



互聯網金融、電子商務
APP開發、大數據分析
以雲服務...
...等

用互連網、
机器学习、
深度学习、
卷积神经网

路径分析的主要应用场景

通过路径分析，可以有效发现用户典型的频繁路径模式（主流的路径、典型的路径），这些发现是对游戏版本更新迭代最基础、最可靠的信息参考之一。

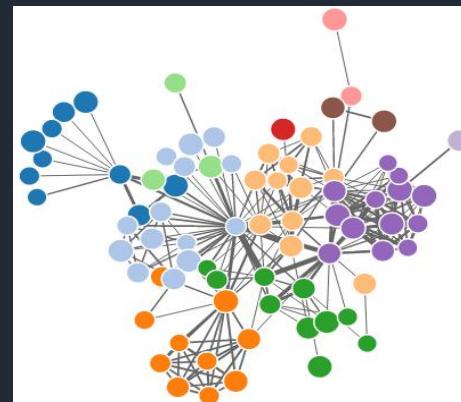
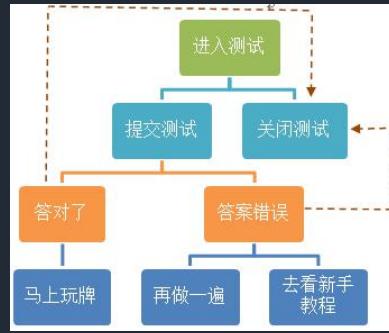


路径本身指的是玩家在游戏中的轨迹，轨迹背后反映的是玩家的行为特征，而这些行为特征对于游戏的运营、管理、客户服务等诸多方面都有重要的参考价值，能够起到借鉴作用。

路径分析对于游戏产品设计和优化的价值是显而易见的，并且是设计和优化的基础和核心来源。游戏数值策划正是通过对用户行为轨迹的观测和分析，来真正从源头上了解和把握玩家在游戏中的实际情况。

路径分析作为更为广泛、更加全面、更加基础的漏斗模型，可以通过对关键环节的转换率来对产品或运营指标进行监控。

路径分析的主要算法



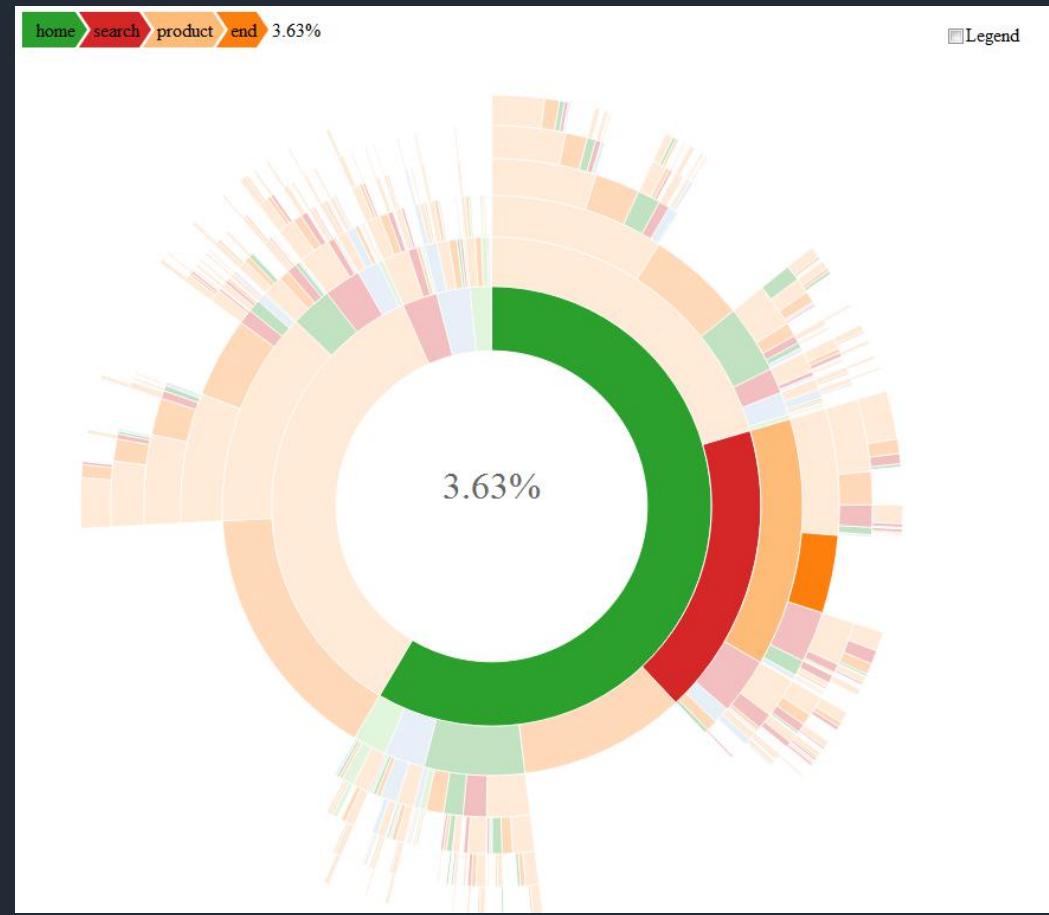


sunburst事件路径图

- 在R中，我们可以利用sunburstR包中的sunburst函数实现sunburst事件路径图
- 可以通过devtools::install_github("timelyportfolio/sunburstR")命令完成安装

```
if(!require(sunburstR))
devtools::install_github("timelyportfolio/sunburstR")
# 导入sequences数据
sequences <- read.csv(
  system.file("examples/visit-
  sequences.csv", package="sunburstR")
,header = FALSE
,stringsAsFactors = FALSE)
head(sequences)
```

V1	V2	Count
account	account	22781
account	account	3311
account	account	906
account	account	1156
account	account	5





基于序列的关联规则

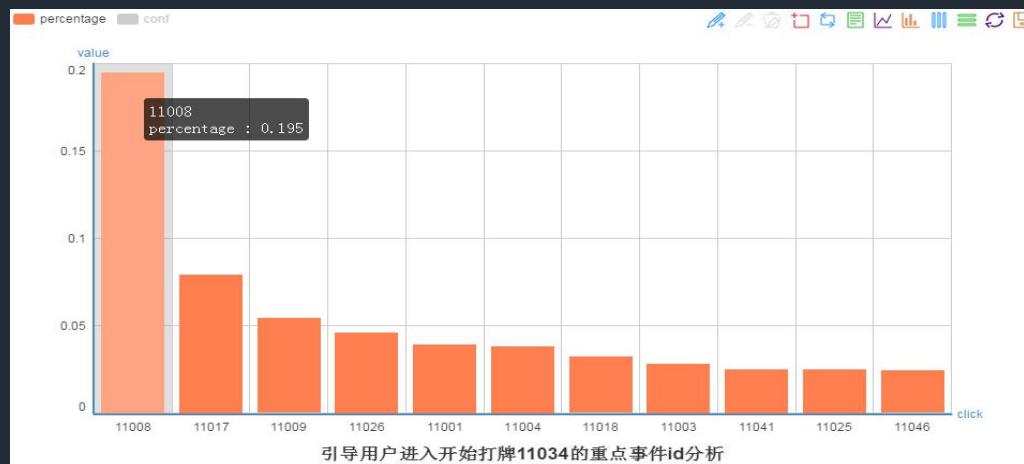
原始
数据

- 基于序列的关联规则常用的算法为由zaki提出的可以使用各种约束限制的cSPADE算法
- 在R中，可以使用arulesSequences包中的核心函数cspade实现。

用户id	时间event	事件id
1	1	11008
1	2	11001
1	3	11034
1	4	11002
2	1	11017
.....
5000	1	11158
5000	2	10034
5000	3	11004



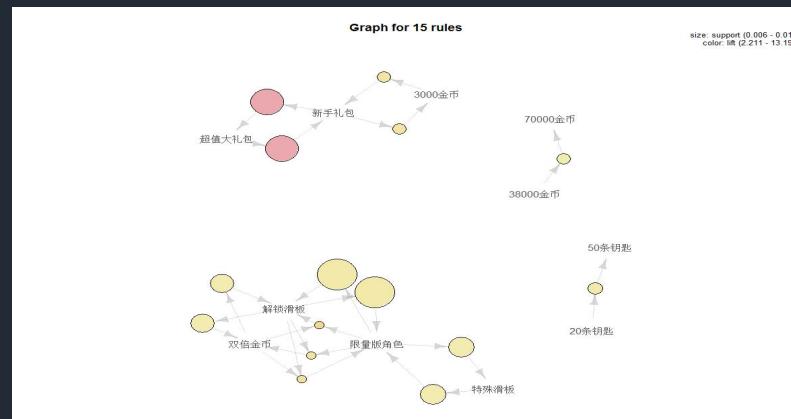
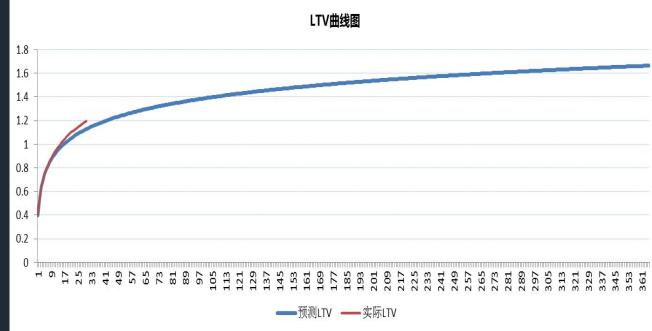
items	transactionID	sequenceID	event
{click=11008}	1		1
{click=11001}	2		2
{click=11034}	3		3
{click=11002}	4		4
{click=11017}	5		1
{click=11004}	6		2
{click=11018}	7		3
{click=11035}	8		4
{click=11036}	9		5
{click=11008}	10		6

转换
数据

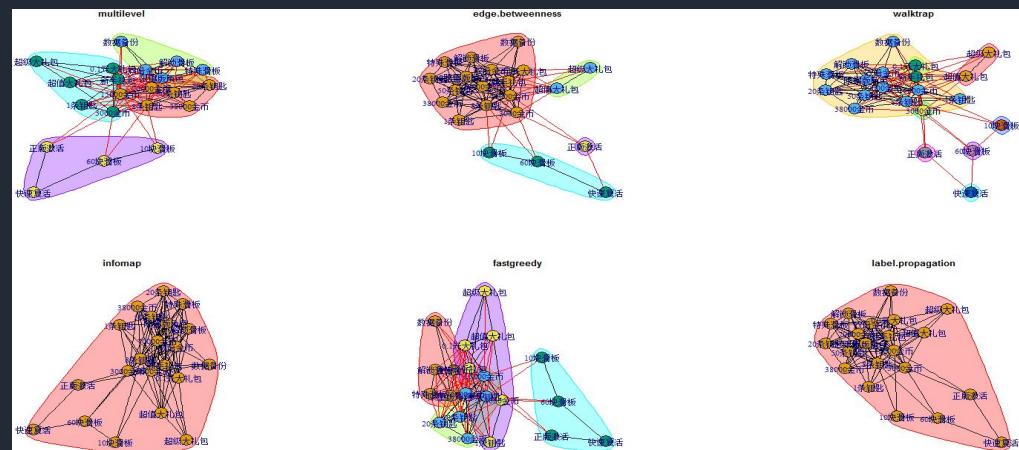
11008是为按钮11034的点击贡献最大的引流按钮，*support*占比为19.5%，接近全部引流按钮的五分之一



付费玩家常用分析方法



```
> as(recom3,"list")[1:5] #查看前五个玩家的 top3 推荐  
$`107204535`  
[1] "超值大礼包" "3000 金币" "超级大礼包"  
$`213666611`  
[1] "15000 金币" "20 条钥匙"  
$`226500629`
```



Apriori算法的三个度

- 假设共有1000人购买了物品（事务集W），其中购买了5000金币的有100人，购买了8000金币的有150人，同时购买了5000金币和8000金币的有80人。

1. 支持度是交易集同时包含X和Y的交易数与事务集W之比。

$$support(X \Rightarrow Y) = count(X \cap Y) / W = 80 / 1000 = 8\%$$

2. 可信度是指包含X和Y的交易数与包含X的交易数之比。

$$confidence(X \Rightarrow Y) = support(X \Rightarrow Y) / support(X) = 80 / 100 = 80\%$$

也就是说有80%的用户在购买了5000金币之后还会购买8000金币。

3. 提升度就是在购买X物品的前提下购买Y物品的可能性与没有购买X物品的情况下购买Y物品的可能性之比。

$$lift(X \Rightarrow Y) = confidence(X \Rightarrow Y) / support(Y) = 80\% / 15\% = 5.34$$

关联规则R语言实现

- 在R语言中，可以用于关联分析的程序包主要包括arules和arulesViz包
- 利用arules程序包中的相关函数可以实现关联规则算法包括Apriori算法、Eclat算法和weclat算法。
- 而另一个包arulesViz包则可以实现关联规则的可视化。

算法	实现函数
apriori	apriori(data,parameter=NULL,appearance=NULL,control=NULL)
eclat	eclat(data,parameter=NULL,control=NULL)
weclat	weclat(data,parameter=NULL,control=NULL)



案例：对玩家物品购买关联分析

原始
数据

player_id	product_name	qty
107204535	感恩大礼包	1
107204535	新手礼包	1
213666611	8条钥匙	1
278620434	15000金币	1
278620434	70000金币	1
278620434	快速复活	2
...

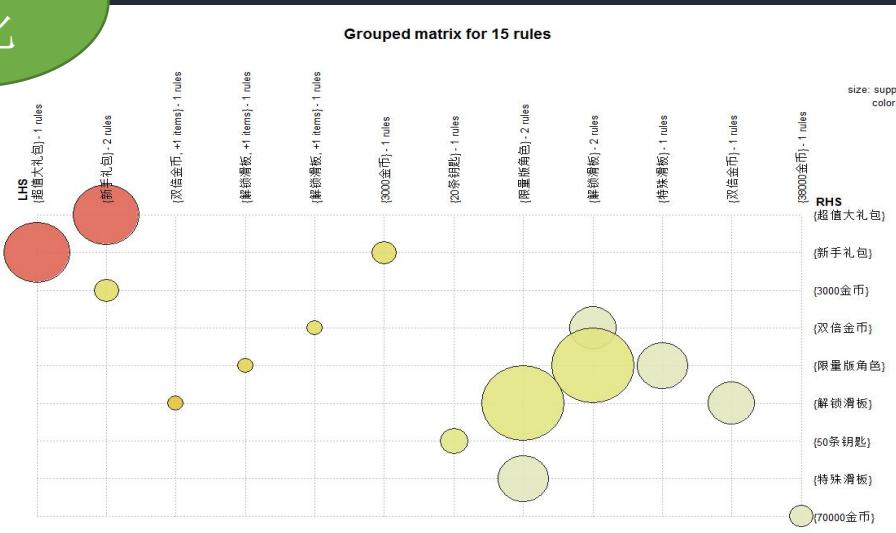


转换
数据

items	transactionID
1 {感恩大礼包,新手礼包}	107204535
2 {8条钥匙}	213666611
3 {0.1元大礼包,8条钥匙,限量版角色}	226500629
4 {38000金币,限量版角色,新手礼包}	230329140
5 {50条钥匙}	264162836
6 {15000金币,70000金币,快速复活}	278620434



规则可
视化



生成
规则

```
> # 对规则按照提升度排序，并输出提升度最大的前六条规则
> inspect(sort(rules, by="lift")[1:6])
  lhs          rhs      support   confidence      lift
  6 {超值大礼包} => {新手礼包} 0.015994882 0.9009009 13.1907084
  7 {新手礼包}   => {超值大礼包} 0.015994882 0.2341920 13.1907084
50 {双倍金币,限量版角色} => {解锁滑板} 0.005918106 0.3523810 5.7372024
48 {解锁滑板,双倍金币}   => {限量版角色} 0.005918106 0.4933333 4.2135524
49 {解锁滑板,限量版角色} => {双倍金币} 0.005918106 0.3083333 3.5241324
  8 {3000金币}      => {新手礼包} 0.007677543 0.2330097 3.4116554
```

基于玩家物品的智能推荐

- 关联规则只能反映一个事物与其他事物之间的关联性，有时候，我们想根据玩家的兴趣特点和购买行为，向玩家推荐他们感兴趣的信息和道具。
- 一个好的推荐系统能够为玩家提供个性化服务，增强用户黏性。
- 智能推荐的方法有很多，包括基于内容推荐、协同过滤推荐、基于规则推荐、基于效用推荐和基于知识推荐。各种推荐算法都有其优缺点。

推荐算法R语言实现

■ 在R语言中，常使用recommenderlab包中的函数构建和评估智能推荐模型。

■ Recommender函数：构建推荐模型

`Recommender(data, method, parameter=NULL)`

其中data为一个ratingMatrix，method的选项包括包括IBCF(基于物品的协同过滤推荐)、UBCF(基于用户的协同过滤推荐)、SVD(矩阵因子化)、PCA(主成分分析)、RANDOM(随机推荐)、POPULAR(基于流行度的推荐)

■ predict函数：预测推荐模型，得到模型的topN列表或者用户的预测评分。

`predict(object, newdata, n = 10, data=NULL, type="topNList", ...)`

其中object为recommender函数生成的推荐模型；newdata为待预测的数据；n为topN的值，默认为10，表示top10推荐；type的参数有"topNList"、"ratings"，当type="topNList"时，predict函数直接返回用户评分最高的前N个item，当type="ratings"时，predict函数预测用户对所有未评分的item打分，返回一个RatingMatrix对象。



案例：对玩家物品购买进行智能推荐

原始数据

player_id	product_name	qty
107204535	感恩大礼包	1
107204535	新手礼包	1
213666611	8条钥匙	1
278620434	15000金币	1
278620434	70000金币	1
278620434	快速复活	2
...



	0.1元大礼包	10块滑板	15元大礼包	1条钥匙	20条钥匙	3000金币	38000金币	50条钥匙	60块滑板	70000金币	8条钥匙	8元大礼包	超级大礼包	超值大礼包	解锁滑板	快速复活	首次复活0.1元	数据备份	双倍金币	礼包
107204535	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
213666611	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
226500629	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
230329140	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
264162836	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
278620434	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	
287632315	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
292033308	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
309973095	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
311844275	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

稀疏矩阵

推荐结果

```

> best <- Recommender(data_class,method="IBCF")
> # 使用predict函数，对玩家进行Top3推荐
> data.predict <- predict(model.best,newdata=data_class,n=5)
> recom3 <- bestN(data.predict,3)
> as(recom3,"list")[1:5] #查看前五个玩家的top3推荐
$`107204535`
[1] "超值大礼包" "3000金币" "超级大礼包"

$`213666611`
[1] "15000金币" "20条钥匙"

$`226500629`
[1] "解锁滑板" "双倍金币" "特殊滑板"

$`230329140`
[1] "超值大礼包" "解锁滑板" "双倍金币"

$`264162836`
[1] "70000金币" "20条钥匙" "15000金币"

```



	0.1元大礼包	10块滑板	15元大礼包	1条钥匙	20条钥匙	3000金币	38000金币	50条钥匙	60块滑板	70000金币	8条钥匙	8元大礼包	超级大礼包	超值大礼包	解锁滑板	快速复活	首次复活0.1元	数据备份	双倍金币	礼包
107204535	false	false	false	false	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	
213666611	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	
226500629	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	
230329140	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	
264162836	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	
278620434	false	false	true	false	false	false	false	false	false	false	true	false	false	false	false	true	false	false	false	
287632315	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	
292033308	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	
309973095	false	false	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	
311844275	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	

rating Matrix

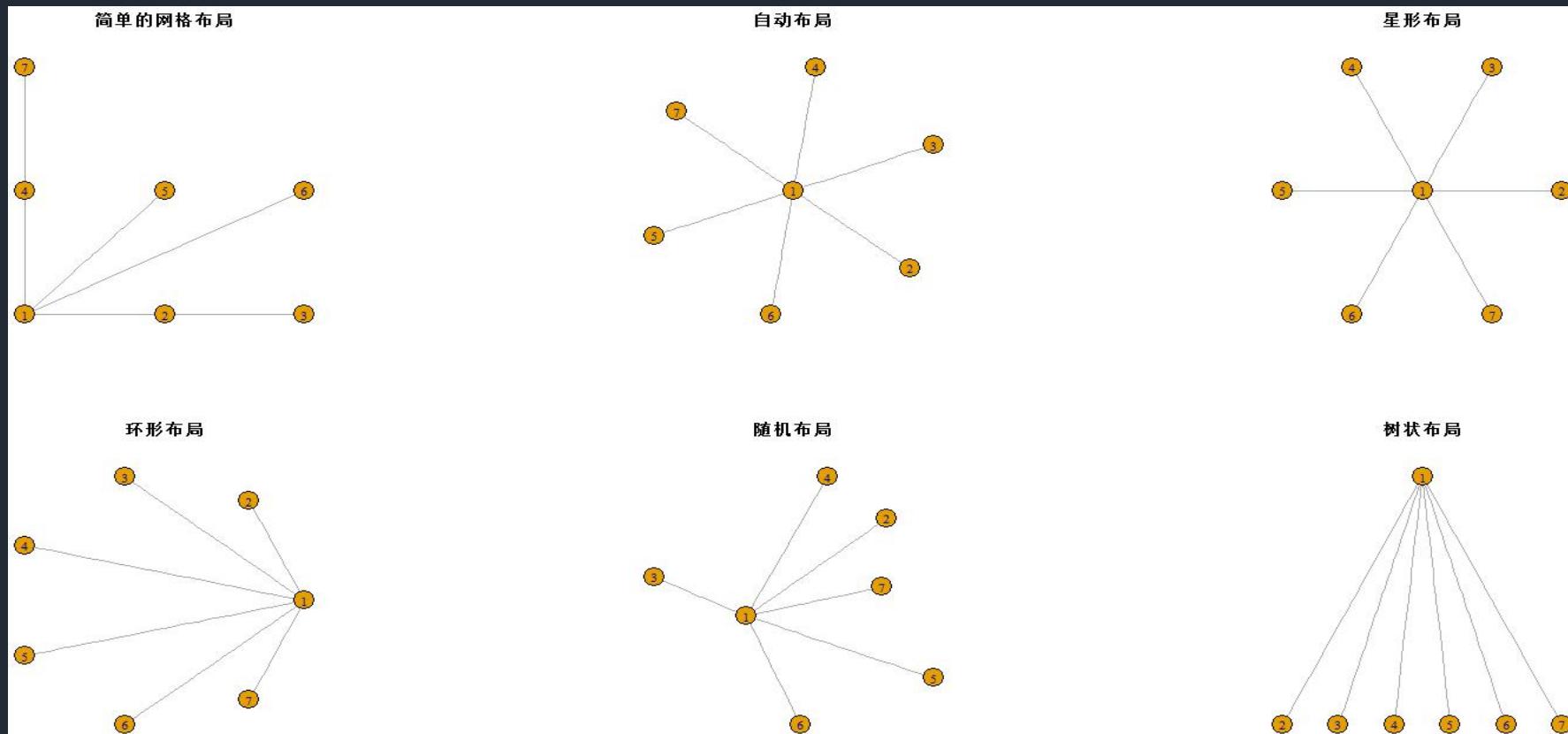
社会网络分析

- 社会网络分析(Social Network Analysis , SNA)是在传统的图与网络的理论之上对社会网络数据进行分析的方法。
- SNA的本质是利用各样本间的关系来分析整体样本的群落现象，并分析样本点在群落形成中的作用以及群落间的关系。
- 而在游戏里社会网络分析也有及其重要的作用，如游戏内社交系统分析、玩家购买道具的网络分析等。



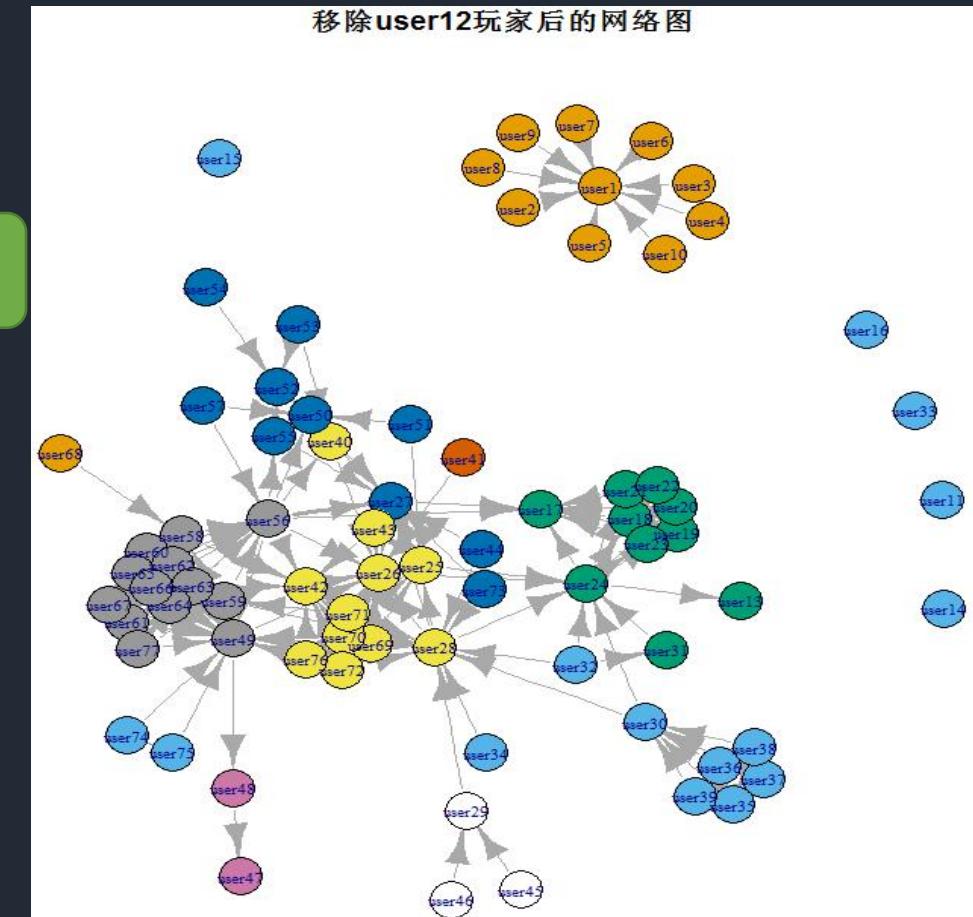
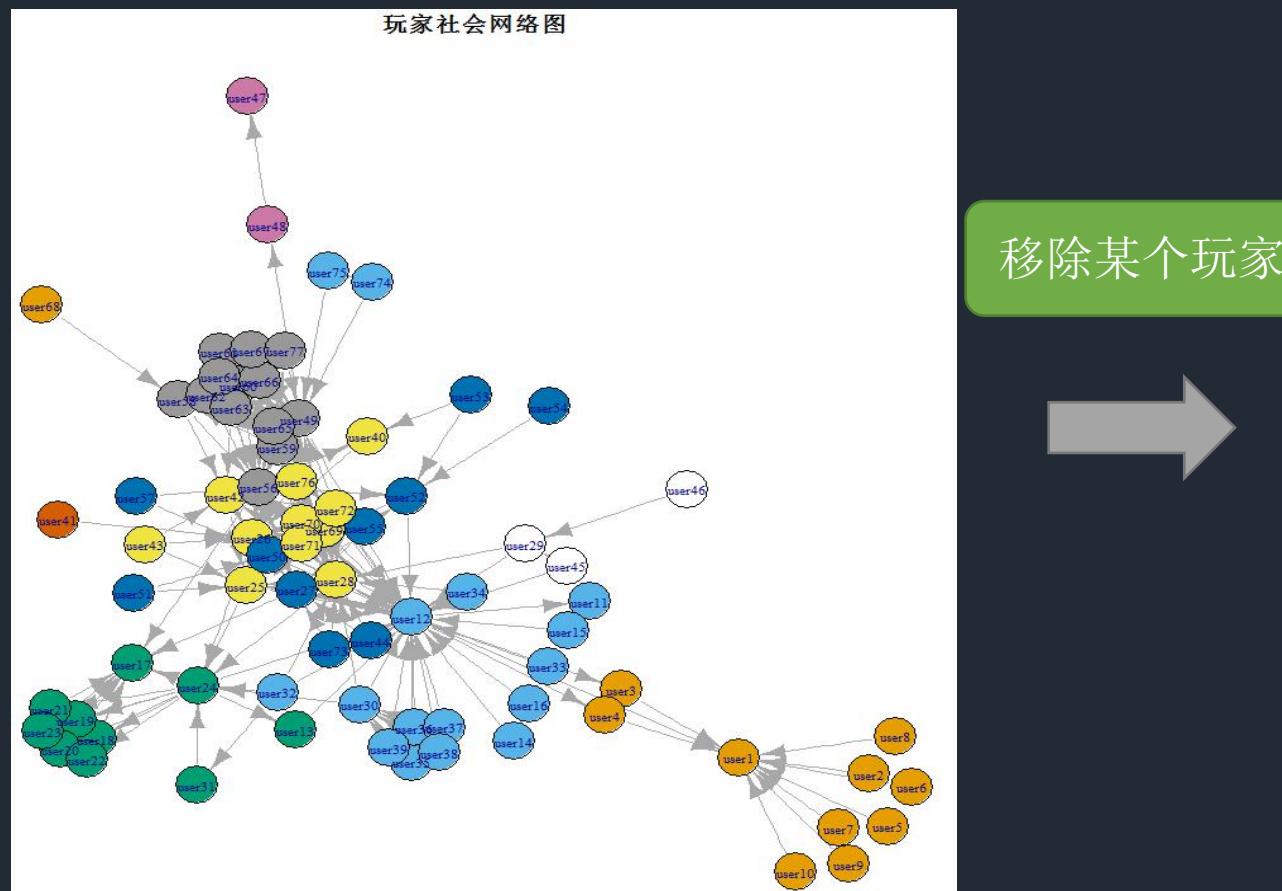
网络图的R实现

- 在R中，igraph包是用来处理网络图的。使用之前先通过`install.packages("igraph")`下载安装
- igraph包非常容易创建各种常规图，常用方法是先创建一个graph对象，然后利用plot函数绘制网络图，并通过设置layout参数来实现不同布局的效果



案例：玩家社交性分析

- 近几年手机端网游越来越重视游戏玩家社交性设计。某款游戏的玩法设计特别强调强社交性：玩家可以在游戏中组建家族，家族成员有不同的职务等级，玩家也可以在游戏中给好友赠送道具。



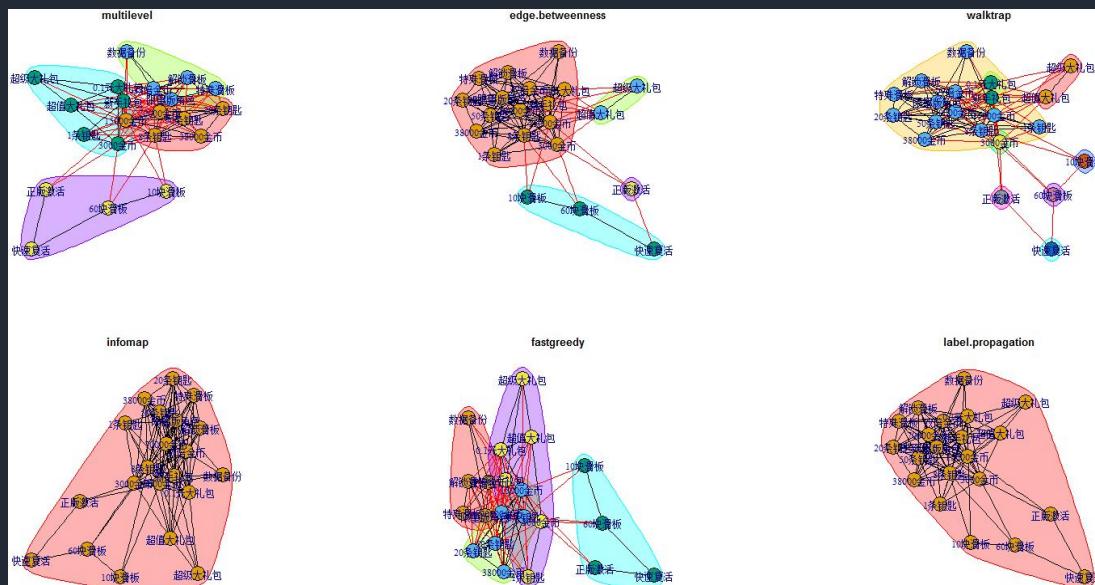
案例：玩家购买物品社群发现

- 社群结构特点：社群内边密度要高于社群间边密度，社群内部连接相对紧密，各个社群之间连接相对稀疏。此发现对于游戏内物品交易也有一定的适用性。

	感恩大礼包	新手礼包	0.1 元大礼包	其他礼包
1072044535	1	1	1	0

需要将其 1072044535 玩家购买数据变成以下格式：

感恩大礼包	新手礼包
感恩大礼包	0.1 元



infomap (*infomap*算法聚类) 和 *label.propagation* (标签传播算法) 不适合于份数据，通过另外四种算法发现的社群结构能帮助我们了解玩家购物喜好分类



乐逗游戏
IDREAMSKY GAMES

