兔展数据处理文档

张芳，董健，刘亮

0. 重要经验总结

0.0 2018 年 1 月 1 日。对节点进行编码有利于提高性能。（部分完成，需要转化）
0.1 2018 年 3 月 16 日。去除数据中自己查看/转发自己的记录，即去环

1. 原始数据集概况

1.1 数据记录格式举例

month78	PageID	IP	Province	SourceID	ViewID	EventType	EventTime	Remark
Tab 隔开(UTF-8)	55306108c51369f023cf59b2	118.181.56.196	甘肃	DGLeERoW	72V0khAu	view	2016-07-10 00:05:00	...

Remark 的内容有两种形式：

view：{"appSrc":"Weixin","isWeixin":true,"lc":8,"mdetail":"HUAWEI C8813DQ","model":"Huawei","netType":"WIFI","screen":"480x855","shareType":"timeline","useragent":"android"}

share：{"shareType":"timeline","staytm":7882,"useragent":"iphone"}

month7	PageID	IP	Province	SourceID	ViewID	EventType	EventTime	Remark

逗号隔开(GBK)	24cd93c8-0072-4504-b835-22ece1acf8ce	27.155.1.191	福建	ZfKIOFTZ	jCdW0XYY	view	2016-07-28 23:02:20	...

Remark 的内容有两种形式：

view:

"{""appSrc"":""Weixin"",""isWeixin"":true,""lc"":13,""model"":""iphone5x"",""netType"":""WIFI"",""screen"":""640x1136"",""shareType"":""groupmessage"",""useragent"":""iphone""}"

share:

"{""staytm"":25753,""useragent"":""android""}"

45days	PageID	SourceID	ViewID	IP	Province	EventTime	EventType	Remark
逗号隔开(UTF-8)	514f4870-bde4-4730-b896-0441f18bd532	TDdn32Fa	vsZ2Lkyn	14.17.37.43	广东	2016-01-14 00:00:00	switchpage	...

Remark 的内容有三种形式：

view：{"lc":3,"shareType":"singlemessage","useragent":"android"}

share：{"staytm":18,"useragent":"android"}

switchpage：{"ctp":9,"staytm":5635,"tgp":10,"ttp":26}

month34	PageID	IP	Province	SourceID	ViewID	EventType	EventTime	Remark
Tab 隔开(UTF-8)	55306108c51369f023cf59b2	125.223.253.27	吉林	uE6nu0lz	lfZRXDIP	view	2017-04-10 03:00:01	...

Remark 的内容有三种形式：

view：{"appSrc":"MicroMessenger","isWeixin":true,"lc":9,"mdetail":"","model":"iphone6x","netType":"WIFI","screen":"750x1334","shareType":"timeline","useragent":"iphone"}

share：{"shareType":"timeline","staytm":27573,"useragent":"android"}

1.2 数据基本情况

说明：D1 与 D2 的 Page 数量、ViewID 数量、ViewID+SourceID 数量都是对原始数据集进行的统计

D3 的 Page 数量、用户数量是对原始数据集进行的统计；大于 1 万的 Page 数量是对原始数据去除 switchpage 记录后进行的统计；ViewID 数量、ViewID+SourceID 数量是对原始数据去除 switchpage 记录并筛选出 Page 数量大于 1 万的数据后进行的统计

Datasets	Period	Days	Page 数量	ViewID 数量	ViewID+SourceID 数量	Cascade	Cascade 用户数量	大于 10000 的 Cascade

						数量		数量
D2	2016.07.01-07.30	30	2947	156911627	164609801			
D3	2017.03.01-04.30	61	2688	110542931	114703538			

Dataset	Period	Days	Cascade 数量	Cascade 用户数量	大于 100 的 Cascade 数量	大于 10000 的 Cascade 数量
		45	229021	7065126	16046	20
D1	2016.01.14-02.27	Page 数量	用户数量	大于 1 万的 Page 数量	ViewID 数量	ViewID+SourceID 数量
		353774	两亿多	2197	120270034	125310531

1.3 数据处理总体框架图

2. Cascade tree

接下来以 month7 数据为例简述 Cascade tree 生成过程，其他两组数据处理过程相同。

2.1 树生成过程

对于 month7 原始数据，包含字段（PageID,IP,Province,SourceID,ViewID,EventType,EventTime,..Useragent,lc,Model,Mdetail..）详情请参考 1.1

要求：1）找出完整传播的 Page：开始记录的 SourceID 为 None，且在采集窗口第一天和最后一天都不存在记录

2） 生成树且节点满足以下要求：

i) SourceID 在该条记录产生之前已经有其转发记录

ii) Share 记录前有其 view 记录（只有 share 没有 view 或者 view 时间在 share 之后的数据全部去除）

iii) 去重，每个用户只记录其第一次查看和转发记录

输出数据（MediatedData），包含字段（PageID SourceID,ViewID,EventTimeStr,ShareTimeStr,IP,ViewDepth,Pr,UserAgent,lc,Model,Mdetail）

①Map 阶段去除无效数据，将 EventTime 转成时间戳，将 Province 转成英文，输出所需字段

无效数据：

i) PageID 格式不正确，即长度不正确

ii) IP 格式不正确

iii) UserID 格式不正确，即包含字母、数字外的其它字符或长度在 5-8 之外（None 除外）

iv) 非移动端查看的记录，即 Useragent 为 pc

Key：PageID

Value：PageTime,IP,Pr,SourceID,ViewID,EventType,UserAgent,lc,Model,Mdetail

②Reduce 阶段找出完整传播的 Page，记录用户第一次查看与分享的时间，以及记录用户为第几级查看用户，最终生成树

Key：pageID

Value：SourceID,ViewID,EventTimeStr,ShareTimeStr,IP,ViewDepth,Pr,UserAgent,lc,Model,Mdetail

其中 EventTimeStr 为用户第一次浏览时间，ShareTimeStr 为该用户第一次转发时间，若用户没有转发，则 shareTimeStr 为 NOSHARE

2.2 拓扑分析

对于数据 MediatedData，包含字段（PageID　SourceID,ViewID,EventTimeStr,ShareTimeStr,IP,ViewDepth,Pr,UserAgent,lc,Model,Mdetail）详情请参考 2.1

要求：对 2.1 生成的树进行拓扑分析，输出数据（TreeAnalysis），包含字段（PageID　size,spread,depth,avgd,apl,dev,st,initProv,lifespan,prov,harePro）

①Map 阶段提取字段

Key：PageID

Value：SourceID,ViewID,ViewTime,Province,ShareTime

②Reduce 阶段对数据按照时间进行排序，使用 Java 的 tree 数据结构，将数据按照时间加入 tree 里，最后统计字段

Key：PageID

Value：size,spread,depth,avgd,apl,dev,st,initProv,lifespan,prov,harePro

字段说明

字段	*含义*	*说明*
size	节点规模	
spread	宽度	
depth	深度	
avgd	平均深度	（保留四位小数）
apl	平均最短路径	（保留四位小数）
dev	度方差	（保留四位小数）
st	初始时间	（单位：h）（除去 pc 端的第一条小时数）
initProv	初始省份	
lifespan	生存周期	（单位：s）　（除去 pc 端帖子的持续时间）
prov	涉及省份数量	（除去 Unknown）（暂不考虑城市）
sharePro	用户转发比	（转发用户数量占所有用户数量比）(保留四位小数）

2.3 时间延迟统计

基于 Cascade 的时间延迟统计

a) 对于数据 MediatedData，包含字段（PageID SourceID,ViewID,EventTimeStr,ShareTimeStr,IP,ViewDepth,Pr,UserAgent,lc,Model,Mdetail）详情请参考 2.1

要求：统计浏览延迟与转发延迟，输出数据（TimeDelay（ViewDelay/ShareDelay）），包含字段（PageID ViewDelays/ShareDelays）

ViewDelays 与 ShareDelays 代表每个相同 PageID 包含的所有用户的浏览延迟序列（单位：s）/转发延迟序列（单位：s）

浏览延迟 = 用户浏览时间 – 被浏览用户的转发时间（s）

转发延迟 = 用户转发时间 – 用户浏览时间（s）

①Map 阶段提取字段

key： PageID

value：SourceID,ViewID,ViewTime,ShareTime

②Reduce 阶段计算浏览延迟/转发延迟

Key：PageID

Value：ViewDelays/ShareDelays

b) 对于数据 MediatedData，包含字段（PageID SourceID,ViewID,EventTimeStr,ShareTimeStr,IP,ViewDepth,Pr,UserAgent,lc,Model,Mdetail）详情请参考 2.1

以及数据 FirstIp，包含字段（ViewID,IP）详情请参考 3.3 a)

i) 以 sourceId 为主键连接 MediatedData 与 FirstIp 并将 Ip 映射成 City，进而映射成 City 编码

输出数据（JoinCityandTime），字段包含（Page,SourceID,ViewID,SourceCode,ViewCode,viewTime,ShareTime）

ii) 计算 ShareDelay 和 ViewDelay，输出数据（CityNetandDelayTime（ShareDelay/ViewDelay））

包含字段（SourceCode,ViewCode,SourceShareWeek(0-6),SourceShareHour (0-23),ViewWeek,ViewHour,ShareDelay/viewDelay(s)）

3. Diffusion Network

接下来以 month7 数据为例简述数据处理过程，其他两组数据处理过程相同。

3.1 数据清洗

对于 month7 原始数据，包含字段（PageID,IP,Province,SourceID,ViewID,EventType,EventTime,..Useragent..）详情请参考 1.1

a） 转码（若数据集编码不为 UTF-8 则需要转码）

b） 去除无效数据

①PageID 格式不正确，即长度不正确

②非移动端查看的记录，即 Useragent 为 pc

③IP 格式不正确

④UserID 格式不正确，即包含字母、数字外的其它字符或长度在 5-8 之外（比如 None）

⑤EventTime 格式不正确，即长度不正确

⑥SourceID 与 ViewID 相同，即去环

c） 根据 eventType（view/share）将数据分成两组数据（DataCleanToViewRecord/DataCleanToShareRecord），包含字段（PageID,SourceID,ViewID,ViewIP,EventTime）

3.2 数据质量检测

对于数据 DataCleanToViewRecord，包含字段（PageID,SourceID,ViewID,ViewIP,EventTime）详情请参考 3.1

a) 统计 ViewID 的个数（不重复）

b) 统计 ViewID 与 ViewIP 的对应情况

① Map 阶段将数据中的 ViewID 与 ViewIP 分离出来

Key： ViewID

Value：ViewIP

② Reduce 阶段定义计数器，分别统计对应一个、两个、三个、多个城市的 ViewID 的个数

c) 统计 ViewID 与 ViewCity 的对应情况

①Map 阶段将数据中的 ViewID 与 ViewIP 分离出来，并且将 ViewIP 根据 IP 库转换成对应的城市

Key： ViewID

Value：ViewCity

②Reduce 阶段定义计数器，分别统计对应一个、两个、三个、多个城市的 ViewID 个数

统计结果：

用户 ID 对应的 IP 数量		1	2	3	3 个以上	All	一一对应百分比
用户数量	D1	83620919	11585220	2331654	1006664	98544457	84.86%
	D2	117764309	18997996	5001599	2607447	144371351	81.57%
	D3	56189132	6653296	1584060	979360	65405848	85.91%

用户 ID 对应的城市数量		1	2	3	3 个以上	All	一一对应百分比
用户数量	D1	90960222	7156138	408508	19589	98544457	92.30%
	D2	135869864	7974775	460705	66007	144371351	94.11%
	D3	62708904	2570213	116406	10325	65405848	95.88%

3.3 构建用户网络

要求：构建用户网络传播关系，包含字段（PageID,SourceID,ViewID,SourceCity,ViewCity,EventTime），若 SourceID 在整个数据集中都找不到其对应的 IP 则去除该条数据。

a) 对于原始数据集，找到每个 ViewID 对应的最早（ViewTime 最早）出现的浏览记录中的 IP，输出数据（FirstIp），包含字段（ViewID,IP）

①Map 阶段将数据中的 ViewID、IP、EventTime 分离出来

Key： ViewID

Value：IP,EventTime

②Reduce 阶段找到 EventTime 最小对应的 IP

Key： ViewID,IP

b) 将 DataCleanToViewRecord/DataCleanToShareRecord 数据（见 3.1）分别与 FirstIp 进行连接，即找到 SourceID 对应的第一次浏览记录中的 IP（SourceIP），并根据 IP 库将 IP 转成对应的城市，得到用户网络（UserNetOfView/UserNetOfShare），包含字段（PageID,SourceID,ViewID,SourceCity,ViewCity,EventTime）

①Map 阶段根据输入的文件分片信息（文件名）分别输出 key/value

Key1：SourceID

Value1：IP

Key2：SourceID

Value2：PageID,SourceID,ViewID,ViewIP,EventTime

②Reduce 阶段找到 SourceID 对应的 SourceIP，并根据 IP 库将 IP 转换为城市（注：香港与澳门在 IP 库里面的格式不一样）

Key：PageID,SourceID,ViewID,SourceCity,ViewCity,EventTime

c) 在 b)的结果基础上统计输出简化的用户网络（SmallUserNetOfView/SmallUserNetOfShare）包含字段（SourceID,ViewID,Weight（PageID 个数））

①Map 阶段将数据中的 SourceID、ViewID、PageID 分离出来

Key：SourceID ,ViewID

Value：PageID

②Reduce 阶段统计 PageID 的个数

Key：SourceID,ViewID,Weight

3.4 构建城市网络

前提：对 IP 库里的 382 个城市进行编码，要求保证长度一致且不重复，文件格式：省份名称，省份名英文，城市名，城市编码（cityInChina.csv）

用四位数对城市进行编码，前两位代表省份，后两位代表该省的某个城市

eg：安徽,Anhui,六安,1208

安徽,Anhui,马鞍山,1213

a) 对于数据 UserNetOfView/UserNetOfShare，包含字段（PageID,SourceID,ViewID,SourceCity,ViewCity,EventTime）详情请参考 3.3

根据上述的城市编码，将 SourceCity,ViewCity 都转成城市编码 SourceCode,ViewCode

输出数据（CityCodeUserNetOfView/ CityCodeUserNetOfShare），包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime）

b) 统计每个城市的用户数量：对于数据 CityCodeUserNetOfView，包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime），统计每个 ViewCode 的 ViewID 数量，输出数据（CounterCitySize），包含字段（ViewCode,ViewIDNum）

①Map 阶段将数据中的 ViewCode、ViewID 分离出来

Key：ViewCode

Value：ViewID

②Reduce 阶段统计 ViewID 的个数

Key：ViewCode,ViewIDNum

c) 构建城市内的用户关系网络：对于数据 CityCodeUserNetOfView，包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime）统计当 SourceCode 与 ViewCode 相同时所有的 SourceID,ViewID 并统计权重（即相同 SourceID,ViewID 的个数），要求 N 个城市输出 N 个文件，文件名为 Net_城市编码，输出数据（UserNetInSameCity），包含字段（SourceID,ViewID,Weigh）

①Map 阶段将数据中的 SourceID、ViewID、SourceCode、ViewCode 分离出来

Key：SourceCode,ViewCode

Value：SourceID,ViewID

②Reduce 阶段先判断 SourceCode 与 ViewCode 是否相同，若相同将 SourceID,ViewID 统计权重后输出到对应的文件（Net_ViewCode）

Key：SourceID,ViewID,Weight

d) 城市内用户查看/被查看行为分布：对于数据 CityCodeUserNetOfView，包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime）

统计 ViewID/SourceID 为某个城市用户查看/被查看行为分布。要求 N 个城市输出 N 个文件，文件名为 View_城市编码

输出数据（UserViewActionDistribution/ UserBeViewedActionDistribution）包含字段为（ViewID/SourceID,DegInCity,DegOutCity）

其中 DegInCity 表示某个用户在该城市内用户网络的度，即查看/被查看了本城市内几次，如果该用户没有出现在城市内用户网络中，则度为 0。DegOutCity 值为该用户查看/被查看其他城市的次数。不记录 DegInCity、DegOutCity 都为 0 的情况。

①Map 阶段将数据中的 SourceID/ViewID、SourceCode、ViewCode 分离出来

Key：ViewCode(SourceCode)

Value：ViewID(SourceID),SourceCode(ViewCode)

②Reduce 阶段统计每个 ViewID/SourceID 中，ViewCode 与 SourceCode 相同的个数以及不同的个数

Key：ViewID(SourceID),DegInCity,DegOutCity

e) 构建城市关系网络：

1）对于数据 CityCodeUserNetOfView/ CityCodeUserNetOfShare，包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime）统计每个 SourceCode,ViewCode 包含的 ViewID 的个数，输出数据（CounterCityNetOfViwe/CounterCityNetOfShare），包含字段（SourceCode,ViewCode,ViewIDNum）

①Map 阶段将数据中的 SourceCode、ViewCode、ViewID 分离出来

Key：SourceCode,ViewCode

Value：ViewID

②Reduce 阶段统计 ViewID 的个数

Key：SourceCode,ViewCode,ViewIDNum

2） 对于数据 CounterCityNetOfViwe/CounterCityNetOfShare，包含字段（SourceCode,ViewCode,ViewIDNum）

将两个文件连接，输出数据（JoinViewAndShareCityNet），包含字段（SourceCode,ViewCode,ViewIDNum,ShareIDNum）

①Map 阶段根据输入的文件分片信息（文件名）分别输出 key/value

Key1：SourceCode,ViewCode

Value1：”view”,ViewIDNum

Key2：SourceCode,ViewCode

Value2：”share”,ShareIDNum

②Reduce 阶段将 ViewNum 与 ShareNum 连接在一起

Key：SourceCode,ViewCode,ViewNum,ShareNum

3.5城市网络及时间延迟

a) 对于数据 CityCodeUserNetOfView/ CityCodeUserNetOfShare，包含字段（PageID,SourceID,ViewID,SourceCode,ViewCode,EventTime）详情参考 3.4 a)

要求同一个 PageID 同一个用户只记录一次行为，根据时间找最早查看或者转发时间的记录，输出数据（FindFirstViewRecord/FindFirstShareRecord）包含字段（PageID,SourceID,ViewID,EventTime,hour,week,”view”/”share”）

①Map 阶段将数据中的 PageID、SourceID、ViewID、EventTime 分离出来

Key：PageID,SourceID,ViewID

Value：EventTime

②Reduce 阶段找出最早时间下对应的记录，并解析出时间对应的小时（0-23），星期（0-6）

Key：PageID,SourceID,ViewID,EventTime,hour,week,”view”/”share”

注：view 与 share 字段是用来区分是转发记录还是查看记录，方便后面的工作

b) 求 ShareDelay：对于数据 FindFirstViewRecord/FindFirstShareRecord，包含字段（PageID,SourceID,ViewID,EventTime,hour,week,”view”/”share”）

转发延迟是用户第一次查看网页到最近一次转发网页的时间延迟

输出数据（ShareDelay）包含字段（SourceID,ViewID,ViewHour,ViewWeek,ShareHour,ShareWeek,shareDelay）

①Map 阶段读取这两个文件

Key：PageID,SourceID,ViewID

Value：PageID,SourceID,ViewID,EventTime,hour,week,”view”/”share”

②Reduce 阶段根据 Value 最后的字段是 view 还是 share 找出转发记录与查看记录，都存在的情况下，计算 ShareDelay（>0），将查看记录输出

Key：SourceID,ViewID,ViewHour,ViewWeek,ShareHour,ShareWeek,shareDelay

C) 求 ViewDelay：对于数据 FindFirstViewRecord/FindFirstShareRecord，包含字段（PageID,SourceID,ViewID,EventTime,hour,week,"view"/"share"）

查看延迟是用户第一次分享网页到最近一次网页被查看的时间延迟

输出数据（ViewDelay）包含字段（SourceID,ViewID,ShareHour,ShareWeek,ViewHour,ViewWeek,ViewDelay）

①Map 阶段将读取这两个文件，根据文件的分片信息（文件名）不同，分别输出 Key/value

若是查看记录：

Key：PageID,SourceID

Value：PageID,SourceID,ViewID,EventTime,hour,week,"view"

若是转发记录：

Key：PageID,ViewID

Value：PageID,SourceID,ViewID,EventTime,hour,week,"share"

②Reduce 阶段根据 Value 最后的字段是 view 还是 share 找出转发记录与查看记录，可能存在 share 记录有多个的情况，则只记录一次 share 记录，将 view 记录保存到集合中，都不为空的情况下，逐一计算 viewDelay（>0）将查看记录输出

Key：SourceID,ViewID,ShareHour,ShareWeek,ViewHour,ViewWeek,ViewDelay

4 Retwitting Prediction & Cascade Prediction （转发预测和级联预测）

总体思路：将 page 分为两类，有始有终一类，其余一类。

1. 统计每个用户参与 page 的数量（参与度 or 活跃度，active），得到一个文件。
2. 特征提取。

   针对每个 page 每个用户，提取以下信息，每个 page 存一个文件。

   userID（用户 ID）, prov（第一次出现的省份）, lc(最早出现层级)，viewNum(转发前查看总次数，相同源只算一次，如果没有转发则计算总次数)，viewTime(第一次查看时间)，isShare(是否转发)，shareTime(最早的转发时间)，active（该用户的参与度），preActive（父节点的参与度；如果父节点为 None，计为 0；如果有多个父节点，记为总和）。

5 浏览量的时序特征提取

针对有始有终的 page，统计每小时的浏览量（不同于 cascade，每个用户只计算一次）。最后得到一个矩阵文件（m*n,m 为数据持续时间的小时分布，n 为 page。没有浏览的记录为 0）。