Application of Clustering Methods in Recommender Systems for User Behavior Analysis

Iryna Kyrychenko
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
iryna.kyrychenko@nure.ua
ORCID 0000-0002-7686-6439

Yehor Nesterenko
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yehor.nesterenko@nure.ua
ORCID 0009-0007-0263-1323

Anastasiya Chupryna
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
anastasiya.chupryna@nure.ua
ORCID 0000-0003-0394-9900

Loreta Savulioniene
Faculty of Electronics and Informatics
Vilniaus Kolegija
Vilnius, Lithuania
l.savulioniene@viko.lt

Paulius Sakalys
Faculty of Electronics and Informatics
Vilniaus Kolegija
Vilnius, Lithuania
p.sakalys@eif.viko.lt

***Abstract*—Recommender systems are crucial in personalizing digital experiences by predicting user preferences. This paper examines the application of the k-means clustering algorithm in recommender systems to segment users based on behavioural patterns, enhancing recommendation accuracy and efficiency. The study also compares k-means with other clustering techniques, analyzing their advantages and limitations in handling sparsity and the cold start problem. The results highlight the effectiveness of k-means for improving user segmentation and optimizing recommendation strategies.**

***Keywords*— collaborative filtering, k-means clustering, recommender systems, user segmentation.**

I. Introduction

Recommender systems have become integral to modern digital platforms, providing personalized content based on user preferences. These systems enhance user experience in e-commerce, media streaming, and online learning by predicting items of interest. Traditional recommendation techniques include collaborative and content-based filtering, which relies on user-item interactions. However, these methods face challenges such as data sparsity and the cold start problem, limiting their effectiveness.

To address these issues, clustering algorithms—particularly k-means—are widely used for user segmentation, allowing more precise recommendations by grouping similar users. This paper explores the application of k-means clustering in RS and compares it with other clustering techniques to assess their efficiency in improving recommendation accuracy.

II. Materials and Methods

Recommender systems are divided into content-based, collaborative filters [1], and knowledge-based, depending on the type of input data and methods used.

Content-based recommendations (sometimes called content-based filtering) are based on attribute vectors of objects created from text related to the objects, such as their description. In the case of books, the characteristics can be genre, topic, or author [2].

Knowledge-based methods suit stores with one-time purchases, such as camera sales. This approach uses technical attributes of objects and user preferences. Attributes often have weighting factors.

Collaborative filters look for similarities between users or objects but only analyze the behavioural archives of registered users. For example, similar users

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol2.8611

usually have the same items in their shopping carts, and the same customers purchase identical objects. Collaborative filters can be classified into model-based and memory-based methods. The first approach creates a model based on ratings, which is then used to generate recommendations. The other approach computes recommendations by searching for similar users or objects in all archived data [3].

Recommender systems face many challenges and problems. Most notably, collaborative filters are considered the most efficient and accurate approach.

A. Major problems of recommender systems

One of the significant challenges in recommender systems is the cold start problem [4]. When a new user interacts with the system for the first time, no historical data is available to generate personalized recommendations. Similarly, when new items are introduced, the system lacks sufficient user interactions to evaluate their relevance, making it challenging to suggest them effectively [5].

Another significant issue is data sparsity. Users often interact with only a small subset of available items, creating incomplete user-item matrices. This sparsity reduces the reliability of similarity-based recommendation techniques and can negatively impact the accuracy of personalized suggestions [6].

Additionally, scalability remains a critical concern, particularly for large-scale applications like news recommendations or e-commerce platforms. Recommender systems must process vast amounts of data and generate real-time recommendations, which requires efficient algorithms and optimized computational resources.

B. Clustering for User Segmentation

Clustering methods group users with similar preferences, enabling systems to recommend items based on cluster characteristics rather than individual interactions. This approach enhances the scalability of collaborative filtering while mitigating data sparsity [7].

Given a set of n users, let X be a matrix of user-item ratings where X = {x_1, x_2, …, x_n}. Clustering seeks to partition these users into k groups (C_1, C_2, …, C_k) such that users within the same cluster exhibit similar preferences.

A common clustering objective function is:

$$J = \sum_{i=1}^{k} \sum_{x_i \in C_i} \left| x_j - \mu_i \right|^2$$

where μ_i is the centroid of cluster C_i and $\|x_j - \mu_i\|^2$ denotes the squared Euclidean distance.

C. k-means Clustering in Recommender Systems

The k-means algorithm is widely used due to its simplicity and efficiency in handling large datasets. This algorithm divides the data into K clusters by minimizing the sum of the squares of the distances between the points and the centres of the clusters.

The algorithm follows these steps:

1) Initialize k cluster centroids randomly.

2) Assign each user x_j to the closest centroid μ_i:

$$d\,(x_i, \mu_j) = \sqrt{\sum_{f=1}^{m} (x_{jf} - \mu_{if})^2}$$

3) Update centroids μ_i as the mean of all users in the cluster:

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

4) Repeat steps 2–3 until convergence.

However, k-means has limitations:

- It requires predefining k, which is non-trivial.
- It assumes spherical clusters, making it unsuitable for complex data distributions.
- It is sensitive to initial centroid placement, affecting convergence quality.

In recommender systems, clustering helps to solve several important tasks:

- User segmentation. Using the k-means method, you can group users based on their preferences, such as movie ratings, genre preferences, etc. Clustering allows you to identify groups of users with similar preferences.
- Content segmentation. Similarly, movies or other items can be grouped by common characteristics, such as genre, rating, or other metadata.
- Improving the relevance of recommendations. After defining clusters, the system can recommend content popular in the user's cluster or items similar to those rated positively by other users in the cluster.

Clustering algorithms play a crucial role in recommender systems by segmenting users and identifying behavioural patterns, enhancing recommendation accuracy [8]. The effectiveness of these algorithms depends on factors such as scalability, data sparsity handling, computational complexity, and adaptability to dynamic user behaviour. This section provides a comparative analysis of various clustering algorithms, focusing on their advantages, limitations, and applicability in recommendation scenarios.

D. k-means Clustering

k-means is widely used in recommender systems due to its efficiency and simplicity. It partitions data points into a predefined number of clusters, where a centroid represents each cluster. The algorithm iteratively updates centroids based on the mean position of points assigned to each cluster. Its computational efficiency and scalability make it suitable for high-dimensional recommendation problems. However, k-means has notable drawbacks:

- Determining the optimal k can be challenging in dynamic environments.
- Initial centroid placement can significantly influence clustering results.
- k-means is best suited for clusters of similar size and shape, limiting its effectiveness with irregularly shaped clusters.

Despite these limitations, *k*-means is commonly applied to segment users based on preferences, providing a foundation for collaborative filtering and personalized recommendations.

E. K-Medoids Clustering

K-Medoids is a variant of k-means that selects actual data points as cluster centres (medoids) rather than computing centroids. This enhances robustness to outliers and noise. By representing clusters through existing user profiles, K-Medoids improve interpretability. However, it is computationally more intensive, especially with large datasets. Key advantages include:

- Less influenced by extreme values compared to k-means.
- Provides consistent clustering results, beneficial in dynamic user environments.

Due to their computational demands, K-Medoids are often applied to smaller datasets or scenarios where interpretability and robustness are prioritized.

F. Hierarchical Clustering

Hierarchical clustering creates a nested structure [9] of clusters, represented as a dendrogram, and operates in two modes:

- Agglomerative (Bottom-Up): Starts with individual points, merging them into larger clusters iteratively.
- Divisive (Top-Down): Begins with the entire dataset, recursively splitting it into smaller clusters.

This method does not require a predefined number of clusters and offers rich interpretability of relationships between users or items. However, it has significant limitations:

- Not suitable for large datasets due to intensive computation.
- Once clusters are formed, reassigning points is not straightforward.

Hierarchical clustering is valuable for exploratory data analysis and understanding the underlying structure of data, often serving as a preprocessing step in complex recommender systems.

G. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN identifies clusters as dense regions separated by areas of lower density. It does not require specifying the number of clusters beforehand and can detect clusters of arbitrary shapes. DBSCAN is particularly effective in handling noise and outliers. Its include:

- Detection of Arbitrary-Shaped Clusters: Capable of identifying clusters of various forms and densities.
- Effectively isolates outliers, preventing them from skewing the results.

However, DBSCAN's performance can decline with varying cluster densities and in high-dimensional spaces, which may limit its applicability in specific recommendation contexts.

H. Spectral Clustering

Spectral clustering utilizes graph theory and eigenvalue decomposition of similarity matrices to identify clusters. It constructs a similarity graph and partitions it based on the graph's spectral properties [4]. This method excels at detecting non-convex clusters and is helpful in complex relationship structure scenarios.

The primary drawback is its computational intensity, particularly with large datasets, due to the eigenvalue decomposition of large matrices.

k-means proves to be the most efficient and scalable clustering algorithm for recommender systems [10]. It balances computational speed, ease of implementation, and effective user segmentation. Unlike hierarchical clustering and DBSCAN, which struggle with scalability and high-dimensional data, k-means efficiently process large datasets while maintaining stable clusters. Despite requiring a predefined number of clusters, optimisation techniques can mitigate this limitation. Its simplicity and adaptability make k-means the preferred choice for improving recommendation accuracy and system performance.

TABLE 1 COMPARISON OF CLUSTERING METHODS

Algorithm	*Scalability*	*Handles Sparse Data*	*Cluster Shape*	*Noise Sensitivity*
k-means	High	Moderate	Spherical	Sensitive
K-Medoids	Medium	Moderate	Spherical	Robust
Hierarchical	Low	Moderate	Arbitrary	Moderate
DBSCAN	Medium	High	Arbitrary	Robust
Spectral Clustering	Low	High	Arbitrary	Moderate

III. RESULTS AND DISCUSSION

The MovieLens dataset is one of the most common datasets used to test and evaluate recommender systems. It contains information about user interactions with movies, including ratings given to movies by different users and metadata about the films, such as title, genre, and year of release. Thousands of users and films are represented in this dataset, and the ratings range from 1 to 5. The following steps can be used to cluster users by their preferences:

- Data preparation. Create a user-movie matrix where rows correspond to users, columns correspond to movies, and the value is the score.
- Clustering. Apply the k-means method to divide users into clusters. For example, users who like dramas can be assigned to one cluster, and those who prefer comedies to another [11].
- Recommendation generation. Based on the cluster profile, the system can recommend movies popular among users of that cluster.

A. Advantages and disadvantages of using the k-means method for recommender systems

The k-means clustering method has several advantages, including high speed and scalability, which allow it to work efficiently even with large data sets. The ease of implementation and the ability to adapt to different tasks make it popular in many areas. Moreover, grouping objects by common characteristics increases the accuracy and relevance of recommendations. At the same time, the method has its drawbacks: the need to determine the number of clusters (k) in advance, which can be difficult for heterogeneous data; sensitivity to the choice of initial cluster centres, which affects the results; and limitations in working with data that form clusters of arbitrary shape, as the algorithm best for spherical clusters.

IV. Conclusions

Among various clustering methods, k-means is the most efficient and scalable for recommender systems. It balances simplicity, speed, and adaptability, making it ideal for handling large datasets. While other methods, such as K-Medoids, Hierarchical Clustering, DBSCAN, and Spectral Clustering, have advantages in specific cases, k-means remains the most practical for user segmentation and recommendation accuracy.

Despite its need to predefine the number of clusters and sensitivity to initialization, k-means effectively mitigates data sparsity and the cold start problem. Future research can focus on optimizing k-means through advanced initialization techniques, adaptive clustering, or hybrid models to enhance recommendation performance.

References

[1] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, "A trust-based collaborative filtering algorithm for E-commerce recommendation system," Journal of Ambient Intelligence and Humanized Computing. [Online]. Available: https://doi.org/10.1007/s12652-018-0928-7.

[2] R. Z. Omarov, A. V. Vostrotina, and A. D. Li, "The cold start problem," Young Scientist, no. 26 (264), pp. 85-88, 2019.

[3] A. Saxena, M. Mittal, and L. M. Goyal, "Comparative Analysis of Clustering Methods," International Journal of Computer Applications, vol. 118, no. 21, pp. 30–35, May 2015. [Online]. Available: https://www.researchgate.net/publication/277907305_Comparative_Analysis_of_Clustering_Methods

[4] M. Koroteev, "Review of Clustering-Based Recommender Systems," arXiv preprint, arXiv:2109.12839, Sep. 2021. [Online]. Available: https://arxiv.org/abs/2109.12839

[5] N. Vara, M. Mirzabeigi, H. Sotudeh, and S. M. Fakhrahmad, "Application of k-means clustering algorithm to improve effectiveness of the results recommended by journal recommender system," Scientometrics, vol. 127, no. 3, pp. 1565–1583, May 2022. [Online]. Available: https://www.researchgate.net/publication/360773132_Application_of_k-means_clustering_algorithm_to_improve_effectiveness_of_the_results_recommended_by_journal_recommender_system

[6] A. Bansal, "Optimizing Customer Segmentation for Enhanced Recommendation Systems through Comparative Analysis of K-Means, Hierarchical Clustering, and DBSCAN Algorithms," International Journal of Core Engineering & Management, vol. 7, no. 6, pp. 12–18, 2023. [Online]. Available: https://www.researchgate.net/publication/384604526_Optimizing_Customer_Segmentation_For_Enhanced_Recommendation_Systems_Through_Comparative_Analysis_Of_K-_Means

[7] G.Proniuk, N. Geseleva, I. Kyrychenko, G. Tereshchenko, Predicate Clustering Method and its Application in the System of Artificial Intelligence, CEUR-WS, 2023, v. 3396, Volume II: Computational Linguistics Workshop, pp. 395-406. ISSN 16130073.

[8] I.Kyrychenko, O. Shyshlo, N. Shanidze, Minimizing Security Risks and Improving System Reliability in Blockchain Applications: a Testing Method Analysis, CEUR-WS, 2023, v. 3403, Volume III: Intelligent Systems Workshop, 2023.pp. 423–433. ISSN 16130073.

[9] K. Smelyakov, P. Dmitry, M. Vitalii and A. Chupryna, "Investigation of network infrastructure control parameters for effective intellectual analysis," 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2018, pp. 983-986, doi: 10.1109/TCSET.2018.8336359.

[10] B. Artley, "Unsupervised Learning: k-means Clustering," Towards Data Science, [Online]. Available: https://towardsdatascience.com/unsupervised-learning-k-means-clustering-2716b95af27. [Accessed: Feb. 2025].

[11] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, 2007, pp. 1027–1035.