Development of Skyline Query Algorithm for Individual Preference Recommendation in Streaming Data

by API

Submission date: 05-Dec-2024 02:01PM (UTC+0700)
Submission ID: 2426664887
File name: Ruhul_TB.docx (166.38K)
Word count: 5214
Character count: 28786

Journal of Applied Data Sciences
Vol. 6, No. 1, January 2025, pp. 76-88

ISSN 2579-7069

Development of Skyline Query Algorithm for Individual Preference Recommendation in Streaming Data

Ruhul Amin [a,b,1,*], Taufik Djatna [c,2], Annisa [a,3], Sukaesih Sitanggang [a,4]

[a] Department of Computer Science, Faculty of Mathematics and Natural Science, IPB University, Bogor 16680, Indonesia
[b] Department of Computer Science, Faculty of Technology Information, Universitas Nusa Mandiri, East Jakarta 13620, Indonesia
[c] Department of Agroindustrial Technology, Faculty of Agriculture Engineering and Technology, IPB University, Bogor 16680, Indonesia
[1] ruhulamin@apps.ipb.ac.id *; [2] taufikdjatna@apps.ipb.ac.id; [3] annisa@apps.ipb.ac.id; [3] imas.sitanggang@apps.ipb.ac.id
* corresponding author

(Received: December 9, 2024; Revised: December 31, 2024; Accepted: January 6, 2025; Available online: January 10, 2025)

Abstract

The ability of recommendation systems to provide relevant results is significantly influenced by their adaptation to users' dynamic individual preferences. Streaming data-based recommendation systems face substantial challenges in adjusting recommendations to the rapid changes in user preferences. Previous research on developing skyline query algorithms has focused on processing efficiency and parallel performance optimization. However, these studies have not considered the dynamic nature of individual user preferences, which is critical for generating relevant and responsive recommendations in streaming data environments. This study aims to develop a skyline query algorithm called Distributed Data Skyline (DDSky) to provide recommendations based on the dynamic individual preferences of users in streaming data. This study employs the Recency, Frequency, Monetary, and Rating (RFMRT) model to capture individual user preferences in real-time, which is then integrated with the skyline query algorithm to generate skyline objects in parallel within the streaming data environment. The DDSky algorithm was tested using a local PALITA application dataset and compared with the Eager algorithm. The results demonstrate that DDSky outperforms Eager, with an average recall of 0.45 and an F1-measure of 0.55, compared to Eager's recall of 0.33 and an F1-measure of 0.47. DDSky also achieved an average precision of 0.73, close to Eager's precision of 0.82. Furthermore, DDSky shows optimal throughput performance for data with up to 10,000 items and high flexibility across various data types.

Keywords: DDSky; Dynamic Individual Preferences; RFMRT model, Streaming data, System Recommendation

1. Introduction

In the big data era, the support of streaming data processing is essential [1] because it enables the continual and real-time processing of large data volumes [2], [3], particularly in applications such as recommendation systems. Streaming-based recommendation systems are influenced by a system's ability to process dynamic transaction histories, which reflect individual preferences that change over time [4]. Individual preferences refer to a person's desires or tendencies toward various products or services [5]. For example, in location-based recommendation systems, these changing preferences also encompass an individual's position during transactions. The ability of a recommendation system to adapt to users' dynamic individual preferences is crucial for providing accurate recommendations [6].

Recommendation system users require query operators to process data to find results that best match their preferences [7]. Relying on exact matches between preferences and data in a non-compliant database through the processing of query operators is unlikely to yield relevant recommendations. This problem is because no recommendation will meet the low price, good taste, high ratings, and proximity criteria. Skyline query are one method that allows the display of a set of data objects that align with user preferences, where no object is dominated by another data object [8]. An object is considered dominant if it does not have a worse value than another object in all dimensions and is at least better in one dimension [9]. In this context, data objects refer to specific products or services. Skyline query can provide recommendations involving multiple attributes [10]. However, this method has a limitation in that the recommendations produced cannot adjust to the dynamic individual preferences of users, meaning preferences may change based on location and evolving user preferences over time [11].

Previous studies [12] have proposed using Local Split Decision (LSD) Trees to speed up the computation of skyline queries on dynamic data. LSD Trees enable geometric structures to store and prune irrelevant data, thus accelerating the repeated processing of skyline queries. However, this approach is less flexible in adapting to the dynamic changes in individual preferences, particularly in real-time environments that demand quick responses to shifts in user preferences and experience performance degradation with high-dimensional data. Another study [13] successfully implemented an effective multicore-based parallel model for continuous skyline queries on high-dimensional data. However, this study did not consider the dynamic changes in individual user preferences. Without adapting to personal preferences, the resulting recommendations become less personalized and responsive to the changes in user preferences. Another study developed a Distributed Parallel Model (DPM) for skyline queries on uncertain data streams in a cloud environment with high scalability [14]. The DPM model improves scalability and load balancing, significantly reducing processing time. Although the DPM model is adequate for processing skyline queries on large datasets, this approach does not consider the dynamic nature of individual user preferences. The parallel DPM model focuses on optimizing parallel performance without mechanisms for updating or adjusting individual preferences within streaming data flows, thereby reducing the relevance of user recommendations.

Based on previous research, no approach has comprehensively addressed the calculation of dynamic individual preferences in streaming data, which can change in real-time according to a user's location or recent preferences. This study aimed to develop a skyline query algorithm for user preference recommendations that can generate recommendations based on dynamic individual user preferences. The algorithm that was developed is called Distributed Data Skyline (DDSky). The DDSky algorithm was designed to provide recommendations based on dynamic individual preferences. DDSky processes streaming data in real-time, allowing changes in user preferences to be captured and recommendations to be promptly adjusted.

Furthermore, DDSky uses streaming data processing and parallel computing technologies to improve the processing efficiency of large and dynamic datasets. By integrating rating indicators and historical transaction data, DDSky provided more accurate and relevant recommendations, thereby enhancing the quality of recommendations compared to previous methods that were less adaptive to dynamic individual preferences. The main contributions of this study are as follows:

- The development of the DDSky algorithm can provide recommendations based on dynamic individual user preferences in streaming data.
- The application of parallel computing enhances efficiency in processing large and dynamic streaming data.
- The dynamic individual preference model captures and analyzes changes in individual user preferences over time.

The remainder of this paper is organized as follows: Section 2 discusses the relevant literature on recommendation systems with skyline queries and skyline queries in streaming data, and Section 3 covers the materials and methods. In Section 4, we present the results and discuss them. Finally, Section 5 concludes with a summary of possible future research extensions.

2. Related Work

Research on skyline query has advanced rapidly, with various innovative approaches to improve computational efficiency, particularly for dynamic and large-scale data. Early studies [15] introduced the nearest neighbour (NN) search operation using an R*-tree structure [16]. This approach was extended using the branch-and-bound skyline (BBS) algorithm [17], which also introduced the concept of a dynamic skyline [18]. In a different approach, [19] utilized an M-tree structure to efficiently prune data, thereby accelerating the computation of the dynamic skyline query. With technological advancements, more recent research has focused on parallelization and implementation in distributed systems to address large-scale data challenges.

A previous study [12] introduced local split decision (LSD) trees to accelerate skyline query computations on dynamic data. This structure allows for the efficient storage and pruning of irrelevant data, thereby speeding up repeated query processing. However, this approach has limitations in adapting to dynamically changing individual preferences, particularly in real-time environments, which require rapid responses to user preference shifts. In addition, LSD Trees exhibit performance degradation when handling high-dimensional data. Another study successfully implemented a multicore-based parallel model for continuous skyline query on high-dimensional data [13]. Although effective in managing data complexity, this model does not account for dynamically changing individual user preferences, resulting in less personalized recommendations that are less responsive to changes in user preferences.

Another study [14] proposed a Distributed Parallel Model (DPM) developed for the skyline query on uncertain data streams in a cloud environment focusing on high scalability. This model improves performance by optimizing load balancing and significantly reducing processing time. However, this approach does not consider mechanisms for updating or adjusting individual user preferences within the streaming data flows, which can reduce the relevance of recommendations in the context of personalization. Although various studies have contributed significantly to improving the computational efficiency of skyline queries, no approach has specifically addressed the dynamic nature of individual user preferences. The lack of adaptation to changes in user preferences limits the effectiveness of recommendations, particularly in streaming data environments that require real-time adjustments. This study aims to fill this gap by developing an algorithm capable of dynamically adjusting individual user preferences to enhance the relevance and quality of the recommendations.

3. Methods

3.1. Development of the Individual Preference Recommendation Model

The individual preference recommendation model was designed to capture the dynamic preferences of users. This model utilizes the users' transaction history's recency, frequency, monetary, and rating (RFMRT) attributes. The developed RFMRT model can process data in real-time (streaming), enabling it to account for changes in individual user preferences over time. The notations used in the individual preference recommendation model are listed in Table 1.

Table 1. Notations of the Individual Preference Recommendation Model (Algorithm 1)

No	Notation	Definition
1	u	*User* of the recommendation system, where $u = 1, 2, 3, \ldots, n$; n n is the total number of *user*.
2	I	Product (*items*) recommended to the *user*, where $I = 1, 2, 3, \ldots, m$; m is the total number of products (*items*).
3	T	A series containing dates in the recommendation system, from the start of the transaction (t_1) to the most recent transaction (t_{new}). In this case, $T = t_1, t_2, t_3, \ldots, t, \ldots, t_{new}$.
4	$TB_{u,I,t}$	A time series corresponding to the time (HH:MM:SS) of item I purchased by user u at times $TB = tb_1, tb_2, tb_3, \ldots, tb, \ldots, tb_{new}$
5	$f_{u,I,t,tb}$	*Frequency* of *user u* puchasing an item I on date t at time tb
6	$F_{u,I}$	The cumulative current frequency of *user u* purchasing item I. This represents the sum of frequencies from the start of transaction up to one transaction before the most recent transaction. 'Current' refers to the event on the current date (t_{new}) and at the current time tb_{new}.
7	$R_{u,i}$	The difference in days between the current date in the system t_{new} and the last transaction date made by *user u* for item I; $R_{u,i}$ is a non-negative integer. In practice, a *user u* may make more than one transaction at t_{new} (transactions occurring on the same day but at different times) for the same item I. In such cases, the value of $R_{u,i}$ is set to 0 ($R_{u,i} = 0$).
8	$M_{u,I}$	The current condition for the accumulation of money spent by *user u* on item I.
9	$RT_{u,I}$	The rating given by *user u* to item I, recorded in the recommendation system at the time of the last transaction. The rating value can range from 1 to 5, where a higher number indicates a better condition.
10	$P_{u,I}$	The current preference value of *user u* for item I.

Dynamic individual user preferences can be formulated using Equation 1 :

$$P_{u,I,new} = (F_{u,I,new} + M_{u,I,new} + RT_{u,I}) - R_{u,i} \quad (1)$$

Algorithm for Individual User Preferences

Algorithm 1 was developed to calculate the current preference value of *user* u for item I.

Algorithm 1: Individual User Preference (*user*)

Input : $R_{u,i}$, $F_{u,I}$, $M_{u,I}$, $RT_{u,I}$

Output : $P_{u,I}$, max ($P_{u,I}$)

Proses :

```
Begin
For u = 1 to n
    For i = 1 to m do
        P_{u,I} = (F_{u,I} + M_{u,I} + RT_{u,I}) − R_{u,i}
    End for
    Max(P_{u,I})
End For
Write ('P_{u,I}'), Max('P_{u,I}')
End
```

The current preference value of the user u for item I can be calculated using Algorithm 1. This algorithm takes inputs from the attributes $R_{u,i}$, $F_{u,I}$, $M_{u,I}$, $RT_{u,I}$. The output of this algorithm is the current preference value of the *user* u for item I ($P_{u,I}$). The process begins by setting the value of u from 1 to n, where n is the number of users for whom the individual preferences are calculated. Next, the preference calculation for each user is performed for each item, considering the various factors that influence the preferences. Then, the preference for each item is calculated by summing the values of the attributes $F_{u,I,new}$, $M_{u,I,new}$, $RT_{u,I}$. The result of this sum is subtracted from the value of the attribute $R_{u,i}$. Finally, the algorithm searches for the maximum value $P_{u,I,new}$ among all items to determine which item has the highest value.

3.2. Development of the Dynamic Individual User Preference-Based Skyline Query Algorithm

This study develops a Distributed Data Skyline (DDSky) algorithm to generate recommendations based on the dynamic individual preferences of users. The algorithm was designed to address several shortcomings in previous research, particularly in handling static data and failing to account for changes in user habits and preferences over time. By leveraging streaming data processing, this new algorithm can provide more accurate recommendations that adapt to rapid changes in preferences, unrestricted by the time or location of the user (spatial-temporal issue).

Algorithm 2. Distributed Data Skyline (DDSky)

Input : User transaction history

Output : Individual preference recommendation

Proses :

1 Read user transaction history
2 Obtain individual user preferences
3 Perform similarity calculation
4 Execute parallel skyline query
5 Is there new data? If:
   - Yes: Proceed to step 6
   - No: Proceed to step 8
6 Use new data to update individual user preferences
7 Display recommendations based on individual user preferences
8 Repeat the process starting from step 1

As shown in Algorithm 2, it receives the input data from the user's transaction history. The user's transaction history is then analyzed to determine the individual preferences using Algorithm 2. Individual user preferences were subsequently compared for similarity with items near the user's location. The result of this similarity calculation will serve as a preference attribute processed alongside the attributes of price and distance in the skyline query. These attributes—price, preference, and distance—are then processed in parallel to obtain skyline objects using Algorithm 2 on the streaming data. Skyline processing continues with each new data input to generate recommendations based on the preferences of the individual users.

Based on Algorithm 2, the skyline query processing is performed in parallel on streaming data. This process generates dynamic individual user preference recommendations more efficiently. Algorithm 3 is an approach for executing the skyline query process in parallel based on dynamic individual user preferences, inspired by the DPM model proposed in the study [20]. The notations used in Algorithm 2 are listed in Table 2.

Table 2. Notations of the Individual Preference Recommendation Model (Algorithm 1)

Notation	Definition
D_s	*Data streams*
W_i	Local sliding window, where $i = 1,2,3,\dots,n;$ n is the total number of local sliding windows.
e_{new}	The newly arrived streaming object in W
e_{old}	The expired streaming object in W
S_i	Local skyline, where S_i is the result of the skyline computation from the local sliding window W_i
S_{global}	Global skyline, where S_{global} is the result of the global skyline obtained from the combination of all local skylines S_i

Algorithm 2 operates by distributing data streams (D_s) to each local sliding window (W_i), where i ranges from 1 to n. Each local sliding window (W_i) manages the maximum data size in a single (W_i) which is equal to the total number of local sliding windows used *(n)*. If the amount of data exceeds the maximum allowed in a local sliding window (W_i), the oldest object is removed when new data enters (e_{new}), following the first-in, first-out (FIFO) principle. When each partition (P_i) contains the maximum amount of data, each node $Pi(1 \leq i \leq n)$ performs parallel computation to find the skyline objects. Eeach skyline object from W_1 to W_n is then merged to obtain the global skyline object. The output of Algorithm 3 is the global skyline object, which is the result of the a combination of W_1 and W_n.

Algorithm 3: DDSky with Parallel Processing

Input : $D_s = \{e_0, e_1, e_2, e_3, e_4, e_5, \dots\}$

Output : S_{global}

Proses :

```
Begin
For Each sliding window W_i (1 ≤ i ≤ n) initialize empty;
   t=0;
   While there comes a new item e_new to W_t Do
      t=(t+1)%n
      W_t receives e_new;
      if size of W_t equals (n+1) Then
         //when W_t is full, remove the e_old
         remove e_old from W_t;
      Endif
      add e_new to W_t;
      True;
      for i=1 to n Do
         if size of W_i is less than (n-1) then
            windowFull = False;
            break;
         Endif
      Endfor
      if windowFull then
         for Each sliding window W_i (1 ≤ i ≤ n)
            skyline_query(Pc, Dt, Pr)
         Endfor
      Endif
   Endwhile
```

```
        prelimanary_global_skyline_set = union of all skyline in W_i;
        global_skyline_set=[];
        for Each item e in prelimanary_global_skyline_set Do
            if not any item O in prelimanary_global_skyline_set dominates e then
                add e global_skyline_set;
        Endfor
    Endfor
    End
```

4. Results and Discussion

In this study, the evaluation was divided into two scenarios. The first evaluation aimed to measure the accuracy of the DDSky algorithm using precision, recall, and F1 measure metrics. The second evaluation focused on the computation time required by the DDSky algorithm to generate skyline objects. The results of the accuracy and computation time produced by the DDSky algorithm were compared with those of other algorithms.

4.1. Dataset

This study uses a dataset of local culinary profiles, local culinary vendor profiles, and user transaction histories sourced from the JALITA (Jajanan Asli Nusantara Pintar) application. JALITA is a mobile-based application for a local culinary recommendation system based on individual user preferences. Another dataset consists of individual user preferences from a questionnaire distributed to users. The JALITA application is a mobile-based recommendation system for Indonesian local cuisines based on user preferences.

4.2. Accuracy Evaluation of the DDSky Algorithm

Accuracy evaluation uses recall, precision, and F1 metrics, which have been applied previously in the research [21]. The definitions and equations used for each of the metrics are as follows:

a. Precision was defined as the percentage of items recommended by the user. Precision measures how well a system recommends relevant and liked items. The precision calculation method for precision is given in Equation (2).

$$\boldsymbol{precision} = \frac{|prefered \cap recommended|}{recommended} \qquad (2)$$

b. Recall is defined as the percentage of liked items that are recommended. Recall measures how well a system recommends items that the user truly likes. The calculation method for recall is given by Equation (3).

$$\boldsymbol{recall} = \frac{|prefered \cap recommended|}{prefered} \qquad (3)$$

c. F1-measure is a balanced combination of precision and recall. The F1-measure combines the precision and recall metrics into a single value that provides an overall view of the algorithm's performance of the algorithm. The calculation method for the F1-measure is given by Equation (4).

$$F1 = \frac{2*recall*precision}{recall+precision} \qquad (4)$$

Where preferred are items liked by the user, and recommended are the set of skyline objects produced by the algorithm.

In the accuracy evaluation, this study compared the DDSky algorithm with the Eager parallel processing algorithm [13]. Both algorithms share fundamental similarities, as they are designed to process skyline queries in streaming data environments, requiring time efficiency and adaptability to real-time data changes. DDSky and Eager implement parallel processing techniques to handle large-scale streaming data, enabling both to achieve high performance in dynamic data scenarios. Additionally, both algorithms utilize a sliding window model to limit the data processed within specific intervals, ensuring that only relevant data is considered. The output of both algorithms is a skyline set comprising optimal objects based on user preferences.

The testing was performed by comparing the recommendation outputs of both algorithms against test data reflecting actual user preferences in dynamic streaming data scenarios. Identical datasets ensured evaluation parity, with normalized attributes to standardize value scales. The accuracy results for the DDSky algorithm are presented in Table 3, demonstrating its superiority in capturing dynamic user preferences. Conversely, Table 4 provides the evaluation results for the Eager algorithm, which excels in processing efficiency but is less responsive to changes in user preferences. This comparison offers a comprehensive overview of the strengths and weaknesses of each algorithm in a streaming data environment.

Table 3. Accuracy Evaluation Results for the DDSky Algorithm

UserID	intersection	prefered	recommended	precision	recall	F1
4	5	53	6	0,83	0,09	0,17
5	5	40	6	0,83	0,13	0,22
6	5	30	7	0,71	0,17	0,27
8	6	42	6	1,00	0,14	0,25
9	3	32	5	0,60	0,09	0,16
10	4	11	4	1,00	0,36	0,53
11	4	47	6	0,67	0,09	0,15
12	4	36	7	0,57	0,11	0,19
13	4	56	5	0,80	0,07	0,13
16	6	48	7	0,86	0,13	0,22
17	5	46	5	1,00	0,11	0,20
18	7	58	7	1,00	0,12	0,22
19	7	58	9	0,78	0,12	0,21
20	4	34	7	0,57	0,12	0,20
21	3	43	7	0,43	0,07	0,12
22	3	52	5	0,60	0,06	0,11
23	5	64	6	0,83	0,08	0,14
24	3	30	4	0,75	0,10	0,18
25	5	64	5	1,00	0,08	0,14
28	5	56	7	0,71	0,09	0,16
32	3	53	4	0,75	0,06	0,11
33	4	23	7	0,57	0,17	0,27
34	4	42	5	0,80	0,10	0,17
40	4	58	5	0,80	0,07	0,13
41	5	37	9	0,56	0,14	0,22
43	7	46	8	0,88	0,15	0,26
45	6	48	9	0,67	0,13	0,21
46	4	31	9	0,44	0,13	0,20
47	4	47	7	0,57	0,09	0,15
48	1	21	4	0,25	0,05	0,08
		average		0,73	0,11	0,19

Table 4. Evaluation of the accuracy results of the Eager algorithm

UserID	intersection	prefered	recommended	precision	recall	F1
4	4	53	4	1,00	0,08	0,14
5	4	40	4	1,00	0,10	0,18
6	3	30	4	0,75	0,10	0,18

UserID	intersection	prefered	recommended	precision	recall	F1
8	4	42	4	1,00	0,10	0,17
9	4	32	4	1,00	0,13	0,22
10	3	11	4	0,75	0,27	0,40
11	3	47	4	0,75	0,06	0,12
12	2	36	4	0,50	0,06	0,10
13	4	56	4	1,00	0,07	0,13
16	4	48	4	1,00	0,08	0,15
17	4	46	4	1,00	0,09	0,16
18	4	58	4	1,00	0,07	0,13
19	4	58	4	1,00	0,07	0,13
20	3	34	4	0,75	0,09	0,16
21	2	43	4	0,50	0,05	0,09
22	3	52	4	0,75	0,06	0,11
23	4	64	4	1,00	0,06	0,12
24	3	30	4	0,75	0,10	0,18
25	4	64	4	1,00	0,06	0,12
28	4	56	4	1,00	0,07	0,13
32	3	53	4	0,75	0,06	0,11
33	2	23	4	0,50	0,09	0,15
34	3	42	4	0,75	0,07	0,13
40	4	58	4	1,00	0,07	0,13
41	3	37	4	0,75	0,08	0,15
43	3	46	4	0,75	0,07	0,12
45	4	48	4	1,00	0,08	0,15
46	3	31	4	0,75	0,10	0,17
47	2	47	4	0,50	0,04	0,08
48	1	21	4	0,25	0,05	0,08
		average		0,82	0,08	0,15

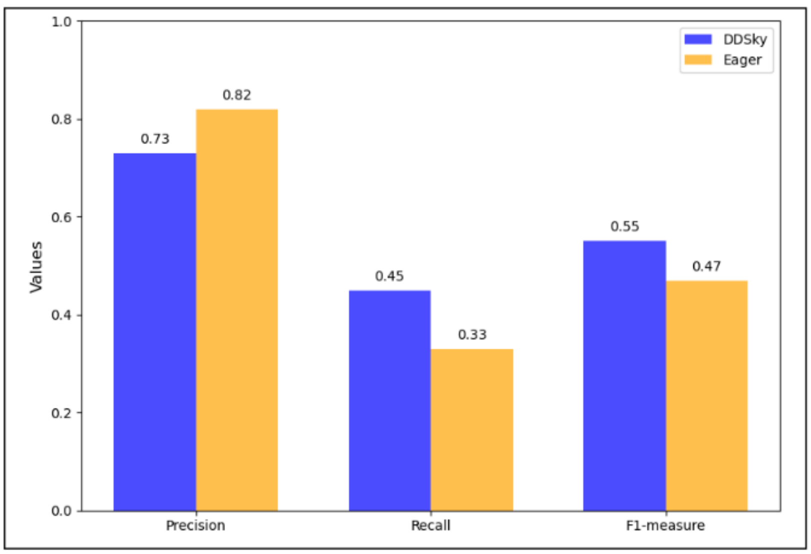

Fig. 1. Comparative analysis of performance metrics between the DDSky and Eager algorithms.

The results shown in Figure 1 indicate that DDSky achieves a precision of 0.73, a recall of 0.45, and an F1-measure of 0.55. In contrast, the Eager algorithm, which emphasizes distributed and parallel processing without considering individual user preferences [13], demonstrates a precision of 0.82, a recall of 0.33, and an F1-measure of 0.47. Based on the accuracy evaluation, while DDSky has a slightly lower precision, it outperforms in capturing dynamic user preferences, as evidenced by its higher recall and F1-measure compared to Eager. Conversely, although Eager has higher precision, its lower recall indicates limitations in adapting to dynamic individual preferences.

One reason for DDSky's lower precision is its design, which adjusts recommendations in response to evolving user preferences over time and location. This adaptability can introduce more significant variability in the recommendation set, occasionally including less relevant items, thereby reducing precision. Overall, in the context of recommendation accuracy, DDSky is better suited to generate recommendations that align with dynamically changing individual user preferences.

4.3. Throughput Evaluation

The second scenario in this evaluation was the throughput testing of the DDSky algorithm. We used a synthetic dataset of various types to test throughput, including independent, correlated, and anti-correlated data. These different dataset types were used to assess the performance of the DDSky algorithm under various data conditions. Throughput is measured to determine how efficiently DDSky processes large volumes of data with diverse types while ensuring that the improvement in recommendation quality does not come at the cost of the algorithm's performance in terms of processing speed. The throughput evaluation results for the DDSky algorithm are shown in Figure 2.

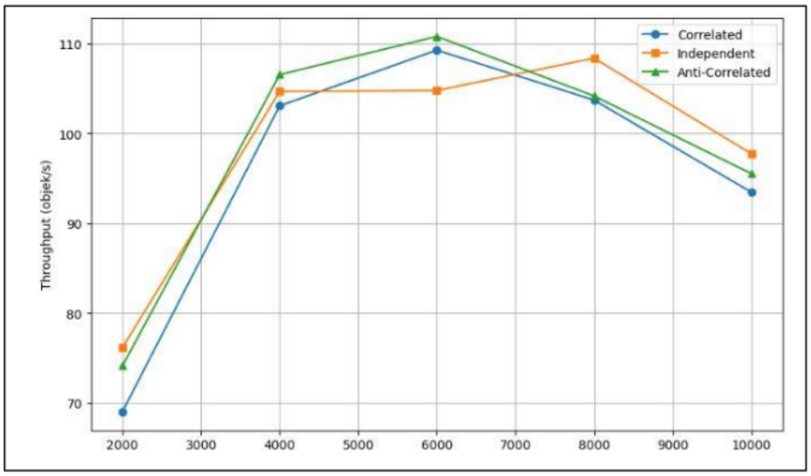

Fig. 2. Throughput Evaluation Results for DDSky

The throughput evaluation results show that the DDSky algorithm achieves optimal performance with approximately 6000 data points for both correlated and anti-correlated data, where the highest throughput was achieved. However, the throughput continues to increase for independent data until it peaks at approximately 8000 data points. The performance of the DDSky algorithm varies depending on the data type being processed. For correlated data, throughput increases sharply up to 6000 data points, then decreases. Anti-correlated data shows a similar pattern but with slightly higher throughput.

Meanwhile, independent data showed a more gradual increase in throughput, peaking at a higher data volume. Despite the varying performance, DDSky can process large volumes of data (up to 10.000) for all data types, with the throughput remaining above 90 objects/s even at the highest data volume. This result indicates the algorithm's flexibility and robustness in handling different conditions and data volumes.

5. Conclusion

This study has developed the DDSky algorithm, designed to provide recommendations based on the dynamic individual preferences of users of streaming data. The results show that DDSky outperforms the Eager algorithm, with an average recall of 0.11 and an F1 measure of 0.19, surpassing Eager, which has a recall of 0.08 and an F1 measure of 0.15. This result indicates that DDSky generates accurate and relevant recommendations more effectively. Additionally, the study successfully developed the RFMRT model, which can identify individual user preferences using transaction history and user ratings data.

The main contribution of this study is the development of a skyline query algorithm that can adapt to dynamic user preferences in the context of streaming data. However, this study has some limitations. Its focus is limited to the Indonesian local culinary recommendation system; therefore, the application of DDSky in other domains, such as e-commerce, social media, and streaming services, has not yet been explored. Furthermore, the developed individual preference recommendation model has not been integrated with machine learning technologies, which could potentially enhance the effectiveness and efficiency of the recommendations.

Therefore, future research is recommended to test the application of the DDSky algorithm in various other domains to evaluate its flexibility and to integrate the individual preference model with machine learning to improve the prediction capabilities and adaptability to changes in user preferences. Evaluations in more complex real-time scenarios are also suggested to ensure the algorithm's effectiveness under real-world conditions. Further studies should also focus on improving the scalability and performance of the algorithm in handling huge data volumes by developing additional optimization techniques and more advanced data processing strategies. Thus, the recommendations generated will be more relevant and responsive to real-time changes in user preferences.

Acknowledgment

Thanks to Universitas Nusa Mandiri for supporting this study through decree no. 1230/3.07/STMIK-NM/IX/2019, and a research team in Computational Intelligence and Optimization (CIO) and industrial and Systems Engineering (ISE) at IPB University.

References

[1] T. Akidau, S. Chernyak, and R. Lax, Streaming systems: the what, where, when, and how of large-scale data processing. O'Reilly Media, Inc, 2018.

[2] H. Kumar, P. J. Soh, and M. A. Ismail, "Big Data Streaming Platforms: A Review," Iraqi journal for computer science and mathematics, vol. 3, no. 2, pp. 95–100, Apr. 2022, doi: 10.52866/IJCSM.2022.02.01.010.

[3] A. Lemzin, "Streaming Data Processing," Asian Journal of Research in Computer Science, vol. 15, no. 1, pp. 11–21, Jan. 2023, doi: 10.9734/AJRCOS/2023/V15I1311.

[4] T. Y. Qian, B. Liu, L. Hong, and Z. N. You, "Time and Location Aware Points of Interest Recommendation in Location-Based Social Networks," J Comput Sci Technol, vol. 33, no. 6, pp. 1219–1230, 2018, doi: 10.1007/s11390-018-1883-7.

[5] E. R. Igou, F. van Dongen, and W. A. P. van Tilburg, "Preference Judgments (Individuals)," Encyclopedia of Human Behavior, pp. 153–159, 2012.

[6] A. N. Noubari and W. Wörndl, Dynamic Adaptation of User Preferences and Results in a Destination Recommender System, (2023).

[7] Y. Gulzar, A. A. Alwan, N. Salleh, and I. F. Al Shaikhli, "Processing skyline queries in incomplete database: Issues, challenges and future trends," 2017. doi: 10.3844/jcssp.2017.647.658.

[8] Annisa, A. Zaman, and Y. Morimoto, "Area Skyline Query for Selecting Good Locations in a Map," Information Processing Vol.24, vol. 24, no. 6, pp. 946–955, 2016, doi: 10.2197/ipsjjip.24.946.

[9] S. Borzsonyil, D. Kossmann, and K. Stocker, "The Skyline Operator," in Proceedings of the 17th International Conference on Data Engineering (ICDE), 2001, pp. 421–430.

[10] Y. Shu, J. Zhang, W. E. Zhang, D. Zuo, and Q. Z. Sheng, "IQSrec: An Efficient and Diversified Skyline Services Recommendation on Incomplete QoS," IEEE Trans Serv Comput, vol. 16, no. 3, pp. 1934–1948, May 2023, doi: 10.1109/TSC.2022.3189503.

[11] Q. Pu, A. Lbath, and D. He, "Location Based Recommendation for Mobile Users Using Language Model and Skyline Query," International Journal of Information Technology and Computer Science, vol. 4, no. 10, pp. 19–28, 2012, doi: 10.5815/ijitcs.2012.10.02.

[12] D. Köppl, "Dynamic Skyline Computation with LSD Trees," Analytics, vol. 2, no. 1, pp. 146–162, 2023, doi: 10.3390/analytics2010009.

[13] T. De Matteis, S. Di Girolamo, and G. Mencagli, "A multicore parallelization of continuous skyline queries on data streams," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9233, pp. 402–413, 2015, doi: 10.1007/978-3-662-48096-0_31.

[14] X. Li, Y. Wang, X. Li, and Y. Wang, "Parallel skyline queries over uncertain data streams in cloud computing environments," International Journal of Web and Grid Services, vol. 10, no. 1, pp. 24–53, 2014, doi: 10.1504/IJWGS.2014.058759.

[15] D. Kossmann, F. Ramsak, and S. Rost, "Shooting Stars in the Sky: An Online Algorithm for Skyline Queries," VLDB '02: Proceedings of the 28th International Conference on Very Large Databases, pp. 275–286, Jan. 2002, doi: 10.1016/B978-155860869-6/50032-9.

[16] H. P. Kriegel, B. Seeger, R. Schneider, and N. Beckmann, "The R-tree: an efficient and robust access method for points and rectangles," ACM, pp. 448–455, 1990, doi: 10.1145/93597.98741.

[17] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An Optimal and Progressive Algorithm for Skyline Queries," Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 467–478, 2003, doi: 10.1145/872811.872814.

[18] M. L. Lo, K. Tang, P. S. Yu, and M. L. Yiu, "Progressive skyline computation in database systems," ACM Transactions on Database Systems (TODS), vol. 31, no. 4, pp. 1256–1274, 2006.

[19] L. Chen and X. Lian, "Dynamic skyline queries in metric spaces," Advances in Database Technology - EDBT 2008 - 11th International Conference on Extending Database Technology, Proceedings, pp. 333–343, 2008, doi: 10.1145/1353343.1353386.

[20] X. Li, Y. Wang, X. Li, Y. Wang, and R. Huang, "Parallelizing probabilistic streaming skyline operator in cloud computing environments," Proceedings - International Computer Software and Applications Conference, pp. 84–89, 2013, doi: 10.1109/COMPSAC.2013.15.

[21] F. Rhimi, S. B. Yahia, and S. B. Ahmed, "Enhancing Skyline Computation With Collaborative Filtering Techniques for QoS-Based Web Services Selection," Proceedings - 14th International Symposium on Network Computing and Applications, pp. 247–250, 2015, doi: 10.1109/NCA.2015.18.

Development of Skyline Query Algorithm for Individual Preference Recommendation in Streaming Data

ORIGINALITY REPORT

9%	6%	6%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.nusamandiri.ac.id Internet Source	1%
2	scholarworks.boisestate.edu Internet Source	1%
3	Ren Jing-xia, Wu Zhi-feng. "Collaborative Filtering Algorithm Based on Dynamic Trust Attenuation", Proceedings of the 2020 3rd International Conference on Big Data Technologies, 2020 Publication	1%
4	cdn.techscience.cn Internet Source	1%
5	Walid Khames, Allel Hadjali, Mohand Lagha. "Parallel continuous skyline query over high-dimensional data stream windows", Distributed and Parallel Databases, 2024 Publication	<1%
6	www-emerald-com-443.webvpn.sxu.edu.cn Internet Source	<1%

7	www2.mdpi.com Internet Source	<1 %
8	Fatma Rhimi, Saloua Ben Yahia, Samir Ben Ahmed. "Enhancing Skyline Computation with Collaborative Filtering Techniques for QoS-Based Web Services Selection", 2015 IEEE 14th International Symposium on Network Computing and Applications, 2015 Publication	<1 %
9	link.springer.com Internet Source	<1 %
10	"Collaborative Computing: Networking, Applications and Worksharing", Springer Science and Business Media LLC, 2024 Publication	<1 %
11	Li, Xiaoyong, Yijie Wang, Xiaoling Li, Yuan Wang, and Rubing Huang. "Parallelizing Probabilistic Streaming Skyline Operator in Cloud Computing Environments", 2013 IEEE 37th Annual Computer Software and Applications Conference, 2013. Publication	<1 %
12	onlinelibrary.wiley.com Internet Source	<1 %
13	Submitted to Universitas Pendidikan Indonesia Student Paper	<1 %

14	Submitted to Nottingham Trent University Student Paper	<1 %
15	Sirojul Munir, Imam Haromain, Riyan Wahyudi, Misna Asqia, Ridwan Raafi'udin. "Wikuliner - Regional Culinary Recommendation System Based on The Web Using Extreme Programming Method", 2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS, 2021 Publication	<1 %
16	dbgroup.eecs.umich.edu Internet Source	<1 %
17	ebin.pub Internet Source	<1 %
18	patents.justia.com Internet Source	<1 %
19	Kian-Lee Tan. "Stratified computation of skylines with partially-ordered domains", Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD 05 SIGMOD 05, 2005 Publication	<1 %
20	Ruhul Amin, Taufik Djatna, Annisa, Imas Sukaesih Sitanggang. "Recommendation System based on Skyline Query: Current and Future Research", 2020 International	<1 %

Conference on Computer Science and Its Application in Agriculture (ICOSICA), 2020
Publication

21	Shi, Jieming, Dingming Wu, and Nikos Mamoulis. "Textually Relevant Spatial Skylines", IEEE Transactions on Knowledge and Data Engineering, 2015. Publication	<1 %
22	Xiaoyong Li, Yijie Wang, Yu Zhao, Yuan Wang, Xiaoling Li. "GPS: A General Framework for Parallel Queries over Data Streams in Cloud", 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013 Publication	<1 %
23	Salem Al-Naemi, Rachid Benlamri, Rehan Sadiq, Aitazaz Farooque, Michael Phillips. "Innovation and Technological Advances for Sustainability", CRC Press, 2024 Publication	<1 %
24	Yonis Gulzar, Ali A. Alwan, Hamidah Ibrahim, Sherzod Turaev, Sharyar Wani, Arjumand Bano Soomo, Yasir Hamid. "IDSA: An Efficient Algorithm for Skyline Queries Computation on Dynamic and Incomplete Data With Changing States", IEEE Access, 2021 Publication	<1 %

25	Jun Liu, Xiaoyong Li, Kaijun Ren, Junqiang Song. " Parallelizing uncertain skyline computation against -of- data streaming model ", Concurrency and Computation: Practice and Experience, 2018 Publication	<1%
26	Mudathir Ahmed Mohamud, Hamidah Ibrahim, Fatimah Sidi, Siti Nurulain Mohd Rum et al. "A Systematic Literature Review of Skyline Query Processing over Data Stream", IEEE Access, 2023 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		