“天池竞赛-商家促销后的重复买家预测”
最终报告

项目分工

成员	分工
周泳宇（3120191079）	数据处理与分析，算法调研与实现，文档撰写
邵江逸（3120191038）	数据处理与分析，算法调研
周田（3220190931）	数据处理与分析，算法调研
林瀚熙（3120191020）	数据处理与分析，结果分析
李彤（3120191018）	数据处理与分析，算法实现

项目代码仓库地址

https://github.com/Carmelo1997/BIT_DM_PROJECT

1. 数据获取及预处理

1.1 数据来源

- **题目的背景，来源**
  本题目来自于天池 IJCAI-15 Contest。我们知道，国内外的电商有各种各样促销活动，像国外的黑五（Black Friday），国内的双十一等等。大量商家通过打折促销吸引顾客。其中，有许多顾客是被商家的促销所吸引的新顾客，那么他们会不会在促销之后继续在这家店买东西呢？本次比赛题目就是预测这些会重复购买的顾客。
- **数据和分析**
  该数据集包含过去 6 个月“双 11”日之前和当天匿名用户的购物日志，以及标签上是否重复购买的信息。由于隐私问题，数据以偏颇的方式进行采样，因此该数据集的统计结果将偏离天猫的实际数据，但它不会影响解决方案的适用性。该问题和数据集与推荐系统领域的 Sequential Recommendation（序列推荐）类似：主要框架为，利用不同的模型（CNN，attention 居多，逐渐取代 RNN）学习每个 user 的 item 序列信息作为其 short-term 特征，单独的 user embedding 视作其 long-term 偏好，两者分开学习，或同时学习，并作最终预测。

1.2 数据说明

该数据集包含过去 6 个月“双 11”日之前和当天匿名用户的购物日志，以及标签上是否重复购买的信息。由于隐私问题，数据以偏颇的方式进行采样，因此该数据集的统计结果将偏离天猫的实际数据，但它不会影响解决方案的适用性。

- 数据描述
  - User Behaviour Logs

Data Fields	Definition
user_id	A unique id for the shopper.
item_id	A unique id for the item.
cat_id	A unique id for the category that the item belongs to.
merchant_id	A unique id for the merchant.
brand_id	A unique id for the brand of the item.
time_tamp	Date the action took place (format: mmdd)
action_type	It is an enumerated type {0, 1, 2, 3}, where 0 is for click, 1 is for add-to-cart, 2 is for purchase and 3 is for add-to-favourite.

  - User Profile

Data Fields	Definition
user_id	A unique id for the shopper.
age_range	User’s age range: 1 for <18; 2 for [18,24]; 3 for [25,29]; 4 for [30,34]; 5 for [35,39]; 6 for [40,49]; 7 and 8 for >= 50;0 and NULL for unknown.
gender	User’ s gender: 0 for female, 1 for male, 2 and NULL for unknown.

  - Training and Testing Data

Data Fields	Definition
user_id	A unique id for the shopper.

Data Fields	Definition
merchant_id	A unique id for the merchant.
label	It is an enumerated type {0, 1}, where 1 means repeat buyer, 0 is for non-repeat buyer. This field is empty for test data.

In [2]:
```
data = pd.read_csv('./data_format2/train_format2.csv', index_col=0)
print('属性类别数:', len(data.columns))
print('总行数:', len(data))
print('示例数据:')
data.head(5)
```

```
属性类别数: 5
总行数: 7030723
示例数据:
```

Out[2]:

user_id	age_range	gender	merchant_id	label	activity_log
34176	6.0	0.0	944	-1	408895:1505:7370:1107:0
34176	6.0	0.0	412	-1	17235:1604:4396:0818:0#954723:1604:4396:0818:0...
34176	6.0	0.0	1945	-1	231901:662:2758:0818:0#231901:662:2758:0818:0#...
34176	6.0	0.0	4752	-1	174142:821:6938:1027:0
34176	6.0	0.0	643	-1	716371:1505:968:1024:3

其中age_range表示用户年龄范围：1表示<18岁；2表示[18,24]；3表示[25,29]；4表示[30,34]；5表示[35,39]；6表示[40,49]；>=50岁时为7和8；
gender表示用户性别：女性为0，男性为1，2和unknown为未知；
merchant_id表示商家的唯一ID标识；
label表示用户是否为重复购买者：'1'表示'user_id'是'merchant_id'的重复购买者，而'0'相反。'-1'表示'user_id'不是给定商家的新客户，"NULL"仅在测试数据中出现，表明它需要预测的；
activity_log表示{user_id，merchant_id}之间的交互记录，其中每个记录都是表示为"item_id：category_id：brand_id：time_stamp：action_type"，"#"用于分隔两个相邻元素，记录未按任何特定顺序排序。

1.3 数据预处理

我们提取了过去 7 个月的数据，以及 20 多个用户观察到的种类，形成最终的数据集。每天的用户行为被视为一个会话，而所有单例会话（即只包含一个物品）被剔除。同时，对于没有被超过 K 个用户共同点击过的物品，我们也将其剔除（K 是一个参数，供后续调整，目前设为 20）。我们将随机选择的 20%用户的最后一个会话作为测试会话，并在每个测试会话中随机删除一个项目作为下一个要预测的项目。然后，所有会话（包括已处理的会话）都被分为长期和短期会话来训练模型。

缺失值个数统计

In [4]:
```
missing_data = data.isnull().sum()
missing_data = missing_data[missing_data != 0]
missing_data
```

Out[4]:
```
age_range       19380
gender          61712
activity_log     2975
dtype: int64
```

将缺失部分剔除

由于缺失值占总数较少，因此可以直接删除掉包含缺失值的整条数据。

In [5]:
```
print('原始数据行数:', len(data))
drop_data = data.dropna(how='any')
print('将缺失部分剔除后数据行数:', len(drop_data))
```

```
原始数据行数: 7030723
将缺失部分剔除后数据行数: 6965801
```

2. 数据分析与可视化

输出数据集中的关联规则，Metric 选为 conviction 和 lift，下表包含了 conviction 大于 1 的规则，其中，发现 18 到 24 岁的用户很少重复去某家店购买商品，同时女性用户不喜欢填写具体的年龄。

	antecedents	consequents	conviction	lift
0	(18~24)	(non-repeat)	1.239229	1.012574
1	(male)	(non-repeat)	1.136363	1.007816
2	(unknown years)	(female)	1.101078	1.043945
3	(unknown years, non-repeat)	(female)	1.088355	1.038862
4	(unknown years)	(non-repeat, female)	1.085804	1.045891
5	(unknown years)	(non-repeat)	1.050385	1.003124
6	(25~29)	(non-repeat)	1.039696	1.002487
7	(non-repeat, female)	(unknown years)	1.013016	1.045891
8	(female)	(unknown years)	1.012457	1.043945
9	(female)	(unknown years, non-repeat)	1.010186	1.038862

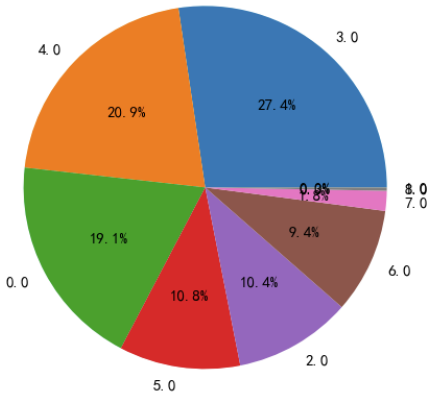
	antecedents	consequents	conviction	lift
10	(non-repeat)	(male)	1.003091	1.007816
11	(non-repeat)	(18~24)	1.001700	1.012574
12	(non-repeat)	(25~29)	1.000902	1.002487
13	(non-repeat)	(unknown years)	1.000876	1.003124

1）年龄和性别数据分布

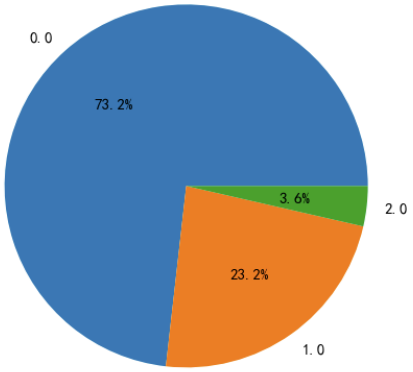
In [7]:
```
plt.rcParams['font.sans-serif'] = [u'SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

In [13]:
```
for field_name, field in [('年龄范围', 'age_range'), ('性别', 'gender')]:
    data_field = drop_data[field].value_counts()
    data_field.plot.pie(autopct='%0.1f%%', title=field_name+'数据分布', figsize=(8, 8),
fontsize=15)
    plt.ylabel('')
    plt.show()
```

年龄范围数据分布



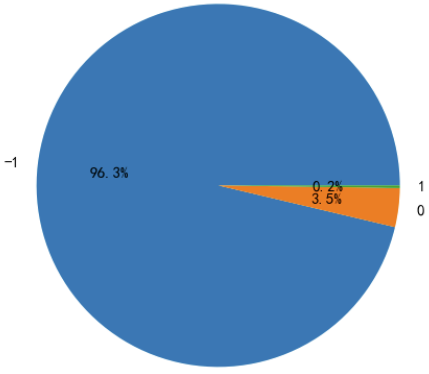
性别数据分布



2）标签数据分布

In [15]:
```
for field_name, field in [('标签', 'label')]:
    data_field = drop_data[field].value_counts()
    data_field.plot.pie(autopct='%0.1f%%', title=field_name+'数据分布', figsize=(8, 8),
fontsize=15)
    plt.ylabel('')
    plt.show()
```

标签数据分布



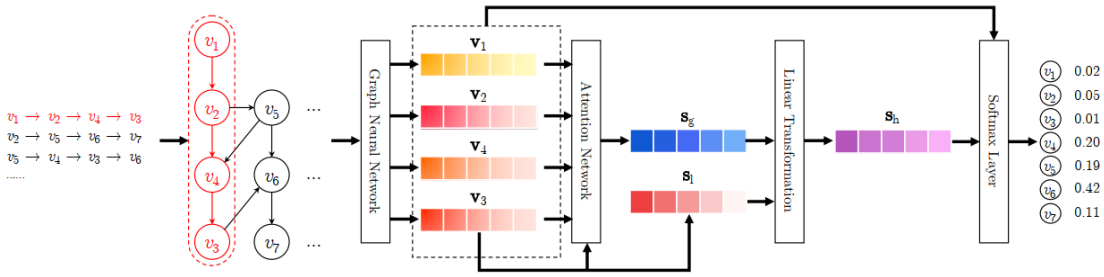
3. 模型选取

根据促销期间用户购买商品的行为，来预测在促销结束后用户购买商品的行为，等价于一个推荐系统(recommendation system)。

- SR-GNN

基于会话的推荐系统旨在通过会话来预测用户行为。但是传统基于序列的推荐算法并不能获取 items 间的复杂的转移关系，因此该模型提出了基于 GNN 的方法。

SR-GNN 的整体框架如下图所示：



在基于会话的推荐系统中，设 $V=\{v_1,v_2,...,v_m\}$。其中，Vs,i 表示用户在会话 $s=[v_{s,1},v_{s,2},...v_{s,n}]$ 中点击的物品。因此基于会话的推荐系统的目标就是预测用户的下一次点击，比如 Vs,n+1。当然在基于会话的推荐系统中，对于某一会话 s，系统一般给出输出概率最高的几个预测点击目标，作为推荐的候选。

每一个会话序列 s 都可以被建模为一个有向图 Gs=(Vs,Es)。在该会话图中，每个节点都代表一个物品 Vs,i，每一条边(Vs,i-1, Vs,i)代表在会话 s 中，用户在点击了物品 Vs,i-1 后点击了 Vs,i。因为许多 item 可能会在会话序列中多次出现，因此论文给每一条边赋予了标准化后的加权值，权重的计算方法为边的出现次数除以边起点的出度。论文将每个 item 通过 GNN 都映射到一个统一的词嵌入空间中，且节点对应的词嵌入向量 v 表示通过图神经网络学到的词嵌入向量。基于每个节点的词嵌入向量的表示形式，每个会话 s 就可以嵌入向量表示：各个节点的词嵌入向量按时间顺序拼接而成。

GNN 十分适合用于基于会话的推荐算法，因为它可以根据丰富的节点连接自动提取会话图的特征。门控图神经网络(Gated GNN)的更新可以由如下公式给出：

$$\mathbf{a}_{s,i}^{t} = \mathbf{A}_{s,i:}\left[\mathbf{v}_1^{t-1}, \ldots, \mathbf{v}_n^{t-1}\right]^{\top}\mathbf{H} + \mathbf{b}, \quad (1)$$

$$\mathbf{z}_{s,i}^{t} = \sigma\left(\mathbf{W}_z\mathbf{a}_{s,i}^{t} + \mathbf{U}_z\mathbf{v}_i^{t-1}\right), \quad (2)$$

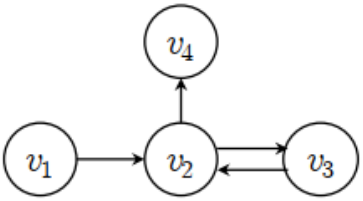
$$\mathbf{r}_{s,i}^{t} = \sigma\left(\mathbf{W}_r\mathbf{a}_{s,i}^{t} + \mathbf{U}_r\mathbf{v}_i^{t-1}\right), \quad (3)$$

$$\widetilde{\mathbf{v}}_i^{t} = \tanh\left(\mathbf{W}_o\mathbf{a}_{s,i}^{t} + \mathbf{U}_o\left(\mathbf{r}_{s,i}^{t} \odot \mathbf{v}_i^{t-1}\right)\right), \quad (4)$$

$$\mathbf{v}_i^{t} = \left(1 - \mathbf{z}_{s,i}^{t}\right) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^{t} \odot \widetilde{\mathbf{v}}_i^{t}, \quad (5)$$

上式中的 H 控制着权重，$z_{s,i}$, $r_{s,i}$分别代表重置(reset)和更新(update)门。$[v_1^{t-1}, ..., v_n^{t-1}]$是会话 s 包含的节点向量。$A_s \in R^{n\times 2n}$是关系矩阵，决定着图中的节点彼此间如何关联的。$A_{s,i:} \in R^{1\times 2n}$是$A_s$中与节点 Vs,i 相关的两列(因为会话图是有向图，因此这的两列分别对应的是当前节点到其他节点和其他节点到当前节点对应的关系系数)。

A_s定义为两个邻接矩阵$A_s^{(out)}$和$A_s^{(in)}$的拼接，分别表示传入边和传出边的连接权重。具体例子如下图所示：



	Outgoing edges				Incoming edges			
	1	2	3	4	1	2	3	4
1	0	1	0	0	0	0	0	0
2	0	0	1/2	1/2	1/2	0	1/2	0
3	0	1	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0

SR-GNN 能够根据不同的会话数据构建相应的会话图，例如当节点存在描述和分类信息等内容特征时，该方法可以进一步推广，具体来说，可以将数据特征与节点向量相对应起来处理这些信息。

对于每个会话图 Gs,门控图神经网络(Gated GNN)同时对所有的节点进行处理。其中(1)式是用于在关系矩阵 As 的监督下进行不同节点间的信息传播，具体而言就是对于每个节点提取其相邻节点的关系生成隐向量输入后续的 GNN 中。然后重置门和更新门分别决定哪些信息需要丢弃和保留。之后，用(4)式中的目前状态，之前的状态和重置门计算候选状态。最后根据(5)式用候选状态，前一状态和更新门计算表出最后的状态。重复上述过程直至收敛，即可得到最后的各个节点的词嵌入向量。

通过将所有的会话图送入 G-GNN 中能够得到所有节点的嵌入向量。接下来，为了将每个会话表示为嵌入向量$s \in R^d$，首先考虑局部嵌入向量 sl,对于会话$s=[v_{s,1},v_{s,2},...v_{s,n}]$，局部嵌入向量可以简单定义为会话中最后点击的物品 Vs,n，对于具体的 session 也可以简单表示为 Vn，即 sl=Vn。

然后，论文结合所有节点嵌入向量来计算会话图的全局嵌入向量 sg，鉴于不同节点信息可能存在不同的优先级，为了使全局嵌入向量有更好的表现，论文引入了 soft-attention 机制。

$$\begin{aligned}\alpha_i &= \mathbf{q}^\top \sigma(\mathbf{W}_1\mathbf{v}_n + \mathbf{W}_2\mathbf{v}_i + \mathbf{c}),\\ \mathbf{s}_g &= \sum_{i=1}^{n} \alpha_i \mathbf{v}_i,\end{aligned} \tag{6}$$

对于每个节点对应的词嵌入向量进行加权求和得到最后的全局词嵌入向量。最后将会话的局部嵌入向量和全局嵌入向量相结合即可得到融合的嵌入向量。

$$\mathbf{s}_h = \mathbf{W}_3 \left[\mathbf{s}_l; \mathbf{s}_g\right], \tag{7}$$

得到每个会话的嵌入向量后论文对每个候选物品计算得分：

$$\hat{\mathbf{z}}_i = \mathbf{s}_h^\top \mathbf{v}_i. \tag{8}$$

将得分经过一个 softmax 激活函数后得到模型的预测输出：

$$\hat{\mathbf{y}} = \mathrm{softmax}\left(\hat{\mathbf{z}}\right), \tag{9}$$

对于每个会话图，损失函数选用常见的交叉熵函数：

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{m} \mathbf{y}_i \log\left(\hat{\mathbf{y}}_i\right) + \left(1-\mathbf{y}_i\right)\log\left(1-\hat{\mathbf{y}}_i\right), \tag{10}$$

模型的训练采用基于时间的方向传播方法(BPTT)，由于在基于会话的推荐方案中，大多数会话长度较短，因此训练迭代次数不要太多，防止过拟合。

4. 挖掘实验的结果

根据具体的实验数据，我们调整了预测的目标，对于测试集中的用户和商家，不是预测用户是否会继续在该商家进行购买。而是根据用户现有 session，去预测这个 session 中用户可能点击的下一个物品。

经过预处理后，我们对数据进行了统计。

用户数量	93302
物品数量	27211
种类数量	793
训练集 session 数量	246882
测试集 session 数量	7705

通过训练模型，我们对 Precision 和 MRR(Mean Reciprocal Rank)进行了评估。

Precision@20：表示前 20 个推荐项目中正确推荐的准确率。

MRR@20：代表正确推荐的物品优先级排名倒数的平均值，只计算前 20 个推荐中正确推荐的排名均值。MRR 度量考虑推荐排名的顺序，其中较大的 MRR 值表示正确的推荐位于排名列表的顶部。

	Precision@K	MRR@K
K=5	25.5678%	20.0627%
K=10	28.0337%	20.4691%
K=20	30.1103%	20.5339%

5. 存在的问题

在现有的模型中，我们只考虑了用户的购买行为，但从数据中可以看到，用户存在各种不同的行为，如点击，购买，加入购物车，收藏等，这些行为没有在模型中进行体现，有待改进。

此外，没有显示考虑用户的兴趣，对于每一个用户，应该存在一个固有的购买习惯或行为，在当前模型中，我们没有考虑这一特征。

6. 下一步工作

对用户的各种行为和用户的固有购买特征进行建模，并在模型中进行体现。

Reference

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, Tieniu Tan: Session-Based Recommendation with Graph Neural Networks. AAAI 2019: 346-353