# 大数据导论
# Introduction to Big Data

## 大数据推荐系统基础

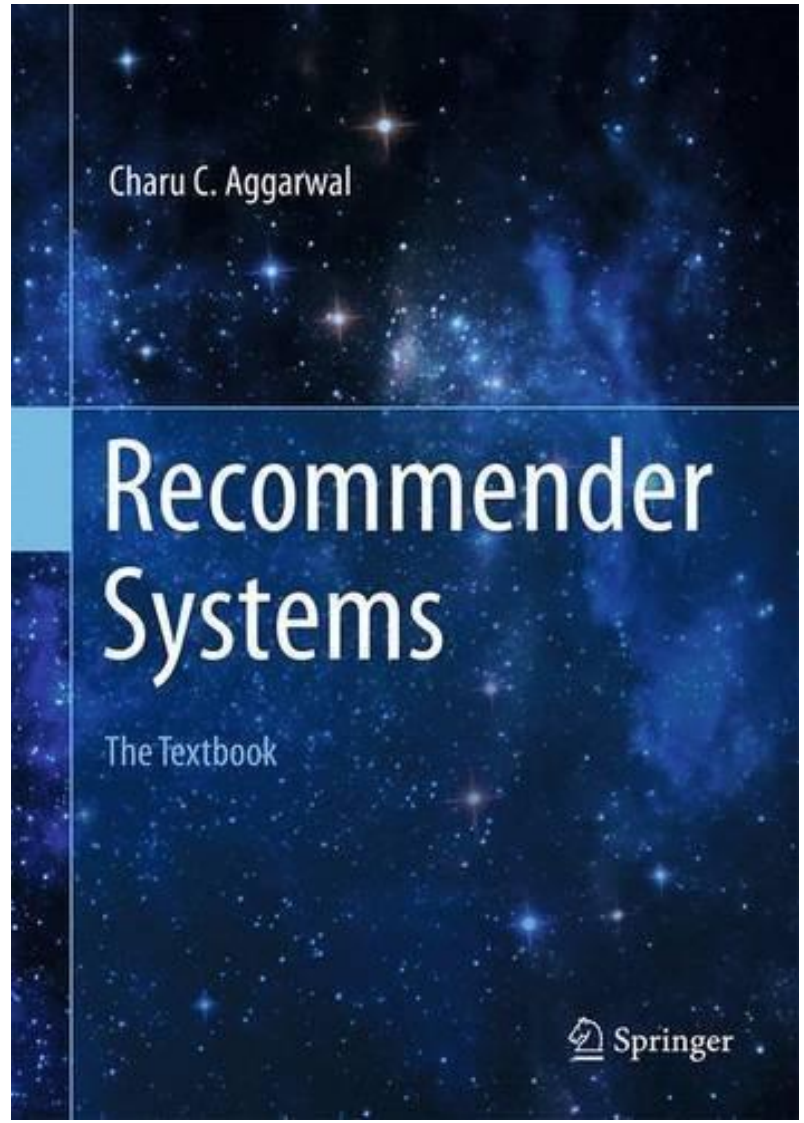

**叶允明**

**计算机科学与技术学院**

**哈尔滨工业大学（深圳）**

# 目 录

- 推荐系统的基本概念
- 协同过滤算法
  - 基于邻居的协同过滤算法（Sec. 2.3）
  - 基于矩阵分解的协同过滤算法 （Sec. 3.6）



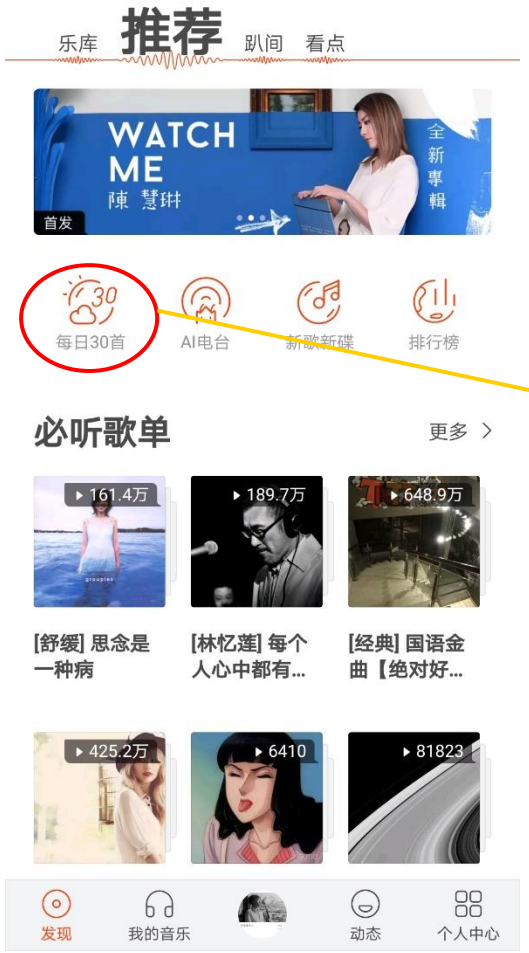
# 推荐系统的基本概念

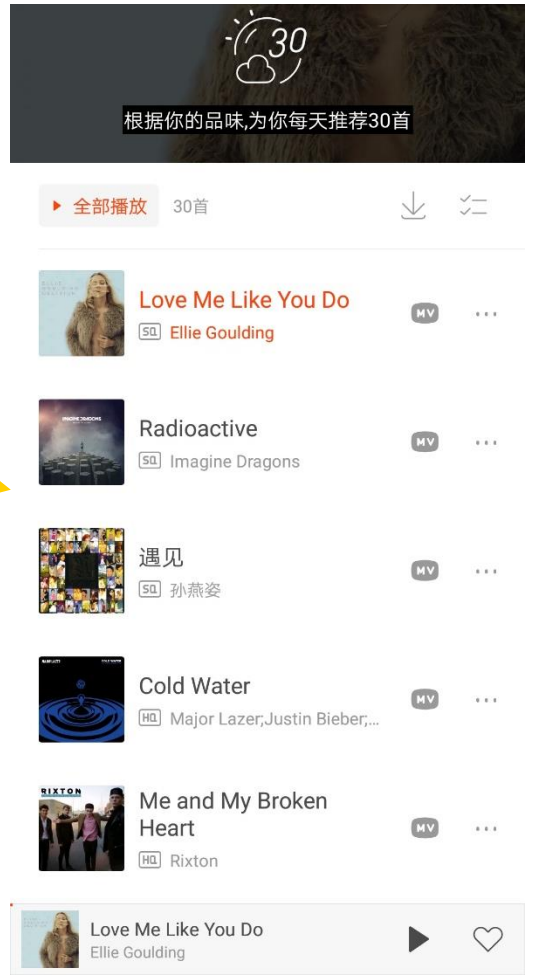
# 推荐系统的典型应用

- **根据浏览记录推荐商品**

# 推荐系统的典型应用

- **根据听歌历史推荐歌单**

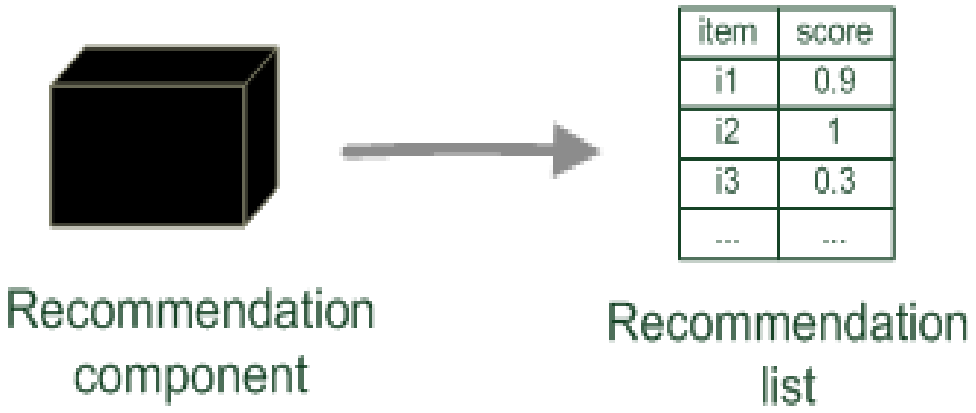


# 推荐系统的典型应用

- **根据浏览历史推荐新闻**

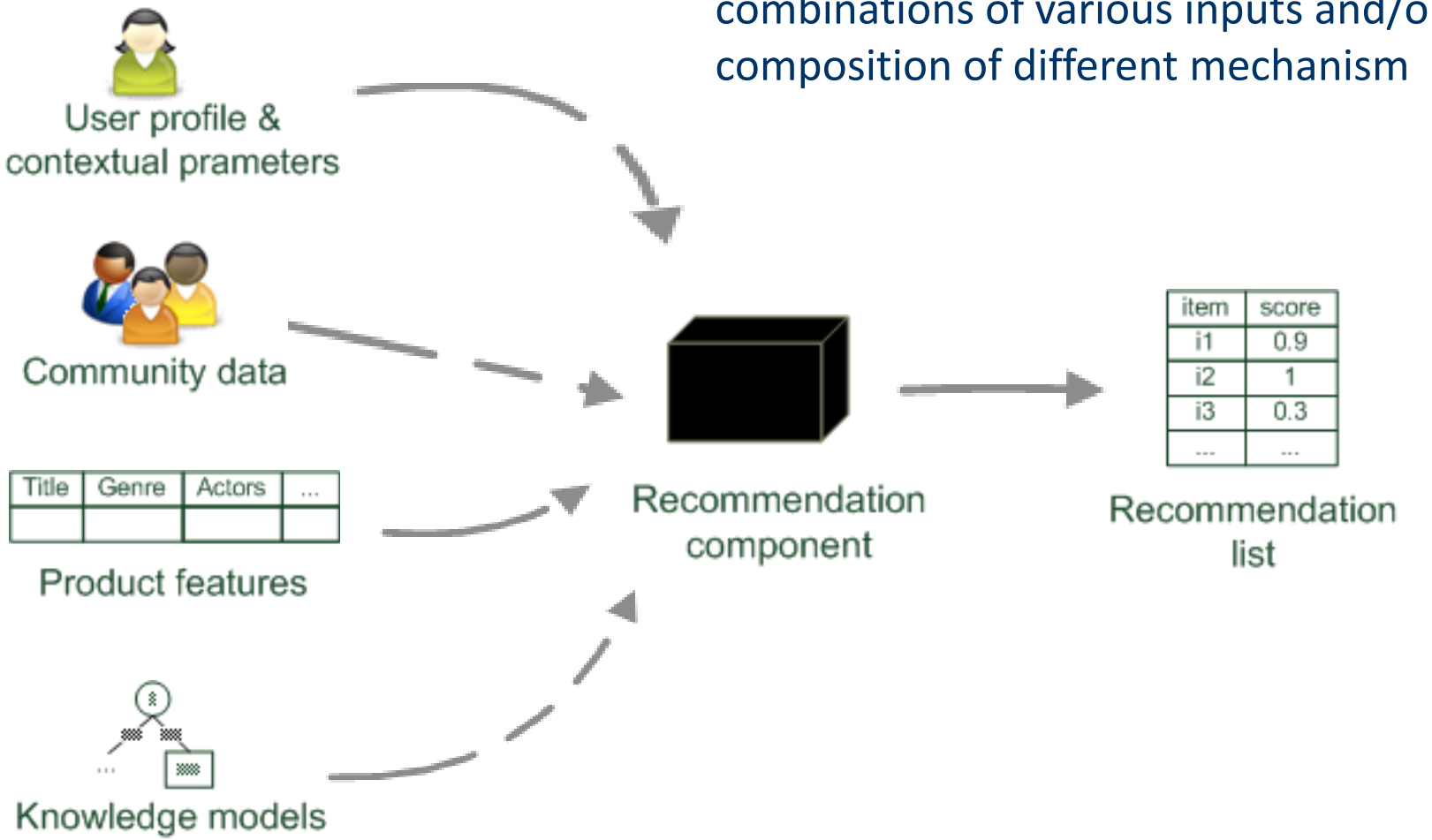
**A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. ——*Wiki***



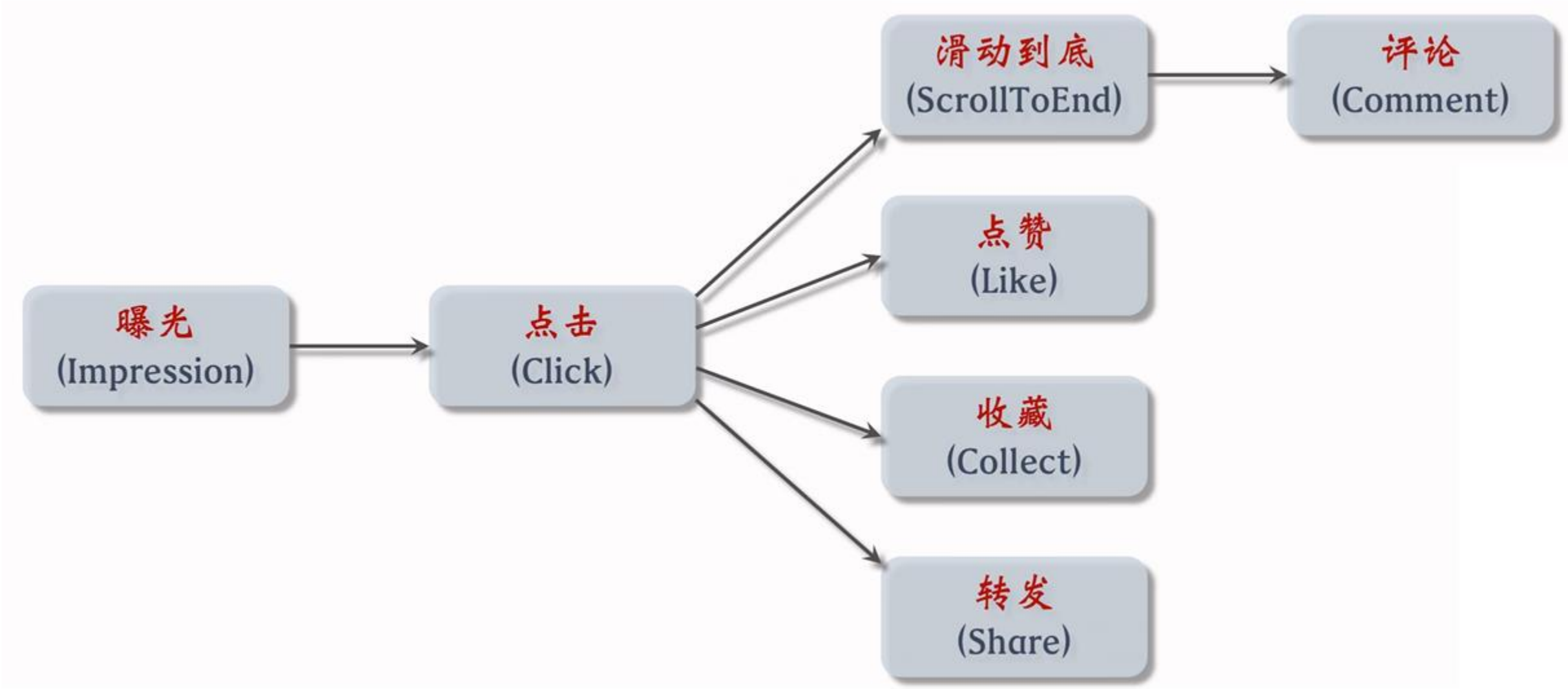

推荐系统：通过相关性估计减少信息过载!

# 推荐系统的多源数据

combinations of various inputs and/or composition of different mechanism



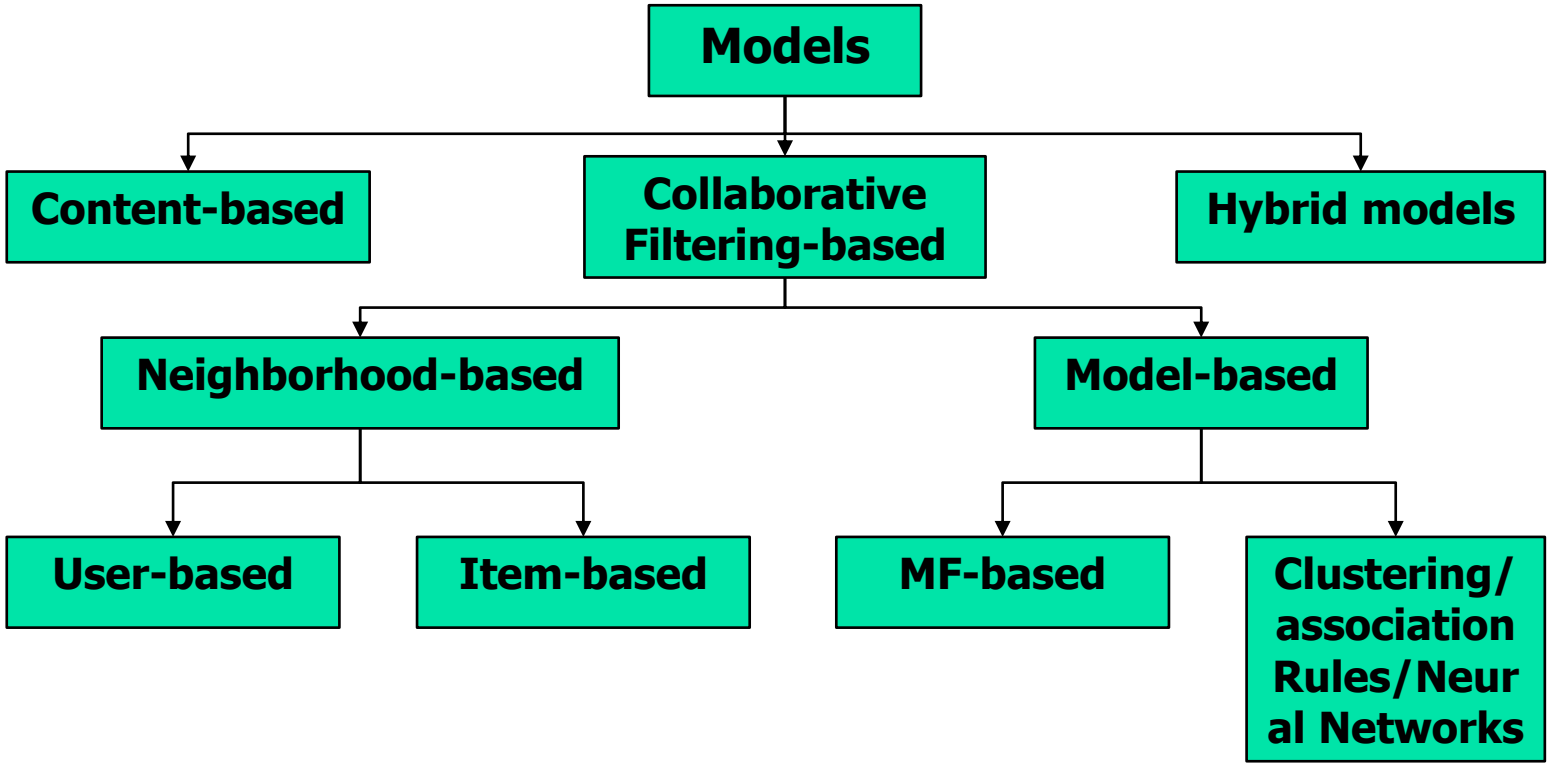
# 常见的推荐系统转化过程



# 常见的推荐系统评价指标

- 点击率 = 点击次数 / 曝光次数
- 点赞率 = 点赞次数 / 点击次数
- 收藏率 = 收藏次数 / 点击次数
- 转发率 = 转发次数 / 点击次数
- 用户规模：
  - 日活用户数（DAU）、月活用户数（MAU）

# 传统推荐模型的主要类别



# 推荐系统的发展历程

| | |
|---|---|
| 2000 | 基于近邻的协同过滤：如KNN Collaborative Filtering |
| 2007-2015 | 矩阵分解（Matrix Factorization），如SVD++ 、FM等 |
| 2015-2017 | 深度学习方法，如FNN, AutoEncoder, CDL, DeepFM等. |
| 2017-Now | 强化学习方法（Reinforcement Learning Methods） |

# 基于邻居的协同过滤

# 什么是协同过滤（Collaborative Filtering, CF)?

| | Book1 | Book2 | Book3 | Book4 | Book5 | Book6 | …... |
|---|---|---|---|---|---|---|---|
| User1 | | | | | | | |
| User2 | | | | | | | |
| User3 | | | | | | | |
| User4 | | | | | | | |
| User5 | | | | | | | |
| User6 | ? | ? | | ? | ? | ? | ? |

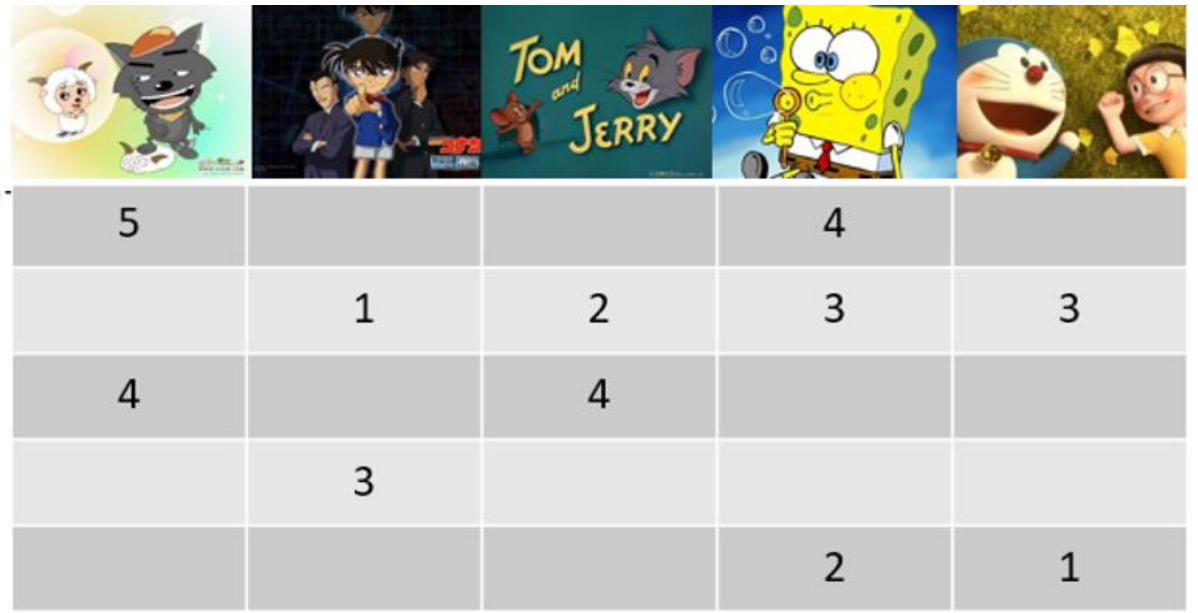
# 基于邻居的协同过滤算法

- **具有广泛应用领域的有效推荐方法**
  - 在网络书店、电影推荐等领域广泛应用
  - 简单有效、且具有良好的可解释性
- **基本假设与思想：**
  - 用户对物品进行显式或隐式的评分（rating）
  - 假设：用户未来的偏好与过去的偏好相似
  - 基于"群体智慧（wisdom of the crowd）"来推荐商品
- **可分为两种算法：**
  - 基于用户的协同过滤（User-based collaborative filtering）
  - 基于物品的协同过滤（Item-based collaborative filtering）

# 评分矩阵（Rating matrix）

| User | Item | Rating |
|---|---|---|
| 1 | 1 | 5 |
| 1 | 4 | 4 |
| … | … | … |
| **u** | **j** | **r** |
| … | … | … |
| … | … | … |

评分矩阵：$R = [r_{uj}]$



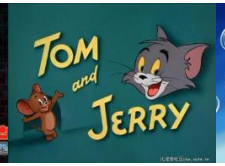

| | | | | |
|---|---|---|---|---|
| | 5 | | | 4 | |
| | | 1 | 2 | 3 | 3 |
| | 4 | | 4 | | |
| | | 3 | | | |
| | | | | 2 | 1 |

# 基于用户的协同过滤算法

| |  |  |  |  |  |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

# 基于用户的协同过滤算法

- **需解决三个问题：**
  - 如何度量相似性？
  - 需要选多少个“邻居”？
  - 如何基于邻居的评分做出预测？

| |  |  |  |  |  |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

# 用户相似性度量方法

- **常用的相似性度量方法：皮尔逊相关系数（Pearson correlation）**

  $u, v$ ：用户

  $r_{uk}$ ：用户 $u$ 对物品 $k$ 的评分

  $I_u$ ：被用户 $u$ 评分过的物品对应的索引号集合

  $\mu_u$ :用户 $u$ 的平均评分（基于其历史评分计算）

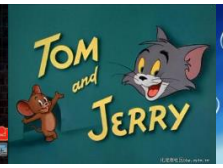
  - 第一步是计算 $\mu_u$ ：

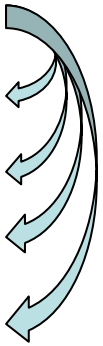
$$\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|} \quad \forall u \in \{1 \dots m\}$$

  - 第二步是计算用户$u$ 和 $v$ 的皮尔逊相关系数：

$$\mathrm{Sim}(u,v) = \mathrm{Pearson}(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

# 用户相似性度量示例

| |  |  |  |  |  |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

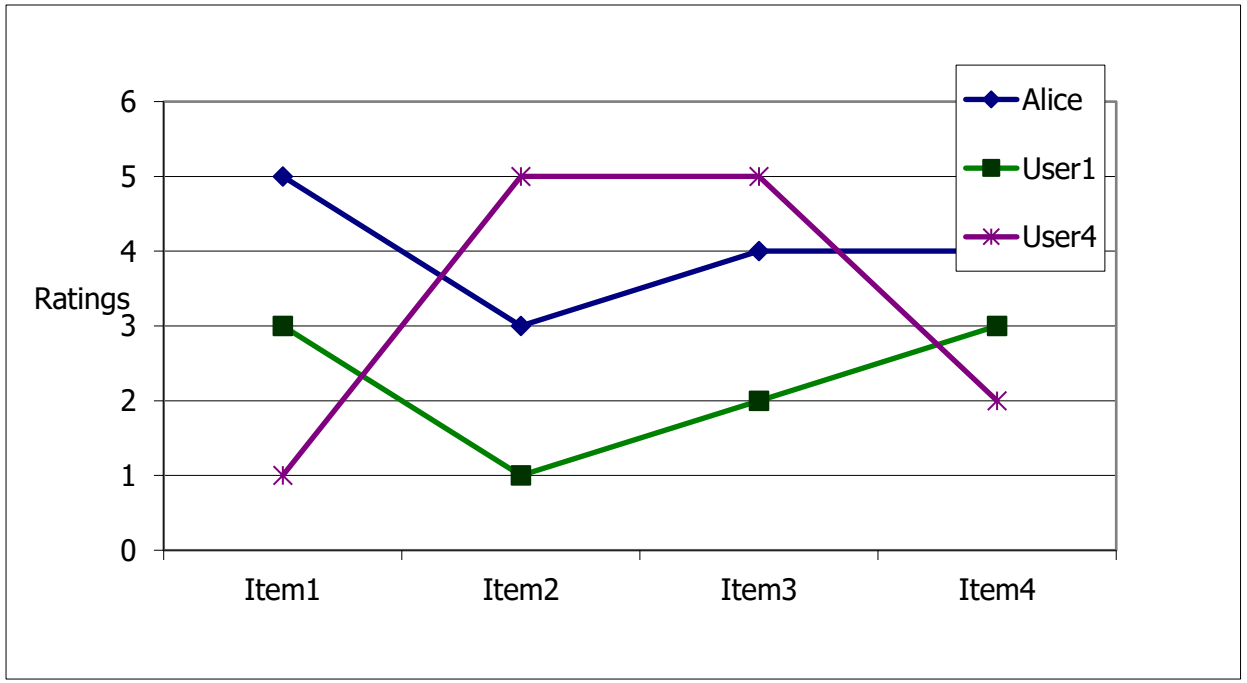


sim = 0.85

sim = 0.00

sim = 0.70

sim = -0.79



# 预测过程

- **第一步：选出 $k$ 个具有最高皮尔逊相关系数的用户作为“邻居”集合**
- **第二步：每个邻居的评分值需进行去均值化（为什么？）：**

$$s_{uj} = r_{uj} - \mu_u \quad \forall u \in \{1 \dots m\}$$

- **第三步：计算预测值**

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot s_{vj}}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|}$$

  - **$P_u(j)$ 表示与用户$u$ 最相似的、且在物品j上有过评分的$k$ 个用户（邻居）集合**

# 预测过程示例

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot s_{vj}}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|}$$


| | | | | | |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

sim = 0.85

sim = 0.00

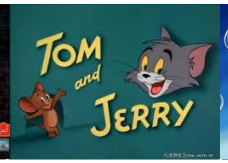
sim = 0.70

sim = -0.79

$$\boldsymbol{pred(Alice, Cartoon5) = 4 + \frac{0.85 * (3 - 2.4) + 0.7 * (4 - 3.2)}{0.85 + 0.7} = 4.69}$$

# 基于物品的协同过滤算法

- 利用物品之间的相似性来做预测

| |  |  |  |  |  |
|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ? |
| User1 | 3 | 1 | 2 | 3 | 3 |
| User2 | 4 | 3 | 4 | 3 | 5 |
| User3 | 3 | 3 | 1 | 5 | 4 |
| User4 | 1 | 5 | 5 | 2 | 1 |

# 物品相似性度量方法

- **物品 $i$ 和 $j$ 的相似性基于在这两个物品上都有评分的用户集合**
  - $U_i$: 在**物品 $i$ 上有评分的用户集合**

$$\text{AdjustedCosine}(i,j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} s_{uj}^2}}$$

  - $S_{ui} = r_{ui} - \mu_u$

# 预测过程

- 选出 $k$ 个具有最高相关系数的物品作为物品 $t$ 的"邻居"集合
- 计算：

$$\hat{r}_{ut} = \frac{\sum_{j \in Q_t(u)} \text{AdjustedCosine}(j,t) \cdot r_{uj}}{\sum_{j \in Q_t(u)} |\text{AdjustedCosine}(j,t)|}$$

  - $Q_t(u)$ 是与商品 t 最相关的 top-$k$ 个商品（且用户 u 有过评分）

$r_{uj}$ 无需再做均值化，为什么?

# 两种方法的特点对比

- **基于物品的推荐算法**
  - 物品的相似性更稳定
  - 推荐的物品更相关, 具有更好的准确度
  - 可能会推荐一些很显然、很普通的物品（如推荐牛奶）
- **基于用户的推荐算法**
  - 稳定性不如前者
  - 推荐的多样性更好, 可能会推荐一些有新意、眼前一亮的商品

# 基于邻居的推荐模型：优缺点分析

- **优点**
  - 简单，容易实现和调优
  - 可解释性好
  - 对增量数据具有比较好的稳定性

**缺点**

  - 推荐阶段需要更多的计算时间（类似于KNN）
  - 容易受数据稀疏性问题（data sparsity problems）的影响

# 数据稀疏性问题

- **冷启动问题（Cold start problem）**
  - 新建设系统中缺少数据积累的问题
  - 新物品、新用户该如何推荐?
- **简单处理方法**
  - 请求或要求用户对一些物品评分
  - 在推荐系统建设的初始阶段采用基于内容的推荐模型
  - 更好的方法……

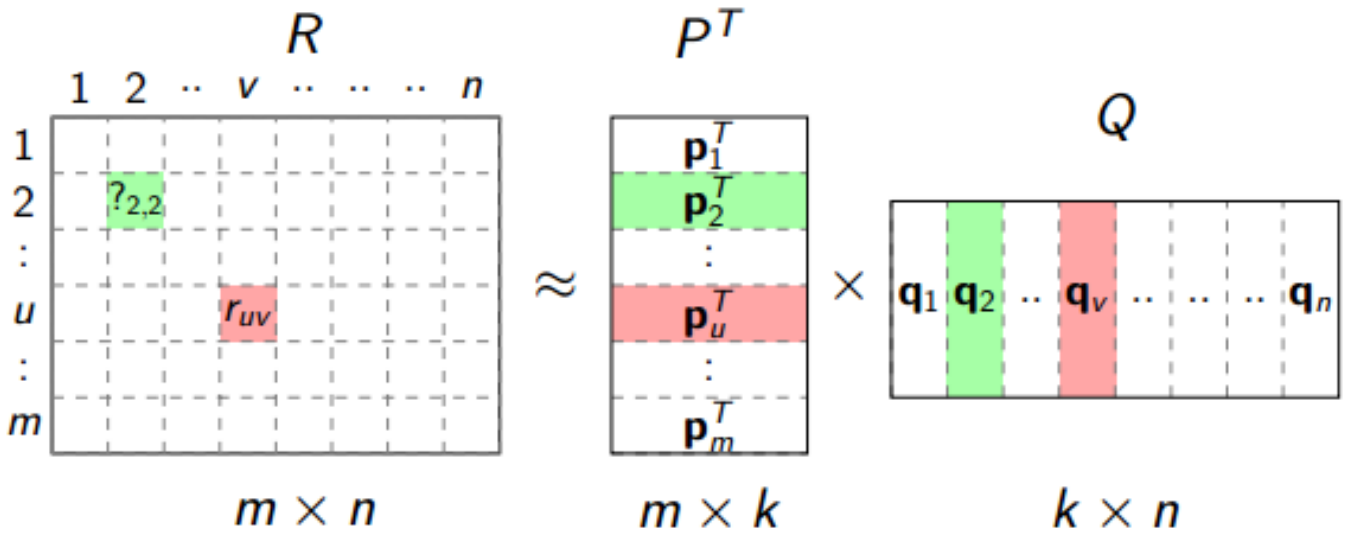
# 基于矩阵分解的协同过滤

# 基本思想

- **潜在因子模型（Latent factor models）**
  - 商品和用户的评分都受到一些共同的 “潜在因子” 的影响
  - 例如：用户-电影的评分矩阵的潜在因子：
    - ✓ 电影可以分为：喜剧片、战争片、烂漫爱情片等
    - ✓ 用户对电影的偏好也可以上面的类别因子来表示
    - ✓ 则以上分类就是电影和用户偏好的共同潜在因子

- **主要方法：基于矩阵分解挖掘潜在因子**
  - 矩阵分解推荐模型曾获得巨大成功：Netflix Prize and KDD Cup 2011

# 矩阵分解




- k: 隐藏因子的子空间维度
- $r_{u,v} = p_u^T q_v$
- $?_{2,2} = p_2^T q_2$

# 基于矩阵分解的推荐模型

| |  |  |  |  |
|---|---|---|---|---|
| A | 5 | 3 | 5.49 | 1 |
| B | 4 | 3 | 4.84 | 1 |
| C | 1 | 1 | 5.19 | 5 |
| D | 1 | 0.70 | 4 | 4 |
| E | 1.59 | 1 | 5 | 4 |

假设有两个潜在因子

| | | |
|---|---|---|
| $\boldsymbol{p}_A$ | 2.38 | 0.40 |
| $\boldsymbol{p}_B$ | 2.04 | 0.41 |
| $\boldsymbol{p}_C$ | 0.32 | 2.19 |
| $\boldsymbol{p}_D$ | 0.27 | 1.72 |
| $\boldsymbol{p}_E$ | 0.62 | 1.78 |

| | | |
|---|---|---|
| $\boldsymbol{q}_1$ | 1.99 | 0.21 |
| $\boldsymbol{q}_2$ | 1.31 | 0.20 |
| $\boldsymbol{q}_3$ | 1.96 | 2.08 |
| $\boldsymbol{q}_4$ | 0.03 | 2.27 |

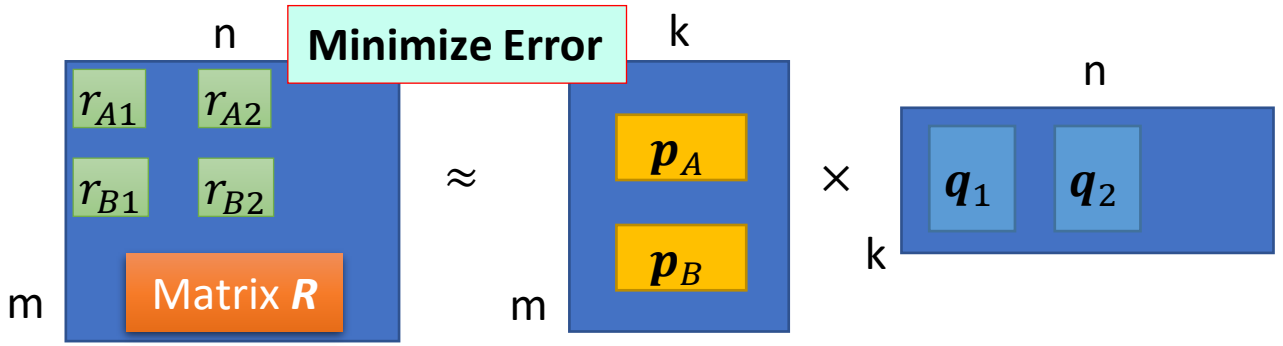
# 基于矩阵分解的推荐模型

No. of User = m


| | | | | |
|---|---|---|---|---|
| A | 5 | 3 | 0 | 1 |
| B | 4 | 3 | 0 | 1 |
| C | 1 | Matrix ***R*** | | 5 |
| D | 1 | 0 | 4 | 4 |
| E | 0 | 1 | 5 | 4 |

No. of Cartoon = n

No. of latent factor = k




$$L = \frac{1}{2}||\boldsymbol{R} - \boldsymbol{P}^T\boldsymbol{Q}||^2$$

$$= \frac{1}{2}\sum_{(u,v)\in\boldsymbol{R}} (r_{uv} - \boldsymbol{p}_u^T \cdot \boldsymbol{q}_v)^2$$

用SVD(Singular value decomposition )求解?

# 基于随机梯度下降的学习算法

- 目标函数:

$$\boldsymbol{P}^*, \boldsymbol{Q}^* = \arg\min_{\boldsymbol{P},\boldsymbol{Q}} L(\boldsymbol{P}, \boldsymbol{Q}) = \arg\min_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{2} ||\boldsymbol{R} - \boldsymbol{P}^T \boldsymbol{Q}||^2$$

$$= \arg\min_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{2} \sum_{(u,v)\in \boldsymbol{S}} (r_{uv} - \boldsymbol{p}_{\mathrm{u}}^T \cdot \boldsymbol{q}_v)^2$$

- 加入正则化:

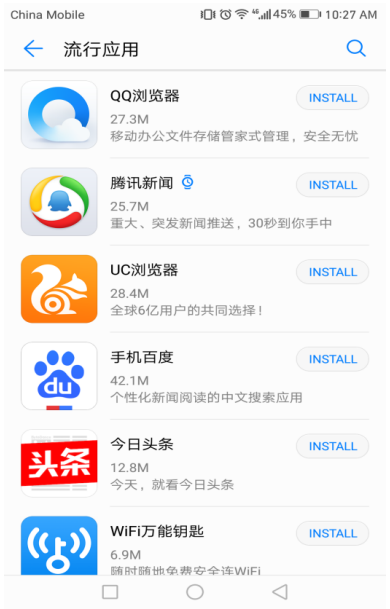
$$\boldsymbol{P}^*, \boldsymbol{Q}^* = \arg\min_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{2} \sum_{(u,v)\in \boldsymbol{S}} ((r_{uv} - \boldsymbol{p}_{\mathrm{u}}^T \cdot \boldsymbol{q}_v)^2 + \lambda_p ||\boldsymbol{p}_{\mathrm{u}}||^2 + \lambda_q ||\boldsymbol{q}_v||^2)$$

$$\arg\min_{\boldsymbol{P},\boldsymbol{Q}} \frac{1}{2} \sum_{(u,v)\in \boldsymbol{S}} ((r_{uv} - \boldsymbol{p}_{\mathrm{u}}^{T} \cdot \boldsymbol{q}_{v})^2 + \lambda_p ||\boldsymbol{p}_{\mathrm{u}}||^2 + \lambda_q ||\boldsymbol{q}_{v}||^2)$$

- 学习算法步骤
  - 随机初始化$\boldsymbol{P}, \boldsymbol{Q}$
  - do
    - ✓ For 每个训练样本（$r_{uv}$，$(u,v) \in \boldsymbol{S}$）do:
      1. $e_{\mathrm{uv}} = \mathrm{r}_{\mathrm{uv}} - \mathbf{p}_{\mathrm{u}}^{\mathrm{T}} \cdot \mathbf{q}_{\mathrm{v}}$
      2. $\boldsymbol{q}_{v} \leftarrow \boldsymbol{q}_{v} + \eta . (e_{\mathrm{uv}} . \mathbf{p}_{\mathrm{u}}^{\mathrm{T}} - \lambda_q . \boldsymbol{q}_{v})$
      3. $\mathbf{p}_{\mathrm{u}} \leftarrow \mathbf{p}_{\mathrm{u}} + \eta . (e_{\mathrm{uv}} . \boldsymbol{q}_{\mathrm{v}}^{\mathrm{T}} - \lambda_p . \mathbf{p}_{\mathrm{u}})$
  - Until stopping criteria satisfied

# 点击率预测（Click-Through Rate）问题

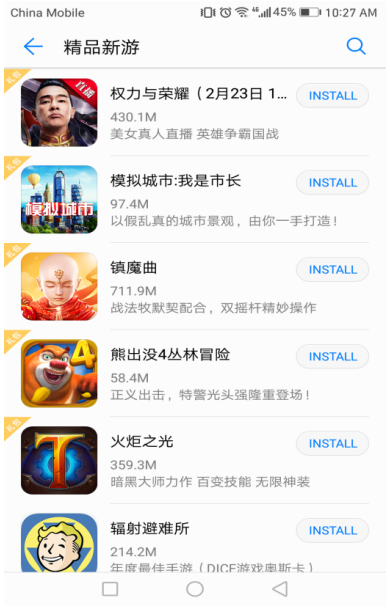
## Normal App




**Ranked by CTR**

**CTR**
**Click Through Rate**

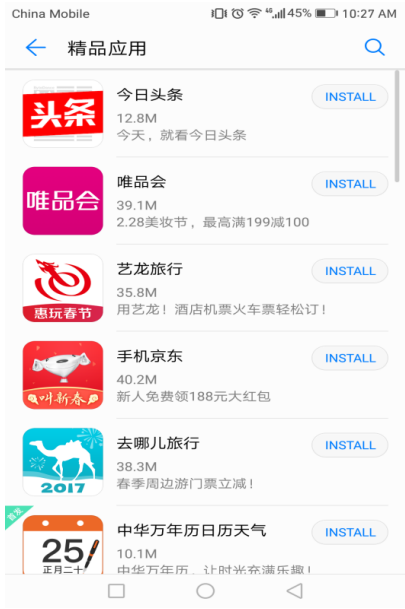
## Game App




**Ranked by CTR ×LTV**

**LTV**
**Life Time Value**

## Advertise App

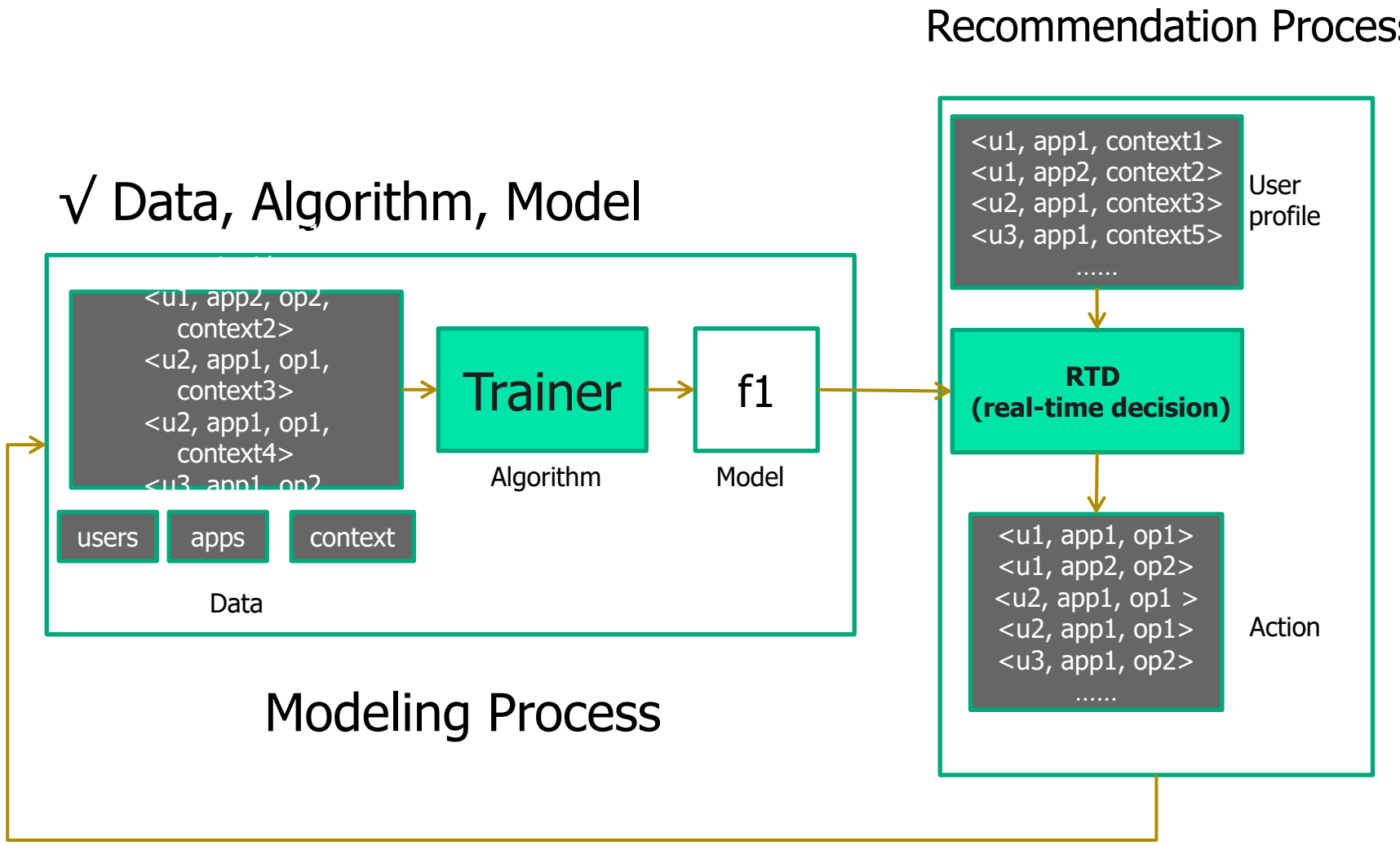



**Ranked by CTR ×CPC**

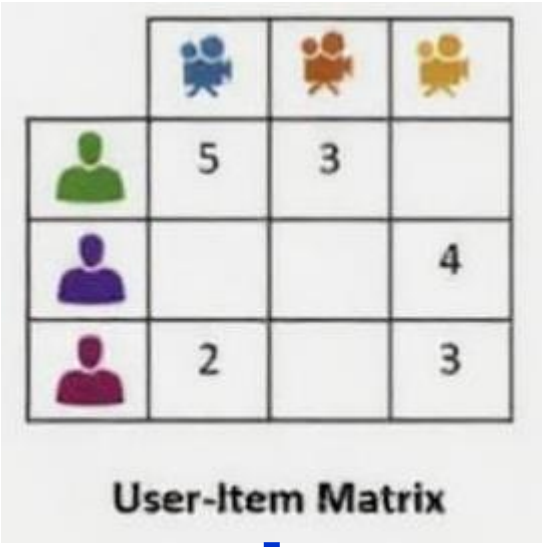
**CPC**
**Cost Per Click**

**CTR is critic in recommender system, not only affect the user experience, but also determine the benefits.**

# 点击率预测（Click-Through Rate）问题



# 从用户-物品矩阵到通用多特征矩阵的挑战




| | Feature vector **x** | | | | | | | | | | | | | | | | | | | | Target y | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | … | TI | NH | SW | ST | … | TI | NH | SW | ST | … | Time | TI | NH | SW | ST | … | | |
| | User | | | | Movie | | | | | Other Movies rated | | | | | | Last Movie rated | | | | | | |
| $x^{(1)}$ | 1 | 0 | 0 | … | 1 | 0 | 0 | 0 | … | 0.3 | 0.3 | 0.3 | 0 | … | 13 | 0 | 0 | 0 | 0 | … | 5 | $y^{(1)}$ |
| $x^{(2)}$ | 1 | 0 | 0 | … | 0 | 1 | 0 | 0 | … | 0.3 | 0.3 | 0.3 | 0 | … | 14 | 1 | 0 | 0 | 0 | … | 3 | $y^{(2)}$ |
| $x^{(3)}$ | 1 | 0 | 0 | … | 0 | 0 | 1 | 0 | … | 0.3 | 0.3 | 0.3 | 0 | … | 16 | 0 | 1 | 0 | 0 | … | 1 | $y^{(2)}$ |
| $x^{(4)}$ | 0 | 1 | 0 | … | 0 | 0 | 1 | 0 | … | 0 | 0 | 0.5 | 0.5 | … | 5 | 0 | 0 | 0 | 0 | … | 4 | $y^{(3)}$ |
| $x^{(5)}$ | 0 | 1 | 0 | … | 0 | 0 | 0 | 1 | … | 0 | 0 | 0.5 | 0.5 | … | 8 | 0 | 0 | 1 | 0 | … | 5 | $y^{(4)}$ |
| $x^{(6)}$ | 0 | 0 | 1 | … | 1 | 0 | 0 | 0 | … | 0.5 | 0 | 0.5 | 0 | … | 9 | 0 | 0 | 0 | 0 | … | 1 | $y^{(5)}$ |
| $x^{(7)}$ | 0 | 0 | 1 | … | 0 | 0 | 1 | 0 | … | 0.5 | 0 | 0.5 | 0 | … | 12 | 1 | 0 | 0 | 0 | … | 5 | $y^{(6)}$ |

# 基于线性回归的CTR模型

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

$$y = w_0 + \sum_{i=1}^{n} w_ix_i$$

**无法挖掘特征之间的非线性关系！**

# 二阶多项式回归模型

| | $x_1$ | $x_2$ | $\cdots$ | $x_{n-1}$ | $x_n$ |
|---|---|---|---|---|---|
| $x_1$ | $x_1x_1$ | $x_1x_2$ | $\cdots$ | $x_1x_{n-1}$ | $x_1x_n$ |
| $x_2$ | $x_2x_1$ | $x_2x_2$ | $\cdots$ | $x_2x_{n-1}$ | $x_2x_n$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $x_{n-1}$ | $x_{n-1}x_1$ | $x_{n-1}x_2$ | $\cdots$ | $x_{n-1}x_{n-1}$ | $x_{n-1}x_n$ |
| $x_n$ | $x_nx_1$ | $x_nx_2$ | $\cdots$ | $x_nx_{n-1}$ | $x_nx_n$ |

**挖掘特征之间的二阶非线性关系**

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j\geq i}^{n} w_{ij} x_i x_j$$

**存在问题：**

**样本中如果没有出现（$x_i$ 与$x_j$）交互的特征组合，则无法对相应参数（$w_{ij}$）进行估计**

# 因子分解机(Factorization Machines, FM)模型

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j \geq i}^{n} w_{ij} x_i x_j$$

**二项式参数 $w_{ij}$ 可以组成一个矩阵 $W$**

根据 Cholesky 分解，则可以分解成

$$W = VV^T$$

$V$ 的第 $j$ 列便是第 $j$ 维特征的**隐向量**。即每个参数

$$w_{ij} = < V_i, V_j >$$

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j$$

# 因子分解机(Factorization Machines, FM)模型

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j \qquad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \qquad (2)$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size $k$:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

# 因子分解机(Factorization Machines, FM)模型

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_i \rangle\, x_i\, x_i$$

$$= \frac{1}{2} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n} \sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i \right)$$

$$= \frac{1}{2} \sum_{f=1}^{k} \left( \left( \sum_{i=1}^{n} v_{i,f}\, x_i \right) \left( \sum_{j=1}^{n} v_{j,f}\, x_j \right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right)$$

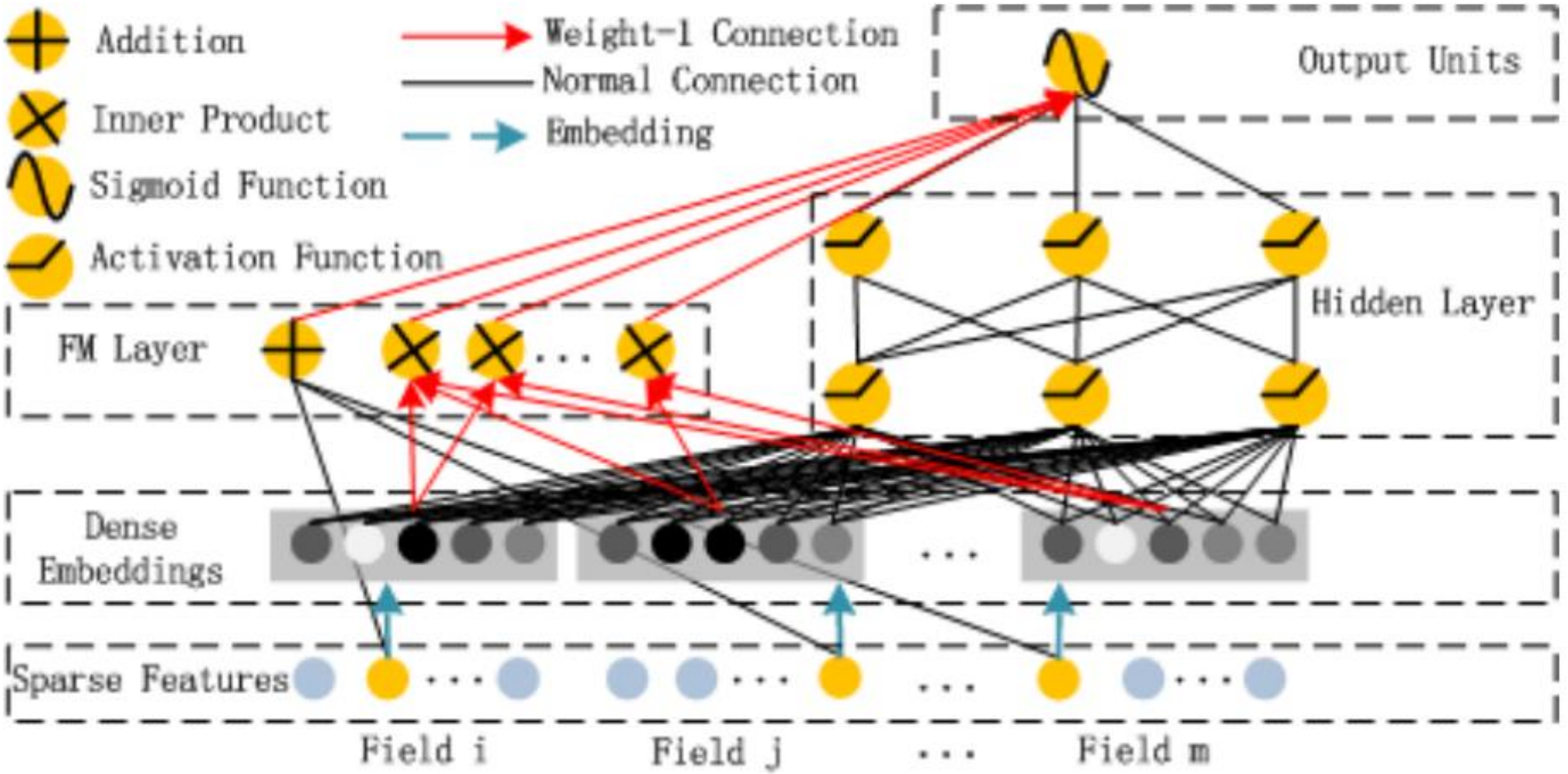
$$= \frac{1}{2} \sum_{f=1}^{k} \left( \left( \sum_{i=1}^{n} v_{i,f}\, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right)$$

# 因子分解机(Factorization Machines, FM)模型

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \frac{1}{2} \sum_{f=1}^{k} \left( \left( \sum_{i=1}^{n} v_{i,f} \, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 \, x_i^2 \right)$$

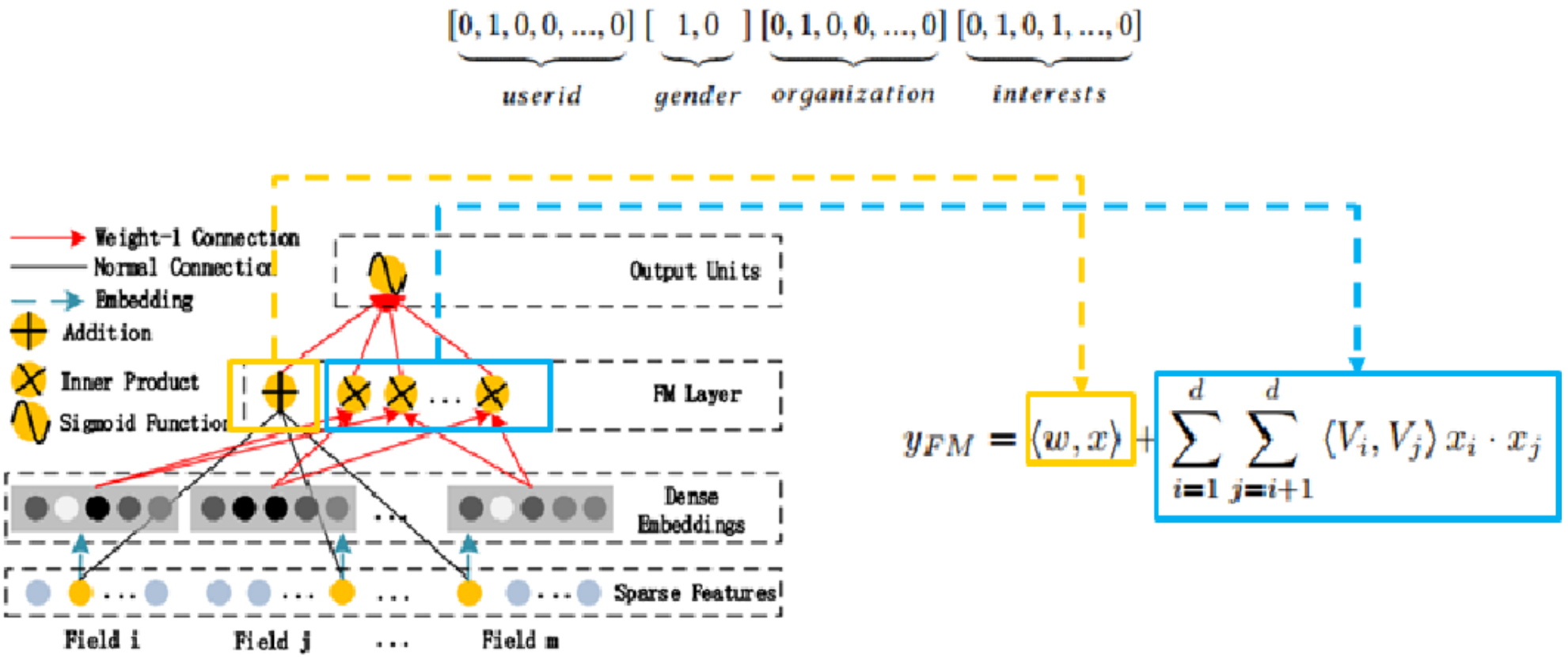
$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

# DeepFM推荐模型


Addition
Inner Product
Sigmoid Function
Activation Function
Weight-1 Connection
Normal Connection
Embedding
Output Units
Hidden Layer
FM Layer
Dense Embeddings
Sparse Features
Field i
Field j
Field m

# DeepFM推荐模型

- FM-Component
  - where w ∈ $R^d$ and $V_i$ ∈ $R^k$ (k is dimension of latent vector which is given) . The Addition unit (<w, x>) reflects the importance of order-1 features, and the Inner Product units represent the impact of order-2 feature interactions.



# Acknowledgements


- Some text, figures and formulations are from WWW. Thanks for their sharing. If you have copyright claim please contact with me at yym@hit.edu.cn.
- This lecture is distributed for nonprofit purpose.

# Thank You for Your Attention

Contact me at: yym@hit.edu.cn

Tel: 26033008, 1376019623

Address: Rm.1402, H# Building