

Indian Institute of Technology Madras
ZANZIBAR CAMPUS

Z3003 · DATA MINING: SCALING TO LARGE DATA

# Frequent Itemsets and the Apriori Algorithm

*Lecture 13 · Wednesday 14 October 2026*

**Dr. Innocent Nyalala** | BS Semester 5 (DSAI) | IIT Madras Zanzibar

# Session Outline

IIT Madras Zanzibar

1. (0–5 min) Recap: support, confidence, lift; the itemset explosion
2. (5–15 min) The Apriori principle — one sentence that prunes almost everything
3. (15–40 min) Apriori by hand on 9 baskets: join, prune, count, repeat
4. (40–52 min) From frequent itemsets to association rules
5. (52–62 min) Apriori on our 120,000 baskets: what it found, how much it pruned
6. (62–75 min) Exercise, summary, questions

# Learning Outcomes

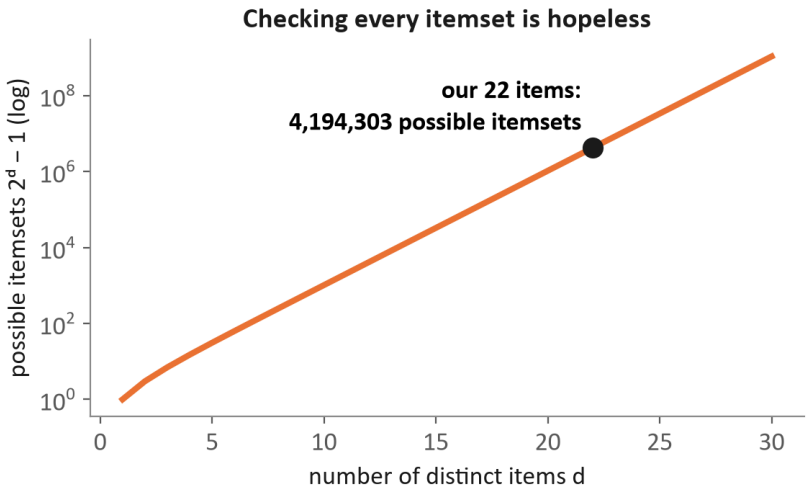
IIT Madras Zanzibar

*By the end of this session, you will be able to:*

| | |
|---|---|
| **State** | the Apriori principle and explain why it is true |
| **Execute** | Apriori by hand: candidates, pruning, counting, level by level |
| **Generate** | association rules from a frequent itemset and filter them by confidence |
| **Count** | how many candidates Apriori saves compared with brute force |
| **Interpret** | Apriori's output on real data, using lift to reject illusions |

# Recap: Three Measures and One Problem

| Measure | Formula | rice seed → urea |
|---|---|---|
| support | baskets with A ∪ B / all | 13,043 / 120,000 = 10.9% |
| confidence | support(A ∪ B) / support(A) | 13,043 / 16,889 = 77.2% |
| lift | confidence / support(B) | 0.772 / 0.114 = 6.80 |




**The problem:** 22 items → 4,194,303 possible itemsets, each needing a pass over 120,000 baskets.

**Today's question:** can we find every frequent itemset WITHOUT checking all 4 million?

# Frequent Itemsets and the Apriori Algorithm

→ **Foundations: The Problem Restated**

The Apriori Principle

Apriori Step by Step (By Hand)

From Itemsets to Rules

Apriori on 120,000 Baskets

In-Class Exercise

# The Task, Precisely

| Symbol | Meaning | Example |
|---|---|---|
| min_sup | minimum support (analyst's choice) | 5% (6,000 baskets) |
| k-itemset | an itemset with k items | {rice seed, urea} is a 2-itemset |
| $C_k$ | candidate k-itemsets we will count | $C_2$ = pairs worth checking |
| $L_k$ | frequent k-itemsets found | $L_2$ = pairs with support ≥ 5% |
| min_conf | minimum rule confidence | 50% |

1 **Stage 1 (expensive)**
find all frequent itemsets: support ≥ min_sup

2 **Stage 2 (cheap)**
from each frequent itemset, list rules with confidence ≥ min_conf

**Why stage 1 is hard:** the number of itemsets doubles with every extra item.

# Frequent Itemsets and the Apriori Algorithm

Foundations: The Problem Restated

→ **The Apriori Principle**

Apriori Step by Step (By Hand)

From Itemsets to Rules

Apriori on 120,000 Baskets

In-Class Exercise

# The Apriori Principle

IIT Madras Zanzibar

**The Apriori principle**

Every subset of a frequent itemset is frequent ⇔ every superset of an infrequent itemset is infrequent.

1 **Why it is true**

any basket containing {R, U, H} also contains {R, U} — so support can only fall as items are added

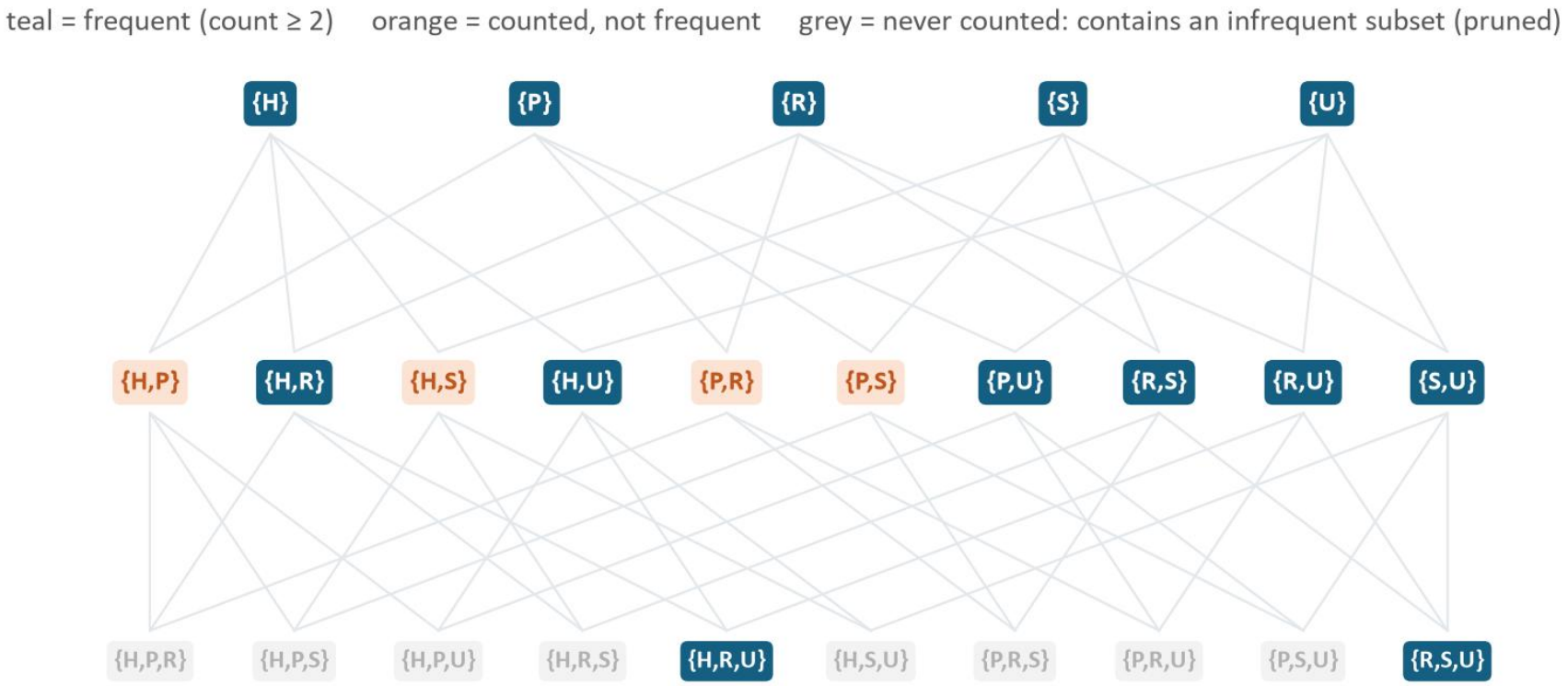
2 **How we use it**

the moment an itemset is infrequent, never count any larger set containing it

$$A \subseteq B \Rightarrow \text{support}(A) \geq \text{support}(B)$$

**Everyday version:** if fewer than 5 students take BOTH Maths and Physics, fewer than 5 can take Maths, Physics AND Chemistry.

# See It: Pruning the Lattice




**The payoff:** every grey itemset was ruled out without reading a single basket.

# Frequent Itemsets and the Apriori Algorithm

Foundations: The Problem Restated

The Apriori Principle

→ **Apriori Step by Step (By Hand)**

From Itemsets to Rules

Apriori on 120,000 Baskets

In-Class Exercise

# Our Worked Example: 9 Baskets

IIT Madras Zanzibar

| basket | items |
|---|---|
| T1 | H, R, U |
| T2 | P, U |
| T3 | S, U |
| T4 | P, R, U |
| T5 | R, S |
| T6 | S, U |
| T7 | R, S |
| T8 | H, R, S, U |
| T9 | R, S, U |

**Items**

R = rice seed · U = urea · H = herbicide · S = sprayer · P = pesticide

! **min_sup = 2 baskets**
= 2 / 9 ≈ 22% support

**Goal:** list every itemset that appears in at least 2 of the 9 baskets — without counting all 31 possible itemsets.

# Level 1: Count Single Items

IIT Madras Zanzibar

| $C_1$ | count | ≥ 2 ? |
|---|---|---|
| R | 6 | ✓ |
| U | 7 | ✓ |
| S | 6 | ✓ |
| H | 2 | ✓ |
| P | 2 | ✓ |

**1** **Pass 1**
read every basket once; count each item

**2** **Keep**
$L_1$ = {R, U, S, H, P} — all reach 2

**On real data:** rare items fail here and vanish — with every pair and triple that contains them.

# Level 2: Join, Then Count Pairs

IIT Madras Zanzibar

| $C_2$ | count | frequent? |
|---|---|---|
| {H,P} | 0 | ✗ |
| {H,R} | 2 | ✓ |
| {H,S} | 1 | ✗ |
| {H,U} | 2 | ✓ |
| {P,R} | 1 | ✗ |
| {P,S} | 0 | ✗ |
| {P,U} | 2 | ✓ |
| {R,S} | 4 | ✓ |
| {R,U} | 4 | ✓ |
| {S,U} | 4 | ✓ |

1. **Join**
   every pair of frequent items: 5 choose 2 = 10 candidates

2. **Pass 2: count**
   count each candidate pair in the 9 baskets

3. **Keep**
   $L_2$ = {H,R}, {H,U}, {P,U}, {R,S}, {R,U}, {S,U}

**Remember the failures:** {H,P}, {H,S}, {P,R}, {P,S}. They prune level 3.

# Level 3: Join, PRUNE, Then Count

| candidate | its 2-subsets | all frequent? | count |
|---|---|---|---|
| {H,R,U} | HR, HU, RU | ✓ keep | 2 |
| {H,R,S} | HR, HS, RS | ✗ prune | — |
| {H,P,U} | HP, HU, PU | ✗ prune | — |
| {H,S,U} | HS, HU, SU | ✗ prune | — |
| {P,R,U} | PR, PU, RU | ✗ prune | — |
| {P,S,U} | PS, PU, SU | ✗ prune | — |
| {R,S,U} | RS, RU, SU | ✓ keep | 2 |

1. **Join**
   7 candidate triples

2. **Prune (no data read)**
   5 contain an infrequent pair → removed

3. **Pass 3: count only 2**
   {H,R,U} = 2 ✓  {R,S,U} = 2 ✓

**Level 4:** {H,R,S,U} contains {H,R,S}, already infrequent → pruned.
No candidates left: stop.
Total frequent itemsets: 5 + 6 + 2 = 13.

# The Apriori Algorithm in Four Lines

```
L₁ = frequent single items                    # pass 1
k = 2
while Lₖ₋₁ is not empty:
    Cₖ = join(Lₖ₋₁)                           # candidates
    Cₖ = prune(Cₖ)                # Apriori principle
    count Cₖ in ONE pass over the baskets
    Lₖ = { c ∈ Cₖ : support(c) ≥ min_sup }
    k = k + 1
return L₁ ∪ L₂ ∪ …
```

1. **Join** — combine frequent (k−1)-itemsets that differ in one item
2. **Prune** — Apriori principle: discard without reading data
3. **Count** — one pass over all baskets per level
4. **Stop** — when no candidates survive

**Cost:** one full pass over the data per level — fine in memory, expensive when the data lives on disk (tomorrow).

# Frequent Itemsets and the Apriori Algorithm

IIT Madras Zanzibar

Foundations: The Problem Restated

The Apriori Principle

Apriori Step by Step (By Hand)

→ **From Itemsets to Rules**

Apriori on 120,000 Baskets

In-Class Exercise

# From a Frequent Itemset to Rules

IIT Madras Zanzibar

| rule A → B | count(A) | confidence = 2 / count(A) | ≥ 50%? |
|---|---|---|---|
| {R,S} → U | 4 | 50% | ✓ |
| {R,U} → S | 4 | 50% | ✓ |
| {S,U} → R | 4 | 50% | ✓ |
| R → {S,U} | 6 | 33% | ✗ |
| S → {R,U} | 6 | 33% | ✗ |
| U → {R,S} | 7 | 29% | ✗ |

**1** **No new passes**
all supports were counted in stage 1

**2** **Split every way**
a k-itemset gives $2^k - 2$ rules

**3** **Filter, then check lift**
keep confidence ≥ min_conf; reject lift ≤ 1

**Shortcut:** moving items from left to right can only lower confidence — if R → {S,U} fails, so does every rule with a smaller left side.

# Frequent Itemsets and the Apriori Algorithm

IIT Madras Zanzibar

Foundations: The Problem Restated

The Apriori Principle

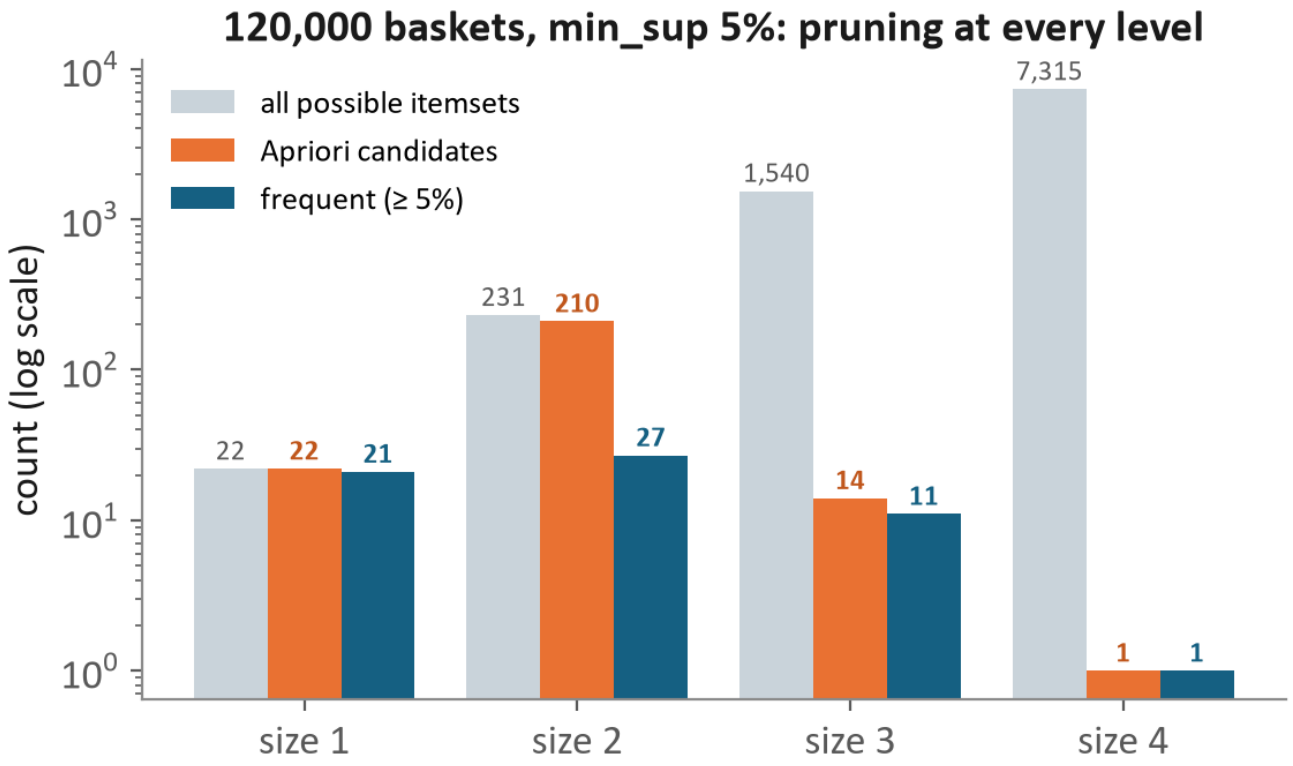
Apriori Step by Step (By Hand)

From Itemsets to Rules

→ **Apriori on 120,000 Baskets**

In-Class Exercise

# Apriori on Our 120,000 Baskets




| level | possible | candidates | frequent |
|---|---|---|---|
| 1 | 22 | 22 | 21 |
| 2 | 231 | 210 | 27 |
| 3 | 1,540 | 14 | 11 |
| 4 | 7,315 | 1 | 1 |

**Result:** 60 frequent itemsets from ≈ 250 counted candidates — not 4 million.

**Biggest saving:** level 3 — 14 candidates instead of 1,540 possible triples.

# What Apriori Found — and What Lift Rejects

IIT Madras Zanzibar

| rule | support | confidence | lift | action |
|---|---|---|---|---|
| NPK fertilizer → cassava cuttings | 6.4% | 95% | **11.5** | bundle them |
| machete → coconut seedlings | 10.1% | 96% | **7.3** | stock together |
| {urea, herbicide} → rice seed | 8.4% | 99% | **7.1** | rice starter kit |
| sprayer → pesticide | 6.3% | 64% | **4.4** | cross-promote |
| clove seedlings → hoe | 11.1% | 55% | **0.92** | ignore: hoe is just popular |

**Lesson again:** confidence ≥ 50% let the hoe rule through; lift < 1 exposes it. Friday: better measures still.

# Apriori in the Real World

IIT Madras Zanzibar

**1 Supermarkets**

bread → butter; shelf layout, promotions

**2 E-commerce**

'frequently bought together' bundles

**3 Agro-dealers**

starter kits: seed + fertilizer + herbicide

**4 Healthcare**

medicines prescribed together; adverse-reaction signals

**5 Web usage**

pages visited together → site navigation

**6 Telecom & mobile money**

services used together → bundle offers

**Common thread:** find what goes together, then keep only rules that beat chance (lift > 1).

# Frequent Itemsets and the Apriori Algorithm

IIT Madras Zanzibar

Foundations: The Problem Restated

The Apriori Principle

Apriori Step by Step (By Hand)

From Itemsets to Rules

Apriori on 120,000 Baskets

→ **In-Class Exercise**

# In-Class Exercise (12 minutes, pairs)

IIT Madras Zanzibar

**IN-CLASS TASK**

Baskets: {A,B,C}, {A,B}, {A,C}, {B,C}, {A,B,C,D}, {B,D}.  min_sup count = 3.

1. Count every single item. Which are frequent ($L_1$)?
2. List $C_2$, count each pair, and give $L_2$.
3. Join $L_2$ into $C_3$. Prune using the Apriori principle. Count what survives.
4. List all rules from your largest frequent itemset with confidence ≥ 60%.

# Solution

IIT Madras Zanzibar

1. **$L_1$**
   A:4, B:5, C:4, D:2 → $L_1$ = {A, B, C} (D fails)

2. **$L_2$**
   AB:3, AC:3, BC:3 → all frequent

3. **$C_3$**
   {A,B,C}: all pairs frequent → count = 2 → NOT frequent (2 < 3)

4. **Rules from {A,B}, {A,C}, {B,C}**
   A→B 3/4 = 75% ✓; B→A 3/5 = 60% ✓; A→C 75% ✓; C→A 75% ✓; B→C 60% ✓; C→B 75% ✓

# Summary & Exam Checkpoints

IIT Madras Zanzibar

**Apriori principle**

Subsets of frequent sets are frequent; supersets of infrequent sets are infrequent.

**Join → prune → count**

Candidates from $L_{k-1}$; drop any with an infrequent subset; one pass to count.

**Rules are cheap**

Split each frequent itemset; confidence = count(itemset) / count(left side). No new passes.

**On our data**

60 frequent itemsets from ≈ 250 candidates. Top rule lift 11.5; hoe rule rejected (lift 0.92).

# Common Questions

IIT Madras Zanzibar

**Q** **Does Apriori ever miss a frequent itemset?**

**A** No. It prunes only sets that are guaranteed infrequent, so the result is exact.

**Q** **How do I choose min_sup?**

**A** Too high: you miss interesting rare patterns. Too low: the candidates explode. Start high and lower it step by step.

**Q** **Why is level 2 the hardest?**

**A** Almost every pair of frequent items becomes a candidate (210 here). Tomorrow's PCY attacks exactly this level.

**Q** **Is a high-confidence rule always useful?**

**A** No — check lift (and Friday's measures). Popular items create misleading rules.

# What's Next

IIT Madras Zanzibar

**Thu** **FP-Growth and limited-pass algorithms**
mining with 2 passes; PCY and SON for data that does not fit in memory

**Tut** **Tutorial 5**
Apriori by hand in code, then mlxtend on 120,000 baskets

**Fri** **Pattern evaluation**
$\chi^2$, all-confidence, Kulczynski, null-invariance

**Mini-project:** dataset approval checkpoint is this Friday, 16 October.

# References & Further Reading


- Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. Proc. VLDB, 487–499.
- Leskovec, J., Rajaraman, A. & Ullman, J. D. Mining of Massive Datasets, 3rd ed. Cambridge — Chapter 6 (frequent itemsets). Course textbook.
- Han, J., Pei, J. & Tong, H. (2023). Data Mining: Concepts and Techniques, 4th ed. — Chapter 4 (pattern mining).