大数据分析与应用

Chapter 8 推荐系统II

8.1 基于近邻的协同过滤（回顾）

8.2 基于模型的方法（基于朴素贝叶斯的推荐、矩阵分解）

8.3 基于内容的方法（标签特征抽取、嵌入、用户画像）

8.* 数据中台软件枚举

8.4 基于上下文的推荐（预算、时效、地理信息）

8.5 其他问题（集成、网络化数据、点击率、排名、组推荐）

8.1 基于近邻的协同过滤（回顾）

- 1．基于用户的协同过滤算法：给用户推荐和他兴趣相似的其他用户喜欢的物品。
- 2．基于物品的协同过滤算法：给用户推荐和他之前喜欢的物品相似的物品

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}}$$

- 缺点:相似性计算的时间复杂度和空间复杂度较高。
- 改进：1）使用聚类方法替代相似性计算:只在簇内计算相似度；
- 改进：2）使用回归模型替代相似性计算：用参数估计代替相似度计算；

$$p(u,i) = \sum_{v} w_{uv} r_{vi}$$

使用回归方法的局限：数据的稀疏性；

8.2 基于模型的协同过滤

- 协同过滤：使用用户行为数据；

- 8.2.1 NBCF 朴素贝叶斯协同过滤
- 8.2.2 LFM 潜在因子模型

8.2.1 朴素贝叶斯协同过滤 NBCF

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

$$P(r_{31}=1|\text{商品}I_i\text{的所有评分})$$

$$\propto P(r_{31}=1)\cdot P(\text{商品}I_i\text{的所有评分}|r_{31}=1)$$

$$P(r_{31}=1|\text{商品}I_i\text{的所有评分})$$

$$\propto P(r_{31}=1)\cdot P(\text{商品}2=1|r_{31}=1)$$

$$\cdot P(\text{商品}3=1|r_{31}=1)$$

$$\cdot P(\text{商品}4=-1|r_{31}=1)$$

$$\cdot P(\text{商品}5=-1|r_{31}=1)$$

$$P(r_{31}=1)=2/4=0.5$$

$$P(\text{商品}2=1|r_{31}=1)=1/2=0.5$$

$$P(\text{商品}3=1|r_{31}=1)=1/1=1$$

$$P(\text{商品}4=-1|r_{31}=1)=2/2=1$$

$$P(\text{商品}5=-1|r_{31}=1)=1/2=0.5$$

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	1	-1	1	-1	1	-1
2	1	1	?	-1	-1	-1
3	?	1	1	-1	-1	?
4	-1	-1	-1	1	1	1
5	-1	?	-1	1	1	1

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

提示：CF看作含缺失数据的分类问题，则可以使用任意分类模型。

8.2.2 潜在因子模型（LFM）

- LFM：Latent Factor Model 潜在因子模型
- LSA：Latent Semantic Analysis 隐含语义分析
- MF： Matrix Factorization 矩阵分解（Decomposition）

$$\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\\0&0&0&1&1&1\\1&1&1&0&0&0\end{bmatrix}=\begin{bmatrix}1&0\\0&1\\0&1\\1&0\end{bmatrix}\begin{bmatrix}1&1&1&0&0&0\\0&0&0&1&1&1\end{bmatrix}$$

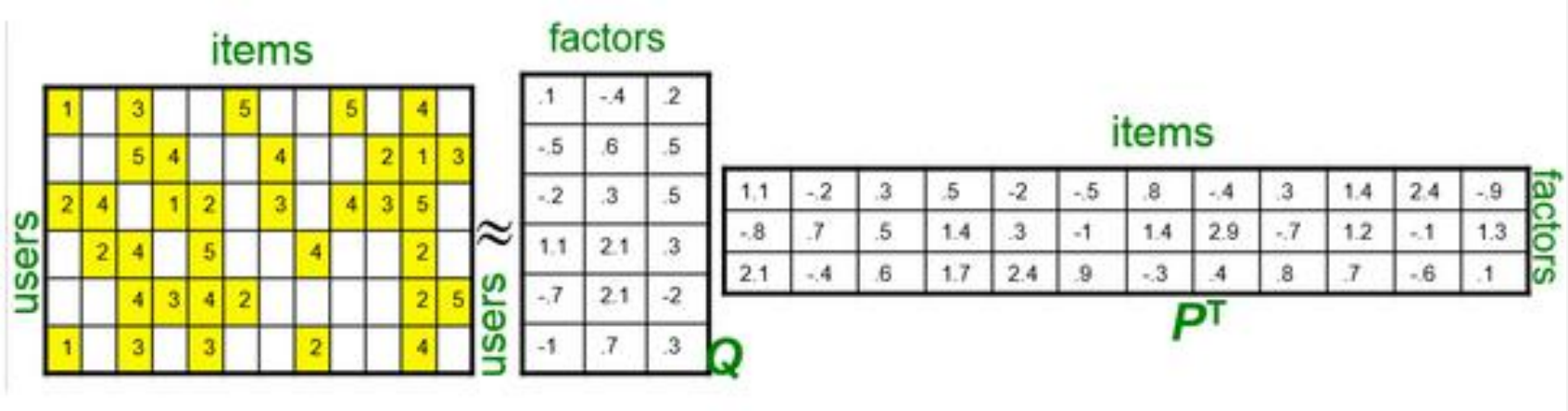
$$U_{m\times k}V^{T}_{n\times k}\approx A_{m\times n}$$

原始的矩阵：m个用户n件商品 → LFM矩阵：m个用户k个兴趣点

兴趣点--商品关系矩阵

8.2.2 潜在因子模型（LFM）

方法：矩阵分解，（通常的选择：SVD奇异值分解）



- 如果评分矩阵非空：SVD分解
- 否则: 回归问题的框架：1）损失函数；2）解算方法：随机梯度下降；

Funk-SVD $$\min \sum_{u,i} (r_{ui} - p_u q_i^T)^2 + \lambda(\|p_u\| + \|q_i\|)$$

NMF $$\min \sum_{u,i} (r_{ui} - p_u q_i^T)^2$$ p,q 非负

8.2.2 潜在因子模型（LFM）

单曲循环=5, 分享=4, 收藏=3, 主动播放=2 , 听完=1, 跳过=-2 , 拉黑=-5，在分析时能获得的实际评分矩阵R。也就是输入矩阵大概是这个样子

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

$$\min \sum_{u,i} (r_{ui} - p_u q_i^T)^2 + \lambda(\|p_u\| + \|q_i\|)$$

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1		2.10	2.08	2.12	1.96		2.12	**2.16**			2.05		
用户2	2.03	2.03	2.01		1.89	2.00	2.04	2.08		**2.10**	1.98	2.08	
用户3	1.78	1.78		1.80				1.83	1.82			**1.83**	
用户4	1.98				1.84	1.95		2.03		**2.05**	1.93		1.95
用户5	1.96			1.98		1.93			2.00	**2.04**		2.01	1.93
用户6	2.05	2.04		2.06	1.90		2.06	2.10		**2.12**	2.00	2.10	2.02
用户7	2.02		2.00	2.03	1.87		2.03	**2.07**	2.05		1.96		
用户8	2.03		2.01	2.05	1.89		2.04		2.07		1.98	2.09	

8.2.2 潜在因子模型（LFM）

1（科幻、惊悚）	3（犯罪）	4（家庭）	5（恐怖、惊悚）
《隐形人》（The Invisible Man，1933）	《大白鲨》（Jaws，1975）	《101真狗》(101 Dalmatians，1996）	《女巫布莱尔》（The Blair Witch Project，1999）
《科学怪人大战狼人》（Frankenstein Meets the Wolf Man，1943）	《致命武器》（Lethal Weapon，1987）	《回到未来》（Back to the Future，1985）	《地狱来的房客》（Pacific Heights，1990）
《哥斯拉》（Godzilla，1954）	《全面回忆》（Total Recall，1990）	《土拨鼠之日》（Groundhog Day，1993）	《异灵骇客2之恶灵归来》（Stir of Echoes: The Homecoming，2007）
《星球大战3：武士复仇》（Star Wars: Episode VI - Return of the Jedi，1983）	《落水狗》（Reservoir Dogs，1992）	《泰山》（Tarzan，2003）	《航越地平线》（Dead Calm，1989）
《终结者》（The Terminator，1984）	《忠奸人》（Donnie Brasco，1997）	《猫儿历险记》（The Aristocats，1970）	《幻象》（Phantasm，1979）
《魔童村》（Village of the Damned，1995）	《亡命天涯》（The Fugitive，1993）	《森林王子2》（The Jungle Book 2，2003）	《断头谷》（Sleepy Hollow，1999）
《异形》（Alien，1979）	《夺宝奇兵3》（Indiana Jones and the Last Crusade，1989）	《当哈利遇到莎莉》（When Harry Met Sally…，1989）	《老师不是人》（The Faculty，1998）
《异形2》（Aliens，1986）	《威胁2 :社会》（Menace II	《蚁哥正传》（Antz，1998）	《苍蝇》（The Fly，1958）

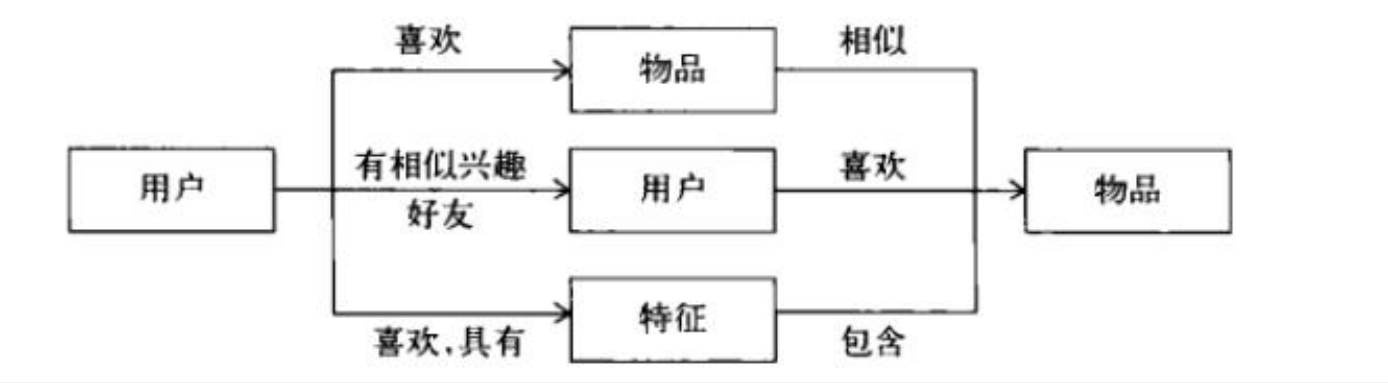
8.3 基于内容的推荐

- 最重要的工作：物品内容与用户内容的提取
- 8.3.1 基于标签的方法（人工提供物品内容的抽象）
- 8.3.2 特征提取与嵌入（自动获取物品内容的抽象）
- 8.3.3 学习用户画像（用户的抽象）

8.3.1 基于标签的推荐

- 标签是一种无层次化结构的、用来描述信息的关键词，它可以用来描述物品的语义。根据给物品打标签的人的不同，标签应用一般分为两种：

1. 一种是让作者或者专家给物品打标签；
2. 另一种是让普通用户给物品打标签，也就是UGC（User Generated Content，用户生成的内容）的标签应用。



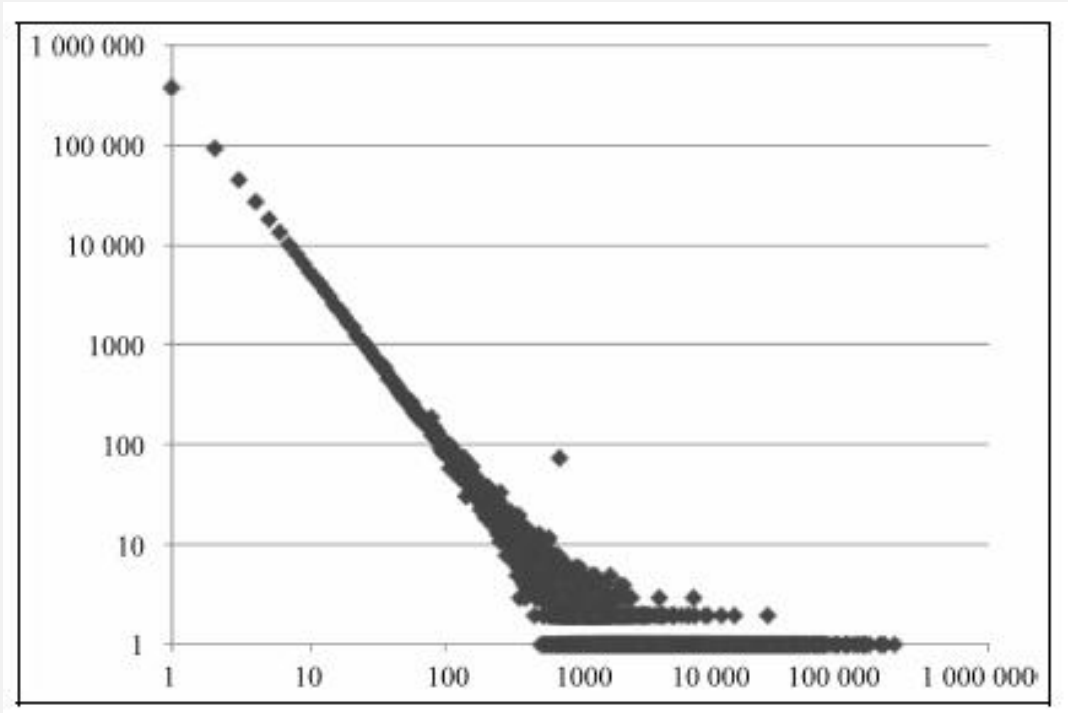
$$p(u,i)=\sum_{b} n_{n,b} n_{b,i}$$

8.3.1 基于标签的推荐

用户、物品、标签、记录的数据量对比

	用户数	物品数	标签数	记录数
Delicious	11 200	8791	42 233	405 665
CiteULike	12 466	7318	23 068	409 220

标签也有长尾分布



8.3.1 基于标签的推荐

- 方便用户输入标签：让用户从键盘输入标签无疑会增加用户打标签的难度，这样很多用户不愿意给物品打标签，因此我们需要一个辅助工具来减小用户打标签的难度，从而提高用户打标签的参与度。
- 提高标签质量：同一个语义不同的用户可能用不同的词语来表示。这些同义词会使标签的词表变得很庞大，而且会使计算相似度不太准确。而使用推荐标签时，我们可以对词表进行选择，首先保证词表不出现太多的同义词，同时保证出现的词都是一些比较热门的、有代表性的词。

推荐策略：

1. 给用户u推荐整个系统里最热门的标签
2. 给用户u推荐物品i上最热门的标签
3. 以加权的同时推荐1、2

8.3.1 基于标签的推荐

■问题：如何从评论中抽取得到标签（特征）

方法：TF-IDF $$TF-IDF = 词频(TF) \times 逆文档频率(IDF)$$

1. TF（term frequency）：

$$词频(TF) = \frac{某个词在文章中的出现次数}{文章的总词数}$$

例文《中国的蜜蜂养殖》，假定该文长度为1000个词，"中国"、"蜜蜂"、"养殖"各出现20次，则这三个词的"词频"（TF）都为0.02。全网数据如下：

	包含该词的文档数（亿）	IDF	TF-IDF
中国	62.3	0.603	0.0121
蜜蜂	0.484	2.713	0.0543
养殖	0.973	2.410	0.0482

2. IDF（Inverse Document Frequency）：

第i个词的独特程度

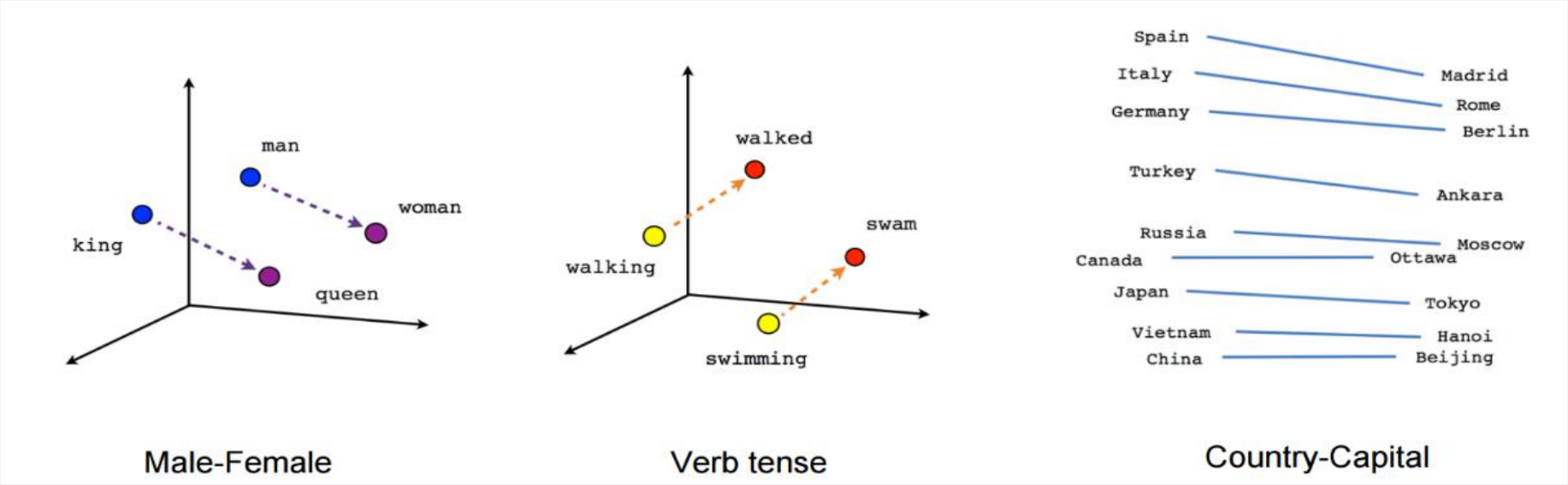
$$逆文档频率(IDF) = \log(\frac{语料库的文档总数}{包含该词的文档数+1})$$

8.3.1 基于标签的推荐

■ TF-IDF算法实现简单快速，但是仍有许多不足之处：

- （1）没有考虑特征词的位置因素对文本的区分度，词条出现在文档的不同位置时，对区分度的贡献不一样。
- （2）按照传统TF-IDF，一些生僻词的IDF(反文档频率)会比较高、因此这些生僻词常会被误认为是文档关键词。
- （3）传统TF-IDF中的IDF部分只考虑了特征词与它出现的文本数之间的关系，而忽略了特征词不同的类别间的分布情况。
- （4）对于文档中出现次数较少的重要人名、地名信息提取效果不佳。（实体词）

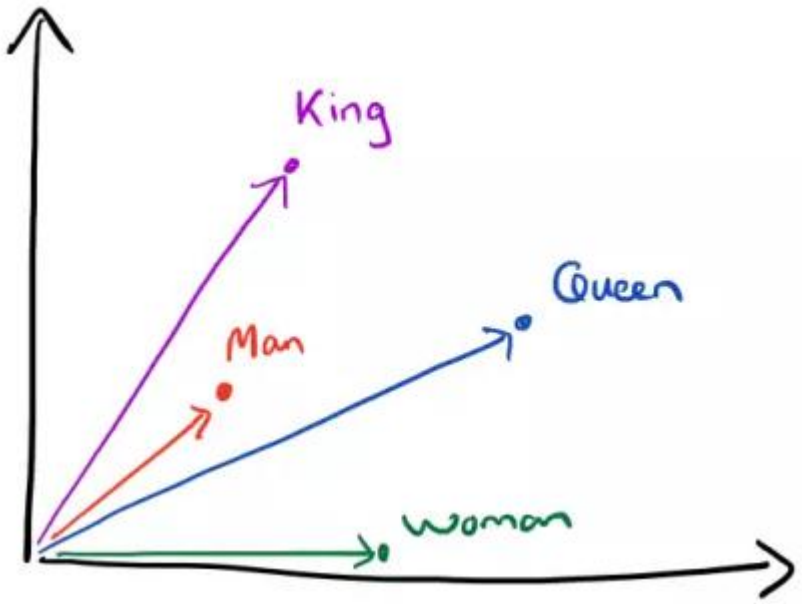
■ 词的出现次数不能反映词的语义



8.3.2 嵌入 Embeddings

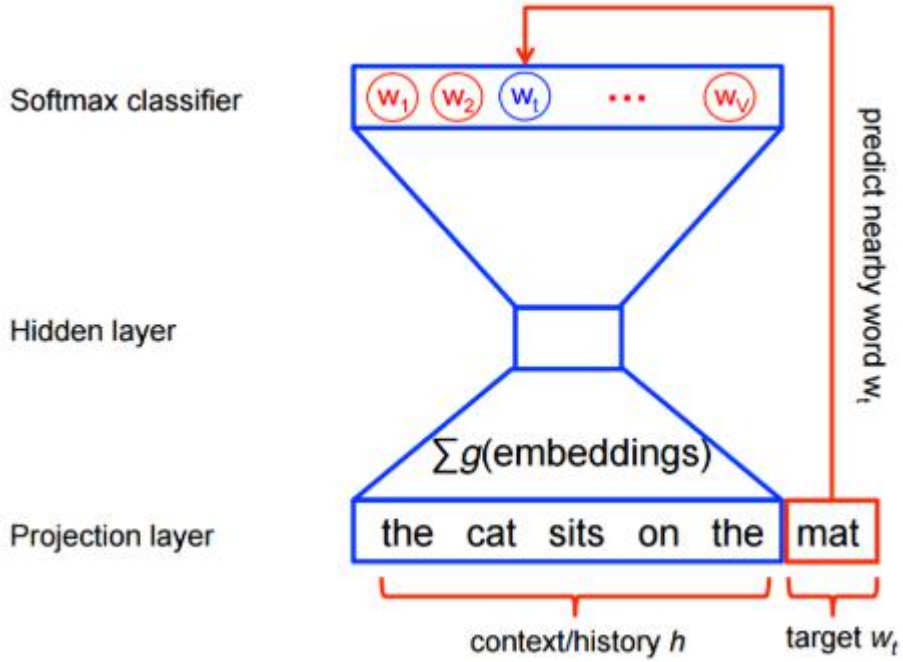
- 假如没有大量用户数据/没有预制标签/想听冷门歌曲，
- 网易云音乐：“从音乐本身寻找答案”

- Word2Vec 以下介绍来自TensorFlow文档
- 自然语言处理系统通常将词汇作为离散的单一符号，例如 "cat" 一词或可表示为 Id537，而 "dog" 一词或可表示为 Id143。这些符号编码毫无规律，无法提供不同词汇之间可能存在的关联信息。换句话说，在处理关于 "dogs" 一词的信息时，模型将无法利用已知的关于 "cats" 的信息（例如，它们都是动物，有四条腿，可作为宠物等等）。可见，将词汇表达为上述的独立离散符号将进一步导致数据稀疏，使我们在训练统计模型时不得不寻求更多的数据。而词汇的向量表示将克服上述的难题。



8.3.2 嵌入 Embeddings

- 向量空间模型 (VSMs)将词汇表达（嵌套）于一个连续的向量空间中，语义近似的词汇被映射为相邻的数据点。向量空间模型在自然语言处理领域中有着漫长且丰富的历史，不过几乎所有利用这一模型的方法都依赖于分布式假设，其核心思想为出现于相似上下文情景中的词汇都有相类似的语义。



$$J_{\text{NEG}} = \log Q_\theta(D=1|w_t,h) + k \mathop{\mathbb{E}}_{\tilde{w}\sim P_{\text{noise}}} [\log Q_\theta(D=0|\tilde{w},h)]$$

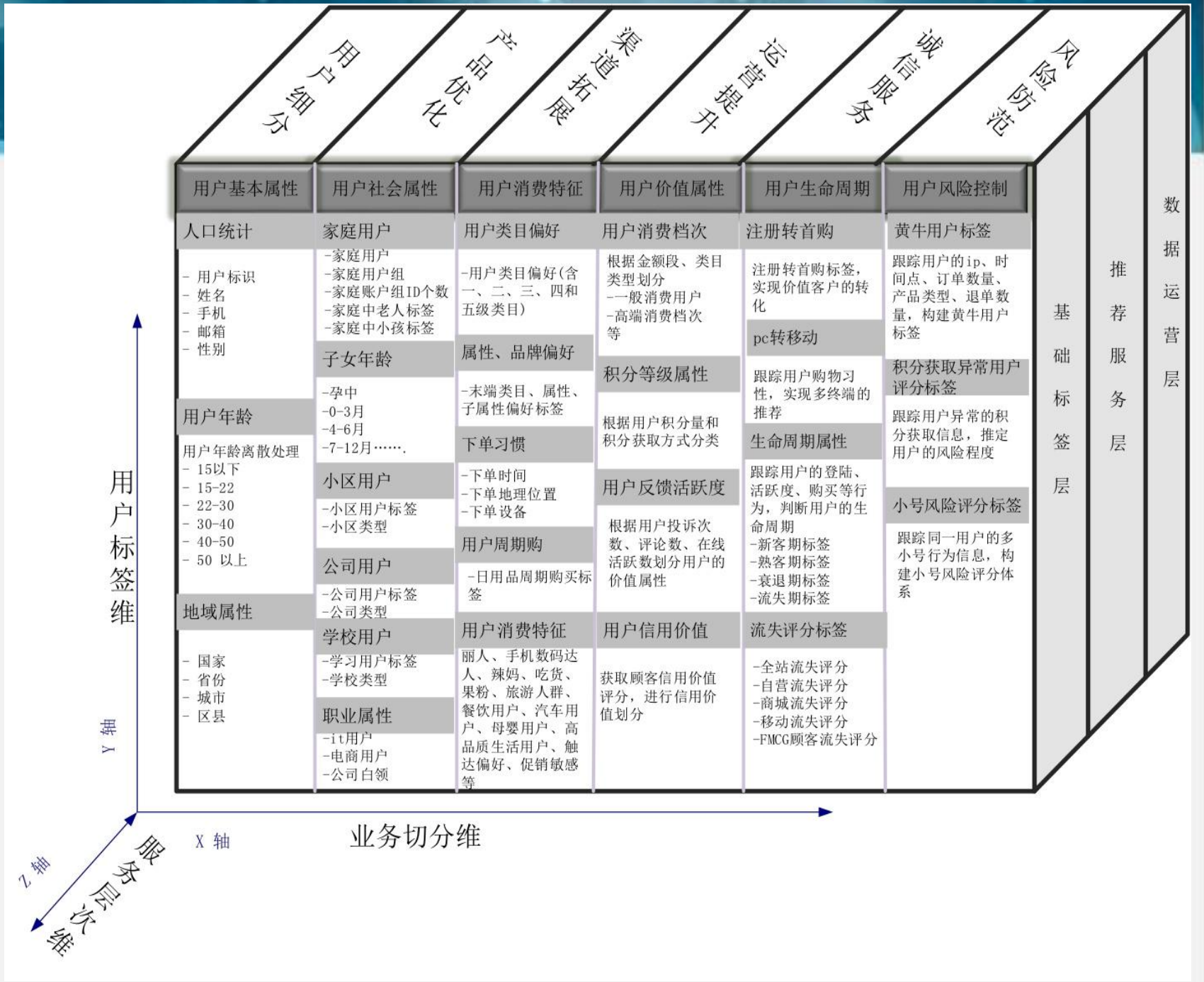
“the quick brown fox jumped over the lazy dog”

([the, brown], quick),
([quick, fox], brown),
([brown, jumped], fox), ...

8.3.3 用户画像

- User persona 往往多应用于产品定义与设计阶段。

- User profile
  - ➢ 为产品用户体验的优化提供方向；
  - ➢ 挖掘用户数据，进行个性化推送；
  - ➢ 衡量用户价值（ARUP值）。



8.3.3 用户画像

■利用数据模型对用户行为数据进行建模与学习，由此产出多级化的用户标签，进而进行用户偏好与行为判断。

模型举例	数据来源	用途举例
是否怀孕模型	关注的话题、公众号、购买记录、消费记录、（就诊记录）	推销婴幼儿产品
文艺青年模型	用户的社交媒体发言、评论数据	媒体产品推荐
客户身高模型	用户购买服装的记录	商品推荐
客户价值模型	消费记录、（工资记录）	判断违约率、贷款额度


```
{
    "ID": 123456,
    "姓名": "王建国",
    "性别": "男",
    "出生年月": 19990909,
    "籍贯": "上海",
    "居住地": "北京",
    "教育背景": {
        "学校":"吉利大学",
        "专业": "中文",
    }

。。。
}
```

8.3.3 用户画像

百度为您找到相关结果约34,700,000个　　搜索工具

Convertlab用户画像_用户数据分析整合_洞察用户全貌

用户画像系统,如何做用户画像,Convertlab用户画像,基于用户画像,实现用户画像,全渠道自有用户数据管理分析工具,描绘完整用户画像，建立标签体系...

host.convertlab.com 2020-05 - 评价 广告

个推用户画像_基于个推大数据_轻松读懂用户

APP开发者可通过个推用户画像 分析用户属性与兴趣偏好,实时判断线下场景.个推用户画像 维度多元,标签丰富,数据接口灵活,可全面实现精细化,场景化运营.

个性化标签定制　满足多样运营需求

新用户冷启动　初始内容精准推荐

换机用户识别　提升用户运营效率

指导流失召回　有效盘活存量用户

查看更多相关信息>>

www.getui.com 2020-05 - 评价 广告

用户画像_百度统计数据

用户画像,产品用户画像,帮助企业洞悉行业竞对,简化投放流程,使广告投放更简单,营销更轻松,统计分析云,依靠百度大数据能力,助企业精准刻画典型用户特征，实现用户转化与..

tongji.baidu.com 2020-05 - 评价 广告

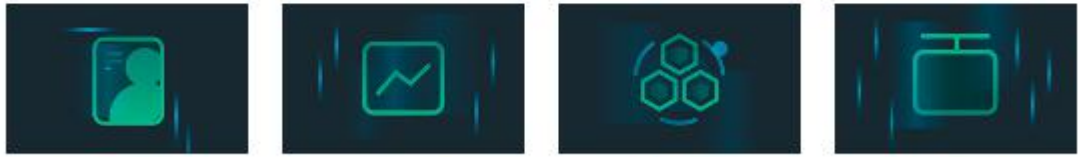
用户画像_数据分析平台

用户画像,用户标签,神策数据,人群画像,深度洞察用户数据,可实现多维度,精细化的统计分析,秒级处理,实时更新，支持私有化部署的数据分析产品.

www.sensorsdata.cn 2020-05 - 评价 广告

用户画像分析_【微盟智营销】丰富标签_精准洞察

用户画像,「微盟智营销」结合年龄段,消费水平等用户属性,近期兴趣偏好特征和实时的线下场景,多指标多维度构建有效精准的用户画像,全面实现精细化,场景化运营.



数据整合　多方客户数据沉淀

精细标签　360度客户全景画像

精细运营　客户生命周期管理

自动营销　智能营销自动执行

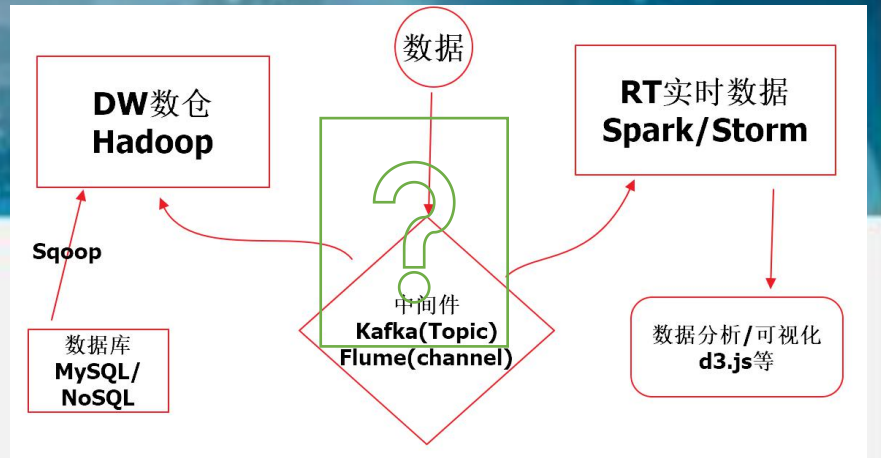
查看更多相关信息>>

www.weimob.com 2020-05 - 评价 广告

8.3.3 用户画像


个推
GeTui
股票代码
300766
开发者服务
效果广告服务
数据解决方案
数据报告
关于我们
登录
注册
捕捉正确的“场景”，找到对的“人”
对用户线上和线下行为深度洞察，构建全面、精准、多维的用户画像体系，为APP
提供丰富的用户画像数据以及实时的场景识别能力，帮助APP全方位了解用户。
行为标签
近期活跃的应用
近期去过的场景
场景标签
机场，商圈，电影院
景区，自定义场景等
属性标签
性别，年龄层次，消费水
平，职业等
兴趣标签
购物，教育，影音，游
戏，金融理财等
定制化标签
定制化标签 A，B，C 等

8.* 数据中台软件枚举



- 收集数据->数据特征提取->构建模型->标签获取->可视化
- *来自Ali数据中台团队

采集与传输层	存储层	计算层（离线）	计算层（在线）	数据服务层
• Sqoop • DataX • Flume • Logstash • Kafka • RocketMQ	• HDFS • HBase • Redis • Ceph	• Hive • Spark • MaxCompute • CDH（Cloudera）	• Storm/Jstorm • Flink • Spark Streaming	• Kylin • Druid • Presto • Lucene • ElasticSearch • Solr

当前典型架构：
离线计算：应用系统日志 -> flume -> kafka -> hdfs -> MapReduce作业
实时计算：应用系统日志 -> flume -> kafka -> blink/jstorm/storm/spark streaming

8.* 数据中台软件枚举

■ 采集&传输层

- **Sqoop**：Hadoop、关系型数据库之间传输数据的工具。传输时，会启动多个MR作业并发的传输数据
- DataX：阿里巴巴开源的数据同步工具，用来在各种异构数据源之间同步数据。比如 RDBMS<->Hadoop/MaxCompute、RDBMS<->hbase/ftp等等。部署、运维非常简单，将DataX的jar包copy到linux系统中即可运行；
- **Flume**：分布式的高可用的数据收集、聚集的工具。通常用于从其他系统搜集数据，如web服务器产生的日志，结合Kafka的消息队列功能，实现实时日志处理、离线日志投递；
- Logstash：服务器端数据收集工具，能够同时从多个来源采集、转换数据。日志收集功能与Flume比较类似；
- **Kafka**：基于发布/订阅机制的分布式的消息系统。常用于日志投递、分发场景；
- RocketMQ：阿里巴巴开源的消息队列工具。经过了双11场景的洗礼，稳定性、可靠性非常好；

8.* 数据中台软件枚举

■ 存储层

- **HDFS**：Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件上的分布式文件系统。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS放宽了一部分POSIX约束，来实现流式读取数据文件；

- **HBase**：Hbase是分布式、KV查询的开源数据库。HDFS为Hbase提供可靠的底层数据存储服务，MapReduce为Hbase提供高性能的计算能力，Zookeeper为Hbase提供稳定服务和Failover机制，LSM数据存储格式提供了高性能读写能力；

- **Redis**：Redis是key-value存储系统。采用ANSI C语言编写、遵守BSD协议、支持网络、可基于内存亦可持久化的日志格式，并提供多种语言的API。提供了哈希(Hash), 列表(list), 集合(sets) 和 有序集合(sorted sets)等数据结构

- Ceph：开源分布式存储系统，提供了块储存RDB、分布式文件储存Ceph FS、以及分布式对象存储Radosgw三大储存功能，是目前为数不多的集各种存储能力于一身的开源存储中间件

8.* 数据中台软件枚举

■ 计算层：离线计算

- **Hive**：Hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务进行运行。 其优点是学习成本低，可以通过类SQL语句快速实现简单的MapReduce统计，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。是事实上的离线数据仓库标准。

- **Spark**：Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark是UC Berkeley AMP lab (加州大学伯克利分校的AMP实验室)所开源的类Hadoop MapReduce的通用并行框架，Spark，拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是——Job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。

- MaxCompute：阿里巴巴开发，基于MR原理的大数据处理平台，已经通过阿里云对外输出，是一种快速、完全托管的TB/PB级数据仓库解决方案。

- CDH：CDH是Cloudera的软件发行版，包含Apache Hadoop及相关项目。所有组件都是100%的开源（Apache许可证）。

8.* 数据中台软件枚举

■ 计算层：实时计算

- **Storm**/Jstorm：分布式的、高容错的实时计算系统，2014年以前应用非常广泛，近几年初步被其他流计算产品替代。

- **Flink**：Flink是一个低延迟、高吞吐、统一的大数据计算引擎。在阿里巴巴的生产环境中，Flink的计算平台可以实现毫秒级的延迟情况下，每秒钟处理上亿次的消息或者事件。同时Flink提供了一个Exactly-once的一致性语义。保证了数据的正确性。这样就使得Flink大数据引擎可以提供金融级的数据处理能力。

- **Spark Streaming**：Spark Streaming 类似于 Apache Storm，是一个流计算处理框架。Spark Streaming 有高吞吐量和容错能力强这两个特点。

8.* 数据中台软件枚举

■ 数据服务层

- Kylin：开源的分布式分析引擎，提供Hadoop/Spark之上的SQL查询接口及多维分析（OLAP）能力以支持超大规模数据。核心原理是数据预计算，利用空间换时间来加速查询模式固定的OLAP查询。最新的版本已经支持了实时数据导入。
- Druid：Druid也是一款非常流行的OLAP引擎，基于MPP(Massively Parallel Processing)架构，采用了 预聚合、列式存储、字典编码、位图索引 4个方法，加速查询性能。 截止2019年9月22日，Druid原生不支持数据精确去重功能。快手已经将Druid应用于生产环境。
- Presto：Presto是一个开源的分布式SQL查询引擎，适用于交互式分析查询，数据量支持GB到PB字节。Presto的设计和编写完全是为了解决像Facebook这样规模的商业数据仓库的交互式分析和处理速度的问题。
- Lucene：Lucene 是一个基于Java 的全文信息检索工具包，目前主流的搜索系统Elasticsearch和solr都是基于lucene的索引和搜索能力进行。
- ElasticSearch：基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎。
- Solr：Solr是Apache Lucene项目的开源企业搜索平台。其主要功能包括全文检索、命中标示、分面搜索、动态聚类、数据库集成，以及富文本的处理。Solr是高度可扩展的，并提供了分布式搜索和索引复制。Solr是最流行的企业级搜索引擎，Solr 4还增加了NoSQL支持。
- Palo：百度开源的OLAP引擎，在百度内部使用比较广泛。基于MPP架构，集成了Google Mesa、Cloudera Impala。

8.4 基于上下文的推荐

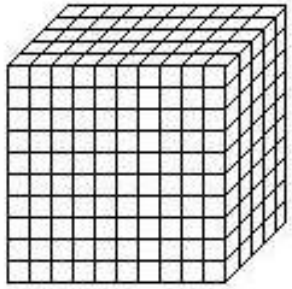
- 8.4.1 通用的基于上下文的推荐
- 8.4.2 基于时间的推荐
- 8.4.3 基于地理位置的推荐

8.4 基于上下文的推荐

■多维模型

$$\text{User} \times \text{Item} \rightarrow \text{Rating}$$

$$\text{User} \times \text{Item} \times \text{Context} \rightarrow \text{Rating}$$



- 上下文可以不止一项；
- 上下文的选择：时间、地点、天气等独立于用户与物品的属性

■多维方法

- U和C看成一个实体；
- I和C看成一个实体；
- U和I看成一个实体

8.4.3与时间有关的推荐方法

■时间协同过滤

- 基于新近的模型：越新的评分越重要；手段：遗忘因子权重/移动窗口
- 周期的基于上下文的模型：周末/平日，春节/平日数据，分开建模；
- 把时间当作独立变量的模型：将评分看作时间的函数；time-SVD++

■序列模式挖掘（频繁模式挖掘带时序的版本）

8.4.2与位置有关的推荐

■偏好位置（位置与用户关联）：不同地区的用户有不同的喜好；

■旅行位置（位置与物品关联）：因为物品的物理属性限制了消费的地点；

- LARS方法：

  假设有一个来自浙江-杭州-西湖区的用户。

  - 首先根据所有用户的行为利用某种推荐算法（假设是ItemCF）给他生成推荐列表；
  - 然后利用浙江省用户的行为给他生成第二个推荐列表；
  - 用杭州市所有用户行为给该用户生成第三个推荐列表；
  - 利用西湖区用户行为给该用户生成第四个推荐列表。

  按照一定的权重将这4个推荐列表线性相加，从而得到给该用户的最终推荐列表。

- 旅行距离惩罚项

  权重定义：RecScore(u,i) = P(u,i) - TravelPenalty(u,i)

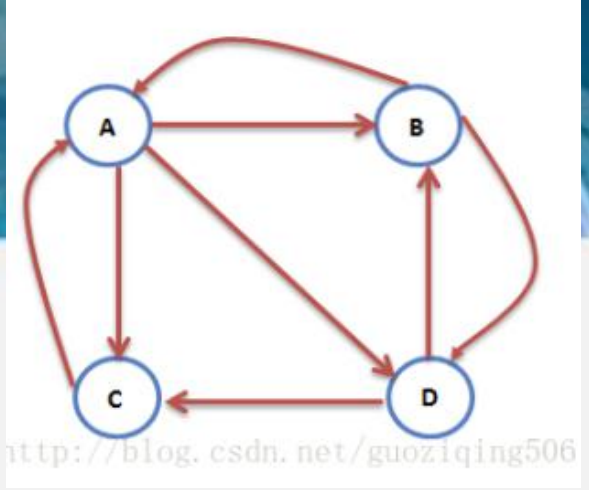
8.5 推荐系统中的其他问题

- 8.5.1 集成方法
- 8.5.2 结构化网络中的推荐
- 8.5.3 点击率预估

8.5.1 集成方法

- 1. 加权型：几个推荐系统得到的分数加权整合成最后的得分；
- 2. 切换型：在不同情境下切换使用不同推荐系统的结果；（例如冷启动等）
- 3. 级联型：后一级推荐算法在前一级的结果上进行优化；
- 4. 特征放大型/元级型：前一级作为后一级的特征使用；
- 5. 特征组合型：不同数据来源组合到一起；
- 6. 交叉型：推荐的结果放到一起进行展示；

8.5.2 结构化网络中的推荐



- 结构化网络本身包含了排序信息：PageRank
- 问题：如何个性化PageRank（对用户感兴趣部分的排序）
- 方法：个性化PageRank/主题敏感PageRank（Topic Sensitive PageRank）

page	A	B	C	D
A	0	1/2	1	0
B	1/3	0	0	1/2
C	1/3	0	0	1/2
D	1/3	1/2	0	0

$$V_1 = MV_0 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

$$V_i = MV_{i-1}$$

- 个性化PageRank与PageRank的区别：随机行走中的跳转行为。
- PageRank：

V= (1 - c)MV + cu，u所有网络概率均等

- Personal PageRank

V= (1 - c)MV + cp，p：用户感兴趣的网页

8.5.3 点击率预估
Click-Through Rate Prediction

程序化广告交易：根据用户行为及时投放在线广告；
目标：把最合适的广告找出来去呈现给最合适的用户

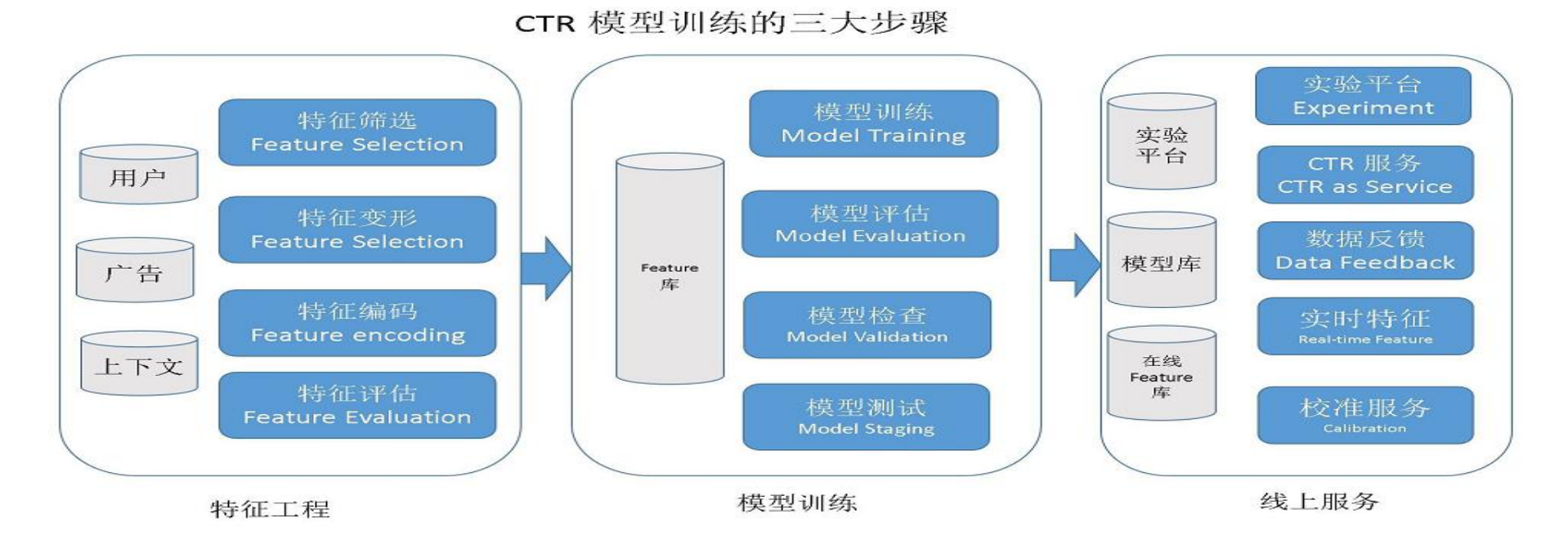
- （1）DSP（Demand-Side Platform）：需求方平台，是为广告主、代理商提供一个综合性的管理平台，通过统一界面管理多个数字广告和数据交换账户。
- （2）SSP（Sell-Side Platform）：即供应方平台，媒体流量主。常见平台有百度SSP、360SSP。
- （3）RTB（RealTime Bidding）：实时竞价，是一种利用第三方技术在数以百万计的网站上针对每一个用户展示行为进行评估以及出价的竞价技术，是一种技术方法。
- （4）DMP(Data-Management Platform)：数据管理平台，是把分散的多方数据进行整合纳入统一的技术平台，并对这些数据进行标准化和细分，让用户可以把这些细分结果推向现有的互动营销环境里的平台。

8.5.3 点击率预估
Click-Through Rate Prediction

■ 为什么要点击率预估？

- 程序化投放【内容来自互联网】：比如金主爸爸A刚好有一个新品要推广，刚好又有钱，可以先直接在媒体大哥666的DSP平台上注册账户充值金额，然后选择要投放的用户群以及要去竞价的金额。这样就可以直接在媒体大哥666收到了钱同时广告素材等通过了审核，此时广告位就可以通过RTB竞价，如果金主爸爸的在这次竞价中赢得了这个广告位，那么就可以在这个广告位中进行广告展示。

■ 点击率预估的目标：给出此次投放广告的价值，作为参与竞价的依据。



目前典型的技术：GBDT+LR

请回顾Chapter 5

没有标签数据怎么办？

强化学习技术

设计推荐系统的经验教训

- (1) 确定你真的需要推荐系统。推荐系统只有在用户遇到信息过载时才必要。如果你的网站物品不太多，或者用户兴趣都比较单一，那么也许并不需要推荐系统。所以不要纠结于推荐系统这个词，不要为了做推荐系统而做推荐系统，而是应该从用户的角度出发，设计出能够真正帮助用户发现内容的系统，无论这个系统算法是否复杂，只要能够真正帮助用户，就是一个好的系统。
- (2) 确定商业目标和用户满意度之间的关系。对用户好的推荐系统不代表商业上有用的推荐系统，因此要首先确定用户满意的推荐系统和商业上需求的差距。一般来说，有些时候用户满意和商业需求并不吻合。但是一般情况下，用户满意度总是符合企业的长期利益，因此这一条的主要观点是要平衡企业的长期利益和短期利益之间的关系。
- (3) 平衡数据和算法之间的关系。使用正确的用户数据对推荐系统至关重要。对用户行为数据的深刻理解是设计好推荐系统的必要条件，因此分析数据是设计系统中最重要的部分。数据分析决定了如何设计模型，而算法只是决定了最终如何优化模型。
- (4) 找到相关的物品很容易，但是何时以何种方式将它们展现给用户是很困难的。不要为了推荐而推荐。
- (5) 不要浪费时间计算相似兴趣的用户，可以直接利用社会网络数据。
- (6) 需要不断地提升算法的扩展性。
- (7) 选择合适的用户反馈方式。

关于网易云音乐的报道

- 人工不可缺少除了算法推荐，在很大程度上，一个流媒体平台也一定会承担人工过滤职责，从产品及运营角度确立人工规则，筛除不符合条件的选项。沈博文告诉我们，他们不只是依赖算法，而是希望通过一些人工的力量，来补偿算法的一些不足。因此，除了有单独的算法团队，网易云音乐也有一个强大的编辑团队。一方面，他们帮助在一开始推荐内容上面做一层筛选，找出那些比较优质的内容，保证整一个推荐库的健康。而另一方面，他们也需要解决算法的一些收敛问题。
- “因为如果纯粹依靠算法推荐的话可能会对一些新内容响应较慢，我们也会用一些人工编辑的方式。去寻找出一些可能我们觉得非常优质的内容，然后推荐给大家”，沈博文表示。此外，即便客服系统在一定程度上依赖AI技术，但由网易云音乐客服部门与技术部门共同组成的“人工反馈小组”，才是让用户对网易云音乐好感up的重要原因很多“秒回”的技术解决方案还被用户戏称为“原来网易云音乐的小编真的是活的”。