

实验三：电子商务网站用户行为建模和协同过滤算法

3170102492 夏豪诚 & 3170101249 姜鹤群

Part1 电子商务网站用户行为建模

Question 1

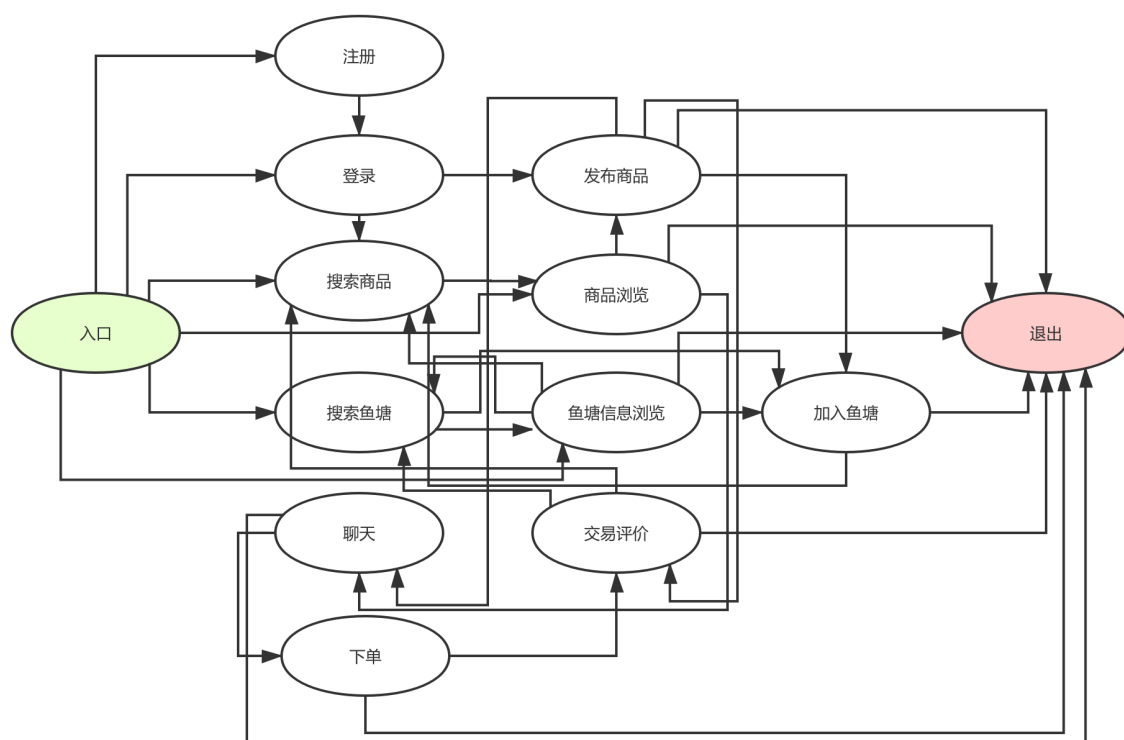
1.1.1 基本核心业务描述

闲鱼电子商务公司的**核心业务**实际上存在多个，包括普通二手商品交易、拍卖、租赁、鉴定担保、回收等等，以下核心业务的状态主要就普通二手商品交易而言（Attention: 表中的状态不包括入口和退出）：

核心业务状态	说明
注册	注册闲鱼账号
登录	登录闲鱼账号（可使用淘宝或支付宝账号）
搜索商品	利用关键词对平台中已发布的二手商品进行搜索
商品浏览	包括在商品推荐页面和其他列表视图以及商品详情页的浏览
鱼塘信息浏览	包括在鱼塘推荐页面和鱼塘详情页的浏览
搜索鱼塘	根据地区和兴趣等信息对鱼塘进行检索
加入鱼塘	选择鱼塘加入
发布商品	用户填写商品信息和商品价格发布二手商品
聊天	用户与用户间在进行交易前后通过聊天进行信息交流
下单	用户填写收货信息后拍下商品并进行支付操作
交易评价	在交易完成后，卖家和买家之间进行互相评价

1.1.2 用户行为模型图

CBMG分析其用户行为模型得到用户行为模型图如下（Attention: 由于没有足够的会话用来统计不同用户在不同会话中在不同状态之间的跳转概率，下方用户行为模型图为静态CBMG）：



Question 2

1.2.0 简述解题思路

1. 首先我们需要温习一下客户行为模型图(Customer Behavior Model Graph)的定义：

客户行为模型图是一种基于转换概率矩阵的图形。它主要用于描绘Web用户会话，捕获用户在某一个站点的浏览模式，也能有效捕获客户如何从一个状态转换到另一个状态，CBMG主要由状态和转换组成。在CBMG中，一个状态使用一个带有状态名和数字的椭圆（为了美观在本题中使用了圆角矩形），一个转化可以用概率（转化率）作标签的箭头符号表示。

根据此定义和给出的客户行为模型的转换概率矩阵 P ，进行客户行为模型图的绘制。

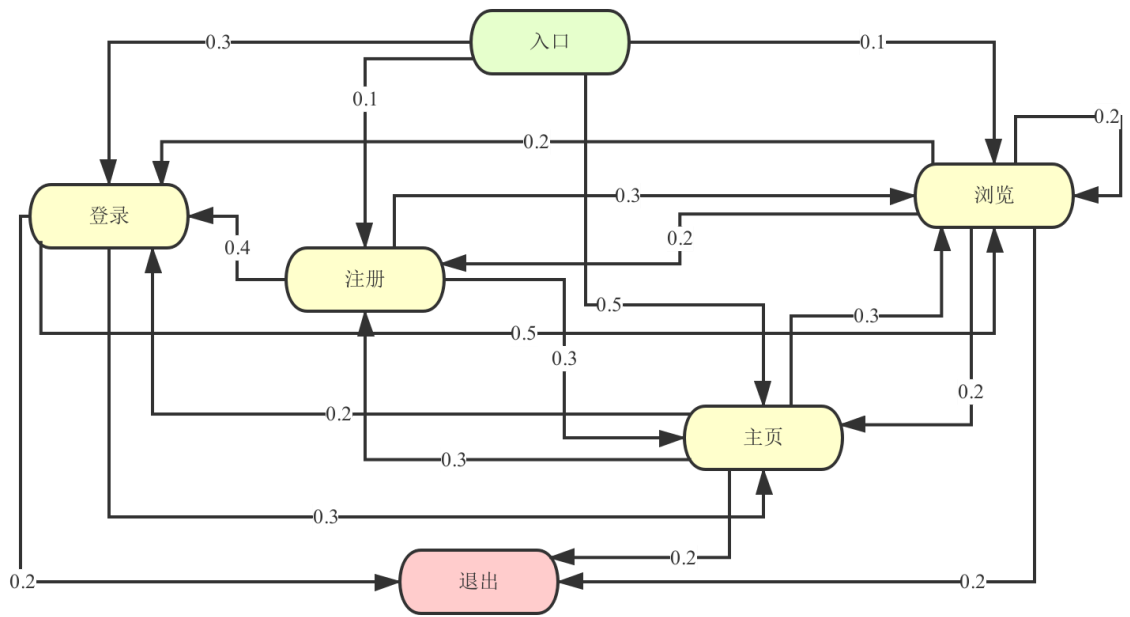
2. 至于CBMG 各个状态的平均访问次数，可以根据公式和定义来进行计算：

定义：设 V 表示CBMG中各个事务被访问到的平均次数， v_i 表示事务 i 在一次会话中被访问到的次数，如果假设 v_0 的次数是 1，即会话开始的次数为 1，那么各个服务被访问的次数可定义为公式(1)的形式，即各服务被访问的次数等于其前驱服务被访问次数与访问该服务概率的乘积。

公式： $v_j = \sum_{i=0}^{n+1} v_i \times p_{i,j}, j = 1, \dots, n+1$

3. 至于计算平均会话长度，因为根据定义平均会话长度是各个状态平均访问次数之和，所以可以利用第二小题中计算得到的数据来进行计算。（Attention: 平均会话长度不包括入口和退出的状态!）

1.2.1 客户行为模型图



1.2.2 平均访问次数

CBMG 各个状态的平均访问次数:

状态	访问比率
入口	1.00
主页	1.30
登录	1.10
注册	0.60
浏览	1.40
退出	1.00

1.2.3 平均会话长度

平均会话长度 = 登录 + 注册 + 浏览 + 主页 = 4.4

Part2 协同过滤算法

2.1 实验结果截图

完成代码后的实验结果截图：

```
Run: ItemBasedModel x
/usr/java/jdk-13.0.1/bin/java -javaagent:/snap/intellij-idea-community/185/lib/idea_rt.jar=46269
0.0 0.9082951062292474 0.8196798155377502 0.11306675421666136 -6.798699777552591E-17
0.9082951062292474 0.0 0.9024377759125219 0.24066597223711836 0.1573133012414231
0.8196798155377502 0.9024377759125219 0.0 0.057475061233962864 0.0
0.11306675421666136 0.24066597223711836 0.057475061233962864 0.0 0.984731927834662
-6.798699777552591E-17 0.1573133012414231 0.0 0.984731927834662 0.0
Sally to sausage: predictedscore 3.0859976747743403
Sally to mutton: predictedscore 3.803426751343522
Tommy to pork: predictedscore 2.0
Tommy to chicken: predictedscore 2.0
warning: no such user named Curry in the dataset
0.0 0.5874799940107168 0.5152540399295447 0.2739544763999135 0.38018781261549794
0.5874799940107168 0.0 0.8770580193070293 0.06661733875264919 0.09245003270420493
0.5152540399295447 0.8770580193070293 0.0 0.0 0.0
0.2739544763999135 0.06661733875264919 0.0 0.0 0.720576692122892
0.38018781261549794 0.09245003270420493 0.0 0.720576692122892 0.0
Sally to sausage: predictedscore 3.36078789311421
Sally to mutton: predictedscore 4.0
Tommy to pork: predictedscore 2.0
Tommy to chicken: predictedscore 2.0
warning: no such user named Curry in the dataset

Process finished with exit code 0
```

2.2 实验结果分析

结果分析：

```
0.0 0.9082951062292474 0.8196798155377502 0.11306675421666136 -6.798699777552591E-17
0.9082951062292474 0.0 0.9024377759125219 0.24066597223711836 0.1573133012414231
0.8196798155377502 0.9024377759125219 0.0 0.057475061233962864 0.0
0.11306675421666136 0.24066597223711836 0.057475061233962864 0.0 0.984731927834662
-6.798699777552591E-17 0.1573133012414231 0.0 0.984731927834662 0.0
Sally to sausage: predictedscore 3.0859976747743403
Sally to mutton: predictedscore 3.803426751343522
Tommy to pork: predictedscore 2.0
Tommy to chicken: predictedscore 2.0
warning: no such user named Curry in the dataset
```

上图为运行样例程序所得结果截图。

其中，矩阵为相似度结果矩阵，通过该矩阵我们可以得出不同User对不同item的预测分。以及判断有无某User。

```

0.0 0.9082951062292474 0.8196798155377502 0.11306675421666136 -6.798699777552591E-17
0.9082951062292474 0.0 0.9024377759125219 0.24066597223711836 0.1573133012414231
0.8196798155377502 0.9024377759125219 0.0 0.057475061233962864 0.0
0.11306675421666136 0.24066597223711836 0.057475061233962864 0.0 0.984731927834662
-6.798699777552591E-17 0.1573133012414231 0.0 0.984731927834662 0.0
Sally to sausage: predictedscore 3.0859976747743403
Sally to mutton: predictedscore 3.803426751343522
Tommy to pork: predictedscore 2.0
Tommy to chicken: predictedscore 2.0
warning: no such user named Curry in the dataset
Jim to mutton: predictedscore 3.9677032487478128
Alice to sausage: predictedscore 3.6953325582581136
Alice to mutton: predictedscore 3.9344755837811736
Kevin to chicken: predictedscore 1.8622530025639272
George to pork: predictedscore 1.0
George to chicken: predictedscore 1.0
Joseph to pork: predictedscore 5.0
Joseph to chicken: predictedscore 4.999999999999999

```

上图为样例程序所有User的结果图，我们可以得出所有用户的预测分。

```

0.0 0.5874799940107168 0.5152540399295447 0.2739544763999135 0.38018781261549794
0.5874799940107168 0.0 0.8770580193070293 0.06661733875264919 0.09245003270420493
0.5152540399295447 0.8770580193070293 0.0 0.0 0.0
0.2739544763999135 0.06661733875264919 0.0 0.0 0.720576692122892
0.38018781261549794 0.09245003270420493 0.0 0.720576692122892 0.0
Sally to sausage: predictedscore 3.36078789311421
Sally to mutton: predictedscore 4.0
Tommy to pork: predictedscore 2.0
Tommy to chicken: predictedscore 2.0
warning: no such user named Curry in the dataset
Jim to mutton: predictedscore 4.0
Alice to sausage: predictedscore 3.786929297704736
Alice to mutton: predictedscore 4.0
Kevin to chicken: predictedscore 1.6038953344193216
George to pork: predictedscore 1.0
George to chicken: predictedscore 1.0
Joseph to pork: predictedscore 5.0
Joseph to chicken: predictedscore 5.0

```

上图为补全程序运行后，第二种方法所得的结果，与上上图对比可得预测分和预测的item很一致，而相似度结果矩阵也并未像同学反映一样出现大量负值，这里我的处理是计算 R_u 时将User-Item二维数组中为0项也计算在内，如果不计算在内会出现0和29的结果，这与原结果差距很大。