【ICDE2014 勉強会】

# No.13 Data Mining ll : Pattern Discovery

担当: 塩井, 萩原(同志社大)

# Data Mining ll: Pattern Discovery

- Automatic Generation of Question Answer Pairs From Noisy Case Logs
  - Jitenda Ajmera, Sachinda Joshi, Ashish verma, Amoi Mittal
- Complete Discovery of High-Quality Patterns in Large Numerical Tensors
  - Loïc Cerf and Wagner Meira Jr.
- Ranking Item Features by Mining Online User-Item Interactions
  - Sofiane Abbar, Habibur Rahman, Saravanan Thirumuruganathan, Carlos Castillo, Gautam Das

※スライドで用いている図は上記論文からの引用です

# Data Mining II: Pattern Discovery

- Automatic Generation of Question Answer Pairs From Noisy Case Logs
  - Jitenda Ajmera, Sachinda Joshi, Ashish verma, Amoi Mittal
- Complete Discovery of High-Quality Patterns in Large Numerical Tensors
  - Loïc Cerf and Wagner Meira Jr.
- Ranking Item Features by Mining Online User-Item Interactions
  - Sofiane Abbar, Habibur Rahman, Saravanan Thirumuruganathan, Carlos Castillo, Gautam Das

※スライドで用いている図は上記論文からの引用です

# Complete Discovery of High-Quality Patterns in Large Numerical Tensors

- 研究の目的
  - これまでにテンソルが用いられた研究は複数存在するが，そのどれも不完全であったり，処理の途中で情報が失われている
  - パターン列挙のための仕組みは最新の抽出器のものを用いるが，新しく定義を追加することで早さの改善，精度の向上

# Complete Discovery of High-Quality Patterns in Large Numerical Tensors

- 既存手法と提案手法
  - DCE (2009)
    - 既存手法の中で，より完全なパターンの抽出が可能
  - Multidupehack
    - ノイズ耐性

      ある程度曖昧な許容誤差範囲 $\epsilon$ を設定し，その範囲より大きく離れている場合には除外することで，パターン抽出を邪魔しない
    - 完全な抽出

      Data-Peeler という彼らの先行研究のアルゴリズムを改良

      (深さ優先，再帰的に計算)

# Complete Discovery of High-Quality Patterns in Large Numerical Tensors

- 評価
  - 人工データに対する比較
    - ランダムに生成したテンソルからパターン抽出

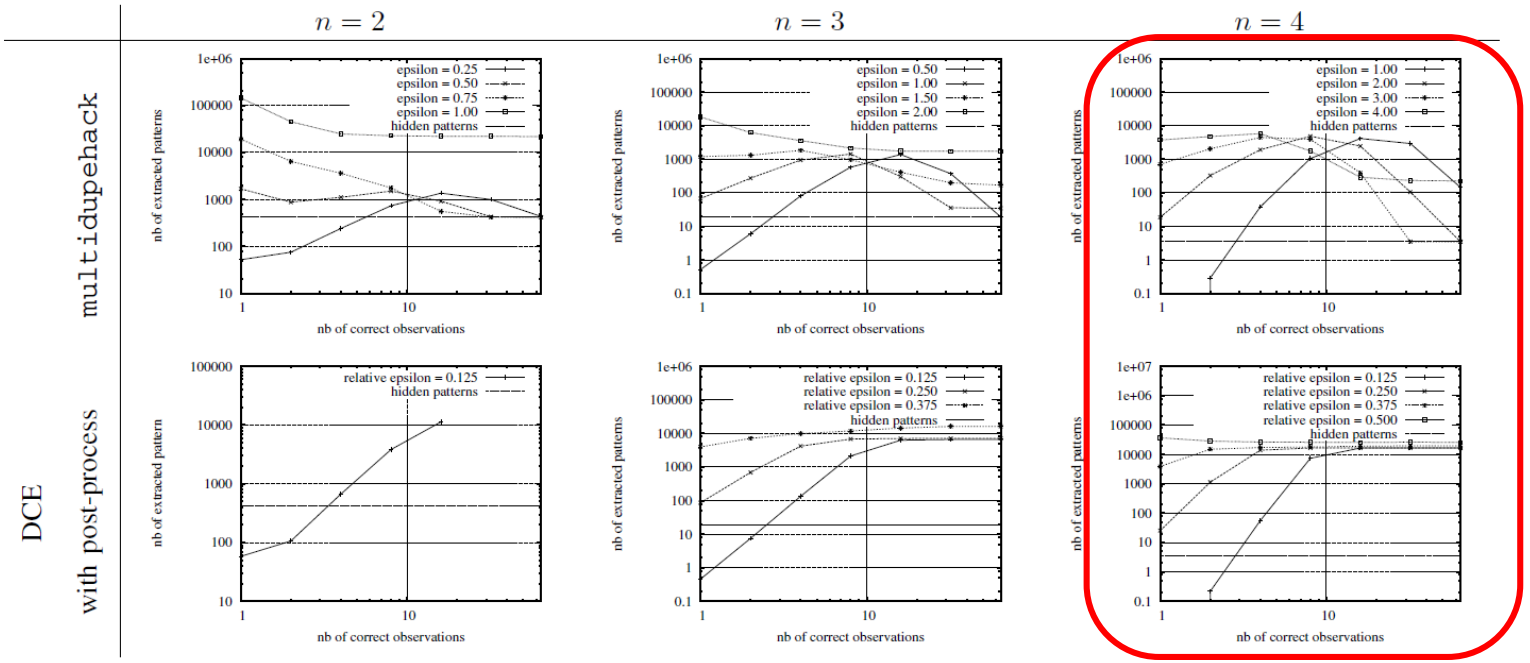
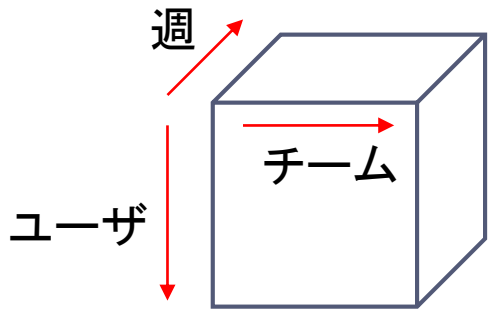



Fig. 6. Number of patterns extracted by `multidupehack` and DCE in synthetic tensors.

# Complete Discovery of High-Quality Patterns in Large Numerical Tensors

- 評価
  - 日常生活でのパターン抽出
    - ツイッターで影響を与えるグループの検出
      3階のテンソル(週，チーム数，ユーザ数)からパターンを抽出
      multidupehack：56 秒
      DCE：100 時間経っても結果でず



# Data Mining II: Pattern Discovery

- Automatic Generation of Question Answer Pairs From Noisy Case Logs
  - Jitenda Ajmera, Sachinda Joshi, Ashish verma, Amoi Mittal
- Complete Discovery of High-Quality Patterns in Large Numerical Tensors
  - Loïc Cerf and Wagner Meira Jr.
- Ranking Item Features by Mining Online User-Item Interactions
  - Sofiane Abbar, Habibur Rahman, Saravanan Thirumuruganathan, Carlos Castillo, Gautam Das

※スライドで用いている図は上記論文からの引用です

- 目的各アイテムの特徴を User-Item Interactionsに基いてランク付け

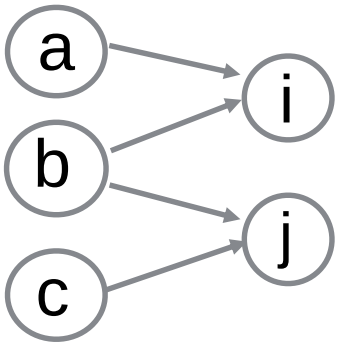
- Feature Ranking(FR) 作成
  - Least Squares(LS)
  - Network Flow(NF)
  - Non-negative Matrix Factorization(NMF)
- 利用する Interactionデータと FR作成の順番に応じて扱う アルゴリズムを変える
  - aggregate level(AGG)：各アイテムと全ユーザーのInteraction
  - individual level(INDIV)：各アイテムと各ユーザーのInteraction
  - matrix W : 各特徴に興味のあるユーザーがそのアイテムに訪問する確率
  - matrix H, vector h : すべての特徴の中でどのくらいのinteractionを持つか

# Ranking Item Features by Mining Online User-Item Interactions

W item-feature　h feature-user　v item-user

User-Item Interaction

3 features　2 items




100 visits

1 visits

$$\begin{array}{c} \\ i \\ j \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 1.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 1.0 \end{pmatrix} \end{array}\begin{pmatrix} h_a \\ h_b \\ h_c \end{pmatrix} = \begin{pmatrix} \frac{100}{101} \\ \frac{1}{101} \end{pmatrix}$$

どちらかを先に割り出す

$$\begin{array}{c} \\ i \\ j \end{array}\begin{pmatrix} W_{ia} & W_{ib} & 0.0 \\ 0.0 & W_{jb} & W_{jc} \end{pmatrix}\begin{pmatrix} \frac{100}{201} \\ \frac{101}{201} \\ \frac{1}{201} \end{pmatrix} = \begin{pmatrix} \frac{100}{101} \\ \frac{1}{101} \end{pmatrix}$$

FR(Feature Ranking) = $W_i \circ h$

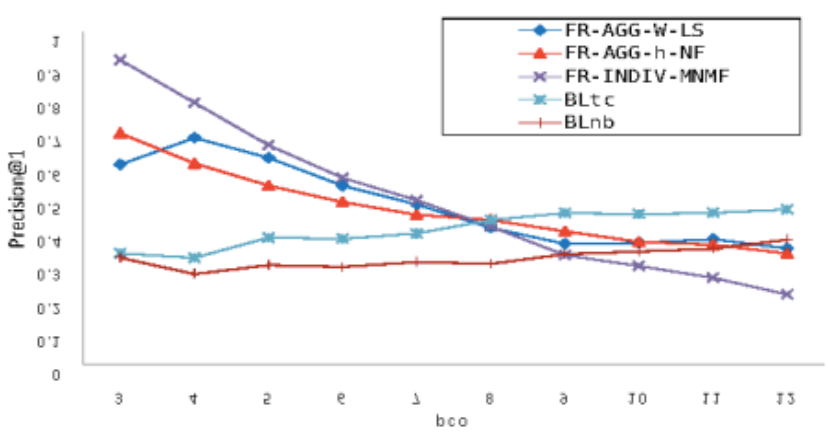
Interaction(AGG or INDIV) → v or V

FR via (W or h) estimation

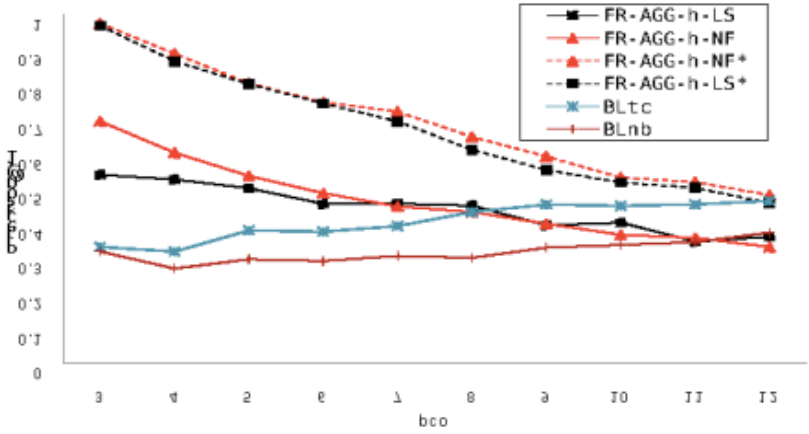
v からW or hを計算
残った方をLS,NF,NMF
を基に割り出し,
$\{W_i \circ h\}$の結果を
ソートすることで
各アイテムごとのFRが作られる

アイテムが含むfeatureの情報は予めわかっている前提で,
v or Vがわかると,W or hも計算できる

# Ranking Item Features by Mining Online User-Item Interactions




(b) Comparative precision@1 curves




(c) precision@1 of different FR−AGG−h variants

- x軸：pco(すべての映画の中で同じ俳優が出現する上限 )
- y軸：予測の精度
- pcoが7以下の時はBLtc(タグクラウド),BLnb(ベイズ)よりも精度が高い

- 新たにvector h* を作ると精度がよくなった.
- h* は, 俳優それぞれの sum( 1 / ある映画の中での,その俳優の順位)で求められる

IMDBの中に映画ごとの俳優の
役柄の重要度が順位が保存されている