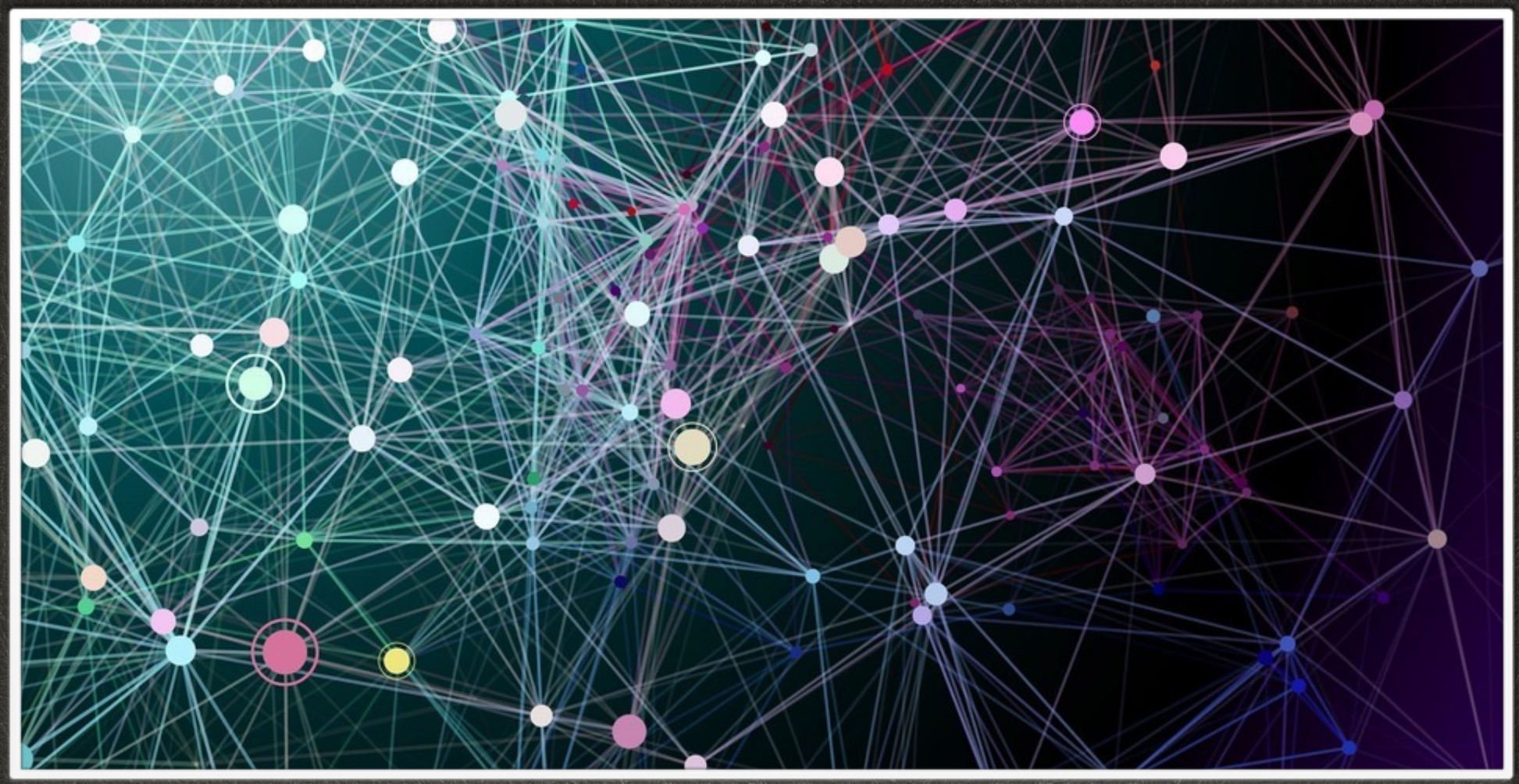


DATA CRACKERS

Guangyu Lin

Background

- Social Network Services (SNS) are becoming increasingly popular in recent years, resulting in a huge amount of information that is growing exponentially. The popularity appeals to the need of accurate recommendation system for SNS users. I will present a recommendation system with time-aware latent factor model for Tencent Weibo, one of the most popular Chinese social networking websites.

- The recommendation system is built using the data set provided by KDD Cup 2012 compared with other recommendation systems proposed by other KDD Cup competitors in accuracy and efficiency.

OVERVIEW

Introduction
Training Model
Data Preprocessing
Training
Recommendation
Evaluation

Introduction

- **Survey of related works**

  KDD Cup 2012 winners and competitors:

  ACMClass@SJTU, factorisation model combined with additive forest

  Shanda Innovations, context-aware ensemble of Multifaceted Factorisation Models

- Data Pre-processing
- Training Model
- Prediction
- Evaluation

Introduction

- Survey of related works
- **Data Pre-processing**

  Active Users & Inactive Users
- Training Model
- Prediction
- Evaluation

Filename	Format & File Size
rec_log_train	.txt (1.99Gb)
user_profile	.txt (55.8Mb)
item	.txt (1.18Mb)
user_action	.txt (217Mb)
user_sns	.txt (740Mb)
user_key_word	.txt (182Mb)

Introduction

- Survey of related works
- Data Pre-processing
- **Training Model**
  - **R_{ij}: Based on the training data for tuples(i, j)**
  - **Latent Factor Model**
  - **$R_{m*n} = P_{m*k} * Q_{k*n}$**
  - **Minimize P * Q: Gradient Descent**
- Prediction
- Evaluation

Introduction

- Survey of related works
- Data Pre-processing
- Training Model
- **Prediction**
  - **Never recommend noisy items**
  - **Observe the attributes of the active users who have actions with items**
- Evaluation

Introduction

- Survey of related works
- Data Pre-processing
- Training Model
- Prediction
- **Evaluation**
  - **a metric**
  - **a validation dataset**

Training Model

- Latent Factor Models
  - feature-biased matrix factorisation
  - Solve Top-N recommendation problems
- Model Derivation Process

  Preliminary Model —> Keyword Model —> Social Network Model —> Group Based Mode —> Combined Model

$$\widehat{r_{ui}} = b_i^I + b_u^U + {p_u}^T q_i$$

b_u^U: user based adjustment
b_i^I: item based adjustment
p_u^k : latent factors for all users
q_i : latent factors for items

$$p_u = \sum_{k \in K(u)} w_{uk}^K p_k^K$$

$$\widehat{r_{ui}} = b_i^I + b_u^U + \left(\sum_{k \in K(u)} w_{uk}^K p_k^K \right) q_i$$

- w_{uk}^k: the weight of user u to keyword k
- p_k^k: latent factors for keywords

$$p_u = \sum_{j \in S(u)} w_{uj}^S p_j^S$$

$$\widehat{r_{ui}} = b_i^I + b_u^U + \left(\sum_{j \in S(u)} w_{uj}^S p_j^S \right) q_i$$

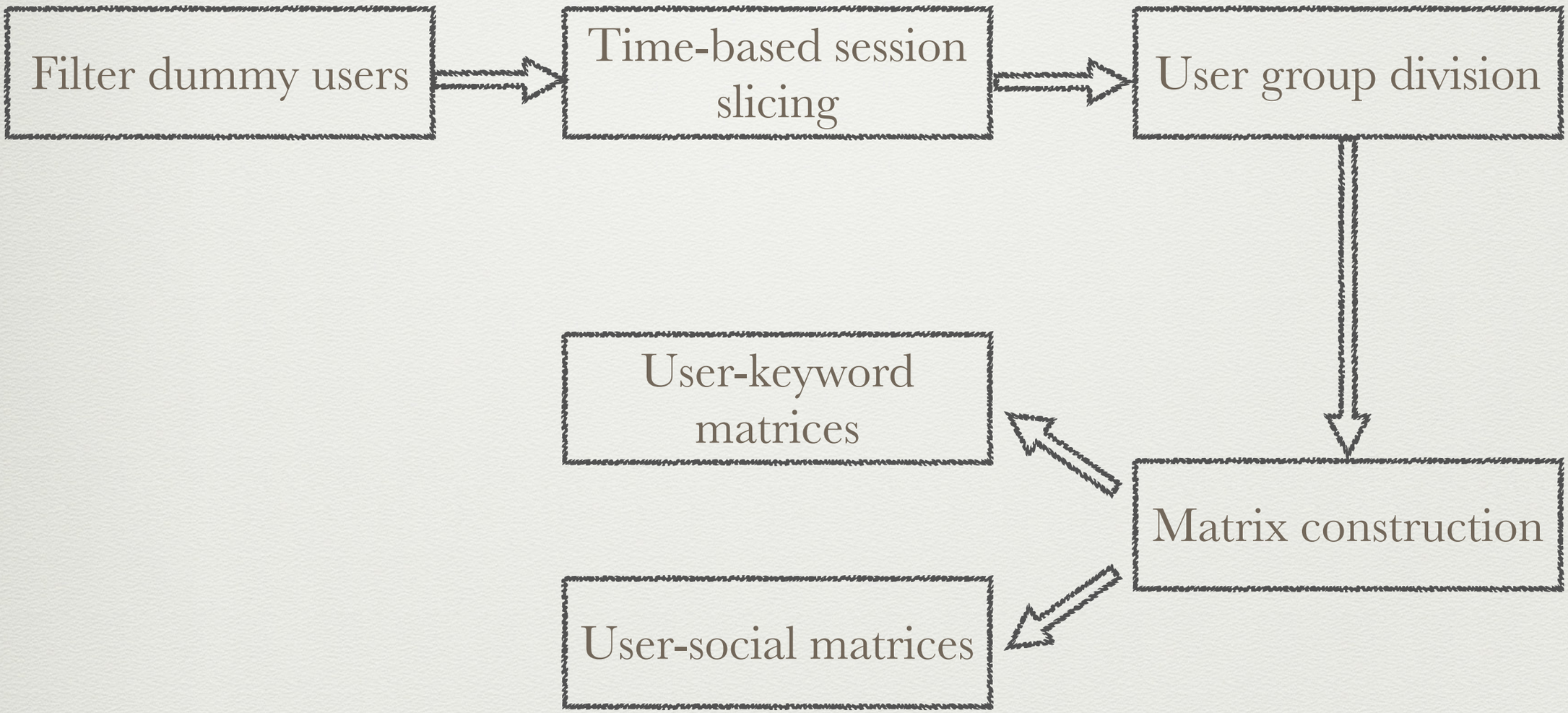
- $\mathbf{w_{uj}^s}$: the weight of interests from user u to user j
- $\mathbf{p_j^k}$ **:** latent factors for all users

- Divide users into different groups: Gender, Age
- Train each model separately

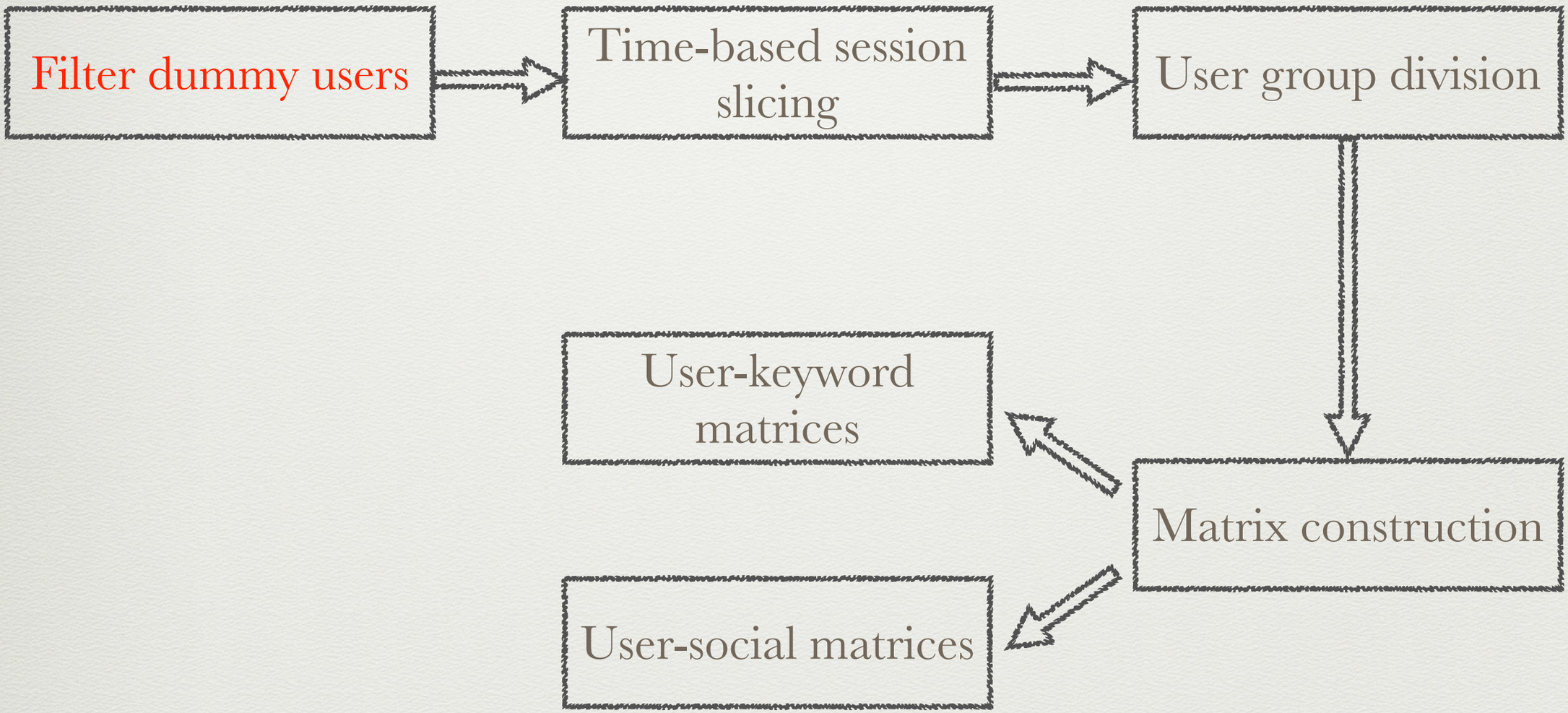
- Incorporate keywords
- Social influence
- Group adjustment

$$\widehat{r_{ui}} = b_i^I + b_u^U + p_u q_i = b_i^I + b_u^U + \left(\alpha \sum\nolimits_{k \in K(u)} w_{uk}^K p_k^K + \beta \sum\nolimits_{j \in S(u)} w_{uj}^S p_j^S\right) q_i$$

Data Processing

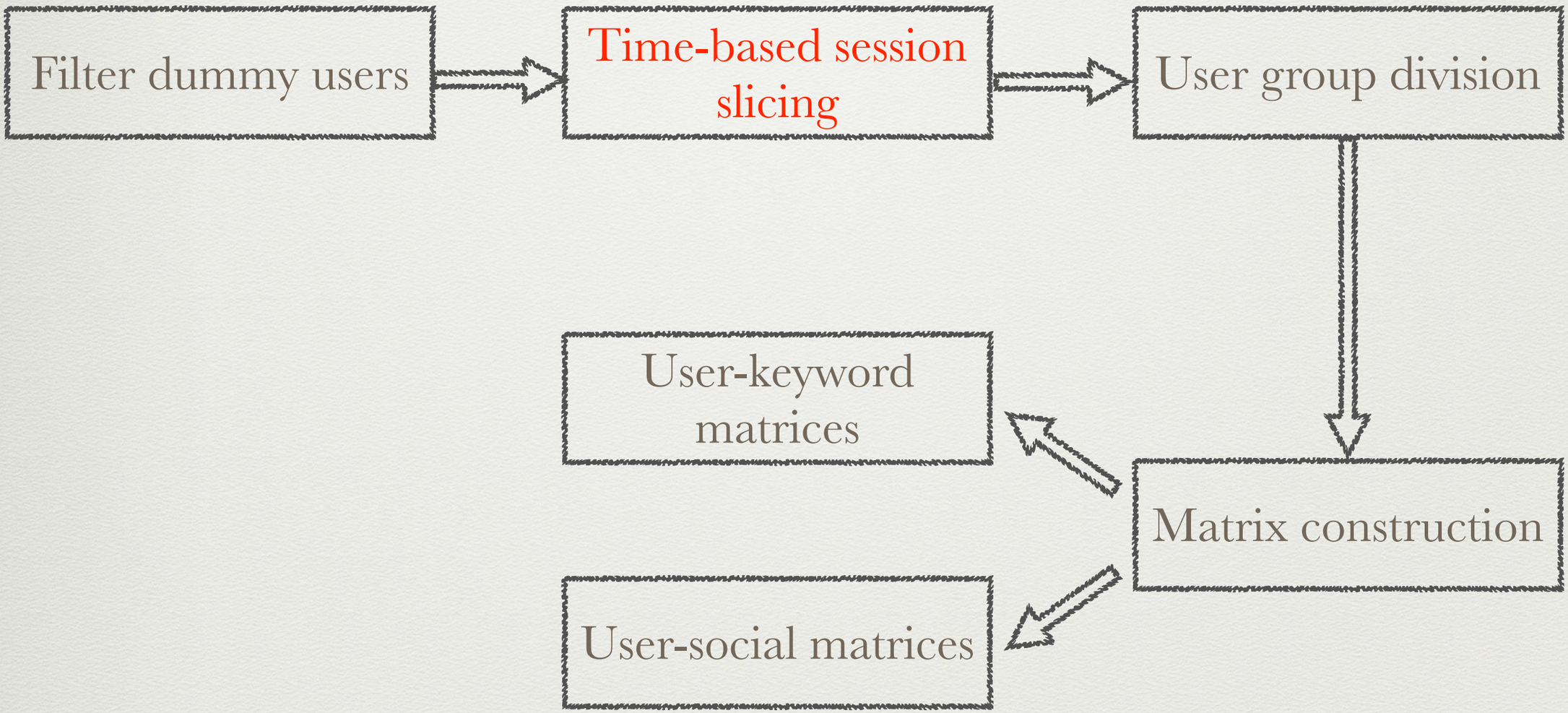


Data Processing



rec_log_train: 73,209,277—>63,551,881

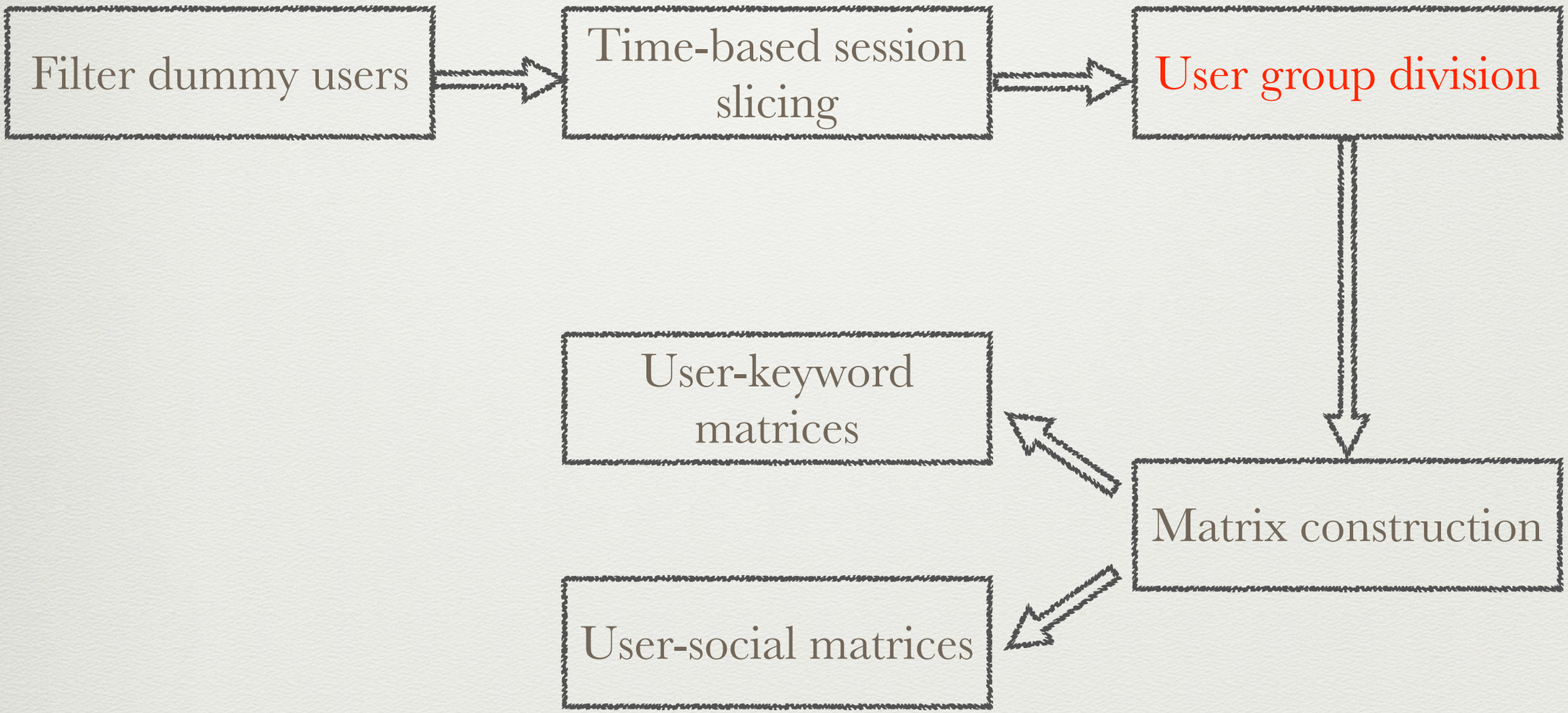
Data Processing



recommendation records: 63,551,881 —> 9,933,395

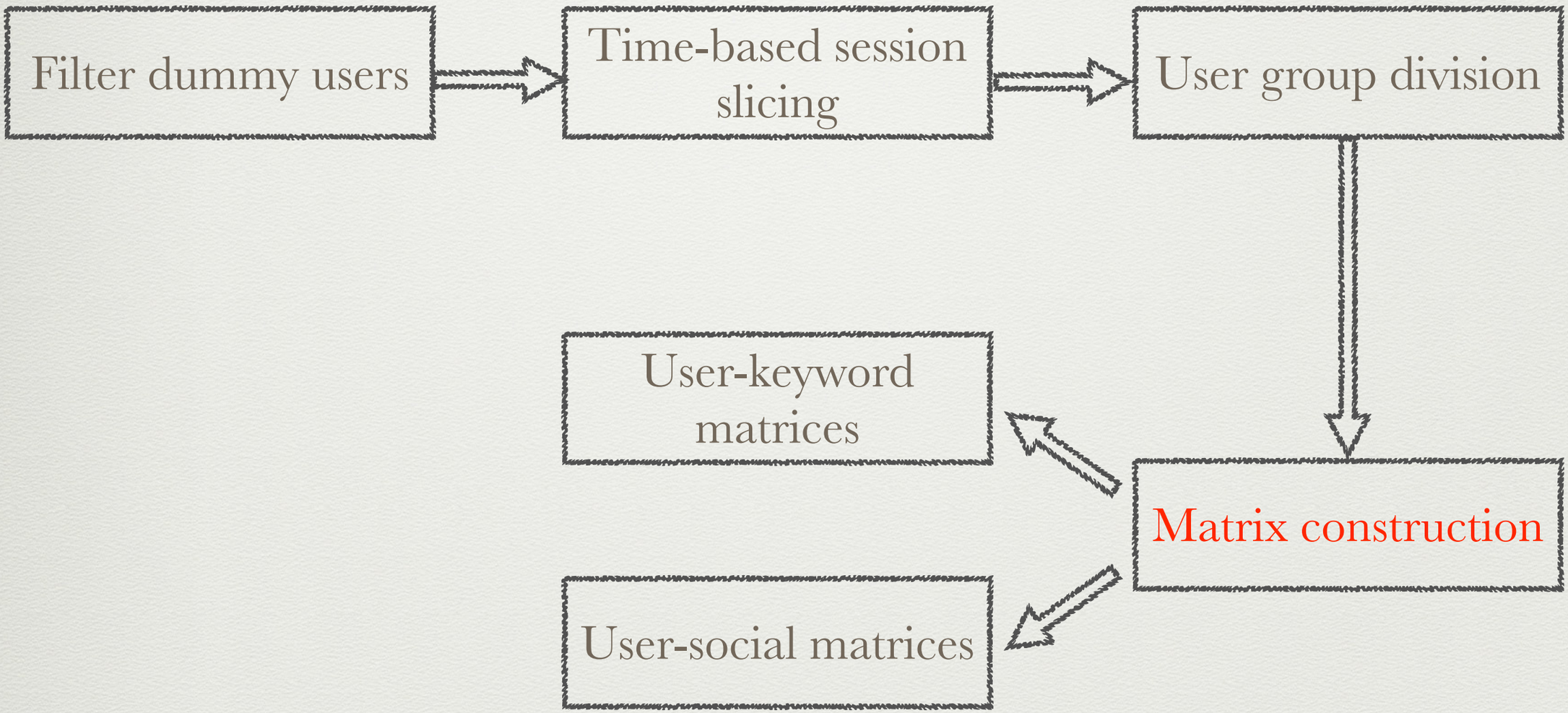
Negative V.S. Positive : 13:1 —>3:2

Data Processing



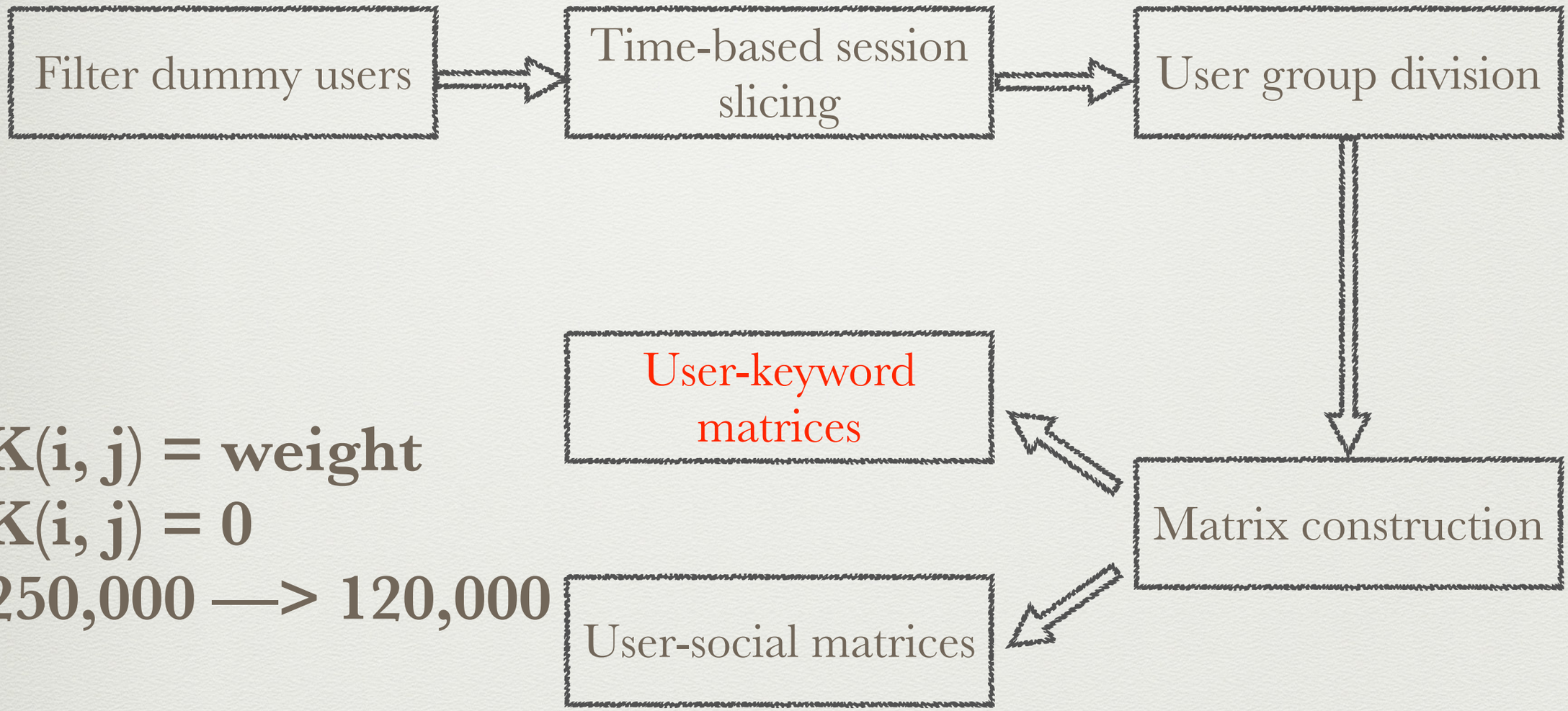
11groups: 0-15, 15-20, 20-25, 25-32, and above 32 two genders and no information

Data Processing



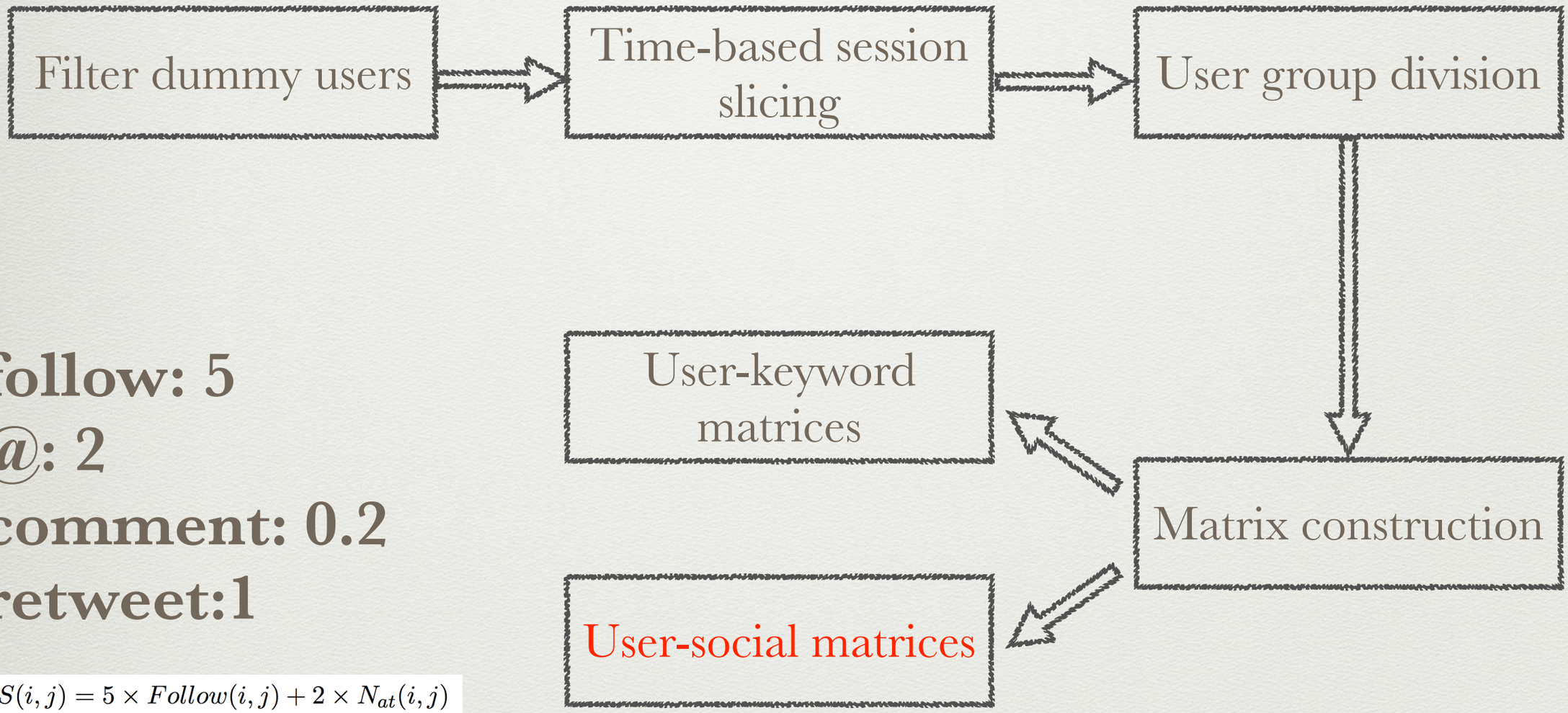
11 user-keyword matrixes and 11 user-user matrixes

Data Processing



K(i, j) = weight
K(i, j) = 0
250,000 —> 120,000

Data Processing



follow: 5
@: 2
comment: 0.2
retweet:1

$$S(i,j) = 5 \times Follow(i,j) + 2 \times N_{at}(i,j)$$
$$+0.2 \times N_{comment}(i,j) + 1 \times N_{retweet}(i,j)$$

Training

- Cost Function

$$C = \sum_{(u,i)\in T} (\widehat{r_{ui}} - r_{ui})^2 + \lambda \sum_{k\in K} \|p_k^K\|^2 + \lambda \sum_{j\in S} \|p_j^S\|^2 + \lambda \sum_i \|q_i\|^2 + \lambda \sum_i (b_i^I)^2 + \lambda \sum_u (b_u^U)^2$$

- Latent Factor Training
  - Gradient Descent
  - Training Step

$$p_k^K = p_k^K - \alpha(\sum_{(u,i)\in T \cap u\in U(k)} (\widehat{r_{ui}} - r_{ui}) w_{uk}^K q_i{}^T + \lambda p_k^K)$$

$$p_j^S = p_j^S - \alpha(\sum_{(u,i)\in T \cap u\in U(j)} (\widehat{r_{ui}} - r_{ui}) w_{uj}^S q_i{}^T + \lambda p_j^S)$$

$$q_i = q_i - \alpha(\sum_{(u)\in U} (\widehat{r_{ui}} - r_{ui}) \left(\sum_{k\in K(u)} w_{uk}^K p_k^K + \sum_{j\in S(u)} w_{uj}^S p_j^S\right)^T + \lambda q_i)$$

$$b_i^I = b_i^I - \alpha(\sum_{(u)\in U} (\widehat{r_{ui}} - r_{ui}) + \lambda b_i^I)$$

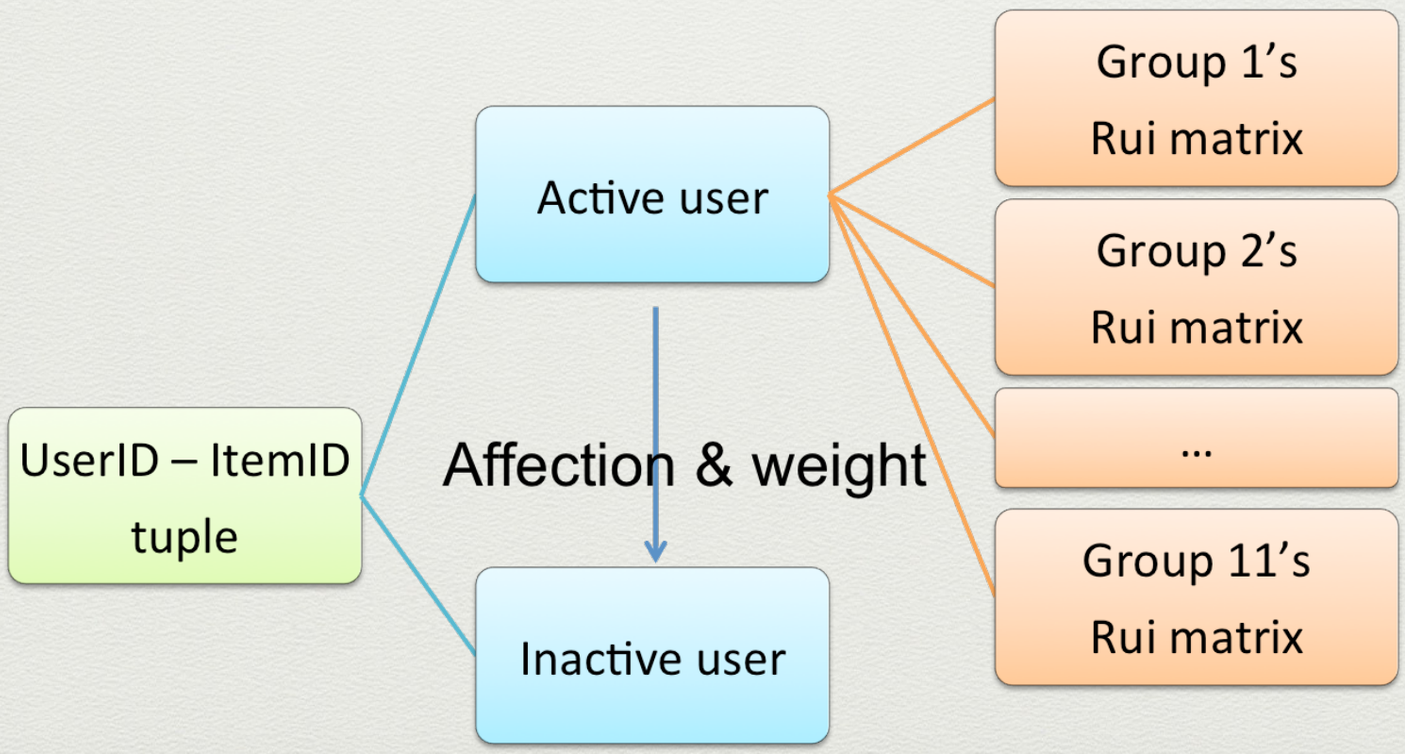
$$b_u^U = b_u^U - \alpha(\sum_{(i)\in I} (\widehat{r_{ui}} - r_{ui}) + \lambda b_u^U)$$

Recommendation

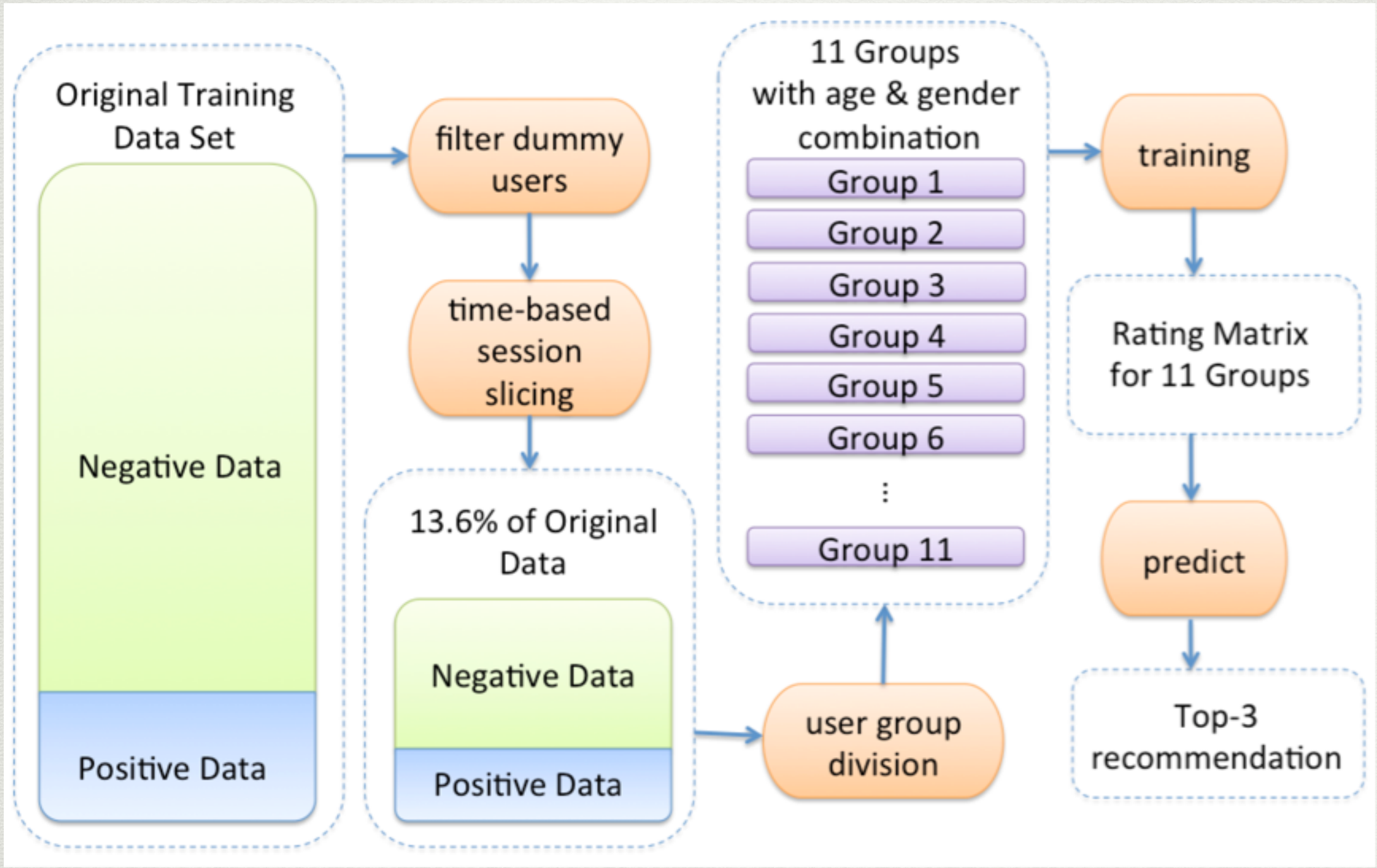
Most 3 items

- Testing Dataset: rec_log_test.txt

- Recommendation Scheme: User Classification



Experiments & Evaluation



Evaluation

- Evaluation metric

Average Precision

$$ap@n = \sum_{k=1...n} P(k)/(\text{number of items clicked in m items})$$

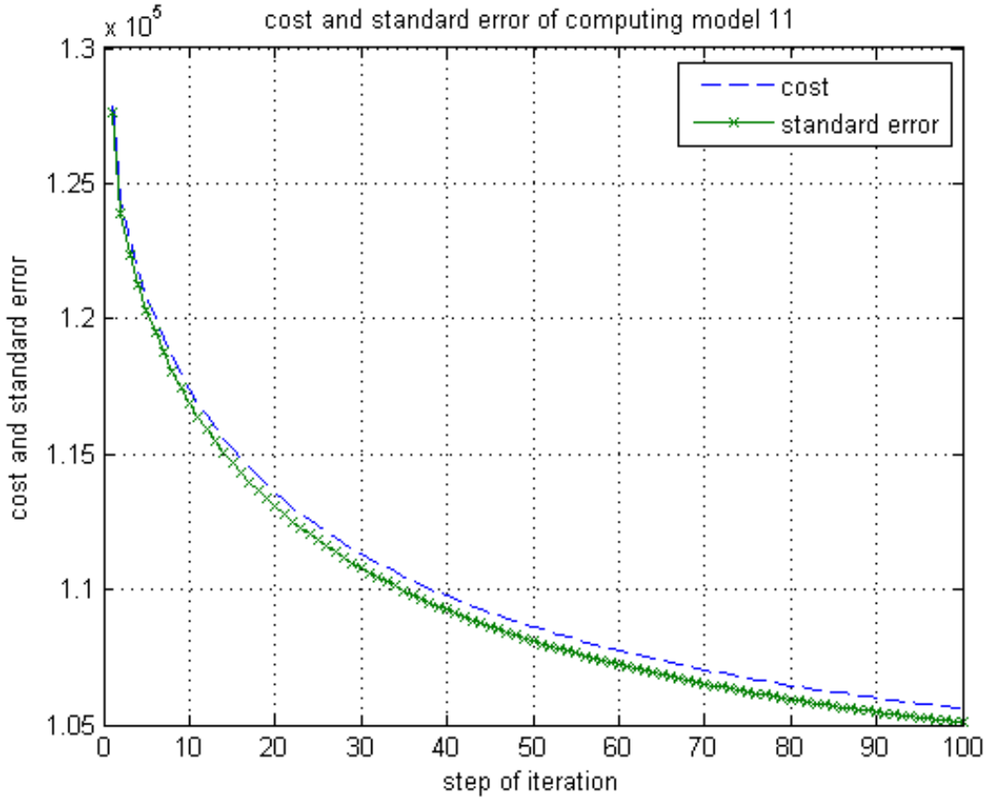
$$AP@n = \sum_{i=1...N} (ap@n)_i/\mathrm{N}$$

For example, if 5 items were recommended to the user, and the user clicked #1,#3 and #4, then

$$ap@3 \equiv \frac{\frac{1}{1} + \frac{2}{3}}{3} \approx 0.56$$

Evaluation

- Parameter Tuning

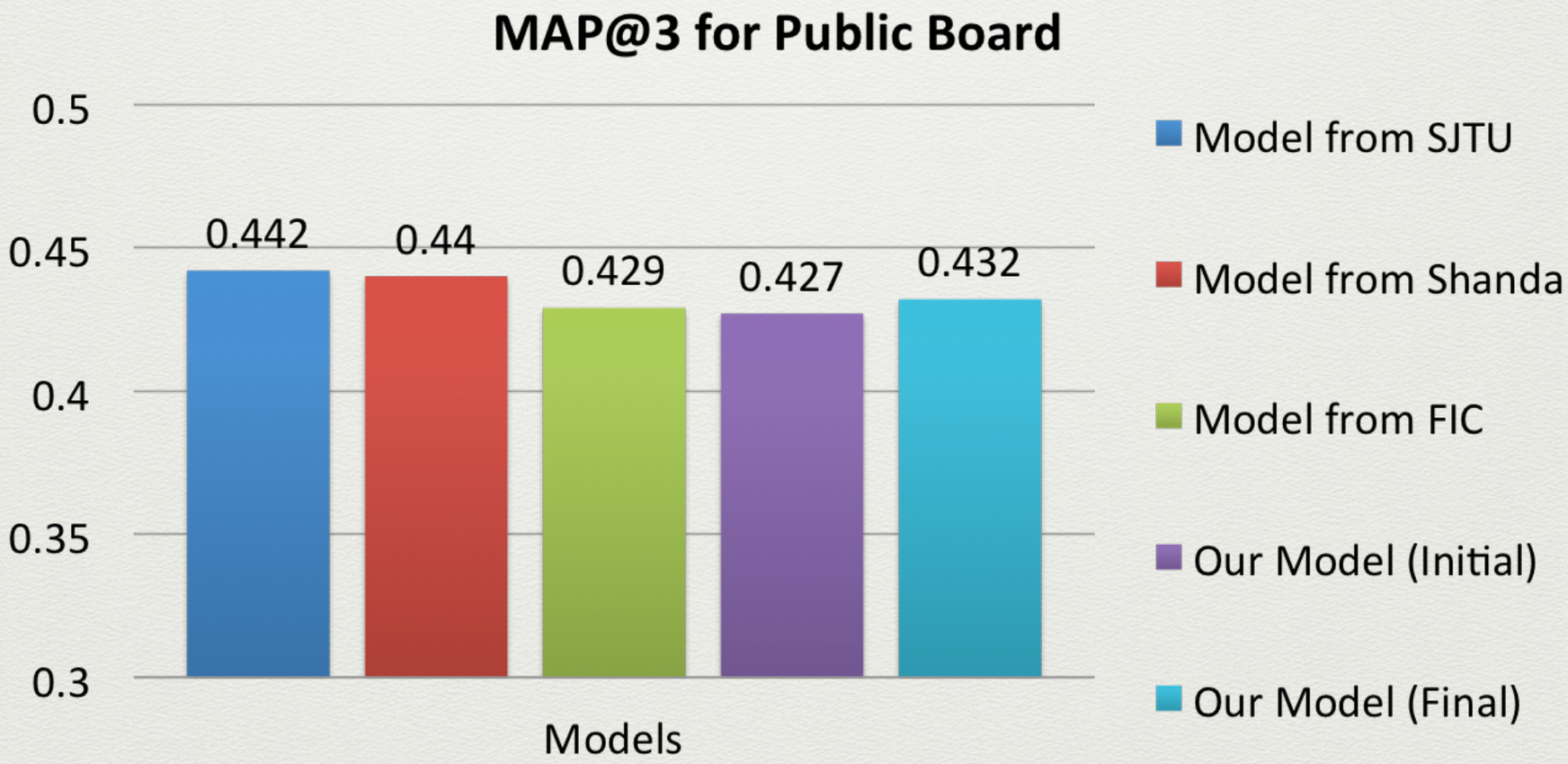


PARAMETER FOR EACH MODEL

Group	α	λ	λ_{bias}	MAP@3
1	0.00004	0.003	0.002	0.431
2	0.00003	0.003	0.001	0.432
3	0.00003	0.003	0.001	0.452
4	0.00004	0.003	0.002	0.446
5	0.00002	0.004	0.001	0.434
6	0.00001	0.004	0.001	0.433
7	0.00003	0.003	0.002	0.391
8	0.00003	0.002	0.002	0.401
9	0.00005	0.003	0.002	0.366
10	0.00005	0.003	0.002	0.390
11	0.00005	0.004	0.002	0.446

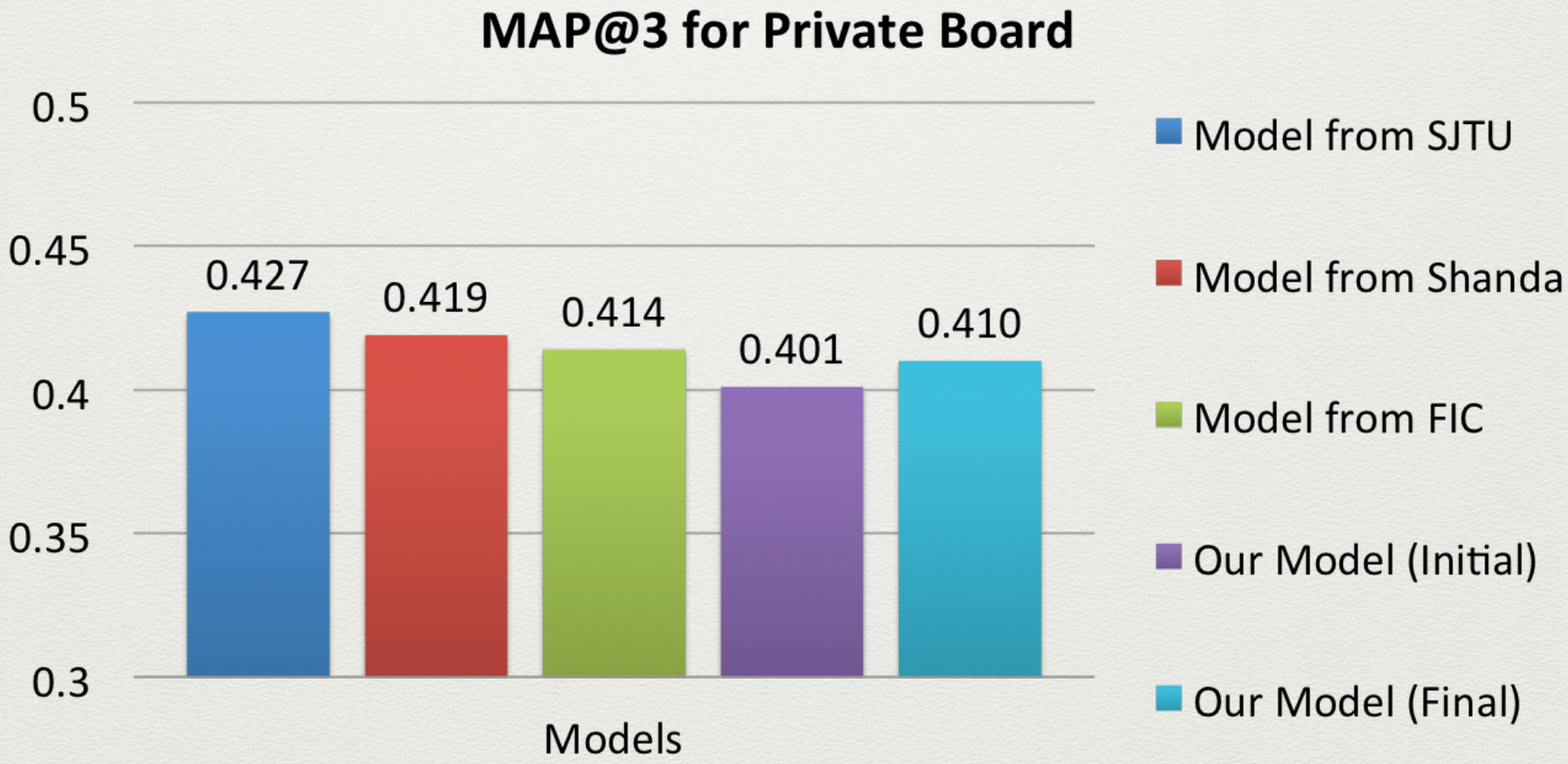
Evaluation

- Recommendation Results



Evaluation

- Recommendation Results



Conclusion

- Study basic form of latent factor model
- Introduce user-keyword latent factors
- Utilise information of users social network
- Cut off noisy data
- Design a time-aware combined model based on latent factor model
- Significantly improve prediction accuracy

REFERENCE

[1] Chen, Tianqi, et al. “Combining factorization model and additive forest for collaborative followee recommendation.” KDD CUP (2012).

[2] Zhao, Xing. “Scorecard with latent factor models for user follow prediction problem.” KDD-Cup Workshop. 2012.

[3] Chen, Yunwen, et al. “Context-aware ensemble of multifaceted factorization models for recommendation prediction in social networks.” KDD-Cup Workshop. 2012.

[4] Bottou, Lon. ”Large-scale machine learning with stochastic gradient descent.”Proceedings of COMPSTAT’2010. Physica-Verlag HD, 2010. 177-186.

[5] http://en.wikipedia.org/wiki/Information_retrieval

https://github.com/gl8429/dataMining

Q & A

Thank you!