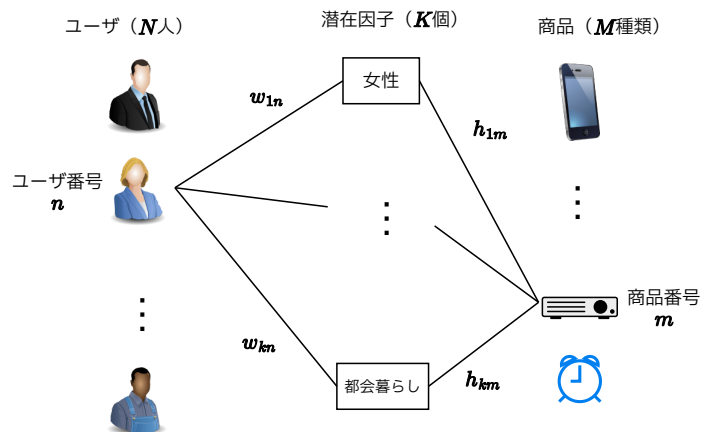


12. パターンマイニング



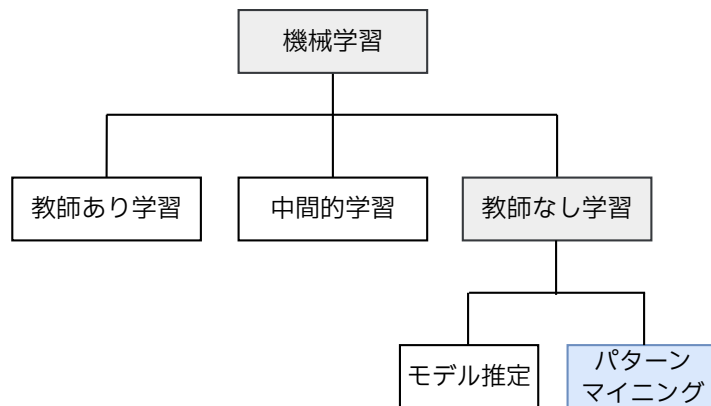
- 12.1 「教師なし・パターンマイニング」の問題の定義
- 12.2 頻出項目抽出
- 12.3 連想規則抽出
- 12.4 行列分解



- 荒木雅弘：『Pythonではじめる機械学習』（森北出版，2025年）
- スライドとコード

12. パターンマイニング

- 問題設定
 - (疎な) 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



- 応用例
 - 推薦システム

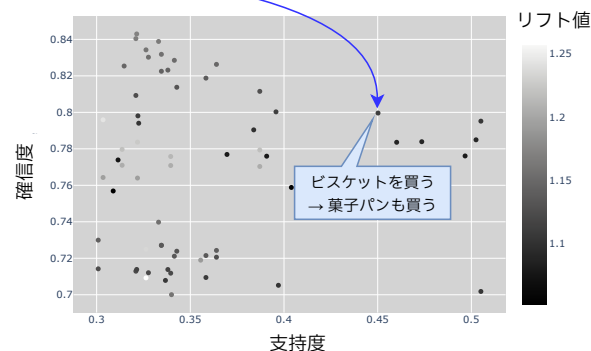
12.1 「教師なし・パターンマイニング」問題の定義 (1/2)

- データセット（正解なし）
 - （疎な） d 次元のカテゴリまたは数値ベクトル $\{\mathbf{x}_i\}$ ($i = 1, \dots, N$)
 - 値が数値でも、順序尺度（例：5段階評価値）のような実質的にはカテゴリとみなせるもの
- 問題設定1
 - データセット中で一定頻度以上で現れる特徴の組み合わせや規則を抽出

菓子パンとビスケットの両方で True が頻出する

	その他 食料品	ベビー 用品	菓子 パン	ビス ケット	クーポン	ジュース	お茶	水
0	False	True	True	True	False	True	False	True
1	False	False	False	False	False	False	False	False
2	False	False	True	True	False	True	False	True
3	False	False	True	True	False	True	False	True
4	False	False	True	True	False	True	True	False
...

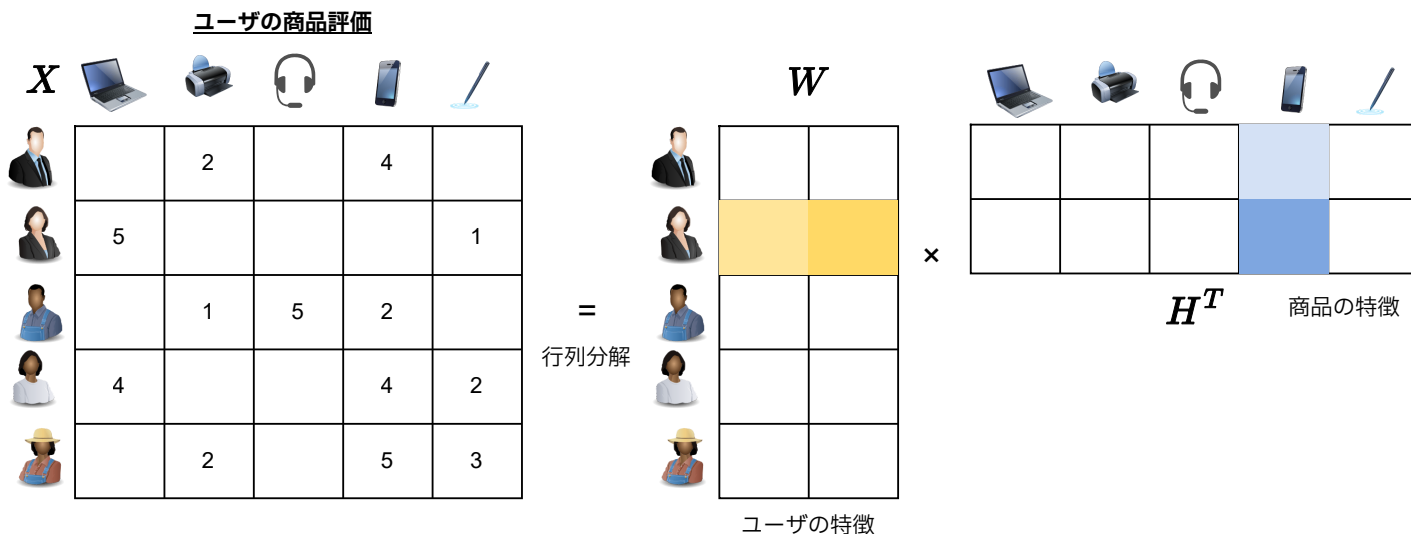
(a) スーパーマーケットの購買記録



(b) 抽出された規則

- 問題設定2

- 似ているデータを参考にして、空所の値を予測する



12.2 頻出項目抽出

12.2.1 頻出の基準と問題の難しさ (1/2)

- 例題：バスケット分析（1件分のデータをトランザクションとよぶ）
 - トランザクション集合中で，一定割合以上出現する項目集合を抽出

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		

12.2.1 頻出の基準と問題の難しさ (2/2)

- 頻出の基準：支持度 (support)
 - 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

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

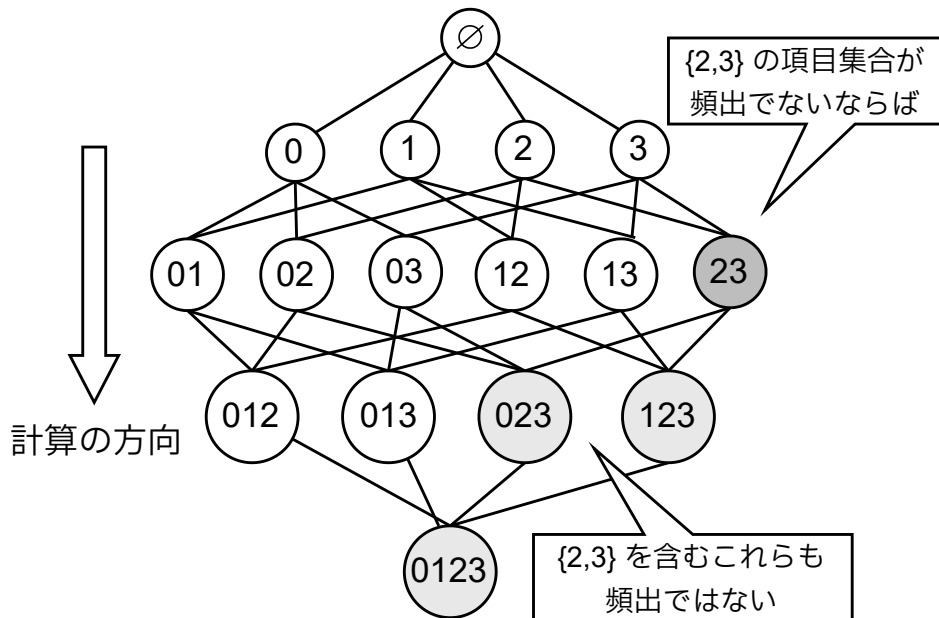
- バスケット分析の問題点
 - すべての可能な項目集合について支持度を計算することは現実的には不可能
 - 商品の種類数 1,000 の店なら，項目集合の数は 2^{1000}
 - 高頻度の項目集合に絞って計算を行う必要がある

12.2.2 アプリオリアルゴリズム (1/3)

- アプリオリな原理: 「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上位集合も頻出ではない」
 - 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 アプリアリアルゴリズム (2/3)

- アプリアリな原理の対偶を用いて頻出項目集合の候補を削減



12.2.2 アプリオリアルゴリズム (3/3)

- 頻出項目集合抽出の手順
 - 入力 : 正解なしデータ D , 支持度の閾値
 - 出力 : 頻出項目集合
 1. 項目数1の集合 C_1 を求める
 2. C_1 から支持度が閾値以下の要素を削除し, 集合 F_1 を求める
 3. $k = 2$ から始め, $F_k = \emptyset$ (空集合) になるまで以下を繰り返す
 - F_{k-1} の要素を組み合わせ, 項目数 k の集合 C_k を作成する
 - C_k の要素で, その部分集合が F_{k-1} に含まれないものを削除する
 - C_k から支持度が閾値以下の要素を削除し, F_k とする
 4. 上記ループ終了時点までの F_k の和集合を返す

12.2.3 FP-Growth アルゴリズム (1/4)

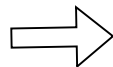
- アプリオリアルゴリズムの高速化
 - トランザクションをコンパクトに表現し，重複計算を避ける
- 1. トランザクションの前処理
 1. トランザクションを出現する特徴名の集合に変換
 2. 各集合を出現頻度順にソート
 3. 低頻度特徴をフィルタリング
- 2. prefixを共有する木構造(FP木)に順次挿入
- 3. FP木を用いて項目集合の出現頻度を高速計算

12.2.3 FP-Growth アルゴリズム (2/4)

- 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
 - 各集合を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}

(a) トランザクション

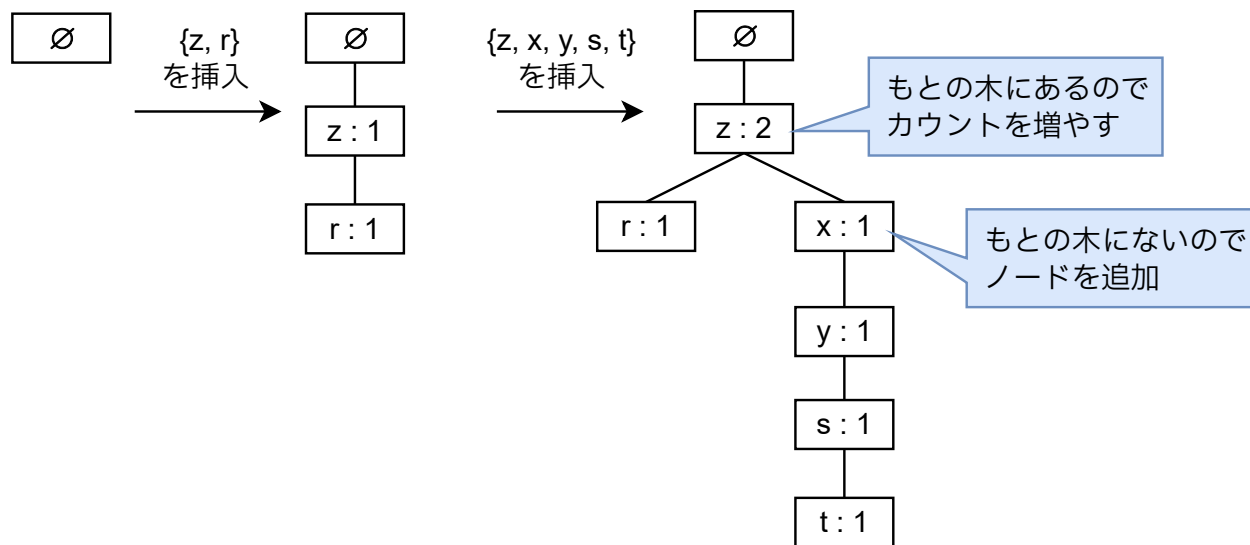


1	{z,r}
2	{z,x,y,s,t}
3	{z}
4	{x,s,r}
5	{z,x,y,r,t}
6	{z,x,y,s,t}

(b) ソート, フィルタリング後のデータ

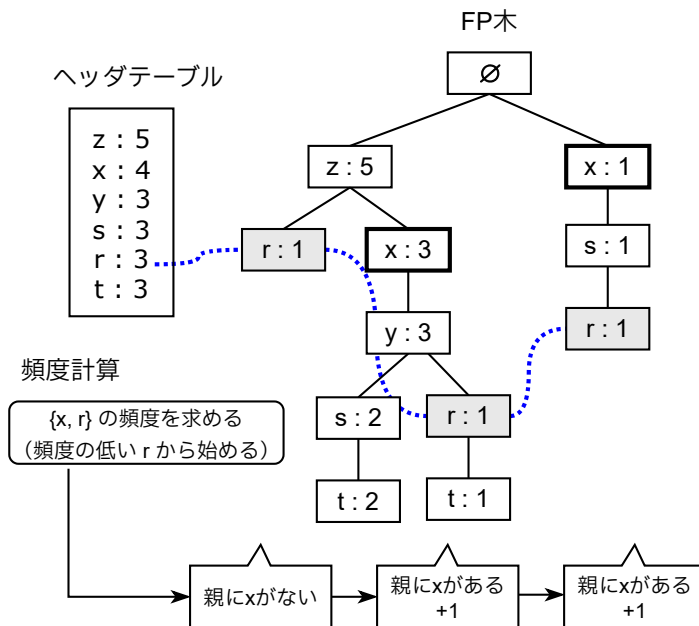
12.2.3 FP-Growthアルゴリズム (3/4)

- prefixを共有する木構造(FP木)を作成
 - 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



12.2.3 FP-Growthアルゴリズム (4/4)

- FP木を用いて項目集合の出現頻度を高速計算



12.3 連想規則抽出 (1/4)

- 連想規則抽出とは
 - 正解付きデータに対して正解を目的変数とみなし, それに対する入力変数の関係を記述
 - 例: 「商品Aを買った人は商品Bも買う傾向がある」というような規則性を抽出したい
 - 評価値の例
 - 確信度: 前提部Aが起こったときに結論部Bが起こる割合

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

- リフト値: 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

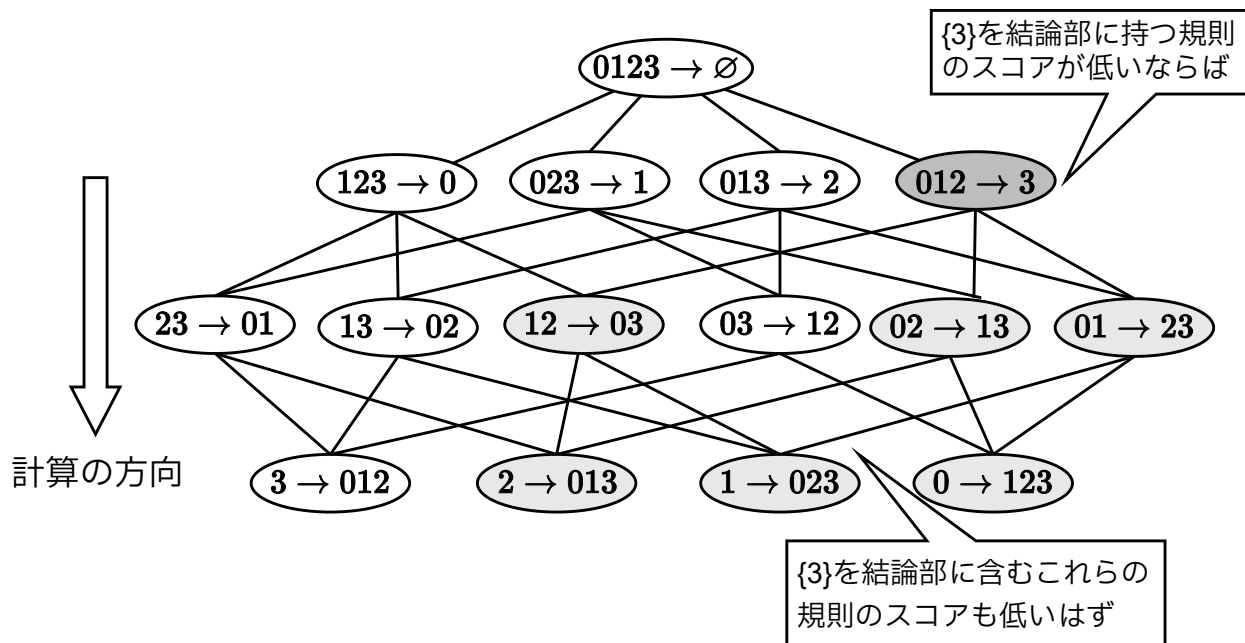
$$\text{lift}(A \rightarrow B) = \frac{\text{confidence}(A \rightarrow B)}{\text{support}(B)}$$

12.3 連想規則抽出 (2/4)

- アプリオリな原理: 「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, 「その上位集合を結論部に含む規則」も頻出ではない
 - 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」の規則も頻出ではない

12.3.4 連想規則抽出 (3/4)

- アプリオリな原理に基づく探索



12.3 連想規則抽出 (4/4)

- 連想規則抽出アルゴリズム

- 入力 : 頻出項目集合 F , 評価値の閾値 θ
- 出力 : 連想規則集合

for f in F :

for all $A \subset f$: # A は f の真部分集合

$B = f \setminus A$ # f から A を除いたもの

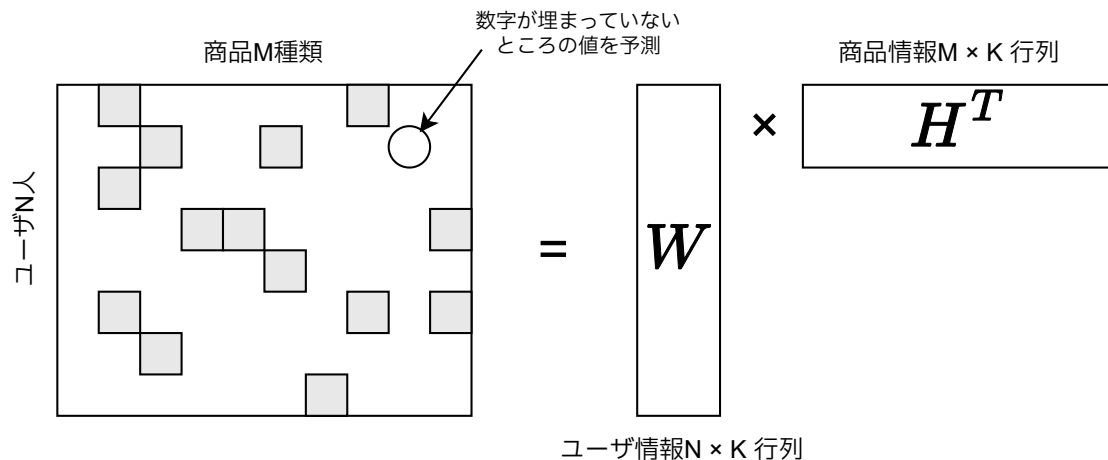
if 評価値($A \Rightarrow B$) $\geq \theta$:

 規則 $A \Rightarrow B$ を連想規則集合に追加

return 連想規則集合

12.4 行列分解 (1/4)

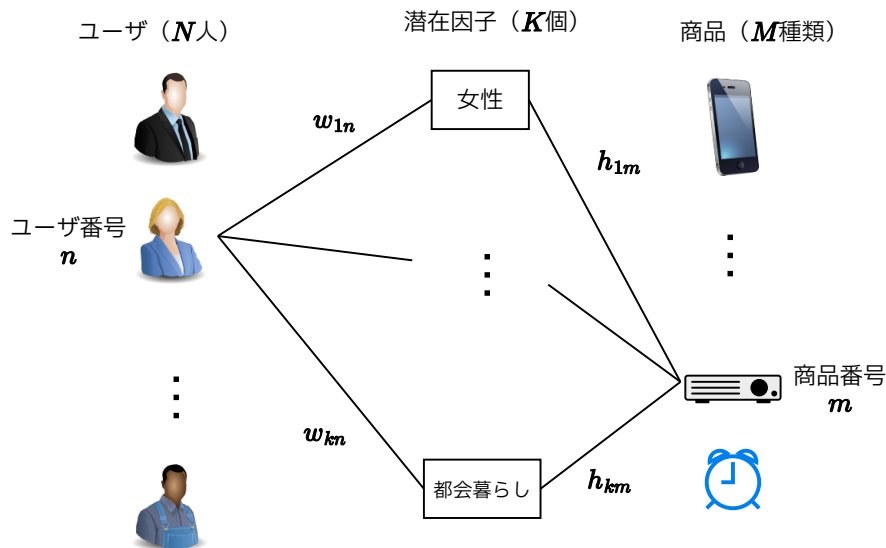
- 協調フィルタリング
 - アイデア：疎な行列は低次元の行列の積で近似できる
 - 少数の潜在因子を設定し、ユーザや商品の特徴をその潜在因子の重みで表現していると考える
 - 空所の値を予測することにより推薦を行う



12.4 行列分解 (2/4)

- 潜在因子によるデータ表現の考え方
 - 図中の潜在因子の名前は行列分解の結果を解釈したものではない

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



12.4 行列分解 (3/4)

- 行列分解の方法
 - $\mathbf{X} - \mathbf{UV}^\top$ の最小化問題を解く

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

- 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二乗誤差を最小化 (+正則化)

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

- \mathbf{U}, \mathbf{V} の要素を非負に限定したものが非負値行列因子分解 (NMF : Nonnegative Matrix Factorization)

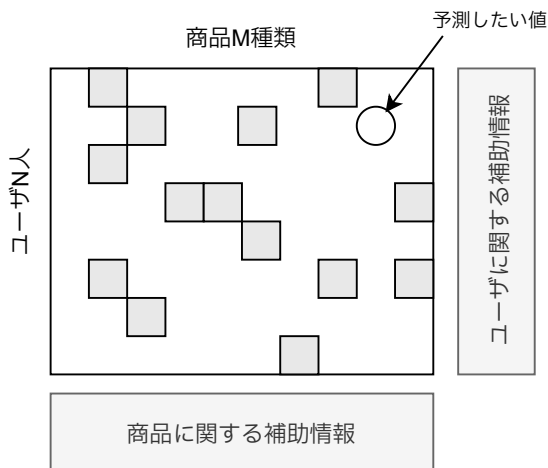
12.4 行列分解 (4/4)

- Factorization Machine

- 補助情報を予測に取り入れることができる行列分解

$$y = w_0 + \sum_i w_i x_i + \sum_{i < j} \mathbf{v}_i^\top \mathbf{v}_j x_i x_j$$

- w_0 : 定数項, w_i : 特徴 x_i の重み, \mathbf{v}_i : 特徴 x_i の潜在ベクトル (セルフアテンションと似たアイデア)



まとめ

- パターンマイニングは有用な規則性を発見する
- アプリアリアルゴリズム
 - 出現頻度の高い項目集合を見つける
 - 出現頻度に基づき、有用な規則を見つける
 - FPGrowth はアプリアリアルゴリズムの高速化版
- 行列分解
 - 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