计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330133
61(5): 1276−1289, 2024

基于集合效用边际贡献学习的可解释薪酬预测算法

孙 莹[1] 章玉婷[2,6] 庄福振[3] 祝恒书[4] 何 清[5,6] 熊 辉[1]

[1](香港科技大学(广州)人工智能学域 广州 511458)

[2](中国科学院计算技术研究所专项技术研究中心 北京 100190)

[3](北京航空航天大学人工智能研究院 北京 100191)

[4](BOSS直聘职业科学实验室 北京 100028)

[5](中国科学院智能信息处理重点实验室(中国科学院计算技术研究所) 北京 100190)

[6](中国科学院大学 北京 101408)

(yings@hkust-gz.edu.cn)

Interpretable Salary Prediction Algorithm Based on Set Utility Marginal Contribution Learning

Sun Ying[1], Zhang Yuting[2,6], Zhuang Fuzhen[3], Zhu Hengshu[4], He Qing[5,6], and Xiong Hui[1]

[1](*Thrust of Artificial Intelligence*, *The Hong Kong University of Science and Technology* (*Guangzhou*), *Guangzhou* 511458)

[2](*Special Technology Research Center*, *Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100190)

[3](*Institute of Artificial Intelligence*, *Beihang University*, *Beijing* 100191)

[4](*Career Science Lab*, *BOSS Zhipin*, *Beijing* 100028)

[5](*CAS Key Laboratory of Intelligent Information Processing* (*Institute of Computing Technology*, *Chinese Academy of Sciences*), *Beijing* 100190)

[6](*University of Chinese Academy of Sciences*, *Beijing* 101408)

Abstract Accurately quantifying the relationship between skills and salary is essential to improve reasonable job salary setting and promote talent attraction and retention. However, the relationship between skills and salary is complex because it involves modeling set utility in a high-dimensional space with massive possible elements. Deep neural networks offer a new solution for complex fitting problems. However, for skill-based fine-grained salary prediction, there still lacks interpretable neural networks that can effectively model set utility under the influence of complex variables. To address this issue, we propose a marginal contribution-based incremental set utility network (MCISUN). MCISUN models the marginal contribution of elements when they are added to the set. In this way, the set utility can be naturally obtained in a flexible and interpretable way. In particular, rather than relying on pooling structures to ensure permutation invariance, MCISUN constructs order-sensitive intermediate results through recurrent attention neural networks and takes advantage of the sets' permutation invariance property to achieve data augmentation, thus improving the model's robustness. We conduct extensive experiments on a real-world large-scale salary dataset. The experimental results show that MCISUN outperforms state-of-the-art models by 30% for skill-based salary prediction. Qualitative experiments show that our model can recognize reasonable skill contribution values and capture the relationship between skills.

Key words set utility modeling; marginal contribution; salary prediction; neural network; interpretability

收稿日期: 2023-03-10; **修回日期**: 2023-07-26

基金项目: 国家自然科学基金项目 (62176014, 61836013); 广州市科技计划市校联合资助项目 (2023A03J0141); 中央高校基本科研业务费专项资金
This work was supported by the National Natural Science Foundation of China (62176014, 61836013), the City-University Joint Funding Project of Guangzhou Science and Technology Plan (2023A03J0141), and the Fundamental Research Funds for the Central Universities.

摘　要　知识技能对薪酬影响作用视为一种多变量影响下高维元素集合的效用建模问题. 深度神经网络为解决复杂问题提供了新的机遇, 但针对知识导向的细粒度薪酬预测问题, 仍缺乏能够对复杂变量影响下的集合效用进行准确、可解释建模的神经网络结构. 为此, 提出一种基于边际贡献的增量式集合效用网络(marginal contribution-based incremental set utility network, MCISUN) 来拟合元素加入时的效用增量, 从而灵活且可解释地建模集合效用. 区别于以往基于池化层的排列不变性建模算法, MCISUN 构建顺序敏感的中间结果, 利用集合的排列不变性实现数据增强, 有效提升模型数据效率及泛化性. 最后, 大规模真实薪酬数据上的实验结果表明所提模型在基于技能的薪酬预测任务上比最先进的（state-of-the-art, SOTA）模型效果提升超过 30%. 同时, 定性实验证明模型能够为技能设置合理的贡献值且发现技能间的关联.

关键词　集合效用建模; 边际贡献; 薪酬预测; 神经网络; 可解释性

中图法分类号　TP391

精准的人才薪酬预测有助于合理化薪酬激励, 促进人才吸引和保留. 基于机器学习的薪酬预测技术近些年引起了学术界广泛关注, 决策树、支持向量机等经典算法可实现基于基本统计特征的薪酬预测[1-4], 但无法建模知识技能的影响. 基于自然语言处理技术[5] 提取岗位描述表征可隐式考虑能力因素[6] 以提升薪酬预测效果, 但依然无法明确建模具体知识技能粒度的薪酬影响. 事实上, 随着劳动力市场转向知识导向的薪酬定价, 技能对薪酬的影响越发重要. 因此, 一些研究提取技能并基于技能集的分类、聚类实现薪酬预测[7-8], 但这些模型表达能力有限, 难以精准量化技能对薪酬的影响作用.

知识技能集与薪酬具有复杂关联. 事实上, 知识技能驱动的薪酬预测可视为一种多变量影响下高维元素空间内复杂集合的效用建模问题. 人才薪酬由其拥有的众多技能共同决定, 技能在相互影响下产生组合增益、边际效应递减等复杂组合效应. 除此之外, 城市、公司、时间等工作场景变量也与技能及薪酬产生复杂交互, 同一技能在不同场景下产生差异化的薪酬作用.

深度神经网络技术的发展使模型具备了从高维输入到输出的高表达性变换能力, 为细粒度量化知识技能的薪酬影响提供了新的机遇. 然而针对薪酬预测问题, 现有神经网络难以有效处理集合数据排列不变性、稀疏性、元素数量不确定性等特点, 欠缺在复杂影响下进行准确集合效用建模的能力. 此外, 现有神经网络内部复杂的连接导致其缺乏解释性, 各元素对集合效用的贡献难以量化, 造成管理者难以理解模型预测依据, 从而降低了薪酬激励决策过程中的模型可用性.

为细粒度量化技能集与薪酬的关系, Sun 等人[9] 提出了一种薪酬-技能价值组成网络 (salary-skill value composition network, SSCN), 可在不同工作场景下将工作技能价值组合为岗位薪酬. 但该工作只关注提取独立的全局技能价值, 将薪酬影响简化为技能价值的加权平均, 忽略了技能集复杂的组合效应. 到目前为止, 仍缺乏一种能够对复杂变量影响下的集合效用进行准确、可解释建模的神经网络结构, 从而支撑有效的薪酬预测.

为此, 本文从薪酬预测问题出发, 将对集合效用施加影响的变量视为边信息, 定义一种边信息作用下的集合效用建模问题, 提出一种基于边际贡献的增量式集合效用网络 (marginal contribution-based incremental set utility network, MCISUN). 通过建模元素顺序加入集合时的边际贡献灵活拟合集合效用, 从而在精准建模的同时解释各元素对输出所产生的贡献. 特别地, 区别于以往利用池化结构保证排列不变性的集合建模算法, MCISUN 通过循环注意力神经网络建模元素边际贡献, 构建对加入顺序敏感的中间结果, 再利用排列不变性实现数据增强, 从而提升模型泛化性. 本文所提模型可以量化各技能所产生的薪酬增益, 并实现可解释、准确的薪酬预测. 最后, 大规模真实薪酬数据上的实验结果表明, 所提模型在知识技能导向的薪酬预测任务上比 SOTA 模型效果提升超过 30%. 同时, 定性实验证明所提模型可合理评估技能贡献并发现技能间的关联关系. 主要贡献总结为 4 个方面:

1) 考虑知识技能及场景信息对薪酬的影响, 将薪酬预测问题定义为边信息作用下的集合效用建模问题, 并提出一种基于深度集合建模的解决方案.

2) 针对边信息作用下的集合效用建模问题, 提出一种基于边际贡献建模的增量式集合效用学习框架, 能够直接量化对集合所产生的贡献, 实现可解释的集合效用建模.

3) 提出一种基于循环注意力机制的顺序敏感网络结构进行集合效用建模，通过构建合理训练策略实现排列不变的集合效用建模，此做法可提升效用建模的泛化性.

4) 在真实数据集上开展大量实验，验证了所提模型在现实薪酬预测任务上的有效性，并证明所提模型的良好解释性，可支持有效的技能薪酬影响分析.

1 相关工作

本节介绍相关工作，包括深度集合模型及薪酬预测.

1.1 深度集合模型

针对集合类型数据的建模[10]是近年来深度学习领域的一个热门研究方向. 不同于网格结构数据建模，集合模型需要满足排列不变性并能处理变长输入. 针对集合数据表征问题，Vinyals 等人[11]利用注意力机制来整合变长输入信息并满足了排列不变性，针对集合数据提出了序列到序列框架的扩展模型，Zaheer 等人[10]进一步将集合数据整合并提出统一的深度集合框架来处理集合数据输入.Lee 等人[12]在此基础上进一步提出了 Set Transformer，基于多头自注意力机制捕捉集合中元素间成对或者更复杂的交互关系. 近些年出现了特征级别排序池化(featurewise sort pooling, FSPool)[13]、Janossy Pooling[14]、注意力聚合(attentional aggregation, AttSets)[15]等针对不同任务场景的集合表征学习算法，进一步提升了模型表达能力和效率. 例如针对集合-集合映射任务，Satio 等人[16]提出一种可有效提取集合间的关联性的特征转换层.Zhang 等人[17]基于所提取集合表征学习置换矩阵以实现集合-序列映射. 本文研究与现有深度集合模型的区别主要包括 3 点：1)现有工作关注于集合表征的独立建模，忽略了集合在不同边信息下的语义和作用变化，本文研究关注边信息影响下的元素语义提取及集合效用建模.2) 区别于现有结构以网络归纳偏置实现排列不变性的做法，本文研究通过元素边际贡献建模构建排列敏感的中间结果并提出针对集合排列不变性的训练策略，更有效提升模型泛化性.3)现有深度集合模型大多把元素组合为集合的过程看作黑箱，无法量化各元素对于集合的作用. 不同于现有工作，本文研究提出一种基于增量式集合效用模型，可直接量化元素对集合的贡献.

1.2 基于机器学习的薪酬预测

基于机器学习的薪酬预测技术近些年引起了学界关注，决策树、支持向量机等经典算法很早就被用于基于人口统计[1-3]、工作环境[4]等特征的薪酬预测模型训练. 贝叶斯模型也被广泛用于薪酬分布建模[18-19]，例如领英公司基于贝叶斯分层平滑算法，利用薪酬分位数、均值等薪酬统计信息建模公司薪酬分布[6]. 这些算法仅考虑基本特征，忽略了知识技能的影响. 自然语言处理技术的发展提高了模型对工作内容的理解能力，一些研究基于词频-逆向文件频率(term frequency–inverse document frequency, TF-IDF)、Doc2-vec、隐含狄利克雷分布(latent Dirichlet allocation, LDA)等文本特征提取算法，从岗位描述中提取特征向量以训练薪酬预测模型[5-6,20]，基于深度学习技术，Meng 等人[21-22]利用 Word2vec 实现基于文本描述的岗位聚合并利用矩阵分解、贝叶斯模型等技术实现薪酬基准预测，Wang 等人[23]利用双向门控循环单元(bidirectional gate recurrent unit, Bi-GRU)处理招聘文本特征以实现端到端的薪酬预测. 这些工作隐式考虑了能力因素对薪酬的影响，但依然无法明确建模知识技能粒度的薪酬影响. 一些研究对知识技能信息提取进行了探索，并通过针对技能集的分类、聚类算法实现薪酬预测[7-8]，例如 More 等人[7]提出一种基于情感分析技术的技能集提取算法并基于线性回归进行薪酬预测. 但现有模型表达能力有限，难以精准量化技能对薪酬的影响作用. 不同于现有工作，本文研究提出一种基于深度集合建模的薪酬预测算法，可以有效建模技能集在复杂场景影响下的组合作用及其对薪酬的复杂影响.

2 问题定义

2.1 薪酬预测及技能贡献解释

基于知识技能的薪酬预测问题旨在预测技能集在不同场景因素下所对应的薪酬. 本文将样本J_j的技能集表示为$S_j=\left\{\left(s^i,lv^i\right)\right\}_{i=1}^{N_j}$，其中$N_j$代表岗位$J_j$的技能数量，$s^i$表示第$i$个工作技能(如“C++”)，$lv^i$表示对第$i$个工作技能的掌握程度(如“熟悉”). 场景因素表示为集合C，包含时间、工作地点、工作环境等影响知识技能所发挥作用的变量. 基于技能的薪酬预测任务表述为给定样本集$D_{\text{train}}=\left\{\left\langle C_j,S_j,Y_j\right\rangle\right\}_{j=1}^{N}$，如何训练模型$f_\theta$，使得输入场景信息$C'$和技能集$S'$输出合理的薪酬估计值$\tilde{Y}=f_\theta(C',S')$，其中$\theta$表示模型参数. 为直观量化知识技能与薪酬的关联，模型需评估各技能对样本薪酬所产生的贡献，实现可解释的薪酬预测.

2.2 边信息感知的集合效用建模及元素归因

本文所提薪酬预测问题主要关注知识技能与薪酬的关联，因而将工作场景信息视为影响技能作用的边信息，建模不同边信息条件下技能集的效用（即薪酬）. 该问题可抽象为一种通用的边信息影响下的集合效用组成问题，其输入包含边信息集及作为直接影响的元素集，模型对边信息影响下元素的作用进行建模，预测元素组合所得集合效用. 由于现实中经常出现混杂变量与所关注因变量共同影响结果的情况，该问题在诸多现实任务中具备普适性.

3 基于边际贡献的增量式集合效用网络（MCISUN）

本节介绍基于边际贡献的增量式集合效用网络（MCISUN），其结构如图 1 所示. 具体而言，首先介绍边信息-元素交互表征网络，然后介绍增量式集合效用建模过程以及基于循环注意力的顺序敏感边际贡献网络实现，最后介绍针对顺序敏感性的训练策略.

3.1 增量式集合效用建模过程

元素对集合效用的贡献往往存在复杂的组合作用. 一方面，元素间存在组合增益，比如拥有大量复杂技能的人才由于稀缺性，岗位薪酬大幅度提高. 同时，集合扩大存在边际效应递减作用，比如技能耦合和上下位关系导致包含很多高级技能的集合在增加更多低级技能时难以带来显著的薪酬增益.

为此，本文提出一种基于边际贡献的增量式集合效用建模算法，灵活建模新元素对集合效用产生的边际贡献，将各元素顺序加入集合时的边际贡献组合为集合效用. 相比 SSCN[9]，该算法放宽元素与集合的线性关系约束，灵活建模元素组合效应，对于广泛的集合效用建模任务具有更高的普适性.

给定集合 $Z=\{z_i\}_{i=1}^{L}$，其中 L 表示集合中元素的个数，z_i 表示集合中第 i 个元素，X 表示所有可能元素的集合. 将 Z 的效用表示为 $\Phi(Z)$，其中 $\Phi:2^X\to\mathbb{R}$ 表示集合效用函数. 基于集合的预测模型通常直接拟合 Φ 的参数化近似 $\hat{\Phi}(\cdot|\Theta)$，其中 Θ 表示参数. 不同于已有算法，本文新提出一种基于元素边际贡献的模型对 Φ 进行增量式拟合. 具体而言，元素边际贡献定义为在给定前置集合 P 的基础上加入新元素 x 后集合效用的增量，$\Delta(x,P):=\Phi(P\cup\{x\})-\Phi(P)$，其中 $\Delta:\left(X,2^X\right)\to\mathbb{R}$ 表示效用增量函数. 给定集合 Z 的任意一个排列 $O=\left(x_{o_1},x_{o_2},\cdots,x_{o_L}\right)$，计算各元素贡献之和为

$$\Phi(O)=\sum_{i=1}^{L}\Delta\left(x_{o_i},\{x_{o_j}\}_{j=1}^{i-1}\right). \tag{1}$$

通过学习 Δ 的参数近似 $\hat{\Delta}\left(\cdot|\Theta\right)$ 以间接估计 $\hat{\Phi}_\Theta$ 为

$$\hat{\Phi}_\Theta(O)=\sum_{i=1}^{L}\hat{\Delta}\left(x_{o_i},\{x_{o_j}\}_{j=1}^{i-1}|\Theta\right). \tag{2}$$

设 $\pi(Z)$ 表示集合 Z 的所有排列，基于集合效用的排列不变性，可得

$$\forall O\in\pi(Z),\qquad \Phi(Z)=\Phi_\Theta(O). \tag{3}$$

由此，可利用实际集合效用构建损失函数，通过最小化估计效用 $\hat{\Phi}_\Theta(O)$ 与实际效用 $\Phi(O)$ 的差别拟合 $\hat{\Delta}\left(\cdot|\Theta\right)$，从而学习到元素边际贡献模型，记作

$$\arg\min_{\Theta}\ \underset{O_Z\in\pi(Z)}{E}\left[\left(\Phi(Z)-\sum_{i=1}^{L}\hat{\Delta}\left(x_{o_i^Z},\{x_{o_j^Z}\}_{j=1}^{i-1}|\Theta\right)\right)^2\right], \tag{4}$$

其中 $x_{o_i^Z}$ 表示 O_Z 中的第 i 个元素. 图 2 举例说明了元素效用增量模型的主要思想，假设集合效用为其中多

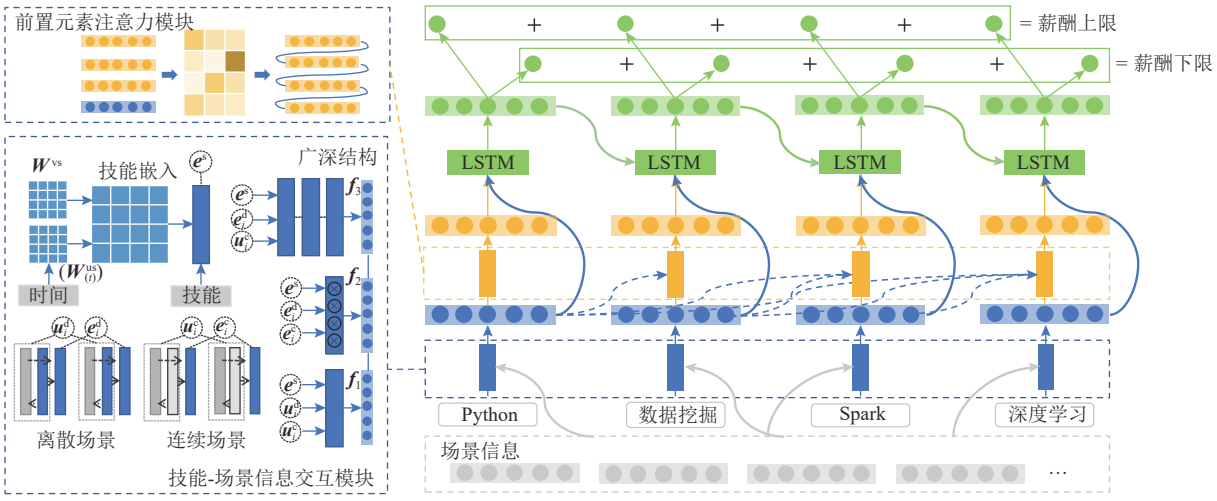

Fig. 1 Overall framework of MCISUN

图 1 MCISUN 总体框架

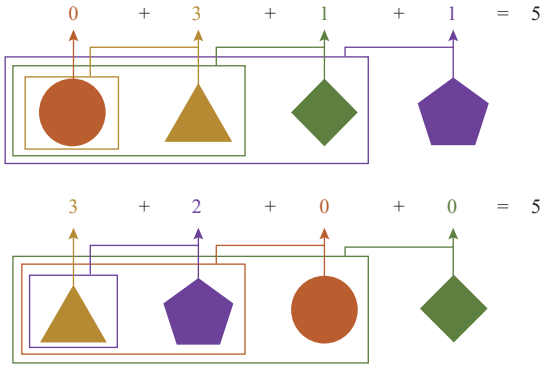

Fig. 2 Main idea of marginal contribution-based incremental set utility modeling

图 2 基于边际贡献的增量式集合效用建模的主要思想

边形的最大边数，不同加入顺序导致各多边形产生边际贡献差异，但最终所得集合效用相等.

3.2 边信息作用下元素表征提取

元素在边信息作用下产生语义变化，导致对集合效用产生不同的作用. 受 SSCN[9] 启发，MCISUN 将边信息融入元素表征提取过程，将样本表示为统一的集合形式 $J_j=\{C_j,s_j^i\}_{i=1}^{N_j}$，以广深结构 (wide-and-deep structure)[24] 取边信息与各元素的多阶交互特征 $\boldsymbol{x}_i=g\left(C_j,s_j^i\right)$ 作为边信息作用下的元素表征向量，并对表征向量集合的效用进行建模. 为方便描述，下文“元素”亦指代元素表征.

具体而言，根据输入形式，模型将边信息分为离散边信息和连续边信息 2 种. 第 i 个离散边信息表示以独热编码的形式输入的城市、时间等信息，第 i 个连续边信息表示公司规模、城市的收入统计等不同域的特征向量. 考虑时间因素的影响，网络以矩阵分解形式学习各元素的时序表征，从而降低模型复杂度并充分建模技能间的语义相关性，形式化写作 $\boldsymbol{E}_{(t)}^s=\boldsymbol{W}_{(t)}^{\mathrm{us}}\boldsymbol{W}^{\mathrm{vs}},\quad t=1,2,\cdots,T$，其中 T 表示数据集中时间段的总个数. $\boldsymbol{E}_{(t)}^s$ 中各行表示不同元素的表征.

基于所学表征，首先进行边信息与元素线性交互特征提取

$$\boldsymbol{f}_1=\sum_i \boldsymbol{W}_i^{\mathrm{cl}}\boldsymbol{u}_i^{\mathrm{c}}+\sum_i \boldsymbol{W}_i^{\mathrm{dl}}\boldsymbol{u}_i^{\mathrm{d}}+\boldsymbol{W}^{\mathrm{sl}}\boldsymbol{e}^{\mathrm{s}}+\boldsymbol{b}^{\mathrm{l}} \tag{5}$$

其中 $\boldsymbol{u}_i^{\mathrm{c}}$ 表示第 i 个连续特征的输入向量，$\boldsymbol{u}_i^{\mathrm{d}}$ 表示第 i 个离散特征的独热编码，$\boldsymbol{e}^{\mathrm{s}}$ 表示元素在当下时间 $\boldsymbol{E}_{(t)}^s$ 中的对应表征. 对于每个连续输入，模型将输入特征向量映射到统一的场景表征空间，写作 $\boldsymbol{e}_i^{\mathrm{c}}=\boldsymbol{u}_i^{\mathrm{c}}\boldsymbol{W}_i^{\mathrm{p}}+\boldsymbol{b}^{\mathrm{p}}$. 而对于每个离散输入，网络学习每个取值的表征，将输入转化为向量表征 $\boldsymbol{e}_i^{\mathrm{d}}=\boldsymbol{u}_i^{\mathrm{d}}\boldsymbol{W}_i^{\mathrm{e}}$. 基于这些表征，网络利用乘性操作提取 2 阶组合特征为

$$\boldsymbol{f}_2=\sum_i\sum_{i\neq j,j}\boldsymbol{e}_i^{\mathrm{c}}\odot\boldsymbol{e}_j^{\mathrm{c}}+\sum_i\sum_{i\neq j,j}\boldsymbol{e}_i^{\mathrm{d}}\odot\boldsymbol{e}_j^{\mathrm{d}}+\sum_i\sum_j\boldsymbol{e}_i^{\mathrm{c}}\odot\boldsymbol{e}_j^{\mathrm{d}}+\boldsymbol{e}^{\mathrm{s}}\left(\sum_i\boldsymbol{e}_i^{\mathrm{c}}+\sum_i\boldsymbol{e}_i^{\mathrm{d}}\right), \tag{6}$$

其中⊙表示元素乘法. 最后利用多层感知机提取高阶特征为

$$\boldsymbol{f}_3=MLP\left(\left(\boldsymbol{u}_1^{\mathrm{c}};\boldsymbol{u}_2^{\mathrm{c}};\cdots;\boldsymbol{e}_1^{\mathrm{d}};\boldsymbol{e}_2^{\mathrm{d}};\cdots;\boldsymbol{e}^{\mathrm{s}}\right)\right). \tag{7}$$

模型将各交互特征连接作为元素表征向量 $\boldsymbol{x}=(\boldsymbol{f}_1;\boldsymbol{f}_2;\boldsymbol{f}_3)$，用于后续效用建模.

3.3 基于循环注意力的边际贡献网络

模型基于所提元素表征对边际贡献进行建模. 为方便描述，将输入序列表示为 $O=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_L)$，并将 $\{\boldsymbol{x}_j\}_{j=1}^{i-1}$ 中的每个元素称为 $\boldsymbol{x}_i$ 的前置元素. 为保证模型对复杂元素贡献的拟合能力，本文提出一种基于循环注意力机制的边际贡献网络. 对于每个输入元素，其综合考虑当前元素、前置元素以及整个加入过程的序列信息来建模元素贡献.

1)基于多头注意力机制的前置表征建模. 元素之间关联性差异导致各前置元素对当前新增元素的影响程度不同. 因此本文研究利用多头注意力机制建模各前置元素影响大小，针对性地提取前置元素集表征. 具体来说，网络以 H 个注意力头学习前置元素集不同侧面的表征. 在第 h 个头的建模中，序列中当前位置 $\boldsymbol{x}_i$ 被用于构建注意力机制中的查询向量 $\boldsymbol{W}_h^{\mathrm{q}}\boldsymbol{x}_i$，同时构建每个前置元素的键值向量 $\boldsymbol{W}_h^{\mathrm{k}}\boldsymbol{x}_j$，利用余弦相似度度量查询和键值之间的相似性 $a_{i,j}^h=(\boldsymbol{W}_h^{\mathrm{q}}\boldsymbol{x}_i)(\boldsymbol{W}_h^{\mathrm{k}}\boldsymbol{x}_j)^{\mathrm{T}}/\sqrt{d_{\mathrm{k}}}$，其中 d_{k} 表示注意力层的维度. 最后对所有前置元素的相似性进行 softmax 操作，得到加和为 1 的注意力分配. 最终为每个元素 $\boldsymbol{x}_i$ 得到 H 个表示前置元素重要性的向量，写作 $\boldsymbol{a}_i^1,\boldsymbol{a}_i^2,\cdots,\boldsymbol{a}_i^H$. 最后构建各个注意力头下各前置元素作为值向量 $\boldsymbol{W}_h^{\mathrm{v}}\boldsymbol{x}_i$，基于重要性在各个头下对值向量进行加权平均，得到 $\boldsymbol{x}_i$ 前置集合表征 $\boldsymbol{p}_i^h$. 将各个头上的前置表征连接作为前置元素的整体表征，写作 $\boldsymbol{P}_i=(\boldsymbol{p}_i^1;\boldsymbol{p}_i^2;\cdots;\boldsymbol{p}_i^H)$. 将集合中各元素与其前置表征连接，构成 $\boldsymbol{xs}_i=(\boldsymbol{x}_i;\boldsymbol{P}_i)$.

2)基于循环神经网络的边际贡献序列建模. 尽管前置元素顺序不影响边际贡献，但在建模过程中前置元素贡献序列可促进当前元素贡献的评估. 具体而言，前置元素加入过程中的边际贡献序列包含集合语义变化动态，其变化存在趋势性和一致性，例如若一段时间内边际贡献持续保持在较低状态，说明边际效应递减明显，此时再加入新元素很可能同

样产生较低增益. 通过对这种变化规律的学习, 模型面对未知样本元素贡献估计时仍可进行符合规律的大致估计, 从而提高模型数据效率及泛化性.

为此, 本文使用长短期记忆网络(long short term memory, LSTM)[25] 元素序列进行顺序敏感的建模, 为输入序列的每个位置抽取序列状态表征 $\boldsymbol{h}^i$, 最终边际贡献建模记作

$$\hat{\Delta}\left(\boldsymbol{x}_i,\{\boldsymbol{x}_j\}_{j=1}^{i-1}|\theta\right)=\sigma\left(\boldsymbol{W}^{\text{fc}}\boldsymbol{h}^i+\boldsymbol{b}^{\text{fc}}\right). \tag{8}$$

3)针对数值区间的边际贡献建模. 由于现实中薪酬通常以非负数值区间形式出现, 本文提出针对数值空间的边际贡献建模方式, 针对集合效用下限及上下限之差分别建模非负边际贡献, 从而保证最终集合效用为合法区间. 在薪酬预测任务中, 由于学习更多技能通常可提高人才薪酬, 因此本文约束元素在任意前置元素集下贡献非负.

3.4 针对顺序敏感性的训练策略

集合效用建模通常需要利用池化等结构直接实现排列不变的归纳偏置, 但 MCISUN 则在建模过程中大量利用序列信息构建对输入顺序敏感的网络中间结果: 1)集合效用的变化过程对顺序敏感, 不同加入顺序导致各元素所产生的边际贡献发生变化; 2) LSTM 模块使得元素贡献对于前置元素加入顺序敏感. 通过参数化地学习满足集合的排列不变性规律, 模型一方面可有效学习元素对不同集合的效用增益以提高效用模型可解释性, 另一方面可基于元素特性针对性地构建确定性排列来提升模型效果. 本文提出 2 种针对元素排列的训练策略.

1)排列不变性数据增强策略. 将训练集中每个集合Z中的元素随机打乱作为模型输入, 由于模型结构对顺序敏感, 不同顺序下的集合在建模中被视为不同样本, 产生不同的模型输出. 但由于集合的排列不变性, 在不同加入顺序下, 各技能贡献之和最终都应等于目标集合效用. 集合排列不变性及模型结构顺序敏感性间的矛盾为网络提供了一个隐含约束, 由于目标函数的优化需要网络稳定建模各排列下的集合效用, 训练集大小由一开始的 $|D_{\text{train}}|$ 扩展为 $\sum_{Z\in D_{\text{train}}}|Z|!$, 降低模型在有限训练数据下过拟合的可能性并提升泛化性. 除此之外, 不同排列中间过程对应差异化的边际贡献, 不合理的贡献建模会导致该顺序下效用的偏差, 因此训练过程中多样化的排列促使网络学习更合理的元素贡献.

2)基于频率的低学习难度排列构建. 除了数据增强策略以外, 在训练数据较少时也可基于元素特点构建确定性偏序, 从而实现集合排列及效用建模的唯一性. 因此, 本文提出一种基于频率的排列构建策略. 具体而言, 训练集中出现次数少的元素由于缺乏足量监督信息, 建模难度更大. 因此, 本文针对训练集稀疏的情况提出一种基于频率的排列构建策略, 将数据集中出现次数少的元素排在前面, 降低其建模过程中的前置集复杂性, 从而减小训练难度. 在确定顺序之下, 假设数据共包含 N_{s} 个元素, 模型需估计的〈前置技能, 新增技能〉二元组的数量由 $O\left(N_{\text{s}}2^{N_{\text{s}}}\right)$ 降至 $\sum_{i=1}^{N_{\text{s}}}2^{i-1}$, 即 $O\left(2^{N_{\text{s}}}\right)$ 个, 可极大降低模型的收敛难度. 此排序在许多任务中具备合理性, 以薪酬预测为例, 样本中出现频率较低的技能往往为稀缺的专业技能, 其相比于基础技能对薪酬产生更大的作用. 将低频元素排在前面, 后续基本技能大概率因为边际效应递减而产生较小贡献, 模型直接基于少量专门技能确定总体薪酬, 可显著降低薪酬估计的复杂程度.

4 实验结果

本节给出 MCISUN 模型的薪酬预测效果验证, 包括总体表现及技能贡献分析.

4.1 实验设置

4.1.1 数据集

本节采用基于技能集的薪酬预测公开数据集[26] 进行实验. 数据集包括来自国内知名互联网招聘平台上采集的来自 13 个城市的 2016 年下半年至 2019 年上半年的招聘数据, 每条数据包括招聘广告中的技能集、工作场景特征及对应薪酬信息. 根据职位和行业, 共划分为信息技术类岗位 (IT) 和设计类岗位 (Designer) 两个数据集. 数据中包含招聘广告信息和公司信息, 其中每条招聘广告包含岗位描述文本、基本信息(如雇主、工作地点、发布时间等)、薪酬范围(单位为人民币), 公司信息包括公司规模、注册资金、成立时间等. 为保证实验结果与相关工作的一致性, 本文采用与 Sun 等人[9] 中相同的数据预处理及特征提取算法, 最终 IT 数据集包含 215 308 个样本, Designer 数据集包含 18 761 个样本, 每个样本包含场景特征集及技能集, 用于模型训练和验证. 其中 IT 技能集合中包含 1 374 个不同的技能, 每条招聘记录中最多包含 40 条技能.Designer 技能集合中包含 138 个不同的技能, 每条招聘记录中最多包含 25 条技能.

4.1.2 模型参数设置

实验中 MCISUN 的主要参数如表 1 所示, 模型

Table 1 Hyper-Parameter Configuration

表 1 超参数设置

参数	参数值	参数	参数值
嵌入大小	128	LSTM 神经元个数	1 024
MLP 层数	3	MLP 隐藏单元	128
注意力头数	16	注意力层维度	64

使用高斯分布[27]初始化训练参数, 利用 Adam[28]算法进行参数优化, 以 Leaky ReLU[29]作为激活函数. 训练中默认使用随机打乱的数据增强策略.

4.1.3 实验环境

本文所进行测试的硬件环境为 2.40 GHz Intel® Xeon®CPU E5-2680 v4 服务器及 500 GB 内存, 深度学习模型训练和测试使用一张 GeForce RTX 2080 Ti GPU 显卡.

4.2 薪酬预测表现

本节展示模型的薪酬预测表现, 包括对比实验、参数实验、数据效率实验、排列消融实验.

4.2.1 对比模型

本实验以现有薪酬预测模型作为对比, 主要包括 6 类算法:

1) 经典回归模型. 包括支持向量机 (support vector machine, SVM)[30]、线性回归 (linear regression, LR)[31]和梯度提升决策树 (gradient boosting decision tree, GBDT)[32]等. 由于这类模型输入为定长向量, 因此将技能集合以多热向量表示, 并与边信息特征连接作为模型输入.

2) 深度神经网络 (deep neural network, DNN)[33]. 将技能与边信息和回归基线模型进行相同处理后, 输入多层全链接网络经过多次变换后进行薪酬预测.

3) 整体薪酬基准矩阵分解 (holistic salary benchmarking matrix factorization, HSBMF)[22]模型 .HSBMF 将岗位按工作内容聚类, 利用矩阵分解预测各岗位的薪酬. 为保证公平性, 使用工作场景信息和技能需求构建正则化矩阵, 从而使 HSBMF 可以考虑工作场景信息.

4) 文本挖掘算法. 将招聘广告视作文本, 并直接基于文本进行薪酬预测. 本类基线包括 2 部分, 第 1 部分直接使用自然语言处理神经网络进行训练, 包括文本卷积神经网络 (text convolutional neural network, TextCNN)[34-35]、分层注意网络 (hierarchical attention network, HAN)[36]及 Transformer-XL[37]. 第 2 部分则在文本预训练模型的基础上进行微调, 包括基于 Transformer 的双向编码表示 (bidirectional encoder representation from Transformers, BERT)[38]、鲁棒优化的 BERT 方法 (a robustly optimized BERT, RoBERTa)[39]和 XLNet[40].

5) SSCN[9]. SSCN 通过建模技能价值和技能支配进行薪酬预测, 是目前效果最好的基于技能集的薪酬预测模型.

6) 消融实验模型. 分别剔除 MCISUN 中的 LSTM 和注意力层, 得到 MCISUN (w/o l) 模型和 MCISUN (w/o a) 模型. 除此之外, 本实验使用 DeepSet 框架[41]替换基于循环注意力的边际贡献网络, 利用边信息-元素表征向量的集合进行端到端的薪酬预测, 得到 MCISUN (DeepSet) 模型.

4.2.2 评价指标

实验选取均方根误差 (root mean square error, *RMSE*)[42]和平均绝对误差 (mean absolute error, *MAE*)[42]作为薪酬预测任务上的性能评价指标, 计算公式为

$$RMSE=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2},\ MAE=\frac{1}{m}\sum_{i=1}^{m}|y_i-\hat{y}_i|, \quad (9)$$

其中 y_i 和 $\hat{y}_i$ 分别表示真实值和预测值.

4.2.3 对比实验

MCISUN 和基线模型在 IT 和 Designer 数据集上的薪酬预测结果分别如表 2 和表 3 所示, 我们在每个模型上重复 10 次留出 (hold out)[43]验证实验, 每次将数据以 4∶1 的比例随机划分为训练集和测试集. 在不同数据集中, *RMSE* 和 *MAE* 的误差结果均以“平均值±标准差”形式列出, 如表 2 和表 3 所示.

4.2.4 参数实验

本实验选取 MCISUN 的 2 个重要参数, 在 IT 数据集上进行不同参数设置下的性能实验, 包括 2 部分:

1) 注意力头数实验. 在其他网络参数不变的情况下分别将注意力头数设置为 4, 8, 16, 32, 并对每组参数进行 10 次留出[43]验证, 实验结果如图 3 所示. 可以发现, 随着注意力头数从 4 提升到 16, *RMSE* 逐渐降低, 而继续提升注意力头数, *RMSE* 升高, 可推测原因为乘法操作复杂度提升导致过拟合. 因此本文选取 16 个注意力头作为默认参数.

2) LSTM 神经元个数实验. 在其他网络参数不变的情况下, 分别将 LSTM 神经元个数设置为 128, 256, 512, 768, 1 024, 并对每组参数进行 10 次留出[43]验证, 实验结果如图 4 所示.观察到 LSTM 复杂度的提升能够显著降低预测误差, 由于 LSTM 复杂度的提高主要提升序列建模效果, 此结果验证了集合建模中加入顺序信息对效果提升的有效性. 随着神经

Table 2 Salary Prediction Errors on IT Dataset

表 2 IT 数据集上薪酬预测误差

模型	薪酬下限		薪酬上限	
	RMSE	*MAE*	*RMSE*	*MAE*
SVM	5.675±0.215	4.120±0.028	10.404±1.202	7.177±0.038
LR	5.386±0.021	4.033±0.013	9.545±0.049	7.139±0.028
GBDT	4.878±0.023	3.651±0.017	8.763±0.032	6.568±0.027
DNN	6.498±0.031	4.999±0.036	11.801±0.021	9.460±0.020
HSBMF	5.291±0.017	3.939±0.015	9.188±0.036	6.800±0.028
TextCNN	4.999±0.028	3.712±0.018	8.800±0.057	6.554±0.057
HAN	4.761±0.043	3.497±0.054	8.333±0.069	6.111±0.092
Transformer-XL	5.459±0.016	4.097±0.045	9.663±0.061	7.278±0.074
BERT	4.592±0.010	3.331±0.011	8.110±0.136	5.841±0.137
RoBERTa	4.642±0.014	3.377±0.011	8.400±0.076	6.122±0.058
XLNet	4.566±0.015	3.333±0.011	8.254±0.060	5.995±0.044
SSCN	4.435±0.061	3.244±0.048	7.686±0.086	5.627±0.060
MCISUN(DeepSet)(本文)	3.439±0.018	2.413±0.015	5.909±0.036	4.193±0.028
MCISUN (w/o l)(本文)	4.336±0.096	3.187±0.092	7.172±0.070	5.273±0.057
MCISUN (w/o a)(本文)	3.243±0.015	2.148±0.014	5.640±0.028	3.778±0.019
MCISUN(本文)	**3.169±0.017**	**2.118±0.012**	**5.505±0.025**	**3.718±0.022**

注：黑体表示最低误差.

Table 3 Salary Prediction Errors on Designer Dataset

表 3 Designer 数据集上薪酬预测误差

模型	薪酬下限		薪酬上限	
	RMSE	*MAE*	*RMSE*	*MAE*
SVM	4.271±0.067	3.137±0.030	7.361±0.101	5.441±0.050
LR	4.183±0.053	3.089±0.029	7.343±0.131	5.436±0.075
GBDT	3.534±0.066	2.585±0.035	6.295±0.110	4.657±0.068
DNN	5.181±0.039	4.117±0.039	9.209±0.107	7.307±0.065
HSBMF	4.587±0.086	3.347±0.036	7.874±0.095	5.814±0.074
TextCNN	4.282±0.148	3.151±0.064	8.800±0.057	5.542±0.107
HAN	4.032±0.123	2.983±0.120	7.126±0.189	5.308±0.139
Transformer-XL	5.075±0.124	3.909±0.132	9.141±0.379	7.151±0.336
BERT	3.797±0.044	2.807±0.027	10.646±0.109	8.343±0.131
RoBERTa	4.272±0.142	3.136±0.075	9.187±0.389	7.522±0.622
XLNet	3.852±0.069	2.864±0.037	4.498±0.009	3.312±0.014
SSCN	3.316±0.036	2.408±0.025	5.887±0.139	4.294±0.107
MCISUN(DeepSet)(本文)	2.604±0.031	1.765±0.030	4.473±0.066	3.110±0.056
MCISUN(w/o l)(本文)	2.939±0.025	2.047±0.024	5.477±0.064	3.738±0.037
MCISUN (w/o a)(本文)	2.657±0.024	1.791±0.017	4.353±0.020	2.940±0.017
MCISUN(本文)	**2.521±0.020**	**1.639±0.012**	**4.170±0.025**	**2.784±0.019**

注：黑体表示最低误差.

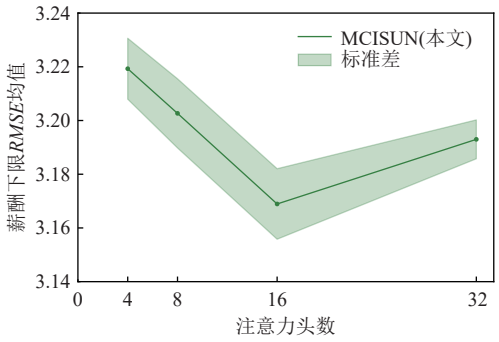

Fig. 3 Hyperparameter experimental result of the number of attention head

图 3 注意力头数超参数实验结果

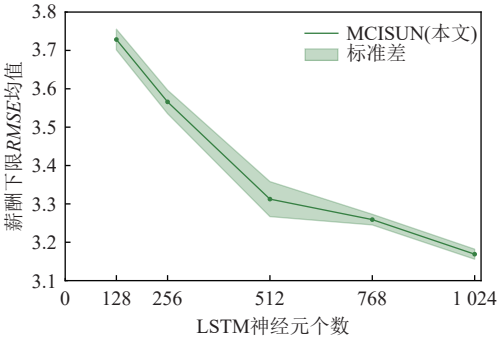

Fig. 4 Hyperparameter experimental result of the number of LSTM units

图 4 LSTM 神经元超参数实验结果

元个数从 128 提升到 512, 预测误差迅速降低, 继续提升 LSTM 神经元个数, 预测误差的降低减缓, 出于模型复杂度的考虑, 本文将 LSTM 神经元个数的默认参数设置为 1 024.

4.2.5 数据效率实验

为验证模型数据效率, 分别在 5%, 10%, 20%, 50% 训练数据规模下验证模型在 IT 数据集上的表现. 各设置上进行 10 次留出[43]验证, 实验结果如图 5 所示. 为了展示的直观性, 图 5 中只对比 6 种有代表性的模型. 观察到不同数据量下, MCISUN 的预测误

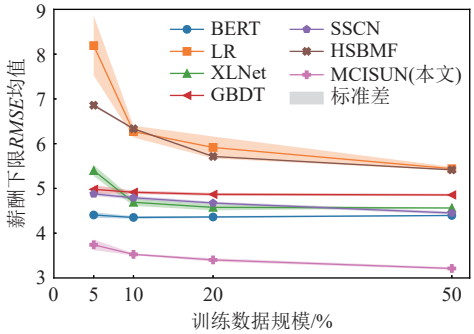

Fig. 5 Data efficiency experiment

图 5 数据效率实验

差均显著低于基线模型, 且误差随数据量变化不明显, 证明模型在训练数据稀疏时保持高泛化性.

4.2.6 排列消融实验

为验证 MCISUN 建模过程中 LSTM 层和顺序敏感性训练策略对模型的影响, 本节进行 3 组消融实验: 1) 固定输入顺序, 使用基于频率的低学习难度序列构建策略 (w/o s); 2) 删除 LSTM 模块 (w/o l); 3) 固定输入顺序且删除 LSTM 模块 (w/o s, w/o l). 在不同规模 IT 训练数据上评估各模型的 *RMSE* 结果如图 6 所示.由图 6 可得: 1)在训练数据规模较小 (1% 和 5%) 时, 基于排列的数据增强策略可降低 *RMSE*, 有效提升模型在数据稀疏时的泛化性. 但数据量较大时模型不容易过拟合, 此时基于可靠性的序列构建策略由于能够降低训练难度, 可取得与数据增强策略相同甚至更好的效果. 2)删除 LSTM 模块后模型在各数据规模下的 *RMSE* 显著上升, 说明基于 LSTM 的序列建模是 MCISUN 的重要组成部分. 3)删除 LSTM 后数据增强策略对性能的提升幅度下降, 说明通过 LSTM 充分建模序列信息能够提升数据增强效果.

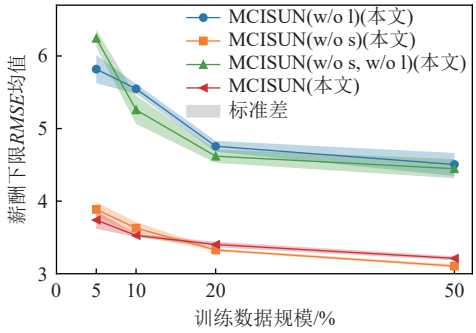

Fig. 6 Permutation ablation experiment

图 6 排列消融实验

4.3 技能贡献分析

本节在 IT 数据集上对 MCISUN 进行定性实验, 通过选取直观上可代表不同工作场景特点的热门通用技能, 分析模型中工作场景对技能贡献的影响.

4.3.1 不同公司中的技能贡献

本实验选取国内 5 家代表性的大型互联网公司并分析对应不同类别工作内容的 6 个关键技能对各公司岗位薪酬的平均贡献, 以观察不同公司在 IT 岗位上的差异性. 具体而言, 对所有该公司发布的包含对应技能的样本进行 100 次随机打乱并计算技能贡献平均值, 从而综合考虑不同顺序下技能的总体贡献, 各技能贡献分布的箱线图如图 7 所示. 观察到各公司的技能贡献分布具有相似性. 如算法和 Python 的贡献总体较高, 与 2016—2019 年间算法研究岗位普遍高薪的现象相符. 同时由于商业策略的差异, 技能贡献在不同公司中仍存在差别. 例如美团和字节跳动公司中算法技能贡献的上下四分位点差距近 1 600, 远大于其他 3 家公司, 体现出美团和字节跳动公司中在算法方向提供更多高薪职位.

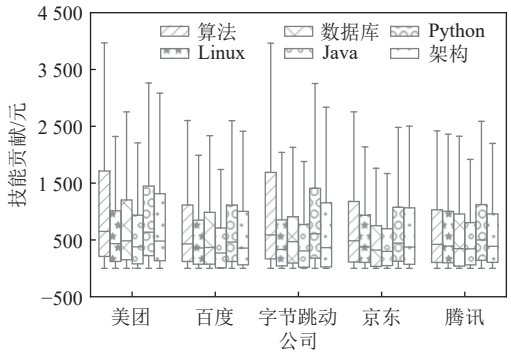

Fig. 7 Boxplots of skill contributions for different Internet companies

图 7 互联网公司中的技能贡献箱线图

4.3.2 技能贡献随时间的变化

本实验选取包括 3 个经典基础技能及 2 个新兴热门技能在内的 5 种常见技能, 计算其在不同时间段的薪酬贡献, 并对比不同类型技能随时间变化的趋势. 与 4.3.1 节实验类似, 本实验对包含对应技能的所有样本进行多次随机打乱并取平均贡献. 由于数据中各时间段样本量充足, 各样本仅随机打乱 20 次. 实验结果如图 8 所示, 观察到推荐系统与 GoLang 的技能贡献较高, 这与近些年电子商务与大规模分布式系统的快速发展相关. 算法和架构的技能贡献偏低, 原因是其常作为大量普通岗位的基本能力要求, 一般不对岗位薪酬产生决定性作用. 同时观察到热门技能的贡献波动和置信区间较大, 显示相关行业发展尚不稳定, 启示劳动者不盲目追求热门技能. 相比之下, C /C++、架构、算法等基础技能一直保持稳定的上升趋势, 映射 2016—2019 年间互联网行业薪酬水平总体提升的现象.

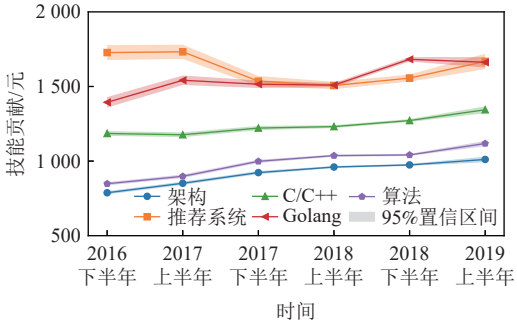

Fig. 8 Distribution of skill contributions over time

图 8 技能贡献随时间的分布

4.3.3 技能间的相互影响

本实验选取 3 种热门面向对象编程语言 Java, C++, Python, 通过分析注意力权值来计算其他技能对相似编程语言的影响差异性. 将所有包含对应技能的样本随机打乱 20 次, 计算其对各前置技能的注意力权重. 对于每个技能选择注意力权重最大的 5 个频繁共现技能(数据集中共现次数不小于 1 000), 其结果如表 4 所示. 观察到 Python 贡献受 R 语言等数据统计分析相关技能影响较大. 可推断数据分析中 R 语言和 Python 使用场景的重合性导致在已有 R 语言的情况下 Python 薪酬贡献受较大影响. 对于 C++而言, 移动端开发受相关技能影响较大, 同时由于 C++与 C 语言常共同出现, 二者耦合极高, 因此 C 语言的出现影响模型对 C++的贡献评估. 最后, Java 是广泛应用于多种系统开发的语言, 观察到其贡献评估受前置技能指代的使用场景而发生变化.

Table 4 Prerequisite Skills That Have the Greatest Impact on the Different Skills

表 4 对不同编程技能影响最大的前置技能

编程技能	Top-5
Python	R 语言、数据分析、 数学、 数据仓库、统计
C++	IOS、 Android、 客户端、 数学、C 语言
Java	项目管理、 Android、 推荐系统、 IOS、 大型软件

4.4 岗位案例分析

为深入了解模型的薪酬建模过程, 本节从数据集中随机选取一份岗位进行案例分析, 其内容如表 5 所示.

Table 5 A Sample Job Post Content

表 5 案例岗位内容

内容条目	内容明细
发布时间	2018 年 10 月
薪酬范围	1.5~3.0 万元
工作地点	北京
技能集	Python、 编程、 编译、 C、 数据结构、机器学习、 Java、 NLP、算法、 C++

4.4.1 整体薪酬变化

为分析薪酬增益的积累过程, 随机选取了一个序列, 得到这个工作中薪酬随技能增长的整体趋势如图 9 所示. 观察到薪酬上限与薪酬下限的整体增长趋势类似, 证明技能对上限及下限的贡献强相关. 薪酬起初增速较小, 符合拥有过少技能一般难以胜任任何工作的常理. 之后, 薪酬增速随技能增多逐渐提升, 当积累足量技能又因边际效应递减而放缓.

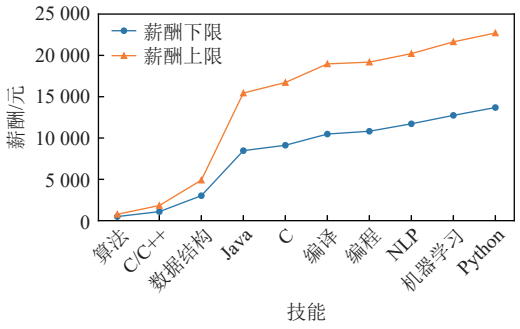

Fig. 9 The changes of salary with skills

图 9 薪酬随技能变化情况

4.4.2 技能贡献

为分析各技能对该岗位薪酬的贡献, 本实验将岗位技能集随机打乱 1 000 次并计算各技能平均贡献, 结果如图 10 所示. 技能从左到右按照加入顺序排列. 观察到机器学习、算法、NLP 等算法类技能的贡献高于 Python、编程、编译等基础技能, 说明算法类技能是该岗位薪酬的决定因素. 因此在求职时人才应注重突出其在机器学习、算法、NLP 方面的能力及经验. 同时, 在各项编程语言技能中, Python 的贡献偏低. 这一方面由于 Python 学习难度低、人才供应充足; 另一方面则由于其与其他技能重合较大, 例如大部分机器学习和 NLP 人才都掌握 Python 技能, 因此在已有机器学习和 NLP 的情况下, Python 是否出现在集合中对薪酬无显著影响. 相比之下, C++学习难度高且与其他技能重合度更低, 因此有更高薪酬贡献. 事实上, 在掌握机器学习、NLP 等高级技能的情况下, C++可使人才胜任复杂的机器学习及 NLP 系统开发及部署工作, 获得更高薪资. 因此若求职者已满足核心算法能力要求, 可多突出 C/C++而非 Python 编程能力.

4.4.3 注意力层可视化

为分析技能之间的相互影响, 图 11 展示此样例岗位中各技能在注意力层中的权值, 颜色越深代表前置技能对目标技能重要性越大.

由图 11 可以发现, 编程语言技能普遍对其他技能有较大影响, 原因为编程语言影响模型对其他技能语义的判断. 以机器学习为例, 与 Python 共现可能代表算法研究工作, 而与 C/C++或 Java 共现可能代表机器学习系统开发工作. 在编程语言中, Python 对其他技能的影响最高, 原因可能为 Python 作为数据分析和算法研究的常见典型技能, 能显著影响模型对工作内容的判断.

4.4.4 前置技能对技能贡献的提升

本实验以机器学习为例观察不同前置技能对技

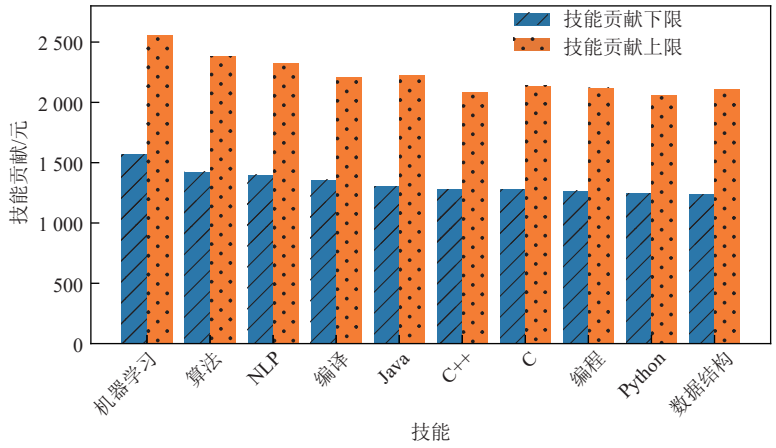

Fig. 10 The average contributions of skills for the sample job post

图 10 技能对样例岗位的平均贡献

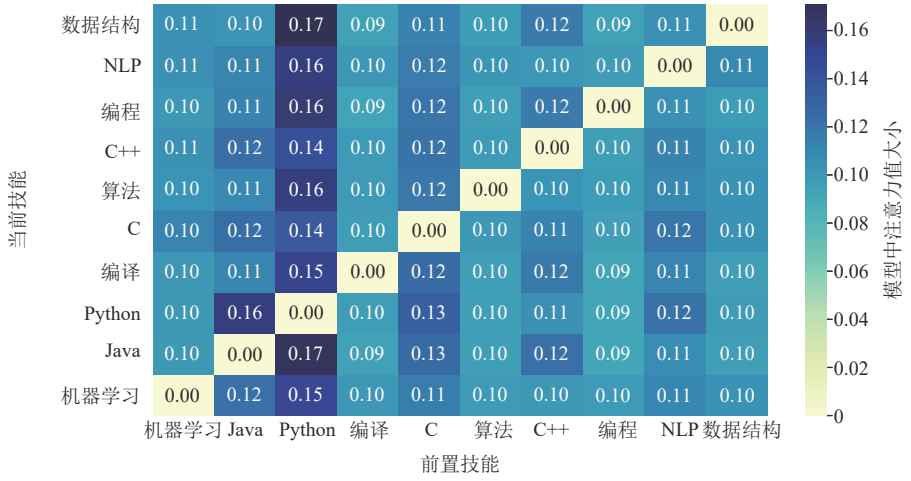

Fig. 11 The heatmap of attention value between skills

图 11 技能间注意力值热力图

能贡献的提升或降低作用. 具体而言，将技能集进行 1 000 次随机排列后分别计算各技能作为前置技能出现时机器学习技能的平均贡献，实验结果如图 12 所示. 观察到算法、NLP、Java 等技能作为前置技能时机器学习技能的平均贡献较大. 有趣的是，注意力层中机器学习对算法的注意力并不高，这说明注意力层中一些高权重的技能对机器学习的贡献降低了. 特别是 C/C++及数据结构技能显著降低机器学习技能的贡

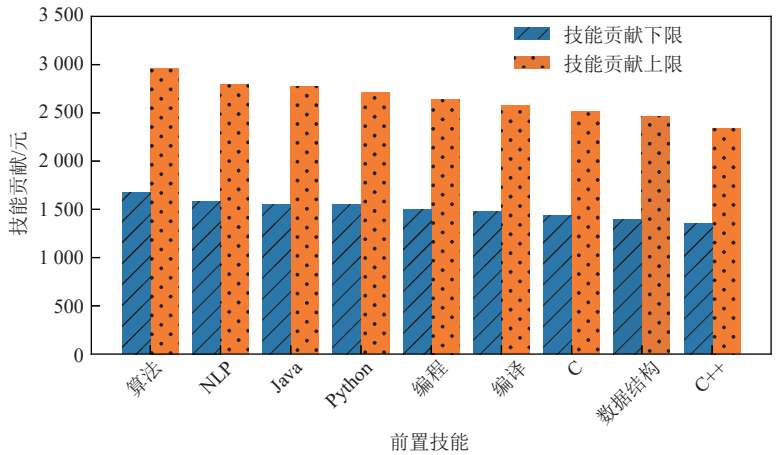

Fig. 12 The impact of prerequisite skills on the contribution of machine learning skill

图 12 前置技能对机器学习技能贡献的影响

献. 这 2 个技能与系统开发工作相关性较高，而开发工作往往着重于系统优化而非对机器学习算法本身的研究，因此机器学习的薪酬贡献下降. 而在算法研究岗位中机器学习知识对于工作结果至关重要，因此机器学习的薪酬贡献会更高. 此结果证明模型通过技能集的组合效应建模岗位的语义，针对性地发现关键技能，进行合理的贡献估计.

5 结　　论

本文针对知识技能导向的薪酬预测问题开展了研究，将薪酬预测问题定义为边信息作用下集合效用建模问题，为实现复杂变量影响下的集合效用有效建模，提出了一种基于边际贡献的增量式集合效用建模网络. 所提网络通过拟合元素加入集合时的效用增量灵活且可解释地建模集合效用. 区别于利用池化结构实现排列不变性的做法，MCISUN 通过循环注意力神经网络构建顺序敏感的中间结果并且利用集合的排列不变性实现数据增强，有效提升了模型泛化性. 大规模真实薪酬数据上的实验结果表明所提模型在基于技能的薪酬预测任务上相比 SOTA 模型效果提升超过 30%. 同时，定性实验证明所提模型能够为技能设置合理的贡献值，并能发现技能间的关联. 本文仍存在一定局限性，即模型中间贡献建模缺乏直接监督，导致数据量较小时输出贡献主要受前置集合大小影响，未来我们将在本文的基础上进一步完善可解释集合效用建模算法，通过增加对中间变量的约束实现更稳定的元素贡献评估.

作者贡献声明：孙莹提出了算法思路和实验方案并撰写论文；章玉婷负责完成实验并撰写论文；庄福振、祝恒书指导论文整体思路及修改论文；何清、熊辉提出指导意见并参与论文校对与审核. 本文的通信作者为庄福振（zhuangfuzhen@buaa.edu.cn）和祝恒书（zhuhengshu@gmail.com）.

参　考　文　献

[1] Hamlen K R, Hamlen W A. Faculty salary as a predictor of student outgoing salaries from MBA programs[J]. Journal of Education for Business, 2016, 91(1): 38−44

[2] Khongchai P, Songmuang P. Implement of salary prediction system to improve student motivation using data mining technique[C/OL]//Proc of the 11th Int Conf on Knowledge, Information and Creativity Support Systems (KICSS). Piscataway, NJ: IEEE, 2016[2023-06-25].https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7951419

[3] Khongchai P, Songmuang P. Random forest for salary prediction system to improve students' motivation[C]//Proc of the 12th Int Conf on Signal-Image Technology and Internet-Based Systems (SITIS). Piscataway, NJ: IEEE, 2016: 637−642

[4] Bansal U, Narang A, Sachdeva A, et al. Empirical analysis of regression techniques by house price and salary prediction[C/OL]// Proc of the IOP Conf Series: Materials Science and Engineering. 2021[2023-06-25].https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012110/pdf

[5] Ma Xinyu, Fan Yixing, Guo Jiafeng, et al. An empirical investigaion of generalization and transfer in short text matching[J]. Journal of Computer Research and Development, 2022, 59(1): 118−126 (in Chinese)
(马新宇, 范意兴, 郭嘉丰, 等. 关于短文本匹配的泛化性和迁移性的研究分析[J]. 计算机研究与发展, 2022, 59(1): 118−126)

[6] Pan Bo, Zhang Qingchuan, Yu Chongchong, et al. Research on the application of Doc2vec in salary forecast[J]. Application Research of Computers, 2018, 35(1): 155−157 (in Chinese)
(潘博, 张青川, 于重重, 等. Doc2vec 在薪水预测中的应用研究[J]. 计算机应用研究, 2018, 35(1): 155−157)

[7] More A, Naik A, Rathod S. Predict-nation skills based salary prediction for freshers[C/OL]//Proc of the 4th Int Conf on Advances in Science & Technology (ICAST2021). Berlin: Springer, 2021[2023-06-25].https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3866758

[8] Martín I, Mariello A, Battiti R, et al. Salary prediction in the IT job market with few high-dimensional samples: A spanish case study[J]. International Journal of Computational Intelligence Systems, 2018, 11(1): 1192−1209

[9] Sun Ying, Zhuang Fuzhen, Zhu Hengshu, et al. Market-oriented job skill valuation with cooperative composition neural network[J]. Nature Communications, 2021, 12(1): 1−12

[10] Zaheer M, Kottur S, Ravanbakhsh S, et al. Deep sets[C]//Advances in Neural Information Processing Systems 30. Cambridge, MA: MIT, 2017[2023-06-25].https://proceedings.neurips.cc/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf

[11] Vinyals O, Bengio S, Kudlur M. Order matters: Sequence to sequence for sets[J]. arXiv preprint, arXiv: 1511.06391, 2015

[12] Lee J, Lee Y, Kim J, et al. Set Transformer: A framework for attention-based permutation-invariant neural networks[C]// Proc of the 36th Int Conf on Machine Learning. New York: ACM, 2019: 3744−3753

[13] Zhang Yan, Hare J, Prügel-Bennett A. FSPool: Learning set representations with featurewise sort pooling[C/OL]//Proc of the 8th Int Conf on Learning Representations. 2020[2023-06-25].https://openreview.net/forum?id=HJgBA2VYwH

[14] Murphy R L, Srinivasan B, Rao V, et al. Janossy Pooling: Learning deep permutation-invariant functions for variable-size inputs[C/OL]// Proc of the 8th Int Conf on Learning Representations. 2020[2023-06-25].https://openreview.net/forum?id=BJluy2RcFm

[15] Yang Bo, Wang Sen, Markham A, et al. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction[J]. International Journal of Computer Vision, 2020, 128(1): 53−73

[16] Saito Y, Nakamura T, Hachiya H, et al. Exchangeable deep neural networks for set-to-set matching and learning[C]//Proc of the 17th European Conf on Computer Vision. Berlin: Springer, 2020: 626−646

[17] Zhang Yan, Hare J, Prügel-Bennett A. Learning representations of sets through optimized permutations[C/OL]//Proc of the 7th Int Conf on Learning Representations. 2019[2023-06-25].https://openreview.net/forum?id=HJMCcjAcYX

[18] Blankmeyer E, LeSage J P, Stutzman J R, et al. Peer-group dependence in salary benchmarking: A statistical model[J]. Managerial and Decision Economics, 2011, 32(2): 91−104

[19] Kenthapadi K, Ambler S, Zhang Liang, et al. Bringing salary transparency to the world: Computing robust compensation insights via LinkedIn Salary[C]//Proc of the 26th ACM on Conf on Information and Knowledge Management. New York: ACM, 2017: 447−455

[20] Zhang Haoyu. Job salary prediction based on text similarity and collaborative filtering[D]. Guangzhou: Zhongnan University of Economics and Law, 2018 (in Chinese)
(张浩宇. 基于文本相似度与协同过滤的岗位薪资预测[D]. 广州：中南财经政法大学, 2018)

[21] Meng Qingxin, Xiao Keli, Shen Dazhong, et al. Fine-grained job salary benchmarking with a nonparametric Dirichlet process–based latent factor model[J]. INFORMS Journal on Computing, 2022, 34(5): 2443−2463

[22] Meng Qingxin, Zhu Hengshu, Xiao Keli, et al. Intelligent salary benchmarking for talent recruitment: A holistic matrix factorization approach[C]//Proc of the 2018 IEEE Int Conf on Data Mining (ICDM). Piscataway, NJ: IEEE, 2018: 337−346

[23] Wang Zhongsheng, Sugaya S, Nguyen D P T. Salary prediction using bidirectional-GRU-CNN model[C/OL]//Proc of the 25th Annual Meeting of the Association for Natural Language Processing. 2019[2023-06-25].https://www.anlp.jp/proceedings/annual_meeting/2019/pdf_dir/F3-1.pdf

[24] Guo Huifeng, Tang Ruiming, Ye Yunming, et al. DeepFM: A factorization-machine based neural network for CTR prediction [C]//Proc of the 26th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2017: 1725−1731

[25] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735−1780

[26] Sun Ying, Zhuang Fuzhen, Zhu Hengshu, et al. Job posting data[CP/OL]. 2021[2023-06-25].https://figshare.com/articles/dataset/Job_Posting_Data/14060498/

[27] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[C/OL]//Proc of the 30th Int Conf on Artificial Intelligence and Statistics. New York: ACM, 2010[2023-06-25]. http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

[28] Kingma D P, Ba J. Adam: A method for stochastic optimization[C/OL]//Proc of the 3rd Int Conf on Learning Representations (Poster). 2015[2023-06-25].https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html

[29] Xu Bing, Wang Naiyan, Chen Tianqi, et al. Empirical evaluation of rectified activations in convolutional network[J]. arXiv preprint, arXiv: 1505.00853, 2015

[30] Noble W S. What is a support vector machine?[J]. Nature Biotechnology, 2006, 24(12): 1565−1567

[31] Montgomery D C, Peck E A, Vining G G. Introduction to Linear Regression Analysis[M]. Hoboken: John Wiley & Sons, 2021

[32] Mason L, Baxter J, Bartlett P, et al. Boosting algorithms as gradient descent[C/OL]//Advances in Neural Information Processing Systems 12. Cambridge, MA: MIT, 1999[2023-06-25].https://proceedings.neurips.cc/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf

[33] Gardner M W, Dorling S R. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences[J]. Atmospheric Environment, 1998, 32(14/15): 2627−2636

[34] Chen Yahui. Convolutional neural network for sentence classification[D]. Waterloo: University of Waterloo, 2015

[35] Zhang Xiang, Zhao Junbo, LeCun Y. Character-level convolutional networks for text classification[C/OL]//Advances in Neural Information Processing Systems 28. Cambridge, MA: MIT, 2015[2023-06-25]. https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f-8dc8b4be867a9a02-Paper.pdf

[36] Yang Zichao, Yang Diyi, Dyer C, et al. Hierarchical attention networks for document classification[C]//Proc of the 15th North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2016: 1480−1489

[37] Dai Zihang, Yang Zhilin, Yang Yiming, et al. Transformer-Xl: Attentive language models beyond a fixed-length context[C/OL]//Proc of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2019[2023-06-25].https://arxiv.org/pdf/1901.02860.pdf%3Ffbclid%3DIwAR3nwzQA7VyD36J6u8nEOatG0CeW4FwEU_upvvrgXSES1f0Kd-

[38] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proc of the 17th Annual Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2019: 4171−4186

[39] Liu Yinhan, Ott M, Goyal N, et al. RoBERTa: A robustly optimized BERT pretraining approach[J]. arXiv preprint, arXiv: 1907.11692, 2019

[40] Yang Zhilin, Dai Zihang, Yang Yiming, et al. XLNet: Generalized autoregressive pretraining for language understanding[C/OL]//Advances in Neural Information Processing Systems 32. Cambridge, MA: MIT, 2019[2023-06-25].https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[41] Zhang Yan, Hare J, Prugel-Bennett A. Deep set prediction networks[C/OL]//Advances in Neural Information Processing Systems 32. Cambridge, MA: MIT, 2019 [2024-03-29]. https://proceedings.neurips.cc/paper_files/paper/2019/file/6e79ed05baec2754e25b4eac73a332d2-Paper.pdf

[42] Botchkarev A. A new typology design of performance metrics to measure errors in machine learning regression algorithms[J]. Interdisciplinary Journal of Information, Knowledge, and Management, 2019, 14: 45−79

[43] Blum A, Kalai A, Langford J. Beating the hold-out: Bounds for *k*-fold and progressive cross-validation[C]//Proc of the 12th Annual Conf on Computational Learning Theory. New York: ACM, 1999: 203−208

Sun Ying, born in 1994. PhD, assistant professor, PhD supervisor. Member of CCF. Her main research interests include machine learning and data mining.

孙　莹, 1994 年生. 博士, 助理教授, 博士生导师. CCF 会员. 主要研究方向为机器学习、数据挖掘.

Zhang Yuting, born in 1998. Master candidate. Her main research interests include machine learning and data mining.

章玉婷, 1998 年生. 硕士研究生. 主要研究方向为机器学习、数据挖掘.

Zhuang Fuzhen, born in 1983. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include machine learning and data mining.

庄福振, 1983 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为机器学习、数据挖掘.

Zhu Hengshu, born in 1986. PhD, professor of engineering. Senior member of CCF. His main research interests include machine learning and data mining.

祝恒书, 1986 年生. 博士, 高级工程师(正研级).CCF 高级会员. 主要研究方向为机器学习、数据挖掘.

He Qing, born in 1965. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include machine learning and data mining.

何　清, 1965 年生. 博士, 研究员, 博士生导师. CCF 高级会员. 主要研究方向为机器学习、数据挖掘.

Xiong Hui, born in 1972. PhD, professor, PhD supervisor. Senior member of CCF. His main research interest includes data and knowledge engineering.

熊　辉, 1972 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为数据与知识工程.