

融合自注意力机制和提示学习的个性化可解释推荐算法^①



吴永庆, 刘 霄

(辽宁工程技术大学 软件学院, 葫芦岛 125105)

通信作者: 吴永庆, E-mail: yqwuyywu@163.com

摘 要: 在个性化可解释推荐系统中, 用户 ID 是实现个性化的重要标识符. 现有的算法通常采用 encoder-decoder 架构来生成个性化可解释的推荐, 然而这种方法增加了算法的复杂性和计算成本, 限制了算法的精度表现. 为了解决这一问题, 本文提出了一个融合自注意力机制和提示学习的个性化可解释推荐算法 (PERSP). 该算法通过在 BERT 的输入层引入提示学习并对其进行微调, 以增强算法的可解释性. 为了克服 BERT 无法直接使用用户 ID 进行个性化推荐, 该算法利用自注意力机制将用户 ID 与其他命令进行拼接, 将拼接后的序列输入到 BERT 的输入层中进行训练和推理. 为了验证该算法的有效性, 在 TripAdvisor、Amazon 和 Yelp 等数据集上进行对比实验. 在 TripAdvisor 数据集上, PERSP 算法相比其他基线算法, RMSE 和 MAE 分别提升了 3.7% 和 4.7%; 在 Amazon 数据集上, 提升了 1.05% 和 4.1%; 在 Yelp 数据集上, 提升了 1% 和 2.5%. 结果表明该算法在个性化可解释推荐任务中具有较好的性能表现, 有效提升了推荐系统的准确性和可解释性.

关键词: 推荐系统; 自注意力机制; 自然语言算法; 图神经网络; 个性化生成; 提示学习

引用格式: 吴永庆, 刘霄. 融合自注意力机制和提示学习的个性化可解释推荐算法. 计算机系统应用. <http://www.c-s-a.org.cn/1003-3254/9844.html>

Personalized Explainable Recommendation Algorithm Integrating Self-attention Mechanism and Prompt Learning

WU Yong-Qing, LIU Xiao

(Software College, Liaoning Technical University, Huludao 125105, China)

Abstract: In personalized explainable recommendation systems, user ID is an important identifier for personalization. Existing algorithms usually adopt the encoder-decoder architecture to generate personalized explainable recommendations, however, this approach increases the complexity and computational cost of the algorithm and limits the accuracy performance of the algorithm. To address this problem, this study proposes a personalized explainable recommendation algorithm (PERSP) that incorporates self-attention mechanism and prompt learning. The algorithm enhances the interpretability of the algorithm by introducing and fine-tuning prompt learning in the input layer of BERT. To overcome the inability of BERT to directly use user IDs for personalized recommendations, the algorithm uses a self-attentive mechanism to splice user IDs with other commands and feeds the sequences into the input layer of BERT for training and inference. To verify the effectiveness of the algorithm, comparative experiments are conducted on TripAdvisor, Amazon, and Yelp datasets. On the TripAdvisor dataset, the PERSP algorithm improves the root mean squared error (RMSE) and mean absolute error (MAE) by 3.7% and 4.7%, respectively, compared to other baseline algorithms; on the Amazon dataset, the improvements are 1.05% and 4.1% respectively; and on the Yelp dataset, the

^① 基金项目: 国家自然科学基金 (52174184)

收稿时间: 2024-10-22; 修改时间: 2024-11-12; 采用时间: 2024-12-04; csa 在线出版时间: 2025-03-24

improvements are 1% and 2.5% respectively. The results show that the algorithm has better performance in personalized explainable recommendation tasks, effectively improving the accuracy and interpretability of recommendation systems.

Key words: recommender systems; self-attention mechanisms; natural language algorithms; graph neural network (GNN); personalized generation; prompt learning

推荐系统在当今众多在线应用程序中发挥着至关重要的作用. 旨在为用户提供个性化推荐, 以满足其兴趣需求, 帮助用户应对信息过载问题. 它们能帮助用户做出更好更快的决策, 增强用户对系统的信任^[1,2]. 解释性推荐算法多种多样, 例如预定义模板^[3-5]、突出显示的图像区域^[6]和自动生成的句子^[7]等. 文本解释生成算法可大致分为3类: 基于模板的方法、基于检索的方法和基于自然语言生成的方法. 第1类基于模板的算法, 通过定义一个或多个句子模板并根据相应的推荐用不同的单词填充它们来生成不同的解释. 这些词语可能包括从用户评论中提取的项目特征术语, 与目标用户的兴趣一致. 然而, 基于模板的方法需要手动定义模板, 这种方法限制了算法表达解释的能力. 第2类基于检索的方法, 旨在从用户评论中检索特定句子作为推荐解释, 从而增强表达句子的多样性. 然而, 这些解释仅限于现有的句子, 妨碍了新解释的产生并导致句子多样性有限. 第3类, 基于自然语言生成的方法, 由于句子结构的灵活性, 近年来引起了研究人员的极大关注. 这些方法旨在从用户评论中自主学习, 以生成灵活的自由文本解释. 在这些算法中, 区分一个用户或项目 ID 与多个用户或项目 ID 是实现个性化推荐的关键. 换言之, 解释性推荐系统的目标是为推荐的项目向用户提供解释, 以展示推荐内容与用户兴趣的相关性.

近年来, 大型预训练算法的语言建模能力越来越强, 广泛应用于解释性推荐系统, 例如 BERT (bidirectional encoder representation from Transformer)^[8], GPT-2 (generative pre-trained Transformer 2)^[9], GPT-3 (generative pre-trained Transformer 3)^[10]和 Transformer^[11]. 以 BERT 为例, 其卓越的语言建模能力为各种任务提供了强大支持. 然而, 在个性化自然语言生成中, 用户 ID 和项目 ID 在区分不同用户和产品方面起着关键作用, 因此 ID 是实现个性化的标识符. 在个性化可解释性推荐系统中, 如何处理 ID 变得至关重要. 传统 encoder-decoder 架构通过使用 MLP (multilayer perceptron)^[12]将用户 ID 和项目 ID 转换为向量, 再用 RNN (recurrent neural

network)^[13]解码为文本解释. 但是 encoder-decoder 架构由于其结构复杂, 在可解释性上不如 BERT. 如果直接将用户 ID、项目 ID 和命令输入到 BERT 算法中, 就会导致生成的解释重复, 这是因为 BERT 算法并不具备个性化, 无法使用个性化的重要标识符 (用户和项目 ID) 生成解释. 解决此问题的方法是利用自注意力机制, 自注意力机制^[14-16]具备关注序列中不同位置的相关性来提取上下文信息, 并为不同位置之间的关系分配权重的能力. 在个性化任务中, 自注意力机制的动态权重分配有助于算法适应不同的上下文条件, 捕捉个性化特征. 因此, 通过自注意力机制将用户和项目 ID 以序列的形式输入到 BERT 算法中, 解决 BERT 无法实现个性化解释的问题.

提示学习 (prompt learning)^[17]通过设计适当的提示使算法适配下游任务, 被认为是一种能够充分利用算法内部知识的方法, 已被证明在多种自然语言处理任务中具有显著效果. 通过在 BERT 的输入层添加提示学习, 可以更好地捕捉用户的个性化需求和偏好, 从而提高 BERT 的个性化程度, 为用户提供更加符合其喜好的推荐结果. 并且添加提示学习可以帮助推荐系统更好地理解用户需求和偏好, 从而提升推荐结果的解释性. 用户可以更清晰地了解为什么会得到某些推荐结果, 增强用户对推荐系统的信任感和使用满意度.

基于上述分析, 本文提出了一个融合自注意力机制和提示学习的个性化可解释推荐算法 (personalized explainable recommendation algorithms incorporating self-attention mechanisms and prompt learning, PERSP), 该算法通过用户和项目 ID 生产解释评估. PERSP 算法的创新之处如下.

(1) 通过引入提示学习, 可以帮助 BERT 算法提高读取序列的效率, 解决算法的弱泛化问题, 从而减少数据冗余.

(2) 构建自注意力编码器, 将用户 ID、项目 ID、特征和解释融合成序列并输入到 BERT 中, 使算法具备个性化能力.

(3) 本文算法不仅根据文本质量指标评估生成解释的质量, 还从项目特征的角度评估可解释性. 实验表明, 该算法优于其他基线算法.

1 相关工作

1.1 可解释推荐

可解释推荐从人机交互和机器学习两个主要角度进行了研究. 从人机交互的角度, 研究关注人们如何感知不同风格的解释; 而从机器学习的角度, 通过设计新的可解释推荐算法提供解释. 然而, 以前的解释依赖于 RNN 及一些预训练算法, 如: LSTM (long short-term memory)^[18], GRU (gated recurrent unit)^[19], 利用预先训练的大型语言算法应用于最近的 PEPLER^[20]算法, 该算法利用 GPT-2 生成解释. PETER^[21]基于 Transformer 架构, 联合生成用于评级预测和文本生成的深层特征表示. 该算法可以在没有项目特征的情况下进行训练并提供解释. 此外, 使用项目特征的 PETER+算法可以对生成的解释实现更好的个性化. CER^[22]算法也使用预先训练的大型语言算法, 通过使用 GPT-2 来生成解释. 但是, 潜在的更有效的预训练算法 BERT 尚未得到充分探索.

1.2 自注意力机制

注意力机制已成为必须捕获全局依赖性的算法的组成部分, 特别是自注意力, 也称为内部注意力, 通过关注同一序列中的所有位置来计算序列中某个位置的响应. 可解释推荐算法可以使用自注意力算法来实现最先进的结果. 将自注意力添加到用于解释生成的推荐算法中, 将其形式化为一种非局部操作, 用于对用户序列中的时空依赖性进行建模. 例如, Att2Seq^[23]提出了一种注意力机制的属性到序列算法, 用于为给定的属性信息生成解释. 自注意力机制在可解释推荐算法中具有多种优点, 包括提高算法透明度、增强推荐结果解释性、提升个性化推荐效果以及支持多维度推荐.

1.3 个性化生成

个性化在许多任务中发挥着至关重要的作用, 例如可解释的推荐、评论摘要和对话系统. 在这些任务中, 用户和项目 ID 是个性化的重要标识符. 这种策略可以在许多应用中找到, 例如 NETE^[24]算法使用有关项目功能的信息来个性化解释. 该算法使用循环单元生成文本, 其内部工作原理可以解释为神经句子模板的生成. 此外, 处理异构数据 (即 ID) 的适当解决方案

尚未发明, 之前使用自注意力机制或用于个性化自然语言生成算法采用将 ID 替换为文本片段的方式去生成个性化解释. 然而, 这些算法不适合已经在大量原始文本上训练过的预训练算法.

1.4 BERT 算法

BERT 是一种预训练算法, 它采用 Transformer 的编码器部分. BERT 仅使用编码器来训练, 可以同时捕捉句子中的上下文信息, 从而实现双向 (即从左到右和从右到左) 的上下文理解能力. 这使得 BERT 能够更好地理解句子的含义, 提高自然语言处理任务和生成解释的性能. 后来的工作表明, 在预训练加微调的范式下, BERT 的有效性已广泛自然语言理解任务上得到证实, 最近的事实证明, 在增加算法大小和训练语料库的数量后, 该算法能够执行训练期间未针对的新任务. 然而, 重新训练此类算法对于没有大量计算资源的研究人员来说可能并不友好. 因此, 出现了一个新的研究方向: 提示学习. 研究人员将任务适应预先训练的算法, 而不需要修改或重新训练它们. 该算法通过引入用户信息来提供高质量的推荐解释, 从而改善用户体验.

1.5 提示学习

提示学习受到大规模预训练语言算法的深刻启发. 这些预训练语言算法在应对语言任务时, 展现出卓越能力. 在此基础上, 研究人员经过深入探索发现, 倘若能够巧妙设计特定的“提示”, 便可以引导这些算法朝着期望的方向生成相应的结果, 发挥它们的优势服务于具体应用场景. 值得一提的是, 提示学习还具备一项十分重要的微调功能. 在部分特定的情况下, 借助这一功能可以对预训练算法进行针对性的微调操作, 使其在特定任务上的表现得以进一步提升, 更好地适应不同任务的差异化需求. 例如, PEPLER 算法就是通过在 GPT-2 中融入提示学习, 并利用其微调功能, 实现对算法的优化调整, 进而有效增强算法整体的可解释性, 为该算法在更多应用场景中推广和使用创造了有利条件.

2 算法

在输入阶段, 算法将用户 ID、项目 ID、特征 $F_{u,i}$ 和解释 $E_{u,i}$ 以向量的形式输入到自注意力编码器中. 然后使用自注意力编码器将向量连接起来形成一个序列 $S_{L,r} = [u, i, F_1, \dots, F_{|F_{u,i}|}, E_1, \dots, E_{|E_{u,i}|}], F_1, \dots, F_{|F_{u,i}|}$, 它是经过关联权重调整后的编码表示. 在序列 $S_{L,r}$ 中 $F_1, \dots, F_{|F_{u,i}|}$ 表示特征, $E_1, \dots, E_{|E_{u,i}|}$ 表示解释中的命令序

列. 序列 $S_{L,r}$ 进入 BERT 的输入层, 被分割成不同类型的词块, 具体涵盖了用户 ID 词块、项目 ID 词块、特征词块及解释词块. 待这些词块离开输入层后, 便进入提示学习环节. 在提示学习阶段, 出于适配后续各类操作、提升计算效率及增强整个流程稳定性等多方面的考量, 会专门针对用户 ID 词块和项目 ID 词块的向量维度展开相应调整工作. 通过这些调整, 不仅有助于提高多词块之间协同预测的能力, 让各个词块在后续协同配合时能够更加顺畅、高效地发挥作用, 进而使预测结果的精准度得以提升; 而且还能保证在后续各种计算及相互比较的操作过程中, 数据所呈现出的特征更具合理性与规范性, 有效避免因数值范围方面存在的问题, 而对词块所蕴含的丰富语义信息造成理解上的偏差, 从而确保可以准确、全面地利用这些语义信息, 为后续任务的开展奠定良好基础. 而经过提示学习精心调试后的这些词块, 会被输送至 BERT 输出层当中. 输出层依托其多层网络结构, 会持续地对输入的词块进行编码处理. 为执行后续相关任务提供有力支撑.

2.1 解释与推荐

序列 $S_{L,r}$ 通过 BERT 的最终表示为序列 $X_L = [X_L, \dots, X_{L,s}]$ 后, 将执行两个任务: 解释生成任务和上下文预测任务. 如图 1 所示.

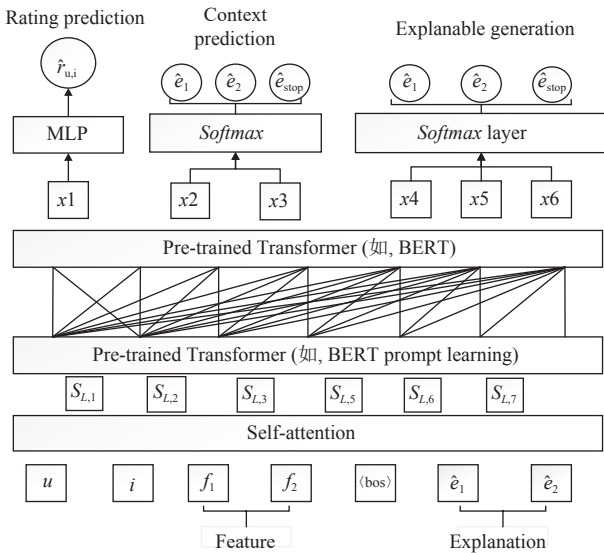


图 1 融合自注意力机制和提示学习的个性化可解释推荐算法

为了个性化解释生成任务, 设计了上下文预测任务. 对于这两个任务, 将对每个标记的最终表示应用线性层, 以将其映射到大小为 $|V|$ 的向量. 例如, 经过这一层后, $X_{L,t}$ 将变为 c_t . 通过使用 BERT, PERSP 可以更好地

处理序列数据, 提高算法的性能和泛化能力. 针对每个标记应用线性层, 可以更有效地将其映射到目标空间, 从而实现更准确的个性化解释生成和上下文预测任务.

$$c_t = \text{Softmax}(W^v S_{L,r} + b^v) \quad (1)$$

其中, W^v 属于 $R^{|V| \times d}$ 且 b^v 属于 $R^{|V|}$ 是权重参数. 向量 c_t 表示词汇表 V 上的概率分布, 可以从中采样概率为 c_t^e .

2.2 解释生成

PERSP 采用负对数似然 (NLL) 作为解释任务的损失函数, 并计算训练集中用户项目对的平均值:

$$L_e = -\frac{1}{|T|} \sum_{(u,i) \in T} \frac{1}{E_{u,i}} \sum_{t=1}^{E_{u,i}} \log \text{Softmax}(W_e x_{|F_{u,i}|+j+2} + b_e)_{e_j} \quad (2)$$

其中, T 是训练数据集, W_e, b_e 是 Softmax 层的权重, e_j 是解释 $E_{u,i}$ 的第 j 个标记. 注意, $|F_{u,i}|$ 的索引 $+j+2$ 只是从输出序列 X 中选择 e_j 的表示形式.

在测试阶段, u, i 和 $F_{u,i}$ 一起为算法提供一个特殊的序列开始标记 ($\langle \text{bos} \rangle$). 根据生成的概率分布 $C_{\langle \text{bos} \rangle}$ 预测命令, 算法可以逐步生成解释. 然后, 采用简单的贪婪解码方法, 即选择具有最大概率的命令进行采样, 并将其连接到序列的末尾以生成下一个命令. 通过重复这一过程, 直到算法生成了特殊的序列结束标记 ($\langle \text{eos} \rangle$), 或者生成的解释 $\hat{E}_{u,i}$ 达到预先定义的长度.

2.3 文本生成

在机器学习中, 上下文预测是一种辅助任务, 它通过仅基于用户和项目的表示来估计解释中所有出现单词的概率分布. 这个任务的目标是加强预测的解释与用户和项目的表示之间的联系, 从而生成更加个性化的解释. 在提出的模型中, 使用 Softmax 层计算输出的概率分布, 并通过交叉熵优化来训练模型.

$$L_c = -\frac{1}{|T|} \sum_{(u,i) \in T} \frac{1}{E_{u,i}} \sum_{t=1}^{E_{u,i}} \log \text{Softmax}(W_c x_t + b_c)_{e_j} \quad (3)$$

其中, T 是训练数据集, W_c 和 b_c 是可学习权重, e_j 是黄金标准解释 $E_{u,i}$ 的第 j 个标记.

2.4 评论生成

推荐系统可以被视为一个预测问题, 其目标是根据用户 u 和项目 i 的 ID 来预测分数 $\hat{r}_{u,i}$. 由于序列中 u 和项目 i 在序列中可以相互关注, 因此它们的最终表示捕获了它们之间的交互. 接着, 将第 1 个表示 $X_{L,1}$ 映射为标量, (因为第 2 个表示会被用于上下文预测). 为

实现这一步,使用了一个具有多个隐藏层的多层感知器 (MLP):

$$\hat{r}_{u,i} = w^r \sigma(W^r S_{L,1} + b^r) + b^r \quad (4)$$

其中, $w^r \in R^{|\mathcal{V}| \times d}$, $W^r \in R^{1 \times d}$, $b^r \in R$ 为权重参数, $\sigma(\cdot)$ 为 Sigmoid 函数. 由此可见,对 BERT 做推荐和解释都是可行的. 对于此任务,PERSP 使用均方根误差 (RMSE):

$$L_r = \frac{1}{|T|} \sum_{(u,i) \in T} (r_{u,i} - \hat{r}_{u,i})^2 \quad (5)$$

其中, $r_{u,i}$ 是真实评级.

2.5 多任务学习

最后,将这 3 个任务集成到一个多任务学习框架中,其目标函数定义为:

$$J = \min_{\theta} (\lambda_e L_e + \lambda_c L_c + \lambda_r L_r) \quad (6)$$

其中, θ 表示算法中所有可训练的参数, λ_e 、 λ_c 和 λ_r 是平衡不同任务学习的正则化权重. 通过这种方式,可以以端到端的方式高效训练算法.

3 实验与分析

3.1 数据集

在实验部分,PERSP 使用了 3 个公开可用的可解释推荐数据集,分别是 TripAdvisor、Amazon 和 Yelp. 数据集中的每条记录都由用户 ID、项目 ID、评级、解释和特征组成. 解释是从用户评论中提取的句子,具体数据见表 1,每个解释至少包含一个项目特征. 为了公平比较,该算法遵循最初引入 PETER 方法的实验设置,包括与相同基线进行比较、采用相同的质量指标. 在对比实验方面,PERSP 区别在于在实验中额外添加了 PEPLER 和 CER.

表 1 实验数据集的统计数据

属性	TripAdvisor	Amazon	Yelp
#user	9765	7506	27147
#items	6280	7360	20266
#records	320023	441783	129347
#features	5069	5399	7340
#records/user	32.77	58.86	47.64
#records/item	59.66	60.02	63.81
#words explanation	13.01	14.14	12.32

3.2 参数设置

为了将实验环节所涉及的算法都调至最优结果进行公平对比,针对当前数据集进行多次实验,选择如下参数设置:针对部分算法训练参数,将 Emsize 大小设

置为 512,并将学习率设置为 0.5;将其隐藏层数设置为 2,每个隐藏层神经元个数为 200,dropout 比率设置为 1.0;推荐任务的正则化参数为 1.0;上下文预测任务的正则化参数为 1.0. 实验配置情况见表 2. 在本文所提出算法中,选取负对数似然 (NLL) 以及均方误差 (MSE) 这两种函数作为损失函数,以此来衡量算法预测结果与真实值间的差异,进而为后续模型优化提供依据,助力算法性能不断提升.

表 2 实验配置

Parameter	Value
CPU	I7-12700H
GPU	NVIDIA GeForce RTX 3070 Ti 8 GB
Language	Python 3.9.12
Operating system	Windows 11
Deep learning framework	PyTorch 1.11.0 + CUDA 11.3

3.3 对比试验

为验证 PERSP 算法的有效性,本文在 TripAdvisor, Amazon 和 Yelp 这 3 个数据集上对比 PERSP 算法与其他可解释推荐算法. 所选择对比算法为学术界和工业界最常使用的可解释推荐算法,具体如下.

(1) Transformer^[25]: 运用预训练算法 Transformer, 通过将用户和项目 ID 视为命令来执行解释生成任务.

(2) Att2Seq^[23]: 通过学习文本数据集中的产品属性和评论之间的关联,来实现从给定产品属性生成符合这些属性的产品评论.

(3) PEPLER^[20]: 通过学习用户应该如何解释推荐结果,来提高推荐系统的可解释性. 该算法不仅能个性化地生成解释性提示,还可以与传统推荐算法进行整合,从而提高推荐系统的准确性和解释性.

(4) ACMLM^[26]: 利用大规模评论数据来验证推荐结果的方法,这种算法可以帮助推荐系统提高推荐结果的准确性和个性化程度,提升用户满意度.

(5) NETE^[24]: 通过将用户行为和推荐结果编码成一个神经模板,来生成用户友好的解释.

(6) NRT^[27]: 采用了一种神经网络算法来预测用户对商品或服务的评分,并生成关于如何改善用户体验的实用建议. 使用神经网络进行训练和预测,为用户提供更好的推荐和建议.

(7) PETER^[21]: 通过使用小型的未经预训练的 Transformer,同时为了弥合用户和项目 ID 与命令之间的差距引入了一项名为“上下文预测”的附加任务来生成解释评论.

(8) PETER+[21]: 个性化 Transformer 算法结合了自注意力机制和特征交互, 旨在提高推荐结果的解释性, 使用户更容易理解和接受推荐结果.

(9) CER[22]: 通过对推荐解释的语义和逻辑结构进行建模, 以及提出一些算法来帮助系统生成一致性解释.

本文采用多样化的指标集合来评估生成解释的可解释性和文本质量以及推荐性能. 使用平均绝对误差 (MAE) 和均方根误差 (RMSE) 来衡量推荐算法性能. 以 BLEU[28]和 ROUGE[29]等指标对生成的解释与数据集中提供的参考进行比较来评估文本质量. 具体地说, 通过测量 BLEU-1 (B1)、BLEU-4 (B4)、ROUGE-1 (R1) 和 ROUGE-2 (R2) 的精度 Precision (P)、召回率 Recall (R) 和分数 F-score (F). 额外添加了 3 个表示算法解释性能的指标, 特征匹配率 (FMR)、特征覆盖率 (FCR)

和特征多样性 (DIV). 提示学习作为一种针对预训练算法 BERT 的微调手段, 在利用相应数据集开展训练的整个过程中, 它自身也同步接受着训练, 从而不断优化自身性能, 使其能更好地适配并提升 BERT 在各类自然语言处理任务中的表现.

3.4 实验分析

PERSP 算法与其他可解释性算法对比的具体实验结果见表 3. 可以看出, 与每个数据集上的最优结果相比, Yelp 数据集上 3 个表示算法解释性能的指标 FMR、FCR 和 DIV 分别提升了 3%、13% 和 16%; Amazon 数据集上 FMR、FCR 和 DIV 分别提升了 3%、16% 和 9%; TripAdvisor 数据集上 FMR、FCR 和 DIV 分别提升了 2%、15% 和 13%. 除此之外, 在文本质量上 PERSP 算法也有着巨大的提升. 综上所述, PERSP 算法在生成解释和文本质量方面要优于其他基线算法.

表 3 PERSP 与其他算法在可解释性和文本质量上的对比实验

Dataset	Model	Explainability			Text quality								
		FMR	FCR	DIV↓	USR	B1	B4	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F
Yelp	Transformer	0.06	0.06	2.46	0.01	7.39	0.42	17.18	10.29	12.56	1.71	0.92	1.09
	NRT	0.07	0.11	2.37	0.12	11.66	0.65	17.69	12.11	13.55	1.76	1.22	1.33
	Att2Seq	0.07	0.12	2.41	0.13	10.29	0.58	18.73	11.28	13.29	1.85	1.14	1.31
	PEPLER	0.08	0.3	1.52	0.35	11.23	0.73	17.51	12.55	13.53	1.86	1.42	1.46
	ACMLM	0.05	0.31	0.95	0.95	7.01	0.24	7.89	7.54	6.82	0.44	0.48	0.39
	NETE	0.80	0.27	1.48	0.52	19.31	2.69	33.98	22.51	25.56	8.93	5.54	6.33
	PETER	0.08	0.19	1.54	0.13	10.77	0.73	18.54	12.20	13.77	2.02	1.38	1.49
	PETER+	0.87	0.38	1.08	0.34	20.80	3.43	35.44	26.12	27.95	10.65	7.44	7.94
	CER	0.86	0.37	1.08	0.30	20.62	3.42	35.51	26.03	27.92	10.74	7.43	7.97
PERSP	0.89	0.42	0.91	0.41	21.17	3.48	35.71	26.39	27.98	10.96	7.51	8.06	
Amazon	Transformer	0.10	0.01	3.26	0.00	0.91	0.59	19.68	11.94	14.11	2.10	1.39	1.55
	NRT	0.12	0.07	2.93	0.17	12.93	0.96	21.03	13.57	15.56	2.71	1.84	2.05
	Att2Seq	0.12	0.20	2.74	0.33	12.56	0.95	20.79	13.31	15.35	2.62	1.78	1.99
	PEPLER	0.11	0.27	2.06	0.38	13.19	1.05	18.51	14.16	14.57	2.36	1.88	1.91
	ACMLM	0.10	0.31	2.07	0.96	9.52	0.22	11.65	10.39	9.69	0.71	0.81	0.64
	NETE	0.71	0.19	1.93	0.57	18.76	2.47	33.87	21.43	24.81	7.58	4.77	5.46
	PETER	0.12	0.21	1.74	0.29	12.77	1.17	19.81	13.80	15.23	2.08	2.08	2.20
	PETER+	0.77	0.31	1.20	0.46	19.75	3.06	34.71	23.99	26.35	9.04	6.23	6.71
	CER	0.78	0.31	1.24	0.44	19.88	3.12	34.98	24.22	26.60	9.24	6.37	3.86
PERSP	0.80	0.36	1.13	0.58	20.54	3.16	35.19	24.49	26.86	9.36	6.45	6.95	
TripAdvisor	Transformer	0.04	0.00	10.00	0.00	12.79	0.71	16.52	16.38	15.88	2.22	2.63	2.34
	NRT	0.06	0.09	4.27	0.08	15.05	0.99	18.22	14.39	15.40	2.29	1.98	2.01
	Att2Seq	0.06	0.15	4.32	0.17	15.27	1.03	18.97	14.72	15.92	2.40	2.03	2.09
	PEPLER	0.07	0.21	2.71	0.24	15.49	1.09	19.48	15.67	16.24	2.48	2.21	2.16
	ACMLM	0.07	0.41	0.78	0.94	3.45	0.02	4.86	3.82	3.72	0.18	2.20	0.16
	NETE	0.78	0.27	2.22	0.57	22.39	3.66	35.68	24.86	27.71	10.20	6.98	7.66
	PETER	0.07	0.13	2.95	0.08	15.96	1.11	19.07	16.09	16.48	2.33	2.17	2.09
	PETER+	0.89	0.35	1.61	0.25	24.32	4.55	37.48	29.21	30.49	11.92	8.48	9.24
	CER	0.88	0.39	1.62	0.32	24.66	4.61	37.04	29.34	30.42	11.84	9.02	9.24
PERSP	0.90	0.45	1.41	0.46	25.03	4.56	37.43	29.26	30.76	12.05	9.10	9.28	

自动评估的平均结果以及每次单独分类器运行的结果见表 4. 分析 Yelp 数据集获得的分数, 可以清楚地

看到预测评分与 PETER+和 CER 生成的解释之间的一致性差异. 10 个评估模型中的每 1 个都表明 PERSP

比 PETER+架构生成更连贯的解释.

3.5 参数实验

3.5.1 学习率对算法效果的影响

学习率在深度学习算法中扮演着关键的角色,它控制着算法参数在训练过程中的更新速度.学习率的选择直接影响着算法的训练效果和性能.设置过大的学习率会导致算法难以收敛,而设置过小则可能无法获得全局最优解.因此,在 TripAdvisor 数据集上进行实验,让学习率分别取值为 0.1、0.3、0.5、0.8、1.0.如图 2 所示,当学习率取值为 0.5 时获得了最优解.

3.5.2 Batch-size 大小对算法效果的影响

在训练神经网络时, Batch-size 指的是每次迭代传入网络的样本数量.合理选择 Batch-size 对算法的训练效果和性能具有重要影响.设置过大的 Batch-size 会导致算法训练速度变慢,内存占用增加,收敛速度减慢,从而降低泛化性能.相反,设置过小的 Batch-size 也会导致问题.因此,在 TripAdvisor 数据集上进行实验,让 Batch-size 分别取值 32、64、128、256、512.如图 3 所示,当 Batch-size 取值为 128 时获得了最优解.

表 4 对所选方法的解释和预测评级的自动一致性评估的结果 (%)

Dataset	Evaluated architecture	Evaluating model										
		1	2	3	4	5	6	7	8	9	10	Mean
Yelp	PETER+	81.16	87.53	77.73	88.41	86.4	88.43	81.77	89.05	91.02	50.55	82.21
	PERSP	78.64	85.58	76.69	85.3	82.3	84.16	79.5	84.6	87.83	48.22	78.45
	Gold standard	90.17	94.72	88.72	95.28	93.83	84.93	90.8	95.96	96.5	73.89	91.48
Amazon	PETER+	86.24	55.81	88.94	47.65	37.33	78.49	73.87	62.97	63.24	83.13	67.77
	PERSP	82.39	50.21	83.29	44.27	35.06	74.18	70.12	60.6	59.62	79.25	64.38
	Gold standard	89.31	57.82	92.03	48.56	36.18	79.78	77.71	65.3	67.82	86.13	70.07
TripAdvosor	PETER+	92.84	93.99	90.02	84.33	95.28	91.1	91.37	92.6	79.13	94.81	90.55
	PERSP	87.49	88.65	74.63	80.26	90.41	85.43	86.88	88.41	75.56	88.84	84.27
	Gold standard	87.71	89.77	85.35	78.88	91.76	84.13	81.52	88.06	69.1	91.72	84.78

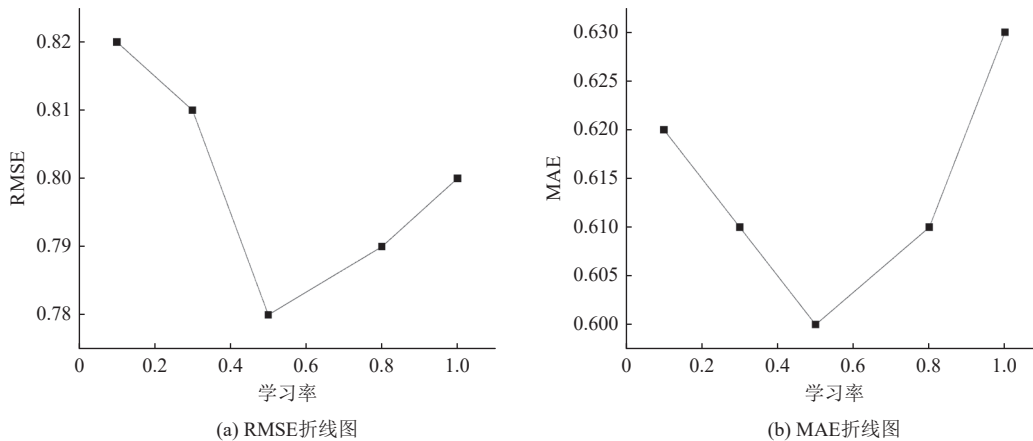


图 2 学习率对算法性能的影响

3.5.3 梯度裁剪对算法的影响

梯度裁剪 (gradient clipping) 是深度学习中常用的技术,用于控制梯度的大小,避免梯度爆炸 (gradient explosion) 或梯度消失 (gradient vanishing) 的问题.在训练神经网络时,梯度过大可能导致算法不稳定,梯度裁剪可以有效解决这个问题.因此,在 TripAdvisor 数据集上进行实验,让梯度分别取值为 0.1、0.5、1.0、1.5、2.0.如图 4 所示,当梯度取值为 1.0 时获得最优解.

3.5.4 Emsize 对实验算法的影响

Emsize 指的是嵌入向量的大小,在实验中需要进行调整.较小的 Emsize 可能无法充分捕捉单词的语义信息,导致算法表现不佳;而较大的 Emsize 可能会增加算法复杂度,导致过拟合或训练速度变慢.因此,选择合适的 Emsize 至关重要.在 TripAdvisor 数据集上,本文使 Emsize 分别取值为 64、128、256、512、1024 进行实验.如图 5, Emsize 取值为 512 时获得了最优解.

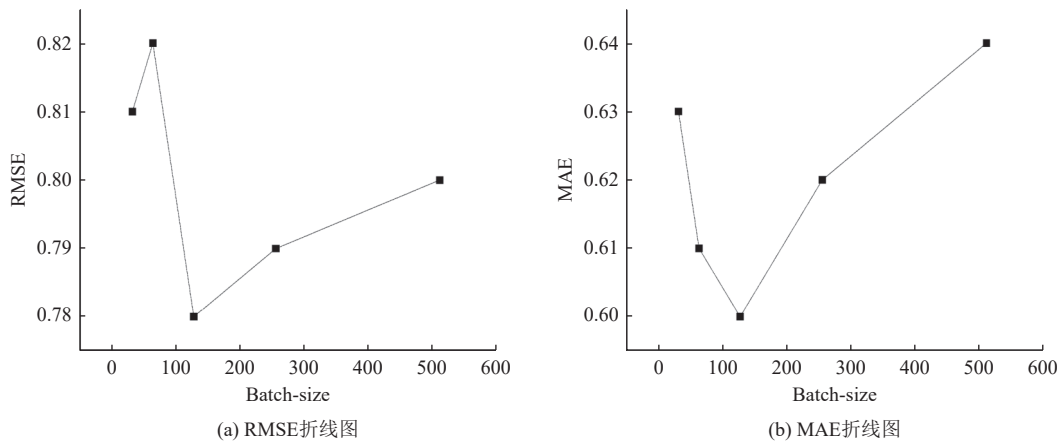


图3 Batch-size 对算法性能的影响

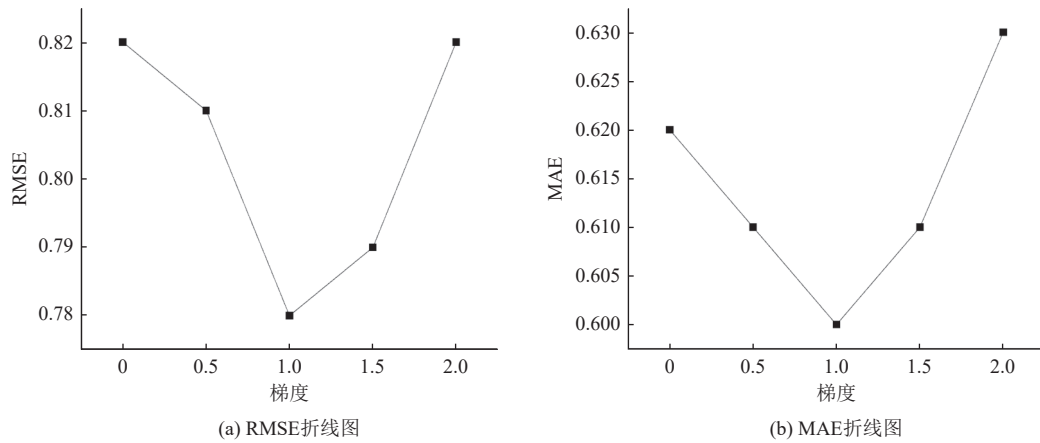


图4 梯度裁剪对算法性能的影响

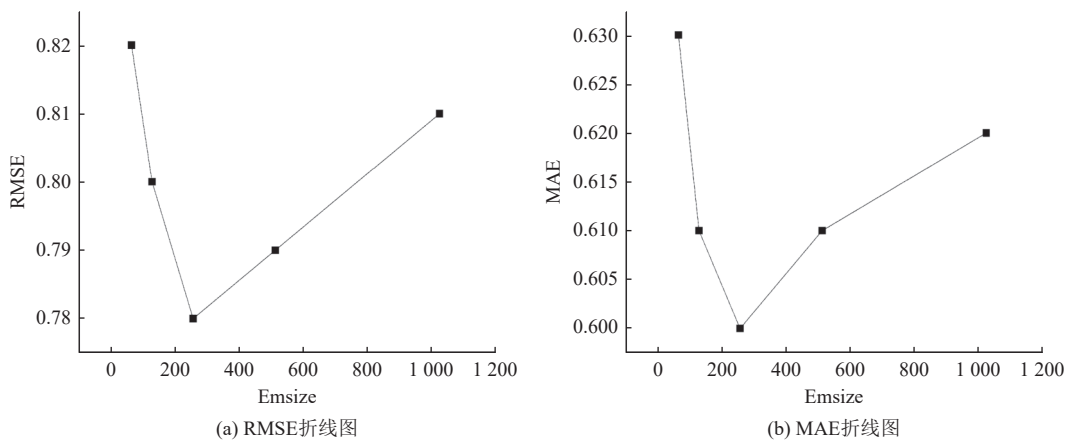


图5 Emsize 对算法性能的影响

3.6 消融实验

本节中, 为了验证自注意力机制、BERT 算法和提示学习对可解释性的影响, 在 TripAdvisor 数据集上对 PERSP 算法设计了以下消融实验.

(1) Disable 自注意力机制: 将自注意力机制从

PERSP 算法删除.

(2) Disable BERT: 将 BERT 从 PERSP 算法删除.

(3) Disable 提示学习: 将 BERT 的提示学习删除.

去除 BERT 后, 自注意力机制在处理大型数据集时会出现计算复杂度升高, 使算法可解释性降低. 去除

BERT 中的提示学习后,降低了算法性能. 无论去掉自注意力机制、BERT 或是提示学习都无法取得最优效

果. 具体实验结果见表 5, 实验表明, 本文所提出的模块在可解释性上有着极好的效果.

表 5 PERSP 算法的消融实验

Model	Explainability			Text quality			Recommendation	
	FMR	FCR	DIV	USR	BLEU-1	BLEU-4	RMSE	MAE
Disable自注意力机制	0.05	0.20	1.60	0.78	0.05	0.80	0.81	0.62
Disable BERT	0.05	0.41	2.14	0.81	0.05	0.80	3.22	3.08
Disable 提示学习	0.04	0.39	2.54	0.85	20.24	3.56	0.83	0.65
PERSP	0.89	0.42	1.41	0.46	25.03	4.56	0.79	0.60

3.7 性能描述

本文提出 PERSP 算法在推荐性能上与经典的不可解释推荐算法进行了比较, PMF^[30]是一种标准的概率矩阵分解方法, 通过潜在因素来表征用户和项目; SVD++^[31]利用用户的交互项目来增强潜在因素.

PERSP 算法性能对比结果见表 6. 由此可见, 本文提出的 PERSP 算法能够有效增强推荐算法在可解释性方面的预估准确性. 通过使用自注意力机制形成序列. 将序列输入到 BERT 算法的提示学习中, 增加 BERT 算法的效率, 解决其弱泛化能力, 减少数据冗余. 综上所述, 本文提出的 PERSP 算法优于其他基线算法, 这验证了本文所提出的通过使用自注意力机制和提示学习在个性化可解释性方面能够为用户提供更为准确的解释.

表 6 RMSE 和 MAE 表示的推荐性能比较

Model	Yelp		Amazon		TripAdvisor	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
PMF	1.09	0.88	1.03	0.81	0.87	0.70
SVD++	1.01	0.78	0.96	0.72	0.80	0.61
NRT	1.01	0.78	0.95	0.70	0.79	0.61
NETE	1.01	0.79	0.96	0.73	0.79	0.60
PETER	1.01	0.79	0.95	0.71	0.81	0.62
CER	1.01	0.79	0.95	0.72	0.81	0.63
PERSP	1.00	0.77	0.94	0.69	0.78	0.60

4 结语

针对 BERT 不具备个性化能力的问题, 本文提出了融合自注意力机制和提示学习的个性化可解释推荐算法. 该算法首先利用自注意力机制将用户 ID、项目 ID、特征和解释融合成序列, 将序列输入到 BERT 算法, 通过在 BERT 算法中引入提示学习并对其进行微调. 最后, 使用 BEER 算法生成解释. 实验结果表明, PERSP 算法在生成高质量的解释上明显优于其他基线算法, 验证了算法的有效性. 未来研究将考虑在可解释性领域引入多模态, 以进一步增强算法的可解释性.

参考文献

- 白鹭, 鲁思琪, 信昆仑, 等. 神经网络与解释模型在非结冰期含沙量预测中的应用. 计算机系统应用, 2023, 32(12): 276–283. [doi: 10.15888/j.cnki.csa.009338]
- Zhang YF, Chen X. Explainable recommendation: A survey and new perspectives. Foundations and Trends® in Information Retrieval, 2020, 14(1): 1–101.
- Li L, Chen L, Dong RH. CAESAR: Context-aware explanation based on supervised attention for service recommendations. Journal of Intelligent Information Systems, 2021, 57(1): 147–170. [doi: 10.1007/s10844-020-00631-8]
- Tan JT, Xu SY, Ge YQ, *et al.* Counterfactual explainable recommendation. Proceedings of the 30th ACM International Conference on Information and Knowledge Management. ACM, 2021. 1784–1793.
- 周晓敏, 滕飞, 张艺. 基于元网络的自动国际疾病分类编码模型. 计算机应用, 2023, 43(9): 2721–2726.
- Chen H, Chen XX, Xu HT, *et al.* Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris: ACM, 2019. 765–774.
- Chen HX, Chen X, Shi SY, *et al.* Generate natural language explanations for recommendation. arXiv:2101.03392, 2021.
- 胥桂仙, 刘兰寅, 王家诚, 等. 基于 BERT 和超图对偶注意力网络的文本情感分析. 计算机应用研究, 2024, 41(3): 786–793.
- Hanna M, Liu O, Variengien A. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans: ACM, 2023. 3322.
- Brown TB, Mann B, Ryder N, *et al.* Language models are few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: ACM, 2020. 159.

- 11 Han K, Xiao A, Wu EH, *et al.* Transformer in Transformer. Proceedings of the 35th International Conference on Neural Information Processing Systems. ACM, 2021. 1217.
- 12 Tolstokhin IO, Houlsby N, Kolesnikov A, *et al.* MLP-mixer: An all-MLP architecture for vision. Proceedings of the 35th International Conference on Neural Information Processing Systems. ACM, 2021. 1857.
- 13 Dhruv P, Naskar S. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. Proceedings of the 2020 International Conference on Machine Learning and Information Processing. Singapore: Springer, 2020. 367–381.
- 14 孟繁林, 何晓曦, 刘应浒, 等. 基于自注意力机制的点云分类分割. 计算机系统应用, 2024, 33(1): 177–184. [doi: [10.15888/j.cnki.csa.009341](https://doi.org/10.15888/j.cnki.csa.009341)]
- 15 葛芸, 马琳, 叶发茂, 等. 基于多尺度池化和范数注意力机制的遥感图像检索. 电子与信息学报, 2022, 44(2): 543–551. [doi: [10.11999/JEIT210052](https://doi.org/10.11999/JEIT210052)]
- 16 Zhao HS, Jia JY, Koltun V. Exploring self-attention for image recognition. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 10073–10082.
- 17 Zhou KY, Yang JK, Loy CC, *et al.* Conditional prompt learning for vision-language models. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 16795–16804.
- 18 Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 2020, 404: 132306. [doi: [10.1016/j.physd.2019.132306](https://doi.org/10.1016/j.physd.2019.132306)]
- 19 Cahuantzi R, Chen XY, Guttel S. A comparison of LSTM and GRU networks for learning symbolic sequences. Proceedings of the 2023 Computing Conference on Intelligent Computing. Cham: Springer, 2023. 771–785.
- 20 Lei L, Zhang YF, Chen L. Personalized prompt learning for explainable recommendation. ACM Transactions on Information Systems, 2023, 41(4): 103.
- 21 Lei L, Zhang YF, Chen L. Personalized Transformer for explainable recommendation. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL, 2021. 4947–4957.
- 22 Raczynski J, Lango M, Stefanowski J. The problem of coherence in natural language explanations of recommendations. Proceedings of the 26th European Conference on Artificial Intelligence. Kraków: ECAI, 2023. 1922–1929.
- 23 Dong L, Huang SH, Wei FR, *et al.* Learning to generate product reviews from attributes. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long studys. Valencia: ACL, 2017. 623–632.
- 24 Li L, Zhang YF, Chen L. Generate neural template explanations for recommendation. Proceedings of the 29th ACM International Conference on Information & Knowledge Management. ACM, 2020. 755–764.
- 25 Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: ACM, 2017. 6000–6010.
- 26 Ni JM, Li JC, McAuley J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 188–197.
- 27 Li PJ, Wang ZH, Ren ZC, *et al.* Neural rating regression with abstractive tips generation for recommendation. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Shinjuku: ACM, 2017. 345–354.
- 28 Papineni K, Roukos S, Ward T, *et al.* BLEU: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia: ACL, 2002. 311–318.
- 29 Lin CY. ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out. Barcelona: ACL, 2004. 74–81.
- 30 Salakhutdinov R, Mbihi A. Probabilistic matrix factorization. Proceedings of the 21st International Conference on Neural Information Processing Systems. Vancouver: ACM, 2007. 1257–1264.
- 31 Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas: ACM, 2008. 426–434.

(校对责编: 王欣欣)