中国平安 PING AN
保险·银行·投资

专业 让生活更简单

基于MongoDB的好房推荐系统

平安好房技术中心
刘诚杰

2018年7月21日

CONTENTS

1 推荐系统介绍

推荐系统是什么？

大数据

机器学习

淘宝、亚马逊

K最近邻算法、协同算法

高、深、复杂

推荐系统是什么？

~~尿布和啤酒~~ 这个例子用烂了

基于用户的协同过滤

2个Tips

1、还有一个月就要七夕了

2、你与丈母娘只差一套房的距离。

好房推荐助你一臂之力

算法低相关

看了又看

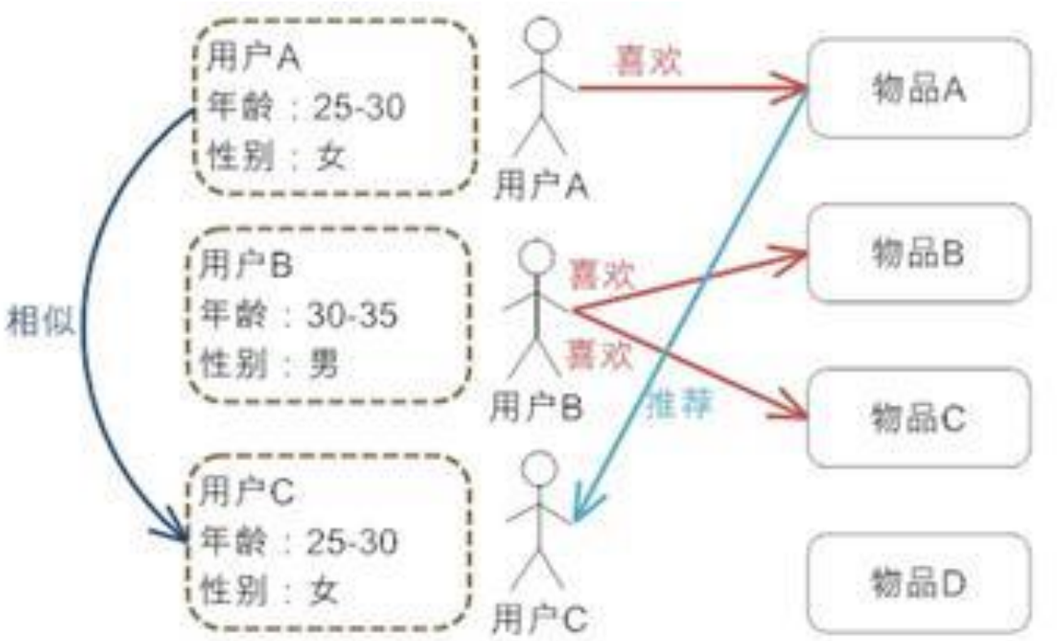
热门推荐

人工推荐

~~竞价排名~~

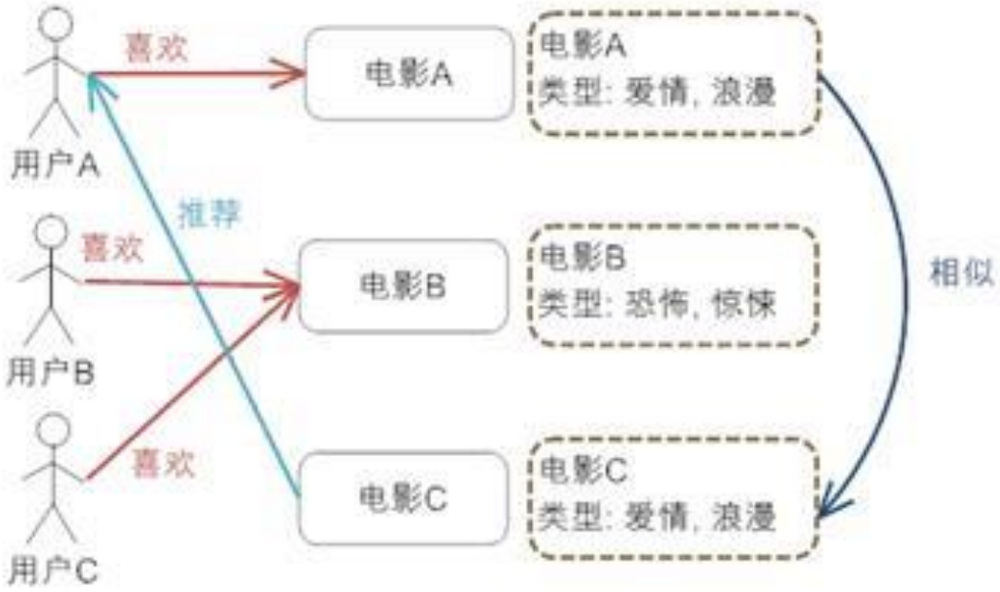
常用推荐算法

基于用户—快速、低效



用户画像

基于内容—精准、片面



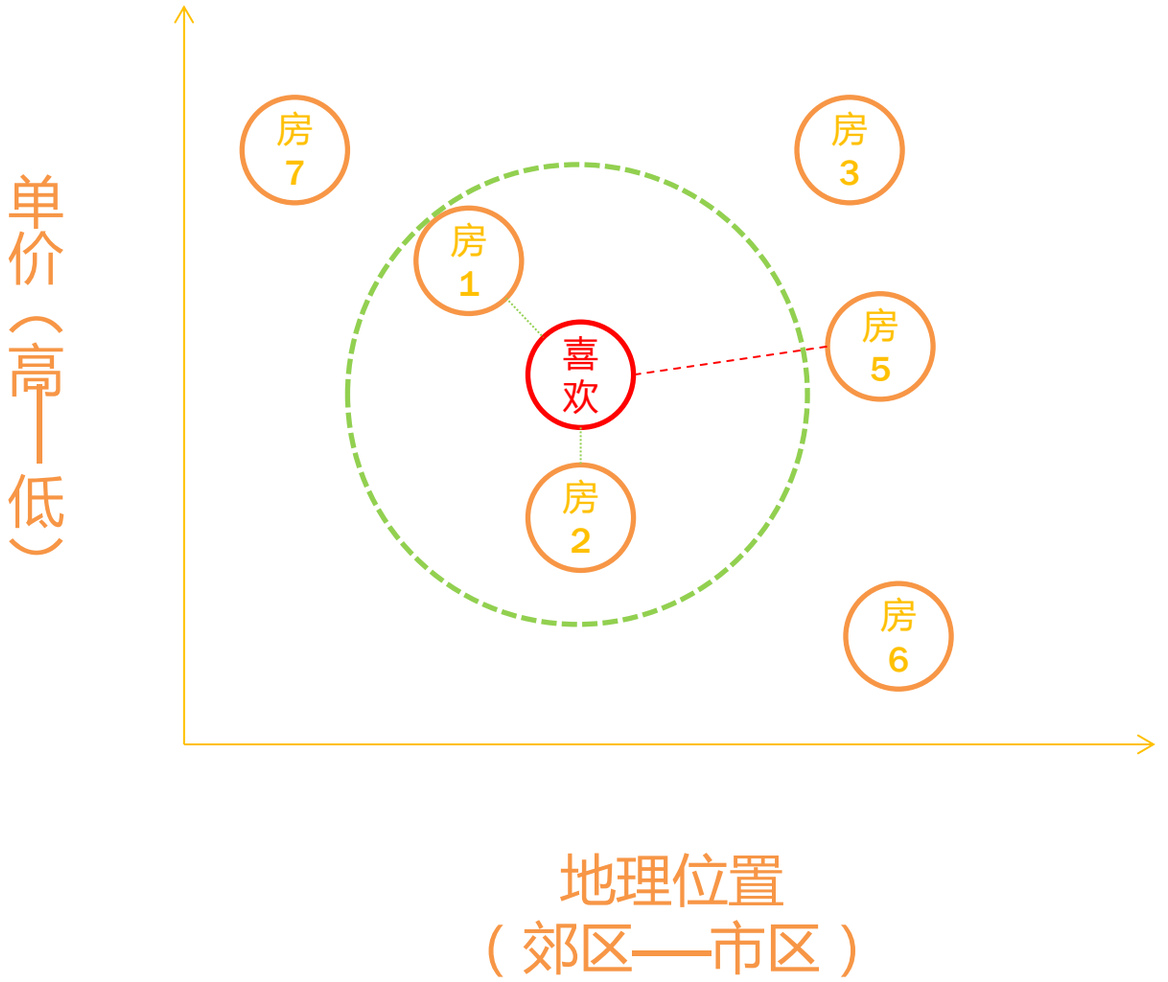
对象特征

协调过滤：再加上历史偏好数据

好房推荐系统基于内容

算法实现

K最近邻算法（KNN）



推荐几何距离更近的房1和房2

可以使用搜索引擎实现寻找相似的内容

推荐的意义

无目的性

信息过载

提高转化率

提升用户体验

2 好房推荐系统

推荐必要性

不必要

- 商品单一，总量少
- 用户对房需求明确（价格，区域已知）

必要

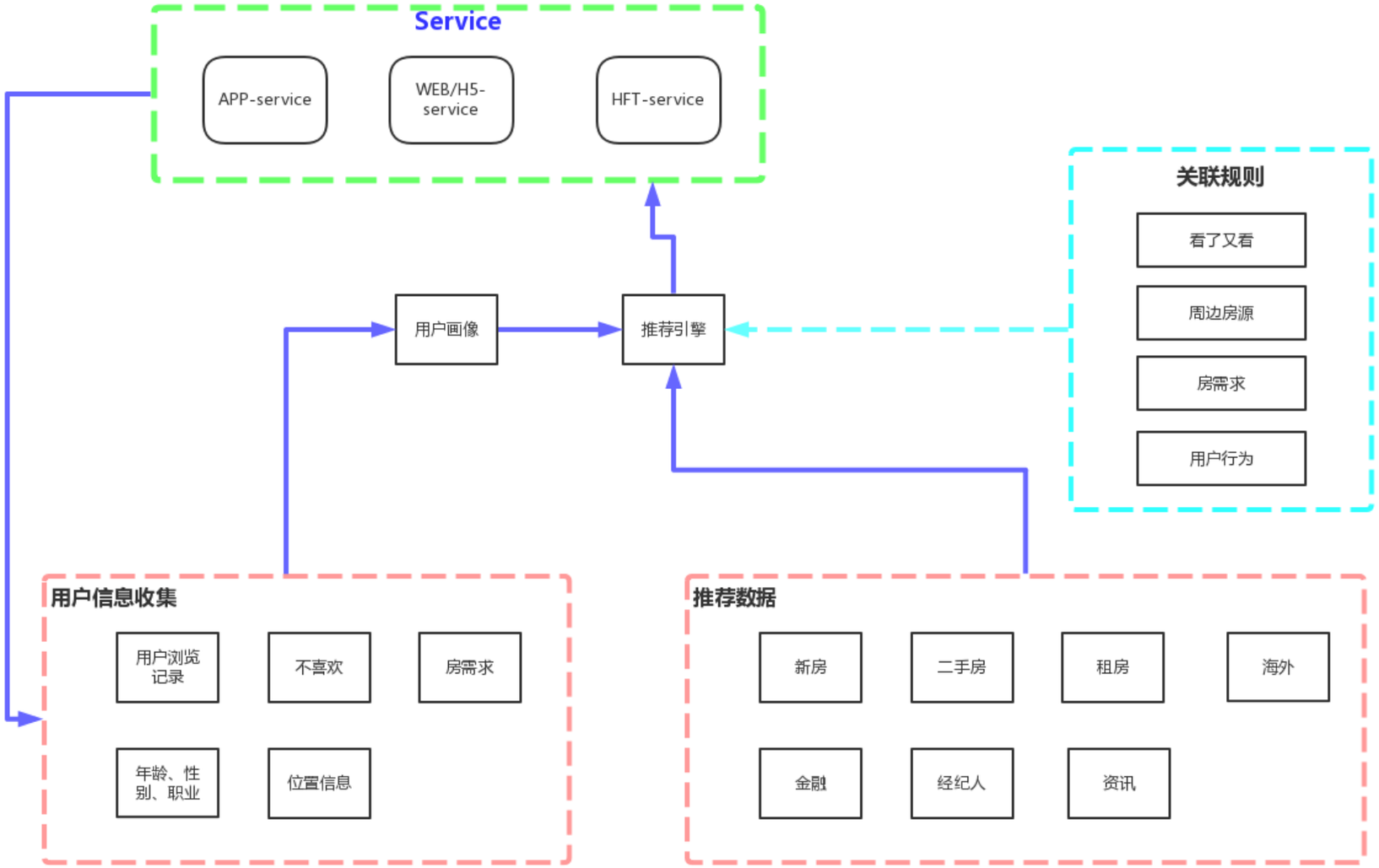
- 信息过载，解决长尾问题
- 房是超高额商品，决策和浏览信息需要很长的时间
- 经纪人、资讯等后期也可以接入推荐

推荐效果

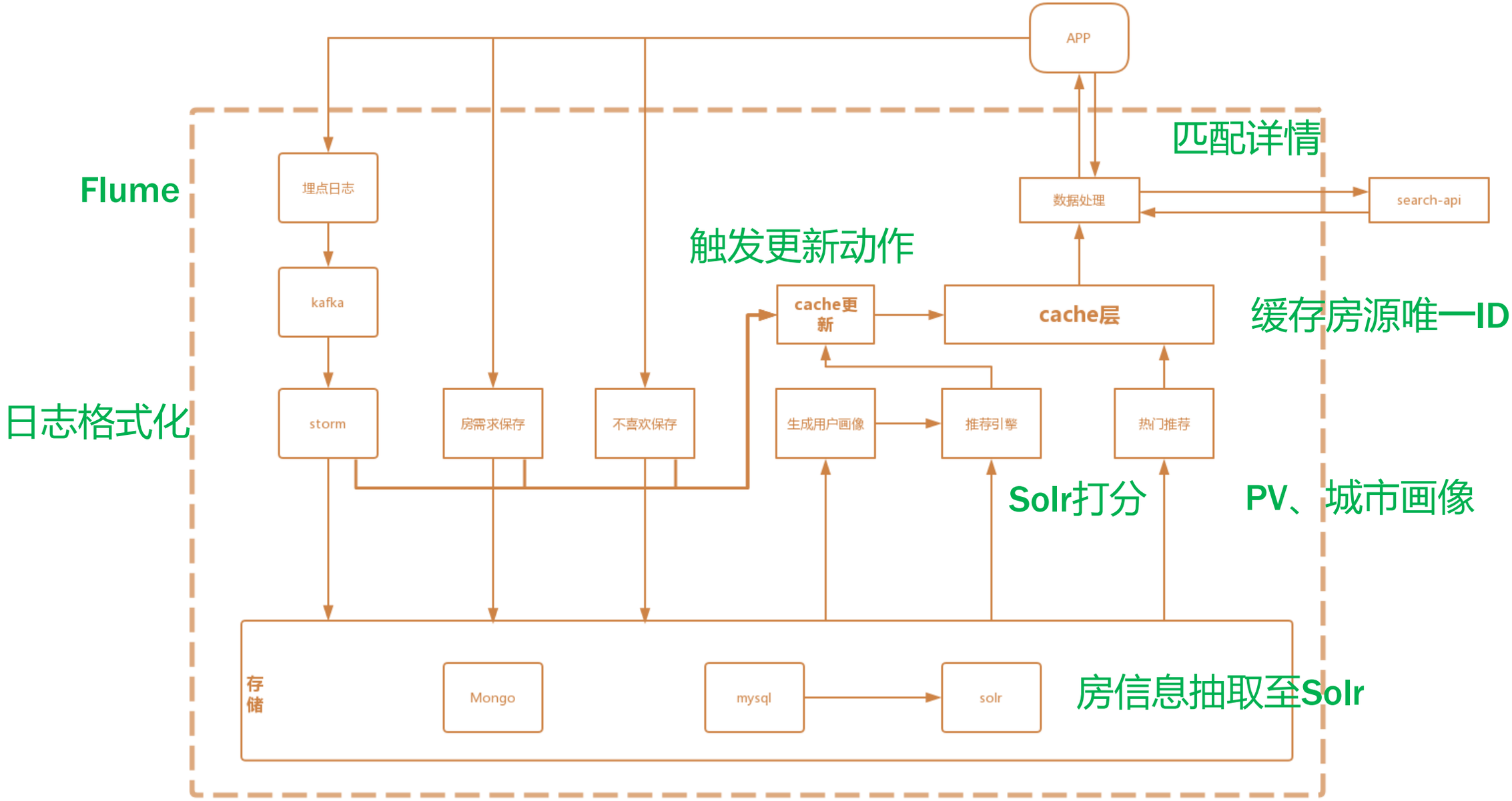
- 首页到详情页的转化暴涨8倍，远超其他列表页
- 用户留存时间显著延长

推荐很有必要

推荐业务结构图

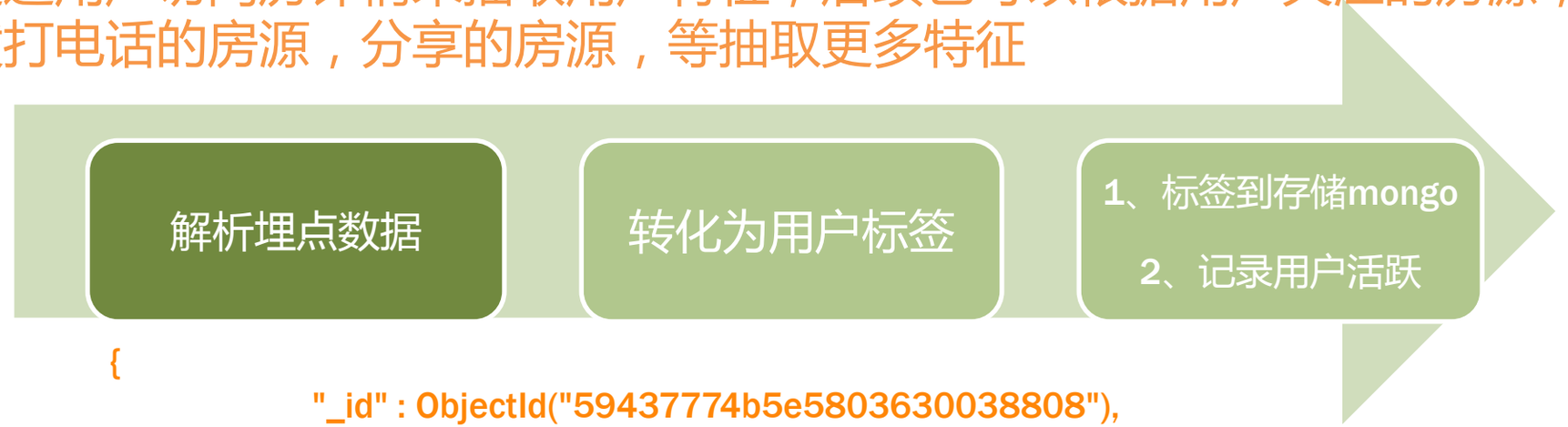

Service
APP-service
WEB/H5-service
HFT-service
关联规则
看了又看
周边房源
房需求
用户行为
用户画像
推荐引擎
用户信息收集
用户浏览记录
不喜欢
房需求
年龄、性别、职业
位置信息
推荐数据
新房
二手房
租房
海外
金融
经纪人
资讯

推荐数据流图


APP
匹配详情
Flume
埋点日志
数据处理
search-api
触发更新动作
kafka
cache更新
cache层
缓存房源唯一ID
日志格式化
storm
房需求保存
不喜欢保存
生成用户画像
推荐引擎
热门推荐
Solr打分
PV、城市画像
存储
Mongo
mysql
solr
房信息抽取至Solr

用户特征采集

通过用户访问房详情来抽取用户特征，后续也可以根据用户关注的房源，拨打电话的房源，分享的房源，等抽取更多特征




```
{
        "_id" : ObjectId("59437774b5e5803630038808"),
        "userId" : "869953021669940",
        "cityId" : "816",
        "recType" : "buy",
        "fangId" : "176907",
        "fangType" : "xf",
        "createTime" : NumberLong("1497504712"),
        "fangLabel" : {
                "area" : [ “50_70",  “70_90" ],
                "layout" : [ “2", "3" ],
                “price” : [ "70_90", "50_70" ],
                "region" : "510106"
        }
}
```

用户偏好

按条件选取的偏好设置


```
{
    "_id" : ObjectId("5b5201c5a8de7b1717eb3fe1"),
    "cityId" : "3566",
    "userId" : "9653181",
    "recType" : "buy",
    "appVersion" : "4.13.0",
    "createTime" : NumberLong(1532101061),
    "buyPreference" : {
        "layout" : {
            "3" : 1
        },
        "region" : {
            "441900" : 1
        },
        "fangType" : {
            "xf" : 1
        }
    }
}
```

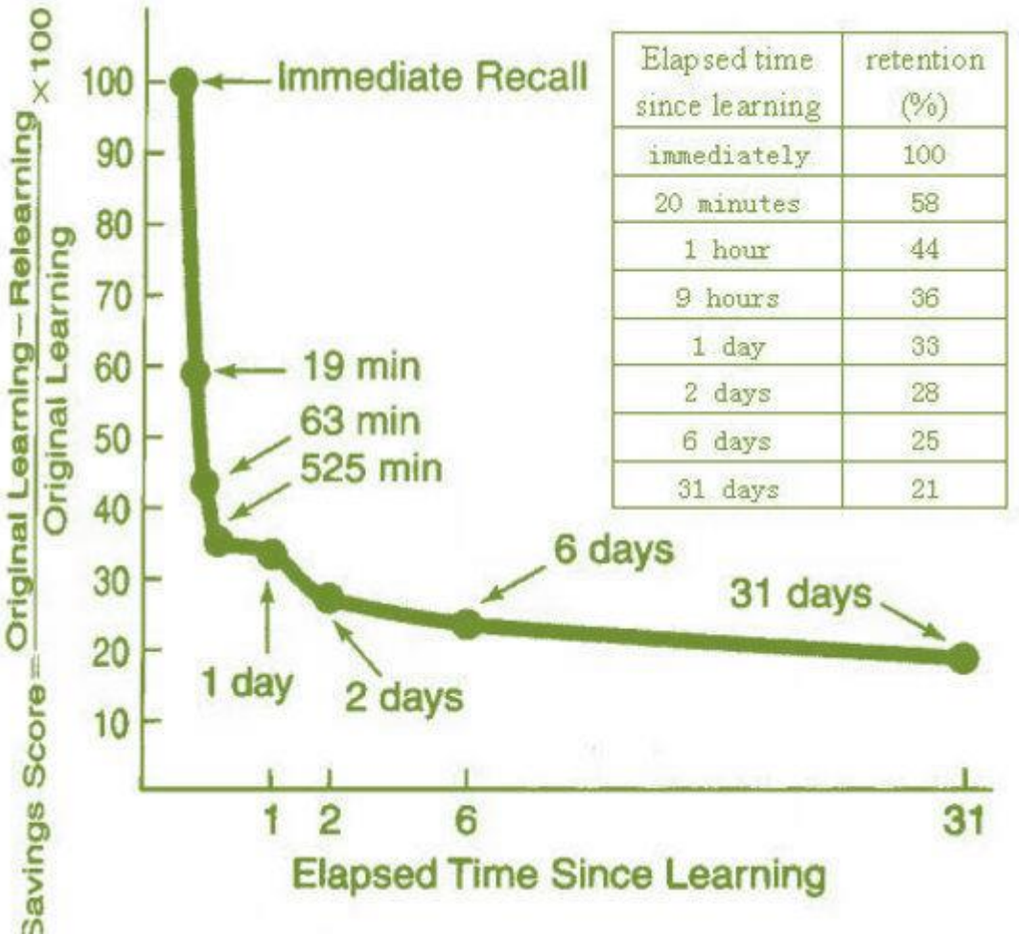
用户不喜欢设置


```
{
        "_id" : ObjectId("5b516a2dd97c2fde7025e711"),
        "userId" : "c9d18861fbfc7f7433076667620c7ce1d1fdee12",
        "cityId" : "2480",
        "fangId" : "20007688",
        "fangType" : "xf",
        "createTime" : NumberLong(1532062253)
}
```

用户画像（JAVA计算结果）


```
{
    "recType": "rent",
    "appVersion": "",
    "fangPreference": {
        "layout": {
            "1": "0.51827",
            "3": "0.28989",
            "4": "0.19184"
        },
        "fangType": {
            "zf_gg": "1.00000"
        },
        "price": {
            "1000_1500": "0.26096",
            "1500_2000": "0.34457",
            "3000_3500": "0.10789",
            "2000_2500": "0.16858",
            "2500_3000": "0.11800"
        },
        "rentType": {
            "1": "0.48347",
            "2": "0.51653"
        },
        "region": {
            "121": "0.15045",
            "166": "0.05058",
            "24": "0.12228",
            "147": "0.11524",
            "53": "0.56145"
        }
    },
    "notLike": {
    },
    "cityId": "1",
    "userId": "9825699855265588"
}
```


行为与偏好结合生成用户画像



Elapsed time since learning	retention (%)
immediately	100
20 minutes	58
1 hour	44
9 hours	36
1 day	33
2 days	28
6 days	25
31 days	21

两者按时间进行比重的衰减

房标签（Solr）


```
{       "area":["70_90",
         "90_110"],
        "loupan_name":"同润金色橘苑",
"layout_building_area":"9000,9200,8800,7800",
        "average_price":"2500000",
        "layoutSource":"3-2-1,3-2-1,3-2-1,2-2-1",
        "cityID":"1",
        "layout":["2",
         "3"],
        "fangType":"xf",
        "max_price":"3200000 ",
        "min_price":"2100000",
        "iupdateTime":1530084093,
        "price":["150_170",
         "170_190",
         "190_210"],
        "id":"209881",
        "icreateTime":1489572812,
        "index_id":"xf_zy_209881",
        "region":"310115",
        "dataUpdateTime":"2018-06-27T15:37:07.718Z",
        "_version_":1604410438038061057}
```


Solr计算（打分）结果


```
"response":{"numFound":76752,"start":0,"max
Score":201.2,"docs":[
{ "community_name":"宁工新寓（二村）",
"regionName":"齐齐哈尔市,,",
"community_address":"草场门大街128号",
"rentType":"1", "cityID":"6636",
"price_source":"3200", "layout":["2"],
"fangType":"zf_zy_f_t",
"iupdateTime":1504195279,
"price":["3000_3500"], "loupanId":"43598",
"id":"23876", "count_date":"",
"icreateTime":1500273241,
"index_id":"tj_23876", "region":"0",
"dataUpdateTime":"2018-05-
21T10:09:56.770Z",
"_version_":1601037765726699526,
"score":1.199997,
"_freq_":2}
```


优先看符条件合项数freq
其次再看分数

其他事项

- 避免信息茧房（因为高分导致同类型房显示过多，按比例将多种类型房显示）
- 第一次访问的用户直接推送热门，热门十分钟更新一次
- 筛选掉推荐给用户，但是用户从不点击的内容

3 MongoDB使用心得

使用MongoDB原因

- JSON格式，可存储数组，且便于程序交互
- 无模式设计，用户行为表非常大，增加字段无代价
- 稳定的高可用与便捷的横向扩展
- 可直接与大数据进行对接（Hadoop-Connector）

TTL索引

```
db.t_user_behavior. createIndex( { "lastModifiedDate": 1 }, { expireAfterSeconds: 2592000 } )
```

时间到期后自动删除

后台触发器每60秒进行一次删除

仅在主节点进行触发

必须包含时间字段

无法进行联合索引

使用collMod修改过期时间

踩坑

Aggregation

- 3.6以前不支持hint，正常条件搜索后，使用_id排序，会使用错误索引
- 可以在排序前使用“$project” 先过滤使用的字段
- 聚合的数据量有限，除了使用落盘操作，也可以使用“$project” 先过滤

Hive（Hadoo-Connector）

- 删除Hive表时，MongoDB中也会删除
- MongoDB做好权限控制
- 连接从库
- 先删掉驱动包，再删Hive表

招人

平安好房了解一下？

- JAVA
- 测试
- 移动端
- 前端
- SA
- DBA

发送简历到
liuchengjie464@pingan.com.cn

感谢提供支持的技术中心同事

谢谢大家

mongoDB | IT大咖说 知识共享平台