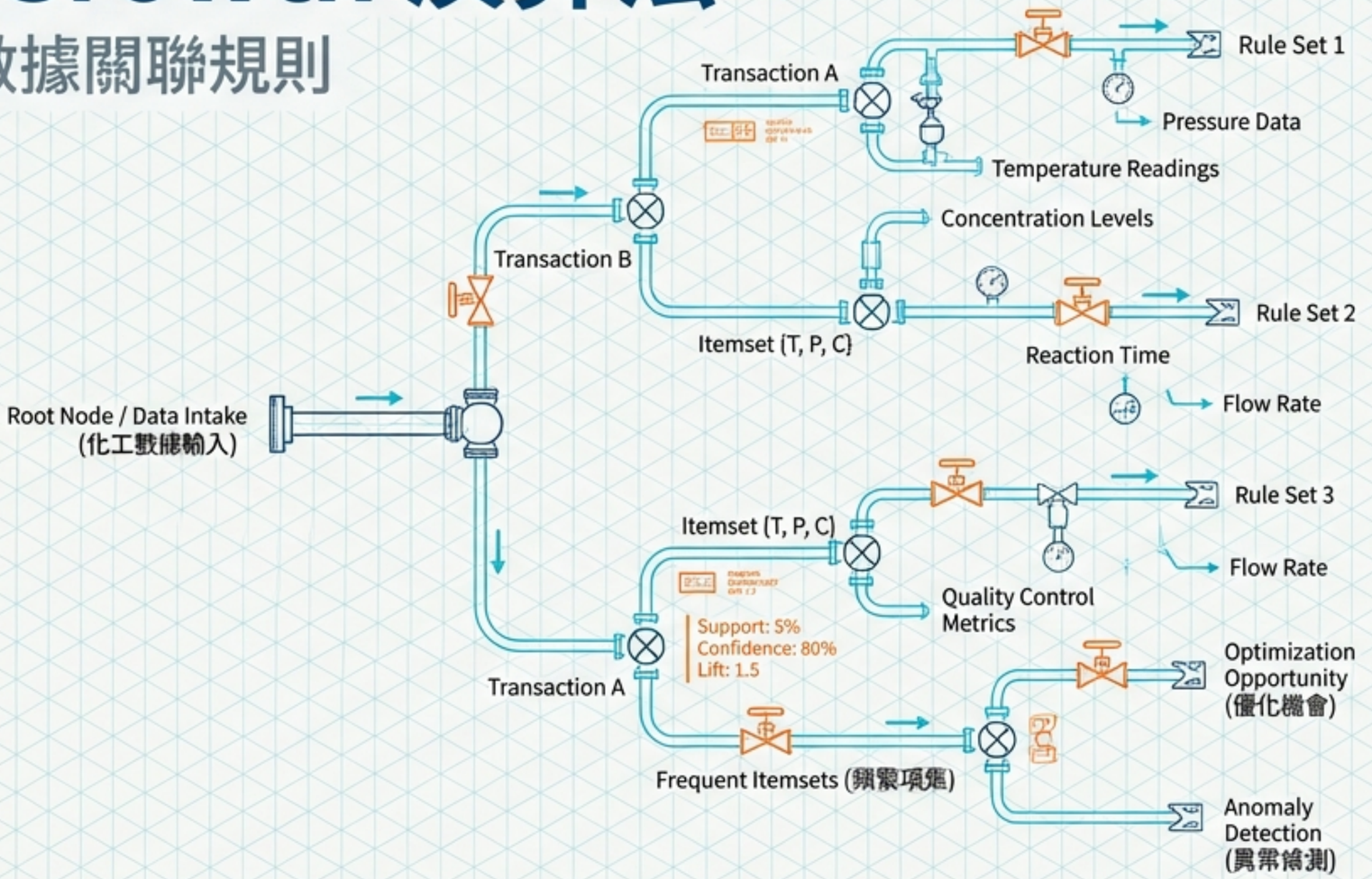
Unit 08: FP-Growth 演算法

高效挖掘大規模化工數據關聯規則

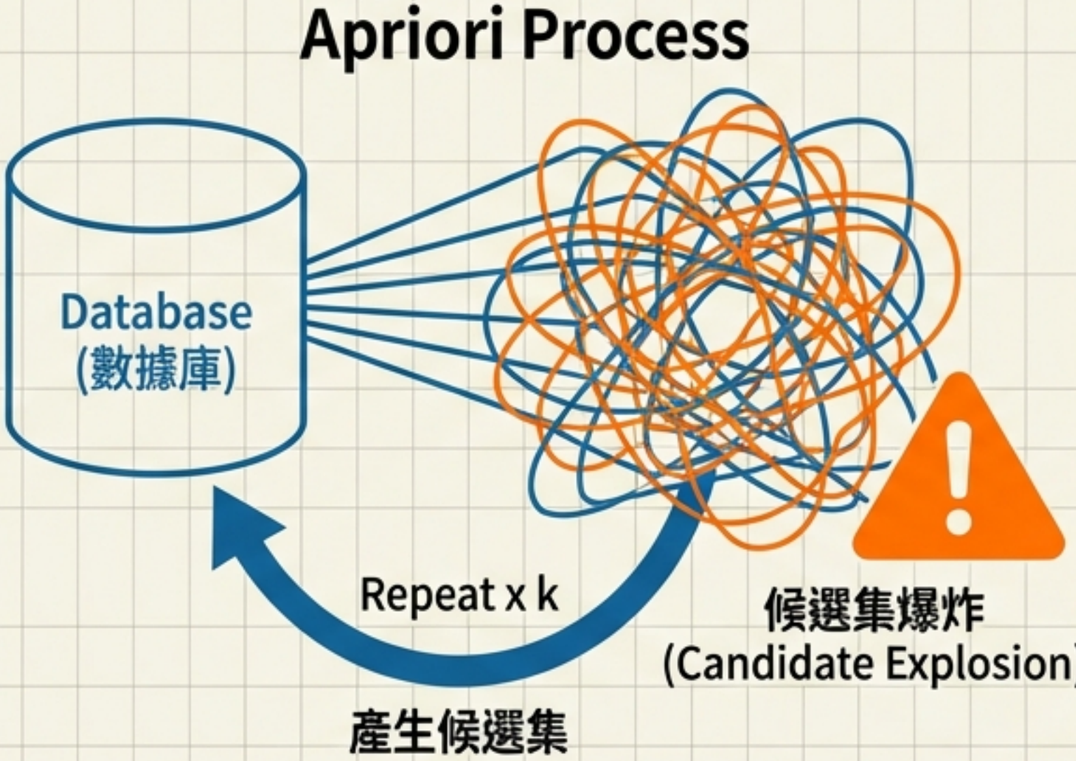


授課教師：莊曜禎 助理教授 | 學期：114學年度第2學期 | 製作單位：逢甲大學 化工系 智慧程序系統工程實驗室

課程地圖與學習路徑

- 核心原理：理解 FP-Tree 資料結構與「分而治之」策略
- 實作技能：使用 Python mlxtend 套件進行大規模運算
- 化工應用：50,000 筆聚合物配方優化案例
- 性能對比：Apriori vs. FP-Growth 效率分析

為什麼 Apriori 遇到瓶頸？



技術瓶頸 (Technical Bottlenecks)

- **多次掃描** (Multiple Scans)
  - 對於 k-項目集，需掃描數據庫 k 次
- **候選集爆炸** (Candidate Explosion)
  - 低支持度下，候選集呈指數級增長
- **內存消耗** (Memory Hog)
  - 維護大量候選集導致 I/O 瓶頸

「Apriori 就像在圖書館找書，每找一本書都要重新把所有書架掃描一次。」

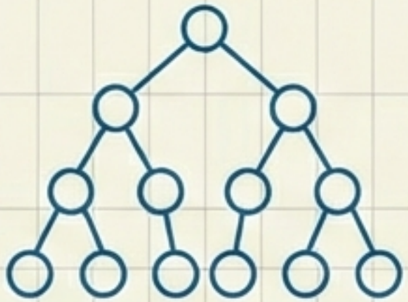
解決方案：FP-Growth 演算法

舊典範 (Old Paradigm) : Apriori

多次掃描 + 大量候選集

VS

新典範 (New Paradigm): FP-Growth



Divide & Conquer
(分而治之)

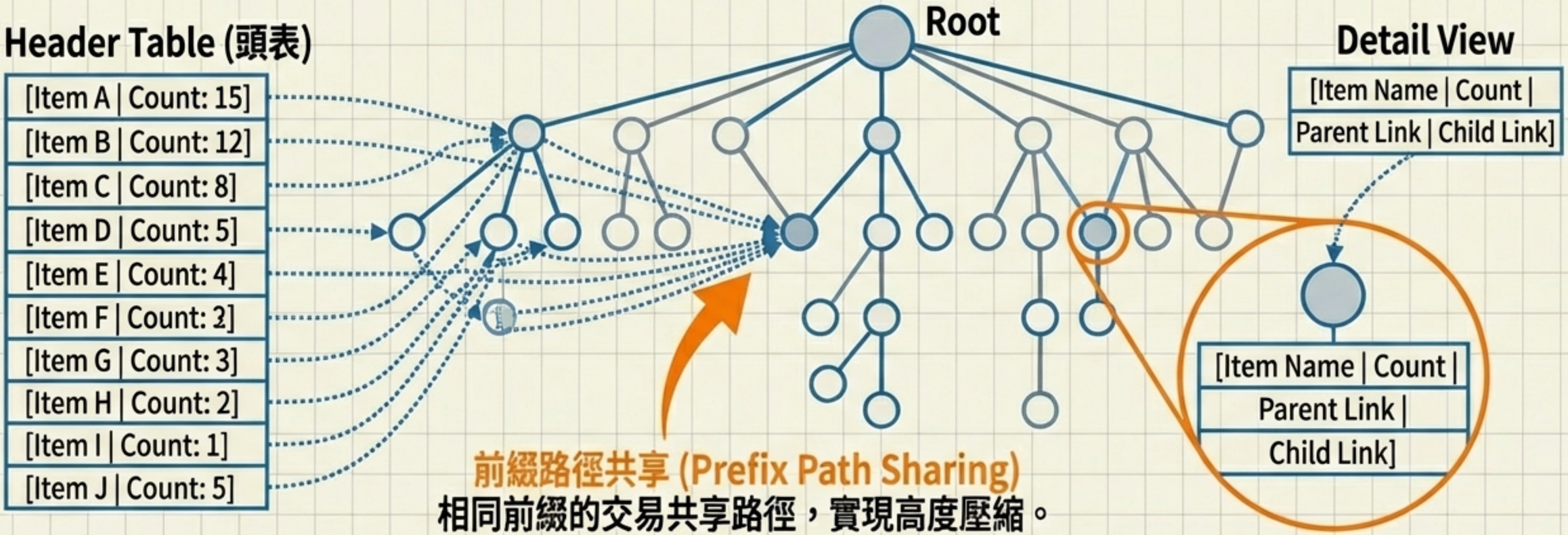
不產生候選集 + 2次掃描

Core Advantages

★ 兩次掃描，搞定一切

1. Scan 1: 統計頻率 (Count)
2. Scan 2: 構建樹結構 (Build Tree)

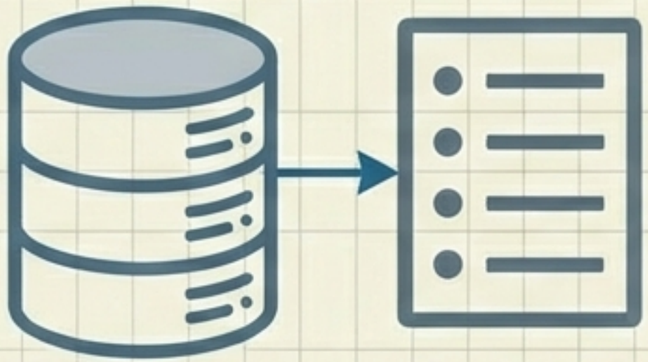
核心引擎：FP-Tree 資料結構



Metaphor: 類似於 Trie (前綴樹)，將大量重複的化工配方壓縮成一棵緊湊的樹。

步驟一：構建 FP-Tree

Step 1: 頻率統計 (Frequency Count)



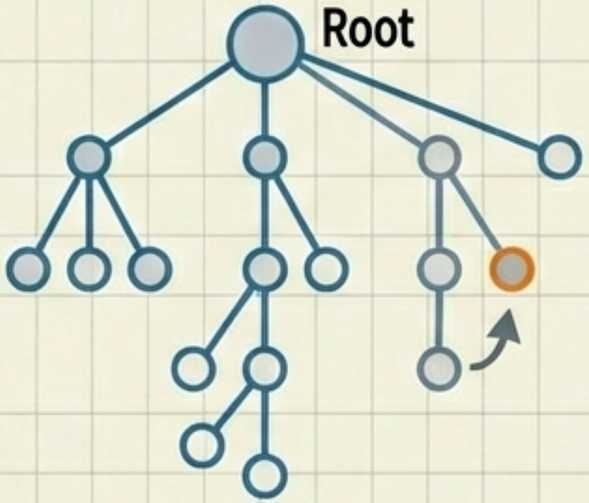
Scan DB, filter by min_support, generate F-List.

Step 2: 重新排序 (Reordering)

Raw:
{催化劑A, 溶劑X, 高溫}

Sorted:
(催化劑A: 0.8) → (溶劑X: 0.8) → (高溫: 0.6)

Step 3: 插入樹中 (Insertion)



Insert sorted items. If path exists → Increment count.

步驟二：頻繁模式挖掘

4. Recursive Mining
(遞歸挖掘)

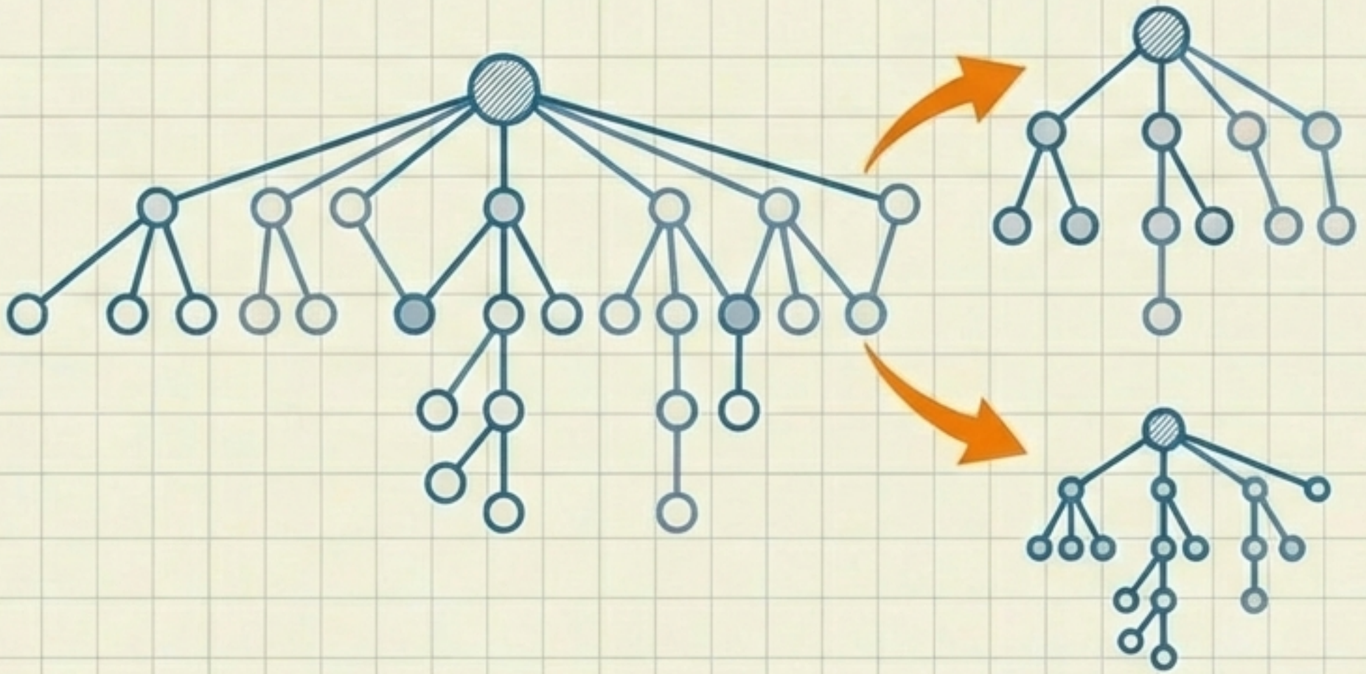
3. Construct Conditional FP-Tree
(構建條件樹)

2. Find Conditional Pattern Base
(找出前綴路徑)

1. Header Table Bottom
(從底部項目開始)

Divide & Conquer (分而治之)

將大數據庫的搜索問題，分解為小樹的遞歸處理。



Python 實作：使用 mlxtend

```python
from mlxtend.frequent_patterns import fpgrowth

# 1. 準備數據 (One-hot Encoding)
# df_encoded = ... (True/False format)

# 2. 執行 FP-Growth
# min_support = 2%
frequent_itemsets = fpgrowth(df_encoded,
                             min_support=0.02,
                             use_colnames=True)

# 3. 結果
print(frequent_itemsets.head())
```

API 與 Apriori 幾乎相同，
輕鬆切換高效引擎。

實戰案例：聚合物配方優化

Project Goal

找出導致「高純度」與「窄分子量分布」的黃金組合。

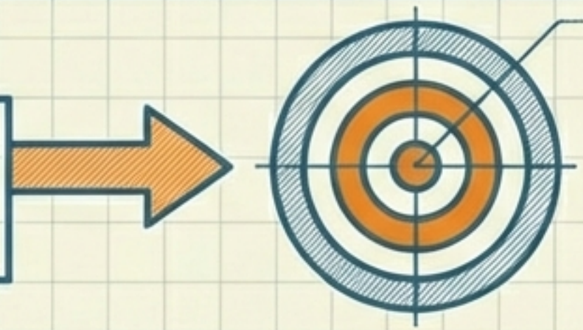
Dataset Specifications

50,000 Historical Recipes
歷史配方

Input Variables

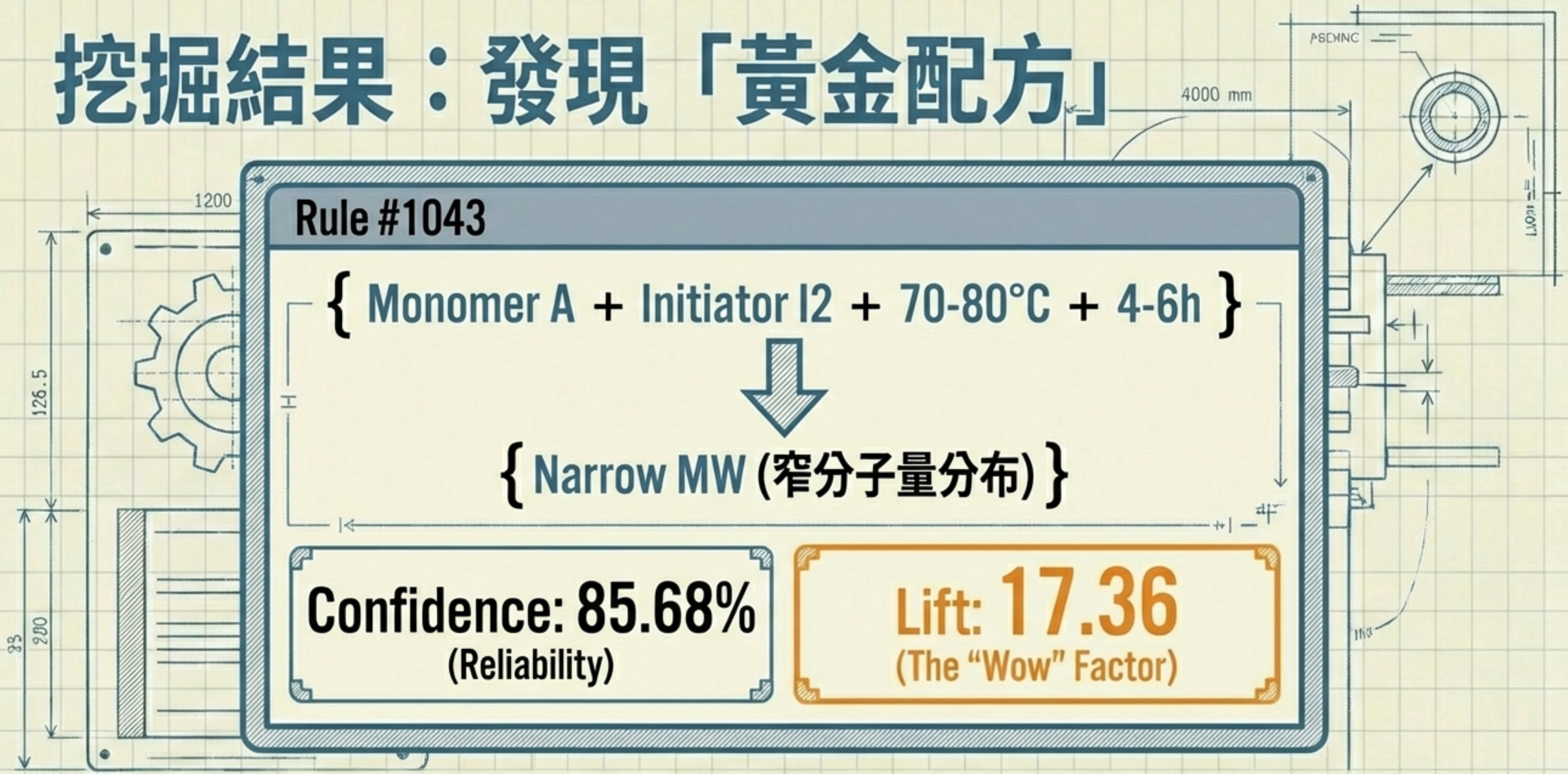
成分 (Ingredients): 單體, 引發劑, 鏈轉移劑, 溶劑

條件 (Conditions): 溫度, 時間



Target:
High Purity
(高純度)

挖掘結果：發現「黃金配方」



Rule #1043

{ Monomer A + Initiator I2 + 70-80°C + 4-6h }

↓

{ Narrow MW (窄分子量分布) }

Confidence: 85.68%
(Reliability)

Lift: 17.36
(The “Wow” Factor)

結論：此組合比隨機猜測高出 17 倍的機率獲得高品質產品。

工程洞察與化學解釋

協同效應 (Synergy)

Chain Transfer Agent + Solvent S2 → High Purity

LIFT VALUE

Lift: 11.74.
揭示了鏈轉移劑與溶劑的特殊化學協同作用。

軟感測器 (Soft Sensor)

Narrow MW 預測
High Conversion

MW INDICATOR

利用分子量分布作為轉化率的早期指標。

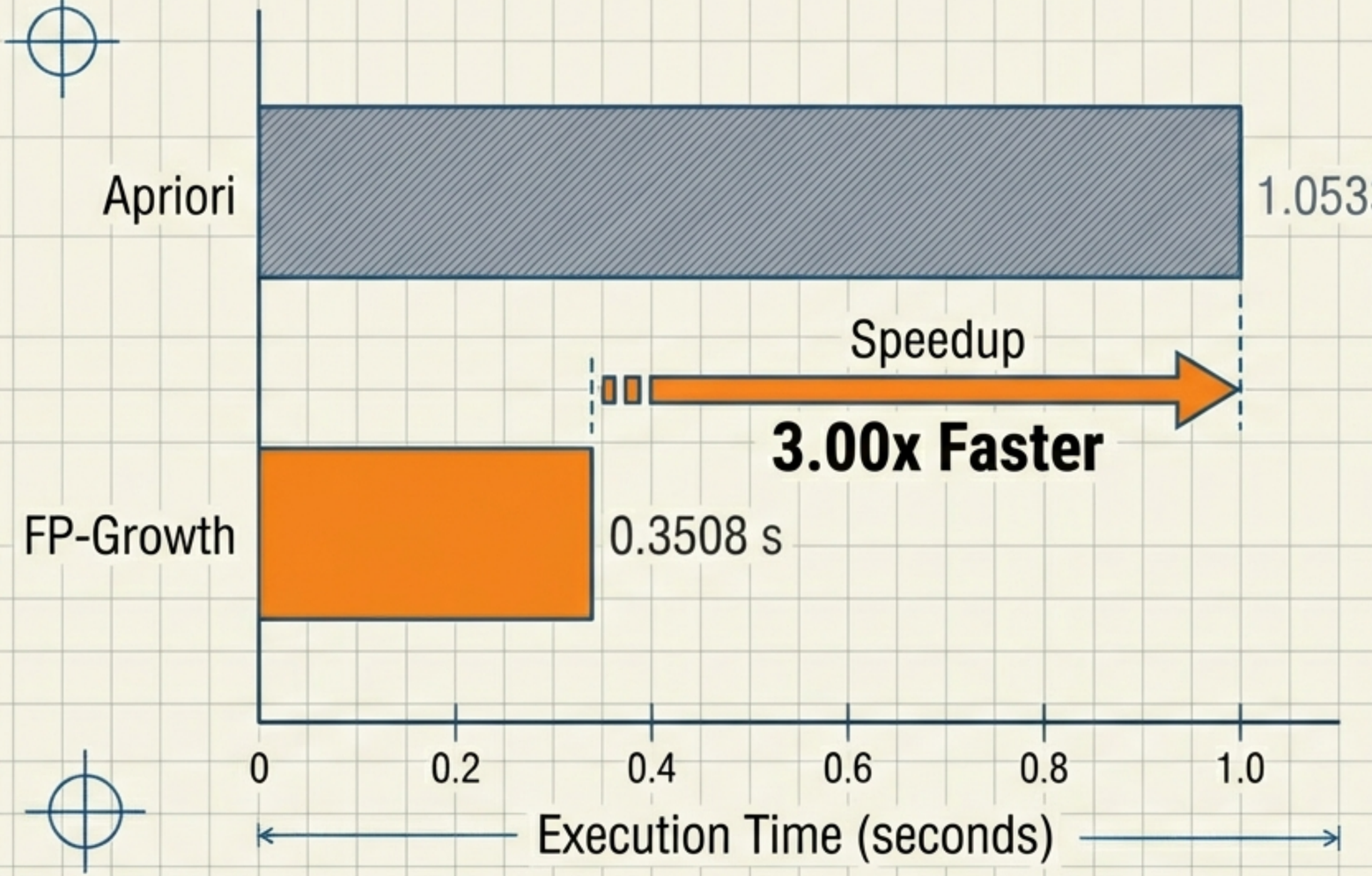
可靠性 (Reliability)

Confidence > 85%

CONFIDENCE LEVEL

高置信度規則可直接轉化為標準作業程序 (SOP)。

效能對決：FP-Growth vs. Apriori



- Conditions: 50,000 recipes, min_support=2%.
- Verdict: 結果完全相同 (100% Accuracy)，但 FP-Growth 在大數據下展現統治級效率。

決策指南：如何選擇演算法？

Data Size & Requirements?

數據大小與需求？

Small Data (< 1k)
OR
Teaching Demo

Big Data (> 10k)
OR
Low Support
OR
Real-time

Use **Apriori**

- 簡單直觀 (Simple)

Use **FP-Growth**

- 極致效能 (High Performance)
- 無候選集 (No Candidates)

單元總結 (Summary)

- **核心結構：FP-Tree** 實現了數據的高效壓縮 (Efficient compression).
- **極致效能：**兩次掃描 + 無候選集生成 (2 Scans + No Candidates).
- **化工價值：**成功應用於大規模配方優化與協同效應發現 (Proven in recipe optimization).
- **未來趨勢：**結合自動化系統進行實時推薦 (Real-time recommendation).

下一步 (Next Steps)

前往 Jupyter Notebook 進行程式演練
Hands-on Practice with mlxtend

「懂原理，更要懂選擇工具。
在大數據時代，效率就是競爭力。」

Unit 08 Completed.