

NLP技术在推荐系统中的应用

张相於

2018/10/16



第十届中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2018



自我介绍


- 阿里高级算法专家
- 关注领域
 - 推荐系统
 - 用户画像
 - 金融风控
 - 机器学习



提纲

- 推荐系统中的关键问题
- 推荐系统中的文本数据
- 推荐系统中的行为数据

推荐系统中的关键问题



相关性计算

物品标签

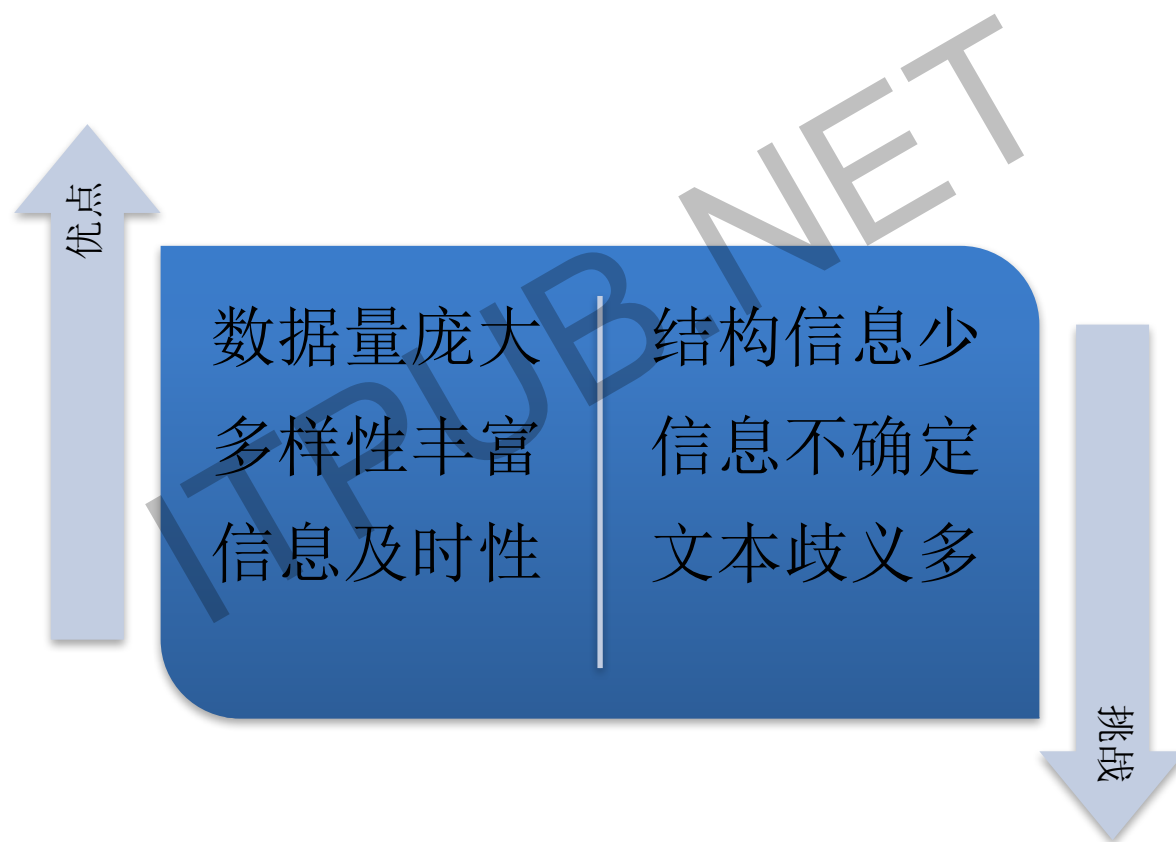
人群标签

排序模型

行为序列分析

推荐系统中的文本数据

文本数据的特点



词袋模型

- 词袋模型的核心假设
 - 文档由词组成
 - 词之间互相独立、无序
 - 只保留词的次数，丢弃其他信息
- 应用场景
 - 相关性计算：用关键词连接用户和物品
 - 排序特征：直接用作高维度特征
 - 物品标签：TF-IDF
- $TF \Rightarrow TF-IDF$
 - 将全局信息加入重要性度量
 - 度量更合理
- TF-IDF的变种
 - TF的log缩放
 - TF的归一化
- N-gram:
 - 2-gram([我, 特别, 爱吃, 鸡翅])=[我特别, 特别爱吃, 爱吃鸡翅]
 - 提高特征区分度，也会带来稀疏性。

向量空间模型

- 问题：
 - 给定两个物品的描述，如何度量它们之间的关系？
- 核心思想
 - 把一组对物体的描述向量化
 - 在此基础上定义相关操作
- 向量点乘：推荐系统中的万能公式
 - 点乘结果大小反应相关性强烈程度
 - 适用于任何数据：文本、行为、等等
- 余弦相似度：归一化的向量点乘
 - $\cos(d_1, d_2) = \frac{V(d_1) \cdot V(d_2)}{|V(d_1)| \times |V(d_2)|}$
- 数据普适
 - 内容、行为、等等
 - 0-1值，连续值，离散值
- 可调
 - $\cos(d_1, d_2) = \frac{V(d_1) \cdot V(d_2)}{|V(d_1)| \times |V(d_2)|}$
- 可解释
 - 总体相关性等于分量相关性的叠加
 - 叠加方式可调节控制
- 算法模块化
 - 原始描述与相似度度量相隔离

问题：同义词、多义词、维数高、不稳定.....

隐语义模型：LSA

- 词袋模型的缺点
 - 维度高、稀疏、信息量小
 - 无法处理近义词、一词多义
 - 缺乏高层次含义（语义）
- 语义模型
 - 引入更高维度的概念：语义（主题）
 - 凝聚更多信息
- 隐语义模型
 - 文章->主题->词
 - 通过观察到的结果（词），推测生成过程。
- 识别“多词同义”、“一词多义”
- 低维表示包含更高抽象层次信息
- 用转换后的维度数据做存储索引（LSI），可提高检索的召回率
- 可看做一个软聚类，是mixture model的基础。

Latent Semantic Analysis

	cat	dog	computer	internet	rabbit
D1	1	1	0	0	2
D2	3	2	0	0	1
D3	0	0	3	4	0
D4	5	0	2	5	0
D5	2	1	0	0	1

$$C \approx C_k = U \Sigma_k V^T$$

$$C_K = \operatorname{argmin}_{\{Z, \operatorname{rank}(Z)=k\}} ||C - Z||_F \quad ||M||_F = \sum m_{ij}^2$$

如何理解LSA

- 目标：找到原始数据背后的深层次因素
- user -> item => user -> latent variable -> item
- $C \approx C_k = U \Sigma_k V^T$
 - U: user -> latent variable
 - V: latent variable -> item
 - $\Sigma_k: n \rightarrow k, k \ll n$

LSA的问题

- SVD计算复杂度高
- 检索复杂度高
- 无概率含义
 - U 和 V 的取值不满足概率原则
 - 可能出现负数

概率隐语义模型：LDA

- 从 $Dir(\alpha)$ 抽取一条样本 θ_i ，对应着一个文本的主题分布概率。
- 从 $Dir(\beta)$ 抽取一组样本 φ ，对应着不同主题下的词分布。
- 对于1到N的词 w_n :
 - 从分布 $Multinomial(\theta_i)$ 中抽样一个主题 c_{ij} 。
 - 从分布 $Multinomial(\varphi_{c_{ij}})$ 中抽样一个词 w_{ij} 。
- 变分推断计算效率高
- 矩阵分解结果有概率含义
- 对应着最小化KL距离的矩阵分解

LDA应用实例：排序特征

- Topic Id作为排序特征
- 两种分布情况
 - 少数主题占据较大概率
 - 大量主题均分概率
- 聚类区分

LDA应用实例：用户&物品标签

- $P(w|u) = \sum_{t,d} P(w|t) \times P(t|d) \times P(d|u)$

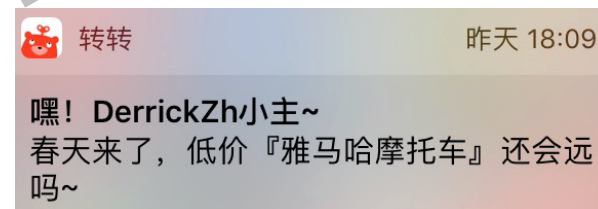
- 启发式规则

- 选取用户行为较多的topic
- 选取该topic下概率较高的词

- 应用：

- 推荐理由
- 个性化推送

- 同理适用于物品标签



LDA应用实例：主题重要性

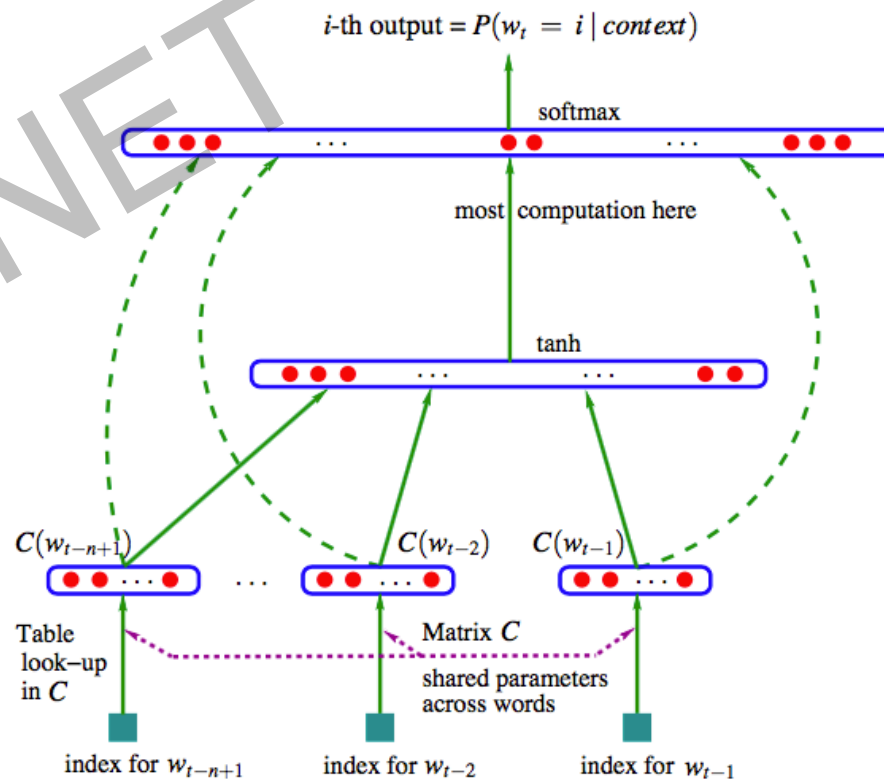
- 主题重要性各不相同
 - 主题1：【教育，学校，读书.....】
 - 主题2：【第一册，第二册，第三册.....】
 - 主题3：【人民教育出版社，高等教育出版社.....】
- 如何度量重要性？
 - 计算每个主题在不同文档下的概率分布
 - 计算信息熵
 - 信息熵小->主题质量好

LDA的问题

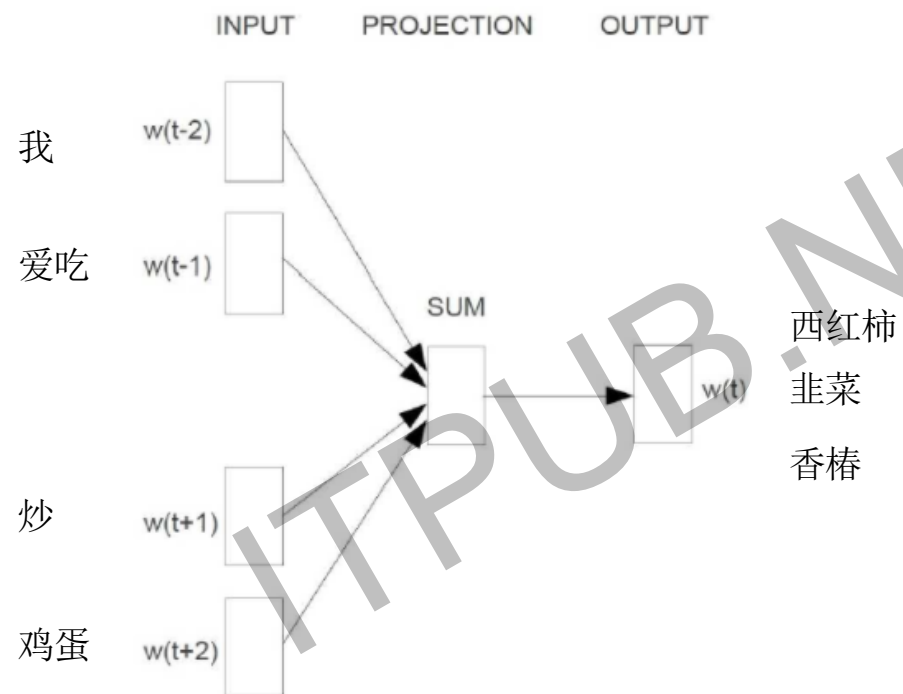
- 核心假设：
 - 词之间可交换顺序（exchangeability）
 - 给定 θ 和 φ ，每个词的生成是相互独立的。
- 忽视了词的上下文环境
 - 我要 吃 鸡腿
 - 鸡腿 吃 我要

神经概率语言模型

- 隐变量 -> 嵌入层
(embedding)
- 上下文和当前词相互预测，上下文可自由定义。
- 引入了可控制的上下文，更广的适用空间。



“过气”网红：word2vec



研究表明，汉字的顺序并不一定影响阅读，比如当你看完这句话后，才发现这里的字全是都乱的。

@全球潮流时尚榜
weibo.com/1411325

Word2vec的应用

- 词聚类、扩展
- 相关性预测
 - 我爱吃？炒鸡蛋=>西红柿、韭菜、香椿
 - 三体，基地，？=>流浪地球、球状闪电
- 维度抽象层次太低导致泛化性能差
 - 新商品来了怎么办？
- 更高维度上训练
 - 搜索词、topic、类别等等
- 时序性
 - 搜索行为的循序渐进

Word2vec应用实例

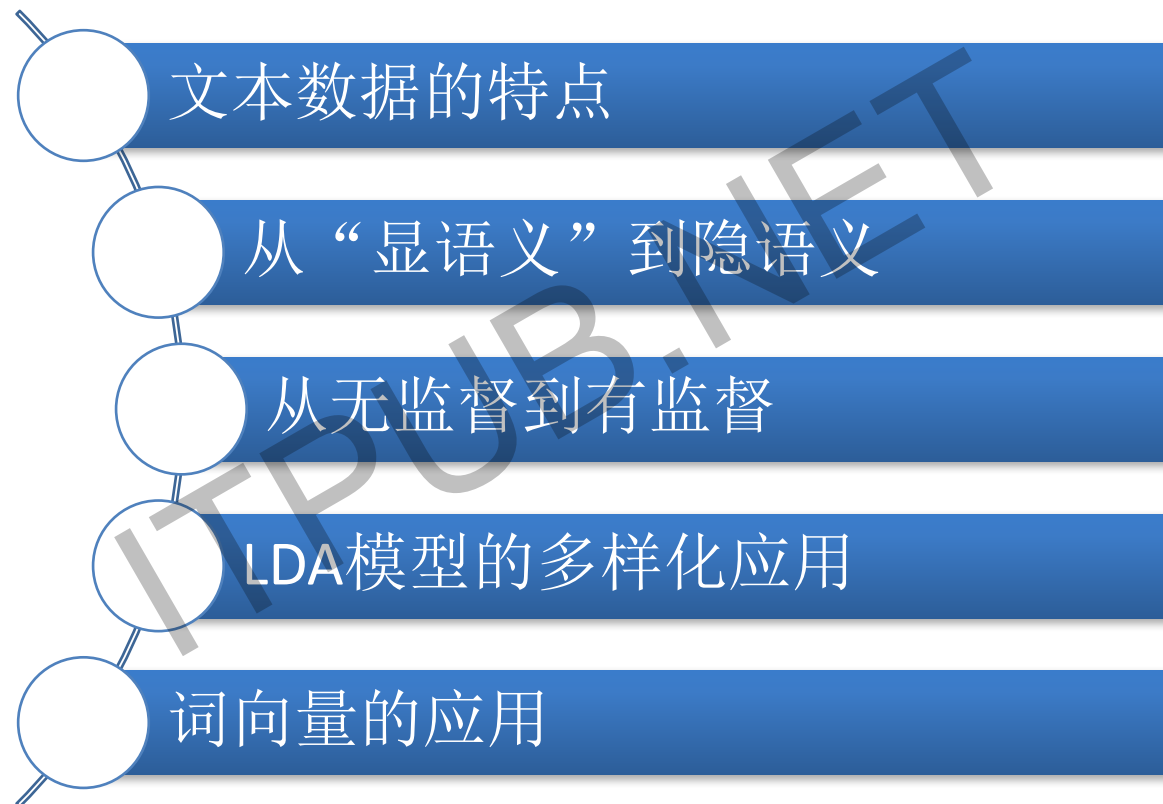
历史搜索词	预测搜索词
书桌，儿童书桌	学生书桌
笔，中性笔	签字笔
二手电动车，电动车	电动车 小龟王
爆米花机，冰激凌机	烤肠机
手表，精工手表	西铁城手表

如何从词向量得到更长的文本向量？

- 深度学习方法
 - Paragraph2vec
 - Doc2vec
- SIF embedding
 - $W2v + \text{average}$
 - $\text{Weight} = a / (a + p_w)$
 - 去除语义无关方向向量
 - 时态、单复数等
- DisC embedding
 - SIF+词序保留
 - Compositional n-gram embedding v_n
 - $\text{concat}(v_1, v_2, \dots)$

【一分钟整明白】不用深度学习的文本向量表示

小结



推荐系统中的行为数据

重新认识文本数据



点击刻画物品

- 电商网站最常用模式：
 - 搜索->浏览->点击 (->购买)
- 文档->词 vs 搜索词 -> 商品
- 用TF-IDF计算搜索词下商品的重要性
- 应用
 - 根据用户搜索召回商品
 - 热搜排行榜

搜索数据文档化

文本场景下

- c1: 该文档总词数
- c2: 该词在该文档出现次数
- c3: 总文档数
- c4: 该词出现的文档数

搜索场景下

- c1: 搜索该query后的总点击数
- c2: 搜索该query后对该商品的点击数
- c3: 总query数
- c4: 该商品出现的query数

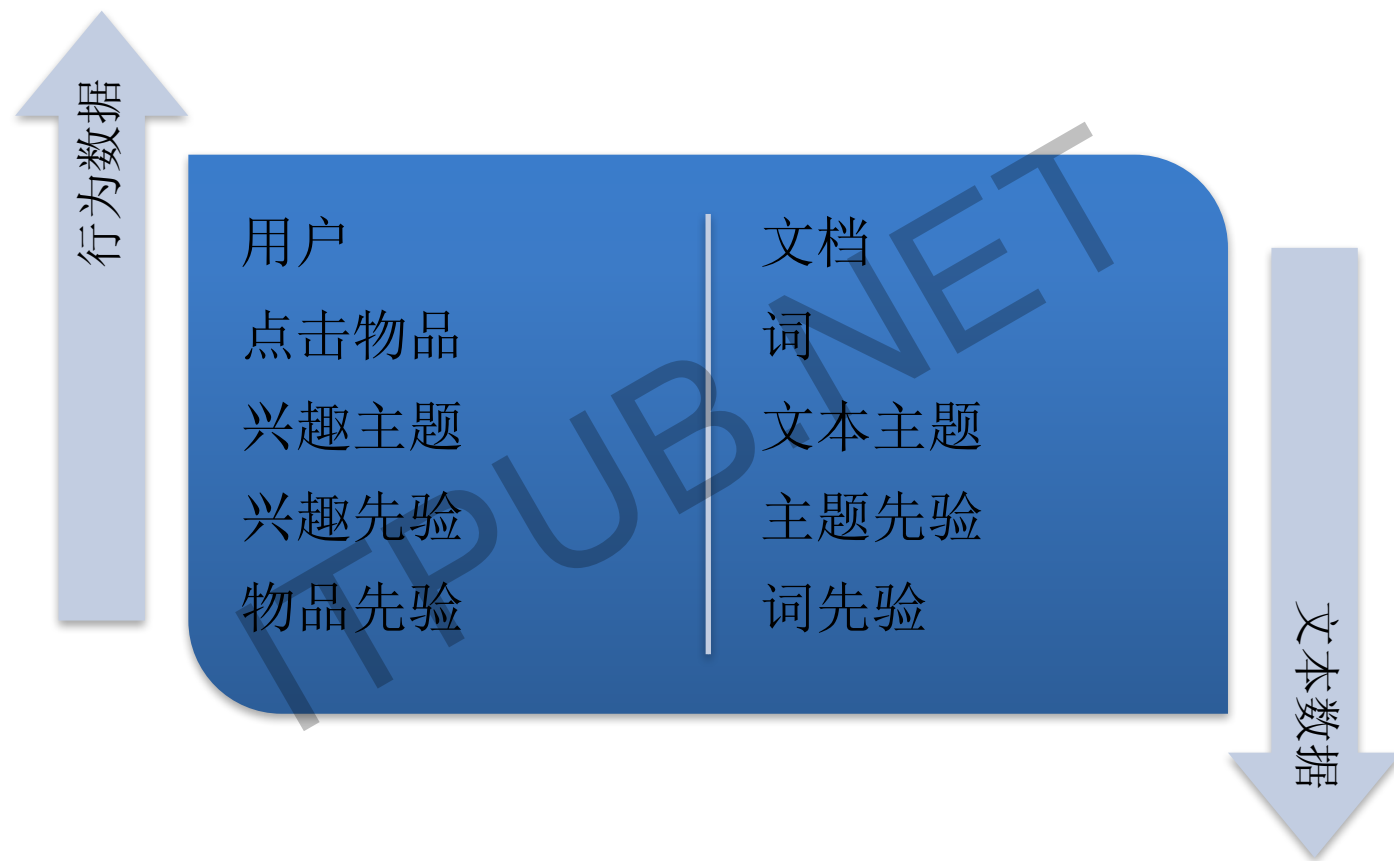
$$\text{tf-idf} = c2/c1 * \log(c3/(c4+1))$$

结果: 每个query下最重要的商品

人群聚类

- 用户行为模式拆解
 - 表现：用户点击了《明朝那些事》
 - 本质：用户属于历史爱好者，历史爱好者喜欢看《明朝那些事》
- $P(item|user) = \sum_{topic} P(topic|user) \times P(item|topic)$
- 用矩阵分解拆解用户中的人群聚类
 - SVD LDA

行为数据文档化



用户行为主题聚类

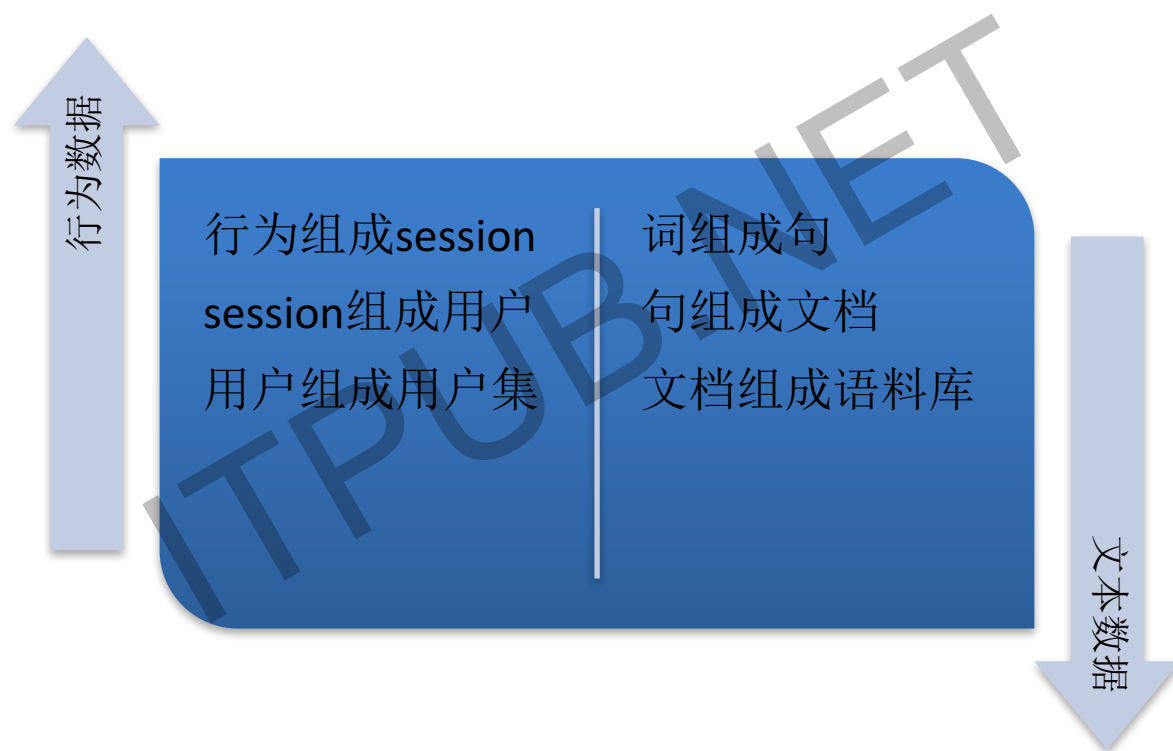


全局兴趣分布

用户兴趣标签

物品兴趣标签

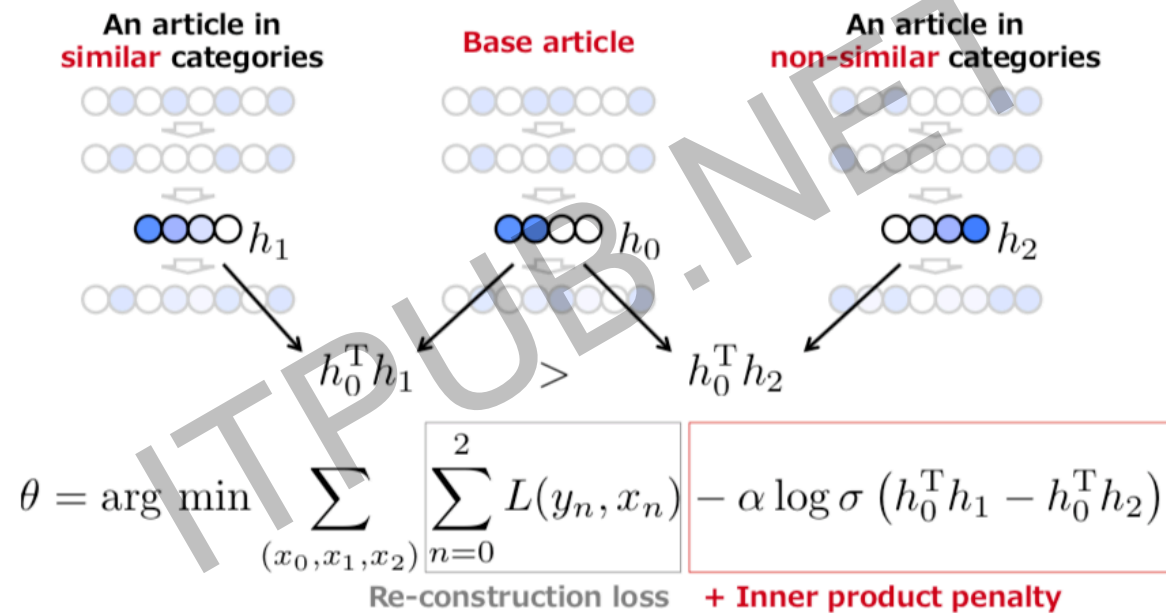
行为序列文档化



行为序列预测

- 给定用户历史行为，预测未来行为
- **BOW word2vec**
 - 根据上下文预测当前词
 - 上下文->前序词
 - 当前词->下一个词
- **node2vec deepwalk**
 - 更好地利用网络结构
- **LSTM**
 - 更好的长依赖处理
 - 更专注与序列预测

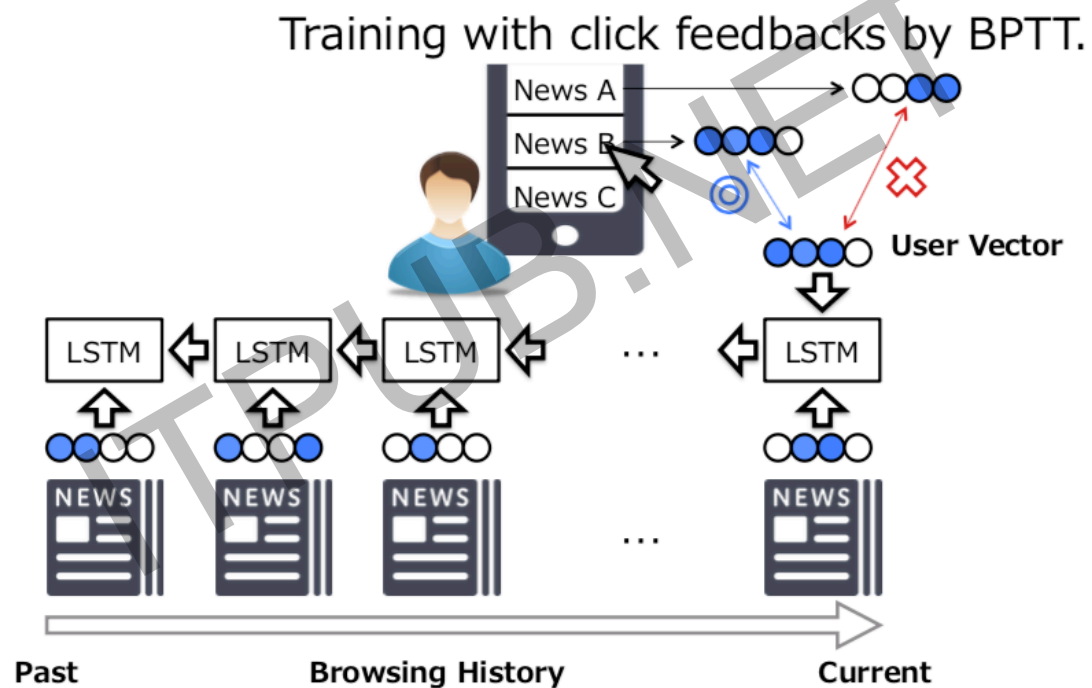
例子：用LSTM做新闻推荐



→ It can construct better **inner product space**.

KDD17, Embedding-based News Recommendation for Millions of Users

例子：用LSTM做新闻推荐



KDD17, Embedding-based News Recommendation for Millions of Users

总结



DuckType思想

用好头部算法

充分认识数据数据



THANKS