

# QCon 全球软件开发大会 【北京站】2016

## 外卖商家排序的算法演进

温赞  
美团外卖

# 外卖商家排序的算法演进

温赞

wenyun@meituan.com



# 内容概览

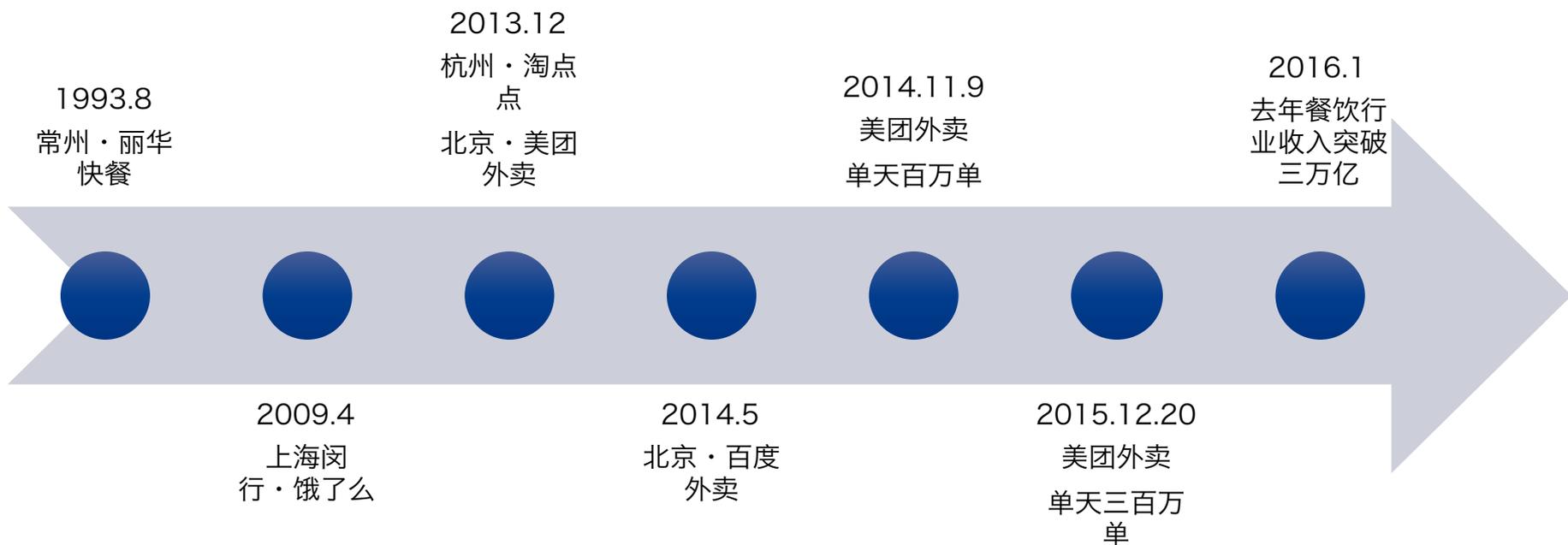
---

- 外卖行业发展简介
- 外卖与团购的异同
- 外卖商家排序的技术演进
- 外卖商家排序的算法挑战与应对
- 小结

---

# 外卖行业发展简介

# 外卖行业发展简介



# 外卖行业发展简介

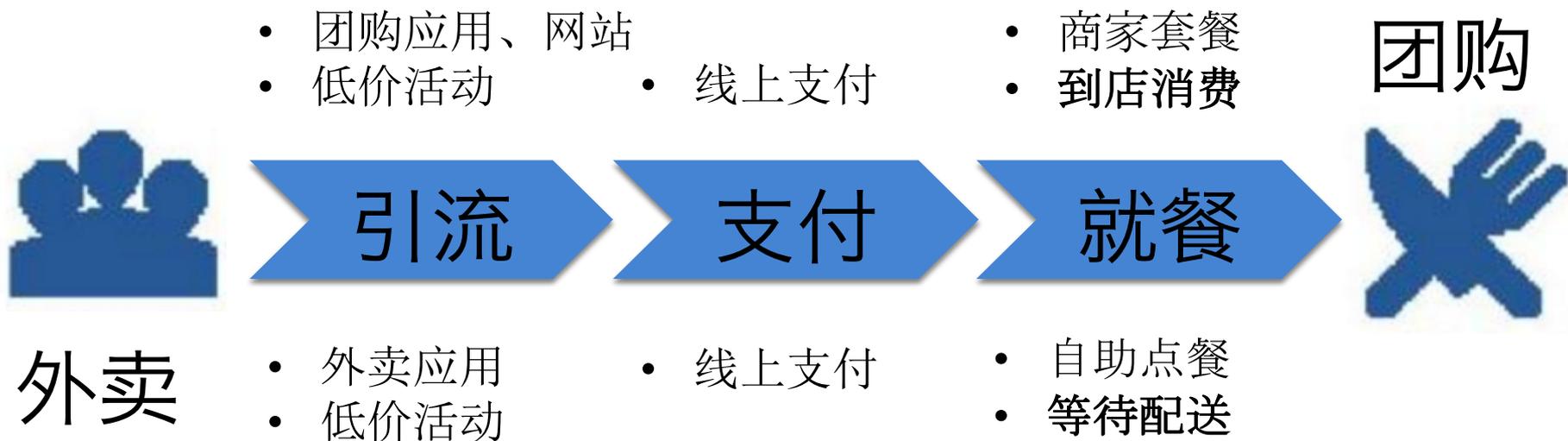


来源：艾瑞网 2015年中国外卖O2O行业发展报告

---

# 外卖与团购的异同

# 外卖与团购的异同

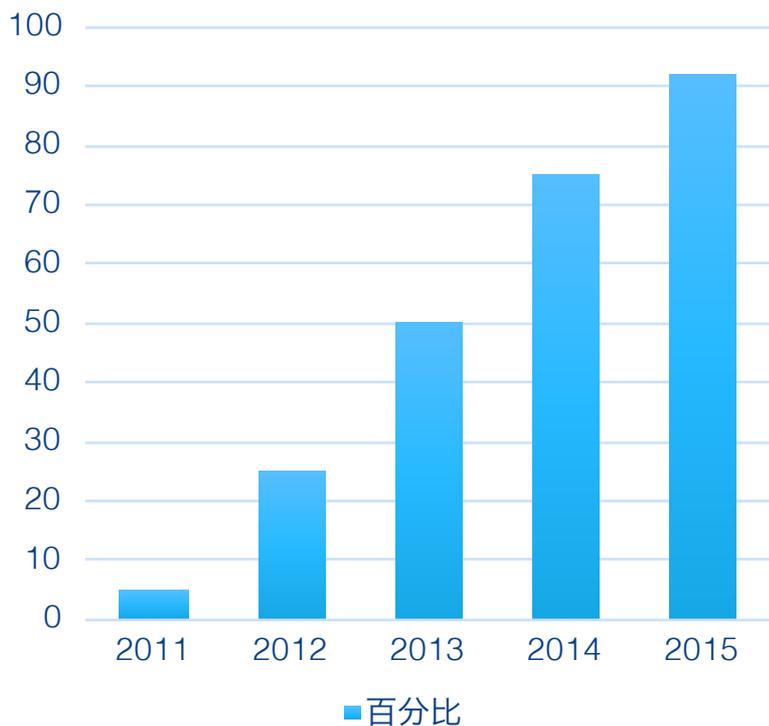


# 外卖与团购的异同

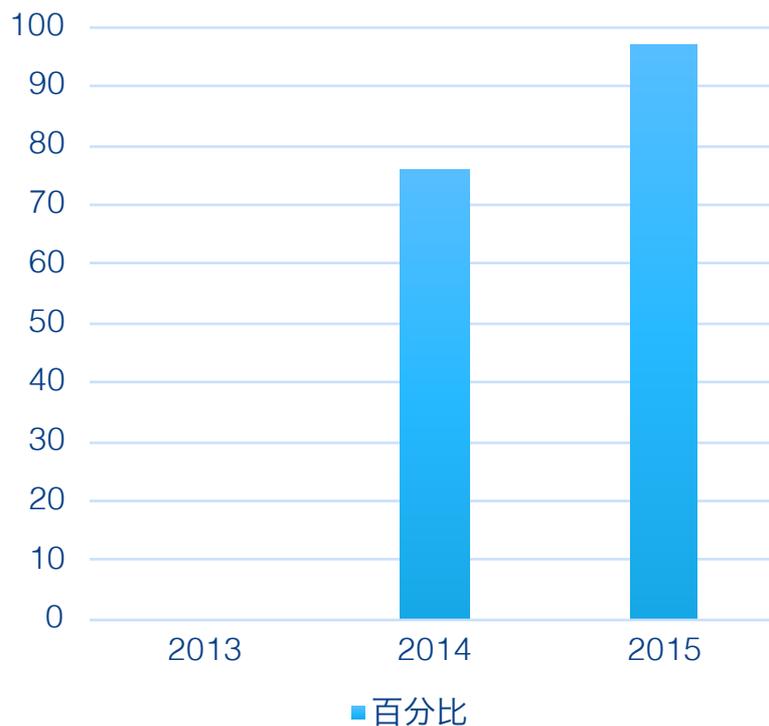
- 移动化
- 本地化
- 场景化
- 时间刚性
- 资源约束

# 移动化

## 美团团购 移动端交易占比

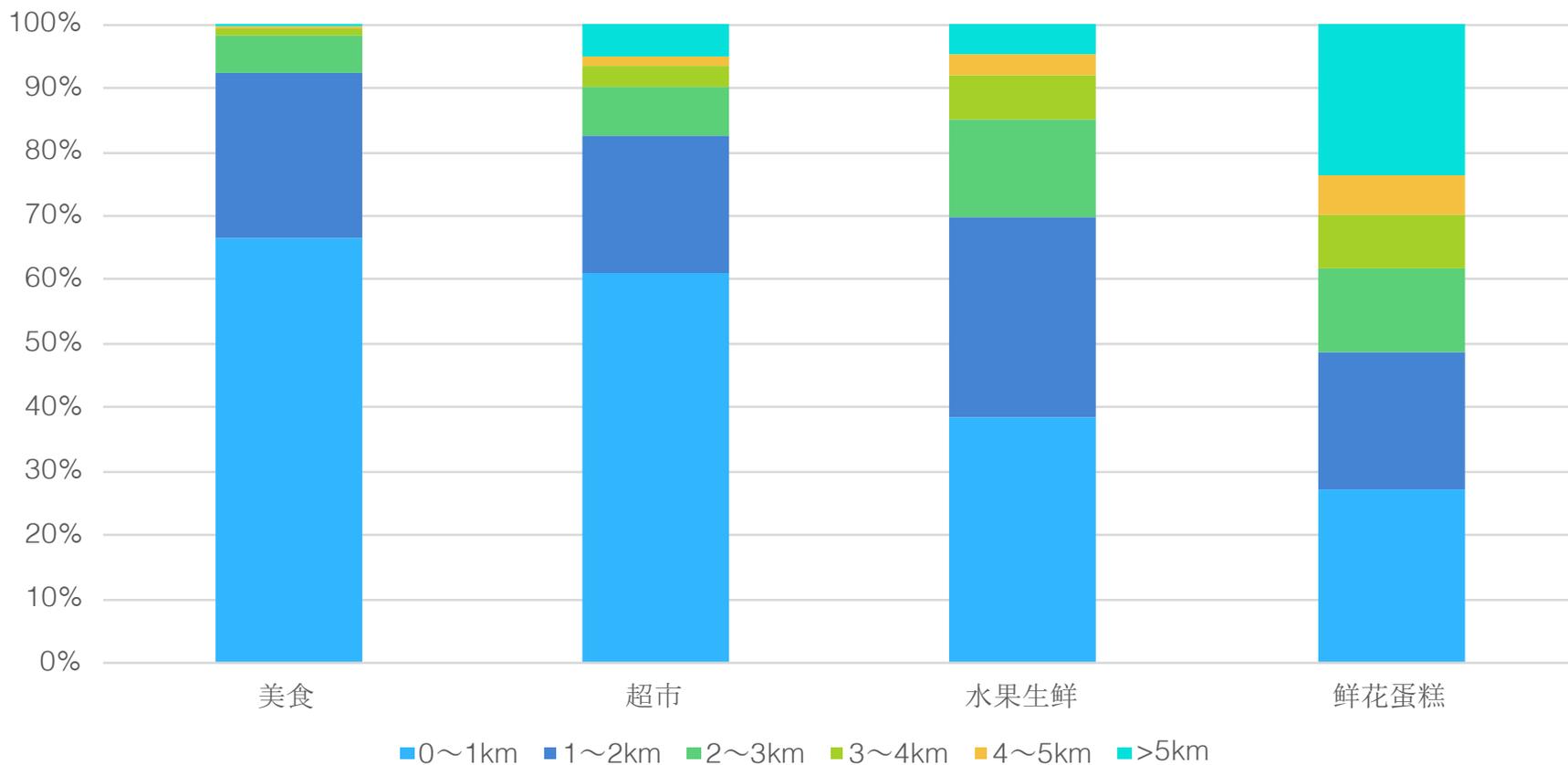


## 美团外卖 移动端交易占比



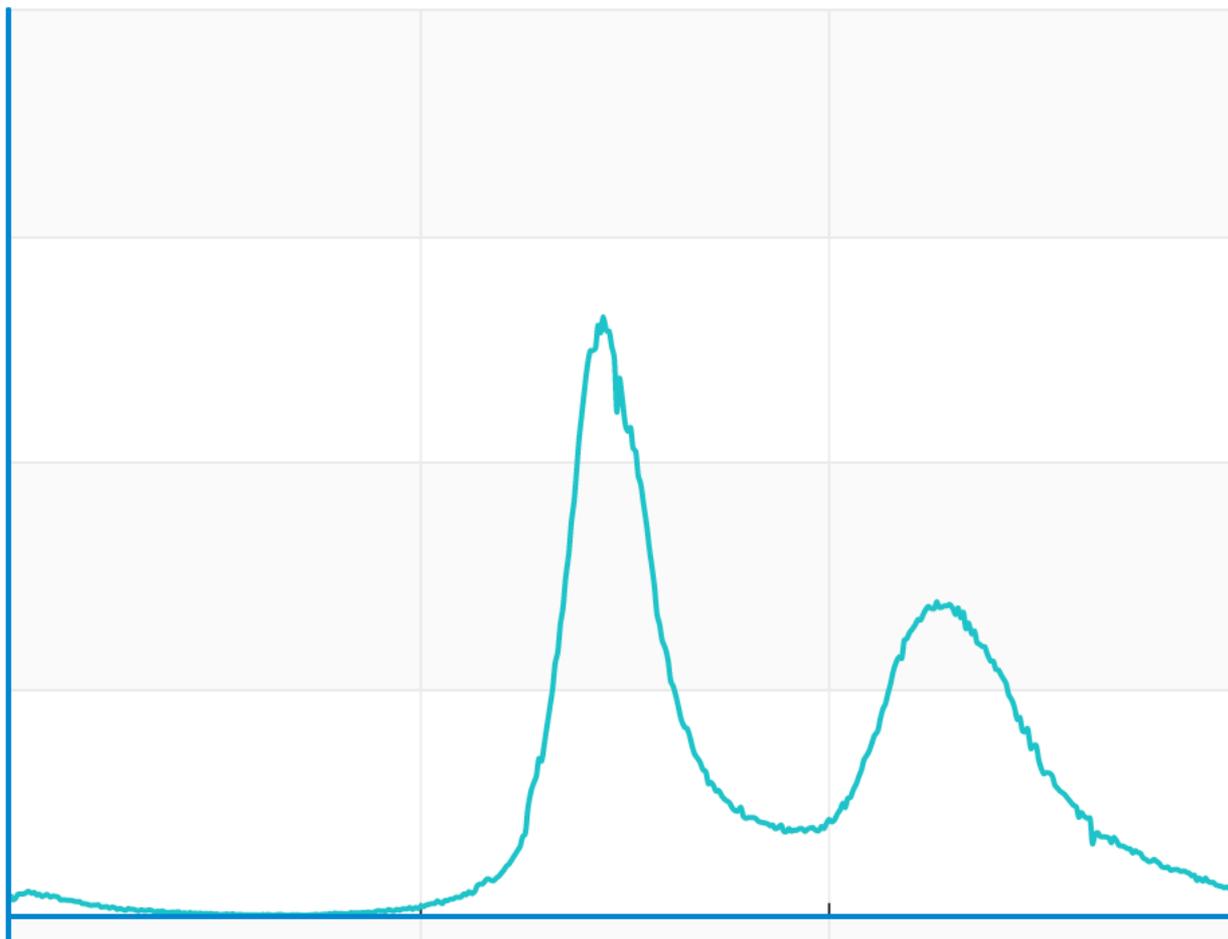
# 本地化

## 美团外卖各品类订单距离分布占比图



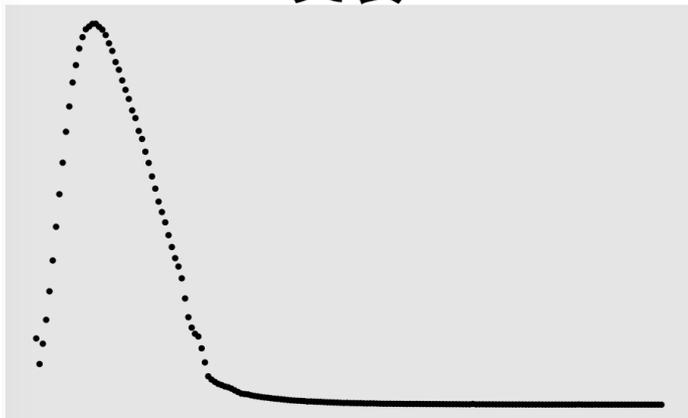
# 场景化

## 美团外卖单天订单趋势图

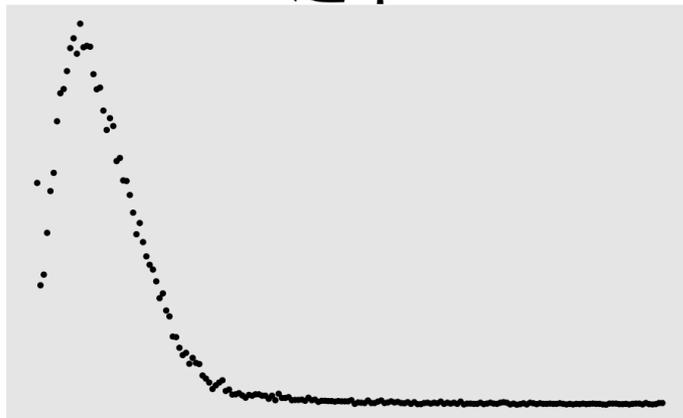


# 时间刚性

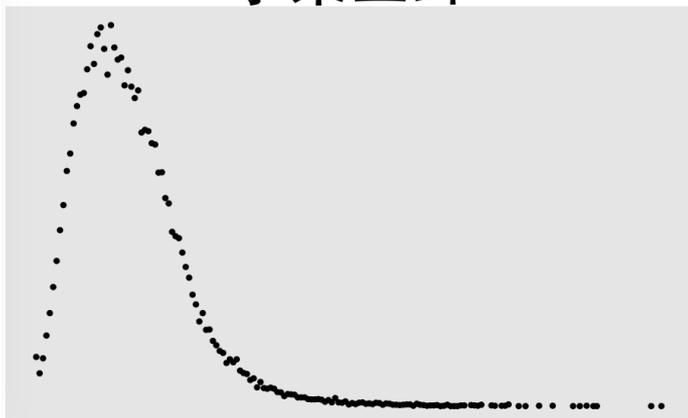
美食



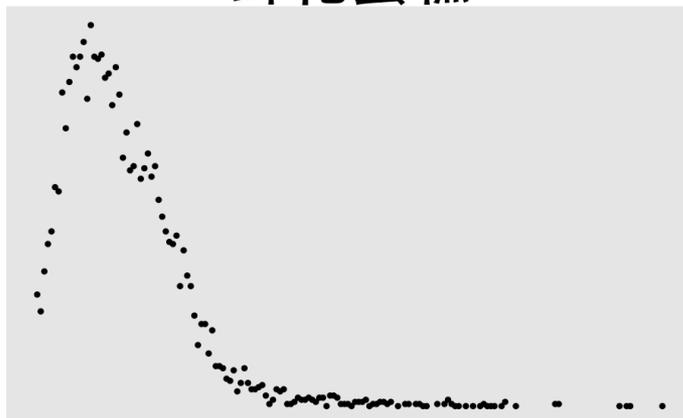
超市



水果生鲜



鲜花蛋糕



# 资源约束

- 食材库存
- 厨房产能
- 配送运力

订餐 出餐 送餐



---

# 外卖商家排序的技术演进

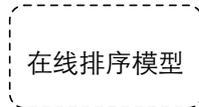
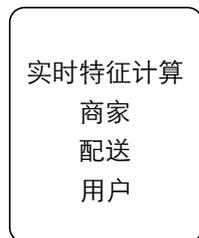
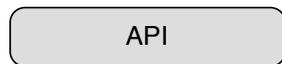
# 产品形态



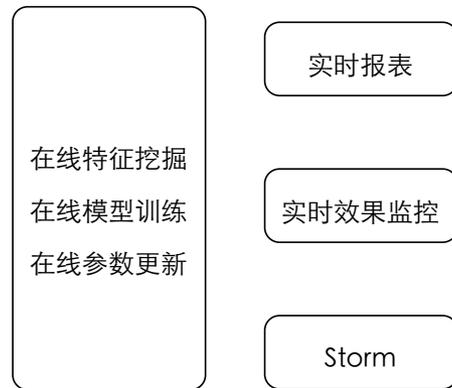
# 技术框架

## 在线服务响应流程

### 离线数据处理



### 在线数据处理



业务数据

APP日志

API日志

Service日志

# 在线服务

- A/B Testing
  - 做AA测试，坚持做
  - 重视抽样误差
  - 关注时间周期效应

```
rule "RULE_WORKER_AA_BASELINEV2"
  salience 2
  when
    construct: java.util.HashMap(construct.get("sceneId") == "HOMEPAGE", s
  then
    construct.put("rank_strategy_tag", "22");
    construct.put("rule_name", "RULE_WORKER_AA_BASELINEV2");
    construct.put("strategyId", "HOMEPAGE_SCENE_BASELINE_STRATEGY_CONFIG");
    construct.put("rank_strategy_version", "10.0.0");
  end
|
rule "RULE_WORKER_AA_BASELINE"
  salience 2
  when
    construct: java.util.HashMap(construct.get("sceneId") == "HOMEPAGE", s
  then
    construct.put("rank_strategy_tag", "23");
    construct.put("rule_name", "RULE_WORKER_AA_BASELINE");
    construct.put("strategyId", "HOMEPAGE_SCENE_BASELINE_STRATEGY_CONFIG");
    construct.put("rank_strategy_version", "10.0.0");
  end
```

# 在线服务

- 插件化策略模块
  - 模块分解可读性 v.s. 代码性能优化
  - 重视接口约束合约
  - 关注插件链的前后依赖

```
scene:"HOMEPAGE",
strategyId:"HOMEPAGE_SCENE_ORD_RERANK_COI
actionChainConfig:{
  "configs":[
    {"name":"Bl", "params":{}},
    {"name":"Ba", "params":{}},
    {"name":"Bo", "params":{"BOOST"},
    {"name":"De", "params":{"V2"},
    {"name":"Gb", "params":{}},
    {"name":"Co", "params":{}},
    {"name":"Le", "params":{}},
    {"name":"Fi", "params":{"unerV2"},
    {"name":"Ch", "params":{"ipTuner"},
    {"name":"Fi", "params":{"Tuner"},
    {"name":"Ne", "params":{"ationTuner"},
    {"name":"CP", "params":{}},
    {"name":"Br", "params":{}},
    {"name":"Po", "params":{}}
  ]
}
```

# 在线服务

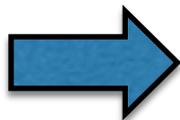
- 调试系统
  - 结果的可解释性
  - 业务、产品的协同

GeoHash:	<input type="text"/>	经纬度	场景:	主列表
userId:	<input type="text"/>	任意数字	uuid:	<input type="text"/>
debugLevel:	<input type="text"/>	(1或-1)	时间:	<input type="text"/>
策略字符串	请选择		<input type="checkbox"/> 策略对比	
筛选/对比:	<input type="text"/>	(逗号分隔)	Wiki链接:	<a href="#">使用Q&amp;A</a>
商家ID列表:	<input type="text"/>			

# 离线流程

- 流量清洗
- 抓取流量
- 无效加载

最后点击/下单

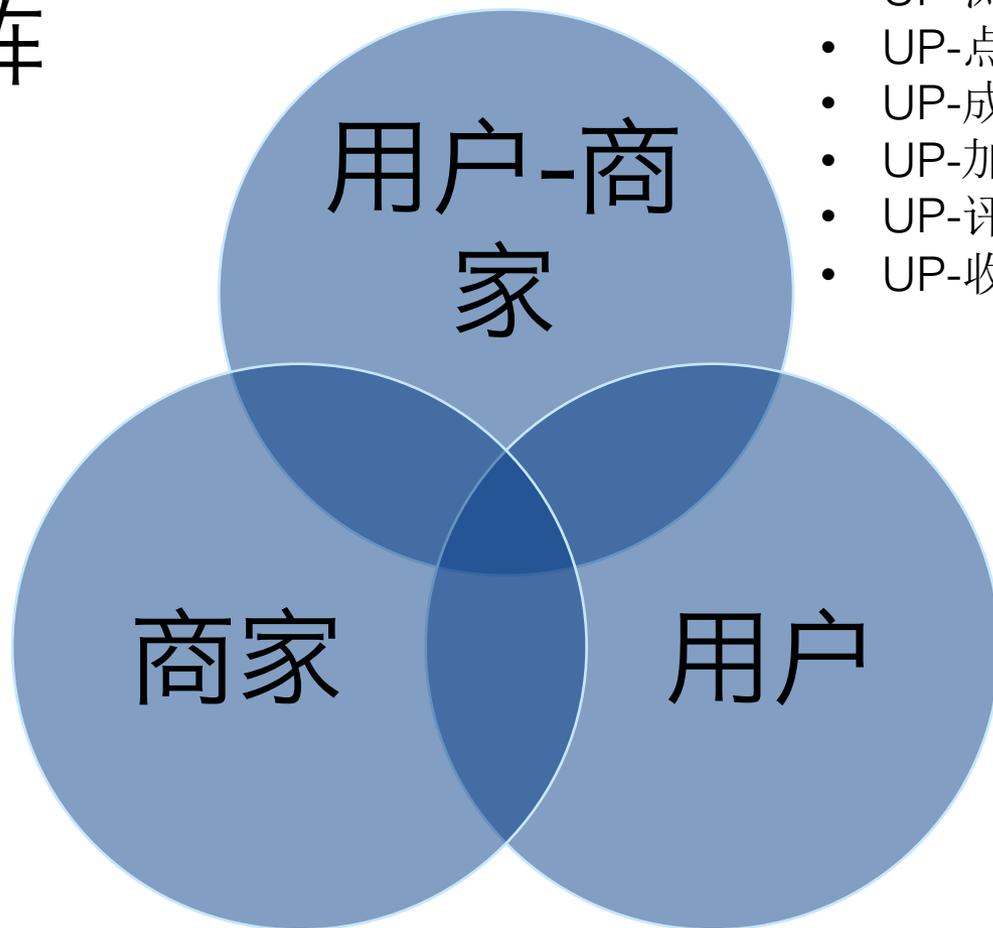


skip above  
+2



# 离线流程

- 特征矩阵



- P-基本信息
- P-转化率
- P-优惠活动
- P-用户评价
- P-热门信息
- P-竞对信息
- P-配送信息
- P-服务质量

- UP-浏览行为
- UP-点击行为
- UP-成单行为
- UP-加购行为
- UP-评论行为
- UP-收藏行为

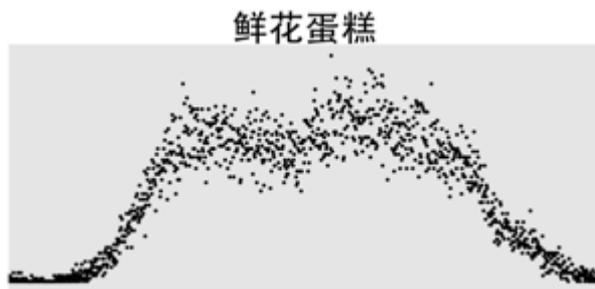
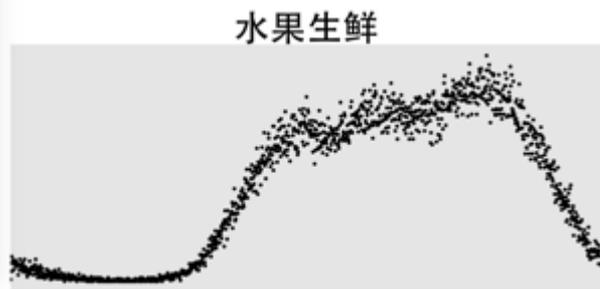
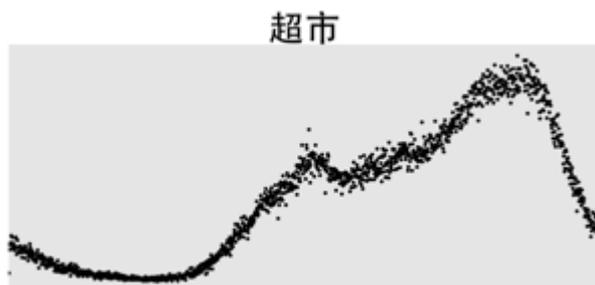
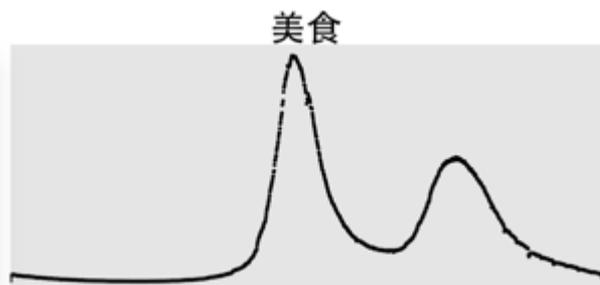
- U-人口统计
- U-品类偏好
- U-购买力
- U-优惠偏好
- U-上下文信息

# 离线流程

- GBDT——机器学习模型Startup
  - Decision Tree 简单贪婪
  - Bagging 民主智慧
  - Boosting 知错就改
  - Shrinkage 小步快跑
- 非线性模型 & 特征选择功能

# 效果里程碑

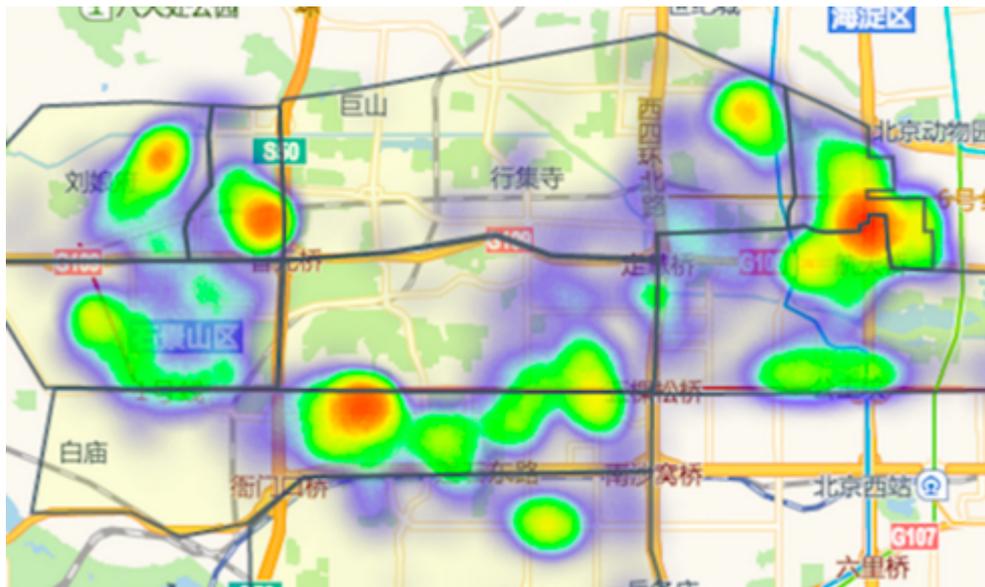
- 时段特征交叉



- 特征工程
- 按照小时数hour划分用户行为数据，取前后n个小时数据进行衰减后线性加权
- 按照星期数weekday划分用户行为数据，按照不同的相似度关系衰减后线性加权
- 问题
- 商家扩展新的营业时段

# 效果里程碑

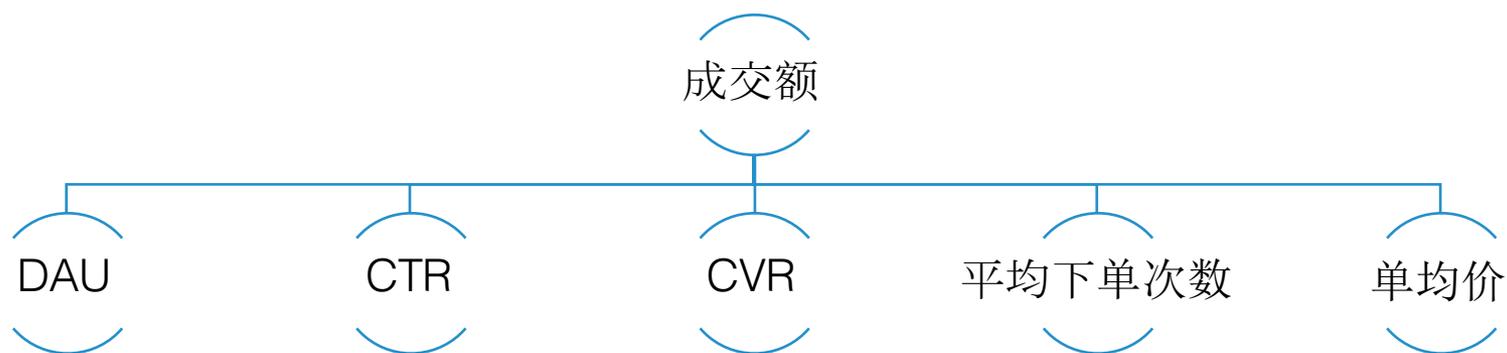
- 距离特征交叉



- 特征工程
- 按照距离区间分段确定转化率基准，供平滑参考
- 按照GeoHash划分区域网格，取上下前后左右共计9个方格确认区域用户行为热度数据
- 问题
- 计算和存储开销明显增大

# 效果里程碑

- CTR还是ROI
  - 基于用户-商家预测下单金额的重排序



---

# 外卖商家排序的算法挑战与应对

# 排序稳定性

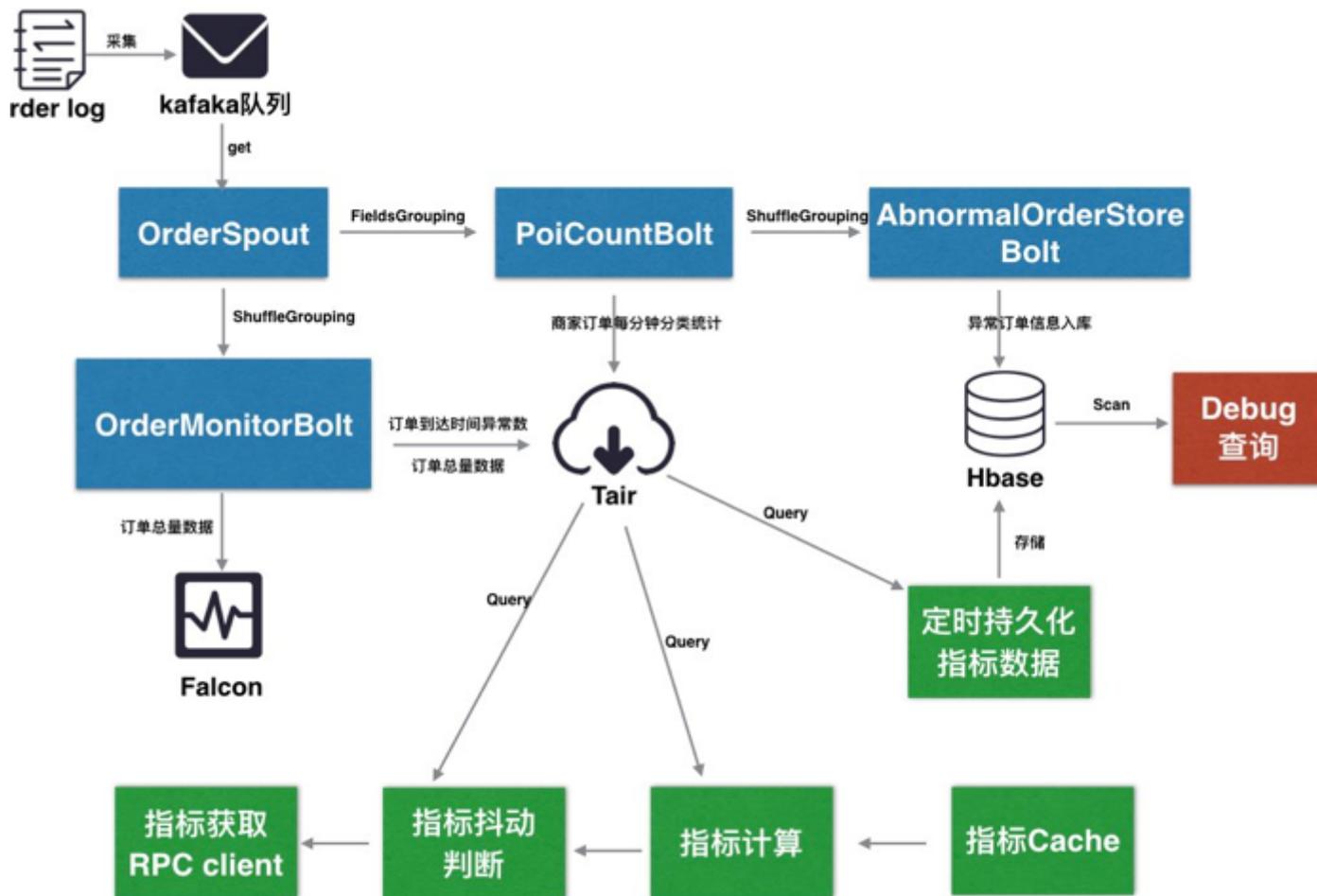
- 排序结果不稳定的成因
  - 不同人——个性化策略
  - 不同地点——LBS检索、商家配送范围
  - 不同时间——时段、星期、场景
  - 不同竞对营业情况——同行歇业
- 效果的稳定性 & 结果的可解释性

# 特征实时化

---

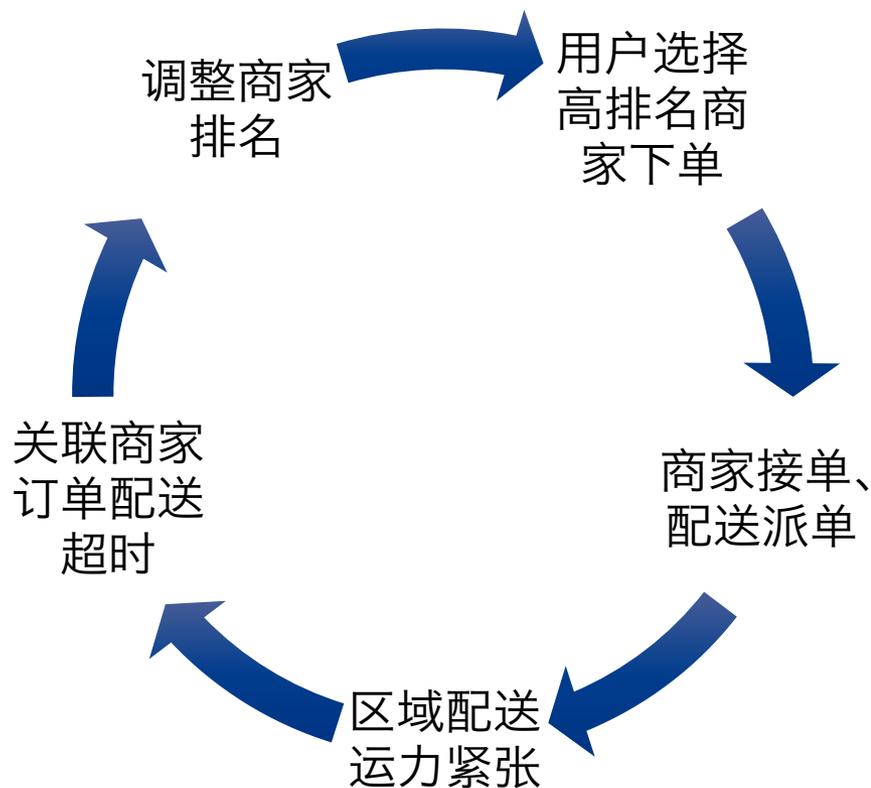
- 商家实时库存
- 商家服务质量
- 商家活动情况
- 配送运力情况

# 特征实时化



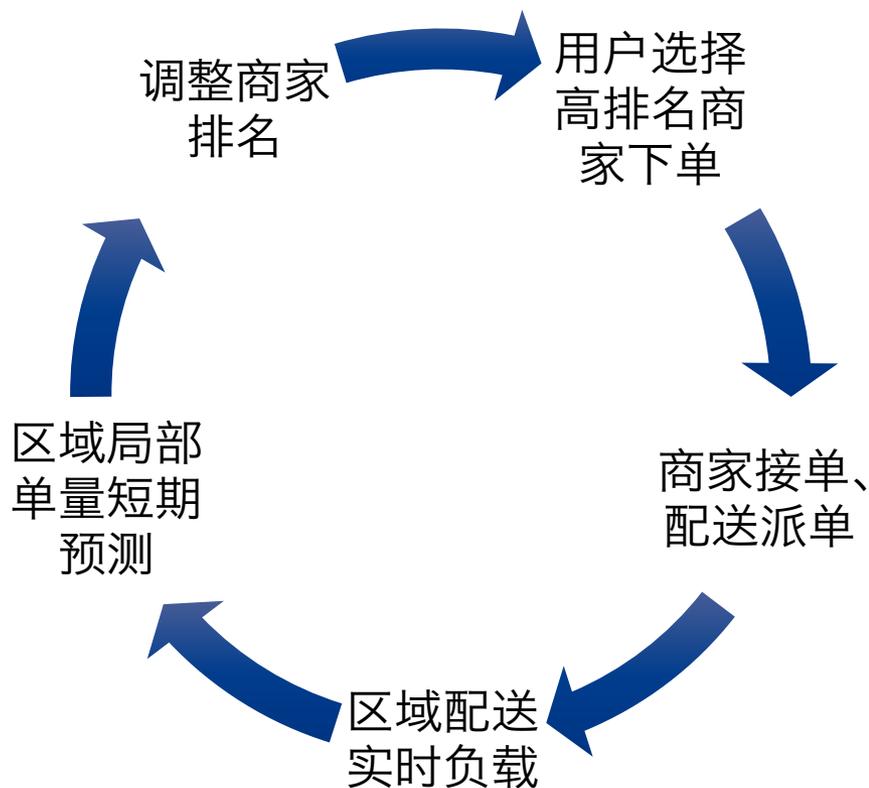
# 跨端系统反馈

- 配送拥塞控制
  - 配送实时负载指标
  - 配送等待时间
  - 配送预计送达时间



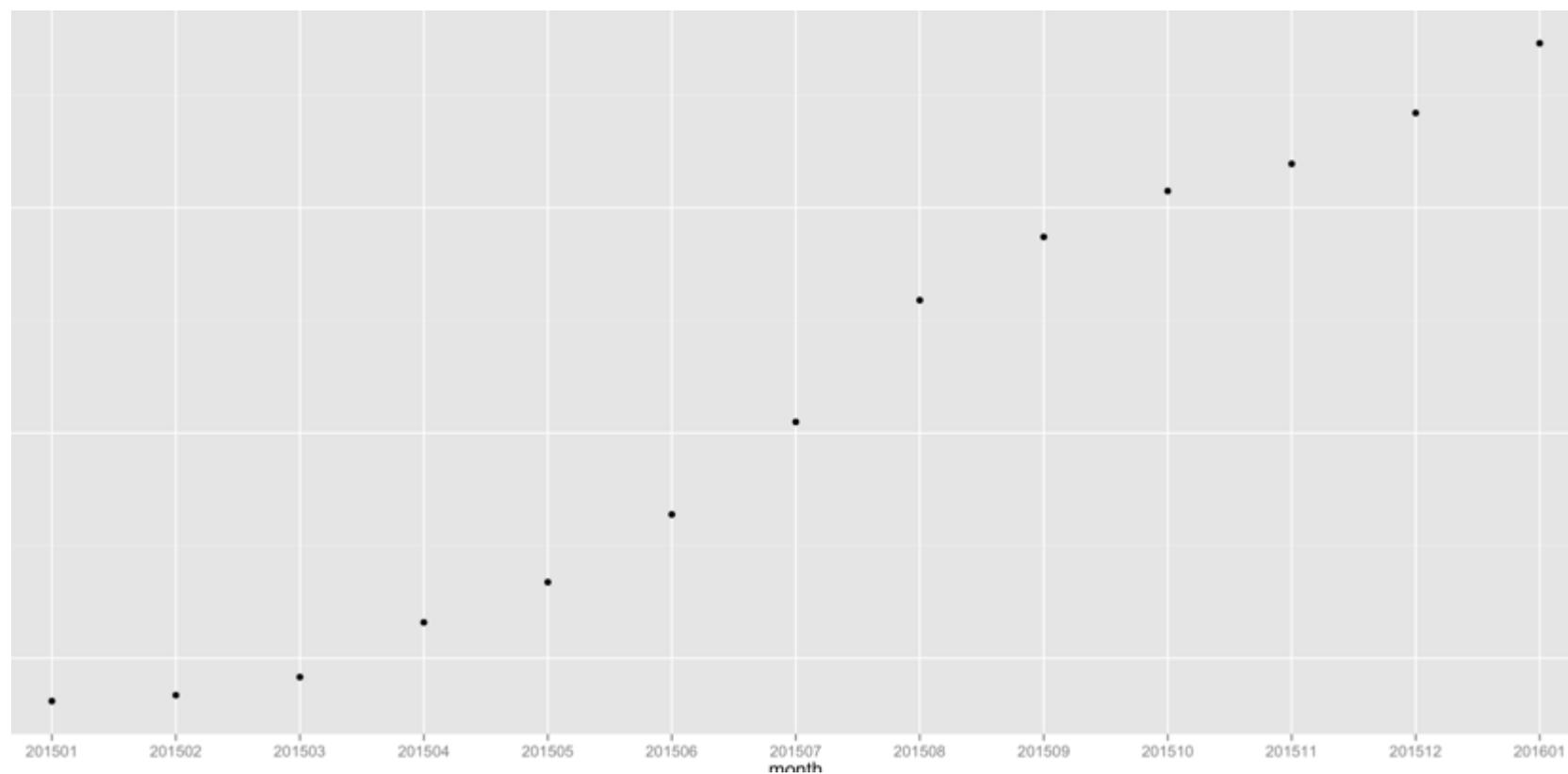
# 跨端系统反馈

- 商家库存售罄预测
- 单量预测



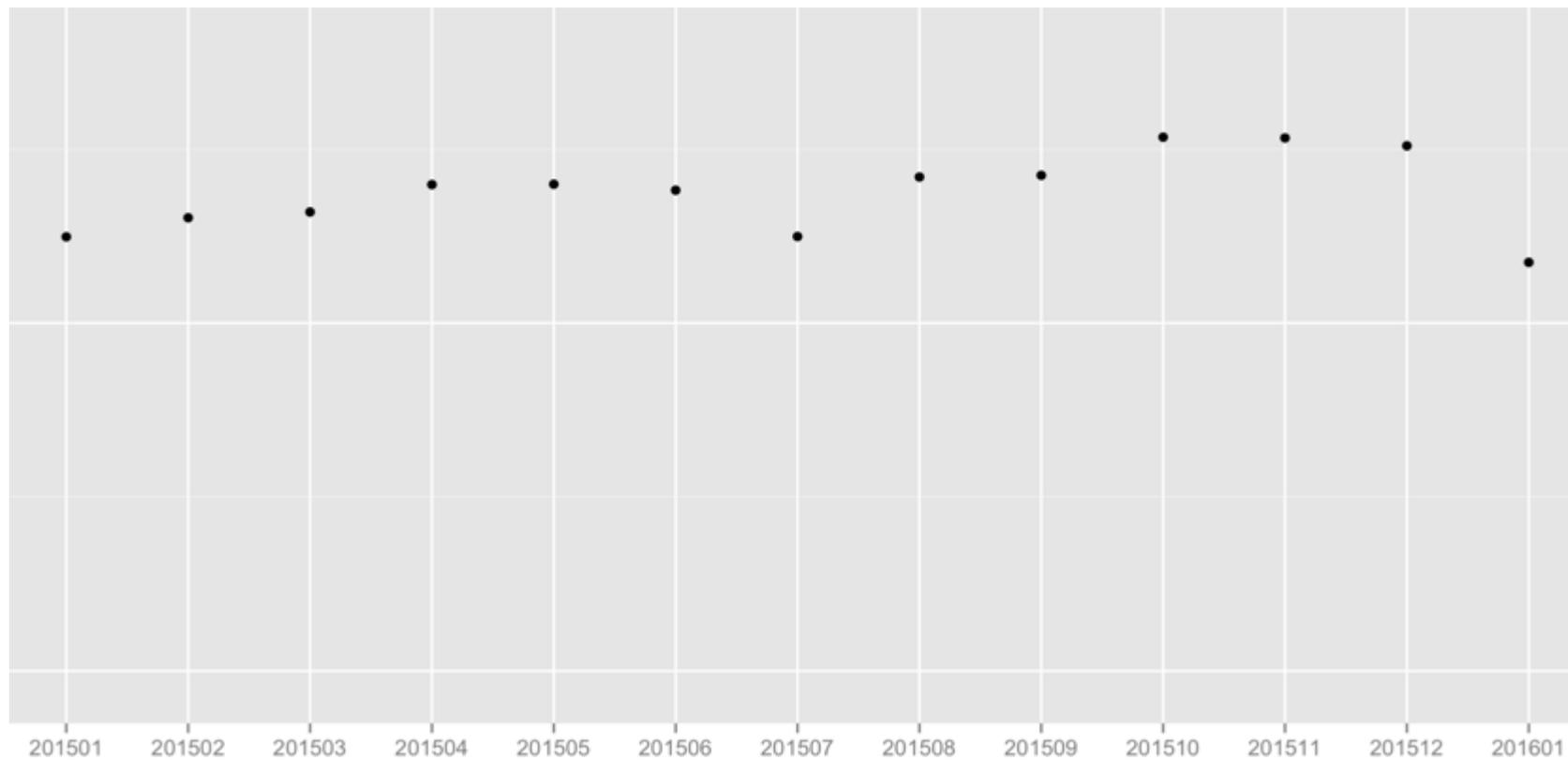
# Explore & Exploit

- 外卖商家供给的演变



# Explore & Exploit

- 外卖平台的基尼系数



# Explore & Exploit

- 多来源数据融合解决冷启动
  - 数据重合覆盖度
  - 迁移学习的可信度
- 基于用户好奇心偏好的探索
  - 专一 v.s. 尝鲜

---

# 小结

# 小结

---

- 技术驱动的前提理解业务
- 做正确的事，不做容易的事
- 在正确的时间做正确的事
- 关注模型策略对数据分布的影响



**THANKS!**