

## 2024美团外卖：基于上下文的长序列建模技术

**SmartMindAI**

专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注

已关注

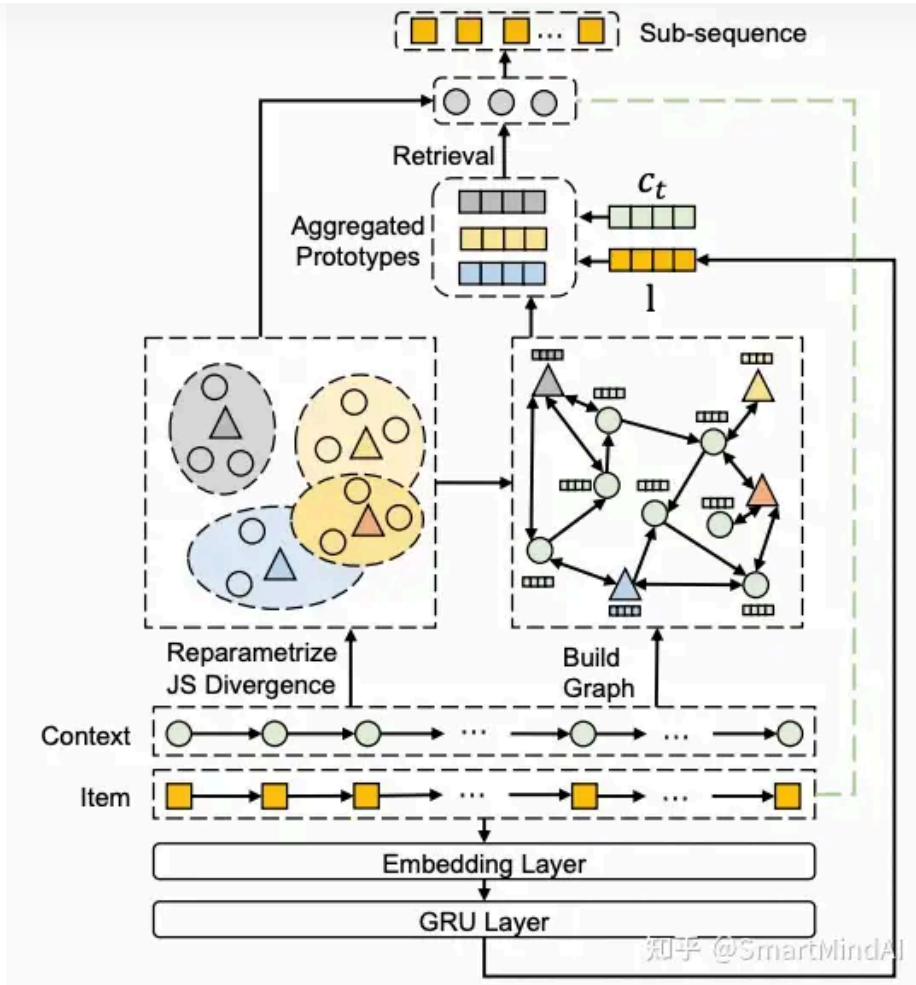
38 人赞同了该文章

### INTRODUCTION

近来，序列推荐的性能改进也逐渐引入上下文信息。尽管这些方法展现出了有效的性能提升，它们通常依赖于自我注意力机制<sup>+</sup>，以捕获序列间元素的依赖性。然而，这种方法在处理含有较长序列的数据时，可能遭遇挑战，特别是在容纳复杂空间中特征交互模式的抽象处理上，容易受到计算成本<sup>+</sup>和模型复杂度的限制。

为了解决这些问题，本研究遵循了双阶段范式，重点放在了第一阶段-----从用户的操作序列中选择一个子序列。考虑到美团外卖上下文依赖的特点，我们提议从用户的过往行为中选择与目标上下文交互的地点作为备选序列，这种方法不依赖于候选地点。然而，如果只保留完全符合目标上下文的行为，则检索到的子序列会过于简短，从而限制了推荐系统<sup>+</sup>的效率。

为此，我们提出了基于上下文的快速推荐策略（名为 CoFARS），以解决美团外卖推荐系统中长序列的问题。我们的目标是在用户偏好基础上确定与目标上下文相似的上下文。我们采用了JS散度<sup>+</sup>来度量不同上下文间地点属性分布的相似性，这既增强了可解释性也提升了精度。为了降低复杂性，我们采用了基于原型的方法，使用原型作为属性相似邻居在偏好表示空间中的中心节点。由于缺乏预先知情的原型知识，我们借鉴了迁移学习的思想，通过编码器将潜在表示转换为概率分布<sup>+</sup>，并使用MSE损失将这些分布与日志数据中不同上下文之间的实际JS散度进行对齐。我们还构建了包含原型和上下文节点的时序图<sup>+</sup>，引入了时间信息。通过考虑用户的短期偏好，我们选择与目标上下文相符的原型，并从用户的操作序列中检索相关子序列。



### Probability Encoder

在美团外卖推荐系统中，地理位置、就餐时间、天气以及假期等诸多上下文特征对用户偏好产生了重大影响。传统的处理方式，例如采用上下文嵌入的余弦相似性<sup>+</sup>方法，旨在识别与目标上下文存在相似用户倾向的上下文以及挑选与目标上下文相关的关键子序列，但在可解释性较差且依赖于嵌入质量等方面遇到了挑战。

然而，我们注意到在某特定上下文中，用户偏好更准确地通过跨越地点属性（如类别、价格）的概率分布来体现。这一发现使我们采用概率分布一致性作为不同上下文间偏好比较的衡量标准，从而既增强了可解释性又提高了准确性。常见的做法是计算Kullback-Leibler (KL) 散度。为避免其固有的不对称性，我们选择了对称的替代方法，即Jensen-Shannon (JS) 散度。我们通过

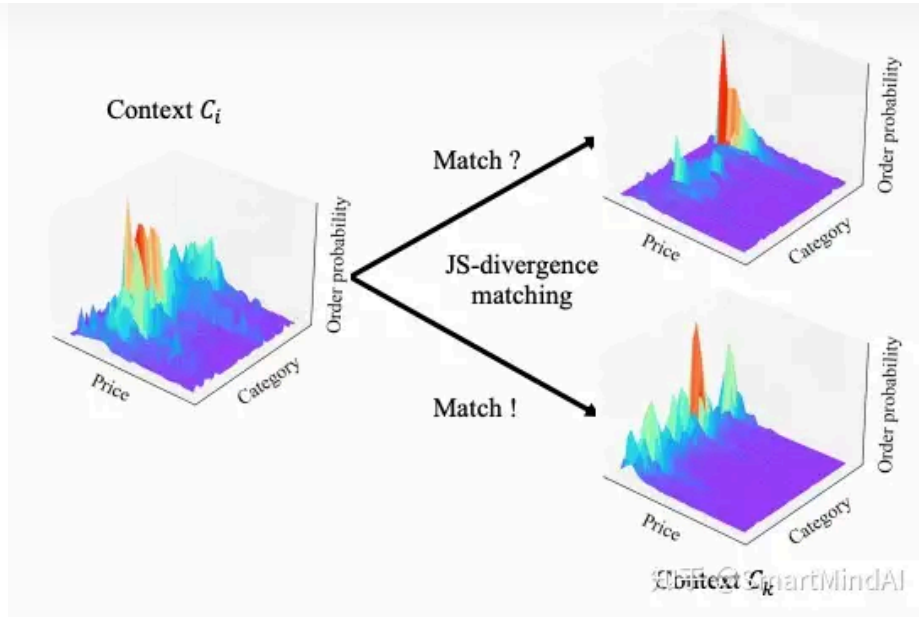
$$\mathcal{D}(\mathcal{C}_i) = p(a_1, a_2, \dots, a_{|A|} | \mathcal{C}_i)$$

来表示在上下文 $\mathcal{C}_i$ 下属性值 $\{a_1, a_2, \dots, a_{|A|}\}$ 的概率分布，这是一 $|A|$ 维向量，记录了从日志数据中获得的对应离散属性值的比例。给定上下文 $\mathcal{C}_i$ 与上下文 $\mathcal{C}_j$ 之间的散度公式如下所示：

$$\text{KL}(\mathcal{C}_i, \mathcal{C}_j) = \sum_{a_1 \in \mathbf{a}_1} \sum_{a_2 \in \mathbf{a}_2} \dots \sum_{a_{|A|} \in \mathbf{a}_{|A|}} \mathcal{D}(\mathcal{C}_i) \log \frac{\mathcal{D}(\mathcal{C}_i)}{\mathcal{D}(\mathcal{C}_j)}$$

$$\text{JS}(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{2} (\text{KL}(\mathcal{C}_i, \mathcal{C}_j) + \text{KL}(\mathcal{C}_j, \mathcal{C}_i))$$

为了直观展示匹配过程，我们采用了基于价格和类别维度的订单概率分布可视化，如图所示。



通过分析每位用户基于日志数据计算得到的订单概率，可以有效地反映用户对不同地点的兴趣在于特定上下文内。这种显示方式不仅清晰地体现了用户偏好在地理、时间和条件变化下的动态，同时也为推荐系统提供了有效的决策支持依据。

这种方法重点在于计算目标上下文与过往历史上下文之间的偏好相似性，并从中挑选出与目标最为匹配的一组上下文<sup>+</sup>。然而，这种方法亦存在几大局限性。首先，它未能彻底解决“冷启动<sup>+</sup>”问题，即在新的或极为罕见的上下文情境下如何进行有效推荐。其次，模型的性能极大地依赖于为每个目标上下文所选取的用于比较的上下文数量，而这一数量在不同场景中高度波动且难于预估。

实际上，用户在特定上下文下的长期行为偏好往往表现出相对的稳定性，这一点为理解用户意图和进行个性化推荐提供了重要的线索。本文引入偏好原型概念，借鉴原型学习理论，用以表示和汇聚语义上相近的上下文。我们定义集合 $\mathcal{O}$ 包含一系列原型，其中集合的大小由 $|\mathcal{O}|$ 表示。每个原型视为偏好表示空间中的核心点，借此聚类相似上下文。匹配某一原型程度极高的上下文视为其同质性表现。由于对原型的潜在表现有所掌握，但缺乏明确的先验定义，依据地点属性直接推导概率分布显得不切实际。为此，本文借鉴迁移学习思路，设计了一种概率编码器，将原型的潜在表示转化为地点属性上的概率分布。通过与日志数据中计算出的上下文间实际JS散度的匹配调整过程，确保分布准确。具体而言，为每个上下文 $\mathcal{C}_i$ 使用向量 $\mathbf{c}_i$ 表示全球偏好的潜在表示，其中 $\mathbf{c}_i = \mathbf{u} + \hat{\mathbf{c}}_i$ 以 $\mathbf{u}$ 为通用向量。利用包含Sigmoid激活函数的多层感知器<sup>+</sup>（MLP）对 $\mathbf{c}_i$ 进行概率分布编码，其输出维度为目标属性值总数，并与所有用户共享以减少参数数量。依此编码，上下文概率分布 $\mathbf{P}(\mathbf{c}_i)$ 被近似为MLP输出。两上下文间的估计JS散度，则基于这些概率分布计算得到。

$$\begin{aligned}\overline{\text{KL}}(\mathbf{c}_i, \mathbf{c}_j) &= \sum_{k=1}^d \mathbf{c}_{i,k} \cdot \log \frac{\mathbf{c}_{i,k}}{\mathbf{c}_{j,k}} \\ \widetilde{\text{JS}}(\mathbf{c}_i, \mathbf{c}_j) &= \frac{1}{2} (\overline{\text{KL}}(\mathbf{P}(\mathbf{c}_i), \mathbf{P}(\mathbf{c}_j)) + \overline{\text{KL}}(\mathbf{P}(\mathbf{c}_j), \mathbf{P}(\mathbf{c}_i)))\end{aligned}$$

对于 $\mathbf{c}_i$ 中的第 $k$ 个元素表示为 $\mathbf{c}_{i,k}$ 。我们的目标是确保估计的JS散度尽可能接近通过日志数据计算得到的特定值。为达成此目的，我们引入了如下约束条件：

$$\mathcal{L}_{MSE} = \frac{1}{|\mathcal{C}|^2} \sum_{i=1}^{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} (\text{JS}(\mathcal{C}_i, \mathcal{C}_j) - \widetilde{\text{JS}}(\mathbf{c}_i, \mathbf{c}_j))^2$$

为了使估计的JS散度与真实值相匹配，我们通过最小化 $\mathcal{L}_{MSE}$ 这一损失函数<sup>+</sup>来实现这一目标。由此，我们可以直接计算潜在表示之间的相似度。

$$\text{sim}(\mathbf{c}_i, \mathbf{c}_j) = 1 - \widetilde{\text{JS}}(\mathbf{c}_i, \mathbf{c}_j)$$

值得注意的是，上下文的聚类<sup>+</sup>会动态调整，过量的原型往往对应于极少的上下文，因此当原型的总数超过一个特定的界限时，模型的性能表现较为稳定，模型的效能并未显著依赖于这个超参数<sup>+</sup>，这一结论在后续的实验中得到了验证。

户的共享表示，它遵循了上下文偏好表示的分布规律，并且利用  $\mathbf{o}_i = \hat{\mathbf{o}}_i + \mathbf{u}$

将这个共享表示个性化为原型 $\mathbf{o}_i$ 。为了确保不同原型之间的独特性，通过鼓励它们的JS散度差异，从而促进模型内部的多样性。

$$\mathcal{L}_{IND} = -\frac{1}{|\mathcal{O}|^2} \sum_{i=1}^{|\mathcal{O}|} \sum_{j=1}^{|\mathcal{O}|} \widetilde{\text{JS}}(\mathbf{o}_i, \mathbf{o}_j)$$

最小化 $\mathcal{L}_{IND}$ 能够促使原型在潜在空间中分布得更为均匀。

### Graph-based Temporal Aggregator

通过概率编码器，成功地将上下文聚类在其潜在空间中，但其对时间序列<sup>+</sup>中的动态信息处理有限。特别是在处理长序列<sup>+</sup>时，这种方法难以避免如RNN模型中出现的记忆泄露，同时基于注意力机制的模型计算复杂度激增。近年来，图神经网络<sup>+</sup>（GNNs）在序列推荐领域崭露头角，通过构建项目之间的连接图来处理用户行为序列，常见做法是利用项目间的共现关系构建图结构。然而，面对稀疏的用户行为序列，所构建的图往往简化为线性结构，显著削弱了GNNs的功效。GNNs的应用更多集中于会话推荐这类频繁重复项目出现的场景，但在上下文序列中，由于受限的上下文多样性和长序列长度，频繁的上下文重复为GNNs提供了一个理想的建模环境。因此，一种改进策略是通过构建基于时间的图，其中每个节点代表上下文，边则连接相邻上下文以反映共现关系。这种结构有效捕捉了用户行为内的序列动态，为深入理解并精细建模用户交互提供了有效途径。

为了将时间维度融入原型中，我们将在图中嵌入原型节点。然而，全面连接原型节点和上下文节点可能导致图的大小变得难以管理。为了解决这个问题，我们提出了一种过滤方法，仅保留相似度得分高于某个阈值的边。然而，传统的硬编码<sup>+</sup>过滤机制不是可微的，阻碍了有效的反向传播训练。为绕过这个问题，我们采用了先前工作中的Gumbel Softmax技巧，这使得对离散输出进行可微学习成为可能。对于每个原型 $i$ 和上下文 $j$ ，等式如下：

$$\beta = \text{sim}(\mathbf{o}_i, \mathbf{c}_j)$$

$$\mathcal{P}(\beta) = \frac{\exp((\log(\beta + g_b)/\tau_p))}{\sum_{b=0}^1 \exp(\log(\beta^b(1-\beta)^{1-b} + g_b)/\tau_p)}$$

其中 $g_b$ 代表从Gumbel分布采样的随机噪声，而温度参数 $\tau_p$ 用于调节噪声的分布宽度，影响生成概率分布的离散度<sup>+</sup>。重要的是，每一对原型-场景的预处理时计算出的相似度分数会被保存，用于后续访问，从而避免了在运行时动态计算相似度的行为。对于所有符合 $\mathcal{P}(\beta) = 1$ 条件的配对，我们为两个节点之间构建双通路。接着，通过应用图注意力网络（Graph Attention Network，简称GAT）来解答节点间的交互关系，这一体系能够有效处理有向图<sup>+</sup>以及未知的图结构，以此整合时间维度，增强节点间时序信息的嵌入。

$$\mathbf{H}^0 = [\mathbf{o}_1; \dots; \mathbf{o}_{|\mathcal{O}|}; \mathbf{c}_1; \dots; \mathbf{c}_{|\mathcal{C}|}]$$

$$\alpha_{ij}^l = \frac{\exp(\text{LeakyReLU}(\text{sim}(\mathbf{H}_i^l \mathbf{W}^l, \mathbf{H}_j^l \mathbf{W}^l)))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\text{sim}(\mathbf{H}_i^l \mathbf{W}^l, \mathbf{H}_k^l \mathbf{W}^l)))}$$

$$\mathbf{H}_i^{l+1} = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^l \mathbf{H}_j^l \mathbf{W}).$$

在步骤中  $\mathbf{H}_i^l \in \mathbb{R}^d$  代表第 $i$ 个节点的 $l$ 层表示，而  $\mathbf{W}^l \in \mathbb{R}^{d \times d}$

是该层的权重参数，值得注意的是 $\sigma$ 启用了ReLU激活函数<sup>+</sup>。 $\mathcal{N}_i$ 是节点 $i$ 的前任节点的集合。通过执行 $L$ 次聚合操作，我们可以提炼出第 $i$ 个聚合后的原型表示  $\tilde{\mathbf{o}}_i = \mathbf{H}_i^L$

以及第 $j$ 个上下文偏好的聚合表示  $\tilde{\mathbf{c}}_j = \mathbf{H}_{|\mathcal{O}|+j}^L$

### Learning and Discussion

在得到聚合节点表示后，我们聚焦于用户最近 $r$ 个活动（ $r$ 远小于 $n$ ），以确定与目标上下文匹配的原型。采用GRU模型对用户的短期行为进行编码。

$$\mathbf{l} = \text{GRU}(\mathbf{e}_{(n-r)}, \mathbf{e}_{(n-r+1)}, \dots, \mathbf{e}_n)$$



在原型。

$$z_i^t = \mathcal{P}(\text{sim}(\widetilde{\mathbf{o}}_i, \widetilde{\mathbf{c}}_t)(\widetilde{\mathbf{o}}_i \cdot \mathbf{1}))$$

其中，**内积\***操作（点表示）用于度量每个原型与候选地点之间的相似性。由于有兴趣地点（Points of Interest, Pols）不存在属性概率分布的定义，因此内积是合适的计算方式。为了保证训练过程的效能，我们采用二元**交叉熵\***损失作为监督手段来优化整个模型。

$$s_{i,v}^t = z_i^t \cdot (\widetilde{\mathbf{o}}_i \cdot \mathbf{v})$$

$$\mathcal{L}_{REC} = \frac{1}{|V|} \sum_{v \in V} \sum_{i=1}^{|\mathcal{O}|} [-y \cdot \log s_{i,v}^t - (1 - y) \cdot \log(1 - s_{i,v}^t)]$$

其中 $V$ 代表训练集，而对于每一组数据点，基于是否为正样本（标记为1），或非正样本（标记为0），我们计算其对应的损失。最终的损失函数则是上述描述下计算得到的总损失。

$$\mathcal{L} = \mathcal{L}_{REC} + \gamma \cdot \mathcal{L}_{MSE} + \lambda \cdot \mathcal{L}_{IND}$$

其中 $\gamma$ 和 $\lambda$ 是用于调节MSE损失和独立性损失权重的超参数。提出的方法的全面学习策略在算法1中进行了概述，整合了这两种损失以优化模型性能。

### Algorithm 1 Learning algorithm for CoFARS

**Input:** Training epoch  $T_{train}$

- 1: Initialize model parameters  $\Theta \leftarrow$  random values;
- 2: **for**  $T_{train}$  epochs **do**
- 3:   Construct a temporal graph of contexts based on the user's historical behavior;
- 4:   Calculate the similarity between prototypes and contexts according to Eq. 6;
- 5:   Integrate prototype nodes into the graph, establishing connections between prototypes and contexts according to Eq. 8;
- 6:   Perform message passing between nodes based on Eq. 10;
- 7:   Derive the user's short-term preferences through Eq.13;
- 8:   Identify the prototypes that match the target context using Eq. 14;
- 9:   Optimize the model parameters by minimizing the loss function Eq. 17;
- 10: **end for**
- 11: **return** All optimal parameters in  $\Theta$ .

知乎 @SmartMindAI

在下一阶段中，对于集合 $\mathcal{O}$ 中每个标记为目标 $z_i^t = 1$ 的原型 $\mathbf{o}_i$ ，收集其涉及的上下文。在这些上下文内，挑选出与用户发生过交互的地点兴趣点（Points of Interest, Pols）形成序列，并运用目标注意力机制对该序列进行深度处理，以精确描绘用户兴趣模式。模型通过交叉熵损失函数进行优化训练。

对于未预见的上下文预测用户的后续访问行为，在线绑定阶段采用不同的策略。对于已知上下文，根据预先计算的原型-上下文对相似性进行预测；对于缺乏信息的上下文，则利用表示 $\mathbf{c}_t$ 计算与原型之间的相似性，并以此为依据，模仿前文阐述的步骤，从中选取序列并建模用户兴趣。

本文提出的**CoFARS**方法将允许选择与目标上下文偏好相似的序列，并避免候选选项的直接影响。在对推理时间执行分析后，我们指出在全局推荐阶段（GSU），计算原型与上下文偏好的相似度的**时间复杂度\***为  $O(|C| \cdot |\mathcal{O}|)$

其中 $|C|$ 远小于序列长度 $n$ ，且 $|\mathcal{O}|$ 为常数。图聚合阶段的时间复杂度为 $O(n \cdot L)$ 。在线阶段，生成体现短期偏好的选择子序列需花费 $O(r + |\mathcal{O}|)$ 的时间，并用此子序列检索其关联的原型。结果表明**CoFARS**方法在处理时间上的效率远超那些依赖两阶段模型的其他方案，后者的复杂度为 $O(B \cdot n)$ ，其中 $B$ 代表候选数量。

为了研究（任务），我们离线测试时选用的是2023年4月1日到4月30日的Meituan外卖记录作为训练数据集的来源，并将5月9日的数据作为验证集<sup>+</sup>。收集的样本包含多种信息：用户档案/统计特征、地点兴趣点（POI）的统计描述、过往行为细节（包括用户与POIs的互动）、点击行为的标签状态等。原始行为特征追溯至过去三年的用户互动目录。同时，我们的数据集考虑了上下文因素，包括地点（地理位置）、消费习惯（用餐时间）、环境条件（天气、节假日）以及目标特性（类别、价格、品质、送餐时间）。特定的数据集细节整理于表中，其中近期平均一栏展示了训练集内的点击行为均值，而历史平均描绘了过去三年用户互动的平均状况，以供模拟和预测用户的偏好。

Table 1: Statistics of the dataset

Field	Size
#Users	0.24 billion
#PoIs	4.37 million
#Records	36 billion
#Recent Avg.	150
#Historical Avg.	4,423

Performance Comparison

我们对比了我们的模型与基线模型的性能。新模型CoFARS以及基线模型的整体比较结果见表。

	Model	CTR AUC	CTCVR AUC
Traditional Models	Avg-Pooling	0.6993	0.7204
	DIN	0.7052	0.7263
	DIEN	0.7069	0.7281
Enhanced Models for Long Sequences	MIMN	0.7083	0.7287
	SIM <sub>hard</sub>	0.7097	0.7300
	SIM <sub>soft</sub>	0.7116	0.7321
	SDIM	0.7123	0.7325
	TWIN	0.7133	0.7336
	CoFARS	0.7188	0.7382

我们有以下几点观察：对于传统的模型，我们发现平均池化方法性能最差。通过使用目标注意力机制为行为序列中的地点分配变化权重，我们发现DIEN和DIN的表现优于无注意力机制的模型，这证明了减少无关地点影响的重要性。进一步地，通过整合时间信息，DIEN超越了DIN，这验证了CoFARS整合序列信息的重要性。

对于专门为长序列增强的推荐模型，它们的性能超越了作为短期模型的DIN和DIEN。这表明长序列在更准确捕捉用户兴趣方面发挥着关键作用。我们还观察到，MIMN在所有增强模型中展现出最低的性能效果。这可能是由于其通过温柔的注意力方式为序列中的地点分配权重来更新内存的策略，潜在的噪音累积可能显著阻碍了序列学习过程。因此，其性能落后于使用硬编码方法<sup>+</sup>的两阶段方法。在的相似探索中，还可以进一步观察这一点。此外，尽管降低了时间复杂度，SDIM的性能超越了SIM<sub>hard</sub>，并与SIM<sub>soft</sub>的性能相近，这证实了SDIM提出的哈希方法的有效性。同样，TWIN在性能上胜过了SDIM，它不仅增强了两阶段表示的一致性，而且还通过选择目标注意力而非规范地点嵌入进行序列建模，从而增强了用户兴趣与目标地点之间相关性的放大。

最后CoFARS不仅在CTR AUC上超越了所有基线模型，并且在CTCVR AUC上相较于TWIN分别提高了0.77%和0.63%，这证明了提出的CoFARS的有效性。通过引入上下文信息CoFARS，借助所提出的可能性编码器和基于图的时间聚合器<sup>+</sup>，能够有效地和高效地识别具有相似偏好的上下文，并根据这些上下文选择与目标上下文相关的目的地地点。

Online A/B Testing

准版本相比，其最终模型实现了4.6%的[点击率+](#)（CTR，Click-Through Rate）提升和4.2%的总商品价值（GMV，Gross Merchandise Volume）提升。这一成果充分展示了**CoFARS**方法在实际应用中的有效性和潜力。目前**CoFARS**已成功部署并作为主流量服务的一部分，为美团外卖系统提供持续优化的推荐体验。

原文《Context-based Fast Recommendation Strategy for Long User Behavior Sequence in Meituan Waimai》

发布于 2024-07-08 11:21 · IP 属地北京

[序列推荐](#) [上下文学习](#) [推荐系统](#)



理性发言，友善互动



发布



还没有评论，发表第一个评论吧

推荐阅读

为什么我的美团店铺没有进店流量？排名也排不到前面去？

如何获取更多免费流量。一、美团外卖APP整体流量情况 竞争这么激烈的情况下，想让消费者点外卖第一眼就看到我们，我们就必须排名更靠前，获得更多流量。其实想要提升流量，主要就是通

飞龙若在天



美团外卖五周年，想爆单一定学会抢流量！

林林星星 发表于外卖运营笔...

2021年的美团外卖真的很难做吗，外卖的怎么样才能做好？

各位对外卖感兴趣的朋友们，由于上期的文章的观点让我得罪了不少对餐饮行业外卖恨之深，爱之切的商家朋友，在这里要给大家道一个歉。由于我没有把自己的观点表述的很清楚，很多人误以为我是...

奇一and奇二



怎么3元开美团会员？

笑死朕了