精准定向的广告系统

从广告的经验看如何把推荐做成一个产业

王益
yiwang@tencent.com

问题：把推荐做成一个产业

- 《推荐系统实践》说了两个事情：
  - 各种源自特定目标（Netflix）和学术研究的推荐方法；
  - 如何用这些方法来实现“一个推荐系统”。
- 明白这两件事，则
  - 可以成就一些推荐系统，一些推荐专家，
  - 不能成就一个像广告那样撑起Google和百度这样的推荐产业。
- 如何成就一个推荐产业：
  - 深刻了解问题：从实证主义工作升华到理论体系。
  - 重视工程实现：可以应付各种推荐需求，可以插拔各种推荐方法，可扩展的离线计算，高并发的在线服务。
- 业界现状
  - 开始出现做“推荐产业”的公司，
  - 借鉴广告技术和业务经验。

推荐和广告的比较

	推荐	广告
目标	从多个项目中选择一个给用户	从多个广告中选一个给用户
输入	用户历史行为中总结的“兴趣”和当前关注点	上下文广告和个性化广告
问题	让用户满意	优化用户、广告主、广告系统（和媒体）的满意度的纳什均衡
优化目标	用户满意=准确度+多样性	Auction model（GSP）
不确定性	预测用户打分	“喜欢”的概率 = 点击率
数据稀疏性	Collaborative filtering Latent factor analysis Heat propagation	从实证主意CTR预估演进为基于模型的CTR预估
冷启动问题	用内容补充用户信息缺乏	Exploraiton and exploitation 理论
产业成熟度	各个公司的推荐团队	著名大公司，成熟的商业模式

情境广告系统介绍

- 情境广告
- 总体技术思路
- 相关性
- 排序
- 系统构架
- 在线实验

精准广告投放

- Query = 用户 + 环境
  - Google AdSense, for mobile and Gmail
  - 个性化广告：百度搜客
- 难点：
  - 在长尾流量上为长尾用户出长尾广告。
  - 满足多方利益：广告主、媒体、用户和引擎。
- 技术思路：
  - 相关性 + 排序
  - 完全预先计算
- 和搜索广告的区别
  - 流量大几个数量级
  - Query的商业价值不直接

广告相关性

- 倒排表
  - 关键词
  - 分类结果
  - 隐含语义
- 倒排表的频繁更新
  - Read-Copy Update (RCU)
  - 增量更新机制
- 信息获取算法
  - 标准布尔模型
  - 向量空间模型
  - 概率模型

语义匹配

- Query和广告都是短文本
  - 信息量不足，所以歧义
- 举例：
  - Q1: apple pie
  - Q2: iphone crack
  - D1: Apple Computer is a well known company located in CA.
  - D2: The apple is a kind of fruit.
- 无法在词空间做匹配

用知识补足信息缺失

- 从大量文本中学习“知识”，来补足短文本的信息。
- “知识”是一组“语义”，每个“语义”是一组“词”。
- 举例：
  - C1: {apple tree pie fruit }
  - C2: {computer iphone ipad apple CA}
  - “apple pie” → C1: 99%, C2: 1%
  - “iphone crack” → C1: 1%, C2: 99%
  - “Apple Computer …” → C1: 1%, C2: 99%
  - “The apple is …” → C1:99%, C2: 1%

语义的学习和理解

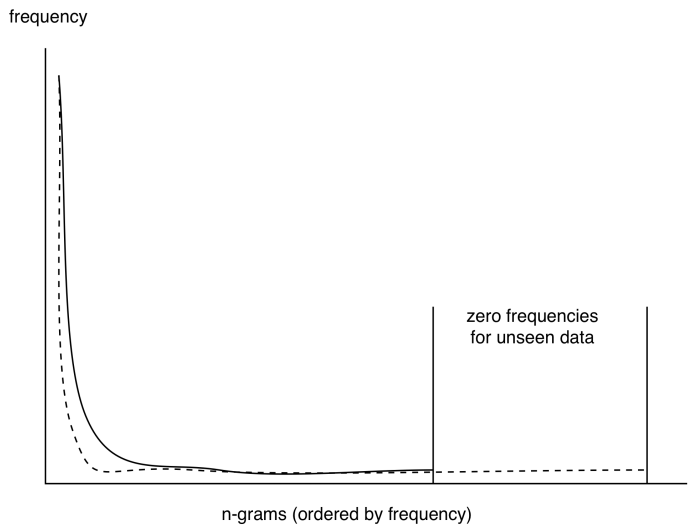
- 学习算法
  - 输入：训练文档、每个文档是一包词
  - 输出：模型（每个语义是一包词）
  - 计算方式：离线、批量、分布式、可扩展
- 理解算法
  - 输入1：一个文档
  - 输入2：模型
  - 输出：文档的语义（每个语义有权重）
  - 计算方式：在线、实时、在速度和精度之前求平衡

语义学习和理解各种思路

- 文本聚类
  - K-means, spectral clustering, graph-cut
  - 可以学习得到词的聚类，但是没有办法理解给定的文档。
- 矩阵分解
  - SVD, NMF
  - 可以把给定文档投影到语义空间，但是语义权重没有解释。
- 概率模型
  - pLSA, 解释语义的算法比较hacky。
- 贝叶斯语义模型
  - LDA, Pachinko allocation, hierarchical LDA
  - 解释语义的算法是“科学”的
  - 无法学习得到“小众”（长尾）语义
- Google Rephil
  - 1,000,000 clusters；comparable to the vocabulary

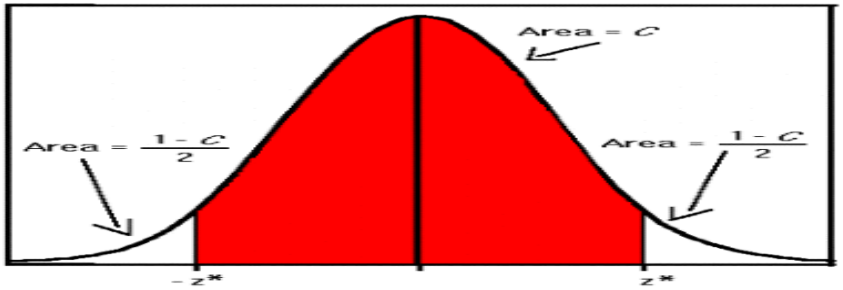
排序、定价和 CTR 预估

- General Second Price 竞价：
  - 目前各种在线广告系统（AdWords、AdSense、凤巢、百度网盟）都符合的竞价模型
  - 排序：pCTR * bid
  - 定价：pCTR2/pCTR1 * bid2
- CTR的实证估计（empirical estimate）
  - pCTR ≈ click/impression
- 如果分子为零怎么办？
  - 数据稀疏，但是“一切皆有可能”
  - Discount smoothing



后验分布和置信区间

- click > 0 就意味着估计准确吗？
  - 不是。更大的 impression 标志更大的可信度。
  - 可信度的体现：后验分布和置信区间：



  - 威尔森区间（Wilson score interval）

$$\frac{\hat{p} + \frac{1}{2n}z^2_{1-\alpha/2} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{1-\alpha/2}}{4n^2}}}{1 + \frac{1}{n}z^2_{1-\alpha/2}}$$

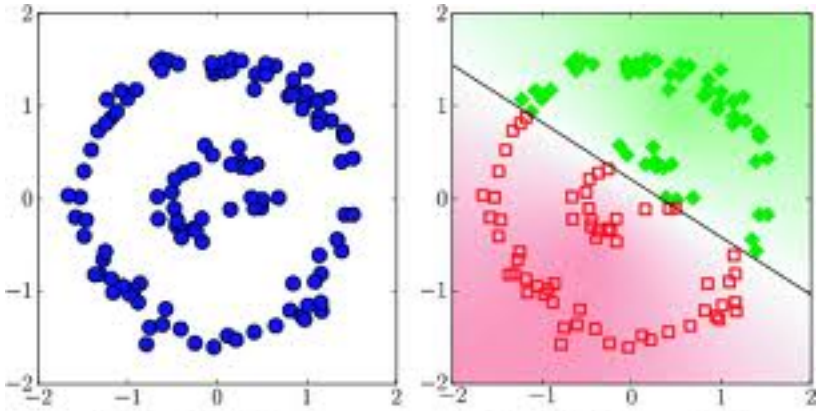
  - Exploitation-and-Exploration

CTR 预估的数学模型

- 如果 impression = 0 怎么办？（冷启动问题）
  - 问题：
    - 一个广告 a 从来没有被展示过，我们无法通过实证主意统计（frequensist）预估点击率。
  - 解法：
    - 如果 a 和一个展示过的广告 b “类似”，则可以预估 a 的点击率和 b 的接近。
  - 如何判断“类似”
    - 投影到特征空间做比较。
  - 建模：
    - 同时考虑特征空间、类似程度计算、和点击率预估
    - P(click=1 | ad, query) = f(ad, query)
  - 进一步的问题
    - 函数 f 的参数华形式什么样的？
    - 特征空间的维度？
    - 需要多少训练数据？

CTR 预估模型的演进

- 早期思路：非线性模型
  - 受启发与非线性模型在搜索引擎的 click-model 中的影响
  - 典型的特征数量：几十到几百
  - 典型的模型：kernel machines, boosting tree
- 胜者为王：线性模型
  - 真实的广告数据就是高维的，不需要投影到更高维度的空间去分类。

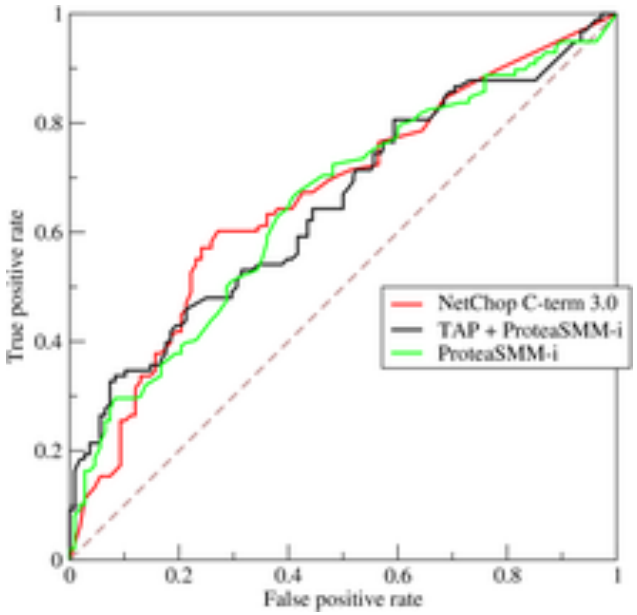


算法研究和大规模系统

- 离线训练
  - L-BFGS: 计算梯度的时候要计算Hessian matrix逆矩阵: $H^{-1}g_t$
  - OWL-QN: 把 $H^{-1}g_t$ 近似为 $H(g_{t-1},g_{t-2},\cdots)g_t$
  - CDN: 数据分布不是横着切，而是竖着切
  - 可以用 MapReduce 并行，可以用更细致的并行方法。
- 在线训练
  - Stochastic gradient descend
  - 可以用 MapReduce 和 Bigtable 一起优化；也可以做在线优化。
- 并行在线训练
  - IPM: 混合多个独立训练的模型，是系统可扩展性的基础；
  - SGD + IPM: 高效处理大流量的在线训练系统。
- 大规模机器学习系统
  - 重要算法的并行话可以以利用现有并行计算系统开始，
  - 但是应当终结于专门设计的并行计算系统。

离线评估和在线实验

- 真实的互联网数据是个长尾的世界
  - 数据稀疏性是无法避免的
- 点击率低是普遍现象
  - 推荐系统、拼写纠错、广告系统
- 传统的离线评估方法无效
  - AUC：数值稳定但是只能评估排序
  - MAE/MSE：能评估准确性但是不稳定
  - 请参照 KDD Cup 2012 评估方案
- 在线实验很重要
  - 业务目标不是准确也不是排序，而是持续盈利。
  - 能尽量复用流量，就能做更多的实验；
  - 做实验快就是占领市场份额。



系统架构和开发

- 再瑰丽的梦想也得写在程序里
- 后台架构技术
  - 用长连接不用短连接，
  - 用TCP不用UDP，
  - 开发专门的RPC系统，不要直接在通信库上开发
- 多层次多维度的测试
  - 单元测试和开发同时进行
  - 小规模集成测试（onebox test）融入CI
  - 预发布测试用真实流量，且先于在线实验和在线发布
- 离线计算基础
  - Distributed filesystem
  - Distributed operating system
  - Parallel computing system (MapReduce 等)
  - 实时的日志分析