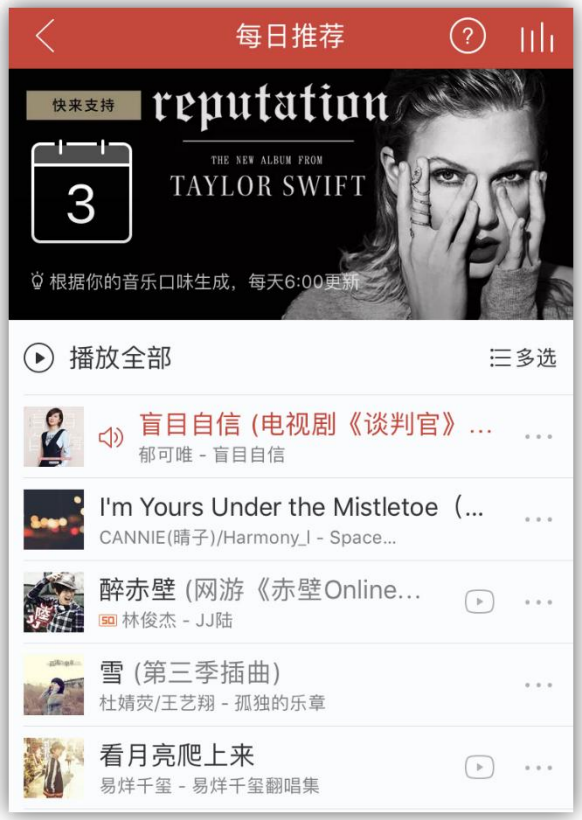
上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

数据挖掘与商务分析

推荐系统与商务实践

主讲教师：肖升生
xiao.shengsheng@shufe.edu.cn

课程导入：推荐系统无处不在



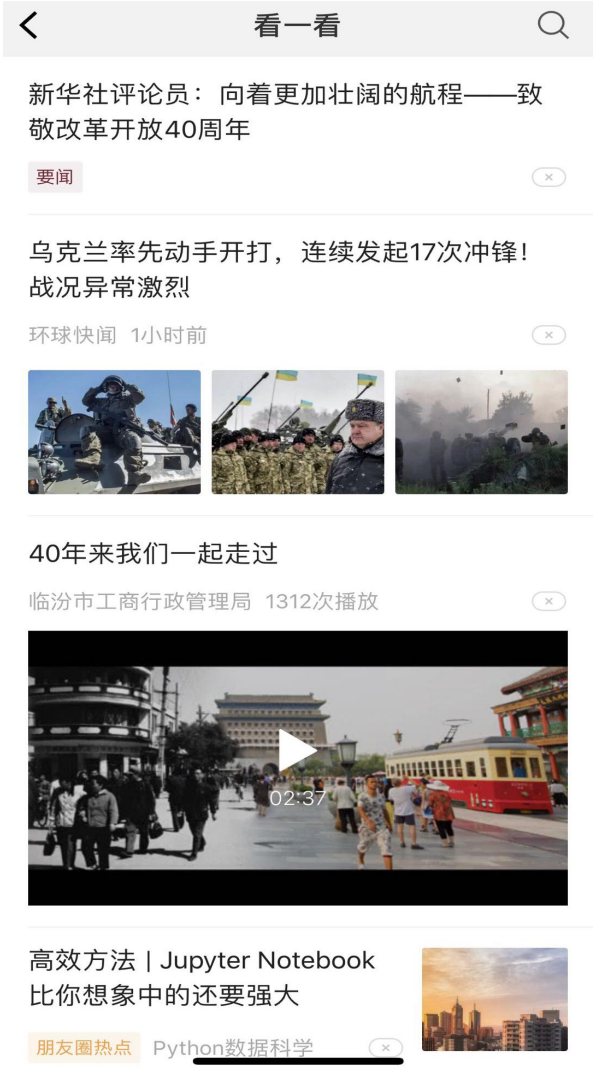
天猫 TMALL.COM

网易云音乐 听见·好时光！

Baidu 新闻

“个性化”推荐

时尚穿搭+心灵鸡汤



军事时政+学术科研

讲授提纲

01 推荐系统基本概念

02 基于内容的推荐

03 基于近邻的协同过滤

04 基于矩阵分解的协同过滤

05 商务案例-电影推荐

讲授提纲

什么是推荐系统？

- 推荐系统是一类信息过滤工具，通过预测用户对产品的评分或偏好，为每个用户生成个性化的推荐列表。

- 多种多样的推荐系统
  - 商品推荐：电商平台的商品、应用商店的移动应用
  - 内容推荐：新闻、博文、音乐、电影推荐等
  - 好友推荐：基于社交网络

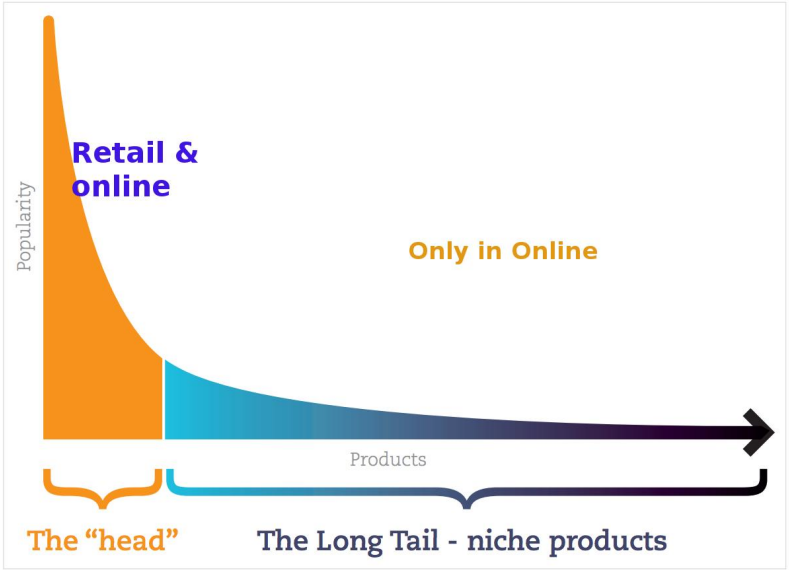
推荐系统 v.s. 搜索引擎

- 二者功能紧密关联，均为信息过滤工具。
- 不同的应用场景，对应不同的目标。
  - 搜索引擎：用户明确搜索目标，反映为查询关键词；
  - 推荐系统：用户无明确搜索目标，需要个性化推荐。

搜索引擎，就是根据**用户需求**，运用特定信息检索技术，从互联网检索出与用户需求相关的特定信息资源。

推荐系统的作用

- 从平台提供者的角度
  - 提高销售额；
  - 提高商品销售的多样性(长尾效应)；
  - 增加用户满意度和用户粘性。
- 从用户的角度
  - 降低搜索成本；
  - 浏览推荐列表本身就是一种娱乐



推荐系统架构

用户信息
（年龄、性别、职业、标签…）

商品信息
（名称、类型、关键词、图片…）

交互信息
（评分、购买、浏览、收藏…）

推荐系统

预测用户对商品的偏好打分

商品集

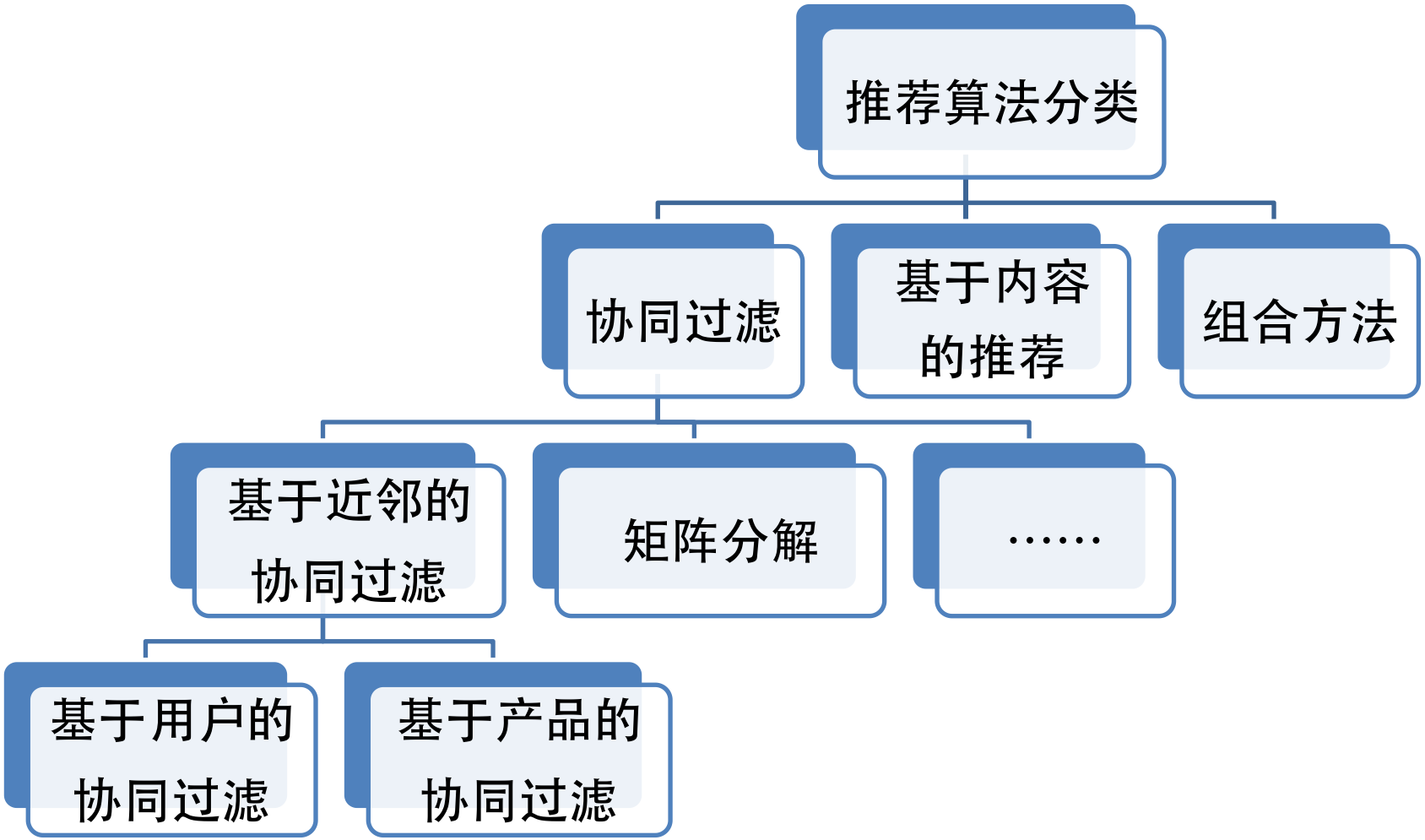
用户集

推荐问题定义

给定用户集合U和产品集合I，推荐算法旨在基于历史评分数据学习用户$u \in U$对于<u>未评分</u>产品$i \in I$的**<u>偏好分值</u>**：

$$f(u,i) \rightarrow r_{ui}$$

推荐算法体系


推荐算法分类
协同过滤
基于内容的推荐
组合方法
基于近邻的协同过滤
矩阵分解
……
基于用户的协同过滤
基于产品的协同过滤

推荐算法评估

离线实验

- 基于用户行为日志收集离线数据集；
- 将数据集划分为训练集和测试集（随机划分、根据时间先后划分等）；
- 在训练集上训练推荐模型，在测试集上评估模型的推荐效果。

用户测试

- 确保测试用户和线上用户服从相同分布；
- 容易获得反映用户主观感受的指标，如满意度、惊喜度等。

线上实验

- 利用AB实验比较不同推荐算法的点击率。

离线实验：推荐准确性

■ 评分预测问题

- 对于有真实评分的测试数据集T，评估推荐算法的预测评分$(\hat{r}_{ui})$与用户真实评分(r_{ui})的差异。

平均绝对误差：

$$MAE = \frac{1}{|T|}\sum_{u,i\in T}|r_{ui} - \hat{r}_{ui}|$$

均方根误差：

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{u,i\in T}(r_{ui} - \hat{r}_{ui})^2}$$

离线实验：推荐准确性

Top-N 推荐问题

- 评估推荐算法生成的推荐列表匹配顾客偏好的程度。

$R(u)$ 表示推荐给用户u的产品集合，

$T(u)$ 表示测试集中用户真实选择的产品集合。

召回度：

$$Recall = \frac{1}{|U|}\sum_{u \in U}\frac{|T(u) \cap R(u)|}{|T(u)|}$$

精度：

$$Precision = \frac{1}{|U|}\sum_{u \in U}\frac{|T(u) \cap R(u)|}{|R(u)|}$$

其他评估维度

- 多样性(Diversity)
- 覆盖度(Coverage)
- 新颖度(Novelty)
- 惊喜度(Serendipity)
- …

Vargas, S. and Castells, P., 2011, October. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 109-116). ACM.

Ge, M., Delgado-Battenfeld, C. and Jannach, D., 2010, September. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 257-260). ACM.

小结

- 推荐系统概念
- 推荐问题定义
- 推荐系统的作用
  - 从平台角度、用户角度
- 推荐算法体系
  - 基于内容的推荐
  - 协同过滤：基于近邻的、基于矩阵分解的……
- 推荐算法评估
  - 离线实验、用户测试、线上实验
  - 准确性评估指标

讲授提纲

01 推荐系统基本概念

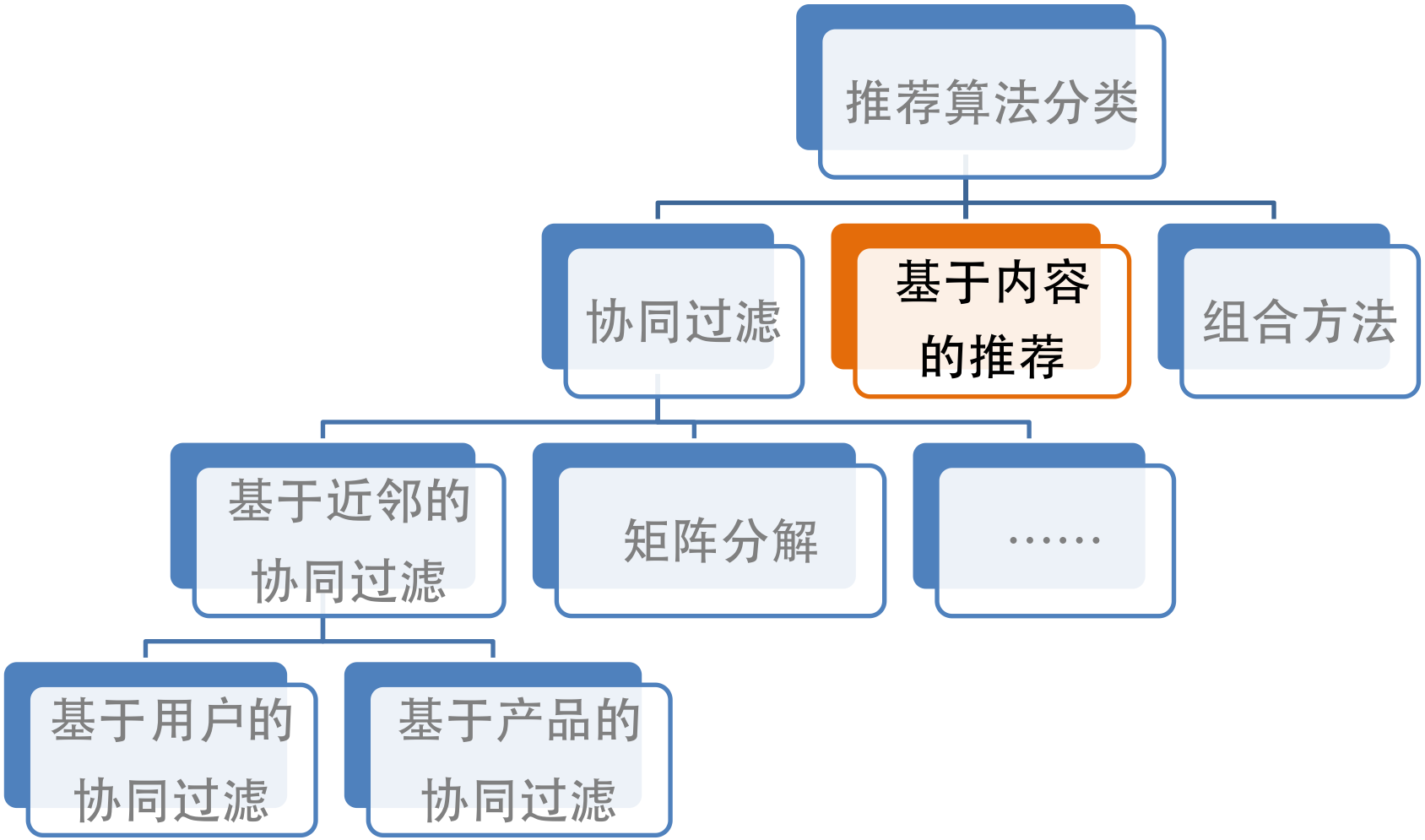
02 基于内容的推荐

03 基于近邻的协同过滤

04 基于矩阵分解的协同过滤

05 商务案例-电影推荐

推荐算法体系（回顾）


推荐算法分类
协同过滤
基于内容的推荐
组合方法
基于近邻的协同过滤
矩阵分解
……
基于用户的协同过滤
基于产品的协同过滤

基于内容的推荐

- 基于内容的推荐（CBR）通过产品的内容描述产品，为用户推荐与喜欢的产品相似的其他产品。
  - 产品内容信息：如名称、类型、标签、图片、用户评论等。
  - 起源于信息检索领域。
  - 最早应用于工业实践的推荐算法；
  - 至今仍广泛应用于新闻推荐领域，例如今日头条；

实现CBR的三个步骤

- **构建产品档案**
  - 基于产品的内容特征描述产品；
- **构建用户档案**
  - 启发式方法，汇总用户喜欢的产品档案；
  - 基于机器学习的分类算法，学习用户偏好模型；
- **生成产品推荐**
  - 推荐与用户档案匹配的产品；

步骤1: 构建产品档案

- 以电影推荐为例，提取描述电影的各项信息：

电影名	类型	导演	主演
流浪地球	科幻、灾难、冒险	郭帆	吴京、屈楚萧、…
战狼	动作、战争、军事	吴京	吴京，斯科特·阿金斯，…
星际穿越	科幻、冒险	诺兰	马修·麦康纳，安妮·海瑟薇，…

- 利用空间向量模型，转化电影类型这个字段：

电影名	科幻	灾难	冒险	动作	战争	军事	…
流浪地球	1	1	1	0	0	0	…
战狼	0	0	0	1	1	1	…
星际穿越	1	0	1	0	0	0	…

步骤2: 构建用户档案

- 利用启发式方法，将用户档案表示为用户喜欢的产品档案的（加权）平均；
  - 给定某用户喜欢“流浪地球”和“星际穿越”这两部电影，其用户档案可以表示为这两部电影档案的平均：

$$\left(\frac{1+1}{2}, \frac{1+0}{2}, \frac{1+1}{2}, 0, 0, 0, \ldots\right) =$$

$$(1, 0.5, 1, 0, 0, 0, \ldots)$$

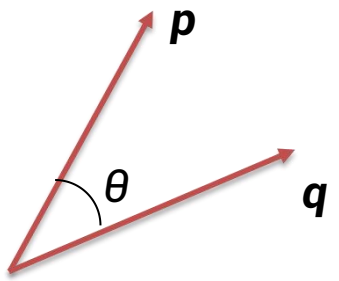
电影名	科幻	灾难	冒险	动作	战争	军事	…
流浪地球	1	1	1	0	0	0	…
战狼	0	0	0	1	1	1	…
星际穿越	1	0	1	0	0	0	…

步骤3: 生成产品推荐（1/2）

- 基于内容的推荐，旨在推荐与用户档案相匹配的产品。
  - 计算用户档案和产品档案之间的匹配度；
  - 采用余弦相似度衡量

给定向量$\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$ 和 $\boldsymbol{q} = (q_1, q_2, \ldots, q_n)$

$$cosine(\boldsymbol{p}, \boldsymbol{q}) = cosine(\theta) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \times \sqrt{\sum_{i=1}^{n} q_i^2}}$$



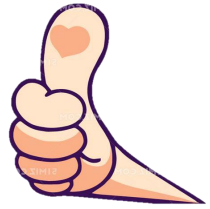
步骤3: 生成产品推荐（2/2）

- 给定用户档案u = (1, 0.5, 1, 0, 0, 0, ...), 推荐下述哪部电影?

电影名	科幻	灾难	冒险	动作	战争	军事	…
战狼	0	0	0	1	1	1	…
盗梦空间	1	0	0	0	0	0	…

$$cosine(u, \text{战狼}) = 0$$

$$cosine(u, \text{盗梦空间}) = \frac{1}{\sqrt{1+\frac{1}{4}+1\times\sqrt{1}}} = \frac{2}{3}$$



拓展1: 引入文本信息构建产品档案

- 文本类型
  - 新闻
  - 产品描述
  - 产品评论
- 文本挖掘技术
  - 分词
  - TF-IDF、LDA等

拓展2: 基于分类算法构建用户档案

- 基于分类算法，学习用户偏好模型；
- 分类的训练数据包括特征向量***x***和类标签*y*：
  - ***x***就是产品档案向量，*y*指的是用户对于电影的正负反馈信息，$y=1$表示喜欢，$y=0$表示不喜欢。
  - 分类算法：决策树、支持向量机、朴素贝叶斯等。

电影名	类型	导演	主演	*y*
流浪地球	科幻、灾难、冒险	郭帆	吴京、屈楚萧、…	1
战狼	动作、战争、军事	吴京	吴京，斯科特·阿金斯，…	0
星际穿越	科幻、冒险	诺兰	马修·麦康纳，安妮·海瑟薇，…	1

讲授提纲

01 推荐系统基本概念

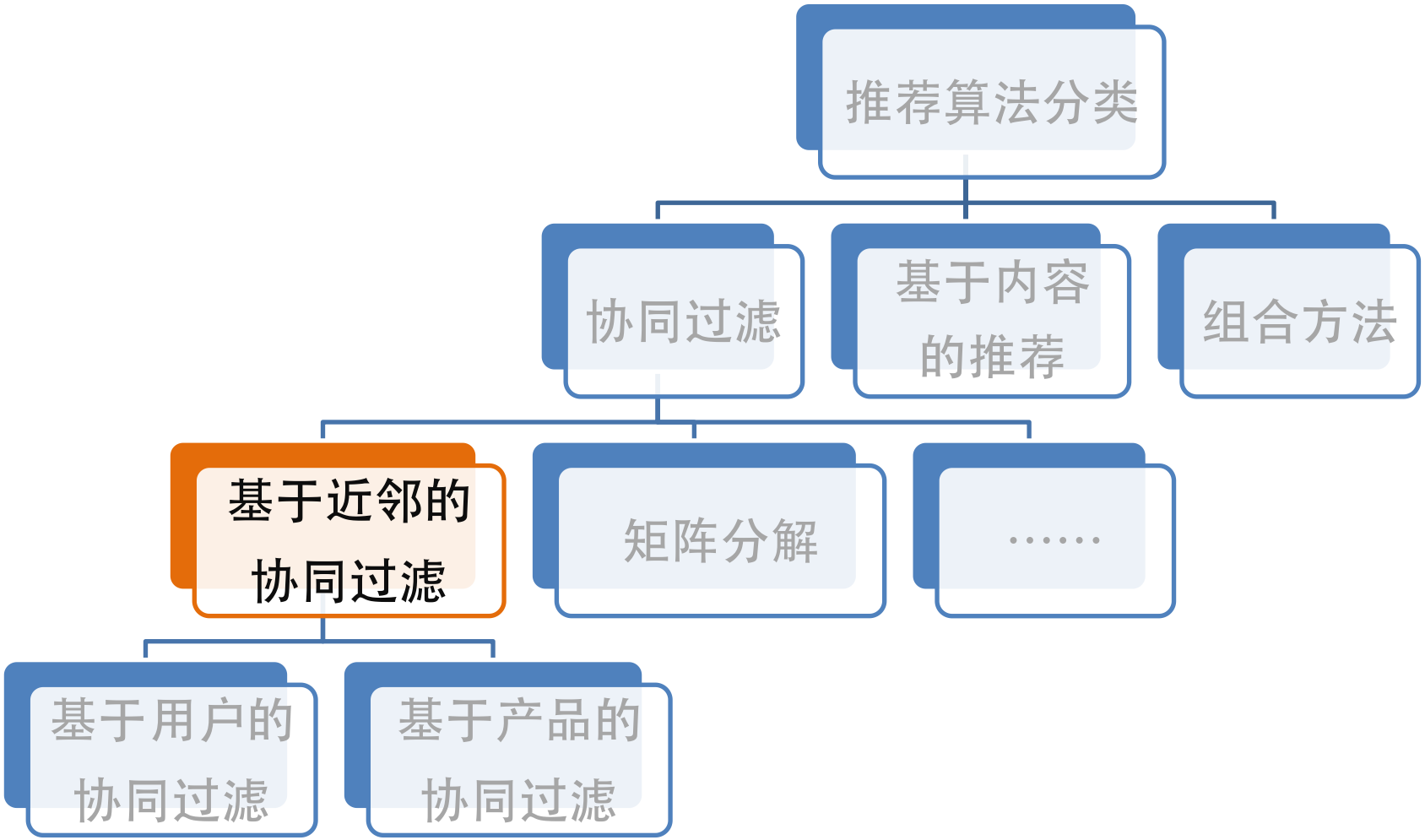
02 基于内容的推荐

03 基于近邻的协同过滤

04 基于矩阵分解的协同过滤

05 商务案例-电影推荐

推荐算法体系（回顾）


推荐算法分类
协同过滤
基于内容的推荐
组合方法
基于近邻的协同过滤
矩阵分解
……
基于用户的协同过滤
基于产品的协同过滤

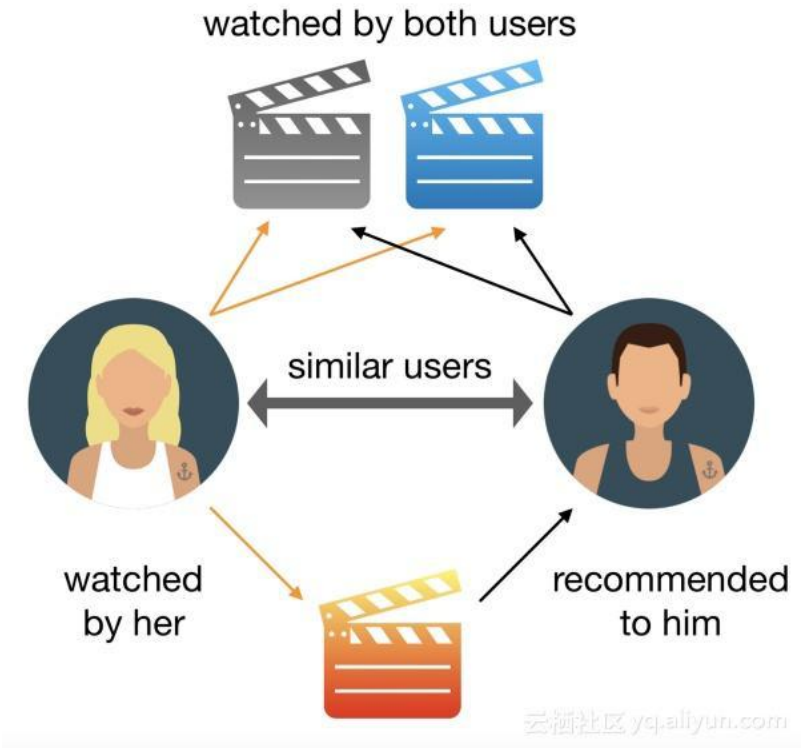
协同过滤的基本思想

过滤的含义

- 从海量产品中挑选出和用户偏好匹配的产品。

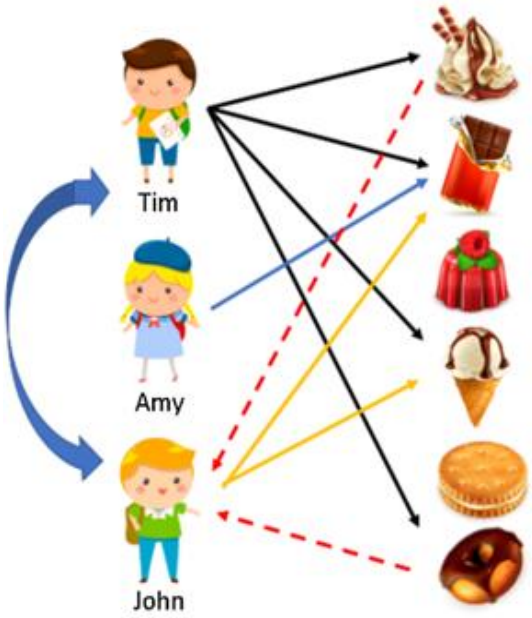
协同的含义

- 综合大量用户的信息，学习用户个体的偏好；
- 对比内容推荐中的“用户独立性”；

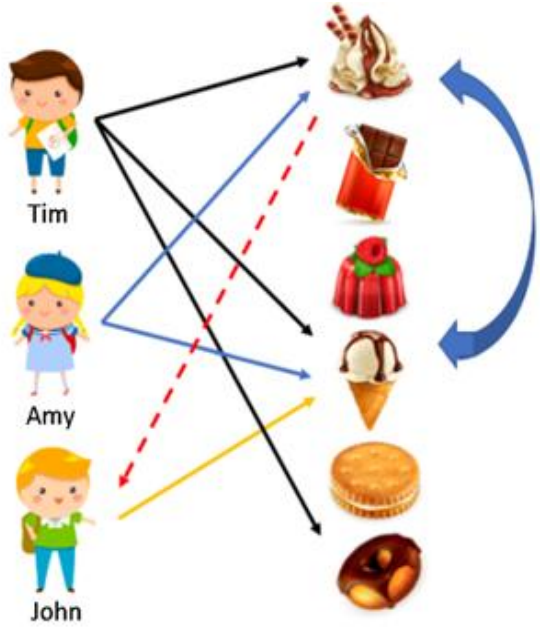


近邻（KNN）的含义

相似的用户：产品偏好相似的用户。



基于用户的协同过滤



基于产品的协同过滤

相似的产品：被相似的用户群偏好。

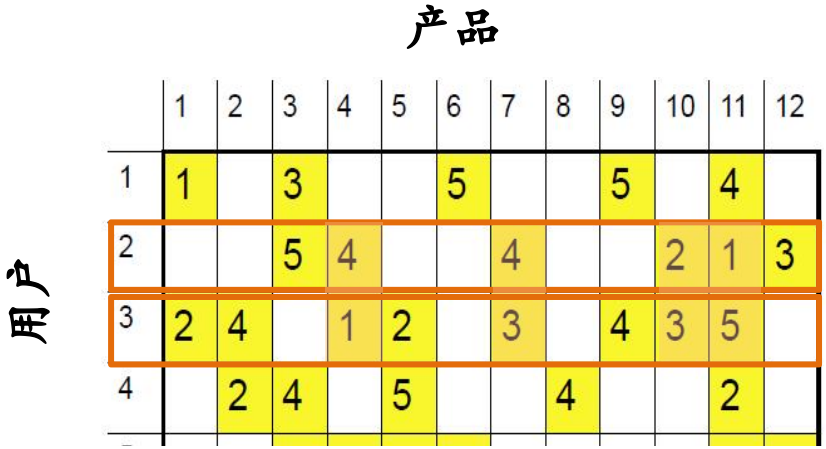
基于用户的协同过滤（UserKNN）

- 基本假设：相似的用户具有相似的偏好。

1. 计算用户相似度
2. 估计用户对于产品的偏好评分

UserKNN: 计算用户相似度

- 用户偏好向量：该用户对产品的评分向量；



$p_2 = (?, ?, 5, 4, ?, ?, 4, ?, ?, 2, 1, 3)$

$p_3 = (2, 4, ?, 1, 2, ?, 3, ?, 4, 3, 5, ?)$

- 用户相似度：计算用户偏好向量之间的余弦相似度；

仅计算两用户均有评分的维度。

$$s(u_2, u_3) = \frac{4*1 + 4*3 + 2*3 + 1*5}{\sqrt{4^2 + 4^2 + 2^2 + 1^2} * \sqrt{1^2 + 3^2 + 3^2 + 5^2}}$$

UserKNN: 估计偏好评分

- r_{uj}：用户 u对产品j的偏好评分

用户相似度作为权重

$$r_{uj} = \frac{\sum_{v \in N_u} sim(u, v) \times r_{vj}}{\sum_{v \in N_u} sim(u, v)}$$

用户 u的平均打分偏好

用户 v的平均打分偏好

$$r_{uj} = \overline{r_u} + \frac{\sum_{v \in N_u} sim(u, v)(r_{vj} - \overline{r_v})}{\sum_{v \in N_u} sim(u, v)}$$

UserKNN：算法流程

计算用户-用户相似度矩阵；

对于每一个用户u

找到其 top-N相似用户N_u；

对于用户u没有评分的任一产品j

计算偏好评分r_{uj}；

根据评分高低生成推荐列表；

该算法有何缺点？

- 频繁更新用户相似度矩阵；
- 计算用户相似度矩阵的时间、空间成本大，难以适应大规模用户的要求。

基于产品的协同过滤（ItemKNN）

- 解决用户相似度矩阵计算、存储开销过大的问题；
- 相对于用户规模，产品数目相对较小；
- 产品相似度矩阵相对稳定，无需频繁更新。

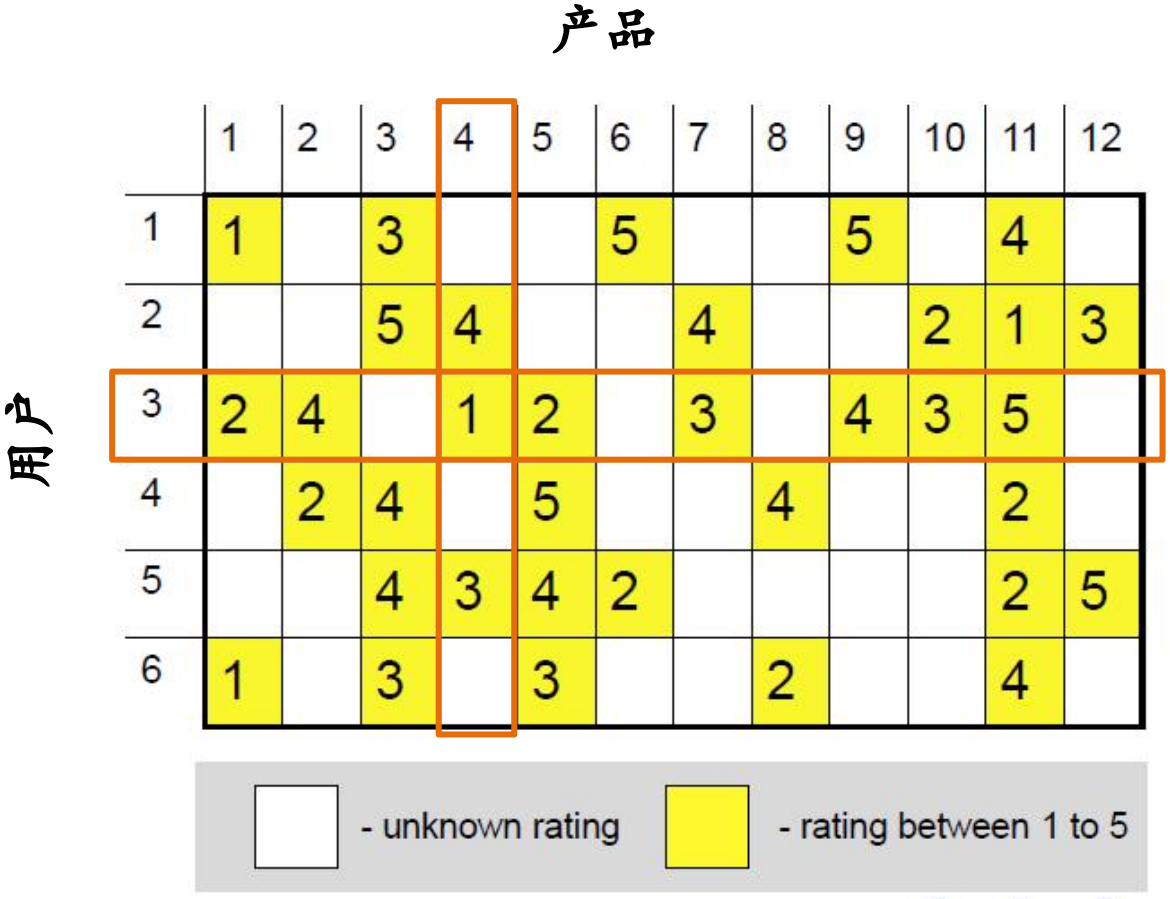
Amazon.com Recommendations

Item-to-Item Collaborative Filtering

Greg Linden, Brent Smith, and Jeremy York • *Amazon.com*

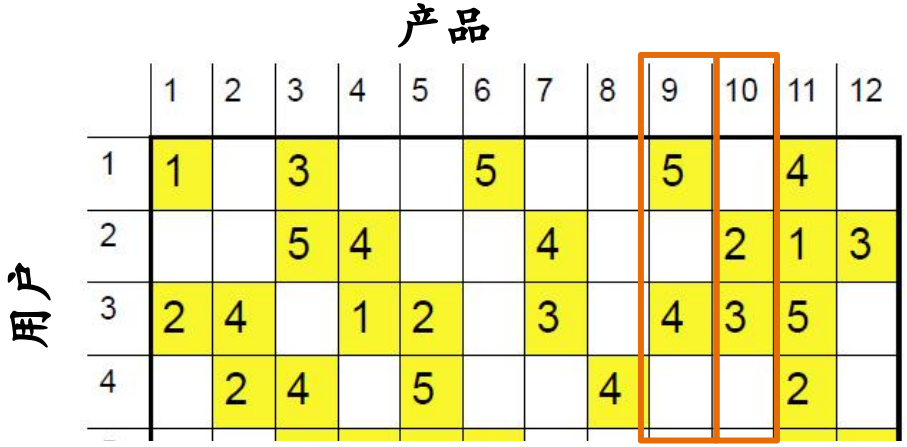
从UserKNN到ItemKNN

- 如何计算产品相似度？如何表征产品？



ItemKNN：计算产品相似度

- 产品特征向量：该产品得到的用户评分向量；



$$q_9 = (5, ?, 4, ?)$$
$$q_{10} = (4, 1, 5, 2)$$

- 计算产品特征向量之间的余弦相似度；

仅考虑均有用户评分的维度

$$sim(i_9, i_{10}) = \frac{q_9 \cdot q_{10}}{\|q_9\| * \|q_{10}\|} = \frac{5*4+4*5}{\sqrt{5^2+4^2} * \sqrt{4^2+5^2}}$$

ItemKNN：估计偏好评分

- r_{uj}：用户 u 对于产品 j 的偏好评分

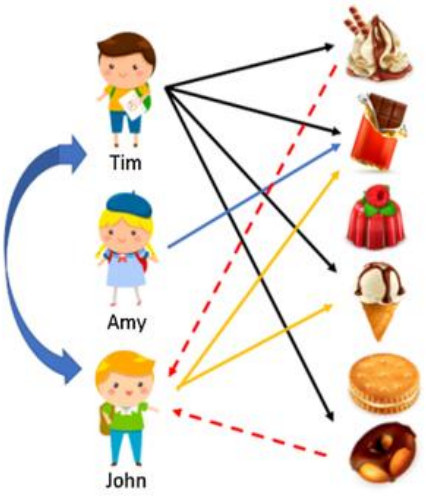
产品相似度作为权重

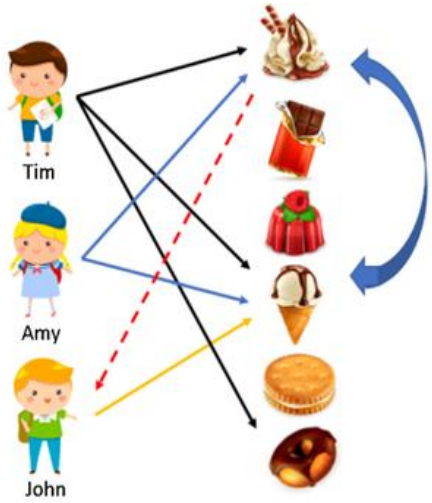
$$r_{uj} = \frac{\sum_{i \in N_j} sim(i, j) \times r_{ui}}{\sum_{i \in N_j} sim(i, j)}$$

产品 j 和产品 i 越相似，同一用户 u 对这两个产品给出的评分（r_{uj} 和 r_{ui}）越相似。

小结

- 基于邻域的协同过滤
  - 协同的含义
  - 邻域的类型
- 基于用户的协同过滤
  - 用户相似度计算
  - 评分预测
- 基于产品的协同过滤
  - 产品相似度计算
  - 评分预测





讲授提纲

01 推荐系统基本概念

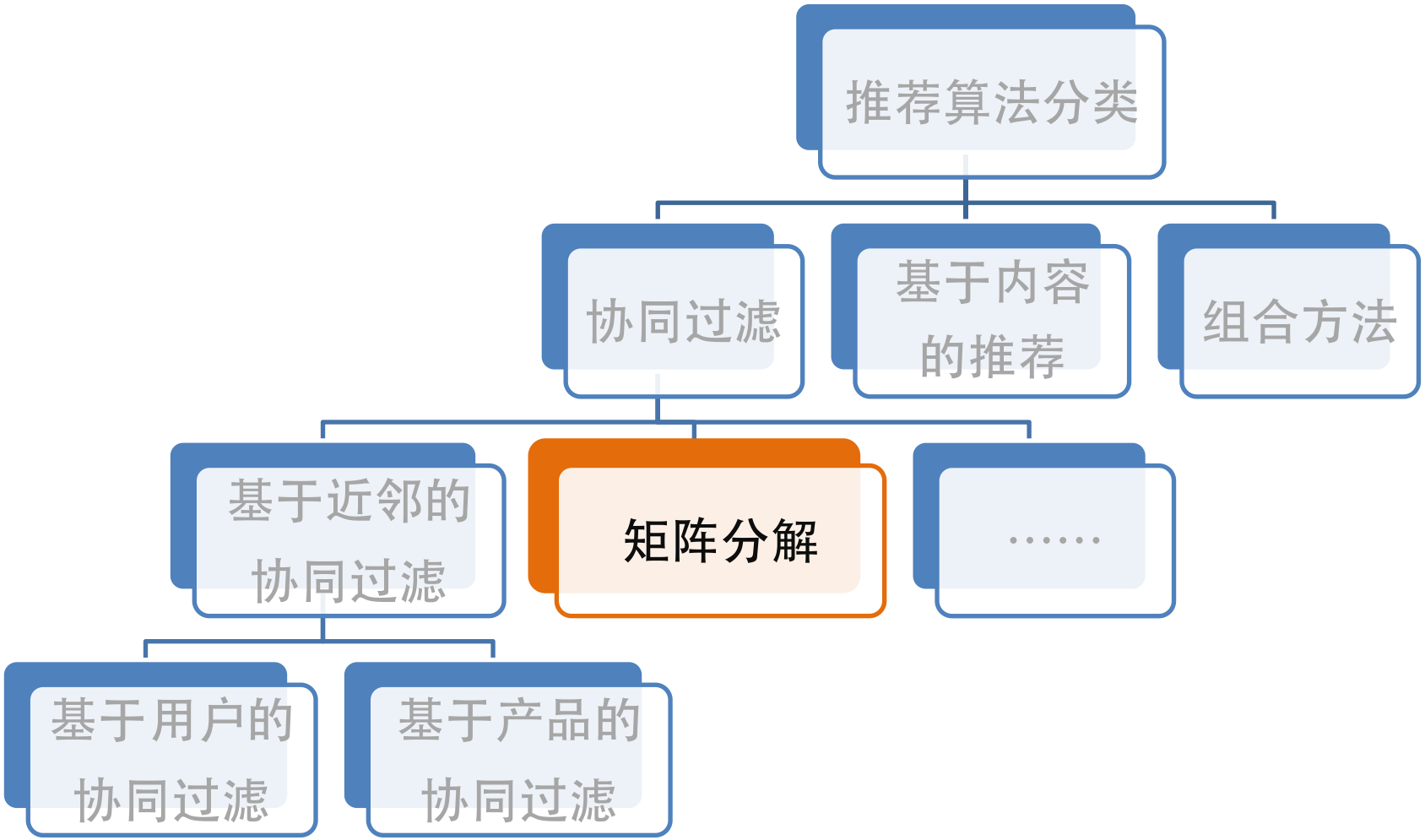
02 基于内容的推荐

03 基于近邻的协同过滤

04 基于矩阵分解的协同过滤

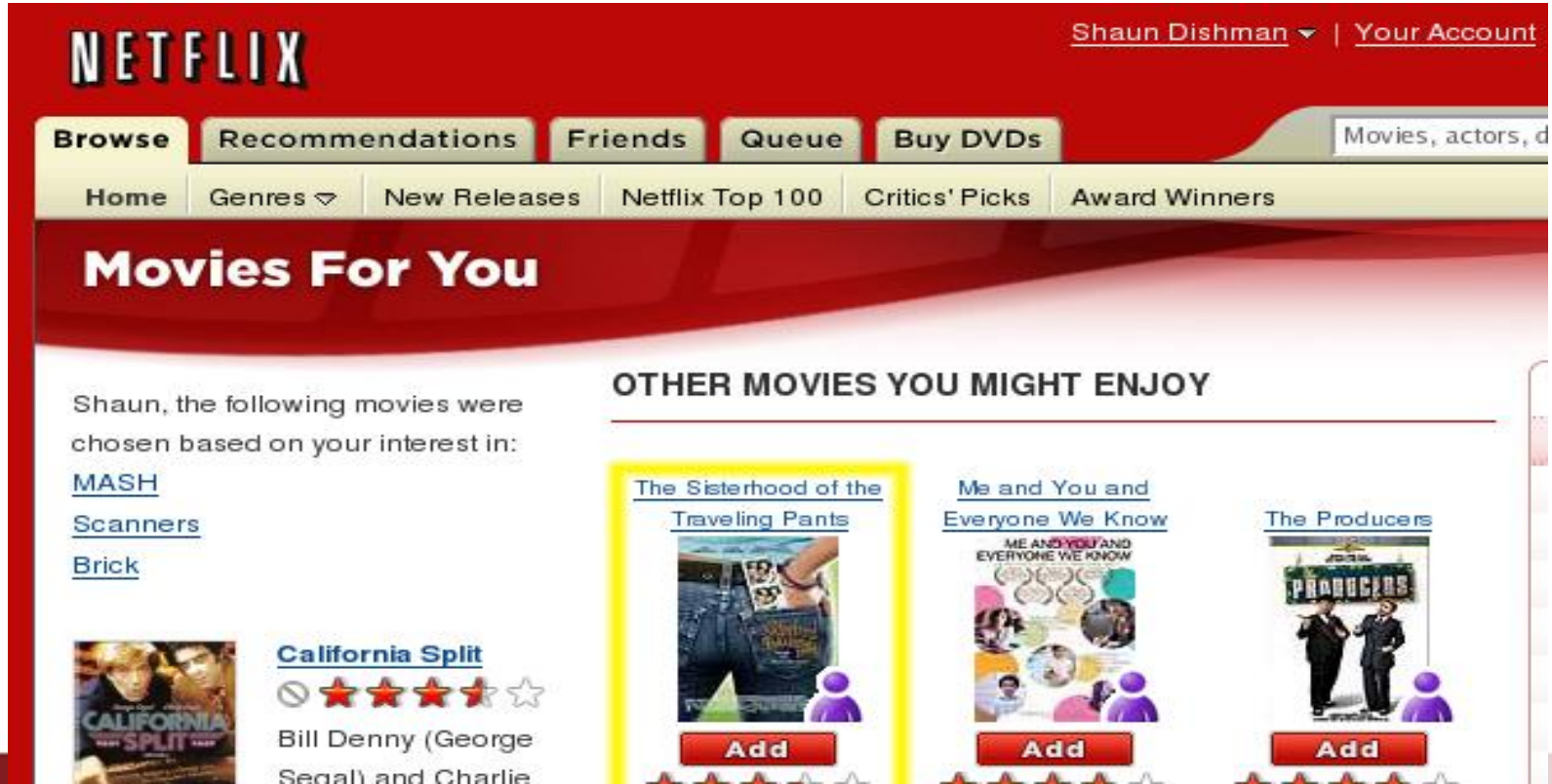
05 商务案例-电影推荐

推荐算法体系（回顾）


推荐算法分类
协同过滤
基于内容的推荐
组合方法
基于近邻的协同过滤
矩阵分解
……
基于用户的协同过滤
基于产品的协同过滤

矩阵分解推荐算法源起

- Netflix推荐系统竞赛
  - 2006年10月，正式启动，提供1亿条评分数据，悬赏100万美元，奖赏提升RMSE指标10%+的算法；
  - 2007年10月，加文·波特考虑心理学的锚定效应，提升9.06%;
  - 2009年7月，BellKor团队提升到10.06%，获得大奖。



Home | Rules | Leaderboard | Update

Leaderboard

https://www.netflixprize.com/leaderboard.htm

Showing Test Score. Click here to show quiz score

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
16	Invisible Ideas	0.8653	9.15	2009-07-15 15:53:04
17	Just a guy in a garage	0.8662	9.06	2009-05-24 10:02:54
18	J Dennis Su	0.8666	9.02	2009-03-07 17:16:17

矩阵分解技术在Netflix竞赛中脱颖而出

MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS

Yehuda Koren, *Yahoo Research*
Robert Bell and Chris Volinsky, *AT&T Labs—Research*

As the Netflix Prize competition has demonstrated, matrix factorization models are superior to classic nearest-neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels.

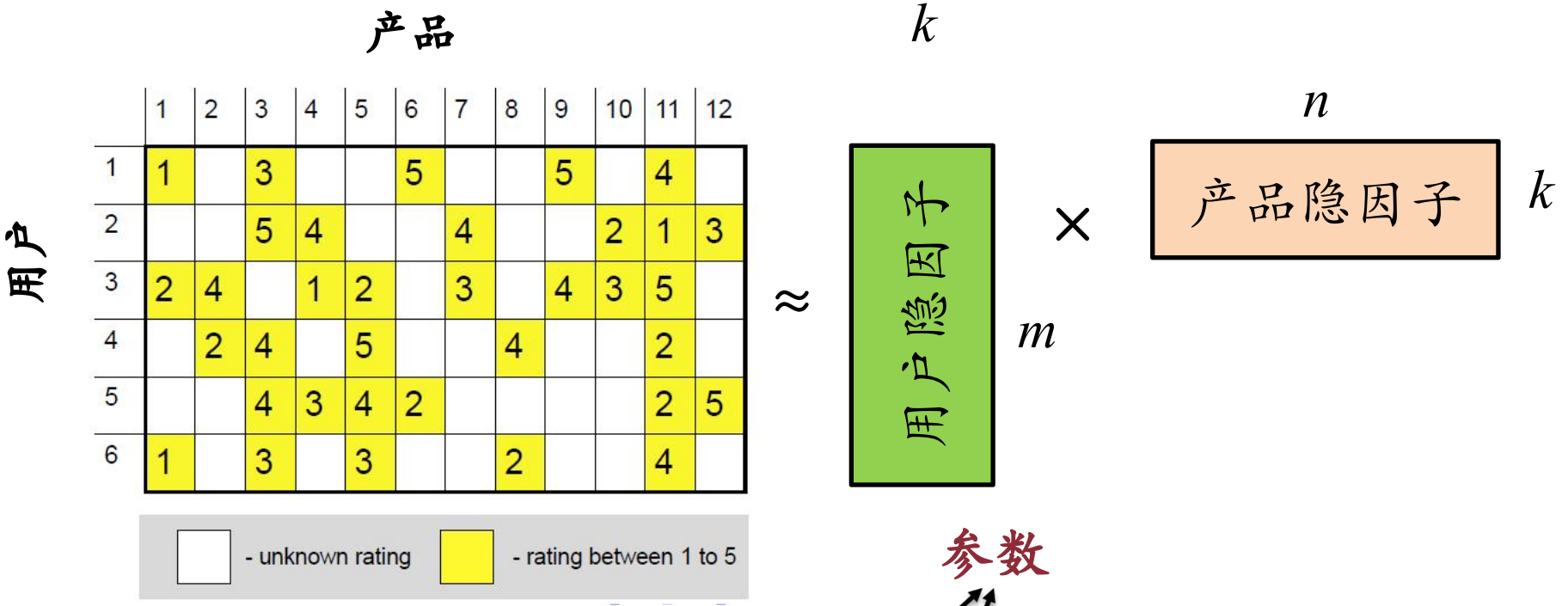
Such systems are particularly useful for entertainment products such as movies, music, and TV shows. Many customers will view the same movie, and each customer is likely to view numerous different movies. Customers have proven willing to indicate their level of satisfaction with particular movies, so a huge volume of data is available about which movies appeal to which customers. Companies can analyze this data to recommend movies to particular customers.

推荐问题定义（回顾）

给定用户集合U和产品集合I，推荐算法旨在基于历史评分数据学习用户$u \in U$对于<u>未评分</u>产品$i \in I$的**<u>偏好分值</u>**：

$$f(u,i) \rightarrow r_{ui}$$

矩阵分解（MF）



- 分解用户-产品评分：

$$r_{ui} = \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i$$

$\boldsymbol{p}_u$为用户u的K维隐因子向量，$\boldsymbol{q}_i$为产品i的K维隐因子向量。

解读隐因子含义

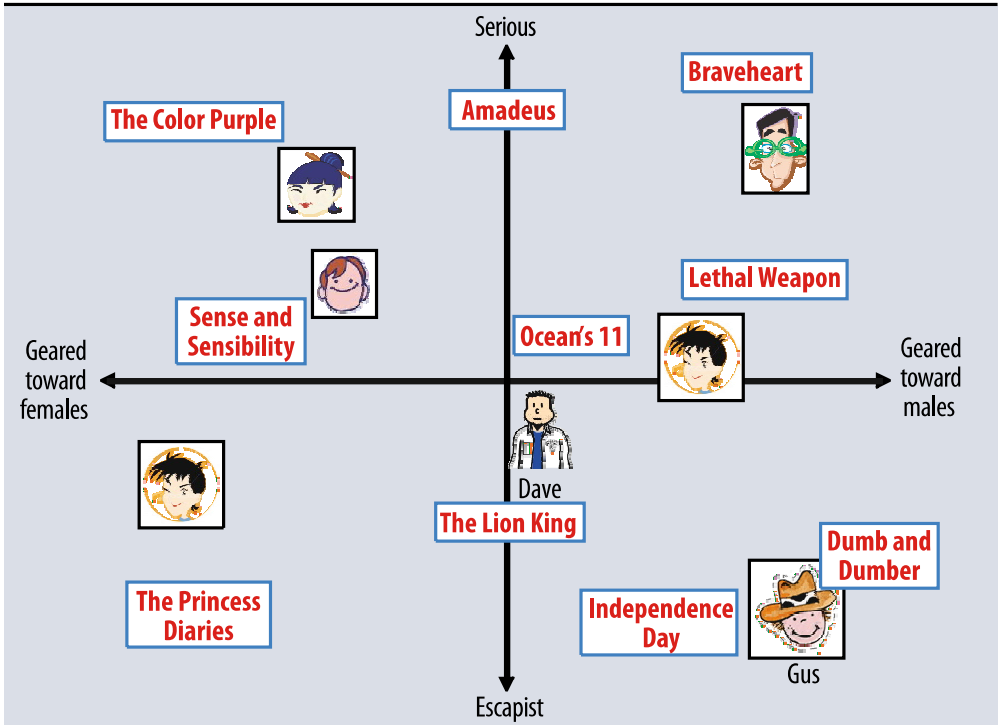


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

$$p_{Gus} = (2.8, -3.2)$$

$$q_{Independence\ Day} = (1.8, -3.2)$$

$$q_{The\ Color\ Purple} = (-2, 3.4)$$

$$p_{Gus} \cdot q_{Independence\ Day} > p_{Gus} \cdot q_{The\ Color\ Purple}$$

点积$p_u^T \cdot q_i$越大，
表示偏好程度越高。

参数学习：估计隐因子

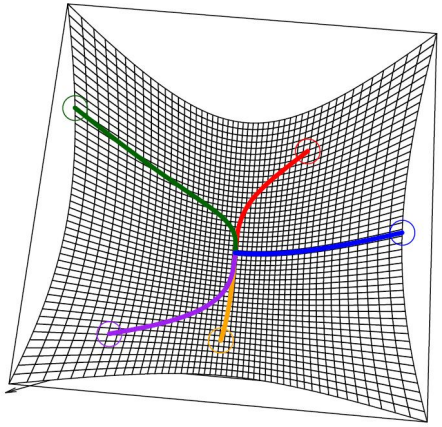
- 将参数学习问题转化为优化问题，最小化真实评分和预测评分之间的差距。
  - 真实评分：评分数据集中观测到的评分r_{ui}
  - 预测评分：基于矩阵分解技术估计的评分$\hat{r}_{ui} = \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i$

- 优化目标：

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{(u,i):r_{ui}\neq ?} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2 \right)$$

L_2正则项

梯度下降法简介



假设初始点x^0，利用梯度信息不断迭代：

$$x^{k+1} = x^k - a_k \nabla f(x^k)$$

其中$f(x)$为目标函数，$-\nabla f(x^k)$为梯度下降方向，a_k为步长。

梯度下降法

- 基于全部训练数据计算梯度，时间开销大；
- 不适合海量数据。

随机梯度下降法

- 对于每一条数据计算梯度，更新模型参数；
- 适合海量数据。

参数学习：基于随机梯度下降

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{(u,i):r_{ui}\neq ?} (r_{ui} - \hat{r}_{ui})^2 + \lambda\left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2\right)$$

对每一个用户-产品评分对r_{ui}，循环迭代直到收敛：

1. 通过参数的偏导数找到最速下降方向 (令 $e_{ui} = r_{ui} - \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i$)

$$\frac{\partial C}{\partial p_{uk}} = -2q_{ik}e_{ui} + 2\lambda p_{uk}$$

$$\frac{\partial C}{\partial q_{ik}} = -2p_{uk}e_{ui} + 2\lambda q_{ik}$$

2. 通过迭代法不断地优化参数 (α 是学习率)

$$p_{uk} = p_{uk} - \alpha \frac{\partial C}{\partial p_{uk}}$$

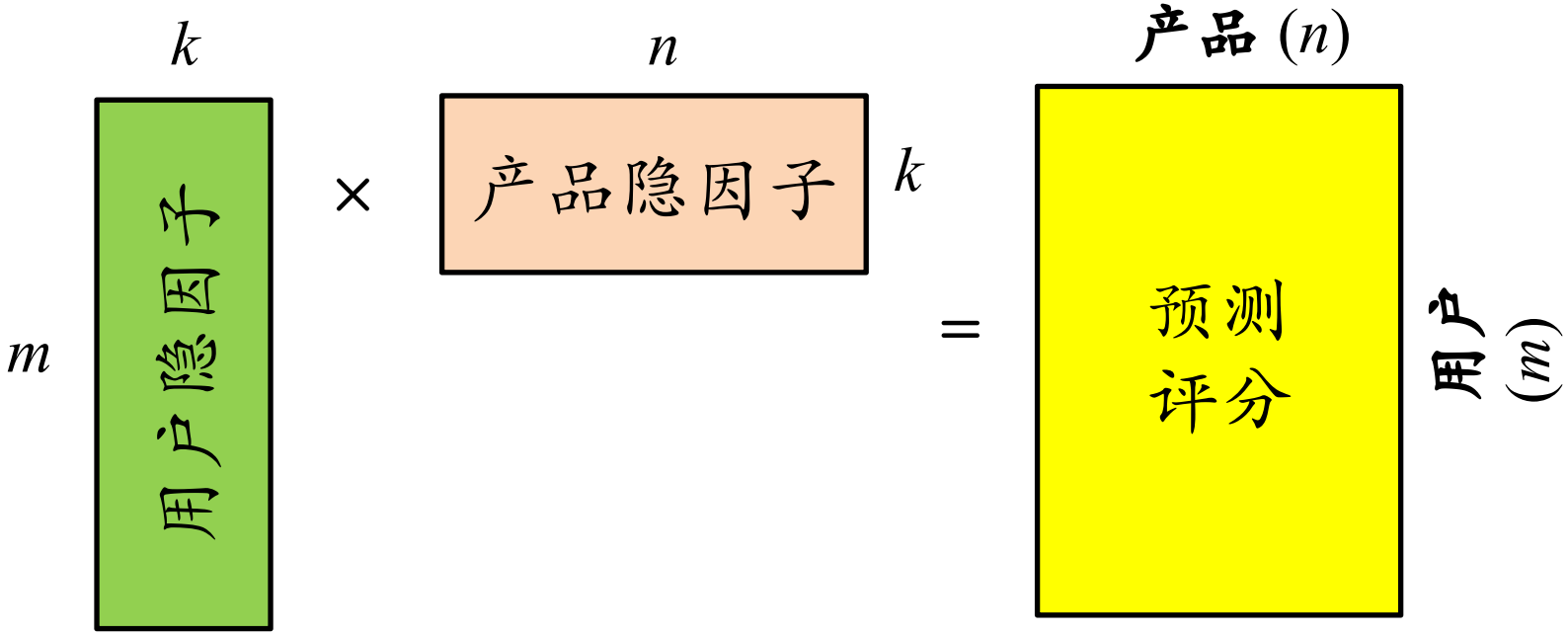
$$q_{ik} = q_{ik} - \alpha \frac{\partial C}{\partial q_{ik}}$$

矩阵分解推荐策略

基于预测评分$\hat{r}_{ui}$填充用户-产品矩阵缺失值；

对于每个用户

根据填充矩阵，推荐top-N评分产品；



拓展总结：如何设计推荐算法？

以MF为例：

预测模型 r_{ui}

$$\hat{r}_{ui} = \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i$$

优化目标 $f(r_{ui}, \hat{r}_{ui})$

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{(u,i):r_{ui}\neq ?} (r_{ui} - \hat{r}_{ui})^2 + \lambda\left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2\right)$$

推荐策略

根据$\hat{r}_{ui}$推荐；

拓展预测模型：从MF到SVD

SVD：

预测模型 r_{ui}

$$\hat{r}_{ui} = \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i + b_u + b_i + b$$

优化目标 $f(r_{ui}, \hat{r}_{ui})$

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{(u,i):r_{ui}\neq ?} (r_{ui} - \hat{r}_{ui})^2 + \lambda\left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2\right)$$

推荐策略

根据$\hat{r}_{ui}$推荐；

拓展优化目标：从MF到BPR

BPR：

预测模型 r_{ui}

$$\hat{r}_{ui} = \boldsymbol{p}_u^T \cdot \boldsymbol{q}_i$$

优化目标 $f(r_{ui}, \hat{r}_{ui})$

$$\min_{\boldsymbol{p},\boldsymbol{q}} \sum_{(u,i,j)\in D_s} -\ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) + \lambda\left(\sum_u \|\boldsymbol{p}_u\|^2 + \sum_i \|\boldsymbol{q}_i\|^2\right)$$

推荐策略

根据$\hat{r}_{ui}$推荐；

Rendle, Steffen, et al. BPR: Bayesian personalized ranking from implicit feedback

小结

- **矩阵分解与Netflix推荐系统竞赛**
- **矩阵分解**
  - 隐因子分解
  - 优化目标：最小化预测评分和真实评分的平方误差
  - 随机梯度下降
- **推荐算法设计的一般思路**
  - 预测模型、优化目标、推荐策略

讲授提纲

01 推荐系统基本概念

02 基于内容的推荐

03 基于近邻的协同过滤

04 基于矩阵分解的协同过滤

05 商务案例-电影推荐