1 赛题背景分析与理解

该赛题是面向大规模电商平台设计的，任务要求在很短时间内从千万级的商品库 C 中为用户挑选出最可能感兴趣的 k 个商品。其中，$k << n$，$n = |C|$。复赛还要求为每个用户进行推荐时的时间复杂度小于 $O(n)$。此外，复赛提交的方案需在一个 8 核 60G P100 的 GPU 容器中对 6 万线上用户进行推荐，限时 1 小时。不仅对复杂度有要求，对内存、CPU 等资源也有限制。

数据集包括用户行为文件、用户信息文件与商品信息文件。用户信息包含用户 ID、性别、年龄与购买力，商品信息包含商品 ID、类目 ID、店铺 ID 与品牌 ID（若有商品价格，有望提高推荐效果），用户行为涉及 16 天（由某个周五开始）的用户对商品的行为日志。

比赛要求预测一组给定用户在第 17 天感兴趣的商品列表。需要注意的是，初赛与复赛的方案评价方式有较大差别：

（1）初赛提供了待预测用户的信息、第 1~16 天的行为日志及感兴趣的商品信息，参赛选手可以仅适用待预测用户的信息设计方案，将预测结果提交到线上进行评测，评价指标为 Recall@50 与 Novel-Recall@50 的加权均值（经我们分析可能为 Recall@50*0.15+ Novel-Recall@50*0.85）。其中，Novel-Recall@50 要求推荐的商品不能与历史感兴趣商品属同一类别，因而难度很大。

（2）复赛将待预测的用户信息等文件置于线上，不允许打印相关信息等内容，而且对运行时间及资源又添加了限制。利用线上用户行为日志等信息建模效果尚可，但复杂度可能会超出要求，因而很多信息及模型需要在线下统计、训练。此外，评价指标变为了 Recall@50，并要求推荐的商品不能与历史感兴趣商品相同。该指标比初赛简单些，因为可以推荐同类商品，这在真实业务及该数据集中都较常见。

2 核心思路

初赛方案仅基于规则做了 Match 阶段，里面有些技巧，感兴趣的同学可以关

注 https://github.com/ChuanyuXue，之后会在上面发布代码。下面重点阐述复赛方案。图 1 给出了推荐系统的经典流程，先从千万级商品库中为指定用户召回几百或几千个候选商品，再建模为候选商品排序，选出少量商品作为最终的推荐列表。

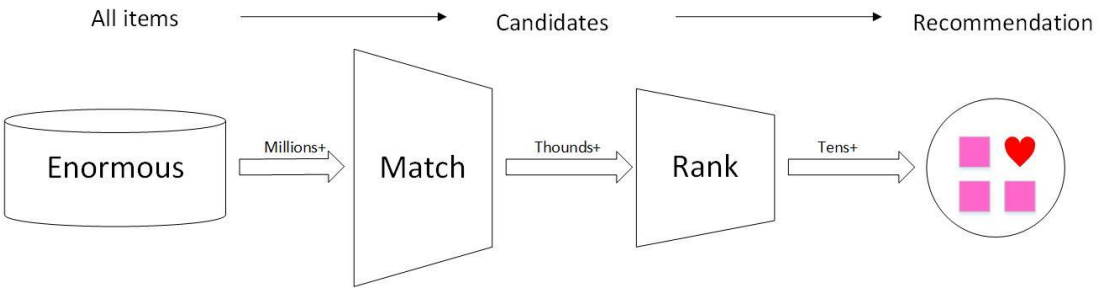


图 1 推荐系统经典流程

2.1 EDA

数据分析与探索对方案设计有重要的指导作用。下面介绍几个关键的分析。在做 EDA 时，我们将数据集切分为了两部分，第 1~14 天日志被视为“历史”行为，第 15 天日志视为“未来”行为，从而可以分析对“未来”行为有重要影响的“历史”行为特点。

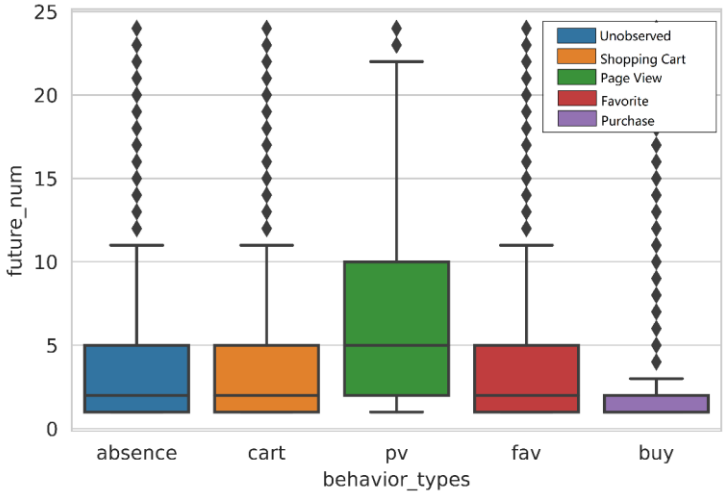


图 2 用户对“历史”感兴趣同类商品的“未来”行为统计分析。

用户行为共有 4 种类型：’pv’（浏览）、’fav’（喜欢）、’cart’（加入购物车）和’buy’（购买）。按照感兴趣程度，可将这 4 种类型的权重依次设为 1、2、3、4

（论坛发布的初赛 baseline 即是这样设置，效果尚可）。图 2 先获取了用户“历史”感兴趣的商品类别，然后统计了“未来”对历史感兴趣的同类别商品的行为。图 2 表明“未来”感兴趣的商品（出现在第 15 天日志中的商品）几乎不会是以往购买过的同类商品。因而，我们在复赛方案中将’buy’的权重设为 1。实际上，4 种行为的权重仍可调优，但我们限于时间和精力未做。

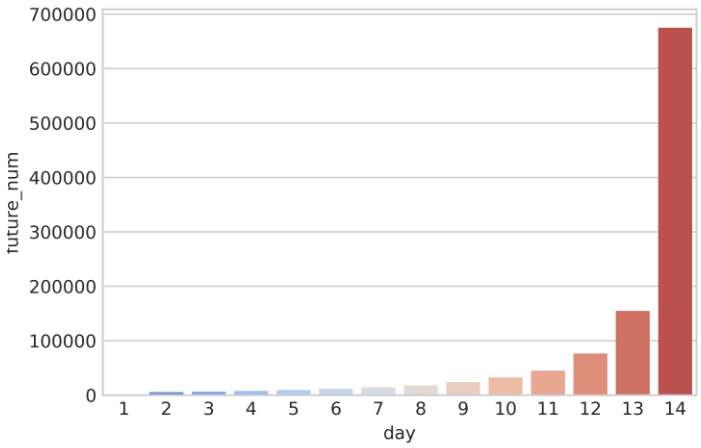


图 3“未来”感兴趣商品在第 1~14 天被感兴趣的次数

如图 3 所示，“未来”感兴趣商品在第 14 天被感兴趣的次数组多，距第 14 天越远次数越少。因而，我们考虑时间因素对行为重要性的影响，按下式调整行为权重：

$$T_{u,i} = 1 - \left(\frac{\max(D) - D_{u,i} + 1}{D_{\max} - D_{\min} + 1} \right)$$

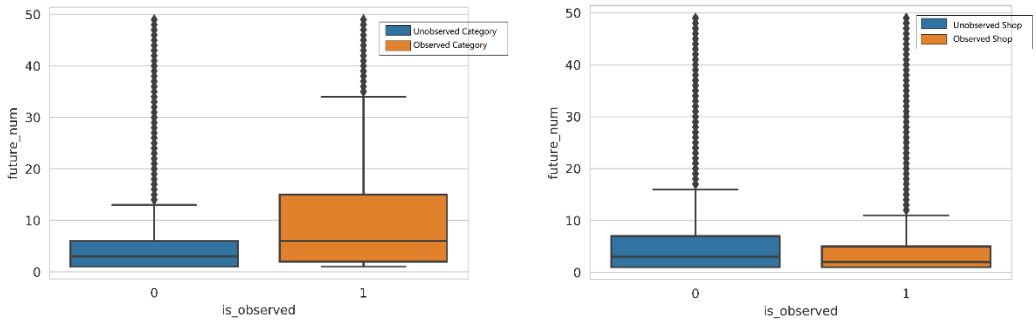
$$V_{u,i} = \max(s_{pv}, s_{fav}, s_{cart}, s_{buy})$$

$$R_{u,i} = T_{u,i} * V_{u,i}$$

其中，s_{pv}、s_{fav}、s_{cart}、s_{buy}是四种行为的权重，$T_{u,i}$代表距最大时间戳$D_{\max}$的远近，$R_{u,i}$是考虑时间因素后评估用户 u 对商品 i 的感兴趣程度。

图 4 没有区分行为的种类，统一分析了用户在“未来”是否仍会对“历史”感兴趣的商品类别及店铺感兴趣。如图 4-(a)所示，用户在“未来”仍会对“历史”感

兴趣的商品类别有较高兴趣；图 4-(b)则表明，用户在“未来”对历史感兴趣的店铺有较低的兴趣。进而，我们针对类别/店铺提取了一些特征，详见对排序阶段的介绍。



(a) (b)

图 4 用户是否仍会对“历史”感兴趣的商品类别及店铺感兴趣。

2.2 召回阶段

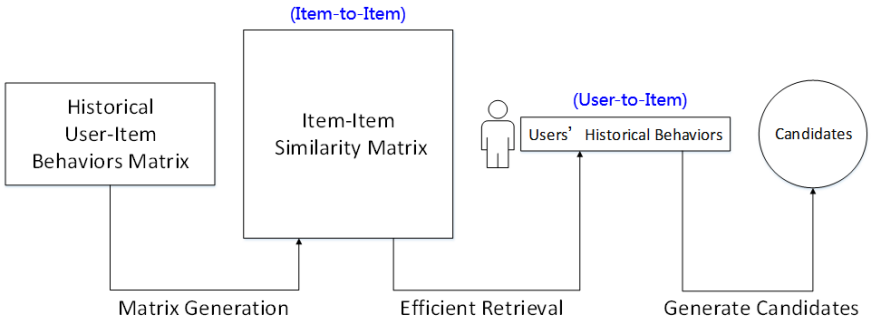


图 5 基于 Item CF 的召回流程

召回的策略有很多，即使是基于规则的策略效果也可以。在复赛后期，我们花费了很大精力实现了一种 Item CF 算法，效果也有明显提升。图 5 给出了基于 Item CF 做召回的流程，先利用庞大的历史日志统计 item-item 相似性矩阵，再结合目标用户的历史行为做推荐。实现的难点在于对约 8000 万历史日志做统计的复杂度太高，需要做优化代码、做并行化处理。

如图 6 所示，我们将用户分为了若干组，并行处理每组内 item-item 共现频率的统计，最终将与每个商品最相似性的 500 个商品存在字典中。实际上，对复赛训练集统计后，发现字典中键值数仅有 40 多万。 为了提高效率，我们使用了 Cpython 实现统计共现频率的代码。整个流程较复杂，感兴趣的同学可以

看答辩后开源的代码。

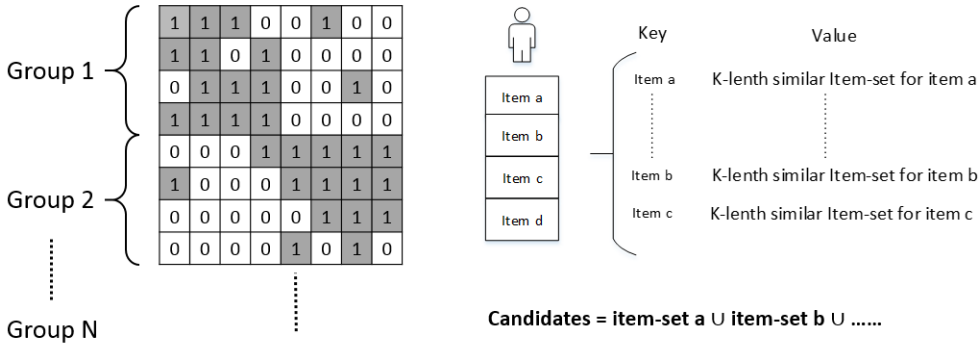


图 6 并行统计 item-item 相似性，并转存为字典

Item CF 相似性指标关乎召回的效果。我们在实现时借鉴了 2015 年腾讯 SIGMOD 论文 [1]。在 9 月初，我们按照关联规则中置信度计算 Item CF 相似性：

$$Sim(i,j) = Confidence(i,j) = P(j|i) = \frac{|U_i \cap U_j|}{|U_i|}$$

其中，U_i代表对商品i感兴趣的用户集合。显然，$Sim(i,j) \neq Sim(j,i)$。基于该指标做召回，线上效果为 0.045。

在此基础上，我们又考虑了用户活跃度（感兴趣的商品数）对相似性的影响，改进了上述指标：

$$\mathrm{Sim}^{w}(i,j) = \frac{\sum_{u \in U} w_u \delta(i,j)}{\sum_{u \in U_i} w_u}$$

其中，U是全体用户集合，U_i是对商品 i 感兴趣的用户集合；w_u代表用户 u 对相似性的贡献度，$w_u = \frac{1}{\log(I_u)+1}$，，$\delta(i,j) = \begin{cases} 1, & i \in I_u \ and \ j \in I_u \\ 0, & else \end{cases}$，$I_u$代表用户感兴趣的商品集合。当$w \to 1$时，$\mathrm{Sim}^{w}(i,j)$等价于$Sim(i,j)$。基于改进指标做召回，并做了些额外处理，线上效果为 0.053。

2.3 排序阶段

召回阶段获得少量（300 或 500）候选商品后，可以构建排序模型获得最终的推荐列表。我们将排序任务转化为二类判别问题。在建模前，需要切分数据集。如图 7 所示，我们利用第 1-15 天数据做召回、生成特征，利用第 16 天的

数据生成标签，从而生成线上训练集；利用 1-16 天数据做召回、生成特征，生成线上测试集，加载训练后的模型及相关文件完成预测。

需要特别注意的是，训练集中的正样本和负样本都是从召回列表中生成的，而不是将每个用户感兴趣的商品都拿出来做正样本。这是因为，很多用户感兴趣的商品对应的特征取值都无法统计，使得这些正样本失去了统计意义，对训练模型有负面影响。

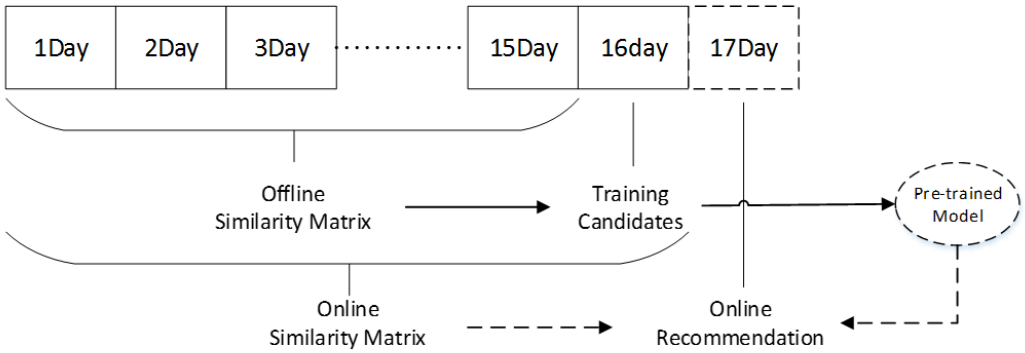


图 7 排序阶段划分数据

图 8 为提取的特征列表，只有 64 个。其中，Item CF 的相似性特征是强特征。我们使用了 Catboost 和 Lightgbm 建模。Catboost 对过拟合的处理较好，使用了全部特征（线上效果为 0.0616）；Lightgbm 使用全部特征效果不佳，故做了特征选择，最终只使用了 36 个特征。

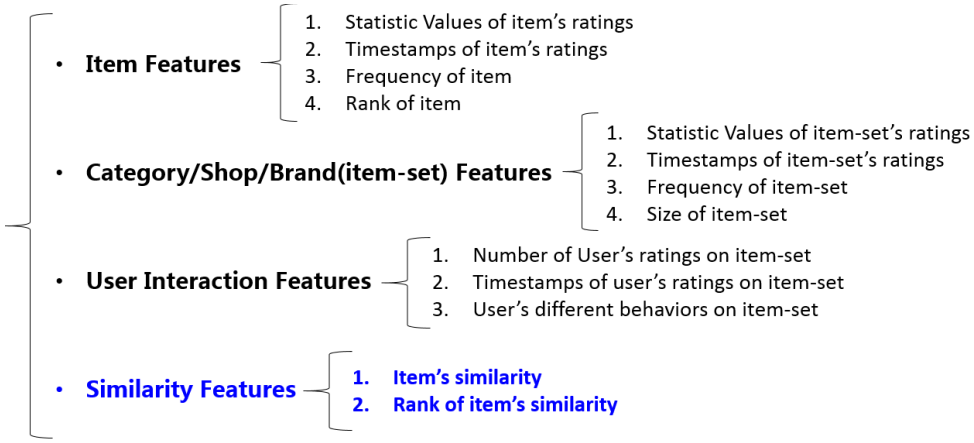


图 8 特征列表（共 64 个）

我们使用了多种特征选择方法，其中一种独立性检验方法表现较好。该方

法基于 Mean Variance Index[2,3]做特征选择，这是首都师范大学崔恒建教授 2015 年发表于统计领域顶刊 JASA 的工作，2018 年进行了拓展，可用于做独立性检验及特征选择。该方法可检验一个离散型变量与一个连续型变量间是否独立，对变量的分布无假定（Distribution free），并且计算简单（只是计数）。这里仅列出其部分理论（图 9），感兴趣的同学可以交流，该方法已被 Chuanyu 做成了工具包，已开源在他的 github。此外，我们在 IJCAI 2018 和资金流入流出预测挑战赛中都使用 Mean Variance Index 做过特征选择，效果都不错。

Feature Selection

- **MV Test**: Mean Variance Test (***JASA** 2015)*
- Distribution free test of Independence (https://github.com/ChuanyuXue/MVTest)
- Mean Variance Index (X: a continuous r.v.; Y: a categorical one):

$$MV(X|Y) = E_X[Var_Y(F(X|Y))] \; where \; F(x|Y) = P(X \le x|Y)$$

- Testing hypothesis:

$H_0: F_r(x) = F(x)$ for any x and r=1,.. .,R
$H_1: F_r(x) \neq F(x)$ for some x and r=1,.. .,R

- Test statistic:

$$T_n = n\widehat{MV}(X|Y)$$
$$= \sum_{r=1}^{R} \sum_{i=1}^{n} \widehat{p_r} * [\hat{F}_r(X_i) - \hat{F}(X_i)]^2$$
$$where, F(x) = P(X \le x), F_r(x) = P(X \le x|Y = y_r)$$

图 9 Mean Variance Test 简介

最后，我们进行了简单的模型融合。为了提高稳健性，我们依次采用了调和平均值、几何平均值和算数表均值（图 10），线上效果为 0.0622。

Model Averaging

- **3 steps**
- Step 1: averaging lightgbm and catboost with Harmonic Mean
- Step 2: averaging lightgbm and catboost with Geometric Mean
- Step 3: Harmonic Mean * 0.5 + Geometric Mean * 0.5

图 10 模型融合

2.4 其他尝试

我们还有一些基于规则的策略及其他方案没有介绍。例如，基于同类商品的

规则做召回、基于同店铺的规则做召回、基于 word2vector 的思路做召回（借助 faiss）、基于 MinHash LSH 做 Item CF、取最近 100 条用户行为做统计等等。感兴趣的同学可以交流。

3 比赛总结和感想

QDU 由青岛大学本科生薛传雨（小雨姑娘）、春秋航空算法工程师张卓然（人畜无害小白兔）、青岛大学讲师吴舜尧（BruceQD）组成，我们都曾在一些数据挖掘比赛取得过 Top 1、Top 2、Top3 或 Top10 的成绩。本攻略与总结由吴舜尧撰写。

参加 CIKM 挑战赛的原因有二：（1）希望验证自身技术和研究价值；（2）参加会议，与专家交流，帮助薛传雨申请博士。受限于复赛任务要求，我们没能在比赛中使用开发的推荐系统框架（一种基于组间效应的增量推荐系统框架[4]）。运气好的是，我们可以去答辩，并有机会去 workshop 分享了。现在如果在比赛中拿不到 Top，很难作为简历的亮点，拿不到好的 offer。12 年我做 KDD CUP 时，只是 100 名经过三轮面试就拿到了百度的 offer（也可能是因为我还有 SIGIR 和 JCDL 的 poster）；17 年我带的研究生找工作时，前 100 或前 50 的排名已经拿不到 offer，幸好之后拿到了两个比赛的 Top1 和 Top2，才拿到了 offer；今年感觉更难了，普通公司都要面试 2~3 轮，好公司要面试 5~6 轮。

想法比套路重要得多。大家在做比赛时，应该把精力放在数据分析与探索，从而提取有用的规则，利用规则进行初步想法的验证；进而，基于规则生成特征，再考虑建模、模型融合。另一方面，建议大家学好统计，读读统计学领域的论文，有助于加深对机器学习的理解。此外，在比赛后几天，要休息好、能沉住气，不能过于急躁。

参考文献

[1] Y. Huang et al. Tencentrec: Real-time stream recommendation in practice. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 2015: 227-238.

[2] H. Cui et al. Model-free feature screening for ultrahigh dimensional discriminant analysis. *Journal of the American Statistical Association*. 2015, 110(510): 630-641.

[3] H. Cui et al. A Distribution-Free Test of Independence and Its Application to Variable Selection. *arXiv preprint arXiv:1801.10559*, 2018.

[4] C. Xue et al. An Incremental Group-Specific Framework Based on Community Detection for Cold Start Recommendation. *IEEE Access*. 2019, 7: 112363-112374.