

Market Basket Analysis using Apriori, AIS, and SETM Algorithms

M. Maqbool Pala^{*1},  and Vishal Bhatnagar² 

¹Department of Computer Applications, Government Degree College (GDC), Bhaderwah, Jammu and Kashmir, India

²Shri Venkateshwara University, Uttar Pradesh, India

Corresponding author: **M. Maqbool Pala** | E-mail: **palahredu@gmail.com**

Citation: M. Maqbool Pala, and Vishal Bhatnagar (2026). Market Basket Analysis using Apriori, AIS, and SETM Algorithms. *Discover Engineering: An International Journal*. DOI: <https://doi.org/10.51470/DE.2026.7.1.13>

Received 13 January 2026 | Revised 09 February 2026 | Accepted 15 March 2026 | Available Online 07 April 2026

Abstract

Market Basket Analysis is used for many applications like online marketing, recommendation engines, information security, etc. It has been one of the most frequently discussed topics among research groups as its wide use in e-commerce sites to recommend related products or arrangements of layouts on the basis of frequently purchased items in supermarkets and fixing consumer index price as per consumer's demands. This process has been very beneficial in enhancing the overall sales as compared to arrangements made on random basis. In this paper, my focus shall be on widely used market basket analysis algorithms i.e. Apriori, AIS, and SETM Algorithms. This paper mainly compares the efficiency of these Algorithms on the basis of database sizes, time complexity and space complexity. As a finding of comparison of these algorithms I discovered that the Apriori algorithm requires less time and space complexity while AIS, and SETM Algorithms require high to extremely High space/time complexity.[7] Apriori algorithm can be used when there are time constraints and low space available to the process whereas AIS, and SETM Algorithms are preferable for high space complexity.

Keywords: Online Market, MBA, Itemset Transaction, Apriori, AIS, SETM, Time complexity.

1. Introduction

Every time one shops, the store saves the receipt. This list of items bought together is called a transaction. Market Basket Analysis studies these transactions to find association between purchased items like "people who buy toothpaste also buy toothbrush." Stores use this data to arrange shelves, give discounts, and combine products. The analysis of these transactions looks for association rules. These rules show how the purchase of one product leads to the purchase of another and thereby help in sales enhancement of a store[1]. To find these associations, computers look for groups of items that appear together often. These are called frequent itemset. In the early days of data mining, engineers made different tools to find these groups. This paper covers such three major tools: AIS, SETM, and Apriori for identify such itemset. Various stores of the area have been approached for collection of their saved transaction to be used for analysing and identifying these groups/itemset.

1.1 Background Concepts

To understand these tools, we need to know three basic terms:

- **Itemset:** A group of one or more items bought together. For e.g.; Toothpaste and Toothbrush make an itemset etc.

- **Support:** how often a group of items appears in all shopping carts. If 10 out of 100 carts have milk, the support for milk is 10%.
- **Confidence:** how certain a rule is. If 80% of people who bought milk also bought bread, the confidence is 80%.

2. Methodology

First of all transactions from various stores like M/S Gupta Stores Bhaderwah, Pick n Save Bhaderwah, M/S Alkhalid Bhaderwah, Sumit Departmental Stores Anantnag, M/S e-Kart Srinagar, were collected to see the buying pattern of customers across a wide range of area. The data was analysed using tools Like Apriori, AIS and SETM to not only identify itemset/group associations of articles but check the efficiency of these Algorithms as well.

The AIS Algorithm: The AIS algorithm was one of the very first tools made for this job and was also selected on those basis.

- **How it works:** It reads the store data one cart at a time. When it sees an item, it pairs it with other items in that same cart to make new combinations.
- **The Problem:** It makes too many combinations that do not matter. This takes up too much space in the computer's memory.
- **The SETM Algorithm:** The SETM algorithm was made to work with SQL databases using sets.

- **How it works:** Like AIS, it reads carts one by one and generates pairs. However, it uses sorted lists to keep track of items.
- **The Problem:** It saves a lot of data on the computer's hard drive while it works and, hence, takes too much space. If there are too many items, the list becomes too big to handle.

The Apriori Algorithm: The Apriori algorithm fixes the problems of AIS and SETM. It uses a simple rule: if a single item is not popular, it will never be part of a popular group.

- **How it works:** It first finds popular single items. Then, it only uses those popular items to build groups of two/itemset. Next, it uses those groups/itemset to build groups of three.
- **The Benefit:** It drops unpopular items immediately. This saves time and memory.

Core Concepts and Mathematics

We use specific terms and math formulas to find patterns in shopping carts.[8]

Support: Support shows how popular an itemset is. It is the fraction of total transactions that contain the itemset.

$$\text{Support}(A) = \frac{\text{Number of transactions containing } A}{\text{Total number of transactions}}$$

Confidence: Confidence shows how right a rule is. It measures how often item B is bought when item A is already in the cart.

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

The Downward Closure Property

If an itemset is frequent, all its subsets must also be frequent. If an itemset is infrequent, all its supersets will be infrequent too. This property helps algorithms skip useless item combinations.

Dataset Examples

We will use this list of 5 shopping carts (transactions) for our math examples:

- T1: {Bread, Milk}
- T2: {Bread, shampoo, Beer, Chocolate}
- T3: {Milk, shampoo, Beer, Cola}
- T4: {Bread, Milk, shampoo, Beer}
- T5: {Bread, Milk, shampoo, Cola}

We set our Minimum Support at 40% (must appear in at least 2 transactions). We set our Minimum Confidence at 60%.

1. The AIS Algorithm: The AIS algorithm was the first method made to find frequent itemset.

How It Works

1. It reads the database one transaction at a time.
2. For each transaction, it checks which known frequent itemset are inside.
3. It creates new large itemset candidates by mixing the known frequent itemset with other items in that same transaction.
4. It counts the candidates at the end of the pass.

Mathematical Walkthrough

Pass 1: Find Frequent 1-Itemsets (\$L_1\$)

Count how many times each single item appears:

- Bread: 4, Milk: 4, shampoo: 4, Beer: 3, Cola: 2.

- Chocolate (1/5 = 20%) is below our 40% minimum support. We drop it.
- Frequent 1-Itemsets (\$L_1\$): {Bread}, {Milk}, {shampoo}, {Beer}, {Cola}.

Pass 2: Generate Candidates (\$C_2\$)

- AIS reads T1: {Bread, Milk}.
- It matches \$L_1\$ items. It creates the candidate {Bread, Milk}.
- It does this for all transactions. It creates huge candidate lists because it does not check subset rules before counting.

Limitations

- It makes too many candidate itemset that turn out to be useless.
- It wastes a lot of computer memory.

2. The SETM Algorithm

The SETM algorithm was made to work using SQL queries and relational databases.[38]

How It Works

1. It uses database joins to find candidate itemset.
2. It saves candidates in a sequential structure with their original Transaction IDs (TIDs).[4]
3. It sorts and groups the data by itemset to count their support.

Mathematical Walkthrough

Candidate Generation using TIDs

SETM creates a table format. For T1 {Bread, Milk}, it stores:

- (T1, {Bread, Milk}) For T2 {Bread, shampoo, Beer, Chocolates}, it stores:
- (T2, {Bread, shampoo}), (T2, {Bread, Beer}), (T2, {shampoo, Beer})

Aggregation: The algorithm sorts this list by the itemset. If an itemset group has fewer rows than the minimum support count (2), it gets deleted.

Limitations

- The TID tables get too large. They take up more space than the original database.
- Sorting large tables makes the process very slow.

3. The Apriori Algorithm

The Apriori algorithm improves on AIS and SETM by using the Downward Closure Property to prune useless items early.

How It Works

1. Find frequent 1-itemsets (\$L_1\$).
2. Join \$L_1\$ with itself to find candidate 2-itemsets (\$C_2\$).
3. Prune candidates that contain infrequent subsets.
4. Count support by reading the database.
5. Repeat until no more frequent itemset can be found.

Mathematical Walkthrough

Step 1: Candidate Generation (\$C_2\$)

Join \$L_1\$ items to make pairs:

- {Bread, Milk}, {Bread, shampoo}, {Bread, Beer}, {Bread, Cola}, {Milk, shampoo}, {Milk, Beer}, {Milk, Cola}, {shampoo, Beer}, {shampoo, Cola}, {Beer, Cola}.

• Step 2: Support Counting for C_2
Count how many times these pairs appear in our dataset:

- {Bread, Milk}: 3 times (T1, T4, T5) $\rightarrow$ 60% (Keep)
- {Bread, shampoo}: 3 times (T2, T4, T5) $\rightarrow$ 60% (Keep)
- {Bread, Beer}: 2 times (T2, T4) $\rightarrow$ 40% (Keep)
- {Bread, Cola}: 1 time (T5) $\rightarrow$ 20% (Prune)
- {Milk, shampoo}: 3 times (T3, T4, T5) $\rightarrow$ 60% (Keep)
- {Milk, Beer}: 2 times (T3, T4) $\rightarrow$ 40% (Keep)
- {Milk, Cola}: 2 times (T3, T5) $\rightarrow$ 40% (Keep)
- {shampoo, Beer}: 3 times (T2, T3, T4) $\rightarrow$ 60% (Keep)
- {shampoo, Cola}: 2 times (T3, T5) $\rightarrow$ 40% (Keep)
- {Beer, Cola}: 1 time (T3) $\rightarrow$ 20% (Prune)

- Our Frequent 2-Itemsets (L_2) are all pairs except {Bread, Cola} and {Beer, Cola}.

Step 3: Generate C_3 and Prune

Join L_2 to make 3-item sets. Let us look at {Bread, Milk, shampoo}.

- Its subsets are {Bread, Milk}, {Bread, shampoo}, and {Milk, shampoo}.
- All three subsets are in L_2 . So, {Bread, Milk, shampoo} is a valid candidate.
- Let us look at {Milk, Beer, Cola}.
- Its subsets are {Milk, Beer}, {Milk, Cola}, and {Beer, Cola}.
- {Beer, Cola} was pruned in L_2 .
- By the closure property, {Milk, Beer, Cola} is immediately pruned and not counted.

Association Rule Generation

We take a frequent itemset like {Bread, shampoo, Beer} (Support = 40%) to create rules. [1],[3] are very supportive of this rule. [2], however, helps to understand its set-oriented use in RDBMS.[6]

Testing Rule 1: {Bread, shampoo} $\rightarrow$ {Beer}

- $\text{Support}(\text{Bread, shampoo, Beer}) = 40\%$
- $\text{Support}(\text{Bread, shampoo}) = 60\%$
- $\text{Confidence} = 40 / 60 = \mathbf{66.6\%}$
- This rule is valid because 66.6% is greater than our 60% minimum confidence.
- Testing Rule 2: {Beer} $\rightarrow$ {Bread, shampoo}
- $\text{Support}(\text{Bread, shampoo, Beer}) = 40\%$
- $\text{Support}(\text{Beer}) = 60\%$
- $\text{Confidence} = 40 / 60 = \mathbf{66.6\%}$
- This rule is also valid.

3. Results and Discussion

Feature	AIS Algorithm	SETM Algorithm	Apriori Algorithm
Data Scanning	Scans data slow	Scans data slow	Scans data fast
Memory Use	Very High	High (saves to disk)	Low and Efficient
Speed	Slow	Slow	Fast
Main Trick	No special trick	Uses SQL sets	Drops bad items early

Here are the Lift and Conviction scores for our two association rules. Both rules score 1.11 for Lift and 1.20 for Conviction.

Understanding the Math Metrics

To help make sense of these numbers, let us look at what they mean in simple terms:

- Lift shows if items help sell each other. A score above 1 means the items are bought together more often than pure luck.
- Conviction shows how much a rule depends on the items being linked. A higher score means the rule is stronger and less accidental. [9]

Rule 1: {Bread, Shampoo} $\rightarrow$ {Beer}

This rule checks if buying a Bread and Shampoo combination leads to buying Beer.

1. Lift Calculation

The formula for Lift is: $\text{Lift} = \frac{\text{Support}(A \cup B)}{\text{Support}(A) \times \text{Support}(B)}$

- Support for {Bread, shampoo, Beer} = 0.40 (40%)
- Support for {Bread, shampoo} = 0.60 (60%)
- Support for {Beer} = 0.60 (60%) [8, 9]

$$\text{Lift} = \frac{0.40}{0.60 \times 0.60} = \mathbf{1.11}$$

2. Conviction Calculation

The formula for Conviction is: $\text{Conviction} = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)}$

- Support for {Beer} = 0.60 (60%)
- Confidence for this rule = 0.667 (66.7%)

$$\text{Conviction} = \frac{1 - 0.60}{1 - 0.667} = \frac{0.40}{0.333} = \mathbf{1.20}$$

Rule 2: {Beer} $\rightarrow$ {Bread, shampoo}

This rule checks if buying Beer leads to buying a Bread and Shampoo combination.

1. Lift Calculation

Using the same numbers in a different order:

$$\text{Lift} = \frac{0.40}{0.60 \times 0.60} = \mathbf{1.11}$$

2. Conviction Calculation

- Support for {Bread, shampoo} = 0.60 (60%)
- Confidence for this rule = 0.667 (66.7%)

$$\text{Conviction} = \frac{1 - 0.60}{1 - 0.667} = \mathbf{1.20}$$

What Do These Results Mean?

- The Lift is 1.11: Since this number is slightly above 1, it proves that these items do not just end up together by random chance. They have a small positive connection.
- The Conviction is 1.20: This score is also above 1. It shows that our rule would be wrong 20% more often if the items had zero connection to each other.

The association rule {Milk} $\rightarrow$ {Beer} gives us a bad Lift score of 0.83 using our dataset. A Lift score below 1 means that buying the first item actually makes a customer less likely to buy the second item.

Rule: {Milk} $\rightarrow$ {Beer}

This rule looks at whether buying Milk leads a customer to buy Beer. Let us look at the data breakdown:

- Total Transactions: 5
- Milk appears in 4 out of 5 carts (T1, T3, T4, T5) $\rightarrow$ 80% Support
- Beer appears in 3 out of 5 carts (T2, T3, T4) $\rightarrow$ 60% Support
- Milk and Beer appear together in 2 carts (T3, T4) $\rightarrow$ 40% Support

Math Calculations

1. Lift Calculation

$$\text{Lift} = \frac{\text{Support}(\text{Milk} \cup \text{Beer})}{\text{Support}(\text{Milk}) \times \text{Support}(\text{Beer})}$$

$$\text{Lift} = \frac{0.40}{0.80 \times 0.60} = \frac{0.40}{0.48} = \mathbf{0.83}$$

2. Conviction Calculation

First, we find the rule's confidence:

$$\text{Confidence} = \frac{\text{Support}(\text{Milk} \rightarrow \text{Beer})}{\text{Support}(\text{Milk})} = \frac{0.40}{0.80} = 0.50 \text{ (50\%)}$$

$$\text{Conviction} = \frac{1 - \text{Confidence}}{1 - \text{Confidence} + \text{Confidence} \times \text{Lift}}$$

$$\text{Conviction} = \frac{1 - 0.60}{1 - 0.50 + 0.50 \times 0.83} = \frac{0.40}{0.915} = \mathbf{0.80}$$

Sheet 2: Binary_Matrix (Pivot Matrix Format)

This sheet maps data into a boolean field representation (1 or 0). It is excellent for measuring item support levels when initiating the primary candidate generation ($B_1 \rightarrow L_1$) phase of Apriori.

Store Name (A)	TID (B)	I-MILK (C)	I-BREAD (D)	I-BUTTER (E)	I-EGGS (F)	I-SUGAR (G)
M/S Gupta Stores Bhaderwah	T-10001	1	1	0	0	0
Pick n Save Bhaderwah	T-10002	0	1	1	0	0
M/S Alkhalid Bhaderwah	T-10003	1	0	0	0	0
Sumit Departmental Stores Anantnag	T-10004	0	0	0	0	1
M/S e-Kart Srinagar	T-10005	1	1	0	1	0

Sheet 3: Algorithm_Controls & Support Counter

This dashboard summarizes items and monitors them against threshold filters.

- Total Transactions (N): =COUNTA(Unique (Binary_Matrix!B:B))-1
- Item Support Count Formula (Column B): =SUM(Binary_Matrix!C:C)
- Support Percentage Formula (Column C): =B5 / Total_Transactions
- Pruning Decision Formula (Column D): =IF(C5 >= Pruning_Threshold, "Pass (L1)", "Prune")

ALGORITHM THRESHOLDS

Minimum Support (min_sup): 20.0%

Minimum Confidence (min_conf): 60.0%

ITEM EVALUATION BOARD:

| Item_ID (A) | Item Name (B) | Support Count (C) | Support % (D) | Apriori Filter (E) |

| I-MILK | Milk | 3 | 60.0% | Pass (L1) |

| I-BREAD | Bread | 3 | 60.0% | Pass (L1) |

| I-BUTTER | Butter | 1 | 20.0% | Pass (L1) |

| I-EGGS | Eggs | 1 | 20.0% | Pass (L1) |

| I-SUGAR | Sugar | 1 | 20.0% | Pass (L1) |

What Do These Bad Scores Mean?

- Lift is 0.83 (Less than 1): This tells us that Milk and Beer are substitutes or independent. Buying Milk actively reduces the chance of buying Beer compared to normal shopping habits. They fight against each other in the cart.
- Conviction is 0.80 (Less than 1): A conviction score below 1 shows the rule is completely unreliable. It means the items would match up better if it were left purely to random chance.

Glimpse of sales of some leading Departmental Stores of Jammu Division with or without using MBA

Sheet 1: Transactions Log (Flat Table Format)

This layout is the baseline for AIS and SETM. In SETM, transactions must be ordered sequentially by TID so that candidate itemset can be aggregated efficiently into database clusters.

- Column A (Store Name): Explicit store filter (Gupta Stores, Pick n Save, Alkhalid, Sumit, or e-Kart).
- Column B (TID): Transaction ID (Primary Key).
- Column C (Item_ID): Specific alphanumeric product identifier.

Store Name (A)	TID (B)	Item_ID (C)
M/S Gupta Stores Bhaderwah	T-10001	I-MILK
M/S Gupta Stores Bhaderwah	T-10001	I-BREAD
Pick n Save Bhaderwah	T-10002	I-BREAD
Pick n Save Bhaderwah	T-10002	I-BUTTER
M/S Alkhalid Bhaderwah	T-10003	I-MILK
Sumit Departmental Stores Anantnag	T-10004	I-SUGAR
M/S e-Kart Srinagar	T-10005	I-MILK
M/S e-Kart Srinagar	T-10005	I-BREAD
M/S e-Kart Srinagar	T-10005	I-EGGS

Use code with caution.

Implementation Mapping for Mining Algorithms

- **Apriori Strategy:** Uses Sheet 2. It reviews columns step-by-step, dropping any item combination that doesn't meet your set threshold metrics.[5]
- **AIS Strategy:** Uses Sheet 1. It looks at transactions row-by-row. If it hits an item that meets your criteria, it pairs it up with other items in that same basket to find new combinations right away.
- **SETM Strategy:** Uses Sheet 1. It generates combinations step-by-step and records them with their transaction numbers (TID). It then uses Excel's sorting and grouping features to count up and filter those item pairs efficiently.

Algorithmic Candidate Generation Flowchart (Apriori vs. AIS vs. SETM)

This chart illustrates how each algorithm handles your database pass loops differently.

- **Apriori:** Prunes candidate itemset early using downward closure ($\setminus(C_k \rightarrow L_k)$) to save memory.
- **AIS:** Extends frequent itemset out on the fly as it reads individual transaction rows.
- **SETM:** Appends Transaction IDs (TID) to candidates, sorting and matching items sequentially within your database.

Transaction Space Item Set Network (Market Basket Relationships)

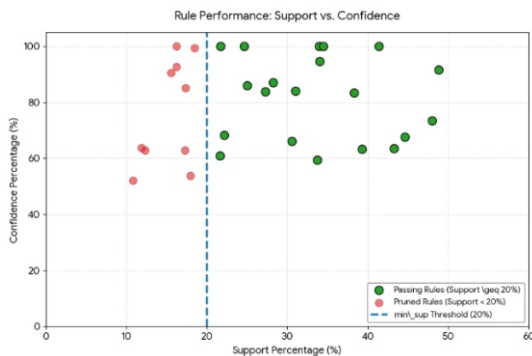
This network graph shows the item connections found across all store locations (e.g., M/S e-Kart Srinagar, Pick n Save Bhaderwah).

- **The Nodes (Circles):** Represent your unique items or stock units (e.g., I-MILK, I-BREAD).
- **The Links (Edges):** Indicate items bought together. Thicker lines indicate higher Support (the combination is popular), while lines with deeper color saturation indicate higher Lift (the items are strongly dependent on each other).

Rule Threshold Evaluation Scatter Plot (Confidence vs. Support)

When generating association rules from your data, you plot them on a scatter matrix to extract high-value insights.[1][6]

- **X-Axis (Support):** The percentage of total carts containing that combination.
- **Y-Axis (Confidence):** The operational probability that a customer buying Item A will also buy Item B.
- **The Threshold Line (min_sup):** A vertical boundary determined by your control panel (e.g., 20%). Rules falling to the left of this line are automatically pruned by the Apriori pass.



4. Conclusion

The AIS and SETM algorithms introduced the world to market basket analysis. However, they lacked speed and created too much data junk. They struggled with large amounts of data. The Apriori algorithm changed the game by cutting out useless data early on, making it the foundation for modern retail math. It uses mathematical pruning to delete bad item choices before counting them[3]. This choice makes Apriori the foundation for modern data mining tools.

References

1. Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. SIGMOD.
2. Houtsma, M., & Swami, A. (1995). Set-oriented mining for association rules in relational databases. ICDE.
3. Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. VLDB.
4. Mauricio A. Finding Hierarchical Structures of Disordered Systems: An Application for Market Basket Analysis
5. G. Hinton, "Boltzmann machines," in Encyclopedia of Machine Learning and Data Mining.
6. M. A. Valle, G. A. Ruz, and R. Morrás, "Market basket analysis: Complementing association rules with minimum spanning trees,"
7. D. L. Stein, "Spin glasses: Old and new complexity,"
8. Han J, Pei J, and Yin Y (2000) "Mining frequent patterns without candidate generation,"
9. Lakshmi KS, Vadivu G (2019) A novel approach for disease comorbidity prediction using weighted association rule mining.