Improve Collaborative Filtering through Bordered Block Diagonal Form Matrices

Yongfeng Zhang, Min Zhang, Yiqun Liu, Shaoping Ma
Dept. CS, Tsinghua University, China

Information Retriever @ TsingHua University

Outline

- Backgrounds
- Our Approach
- Experiments
- Discussions
- Conclusions

Backgrounds

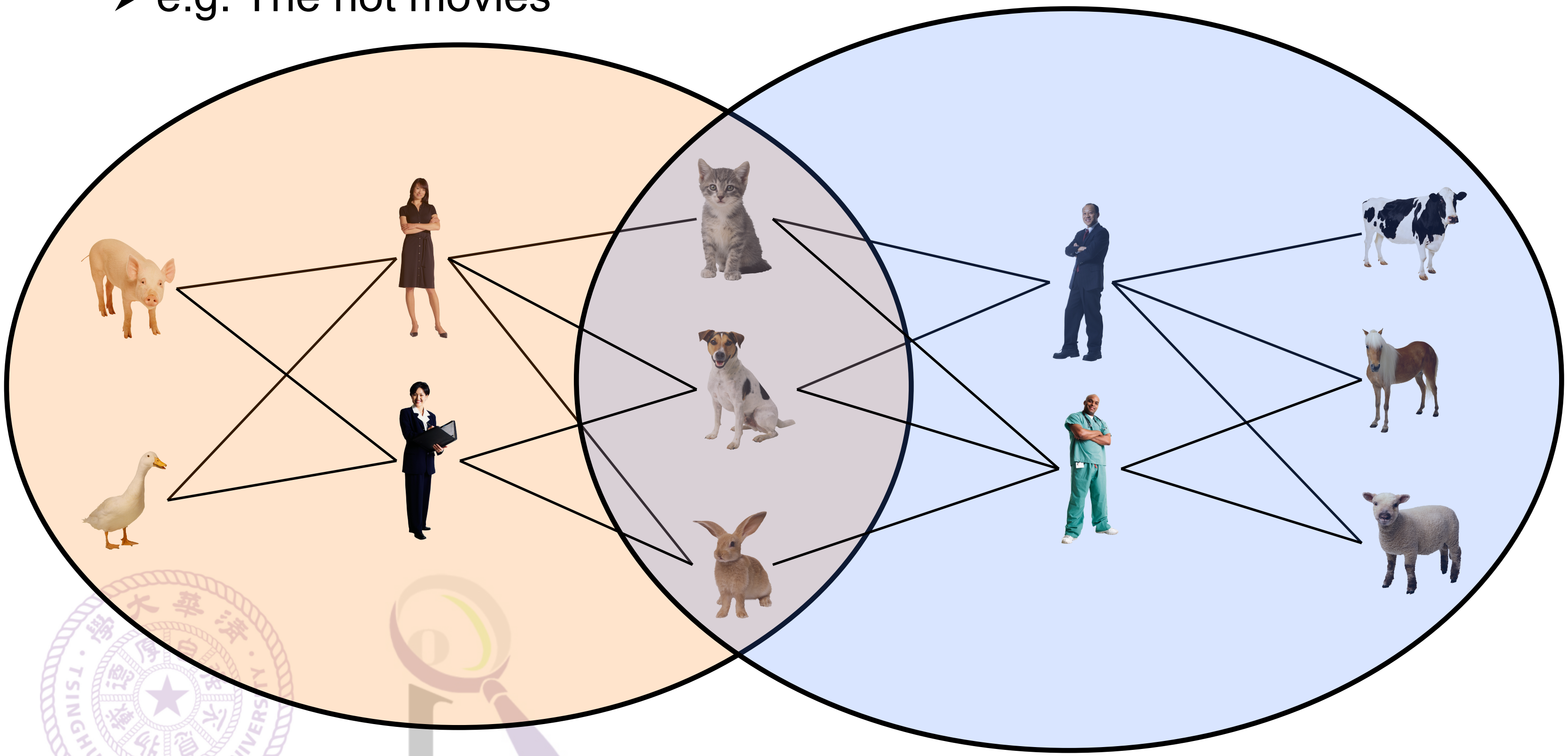
- Recommender Systems
  - Playing an important role on the web
  - E-Commerce and Review Services, e.g. Amazon and Yelp
- Collaborative Filtering
  - The ability to recommend without clear content information
  - Have achieved significant success
- Rating Prediction
  - Make rating predictions on user-item rating matrix based on observed ratings
  - One of the core tasks of CF
  - Widely investigated

Backgrounds

- The use of user-item communities
  - Benefits the efficiency and effect in many cases
- Matrix Clustering
  - Extract user-item sub-matrices (clusters)
  - Conduct Collaborative Filtering on each sub-matrices
- Some existing popular approaches
  - User / Item Clustering [Corner & Herlocker, SIGIR'99]
  - Co-Clustering [George & Merugu, ICDM'05]
  - User-Item Subgroups Mining [Xu & Bu et al, WWW'12]
- Our Concerns
  - Clusters may not be a 'natural' representation of communities
  - Usually forces a user/item to be in a single cluster

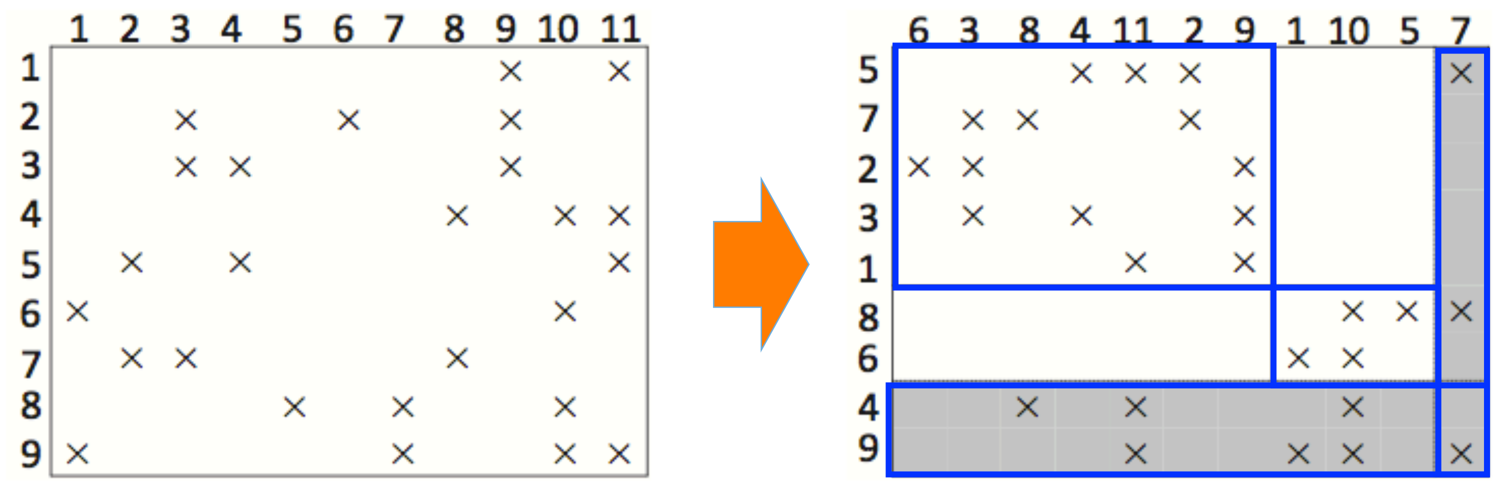
Observations

- Common Interests and Special Interests
  - Common Interests: Items favored by users from different communities
  - Special Interests: items favored by some specific groups of users
  - Common Interests can be shared by different user groups
    - e.g. The hot movies



the BBDF structure

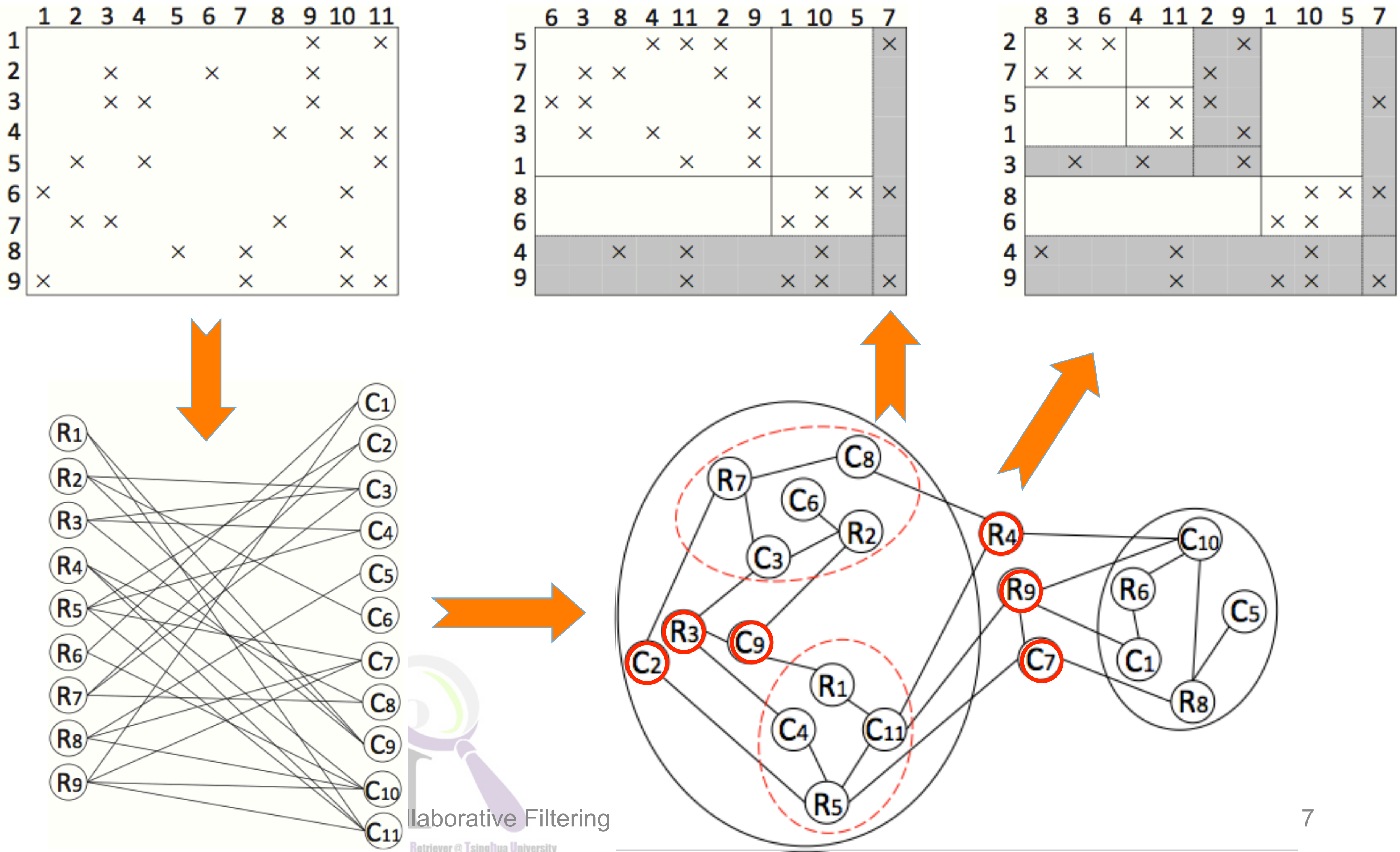
- Bordered Block Diagonal Form (BBDF) structure



- The Intuition
  - Row Borders: Super Users
  - Column Borders: Super Items, e.g. hot movies
  - Diagonal Blocks: User-Item Communities

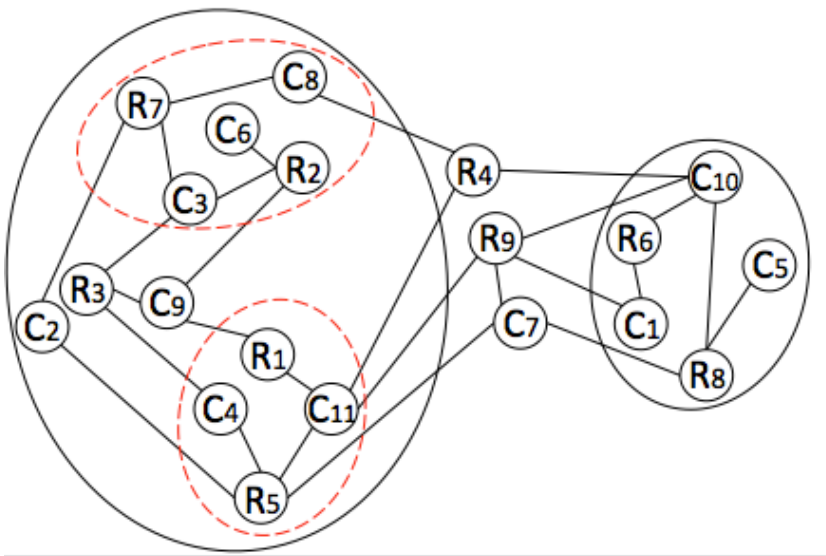
BBDF and GPVS

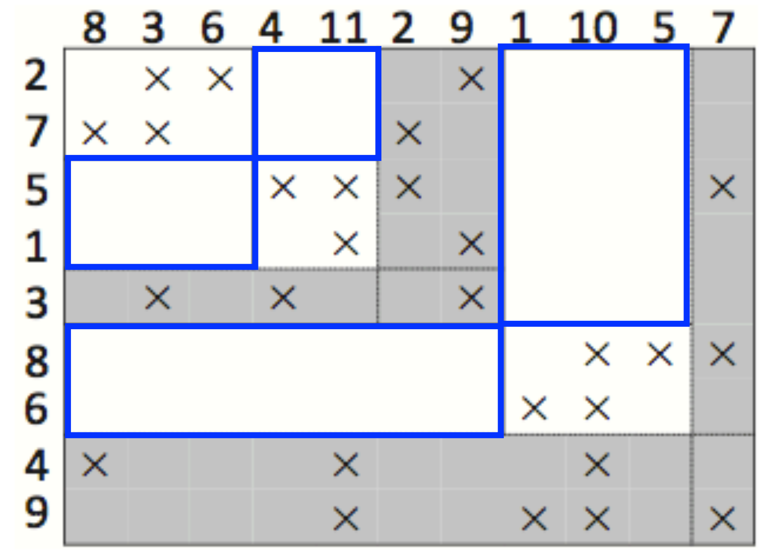
- Graph Partitioning by Vertex Separator (GPVS[Karypis,2011])



the ABBDF structure

- An underlying assumption in BBDF structure.
  - There is no edge between communities.

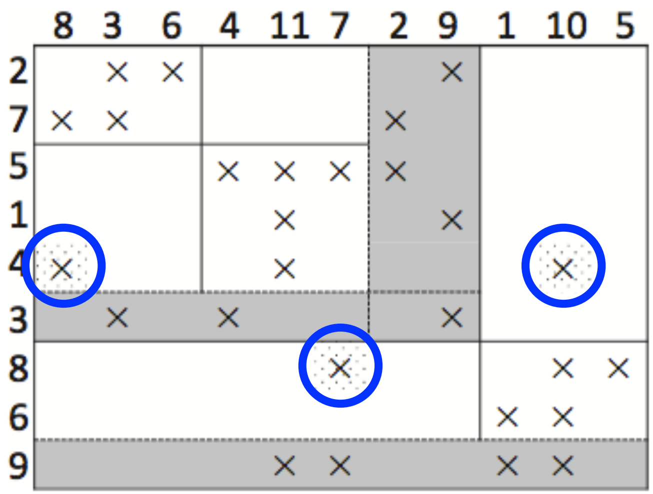


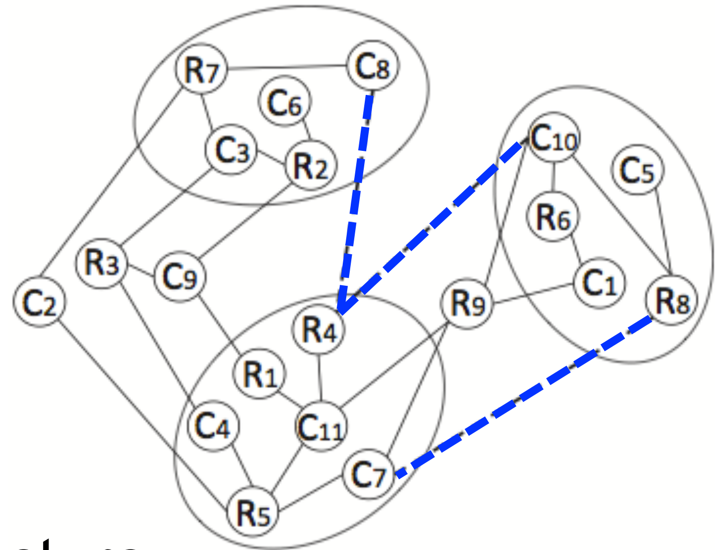


- May not be a reasonable assumption
  - User might indeed focus on some domains
  - They do step into other domains sometimes

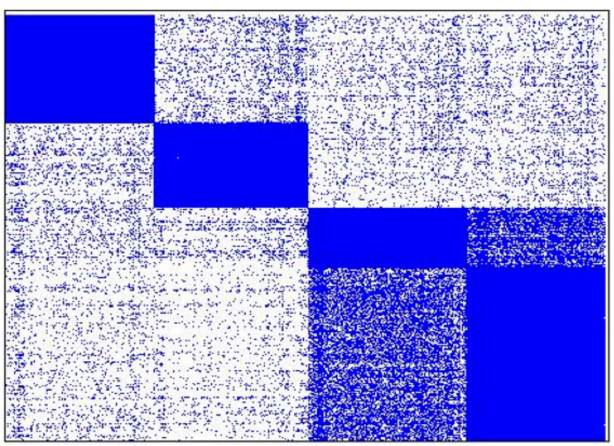
the ABBDF structure

- Approximate Bordered Block Diagonal Form (ABBDF)





- A special form of ABBDF structure



* The ABBDF structure without border
* Can be achieved with Graph Partitioning by Edge Separator (GPES) algorithms
* Remove some edges (non-zeros in off-diagonal areas) and split the graph

(A)BBDF and Community Detection

- More general conclusions

- Any Community Detection result on a bipartite graph can be represented as an ABBDF structure
  - Not only GPVS or GPES algorithms

- Corollary: Can be represented as an BBDF structure if there is no inter-community edge.

Algorithms

- How to permute matrices into (A)BBDF structures?
- BBDF Permutation Algorithm
  - Algorithm1, Basic-BBDF-Permutation procedure
  - Algorithm2, BBDF-Permutation procedure
- ABBDF Permutation Algorithm
  - Algorithm3, ABBDF-Permutation procedure
  - Algorithm4, Improve-Density procedure

BBDF permutation algorithm

- The basic procedure for BBDF permutation

Algorithm 1 Basic-BBDF-Permutation$(A, \mathcal{G})$

Require:

User-Item rating matrix A.

Bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (\mathcal{R} \cup \mathcal{C}, \mathcal{E})$ of A. $\triangleright$ $\mathcal{R}/\mathcal{C}$ are row/column vertex sets of $\mathcal{V}$ correspondingly.

Ensure:

Average density of resulting diagonal blocks $\bar{\rho}$.

1: $\Gamma_v \leftarrow \{\mathcal{V}_1 \mathcal{V}_2 \cdots \mathcal{V}_k; \mathcal{V}_S\} \leftarrow \mathrm{GPVS}(\mathcal{G})$

2: Permute rows of A in order of $\mathcal{R}_1 \mathcal{R}_2 \cdots \mathcal{R}_k \mathcal{R}_S$

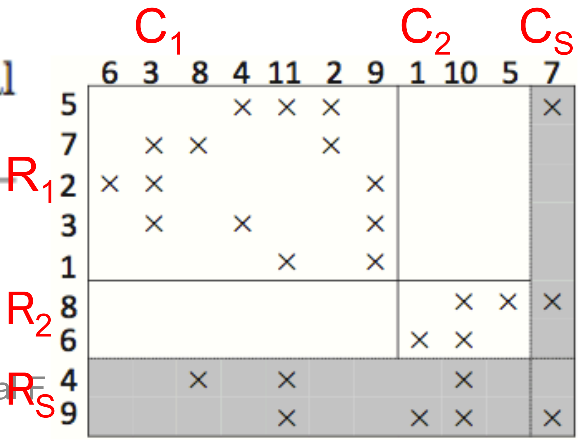
3: Permute columns of A in order of $\mathcal{C}_1 \mathcal{C}_2 \cdots \mathcal{C}_k \mathcal{C}_S$

4: **return** $\bar{\rho}(D_1 D_2 \cdots D_k)$ $\triangleright$ D_i denotes the i-th diagonal block which corresponds to vertex set $\mathcal{V}_i = \mathcal{R}_i \cup \mathcal{C}_i$

Remove a set of vertices V_S and split the graph into k connected components.

Remove the vertex set V_S to borders and permute the reaming to diagonals

Return the average density of resulting diagonal blocks in this stage



BBDF permutation algorithm (cont.)

- BBDF Permutation algorithm
  - Permute sub-matrices into BBDF structure recursively

Algorithm 2 BBDF-Permutation($A, \mathcal{G}, \rho$)

Require:

User-Item rating matrix A.

Bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of A.

Density requirement ρ.

Ensure:

Matrix A permuted into BBDF structure.

```
1: ρ_A ← ρ(A)
2: if ρ_A < ρ then                          ▷ else do nothing
3:    ρ̄ ← Basic-BBDF-Permutation(A, G)
4:    if ρ̄ > ρ_A then                       ▷ else do nothing
5:       for each diagonal block D_i in A do
6:          BBDF-Permutation(D_i, G_{V_i}, ρ)   ▷ V_i denotes
   the vertex set of D_i, G_{V_i} is the subgraph induced by V_i
7:       end for
8:    end if
9: end if
```

The expected minimum average density of diagonal blocks

If the density of a sub-matrix has not reached the requirement, split it using the basic procedure

If the average density improves after split, take the split and recurse. Else, stop recursion.

ABBDF permutation algorithm

Algorithm 3 ABBDF-Permutation$(A, \mathcal{G}, \rho)$

Require:

User-Item rating matrix A.
Bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (\mathcal{R} \cup \mathcal{C}, \mathcal{E})$ of A.
Density requirement ρ.

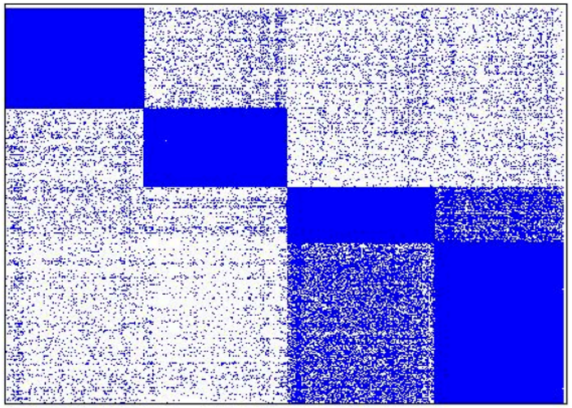
Ensure:

Matrix A permuted into ABBDF structure.

```
if ρ(A) ≥ ρ then
    return
else
    Γ_e ← {V_1 V_2 ··· V_k} ← GPES(G)
    Permute rows of A in order of R_1 R_2 ··· R_k
    Permute columns of A in order of C_1 C_2 ··· C_k
    {V'_1 V'_2 ··· V'_k; V'_S} ← Improve-Density(A, G, Γ_e)
    for each diagonal block D_i in A do
        ABBDF-Permutation(D_i, G_{V'_i}, ρ)
    end for
end if
```

Split the corresponding graph using GPES, resulting in a ABBDF matrix without borders.

If the average density of diagonal blocks didn't improve, try to improve it by moving some rows/ columns to borders.



ABBDF permutation algorithm(cont.)

Algorithm 4 Improve-Density$(A, \mathcal{G}, \Gamma_e)$

Require:
User-Item rating matrix A.
Bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (\mathcal{R} \cup \mathcal{C}, \mathcal{E})$ of A.
GPES result $\Gamma_e = \{\mathcal{V}_1 \mathcal{V}_2 \cdots \mathcal{V}_k\}$ of $\mathcal{G}$.

Ensure:
Average density of diagonal blocks greater than $\rho(A)$.

$\{\mathcal{V}'_1 \mathcal{V}'_2 \cdots \mathcal{V}'_k; \mathcal{V}'_S\} \leftarrow \{\mathcal{V}_1 \mathcal{V}_2 \cdots \mathcal{V}_k; \emptyset\}$
while $\bar{\rho}(D_1 D_2 \cdots D_k) < \rho(A)$ **do**
  $l', i' \leftarrow 0, \bar{\rho}' \leftarrow 0$
  for each diagonal block D_i **do**
    for each line l in D_i **do**
      $\bar{\rho} \leftarrow \frac{\sum_{j=1}^{k} \mathrm{n}(D_j) - \mathrm{n}(l(D_i))}{\sum_{j=1}^{k} \mathrm{area}(D_j) - \mathrm{area}(l(D_i))}$
      if $\bar{\rho} > \bar{\rho}'$ **then**
        $l' \leftarrow l, i' \leftarrow i, \bar{\rho}' \leftarrow \bar{\rho}$
      end if
    end for
  end for
  Permute line l' to borders
  $\mathcal{V}'_{i'} \leftarrow \mathcal{V}'_{i'} - \{\mathrm{node}(l')\}$
  $\mathcal{V}'_S \leftarrow \mathcal{V}'_S \cup \{\mathrm{node}(l')\}$ ▷ node(l') denotes the node in $\mathcal{V}'_{i'}$ corresponding to line l'
end while
return $\{\mathcal{V}'_1 \mathcal{V}'_2 \cdots \mathcal{V}'_k; \mathcal{V}'_S\}$

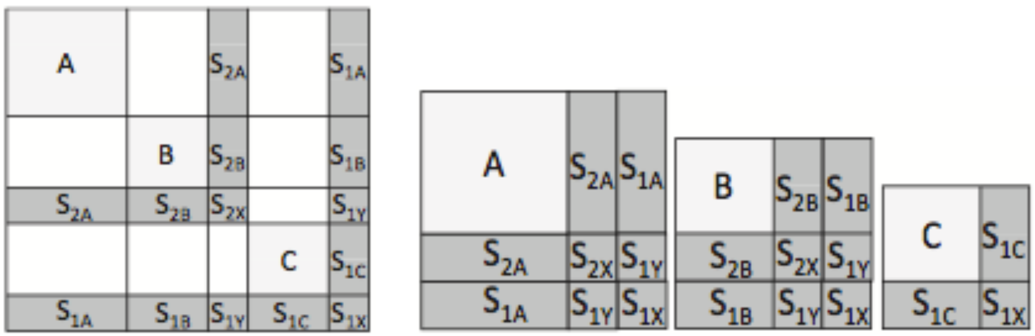
For each row and column from each diagonal block, check whether its removal improves average density

Permute the row/column to borders whose removal improves average density most

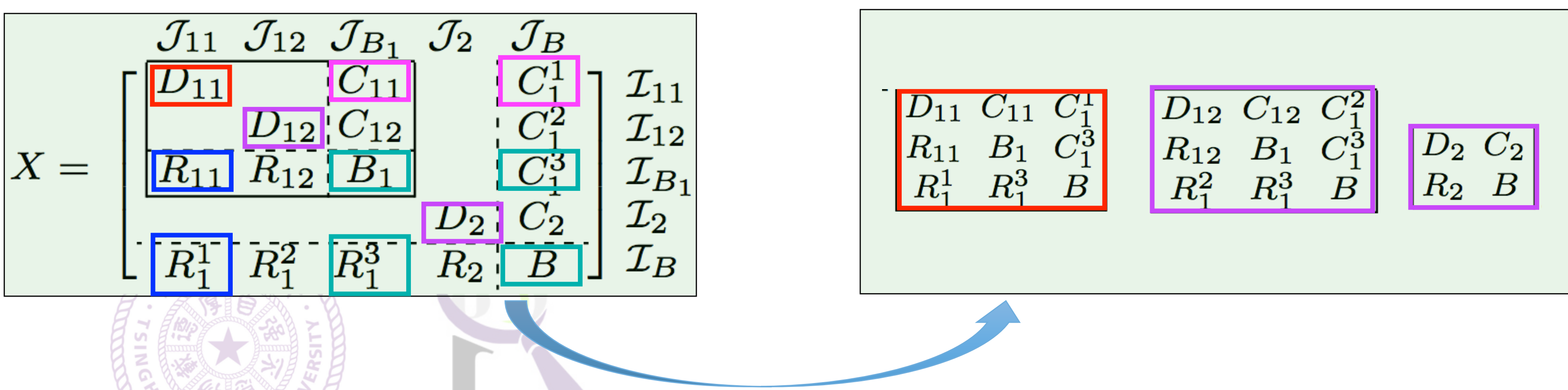
Until average density is higher than the original matrix

Make Rating Predictions

- Extract sub-matrices representing communities from the (A)BBDF structure

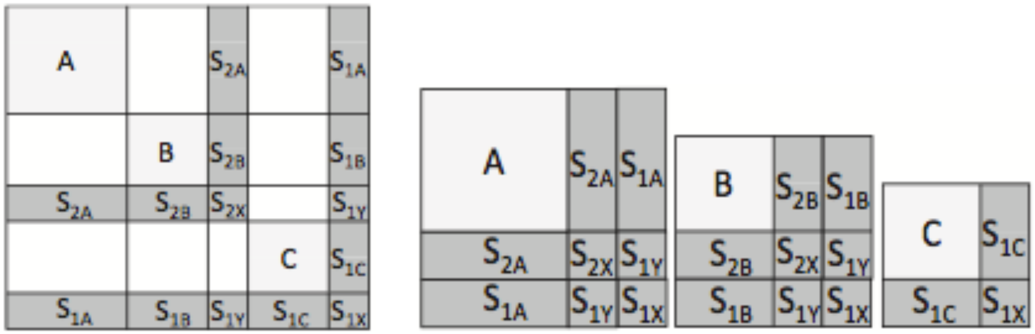


(a) BBDF matrix (b) Submatrices extracted



Make Rating Predictions (cont.)

- Make rating predictions in 2 steps:
  - Step1 : Conduct CF in each of the sub-matrices
  - Step2: Average predictions in duplicated blocks
    - E.g. S_{2x} is predicted twice in sub-matrices A and B

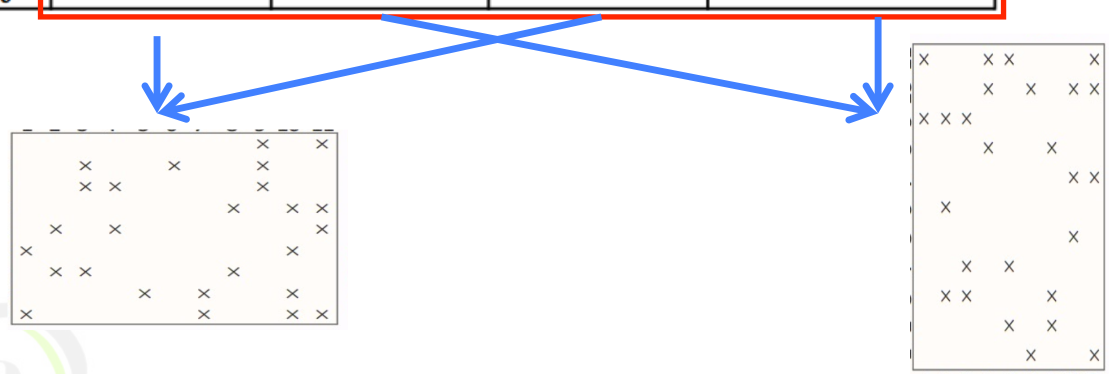


(a) BBDF matrix (b) Submatrices extracted

Experiment Setup

- Dataset Description
  - 4 real-world datasets: MovieLens-100k, MovieLens-1m, Dianping, and Yahoo! Music.

	ML-100K	ML-1M	DianPing	Yahoo!Music
#users	943	6,040	11,857	1,000,990
#items	1,682	3,952	22,365	624,961
#ratings	100,000	1,000,209	510,551	256,804,235
#ratings/user	106.045	165.598	43.059	256.550
#ratings/item	59.453	253.089	22.828	410.912
average density	0.0630	0.0419	0.00193	0.000411



Experiment Setup (cont.)

- Experimented the framework on 4 CF algorithms
  - User-based
  - Item-based
  - SVD (Singular Value Decomposition)
  - NMF (Nonnegative Matrix Factorization)

- Evaluation Metric
  - Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(r_i - \hat{r}_i)^2}{N}}$$

Community Analysis

- Density requirement v.s. # diagonal blocks
  - Low density -> A small number of big communities
  - High density -> A large number of small communities
- Example of BBDF permutation results on DianPing

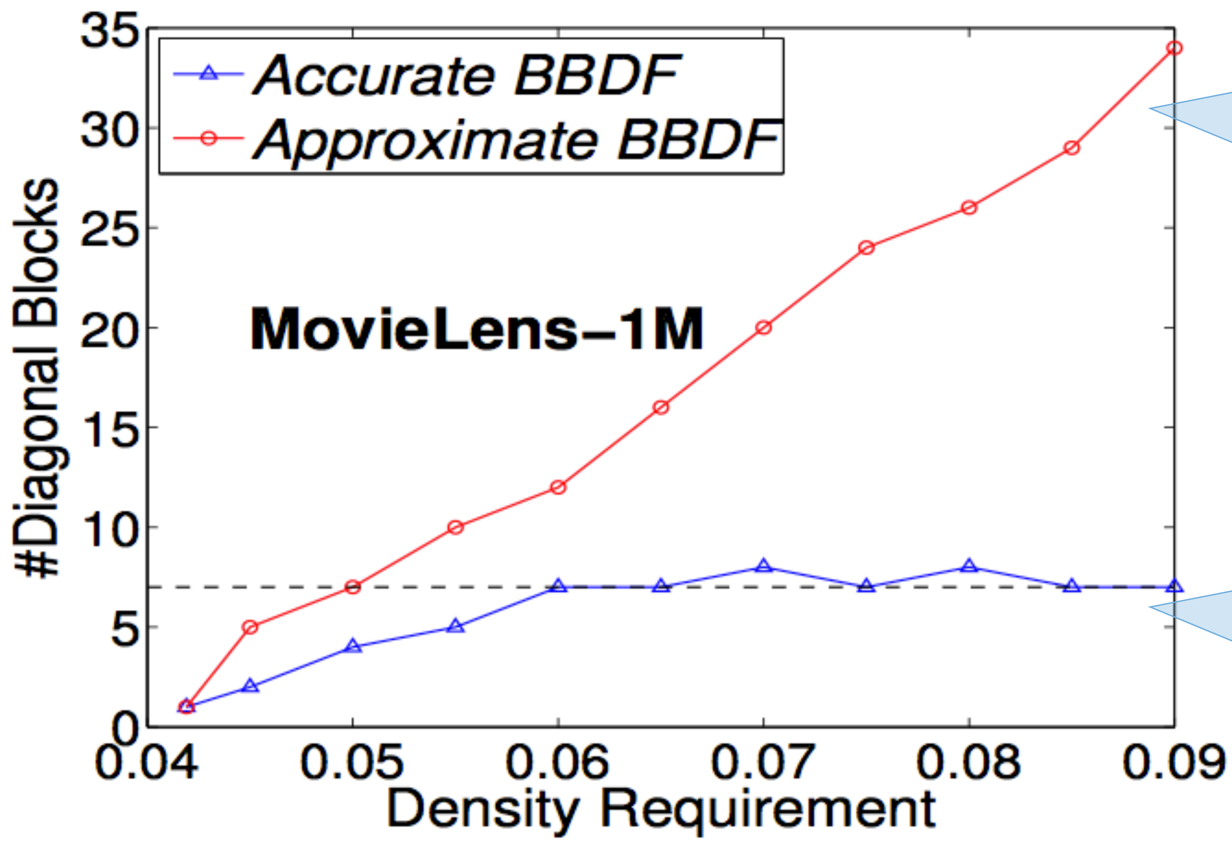
(a) BBDF $\rho = 0.005$ (b) BBDF $\rho = 0.01$

Community Analysis (cont.)

- An appropriate density requirement gives reasonable community detection results.

Community Analysis (cont.)

- Density requirement v.s. # diagonal blocks

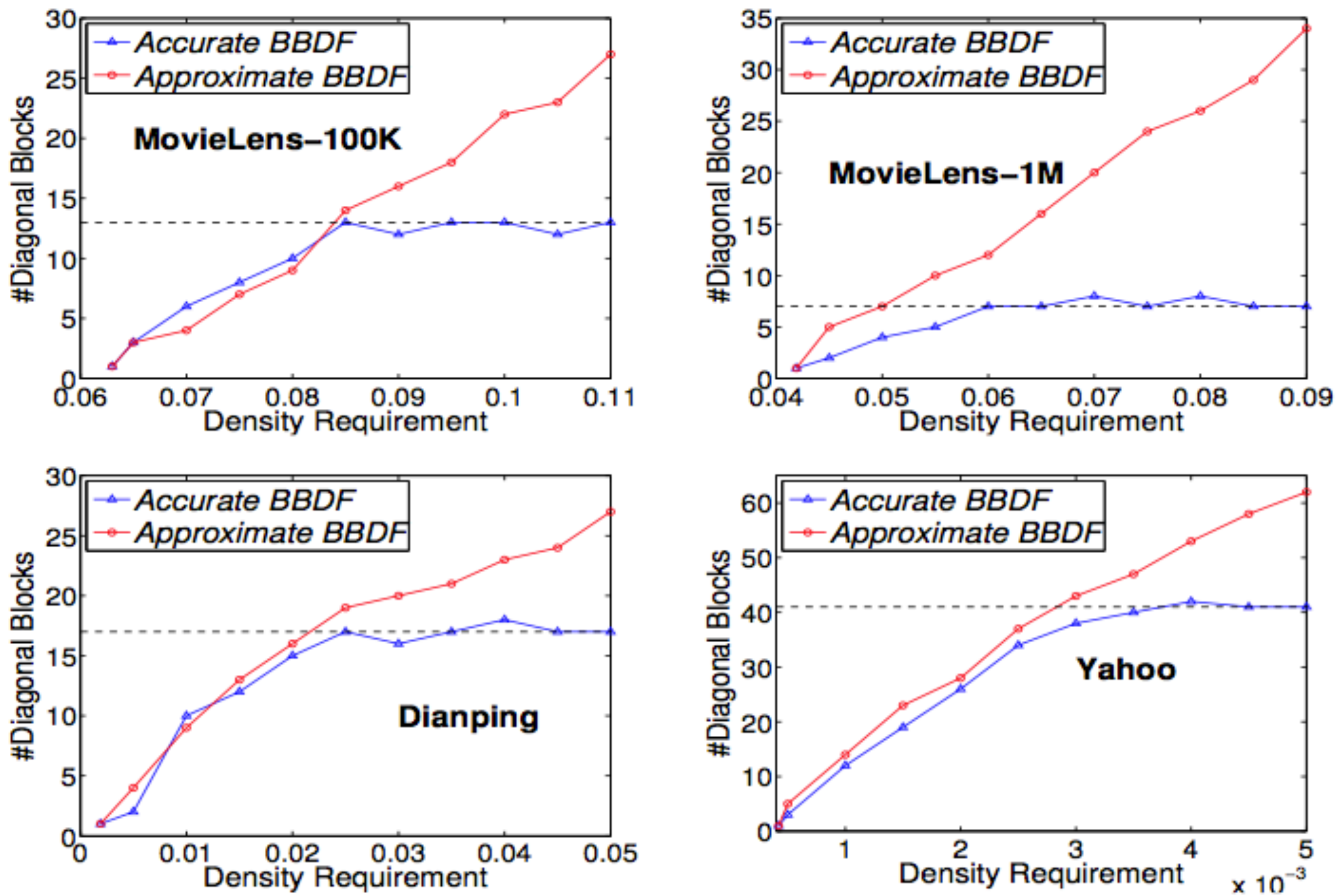


ABBDF: # Diagonal Blocks grows consistently with the growth of density requirement

BBDF: # Diagonal Blocks grows at first and tends to be stable at last

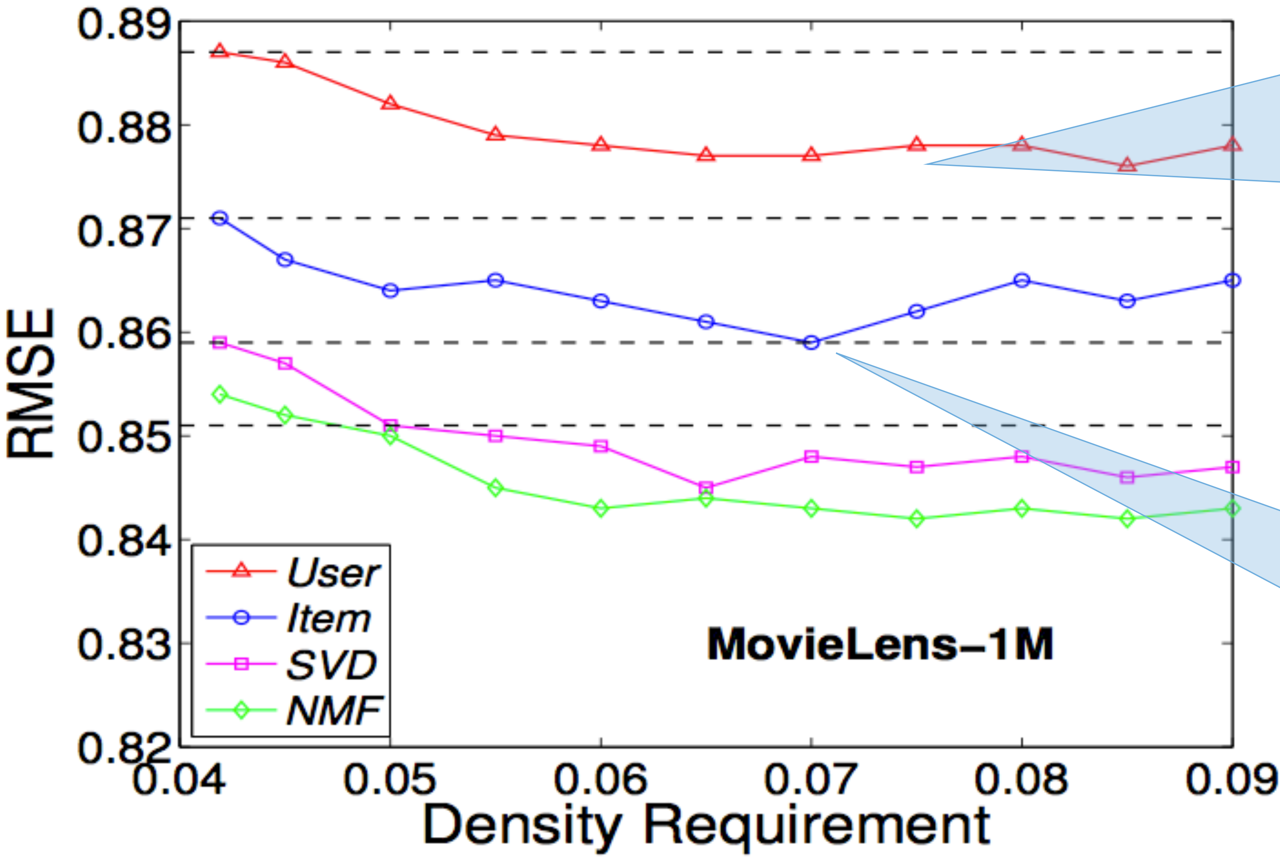
Community Analysis (cont.)

- Similar results are observed on the other datasets



Prediction Accuracy

- BBDF: RMSE v.s. Density Requirements

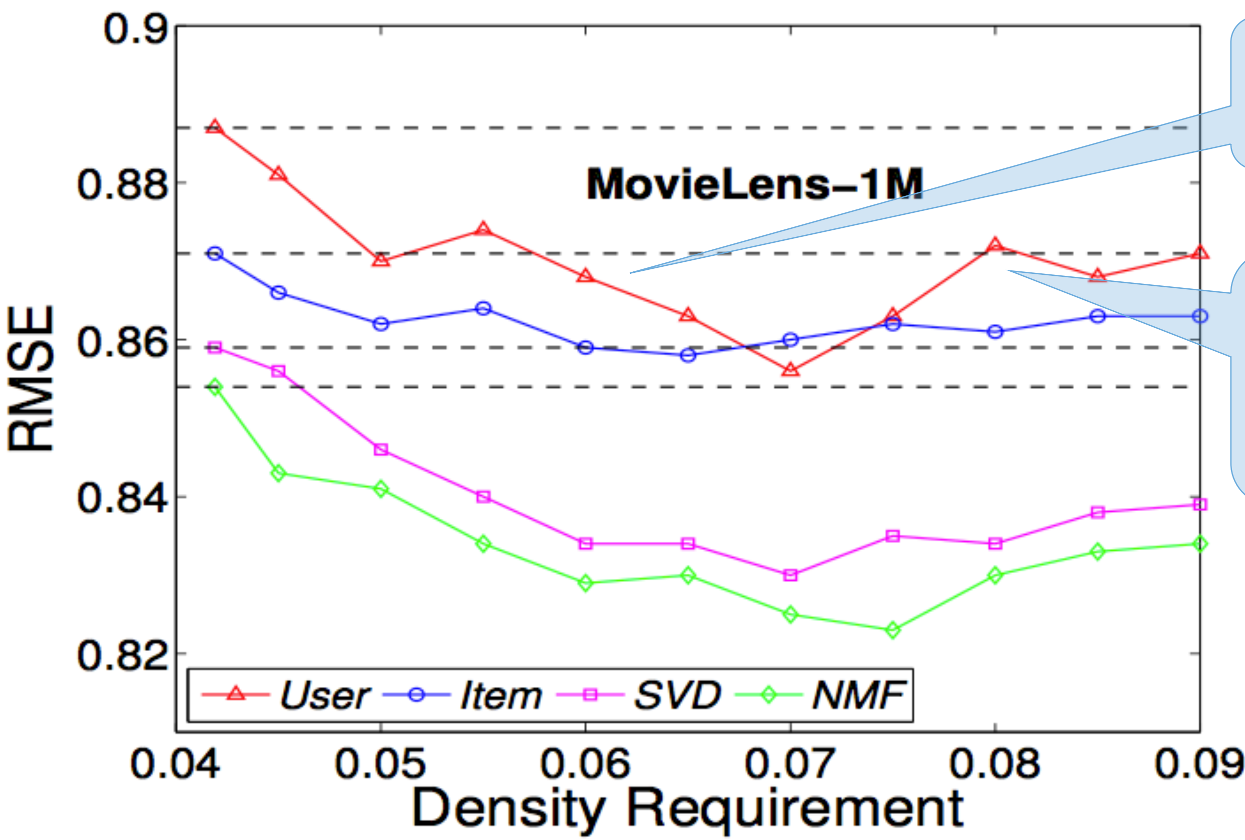


Prediction accuracy tends to be stable (the BBDF algorithm stop to split matrices when density requirement is too high.)

Gains better prediction accuracy given appropriate density requirement

Prediction Accuracy (cont.)

- ABBDF: RMSE v.s. Density Requirements



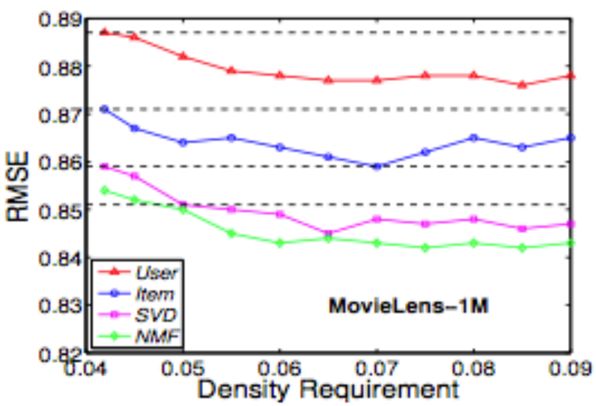
Tends to gain better prediction accuracy at first

But the performance tends to drop rapidly given high density requirements.

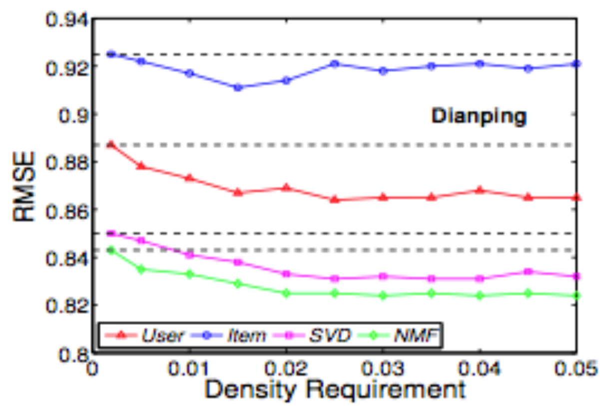
The algorithm leads to many small scattered communities

Prediction Accuracy (cont.)

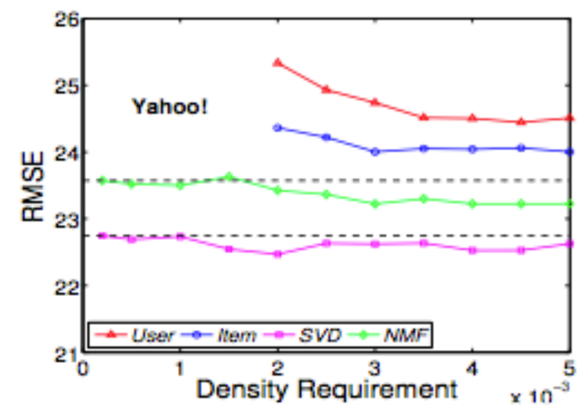
- Similar results were observed on the other datasets



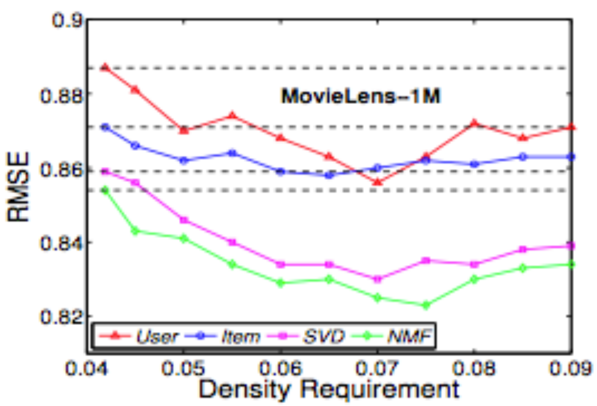
(b) acc.BBDF ML-1M

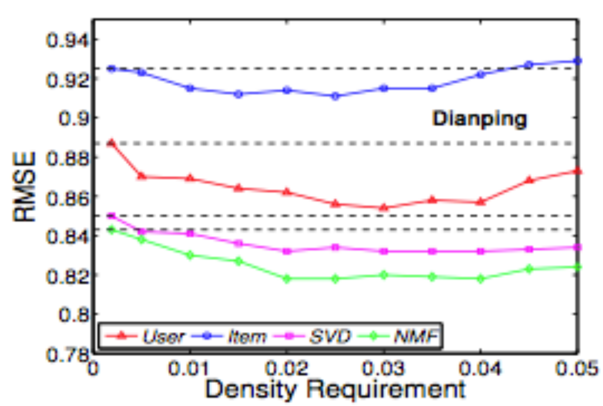


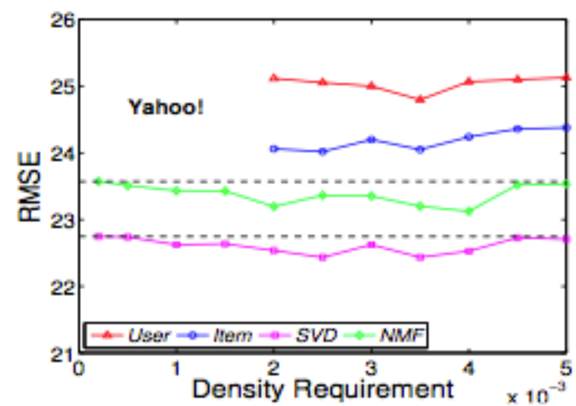
(c) acc.BBDF Dianping



(d) acc.BBDF Yahoo!

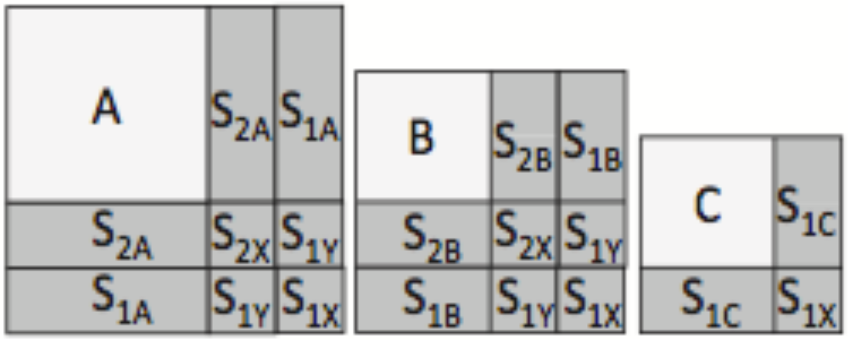






Discussions

- Potential advantage: Selective re-training in practical systems
  - Ratings are made by users continuously in real-world systems
  - Have to retrain a CF model every period of time
  - Only need to retrain those really in need of re-training
    - E.g. The RMSE has reached a criterion



Wrap up

- In this work:
  - Investigated the relationship between (A)BBDF structure and community detection
  - Designed density-based algorithms to transform a matrix into (A)BBDF structure
  - Proposed a framework to make rating predictions on this structure
- Future directions
  - (A)BBDF structure is independent of specific community detection algorithm
    - Investigate other kinds of (A)BBDF permutation algorithms except for GPVS and GPES
  - Conduct selective re-training using our framework

Thanks!