Alleviating the Sparsity in Collaborative Filtering using Crowdsourcing

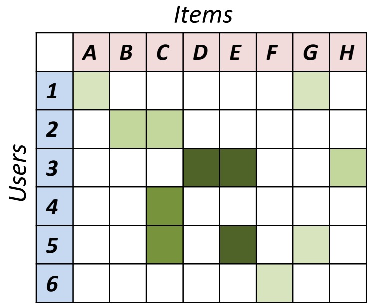
Oct. 12, 2013

Jongwuk Lee[1], Myungha Jang[1], Dongwon Lee[1], Won-Seok Hwang[2], Jiwon Hong[2], Sang-Wook Kim[2]

The Pennsylvania State University[1], USA
Hanyang University[2], Korea

Motivation (1/3)

- **Sparsity of benchmark datasets**
  - MovieLens and Netflix: **95.7%** and **98.8 %** of ratings are missing
- **Existing solutions: reduce sparsity**
  - Impute missing ratings
  - Exploit external information
    - User demographics
    - Trust relationships in social networks
  - Active learning
    - Acquire more ratings for a target user



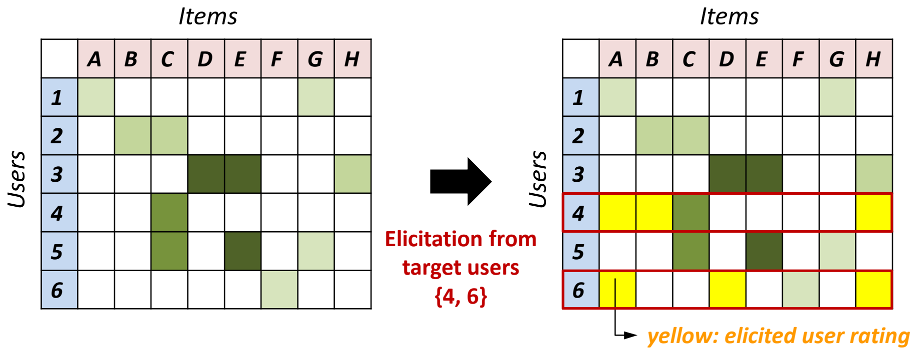
*The higher the rating is, the darker the color is. (**green: observed**)*

- **Our proposal**
  - Append ***more*** ratings using ***crowdsourcing***

2

Motivation (2/3)

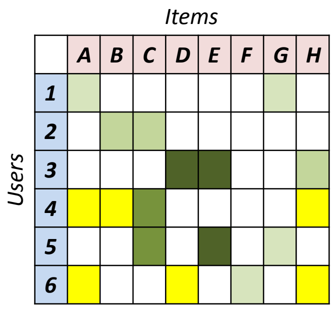
- How does *active learning* work for collaborative filtering?



- Challenging issues
  - **Not** all ratings are **equally** useful
  - To **minimize** user **efforts**, only some of them should be requested and acquired

Motivation (3/3)

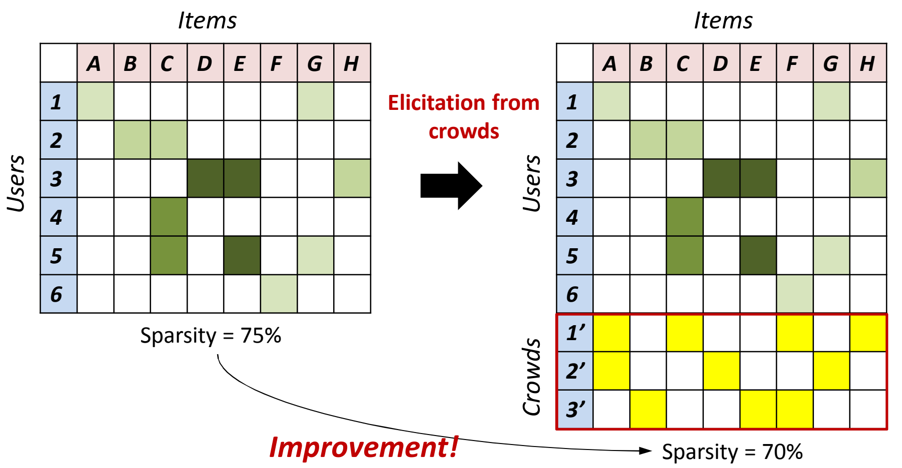
- Differences between active learning and crowdsourcing
  - **Quality of ratings**
    - Accurate vs. Noisy
  - **Human scalability**
    - Single (or a few) users vs. Massive crowds
  - **System environments**
    - Interactive vs. Batch (or one-shot)



The active learning setting is inappropriate to crowds

Our Problem (1/2)

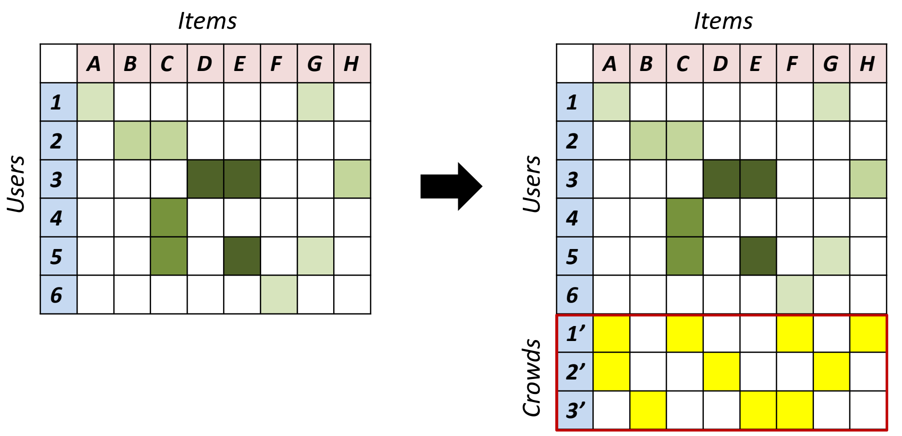
- **How does *crowdsourcing* work for collaborative filtering?**
  - **Crowds** are regarded as *new users* instead of simulated target users



5

Our Problem (2/2)

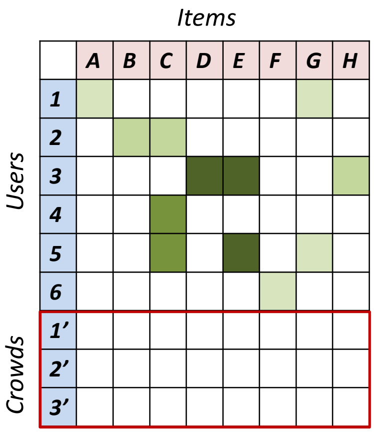
- **Challenging issues**
  - How to **select items** to be shown to crowds?
  - How to decide the **minimum quantity** asked to crowds?
  - How to **handle noisy ratings** by crowds?



6

Challenge 1: Item Selection Strategies (1/7)

- **Strategy: show *s* items out of *n* items to crowds**
  - In the system side, select ***s*** items to improve the accuracy of collaborative filtering



Which items should be shown to crowds?

Challenge 1: Item Selection Strategies (2/7)

- **Two key criteria for designing item selection strategies**
  - **Popularity**: the number of collected (existing) ratings
  - **Usefulness**: more informative ratings

- **Comparisons with 7 strategies**
  - Random, Popularity [1], Highest rating [3], Entropy [1]
  - Highest rating0, Entropy0 [2]
  - Harmonic mean of entropy and logarithm of frequency (HELF) [2]

[1] A. M. Rashid et al., Getting to know you: learning new user preferences in recommender systems. In IUI, 127-134, 2002

[2] A. M. Rashid, G. Karypis, and J. Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. SIGKDD Explorations, 10(2):90-100, 2008.

[3] M. Elahi, V. Repsys, and F. Ricci. Rating elicitation strategies for collaborative filtering. In EC-Web, pages 160-171, 2011.

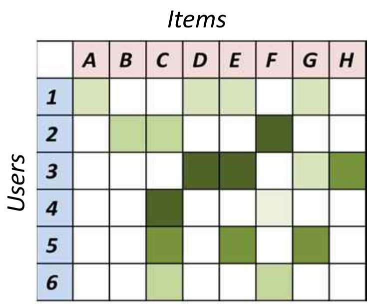
Challenge 1: Item Selection Strategies (3/7)

PENNSTATE

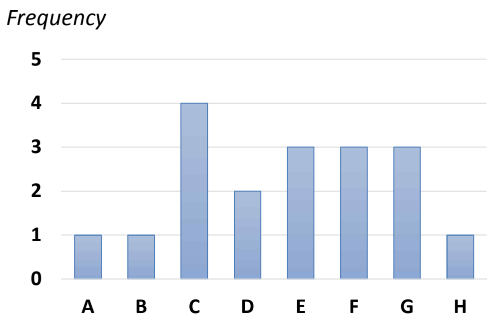
☐ Popularity-based strategy

- Sort items in the decreasing order of the **frequency** in the **observed** matrix
- Choose the top *s* items

$$freq\,(e_j) = \sum_{i=1}^{m} y_{ij}$$



Observed matrix ***O***



Decreasing order of popularity (***s*** = 4):
{<u>*C*, *E*, *F*, *G*</u>, ***D***, ***A***, ***B***, ***H***}

9

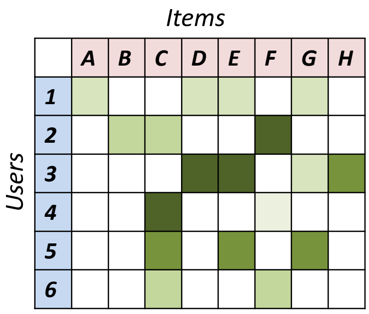
Challenge 1: Item Selection Strategies (4/7)

PENNSTATE

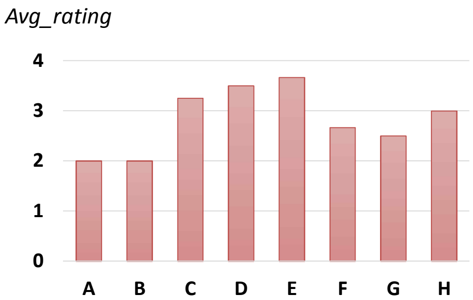
☐ Highest-rating-based strategy

- Sort items in the decreasing order of the **average ratings** in the observed matrix
- Choose the top *s* items

$$Avg\_rating\,(e_j) = \frac{\sum_{i=1}^{m} e_{ij}}{\sum_{i=1}^{m} y_{ij}}$$



Observed matrix ***O***



Decreasing order of avg. ratings (***s*** = 4):
{<u>*E*, *D*, *C*, *H*</u>, ***F***, ***G***, ***A***, ***B***}

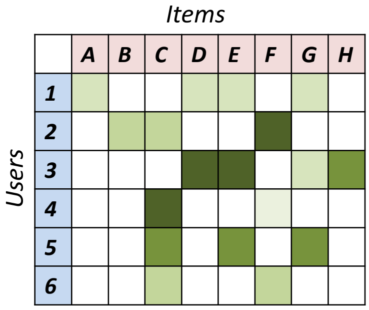
10

Challenge 1: Item Selection Strategies (5/7)

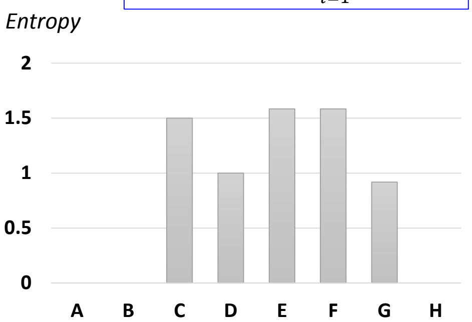
Entropy-based strategy

- Sort items in the decreasing order of the **entropy** in the observed matrix
- Choose the top *s* items

$$Entropy\ (e_j) = -\sum_{i=1}^{5} p_{ij} \log_2 p_{ij}$$



Observed matrix ***O***



Decreasing order of entropy (***s*** = 4):
{***E***, ***F***, ***C***, ***D***, ***G***, ***A***, ***B***, ***H***}

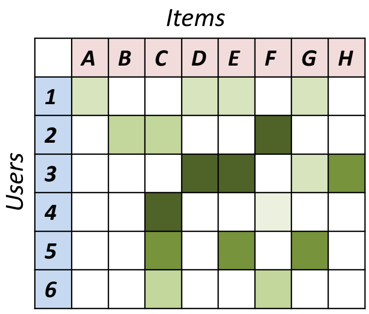
11

Challenge 1: Item Selection Strategies (6/7)

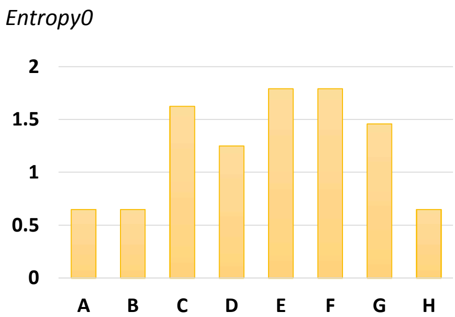
Entropy0 strategy

- Missing ratings are considered as **zero scores**, and compute the entropy
- Choose the top *s* items

$$Entropy0\ (e_j) = -\sum_{i=0}^{5} p_{ij} \log_2 p_{ij}$$



Observed matrix ***O***



Decreasing order of entropy0 (***s*** = 4):
{***E***, ***F***, ***C***, ***D***, ***G***, ***A***, ***B***, ***H***}

12

Challenge 1: Item Selection Strategies (7/7)

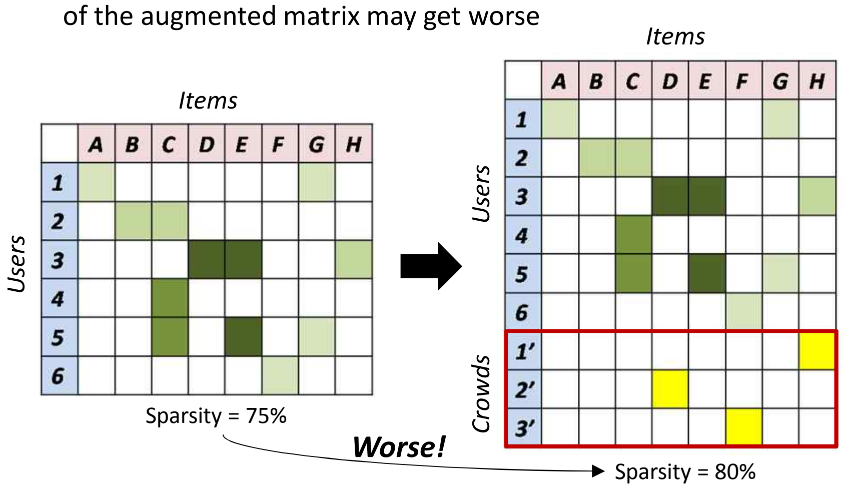
- **Highest-rating0 strategy**
  - When computing the average, missing ratings are considered as zero scores

- **Harmonic mean of entropy and logarithm of frequency (HELF)**
  - Combine the entropy and popularity using a harmonic mean

$$HELF(e_j) = 2 \times \frac{\log_2 freq(e_j) \times entropy(e_j)}{\log_2 freq(e_j) + entropy(e_j)}$$

Challenge 2: Minimum Quantity for Crowds (1/3)

- **Problem for crowd elicitation**
  - If the number of ratings in a crowd matrix is **too small**, the sparsity of the augmented matrix may get worse



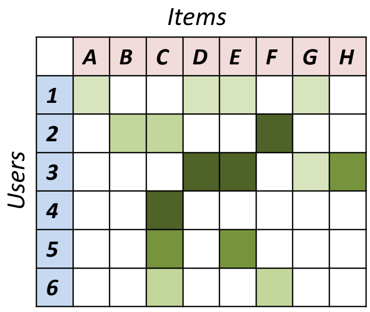
Challenge 2: Minimum Quantity for Crowds (2/3)

- **Solution**
  - Exploit ***the frequency of item ratings per user*** in observed matrix O

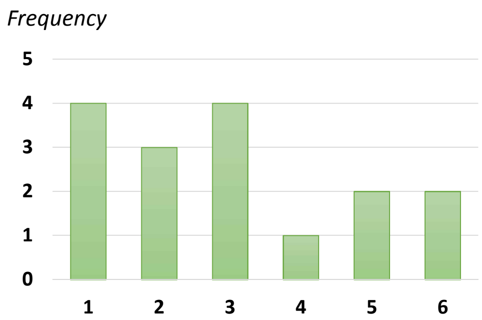
- **Three heuristic solutions**
  - **Minimum frequency**
    - Reduces the burden for crowds, but may increase the sparsity
  - **Average frequency**
    - Reduces the sparsity for the augmented matrix
  - **Median frequency**
    - May reduce the sparsity for the augmented matrix

Challenge 2: Minimum Quantity for Crowds (3/3)

- **Three heuristic solutions**
  - Average vs. Minimum vs. Median



Observed matrix ***O***



Average frequency = 3

Minimum frequency = 1

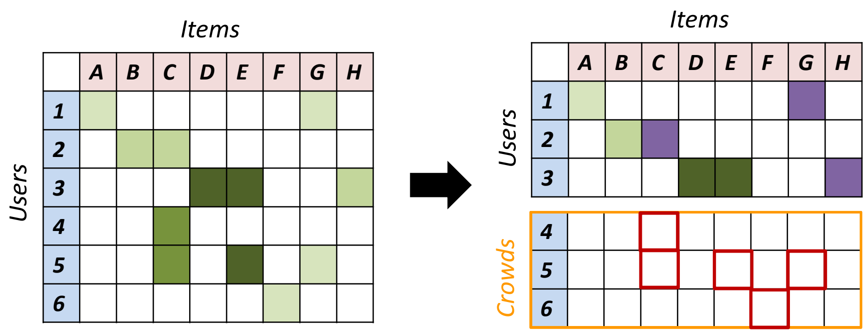
Median frequency = 2

Challenge 3: Filtering Noisy Ratings

- **Problem**
  - The collected ratings from crowds can be **erroneous**
- **Solutions**
  - Task-independent features
    - Exploit ill-qualified workers using AMT features
    - E.g., Geo-location, **HIT approval rate**, and category of workers
  - Task-dependent features
    - **Fake movie filtering**: coming-soon movies to be released in 2014 and fan-made non-existent movies
    - **Time spent for a HIT**
    - **Correlation** between ratings by a worker and the average ratings
- **In our real-life evaluation, both task-independent and dependent features were exploited**

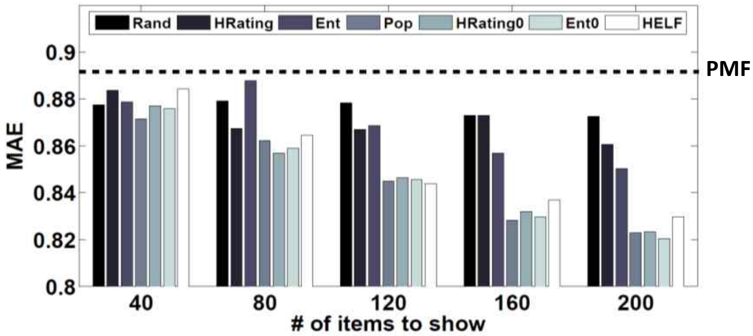
Simulated Experimental Setting

- **Simulated evaluation**
  - Split a dataset into three partitions: **crowdsourced**, **observed**, and **validated**
  - MovieLens: 943 users and 1,682 movies with 100,000 ratings
    - 100-400 users as crowd workers
    - At least 70 ratings per worker



Experimental Results (1/2)

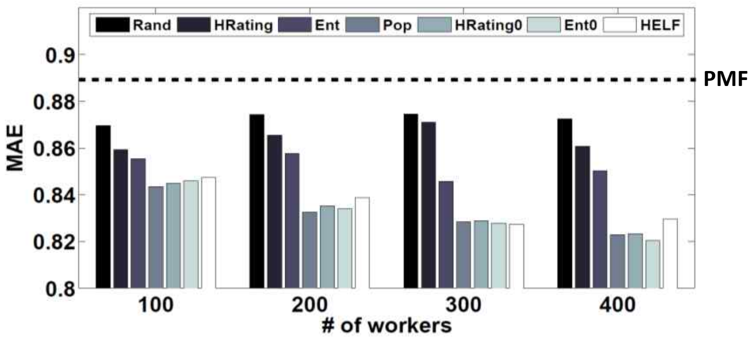
- Impact of the **number of items to be shown**
  - A dotted line represents a collaborative filtering algorithm without the crowd matrix
    - Use probabilistic matrix factorization (PMF) as a baseline



Ent0 and **Pop** are the overall winners.

Experimental Results (2/2)

- Impact of the **number of workers**
  - A dotted line represents a collaborative filtering algorithm without the crowd matrix
    - Use probabilistic matrix factorization (PMF) as a baseline



Real-life Experimental Setting

☐ Real-life evaluation

Parameters	Value
Strategies	HRating, Ent, Pop, HELF
# of items to be shown	204 (real: 180, fake: 24)
Min # of requested items	20
# of workers per strategy	100
Reward per worker	$0.7

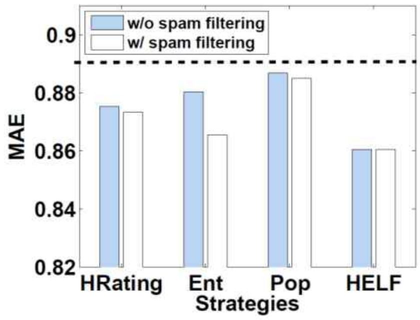
☐ Spam filtering

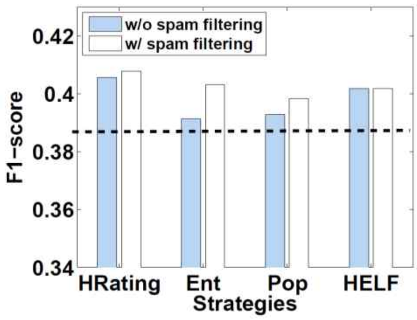
- The approval rate of workers ≥ 90%
- They are rejected if
  - ☐ # of clicked fake movies > 3
  - ☐ The average work time per clicked movie < 10 seconds

Snapshot of our HIT design

Real-life Experimental Results

- Real-life evaluation
  - Crowd-enabled ratings yield better accuracy than the baseline without crowdsourcing
  - Overall, HELF shows the best accuracy among 4 strategies





Conclusions

- **Combining collaborative filtering with crowdsourcing**
- **Addressing three key challenges**
  - Item selection strategies
  - Minimum quantity
  - Spam filtering
- **Evaluating simulated and real-life evaluation on AMT**

Thank you!
Questions?