UBCF suffers from well-known limitations such as sparsity, cold-start issues, and scalability concerns in large datasets.

Several works have proposed enhancements to mitigate these problems. For example, item clustering [Linden et al., 2003], neighborhood pruning [Desrosiers & Karypis, 2011] (Desrosiers and Karypis, 2011), and hybrid models that combine user and item-based signals have shown promise. However, most UBCF approaches are shallow in representation and fail to capture the rich structural or temporal context of user behavior, especially in domains with high item churn like e-commerce.

2.2 Graph-Based Recommendation Models

Graphs have emerged as powerful structures for modeling interactions between users and items. Bipartite user–item graphs allow for the capture of both direct and higher-order co-occurrence patterns. Early methods such as random walks [Tong et al., 2006] (Tong et al., 2006) and label propagation have demonstrated success in capturing implicit feedback paths. More recently, the use of Graph Neural Networks (GNNs) has brought significant advances. Models such as GraphSAGE [Hamilton et al., 2017] (Hamilton et al., 2017), PinSage [Ying et al., 2018] (Ying et al., 2018), and LightGCN [He et al., 2020] (He et al., 2020) have been shown to learn high-quality user and item embeddings directly from the interaction graph.

In our work, we adopt GraphSAGE due to its ability to generalize to unseen nodes and inductively learn embeddings by aggregating information from a node's neighborhood. This approach is particularly well-suited for sparse or dynamic e-commerce graphs, where new users and products frequently enter the system.

2.3 Embedding-Based User Modeling

Learning low-dimensional embeddings to represent users and items has become a dominant strategy in modern recommender systems. These embeddings can be learned via matrix factorization [Koren et al., 2009] (Koren et al., 2009), sequence models (e.g., GRU4Rec [Hidasi et al., 2016] (Hidasi et al., 2016), SASRec [Kang & McAuley, 2018] (Kang and McAuley, 2018)), or graph-based methods like Node2Vec [Grover & Leskovec, 2016] (Grover and Leskovec, 2016). In e-commerce, item embeddings are often trained on user interaction sequences or product co-occurrence graphs, capturing substitute or complementary relationships between products.

Our approach builds on this by learning product embeddings via Node2Vec on a product–product co-occurrence graph constructed from user browsing and purchase sessions. These embeddings are then averaged per user to construct behavioral user vectors, allowing us to compute similarity in a latent intent space. Similar techniques have been explored in session-based recommendation [Quadrana et al., 2017] (Quadrana et al., 2017) and offline customer modeling [Grbovic & Cheng, 2018] (Grbovic and Cheng, 2018).

2.4 Diversity and Complementarity in Recommendations

Another key challenge in recommender systems is balancing relevance with diversity and novelty. Without explicit constraints, many systems tend to over-recommend popular or similar items, leading to user fatigue. Approaches such as result re-ranking [Ziegler et al., 2005] (Ziegler et al., 2005), determinantal point processes (DPP) [Kulesza & Taskar, 2012] (Kulesza and Taskar, 2012), and intent-aware diversification [Vargas & Castells, 2011] (Vargas and Castells, 2011) have been proposed to address this issue. Our round-robin sampling strategy across categories aims to increase inter-category diversity while preserving topical relevance by incorporating recent browsing context.

Finally, the concept of recommending from complementary categories, rather than just similar ones, has been underexplored, despite its relevance in domains like e-commerce where users often buy related products over time (e.g., phone $\rightarrow$ case $\rightarrow$ charger). Recent works in complementary item prediction [McAuley et al., 2015] (McAuley et al., 2015) and basket completion have laid the groundwork for this direction, which our method expands upon.

3 METHODOLOGY

We designed and implemented a user2user recommendation framework tailored for a large-scale e-commerce platform. The system is built upon behavioral data spanning the past 12 months, and it targets users with at least 10 distinct product purchases. This threshold ensures that the similarity estimation is grounded in meaningful behavioral patterns.

The goal is to retrieve the top-10 most similar users for a given target user, and recommend products that those similar users have purchased but the target user has not. To capture different aspects of user similarity, we developed and evaluated three distinct approaches: category-based sparse vectors, graph-based embeddings, and behavior-driven product vector aggregation.

3.1 Category-Based User Similarity

In this method, we represent each user as a high-dimensional binary vector indicating their interactions with product categories. Each vector dimension corresponds to a product category, where a value of 1 indicates at least one purchase in that category. Given the sparsity of the data, we utilize the sparse_dot_topn (Géron, 2018) library to efficiently compute approximate top-N cosine similarities. This allows us to retrieve the 10 most similar users for each target user at scale.

	T-shirt	Jacket	Laptop	TV	Carpet	...	Bicycle
User 1	1	0	0	0	0	..	0
User 2	0	0	1	1	1	...	1
User 3	1	1	1	0	0	...	0
User 4	0	1	0	0	0	...	0

Figure 1: User-Category Binary Matrix.

The Fig 1 illustrates a sparse binary matrix in Compressed Sparse Row (CSR) format, where rows corresponding to users and columns to product categories. A value of 1 indicates at least one purchase in that category, capturing user-category interactions compactly.

3.2 Graph-Based User Embeddings

To model deeper structural relationships between users and products, we construct a bipartite graph where nodes represent users and products, and edges denote purchase interactions (Kazemi et al., 2019). We then train a GraphSAGE model using a link prediction objective to learn node embeddings. GraphSAGE operates by iteratively aggregating and transforming feature information from a node's local neighborhood. For each node v at layer k, its embedding $\mathbf{h}_v^k$ is computed based on its own embedding from the previous layer $\mathbf{h}_v^{k-1}$ and the aggregated information from its neighbors $\mathcal{N}(v)$. A common approach, using the mean aggregator, can be expressed as:

$$\mathbf{h}_{\mathcal{N}(v)}^k = \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1} : u \in \mathcal{N}(v)\}) \quad (1)$$

Here, AGGREGATE$_k$ typically refers to an aggregation function (such as mean, sum, or max-pooling) applied to the embeddings of node v's neighbors from the previous layer $k-1$. For the mean aggregator, this would like:

$$\mathbf{h}_{\mathcal{N}(v)}^k = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{k-1} \quad (2)$$

Following aggregation, the node's embedding is updated by combining its own previous embedding with the aggregated neighbor information, typically followed by a non-linear activation:

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k) + \mathbf{b}^k\right) \quad (3)$$

In this update step, σ is a non-linear activation function, $\mathbf{W}^k$ is a learnable weight matrix, $\mathbf{b}^k$ is a learnable bias vector, and CONCAT represents the concatenation of the node's embedding and its aggregated neighborhood embedding. This iterative process allows each node's embedding to capture increasingly broader structural and feature information from its multi-hop neighborhood.

This enables us to encode not only direct user-product interactions but also higher-order co-purchase patterns. After training, we extract user embeddings and perform approximate nearest neighbor (ANN) search using the HNSW (Hierarchical Navigable Small World) (Malkov and Yashunin, 2016) algorithm to find the top 10 most similar users.

3.3 Behavioral Product Embeddings and Averaging from Orders

Our experiments also evaluate two production-grade user modeling baselines that generate user embeddings by averaging behavioral product vectors. In both methods, user embeddings are formed by averaging the embeddings of products found in their historical orders. The key difference lies in how the individual product embeddings are constructed: the V2V model (Attokurov et al., 2022) learns product embeddings from co-view patterns within the same session (i.e., "users who viewed this also viewed"), while

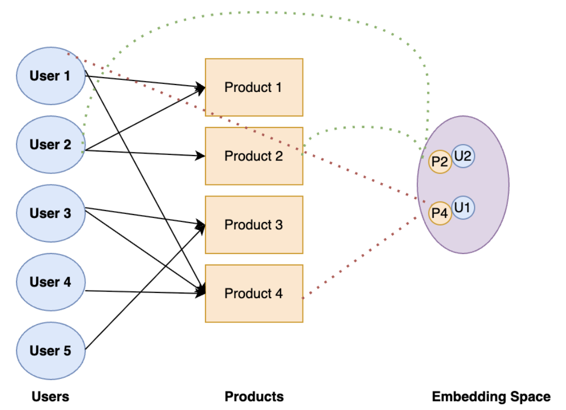

Figure 2: Overview of the methodology: Bipartite user-product interaction graph and its corresponding embedding space. On the left, users (User1, User2, etc.) are connected to products (Product1, Product2, etc.) they have purchased or interacted with. Through network embedding techniques GraphSAGE, the graph structure is transformed into a latent vector space (right), where proximities reflect behavioral or structural similarities. This representation enables user-to-user or item-to-item recommendation based on spatial closeness.

the FBT model (Keskin et al., 2024) trains on co-purchase graphs derived from same-day orders (i.e., "frequently bought together"). Both models utilize the Node2Vec algorithm over their respective product–product graphs. These embeddings serve as representative behavioral priors in our user2user similarity computations.

Figure 3: Getting average product embedding from user history

4 RECOMMENDATION GENERATION

Once the top-10 most similar users are identified for a given seed user using one of the methods described in the previous section, we proceed to generate personalized product recommendations by leveraging the behavioral footprints of those similar users. Our goal is not merely to recommend frequently purchased items, but to surface products that are novel, contextually relevant and diverse.

4.1 Extracting Candidate Categories

For each target user, we begin by aggregating the product categories purchased by the 10 nearest users. This aggregated set reflects the broader interest space of similar users. To ensure novelty, we filter out the categories in which the seed user has already made purchases, and instead focus on:

- Categories the user has previously browsed or interacted with, but not purchased
- And/or categories that are considered complementary to the user's previous purchases (e.g., if a user bought a camera, accessories like tripods or memory cards are deemed complementary).

This step ensures that recommendations explore adjacent areas of interest while maintaining contextual relevance.

4.2 Candidate Product Pool Construction

We enhanced the recently purchased products of users by incorporating substitute and complementary categories. This creates a large pool of candidate items. To avoid temporal bias and ensure relevance, we further rank these candidate products based on the recency of the seed user's browsing activity in each corresponding category. That is, products from categories the user has recently interacted with are prioritized higher.

4.3 Ranking and Diversification Strategy

To construct the final recommendation list, we adopt a round-robin selection strategy across categories to increase diversity and avoid item redundancy. Rather than recommending multiple products from the same category consecutively which often leads to diminishing returns in user engagement we iteratively sample one product from each relevant category in a rotating manner. This ensures that the resulting recommendation list is category-balanced and prevents overconcentration in any single interest area.

The final ranking thus reflects a careful balance between:

- Behavioral similarity (from user2user modeling)
- Recency-aware personalization
- And inter-category diversity

This approach is particularly effective in e-commerce platforms where users are often multi-

Table 1: Offline performance comparison of five user similarity methods on HitRate and NDCG at different cutoffs.

Method	HitRate			NDCG		
	@10	@20	@50	@10	@20	@50
SparseDotTopn	0.001041	0.001387	0.001635	0.000586	0.000677	0.000737
V2V	**0.002684**	**0.004016**	**0.004858**	**0.001380**	**0.001722**	**0.001908**
FBT	0.001174	0.001390	0.001408	0.000711	0.000774	0.000778
GraphSAGE (raw)	0.000839	0.001131	0.001319	0.000456	0.000532	0.000575
GraphSAGE (avg)	0.000656	0.000854	0.000994	0.000362	0.000413	0.000444

intent, and engagement improves with exposure to a variety of relevant options rather than a narrow focus.

5 EVALUATION

To evaluate the effectiveness of our user similarity methods in a real-world e-commerce setting, we conducted an offline evaluation using historical purchase data. For each eligible user, we held out the products purchased in the following 30 days as ground truth, simulating a next-period recommendation task. Our goal was to assess whether the users retrieved by our similarity methods would help uncover relevant products that the target user is likely to purchase in the near future.

We compared five methods for user similarity:

- **SparseDotTopn**: cosine similarity over binary category vectors.
- **V2V**: user embeddings averaged from product vectors trained on session-based co-view graphs.
- **FBT**: user embeddings averaged from product vectors trained on same-day co-purchase graphs.
- **GraphSAGE (raw)**: user embeddings directly obtained from a trained GraphSAGE model on the user–product bipartite graph.
- **GraphSAGE (avg)**: user embeddings obtained by averaging product embeddings from GraphSAGE.

Performance was measured using standard top-*k* ranking metrics: HitRate@k and nDCG@k (Normalized Discounted Cumulative Gain). HitRate@k checks whether at least one of the held-out items appears in the top-*k* recommended products, while nDCG@k takes into account the rank position of correct predictions, rewarding higher placements.

As shown in Table 1, the **v2v** method achieved the best overall performance across all metrics and cutoffs. This suggests that session-based co-view embeddings provide a strong behavioral signal for retrieving similar users. The **fbt** approach also performed well, outperforming GraphSAGE variants and SparseDotTopn in most cases, reflecting the effectiveness of co-purchase-based embeddings.

The **GraphSAGE (avg)** method, while conceptually simple, lagged behind v2v and fbt, potentially due to over-smoothing effects during aggregation. **GraphSAGE (raw)** produced more expressive representations but struggled with sparsity and noise in the bipartite graph. Finally, the **SparseDotTopn** method offered moderate performance, demonstrating that even lightweight sparse vector techniques can be competitive with proper category engineering.

These results highlight the trade-offs between simplicity, scalability, and expressiveness when designing user representation models for e-commerce recommendations.

6 CONCLUSION AND FUTURE WORKS

In this work, we proposed a scalable user-based recommendation framework for large-scale e-commerce platforms by implementing binary category interaction vectors and graph-based embedding techniques. Our method effectively identifies similar users through approximate top-N cosine similarity computations, enabling personalized and interpretable product recommendations. We demonstrated that simple, sparse interaction-based modeling (v2v) outperforms more complex graph neural approaches in cold-start and sparse settings, especially when relying solely on purchase data.

In future work, we plan to incorporate richer behavioral signals such as product clicks, add-to-cart events, and dwell time to construct more fine-grained user representations. By integrating these signals, we expect to capture earlier stages of user intent and extend the recommendation coverage to a broader user base including those without recent purchase history. Moreover, temporal modeling and causal inference approaches could help further refine recommendation quality and robustness.

ACKNOWLEDGEMENTS

This project was made possible by the individual contributions of each member of the recommendation team within Hepsiburada technology group. Also, this project would not have been possible if the technology group management of Hepsiburada had not supported and encouraged the data science team in innovation.

REFERENCES

Attokurov, U., Kaya, O., and Sezgin, M. S. (2022). Product recommendation based on embeddings: People who viewed this product also viewed these products. In *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 296–299. IEEE.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook*, pages 107–144.

Grbovic, M. and Cheng, H. (2018). Real-time personalization using embeddings for search ranking at airbnb. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 311–320.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864.

Géron, A. (2018). sparse-dot-topn: Efficient sparse matrix multiplication for top-n cosine similarity. https://github.com/ing-bank/sparse_dot_topn. Accessed: 2025-07-20.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1024–1034.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., and Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648.

Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kang, W.-C. and McAuley, J. (2018). Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 197–206.

Kazemi, S. M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., and Poupart, P. (2019). Relational representation learning for dynamic (knowledge) graphs: A survey. *CoRR*, abs/1905.11485.

Keskin, M., Teper, E., and Kurt, A. (2024). Comparative evaluation of word2vec and node2vec for frequently bought together recommendations in e-commerce. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 1–5. IEEE.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Kulesza, A. and Taskar, B. (2012). Determinantal point processes for machine learning. In *Foundations and Trends in Machine Learning*, volume 5, pages 123–286. Now Publishers Inc.

Malkov, Y. A. and Yashunin, D. A. (2016). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *CoRR*, abs/1603.09320.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Quadrana, M., Karatzoglou, A., Hidasi, B., and Cremonesi, P. (2017). Personalizing session-based recommendations with hierarchical recurrent neural networks. *CoRR*, abs/1706.04148.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW)*, pages 285–295. ACM.

Tong, H., Faloutsos, C., and Pan, J.-Y. (2006). Fast random walk with restart and its applications. *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM)*, pages 613–622.

Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM.