Market Basket Analysis

*Apriori Algorithm

1st Omar Abdellatif
Electronics Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
0009-0004-3780-4886

***Abstract*—The Apriori algorithm is a widely used method in data mining that helps uncover patterns in large sets of transaction data like what items people often buy together. It's especially useful in Market Basket Analysis [1], where businesses look for combinations of products that frequently appear in the same shopping carts. This insight can support smarter decisions around product placement, marketing, and sales strategies. Apriori works by first identifying commonly purchased individual items, then gradually building up to larger item combinations, keeping only those that appear often enough to be meaningful. It then creates rules that show how buying one item might suggest the purchase of another. By using simple measures like support, confidence, and lift, Apriori turns raw shopping data into valuable business insights in a clear and structured way.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

The Apriori algorithm is a foundational method in the field of data mining, specifically designed for discovering frequent itemset and generating association rules from large transactional datasets. It plays a central role in Market Basket Analysis, a technique used by retailers and analysts to uncover hidden patterns in customer purchasing behavior. The algorithm enables businesses to identify item combinations that frequently co-occur in transactions, leading to insights that support cross-selling, shelf organization, and targeted marketing strategies. [2]

II. Related Work

Association rule mining is basically a way to find interesting connections in large piles of data — kind of like spotting what items people tend to buy together at the grocery store. Earlier work by Mendonça and Ovalle [8] showed how these rules can be used beyond just shopping — for example, in software architecture to discover patterns that aren't obvious at first glance. Their research proved that association rules are pretty flexible and useful for many different fields.

More recently, researchers have been focusing on making these techniques faster and more accurate, which makes a lot of sense given how much data we collect these days. Martinez and his team came up with a smart way to speed up similarity checks by sampling data instead of comparing everything directly. This means you get nearly the same results, but way quicker something really helpful when you're dealing with thousands or millions of transactions.

Another important point is that relying just on traditional numbers like support and confidence can sometimes be misleading. That's why people started using other measures like Kulczynski and Jaccard scores, which help give a better sense of how strong or reliable a rule really is. These metrics balance frequency with relevance, so you don't end up chasing patterns that happen a lot by chance. Both Mendonça's and Martinez's work emphasize this more careful approach to evaluating patterns. [9]

All in all, the field has come a long way. It's not just about mining data blindly anymore it's about being smart with the math and the tools to find insights that truly matter. Our work builds on these ideas, applying them to grocery shopping data to see what real patterns we can uncover that might actually help stores and shoppers alike.

III. How the Apriori Algorithm Works

At its core, Apriori operates on the principle that: "If an itemset is frequent, all of its subsets must also be frequent." This downward-closure property allows the algorithm to prune large parts of the search space early, improving efficiency. The algorithm proceeds in the following steps:

A. Identify Frequent Itemsets

It begins by scanning the dataset to find individual items (1-itemsets) that meet a predefined minimum support threshold—i.e., items that appear in a significant number of transactions.

B. Generate Candidate Itemsets

Once frequent itemset are identified, the algorithm derives association rules that describe how the presence of one set of items in a transaction influences the presence of another. These rules are evaluated using three key metrics: support, confidence, and lift.

C. Association Rule Generation

Once frequent itemsets have been identified in a transaction dataset, the next step in association rule mining is the generation of *association rules*. An association rule is an implication expression of the form:

$$X \to Y$$

Identify applicable funding agency here. If none, delete this.

where X and Y are itemsets, and $X \cap Y = \emptyset$. The interpretation is that transactions containing the items in X tend to also contain the items in Y.

To determine the usefulness and validity of such rules, two fundamental metrics are used:

- **Support**: The support of an itemset Z is the proportion of transactions in the dataset D that contain Z:

$$\text{supp}(Z) = \frac{|\{t \in D \mid Z \subseteq t\}|}{|D|}$$

  Support measures the statistical significance of the itemset.
- **Confidence**: The confidence of the rule $X \to Y$ is the conditional probability that a transaction contains Y given that it contains X:

$$\text{conf}(X \to Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

  Confidence measures the predictive strength of the rule.

Only those rules that meet or exceed user-defined thresholds for support and confidence are considered *strong* and retained for analysis. [3]

Algorithmic Approach: Let $\mathcal{L}$ be the set of all frequent itemsets mined from the dataset using an algorithm such as Apriori. For each frequent itemset $L \in \mathcal{L}$ and each non-empty proper subset $A \subset L$, an association rule can be generated in the form:

$$A \to (L \setminus A)$$

- **Support:** This is the percentage of total transactions that contain both X and Y. If this value exceeds a predefined threshold, the rule is said to have sufficient support.
- **Confidence:** This measures how often Y appears in transactions that also contain X. It gives an idea of the rule's reliability.

D. Apriori Algorithm

Input: Database of transactions D; minimum support threshold min_sup

Output: Frequent itemsets L in D

$L_1 \leftarrow$ find frequent 1-itemsets in D $k = 2; L_{k-1} \neq \emptyset; k \leftarrow k + 1$ $C_k \leftarrow$ apriori_gen$(L_{k-1}, \text{min_sup})$ transactions $t \in D$ $C_t \leftarrow$ subset(C_k, t) candidates $c \in C_t$ c.count $\leftarrow$ c.count $+ 1$ $L_k \leftarrow \{c \in C_k \mid c.\text{count} \geq \text{min\_sup}\}$ **return** $L = \bigcup_k L_k$

IV. Apriori Algorithm Implementation and Analysis

This section presents the practical implementation of the Apriori algorithm on a retail transactional dataset. The process includes data loading and exploration, item frequency analysis, co-occurrence visualization, frequent itemset mining, association rule generation, and results visualization.

A. Dataset Loading and Exploration

The dataset consists of individual grocery transactions stored in the file `Groceries_dataset.csv`. [4] Each row represents a purchased item associated with a unique member and date. The dataset was loaded using the `pandas` library:

```
my_data = pd.read_csv("Groceries_dataset.csv")
```

Top and bottom selling items were identified using value counts on the `itemDescription` column, revealing the most frequently and least frequently sold items.

B. Item Frequency Visualization

A bar chart depicting the top 15 most sold items was generated using `matplotlib` and `seaborn` [5] to visualize the distribution of sales frequency. This offers insight into dominant products influencing purchasing patterns.

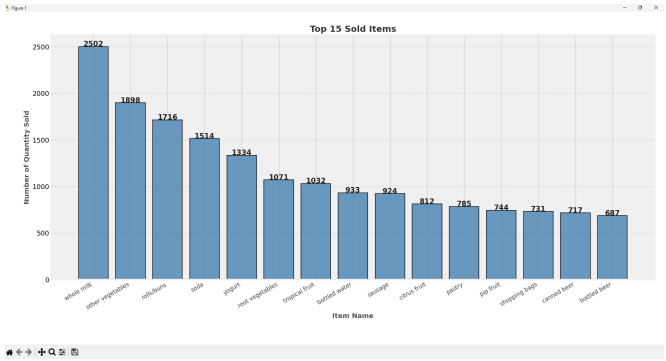

Fig. 1. Example of a figure caption.

C. Co-occurrence Analysis

To understand item pair relationships, item co-occurrence matrices were computed from transactions aggregated by unique members:

$$\text{Transactions} = \{T_1, T_2, \ldots, T_N\}, \quad T_i \subseteq \mathcal{I}$$

where $\mathcal{I}$ is the set of all items. All item pairs (i, j) within transactions were enumerated via combinations:

$$\text{Pairs}(T_i) = \{(i_a, i_b) : i_a, i_b \in T_i, i_a \neq i_b\}$$

Counting the frequency of each pair across transactions generated the co-occurrence matrix C for the top 10 items. This matrix was visualized as a heatmap, illustrating which items are frequently purchased together.

D. Network Graph of Item Co-occurrences

The 60 most frequent item pairs were used to construct an undirected weighted graph $G = (V, E)$, where nodes V represent items and edges E indicate co-occurrence frequency weights. Node sizes and colors correspond to individual item sales frequency, while edge widths represent pair frequencies. This network visualization highlights strongly connected items in the purchase network.

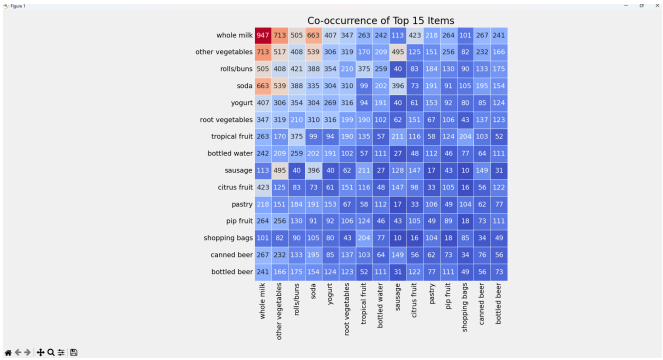

Fig. 2. Example of a figure caption.

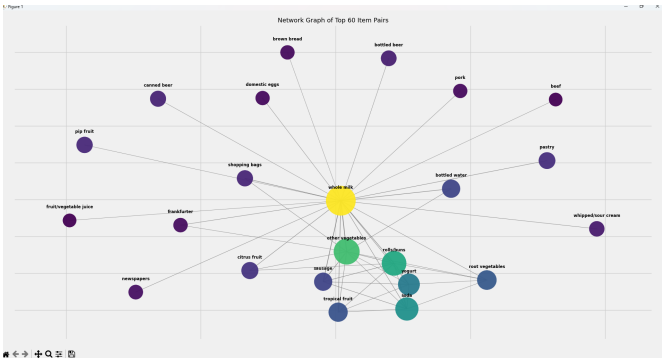

Fig. 3. Example of a figure caption.

E. Data Preparation for Apriori Algorithm

Transactions were redefined by concatenating `Member_number` and `Date` to form unique transaction identifiers:

$$\text{singleTransaction} = \text{Member\_number} \,||\, \text{Date}$$

The list of items per transaction was aggregated:

$$\{T_i\} = \texttt{groupby(singleTransaction)} \rightarrow \text{list of items}$$

Using the `TransactionEncoder` from `mlxtend`, the transaction dataset was converted into a binary matrix $M \in \{0,1\}^{N \times m}$, where m is the number of unique items, indicating presence (1) or absence (0) of items per transaction.

F. Frequent Itemset Mining

The Apriori algorithm was applied with a minimum support threshold $\sigma = 0.0005$. The support of an itemset X is:

$$\text{supp}(X) = \frac{\#\text{transactions containing } X}{N}$$

Itemsets meeting $\text{supp}(X) \geq \sigma$ were extracted, and their lengths recorded. The top frequent itemsets by support were printed for inspection.

G. Association Rule Generation

From the frequent itemsets, association rules $A \rightarrow B$ were generated with minimum confidence threshold $\gamma = 0.1$. The confidence is defined as:

$$\text{conf}(A \rightarrow B) = \frac{\text{supp}(A \cup B)}{\text{supp}(A)}$$

Additional interestingness metrics were computed:

- **Jaccard Index:**

$$J(A,B) = \frac{\text{supp}(A \cup B)}{\text{supp}(A) + \text{supp}(B) - \text{supp}(A \cup B)}$$

- **Certainty Factor:**

$$\text{certainty}(A \rightarrow B) = \text{conf}(A \rightarrow B) - \text{supp}(B)$$

- **Kulczynski Measure:**

$$K = \frac{1}{2}\left(\text{conf}(A \rightarrow B) + \frac{\text{supp}(A \cup B)}{\text{supp}(B)}\right)$$

V. Results

TABLE I
SAMPLE OF ASSOCIATION RULES

Antecedents	Consequents	Certainty	Kulczynski	Jaccard
(pork, sausage)	(whole milk)	0.233	0.198	0.153
(sweet spreads)	(pip fruit)	0.069	0.064	0.055
(sweet spreads)	(tropical fruit)	0.094	0.086	0.070
(rolls/buns, whipped/sour cream)	(yogurt)	0.119	0.106	0.086
(spices)	(soda)	0.128	0.116	0.095
(sausage, shopping bags)	(other vegetables)	0.154	0.140	0.112
(yogurt, whole milk)	(sausage)	0.071	0.078	0.057
(pastry, soda)	(sausage)	0.071	0.070	0.053
(brandy)	(whole milk)	0.184	0.174	0.139
(pork, whole milk)	(sausage)	0.060	0.065	0.048

The analysis of the transactional grocery dataset revealed meaningful associations among frequently purchased items. Using association rule mining, several strong and interpretable patterns were discovered. These rules help uncover latent structures in customer purchasing behavior and can be beneficial for strategic decisions in product placement.

Each rule was evaluated using standard interest measures, including **confidence**, **kulczynski**, and **jaccard** [8]. Confidence indicates the conditional probability that the consequent is purchased given the antecedent. For instance, a rule such as *(pork, sausage) → (whole milk)* with a confidence of 0.233 suggests that 23.3% of the transactions that include pork and sausage also include whole milk.

The **kulczynski** measure was used to account for symmetric dependencies between items, averaging the confidence of both the forward and reverse associations. This helps mitigate cases where high confidence is skewed by the frequency imbalance between antecedents and consequents. Similarly, the **jaccard** index quantifies the similarity between the antecedent and consequent itemsets in terms of co-occurrence, providing

insight into how frequently these items appear together relative to their total support. [9]

Overall, the top rules reveal recurring item combinations, such as dairy products being commonly associated with meat or baked goods. These results reflect real-world shopping tendencies and validate the efficacy of association rule mining in extracting actionable insights from transactional data.

ACKNOWLEDGMENT

I would like to sincerely thank Professor Stefan Henkler and Professor Martin Hirsch for their invaluable guidance, support, and encouragement throughout this valuable subject. Their expertise and insightful feedback greatly added to the quality and direction of this work. I am deeply thankful for the opportunities and resources provided during this subject.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] A. Bharathi and S. Nandhini, "A survey on association rule mining with genetic algorithm," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, Kovilpatti, India, 2016, pp. 1–4. doi: 10.1109/ICCTIDE.2016.7483396.

[3] M. Al-Maolegi and B. Arkok, "An improved Apriori algorithm for association rules," *arXiv preprint* arXiv:1403.3948, Mar. 2014. doi:10.48550/arXiv.1403.3948.

[4] H. Dedhia, "Groceries Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset

[5] M. Waskom, "Seaborn: Statistical data visualization," Seaborn, 2024. [Online]. Available: https://seaborn.pydata.org/

[6] R. Fabrino Mendonça and M. T. Ovalle, "Mining architectural patterns using association rules," in *Proceedings of the 25th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, Pittsburgh, PA, USA, July 2013, pp. 375–380.

[7] C. Martínez, A. Viola, and J. Wang, "Fast and accurate similarity estimation via sampling," *arXiv preprint arXiv:2308.13228*, 2023. [Online]. Available: https://www.researchgate.net/publication/373434197$_{F}ast_{a}nd_{A}ccurate_{S}imilarity_{E}stimation_{v}ia_{S}amplingR.FabrinoMendonçaandM.T.Ovalle$, "$Mininga$ -380.

[9] C. Martínez, A. Viola, and J. Wang, "Fast and accurate similarity estimation via sampling," in *Lecture Notes in Computer Science, Similarity Search and Applications – SISAP 2023*, A Coruña, Spain, Oct. 2023, pp. 56–63.