法律声明

- 课程详情请咨询
  - 微信公众号：北风教育
  - 官方网址：http://www.ibeifeng.com/

人工智能之推荐系统

推荐系统(Recommender System)

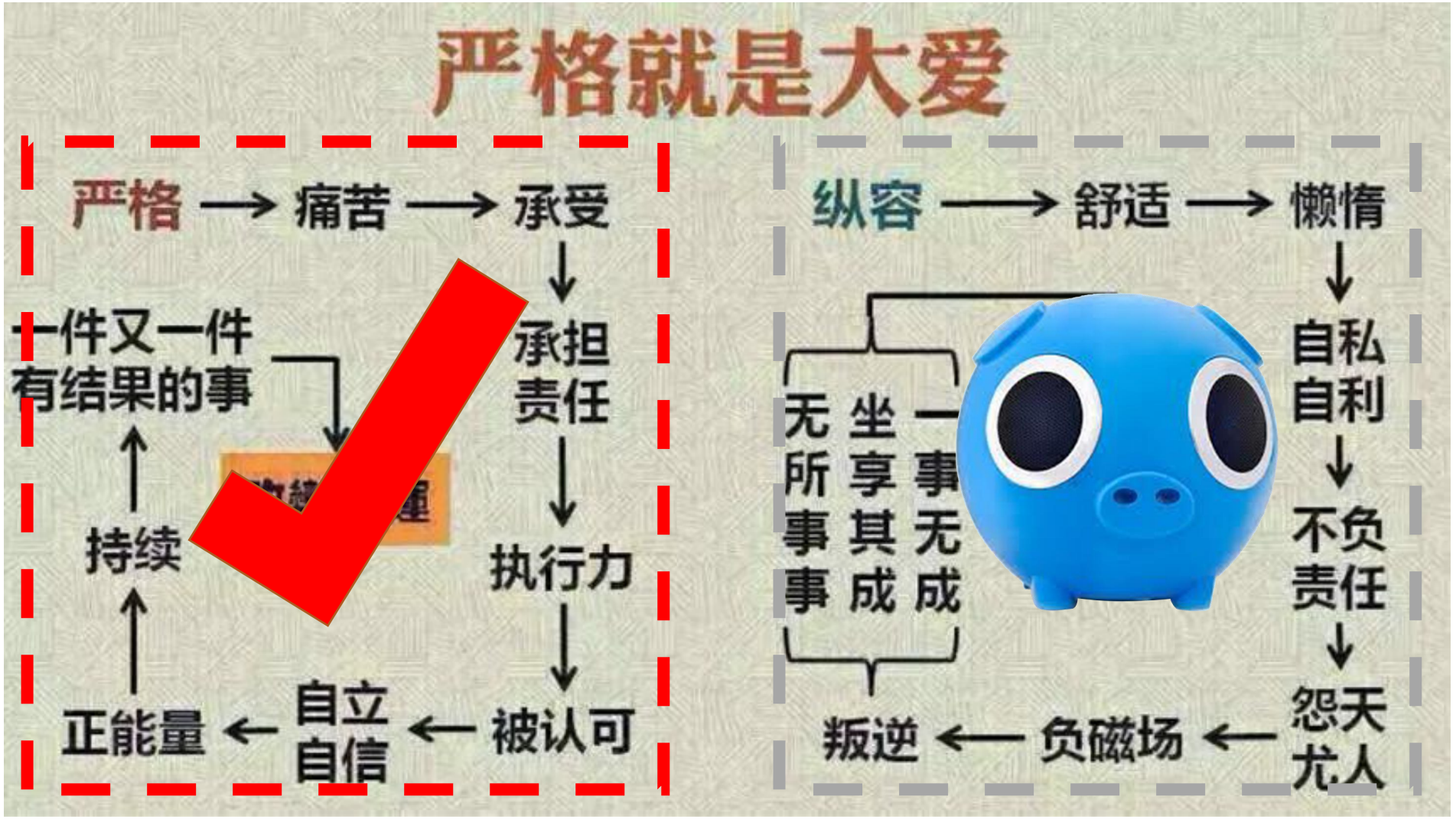
主讲人：Gerry

上海育创网络科技有限公司

课程要求

- 课上课下“九字”真言
  - 认真听，善摘录，勤思考
  - **多温故，乐实践**，再发散
- 四不原则
  - 不懒散惰性，不迟到早退
  - 不请假旷课，不拖延作业
- 一点注意事项
  - 违反“四不原则”，不包就业和推荐就业

严格是大爱


严格就是大爱
严格
痛苦
承受
承担责任
执行力
被认可
自立自信
正能量
持续
一件又一件有结果的事
纵容
舒适
懒惰
自私自利
不负责任
怨天尤人
负磁场
叛逆
一事无成
坐享其成
无所事事

寄语

做别人不愿做的事，

做别人不敢做的事，

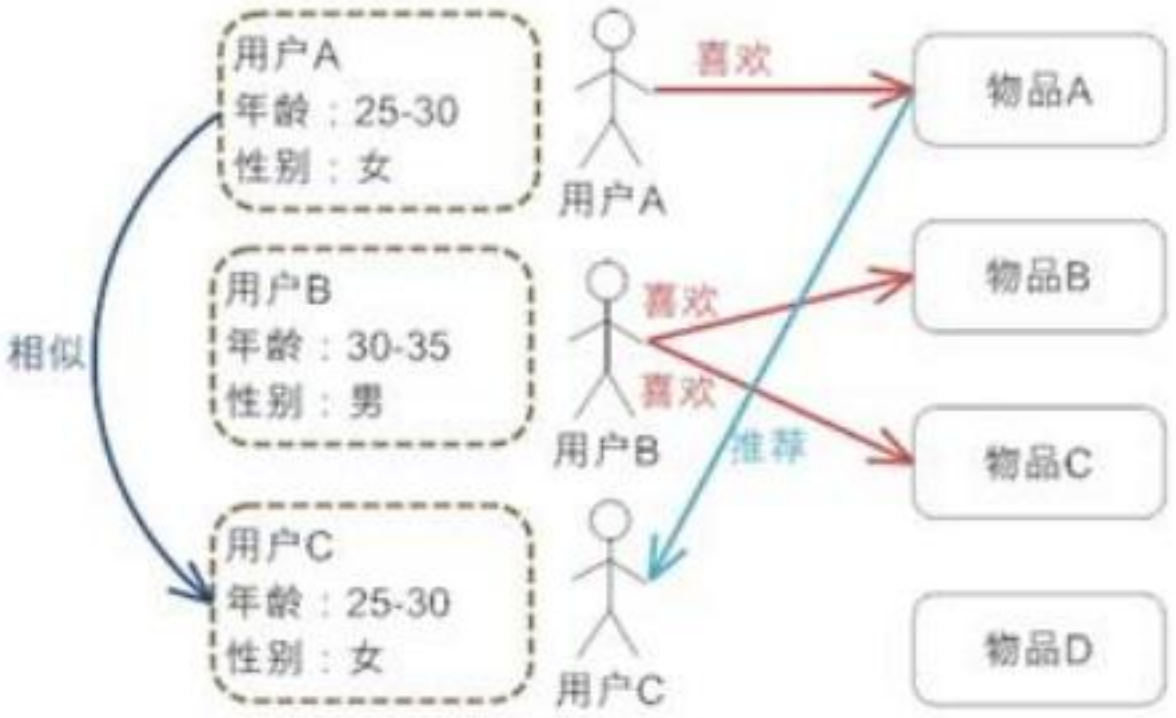
做别人做不到的事。

课程内容

- 基于内容的推荐
- 基于知识的推荐
- 混合推荐方法

Based Recommendation

- **基于人口统计学的推荐(Demographic-based Recommendation)**是一种根据系统用户的基本信息发现用户之间的相似程度，然后将相似用户喜爱的物品推荐给当前用户。



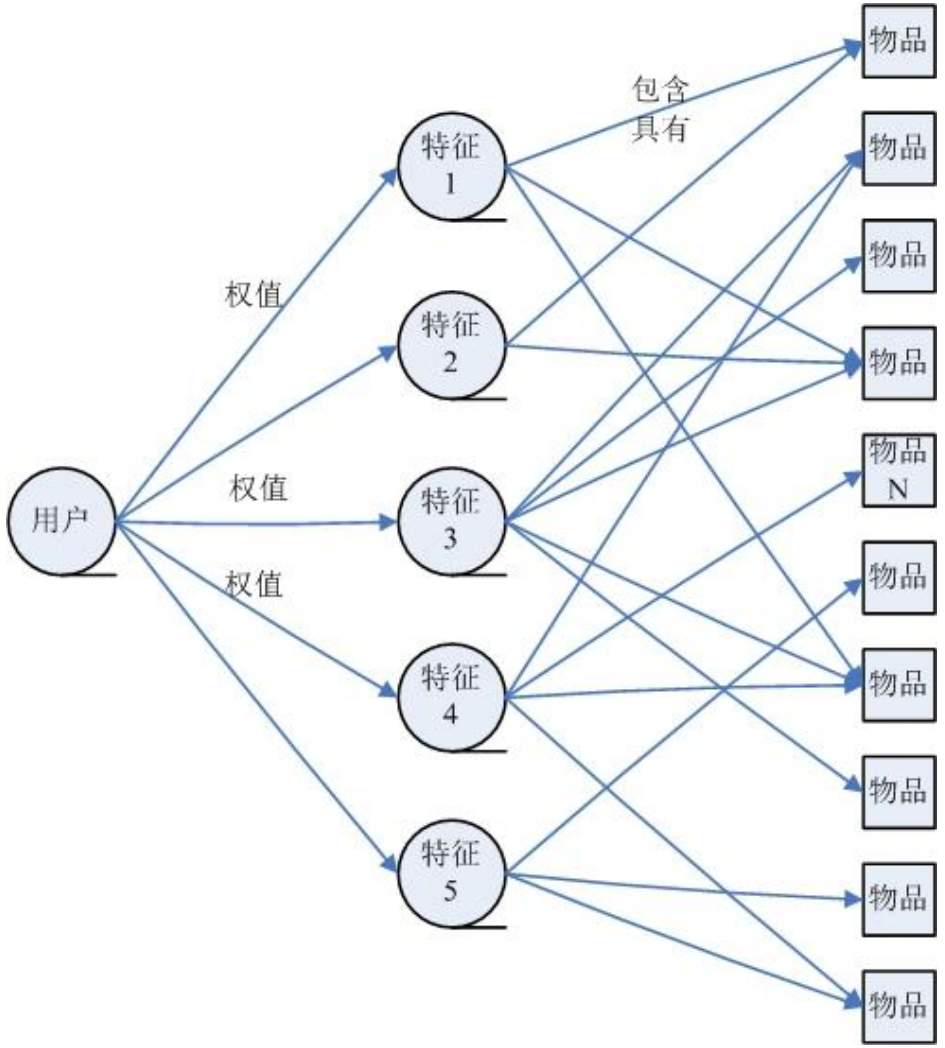
CB

- **基于内容的推荐算法**(Content-based Recommendations, CB)也是一种工业界应用比较广的一种推荐算法。由于协同过滤推荐算法中仅仅基于用户对于商品的评分进行推荐，所以有可能出现冷启动的问题，如果可以根据物品的特性和用户的特殊偏好等特征属性进行比较直观的推荐就可以解决这个冷启动的问题。
- CB算法虽然需要依赖物品和用户偏好的额外信息，但是不需要太多的用户评分或者群体记录，也就是说，就是只有一个用户也可以完成推荐功能，产生一个物品推荐列表。
- CB算法的初始设计的目标是推荐有意思的文本文档，现阶段也会将该算法应用到其它推荐领域中。

CB和CF的区别

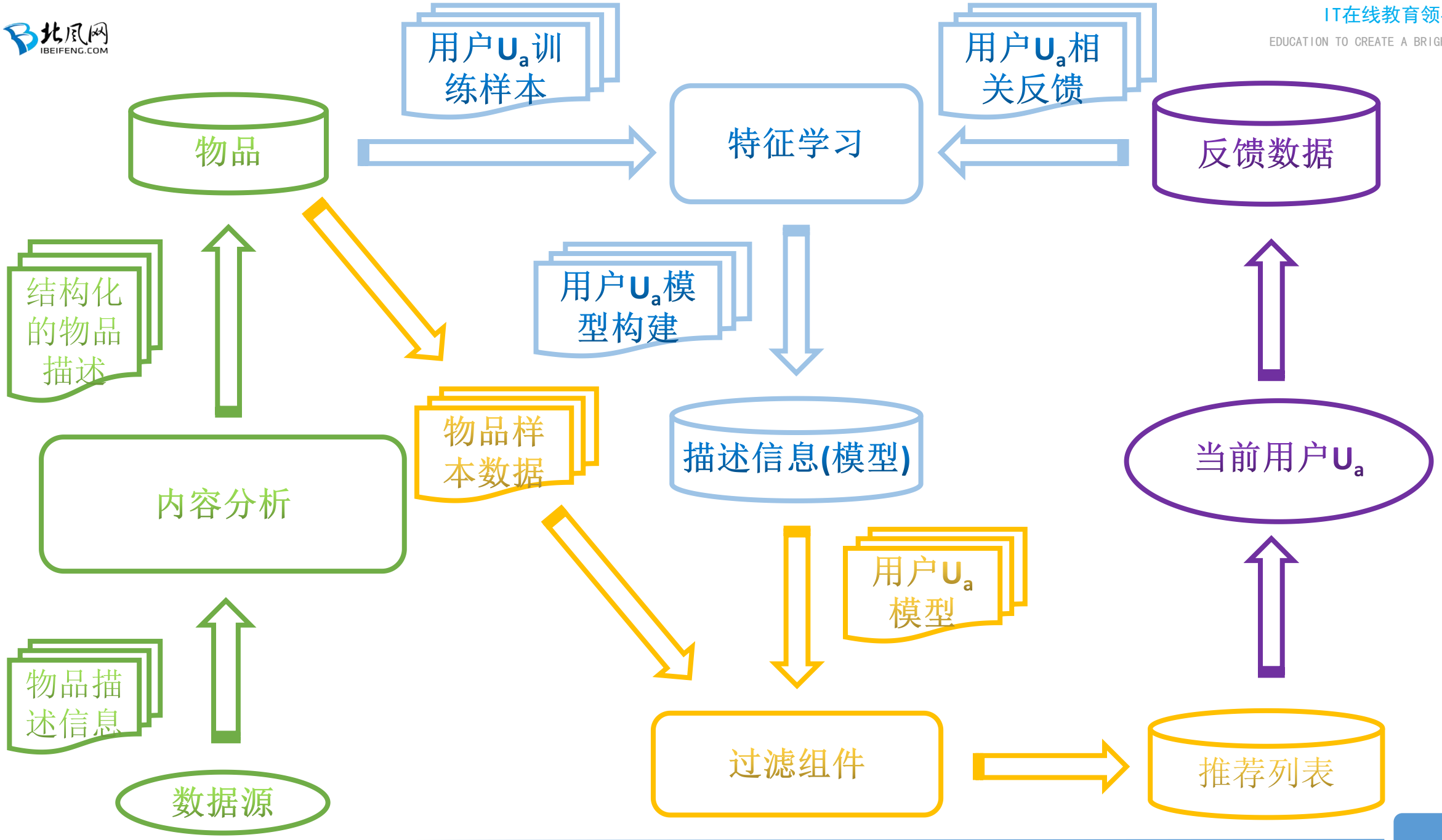
- CB(基于内容的推荐算法)的推荐系统会试图推荐给给定用户过去喜欢的相似物品。CB不需要用户-物品评分矩阵。
- CF(协同过滤的推荐算法)的推荐系统会试图识别出具有相同爱好的用户，并推荐他们喜欢过的物品。CF算法是基于用户-物品评分矩阵来进行推荐的。

CB结构原理



CB-结构

- CB算法主要包含三个步骤：
  - Item Representation： 为每一个item抽取一些特征属性出来，也就是结构化物品的描述操作，对应的处理过程叫做: Content Analyzer（内容分析）
  - Profile Learning：利用一个用户过去喜欢(不喜欢)的item特征数据，来学习该用户的喜好特征(profile)；对应的处理过程叫做：Profile Learning（特征学习）
  - Recommendation Generation：通过比较上一步得到的用户profile与特征item的特征，为此用户推荐一组相关性最大的item即可；对应的处理过程叫做：Filtering Component（过滤组件）


用户U_a训练样本
用户U_a相关反馈
物品
特征学习
反馈数据
结构化的物品描述
用户U_a模型构建
物品样本数据
描述信息(模型)
当前用户U_a
内容分析
用户U_a模型
物品描述信息
过滤组件
推荐列表
数据源

CB-Item Representation

- 对于物品特征属性的抽取类型机器学习中采用的方式，主要包括对数值型数据的处理和对非数值型数据的处理，主要处理方式如下：
  - 数值型数据归一化
  - 数值型数据二值化
  - 非数值型数据词袋法转换为特征向量
  - TF-IDF
  - Word2Vec
  - 深度学习

CB-Profile Learning

- 假设用户u对于一些item已经给出了喜好判断，喜欢其中的一部分item，不喜欢其中的另外一部分，那么该过程就是通过用户u过去的这些喜好判断，构建一个判别模型，最后可以根据这个模型判断用户u对于一个新的item是否会喜好。所以说这是一个比较典型的有监督学习问题，理论上可以使用机器学习的分类算法求解出所需要的判别模型。
- 常用的算法有：
  - 最近邻方法(K-Nearest Neighbor, KNN)
  - 决策树算法(Decision Tree, DT)
  - 线下分类算法(Linear Classifer, LC)
  - 朴素贝叶斯算法(Naive Bayes, NB)

CB-优缺点

- 优点：
  - 用户独立性： 在构建模型的过程中，仅仅只需要考虑当前用户信息即可
  - 透明度：通过显示地列出使得物品出现在推荐列表中的内容特征或者描述，可以比较明确的解释推荐系统是如何工作的。
  - 新物品：在没有任何评分的情况下，也可以进行推荐。
- 缺点：
  - 可分析的内容有限/特征抽取比较难：与推荐对象相关的特征数量和类型上是有限制的，而且依赖于领域知识。
  - 无法发现用户的潜在兴趣/过度特化：由于CB中的推荐结果是和该用户以前喜欢的item类似的，所以如果一个用户在一个网站仅仅只对一个item表达出喜欢的情绪，那么推荐系统也就无法发现这个人可能还喜欢其它物品。
  - 无法为新用户产生推荐：由于CB算法需要依赖用户的历史数据，那么对于新用户而已就有可能无法产生一个比较可靠的推荐。

KB

- 传统的推荐算法(CB和CF)适用于推荐特性或者口味相似的物品，比如：书籍、电影或者新闻。但是在对某些产品进行推荐的过程中，就有可能不是特别适合的方法，比如汽车、电脑、房屋、或者理财产品等等。主要是两个原因：**很难在一个产品上获取大量的用户评分信息**以及**获得推荐的用户不会对这些已经过时的产品产生一个满意的回馈。**
- **基于知识的推荐技术(Knowledge-based Recommendations, KB)**是专门解决这类问题的一种新的推荐技术，高度重视知识源，不会存在冷启动的问题，因为推荐的需求都是被直接引出的。缺点是：所谓的知识的获取比较难，需要知识整理工程师将领域专家的知识整理成为规范的、可用的表达形式。
- 基于知识的推荐技术需要主动的询问用户的需求，然后返回推荐结果。

KB-会话式推荐系统交互形式

- 一般的交互过程如下：
  - 用户指定自己的最初偏好，然后一次性问完所有问题，或者逐步问完问题。
  - 当收集到足够多有关用户需求和偏好的信息后，会提供给用户一组匹配产品。
  - 用户可能会修改自己的需求
- 类似于搜索过程，只是将搜索过程中给定的参数输入到基于知识的推荐系统中。
- 系统开发中需要考虑的问题：
  - 需要一些比较高精度的推荐结果
  - 当没有完全匹配物品的时候，需要给定解决方案，比如主动提供某些的候选结果

KB-分类

- 基于知识的推荐系统分为两大类:**基于样列的推荐**和**基于约束的推荐**；这两种方法非常相似：先收集用户需求，在找不到推荐方案的情况下，自动修复与需求的不一致性，并给出推荐的解释。区别在于：推荐方案是如何被计算出来的。
- 基于样列的推荐方法通过相似度衡量标准从目录中检索物品。
- 基于约束的推荐方式主要是利用预先定义好的推荐知识库，即一些描述用户需求以及与这些需求相关的产品信息特征的显示关联规则；也就是使用约束求解器解决的约束满足问题或者通过数据库引擎执行并解决的合取查询形式。

KB

- 基于知识的推荐系统一般情况下需要依赖物品特征的详细知识；简单来讲，推荐就是从物品特征数量表中挑出能够匹配用户需求、偏好和硬件需求的物品；用户的需求可能会表达成为：价格不超过在2200元的物品或者能够防水等等

id	价格(￥)	兆级像素	光学对焦	显示屏尺寸	录像	声音	防水
p1	1580	8.0	4x	2.5	否	否	是
p2	1820	8.0	5x	2.7	是	是	否
p3	1890	8.0	10x	2.5	是	是	否
p4	1960	10.0	12x	2.7	是	否	是
p5	1510	7.1	3x	3.0	是	是	否
p6	1990	9.0	3x	3.0	是	是	否
p7	2590	10.0	3x	3.0	是	是	否
p8	2780	9.1	10x	3.0	是	是	是

基于约束的推荐技术-基本概念

- 基于约束的推荐技术可以使用一组(V,D,C)来描述，其中：
  - V：是一组变量集合，主要是:V_c和V_{PROD};
  - D：是一组这些变量的有限域;
  - C：是一组约束条件，描述了这些变量能够同时满足的取值的组合条件；主要是C_R、C_F、C_{PROD};
- 实际上基于约束的推荐技术就是在约束($V=V_c \cup V_{PROD}$, D, $C = C_R \cup C_F \cup C_{PROD}$)情况下，给定一个需求REQ，给出一个最终的RES推荐结果。

基于约束的推荐技术-基本概念

- **用户属性**(V_C)：描述了客户的潜在需求，即用户需求的特征属性实例化。比如：max-price表示用户能够接受的最高价格。
- **产品属性**(V_{PROD})：描述了一个给定产品种类的特征属性，比如mpix表示分辨率。
- **一致性约束条件**(C_R)：定义了允许范围内的用户实例对象，也就是对客户需求可能的实例化的系统约束，比如：如果要去相机能够打印大尺寸的照片，则最大可接受价格必须高于1500。
- **过滤条件**(C_F)：定义了在那种条件下应该选择的哪种产品，也就是定义了用户属性和产品属性之间的关系，比如：大尺寸照片打印功能要求相机的分辨率至少大于5mpix。
- **产品约束条件**(C_{PROD})：定义了当前有效的产品分类

基于约束的推荐技术-基本概念

V_c	{max-price(0...1000), usage(digital, small-print, large-print), photography(sports, landscape, portrait, macro), waterproof(yes,no)}
V_{PROD}	{price(0...10000), mpix(3.0...12.0),,....,waterproof(yes,no)}
C_F	{usage=large-print -> mpix > 5.0}（usage是用户需求特性，mpix是商品特性）
C_R	{usage=large-print -> max-price > 1500}（usage和max-price都是用户需求特性）
C_{PROD}	{(id=p1^price=1480^mxip=8.0^...^waterproof=no) v (id=2 ^..^ waterproof=yes) v ...}
REQ	{max-price=2000, waterproof=yes, usage=large-print}
RES	{id=p4 & id=p1}

基于约束的推荐技术-默认值设置

- **默认值设置**的主要目的是帮助用户说明需求的一种方式，当用户给定一个比较模糊、泛化的需求的时候，系统可以对该属性指标进行解析转换，得到更加丰富的需求条件列表。比如：当一个用户需要的是一个可以打印大尺寸照片的时候，我们可以默认认为他需要的相机的像素必须大于3兆；给定默认的方式如下：
  - **静态默认设置**：每一个属性都具有一个默认值
  - **条件默认设置**：根据用户给定的需求条件，生产一个默认值
  - **派生默认设置**：利用以前所有用户的交互日志和当前用户给定的参数，进行分析建模，得到每一个属性的默认值。最常用的方法为:**1最近邻**和**加权多数投票**。

基于约束的推荐技术-处理空结果集

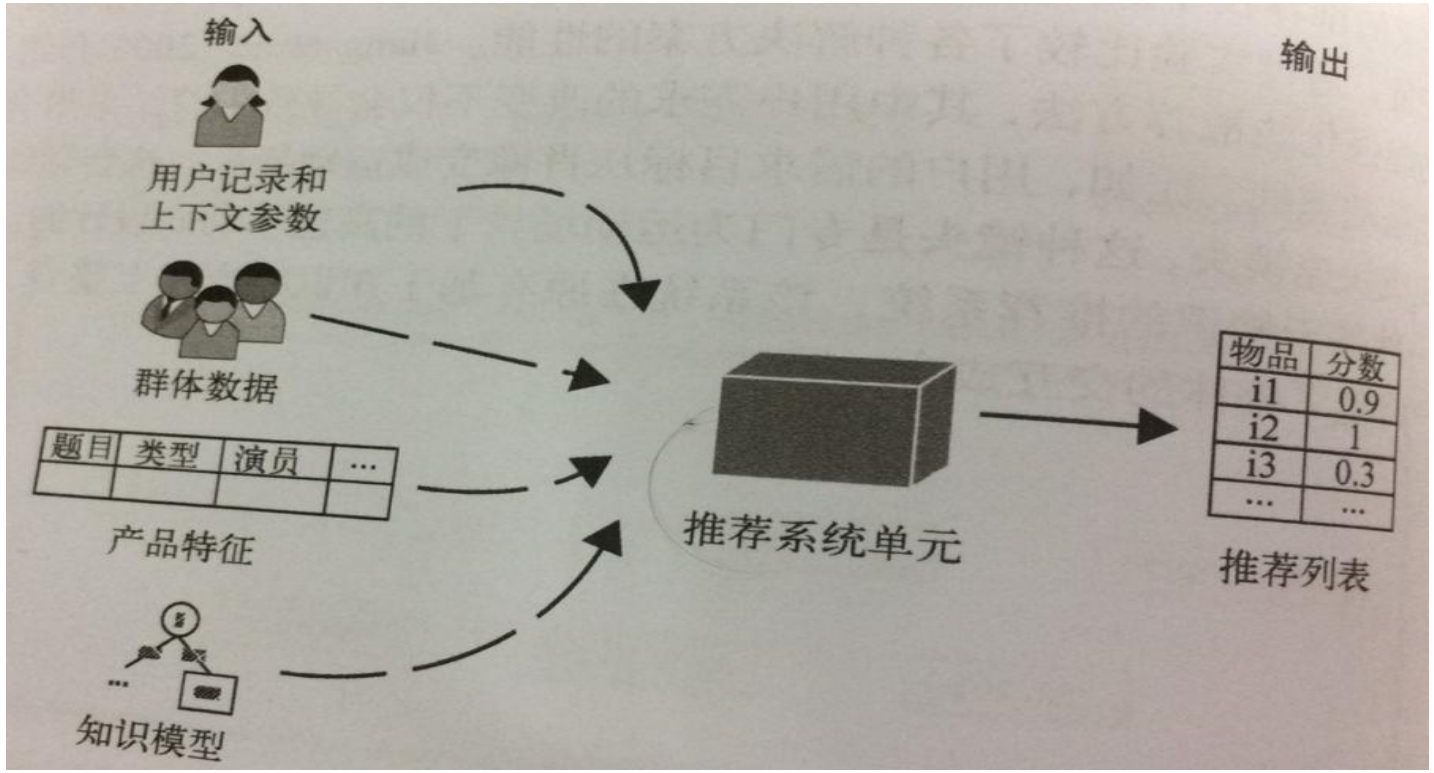
- 实际上，当用户给定的需求太多的时候，就有可能产生没有任何一个物品是符合给定需求的，也就会产生一个空推荐结果集，基本上所有的推荐系统都不能完全解决这种“无米之炊”的难题，常用的一种解决方案是：
  - 给定用户需求特征属性的优先级别
  - 按照属性的优先级别删除原始需求中的需求，得到一个新的需求条件列表，重新获取推荐数据，直到有结果产生。

KB-总结

- 基于知识的推荐系统在协同过滤或者基于内容的推荐技术有明显缺点的时候十分有用，并且能够很好的应用到大型的推荐系统中，但是基于知识的推荐系统还是存在着一系列的问题：
  - 基于约束的推荐技术构建约束条件需要比较多的一个领域知识，比较难。
  - 基于样列的推荐技术当计算物品和需求之间相似度公式效果不佳的时候，推荐的结构比较不好，而且结构化物品特征数量以及构建特征属性和需求之间的相似度计算规则比较难，需要比较高的一个领域知识。
- 未来是一个发展方向，但是在当前推荐领域中实际应用的不多。

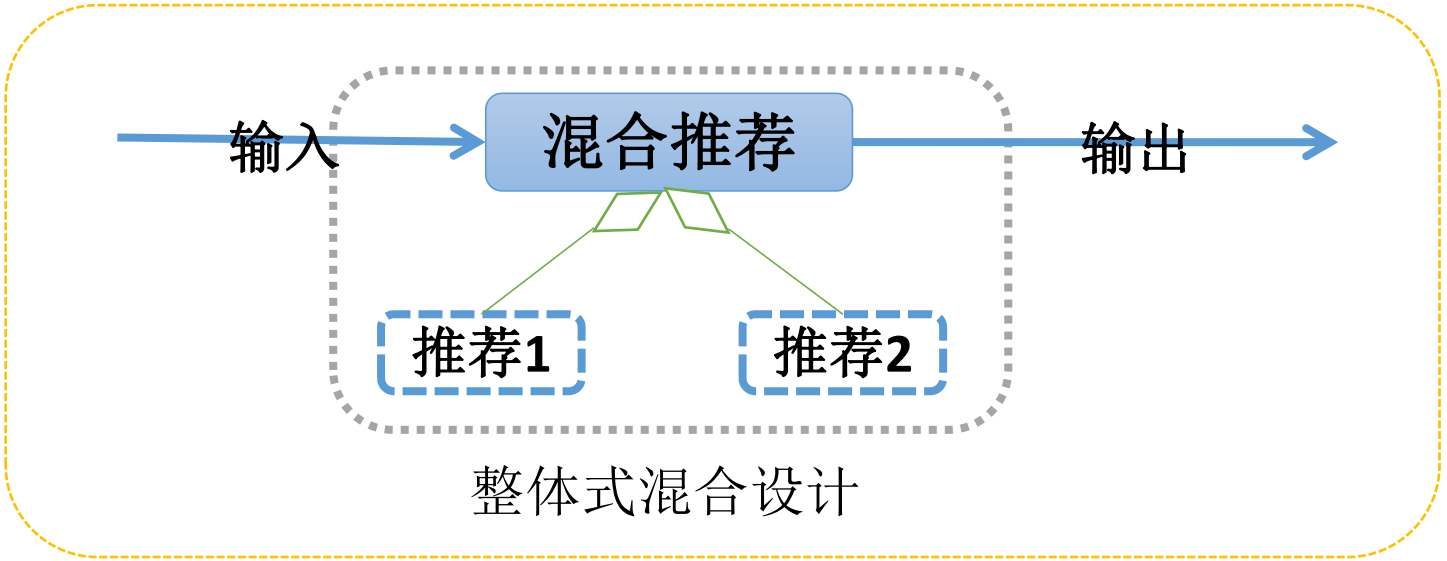
混合推荐系统

- 我们可以把推荐系统看成一个黑盒子，可以将输入的数据转换成为一个有序的物品列表再进行输出，输入的数据主要包含用户记录、上下文信息、产品特征、知识模型等等，但是没有任何一个推荐算法可以利用到这所有的输入数据，所以可以考虑将多个推荐系统模型的结果混合到一起来作为最终的推荐结果。

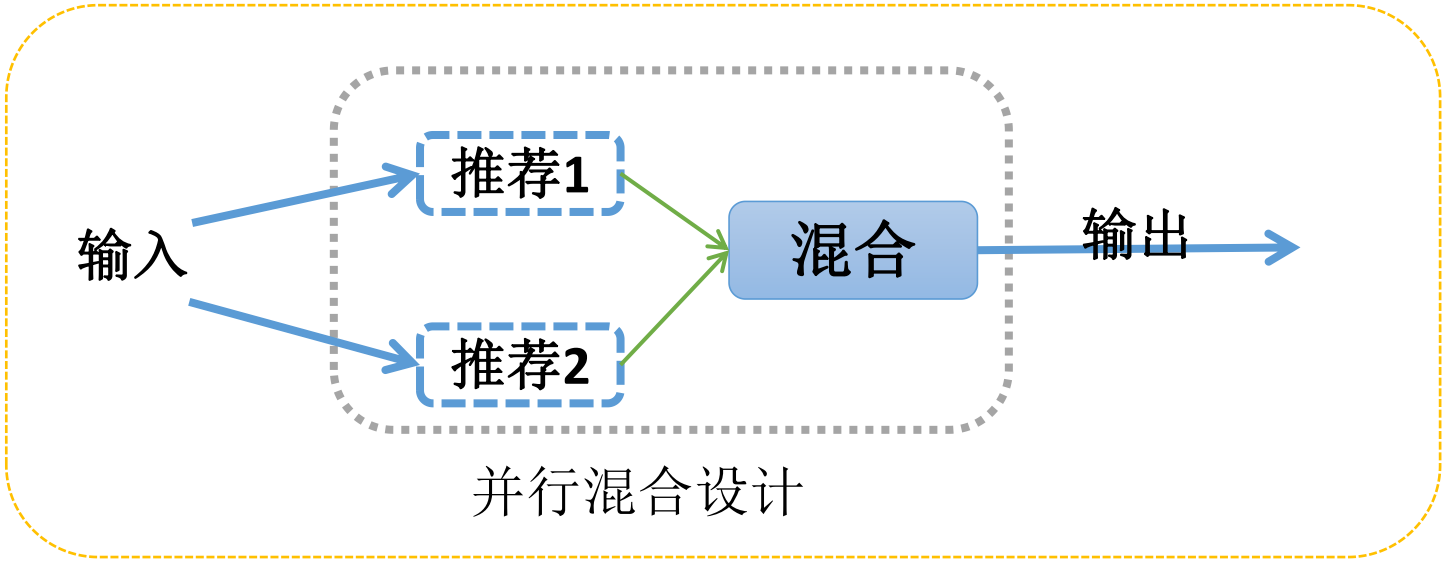


混合推荐系统

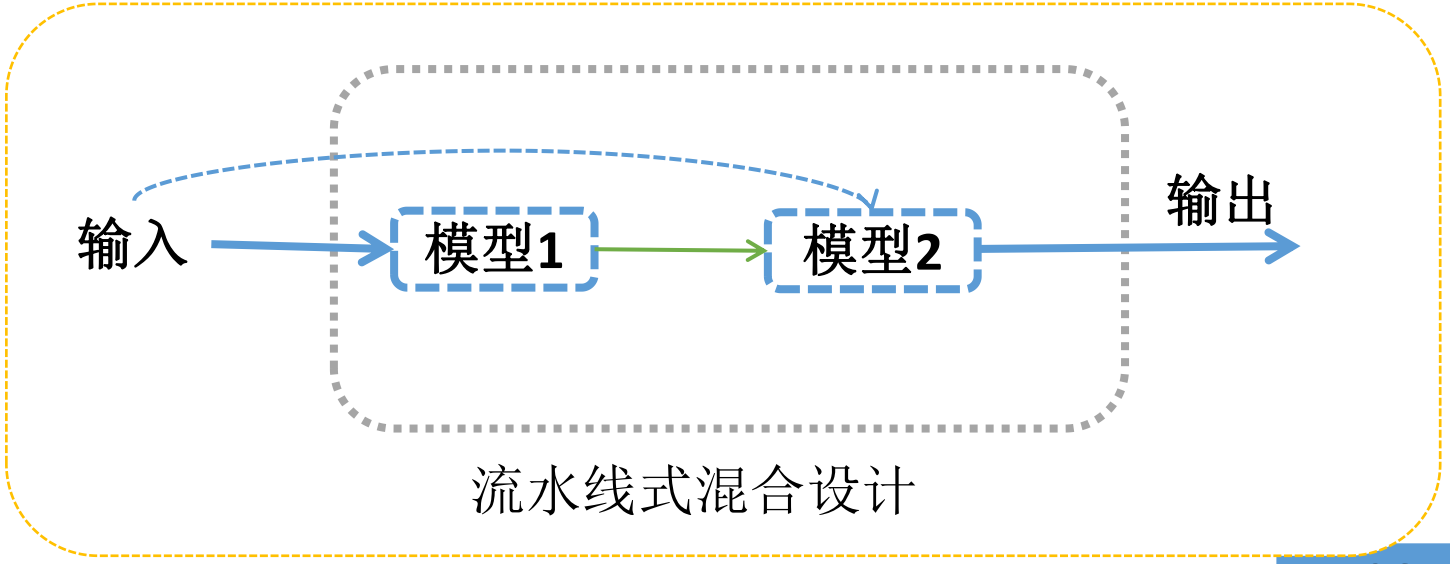
- 混合推荐系统的设计结构主要分为三大类，分别是：**整体式混合设计、并行式混合设计、流水线式混合设计。**



混合推荐系统



并行混合设计



流水线式混合设计

推荐系统扩展-攻击

- 在实际应用中，由于推荐系统的建议可能会影响用户的购买行为，带来经济效益的时候，我们并不能假设所有的用户都是诚实公平的，也就是说存在的恶意用户有可能会影响推荐系统的运行效果，让推荐列表经常(或者很少)包含某类商品，这种问题就叫做推荐系统攻击。
- 解决方案：
  - 尽可能的提高“可信”朋友的评分权重
  - 过滤异常数据，因为只有大量异常数据的存在才有可能对最终结果产生不好的影响，那么只需要过滤这部分的异常数据就可以解决这个问题。

THANK YOU

上海育创网络科技有限公司