

用户质量评分模型搭建

谢佳标 (Daniel.xie)

数据驱动下的最佳渠道优化策略

基于数据驱动最佳渠道优化策略

定位目标



获取数据



渠道检测

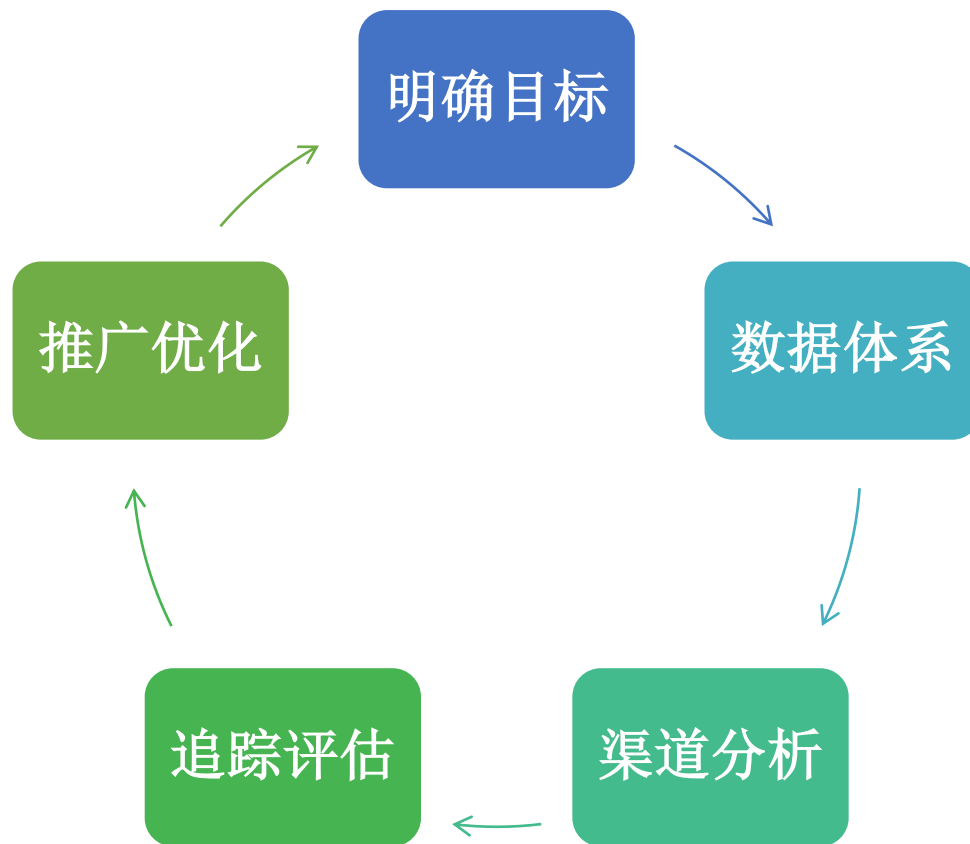


策略优化



深度推广

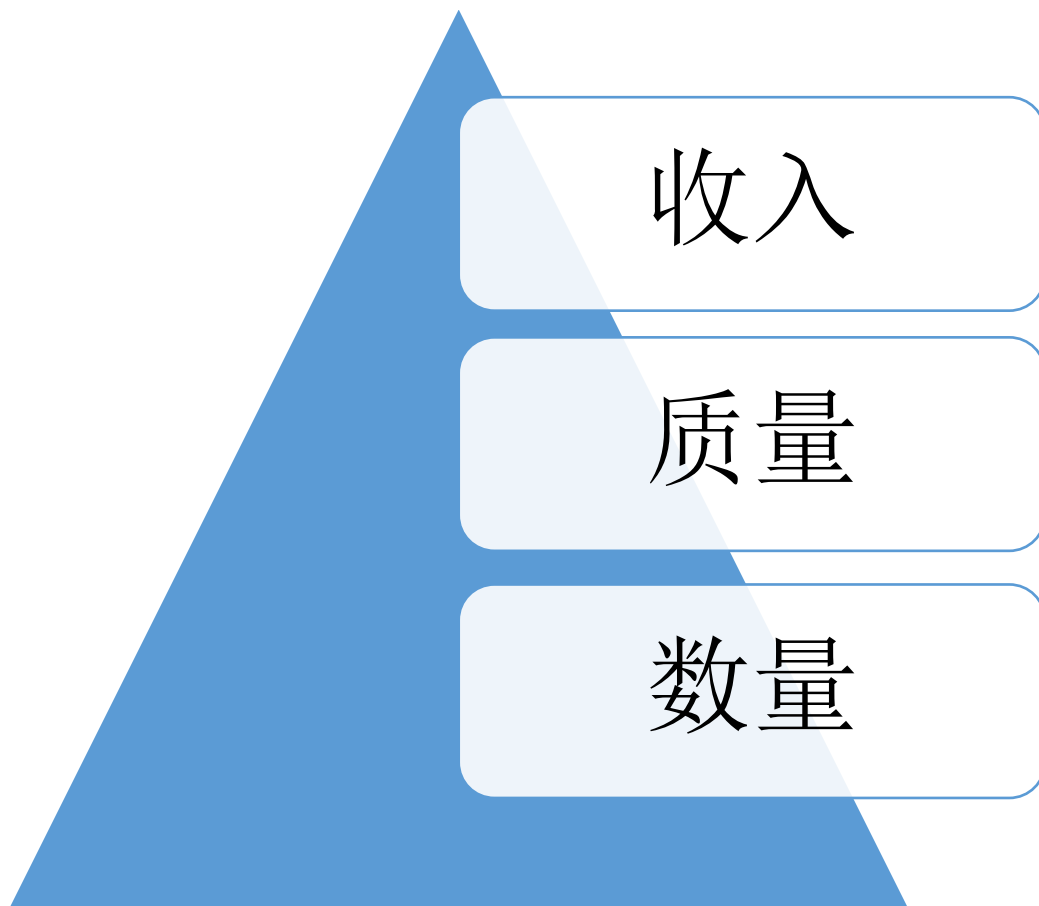
渠道推广分析的闭环



- 渠道分析-追踪评估-推广优化：这3部分是密不可分的，是整个分析闭环的核心；
- 所有的分析结果最终都要服务于ROI，对于一个优质的渠道，应该是在保证用户数量质量之外，更能给公司带来收入的渠道。

评估渠道质量的3部分

- 通过数量、质量和收入3大指标维度进行渠道的分析找出最优渠道，进而有效地推广策略优化、流量优化、实现利益最大化。



业务案例：利用四象限法研究渠道用户

- 我们利用Median-IQR方法分析ARPPU、ARPU、新增次日留存率和新增七日留存率等指标，查看不同渠道在一月份数据的集中及离散程度。这里使用中位数作为衡量中心的统计量，应用四分位距（IQR）作为离散指标的统计量更有意义。与更常用的均值和标准差相比，这些统计量在有离群值存在时更加稳健。
- 计算每个渠道所有数据的这两个统计量代码如下：

```
# 导入渠道日运营数据
```

```
channel <- read.csv("data/渠道日概况.csv",header=TRUE,fill=TRUE)
```

```
# 利用Median-IQR方法分析ARPPU、ARPU、新增次日留存率和新增七日留存率
```

```
# 自定义求中位数和四分位距函数
```

```
mystats <- function(x) c(Median=median(x,na.rm=T),IQR=IQR(x,na.rm=T))
```

```
# 求出各指标的统计值
```

```
library(doBy)
```

```
result <- summaryBy(ARPPU+ARPU+新增次日留存率+新增7日后留存率~渠道名称,
```

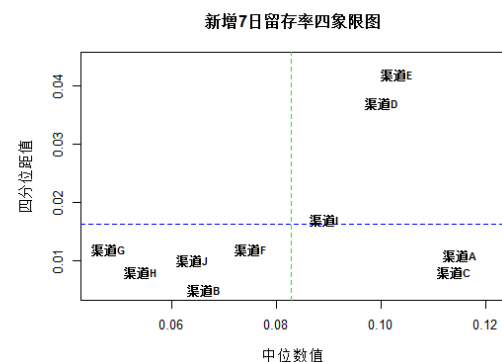
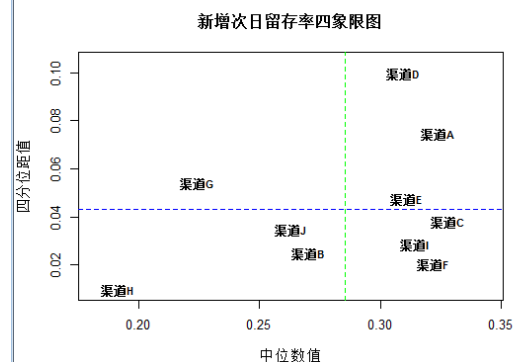
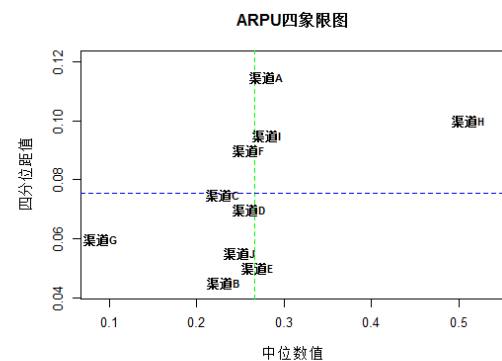
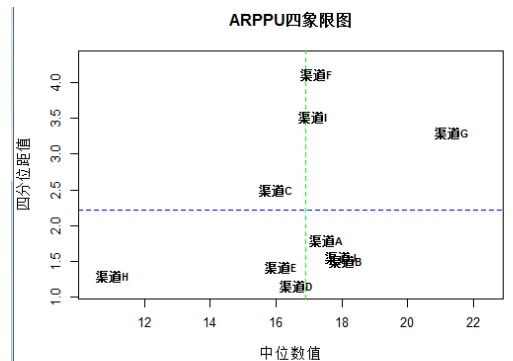
```
data=channel,FUN = mystats)
```

```
result
```

业务案例：利用四象限法研究渠道用户

• 对结果进行四象限法可视化展示：

```
# 对各指标绘制四象限图，查看渠道分布情况
# 将需要绘制四象限图的指标赋予name对象
name <- c("ARPPU","ARPU","新增次日留存率","新增7日留存率")
# 利用for循环绘制四个四象限分布图
par(mfrow = c(2,2))
for(i in 1:4){
  plot(result[,2*i],result[,2*i+1],type = "n",
        xlim=c(0.95*min(result[,2*i]),1.05*max(result[,2*i])),
        ylim=c(0.95*min(result[,2*i+1]),1.05*max(result[,2*i+1])), main =
        paste0(name[i],"四象限图"),
        xlab = "中位数值",ylab = "四分位距值") #绘制散点图
  abline(v = mean(result[,2*i]),lty = 2,col = "green") # 添加垂直直线
  abline(h = mean(result[,2*i+1]),lty = 2,col = "blue") # 添加水平直线
  text(result[,2*i],result[,2*i+1],result[,1],
        col = "black",cex = 0.8,font=2) # 在图中打印出渠道名称
}
par(mfrow=c(1,1))
```



渠道质量评估方法

- 目前渠道分析中通常都是将数量、质量、收入等维度单独研究，不能很好体现渠道的整体质量，也不容易进行不同渠道的横向对比。
- 此研究的评估方法是利用渠道在运营中的关键维度数据，构建通用的渠道质量评价指标，最后对渠道用户进行打分。



- 需要一定效度：能够准确代表渠道品质，反映该渠道在整体中的一个量级；
- 需要一定信度：在不同的阶段和时间内保持稳定；
- 需要一定普适性：采用同一算法，指标表现需要排除差异。

打分规则

- 1、选取某一作为起始周，假设全部渠道都处于同一起跑线，默认都是 10 分⁺
- 2、波动性得分 = $5 * (\text{本周实际值} - \text{上周实际值}) / \text{最近四周的最大值}$ ⁺
- 3、量级得分 = $5 * \text{渠道本周值} / \text{所有渠道本周总值}$ (只有指标 1、指标 2 有量级指标，其他三个指标不用考虑量级得分)⁺
- 4、各指标得分 = 上周得分 + 波动性得分 + 量级得分⁺
- 5、渠道用户综合得分 = $0.3 * \text{指标 1 得分} + 0.2 * \text{指标 2 得分} + 0.2 * \text{指标 3 得分} + 0.15 * \text{指标 4 得分} + 0.15 * \text{指标 5 得分}$ (权重默认值可以按照实际情况调整)⁺

自定义打分函数

```
# 自定义channel_score()实现指标打分模型
# 渠道得分函数
channel_score <- function(data,amount=T){

  # 进行指标的波动性打分
  library(reshape)
  data <- cast(data,渠道名称~自然周)
  # 利用apply函数分渠道求出当前周与上周的差值
  x <- t(apply(data[,-1],1,diff))
  # 利用as.data.frame函数将x转换成数据框形式
  x <- as.data.frame(x,row.names = as.character(data[,1]))
  # 利用colnames函数对x列名重新赋值
  colnames(x) <- colnames(data[3:ncol(data)])
  # 找出最近四周的最大值
  # 自定义函数mystat求最近四周的最大值
  mystat <- function(x){
    m <- rep(0,(ncol(data)-1))
    for(i in 1:(ncol(data)-1)){
      if(i <=3){
        m[i] <- max(x[1:i])
      } else {
        m[i] <- max(x[(i-4):i])
      }
    }
    return(m)
  }
  # 利用apply函数按分渠道求最近四周最大值
  y <- t(apply(data[,-1],1,mystat))
  # 利用as.data.frame函数将y转换成数据框形式
  y <- as.data.frame(y,row.names = as.character(data[,1]))
  # 利用colnames函数对y列名重新赋值
  colnames(y) <- colnames(data[2:ncol(data)])
  # 计算波动变化得分
  reliability_score <- 5*round(x/y[,-1],3)
  reliability_score
  if(amount) {
```

```
    if(amount) {
      # 进行指标的量级打分
      # 利用colSums函数进行按列求和
      x <- colSums(data[,-1])
      x <- as.data.frame(matrix(rep(x,nrow(data)),nrow = nrow(data),byrow = T))
      amount_score <- 5*round(data[,-1]/x,3)
      rownames(amount_score) <- rownames(reliability_score)
      amount_score

      # 计算指标得分
      # 利用cumsum函数进行对波动变化值进行累积求和
      rs_cumsum <- apply(reliability_score,1,cumsum)
      rs_cumsum <- as.data.frame(t(rs_cumsum))
      score <- 10+rs_cumsum+amount_score[,-1] #起始分+波动变化得分+量级得分
      score[,colnames(data)[2]] <- 10
      score <- score[,c(ncol(score),1:ncol(score)-1)] # 改变列的顺序
    } else {
      # 利用cumsum函数进行对波动变化值进行累积求和
      rs_cumsum <- apply(reliability_score,1,cumsum)
      rs_cumsum <- as.data.frame(t(rs_cumsum))
      score <- 10+rs_cumsum #起始分+波动变化得分
      score[,colnames(data)[2]] <- 10
      score <- score[,c(ncol(score),1:ncol(score)-1)] # 改变列的顺序
    }
  }
  return(score)
}
```

渠道各周得分走势

