# 职位推荐：分类广告协同过滤方法的基准测试

RobertKwieci´nski[a, b] (r.kwiecinskipl@gmail.com), Agata Filipowska[a, c] (agata.filipowska@ue.poznan.pl), Tomasz G´orecki[b] (to masz.gorecki@amu.edu.pl), Viacheslav Dubrov (slavadubrov@gmail.com)

[a] OLX Group, Kr´olowej Jadwigi 43, 61-872 Pozna´n, 波兰[b] 亚当·密茨凯维奇大学, Wieniawskiego 1, 61-712 Pozna´n, 波兰[c] Pozna´n University of Economics and Business, Al. Niepodleg-lo´sci 10, 61-875 Pozna´n, 波兰

**通讯作者：**

Robert Kwieci´nski 亚当·密茨凯维奇大学, Wieniawskiego 1, 61-712 Pozna´n, 波兰 电话: +48 888 791 384电子邮箱: r.kwiecinskipl@gmail.com

arXiv:2301.07946v1 [cs.IR] 19 Jan 2023

# 职位推荐：分类广告协同过滤方法的基准测试

Robert Kwieci´nski[a, b, *], Agata Filipowska[a, c], Tomasz G´orecki[b], Viacheslav Dubrov

[a]*OLX Group, Królowej Jadwigi 43, 61-872 Poznań, Poland*[b]*Adam Mickiewicz University, Wieniawskiego 1, 61-712 Poznań, Poland*[c]*Poznań University of Economics and Business, Al. Niepodległości 10, 61-875 Poznań, Poland*

**摘要**

由于有关用户和物品的信息有限，分类广告给推荐方法带来了诸多挑战。本文以 OLX Jobs 为例，探讨了适用于分类广告的推荐方法。

本文旨在对不同的职位分类广告推荐方法进行基准测试，以提高广告转化率和用户满意度。

在研究中，我们实现了具备可扩展性且代表不同推荐思路的方法，分别为 ALS、LightFM、Prod2Vec、RP3beta 和 SLIM。我们从准确率、多样性和可扩展性（训练和预测期间的内存及时间消耗）等方面，对这些方法进行了实验室对比。我们还通过发送数百万条推荐消息开展了在线 A/B 测试，以在真实场景中评估模型。

此外，我们还公开了为本研究创建的数据集。据我们所知，这是首个此类数据集。该数据集记录了 2020年 两周内 3,295,942 名用户在 OLX Jobs 上执行的 65,502,201 次事件，这些用户与 185,395 条招聘广告进行了互动（浏览、回复或添加书签）。

我们通过实验室环境下的测试证明，RP3beta、SLIM 和 ALS 的表现显著优于 Prod2Vec 和 LightFM。在线 A/B 测试还表明，发送由 ALS 和 RP3beta 模型生成推荐结果的消息，可以增加联系广告发布者的用户数量。此外，RP3beta 对这一指标的影响比 ALS 高出 20%。

关键词：职位推荐、分类广告、协同过滤、可扩展性测试、A/B 测试

## 1. 引言

在线分类广告是指广告发布者发布有关销售或租赁服务或产品广告的网站（Davis，2021年）。预计全球在线分类广告市场将在2024年达到323.3 亿美元，2020年至2024年的复合年增长率为 9.40%（Research and Markets，2021年）。本文以OLX Jobs为例，探讨在线广告。

*通讯作者*Email addresses:* r.kwiecinskipl@gmail.com (Robert Kwieci´nski ), agata.filipowska@ue.poznan.pl (Agata Filipowska), tomasz.gorecki@amu.edu.pl (Tomasz G´orecki), slavadubrov@gmail.com (Viacheslav Dubrov)

分类广告平台努力满足客户的需求，包括广告发布者和在线广告的响应者。为了更好地匹配这两类群体，推荐方法日益重要。与传统电子商务（例如亚马逊或易贝）相比，分类广告平台需要应对不同的挑战。其中之一是广告的时间可见性，这既可能由展示时长有限造成，也可能是因为广告发布者通常只需要一位用户来完成交易。

我们的研究聚焦于协同过滤领域中涌现的方法。协同过滤模型即使不利用有关用户或物品的额外信息，也已证明具有较高的准确率，而这些额外信息在实践中有时难以获取或处理。本文旨在评估不同推荐方法在分类信息行业中的应用。数据集的高度稀疏性，以及广告和用户数量庞大，都增加了研究问题的复杂性。

我们的主要研究贡献如下：在实验室环境中全面评估了多种可扩展的协同过滤方法，并提供了针对超过 100 万名用户开展的在线 A/B 测试结果。此外，我们还发布了一个求职互动数据集。据我们所知，这是目前公开可用的规模最大的求职互动数据集。

本文共分为七个章节。第 2 节回顾了相关领域的研究进展。第 3 节介绍了我们开发并发布于 Kaggle 的数据集。第 4 节介绍了实验室实验的设计以及各方法的实现。第 5 节重点介绍各项实现的实验室评估结果，并讨论结果的统计显著性。第 6 节介绍了在面向用户开展 A/B 测试过程中进行在线实现所取得的结果，并对结果进行了讨论。第 7 节给出结论。

## 2. 相关工作

### *2.1.* 分类广告网站

分类广告网站（分类广告）是报纸广告版块的在线版本，刊登有关车辆、房地产、就业、宠物等方面的广告。分类广告网站是一种特殊类型的电子商务网站，因为它们不支持交易双方完成交易，而是用于交流产品或服务信息并安排会面，交易则在线下进行；这使其有别于易贝或亚马逊等传统电子商务网站（Jones，2009年）。我们的研究聚焦于 OLX，该平台在全球 30 多个国家开展业务。

分类广告网站刊登的广告类型之一是职位招聘。Johnson and Johnson（1995年）分析了招聘广告的格式，以及这些广告如何描述雇主对候选人所需技能的偏好。在我们的研究中，我们确定了以下特征，这些特征使分类广告网站上的招聘广告有别于例如汽车或房地产广告：

- 广告中列出的要求表明了可能会响应广告的用户类型——并非所有用户都具备所需的能力。

- 用户在搜索过程中参与度更高，因为与例如买车相比，找工作是更为迫切的需求。

- 工作的地理位置对用户影响很大——购买手机或汽车时，用户可以亲自前往或使用快递服务运送物品，但如果工作地点远离用户目前的居住地，通常就需要搬迁。

- 从技术角度来看，职位描述有许多定性方面，而且根据职位不同，即使是同一家公司的广告，彼此之间也可能存在显著差异。

除了这些特点，在研究过程中，我们还发现了分类广告推荐方法面临的以下挑战，这些挑战影响了我们对这些方法的实现：

- 用户数量和职位数量：在 OLX Jobs 的案例中，我们面对的是数百万用户和数万个职位，而且两者都在不断变化。

- 用户无需创建个人资料即可与广告互动，这意味着无法利用个人资料中的信息进行推荐；推荐结果取决于用户浏览时的用户行为。

- 广告的访客（与广告互动的用户）数量通常有限，因为广告通常针对某个独特的职位空缺；需求得到满足后，广告就会被停用。

- 无法得知是否发生了交易；在分类广告平台上，转化与否取决于是否收到对已发布广告的回复，因为任何交易都是在线下完成的。

这些挑战不仅给实现带来了困难，也使测试所提方法的准确率变得困难。我们在准备数据集以及实现本文所述方法时，都考虑了这些挑战。

### *2.2.* 推荐系统

研究表明，过度选择 （Gourville 和 Soman，2005年）可能会降低产品消费量（Chernev，2003年；Iyengar 和 Lepper，2001年），并减少网站收入（Boatwright 和 Nunes，2001年）。为了减少向用户提供的选项数量，服务提供商可以询问用户更多有关其对物品的期望的信息。即使用户愿意花时间提供这些信息，可能的选项数量通常仍然太多，无法全部呈现给用户。为了解决这一问题，Netflix（G´omez-Uribe 和 Hunt，2015年）和亚马逊（Smith 和 Linden，2017年）等许多公司都创建了个性化推荐系统，用于向用户推荐相关物品（Aggarwal，2016年）。

推荐结果应用于多个领域，包括音乐（Song等人，2012）、旅游（Levi等人，2012）、时尚（Hwangbo等人，2018）和食品（Ge等人，2015）。电子商务（Li和Karahanna，2015；Jiang等人，2019）、分类广告（Twardowski，2016）和职位推荐（Alotaibi，2012；RecSys 挑战赛 ’16；RecSys 挑战赛 ’17）领域也开展了多项研究。

通常，我们将提供推荐结果的方法分为两类：基于内容的推荐和协同过滤（Aggarwal, 2016）。

在基于内容的推荐中（Pazzani and Billsus，2007年；Javed等人，2021年），我们利用有关用户（例如，爱好、教育、技能）和物品（例如，价格、位置、类别）的信息。这些系统通常也会利用单个用户的交互历史（评分、访问、购买、回复），但不会同时考虑所有用户的评分。

收集和处理有关用户和物品的信息并非总是可行的。在协同过滤方法中（Chen等人，2018年；Wang等人，2019年），我们仅利用用户与物品之间交互的信息。即使缺少用户特征和物品特征的信息，这些系统也能够提供非常准确的推荐结果。我们为这个问题选择了这种方法，因为缺乏用户特征，而且在生产环境中利用物品特征很难超越协同过滤（我们将在第6.1节中详细讨论这一点）。

协同过滤最大的缺点之一，是无法为没有任何交互的用户和物品提供推荐，这被称为冷启动问题（Schafer等人，2007年）。减少这一问题的一种可能解决方案是频繁重新训练模型。当用户交互数量庞大时，这一策略就要求重点关注模型的可扩展性（Tak´acs等人，2009年）。

决定模型选择的另一个重要因素，是用户与物品之间的交互类型（Schafer等人，2007年）。当用户明确表示自己对物品的偏好时（例如，在1–5的量表上为其评分），我们称这种反馈为显式反馈 。另一种类型是隐式反馈，此时我们只能根据用户行为（例如购买物品、观看电影）推断用户对物品的偏好。隐式反馈的准确性较低、数量较多，而且通常只表达正向偏好（Jawaheer等人，2010年）。

对推荐系统进行实验室评估通常基于准确率指标（Tamm等人，2021年）。近年来，人们日益关注其他方面，如多样性、意外发现、新颖性和覆盖率（Kaminskas和Bridge，2016年；Kunaver和Pozrl，2017年）。对于分类广告，某个物品通常只能被一位用户购买，因此我们应避免向大量用户推荐同一物品。因此，关注目录覆盖率非常重要（Ge等人，2010年）。评估推荐系统最具挑战性的任务之一，是找出与在线目标相关性最高的离线指标（Mogenet等人，2019年）。

本文对大型隐式反馈数据集上的协同过滤模型进行基准测试。我们在实验室环境中比较了所选推荐方法的准确率、多样性和效率，随后通过面向用户的在线测试验证这些方法的质量。

### *2.3.* 推荐方法

在为研究选择方法时，我们重点关注以下特性：方法经过验证的性能（尤其关注文献中经常测试的方法），以及

该解决方案是否已在生产环境中经过测试（由公司在实际场景中进行测试），以及可扩展性（解决方案能否服务数百万用户并处理数百万件物品）。我们还希望测试代表不同“类别”的方法，以评估它们在分类广告中的适用性。最终，我们选定 ALS、LightFM、Prod2Vec、RP3beta 和 SLIM 进行测试和评估。以下各节将介绍这些方法。

*2.3.1.* 矩阵分解模型

自 Netflix 奖竞赛（Koren 等，2009年）以来，矩阵分解技术已成功应用于推荐系统。其核心思想是将稀疏的用户–物品交互矩阵近似为两个较小的稠密矩阵的乘积。我们将评估两种矩阵分解方法：LightFM 和 ALS。

**LightFM** 模型（Kula，2015年）旨在解决矩阵分解方法的冷启动问题。在我们的场景中，由于没有关于用户和物品的额外信息，该模型退化为经典矩阵分解。尽管如此，我们还是决定采用这种方法，因为它有高效的官方实现，并支持多种损失函数：逻辑损失、贝叶斯个性化排序（Rendle 等，2012年）、WARP（Weston 等，2011年）以及 $k$-OS WARP（Weston 等，2013年）。

设 U 为用户集合，I 为物品集合。对于给定的用户 $u \in$U 和物品 $i \in$I，我们定义$r_{ui} = 1$：若用户 $u$ 与物品 $i$发生过交互，则为真；否则 $r_{ui} = 0$。LightFM 模型预测分数 $r_{ui}$ 如下：

$$\hat{r}_{ui} = \boldsymbol{x}_u \cdot \boldsymbol{y}_i + b_u + b_i,$$

其中，$\boldsymbol{x}_u$ 和 $\boldsymbol{y}_i$ 分别表示用户 $u$和物品 $i$的 $k$维潜在表示，而 $b_u$ 和 $b_i$分别是用户 $u$和物品 $i$的偏置。所有这些参数均在训练过程中学习得到。

当正反馈和负反馈都可用时，应使用逻辑损失函数，因此它不适用于我们的数据集；我们在超参数调优过程中通过实验确认了这一点。

贝叶斯个性化排序、WARP 和 $k$-OS WARP 损失函数都是成对学习排序方法，与我们的 top-$k$ 推荐任务更加相关。它们试图调整模型参数，以最大化用户 $u$ 进行过交互的物品 $i \in$ D$_u$ 与用户未进行过交互的物品 $i \notin$D$_u$ 之间的得分差异。

具体而言，在 **贝叶斯个性化排序损失函数** （Rendle et al., 2012）中，我们试图最大化以下表达式：

$$\sum_{(u,i,j)\in\mathcal{S}} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) - \lambda_{\boldsymbol{\Theta}}||\boldsymbol{\Theta}||^2,$$

其中

$$\mathcal{S} = \{(u,i,j)|u \in \mathcal{U}, i \in \mathcal{D}_u \text{ and } j \notin \mathcal{D}_u\},$$

$$\sigma(x) = \frac{1}{1+e^{-x}},$$

$\lambda_{\boldsymbol{\Theta}}$ 是模型特定的正则化参数，$\boldsymbol{\Theta}$ 是参数向量，$||\cdot||$ 是欧几里得范数。

我们将把 WARP 作为 $k$-OS WARP 的一个特例来介绍。$k$-OS WARP 损失函数是针对每个用户的损失之和：

$$\sum_{u \in \mathcal{U}} L(\hat{\boldsymbol{r}}_u, \mathcal{D}_u),$$

其中，$\hat{\boldsymbol{r}}_u$ 是一个向量，包含每个物品 $i \in \mathrm{I}$ 的估计分数 $\hat{r}_{ui}$。

对于给定的用户 $u$，我们按照估计分数从高到低对交互项目 $\mathrm{D}_u$ 进行排序。也就是说，我们将物品排序为 $u_{i_1}, u_{i_2}, \ldots, u_{i_{|\mathcal{D}_u|}}$，其中索引 $i_k$ 的选取方式满足：

$$\hat{r}_{ui_1} \geq \hat{r}_{ui_2} \geq \ldots \geq \hat{r}_{ui_{|\mathcal{D}_u|}}.$$

于是：

$$L(\hat{\boldsymbol{r}}_u, \mathcal{D}_u) = \frac{1}{Z} \sum_{s=1}^{|\mathcal{D}_u|} P(s) \Phi(\mathrm{rank}_{i_s}(\hat{\boldsymbol{r}}_u)),$$

其中，$\mathrm{rank}_i(\hat{\boldsymbol{r}}_u) = \sum_{j \in \mathcal{I} \setminus \mathcal{D}_u} \mathbb{1}(1 + \hat{r}_{uj} \geq \hat{r}_{ui})$，$\mathbb{1}$ 是指示函数，$\Phi(n) = \sum_{m=1}^{n} \frac{1}{m}$，而 $Z = \sum_{s=1}^{|\mathcal{D}_u|} P(s)$ 则对由 $P$ 诱导的权重进行归一化。

实际应用中，我们按照加权函数 $P$ 对一个正物品 $i$ 进行采样，并随机选取未见过的物品 $j$，直到经过 $N$ 次采样后，满足 $1 + \hat{r}_{uj} \geq \hat{r}_{ui}$。然后，我们执行一个梯度步，以最小化 $\Phi(\lfloor \frac{|\mathcal{I} \setminus \mathcal{D}_u|}{N} \rfloor)(1 - \hat{r}_{ui} + \hat{r}_{uj})$。

最后，当 $m \neq k$ 时，$P(k) = 1$ 和 $P(m) = 0$ 可达到 $k$-OS WARP 损失。对于 $m \in \mathbb{N}$，当我们设置 $P(m) = 1$ 时，即可达到 WARP 损失。在这种情况下，为简化计算，我们可以跳过对 $\mathrm{D}_u$ 中正向物品的排序。

在我们的研究中，所有损失函数（逻辑损失、贝叶斯个性化排序、WARP 和 $k$-OS WARP）都被视为 LightFM 模型的超参数，并在训练过程中进行优化。

**ALS** 模型（交替最小二乘），又称 WRMF（加权正则化矩阵分解）（Hu 等人，2008年），是专为隐式反馈数据集设计的矩阵分解方法之一。我们选择这种方法，是因为它在性能、可扩展性方面经过验证，且在学术研究和工业界广受欢迎。此外，OLX 已实现并采用了 ALS 模型。

我们将一次观测的**置信度** 定义为 $c_{ui} = 1 + \alpha r_{ui}$，其中 $\alpha$ 是一个超参数。我们还将**偏好** 定义为 $p_{ui} = \mathbb{1}(r_{ui} > 0)$。然后，在 ALS 中，我们优化以下表达式：

$$\min_{\boldsymbol{x}_*, \boldsymbol{y}_*} \sum_{u,i} c_{ui}(p_{ui} - \boldsymbol{x}_u^\top \boldsymbol{y}_i)^2 + \lambda(\sum_u ||\boldsymbol{x}_u||^2 + \sum_i ||\boldsymbol{y}_i||^2),$$

其中，$\boldsymbol{x}_u$ 是用户 $u$ 的 $f$ 维嵌入，$\boldsymbol{y}_i$ 是物品 $i$ 的 $f$ 维嵌入，$\lambda$ 是一个正则化参数。模型的名称指的是优化该函数所用的方法。交替最小二乘过程假设用户或物品的潜在因子之一固定，并通过解析方式确定另一方。该过程以极高的效率重复数次（Hu 等人，2008年；Tak´acs 等人，2011年）。

*2.3.2.* 基于邻域的模型

**SLIM** 模型（用于 Top-N 推荐系统的稀疏线性方法；Ning 和 Karypis（2011年））基于基于物品的最近邻回归（Aggarwal，2016年）。我们选择这种方法，是因为它优于基于物品的 $k$ 近邻方法和矩阵分解模型（Ning 和 Karypis，2011年），并且相较于非神经网络和神经网络方法表现良好（Dacrema 等，2021）。

在该模型中，我们通过学习物品-物品稀疏相似度矩阵 $\boldsymbol{W}$ 来近似用户-物品矩阵 $\boldsymbol{R}$ 。具体而言，通过最小化以下表达式来学习 $\boldsymbol{W}$

$$\frac{1}{2}||\boldsymbol{R}-\boldsymbol{R}\boldsymbol{W}||_F^2+\frac{\beta}{2}||\boldsymbol{W}||_F^2+\lambda||\boldsymbol{W}||_1,$$

其中，$w_{ij}\geq 0$ 和 $w_{ii}=0$ 对所有 $i$、$j$、$1\leq i,\ j\leq |\mathcal{I}|$，

$||\cdot||_F$ 是矩阵弗罗贝尼乌斯范数：

$$||\boldsymbol{W}||_F=\sqrt{\sum_{i=1}^{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{I}|}w_{ij}^2},$$

$||\cdot||_1$ 是逐元素 $l_1$-范数：

$$||\boldsymbol{W}||_1=\sum_{i=1}^{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{I}|}|w_{ij}|$$

而 $\beta$ 和 $\lambda$ 是正则化参数。

我们可以看到，$\boldsymbol{W}$ 的每一列都可以独立学习，从而提高了该模型的可扩展性（Ning 和 Karypis，2011年）。

*2.3.3.* 基于图的方法

**RP3beta** （Christoffel 等人，2015年）是一种基于图的协同过滤方法。与 Dacrema 等人（2021年）类似，我们将把该模型介绍为 P3alpha 模型（Cooper 等人，2014）的扩展。我们考虑一个二分无向图 $G=(V,E)$，其中每个顶点 $v\in V$ 表示一个用户或物品，每条边 $\{v_u,v_i\}\in E$ 表示给定用户 $u$ 与给定物品 $i$之间的交互。令 $\boldsymbol{A}$ 为图的邻接矩阵，$\boldsymbol{D}$ 为度矩阵 $G$。那么，P3alpha 的得分存储在矩阵中

$$((\boldsymbol{D}^{-1}\boldsymbol{A})^{\circ\alpha})^3,$$

其中 $\circ$ 表示矩阵的逐元素乘方（哈达玛幂）。更准确地说，得分 $\hat{r}_{ui}$ 存储在表示用户 $u$ 的行与表示物品 $i$的列的交叉处。我们可以看到，该得分是图 $G$中连接用户 $u$ 和物品 $i$ 的长度为 3 的路径所对应得分之和。

在 RP3beta 中，分数通过 P3alpha 计算，并除以物品流行度的 $\beta$ 次方，以减轻流行度偏差。

我们选择该模型，是因为它简单、可扩展且性能经过验证（Dacrema 等，2021；Anelli 等，2021），尤其考虑到我们的数据集具有较高的稀疏性，这显著降低了模型训练时的内存利用率。这样，我们便可以直接计算并在内存中存储物品–物品相似度矩阵，而不必采用原论文提出的随机游走近似。

*2.3.4. Prod2Vec*

Prod2Vec（Grbovic 等，2015）（或 Item2vec（Barkan 和 Koenigstein，2016））模型基于广泛应用于自然语言处理的 Word2Vec 模型（Mikolov 等，2013）。该模型将物品序列视为“句子”，将序列中的物品视为“词”，以此学习物品的向量表示。选择这种方法是因为它具有可扩展性，且与用于比较的方法截然不同。此外，OLX 其他类别中使用的当前最先进的混合深度神经网络推荐系统也基于这种方法。

Prod2Vec 使用跳字模型，通过最大化物品序列集合 S 上的目标函数，其定义如下：

$$L = \sum_{s \in \mathcal{S}} \sum_{p_i \in s} \sum_{\substack{-c \le j \le c \\ j \neq 0}} \log \mathbb{P}(p_{i+j} | p_i),$$

其中，$c$ 是物品序列的上下文长度，同一序列中的物品按任意顺序排列。给定当前物品 $p_i$，观察到相邻物品 $p_{i+j}$ 的概率 $\mathbb{P}(p_{i+j}|p_i)$ 使用 Softmax 函数定义：

$$\mathbb{P}(p_{i+j} | p_i) = \frac{\exp(\boldsymbol{v}_{p_i}^{\top} \boldsymbol{v}'_{p_{i+j}})}{\sum_{p=1}^{P} \exp(\boldsymbol{v}_{p_i}^{\top} \boldsymbol{v}'_{p})},$$

其中，$\boldsymbol{v}_p$ 和 $\boldsymbol{v}'_p$ 分别是物品 $p$的输入向量表示和输出向量表示，$P$ 是词汇表中不同物品的数量。Prod2Vec 对物品序列的上下文进行建模，上下文相似的物品（即具有相似相邻交互的物品）具有相似的向量表示。

为了生成推荐结果，我们将用户表示为其交互物品的输入向量表示的平均值。然后，我们搜索输入表示与用户表示的余弦相似度最高的物品（潜在空间中的 $k$近邻）。

*2.4.* 研究空白

本文旨在针对分类广告中的招聘领域这一特定研究空白，对不同方法进行基准测试。据我们所知，所涉及的推荐方法从未在招聘领域的大规模在线环境中进行过测试或基准测试。在本研究中，我们不仅使用我们发布的数据集，在实验室环境中比较这些方法，还在生产环境中开展了在线 A/B 测试。

## 3. 数据集

本研究提供了一个商业数据集，其中包含 2020 年两周内 3,295,942 名用户在 OLX Jobs 上进行的 65,502,201 次事件；这些用户与 185,395 条招聘广告进行了互动。我们已将该数据集发布在 Kaggle 上。[1]

| User | Item | 事件 | 时间戳 |
|---|---|---|---|
| 1745587 | 168661 | 点击 | 1582216025 |
| 843008 | 62838 | 点击 | 1582485868 |
| 14285 | 30469 | 书签 | 1582247367 |
| 1142944 | 80122 | 点击 | 1581805847 |
| 2835659 | 23728 | 聊天点击 | 1582397836 |

来源：本研究

表1：数据集中的示例行。

数据集的每一行表示某个用户在特定时间对某则广告执行的事件（表1）。其中包括以下涉及用户行为的事件：

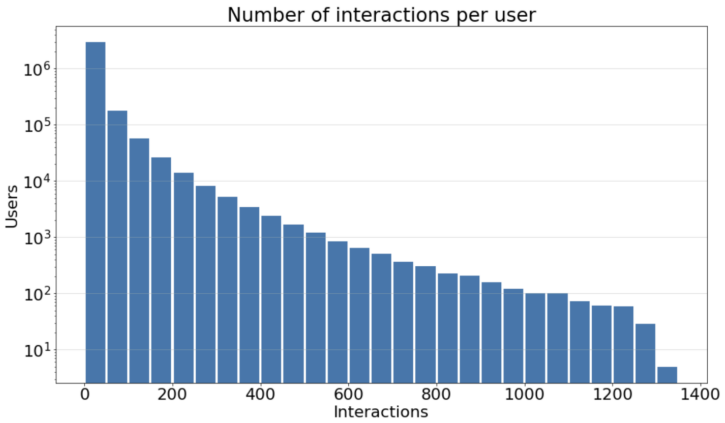
- 点击：用户访问了物品详情页；
- 书签：用户将物品添加到了书签；
- 聊天点击：用户打开聊天窗口，联系物品所有者；
- 联系电话点击1：用户显示了物品关联的电话号码；
- 联系电话点击2：用户点击拨打物品所有者的电话；
- 联系电话点击3：用户点击向物品所有者发送短信；
- 联系合作伙伴点击：用户点击访问物品所有者的外部页面；
- 联系聊天：用户向物品所有者发送了一条消息。

数据稀疏度（用户-物品矩阵中缺失条目的百分比）为99.9893%，非常低，但对于分类广告而言很常见（作为比较，MovieLens 20M数据集的数据稀疏度为99.47%，密集程度几乎是其50倍）。每位用户的平均交互次数为19.874（标准差为47.762，最大值为1310）。每件物品的平均交互次数为353.312（标准差为624.773，最大值为15,480）。数据集的统计信息见图1和图2以及表2。请注意，我们使用了对数刻度，因为这些分布呈现出明显的右偏。

需要指出的是，在准备该数据集时，保护广告和用户的机密性是首要任务。为保护隐私采取的措施包括：

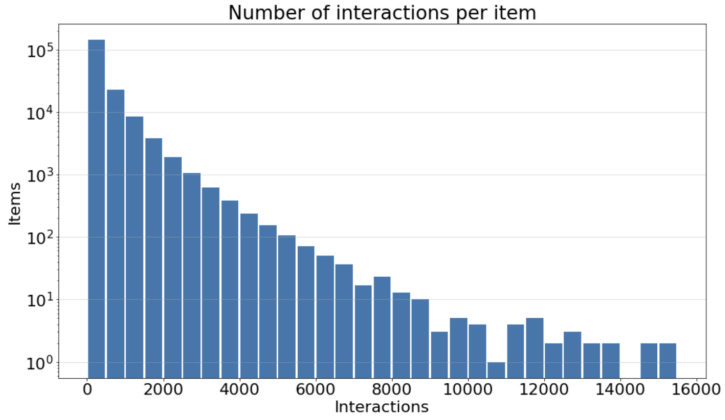
- 原始用户和物品标识符已替换为唯一的随机整数；
- 所有时间戳都加上了某个未公开的常整数；

[1] https://www。kaggle。com/olxdatascience/olx-jobs-interactions，访问日期：2022-08-06




| Quantile | Value |
|---|---|
| 0.01 | 1 |
| 0.10 | 1 |
| 0.25 | 2 |
| 0.50 | 6 |
| 0.75 | 18 |
| 0.90 | 46 |
| 0.99 | 221 |

图1：每位用户的交互次数。来源：本研究




| Quantile | Value |
|---|---|
| 0.01 | 1 |
| 0.10 | 1 |
| 0.25 | 3 |
| 0.50 | 137 |
| 0.75 | 428 |
| 0.90 | 950 |
| 0.99 | 2944 |

图2：每件物品的交互次数。来源：本研究

- 部分交互被过滤掉；
- 添加了一些额外的人工交互 .

## 4. 实验设置

### *4.1.* 实现

我们基于先前的开发成果实现了上述方法。表3列出了我们的实现与先前测试方法的差异详情。为确保结果的可复现性，这些方法的实现代码可在 GitHub 上获取。[2]

### *4.2.* 实验室研究和在线测试目标

我们的最终业务目标是增加通过 OLX Jobs 申请工作的用户数量。为此，我们尝试解决一个排序问题；更准确地说，我们尝试推荐最有可能获得特定用户交互的 10 个物品。寻找一种

[2] https://github.com/rob-kwiec/olx-jobs-recommendations，访问日期：2022-08-06

| 操作 | 频率 (%) |
|---|---|
| 点击 | 89.794 |
| 联系电话点击1_ _ | 2.628 |
| 书签 | 2.511 |
| 聊天点击 | 2.136 |
| 联系聊天 | 1.448 |
| 联系合作伙伴点击 | 0.701 |
| 联系电话点击2_ _ | 0.679 |
| 联系电话点击3_ _ | 0.103 |

来源：本研究

表2：事件分布。

与招聘领域申请率相关性最高的离线指标这一问题由 Mogenet 等（2019）进行了探讨。由于我们的目标非常相似，因此决定采用他们的研究结果，优化精确率@10（前 10 个推荐项目的精确率（Tamm等人，2021））。不过，我们也报告并讨论了其他几项准确率、多样性和效率指标。

我们关注通过与招聘广告交互来积极求职的用户。因此，我们不考虑没有任何交互的用户（冷启动用户）。

### *4.3.* 数据集划分

根据研究方法的要求，第3节所述的数据集被划分为训练集与验证集以及测试集。测试集包含时间上最新的20%交互。这意味着，在14天的数据中，约有3天的数据被纳入测试集（见图3）。

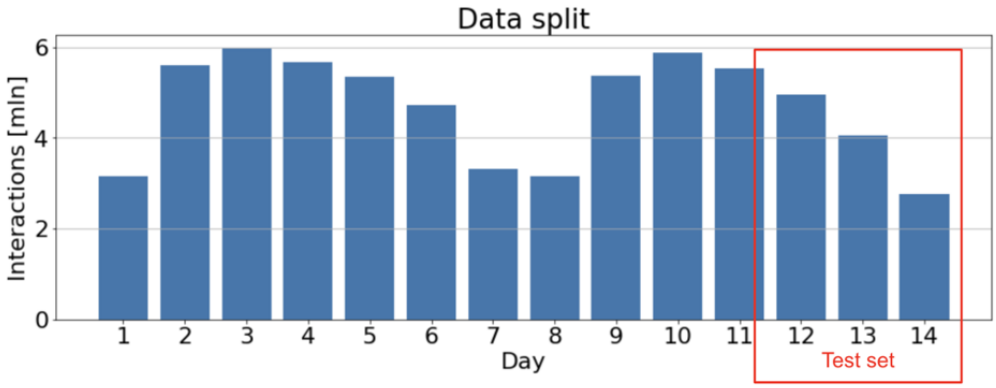



图3：数据集准备（根据天数划分为训练集与验证集以及测试集）。来源：本研究

随后，我们在两个数据集中都将交互次数限定为用户与物品之间的唯一交互。如果用户与某件物品交互过多次，则只计入首次交互，并将该次交互的时间戳与该物品关联。我们也不区分交互类型。此外，还进行了两项修改：仅将出现在训练集与验证数据集中的用户纳入测试集（因为

| 方法 | 类别 方法 | 来源 | 与.....的区别 方法 方法 |
|---|---|---|---|
| LightFM | 矩阵 fac- 分解 | 实现 基于: (Kula, 2022年)。支持 论文：(Kula，2015年)。 | None |
| ALS | 矩阵 fac- 分解 | 实现 基于: （Frederickson，2022年）。Sup- 移植 论文: （Tak´acs 等人，2011年；Hu 等人，2008年）。 | None |
| SLIM | 基于邻域的 基于 | 启发我们实现的来源 实现：（Levy 和 Grisel ， 2021年； Quadrana, 2021年）。支持论文: （Ning 和 Karypis，2011年）。 | None |
| RP3beta | 图 驱动 | 启发我们实现的来源: 实现: （Bernardis, 2022年）。 支持 pa- 人员: （Paudel等, 2016年; Dacrema 等，2021）。 | 进行了直接计算 基于稀疏矩阵，而非 随机游走近似。 |
| Prod2Vec Word2Vec | | 实现 基于 on: (˘Reh ˇu˘rek 和 Sojka，2010年)。 支持论文 : (Gr- bovic 等, 2015年; Barkan 和 Koenigstein，2016年）。 | 交互序列，或—— 按时间戳排序。将用户表示为交 互物品表示的平均值。 交互物品的表示。 也尝试了 CBOW （连续词袋）。 |

来源：本研究

表3：为满足研究需要而重新实现的方法。

如果没有关于其他用户的额外信息，我们就无法向他们提供个性化推荐），并过滤掉训练集中已有的所有用户-物品对（以避免推荐用户已经看过的物品）。最终，训练集约有 3800 万行，测试集约有 600 万行。

### *4.4.* 模型调优

为节省成本和时间，在优化参数时，我们将范围限制为训练集与验证集中 20% 的用户和 20 % 的物品。随后，我们按照上一节所述的规则，将受限后的数据集拆分为训练集和验证数据集。对于每个模型，我们使用 ScikitLearn 实现的高斯过程进行贝叶斯优化，并选择了 100 次迭代。[3] 配置文件定义了超参数的搜索空间，可在我们的代码仓库中找到。[4] 例如，对于 RP3beta，我们将搜索范围限制为

[3] `https://scikit-optimize.github.io/stable/modules/generated/skopt.gp minimize.html` ，访问时间：2022年 08 - - 06 [4] `https://github.com/rob-kwiec/olx-jobs-recommendations/blob/main/src/tuning/config.py`，访问时间：2022-08-06

在区间 [0,2]内搜索 $\alpha$ 和 $\beta$ 。

优化过程中为30,000名用户生成了推荐结果，如上文所述，使用的优化评分指标为精确率@10。图4展示了精确率随参数变化的情况。可以看出，ALS在此过程中并未得到充分优化，但对于其他方法而言，这种优化对提高精确率至关重要。所有模型的超参数均列于表4中。

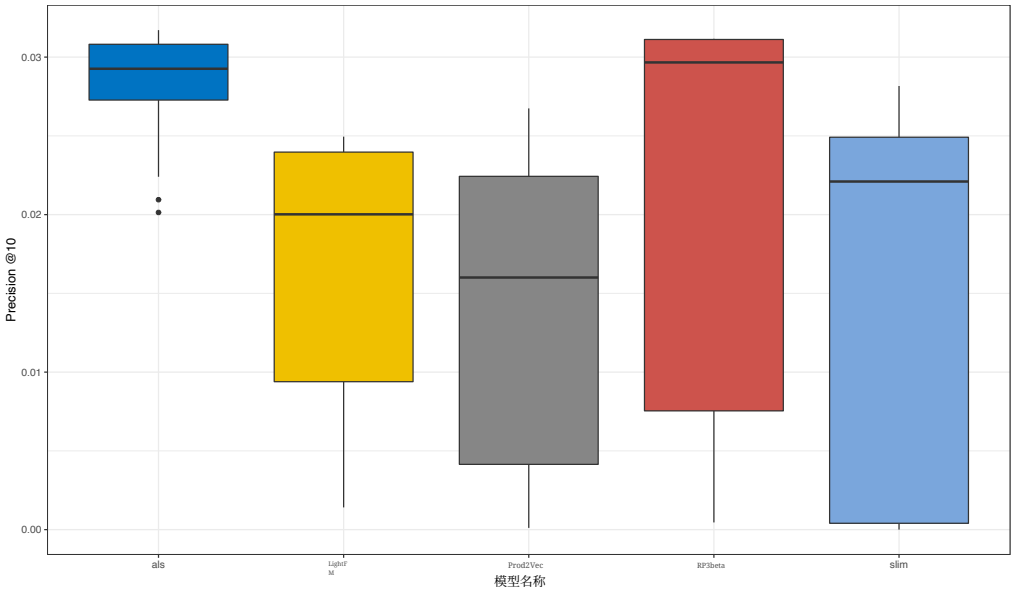



图4：各模型的精确率与参数的关系。来源：本研究

| 模型 | 模型超参数 |
|---|---|
| **als** | {’因素’: 357, ’正则化’: 0.001, ’迭代次数’: 20, ’事件权重乘数’: 63} |
| **LightFM** | {’无组件’: 512, ’k’: 3, ’n’: 20, ’学习率调度’: ’Adadelta’, ’损失函数’: ’WARP’, ’最大采样数’: 61, ’训练轮数’: 11} |
| **Prod2Vec** | {’向量大小’: 168, ’alpha’: 0.028728, ’窗口’: 20, ’最小计数’: 16, ’采样’: 0.002690026, ’最小_alpha’: 0.0, ’sg’: 1, ’hs’: 1, ’negative’: 200, ’ns_exponent’: -0.16447846705441527, ’cbow_mean’: 0, ’训练轮数’: 22} |
| **RP3beta** | {’alpha’: 0.61447198, ’beta’: 0.1443548} |
| **slim** | {’alpha’: 0.00181289, ’L1_比率’: 0.0, ’迭代次数’: 3} |

来源：本研究

表4：模型超参数

## 5. 实验室评估

本节介绍所讨论的方法在构建的数据集上的实验室评估。我们使用了上一节中找到的性能最佳的超参数。每个模型都在完整的训练集与验证数据集上进行训练，并为测试集中的全部619,389 位用户生成了推荐结果。为了更好地呈现各项离线指标的数值，我们将

所述方法与简单的非个性化基线进行了比较：最受欢迎的产品推荐方法和随机推荐。

### *5.1.* 准确率

#### *5.1.1.* 准确率评估

本节所呈现的所有指标均基于前 10 条推荐结果计算，例如精确率@10、召回率@10。指标值越高，表示推荐结果质量越高。由于指标在很大程度上取决于数据集的分布（例如较高的稀疏性）以及训练/测试集划分策略，因此不应将这些数值与其他数据集上取得的结果直接比较。表5展示了评估结果。由于评估的用户数量庞大，大多数观察到的差异都具有统计显著性。我们将在第 5.1.2 节进一步讨论结果的统计显著性。

| 模型 | RP3beta | SLIM | ALS | Prod2Vec | LightFM | 最受欢迎 | 随机 |
|---|---|---|---|---|---|---|---|
| 精确率 | **0.0484** | 0.0472 | 0.0434 | 0.0368 | 0.0359 | 0.0012 | 0.00006 |
| 召回率 | **0.0783** | 0.0736 | 0.0657 | 0.0580 | 0.0564 | 0.0012 | 0.00005 |
| ndcg | **0.0759** | 0.0721 | 0.0657 | 0.0567 | 0.0545 | 0.0016 | 0.00007 |
| mAP | **0.0393** | 0.0365 | 0.0329 | 0.0282 | 0.0264 | 0.0006 | 0.00002 |
| MRR | **0.1365** | 0.1314 | 0.1230 | 0.1065 | 0.1034 | 0.0038 | 0.00019 |
| LAUC | **0.5391** | 0.5368 | 0.5328 | 0.5289 | 0.5281 | 0.5006 | 0.49999 |
| HR | **0.3131** | 0.3066 | 0.2878 | 0.2537 | 0.2547 | 0.0112 | 0.00059 |

来源：本研究

表5：准确率：评估结果。所有列出的指标均由 Tamm 等（2021）描述。对于平均准确率均值（mAP）指标，我们使用了带有 $x = \min(k,\ r)$ 的变体。

可以看出，我们的所有方法都大幅优于随机推荐和最热门物品推荐。最热门方法的性能较低，源于分类广告的特性，尤其是在招聘领域，用户通常只关注特定的位置和类别。在这种情况下，构建更复杂的个性化推荐系统是合理的。

结果表明，在准确率指标方面，RP3beta 优于其他方法。我们认为，这与数据集的高稀疏性有关。RP3Beta 以确定性方式计算推荐结果（利用用户–物品二部图中长度为 3 的路径）。其他所有方法都使用机器学习技术来寻找用户表示（ALS、LightFM）、物品表示（ALS、LightFM、Prod2Vec）或直接的物品间相似度（SLIM）；当用户或物品的交互次数较少时，这可能会带来挑战。

#### *5.1.2.* 精确率*@10*的统计比较

为了识别不同方法之间的差异，我们针对主要指标精确率@10进行了详细的统计比较。首先，我们检验所有方法表现相同且观察到的差异仅仅是随机因素造成的零假设（总体检验）。Friedman

检验（Friedman，1937年，1940年）及其 Iman 和 Davenport（1980年）扩展可能是最受欢迎的总体检验；在比较五种以上不同算法时，它通常是一个不错的选择（Garc´ıa 和 Herrera，2008年；Garc´ıa 等，2010年）。令 $R_{ij}$ 表示第 $N$ 个样本中第 $K$种方法的第 $j$个秩，并且

$$R_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}.$$

该检验比较各方法的平均秩，基于以下统计量

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2},$$

其中

$$\chi_F^2 = \frac{12N}{K(K+1)}\sum_{i=1}^{K} R_i^2 - 3N(K+1)$$

是 Friedman 统计量，其服从自由度为 $K-1$ 和 $(K-1)(N-1)$ 的 $F$ 分布。该检验得到的 $p$ 值等于 0。所得 $p$ 值表明，我们可以放心地拒绝所有算法性能相同这一零假设。因此，我们可以继续进行事后检验，以检测所有方法之间显著的成对差异。Dem˘sar（2006年）建议采用 Nemenyi 检验（Nemenyi，1963年），该检验对所有算法进行两两比较。对于显著性水平 $\alpha$，该检验会确定临界差异（CD）。如果两个算法的平均排名之差大于

$$\mathrm{CD} = q_\alpha\sqrt{\frac{K(K+1)}{6N}}$$

则拒绝算法性能相同这一零假设（ $q_\alpha$ 基于学生化极差统计量除以√2）。Dem˘sar（2006年）建议使用临界差异图（CD plot）直观检查差异。在图中，没有被线段连接的算法可视为彼此不同。

在本案例中，当显著性水平为 $\alpha = 0.05$ 时，任何两个算法的平均排名差异超过 0.0114，均将被视为不相等（图 5）。

### *5.1.3.* 不同交互次数用户的准确率

关于用户，我们唯一掌握的信息就是他们的交互。在图6中，我们考察了用户交互过的物品数量对主要指标——精确率@10——的影响。我们根据交互次数将用户分成了10组，每组人数大致相同。可以看到，按精确率@10排序的模型顺序不受用户组别影响；唯一的例外是，对于交互次数至少为22次的用户，SLIM的表现优于RP3Beta。这种差异对于每一组用户而言都具有统计显著性（详见第5.1.4节）。RP3Beta 模型可视为基于物品的协同过滤方法的一种特殊情况，该方法使用物品间确定性的相似度度量（Dacrema 等人，2019年）。另一方面，SLIM 使用机器学习方法来学习相似度。我们认为，这种学习得到的相似度分数偏差较小，但方差较大；由于对更多分数取平均，因此对于交互次数较多的用户，模型性能更好。

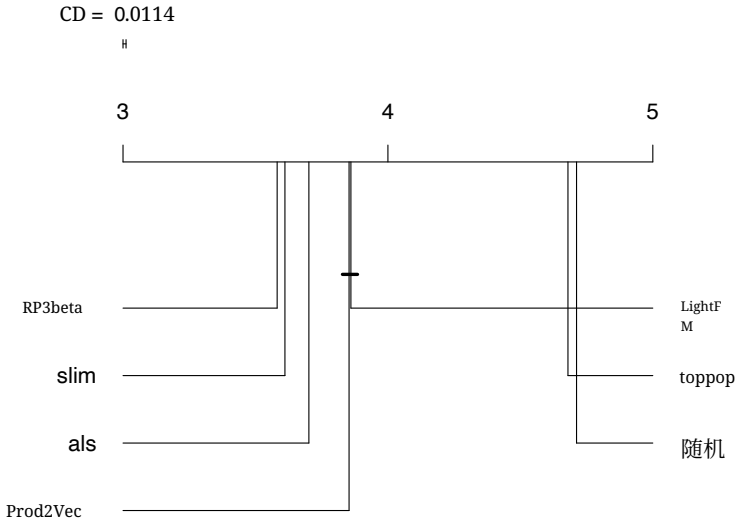



图 5：临界差异图（精确率）。来源：本研究

*5.1.4.* 不同交互次数用户的准确率统计评估

本小节讨论 RP3Beta 和 SLIM 模型之间的差异如何随用户交互数量而变化，如图6所示。

为了在多个数据集上对两种方法进行统计比较，Demˇsar（2006年）建议使用 Wilcoxon 符号秩检验（Wilcoxon，1945年）。Wilcoxon 符号秩检验是配对 $t$检验的一种非参数替代方法。该检验对每个数据集中两种算法的性能差异进行排序（忽略差异的符号），并比较正差异与负差异对应的秩。

从表6可以看出，对于每组用户，RP3Beta 和 SLIM 模型之间的差异在统计上均显著。

| **bin** | **p值** |
|---|---|
| [1.0, 3.0) | 0 |
| [3.0, 5.0) | 0 |
| [5.0, 8.0) | 0 |
| [8.0, 11.0) | 0 |
| [11.0, 16.0) | 0 |
| [16.0, 22.0) | 1.25e-6 |
| [22.0, 31.0) | 9.27e-7 |
| [31.0, 45.0) | 0 |
| [45.0, 74.0) | 0 |
| [74.0, 852.0) | 0 |

来源：本研究

表6：根据用户交互过的物品数量，对 RP3Beta 与 SLIM 模型进行 Wilcoxon符号秩检验的结果。

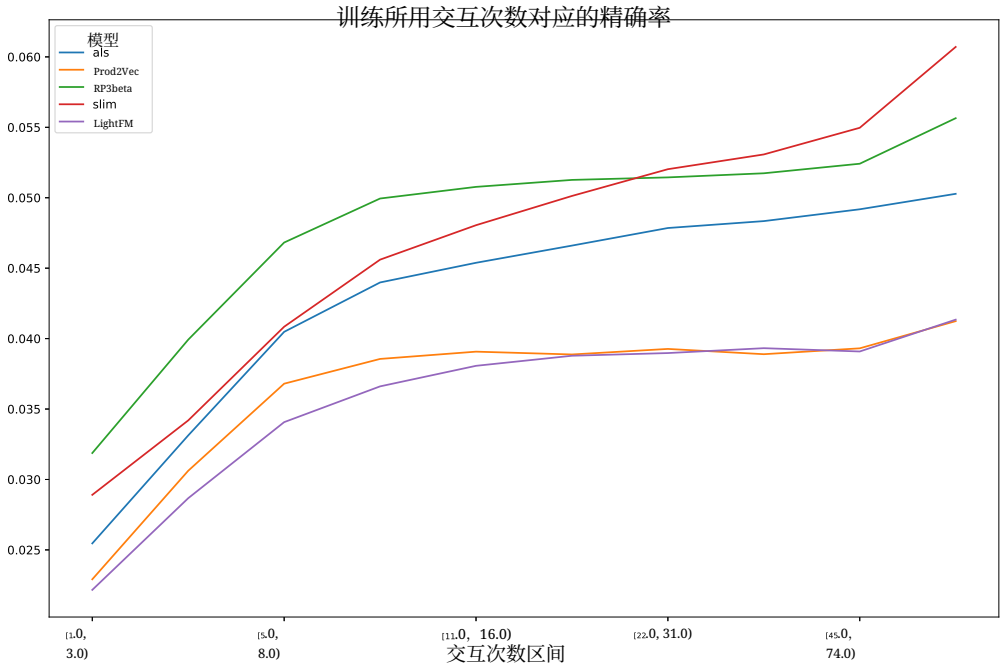



图6：各模型的精确率@10，取决于用户交互过的物品数量。建议彩色查看。来源：本研究

### *5.2.* 多样性

| 模型 | RP3beta | SLIM | ALS | Prod2Vec | LightFM | 最受欢迎 | 随机 |
|---|---|---|---|---|---|---|---|
| 测试覆盖率 | 0.5725 | 0.5171 | 0.3038 | **0.7400** | 0.7031 | 0.0002 | 0.9778 |
| 香农 | 9.5271 | 9.6728 | 9.6270 | **10.4031** | 10.1385 | 2.3296 | 11.7267 |
| Gini | 0.9083 | 0.9029 | 0.9120 | **0.7956** | 0.8397 | 0.9999 | 0.1159 |

来源：本研究

表7：多样性：评估结果。

在分类广告领域，尤其是招聘领域，通常只需要一位用户就能完成针对某则广告的交易。在这种情况下，推荐结果的多样性比其他领域（如音乐、电影，甚至除分类广告以外的其他电子商务领域）更为重要。为评估这一方面，我们采用三种度量指标：测试覆盖率、香农熵（Shani 和 Gunawardana，2011）以及基尼指数（Shani 和 Gunawardana，2011）。测试覆盖率指至少向一位用户推荐过的测试物品所占的比例。计算香农熵和基尼指数时，我们忽略测试集以外的物品。测试覆盖率和香农熵的值越大，表示多样性越高；而基尼指数的值越大，则表示多样性越低。从表7可以看出，排除随机推荐（按定义其多样性应当最高）后，Prod2Vec 和 LightFM 提供的多样性最高。准确度最高的模型 RP3Beta、SLIM 和 ALS 的多样性相近，但 ALS 的测试覆盖率较低。

### *5.3.* 方法重叠度

我们的研究聚焦于值得通过真实用户进行测试的方法。因此，我们研究了不同已实现方法针对用户-物品对所提供推荐结果之间的重叠情况（重叠系数（Vijaymeena 和 Kavitha，2016年））。我们预计，重叠度较低的方法之间，推荐结果的差异会更加显著。

该实验的结果见表 8。RP3beta 和 SLIM 提供的推荐结果相似，这是因为两者都使用稀疏的物品–物品相似度矩阵，尽管矩阵的计算方式不同。与其他方法相比，ALS 的相似度处于中等水平。从重叠度来看，差异最大的也是最多样化的模型，即 Prod2Vec 和 LightFM。ALS 与 LightFM 的低重叠度凸显了损失函数的选择对矩阵分解方法的重要性。

| 模型 | RP3beta | SLIM | ALS | Prod2Vec | LightFM |
|---|---|---|---|---|---|
| RP3beta | 100% | **73%** | 53% | 37% | 38% |
| SLIM | **73%** | 100% | 50% | 35% | 35% |
| ALS | 53% | 50% | 100% | 38% | 37% |
| Prod2Vec | 37% | 35% | 38% | 100% | 28% |
| LightFM | 38% | 35% | 37% | 28% | 100% |

来源：本研究

表 8：模型重叠。

### *5.4.* 效率评估

实验室评估的最后一步是评估效率。为此，我们采用了以下配置：AWS Sagemaker ml.m5.4xlarge 实例，配备 64 GB 内存和 16 个虚拟 CPU，搭载 Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz，运行 Amazon Linux AMI 2018.03。

在本实验中，我们从三个方面评估了所有方法：数据预处理、模型拟合和物品推荐。执行时间按模型计算所需的总时间（小时）计算。内存峰值以 GB 为单位，指执行给定功能期间的最大内存使用量。本次评估使用 tracemalloc 库完成。[5] 效率评估结果如图7所示。我们观察到，在执行时间方面，RP3beta、ALS 和 LightFM 优于 SLIM 和 Prod2Vec，而 Prod2Vec 的内存需求最少。

### *5.5.* 总结

实验室研究使我们能够选出适合在 OLX 用户中进行在线测试的模型。首先，我们选择了 RP3Beta 模型进行在线测试，因为它在

[5] https://docs。python。org/3/library/tracemalloc。html，访问日期：2022-08-06

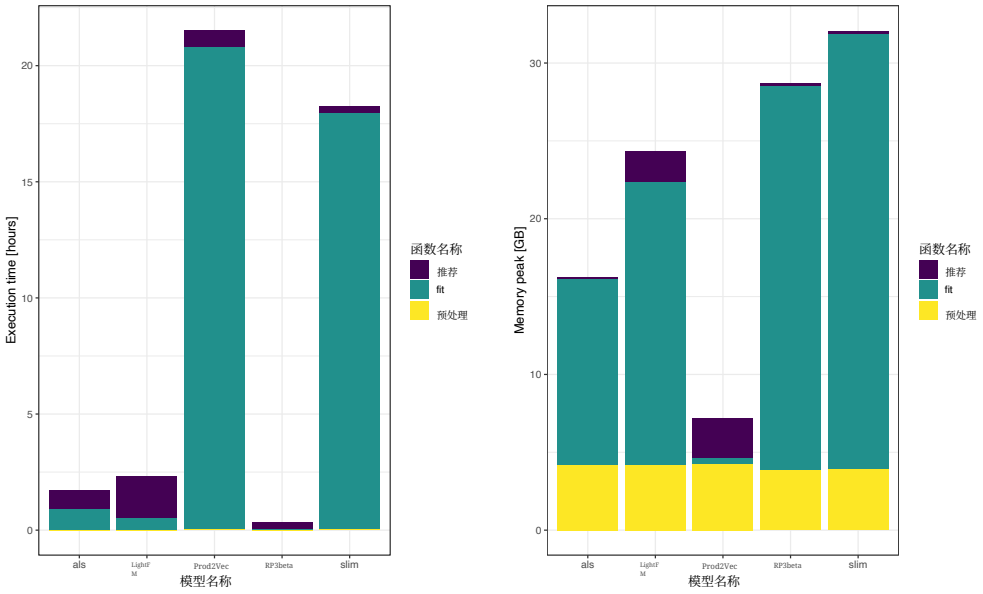



图7：执行时间和内存利用率。彩色查看效果最佳。来源：本研究

y

准确率和执行时间方面表现最佳。其内存消耗相对较高，但仍足够低，因此该模型可以在生产环境中进行测试。研究这种方法多样性相对较低所带来的影响，或许会是一个有趣的未来研究方向。

在精确率@10方面，表现第二好的模型是 SLIM。不过，由于效率较低，且与已选定的 RP3Beta 模型重合度较高，我们没有选择这种方法进行在线评估。提高 SLIM 实现的效率或许会改变这一决定。

在其他模型中，ALS 的精确率@10最高。此外，ALS 已经实现，并用于 OLX 的职位推荐；因此，我们决定将其纳入在线比较。

尽管 Prod2Vec 和 LightFM 生成的推荐结果最多样化，但由于它们的精确率@10显著较差，而且 Prod2Vec 的效率也较低，我们决定不在线测试它们。

## 6. 在线评估

为了评估所选方法的有效性，我们在 OLX Jobs 用户中进行了两次在线 A/B 测试。实验的第一步旨在回答以下问题：引入推荐方法是否会增加申请职位的用户数量？我们重点关注近期在平台上浏览过招聘广告的用户。在这项测试中，我们将用户分为两组：一组为不接收任何推荐的对照组，另一组则接收使用 ALS 模型生成的推荐结果。

该测试于2021年3月进行了25天。我们发送了包含10条推荐招聘广告的电子邮件；对于安装了OLX 应用程序的用户，还发送了一条包含一条推荐

招聘广告的推送通知。随后，我们观察用户是否在接下来的48小时内申请了任何职位（转化用户）。结果见表9。我们使用卡方检验检查了 ALS 相较于对照组的优势是否具有统计显著性，发现$p$-值为0。这项实验证实了我们的假设：推荐结果会影响用户活跃度，因为收到通知后，转化用户的比例显著提高了 (16.83% – 15.98%)/15.98% $\approx 5.3\%$。观察到的增长意味着每天有数百名额外用户通过 OLX Jobs 申请职位。

| 变体 | 用户 | % 转化用户 |
|---|---|---|
| 控制 | 129,308 | 15.98% |
| ALS | 1,170,262 | 16.83% |

来源：本研究

表9：有无推荐时的活动情况对比。

在第二次实验中，我们测试了实验室测试中表现最佳的模型是否优于 ALS。我们将用户分成三组，分别接收来自不同推荐系统的推荐结果：ALS、RP3beta，以及一种混合变体，其中一半推荐结果由 ALS 生成，另一半由 RP3beta 生成。第二次实验于2021年三月和四月进行，历时28天。结果见表10。此处转化用户的百分比是相对于所有用户计算的；然而，更换推荐系统并未影响未打开消息的用户。我们决定将这些用户筛除，因为他们的行为给数据带来了不必要的噪声。表11展示了对打开消息的用户产生的影响。RP3beta 相较于 ALS 的优势具有统计显著性（$p$-值 $\approx 10^{-6}$）。RP3beta 和 ALS+RP3beta 这两种变体之间的差异不具有统计显著性（ $p$-值为 0.19）。

| 变体 | 用户 | % 转化用户 |
|---|---|---|
| ALS | 343,892 | 15.25% |
| RP3beta | 345,273 | 15.40% |
| ALS+RP3beta | 343,896 | 15.30% |

来源：本研究

表10：不同推荐方法的比较。

| 变体 | 用户 | % 转化用户 |
|---|---|---|
| ALS | 44,775 | 19.66% |
| RP3beta | 46,097 | 20.94% |
| ALS+RP3beta | 45,469 | 20.59% |

来源：本研究

表11：打开消息的用户所使用的不同推荐方法比较。

在打开消息的用户中，RP3beta 带来的转化用户比 ALS 多 1.28 个百分点。由于消息打开率为 13.2%，这相当于

所有目标用户中增加了 $1.28 \cdot 0.132 \approx 0.17$ 个百分点。根据第一项实验的结果，ALS 相较于对照组带来了 0.85 个百分点的提升。这意味着，用 RP3beta 替代 ALS，可使我们对所有目标用户的影响提升约 $0.17/0.85 = 20\%$。

### *6.1.* 讨论

我们已证明，使用 ALS 模型发送职位推荐，可使响应招聘广告的用户数量增加 5% 以上。此外，RP3Beta 模型的表现显著优于 ALS。

此外，我们还开展了另一项在线 A/B 测试，将 RP3Beta 模型与一个变体进行比较：该变体将一半的 RP3Beta 推荐结果替换为 OLX 所采用的深度学习混合推荐系统生成的推荐结果。[6] 我们决定不在本文中描述这项测试，因为本文聚焦于协同过滤方法。此外，我们希望确保所有结果都能基于已发布数据集完全复现。遗憾的是，我们无法公开内部模型使用的全部特征。这里只想说明，尽管第二个模型利用了有关招聘广告的额外信息，我们并未观察到其表现优于 RP3Beta。因此，我们确认，经过恰当确定和调优的简单模型可能胜过先进得多的深度模型。这与 Dacrema 等人（2019年）和 Anelli 等人（2021年）的结果一致；他们将包括 RP3Beta 在内的若干非神经网络方法与更先进的神经网络模型进行了比较。

RP3Beta 模型的高效率使 OLX Jobs 能够每天以极低的成本（在中央处理器上）从头开始训练多次。目前，它是 OLX Jobs 最先进的协同过滤模型。我们正在努力改进该模型，同时避免效率大幅下降；在现实世界应用中，这是一个非常重要的限制。我们观察到，其他公司也倾向于采用更简单的模型；例如，Netflix 决定不实施 Netflix 奖竞赛的获胜算法（Bennett 等人，2007年），因为所需的工程投入过于巨大。[7]

在这项工作中，我们针对招聘领域中缺乏对经典协同过滤方法进行比较这一研究空白，通过使用真实世界数据集开展全面的实验室研究，并对数百万用户进行在线评估，填补了这一空白。此外，我们还公开了用于离线评估的源代码和数据集。我们鼓励其他研究人员开发新的推荐方法，希望这能帮助用户更快地找到理想的工作。

[6] https://towards...-neural-item-embeddings-to-enhance-recommendations-1fd948a6f293，访问日期：2022-08-06

[7] https://towardsdatascience。com/deep-dive-into-netflixs-recommender-system-341806ae3b48，访问日期：2022-08-06

## 7. 总结

本文以 OLX Jobs 为例，探讨了在线分类广告中的职位推荐问题。我们对代表不同推荐方法类别的多种方法进行了广泛评估。

在实验室研究中，我们证明 RP3beta、SLIM 和 ALS 的表现显著优于 LightFM 和 Prod2Vec。我们还发现，Prod2Vec 的多样性最高，这在提升推荐结果的新颖性时可能很重要。此外，我们还检查了不同模型生成的推荐结果之间的相似程度。

我们还通过向在线用户发送数百万条消息进行了评估。我们使用 ALS 和 RP3beta 开展了 A/B 测试。测试结果表明，发送这些模型生成的职位推荐，能够增加联系 OLX Jobs 广告发布者的用户数量。值得注意的是，在生产环境中部署 RP3beta 后，我们的成效提升了 20%（以用户的响应为衡量依据）。

从分类广告的角度评估这些方法时，需要注意的是，推荐最受欢迎（但用户尚未看过）的物品的模型并不适用于职位推荐。我们还加入了基于随机推荐的推荐模型作为对照，以展示所研究方法的优势。

这项研究的另一个重要成果是该数据集，它将使人们能够针对分类广告的需求开展更多推荐结果相关研究。

未来与我们研究相关的工作将纳入不同类型的交互，以区分用户对某件物品的偏好（例如，查看广告或对广告作出回应），并考察多次访问同一广告的频率。我们在本研究中介绍的数据集使此类研究成为可能。

**参考文献**

Aggarwal, C.C., 2016年。《推荐系统：教科书》。第1版，施普林格出版公司。

Alotaibi, S., 2012年。职位推荐系统综述。International Journal of the Physical Sciences 7, 5127–5142。doi:10.5897/IJPS12.482。

Anelli, V.W., Bellog´ın, A., Di Noia, T., Pomo, C., 2021年。在《第十五届 ACM 推荐系统会议》中重新审视神经协同过滤与矩阵分解之间的比较，计算机协会，纽约州纽约，美国。第 521–529 页。doi:10.1145/3460231.3475944。

Barkan, O., Koenigstein, N., 2016年。Item2vec：用于协同过滤的神经物品嵌入。2016年 IEEE 第 26 届信号处理机器学习国际研讨会（MLSP），1–6doi:10.1109/MLSP.2016.7738886。

Bennett, J., Lanning, S., 等，2007年。Netflix 奖，载于：KDD Cup 及研讨会论文集，纽约州纽约，美国。第 35 页。

Bernardis, C., 2022年。RP3beta – GitHub 仓库。网址：https://github.com/MaurizioFD/RecSys2019 DeepLearning Evaluation/blob/master/GraphBased/RP3betaRecommender.py。访问日期：2022-07-12。

Boatwright, P., Nunes, J., 2001年。缩减商品组合：一种基于属性的方法。《营销学杂志》– J MARKETING 65, 50–63。doi:10.1509/jmkg.65.3.50.18330。

Chen, R., Hua, Q., Chang, Y.S., Wang, B., Zhang, L., Kong, X., 2018年。基于协同过滤的推荐系统综述：从传统方法到基于社交网络的混合方法。IEEE Access 6, 64301–64320。doi:10.1109/ACCESS.2018.2877208。

Chernev, A., 2003. 多即是少，少即是多：理想点可用性和商品组合在消费者选择中的作用。《消费者研究杂志》30, 170–183. doi:10.1086/ 376808.

Christoffel, F., Paudel, B., Newell, C., Bernstein, A., 2015年. 热门商品与冷门商品：通过随机游走实现准确、多样且可扩展的推荐结果，载于：第9届ACM推荐系统会议论文集，计算机协会，纽约州纽约，美国。第163–170页。URL: https://doi.org/10.1145/2792838.2800180, doi:10.1145/2792838.2800180.

Cooper, C., Lee, S.H., Radzik, T., Siantos, Y., 2014。《推荐系统中的随机游走：精确计算与模拟》，载于：第23届国际万维网大会论文集，计算机协会，纽约，纽约州，美国。第811–816页。doi:10.1145/2567948.2579244。

Dacrema, M.F., Boglio, S., Cremonesi, P., Jannach, D., 2021年。《推荐系统研究中可复现性与进展问题的分析》。ACM信息系统汇刊 39，1–49。doi:10.1145/3434185。

Dacrema, M.F., Cremonesi, P., Jannach, D., 2019年。我们真的取得了很大进展吗？对近期神经推荐方法的令人担忧的分析，载于：第13届ACM推荐系统会议论文集，计算机协会，纽约，纽约州，美国。第101–109页。doi:10.1145/3298689.3347058。

Davis, M., 2021年。什么是在线分类广告？BigEasy Magazine。网址：https://www.bigeasymagazine.com/2021/01/04/what-are-online-classifieds/。

Demˇsar, J., 2006年。对多个数据集上的分类器进行统计比较。《机器学习研究杂志》7，1–30。

Frederickson, B., 2022年。ALS – GitHub 仓库。网址：https://github.com/benfred/implicit/blob/master/implicit/als.py。访问日期：2022-07-12。

Friedman, M., 1937年。利用秩避免方差分析中隐含的正态性假设。美国统计学会杂志 32，675–701。

Friedman, M., 1940年。针对 *m* 排名问题的显著性检验方法比较。数学统计年刊 11，86–92。

Garc´ıa, S., Fern´andez, A., Luengo, J., Herrera, F., 2010年。计算智能与数据挖掘实验设计中多重比较的高级非参数检验：检验效能的实验分析。信息科学 180，2044–2064。

Garc´ıa, S., Herrera, F., 2008年。对“多个数据集上的分类器统计比较”的扩展，适用于所有两两比较。机器学习研究杂志 9，2677–2694。

Ge, M., Delgado-Battenfeld, C., Jannach, D., 2010年。超越准确率：通过覆盖率和意外发现评估推荐系统，载于：第四届 ACM 推荐系统会议论文集，计算机协会，纽约州纽约，美国。第 257–260 页。doi:10.1145/1864708.1864761。

Ge, M., Ricci, F., Massimo, D., 2015年. 健康导向的食品推荐系统，载于：第9届ACM推荐系统会议论文集，第333–334页。

Gourville，J., Soman，D., 2005年. 过度选择与品类类型：多样性何时以及为何适得其反。《营销科学》24，382–395。doi:10.1287/mksc.1040.0109。

Grbovic，M.、Radosavljevic，V.、Djuric，N.、Bhamidipati，N.、Savla，J.、Bhagwan，V.、Sharp，D.，2015年。收件箱中的电子商务：大规模产品推荐，载于：第21届ACM SIGKDD国际知识发现与数据挖掘会议论文集，

计算机协会，纽约，纽约州，美国。第1809–1818页。doi:10.1145/ 2783258.2788627。

G´omez-Uribe, C., Hunt, N., 2015年。Netflix 推荐系统。ACM 管理信息系统汇刊 6，1–19。doi:10.1145/2843948。

Hu, Y., Koren, Y., Volinsky, C., 2008年。用于隐式反馈数据集的协同过滤，载于：2008年 IEEE 第八届国际数据挖掘会议论文集，IEEE 计算机学会，美国。第 263–272 页。doi:10.1109/ICDM.2008.22。

Hwangbo, H., Kim, Y.S., Cha, K., 2018年。时尚零售电子商务推荐系统开发。电子商务研究与应用 28，94–101。doi:10.1016/ j.elerap.2018.01.012。

Iman, R.L., Davenport, J.M., 1980。Friedman 统计量临界域的近似。统计学通讯：理论与方法 9，571–595。

Iyengar, S., Lepper, M., 2001年。选择何以令人失去动力：人们会不会对美好事物贪求过多？人格与社会心理学杂志 79, 995–1006。doi:10.1037/0022- 3514.79.6.995。

Javed, U., Shaukat Dar, K., Hameed, I., Iqbal, F., Mahboob Alam, T., Luo, S., 2021年。基于内容和基于上下文的推荐系统综述。国际学习新兴技术杂志 (iJET) 16, 274–306。doi:10.3991/ijet.v16i03.18851。

Jawaheer, G., Szomszor, M., Kostkova, P., 2010年。在线音乐推荐服务中的隐式反馈与显式反馈比较，载于：第1届推荐系统信息异质性与融合国际研讨会论文集，第47–51页。doi:10.1145/1869446.1869453。

Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., Wang, X., 2019年. 一种用于电子商务推荐系统的基于信任的协同过滤算法. 环境智能与人性化计算杂志 10, 3023–3034. doi:10.1007/s12652-018-0928-7.

Johnson, L.M., Johnson, V.E., 1995. 招聘会计师：分类广告揭示的雇主期望. 商业教育杂志 70, 130–134. doi:10.1080/ 08832323.1995.10117738, arXiv:https://doi.org/10.1080/08832323.1995.10117738.

Jones, S., 2009年. 在线分类广告. 皮尤互联网与美国生活项目.

Kaminskas, M., Bridge, D., 2016年. 多样性、意外发现、新颖性与覆盖率：对推荐系统中超越准确率的目标进行的调查与实证分析. ACM交互式智能系统汇刊 7, 1–42. doi:10.1145/2926720.

Koren, Y., Bell, R., Volinsky, C., 2009年。面向推荐系统的矩阵分解技术。《Computer》42, 30–37。doi:10.1109/MC.2009.263.

Kula, M., 2015年。用于用户和物品冷启动推荐的元数据嵌入。arXiv 预印本 arXiv:1507.08439。Kula, M., 2022年。LightFM – GitHub 仓库。网址：https://github.com/lyst/lightfm。访问日期：2022-07-12。Kunaver, M., Pozrl, T., 2017年。推荐系统中的多样性：综述。《知识型系统》123, 154–162。doi:10.1016/j.knosys.2017.02.009. Levi, A., Mokryn, O., Diot, C., Taft, N., 2012。如何在评论的茫茫大海中寻针：基于情境的冷启动酒店推荐系统，载于：第六届 ACM 推荐系统会议论文集，计算机协会，纽约州纽约，美国。第 115–122 页。网址：https://doi.org/10.1145/2365952.2365977，doi:10.1145/2365952.2365977。Levy, M., Grisel, O., 2021年。SLIM – GitHub 仓库。网址：https://github.com/Mendeley/mrec/blob/master/mrec/item similarity/slim.py。访问日期：2022-07-12。Li, S., Karahanna, E., 2015年。B2C 电子商务环境中的在线推荐系统：综述与未来方向。《信息系统协会期刊》16, 72–107。doi:10.17705/1jais.00389.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. 词和短语的分布式表示及其组合性，载于：第 26 届神经信息处理系统国际会议论文集——第 2 卷，Curran Associates Inc.，美国纽约州雷德胡克。第 3111–3119 页。

Mogenet, A., Pham, T.A.N., Kazama, M., Kong, J., 2019. 使用离线评估预测职位推荐系统的在线性能，载于：《第13届ACM推荐系统会议论文集》，计算机协会，纽约，纽约州，美国。第477–480页。doi:10.1145/3298689.3347032。Nemenyi, P., 1963. 无分布假设的多重比较。博士论文。普林斯顿大学。Ning, X., Karypis, G., 2011. 面向 Top-N 推荐系统的稀疏线性方法，载于：《2011年IEEE第11届国际数据挖掘会议论文集》，IEEE计算机学会，美国。第497–506页。URL：https://doi.org/10.1109/ICDM.2011.134，doi:10.1109/ ICDM.2011.134。Paudel, B., Christoffel, F., Newell, C., Bernstein, A., 2016. 面向交互式应用的可更新、准确、多样且可扩展的推荐结果。《ACM交互式智能系统汇刊》7，1–34。doi:10.1145/2955101。Pazzani, M.J., Billsus, D., 2007. 基于内容的推荐系统，载于：《自适应 Web》，施普林格。第325–341页。doi:10.1007/978-3-540-72079-9 10。

Quadrana, M., 2021. SLIM – GitHub 仓库。URL：https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/SLIM_ElasticNet/SLIMElasticNetRecommender.py。访问日期：2022-07-12。

RecSys 挑战赛 '16，2016年。RecSys 挑战赛 '16：推荐系统挑战赛论文集，计算机协会，纽约，纽约州，美国。

RecSys 挑战赛 '17，2017年。RecSys 挑战赛 '17：2017年推荐系统挑战赛论文集，计算机协会，纽约，纽约州，美国。

Řehůřek，R.，Sojka，P.，2010年。大型语料库主题建模软件框架，载于：LREC 2010自然语言处理框架新挑战研讨会论文集，欧洲语言资源协会，瓦莱塔，马耳他。第45–50页。http://is.muni.cz/publication/884893/en。

Rendle，S.，Freudenthaler，C.，Gantner，Z.，Schmidt-Thieme，L.，2012年。贝叶斯个性化排序：基于隐式反馈的贝叶斯个性化排序。《第25届人工智能不确定性会议论文集》，UAI 2009，452–461。

Research and Markets，2021年。《全球在线分类广告市场（2020年至2024年）——洞察与预测，以及新冠肺炎的潜在影响》。环球通讯社。网址：https://www.globenewswire.com/en/news-release/2021/01/05/2153340/28124/en/Global-Online-Classifieds-Market-2020-to-2024-Insights-Forecast-with-Potential-Impact-of-COVID-19.html。访问日期：2022-07-12。

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S., 2007年。《自适应 Web》中的协同过滤推荐系统。施普林格，第291–324页。doi:10.1007/978-3-540-72079-9 9。

Shani, G., Gunawardana, A., 2011年。《推荐系统评估》。施普林格美国，波士顿，马萨诸塞州。第257–297页。网址：https://doi.org/10.1007/978-0-387-85820-3 8，doi:10.1007/978-0-387-85820-3\ 8。

Smith, B., Linden, G., 2017年。亚马逊推荐系统的二十年。《IEEE 互联网计算》21，12–18。doi:10.1109/MIC.2017.72。

Song, Y., Dixon, S., Pearce, M., 2012。《音乐推荐系统综述与未来展望》，载于：第九届计算机音乐建模与检索国际研讨会，Citeseer。第395–410页。

Tak´acs, G., Pil´aszy, I., Tikk, D., 2011年。用于隐式反馈协同过滤的共轭梯度法应用，载于：第五届 ACM 推荐系统会议论文集，计算机协会，纽约，纽约州，美国。第 297–300 页。URL：https://doi.org/10.1145/2043932.2043987，doi：10.1145/2043932.2043987。

Tak´acs, G., Pil´aszy, I., N´emeth, B., Tikk, D., 2009年。面向大型推荐系统的可扩展协同过滤方法。《机器学习研究杂志》10，623–656。doi：10.1145/ 1577069.1577091。

Tamm, Y.M., Damdinov, R., Vasilev, A., 2021年。推荐系统中的质量指标：我们是否以一致的方式计算指标？，载于：第十五届ACM推荐系统会议，计算机协会，纽约，纽约州，美国。第708–713页。URL：https:// doi.org/10.1145/3460231.3478848, doi:10.1145/3460231.3478848。

Twardowski, B., 2016年。利用神经网络对会话感知推荐系统中的上下文信息进行建模，载于：第十届ACM推荐系统会议论文集，计算机协会，纽约，纽约州，美国。第273–276页。doi:10.1145/2959100.2959162。

Vijaymeena, M., Kavitha, K., 2016年。文本挖掘中相似度度量方法综述。《机器学习与应用：国际期刊》3，19–28。doi:10.5121/mlaij.2016.3103。

Wang, X., He, X., Wang, M., Feng, F., Chua, T.S., 2019年. 神经图协同过滤，载于：第42届ACM SIGIR信息检索研究与发展国际会议论文集，计算机协会，纽约，纽约州，美国。第165–174页。网址：https://doi.org/10.1145/3331184.3331267，doi:10.1145/3331184.3331267。

Weston, J., Bengio, S., Usunier, N., 2011年. WSABIE：扩展至大词汇量图像标注，载于：第二十二届人工智能国际联合会议，第2764–2770页。

Weston, J., Yee, H., Weiss, R.J., 2013. 使用k阶统计量损失函数学习推荐结果排序，载于：第7届ACM推荐系统会议论文集，计算机协会，纽约，纽约州，美国。第245–248页。网址：https://doi.org/10.1145/ 2507157.2507210，doi:10.1145/2507157.2507210。

Wilcoxon, F., 1945. 通过排序方法进行个体比较。生物统计学通报 1, 80–83. URL: http://www.jstor.org/stable/3001968.