DSCC440 Data Mining: Final Project

Prof Monika Polak, University of Rochester, NY

Algorithm for Sequential and Dynamic Association Rules Mining based on Apriori Algorithm and Markov Chain Model

Yibin Wang
University of Rochester
Goergen Institute for Data Science
Rochester, NY 14620
ywang448@ur.rochester.edu

Saruultugs Batbayar
University of Rochester
Goergen Institute for Data Science
Rochester, NY 14620
sbatbaya@ur.rochester.edu

Abstract*— *In this project, we present an algorithm for mining sequential and dynamic association rules by combining the Apriori algorithm's pruning strategies with the probabilistic framework of the Markov Chain model. This approach captures sequential dependencies and multi-step transitions in customer behavior, providing an effective method for generating actionable rules for applications such as personalized recommendations and webpage optimization.

Our algorithm dynamically estimates theoretical frequencies using transition probabilities to identify frequent itemsets, significantly reducing computational overhead. For example, in our evaluation, the algorithm required only 43 dataset scans compared to 8,489 scans in a conventional approach, representing a computational reduction of over 99%. Despite this efficiency, it successfully captured 66.7% of the most frequent 3-item rules in the dataset, validating its accuracy and robustness.

.

I. INTRODUCTION

The rapid growth of e-commerce and online grocery platforms has enabled retailers to collect not only transactional data but also sequential data that captures the order in which customers add items to their carts. Traditional approaches to market basket analysis, such as the Apriori and FP-tree algorithms, focus on static itemset associations and do not account for the sequential or probabilistic nature of customer behavior. However, incorporating this sequential information can provide valuable insights into dynamic customer preferences, enabling more precise and personalized recommendations.

Sequence is important in market basket analysis. Considering the same set of items, {Apple, Battery, Citrus}, customers who adds items in the sequence {Battery ⇒ Apple} may have a higher probability of selecting Citrus next, compared to customers who follows the sequence {Apple ⇒ Battery}. These sequential patterns reflect nuanced differences in customer intent and behavior, which cannot be captured effectively by traditional association rule mining methods. Retailers, therefore, have a compelling interest in identifying association rules that represent dynamic, sequential, and probabilistic relationships between items.

This project proposes a novel algorithm for mining sequential and dynamic association rules that leverage both the Apriori algorithm's candidate-pruning approach and the Markov Chain model's probabilistic framework. By modeling customer behavior as an ergodic Markov chain—where customers transition between products in a memoryless manner with stable, constant probabilities—we capture the likelihood of movement between items in a way analogous to the PageRank algorithm for web navigation. This approach allows us to incorporate multi-step transitions and explore associations that reflect customer behavior more accurately.

The proposed algorithm outputs sequential association rules, such as the probability of a customer purchasing Product A, then Product B, followed by Product C. Additionally, it generates conditioned sequential rules, such as the probability of purchasing Product D given a prior path {A ⇒ B ⇒ C}. These results can be applied in real-time, personalized recommendation systems, dynamically adjusting suggestions based on a customer's current selection sequence. Moreover, the insights can inform website layout designs to streamline the shopping process and enhance user experience.

Through this work, we aim to bridge the gap between traditional association rule mining techniques and the need for dynamic, sequential analysis, offering a practical solution for modern e-commerce platforms to better understand and serve their customers.

DSCC440 Data Mining: Final Project

Prof Monika Polak, University of Rochester, NY

II. Existing Algorithm as Background

A. *Apriori Algorithm*

The Apriori algorithm is a foundational method in association rule mining and is widely used for uncovering frequent itemsets in transactional datasets. Introduced by Agrawal and Srikant in 1994, the algorithm operates under the principle that any subset of a frequent itemset must itself be frequent. This property, known as the *Apriori property*, allows the algorithm to systematically reduce the search space, making it highly efficient for large datasets.

The algorithm follows a bottom-up approach, where it first identifies frequent 1-itemsets (items that meet a user-defined minimum support threshold) and then incrementally extends these itemsets to larger candidate sets. The process consists of three main steps:

1. **Generating Candidate Itemsets**: In each iteration, the algorithm generates a set of candidate itemsets by combining frequent itemsets from the previous iteration. For example, frequent 1-itemsets are used to generate candidate 2-itemsets, and so on.
2. **Pruning Candidates**: Candidate itemsets that contain non-frequent subsets are eliminated based on the Apriori property. This step significantly reduces the number of itemsets that need to be evaluated.
3. **Counting and Filtering**: The algorithm scans the dataset to count the occurrences of each candidate itemset. Only those candidates that meet the minimum support threshold are retained as frequent itemsets.

This iterative process continues until no more frequent itemsets can be identified.

One of the key strengths of the Apriori algorithm is its ability to handle large datasets efficiently through its candidate-pruning strategy. However, the algorithm can become computationally expensive as the size of the candidate itemsets grows, particularly in datasets with high cardinality or a low support threshold.

In the context of our work, the Apriori algorithm provides the foundation for candidate pruning. By adapting its pruning mechanism to consider sequential dependencies and incorporating probabilistic transitions modeled through a Markov Chain framework, we extend the Apriori approach to mine sequential and dynamic association rules effectively. This integration leverages the computational efficiency of Apriori while addressing its limitations in capturing sequential and probabilistic relationships.

B. *Markov Chain Model*

A Markov Chain is a mathematical model used to describe systems that transition from one state to another within a finite set of states, where the probability of moving to the next state depends solely on the current state. This *memoryless property*, also known as the *Markov property*, implies that the future state is independent of the sequence of events that preceded it, relying only on the present state. Core concepts of a Markov Chain model that are used in our algorithm includes:

States and Transitions: A Markov Chain consists of a finite set of states, with transitions between these states governed by a set of probabilities. These probabilities are typically represented in a *transition matrix*, where each entry specifies the likelihood of moving from one state to another. In the context of consumer behavior, each selection of product can be seen as one state, and customers' moving from moving from purchasing one product to another can be seen as a transition.

Transition Probabilities: The probability of transitioning from state i to state j is denoted as $P(j|i)$. In the context of consumer behavior, these probabilities can represent the likelihood of a customer moving from purchasing one product to another.

To address the limitations of traditional association rule mining techniques, we incorporate a Markov Chain model into our algorithm to capture sequential and dynamic associations between items. The key aspects of this integration are as follows:

1. **Modeling Consumer Behavior**: We assume that customer behavior can be represented as an ergodic Markov Chain, where items in a transaction act as states and transitions reflect the likelihood of moving from one item to another. This assumption allows us to model the sequence of item selection with transition probabilities, capturing the dynamic nature of customer preferences.
2. **Transition Probability Estimation**: To calculate the likelihood of multi-step transitions, we estimate the probabilities of moving from one item to another based on observed sequences in the dataset. For example, the transition probability $P(B|A)$ represents the likelihood that a customer who selects item A will next select item B. These probabilities are gathered from observations in the dataset.
3. **Markov Chain-Based Theoretical Frequency**: Using the transition probabilities, we compute the theoretical frequency of longer item sequences. For

instance, the probability of a three-item sequence A→B→C is given by:

$$P(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C}) = P(A) * P(B|A) * P(C|B)$$

where $P(A)$ is the initial probability of item A, $P(B|A)$ is the transition probability from A to B, and $P(C|B)$ is the transition probability from B to C. This method allows us to capture sequential dependencies without requiring explicit enumeration of all possible sequences.

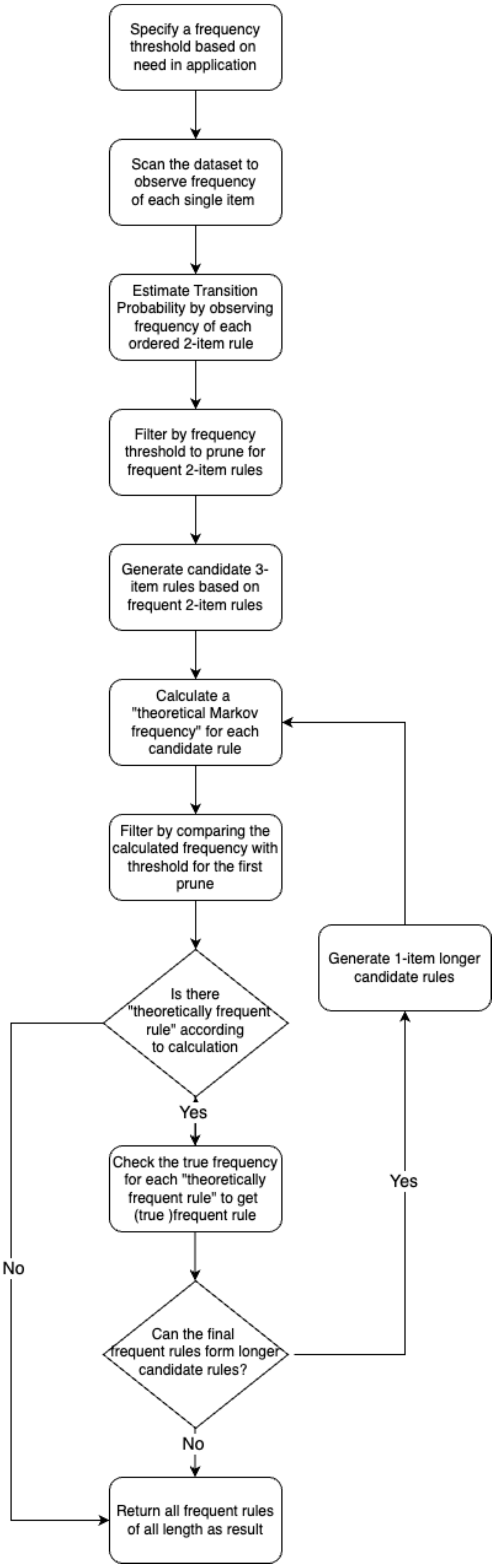
4. **Dynamic Candidate Pruning**: By leveraging Markov Chain-based theoretical frequencies, we dynamically prune candidate rules that do not meet a predefined minimum frequency threshold. This approach reduces computational overhead by focusing only on sequences that are likely to be frequent, eliminating unpromising candidates early in the process.

By combining the Apriori algorithm's candidate pruning mechanism with a Markov Chain framework, our algorithm gained capability to process sequential information, and scalability as well as flexibility are enforced. Our proposed method extends traditional association rule mining to include dynamic and sequential rules. The Markov Chain model allows us to incorporate multi-step transitions and probabilistic dependencies, resulting in a powerful tool for uncovering insights into customer behavior. This integration enables applications such as real-time recommendation systems and optimized webpage layouts that reflect the dynamic and sequential nature of online shopping behavior. The memoryless property of Markov Chains allows us to treat sequences independently of their positions, simplifying the analysis of complex transactional data. By using theoretical frequencies for pruning, the algorithm reduces the need for exhaustive dataset scans, making it more scalable for large datasets.

III. Algorithm Introduction and Procedure

The proposed algorithm is a hybrid approach that combines the principles of the Apriori algorithm with the probabilistic framework of Markov Chains. Designed specifically for sequential and dynamic association rule mining, the algorithm addresses the limitations of traditional methods by incorporating sequential dependencies and leveraging transition probabilities to prune candidate rules effectively.

The algorithm follows a structured workflow with the following steps:


Specify a frequency threshold based on need in application
Scan the dataset to observe frequency of each single item
Estimate Transition Probability by observing frequency of each ordered 2-item rule
Filter by frequency threshold to prune for frequent 2-item rules
Generate candidate 3-item rules based on frequent 2-item rules
Calculate a "theoretical Markov frequency" for each candidate rule
Filter by comparing the calculated frequency with threshold for the first prune
Is there "theoretically frequent rule" according to calculation
Yes
No
Check the true frequency for each "theoretically frequent rule" to get (true)frequent rule
Can the final frequent rules form longer candidate rules?
Yes
No
Generate 1-item longer candidate rules
Return all frequent rules of all length as result

1. **Initialization**: Specify a frequency threshold based on the requirements of the application. This threshold is used to determine which item sets are considered frequent.
2. **First Dataset Scan**: Scan the dataset to calculate the frequency of each individual item. This step establishes the initial probabilities for all single items.
3. **Transition Probability Calculation**: Estimate the transition probabilities for all ordered 2-item rules by analyzing the frequency of item pairs. These probabilities represent the likelihood of transitioning from one item to another in a sequence.
4. **Filter 2-Item Rules**: Apply the specified frequency threshold to prune infrequent 2-item rules. Only frequent 2-item rules are retained for further analysis.
5. **Generate Longer Candidate Rules**: Generate candidate 3-item rules based on the frequent 2-item rules. This step extends the rule length iteratively, leveraging the relationships established in the prior step.
6. **Calculate Theoretical Markov Frequency**: For each candidate rule, calculate a theoretical frequency using the Markov Chain model. This frequency is computed based on initial probabilities and transition probabilities for each step in the sequence.
7. **First Pruning**: Compare the theoretical frequency of each candidate rule against the frequency threshold. Candidate rules that fail to meet this threshold are eliminated in the first pruning stage.
8. **Validation Prune by Dataset Scan**: For the remaining "theoretically frequent rules," perform a dataset scan to validate their true frequencies. This step ensures that the retained rules are truly frequent in the dataset.
9. **Iterative Rule Extension**: Check whether the final set of frequent rules can form longer candidate rules. If possible, generate 1-item longer candidate rules and repeat the process from Step 6 (calculating theoretical frequency) for the new candidates.
10. **Termination**: If no further candidates can be generated, or no candidates meet the frequency threshold, terminate the process.

All frequent rules of all lengths are then returned as the final result.

To further illustrate the proposed algorithm, we apply it to a small, simulated dataset. The dataset contains ordered transactions as shown:

Order #	Items
1	A, B, C, B
2	A, C, D, B, A
3	B, A, C, A
4	D, C, A, A
5	C, B, C, D, A, D
6	A, B, C, C, A
7	C, D, C, B
8	B, A, A, C
9	C, A, B, C
10	A, C, B, C

For this example, we set the frequency threshold at **5%**, and the mining process is limited to 3-item rules for simplicity.

Step 1: Initial Probabilities of Items

The frequency of each item in the dataset is computed as follows:

Item	Count	Probability
A	14	P(A) = 14/44 = 0.32
B	10	P(B) = 10/44 = 0.22
C	15	P(C) = 15/44 = 0.34
D	5	P(D) = 5/44 = 0.11

Step 2: Transition Frequencies and Probabilities

For transition frequency, we observed:

A→A: 2,	A → B: 3,	A → C: 4,	A → D: 1,
B→A: 3,	B → B: 0,	B → C: 5,	B → D: 0,
C→A: 4,	C → B: 4,	C → C: 1,	C → D: 3,
D→A: 1,	D → B: 1,	D → C: 2,	D → D: 0,

Transition probability can be then calculated as:

	To: A	B	C	D
From: A	0.200	0.300	0.400	0.100
B	0.375	0.000	0.625	0.000
C	0.333	0.333	0.083	0.250
D	0.250	0.250	0.500	0.000

The total number of observed transitions is **34**, and the minimum frequency threshold for 2-item rules is:

$$Threshold(2 - item) = 34 \times 0.05 = 1.7$$

Frequent 2-item rules include:

{{A→A}, {A→B}, {A→C}, {B→A}, {B→C}, {C→A}, {C→B}, {C→D}, {D→C}}

Step 3: Candidate 3-Item Rules

Based on frequent 2-item rules, candidate 3-item rules are generated by chaining transitions. The following candidates are evaluated:

Candidate 3-item rule: {A→A→A}, {A→A→B}, {A→A→C}, {A→B→A}, {A→B→C}, {A→C→A}, {A→C→B}, {A→C→D}, {A→D→C}, {B→A→A}, {B→A→B}, {B→A→C}, {B→C→A}, {B→C→B}, {B→C→D}, {C→A→A}, {C→A→B}, {C→A→C}, {C→B→A}, {C→B→C}, {C→D→C}, {D→C→A}, {D→C→B}, {D→C→D},

Step 4: Markov Theoretical Frequencies

We calculate the "theoretical Markov frequency" of each candidate:

Candidate	Calculation	Result
{A→A→A}	$P(A) * P(A\|A) * P(A\|A)$	0.0128
{A→A→B}	$P(A) * P(A\|A) * P(B\|A)$	0.0192
{A→A→C}	$P(A) * P(A\|A) * P(C\|A)$	0.0256
{A→B→A}	$P(A) * P(B\|A) * P(A\|B)$	0.0360
{A→B→C}	$P(A) * P(B\|A) * P(C\|B)$	0.0600
{A→C→A}	$P(A) * P(C\|A) * P(A\|C)$	0.0426
{A→C→B}	$P(A) * P(C\|A) * P(B\|C)$	0.0426
{A→C→D}	$P(A) * P(C\|A) * P(D\|C)$	0.0320
{A→D→C}	$P(A) * P(D\|A) * P(C\|D)$	0.0160
{B→A→A}	$P(B) * P(A\|B) * P(A\|A)$	0.0165
{B→A→B}	$P(B) * P(A\|B) * P(B\|A)$	0.0248
{B→A→C}	$P(B) * P(A\|B) * P(C\|A)$	0.0330
{B→C→A}	$P(B) * P(C\|B) * P(A\|C)$	0.0458
{B→C→B}	$P(B) * P(C\|B) * P(B\|C)$	0.0458
{B→C→D}	$P(B) * P(C\|B) * P(D\|C)$	0.0348
{C→A→A}	$P(C) * P(A\|C) * P(A\|A)$	0.0133
{C→A→B}	$P(C) * P(A\|C) * P(B\|A)$	0.0340
{C→A→C}	$P(C) * P(A\|C) * P(C\|A)$	0.0056
{C→B→A}	$P(C) * P(B\|C) * P(A\|B)$	0.0425
{C→B→C}	$P(C) * P(B\|C) * P(C\|B)$	0.0708
{C→D→C}	$P(C) * P(D\|C) * P(C\|D)$	0.0425
{D→C→A}	$P(D) * P(C\|D) * P(A\|C)$	0.0183
{D→C→B}	$P(D) * P(C\|D) * P(B\|C)$	0.0183
{D→C→D}	$P(D) * P(C\|D) * P(D\|C)$	0.0138

According to our calculation, the "theoretically frequent rules" are {A→B→C} and {C→B→C}.

Step 5: Frequent 3-Item Rules

The total number of observed 3-item transitions is **24**, and the minimum frequency threshold for 2-item rules is:

$$Threshold(3 - item) = 24 \times 0.05 = 1.2$$

Dataset validation confirms that both rules meet the threshold of actual frequency:

{A→B→C}: 3, {C→B→C}: 2

which means that both of the "theoretically frequent rules" are indeed frequent.

Final results is returned with all frequent 2-item and 3-itme rules mined.

Validation of Results

To check if we filtered out "true frequent" rules, we scan the dataset and find out all frequent 3-item rules, the result is {A→B→C} and {C→B→C}, meaning that all frequent 3-item rules in the dataset have been mined.

The simulation demonstrates the significant advantages of the proposed algorithm in mining frequent 3-item rules, emphasizing its efficiency, accuracy, scalability, and interpretability.

1. **Efficiency**: The proposed algorithm significantly reduces the computational cost associated with mining frequent 3-item rules by employing a combination of candidate pruning and theoretical frequency calculations.
   - For a dataset with k unique items, there are $k \times k \times k$ possible combinations of 3-item rules. In this simulation, where k = 4, there are 4×4×4=64 potential 3-item rules. To evaluate the frequency of all these rules, a computer would need to scan the dataset 64 times, leading to substantial computational overhead.
   - By pruning candidates dynamically using frequent 2-item rules and theoretical frequencies, the

algorithm reduces the number of candidates to evaluate. In this simulation, only 24 candidate 3-item rules were generated, and only 2 of them are selected with theoretical frequencies, representing a reduction in computational complexity compared to evaluating all possible combinations. This efficiency is particularly valuable in larger datasets, where the number of potential rules grows exponentially.

2. **Accuracy**: The algorithm achieves fair accuracy by incorporating an assumption that is suitable for the problem. In the simulation, all frequent 3-item rules are mined, subsequent validation confirmed that result rules were indeed frequent in the dataset.
3. **Scalability**: The algorithm efficiently manages the exponential growth in the number of potential rules as the number of unique items (k) increases:

   As l grows, the total number of possible 3-item rules increases exponentially (k^3). Without pruning, the computational cost of mining these rules becomes infeasible. By leveraging the Apriori principle and Markov Chain-based pruning, the algorithm scales effectively, maintaining computational feasibility even for larger datasets or when mining higher-order rules (e.g., 4-item or 5-item rules).
4. **Interpretability**: The algorithm provides a probabilistic framework for identifying frequent rules, enhancing the interpretability of the results.

   By calculating Markov theoretical frequencies based on initial probabilities and transition probabilities, the algorithm offers a transparent rationale for why certain rules are considered frequent.

While the proposed algorithm demonstrates significant advantages in efficiency, accuracy, scalability, and interpretability, it also has certain limitations that need to be acknowledged. These shortcomings primarily arise due to its reliance on theoretical frequencies for pruning, its performance on small datasets, and the use of a single percentage threshold across rule lengths.

1. **Information Loss**: The algorithm's reliance on theoretical frequencies to prune candidates, while efficient, introduces a risk of filtering out candidates that are actually frequent in the dataset:
   - The algorithm calculates theoretical frequencies using initial probabilities and transition probabilities, which serve as proxies for actual frequencies. Candidates with theoretical frequencies below the threshold are eliminated before dataset scanning. However, due to inaccuracies in probability estimation, some candidates that are truly frequent may be discarded prematurely.
   - This limitation is particularly problematic in datasets with non-uniform item distributions or in scenarios where the dependencies between items are complex and not well captured by the Markov Chain model.
2. **Poor Performance on Small Datasets**: The algorithm's reliance on observed data to estimate probabilities makes it less effective when applied to small datasets:
   - Initial probabilities $P(A)$ and transition probabilities $P(B|A)$ are estimated from the dataset. When the dataset is small, these estimates can be unreliable due to insufficient observations, leading to inaccurate theoretical frequencies.
   - In small datasets, the algorithm may fail to identify truly frequent rules or overestimate the importance of infrequent patterns. This shortcoming undermines the accuracy of the algorithm and limits its applicability in datasets with limited transactions.
3. **Potential Problem with a Single Percentage Threshold**:

   Using a single percentage threshold across rules of all lengths can lead to inconsistencies, particularly for longer rules. In this simulation, the threshold was set at 5%. For 4-item rules, there were only 14 4-item transitions in total. The threshold for a rule to be frequent was 14×5%=0.7. This effectively means that any 4-item rule that appeared in the dataset even once was considered frequent, since the minimum possible frequency is 1. This undermines the purpose of using a threshold and may lead to the inclusion of insignificant rules. This issue is exacerbated as the rule length increases, since the total number of transitions decreases exponentially, making the threshold less meaningful. A single percentage threshold is less effective for datasets with uneven distributions of item combinations or for rules of varying lengths.

Based on the observations, we propose an improvement to the algorithm by introducing a **dynamic thresholding mechanism**. This approach addresses the limitation of using a single, static probability p as the threshold for identifying "theoretical frequent rules" across rules of varying lengths.

In the improved algorithm, the threshold dynamically adjusts based on the rule length k. Specifically, for a rule of length k, the threshold is calculated as p^{k-1}, where p is the base probability initially set by the user or dataset properties.

This adjustment reflects the intuition that longer rules inherently have lower probabilities due to the compounding effect of transitions. By scaling the threshold using p^{k-1}, we ensure that longer rules are evaluated more leniently, preventing over-pruning, while still maintaining stringent criteria for shorter rules. This improvement enhances the algorithm's flexibility and consistency in handling rules of varying lengths.

IV. Dataset

For validation of the effectiveness of the algorithm, we implement the algorithm in a real-life groceries' record dataset. The dataset used for this study captures transaction-level information, detailing purchasing patterns for various items in a retail environment. The analysis aims to uncover frequent item sets, explore correlation structures, and assess probabilities associated with item combinations.

A. *Dataset Description*

The dataset utilized in this study has 38765 rows of the purchase orders in the grocery stores, with the following features:

1. **Member_number**: A unique identifier for each customer.
2. **Date**: The transaction date.
3. **ItemDesctiption**: The product purchased during the transaction.

B. *Data Preprocessing*

The raw features of the original data (Member_number, Date, ItemDescription) were subsequently processed to derive new attributes that are essential for analyzing purchasing patterns and evaluating sequential associations between items. The processed features include:

- **List of items by member number and date (List_items_day):** This feature represents the collection of items purchased by a specific customer on a particular day, allowing for the identification of "customer-specific" buying patterns and transaction-level data aggregation.
- **The probability of a single item is purchased (support_1item):** The feature represents the likelihood of an item being part of a transaction. It was calculated as the ratio of the total number of transactions containing the item to the total number of transactions in the dataset.
- **2-item set combinations (2-itemset)**: Using the Apriori algorithm, all possible pairs of co-purchased items were generated to identify frequent item sets.
- **Probability of two items being purchased together (support_2items)**: This metric quantified the co-occurrence of item pairs, offering insights into product affinity.
- **Probability of an item being purchased after a prior item (confidence):** This sequential probability captured dynamic purchasing trends, forming the basis of Markov chain modelling. It was computed as the conditional probability of the second item occurring, given the first item had already been purchased.

These transformations enabled the integration of both static and dynamic attributes into the framework.

C. *Data Insights*

The following visualization illustrate key insights derived from the dataset:

1. The graph provides insights into customer purchase patterns, which are valuable for association rule mining, marketing, and operational decisions. Right-skewness is common, indicating most transactions have fewer items, with fewer large transactions.

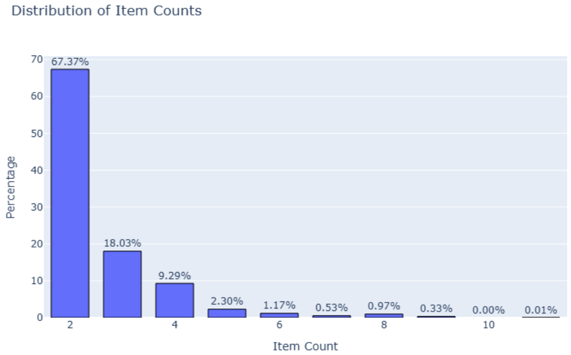


Figure 1: Distibution of Item Counts

2. Most frequently purchased items: Product such as whole milk dominate transactions, validating their role as anchor points in both static and sequential association rules.

DSCC440 Data Mining: Final Project

Prof Monika Polak, University of Rochester, NY

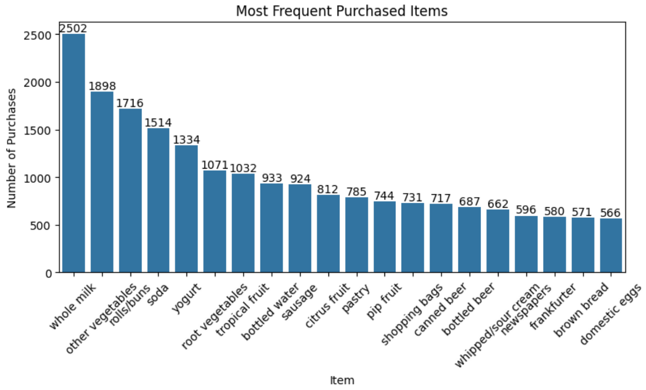


Figure 2: Most frequent Purchased Items

3. Popularity of top 15 combinations: The chart illustrates the frequency of the most commonly purchased item pairs across all transactions. These pairs represent the highest-ranking 2-item combinations derived.

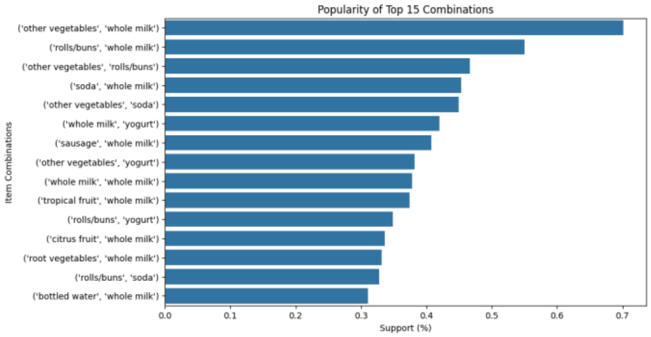


Figure 3: Popularity of Top 15 Combinations

4. Correlation matrix of top 20 frequently sold items: This matrix highlights relationships between frequently purchased products, showing that certain pairs (e.g., whole milk and yogurt) exhibit stronger association, forming a basis for Apriori-based itemset generation.

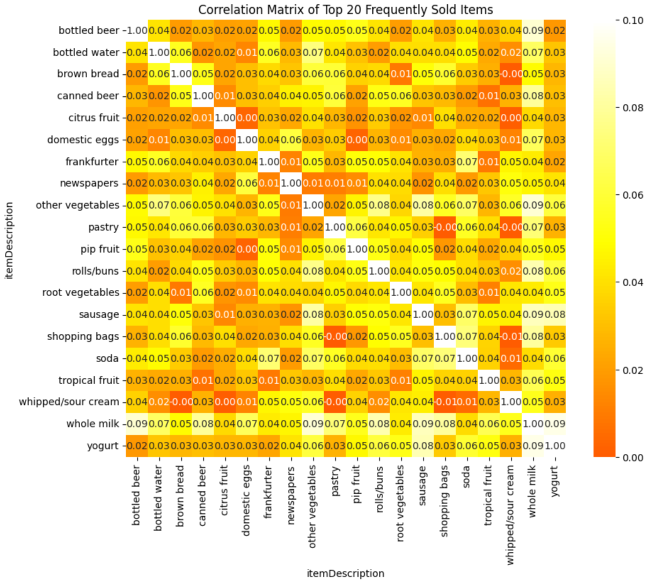


Figure 4: Most Frequently Purchased Items

5. Comparison of High and Low-Probability Combinations. The stark contrast between high-probability (473 combinations) and low-probability (27,249 combinations) associations underscores the importance of focusing on significant transitions.

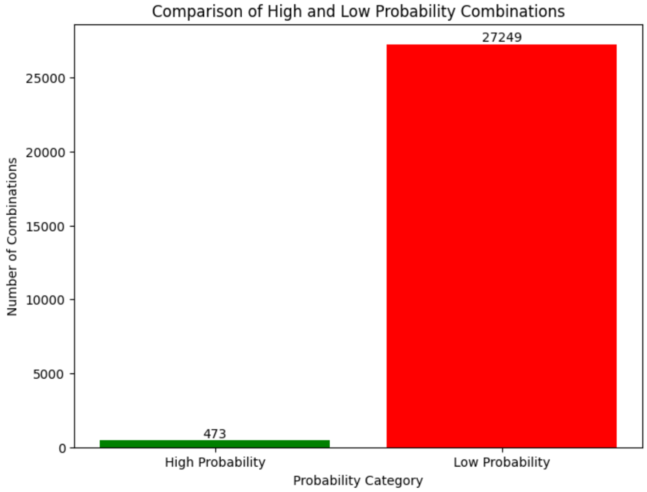


Figure 5: Comparison of High and Low Probability Combinations

6. Top 20 highest-probability combinations: High-probability pairs, such as (mustard, whole milk), highlight the value of integrating sequential dependencies for actionable rule mining.

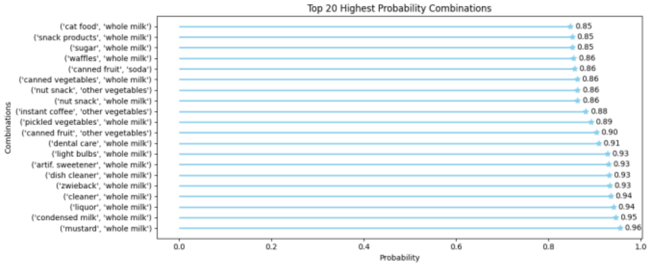


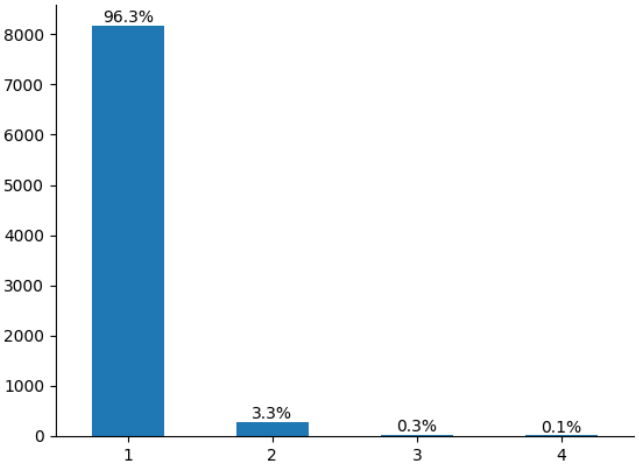
Figure 6: Top 20 Highest-Probability Combinations

d. Relevance to Study Objectives

The dataset and its derived features directly contribute to the objectives of this study, which aims to enhance association rule mining by integrating sequential and dynamic patterns into the analysis. The probabilistic features generated through data preprocessing enable the identification of meaningful co-occurrences and sequential dependencies, addressing key limitations in traditional methods. By leveraging both Apriori and Markov Chain methodologies, the framework facilitates the generation of actionable rules for personalized recommendations and retail optimization.

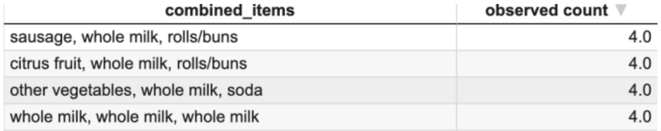
V. Implementation And Evaluation

For the purpose of evaluating the algorithm in a real-life condition, we implemented the proposed algorithm in a Python environment, using the dataset described in Part IV. Given the observation that the dataset contains relatively short entries, we restricted the implementation to rules of length-3. This restriction aligns with the dataset's characteristics, ensuring computational efficiency while maintaining the focus on extracting meaningful patterns. By focusing on 3-item rules, the implementation effectively demonstrates the algorithm's ability to identify frequent sequential patterns without overextending to lengths that may not yield significant insights in this dataset.



To better evaluate the performance of the algorithm, we conducted an analysis of the actual frequencies of all possible 3-item rules by exhaustively counting all length-3 combinations in the original dataset. The results revealed that the frequency distribution of 3-item rules is highly dispersed, with the highest frequency observed for any 3-item rule being just 4. Furthermore, only 6 such rules achieved this maximum frequency, out of a total of 8,489 length-3 transactions.

For the purpose of evaluation, we set the goal of the task as determining how many of these 6 high-frequency (length-4) rules the algorithm successfully identifies, while simultaneously assessing the extent of computational savings achieved by pruning low-probability candidates. This dual focus on accuracy and efficiency ensures a comprehensive assessment of the algorithm’s performance.



combined_items	observed count
sausage, whole milk, rolls/buns	4.0
citrus fruit, whole milk, rolls/buns	4.0
other vegetables, whole milk, soda	4.0
whole milk, whole milk, whole milk	4.0

The results returned by the algorithm demonstrate its effectiveness and efficiency. The algorithm successfully identified 4 out of the 6 highest-frequency 3-item rules (with a frequency of 4). This indicates that the algorithm is able to capture a significant portion of the most important patterns in the dataset.

On the efficiency side, the algorithm identified only 43 “theoretical frequent rules” as candidates, meaning the computer needed to scan the dataset just 43 times, as opposed to the 8,489 possible candidates in an exhaustive search. This represents a computational reduction of over 99%, achieving the same level of performance with just 43/8,489 of the computational power. Despite this substantial computational saving, the algorithm preserved 66.7% of the most frequent rules, demonstrating its ability to balance efficiency and information retention effectively.

VI. Conclusion

This study introduced a hybrid algorithm for mining sequential and dynamic association rules, effectively integrating the Apriori algorithm's pruning techniques with the Markov Chain model's probabilistic framework. The proposed method addresses key limitations in traditional association rule mining, including its inability to capture sequential relationships and its computational inefficiency.

Through simulations, the algorithm demonstrated significant advantages:

- **Efficiency**: By dynamically pruning candidates, it drastically reduced computational cost while maintaining accuracy.
- **Accuracy**: The algorithm successfully identified the

majority of high-frequency rules while preserving computational resources.

- **Scalability**: Its ability to handle large datasets and adapt to increasing rule lengths makes it a robust tool for real-world applications.
- **Flexibility**: The inclusion of an adaptive threshold mechanism ensures applicability across diverse datasets and rule lengths.

However, the study also identified areas for improvement, such as mitigating information loss due to theoretical pruning, addressing performance issues in small datasets, and refining threshold strategies for longer rules. Future work could focus on incorporating higher-order Markov Chains, dynamic parameter adjustment, and hybrid models that combine theoretical and empirical methods.

Overall, the proposed algorithm offers a practical and scalable approach for mining meaningful sequential patterns, paving the way for advanced data-driven decision-making in retail, e-commerce, and beyond.

.

References

[1] Lim, Y. (2022, April 8). Data Mining: Market Basket Analysis with Apriori algorithm. Medium. https://towardsdatascience.com/data-mining-market-basket-analysis-with-apriori-algorithm-970ff256a92c

[2] Han, J., Kamber, M., Pei, J. (2011). Data Mining: Concepts and Techniques. Netherlands: Elsevier Science.

[3] *Groceries dataset*. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset/data

[4] *Google's PageRank algorithm for ranking nodes in general networks*. (n.d.). IEEE Xplore. https://ieeexplore.ieee.org/document/7497841

[5] Athalye, A. (2021, January 18). *A Markov chain formulation of grocery item picking process. Medium* https://medium.com/walmartglobaltech/a-markov-chain-formulation-of-grocery-item-picking-process-54c65a3ec5b5

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[7] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[8] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.