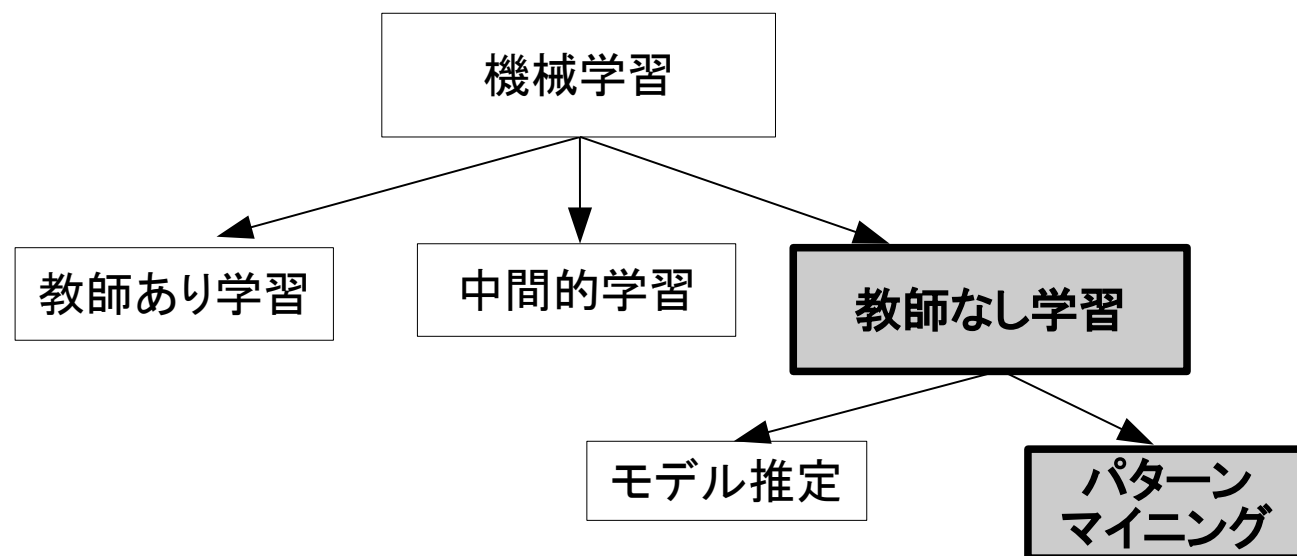


# 12. パターンマイニング

- パターンマイニングの問題設定
  - 入力：カテゴリ特徴の教師なしデータ
  - 出力：頻出項目、連想規則、未観測データ



No.	ミルク	パン	バター	雑誌
1	t	t		
2		t		
3				t
4		t	t	
5	t	t	t	
6	t	t		

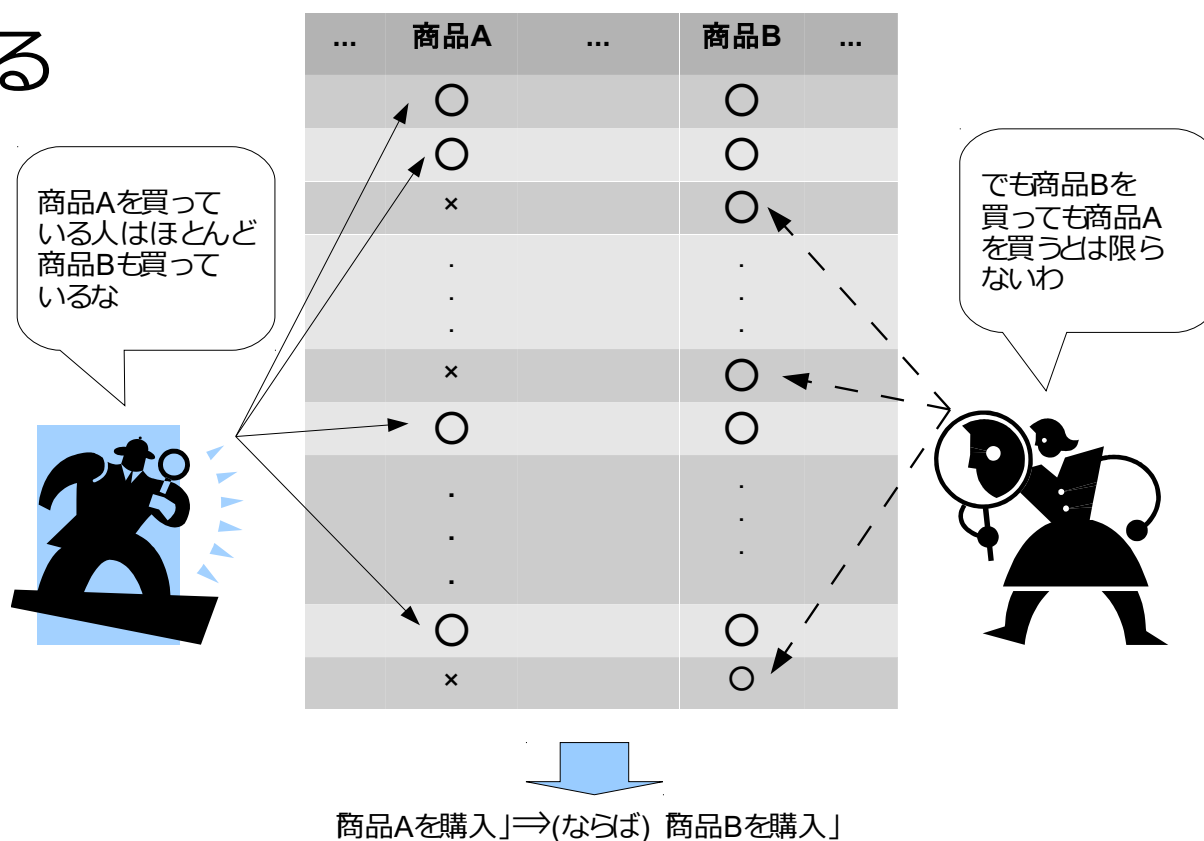
# 12.1 問題の定義

- 学習データ

$$\{\mathbf{x}^{(i)}\} \quad i = 1, \dots, N$$

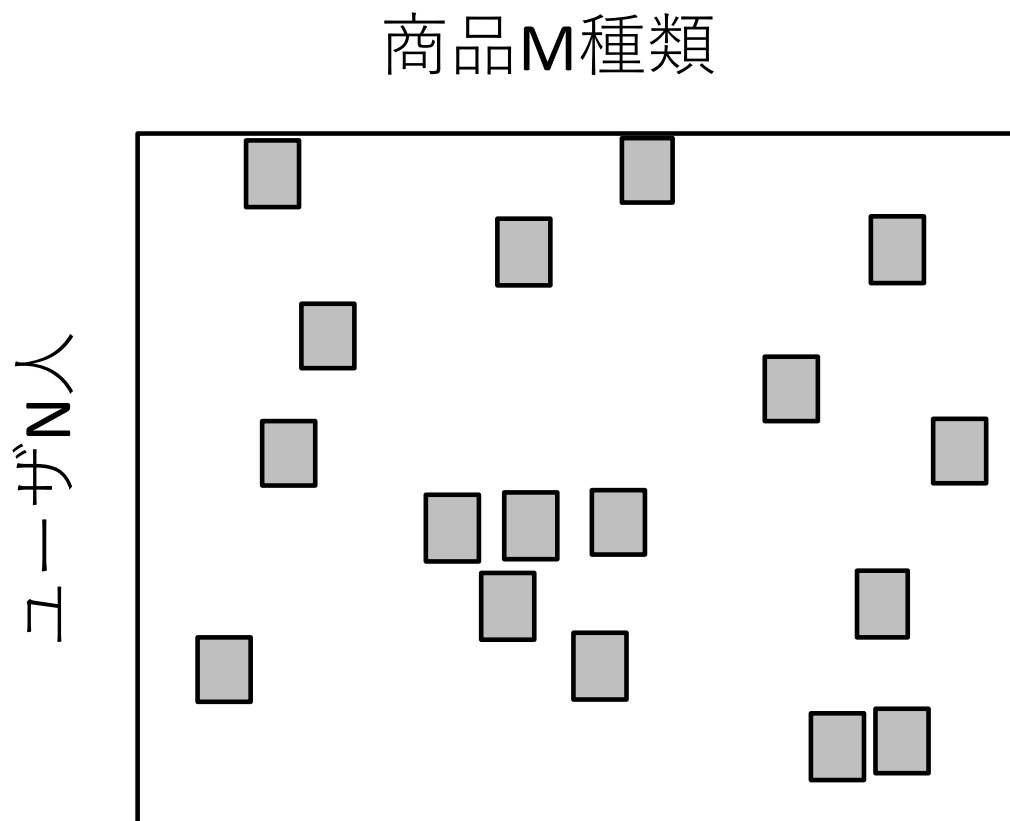
- 問題設定 1

- データ集合中で、一定頻度以上で現れるパターンを抽出する



# 12.1 問題の定義

- 問題設定 2
  - 疎な数値データ（カテゴリ特徴とみなせる離散値）  
を行列とみなして、空所の値を予測する



## 12.2 頻出項目抽出

- 例題：バスケット分析

No.	ミルク	パン	バター	雑誌
1	t	t		
2		t		
3				t
4		t	t	
5	t	t	t	
6	t	t		

バスケット分析では、1件分のデータをトランザクションとよぶ

- 支持度
  - 全トランザクション数  $T$  に対して、ある項目集合 (items) が出現するトランザクションの割合

$$\text{support}(\text{items}) = \frac{T_{\text{items}}}{T}$$

## 12.2 頻出項目抽出

- バスケット分析の目的
  - 支持度の値が閾値以上の項目集合を抽出したい
- バスケット分析の問題点
  - すべての可能な項目集合について、支持度を計算することは現実的には不可能

項目集合の種類数は 2 の商品数乗  
商品数 1,000 の店なら  $2^{1000}$



高頻度の項目集合だけに絞って計算を行う必要がある

## 12.2.2 Apriori アルゴリズムによる頻出項目抽出

- a priori な原理

ある項目集合が頻出ならば、その部分集合も頻出である

例) 「パン・ミルク」が頻出  
ならば「パン」も頻出

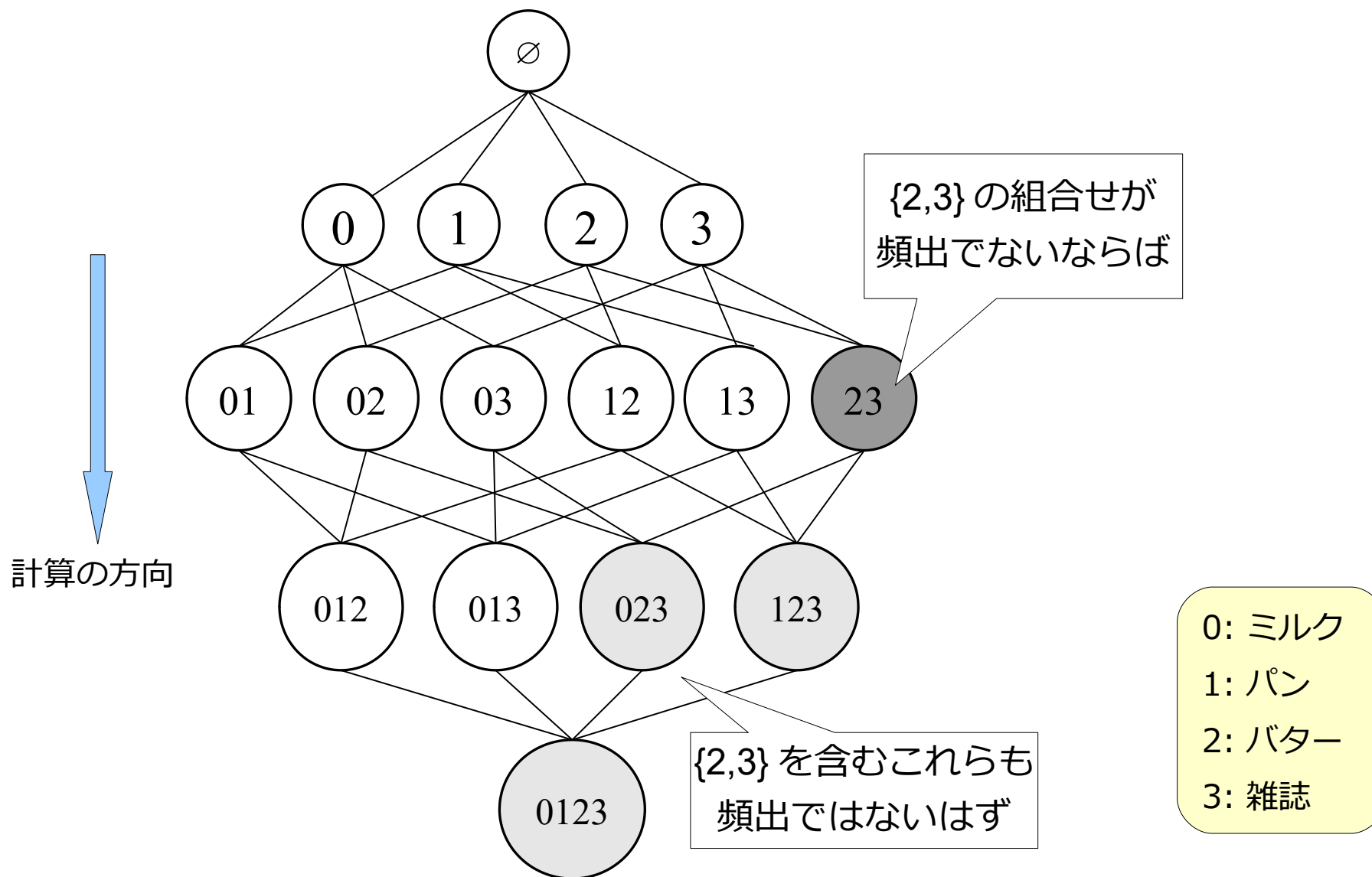


対偶

ある項目集合が頻出でないならば、  
その項目集合を含む集合も頻出でない

例) 「バター・雑誌」が頻出でない  
ならば「バター・雑誌・パン」  
も頻出でない

## 12.2.2 Apriori アルゴリズムによる頻出項目抽出



## 12.3 連想規則抽出

- 連想規則抽出の目的
  - 「商品 A を買った人は商品 B も買う傾向が強い」というような規則性を抽出したい
- 確信度またはリフト値の高い規則を抽出

$$\text{confidence}(A \rightarrow B) = \frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{T_{A \cup B}}{T_A}$$

条件部 A が起こったときに  
結論部 B が起こる割合

$$\text{lift}(A \rightarrow B) = \frac{\text{confidence}(A \rightarrow B)}{\text{support}(B)}$$

B だけが単独で起こる割合と  
A が起こったときに B が起こ  
る割合との比



## 12.3 連想規則抽出

- 連想規則抽出の手順
  - 頻出項目集合を求める
  - 項目集合を条件部、空集合を結論部とした規則を作成する
  - 条件部から結論部へ項目を 1 つずつ移動し、評価する

## 12.3 連想規則抽出

- a priori な原理

ある項目集合を結論部に持つ規則が頻出ならば、  
その部分集合を結論部に持つ規則も頻出である



対偶

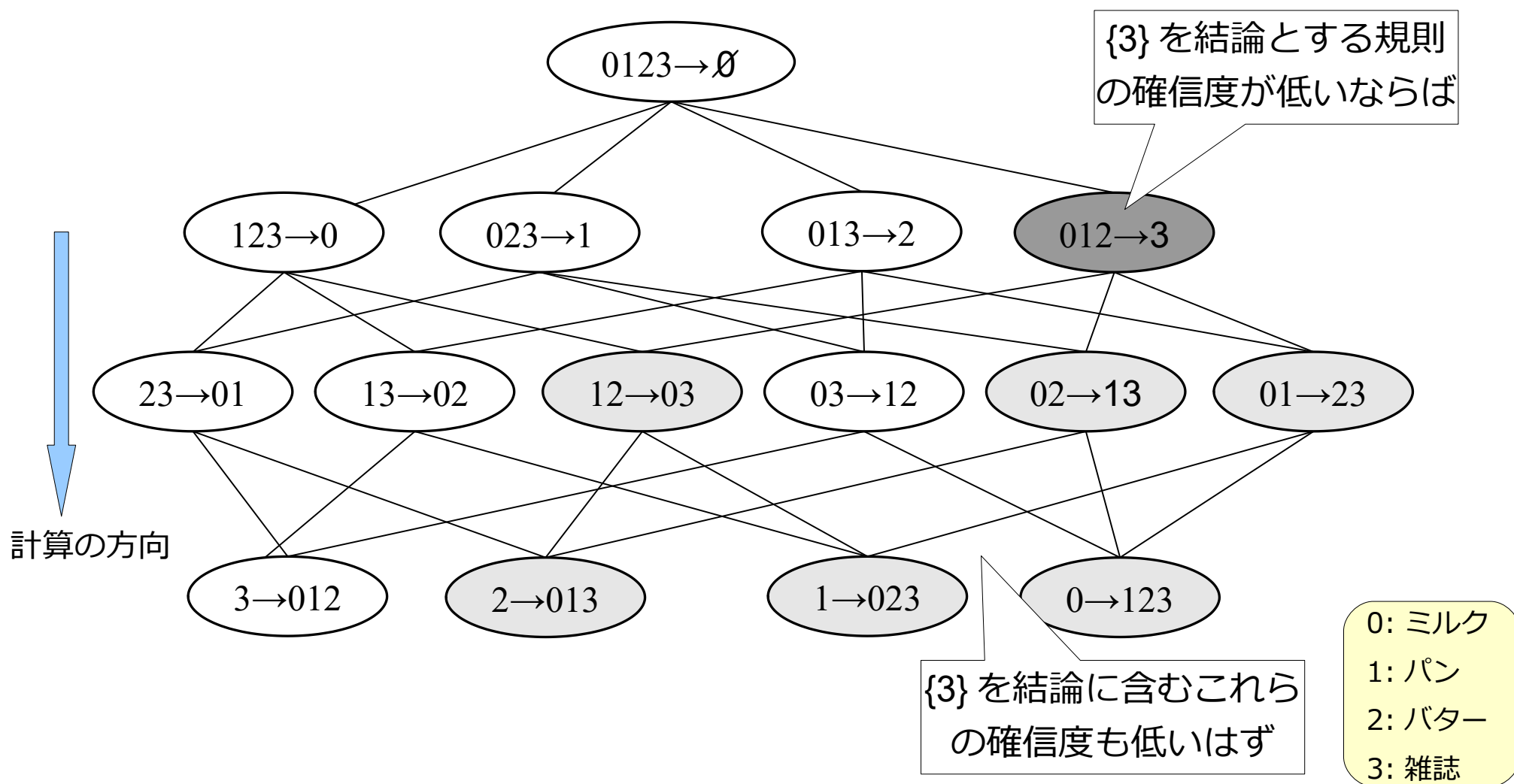
例) 結論部が「パン・ミルク」の規則が  
頻出ならば、結論部が「パン」の  
規則も頻出である

ある項目集合を結論部に持つ規則が頻出でないならば、  
その項目集合を結論部に含む規則集合も頻出でない

例) 結論部が「雑誌」の規則が頻出でない  
ならば、結論部が「パン・雑誌」の  
規則も頻出でない

## 12.3 連想規則抽出

- a priori 原理に基づく探索

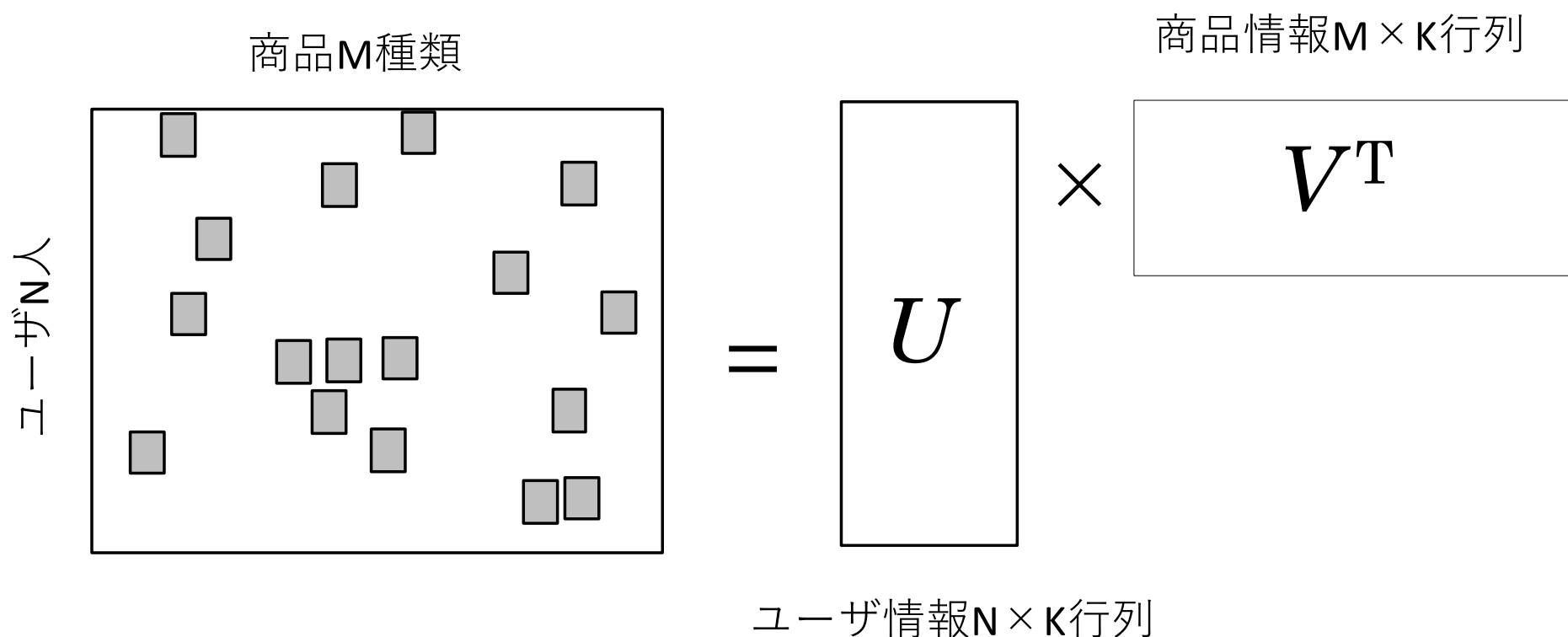


# 12.4 FP-Growth アルゴリズム

- Apriori アルゴリズムの高速化
  - トランザクションをコンパクトに表現し、重複計算を避ける
- 1. トランザクションの前処理
  - トランザクションを、出現する特徴名の集合に変換
  - 出現頻度順にソート
  - 低頻度特徴をフィルタリング
- 2. prefix を共有する木構造 (FP 木) に順次挿入
- 3. FP 木を用いて項目集合の出現頻度を高速計算

# 12.5 推薦システムにおける学習

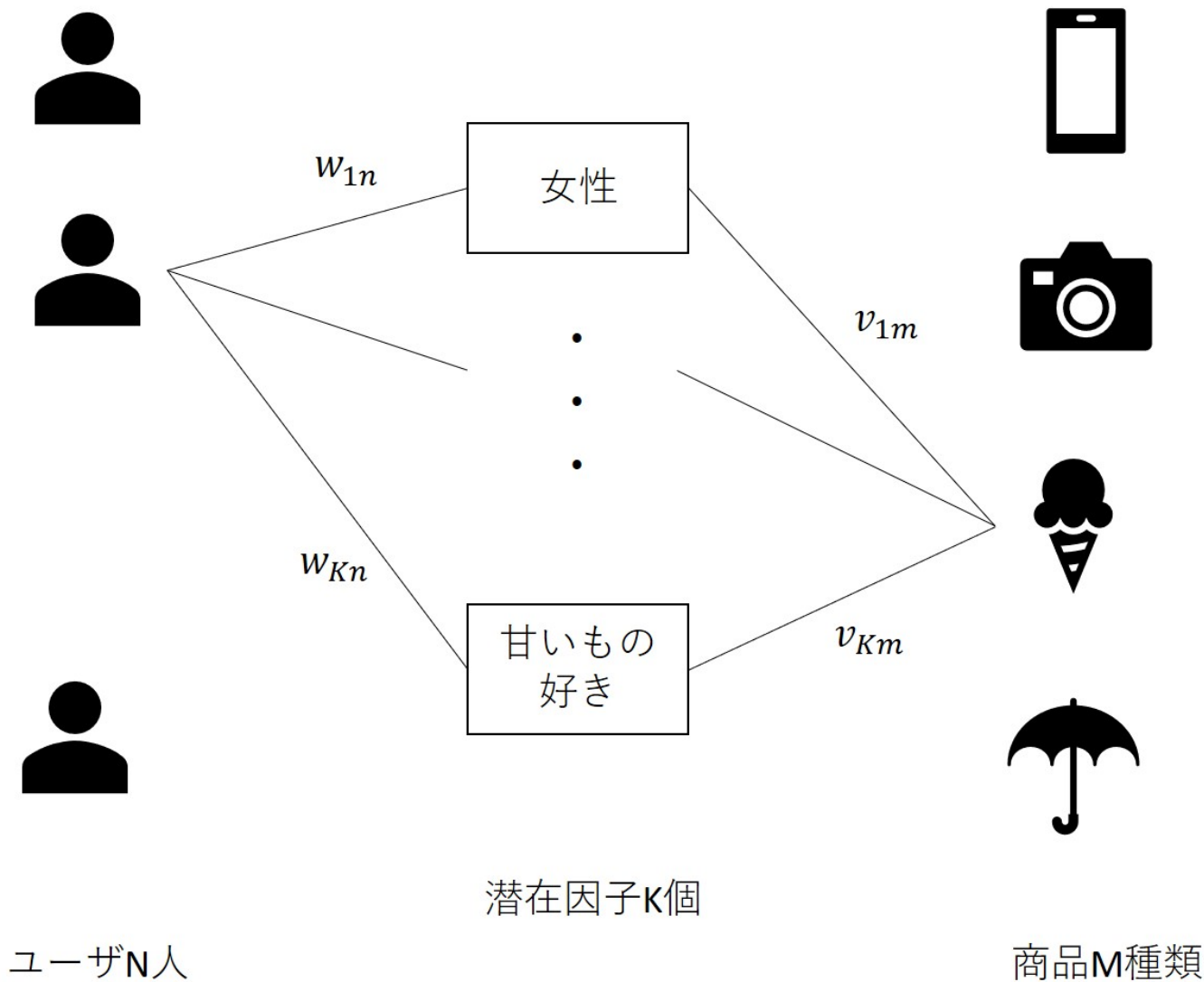
- 協調フィルタリング
  - アイデア：疎な行列は低次元の行列の積で近似できる
  - 値のある部分だけで行列分解を行う
  - 空所の値を予測する



# 12.5 推薦システムにおける学習

- 潜在因子によるデータ表現の考え方

$$x_{mn} = w_{1n}v_{1m} + w_{2n}v_{2m} + \cdots + w_{kn}v_{km}$$



# 12.5 推薦システムにおける学習

- 行列分解の方法
  - $X - UV^T$  の最小化問題を解く

$$\min_{U, V} \frac{1}{2} \|E\|_{\text{Fro}}^2 = \min_{U, V} \frac{1}{2} \|X - UV^T\|_{\text{Fro}}^2$$

空欄を値 0 とみなしてしまっている

- 値が存在する要素だけに限って 2 乗誤差を最小化

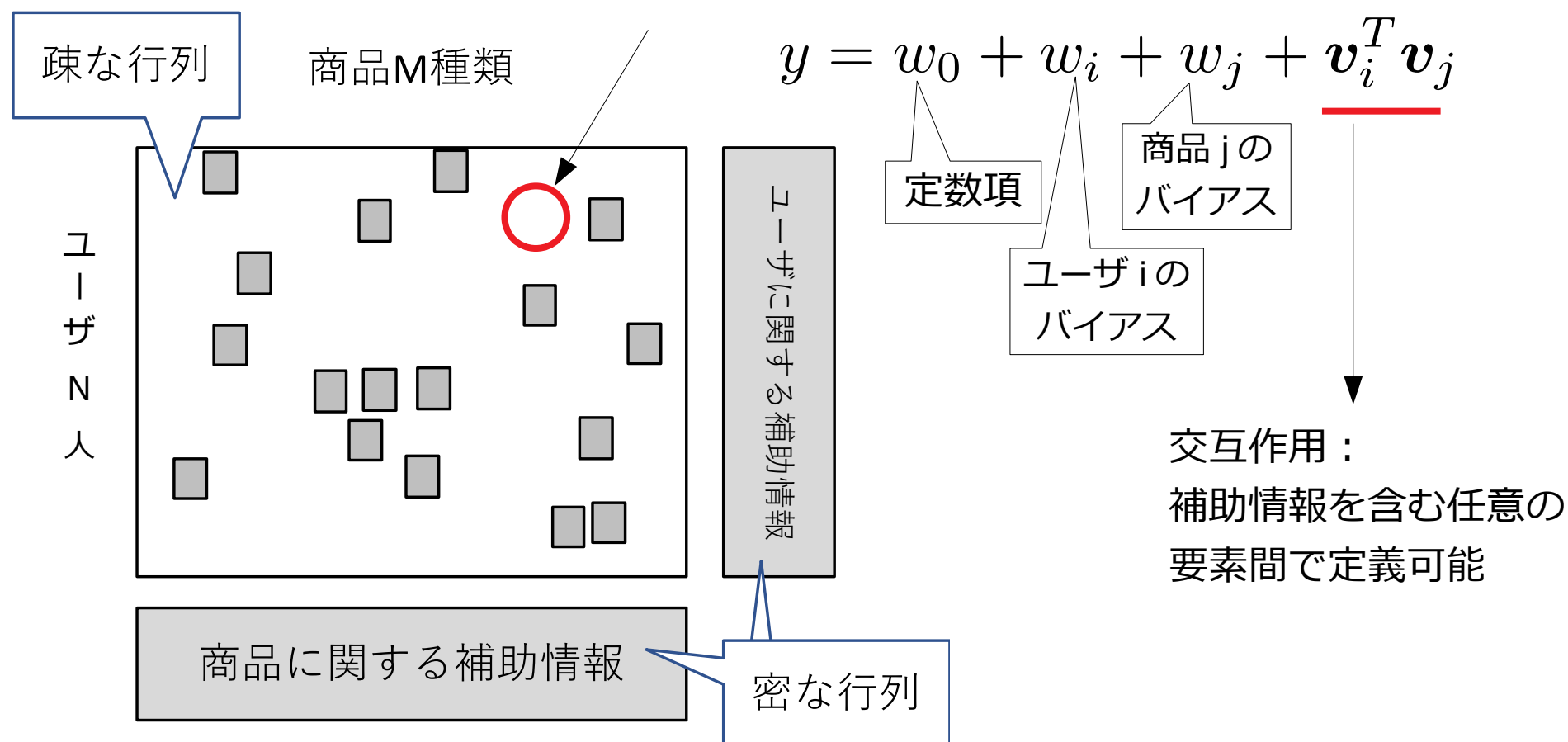
$$\min_{U, V} \sum_{(i, j) \in \Omega} (x_{ij} - u_i^T v_j)^2 + \lambda_1 \underbrace{\|U\|_{\text{Fro}}^2}_{\text{正則化項}} + \lambda_2 \underbrace{\|V\|_{\text{Fro}}^2}_{\text{正則化項}}$$

Fro (フロベニウスノルム) : 行列の要素の二乗和の平方根

# 12.5 推薦システムにおける学習

- Factorization Machine

予測したい値  $y$ : ユーザ  $i$  が商品  $j$  を買うか





# まとめ

- Weka デモ
  - supermarket データ
  - Apriori, FPGrowth
- Python デモ
  - 行列分解
- パターンマイニング
  - 大規模データに対する効率的な数え上げを実現
- 行列分解
  - 低次元の潜在ベクトルを求める