Discrete Collaborative Filtering

Hanwang Zhang[1], Fumin Shen[2], Wei Liu[3],
Xiangnan He[1], Huanbo Luan[4], Tat-Seng Chua[1]

Presented by Xiangnan He

1. National University of Singapore
2. University of Electronic Science and Technology of China
3. Tencent Research
4. Tsinghua University

19 July 2016

Highlights of
Discrete ***C***ollaborative ***F***iltering

Online Recommendation

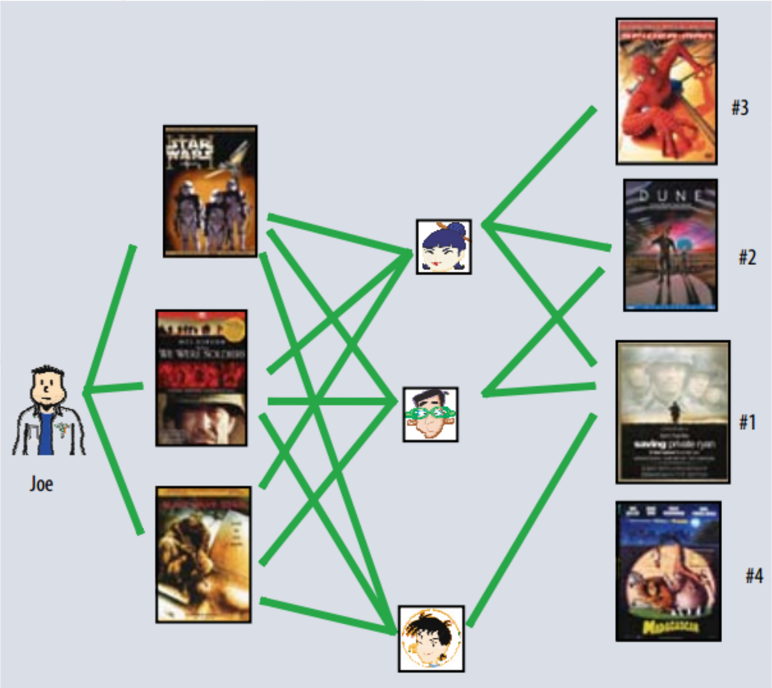
- An ***Efficient*** Recommender System
- Latent Model: ***Binary*** Representation for Users and Items
- Recommendation as ***Search*** with Binary Codes

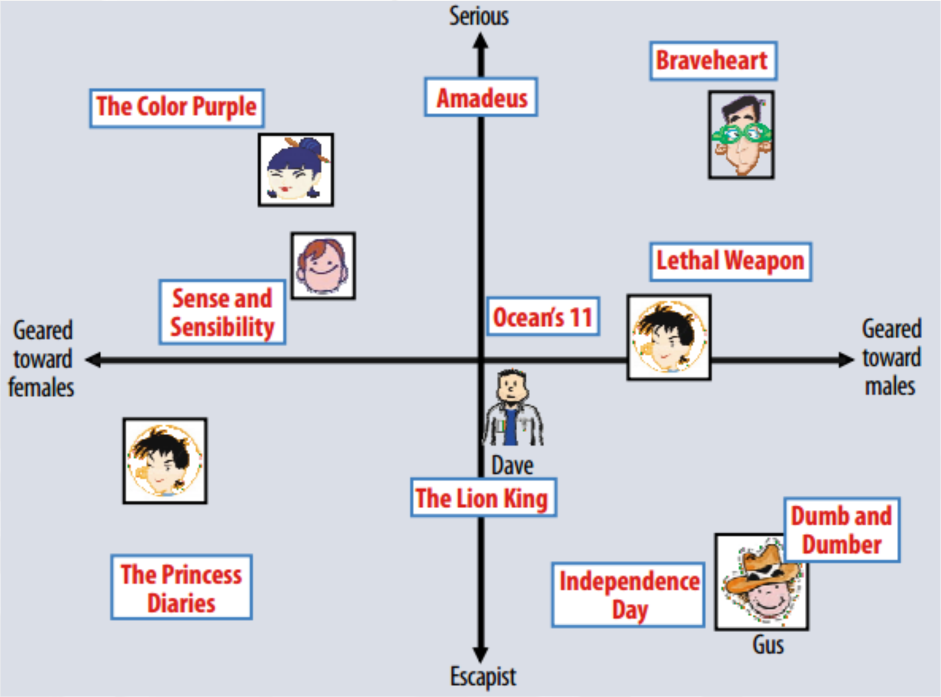
Offline Training

- ***End-to-end*** binary optimization
- ***Balanced*** and ***Decorrelated*** Constraint
- Small ***SVD*** + ***Discrete*** Coordinate Descent

Collaborative Filtering

Latent Factor Approach [Koren et al. 2009]





User-Item Matrix

Latent Space

Efficient CF:
Hashing Users & Items

Recommendation is ***Search***

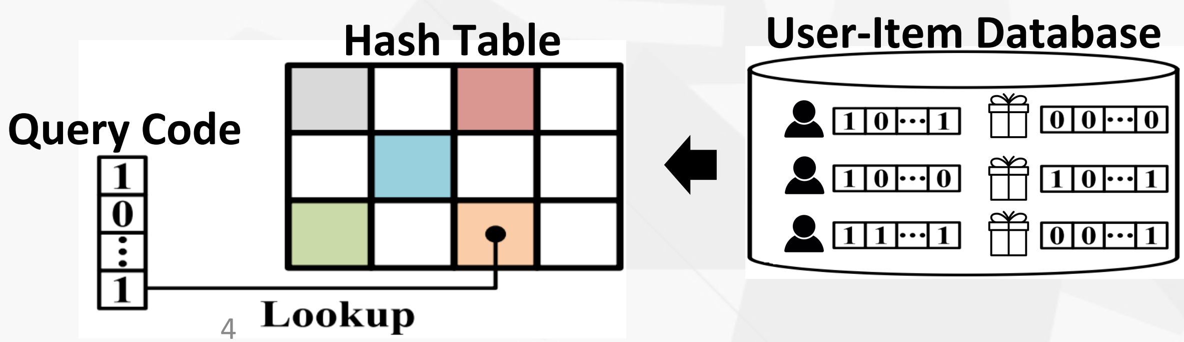
Ranking by **<user vector, item vector>**

Search in Euclidean space is ***slow***

Requires **float** operations & **linear** scan of the data

Search in ***Hamming Space*** is ***fast.***

Only requires **XOR** operation & **constant-time** lookup



Existing Hashing for CF: Two-Stage Approach

[Zhang et al, SIGIR'14; Zhou et al, KDD'12]

- Stage 1: Relaxed Real-Valued Problem

  {B, D} ← Continuous CF Methods

- Stage 2: Binarization

  B ← sgn (B), D ← sgn (D)

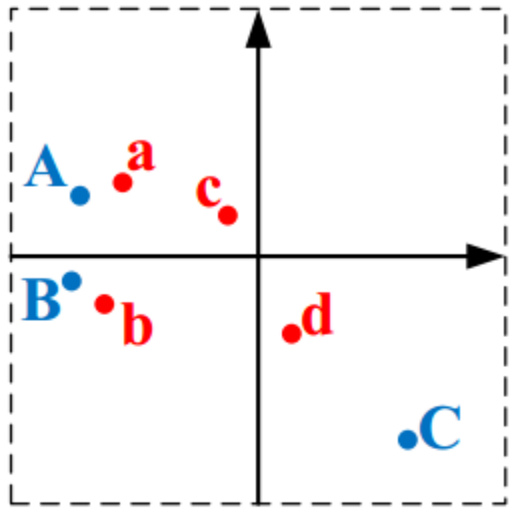
Code learning and CF are isolated

Quantization Loss

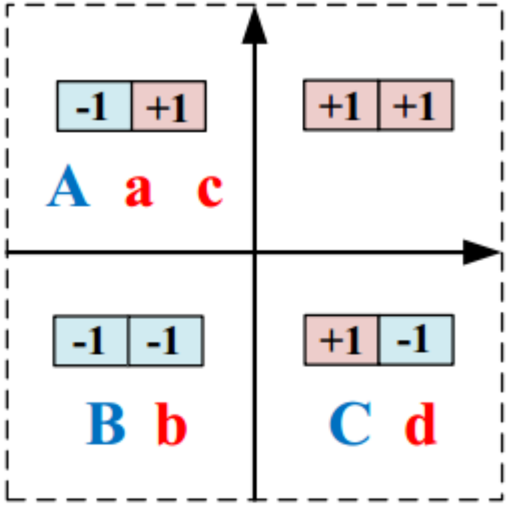
Quantization Loss

User-Item Matrix

	a	b	c	d
A	1	.8	.4	?
B	.8	.8	?	.4
C	.2	?	.2	.4



Relaxed Solution



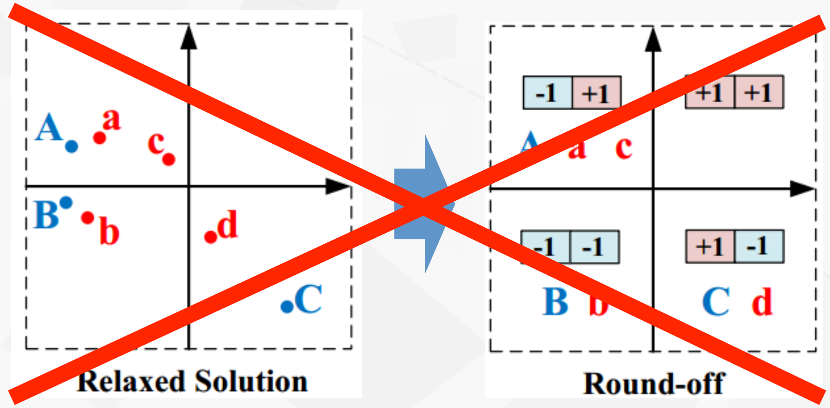
Round-off

1. A,B,a,b are close but they are separated into different quadrants
2. C, d should be far but they are assigned to the same quadrant

Tackling Quantization Loss

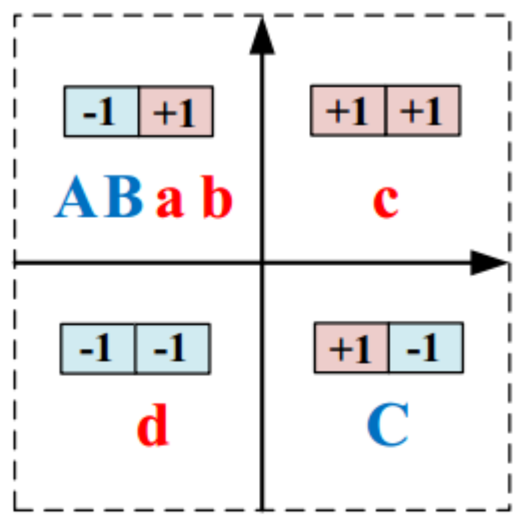
User-Item Matrix

	a	b	c	d
A	1	.8	.4	?
B	.8	.8	?	.4
C	.2	?	.2	.4



Relaxed Solution

Round-off



Discrete Optimization

Intuitive Solution: Binary Constraints

Observed rating / User code / Item code

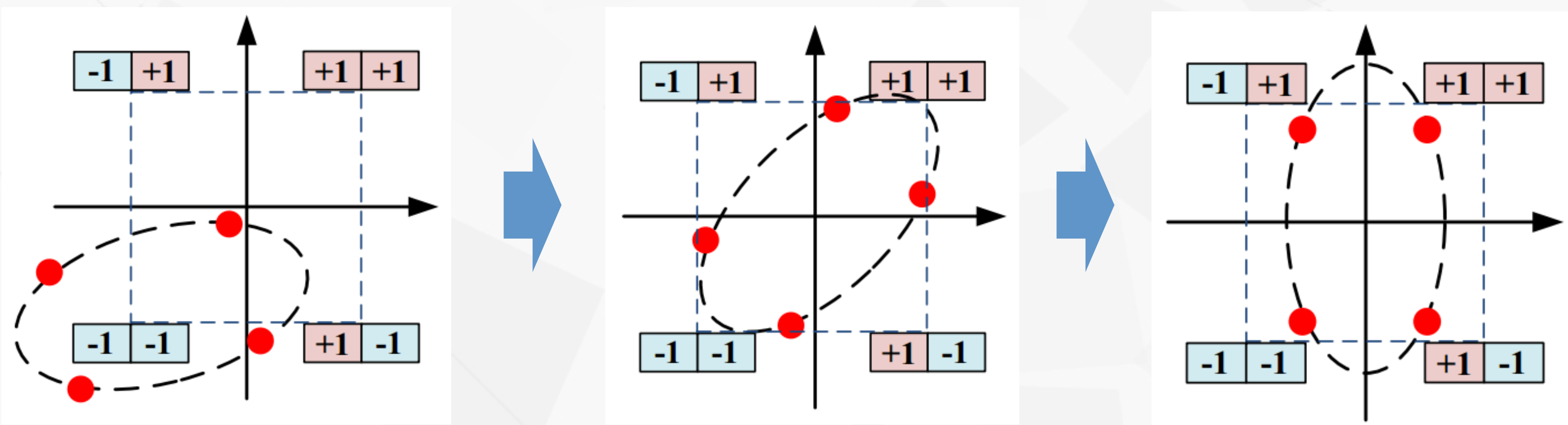
$$\underbrace{\underset{\mathbf{B},\mathbf{D}}{\operatorname{argmin}} \sum_{i,j\in\mathcal{V}} \left(S_{ij} - \mathbf{b}_i^T \mathbf{d}_j\right)^2}_{\text{Rating Prediction}} \qquad \underbrace{\mathbf{B} \in \{\pm 1\}^{r\times m}, \mathbf{D} \in \{\pm 1\}^{r\times n}}_{\text{Binary Constraint}}$$

However, it may lead to non-informative codes, e.g.:

1. Unbalanced Codes → each bit should have split the dataset evenly
2. Correlated Codes → each bit should be as independent as possible

Balanced and Decorrelated Constraint

Illustration of the effectiveness of the two constraints in DCF



Without any constraints:
3 points are (-1, -1) and 1 point is (+1, -1), which is not discriminative.

Balanced:
Separated in the 1^{st} & 3^{rd} quadrant

Decorrelated:
Well separated

Solution with the Two Constraints

$$\underset{\mathbf{B},\mathbf{D}}{\operatorname{argmin}} \sum_{i,j\in\mathcal{V}} \left(S_{ij} - \mathbf{b}_i^T \mathbf{d}_j\right)^2,$$

$$s.t.\ \mathbf{B} \in \{\pm 1\}^{r\times m}, \mathbf{D} \in \{\pm 1\}^{r\times n}$$

$$\underbrace{\mathbf{B1} = 0, \mathbf{D1} = 0,}_{\text{Balanced Partition}}\ \underbrace{\mathbf{BB}^T = m\mathbf{I}, \mathbf{DD}^T = n\mathbf{I}}_{\text{Decorrelation}}$$

However, the hard constraints of zero-mean and orthogonality may not be satisfied in Hamming space!

Our DCF Formulation

Objective function:

$$\underbrace{\sum_{i,j\in\mathbb{V}}(S_{ij}-\mathbf{b}_i^T\mathbf{d}_j)^2}_{\text{Rating Prediction}} + \underbrace{2\alpha||\mathbf{B}-\mathbf{X}||^2+2\beta||\mathbf{D}-\mathbf{Y}||^2}_{\text{Constraint Trade-off}}$$

Binary Constraint

$$\mathbf{B}\in\{\pm1\}^{r\times m}, \mathbf{D}\in\{\pm1\}^{r\times n}$$

Delegate Code Quality Constraint

$$\overbrace{\mathbf{X1}=0, \mathbf{Y1}=0,}^{\text{Balanced Constraint}} \overbrace{\mathbf{XX}^T=m\mathbf{I}, \mathbf{YY}^T=n\mathbf{I}}^{\text{Decorrelated Constraint}}$$

Mixed-Integer Programming NP-Hard [Hastad 2001]

Our Solution: Alternating Optimization

Alternative Procedure

- B-Subproblem

$$\underset{\mathbf{B}}{\text{argmin}} \sum_{i,j \in \mathcal{V}} \left(S_{ij} - \mathbf{b}_i^T \mathbf{d}_j\right)^2 - 2\alpha tr(\mathbf{B}^T \mathbf{X}) \; s.t., \mathbf{B} \in \{\pm 1\}^{r \times m}$$

- D-Subproblem

$$\underset{\mathbf{D}}{\text{argmin}} \sum_{i,j \in \mathcal{V}} \left(S_{ij} - \mathbf{b}_i^T \mathbf{d}_j\right)^2 - 2\beta tr(\mathbf{D}^T \mathbf{Y}) \; s.t., \mathbf{D} \in \{\pm 1\}^{r \times n}$$

- X-Subproblem

$$\underset{\mathbf{X}}{\text{argmin}} \; -2\alpha tr(\mathbf{B}^T \mathbf{X}) \; s.t., \mathbf{X}\mathbf{1} = 0, \mathbf{X}\mathbf{X}^T = m\mathbf{I}$$

- Y-Subproblem

$$\underset{\mathbf{X}}{\text{argmin}} \; -2\beta tr(\mathbf{D}^T \mathbf{Y}) \; s.t., \mathbf{Y}\mathbf{1} = 0, \mathbf{Y}\mathbf{Y}^T = n\mathbf{I}$$

B-Subproblem for Binary Codes

Objective Function

$$\underset{\mathbf{B}}{\operatorname{argmin}} \sum_{i,j\in\mathcal{V}} \left(S_{ij} - \mathbf{b}_i^T\mathbf{d}_j\right)^2 - 2\alpha tr(\mathbf{B}^T\mathbf{X})\ s.t., \mathbf{B} \in \{\pm 1\}^{r\times m}$$

For each user code ***b**$_i$*, optimize ***bit by bit***

for *i=1* **to** m **do** — Parallel **for** loop over ***m*** users

  repeat — Usually converges in **5** iterations

    for *k=1* **to** r **do** — **for** loop over ***r*** bits

      $\hat{b}_{ik} \leftarrow \sum_{j\in\mathcal{V}_i}\left(S_{ij} - \mathbf{d}_{j\bar{k}}^T\mathbf{b}_{i\bar{k}}\right) d_{jk} + \alpha x_{ik};$

      $b_{ik} \leftarrow \operatorname{sgn}\left(K(\hat{b}_{ik}, b_{ik})\right);$

    end

  until *converge*;

end

D-Subproblem can be solved in a similar way

B-Subproblem Complexity

#bits

#bit-by-bit iterations

#computing threads

$$\mathcal{O}(r^2 T_s |\mathcal{V}|/p)$$

#training ratings

Linear to data size!

X-Subproblem for Code Delegate

Objective Function

$$\underset{\mathbf{X}}{\operatorname{argmin}} \; -2\alpha tr(\mathbf{B}^T\mathbf{X}) \quad s.t., \mathbf{X}\mathbf{1} = 0, \mathbf{X}\mathbf{X}^T = m\mathbf{I}$$

r x ***m*** row-centered user code matrix

Small SVD ***r*** x ***m***

$$\left([\mathbf{P}_b\widehat{\mathbf{P}}_b], \mathbf{Q}_b\right) \leftarrow \mathrm{SVD}\boxed{(\overline{\mathbf{B}})}$$

Orthogonalization

$$\widehat{\mathbf{Q}}_b \leftarrow \mathrm{GramSchmidt}\left([\mathbf{Q}_b \mathbf{1}]\right)$$

$$\mathbf{X} \leftarrow \sqrt{m}[\mathbf{P}_b \; \widehat{\mathbf{P}}_b][\mathbf{Q}_b \; \widehat{\mathbf{Q}}_b]^T$$

Y-Subproblem can be solved in a similar way

X-Subproblem Complexity

$$\underbrace{\mathcal{O}(r^2 m)}$$

#bits #users

$$\mathcal{O}(\overset{\text{\#bits}}{r^2}\,\overset{\text{\#users}}{m})$$

Linear to data size!

Summary

- Recommendation is ***search***
- We can accelerate search by ***hashing***
- Unlike previous erroneous ***two-stage*** hashing, ***DCF*** is an ***end-to-end*** hashing method
- Fast ***O(n) discrete optimization*** for DCF

Evaluations

- Dataset (filtering threshold at 10):

Table 1: Statistics of datasets in evaluation.

Dataset	Rating#	User#	Item#	Density
Yelp	696,865	25,677	25,815	0.11%
Amazon	5,057,936	146,469	189,474	0.02%
Netflix	100,480,507	480,189	17,770	1.18%

- Random split: 50% training and 50% testing.
- Metric: NDCG@K
- Search Protocol:

Hamming ranking or hash table lookup

Evaluation 1: Compared to state-of-the-art

- **MF:** ***M***atrix ***F***actorization [Koren et al 2009]

  Classific MF, Euclidean space baseline

- **BCCF:** ***B***inary ***C***ode learning for ***C***ollaborative ***F***iltering [Zhou&Zha, KDD 2012]

  MF+balance+binarization

- **PPH:** ***P***reference ***P***reserving ***H***ashing [Zhang et al. SIGIR 2014]

  Cosine MF + norm&phase binarization

- **CH:** ***C***ollaborative ***H***ashing [Liu et al. CVPR 2014]

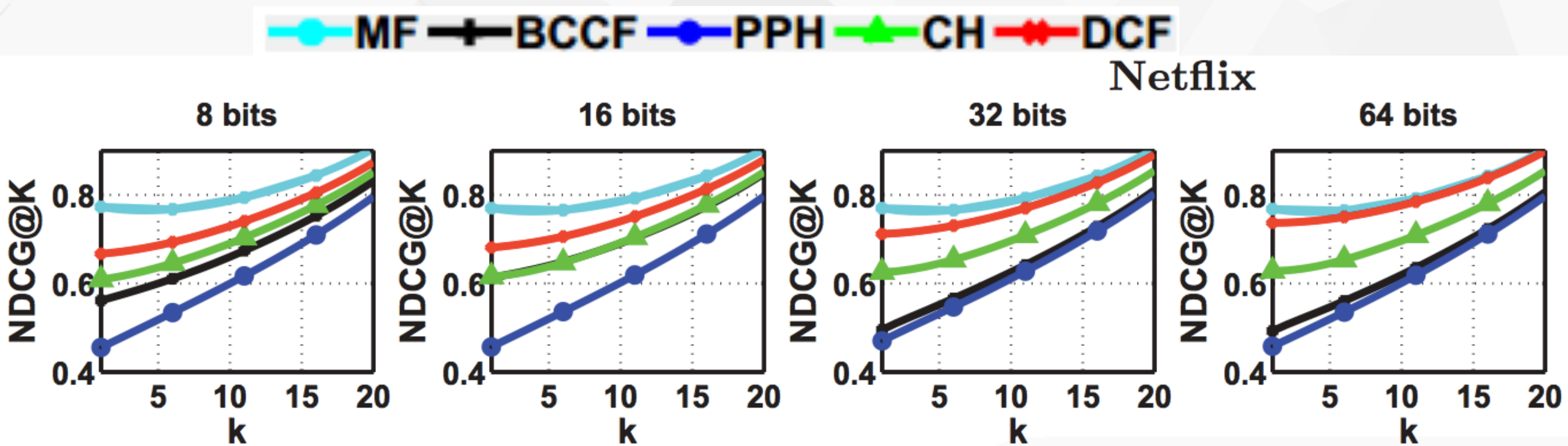
  Full SVD MF + balance + binarization

Evaluation 1
DCF is a new state-of-the-art

Performance of NDCG@10.

Protocol	Hamming Ranking		
Dataset	Yelp	Amazon	Netflix
BCCF(128)	0.623	0.827	0.633
PPH(128)	0.643	0.841	0.587
CH(128)	0.655	**0.922***	0.693
DCF(8)	**0.684***	0.890	**0.730***

1. DCF learns compact and informative codes.
2. DCF's performance is most close to the real-valued MF.
3. End-to-end > Two stage



Evaluation 2
DCF generalizes well to unseen users

Training: full histories of 50% users
Testing: the other 50% users that have no histories in training
Evaluation: simulate online learning scenario.

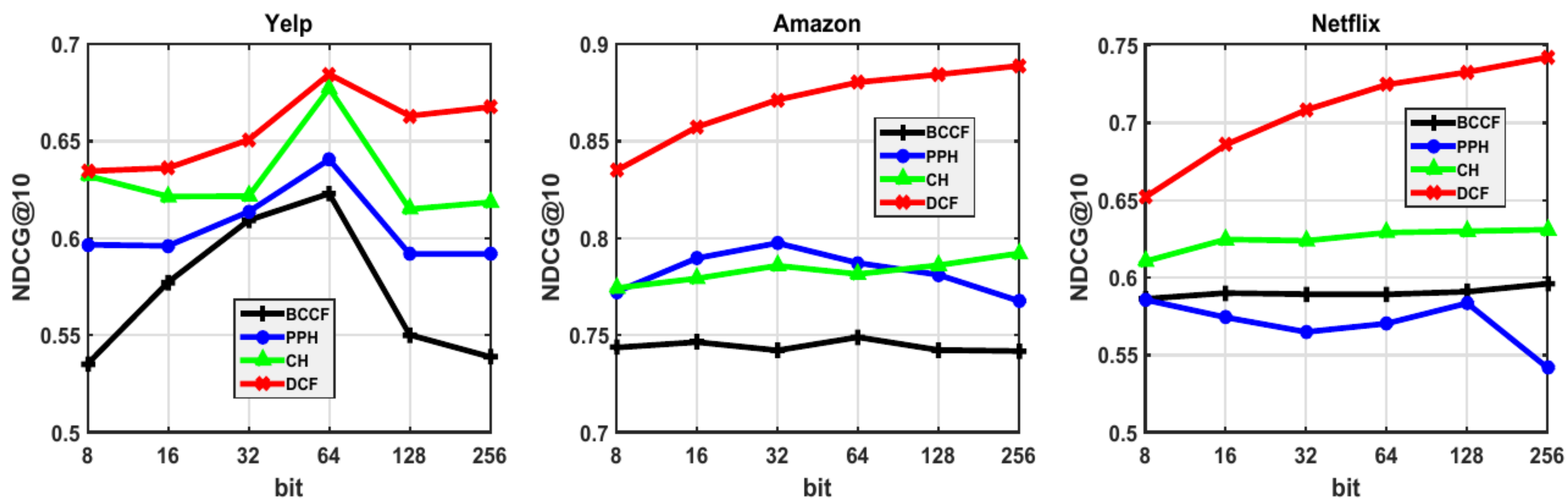


Figure 4: Recommendation performance (NDCG@10) on 50% held-out "new" users (RQ 2).

Evaluation 3
Balanced and Decorrelated Constraints are necessary

MF: original MF
MFB: MF+Binarization
DCFinit: the variant of DCF that discards the two constraints.

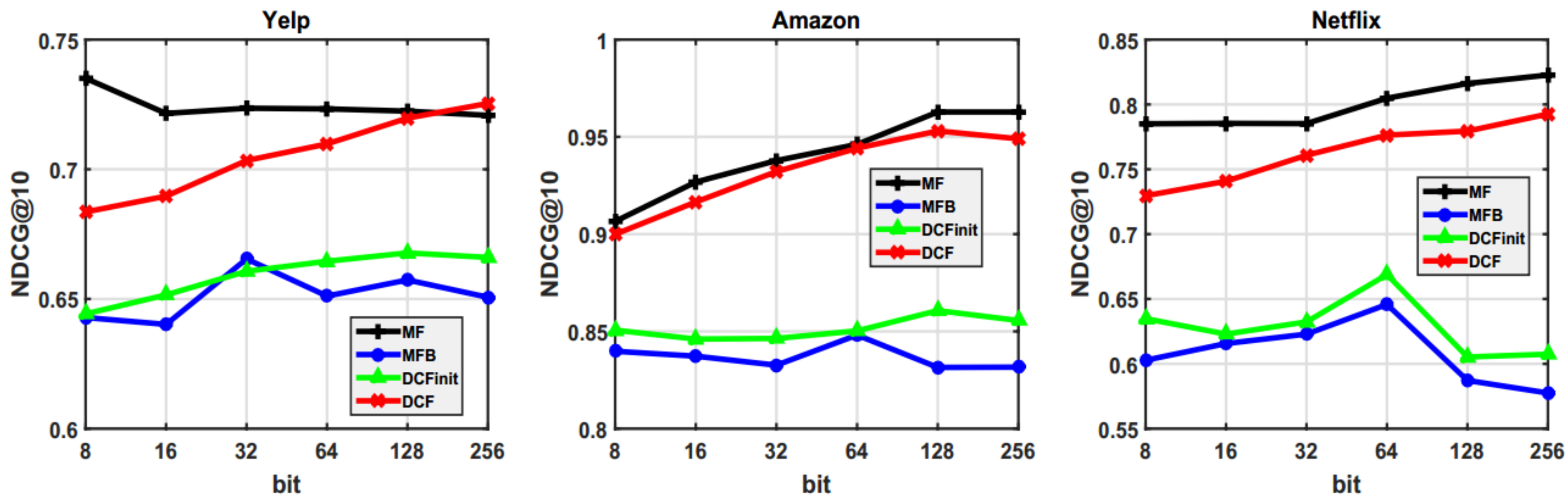


Figure 5: Recommendation performance (NDCG@10) of CF and DCF variants (RQ 3).

Conclusions

- ***D***iscrete ***C***ollaborative ***F***iltering: an end-to-end hashing method for efficient CF
- A fast algorithm for DCF
- DCF is a general framework. It can be extended to any popular CF variants, such as SVD++ and factorization machines.

Thank you

Code available: https://github.com/hanwangzhang/Discrete-Collaborative-Filtering