第九届中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2017

推荐系统的架构演进

程晓澄@第四范式 2017.10.21

第四范式　守正·出奇

AI For Everyone

一切以为客户创造价值为出发点,突破思维,同时不固守常规

戴文渊

创始人CEO

机器学习全球领军学者

多次获得KDD Cup，ACM世界冠军

曾任最年轻的百度高级科学家

曾任华为诺亚方舟实验室主任科学家

曾指导百度凤巢在线营销系统、百度大脑等多个重量级核心产品，使百度变现能力4年提升8倍，移动端变现能力持续超华尔街预期的核心驱动力

陈雨强

资深互联网数据建模专家

世界级深度学习专家

曾任百度凤巢系统模型专家

曾任今日头条推荐系统负责人

周开拓

推荐系统架构专家

5年+的推荐算法和产品经验

曾负责手机淘宝推荐系统

田　枫

资深数据分析与挖掘专家，在线精准营销最早实践者

曾任百度商业广告体系

曾任奇虎商业广告体系BA团队负责人

程晓澄

资深数据科学家

曾负责豆瓣FM推荐系统

前言

推荐系统的诞生土壤和早期演进

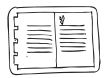
推荐系统当下的基本架构

搭建一个推荐系统

前言

推荐系统的诞生土壤和早期演进

- 诞生土壤
- 按经验策略排序阶段
- 机器学习阶段



推荐系统当下的基本架构

- 召回
- 排序
- 优化列表

搭建一个推荐系统

- 工程实践
- 机器学习实践
- 研究热点

前言

推荐系统的诞生土壤和早期演进

- 诞生土壤
- 按经验策略排序阶段
- 机器学习阶段

推荐系统当下的基本架构

- 召回
- 排序
- 优化列表

搭建一个推荐系统

- 工程实践
- 机器学习实践
- 研究热点

推荐系统的诞生土壤

- 1995 ： 亚马逊网上书店
- 1998 ： Yahoo! -> Google
- 2004 ： 门户网站 -> web2.0
- *信息分发

什么是好的帖子：聪明才智 + 经验 rank

- Reddit / Quora / Stack Overflow
- 大众点评 / 糗事百科
- facebook newsfeed @ 2006

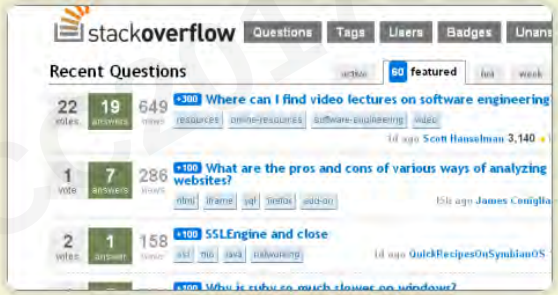
2012年

- 高斯模糊的算法（47@2012.11.14）
- 贝叶斯推断及其互联网应用（三）：拼写检查（35@2012.10.16）
- 虚数的意义（101@2012.09.24）
- 基于用户投票的排名算法（六）：贝叶斯平均（24@2012.03.28）
- 基于用户投票的排名算法（五）：威尔逊区间（41@2012.03.20）
- 基于用户投票的排名算法（四）：牛顿冷却定律（20@2012.03.16）
- 基于用户投票的排名算法（三）：Stack Overflow（20@2012.03.11）
- 基于用户投票的排名算法（二）：Reddit（33@2012.03.07）
- 基于用户投票的排名算法（一）：Delicious和Hacker News（37@2012.02.24）

$$\frac{(log_{10}Qviews)\times 4+\frac{Qanswers\times Qscore}{5}+sum(Ascores)}{((Qage+1)-(\frac{Qage-Qupdated}{2}))^{1.5}}$$

在Stack Overflow的页面上，每个问题前面有三个数字，分别表示问题的得分、回答的数目和该问题的浏览次数。以

础，就可以设计算法了。



创始人之一的Jeff Atwood，曾经在几年前，公布过排名得分的计算公式。

写成php代码，就是下面这样：

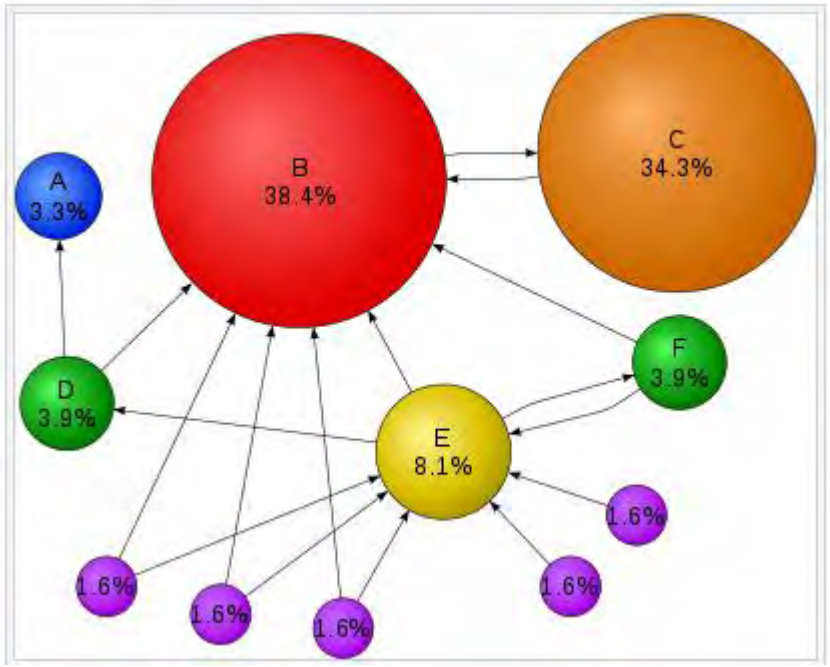
```
function hot($Qviews, $Qanswers, $Qscore, $Ascores, $date_ask, $date_active)
{
    $Qage = time() - strtotime(gmdate("Y-m-d H:i:s",strtotime($date_ask)));
    $Qage = round($Qage/3600, 1);

    $Qupdated = time() - strtotime(gmdate("Y-m-d H:i:s",strtotime($date_active)));
    $Qupdated = round($Qupdated/3600, 1);

    $dividend = (log10($Qviews)*4) + (($Qanswers * $Qscore)/5) + $Ascores;
    $divisor = pow((($Qage + 1) - ($Qage - $Qupdated)/2), 1.5);

    echo $dividend/$divisor . "\n";
}
```

什么是好的帖子：聪明才智 + 经验 rank

- 离线计算：
  - for (item in all_items):
    - score = score_func(item)
    - “UPDATE item SET score = %d” % score
- 在线查询
  - “SELECT item_id FROM item ORDER BY score”

什么是好的网站：聪明才智 + 经验 rank

- Google PageRank，季度更新



$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

什么是好的帖子：聪明才智 + 经验 rank

- 离线计算：
  - Map-reduce / MPI：矩阵迭代计算
- 在线查询
  - “SELECT score FROM item WHERE page_id = %s” % page_id

协同过滤

- 评分矩阵

ItemID \ user	1	2	3
101	0	0.1	0
102	0.1	0.7	0.5
103	0.2	0	0
104	0.3	0	0.7
105	0	0.9	0.5

- user based recommendation

$$r_{u,i} = k \sum_{u' \in U} \text{simil}(u, u') r_{u',i}$$

- item based recommendation

点积：

$$Inner(x,y) = \sum_i x_i y_i = \langle x, y \rangle$$

cosine相似度：

$$CosSim(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}} = \frac{\langle x, y \rangle}{||x||\,||y||}$$

Pearson相关系数：

$$\begin{aligned} Corr(x,y) &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2}\sqrt{\sum (y_i - \bar{y})^2}} \\ &= \frac{\langle x - \bar{x},\, y - \bar{y} \rangle}{||x - \bar{x}||\,||y - \bar{y}||} \\ &= CosSim(x - \bar{x}, y - \bar{y}) \end{aligned}$$

协同过滤

- 离线计算：
  - Map-reduce：
  - sim[i][j] = sim_score_func(i, j)
  - redis.set(“%s_sim_items” % i, sorted(score[i]))
- 在线查询
  - redis.get(“%s_sim_items” % item_id)

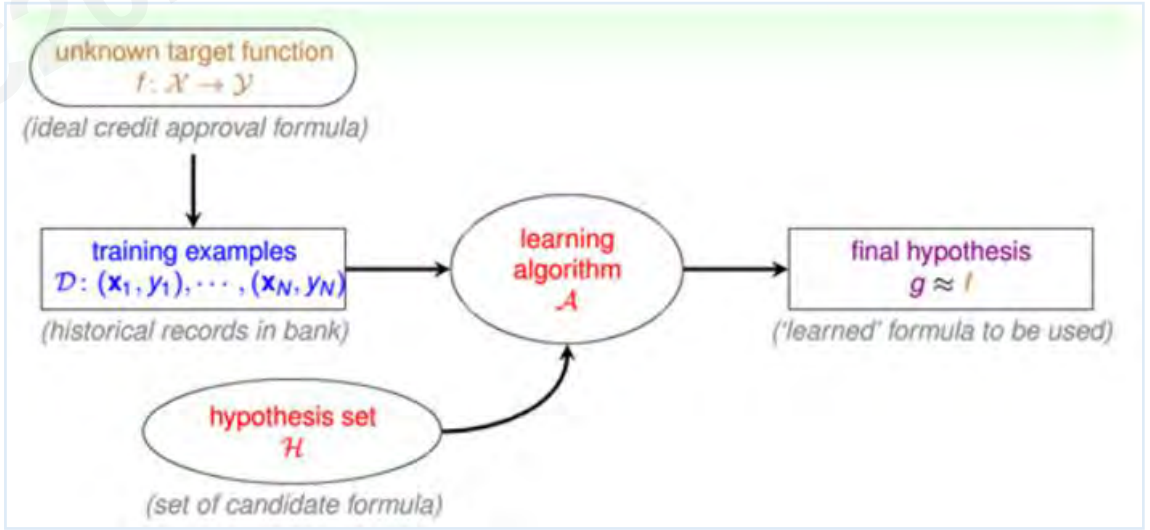
不足

- 评分 = 写死的公式(统计数据)
  - 抽象假设：用户喜欢看相似的内容、用户喜欢相似的人的推荐
  - 实施环节：这个相似度的度量方式是好的
- 真的吗
  - 每个用户都喜欢这样的推荐吗？
  - 每个 item 的相似都是这样度量的吗？
  - 每个 item 、每个 category 的权重、表达的信息含量一样吗？
  - 如果不是，那是怎样的？
  - 有没有更好的假设？

- 比随机或热度好，
- 能否有更丰富的假设，并针对每个用户的正负反馈自我修正、选择能力

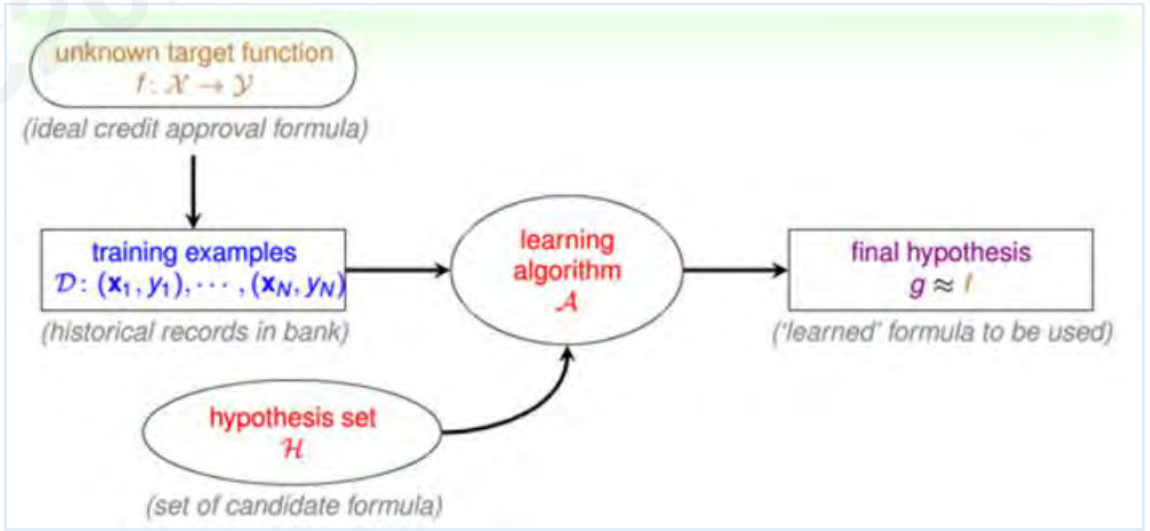
机器学习

- 监督学习
  - Y = f(X)
  - 损失函数 Cost:
    - cost(Y, g(X))
  - 优化算法 A:
    - 最小化 cost
    - 使 g 逼近于 f



机器学习

- 监督学习
  - Y = f(X) ， 如 $Y = ax_1 + bx_2 + c$
  - 损失函数 Cost:
    - cost(Y, g(X)) ， 如 $(Y - g(x))^2$
  - 优化算法 A:
    - 最小化 cost
    - 找到最佳 parameters：{a=3,b=4,c=2}
    - 使 g 逼近于 f ， $g(x) = 3x_1 + 4x_2 + 2$



Y ：点击率 / 观看时长 / 评分 / 其他量化体验或营收的值

矩阵分解

- 2007年 NETFLIX
- $1million

- Y = f(X)
- rating = f(u, i)

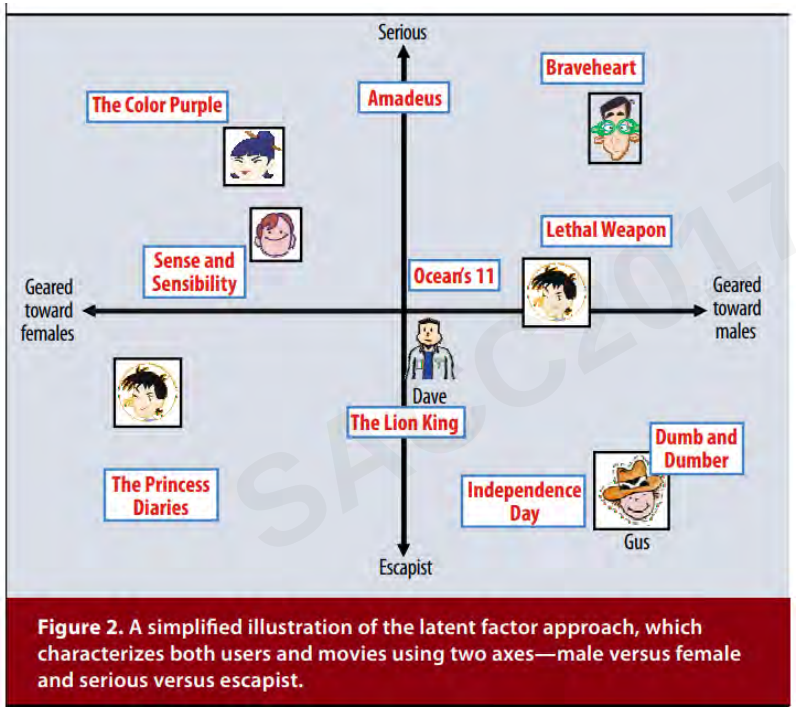


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

$$\hat{r}_{ui} = q_i^T p_u$$

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

$$\min_{p^*,q^*,b^*} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda$$

$$(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2)$$

局限

- 用到的信息只有 uid，iid，历史评分
- 新用户、新内容就无法刻画了吗？
- 用户的信息、内容的特征无法利用吗？

举例

上下文信息

- 顾客有没有带男朋友
- 当前季节，时间
- 店铺位置

人是怎么判断的？

观察客户

- 顾客穿什么样的，什么牌子的衣服
- 顾客拎的什么包
- 顾客之前买了什么东西
- 顾客进店以后看了什么商品，花了多少时间？

了解商品

- 衣服的过往销量
- 衣服的款式、面料、品牌故事
- 衣服的价格、打折幅度、活动

为机器学习模型注入更多的特征

- 为了用上更多的特征，把更多的猜想、假设空间扔给机器学习引擎去学习

- Y = f(X) = f(user_info, item_info, context,
- user_history, item_history,
- score, rank, similarity, …)

- 成为广告投放、搜索排序、电商、内容推荐领域的核心引擎

机器学习

- 离线计算 基于参数服务器：
- 十亿级别训练数据，亿级别特征数，多 worker 异步更新
  - for i in range(PASS_NUMBER):
    - @async
    - for worker in workers:
      - while(training_instances = fetch_training_instances(instance_count=100)):
        - parameters_grad = sgd(training_instances).
        - parameter_server.update(parameters_grad)
  - parameters = dump_result_from_parameter_server()
- 在线查询
  - score = f(parameters, user_features, item_features, …)　　//百千级别parameters量

前言

推荐系统的诞生土壤和早期演进

诞生土壤

按经验策略排序阶段

机器学习阶段

推荐系统当下的基本架构

- 召回
- 排序
- 优化列表

搭建一个推荐系统

- 工程实践
- 机器学习实践
- 研究热点

典型架构

从客户提供的数百亿内容中选出数千

大规模机器学习模型排序

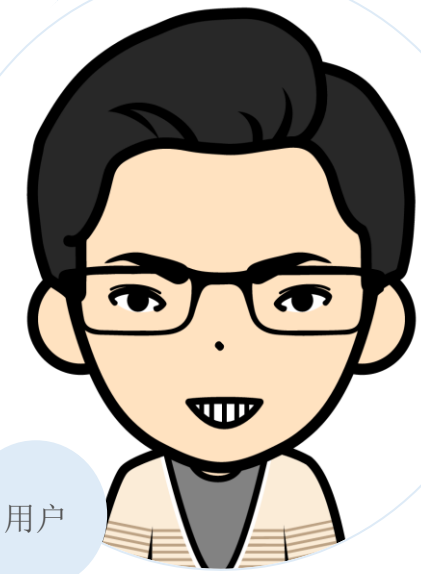
基于场景进行去重、多样性控制、加权，生成最终的推荐列表

线上实时产生并更新

典型架构-召回

- 排序召回
  - 最新、最热、最近、最常光顾、
  - 各种经验上的评分公式…
- 简单模型、rank 召回
  - item based / user based
  - 矩阵分解…
- 规则召回
  - 天气、近期搜索浏览、朋友的购买、同期过往习惯…

排序：考虑更多的因素



用户

1 秋天到了，推荐的商品应季吗？

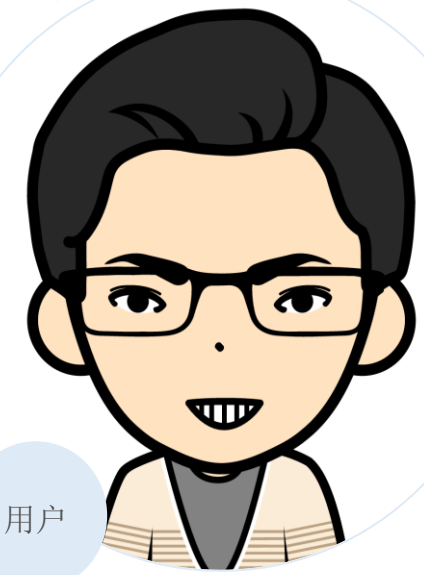
2 他在回家的路上，手机信号4G

3 他在你的app上非常活跃，有大量的社交行为

4 这是他在你平台买过的裤子，代表了他的品味

5 他在什么地方？

排序：考虑更多的因素



用户

1 秋天到了，推荐的商品应季吗？

- 时间、季节、周日

2 他在回家的路上，手机信号4G

- 运营商信息、连接方式

3 他在你的app上非常活跃，有大量的社交行为

- 行为的序列、社交关系

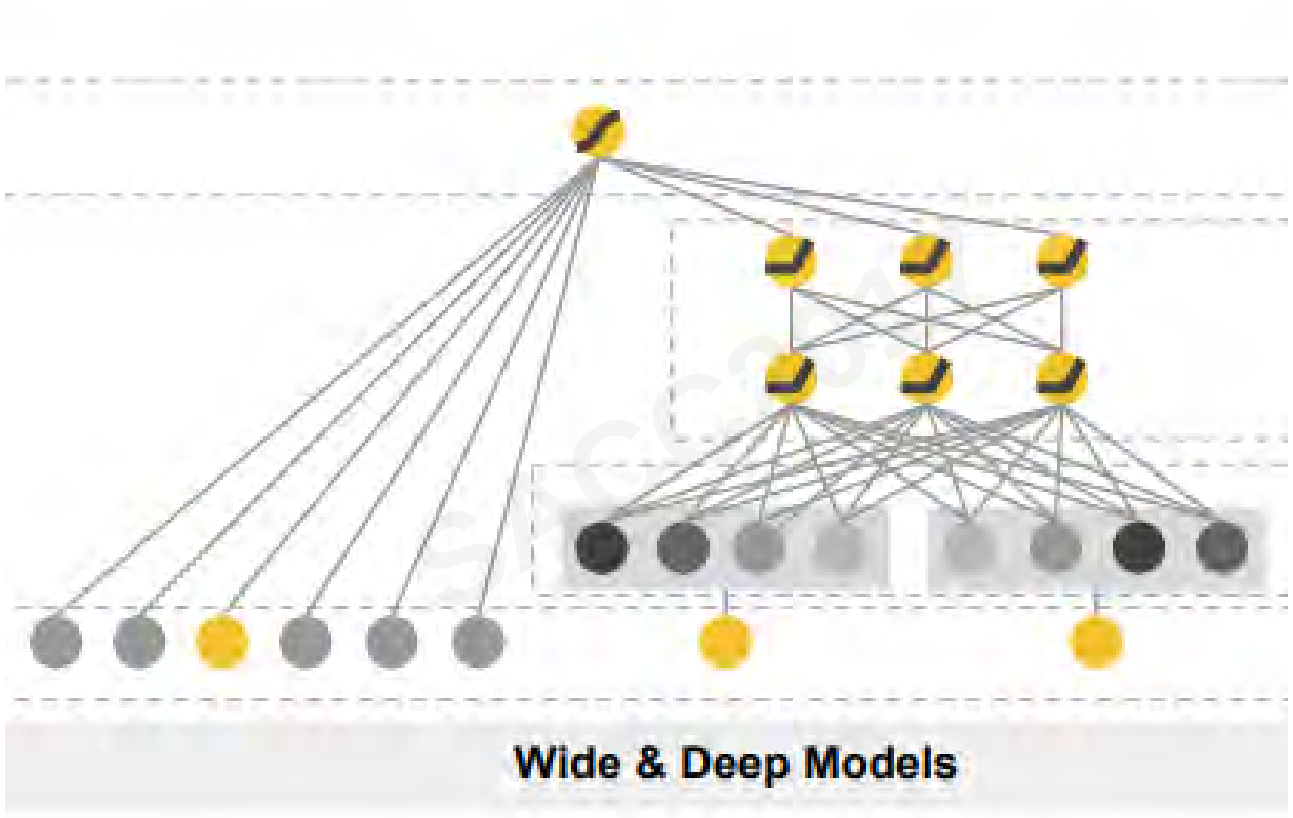
4 这是他在你平台买过的裤子，代表了他的品味

- 历史行为内容序列

5 他在什么地方？

- GPS 坐标、运营商信息、IP地址

典型架构-排序



时间、社交关系、GPS、
....
商品属性、卖家信息、价格....

观看、收藏、购买、搜索序列（embedding）

图片信息（CNN）　文字信息（word2vec）

Rank、Score、Similarity

典型架构-生成推荐结果

- 优化整体体验
  - 多样化
  - Exploit vs Explore
  - 准确性 vs 多样性 vs 新颖性

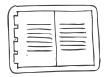
前言

推荐系统的诞生土壤和早期演进

诞生土壤

按经验策略排序阶段

机器学习阶段



推荐系统当下的基本架构

- 召回
- 排序
- 优化列表

搭建一个推荐系统

- 工程实践
- 机器学习实践
- 研究热点

工程实践：线上请求

物料库召回（CF、MF）

关注的人、内容

人群、地域

我们认为你可能喜欢的

机器学习推荐模型排序

对选出内容进行转化率预估

模型

定制化重排序，生成推荐列表

高转化推荐

低转化推荐

优化推荐列表结构

保证内容多样性
保证内容新鲜
综合多种来源

高转化推荐

低转化推荐

关系数据库、NoSQL、缓存

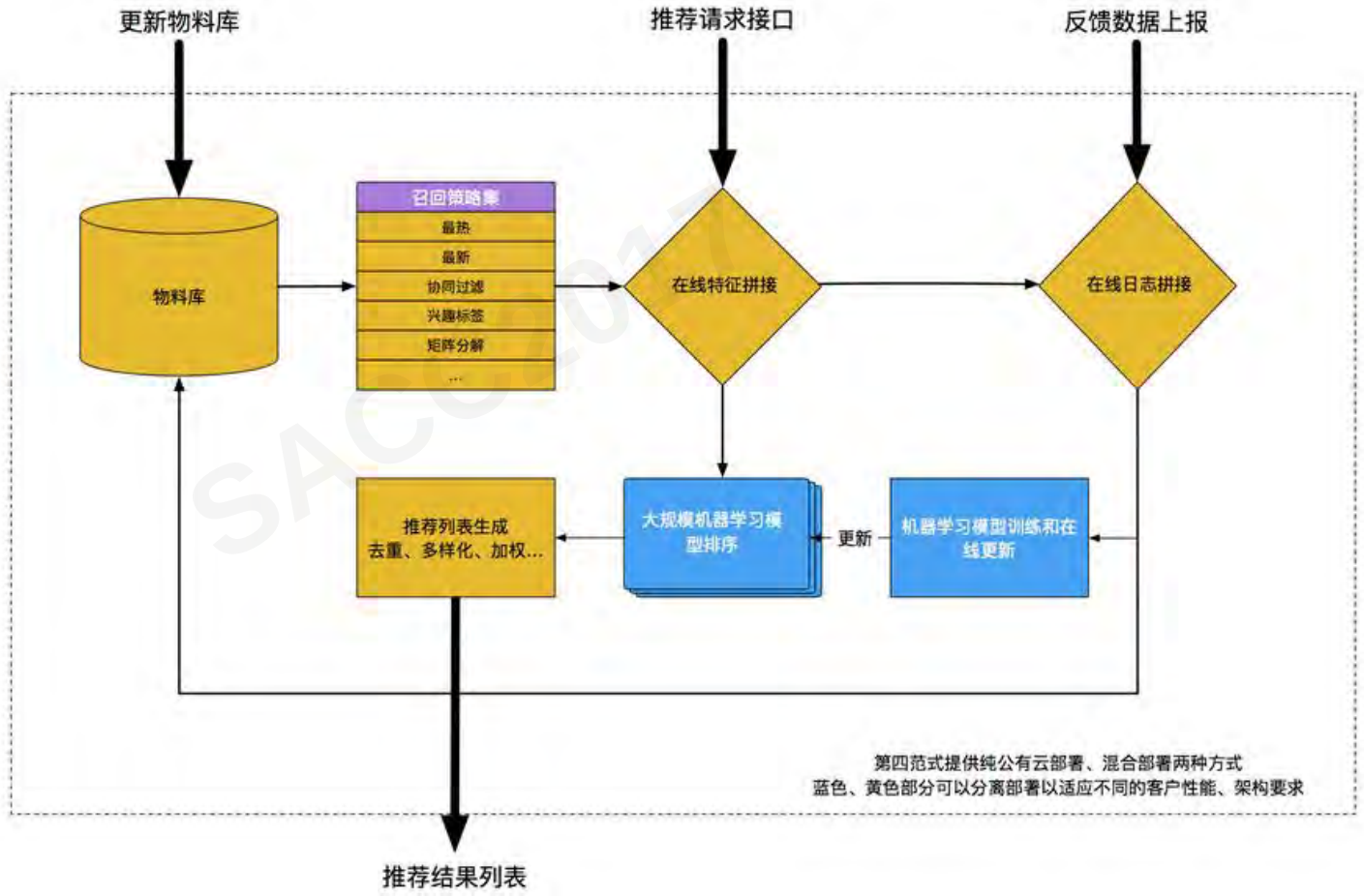
读取模型参数进行计算

基于场景进行去重、多样性控制、加权，生成最终的推荐列表

线上实时产生并更新

工程实践：线下数据流闭环

- 实现或应用一个推荐系统
  - 物料库
  - 原始特征记录
  - 反馈数据拼接



工程实践：数据分析、算法实验

“一个象棋大师会被一个每回合走两步的业余选手轻松击败”

用户的困惑

为毛总是推荐美国大选的新闻给我

我在上班呢，能别推我女儿的泳装照给我吗

这推荐新闻全部是标题党

总是推荐全世界人民都看过的内容给我

工程实践：数据分析、算法实验

“一个象棋大师会被一个每回合走两步的业余选手轻松击败”

- 可响应产品、性能、算法需求的架构
- 可同时进行大量实验的环境机制

机器学习实践面临的挑战

Speed：

- 数据的增长不受技术限制，在有限的时间内完成模型训练是机器学习计算框架最大的挑战之一
- 使用更多的特征、更复杂的模型会提高效果，然而由于计算资源限制不得不在效果和成本间进行取舍
- 需要有专门为机器学习任务优化的计算框架

Scalability：

- 业务增长的速度不仅是量的增长，更是维度的增长
- 快速发展的创业公司需要能匹配自己增长全周期的机器学习解决方案

研究

Y = f(X)

Y:

产品、交互设计

X :

特征更多种类的特征：挖掘图像、音频、文本特征、Session 类特征

特征工程：通过特征组合、特征变换，丰富假设空间

f:

模型抽象与相匹配的优化算法

商务合作

第四范式数据技术有限公司
Add: 北京.海淀.上地东路35号院.颐泉汇303
www.4paradigm.com