12. パターンマイニング

- 本章の説明手順

1.教師なし、パターンマイニングの問題設定

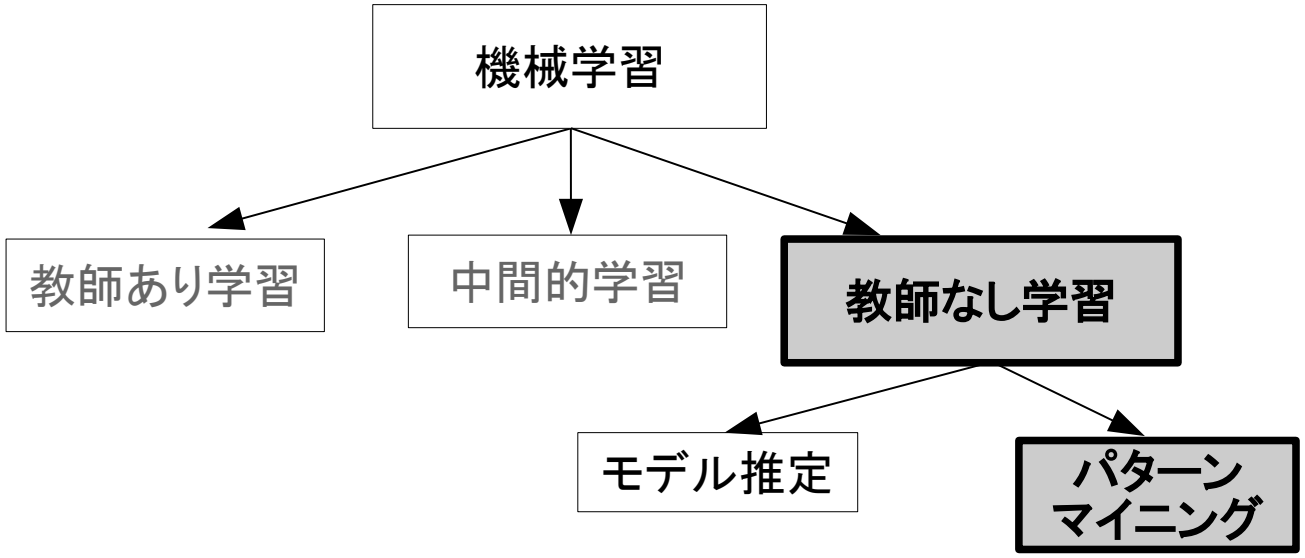
2.出現頻度の高い項目集合を見つける

3.2. の結果に基づき、有用な規則を見つける

4.低次元ベクトル表現を見つけることにより、未知の値の予測を行う

12. パターンマイニング

- 問題設定
  - 教師なし学習
  - （疎な）数値またはカテゴリベクトル → 規則性
    - 規則性の例
      - 頻出項目、連想規則、低次元ベクトル



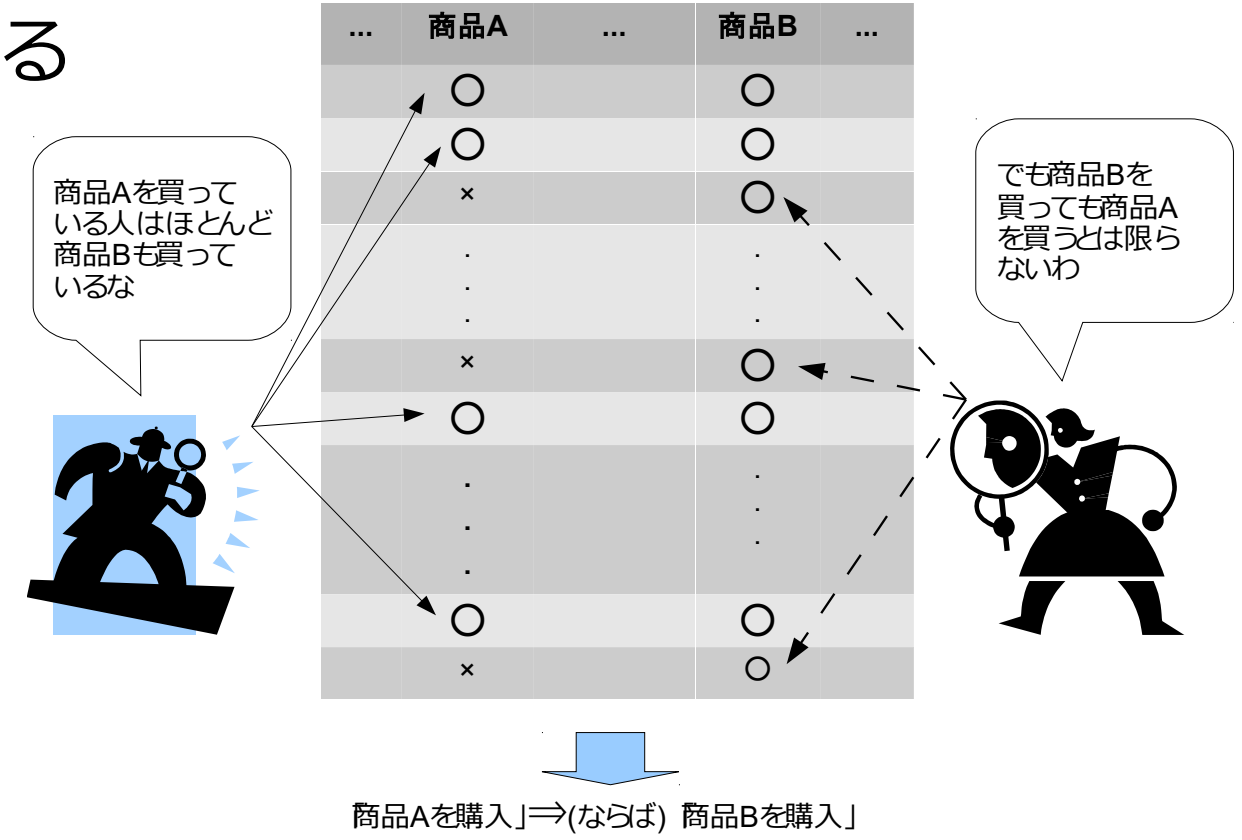
No.	ミルク	パン	バター	雑誌
1	t	t		
2		t		
3				t
4		t	t	
5	t	t	t	
6	t	t		

12.1 カテゴリ特徴に対する「教師なし・パターンマイニング」問題の定義

- 学習データ

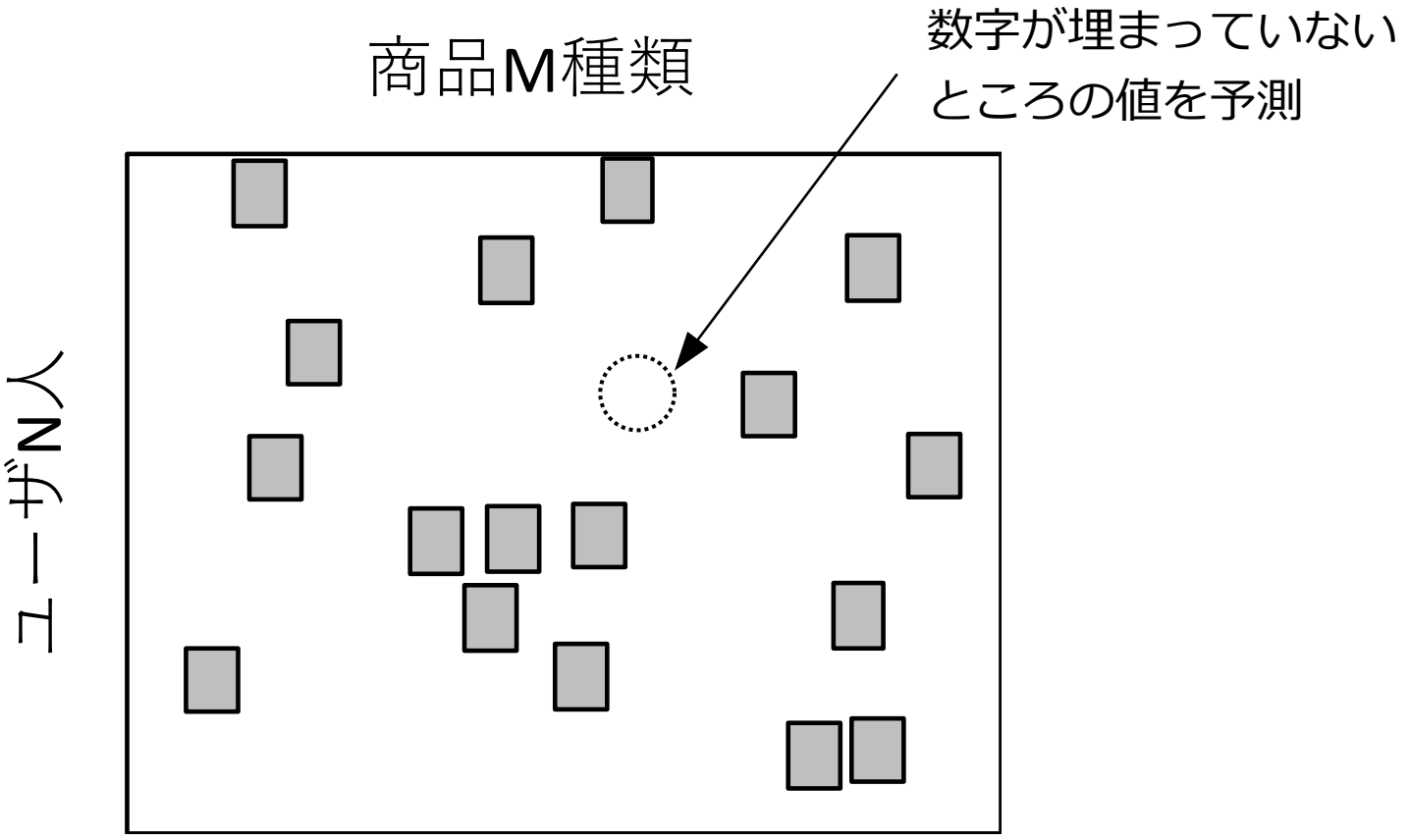
$$\{\boldsymbol{x}_i\} \quad i = 1, \ldots, N$$

- 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 カテゴリ特徴に対する「教師なし・パターンマイニング」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）を行列として扱い、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

- バスケット分析の目的
  - トランザクション中で、一定割合以上出現する項目集合を抽出する

12.2.1 頻出の基準と問題の難しさ

- 支持度
  - 全トランザクション数 T に対する、項目集合 $items$ が出現するトランザクション数 T_{items} の割合

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

- 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

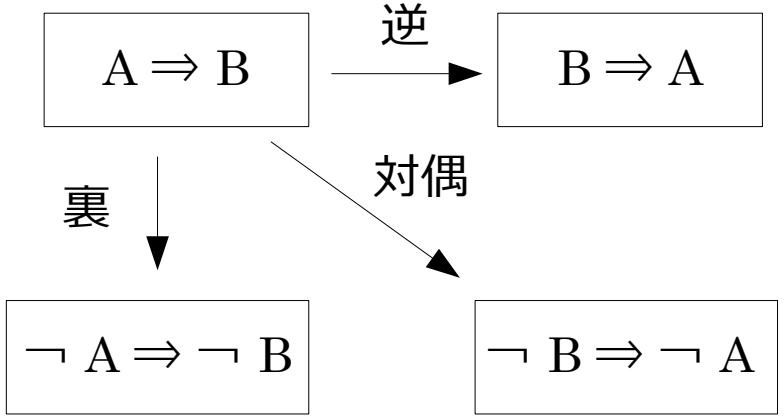
対偶

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

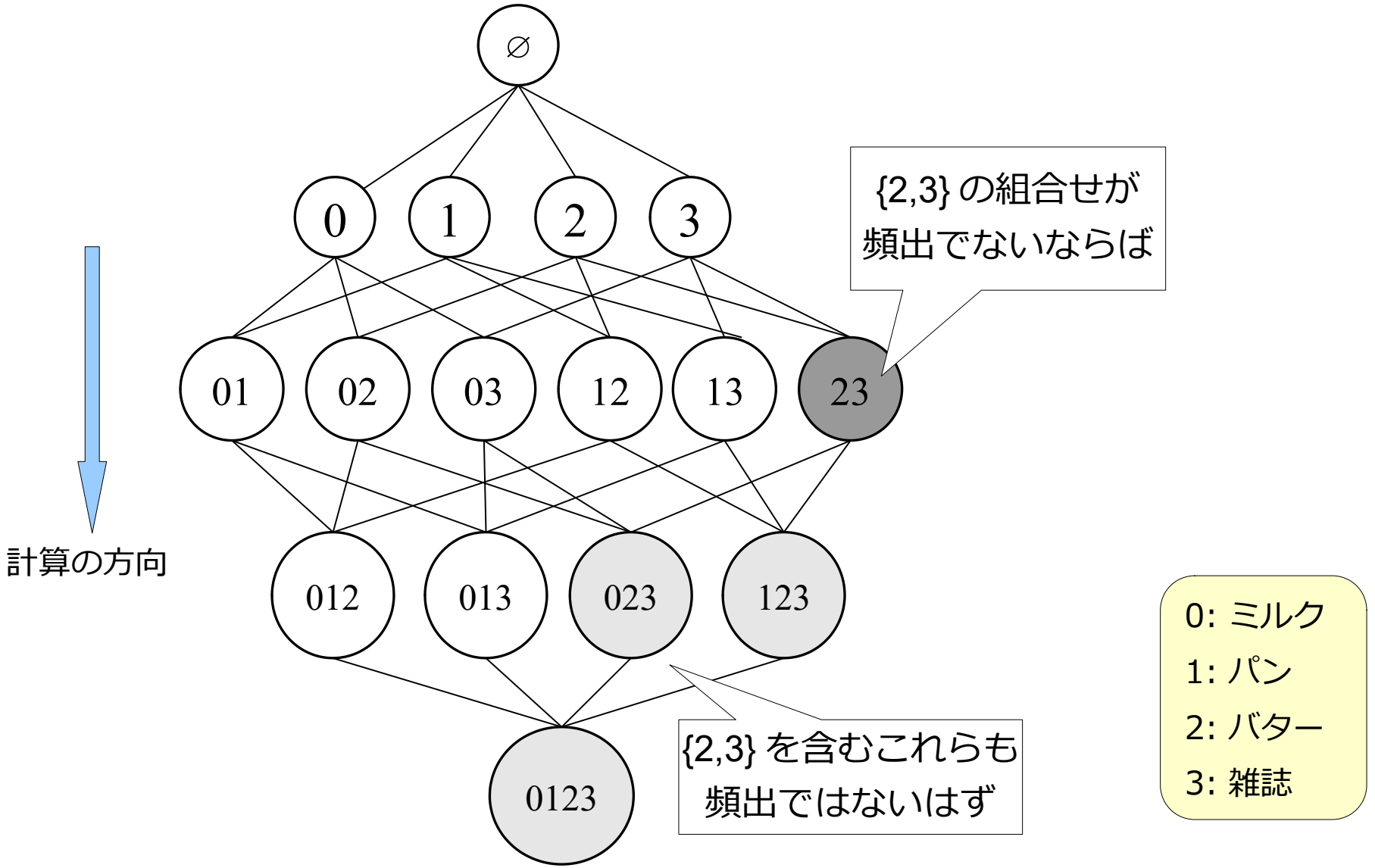
12.2.2 Apriori アルゴリズムによる頻出項目抽出

- 命題論理
  - 「 A ならば B 」が成り立つなら、必ずその対偶である「¬ B ならば¬ A 」が成り立つ



「 A ⇒ B 」は「¬ A∨B 」と定義されている。
一方、「¬ B ⇒ ¬ A 」は
　¬（¬ B ）∨ (¬ A)
なので、
　B ∨ ¬ A
となり、「¬ A∨B 」と等しい

12.2.2 Apriori アルゴリズムによる頻出項目抽出



12.2.2 Apriori アルゴリズムによる頻出項目抽出

Algorithm 12.1 Apriori アルゴリズム（頻出項目抽出）

入力: 正解なしデータ D

出力: 頻出項目集合

$F_1 \leftarrow$ 要素数 1 の頻出項目集合
$k = 2$
while $F_{k-1} \neq \emptyset$ **do**
  $C_k \leftarrow F_{k-1}$の各要素を組み合わせ
  for all $\boldsymbol{x} \in D$ **do**
    for all $c \in C_k$ **do**
      if $c \subset \boldsymbol{x}$ **then**
        $c.count \leftarrow c.count$
      end if
    end for
    $F_k \leftarrow \{c \in C_k \mid c.count >$ 閾値 $\}$
  end for
  $k \leftarrow k + 1$
end while
return $\bigcup_k F_k$

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

$$\mathrm{confidence}(\mathrm{A} \Rightarrow B) = \frac{\mathrm{support}(A \cup B)}{\mathrm{support}(A)} = \frac{T_{A \cup B}}{T_A}$$

前提部 A が起こったときに結論部 B が起こる割合

$$\mathrm{lift}(\mathrm{A} \Rightarrow B) = \frac{\mathrm{confidence}(A \Rightarrow B)}{\mathrm{support}(B)}$$

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

- 支持度・確信度・リフト値の意味
  - support({ ハム , 卵 }) : 0.1
  - confidence(ハム⇒卵) : 0.7　lift(ハム⇒卵): 5
  - 「全体顧客の 10% がハムと卵を一緒に購入しており、ハム購入者の 70% が卵も購入している」ということになる。この時のリフト値 5 は、「ランダムに選んだ顧客が卵を買う確率に対して、ハムを買った顧客が卵を買う確率は 5 倍大きい」という意味を表している。

12.3.4 Apriori アルゴリズムによる連想規則抽出

- 連想規則抽出の手順
  - 頻出項目集合を求める
  - 項目集合を前提部、空集合を結論部とした規則を作成する
  - 前提部から結論部へ項目を 1 つずつ移動し、確信度またはリフト値で評価する

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

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

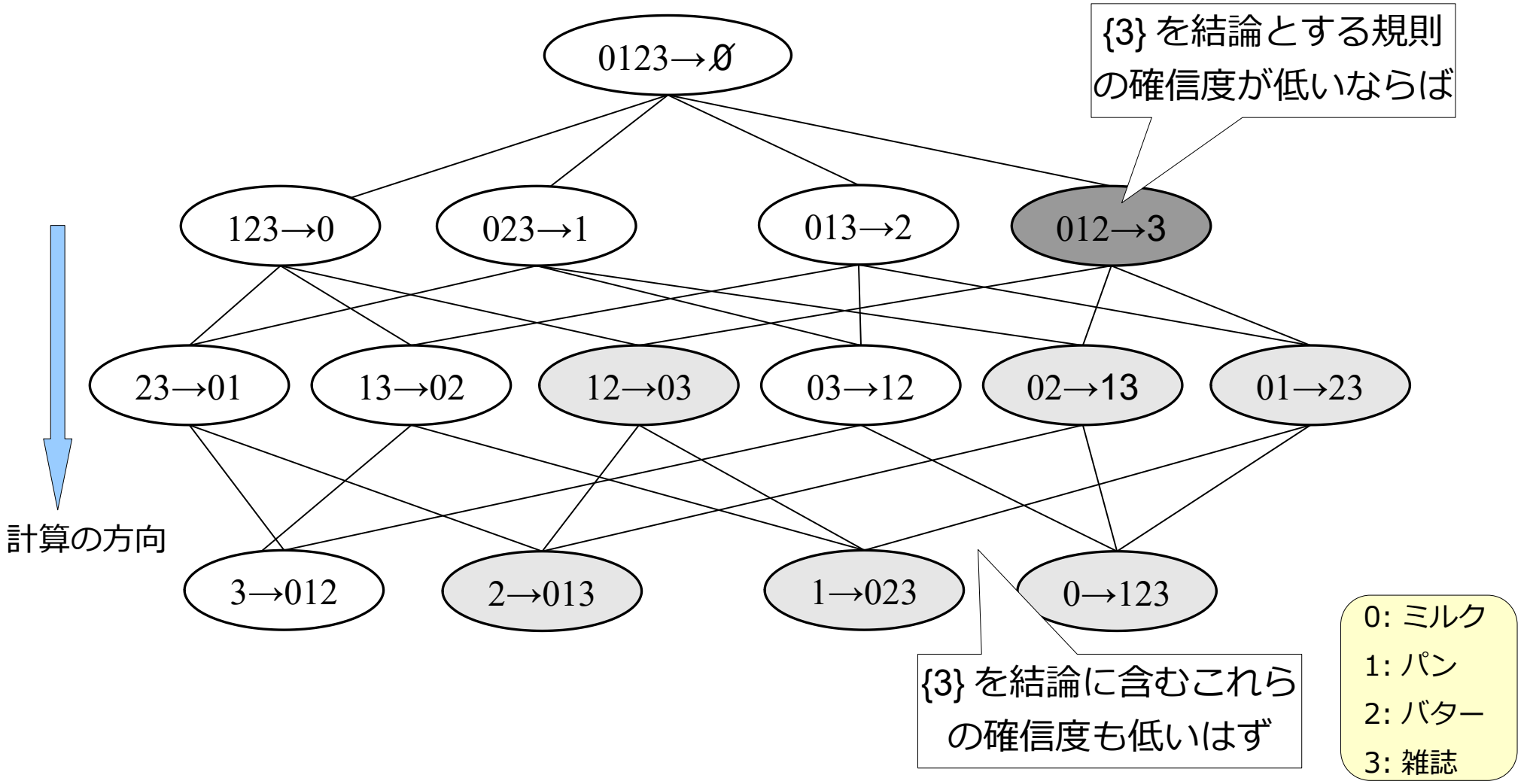
対偶

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

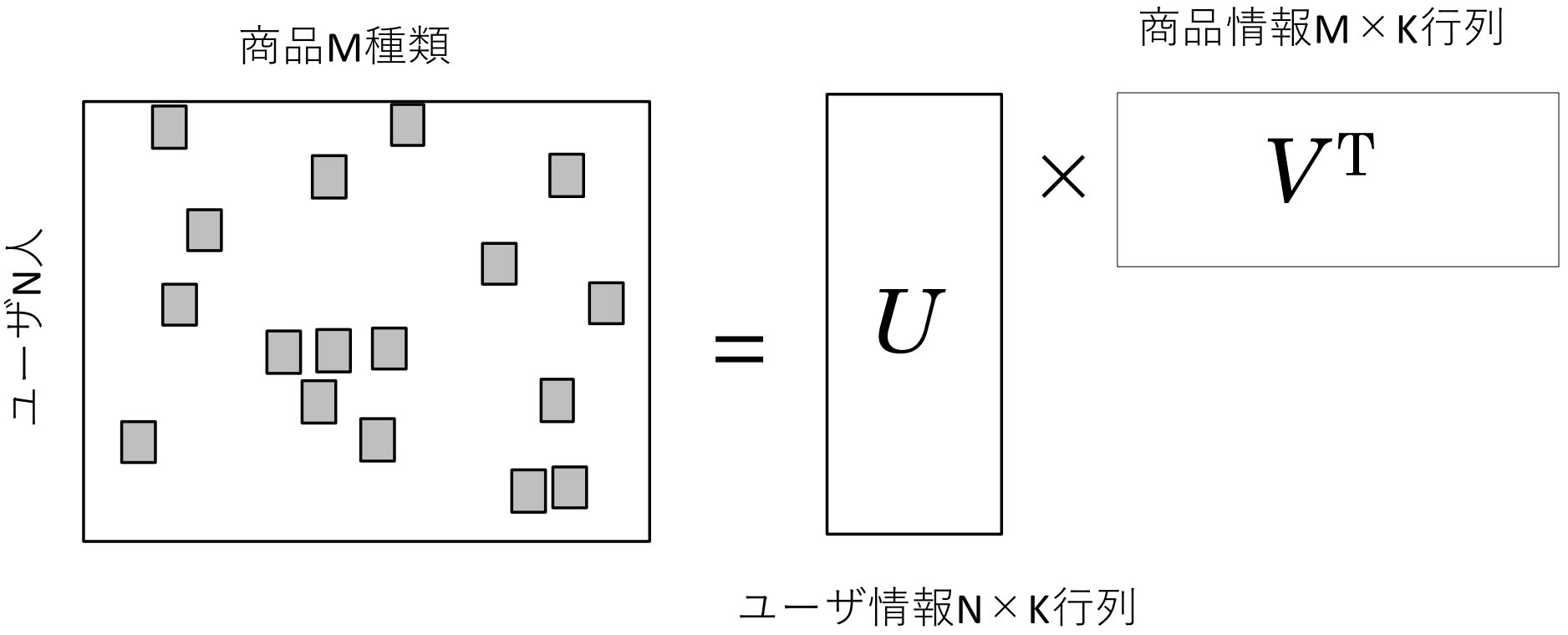
12.3 連想規則抽出

- a priori 原理に基づく探索



12.5 推薦システムにおける学習

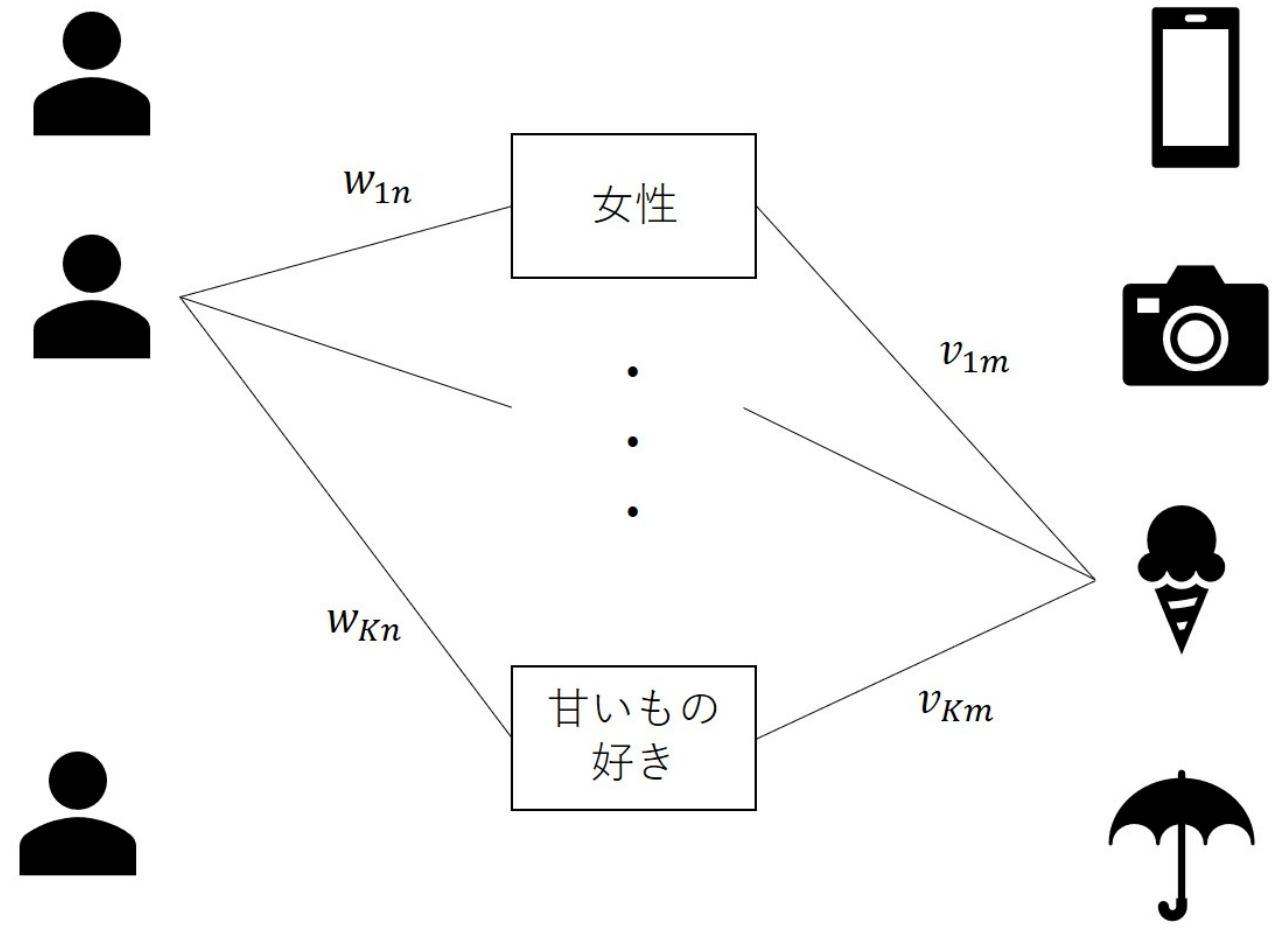
- 協調フィルタリング
  - アイデア：疎な行列は低次元の行列の積で近似できる
  - 値のある部分だけで行列分解を行う
  - 空所の値を予測する



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
  - $\boldsymbol{X}$-$\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$ の最小化問題を解く

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

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

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{(i,j)\in\Omega} (x_{ij} - \boldsymbol{u}_i\boldsymbol{v}_j^T)^2 + \lambda_1\|\boldsymbol{U}\|_{\mathrm{Fro}}^2 + \lambda_2\|\boldsymbol{V}\|_{\mathrm{Fro}}^2$$

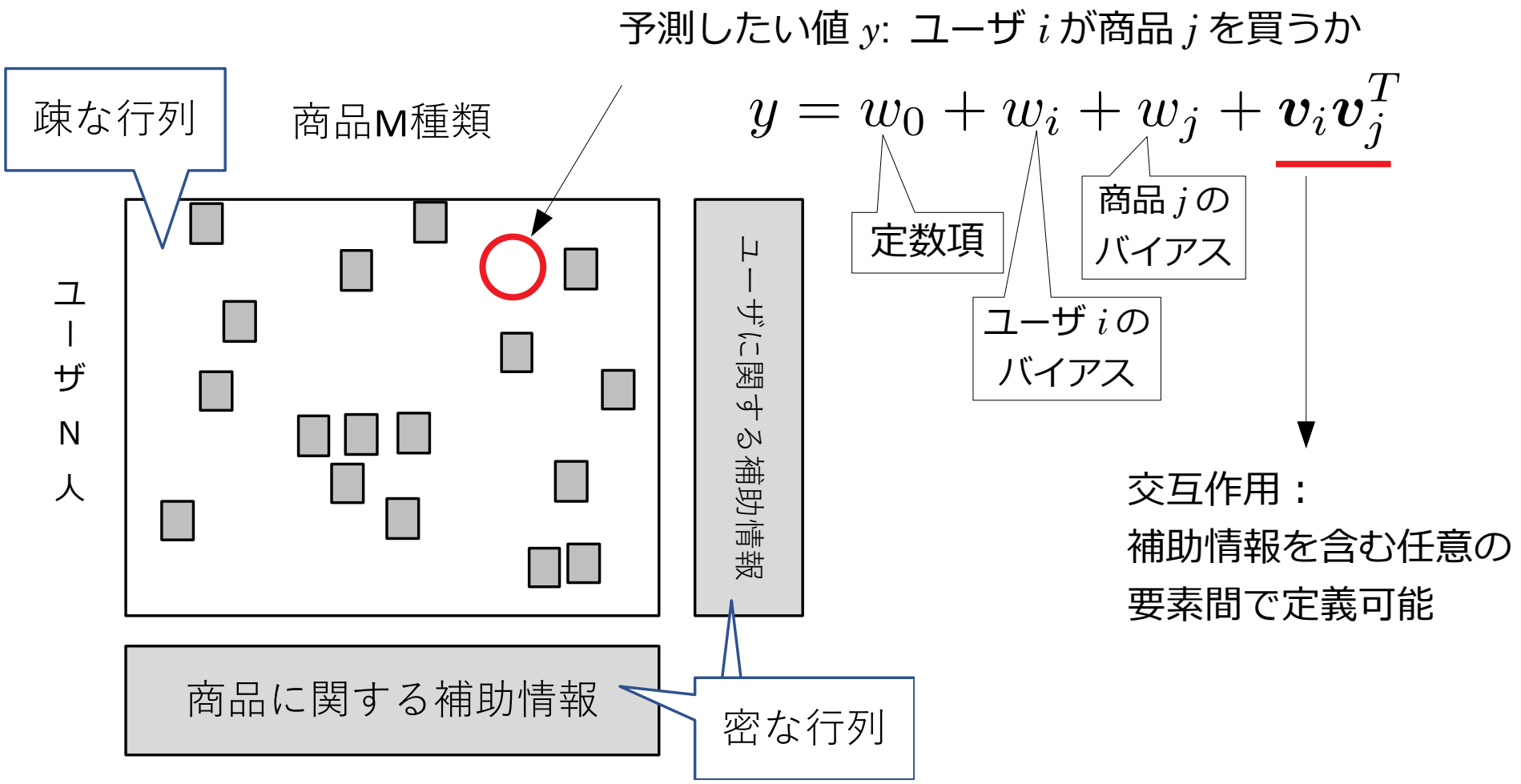
正則化項

Fro（フロベニウスノルム）：行列の要素の二乗和の平方根

  - $\boldsymbol{U},\boldsymbol{V}$ の要素を非負に限定したものが NMF

12.5 推薦システムにおける学習

- Factorization Machine
  - 補助情報を予測に取り入れることができる



補足

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

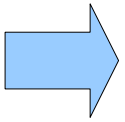
1. トランザクションの前処理
   - トランザクションを、出現する特徴名の集合に変換
   - 出現頻度順にソート
   - 低頻度特徴をフィルタリング
2. prefix を共有する木構造 (FP 木) に順次挿入
3. FP 木を用いて項目集合の出現頻度を高速計算

12.4 FP-Growth アルゴリズム

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

```
1 {r,z,h,j,p}
2 {z,y,x,w,v,u,t,s}
3 {z}
4 {r,x,n,o,s}
5 {y,r,x,z,q,t,p}
6 {y,z,x,e,q,s,t,m}
```

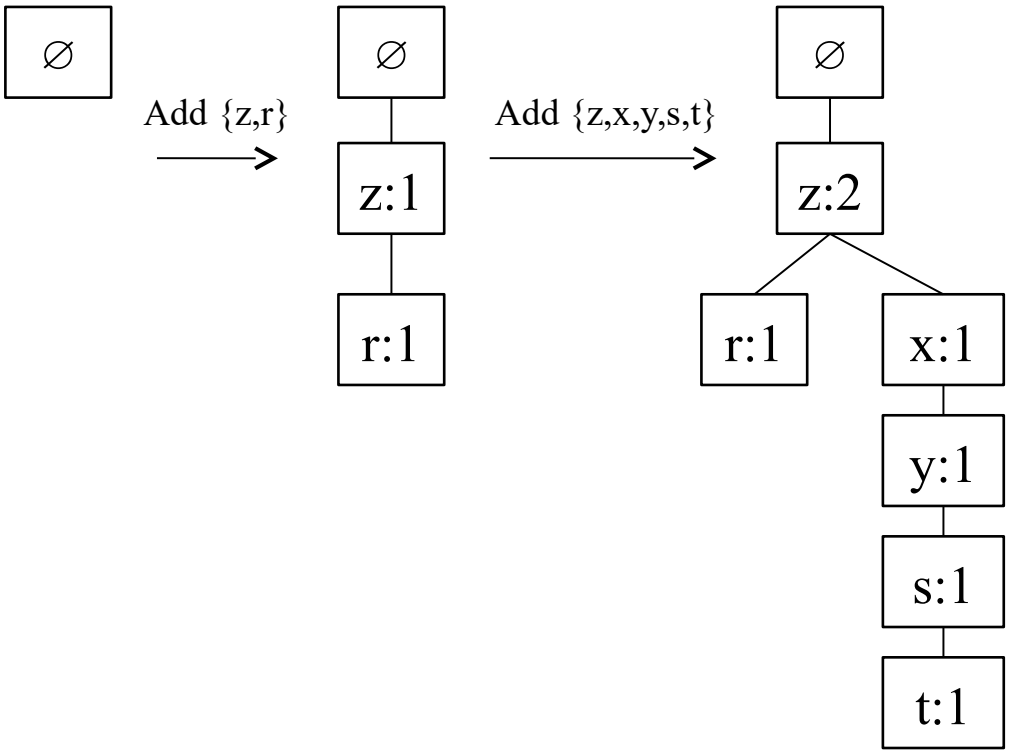


```
1 {z,r}
2 {z,x,y,s,t}
3 {z}
4 {x,s,r}
5 {z,x,y,r,t}
6 {z,x,y,s,t}
```

12.4 FP-Growth アルゴリズム

2.prefix を共有する木構造 (FP 木) に順次挿入

- ソート、フィルタリング後のトランザクションデータを順次 FP 木に挿入



12.4 FP-Growth アルゴリズム

3.FP 木を用いて項目集合の出現頻度を高速計算