



# Intelligent Recommender System —— An Introduction

---

School of Computer Science and Technology  
East China Normal University

Wei Zhang (张伟)

# Brief Course Introduction

---

- Recommender Systems (RS) are key technical components in many online applications, such as product recommendation, video recommendation, or stream recommendation in social networks.
- In this course, we focus on some basic and advanced **RS algorithms** and their **applications**.
- Goals
  - Understand basic concepts and procedures of RS
  - Master basic recommendation algorithms
  - Practice coding

# Contact Information

---

- **Wei Zhang (张伟)**

- Associate Researcher, School of Computer Science and Technology, East China Normal University
- Research interest: Data mining, machine learning application
  - Focusing on applications in **recommender system**
  - Homepage: <https://weizhanglitt.github.io/>
  - Mail: [zhangwei.thu2011@gmail.com](mailto:zhangwei.thu2011@gmail.com)
- Office: Room B903, Science Building
  - Office hours: 10:00-11:30 (Tuesday)

# Prerequisite

---

- **Python**

- Suitable for data science (e.g., processing data)
- Many third-party libraries for scientific computing
  - Deep learning: Tensorflow, PyTorch

- **Artificial Intelligence**

- Some basic concepts in AI
  - E.g., cost function, optimization

- **Basic Mathematics**

- E.g., matrix computation

# Evaluation

---

- **Course Grade (100%)**

- 10% class participation, 45% homework (15% each), 45% final project
- Class participation
- Homework
  - 3 programming assignments, about 2 weeks for each
- Final project
  - Working as a group (2-3 persons)
  - Multiple choices (e.g., data science competition, self-defined task w.r.t. recommendation, paper survey)
  - Note: If you want only one or more than 3 persons in a group, please contact me.

# Programming Assignments

---

- **Assignment 1**

- User-based collaborative filtering (UCF)

- **Assignment 2**

- Content-based recommendation

- **Assignment 3**

- Latent factor model (matrix factorization)

- **Req.:** Each should include **code** (executable), **report**

- More details will be clarified in later courses
- Should be ***executable***

# Course Materials

---

- **大夏学堂**

- Platform for material uploading
- URL: <https://elearning.ecnu.edu.cn/>

- **包括**

- **Slides**

- **Assignments**

- Requirement
- Data

- **Others**

- Including some suggested papers, or books
- Releasing assignment scores
- Course WeChat group

# Reference Books

- 推荐系统：技术、评估及高效算法
  - 机械工业出版社
- 推荐系统实践
  - 项亮，人民邮电出版社
- 推荐系统
  - 人民邮电出版社
- 深度学习推荐系统
  - 人民邮电出版社





# Introduction



*Which digital camera should I buy?*

*What is the best holiday for me and my family?*

*Which is the best investment for supporting the education of my children?*

*Which movie should I rent? Which web sites will I find interesting?*

*Which book should I buy for my next vacation?*

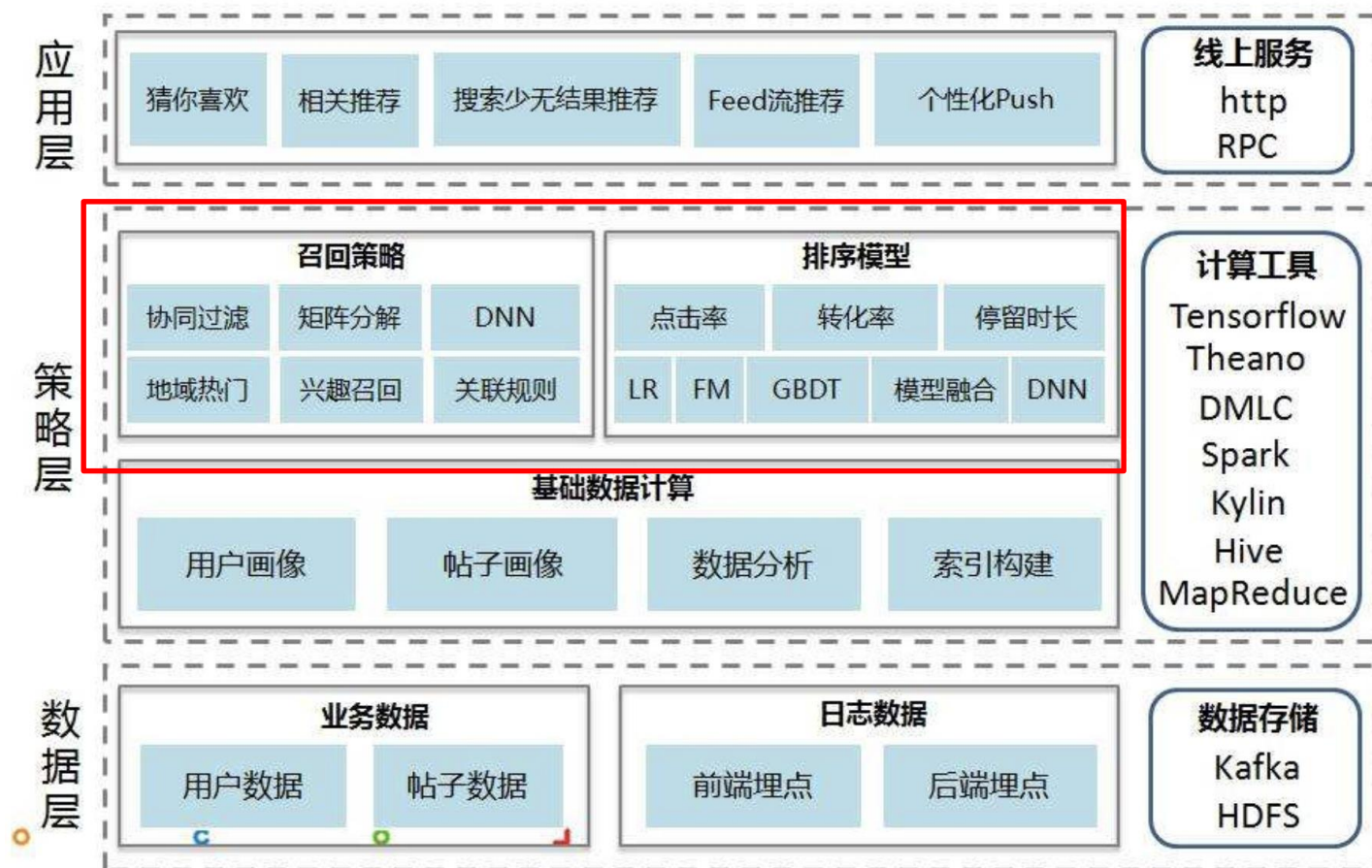
*Which degree and university are the best for my future?*

---

# **What is recommender system?**

# Architecture of Full RS

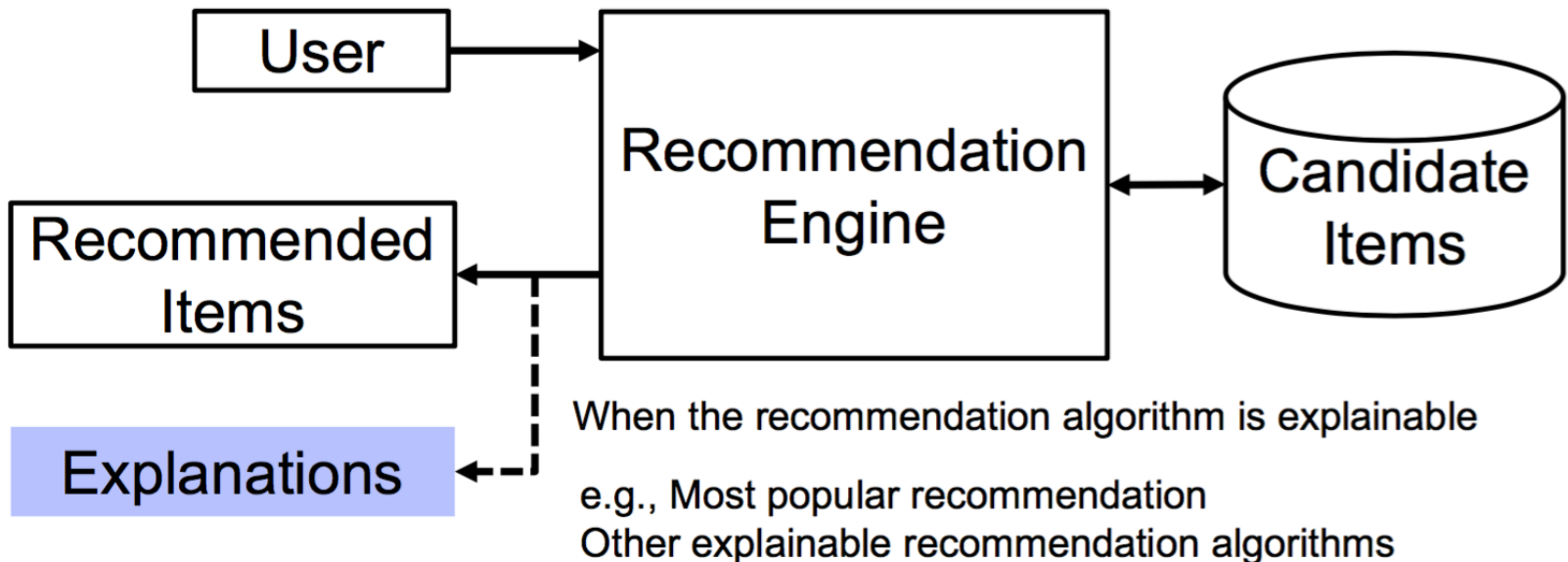
- RS is a system including many parts
  - Storage, computation engines, web service, etc.
  - Many components, very complex



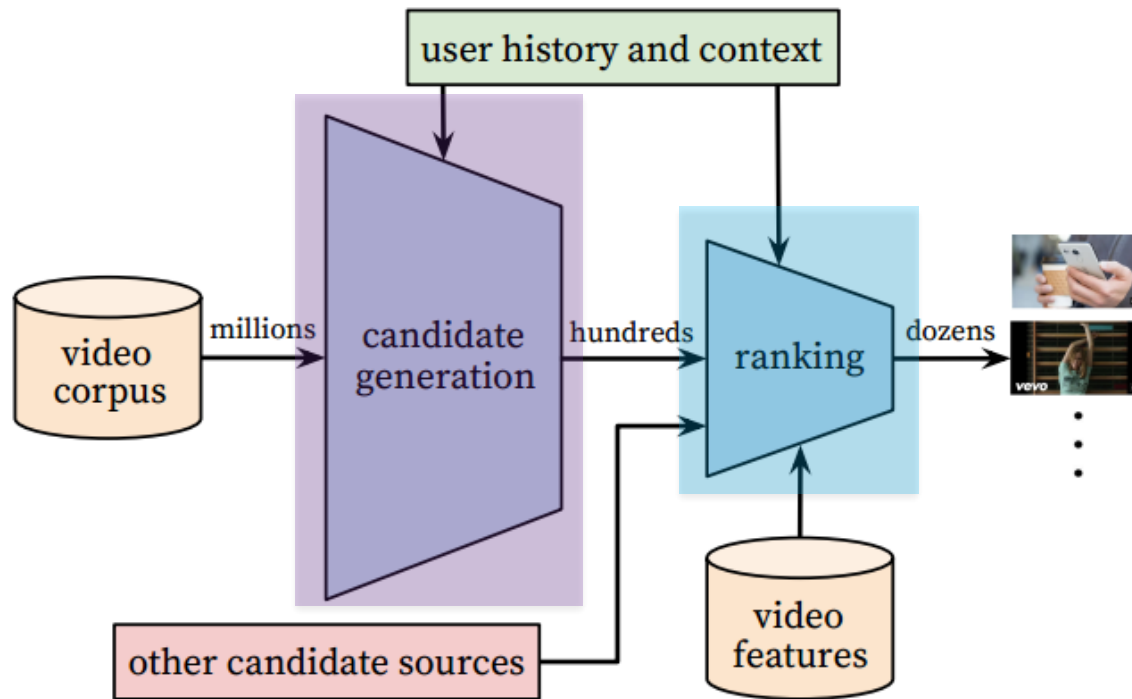
# An Overview of RS (from Alg. Perspective)

- **Key components**

- Data: User, Item, User-Item Interaction (e.g., score)
- Algorithm
- Explanations:
  - Why recommending these items (optional)



# Pipeline of RS in Industry



- **Coarse ranking**

- From millions of items to hundreds

- **Fine-grained ranking**

- From dozens to hundreds

**Two-stage approach**

**Note: third stage -> reranking (optional)**

# More Detailed Pipeline

---

- **Stage 1: Retrieval**

- Retrieval from many sources (e.g., real-time news, question-answers, short videos)

- **Stage 2: Rough ranking**

- Using simple models or complicated rules to generate candidate recommendation
- Size: Several hundred to several thousand
  - E.g., 1000 candidate recommendation

- **Stage 3: Candidates pass to next stage**

- Manual rules to filter or adjust some candidates

# More Detailed Pipeline

---

- **Stage 4: Sophisticated ranking**
  - Complicated machine learning (ML) models
  - Size: Dozens to hundreds
    - E.g., 100 candidate recommendation
- **Stage 5: Diversity**
  - Multiple diversity criteria
- **Stage 6: Blending rules (business)**
  - E.g., adding local news

# What we focus in this course

---

- **Stage 2: Rough ranking**

- Using simple models or complicated rules to generate candidate recommendation
- Size: Several hundred to several thousand
  - E.g., 1000 candidate recommendation

- **Stage 4: Sophisticated ranking**

- Complicated machine learning (ML) models
- Size: Dozens to hundreds
  - E.g., 100 candidate recommendation



# RS as a Function

---

- **Function**

- score =  $f(\text{user}, \text{item})$

- **Input:**

- user:
  - demographics: age, gender, income
  - preferences: bought products, watched videos
  - situational context: time, where, emotion
- item: *to be recommended*
  - description: image, text
  - attributes: price, location, category

- **Output:**

- score: relevance score; used for ranking

# Considerations for RS function

---

- **Function**

- score =  $f(\text{user}, \text{item}; w)$
- $w$ : parameters

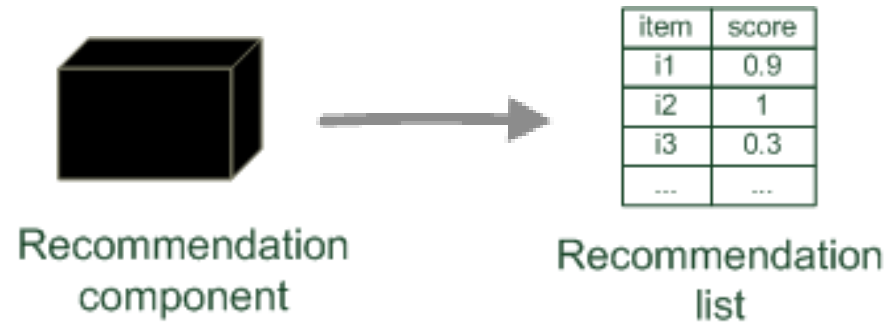
## Major Consideration for Specific Applications:

1. **user, item** – What information of them to consider
2. **f** – How to build the RS function (a.k.a., model)
  - e.g., collaborative filtering, matrix factorization, neural networks
3. **w** – How to optimize the parameters  $w$

# Paradigms of RS

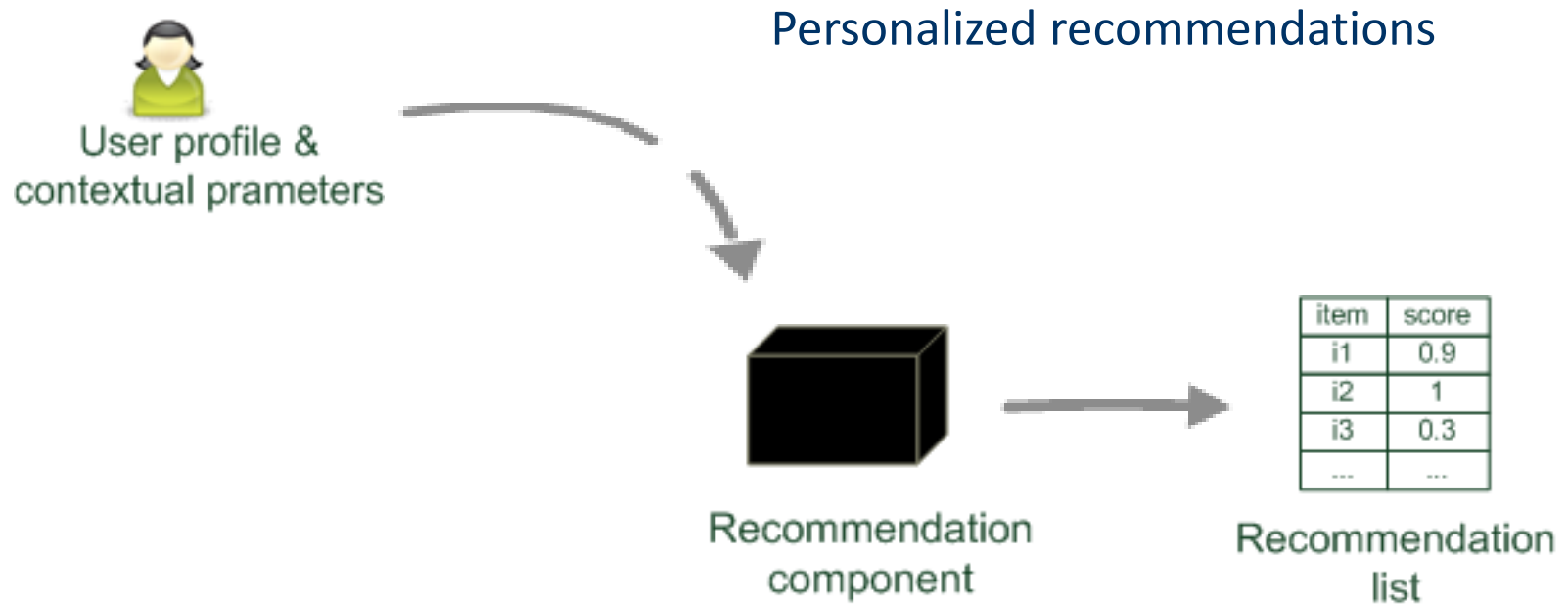
---

Recommender systems reduce information overload by estimating relevance

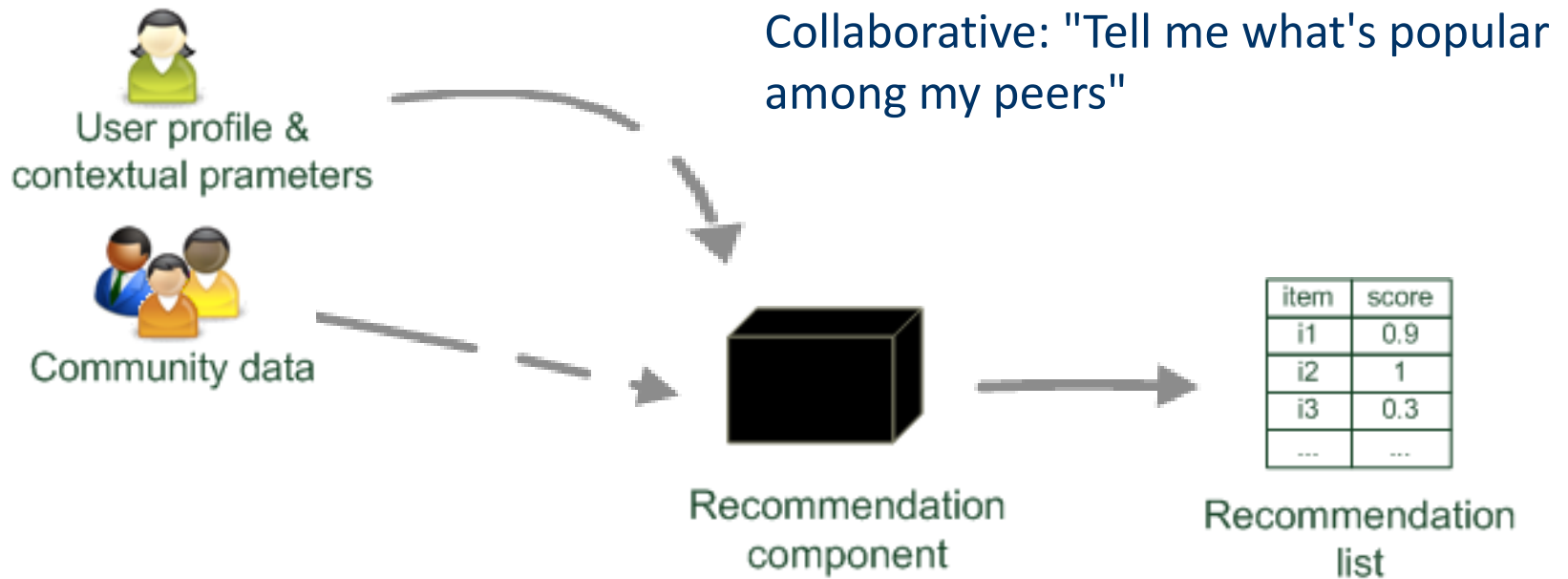


# Paradigms of RS

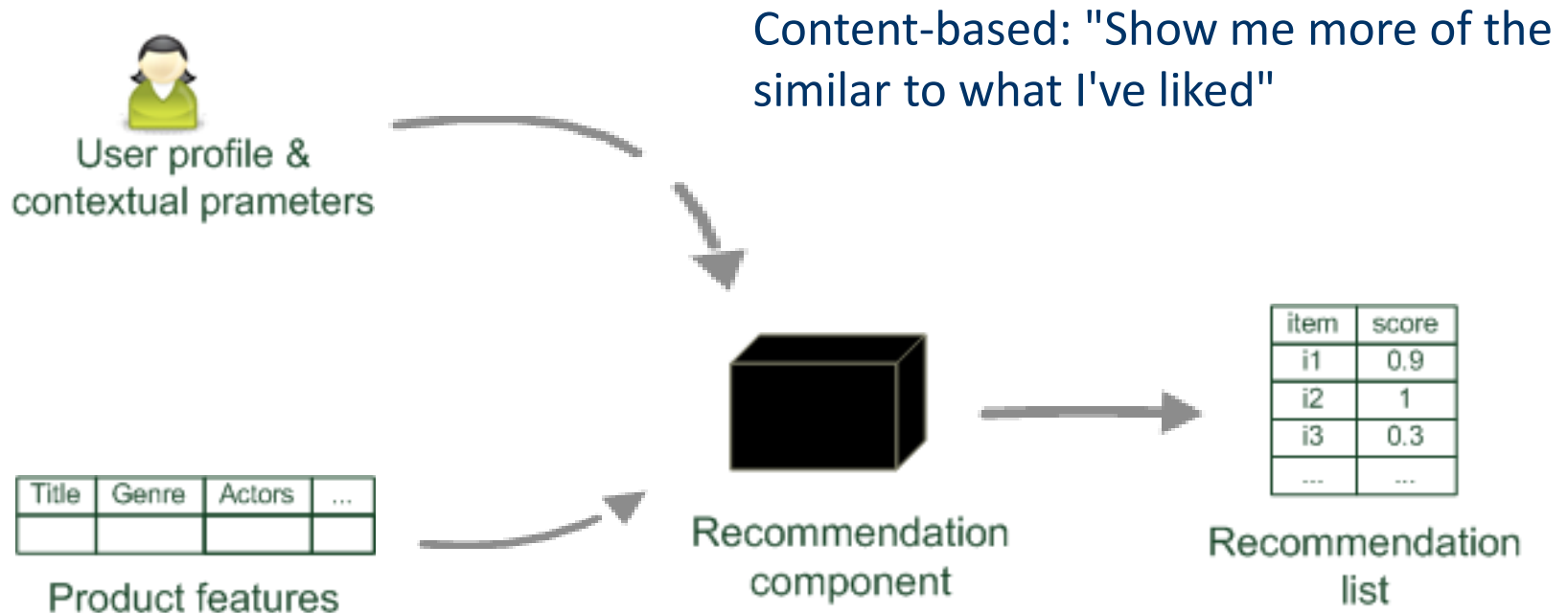
---



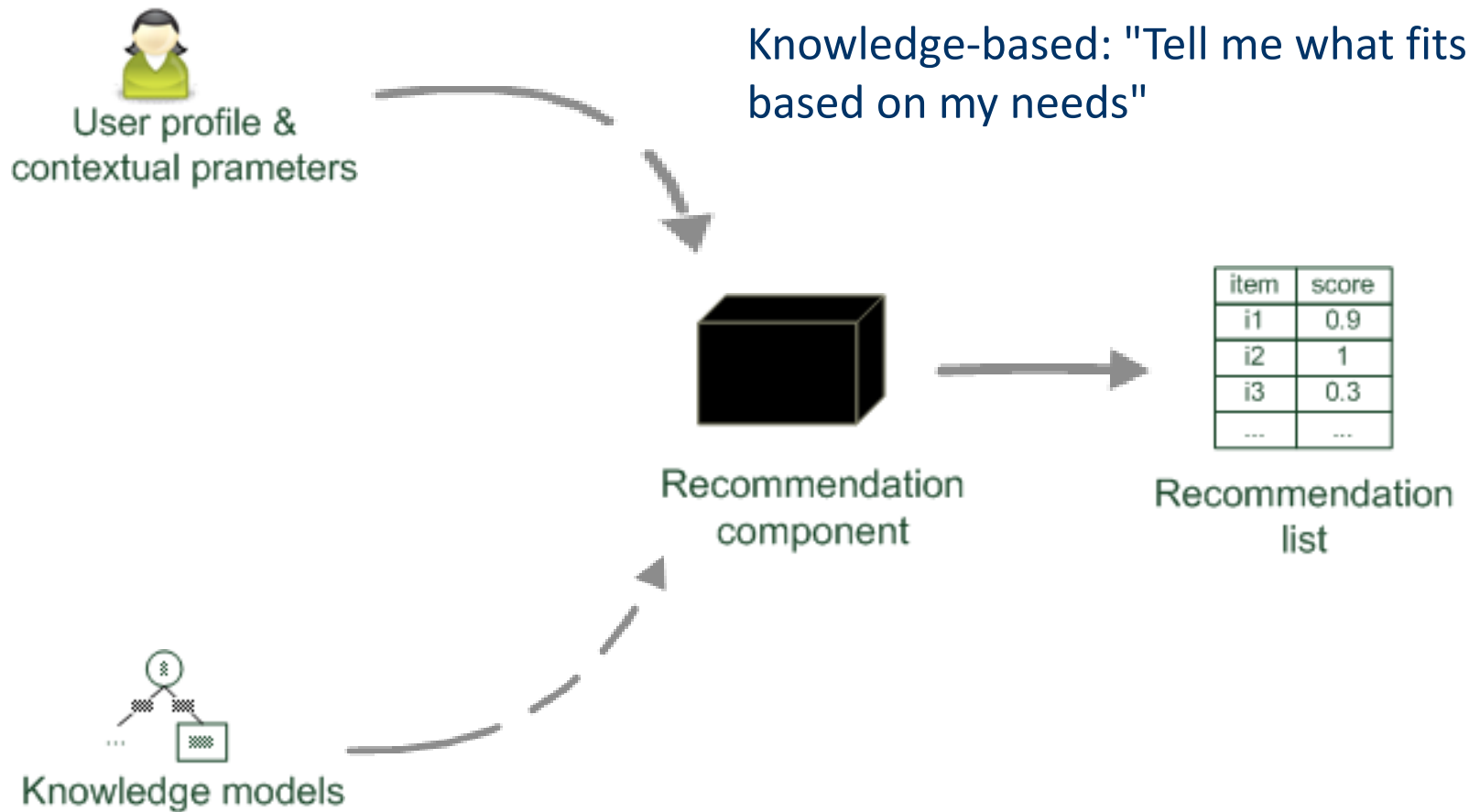
# Paradigms of RS



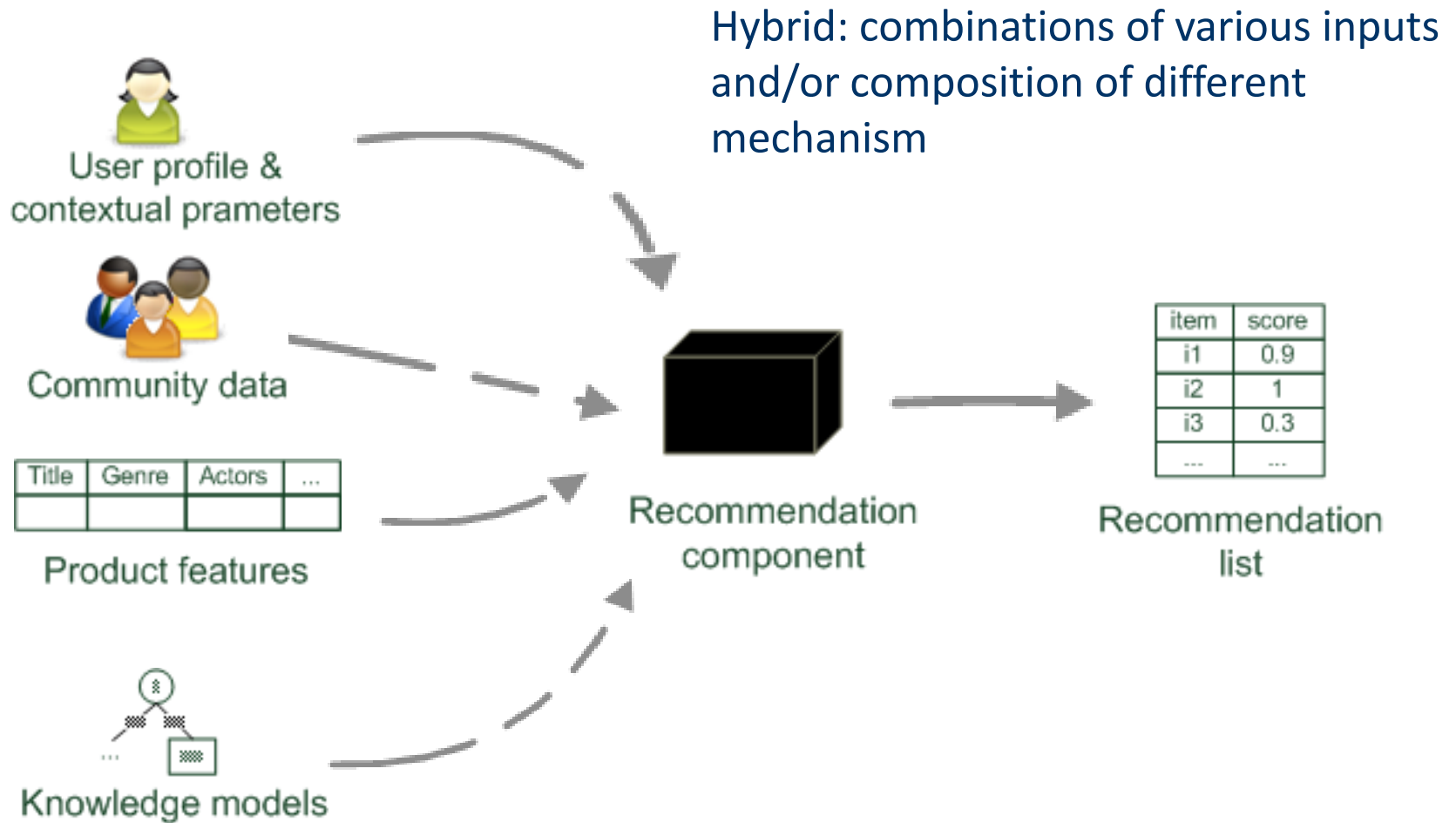
# Paradigms of RS



# Paradigms of RS



# Paradigms of RS



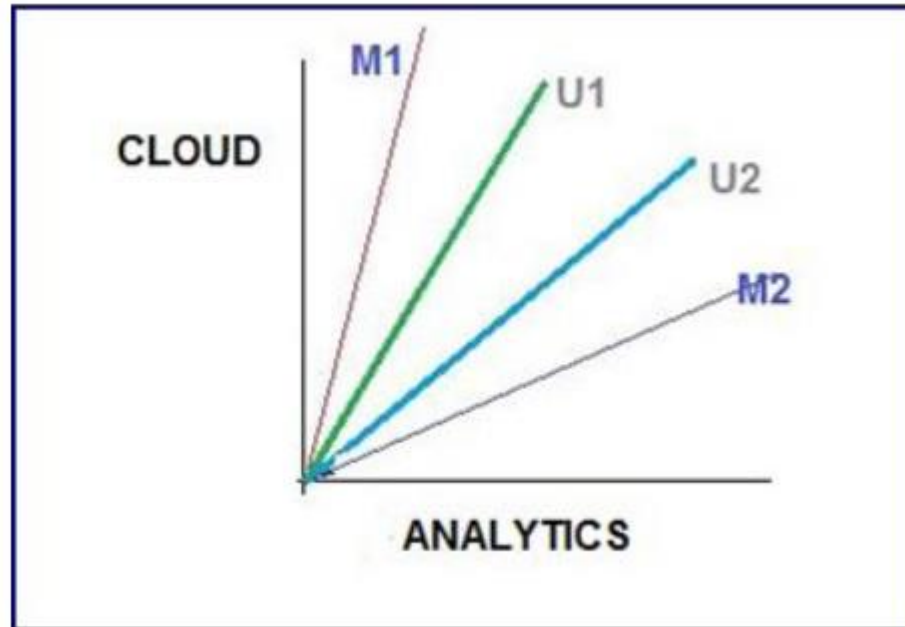


# Representative RS algorithms

- **Content-Based Algorithm**

- **Vector Space Model**

- Using content to represent an item as a vector



# Representative RS algorithms

- **Collaborative Filtering**

- Collaborative Signals: Behavior Similarity of Users
- Memory-based CF
  - User-CF
  - Item-CF
- Model-based CF
  - Latent factor model

$(u_1 \text{ (blue)} \quad i_1 \text{ (white)})$	5
$(u_2 \text{ (blue)} \quad i_1 \text{ (white)})$	3
$(u_2 \text{ (blue)} \quad i_2 \text{ (white)})$	4
$(u_3 \text{ (blue)} \quad i_2 \text{ (white)})$	1
$(u_3 \text{ (blue)} \quad i_3 \text{ (white)})$	2
$(u_3 \text{ (blue)} \quad i_4 \text{ (white)})$	4
... ..	...

		item			
		1	2	3	4
user	1	5	?	?	?
	2	3	4	?	?
	3	?	1	2	4
		...	...	...	...

Interaction Matrix

## 1. Memory-based CF:

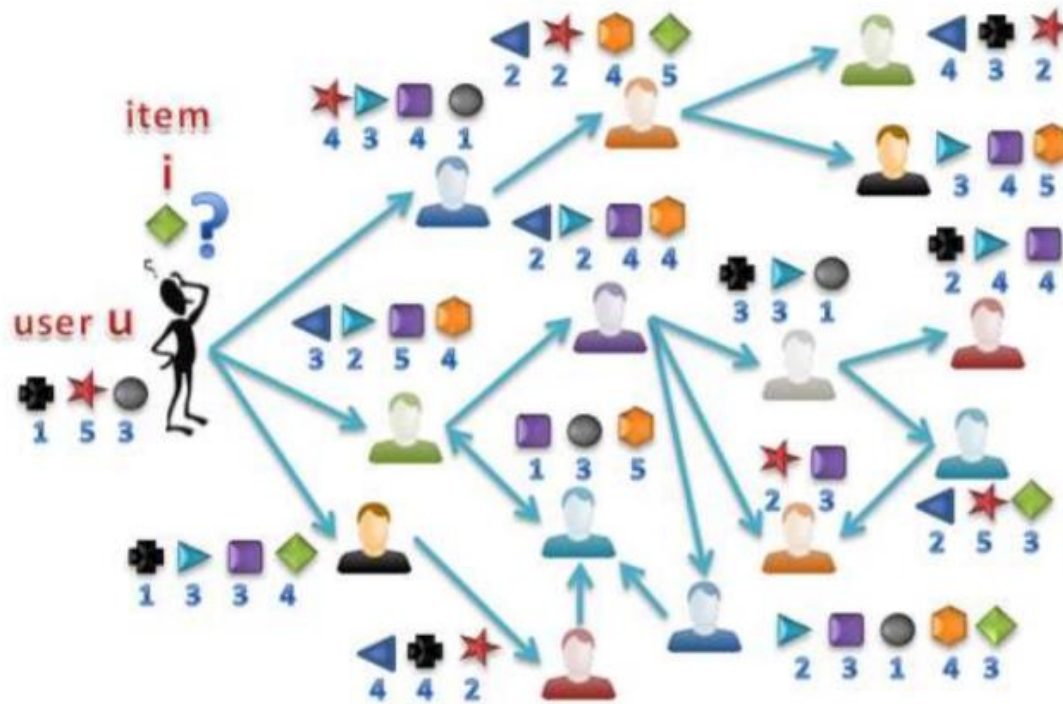
Predict by **memorizing** similar users' (or items') ratings

## 2. Model-based CF:

Predict by **inferring** from an underlying model.

# Representative RS algorithms

- **Community-Based Algorithm**
  - Social-based recommendation algorithms
    - TrustWalker<sup>[1]</sup>, SocialRec<sup>[2]</sup>, etc.

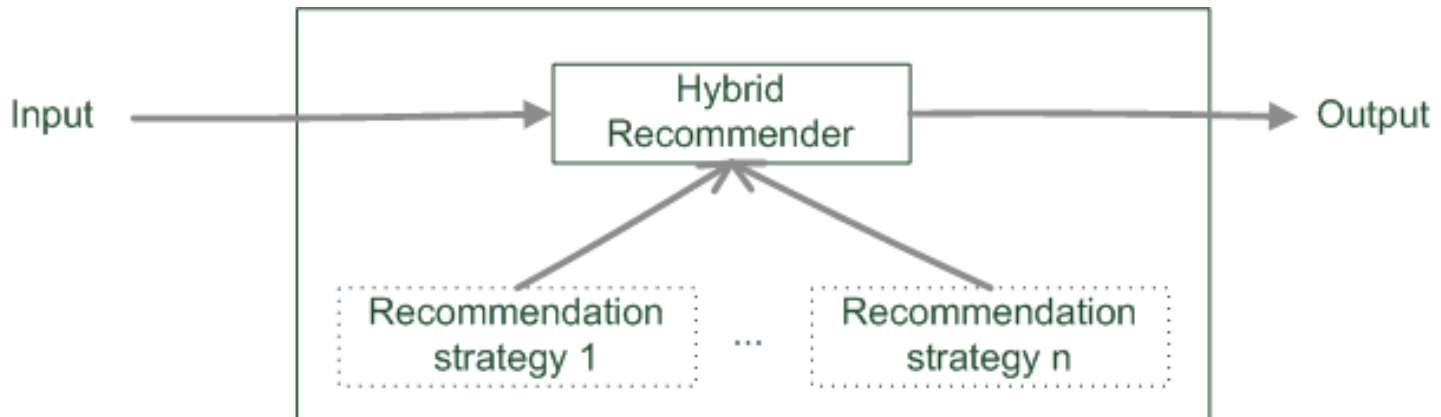


[1] TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation, KDD'09

[2] Recommender systems with social regularization, WSDM'11

# Representative RS algorithms

- **Hybrid Recommendation Algorithm**
  - Merge multiple RS algorithms
    - CTR<sup>[1]</sup>

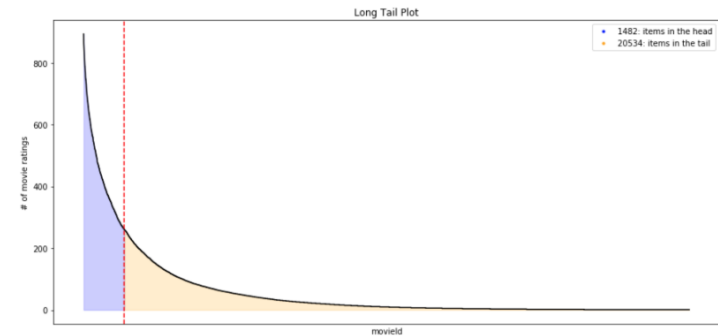


---

**Why do we need  
recommender system?**

# Benefits of RS (Why Companies Need RS?)

- **Increasing sales volume of commodities**
- **Increasing consumption of more item types**
  - Beneficial for long-tail items
- **Increasing user satisfaction**
- **Increasing user loyalty**
  - User historical behaviors are stored
- **Better learning user preference**



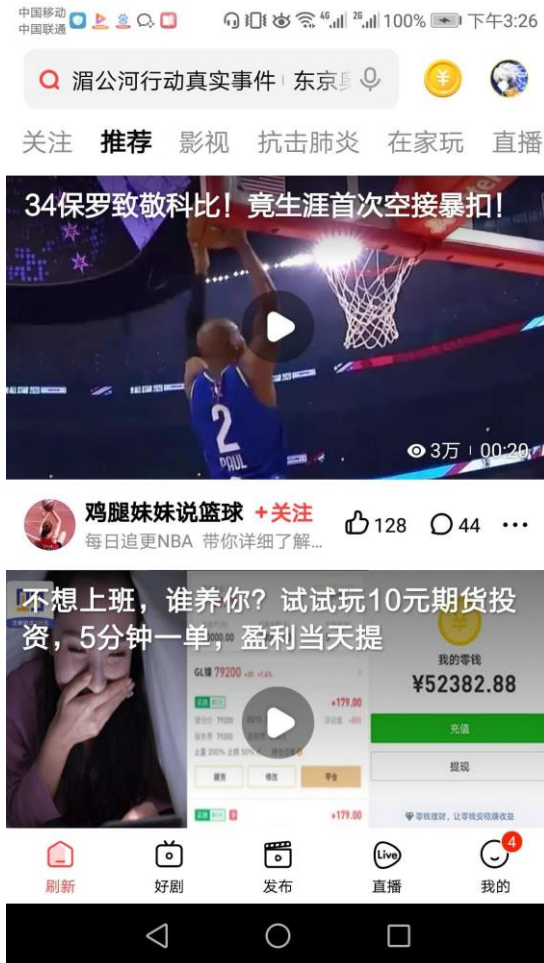
Long tail plot. (Sample of Movielens 20m ratings data)

# Real Applications

---

- **Entertainment**
  - Music, movie
- **Content**
  - News feeds, news portal
- **E-commerce**
  - Products
- **Service**
  - Tourism, knowledge sharing

# Homepage RS: Entertainment



Streaming Short Video



Long Video



# Homepage RS: Content



News Feeds

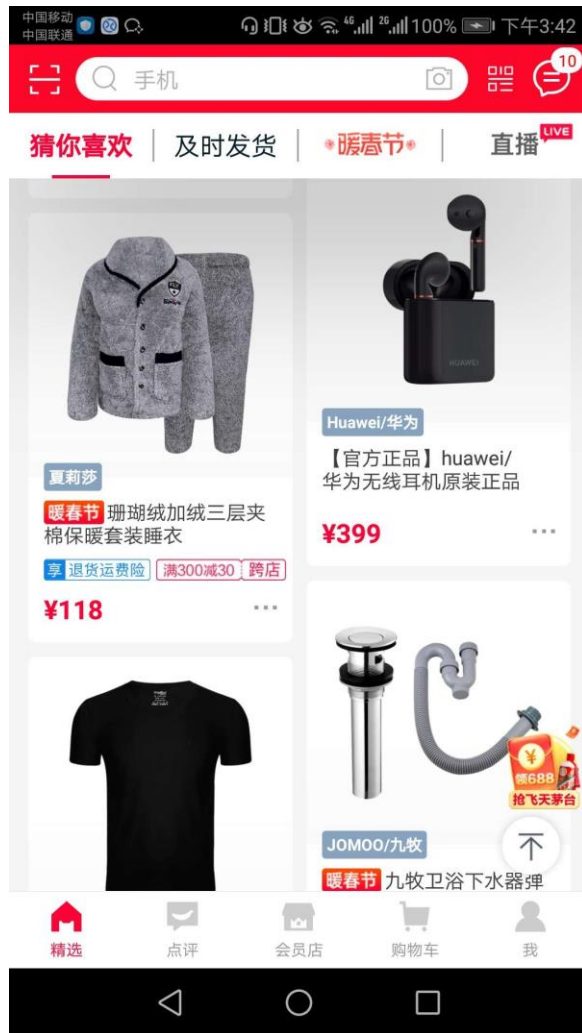


## 猜你喜欢

- 工信部提醒：短信可以为您提供“行程证明”
- 美媒妄称“中国是亚洲病夫” 接到中方交涉后仍不道歉
- 日本疫情蔓延中国是否提供支持或援助?外交部回应
- 一个美国人发问：如果疫情发生在休斯敦 该怎么办
- 为什么要解剖新冠肺炎逝世患者遗体？
- 纽约载客司机担心感染拒拉中国乘客 市长怒了
- 新发明隔离帽来了：让医护的口罩勒痕消失
- 俄媒：科学家发现病毒致命弱点 有望促成疫苗研发

News Portal

# Homepage RS: E-commerce



Product App

# Homepage RS: Service



Knowledge Sharing



Trip

# Relevance Recommendation



自营 西铁城手表京东自营旗舰店 联系客服 关注店铺

### 男表NH8350-08AB

距离结束 23 : 57 : 03

累计评价 3.8万+


50.00, 超出数量以结算价为准, 仅限

99元免基础运费(20kg内)

8:00前下单, 预计02月21日(周五)送达, 受全国道路等待。

看了又看


CITIZEN 品牌特和旗舰店



西铁城 (CITIZEN) 手...

¥ 2520.00

CITIZEN 品牌特和旗舰店



西铁城(CITIZEN)手表 ...

¥ 1664.00

---

# **How to evaluate recommender system?**

# Evaluation Method

---

- **Offline experiment**

- Use historical behavior data for testing
- Segmenting data into training and testing.

- **Online experiment**

- Use instant interactions with users for testing

- **User study**

- Hiring users for testing

# Evaluation Metric

---

- Prediction accuracy
  - E.g., Root Mean Squared Error (RMSE), NDCG
- Coverage
  - Percent of items in the training data the model is able to recommend on a test set
- Novelty
- Diversity
- ...

---

# **Where to find references for advanced RS?**



# DBLP and Google

---

- Recommender System
  - **ACM Recommender Systems conference (RecSys)**
- Intelligent Information Processing
  - ACM SIGKDD Int. Conf. on Knowledge Discovery in Databases and Data Mining (**KDD**)
  - ACM SIGIR Conference on Research and Development in Information Retrieval (**SIGIR**)
  - The Web Conference (**WWW**)
  - Int. Conf. on Web Search and Data Mining (**WSDM**)
- Other related conferences
  - ML conferences: ICML, NIPS
  - AI conferences: AAAI, IJCAI
- Journals
  - IEEE Trans. On Knowledge and Data Eng. (TKDE)
  - ACM Transactions on Information Systems (TOIS)
  - ACM Trans. on KDD
  - Data Mining and Knowledge Discovery (DAMI or DMKD)

# Experience Sharing of Competition

- Kaggle
  - <https://www.kaggle.com/competitions>
    - Winner Experience:
      - <https://www.kaggle.com/c/criteo-display-ad-challenge/discussion/10555>
- 阿里天池
  - <https://tianchi.aliyun.com/competition/gameList/algorithmList>
- Biendata
  - <https://biendata.com/competition>





# Thanks! Questions?

**Wei Zhang**

✉ zhangwei.thu2011@gmail.com

🏠 <https://weizhangltt.github.io/>