推荐算法综述

郭鑫鹏

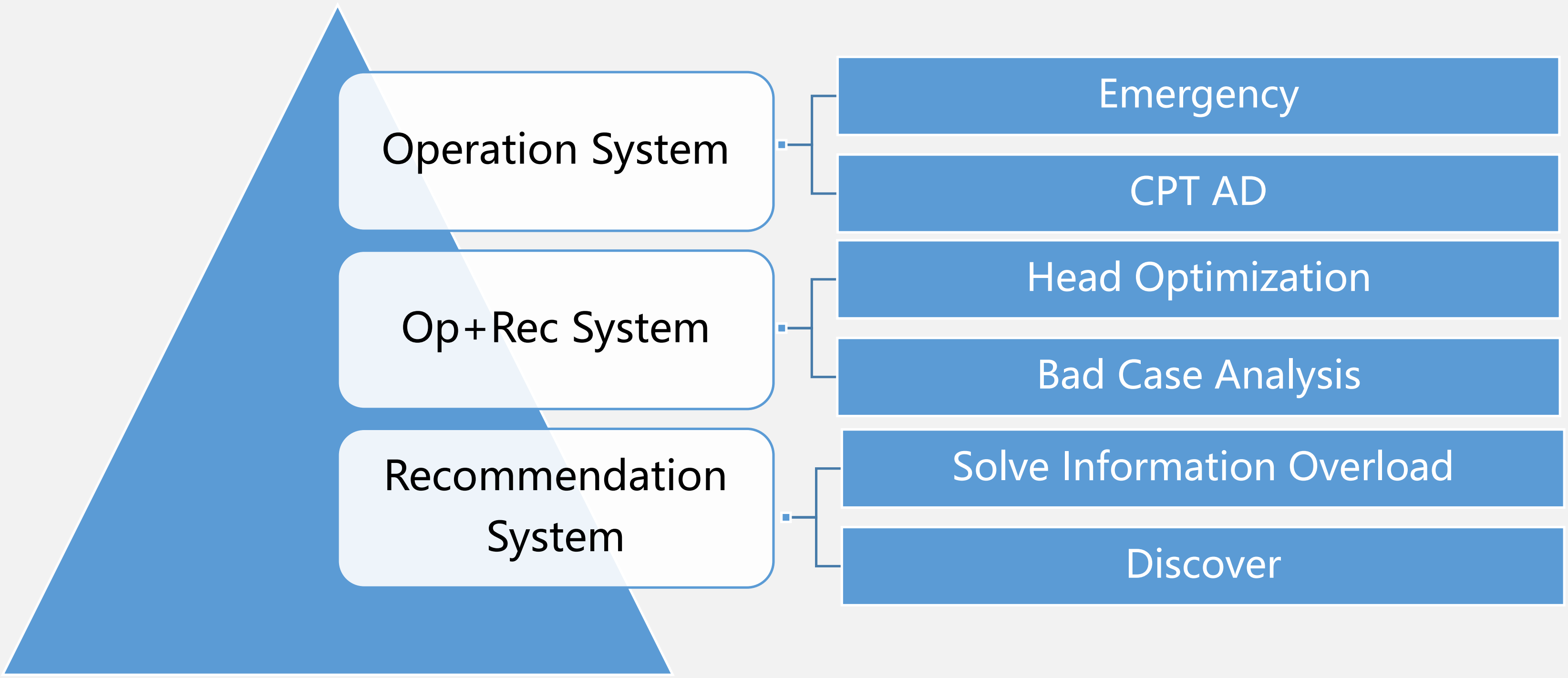
互联网基础平台部 数据挖掘组

CONTENTS

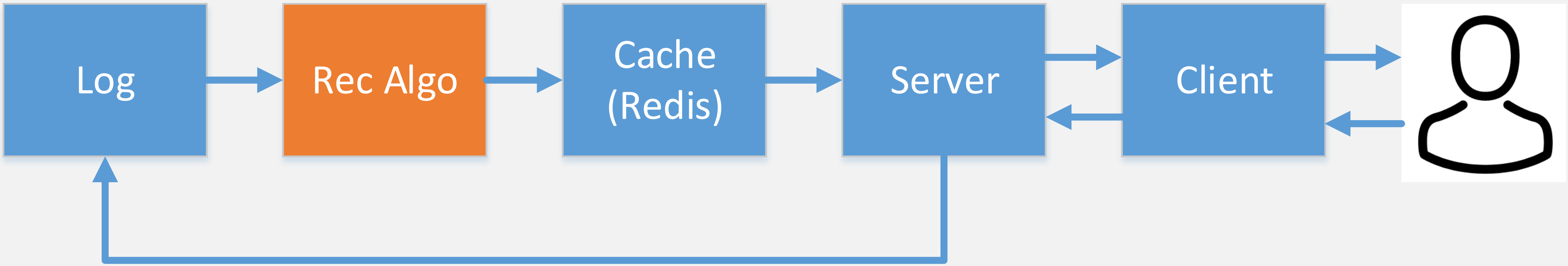
Why RecSys

- 信息过载
- 信息获取场景
  - 搜索
    - 主动需求
    - 明确需求（指导自己想要什么）
  - 推荐：浏览需求
    - 浏览需求
    - 模糊需求（不知道如何刻画）
    - 探索发现（Something wonderful that you did not know existed）

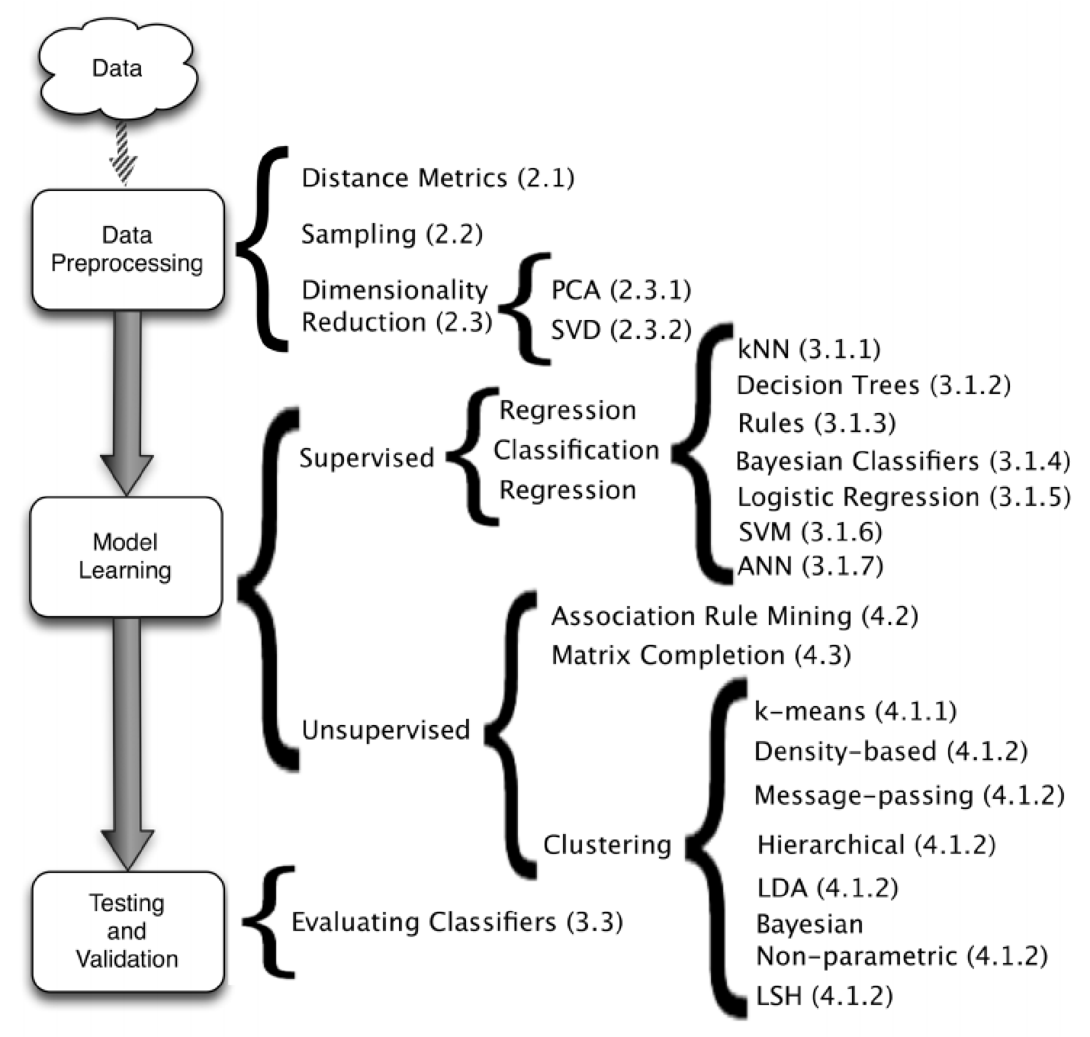
Three Layers



Data Closed Loop



Recommendation as data mining



The core of the Recommendation Engine can be assimilated to a general data mining problem

(Amatriain et al. Data Mining Methods for Recommender Systems in Recommender Systems Handbook)

CONTENTS

Two Basic Entity

User

Item

1. Song
2. Singer
3. Song List
4. MV
5. Album

...

Three Basic Relationships

User-User

1. Circle
2. Discover New People (Famous People)
3. People U may know (Social Relation Chain)

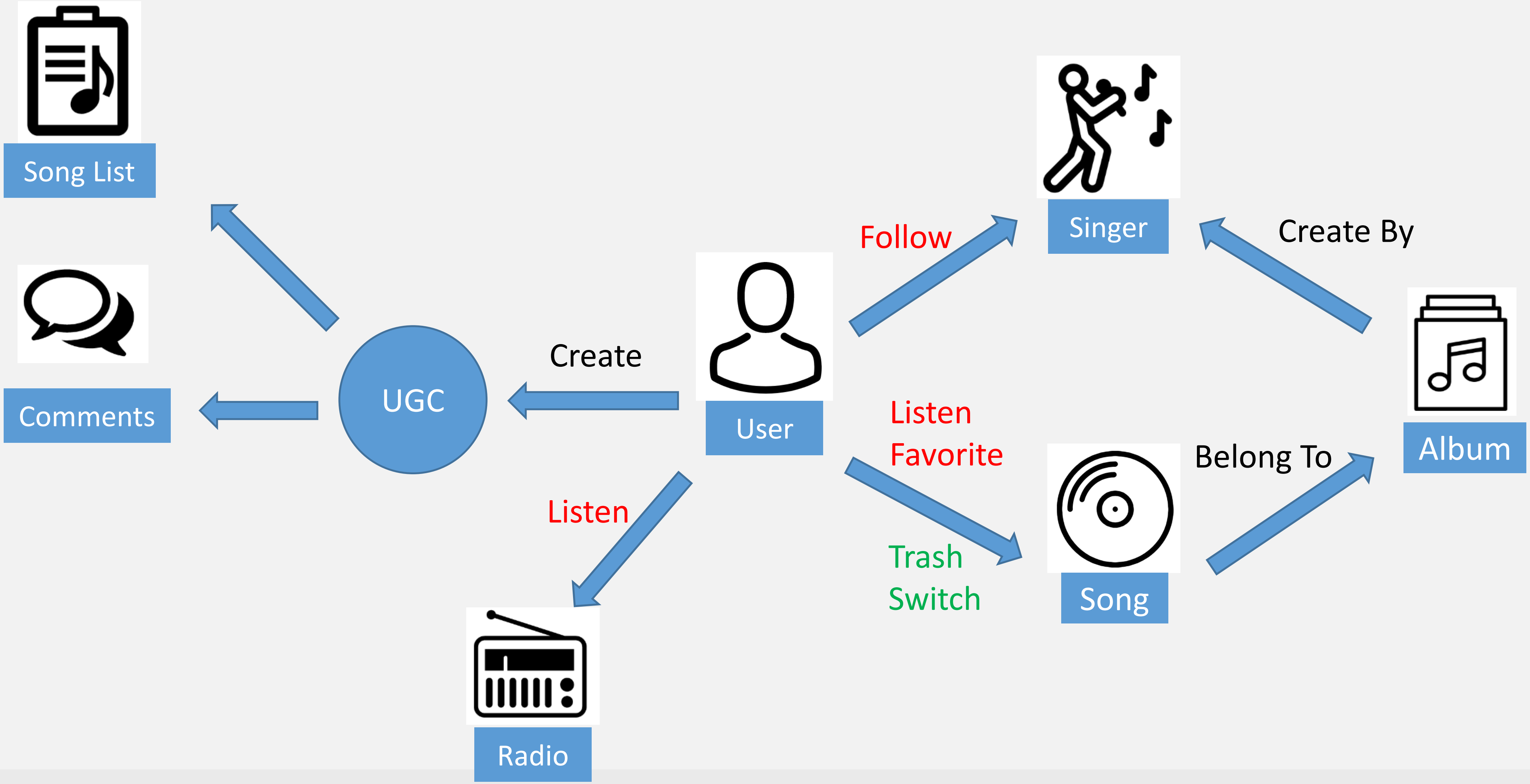
Guess U like

User-Item

Item-Item

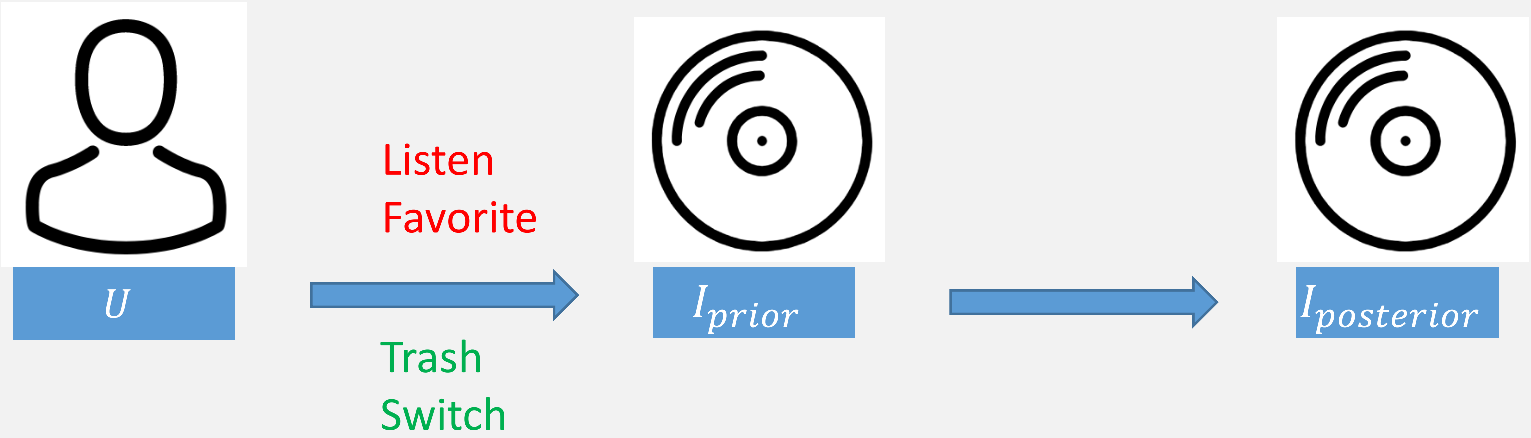
1. Similar Songs
2. Single Song Radio
3. Buy This Also Buy
4. View This Also View

Knowledge Graph

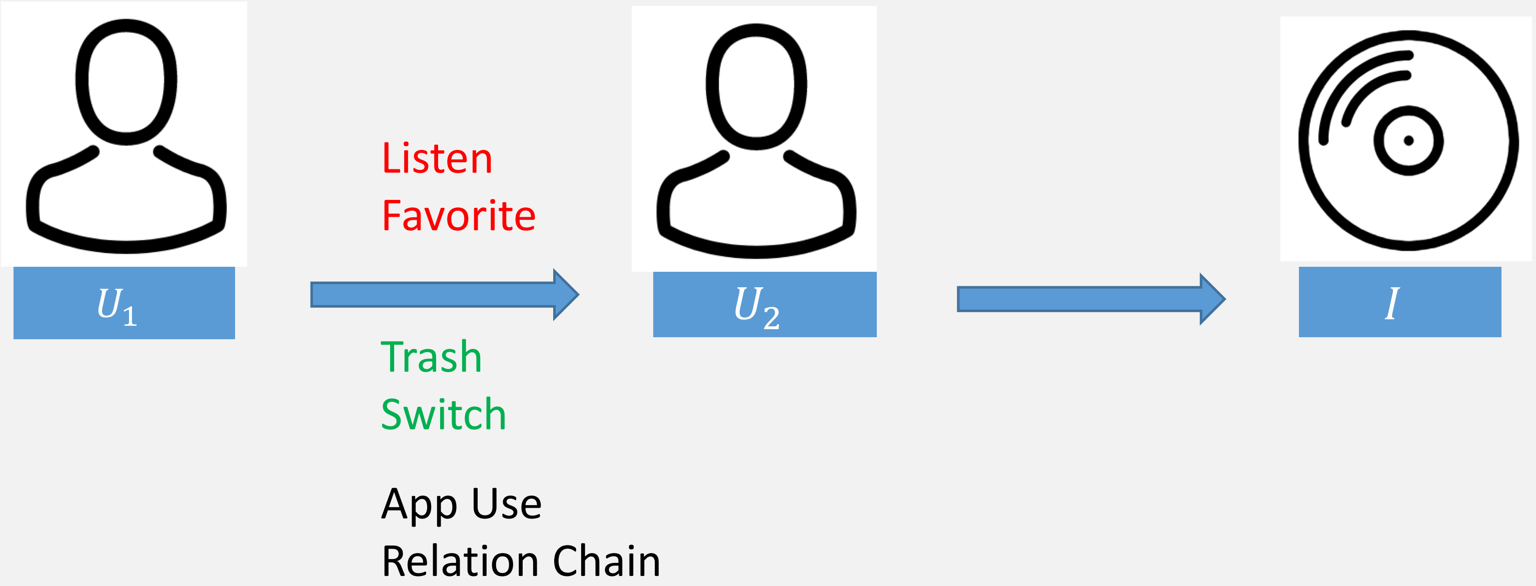


CONTENTS

Item-CF

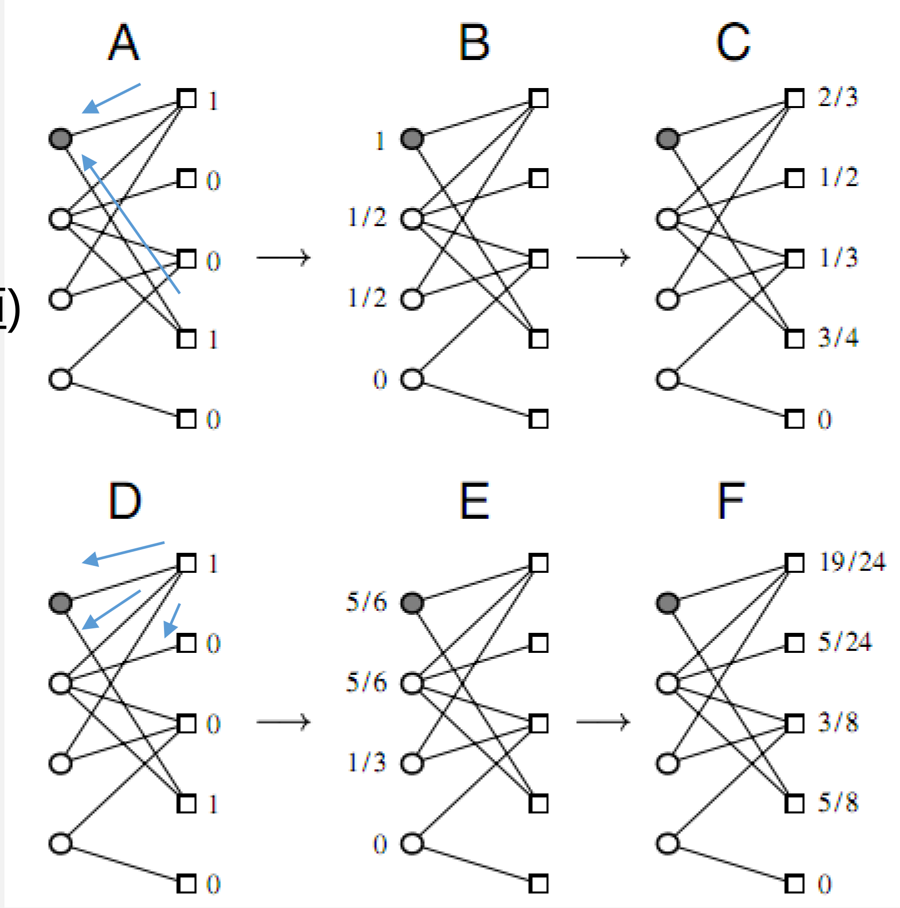


User-CF



Probs & Heats

Heats
(不守恒)



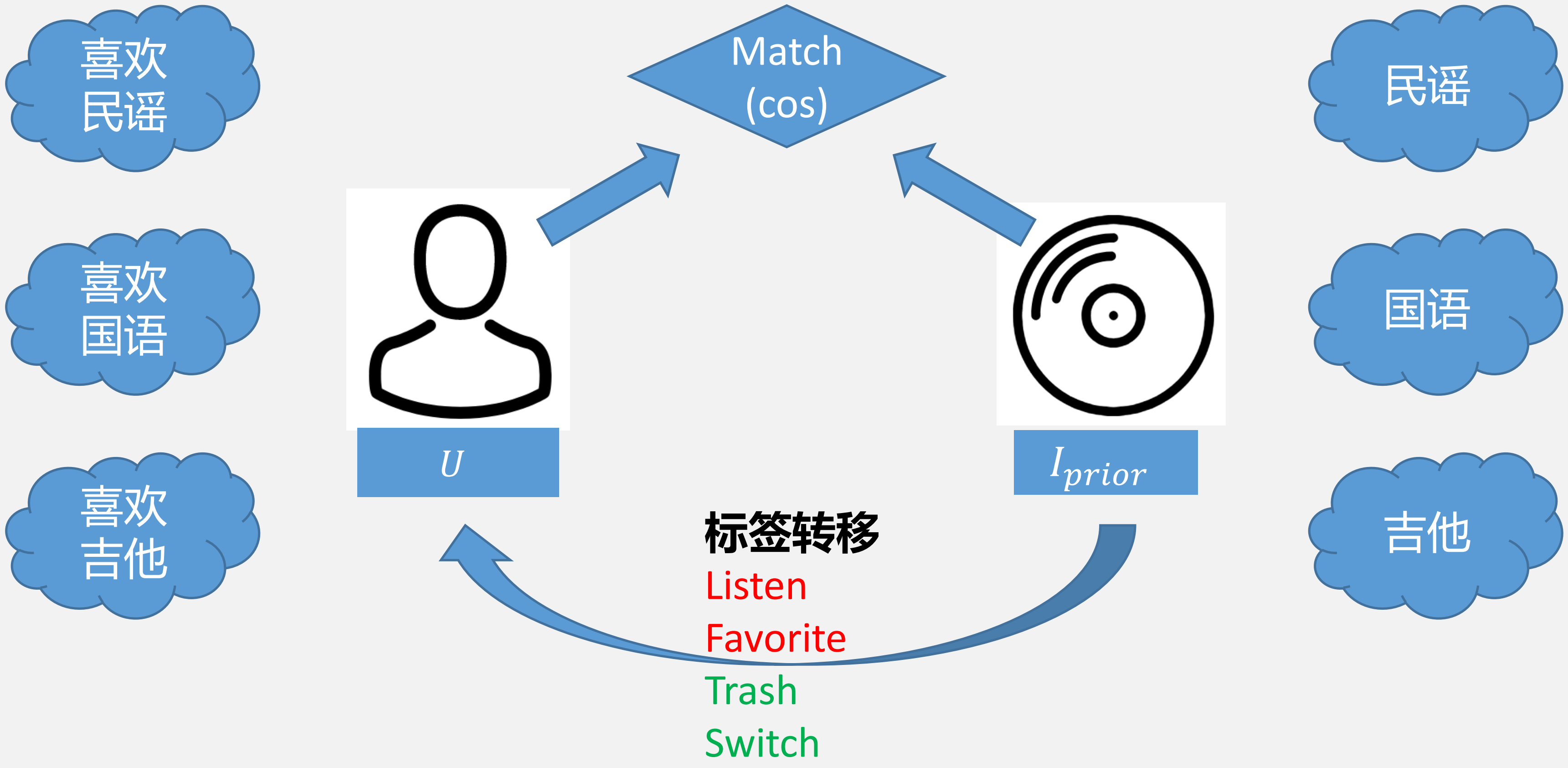
Probs
(守恒)

Fig. 1. The HeatS (*A*, *B*, *C*) and ProbS (*D*, *E*, *F*) algorithms (Eqs. **1** and **2**) at work on the bipartite user-object network. Objects are shown as squares, users as circles, with the target user indicated by the shaded circle. While the HeatS algorithm redistributes resource via a nearest-neighbor averaging process, the ProbS algorithm works by an equal distribution of resource among nearest neighbors.

$$W_{\alpha\beta}^{H+P} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^{\lambda}} \sum_{j=1}^{u} \frac{a_{\alpha j} a_{\beta j}}{k_j},$$

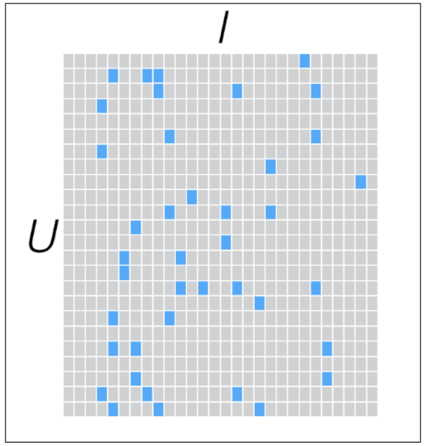
- Heats provides personalization and novelty
- Probs provides accuracy

Content Based CF



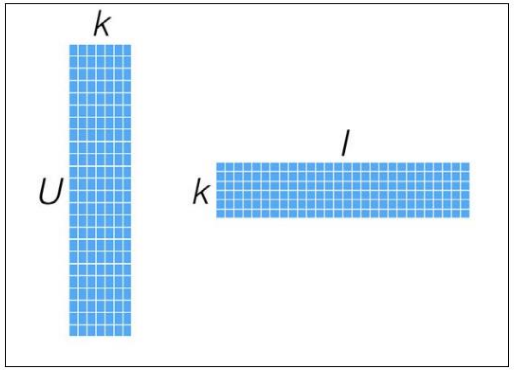
CONTENTS

ALS



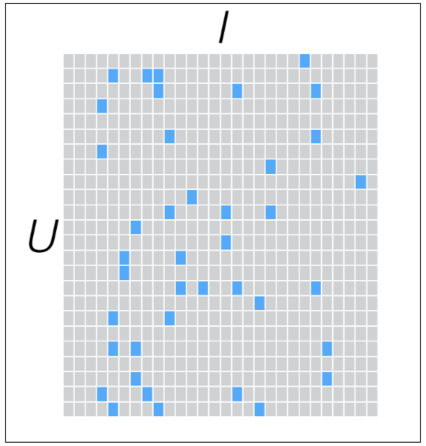
A sparse ratings matrix

分解



The user- and item-factor matrices

重组



A sparse ratings matrix

In this paper, we use *alternating least squares* (ALS) to solve the low-rank matrix factorization problem as follows:

Step 1 Initialize matrix M by assigning the average rating for that movie as the first row, and small random numbers for the remaining entries.

Step 2 Fix M, Solve U by minimizing the objective function (the sum of squared errors);

Step 3 Fix U, solve M by minimizing the objective function similarly;

Step 4 Repeat Steps 2 and 3 until a stopping criterion is satisfied.

Implicit ALS

- Explicit：Star Ratings
- Implicit：views, clicks, purchases, likes, shares etc

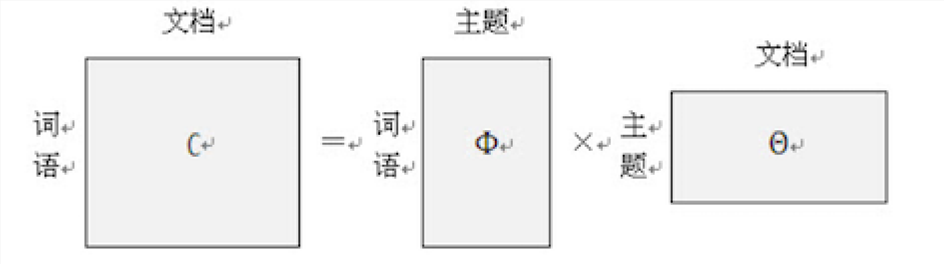
$$L_{WRMF} = \sum_{u,i} c_{ui} \left(p_{ui} - \mathbf{x}_u^{\intercal} \cdot \mathbf{y}_i \right)^2 + \lambda_x \sum_u \|\mathbf{x}_u\|^2 + \lambda_y \sum_u \|\mathbf{y}_i\|^2$$

用户和物品之间有交互就让pui等于1，没有就等于0

函数中还有一个c_{ui}的项，它用来表示用户偏爱某个商品的置信程度

LDA

- **痛点**
  “乔布斯离我们而去了” 和 “苹果什么时候降价”如何关联？
- **思路**
  - 将word映射到topic维度



  - 概率表示

$$P(\text{词语}|\text{文档}) = \sum_{\text{主题}} (\text{词语}|\text{主题}) * (\text{主题}|\text{文档})$$

$$P(\text{文档}|\text{词语}) = \sum_{\text{主题}} (\text{文档}|\text{主题}) * (\text{主题}|\text{词语})$$

CONTENTS

LR/FTRL

- $Linear: y \coloneqq w0 + sum(wi * xi)$
- $\text{Logistic}: y \coloneqq sigmoid(w0 + sum(wi * xi))$

- 优点
  - 简单；可解释；易扩展；效率高；易并行
- 缺点
  - 无法捕捉特征组合

$$\text{FTRL-Proximal} \quad x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + \quad t\Psi(x) \quad + \tfrac{1}{2} \textstyle\sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - x_s)\|_2^2$$

Feature Interactions

- Download apps for food delivery at meal time(app category & time-stamp)
- Male Teenagers like shooting games and RPG games(app category & user gender & user age)

FM

$$\text{FM}: \hat{y}(x) := \underbrace{w_0 + \sum_{i=1}^{n} w_i x_i}_{\text{LR模型}} + \underbrace{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j}_{\text{Dense化两两特征}}$$

$$\langle v_i, v_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} = \underbrace{\begin{bmatrix} 0.1 & 0.2 & 0.6 & 0.8 & 0.1 & 0.2 \end{bmatrix}}_{v_i} \quad \left.\begin{bmatrix} 0.4 \\ 0.2 \\ 0.4 \\ 0.2 \\ 0.4 \\ 0.2 \end{bmatrix}\right\} v_j$$

GBDT

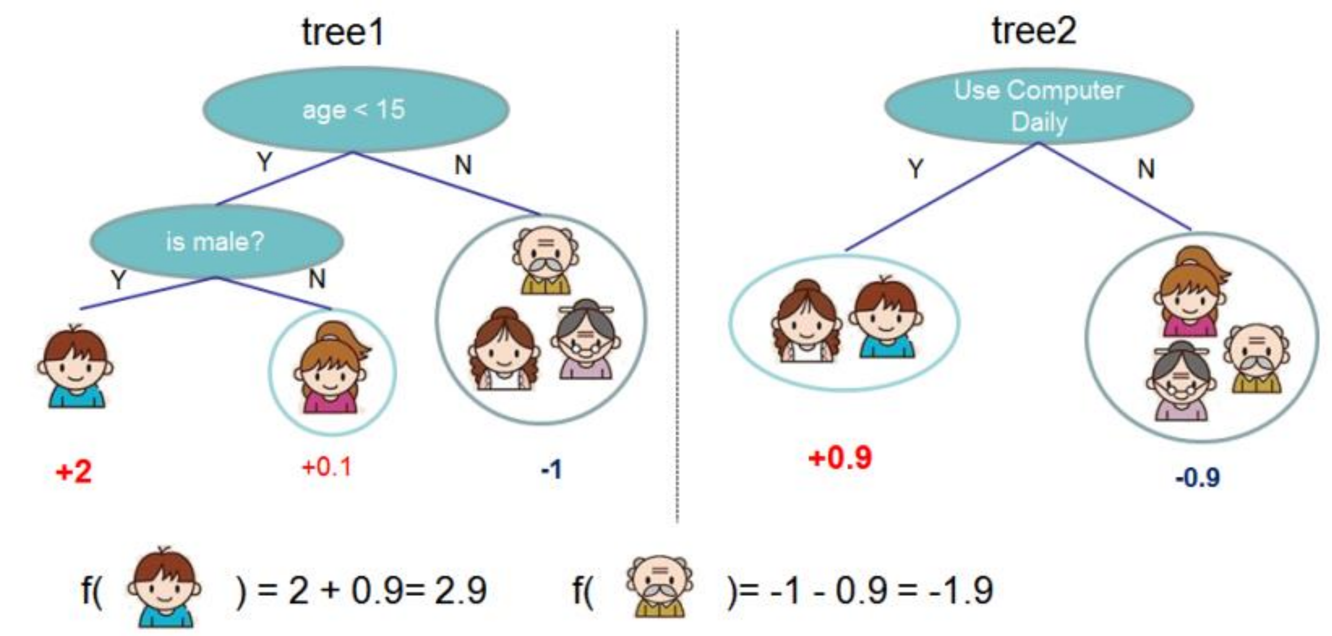


Figure 1: Tree Ensemble Model. The final prediction for a given example is the sum of predictions from each tree.

GBDT+LR

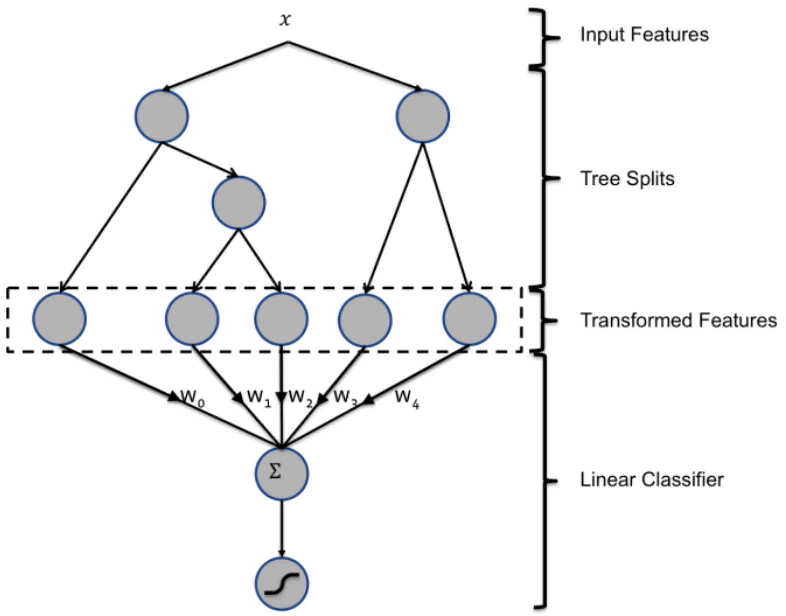


Figure 1: Hybrid model structure. Input features are transformed by means of boosted decision trees. The output of each individual tree is treated as a categorical input feature to a sparse linear classifier. Boosted decision trees prove to be very powerful feature transforms.

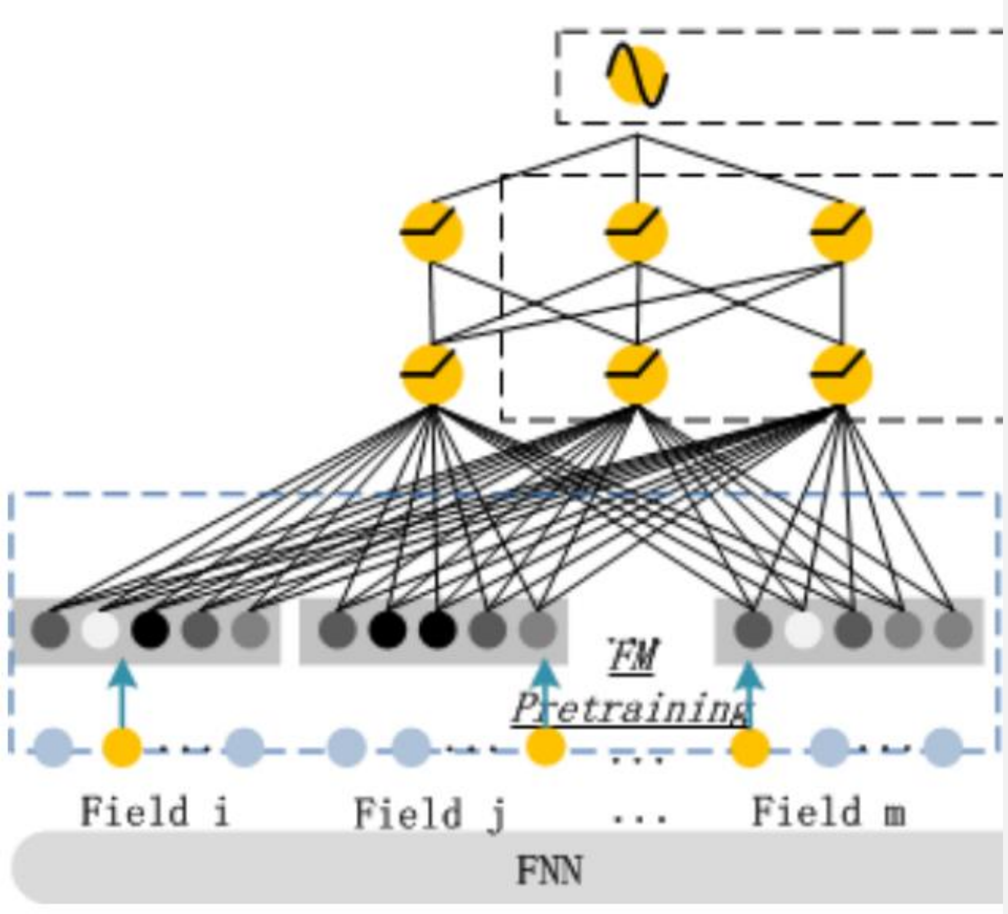
- GBDT特征工程
  - 特征分段
  - 特征组合
  - 特征选择
- LR建立模型

CONTENTS

特征

- 连续特征
  - 收入，身高，体重
  - 适合DNN处理
- 离散特征
  - 职业，性别，学校
  - 不适合DNN处理

FNN



- Embedding vectors of categorical data
- PreTrained FM
- 串联结构

PNN

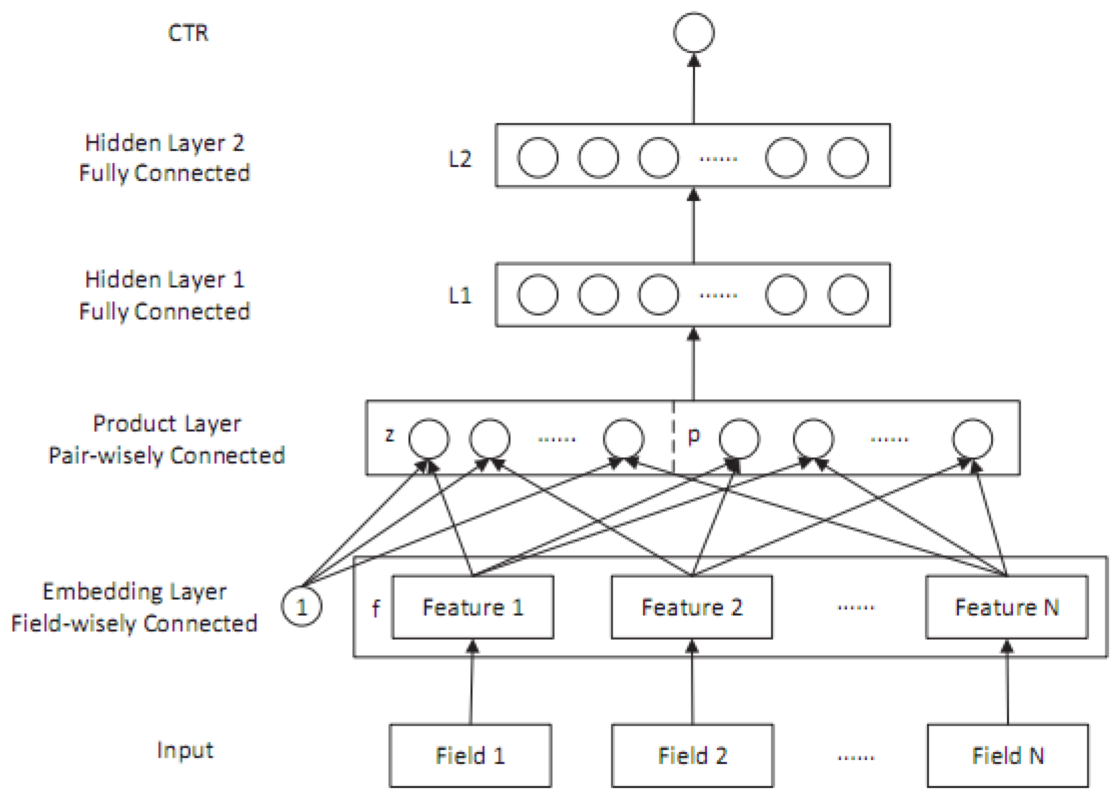
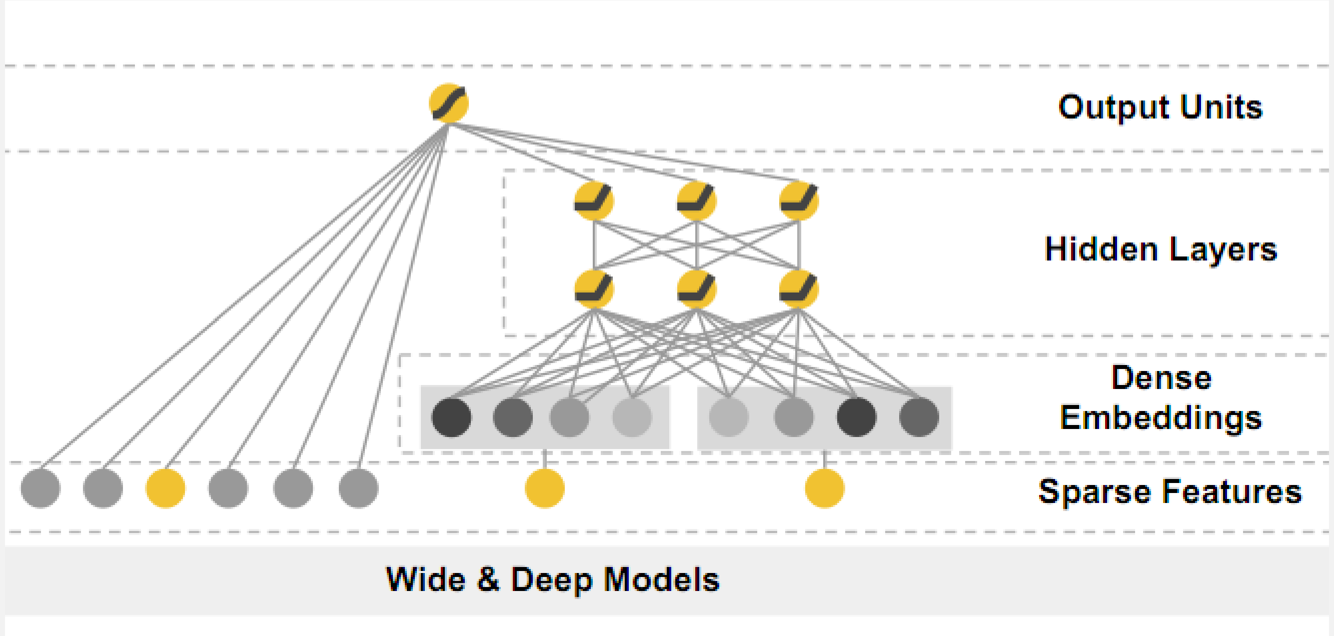


Fig. 1: Product-based Neural Network Architecture.

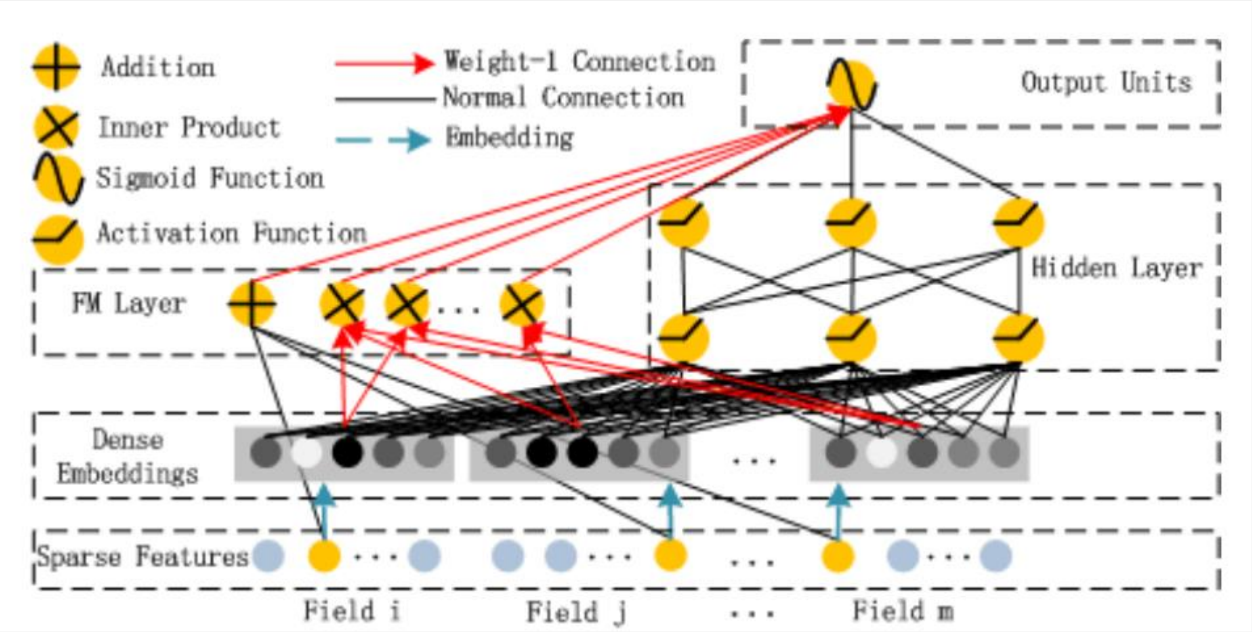
- Without PreTraining
- Product layer to model inter-field feature interactions

Wide_n_Deep



- Joint Training
- Continuous VS Categorical(embedding)
- ReLU VS SoftMax
- Memorization(wide) VS Generazation(deep)

DeepFM



W&D缺点

- A strong bias towards low- or high- interactions
- Require Expertise engineering

所以用FM代替Wide

CONTENTS

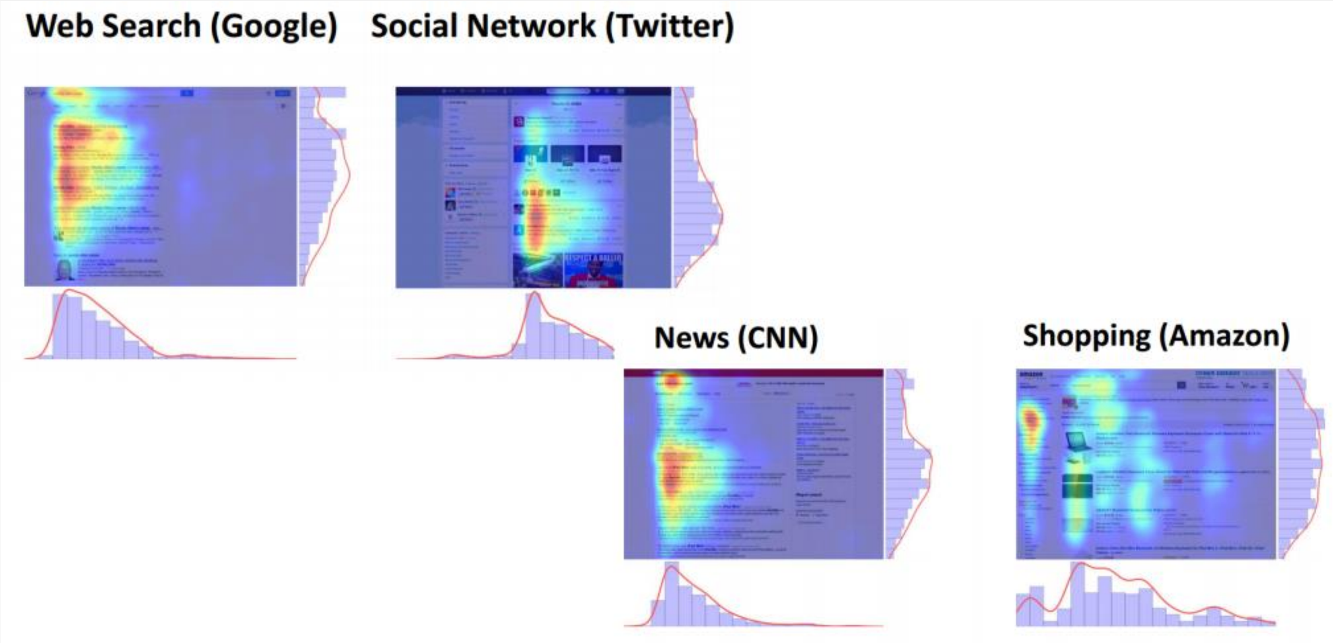
Pros & Cons

- Limitations of Collaborative Filtering
  - Cold Start
  - Popularity Bias
- Limitations of Content Based
  - Require content that can be encoded as meaningful features

Some other things

- User Interface
- System requirements(efficiency, scalability, privacy)
- Explanations
- Diversity vs Accuracy
- Freshness vs Stability
- Depth vs Coverage
- MAB

User Attention



?	**?**	**?**
?	?	?
?	?	?

排序列表如何放置？

Other Models

- 强化学习
- 迁移学习

Reference

- **【Two Decades of Recommender Systems at Amazon.com】** https://www.computer.org/csdl/mags/ic/2017/03/mic2017030012.html
- 【PNN】 https://arxiv.org/pdf/1611.00144.pdf
- 【Wide&Deep】 https://arxiv.org/pdf/1606.07792.pdf
- 【DeepFM】 https://arxiv.org/pdf/1703.04247.pdf

Q&A