

可解释推荐系统中的文本处理技术综述

张翼鹏

华东师范大学

摘 要

推荐系统的目标是为用户推荐可能会感兴趣的物品。通过算法推荐达到节省用户时间、提升用户满意度、为公司创造更多的商业价值的目的。为了获取用户对该系统的信任，一种比较好的方法是为用户推荐标的物的同时给用户推荐的理由，向用户解释清楚是基于什么原因推荐的。另外，大数据时代背景下，海量的文本数据背后往往有很多隐藏信息需要我们去进行挖掘，这使得可解释推荐系统和自然语言处理领域建立了密不可分的联系。

本文着眼于可解释推荐系统中常见的文本处理方法，总结并详细介绍了当下主流模型及评估方法，并将一些论文所创新的模型与基础模型进行联系，最后笔者也总结了这些技术的应用场景和自己的思考，希望读者读完本文后对可解释性推荐系统中的文本处理技术有一个大致了解，并对当前的学术前沿有一定的认知和把握。

关键词：智能推荐系统，自然语言处理，LFM,LDA,Word2Vec, 深度学习，知识图谱

1 引言

随着推荐系统在工业界的投入使用，在过去的几年里，研究人员发现，构建可解释的推荐系统可以极大地增加推荐系统的可信度，从而创造更多的经济利益。可解释推荐系统使得整个系统变得更加透明，并有可能允许用户判断系统何时错误（可查看性），增强用户对系统的信心或信任，帮助用户做出更有效、更快速决策，说服用户尝试或购买推荐的物品，改善用户使用的体验感。已经提出了各种技术来生成解释。

鉴于推荐系统在生活中有很好的落地使用，其使用和训练过程中不可避免地会使用大量的非结构化数据，其中的自然语言文本信息是本文聚焦的重点，从可解释推荐系统的角度，其“输入”和“输出”的形式往往都是自然语言——系统训练的“输入”往往来自用户偏好、商品信息等，前者通常来自用户使用某物品后给出的评论信息、用户的文本化画像与偏好；后者给出的解释文本，如推荐标签、文案，呈现方式也往往是文本类型。

笔者通过阅读近几年的前沿文献对文本处理技术在可解释推荐系统中的应用进行了一些总结与思考，后文中将从文本来源、常用模型、模型评估、应用场景及展望等方面进行系统的归纳。

2 文本信息来源

为了追根溯源，我们在本章中理一下文本信息的来源。从基于内容的推荐系统这一大方向入手，这个过程可以简单认为是，对物品进行建模，对用户进行画像，根据以上两个步骤对用户进行物品推荐。从这个角度，笔者将文本信息的来源归为三大类。

2.1 物品画像

物品画像也即物品基本属性，如分类、标题、售价、商家等基础信息，可以来自定向的商户登记和不定向的信息爬取，因此这部分数据可能会出现一些问题，如标签缺失、标签体系不准确等等。比如一个淘宝卖家上传了一个新的商品，为了确保用户能在搜索端搜索到，卖家往往会让商品标题尽可能长（哪怕不通顺），糅合进更多的关键字，如“小熊煮蛋器可爱系单人居家使用健康早餐...”，这段内容，描述了很多物品的基本属性，例如品牌：小熊，商品：煮蛋器，适合两个人，可爱系等等，这类标签需要使用自然语言处理技术如分词、词性判断、实体提取等一系列词法与语法分析。

2.2 用户画像

用户画像包括用户个性标签、偏好等。如很多应用为了解决冷启动问题，会要求用户注册时让其自己选择兴趣标签以更精准地推荐，如左图；还有些应用会根据一些客观的、直接的特征如地理位置等提供可解释的服务，右图为美团 APP 使用地理位置根据距离推荐附近商家。



图 1: 用户画像构建实例

2.3 交互行为

交互行为即用户和物品发生的交互记录，在文本领域最为关键的就是用户对物品的评论，从这些往往伴随着评分的评论中可以提取用户的隐式兴趣标签、对某种商品的感情态度等，这是后文很多模型的重要训练数据来源。

3 基于文本信息的可解释推荐系统模型

3.1 隐变量模型

2006 年 Netflix Prize 开始后, Simon Funk 提出了矩阵分解方法, 后来在 Netflix Prize 的冠军 Koren 的经典论文 [1] 中被系统化总结为 LFM (Latent Factor Model, 隐变量模型, 或称隐语义模型等), 该论文于 2009 年发表在 IEEE 下的 “COMPUTER” 期刊上, 是推荐系统领域第一篇比较正式、全面介绍融合了机器学习技术的矩阵分解算法, 对于近几年基于矩阵分解的推荐算法的研究起到了非常大的影响。

矩阵分解是构建隐变量模型的主要方法, 即通过把整理、提取好的 “用户—物品” 评分矩阵进行分解, 来得到一个用户隐向量矩阵和一个物品隐向量矩阵。假设现在有一个 $M \times N$ 的矩阵, M 代表用户数, N 代表物品数, 想将用户、物品分别训练出两个隐属性, 即每个用户、每个物品都对应着一个二维向量, 即得到了一个 $M \times 2$ 的用户隐变量矩阵和一个 $2 \times N$ 的物品隐变量矩阵, 我们可以将每个用户、物品对应的二维隐变量看作是一个坐标, 将其画在坐标轴上。虽然我们得到的是不可解释的隐变量, 但是可以为其赋予一定的意义来帮助我们理解这个分解结果。比如我们把用户、物品的 2 维的隐向量赋予严肃文学 (Serious) vs. 消遣文学 (Escapist)、针对男性 (Geared towards males) vs. 针对女性 (Geared towards females), 那么可以形成论文中那样的可视化图片:

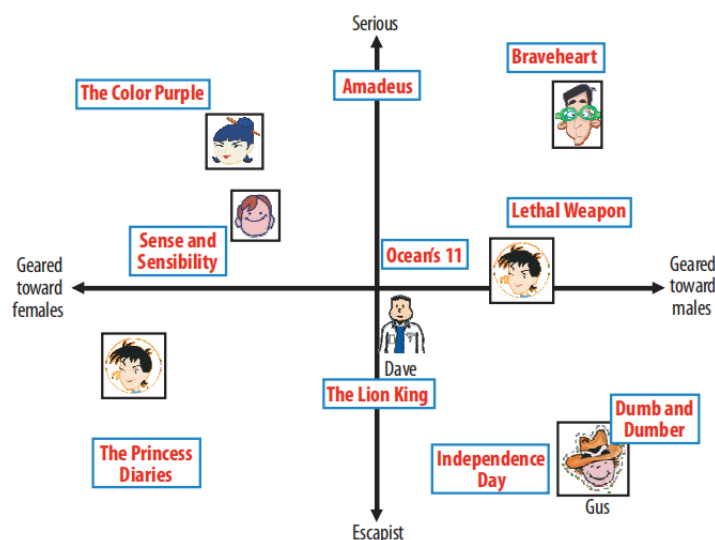


图 2: 选取严肃文学 vs. 消遣文学、针对男性 vs. 针对女性作为隐变量的简单例子

用户矩阵的各行是用户的向量表示 p_u , 物品矩阵各列是物品的向量表示 q_i , 二者的内积就是用户对物品的评分:

$$\hat{r}_{ui} = q_i^T p_u$$

可以看出用户和物品向量越契合, 即物品的各种特征恰好是用户喜欢的, 那么评分就高。但由于用户-物品评分矩阵往往是非常稀疏的, 直接采用线性代数中的矩阵分解策略是行不通的。一种想法是利用矩阵中已有的值, 期望 $q_i^T p_u$ 尽可能地接近这些值。得到稀疏的用户-物品评分矩阵后, 就可以恢复出完整的评分矩阵, 以此预测用户对于没有评分过的物品的评分。

这样以来矩阵分解可以转变成下面的优化问题:

$$\min_{q^* p^*} \sum_{(u,i) \in \mathbf{K}} \left(r_{ui} - q_i^T p_u \right)^2 + \lambda \left(\|q_i\|^2 + \|p_u\|^2 \right)$$

对原评分矩阵中存在的值，希望用户向量和物品向量相乘后尽可能地接近该值。上式中另外加入了正则化项，防止过拟合。因为目的不单单是逼近评分矩阵中存在的值，也希望能够最好地预测未知的数。

解上面的优化问题，可以使用随机梯度下降法，定义：

$$e_{ui} = r_{ui} - q_i^T p_u$$

求偏导数即可得出下降公式

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$

也可以使用交替最小二乘法，因为 q_i 和 p_u 都是未知的，前面的优化目标是非凸函数，不好求解，但是如果能够固定 q_i 和 p_u 中的一个，交替地更新另外一个，公式中只有一个变量，而且是二次的，优化问题就更容易得到最优解。可以看出梯度下降法更容易实现且更快，但交替最小二乘法可以并行化独立更新 q_i 和 p_u 。

本文重点讨论可解释推荐系统中文本处理技术，直观上看隐变量模型似乎与文本没有直接关系，但鉴于接下来的几个模型都或多或少牵涉到这个模型，将这个模型作为第一个进行叙述。

考虑到现实应用场景，作者还提出了四种优化策略：

(1) 加入对偏置项的考量

考虑到不同用户评分严格程度不同，打分范围不同。比如用户给电影打分，有的用户很严格，给分都在 4 分以下，有些用户较为宽容，即使是很差的电影也能给到 3 分，这就是用户的偏置项，物品也类似。考虑到上面这些因素，对评分的估计为：

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

其中 μ 为全局平均评分， b_i 为物品偏置项， b_u 为用户偏置项，从而优化目标变为：

$$\min_{p^*, q^*, b^*} \sum_{(u,i) \in \mathbf{K}} \left(r_{ui} - \mu - b_u - b_i - p_u^T q_i \right)^2 + \lambda \left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 \right)$$

(2) 引入附加输入信息

推荐系统往往需要处理冷启动问题，很多用户可能只对个别物品进行了评分，这就很难得出可靠的用户向量表示。引入其他的信息能够解决这种信息较少的问题。

推荐系统可以利用隐式信息，比如用户的浏览记录、搜索记录、鼠标停留信息等，在没有足够的明确信息（购买、评分）时，此类信息也能在一定程度上对用户进行刻画。

考虑到上面这些，作者引入用户的浏览记录和用户个人属性等信息。

定义 $N(u)$ 为用户浏览过的物品集合，每一个物品对应一个向量 $x_i \in \mathbb{R}^f$ ， $N(u)$ 中的物品给用户带来的特征可以表示为：

$$|N(u)|^{-0.5} \sum_{i \in N(u)} x_i$$

其中 $|N(u)|^{-0.5}$ 用于归一化。另外用户自身的属性也是一个信息来源，设用户有一组特征 $A(u)$ ，每个特征用向量表示 $y_a \in \mathbb{R}^f$ ，用户的属性给用户来的特征可以表示为：

$$\sum_{a \in A(u)} y_a$$

此时用户对物品的评分可以表示为：

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left[p_u + |N(u)|^{-0.5} \sum_{i \in N(u)} x_i + \sum_{a \in A(u)} y_a \right]$$

(3) 引入时序变量

有很多因素会随时间变化，比如用户的品味，打分的风格等。引入时序信号，可以捕获到用户或物品随着时间的改变：

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$$

$b_i(t)$ 是物品的偏置项，它会随时间改变，比如《甄嬛传》这部电视剧在刚上映时豆瓣评分不高，后来人们开始挖掘这部宫斗剧背后的职场价值、人生哲学等，评分一路飙升。 $b_u(t)$ 是用户偏置，如前所述，用户的品味会变化。 $p_u(t)$ 是用户向量，用户对各种电影的喜好会变化，比如之前喜欢看偶像剧，最近喜欢看历史剧。 q_i 是隐因素向量，因为物品的各种因素相对稳定，因此不需要引入时序变量。

(4) 考虑输入的可信程度

不是所有评分都有一样的权重，因此，作者对每个观察到的评分引入了可信程度，优化目标变为：

$$\min_{p^*, q^*, b^*} \sum_{(u,i) \in K} c_{ui} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

3.2 概率主题模型

最具有代表性的概率主题模型 LDA (Latent Dirichlet Allocation, 隐狄利克雷分布, 简称 LDA) 由 DM Blei 在 2003 年提出 [2]，是一种无监督的学习方法，它可以将文档集中每篇文档的主题以概率分布的形式给出，从而通过分析一些文档抽取出它们的主题后，便可以根据主题进行聚类或文本分类。同时，LDA 是一种典型的词袋模型。所谓词袋模型，是将一篇文档，我们仅考虑一个词汇是否出现，而不考虑其出现的顺序。比如在词袋模型中，“我喜欢海绵宝宝”和“海绵宝宝喜欢我”是等价的。

要深入了解该模型，首先回顾一下 Beta 分布、多项式分布和狄利克雷分布。Beta 分布是二项分布的共轭先验分布。给定参数 $\alpha, \beta > 0$ ，取值范围为 $[0, 1]$ 的随机变量 x 的概率密度函数：

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

其中:

$$\frac{1}{B(\alpha, \beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

多项分布是二项分布的推广形式, 令 $x_1 + x_2 + \dots + x_k = n$, $p_1 + p_2 + \dots + p_k = 1$, 则多项分布的概率密度函数:

$$f(x_1, x_2, \dots, x_k | n, p_1, p_2, \dots, p_k) = \frac{n!}{x_1! \dots x_k!} p_1^{n_1} \dots p_k^{n_k}$$

其中 $X \sim \text{Mult}(n, p_1, \dots, p_n)$ 。

狄利克雷分布是 Beta 分布在高维度上的推广, 密度函数:

$$f(x_1, x_2, \dots, x_k; \alpha_1, \alpha_2, \dots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^k x_i^{\alpha_i-1}$$

其中:

$$B(\alpha) = \frac{\prod_{i=1}^k \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^k \alpha_i)}, \sum x_i = 1$$

可以看出狄利克雷分布是多项式分布的共轭先验概率分布, 一般表达式如下:

$$\text{Dir}(\vec{p} | \vec{\alpha}) + \text{Mult Count}(\vec{m}) = \text{Dir}(p | \vec{\alpha} + \vec{m})$$

接着从 McAuley, Julian and Leskovec 总结 PLSA 模型的 [3] 入手。人类生成文档的过程可以概括为: 1) 先列出几个主题; 2) 以一定的概率选择主题, 以一定的概率选择这个主题包含的词汇; 3) 组合成一篇文章。如下图所示 (其中不同颜色的词语分别对应上图中不同主题下的词):

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

图 3: 文档中不同的关键词对应不同的主题 (颜色)

一篇文章往往有多个主题, 只是这多个主题各自在文档中出现的概率大小不一样。比如介绍一部影视剧的文档中, 往往会分别从类型、剧情、演员等多个主题进行介绍。假定共有 K 个可选的主题, 有 V 个可选的词, 令 $K=3$, 制作一个 3 面的骰子, 三个词分别为: 类型、剧情、演员; $V=3$, 制作 3 个有着 3 面的“主题-词项”骰子, 其中, 类型主题骰子的 3 个面上的词可以是: 古装、架空、小说改编, 剧情主题骰子的 3 个面上的词可以是: 夺嫡、复仇、江湖, 演员主题骰子的 3 个面上的词可以是: 胡歌、王凯、刘涛。

每写一个词，先扔该“文档-主题”骰子选择主题，得到主题的结果后，使用和主题结果对应的那颗“主题-词项”骰子，扔该骰子选择要写的词。先扔“文档-主题”的骰子，假设（以一定的概率）得到的主题是类型，所以下一步便是扔类型主题筛子，（以一定的概率）得到类型主题筛子对应的某个词：古装。

上面这个投骰子产生词的过程简化下便是：“先以一定的概率选取主题，再以一定的概率选取词”。不停的重复扔“文档-主题”骰子和“主题-词语”骰子，重复 N 次（产生 N 个词），完成一篇文档，重复这产生一篇文档的方法 M 次，则完成 M 篇文档。

上述过程抽象出来即是 PLSA 的文档生成模型。在这个过程中，我们并未关注词和词之间的出现顺序，所以 PLSA 是一种词袋方法。生成文档的整个过程便是选定文档生成主题，确定主题生成词。

文档 d 和词 w 是我们得到的样本，可观测得到，所以对于任意一篇文档，其 $P(w_j|d_i)$ 是已知的。从而可以根据大量已知的文档-词项信息 $P(w_j|d_i)$ ，训练出文档-主题 $P(z_k|d_i)$ 和主题-词项 $P(w_j|z_k)$ ，如下公式所示：

$$P(w_j | d_i) = \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i)$$

故得到文档中每个词的生成概率为：

$$P(d_i, w_j) = P(d_i) P(w_j | d_i) = P(d_i) \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i)$$

由于 $P(d_i)$ 可事先计算求出，而 $P(w_j|z_k)$ 和 $P(z_k|d_i)$ 未知，所以 $\theta = (P(w_j|z_k), P(z_k|d_i))$ 就是我们要估计的参数（值），通俗点说，就是要最大化这个 θ ，常用的参数估计方法有极大似然估计 MLE 、最大后验证估计 MAP 、贝叶斯估计等等。

LDA 中，选主题和选词依然都是两个随机的过程，依然可能是先从主题分布中抽取出主题：类型，然后再从该主题对应的词分布中抽取出词：古装。但和 PLSA 区别在于，PLSA 中主题分布和词分布是唯一确定的，能明确的指出主题分布可能就是 {类型：0.4，演员：0.3，剧情：0.3}，词分布可能就是 {古装：0.4，架空：0.3，小说改编：0.3}；但在 LDA 中，主题分布和词分布不再唯一确定不变，即无法确切给出。例如主题分布可能 {类型：0.4，演员：0.3，剧情：0.3}，也可能是 {类型：0.5，演员：0.3，剧情：0.2}，到底是哪个我们不再确定（即不知道），因为它是随机的可变化的。但再怎么变化，也依然服从一定的分布，即主题分布跟词分布由狄利克雷分布先验随机确定。正因为 LDA 是 PLSA 的贝叶斯版本，所以主题分布跟词分布本身由先验知识随机给定。

换言之，LDA 在 PLSA 的基础上给这两参数 ($P(z_k|d_i)$ 、 $P(w_j|z_k)$) 加了两个先验分布的参数（贝叶斯化）：一个主题分布的先验分布狄利克雷分布， α 和一个词语分布的先验分布狄利克雷分布 β 。

综上，LDA 也就是 PLSA 的贝叶斯版本，文档生成后，两者都要根据文档去推断其主题分布和词语分布（即两者本质都是为了估计给定文档生成主题，给定主题生成词语的概率），只是用的参数推断方法不同，在 PLSA 中用极大似然估计的思想去推断两未知的固定参数，而 LDA 则把这两参数设为随机变量，且加入狄利克雷先验。

LDA 模型中一篇文档生成的方式如下：

- (1) 从狄利克雷分布 $\text{Dir}(\vec{\alpha})$ 中取样生成文档 m 的主题分布 $\text{Mult}(\vec{\theta}_m)$
- (2) 从主题的多项式分布 $\text{Mult}(\vec{\theta}_m)$ 中取样生成文档 m 第 n 个词的主题 $z_{m,n}$
- (3) 从狄利克雷分布 $\text{Dir}(\vec{\beta})$ 中取样生成主题 $z_{m,n}$ 对应的词语分布 $\text{Mult}(\vec{\phi}_{z_{m,n}})$
- (4) 从词语的多项式分布 $\text{Mult}(\vec{\phi}_{z_{m,n}})$ 中采样最终生成词语 $w_{m,n}$

第 m 篇文档中的第 n 个词是 t 的概率为：

$$P(w_{m,n} = t \mid \vec{\theta}_m, \Phi) = \sum_{k=1}^K P(w_{m,n} = t \mid \vec{\phi}_k) P(z_{m,n} = k \mid \vec{\theta}_m)$$

文档 m 中词、隐变量和参数的联合概率分布为：

$$P(\vec{w}_m, \vec{z}_m, \vec{\theta}_m, \Phi \mid \vec{\alpha}, \vec{\beta}) = P(\vec{\theta}_m \mid \vec{\alpha}) P(\Phi \mid \vec{\beta}) \prod_{n=1}^{N_m} P(w_{m,n} \mid \vec{\phi}_{z_{m,n}}) P(z_{m,n} \mid \vec{\theta}_m)$$

所有文档的联合概率分布为：

$$P(\mathcal{W}, \mathcal{Z}, \theta, \phi \mid \vec{\alpha}, \vec{\beta}) = \prod_{k=1}^K P(\vec{\phi}_k \mid \vec{\beta}) \prod_{m=1}^M \left(P(\vec{\theta}_m \mid \vec{\alpha}) \prod_{n=1}^{N_m} P(w_{m,n} \mid \vec{\phi}_{z_{m,n}}) P(z_{m,n} \mid \vec{\theta}_m) \right)$$

上述公式中， $\vec{\alpha}$ 表示语料级别的参数，用于生成每篇文档的主题分布， $\vec{\beta}$ 表示语料级别的参数，用于生成每个主题的词汇分布， ϕ 表示语料级别的参数，即每个主题的词汇分布， θ 表示文档级别的参数，即每个文档的主题分布， $z_{m,n}$ 表示词级别的参数，即每个词对应的主题。

给定一篇文档，要反向求出文档的主题分布，就需要推出这些参数。涉及到 Gibbs Sampling (吉布斯采样) 的细节这里暂不展开，最终我们训练得到 LDA 模型：

- (1) 对于文档中的每个词汇 w ，随机的赋予一个 topic 编号 z
- (2) 重新扫描语料库，对每个词 w ，使用 Gibbs Sampling 公式对其采样，求出它的 topic，在语料中更新
- (3) 重复步骤 (2)，直到 Gibbs Sampling 收敛
- (4) 统计语料库的 topic-word 共现频率矩阵，该矩阵就是 LDA 的模型

McAuley and Leskovec[4] 提出了基于从评论中学习到的潜在主题来理解隐变量模型中的隐变量。在此之前，对用户评分的预测往往是基于历史信息，比如协同过滤算法，实质上进行推荐的依据都是基于用户和物品的行为，如用户给哪些物品打了分，物品被哪些用户打了分，这并没有利用到用户和物品本身的特征信息，因此可解释性较差。这篇文章中，作者将用户给出的评论文本作为用户评分的“解释信息”，提出了 HFT(Hidden Factor and Topic) 模型，它将隐变量模型 LFM 和 LDA 联系起来，将隐变量的每个维度与 LDA 主题分布向量的一个维度联系起来，通过将每个用户的隐变量投射到 LDA 的潜在主题空间中，可以帮助理解为什么用户对目标物品进行特定的评价。最终获得了用户对物品评分的高度可解释性的文本标签，有助于利用评论文本“解释”评级，也能据此更准确地进行评分预测。

McAuley 和 Leskovec 最大的贡献点在于他们首次将用户评论文本纳入考量。接着，Tan 等人在 2016 年 [5] 提出了 RBLT(Rating-Boosted Latent Topics) 模型，在同一空间中映射物品可推荐性和用户偏好分布。整个过程大致分为：1) 可推荐学习提取评级提升潜在主题空间项目的推荐分布；2) 与可推荐分布在同一主题空间识别用户偏好分布的偏好学习；3) 评级预测或顶级推荐

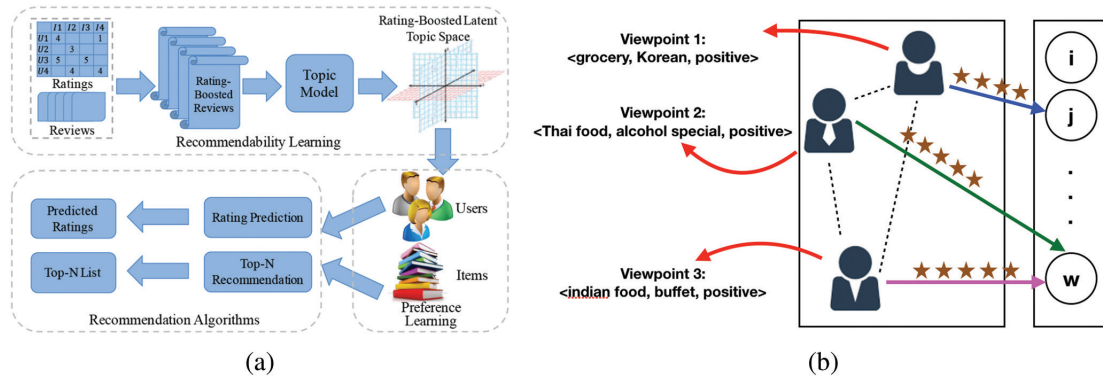


图 4: (a):RBLT 模型框架,(b): 朋友关系、文本评论和评分的例子

的推荐算法。实验表明其在 top-N 推荐的场景中取得了良好的效果。让笔者觉得很巧的是这篇文章中采用了“强化评论”的方法，也即作者认为，如果一个物品得到了较高的评分，我们认为这条评分记录所对应的评论文本中出现的特征或主题更为重要，“强化”的手段就是在建模的过程中将这条评论多重复几次，这个方法与笔者在第二次课程作业中采用的新闻摘要建模相同，对于关键词，如动词、名词，我们认为其在词袋模型中能反应出更多信息、权重较大，因此我们对文本建模时会将这些词多重复几次。

以上的模型中都给物品主题和隐变量建立了一一对应的关系，2019 年，cheng 等人 [6] 提出这样做有失偏颇：一个主题特征相对于不同的用户考虑不同的物品的相对重要性可能会非常不同。例如有些用户在看影视剧时会关注剧情而不在乎演员是谁，而有些用户更在乎剧中有没有自己喜欢的演员。因此，为了进行准确的预测，准确地捕捉每个隐变量对于用户对某个项目的重要性是很重要的。传统的隐变量模型对于每一个物品的因子都是以同样的重要性和相似度对待的，在评价预测中，每个主题特征对用户都是同等重要的，这很容易使得结果陷入局部最优。这些作者提出了一个多模态方面感知的主题模型 MMALFM，利用用户评论和项目图像一起学习潜在主题，这些主题用于建模用户在不同方面的偏好和项目属性和估计一个特征的重要性。然后将这些结果集成到一个提出的面向方面的隐变量模型中，以估计面向特征的评级，从而进行最终的评分预测。特别地，引入了一个权重矩阵，将隐变量与不同的主题特征联系起来，使得该模型能够预测特征的评级，整体评分是由各特征评分的线性组合得出。

Wu 和 Ester 在 2015 年 [7] 研究了基于物品特征的个性化情绪评估问题也考虑了用户对不同特征的偏好，这是对 zhang 等人的 ETF 模型 [8] 的一个改进，另外，ETF 模型将基于特征的偏好挖掘作为预处理步骤，而 Wu 的模型是一个统一的模型，将偏好挖掘作为模型的一部分。这使得我们的方法能够用于分析评论的特征分布和评论中表达的隐变量的评分。

基于可信任的社会关系的推荐是我们在生活中最常见到的可解释推荐之一，如 QQ 音乐的“好友某某最近在看”，微信的“在看”等等。引入对可信社交关系的考量，ren 等人在 2017 年 [9] 中提出了 SCVR(Social-Collaborative Viewpoint Regression) 模型，即社会协作观点回归模型，通过联合建模概念、主题、情绪标签和社会关系来预测用户评分，具体来说，抽象出元组 viewpoint (用户评论，可信社会关系，主题和情感)，并通过实验在三个基准数据集上证明了模型优于最先进的基线。

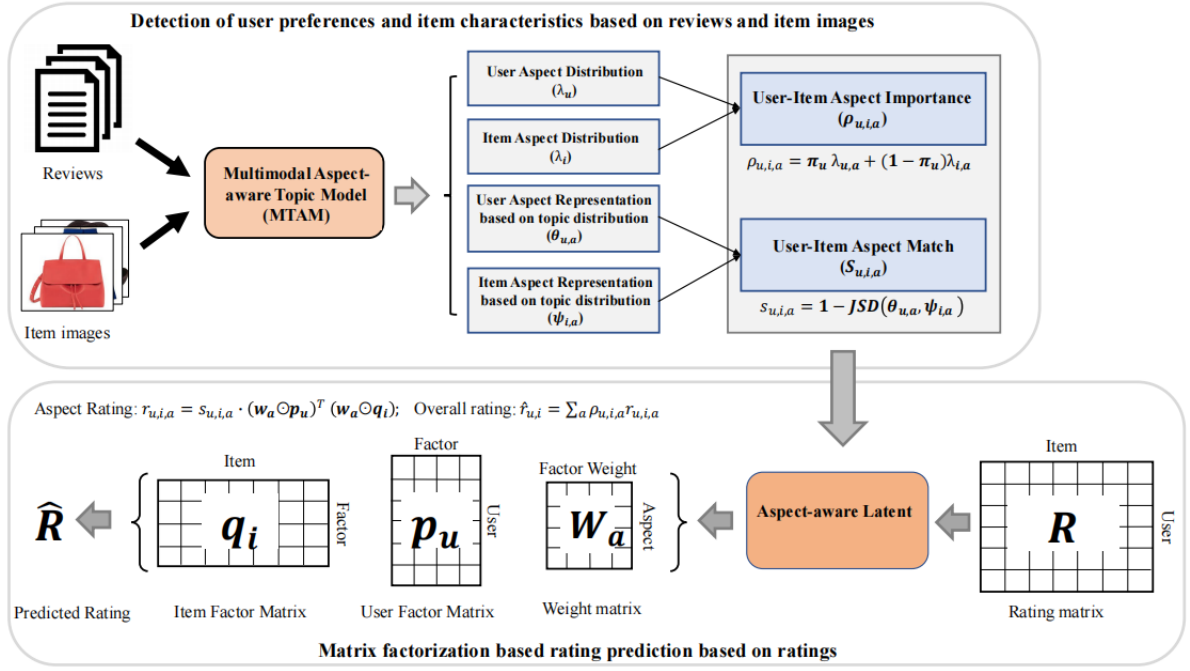


图 5: MMALFM 模型框架

3.3 词向量模型

来自 Google 的 Mikolov 和 Chen 等的经典论文 [10] 和 [11] 在借鉴和创新的同时，采用了简单模型 + 大规模样本的实用路线，系统化总结了词向量模型。作者将该模型开源，加之 Google 的业界影响力，将词向量技术一跃而起，迅速成为最热门的研究方向之一。

首先需要了解词嵌入，起先源自于自然语言处理领域，主要是利用背景信息构建词汇的分布式表示，最终可以得到一种词的向量化表达，即用一个抽象的稠密向量来表征一个词。“嵌入”本身是一个数学概念。通俗地讲，如果将词看作是文本的最小单元，词嵌入本身是一种特别的映射过程，即将文本空间中的某个词，通过一定的方法，映射或者说嵌入到另一个数值向量空间，原来的整数全部变为实数，是用连续向量表示离散变量的方法。

是用浅层神经网络学习得到每个词的向量表达，而且在工程上进行了优化，使得百万词的规模在单机上可以几分钟轻松跑出来。它有两种网络结构，分别是 CBOW (Continues Bag of Words) 和 Skip-gram:

这两种结构都是假定每个词都跟其相邻的词的关系最密切，不同的是 CBOW 模型的动机是每个词都是由相邻的词决定的，所以 CBOW 的输入是 w_t 周边的词，预测的输出是 w_t ，而 Skip-gram 模型的动机则是每个词都决定了相邻的词，输入是 w_t ，输出是 w_t 周边的词。这里相邻的词通常由一个滑动窗口来获得，滑动窗口的长度为 $2c + 1$ (目标词前后各选 c 个词)，从句子左边滑到右边，每滑一次，窗口中的词就形成了一个正样本。

经验上讲 Skip-gram 的效果要更好一点，以 Skip-gram 为例，理解一下原理。上面讲到通过滑动窗口可以获取到训练样本，有了训练样本之后我们就可以着手定义优化目标了，既然每个词 $p(w_{t+j}|w_t)$ 都决定了相邻词 w_{t+j} ，基于极大似然，我们希望所有样本的条件概率 $p(w_{t+j}|w_t)$

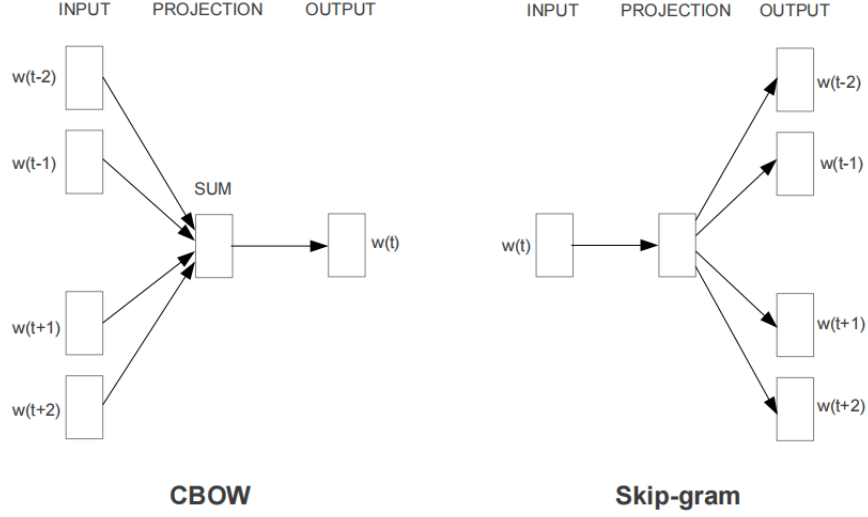


图 6: 词向量模型的两种网络结构

之积最大，于是有：

$$J'(\theta) = \prod_{t=1}^T \prod_{-c \leq j \leq c, j \neq 0} p(w_{t+j} | w_t; \theta)$$

转为最大化对数似然：

$$J(\theta) = \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

其中， T 是文本长度，即单词总数， c 是窗口大小，在实际训练学习过程中，训练样本是由中心词和其上下文词组成一个个 Pair 对，所以可以将目标函数写成一个更容易理解的式子：

$$\arg \max_{\theta} \sum_{(w,c) \in D} \log p(c | w)$$

其中， (w, c) 表示中心词 w 与其上下文词 c 构成的样本对， D 是语料中所有单词及其上下文词构成的样本对集合，这里的 θ 为待定参数集。

接下来的问题是怎么定义 $p(c|w)$ ，作为一个多分类问题，最简单最直接的方法是直接用 Softmax 函数，同时希望用隐变量来表示每个词。

于是设 v_w 为中心词 w 的词向量， v_c 为上下文词 c 向量，向量维度等于设置的隐藏层单元数量。整个语料共有 V 个单词，也就是词汇表单词数。

因此：

$$p(c | w; \theta) = \text{Softmax}(v_w \cdot v_c) = \frac{\exp(v_w \cdot v_c)}{\sum_{i=1}^V \exp(v_w \cdot v_i)}$$

代入上面的目标函数，可以得到：

$$\arg \max_{\theta} \sum_{(w,c) \in D} \log p(c | w) = \sum_{(w,c) \in D} \left(\log e^{v_w \cdot v_c} - \log \sum_{i=1}^V e^{v_w \cdot v_i} \right)$$

其中，参数 θ 是 v_{w_i} 和 v_{c_i} ， $i \in 1, \dots, d$ ， d 是向量维度。

这两个单词的隐向量的点积表示语义的接近程度，其实点积的物理意义就是词之间的距离，点积越大，表明两个单词的语义越接近，向量点积展开如下：

$$v_w \cdot v_c = v_w^T v_c = v'_w v_c = \sum_{i=1}^d v_w^i \times v_c^i$$

而 Softmax 的作用就是将点积转换成概率，下图的例子可以辅助理解：

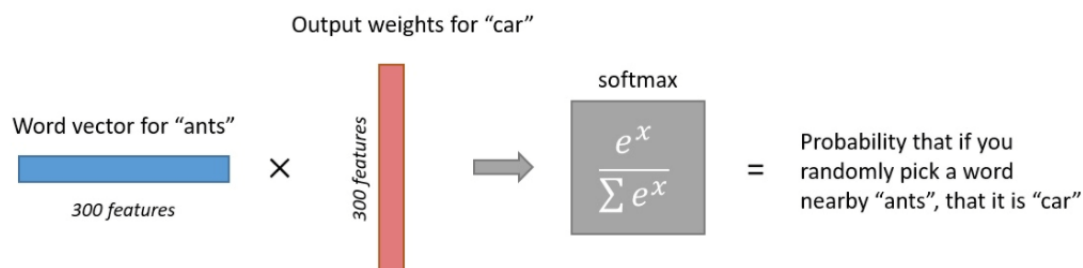


图 7: 点积转换成概率

两种模型都有各自的优缺点，不过根据 Mikolov 的原论文 [10]，Skip-gram 在处理少量数据时效果很好，可以很好地表示低频单词。而 CBOW 的学习速度更快，对高频单词有更好的表示。实际应用中，Skip-gram 方式会更常见。

鉴于词嵌入技术的优点，人们不满足仅局限在自然语言处理领域，很快有人就尝试把这种嵌入式表示的思想应用到推荐系统中。最常见的就是构建实体的嵌入式表示，使得多种实体的嵌入式表示存在于同一个隐含空间内，进而可以计算两个实体之间的相似性。2016 年，研究人员 [12] 将词向量的算法思路迁移到基于物品的协同过滤 ICF (Item-Based CF) 上，以物品的共现性作为自然语言中的上下文关系，构建神经网络学习出物品在隐空间的向量表示 Item2Vec。在 Item2Vec 中，样本数据可以有两种看待方式：

(1) 基于时序

认为物品之间存在强时序关系，即前面物品对后面物品的产生有很大的影响，比如一个人看了电视剧《琅琊榜》的第一集，他接下来很有可能会继续看第二集，那么可以把一段时间内连续发生的物品序列看作是句子，序列中的物品看作是句子中的词，这样与 Word2Vec 的训练数据的构造就没什么区别了，因此可以直接按照 Word2Vec 的方法进行词嵌入训练。

(2) 基于集合

认为物品之间存在非常弱的时序关系，或者因为某种原因，我们无法获得物品的序列。那么这种情况只能放弃考虑物品间的时空信息，转而使用物品集合来取代序列。通常我们把用户的行为序列视为一个集合（当然还有其他集合构造方式），我们假定一个静态的环境，在这个环境中，共现在同一个集合的物品就可以认为它们是相似的，然后视为正样本，不管它们是以什么样的顺序产生的，如果某两个物品频繁共现，二者的 Embedding 则可能也很相似。很明显，这种利用集合共现性构建样本的方式与 Word2Vec 的输入样本还是不一样的，因此无法直接应用 Word2Vec 的算法，这就需要对 Word2Vec 的过程进行调整。另外还需要注意的是，上面的这种假设是有局限性的，不一定适用于其它场景。

3.4 深度学习模型

常见的深度学习模型如 CNN、RNN 等在可解释推荐系统领域有出色的表现，它们还适用于不同的可解释推荐任务，如评分预测、top-K 推荐和序列推荐等。在本节中，我们将回顾可解释推荐的深度学习方法。

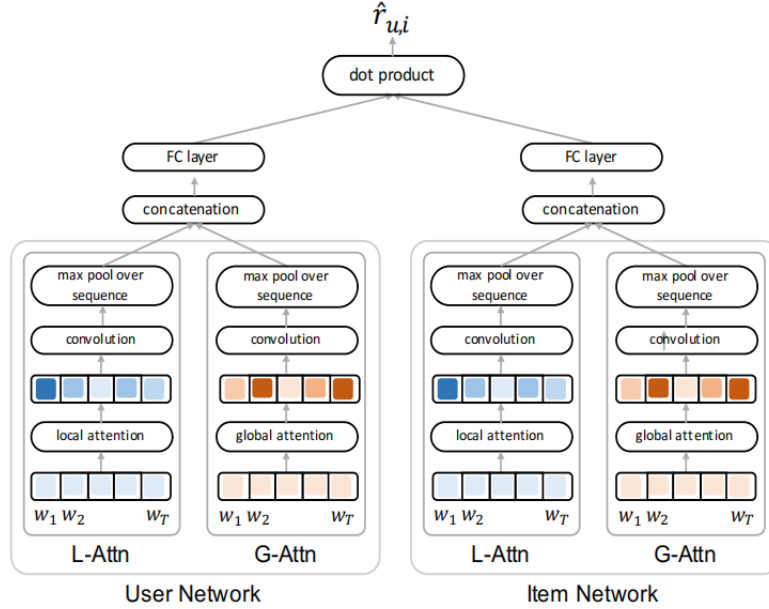


图 8: 双重注意力模型架构

2017 年 Seo 等人 [13] 提出在评论文本上，使用卷积神经网络 (CNNs) 对用户偏好和物品属性进行建模，如模型架构所示，其中左侧 User Network 的输入是该用户给出的所有评论；右侧 Item Network 的输入是该产品收到的所有评论。输出是该用户对给定产品的评分。它使用局部和全局的双重注意力进行解释，在预测用户-物品评分时，该模型会有选择地选取不同注意力权重的评论词语。根据学习到的注意力权重，该模型可以显示出评论的哪一部分对输出结果更重要。此外，该模型可以突出评论中的重要单词作为解释，帮助用户理解推荐。

Wu [14] 等人提出以前的模型学到的用户和物品的隐层向量都是一个静态的，没有考虑到用户对物品的偏好，于是他们团队提出了 CARL 上下文感知模型，使用用户的评论和物品总评论，通过用户-物品对进行 CNN 训练，加入了注意力层，摘要层，学习到的是用户和物品的联合表达。更倾向于自然语言处理的论文，和传统的推荐模型差距比较大。

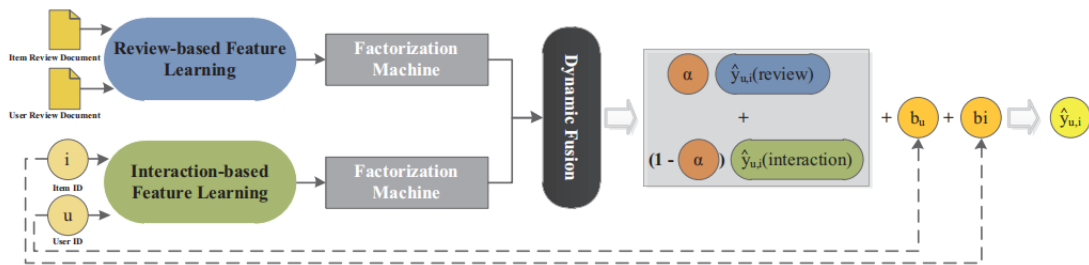


图 9: CARL 模型框架

2018 年，LU 等人 [15] 改进了基于评论的推荐算法，提出了基于注意力机制的 GRU 神经网络

络模型（attention-GRU 模型）对评论文本集合建模，之后结合矩阵分解（MF）应用在推荐任务中。对于每个用户或者物品来说，评论文本集合会由很多条单独的 review 构成，作者首先对每条评论应用 attention-GRU 模型得到表征向量，之后 user reviews 和 item reviews 的表征向量由各个 review 的表征向量取平均得到。attention-GRU 模型结构如下：

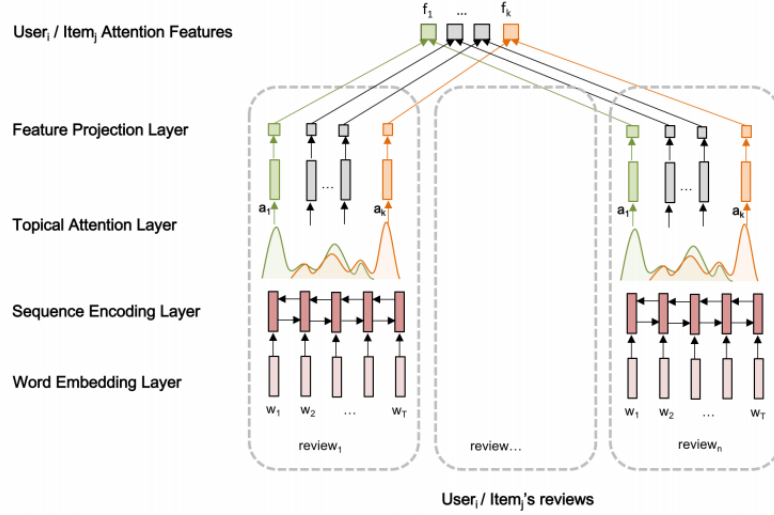


图 10: attention-GRU 模型框架

这里主要强调一下下面的三层。其中 Word Embedding 层采用 word2vec 初始化方式；序列编码层采用双向 GRU 建模，sentence 中第 t 个单词的表示可以是 $h_t = [h_t^{\rightarrow}, h_t^{\leftarrow}]$ ，其中 h_t^{\rightarrow} 表示 forward GRU 的 t 时刻向量， h_t^{\leftarrow} 表示 backward GRU 的 t 时刻向量；主题注意层中，如果把 review 最终表征向量的每一维当作一种与用户特性或者物品特性相关主题，每一维的值表示该主题的强度。对于一个特定的主题来说，评论中的每个词对该主题的贡献程度是不一样的，因此作者提出注意力机制加权方式，即每个单词对每个主题都会学习一个 attention，之后加权得到该主题的代表向量。

LI 等人在 2017 年 [16] 提出了一个基于深度学习的命名框架 NRT，它可以同时预测精确的评分，并生成高质量的抽象提示，模拟用户体验和感受。对于抽象提示生成，在来自不同领域的基准数据集上进行的广泛实验表明，NRT 比最先进的方法取得了显著的改进。

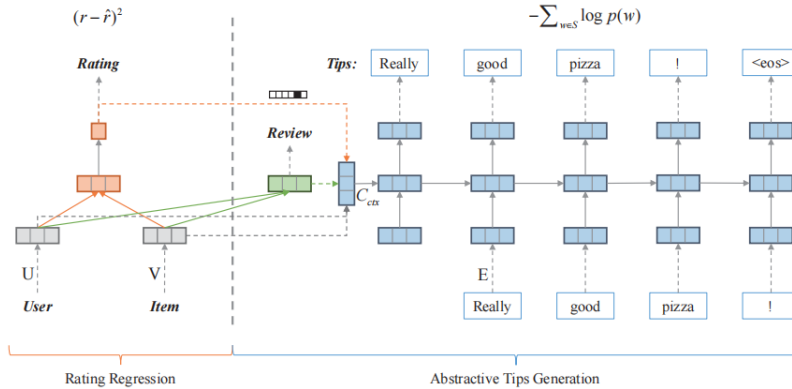


图 11: NRT 模型框架

Costaet 等在 2018 年 [17] 取得了一个突破，与之前的标签化解释、短语解释不同的是，他

们提出了一种基于字符级 RNN 的自然语言解释自动生成方法。该模型将用户评分作为辅助信息连接到输入组件中，使模型能够根据预期的评分（情感）生成评论。与许多基于预定义模板生成解释的可解释推荐模型不同，该模型能够以自然语言的方式自动生成解释。通过选择不同的参数，模型可以产生不同的解释来吸引用户。

Chen [18] 等在 2019 年提出了一个可解释推荐的协同注意多任务学习模型 CAML (Co-Attentive Multi-Task Learning)，通过充分利用推荐任务和解释任务之间的相关性，提高了推荐的预测准确性和可解释性。特别地，该模型受到认知心理学中人类的信息处理模型的启发，设计了一个编码器-选择器-解码器架构，他们还提出了一个层次的协同注意选择器来有效地建模这两个任务转移的交叉知识不仅提高了推荐任务的预测准确性，而且还生成了流畅、有用和高度个性化的语言解释。

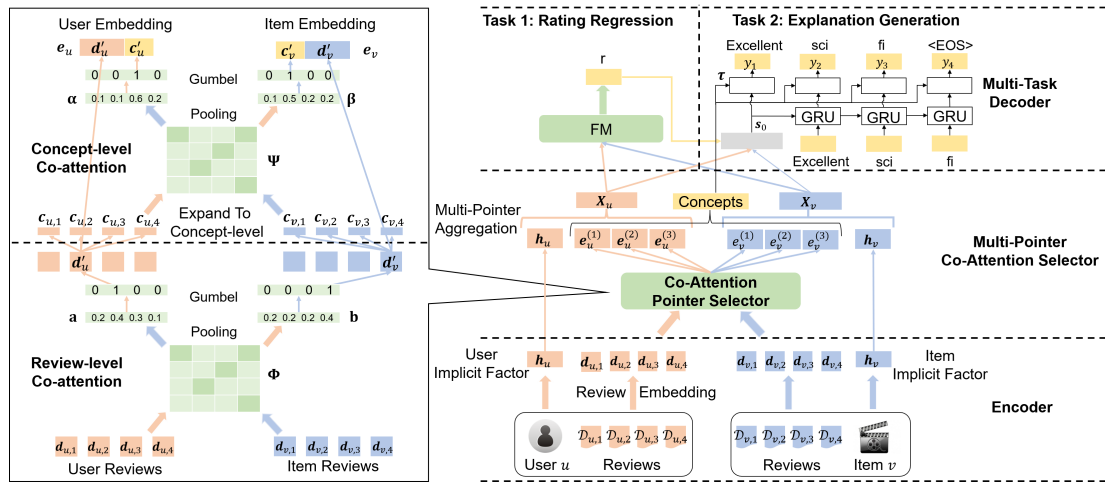


图 12: CAML 模型框架

Chang 等人在 2016 年 [19] 提出了另一种生成自然语言解释的方法，该方法首创性引入众包思想。笔者认为该思想可以理解为，通过集合人类共同的智慧修正模型，以华东师范大学在线评测系统为例，该系统会根据算法题的算法标签向用户推荐其尚未通过的题目，目前支持已经通过某题目的用户对题目的算法标签进行编辑，一方面，集众人之力使得算法题的标签更加准确；另一方面，对众包对象做了限制，认为顺利通过一道题目的人才拥有话语权。

类似地，论文作者们认为，算法生成的解释可能过于简单和没有说服力，而人类可以克服这些限制。受人们如何解释口碑推荐的启发，作者设计了一个将众包和计算结合起来生成解释的流程。他们首先基于非监督学习方法提取电影的主题特征，然后为主题特征生成自然语言解释。更具体地说，作者收集了每个方面的相关评论引用，然后请大众工作者将这些引用合成为解释。最后，作者根据用户的活动建立用户偏好模型，并以个性化的方式给出解释。实验通过在 220 名 MovieLens 用户中进行，对照评估了个性化自然语言解释与个性化标签解释的效率、有效性、信任度和满意度。

随着近几年机器学习领域的飞速发展，可解释推荐的深度学习的范围和文献并不局限于本小节介绍的研究。深度学习模型在本质上通常是复杂的，有时很难确定模型所提供的解释是否真的反映了产生建议或决策的真实机制，其可解释性仍然是一个重要的开放问题，需要更高级的解释模型来理解的行为。

3275. 章鱼哥没有日历

题面 统计数据 4 个讨论

单点时限: 2.0 sec
内存限制: 256 MB

一月到十二月, 这种无聊的东西, 对于章鱼哥这种神奇的生物来说, 当然是没有意义的。对于章鱼哥来说, 只有年和日, 没有月。他只关心某一天是这一年的第几天。



如果你要告诉它华东师范大学的校赛将在 2017 年 5 月 20 日举行, 他就会非常郁闷了, 因为他对 5 月 20 日根本就没有概念。他只关心 5 月 20 日是 2017 年的第几天。

如果你要告诉它 2017 年 2 月 29 日, Uncle John 要举行婚礼了, 他也会似懂非懂。因为他对月没有概念, 所以他根本就不知道 2017 年 2 月 29 日存不存在 (事实上是不存在的)。

现在给出 2017 年某一天的日期, 你要告诉章鱼哥这是 2017 年的第几天。如果根本没有这一天, 你也要告诉章鱼哥这一天「不存在的」。

707 人解决, 748 人已尝试。
850 份提交通过, 共有 2400 份提交。
1.0 EMB 奖励。

创建: 4 年, 1 月前。
修改: 3 年, 9 月前。
最后提交: 1 月前。
来源: 2017 华东师范大学校赛

标签

date × implementation ×
language exercise ×

保存标签

图 13: 华东师范大学在线评测系统-众包编辑题目算法标签

3.5 知识图谱模型

2012 年 12 月, 谷歌正式提出了“知识图谱”这个术语, 知识图谱旨在描述真实世界中存在的各种实体或概念, 以及他们之间的关联关系。其中, 每个实体或概念用一个全局唯一确定的 ID 来标识, 每个属性-值对用来刻画实体的内在属性, 而关系用来连接两个实体, 刻画它们之间的关联。知识图谱可以融合多种数据源丰富数据语义信息, 并且可以结合推理得到的隐含信息为用户提供服务。随着信息检索、智慧城市等应用领域的发展要求, 将知识图谱应用于这些领域以提高用户体验和系统性能, 成为学术界和工业界关注的热点。

这个模型和上述的几种不同, 直接通过具体例子进行介绍。

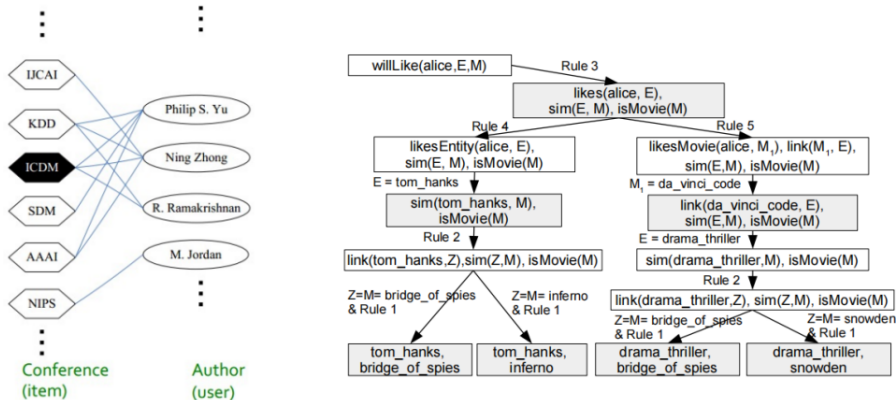


图 14: 左图为课程 PPT 中 PageRank 的示例图 [20], 右图为论文中电影推荐的简单例子

Catherine 等人在 2017 年 [21] 的这篇短小精悍的只有两页的论文首次将知识图谱引入推荐系统领域, 使用 Personalized PageRank 程序对物品和知识图谱实体进行联合排序的方法, 该方法生成推荐及其解释。该文章以电影推荐场景为研究对象, 通过将实体与对应电影一起排列来产生一个被排序的实体列表作为解释。以一组规则和一个事实数据库作为输入, 并使用这些规则在一个小图中生成每个查询的近似局部基础。查询的候选答案是图中满足规则的节点。然后, 通过在图上运行一个个性化的 PageRank 算法, 对候选人进行排名。概括为两个关键步骤: 1) 使用个性化的 PageRank 算法联合对用户的项目和实体进行排序; 2) 将结果整合为建议和解释。

Ai 等人认为 [22] 以往的可解释推荐系统通常使用各种非结构化信息源进行推荐，如文本评论、视觉图像和各种隐式或显式反馈。该团队试图为推荐系统提供结构化知识，从而帮助系统为推荐项目生成更可信的解释从而提高推荐系统的说服力、效率、有效性和透明度。这篇文章中，为了生成基于自然语言的解释语句，引入了“解释路径”的概念，即认为生成建议解释的关键是从用户到知识图中的项目找到一个合理的逻辑推理序列。在这项工作中，通过构建潜在知识空间中用户和项目之间的解释路径来找到这样一个序列，如解释 Bob 购买 iPad 行为，可能是因为 Bob 在评论中经常提到 IOS 系统，也可能是因为 Bob 购买过苹果的产品而 iPad 也是苹果的产品，这就是对这种行为的两种解释路径。

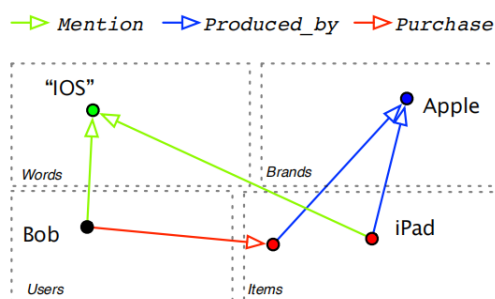


图 15: Bob 购买 iPad 的两种解释路径

Ma 等人 [23] 提出了一种新的联合学习框架 RuleRec，该框架通过规则学习模块和推荐模块的联合学习实现了较高准确度的可解释推荐。该框架鼓励两个模块在生成有效和可解释的建议方面相互补充：1) 从以物品为中心的知识图中挖掘的归纳规则，总结推断不同的项目关联的常见多跳关系模式，并为模型预测提供人类可读的解释；2) 推荐模块可以通过诱导规则来增强，从而对处理冷启动问题具有更好的泛化能力，在真实数据集上的项目推荐方面取得了显著的改进。

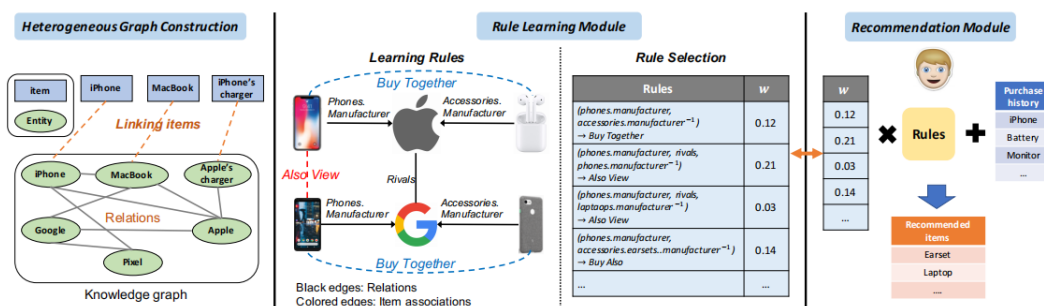


图 16: RuleRec 模型框架

3.6 小结

本章系统总结了可解释推荐系统领域和文本直接或间接相关的一些模型，为后文的应用提供了理论支持。

4 模型评估

需要注意的是，推荐系统的可解释性和准确性（推荐性能）往往会存在矛盾，二者不可兼得，从模型评估的角度往往要把二者分开来看。

4.1 解释性能评估

解释性能评估方法包含在线评估、离线评估和用户研究、案例研究评估。通常，在线评估更容易实施，而在线评估和用户研究取决于实际系统中数据和用户的可用性。

- 离线评估

离线评估通常有两种评估推荐解释的方法。一种是忽略解释质量，评估可解释推荐模型可以解释的推荐项目的百分比；第二种方法是准确评估解释的质量。但是，为了更全面地评估建议解释，需要更多的离线评估措施和协议。

- 在线评估

评估解释建议的另一种方法是通过在线实验，也是基于转换率（CR）和点击率（CTR）等在线测量，类似于推荐性能的在线评估。可能有几种不同的观点需要考虑在线评估解释，包括说服力，有效性，效率和解释的满意度。

- 基于用户研究的模拟在线评估

在线评估需要一个拥有大量用户的已部署系统，这通常需要广泛的协作或与商业公司合作。一种较为简单的方式是通过基于志愿者或付费实验科目的用户研究，模拟在线评估。志愿者或有偿受试者可以直接由研究人员雇用，也可以基于各种在线众包平台聘用。

- 案例研究定性评估

作为定性分析的案例研究也经常用于可解释的推荐系统研究。提供案例研究有助于理解可解释推荐模型背后解释的有效性，并且有助于读者理解提议的方法何时起作用以及何时不起作用。研究者提供了解释顺序的案例研究，作者发现许多顺序建议都可以根据用户的“一对多”或“一对一”行为模式进行解释，这些解释可以帮助用户清楚地理解为什么推荐一个项目，以及推荐的方式。

4.2 推荐性能评估

推荐性能的评估方法也包括离线评估和在线评估。在离线方法中，通常可以使用平均绝对误差（MAE）和均方根误差（RMSE）来评估评级预测的性能。对于 top-K 推荐，可以使用许多排名度量进行评估。最常用的度量可以是精确度，召回率，F1 度量值，归一化折扣累积增益量（NDCG）等。在在线方法中，通过点击率（CTR），转换率（CR）以及其他与业务相关的衡量指标进行评估。在线评估通常被商业公司使用，可以通过大量用户的访问进行实验。

5 应用场景

从笔者的角度，虽然可解释推荐系统的模型千变万化，但落实到应用场景，大致可以分为两种类型：文本生成与情感分析；前者应用在召回场景中，用于向用户直接推荐商品，后者应

用在模型修正中，从用户反馈的文本信息中提取用户情感，以更好地刻画用户画像、评估推荐效果等。

5.1 文本生成

在可解释推荐系统中，文本生成多数指的是生成解释内容，如 Wu 等人的 [7]ETF 模型可以生成词云来解释推荐的原因：



图 17: 基于字符级 RNN 的评论生成

可以看出不仅对旅馆生成了用户所关注的内容标签，也对被给出低分和高分的旅馆生成了形容词标签，被打低分是因为服务员态度粗略，打高分是因为服务态度好，给人温馨感等等。

Costa [17] 的模型可以调整参数生成不同的解释文本：

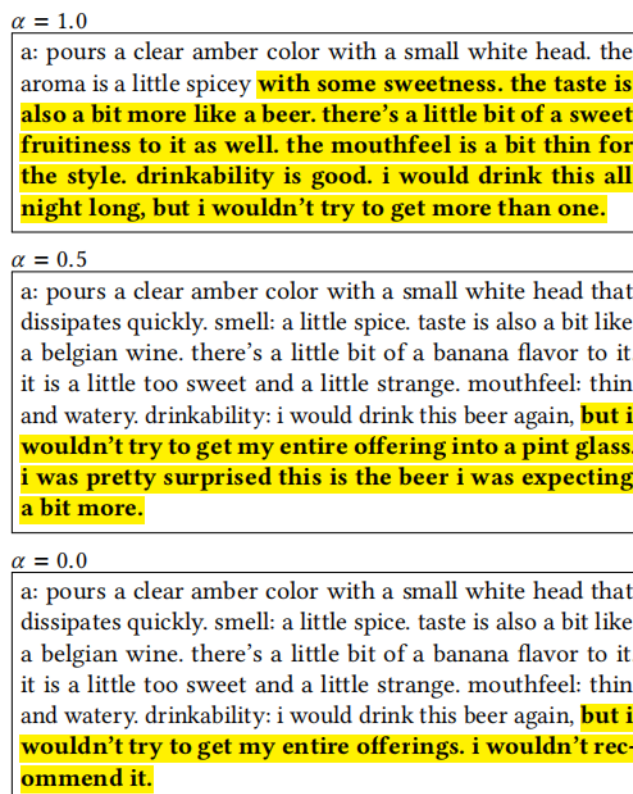


图 18: ETF 模型生成的旅馆标签词云

也有比较少见的用于物品画像的例子，比如阿里巴巴团队提出的使用多模态生成对抗网络生成移动电子商务中短产品标题的例子 [24]。由于商家通常倾向于描述冗余和信息过多的产品标题来吸引客户的注意，所以在有限的手机屏幕上简明地显示简短的产品标题是很重要的。他

们采用基于多模态生成对抗网络 (MM-GAN) 的短产品标题生成，创新地结合了来自产品的图像信息和属性标签，以及来自原始长标题的文本信息，现在已经部署到了淘宝。

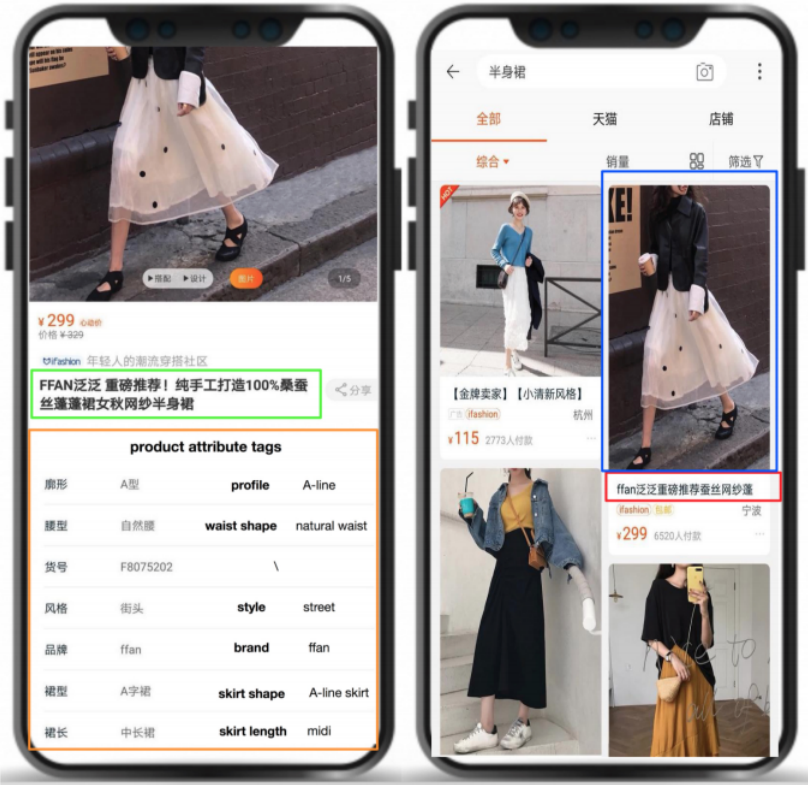


图 19: 短标题生成的例子

5.2 情感分析

用户评论中总是带有很明显的主观情感色彩，典型的论文 [8] 中建议从这些评论中提取元组 (用户，主题，情感态度)，这对自然语言处理技术有一定的要求，如分词、词性还原、词性提取等。

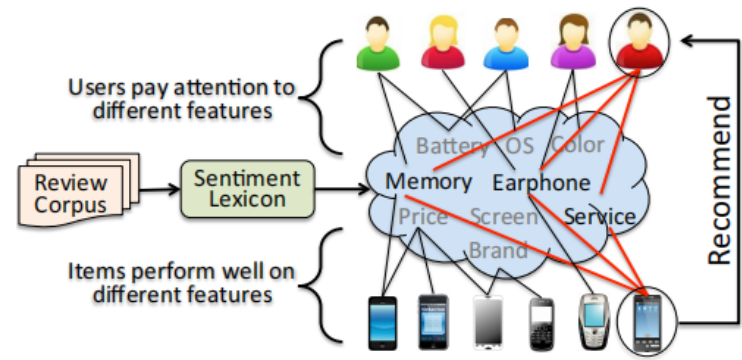


图 20: 从用户评论中提取用户对手机产品的特征兴趣

6 展望

互联网上的文本数据正在日益不断地增长，工业界对可解释推荐系统的需求也在逐渐提升，总结了前人的研究成果可以看到，该领域还有很多值得深入挖掘和学习的地方，在这里笔者也提出一些简单的想法，也期待在以后的学习和研究中有机会落实。

- 可解释推荐系统的优势在于用户可以直观理解为什么被推荐了某些物品，而有些隐变量模型虽然直观上难以解释却能达到很好的推荐效果，也就是说，可解释和准确性二者之间有时候会存在矛盾，如何平衡和取舍是算法工程师们应该解决的问题。
- 对于重要信息来源——用户评论，往往需要更好地利用。例如淘宝网、大众点评等主流电商平台目前的用户评论信息更多地被用于评判商家的好坏（或许更直接地只提取评分，用户评论只是供其他用户参考）而没有与用户偏好联系上，这是一种文本资源的浪费。
- 在实际应用场景中，用户评论往往有很多“噪声”，比如用户为了刷经验、商家为了刷好评，会出现很多没有意义的评论，即使去掉了停用词，这些噪声也很难降低，如何甄别有效评论是一个重要的问题。
- 目前的推荐往往针对的是用户的兴趣推荐其感兴趣的物品，那么反向地，在刻画用户画像时是否可以使用相同的模型去给用户反感的物品类型建模，这样可以在用户浏览时屏蔽到其不感兴趣甚至讨厌的物品，从而增强用户体验感（比如笔者不喜欢看养生类新闻，永远都不会点开看，但浏览器经常会推送奇怪的养生新闻）
- 近年来推荐系统的隐私问题常常为人们诟病，信息固然重要，但哪些是用户隐私，哪些可以对外开放，比如用户的评论信息可以用来建模，但商家为了更准确地推荐提取用户私人信息如性别、年龄等，这种行为是否属于侵犯个人隐私，是合理参考利用和不合理的窥取这些都有待商榷和解决。
- 目前的推荐系统对时间的考量往往是基于纵向的用户兴趣变化，笔者认为还有一种引入时间的角度可以用来把握推荐效果的好坏。比如系统给用户推荐了一条新闻，用户可能对推荐结果并不满意，打开以后就立刻退出了，或许我们可以在后台记录用户的浏览时间并与该用户的平均浏览时间相比较，如果明显小于该时间也可以认为推荐效果不好，接着提取该新闻中的主题特征，将其从用户偏好中去掉。

7 总结

本文通过回顾近年来的一些论文系统化总结了可解释推荐系统领域常用的模型和文本处理技术，并总结了模型的评估方法和一些常用的应用场景，还提出了自己的一些展望和想法，深感学到了很多知识并对学术前沿进展有了一定的把握，总体来说是很有意义的一次作业任务，笔者希望在以后的学习和研究中能更加深入地了解这些模型并争取复现。

参考文献

- [1] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42: 30-37.
- [2] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. J. Mach. Learn. Res., 2003, 3(null):993–1022.
- [3] HOFMANN T. Probabilistic latent semantic indexing[C/OL]//SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Association for Computing Machinery, 1999: 50–57. <https://doi.org/10.1145/312624.312649>.
- [4] MCAULEY J, LESKOVEC J. Hidden factors and hidden topics: Understanding rating dimensions with review text [J/OL]. 2013:165–172. <https://doi.org/10.1145/2507157.2507163>.
- [5] TAN Y, ZHANG M, LIU Y, et al. Rating-boosted latent topics: Understanding users and items with ratings and reviews [C]//IJCAI'16: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. [S.l.]: AAAI Press, 2016: 2640–2646.
- [6] CHENG Z, CHANG X, ZHU L, et al. Mmalfm: Explainable recommendation by leveraging reviews and images[J/OL]. ACM Trans. Inf. Syst., 2019, 37(2). <https://doi.org/10.1145/3291060>.
- [7] WU Y, ESTER M. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering [C/OL]//WSDM '15: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. New York, NY, USA: Association for Computing Machinery, 2015: 199–208. <https://doi.org/10.1145/2684822.2685291>.
- [8] ZHANG Y, LAI G, ZHANG M, et al. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis[C/OL]//SIGIR '14: . New York, NY, USA: Association for Computing Machinery, 2014: 83–92. <https://doi.org/10.1145/2600428.2609579>.
- [9] REN Z, LIANG S, LI P, et al. Social collaborative viewpoint regression with explainable recommendations[C/OL]//WSDM '17: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. New York, NY, USA: Association for Computing Machinery, 2017: 485–494. <https://doi.org/10.1145/3018661.3018686>.
- [10] MIKOLOV T, CHEN K, CORRADO G S, et al. Efficient estimation of word representations in vector space[C/OL]//2013. <http://arxiv.org/abs/1301.3781>.
- [11] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]//NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. Red Hook, NY, USA: Curran Associates Inc., 2013: 3111–3119.
- [12] BARKAN O, KOENIGSTEIN N. Item2vec: Neural item embedding for collaborative filtering[J]. 2015.
- [13] SEO S, HUANG J, YANG H, et al. Interpretable convolutional neural networks with dual local and global attention for review rating prediction[C/OL]//RecSys '17: Proceedings of the Eleventh ACM Conference on Recommender Systems. New York, NY, USA: Association for Computing Machinery, 2017: 297–305. <https://doi.org/10.1145/3109859.3109890>.
- [14] WU L, QUAN C, LI C, et al. A context-aware user-item representation learning for item recommendation[J/OL]. ACM Trans. Inf. Syst., 2019, 37(2). <https://doi.org/10.1145/3298988>.
- [15] LU Y, DONG R, SMYTH B. Coevolutionary recommendation model: Mutual learning between ratings and reviews [C/OL]//WWW '18: Proceedings of the 2018 World Wide Web Conference. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018: 773–782. <https://doi.org/10.1145/3178876.3186158>.
- [16] LI P, WANG Z, REN Z, et al. Neural rating regression with abstractive tips generation for recommendation[C/OL]//SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Association for Computing Machinery, 2017: 345–354. <https://doi.org/10.1145/3077136.3080822>.
- [17] COSTA F, OUYANG S, DOLOG P, et al. Automatic generation of natural language explanations[C/OL]//IUI '18 Companion: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion. New York, NY, USA: Association for Computing Machinery, 2018. <https://doi.org/10.1145/3180308.3180366>.
- [18] CHEN Z, WANG X, XIE X, et al. Co-attentive multi-task learning for explainable recommendation[C/OL]//Proceedings

- of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, 2019: 2137-2143. <https://doi.org/10.24963/ijcai.2019/296>.
- [19] CHANG S, HARPER F M, TERVEEN L G. Crowd-based personalized natural language explanations for recommendations[C/OL]//RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems. New York, NY, USA: Association for Computing Machinery, 2016: 175-182. <https://doi.org/10.1145/2959100.2959153>.
- [20] 张伟. 智能推荐系统课程 PPT9-graph-based[J]. 2021.
- [21] CATHERINE R, MAZAITIS K, ESKÉNAZI M, et al. Explainable entity-based recommendations with knowledge graphs [J]. ArXiv, 2017, abs/1707.05254.
- [22] AI Q, AZIZI V, CHEN X, et al. Learning heterogeneous knowledge base embeddings for explainable recommendation [J/OL]. Algorithms, 2018, 11(9). <https://www.mdpi.com/1999-4893/11/9/137>. DOI: 10.3390/a11090137.
- [23] MA W, ZHANG M, CAO Y, et al. Jointly learning explainable rules for recommendation with knowledge graph[C/OL]// WWW '19: The World Wide Web Conference. New York, NY, USA: Association for Computing Machinery, 2019: 1210-1221. <https://doi.org/10.1145/3308558.3313607>.
- [24] ZHANG J G, ZOU P, LI Z, et al. Multi-modal generative adversarial network for short product title generation in mobile e-commerce[J]. 2019.
- [25] ZHAO K, CONG G, YUAN Q, et al. Sar: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews[J]. 2015 IEEE 31st International Conference on Data Engineering, 2015:675-686.