

基于用户行为和社会热点的细粒度场景挖掘和推荐

本项目通过整合用户行为与社会热点数据，构建细粒度场景挖掘与推荐系统。系统由场景发现、场景识别、应用/服务推荐三大核心模块组成：在场景发现模块中，系统对多源用户行为数据进行清洗、归一化及语义增强处理，通过多模态特征编码融合应用行为、时空特征，形成用户场景表征。基于自监督预训练与对比学习框架，结合层级聚类与原型学习方法，从无标注数据中自动挖掘结构化场景簇，并具备新场景发现能力。场景识别模块聚焦用户单时间片行为数据，构建融合应用使用、位置、时间及设备状态的多模态表征。采用基于 Transformer 的多标签分类模型实现场景判定，创新性引入分层注意力机制捕捉长程依赖，结合信息增益引导的特征建模策略，有效应对原始数据的高维稀疏性与冗余问题，显著提升分类精度与模型鲁棒性。在应用/服务推荐模块中，系统以用户当前识别出的行为场景与个体静态画像为核心输入，构建知识图谱驱动的候选集筛选机制，并融合多阶段序列建模策略实现推荐生成。推荐模型集成前缀融合、逐项融合、Transformer 建模和后端融合四大机制，结合对比学习与数据增强方法，有效捕捉用户长期偏好与短期动态行为之间的深层关联。三个模块形成完整技术闭环，在数据表征、场景建模、推荐适配等关键环节实现技术创新，为个性化服务提供精准场景理解与决策支持。针对端侧实时性需求，模型采用轻量化设计，通过知识图谱剪枝、特征选择优化和量化压缩技术，将动态内存占用控制在 5MB 以内，端到端响应时延压缩至 300ms 以内，在保证精度的前提下实现高效部署。最终模型支持在冷启动、稀疏数据等复杂环境下实现鲁棒、个性化的服务推荐，显著提升推荐准确率与用户体验。

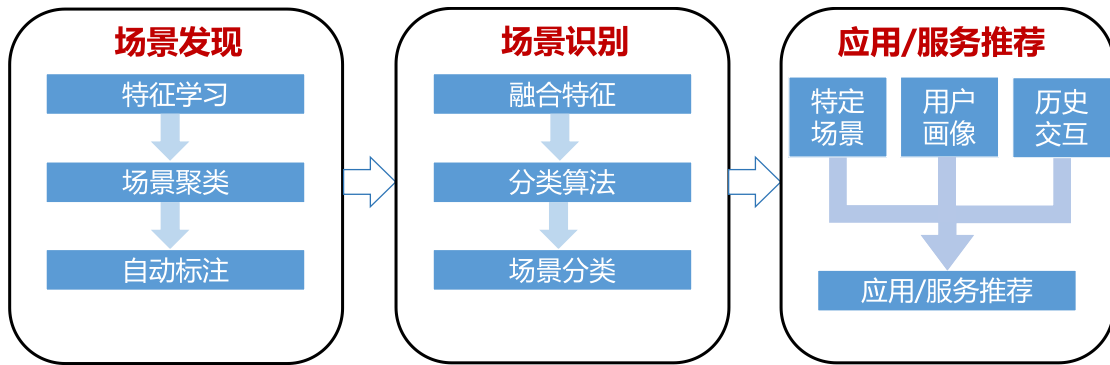


图 1: 基于用户行为和社会热点的细粒度场景挖掘和推荐

1. 基于移动用户数据的场景发现

1.1 洞察总结

在移动用户数据理解领域，学术界和工业界均开展了一些相关研究探索和工程实践。在学术界，早期工作多依赖隐马尔可夫模型（HMM）或高斯混合模型（GMM）对用户行为的时空序列进行建模，但受限于浅层特征的表征能力，难以捕捉复杂场景的语义关联。随着深度学习的兴起，基于 Transformer 的预训练方法通过掩码序列预测任务学习用户行为序列的全局上下文表征，显著提升了场景动态感知的精度。图神经网络（GNN）则通过构建用户-位置-时间的异构图，补全稀疏语义标签并挖掘潜在场景模式。针对标注数据稀缺的核心挑战，半监督学习（SSL）通过对比学习与一致性正则化策略，利用无标签数据增强场景表征的鲁棒性，而广义类别发现（GCD）进一步突破传统类别限制，允许无标签数据包含未知场景类别，通过参数化分类器与深度聚类的协同优化，实现已知场景分类与未知场景发现的动态平衡。在工业实践中，多源异构数据的融合与场景识别已形成多维度技术范式。以谷歌为例，其通过整合 Android 系统传感器、Google

Maps 位置历史及 Gmail 日程等跨模态数据,构建用户行为特征图谱,利用 LSTM 模型预测各场景。苹果则采用差分隐私与本地化模型,在设备端通过 Core ML 分析应用启动时序与模糊化位置轨迹,使锁屏推荐点击率提升的同时,将隐私泄露风险降低。Uber 的动态定价系统结合 Apache Flink 实时处理用户位置流与社会热点事件,通过时空图神经网络预测区域供需失衡。亚马逊则利用设备类型与 IP 位置的协同分析,构建多设备行为图谱,区分不同场景,使跨场景 GMV 大幅增长。

移动用户数据的自动场景发现技术在实际落地中面临一系列复杂挑战,这些挑战根植于数据本身的特性与应用场景的复杂性。首先,用户行为数据的多源性导致关键信息(如位置标签)的普遍缺失。例如,许多应用使用记录中缺乏经纬度或语义标签,使得用户在商场、办公楼等具体场景中的行为模式难以被精准识别——尽管这些场景下用户的应用偏好可能差异显著(如购物类应用与办公软件的切换),但稀疏的位置数据削弱了场景划分的准确性。与此同时,数据分布的不平衡性进一步加剧了建模难度:高频场景(如日常通勤、工作时间)因样本充足容易被模型捕捉,而低频场景(特定娱乐活动)则因样本稀缺难以形成规律性表征,导致推荐系统在面对长尾场景时泛化能力不足。更棘手的是,用户行为数据中混杂着大量噪声,例如误触操作或短暂使用的无关应用,这些离群点若未被有效过滤,可能掩盖真实场景特征,使模型误判用户意图。

从数据特性来看，移动用户行为呈现多维动态交织的特点。一方面，其多模态性体现在应用使用时长、位置语义解析、时间周期性等异构特征的深度融合。例如，用户在早晚通勤时段频繁使用导航与音乐应用，这种时序依赖性与地理位置（如地铁站、通勤路线）形成强关联；而周末的娱乐行为则可能集中在商圈或家庭场景，突显时空维度的协同影响。另一方面，数据中的噪声与长尾分布现象显著，低频场景样本易被高频场景淹没，例如临时旅游需求的数据可能因出现频次低而被模型忽略。此外，用户行为的动态性和上下文敏感性对技术方案提出了更高要求。突发社会事件（如极端天气、节日促销）可能瞬间改变用户的应用偏好，而社会热点的即时融入则要求模型具备实时感知与自适应能力。这些特性共同构成了移动用户数据的复杂性，既为场景发现提供了丰富的信息源，也对其技术实现的鲁棒性和灵活性提出了严峻考验。

1.2 技术路线

自动场景发现是为用户 app 推荐系统提供额外的场景特征，将原先没有明确界限的用户使用行为划分为不同场景。由于通常人为定义场景类别非常受限，因此我们提出自动场景发现来进一步拓宽海量用户数据下的场景定义，从而引导用户推荐策略在同场景约束下更加聚焦，进而提升用户推荐 app 的准确率。然而，这一任务面临着用户数据稀疏、数据不平衡以及大量离群噪声数据的挑战。

用户数据达千亿级规模，我们通过分析单日共 12 亿条用户数据分析，发现数据呈明显的高频与低频分布特征。高频数据主要包括用户应用使用记录（应用名称、使用时刻、时长等交互数据）以及覆盖率达 99.7% 的 WIFI/蓝牙连接状态，这些数据具有连续性强、稳定性高

的特点，应用使用记录能够可靠地反映用户设备使用习惯，而 WIFI/蓝牙连接状态能够反映出网络环境特征，用户是否在室内或常驻环境。同时，运动状态数据占比 27.33%，能有效反映用户当前活动状态。低频数据中 GPS 经纬度信息虽仅占 0.16%，但仍具有百万规模，可以反映这部分用户的当前场景。同时 POI 数据占 10.49%，其精细的二级分类体系共 136 类可以提供丰富的用户近端预期场景语义信息。深入分析显示，POI 数据的结构化分类（如商场、铁路、医院等）与用户应用使用行为及场景存在显著关联性。基于此，我们确定以应用使用记录和 POI 数据作为核心特征，结合高覆盖率的网络连接状态和中频的运动状态数据反映用户活动状态，同时考虑占比较少的用户 GPS 经纬度信息，构建融合多维特征的用户行为画像体系，以达到对用户当前场景及邻近预期场景的建模。

针对上述挑战，本报告提出了一种高效的场景发现技术方案，首先利用大语言模型对用户数据中一些较易分辨的通用性场景进行标注，再通过 BERT 无监督训练的方法提炼用户应用使用特征。之后通过借鉴对比聚类领域的最新研究进展，采用自监督学习策略实现用户表征学习。该方案通过优化数据处理与表征学习机制，解决了稀疏位置信息记录、数据不平衡以及离群点普遍存在等关键问题，不仅在技术实现上具有较强的可扩展性和稳健性，还能够显著提升场景发现的精准度，为服务和应用推荐系统的智能化提供了有力支持。

（1）自动预处理场景标签

针对现有用户数据中没有场景标签信息，该自动场景标签预处理部分采用大语言模型对现有多模态用户数据进行初步场景标注，具体方案如下：

1. 我们首先手动定义少部分用户数据作为示例来使得模型可以先明确我们需要他根据提供的用户数据得到怎样的场景标注结果，再利用大语言模型的泛化能力得到更多数据标注结果；

2. 选择用户数据中的应用名（com.smile.gifmaker：快手）、应用使用时长、应用交互次数、运动状态、蓝牙 wifi 连接情况、POI 信息（0801000000：机场）等与场景标注关联性较高的信息生成语义化的数据作为大语言模型输入文本信息；

3. 将选择的用户数据转化为可从大语言模型得到期望文本的场景标注的提示词。示例：(1)对于当前时间有应用记录的数据语义化为：“用户在 2025-05-03 13:09:08 正在使用微信（社交通讯应用），持续时间 0.0 秒，此时 POI 信息可能是：快餐厅。WIFI 已连接，蓝牙未连接，运动状态：still。在该时刻前后的服务使用记录如下：在 2025-05-03 12:49:15 时使用微信（社交通讯应用）持续 810.766 秒，在 2025-05-03 13:13:25 时使用抖音（拍摄美化应用）持续 1535.779 秒。使用微信共 810.766 秒，期间交互 1 次；使用抖音共 1535.779 秒，期间交互 1 次”；(2)对于当前时间没有应用记录的数据语义化为：“用户此时 POI 信息是：便利店。WIFI 未连接，蓝牙未连接，运动状态：walking。在该时刻前后的服务使用记录如下：在 2025-05-04 08:38:29 时使用微信（社交通讯应用）持续 243.2 秒。使用微信共 243.2 秒，期间交互 1 次。”

4. 将上述语义化之后的数据作为大语言模型的输入，使用 prompt 内容为：“你是一个数据标注员。现在的任务是对用户当前所处的场景进行预测以便于对用户进行个性化推荐，你需要对每条数据进行场景预测并标注，我们希望你能够进行一步一步的推理，根据这个数据

的语义化信息推断用户可能正在进行的场景，并给出最终结果。有如下要求：(1)快速解答、不需要进行深度思考；

(2)假设场景标注的总分是 100 分，请你给出每条数据与预测场景的对应置信分数；

(3)最终预测的结果后必须以" (end) "结束。

示例：

输入：用户在 2025-05-03 18:40:13 正在使用抖音（拍摄美化应用），持续时间 158.857 秒，此时 POI 信息可能是：未知位置。Win 已连接，蓝牙未连接，运动状态：still。在该时刻前后的服务使用记录如下：在 2025-05-03 18:36:04 时使用抖音（拍摄美化应用）持续 248.721 秒，在 2025-05-03 18:49:36 时使用微信（社交通讯应用）持续 36.629 秒。使用抖音共 248.721 秒，期间交互 1 次；使用微信共 36.629 秒，期间交互 1 次。

输出：情景预测：室内休闲娱乐；置信分数：85(end)

*注意*输出格式严格按照样例”

4. 由于大语言模型的场景标注准确度有限，因此我们采用基于提示词的大语言模型生成结果自检方法。我们通过大语言模型对当前用户数据对应的场景标注给出置信分数，然后手动筛选场景标注置信分数为前 25% 的用户数据作为预先场景标注数据。(示例：(1)情景预测：餐厅用餐；置信分数：90；(2) 情景预测：便利店购物；置信分数：90) 。

5. 将 10 万条标注数据的预测场景进行归纳整理，由于大语言模型 api 输入的限制，将其数据分成 10 个批次分别送入大语言模型分别总结 100 个类别，提示词为：“你是总结归纳员，数据是很多场景的名称，你需要根据其内容，把各种场景总结归纳成 100 个。”，最

终再将得到的有重复的 1000 个左右的场景用同样的 prompt 归纳并通过人工处理得到大概 150 类的场景，将其回溯到原数据作为新的标签。

经过初步利用大模型本身的泛化场景定义能力，我们已经可以得到一些尾端场景，如：婚礼筹备、快递收发、电影院观影、运动健身等。经过标注，包括 POI 信息的用户数据的场景标注置信度相比不包括该信息的用户数据往往具有更高的置信度。由此我们可以得到部分拥有高评分的场景标注的用户数据，其中大部分标注用户数据包括 POI 信息，而其余未标注用户数据则需要经过我们之后的场景发现来分类或聚类，之后我们将依据现有的数据开展模型训练与方法检验以形成最终场景发现方案，并依据场景发现方案开展新场景探索。

(2) 数据预处理

在数据预处理中，我们首先进行数据清洗与归一化处理。对于活动水平较低的记录，这些数据对场景发现的贡献有限，因此需要进行过滤。过滤条件为活动水平大于等于阈值 θ ，该阈值可根据具体场景需求设定，以剔除不重要或无意义的数。同时，为了消除特征尺度差异对模型学习的影响，需要对应用使用时长和位置评分进行归一化处理。归一化公式为：

$$x_{norm} = \frac{x - \mu}{\sigma}$$

其中 x 为原始数据值， μ 为均值， σ 为标准差。归一化后，数据分布更加适合模型的学习需求。此外，稀疏的位置信息需要通过语义增强进行补充。我们为位置信息添加语义标签（例如“购物中心”、“办公楼”等），以提升数据的上下文信息丰富度。这些数据预处理步骤有效提高了数据质量和信息密度，为后续场景发现和推荐模型的优化提供了坚实的基础。

(3) 特征表示

针对用户交互行为数据的多模态特性，我们设计了一套复合特征编码方案，以有效融合类别型、语义型和时序型三类异构特征：

应用特征：包括应用的嵌入表示 \mathbf{e}_{app} 、应用使用时长 d_{app} 和用户与应用的交互次数 c_{app} 。这些特征用于描述用户与不同应用的行为模式，并为模型提供与用户应用偏好相关的丰富信息。对应用使用时长进行数据预处理中的标准化以避免由于应用间的不同特性导致使用时长差异大（如视频应用 vs 短信应用）而产生过拟合。应用使用时长 d_{app} 和用户与应用的交互次数 c_{app} 结合以反映应用对当前场景的重要程度。

位置特征：部分用户数据中的经纬信息需转换为语义文本信息。我们采用经纬逆编码的方法将经纬信息转换为街道地点，然后将无实际语义意义的街道地点信息作为大语言模型输入转换为对应地点类别语义名称以更好适配任务场景语义。将当前位置和兴趣点语义嵌入加权表示为 \mathbf{e}_{loc} 。这些位置特征结合了地理信息和语义信息，用于更准确地刻画用户的行为场景。

时间特征：包括一天中的小时信息，通过正弦和余弦编码进行周期性表示，公式为：

$$\mathbf{H}_{sine} = \sin\left(\frac{2\pi \cdot hour}{24}\right), \mathbf{H}_{cosine} = \cos\left(\frac{2\pi \cdot hour}{24}\right).$$

这种时间特征 \mathbf{h}_t 的处理方式能够捕捉时间周期性变化，为模型提供时间维度上的行为规律信息。

这些特征的多维度分类和编码方法相结合，为模型提供了应用、位置和时间三个方面的丰富输入特征，有助于提高场景发现与用户行为建模的精确性。

(4) 用户交互序列表征预训练方法

针对移动计算场景下的用户行为分析需求，本研究提出一种基于 Transformer 架构的序列表征预训练方法。用户与应用服务的交互行为天然具有时序依赖性和场景相关性特征，这为自监督学习提供了理想的数据基础。通过分析发现，用户在不同时空场景下的应用使用模式呈现显著规律性，这种规律性使得基于上下文预测的预训练任务具有可行性。

本方法创新性地将用户交互序列建模为多模态 Token 序列，每个交互事件 Token 由多种模态异构特征构成：应用标识特征(App Name)、时序特征（交互时刻的时间戳）、持续时间特征（使用时长）、位置信息（可能缺失）是否有 WIFI 使用等。受 BERT 的掩码语言模型启发，如图 1.1 我们设计了基于动态掩码的预训练策略：

1. 输入序列首部添加可学习的[CLS]标志位
2. 随机掩蔽 15%的交互 Token（采用全掩码策略）
3. 模型通过双向 Transformer 编码器学习上下文感知的序列表征
4. 预训练目标：联合优化以下任务：
 - 被掩蔽 App ID 的分类预测（主任务）
 - 交互时间的回归预测（辅助任务）
 - 使用时长的回归预测（辅助任务）

该方法具有下优势：

- 异构特征融合：通过联合编码多模态交互特征，有效捕捉行为模式的复合规律
- 场景感知能力：利用双向注意力机制建模长程依赖关系，自动发现场景相关的使用模式
- 数据效率：自监督范式可充分利用海量无标注行为数据

。迁移友好：[CLS]位置的聚合表征可直接用于下游场景分类、用户画像等任务。

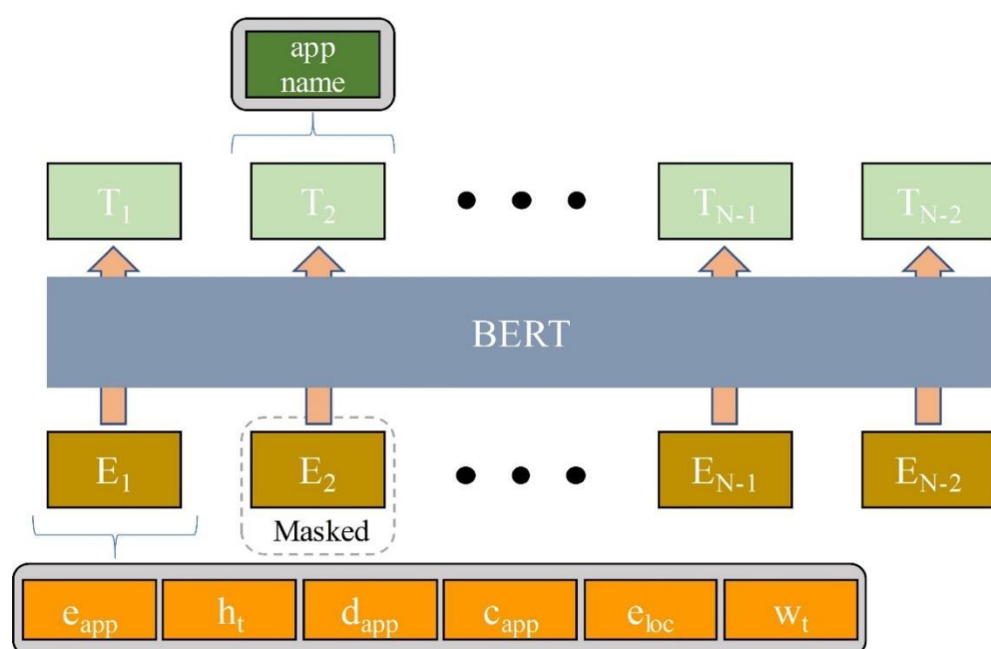


图 1.1: 基于掩码预测的交互序列表征预训练示意图

(5) 数据增强

本项目使用对比学习的方式学习用户特征，因此需要定义和构建正样本对。在有监督场景下，正样本对可以被定义为针对同样场景的用户数据。在无监督场景下，需要通过数据增强的方式获得正样本对。针对用户 APP 和位置数据的对比学习，可以通过多种数据增强方式提升数据多样性和模型鲁棒性。在 APP 数据方面，可以通过类别扰动、时长和交互次数的随机扰动以及组合应用模拟，生成更多样化的用户行为数据。在位置数据方面，可以通过位置嵌入的微小扰动、语义上下文的替换以及邻近位置的模拟，增强位置信息的表达。

(6) 特征学习

本项目提出了两种用于场景发现的用户特征学习方法。第一种方法结合已标注场景的用户数据和无标注数据进行学习，称为广义场景

发现。第二种方法是一种无需依赖场景标签的特征学习范式，仅利用用户数据即可完成学习，称为自监督用户特征学习。

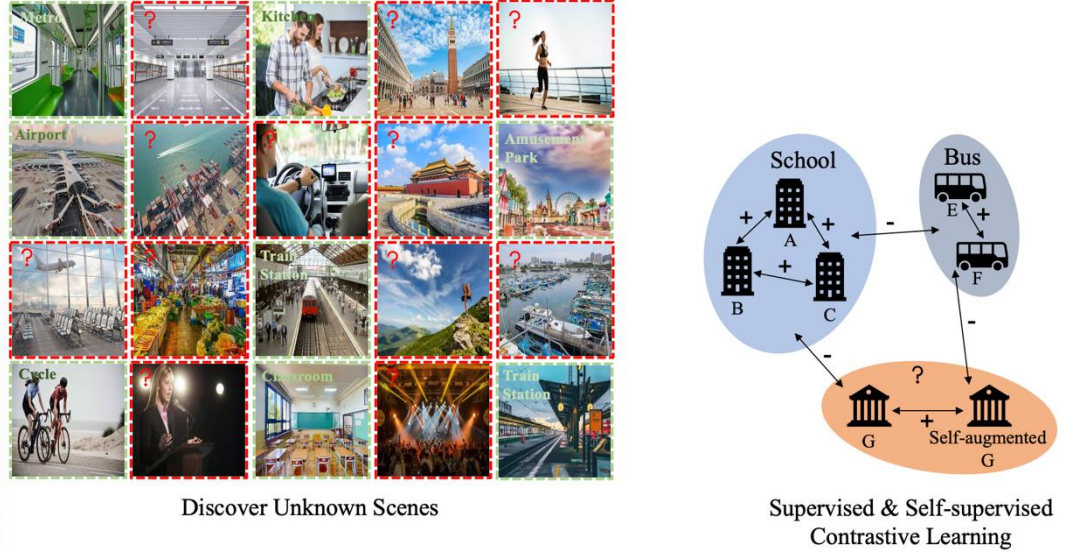


图 1.2: 广义场景发现示意图

广义场景发现:

该方法旨在基于已标注数据来推理出其他无标注数据的场景分类，无标注数据的场景既可能是基于已标注数据的场景，也可能来源于全新的未知场景。因此广义场景发现需要解决两个问题：（1）将已标注数据与具有相同场景的未标注数据分类；（2）将未知场景的未标注数据聚类。用户数据 \mathcal{D} 中包含已标注数据 $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}_L$ 和无标注数据 $\mathcal{D}_U = \{(x_i, y_i)\}_{i=1}^M \in \mathcal{X} \times \mathcal{Y}_U$ ，其中 $\mathcal{Y}_L \subset \mathcal{Y}_U$ 。在训练期间我们仅能获得 \mathcal{Y}_L ，因此我们的目标是通过 \mathcal{Y}_L 推理出 \mathcal{Y}_U 。对于已标注数据 \mathcal{D}_L 和无标注数据 \mathcal{D}_U ，我们分别采用有监督和无监督对比学习。

考虑用户数据 x_i 及对应的增强数据 x'_i ，无监督对比损失函数可以表示为：

$$\mathcal{L}_i^u = -\log \frac{\exp\left(z_i \cdot \frac{z'_i}{\tau}\right)}{\sum_n \mathbb{1}_{[n \neq i]} \exp\left(z_i \cdot \frac{z_n}{\tau}\right)},$$

其中 $\mathbf{z}_i = \mathbf{f}(\mathbf{x}_i)$, $\mathbb{1}_{[n \neq i]}$ 表示指示函数用以约束 $n \neq i$, τ 是温度系数, \mathbf{f} 表示特征网络。同时, 有监督对比学习损失函数可以表示为:

$$\mathcal{L}_i^s = -\frac{1}{|N(i)|} \sum_{q \in N(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_q / \tau)}{\sum_n \mathbb{1}_{[n \neq i]} \exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)},$$

其中 $N(i)$ 表示其他同数据 \mathbf{x}_i 有相同标注的数据集。因此最终的损失函数可以表示为:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}^u + \mathcal{L}^s,$$

其中 λ 表示损失函数组合权重系数。

在推理阶段, 我们采用 K 均值算法将无标注用户数据特征聚类以实现相同场景的分类, 进而发现无标注场景。

基于聚类的自监督特征学习:

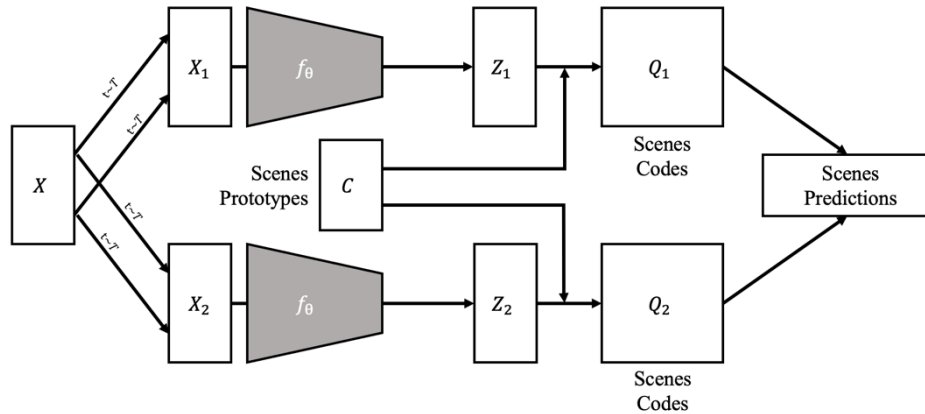


图 1.3: 基于原型的自监督学习示意图

该方法旨在通过全无标签的用户数据推理出相应的场景分类。在训练阶段, 用户数据 \mathcal{D} 中仅存在无标注的数据集 $\mathcal{D}_U = \{(\mathbf{x}_i)\}_{i=1}^M \in \mathcal{X}$, 用户本身所代表的类别属性不可知, 这种场景类别需要通过最终的推理过程得到。因此在特征学习阶段仅能通过单个用户本身的数据及其增强数据来学习用户自身特征。在传统自监督的特征学习过程中, 因缺乏监督而产生无意义特征, 导致学习过程不易聚类。针对这一局限性, 我们采用基于原型的自监督学习方法表征单个用户的各项数据。

具体来说, 基于原型 $\mathbf{c} \in \mathbb{R}^{D \times K}$ 的自监督学习引入聚类中心特征来明确聚类范式, 进而约束表征过程, 使表征模型 f_{θ} 能面向聚类中心提取有意义的特征; 随后, 结合聚类中心特征和用户数据特征来优化单通路的预测结果, 即用户数据编码 Q ; 最后, 通过结合多通路的用户编码得到最终的场景预测结果。其中场景原型 \mathbf{C} 是被所有特征通路所共享的, 因此可以实现单个用户的多通路数据特征之间的相互预测, 以达到单个用户的自监督学习。

同时, 考虑到用户数据中存在离散型长尾分布, 而长端数据在基于原型的自监督学习中会影响尾端数据表征, 因此在训练阶段交替地使用层级 K 均值迭代优化算法来消除长尾分布不平衡现象, 以增强尾部数据表征并消除无意义离群数据的影响。

自监督学习:

采用基于原型的自监督对比学习来表征全无标签的用户数据。该方法引导用户单通路特征预测其他通路特征对应的场景。通过这种方式实现多通路相互预测场景, 进而约束特征学习过程。其目标函数可以定义为:

$$\mathcal{L}(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t),$$

其中 \mathbf{z}_t 和 \mathbf{z}_s 是用户数据及其增强数据的特征, 而 \mathbf{q}_t 和 \mathbf{q}_s 是表示预测结果的两个编码。单个损失分量可以描述为:

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k q_s^{(k)} \log p_t^{(k)}, \quad p_t^{(k)} = \frac{\exp(\mathbf{z}_t^T \mathbf{c}_k / \tau)}{\sum_{k'} \exp(\mathbf{z}_t^T \mathbf{c}_{k'} / \tau)}$$

其中 τ 表示温度系数, $\mathbf{c}_k \in \mathbb{R}^D$ 表示第 k 个原型。编码 $Q = [q_1, \dots, q_B]$ 的作用是最大化特征与原型之间的相似度, 并在熵约束 $H(Q) = -\sum_{ij} Q_{ij} \log Q_{ij}$ 下逐渐进行优化, 可以表示为:

$$\max_{Q \in \mathcal{Q}} \text{Tr}(Q^T \mathbf{C}^T \mathbf{Z}) + \varepsilon H(Q),$$

其中 Q 采用软编码即不采用一般的取整方式, 可以表示为:

$$Q = \text{Diag}(\mathbf{u}) \exp\left(\frac{C^T Z}{\varepsilon}\right) \text{Diag}(\mathbf{v}),$$

其中 \mathbf{u} 和 \mathbf{v} 是 \mathbb{R}^K 和 \mathbb{R}^B 中重归一化的向量，使用迭代的 Sinkhorn-Knopp 算法通过少量的矩阵乘法来计算重归一化向量。

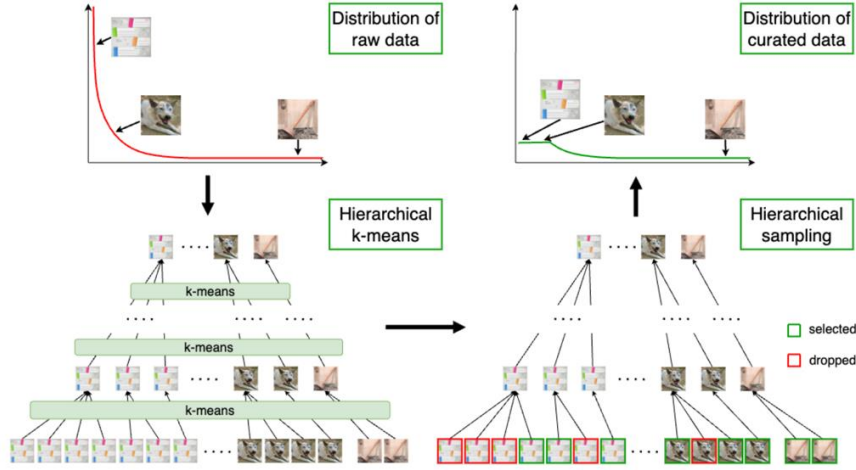


图 1.3: 长尾数据迭代优化示意图

迭代优化:

K 均值算法能够使用距离度量对未标注数据进行分类。然而，在不平衡的情况下对占据主导地位的数据进行聚类会影响度量指标。因此，层次 K 均值算法更适合拟合不平衡分布。假设存在一个均匀概率分布 U ，在数据分布 P 的集合 R 上，其密度为 $\frac{1}{|R|} \mathbb{1}_R$ ，其中 R 表示支撑集，定义为 $R = \{x \mid p(x) > 0\}$ ， P 表示数据分布。因此，一个理想的 U 需满足：

$$\text{Dist}_{\text{KL}}(G \parallel U) \leq \text{Dist}_{\text{KL}}(P \parallel U),$$

其中 G 是聚类中心分布，它服从未归一化密度 $p^{\frac{d}{d+2}}$ ，该密度更接近 U 。为了符合期望的分布，距离度量修改为 $d(x, y) = \|x - y\|^s$ ，根据实证实验 $s > 2$ 。因此，提出了一种经过 T 次迭代的层次 K 均值算法以渐近拟合 G 。

重采样:

我们首先在层级 t 处使用 K 均值算法得到聚类集 L_t 。然后，为了避免采样过程中数据呈指数级减少，我们对 L_t 中每个质心最接近的 r_t 个数据点进行重采样，以形成子集 D ，并在 D 上实施 K 均值算法来更新聚类集 L_t 。由于 D 中的数据分布更接近 U ，因此聚类集 L_t 变得更接近 U 。

确定数据层级：

给定数据中的用户规模 N 以及聚类规模 s_j (其中 $1 \leq j \leq k$)，我们通过在区间 $[0, N]$ 内进行二分搜索，找到使 $|N - \sum_j^k \min(n, s_j)|$ 最小的整数 n 。尽管最高层级上的分布在支撑集 R 上是均匀的，但数据量可能会少于所需数量。为了解决这个问题，我们从最高层级到底层层级进行数据采样，每个聚类中取 n 个点。最终，我们得到了一个消除不平衡影响的数据分布。在整个训练过程中交替地采用基于原型的自监督学习和层级 K 均值平衡算法，不断优化表征结果和数据分布情况。

未知场景定义：

考虑到场景发现后除第一阶段通过大语言模型预处理得到的已定义场景外，本算法会基于用户特征聚类而产生新的场景，为同先前的语义化场景标注对齐，我们采用基于大语言模型的未知场景定义，具体步骤如下：

1. 将同类场景下的用户特征进行回溯与用户数据相对应；
2. 将的用户数据中的应用名、应用使用时长、应用交互次数、使用位置 POI 信息等与场景标注关联性较高的信息转换为文本信息；
3. 将用户数据转化为可从大语言模型得到期望文本的场景标注的提示词。示例：用户在 2025-01-26 10:38:09：使用计算器应用 16 秒，

2025-01-26 10:40:36: 使用计算器应用 5.7 秒, 2025-01-26 10:40:43: 使用通讯录应用 1.4 秒, 2025-01-26 10:40:45: 使用通话界面应用 25.2 秒, 2025-01-26 10:41:10: 使用通讯录应用 27.0 秒, 2025-01-26 10:41:39: 使用通讯录应用 2.4 秒, 2025-01-26 10:41:42: 使用通话界面应用 45.9 秒, 2025-01-26 10:36:14: 使用 qq 音乐持续 2.4 秒, 2025-01-26 10:36:19: 使用 qq 音乐持续 3.5 秒。使用通讯录 app 共 31 秒, 期间交互 3 次; 处于通话界面共 6 分 17 秒, 期间交互 2 次; 使用计算器 app 共 21.7 秒, 期间交互 2 次; 使用 qq 音乐 app 共 6 秒, 期间交互 2 次。

4. 为得到最优场景描述文本, 选择同类场景下的出现频次最高的场景描述以得到未知场景的定义。

1.3 实验分析

通过大语言模型对 10 万条初标注的数据进行归纳总结出 150 个类别, 具体为:

一、居家生活类

居家休闲娱乐/居家社交沟通/居家观影/居家阅读/居家音乐欣赏/
居家智能控制/整理照片/居家烹饪/居家睡眠/居家用餐

二、社交互动类

朋友聚会/约会社交/商务社交/线上社交/电话通讯/视频通话/社交
媒体浏览/社交媒体互动/社交分享/社交用餐/社交活动/社交浏览/
社交记录

三、餐饮美食类

餐厅就餐/快餐店用餐/休闲馆休闲/酒吧社交/外卖点餐/外卖配送
/早餐时间/餐饮支付/烹饪学习

四、购物消费类

超市购物/商场购物/便利店购物/线上购物/购物比价/购物支付/购物浏览/建材选购/汽车选购/珠宝选购/药店购物/市场交易/购物等待

五、旅游出行类

旅游观光/旅游导航/旅游社交/旅游购物/旅游用餐/旅游住宿/旅游交通/旅游规划/旅游预订/旅游信息查询

六、交通通勤类

自驾出行/公交出行/地铁通勤/火车出行/高铁出行/飞机出行/网约车/代驾工作/步行通勤/骑行通勤/停车场/加油站/交通导航/出行规划/交通等待

七、办公商务类

工作办公/商务出差/会议交流/工作沟通/工作间隙/文件管理/订单处理/库存管理/求职招聘/业务咨询

八、教育培训类

在校学习/在线学习/图书馆学习/培训机构学习/驾校学习/考试备考/技能培训/亲子教育/语言学习

九、医疗健康类

医院就诊/医院陪护/药店购药/健康监测/养生保健/健身运动/瑜伽练习/按摩护理/美容护理/美发服务/宠物美容

十、休闲娱乐类

影院观影/KTV 娱乐/网吧游戏/游乐场娱乐/公园休闲/运动健身/游

泳馆/台球娱乐/棋牌活动/音乐欣赏/游戏娱乐/短视频浏览/直播观看/摄影创作/露营体验

十一、公共服务类

银行服务/政务办理/快递收发/物业管理/宗教活动/安防监控/紧急救助

十二、设备操作类

设备调试/照片编辑/视频剪辑/录音操作/遥控操作/监控查看

十三、特定场景类

婚礼筹备/体育赛事/戏剧演出/博物馆参观/产品体验/维修等待/洗衣店洗衣

十四、特殊时段类

早晨准备/通勤时段/午休时间/下午茶时间/晚间放松前准备/夜间活动

后续将这 150 类作为数据标签，以检验场景发现的训练结果

1.4 可行性分析

在数据预处理阶段，通过清洗低活跃度数据（如过滤单次点击行为）和标准化处理（如归一化应用使用时长），有效解决了特征尺度差异问题。同时，利用高德地图 API 对稀疏位置信息进行语义增强（如将坐标解析为“写字楼”“商场”），补充了上下文信息，缓解了位置数据缺失对场景划分的影响。为进一步提升模型鲁棒性，方案引入对比学习策略，通过对应用使用时长、位置嵌入进行随机扰动或邻近

位置模拟，生成多样化的正样本对，从而增强模型对噪声和长尾数据的适应能力。

在特征表示与融合层面，方案设计了复合编码方案以整合多模态数据。例如，应用特征通过嵌入表示捕捉用户偏好，位置特征结合地理坐标与兴趣点（POI）的加权语义嵌入，时间特征则通过正弦/余弦编码捕捉周期性规律。这种多维特征融合为模型提供了全面的场景刻画能力。此外，基于 Bert 架构的预训练方法通过掩码预测任务（如预测被遮蔽的应用 ID、交互时间）学习用户行为序列的上下文表征，有效建模了长时序依赖关系，例如发现用户在通勤场景中高频使用导航与音乐应用的连贯模式。

在模型训练策略上，方案结合了自监督与对比学习的优势。广义场景发现方法利用少量标注数据（如已知“工作”“娱乐”场景）引导无标注数据的分类，同时通过层级 K 均值聚类迭代优化，逐步识别未知场景。针对长尾分布问题，原型聚类方法通过重采样平衡数据分布，优先保障低频场景的聚类效果。例如，在迭代过程中对尾部场景样本进行重复采样，避免其被主导场景覆盖。这种动态优化机制显著提升了模型对复杂数据分布的适应性。

总体而言，该方案通过创新的数据处理、特征融合与自监督学习框架，系统性解决了移动用户数据场景发现的核心痛点。其技术优势体现在三方面：一是高效应对数据稀疏性与噪声问题，通过语义增强与对比学习提升数据质量；二是场景感知能力强，多模态特征与 Bert 架构精准捕捉用户行为模式；三是可扩展性高，自监督学习减少对标

注数据的依赖，适合海量用户数据的动态扩展。因此，该方案不仅在技术层面具备可行性，还为个性化推荐、智慧城市等应用场景提供了可落地的解决方案。

2. 用户场景识别与分类

2.1 洞察总结

场景识别技术的发展，特别是在基于多源数据融合模型架构和深度学习方法上的创新，使得该领域逐渐成为计算机视觉、自然语言处理与地理信息科学交叉的核心方向之一。

多模态数据融合是国外场景识别领域的研究热点之一。研究表明，地理数据与文本数据的联合建模可以有效提升场景分类精度。例如，Geo-Text 数据的分析通过将地理坐标与文本（如地理标记的推文、新闻报道等）结合，能够为场景识别提供更多维度的信息。在这一过程中，文本数据的自然语言处理（NLP）技术（如命名实体识别（NER）和情感分析（Sentiment Analysis））与空间数据处理（如地理标签解析）的结合，已经成为提高模型精度的关键。

Transformer 架构的引入，为场景识别带来了革命性进展。尤其在时序建模方面，Transformer 比传统的 RNN、LSTM 模型更擅长捕捉长距离依赖。通过自注意力机制，Transformer 能够同时关注输入序列中所有位置的关系，有效处理多模态信息和长时序数据。例如，BERT4Rec 模型利用 Transformer 架构对用户行为进行序列建模，并结合多模态特征（如行为序列、位置和时间信息），在推荐系统中的应用取得了极大的成功。

在处理时空数据时，国外的研究也提出了时空聚类 and 空间自相关分析方法，利用时空数据分析来构建用户的行为图谱。研究者如 Li et al. (2012) 提出，通过核密度估计（KDE）方法分析地理空间数据，能

够更准确地获取用户在不同场景下的行为模式。结合时空聚类技术，他们能够从大规模社交媒体数据中自动提取出多个场景的空间分布特征。

国内的研究在场景识别中的技术应用同样取得了显著进展。许多企业，尤其是阿里巴巴、腾讯等，已经实现了将深度学习与大数据分析相结合，通过用户行为数据、地理位置信息以及社交媒体内容，实现了精准的场景分类和推荐。这些企业依赖于基于 Transformer 的序列建模，并结合用户画像与行为轨迹分析进行深度优化。例如，在电商推荐系统中，利用用户的浏览行为、地理位置、购买历史等信息，模型能够实时动态地为用户提供个性化推荐。

在学术界，国内的研究则侧重于利用多源异构数据，提出基于图的场景分类模型。例如，图卷积网络（GCN）被广泛应用于场景识别中，通过构建行为图谱，对用户与场景之间的关系进行建模。这种方法有效解决了单一行为序列中信息缺失的问题，并提升了场景识别的准确性。

随着技术的不断发展，场景识别逐渐向以下几个方向发展：

自监督学习与预训练模型。自监督学习技术，通过设计辅助任务让模型从大量未标注数据中学习有效特征，已经成为当前模型训练的核心。例如，BERT 和 GPT 系列模型的成功为无监督预训练提供了重要借鉴。BERT 通过掩码语言模型（Masked Language Modeling, MLM）任务，使模型能够在没有显式标签的情况下学习上下文关系，进而提升了在下游任务中的表现。

在场景识别中，自监督学习的应用能够有效解决标签稀缺的问题，尤其是在复杂场景的分类任务中，利用无标注数据生成高质量的特征表示，能够大大提高模型的泛化能力。

多模态学习与深度融合。多模态学习正在成为场景识别中的重要方向。通过融合不同类型的特征，如图像、文本和地理位置信息，模型能够更全面地理解场景信息。在此过程中，注意力机制和图神经网络（GNN）等技术也被广泛应用于不同模态信息的融合与建模。例如，基于 Transformer 架构的多模态融合模型，通过自注意力机制（Self-attention）对各模态之间的关系进行建模，有效提升了场景分类的精度。

特别是在处理 Geo-Text 数据时，结合地理信息和自然语言文本，采用地理词条解析（Geoparsing）技术，可以有效提取地理信息中的上下文语义。这种文本与空间信息的深度融合方法，已经成为精准场景识别的重要技术。

时空建模与深度时序分析。时空建模是场景识别中的一项重要技术，尤其是在处理具有时间和空间维度的数据时，时空建模能够帮助模型更准确地捕捉场景中的动态变化。例如，基于 Transformer 架构的模型能够处理时间序列数据并捕捉时间上的长距离依赖关系，极大提升了时空信息的理解能力。

在实际应用中，研究者们采用时空聚类算法和时空深度学习模型，对用户行为数据进行建模。例如，基于时空卷积神经网络（TCN）的方法，能够处理时空数据中的多层次信息，从而为场景分类提供更加准确的时空推理能力。

尽管场景识别技术在多个领域取得了显著进展，但在实际应用中仍面临许多挑战。

数据质量与标注稀缺。场景识别中的数据主要来自于社交媒体、传感器数据、用户行为数据等多源异构数据，这些数据通常存在质量

参差不齐的问题。尤其是在标注稀缺的情况下，如何利用无监督学习和自监督学习方法，最大化地利用未标注数据，成为当前的研究重点。

模型的可解释性不足。虽然深度学习模型在精度上表现优异，但其“黑箱”特性使得模型的决策过程难以理解。可解释性问题尤其在医疗、金融等领域显得尤为重要。未来的研究将更多关注模型透明性和决策过程可追溯性的提升，通过注意力机制和可解释的 AI 模型，使得深度模型在复杂场景下更具信任度。

跨领域迁移能力。当前模型往往在特定领域的训练数据上表现优异，但跨领域应用时其表现可能会大幅下降。如何设计具有较强迁移能力的通用场景识别模型，成为解决多领域应用的关键问题之一。

未来的场景识别技术将进一步向更加精准、实时的方向发展。特别是随着边缘计算和 5G 技术的普及，未来的场景识别不仅会向移动端推移，还会向实时推理和低延迟处理方向发展。通过大规模预训练模型、深度融合、时空建模等技术，场景识别将在个性化推荐、智能城市、医疗健康等领域展现出巨大的潜力。

2.2 基于应用数据的用户行为特征建模

(1) 基于用户轨迹数据的整理与伪标签生成

手机用户的行为包含应用的使用记录，地点数据，设备状态，时间（时间戳、是否为工作日），网络连接状态（WiFi、蓝牙、基站等）和应用来源等。地点数据可通过基站的语义信息、地理坐标及辅助状态字段综合描述。基站信息包含基站的地理位置和基站名字，基站的使用序列能够确定用户的移动轨迹、停留点、会晤事件等。

用户在不同的位置上使用相同应用会留下不同的使用痕迹，其在一定程度上也勾勒了用户的部分行为特征，如图 2.2 所示，靠近公司的位置信息会呈现工作性质的应用使用频繁，非工作时间使用时间稀

少的特征，相反，靠近住宅的位置信息会呈现工作性质的应用使用稀少，非工作时间使用频繁的现象。

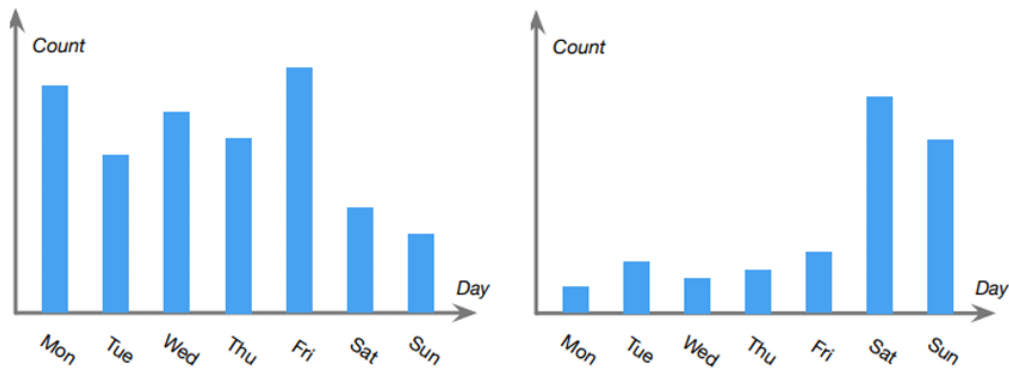


图 2.2：用户在不同位置信息访问相同 App 频率，左图为公司附近，右图为住宅附近

为了更好地支撑本项目的模型训练及特征工程需求，需对原始数据库中的用户 30 分钟内 App 使用跟踪数据进行系统性的数据处理。原始数据以表格形式存储，其中每条记录包含多个字段，尤其重要的是 content 字段，该字段内嵌约 45 个以竖线“|”分隔的具体内容字段，如“serviceUsage”、“serviceExposure”、“serviceSource”、“motionState”、“sceneIds”等，内容丰富但存在数据缺失现象。

针对用户的数据特点，本项目首先需要基于“udid”字段进行同一用户数据的归并处理，即将所有属于同一“udid”的记录需按照时间范围收集起来，将其中“content”字段以“#####”符号连接，形成新的合并字段“merged_content”，便于后续批量拆分；同时记录每个 udid 对应的原始数据条数，新增“count”字段作为用户活跃度的参考指标。合并后的数据需进一步细化处理，针对每条“merged_content”内容，首先按“#####”拆分还原为单条记录，再对每条记录基于竖线“|”符号进行二次拆分，解析出 54 个具体字段，并将各字段分别映射至独立列中，确保字段与数据对应准确无误，形成结构化、标准化的数

据表。由于“content”中的字段存在不同程度的缺失，因此在完成拆分后，需要对每一个字段计算其缺失率，统计全量数据中每列缺失值占比，并生成缺失率统计表，按缺失率由高到低排序，为后续特征筛选提供依据。一般来说，缺失率过高（如 90% 以上）的字段可考虑剔除或填充处理，以保证模型训练的数据质量和鲁棒性。此外，为了丰富数据标签体系并加快模型训练进度，本项目计划基于当前结构化数据，使用预训练大模型对每条数据样本进行初步的场景推断与预标注，结合“motionState”、“sceneIds”、“connectedDeviceList”、“wifi”状态等多字段特征综合分析，自动生成软标签“pseudo_label”字段，形成带有初步场景标签的样本集，为后续监督学习模型训练提供高质量伪标签。整体的数据处理流程需要模块化设计，并将各步骤封装成独立处理脚本，形成如 data_merge.py、feature_split.py、missing_rate_analysis.py、pseudo_label_generation.py 等脚本文件，以提高工程开发的可维护性、可复现性和扩展性，确保在不同版本数据或不同批次数据更新时能够快速适配和处理，为模型训练打下坚实的数据基础。

（2）特征字段筛选与数据加载模块设计

在完成初步的数据处理与字段拆分后，经过对各字段缺失率、信息量、业务相关性及模型输入可行性进行综合评估，本项目最终选择“wifiConnectedState”、“longitude”、“latitude”、“unionPreServiceUsages”、“unionPostServiceUsages”、“unionFutureServiceUsages”以及“hwPoiInfo”这七个字段作为后续模型训练的核心输入特征。“wifiConnectedState”字段能够有效反映用户所处环境及网络连通性，对场景分类具有高度指示性；“longitude”与“latitude”作为地理位置信息，是推断用户场景地理特征不可或缺

的基础特征，且在配合高阶特征时能极大提升场景识别准确率；“unionPreServiceUsages”、“unionPostServiceUsages”与“unionFutureServiceUsages”分别对应用户在观察时间窗口之前、之后以及预测时刻可能涉及的应用使用信息，能够从行为轨迹角度捕捉用户兴趣偏好及活动模式变化，极大丰富了模型对于时间动态特征的感知能力；“hwPoiInfo”字段包含了用户所处位置关联的 POI (Point of Interest) 信息，如商场、车站、住宅区等，是直接表征环境语义的强特征。因此，以上七个字段既覆盖了物理空间特征，又融入了行为特征，并结合了环境语义特征 (POI)，在特征多样性与表达能力上形成了良好的互补，能为后续的场景分类模型提供充分、有效的输入支撑。

基于上述特征选择，数据加载模块需设计为结构化地读取并输出上述七个字段，确保数据在送入模型前的正确解析与规范处理，如图 2.3 所示，采用 BERT 模型编码，获取高维语义向量。在具体实现上，该部分自定义 Dataset 类，初始化时完成数据文件读取与必要的字段筛选，将每条数据的“wifiConnectedState”数值化处理（如 0/1 表示连接状态），将“longitude”与“latitude”归一化至统一量纲，对“unionPreServiceUsages”、“unionPostServiceUsages”、“unionFutureServiceUsages”进行标签编码，同时对“hwPoiInfo”字段提取关键词或类别编码，形成固定长度的数值输入向量；此外数据加载模块支持 batch 加载，shuffle 打乱，以及按需支持异步加载提升训练性能。整体设计需注重模块化、扩展性及高效性，便于后续特征增减调整与不同模型框架适配，打下坚实的基础。

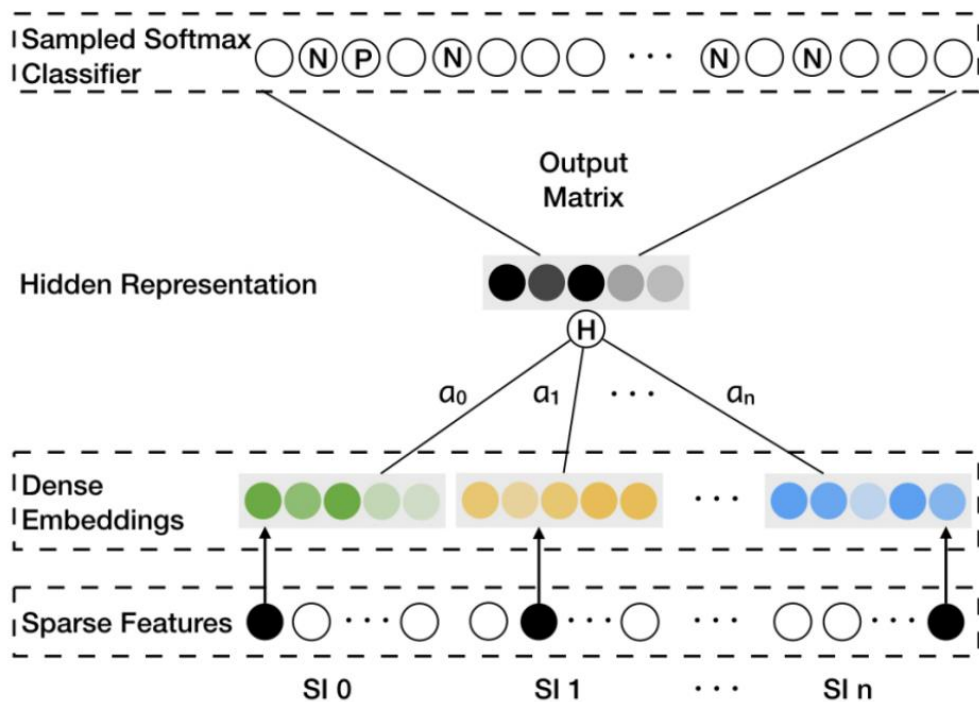


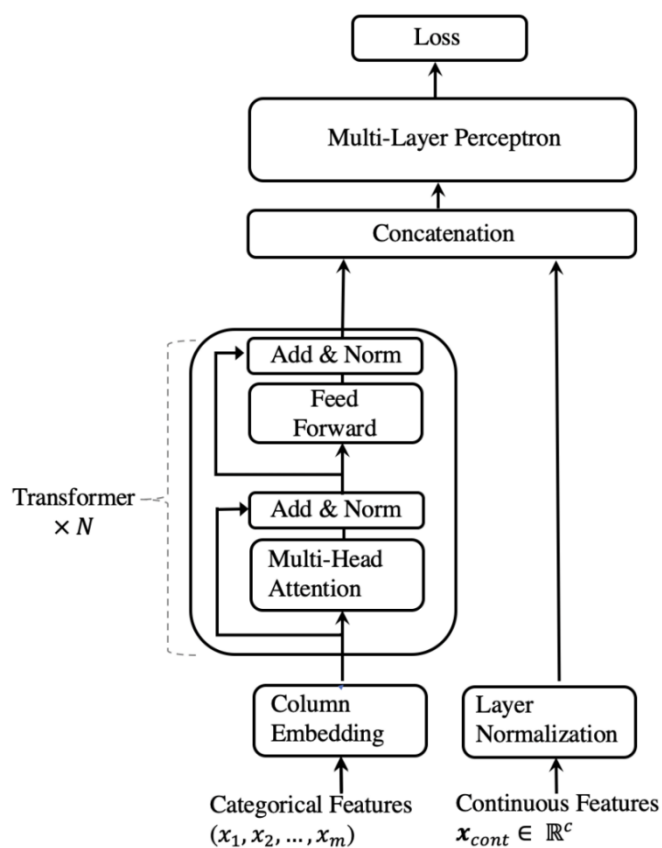
图 2.3 基于应用数据的用户表征示例

2.3 基于用户行为特征建模的用户场景识别与分类

基于前述数据处理与特征筛选结果，为有效建模不同类型输入特征之间的复杂关系并充分挖掘高阶交互特性，本项目决定采用 TabTransformer 作为核心模型框架对结构化数据进行场景分类建模。TabTransformer 通过引入 Transformer 自注意力机制对表格数据中的类别特征进行编码处理，能够显著提升模型在捕捉特征交互关系与非线性表达能力方面的性能，尤其适合本项目中既包含类别型特征，如“wifiConnectedState”、“hwPoiInfo”、“unionPreServiceUsages”等，又包含连续型特征，如“longitude”、“latitude”的混合型输入场景。具体而言，“wifiConnectedState”字段在预处理阶段将作为二值类别输入，“hwPoiInfo”字段提取出的 POI 类别编码，以及“unionPreServiceUsages”、“unionPostServiceUsages”、“unionFutureServiceUsages”经过标签编码处理后也作为类别向量输入，均将通过 TabTransformer 内部的嵌入层转化为固定维度的稠密向

量；“longitude”与“latitude”则作为连续数值特征，直接并入后续多层感知机（MLP）部分进行处理。在模型架构设计上，类别特征首先经过 Embedding 层后，通过一组堆叠的 Transformer Encoder Blocks 进行特征间自适应交互建模，自注意力机制能够在无需明确指定特征交互规则的情况下，自主学习到重要特征组合关系，有效提升模型的泛化性能和表示能力。随后，将 Transformer 编码后的输出向量与连续型特征拼接，通过标准化的全连接 MLP 层进行进一步的特征融合与分类预测，输出最终的场景类别概率分布。在训练流程上，采用交叉熵损失作为优化目标，优化器选择 AdamW 以兼顾训练稳定性与收敛速度；为防止过拟合，Transformer Block 与 MLP 部分均引入 Dropout 机制，Dropout 率初步设定为 0.2。训练过程中按照统一的 batch size 进行训练，每轮 epoch 后在验证集上评估 accuracy、macro-F1 等指标，实时监控并保存最优模型参数节点。

采用 TabTransformer 的最大优点在于，其一方面能够充分利用类别型特征的高阶语义关系，通过自注意力机制自动建模复杂特征交互，无需手动特征工程，大幅降低工程复杂度；另一方面相比传统深度学习方法如 DNN、MLP，TabTransformer 在类别稀疏、特征异质性场景下有更好的适应性和更强的泛化能力，尤其适合本项目中应用使用记录、POI 类别、连接状态等强语义类别特征丰富、连续特征数量适中的数据结构。此外，由于 TabTransformer 对输入结构要求较为灵活，未来若有特征扩展或数据源更新，可通过简单增加 Embedding 表或调整输入通道数快速适配，无需大规模修改模型主干，具备良好的扩展性和工程落地性。因此，本项目采用 TabTransformer 作为主要建模方案，将极大提升结构化轨迹数据的建模效率与分类准确率，为最终场景分类系统提供坚实的模型基础。



2.4 模型评估与结果分析

(a) (b)

图 2.5 模型训练损失(a)与准确率(b)变化情况

从上图 (a) 中可以看到，训练损失（蓝色曲线）与验证损失（橙色曲线）均随训练轮数（Epoch）增加而逐渐下降，在大约第 30 轮之后趋于平稳。模型在训练集与验证集上的损失表现相近，说明当前模型并未发生明显的过拟合现象，具备一定的泛化能力。展示了模型在训练集与验证集上的分类准确率变化。从上图 (b) 可以观察到，模型准确率在初期迅速提升，并在第 15 至第 25 轮之间达到相对稳定状态。训练与验证准确率始终保持接近，进一步说明模型具有良好的泛化表现。最终，在 1000 条数据的基础上进行初步训练后，模型在验证集上的平均分类准确率达到 0.71。虽然该结果尚不能代表模型在大规模数据集下的最终性能，但已经表明该模型具备初步的场景识别能力。

目前该模型表现良好，尚未出现明显的过拟合或欠拟合现象。未来该部分的工作将包括：增加训练数据量，以进一步提升模型的泛化能力与准确率；引入更多特征增强模型输入；尝试更复杂的模型结构以更好捕捉用户行为时序特性；使用更严谨的交叉验证方法进行模型效果评估。

2.5 可行性分析及项目周期安排

本项目提出的用户场景识别方法，旨在基于用户在单时间片内的行为特征进行多标签分类，识别其所处的真实使用场景。该模块以多源行为数据为输入，采用基于 TabTransformer 架构的深度学习神经网络进行建模，并以多标签概率分布形式输出场景判别结果。从现有研究基础、数据结构分析、模型选择的合理性等方面综合来看，所提出的技术路径具有较强的可行性和落地性，具体分析如下：

数据维度丰富，具备高度结构化与语义化特征。用户行为数据作为场景识别的核心输入，其质量、粒度和表达能力直接决定了模型判

别能力的上限。从字段设计上看,数据涵盖了时间戳、应用使用信息、用户状态、位置信息等多个维度。这些字段分别对应时间、语义、上下文、空间等要素,构成了一个典型的多模态特征体系,能够完整描述用户在某一时刻的行为上下文。

Transformer 架构适应多模态序列建模趋势,具备先进性。在模型选择方面,本项目采用基于 Transformer 的架构进行场景识别建模,相较于传统的 RNN、CNN 结构,该模型在处理异构特征序列方面具有显著优势。一方面,Transformer 的自注意力机制能够有效建模行为序列中不同时间点特征之间的依赖关系,捕捉用户使用行为的上下文规律;另一方面,多头注意力机制可以并行处理多种特征间的交互模式,适应多模态输入场景,提升模型泛化能力。

此外,模型输出结果还可以作为推荐系统的输入特征,进一步参与候选应用的排序阶段,有助于实现“场景-服务”之间的动态匹配。通过结合用户画像与行为序列,还可支持多粒度的动态用户建模,增强模型长期和短期偏好的协同建模能力。

综上所述,本项目在用户场景识别模块所采用的建模路径,从数据结构、模型选择、任务设计到训练策略,均符合当前推荐系统与用户建模的技术趋势。在数据可获取性强、结构清晰、语义丰富的前提下,基于 Transformer 的多标签分类模型具备良好的表现能力和迁移性。方案具有扎实的研究基础与可实现性,为下游个性化服务推荐提供了稳定且可扩展的语境识别能力。

为确保项目顺利推进并高效交付,制定如下详细项目计划安排:第 1 周为数据准备与预处理阶段,完成原始数据的收集、清洗、按 uid 合并、content 拆分及缺失率统计分析,明确最终使用的特征字段,同时完成初版预标注生成;第 2 周为特征工程与数据加载模块开发,完

成特征数值化、编码标准制定、自定义 Dataset 与 DataLoader 模块开发与验证,确保数据流通顺畅;第 3 周进入模型搭建与初步训练阶段,完成 TabTransformer 模型框架实现,确定 Embedding 维度、Transformer Block 堆叠层数、MLP 结构等关键超参数配置,并完成在小样本集上的初步训练测试;第 4 周为完整数据集训练与调优阶段,全面铺开正式训练,进行多组超参数搜索,并根据验证集效果实时调整训练策略;第 5 周为模型评估与优化阶段,深入分析训练结果,细化模型性能指标,对特征重要性进行解释性分析,必要时进行轻量化模型剪枝或蒸馏优化;第 6 周为推理优化与工程集成阶段,完成模型推理接口封装,编写推理测试脚本,进行模型性能压测,确保满足实际应用时延与资源占用要求,并预留时间处理模型部署相关适配;第 7 周为最终验收与文档整理阶段,整理完整技术文档、实验报告、模型版本记录及使用指南,同时根据验收标准完成最终演示与交付。整体安排确保各阶段衔接紧密,任务划分合理,进度可控,并留有一定的机动时间应对可能出现的问题或调整,最大限度保障项目按时保质完成。

3. 基于用户画像和用户场景的应用/服务推荐

3.1 洞察总结

随着移动互联网的深入发展与智能终端的普及,用户在日常生活所处的使用情境愈发复杂和多样化。传统的推荐系统多基于用户历史行为进行推荐,难以精准把握用户在不同时间、地点、状态下的即时需求。而本项目研究的“场景化推荐”逐渐成为推荐系统演进的重要方向之一,强调在特定的情境下,为用户提供更契合当下环境与目的的服务组合。本项目旨在面向“出行”这一垂直领域,结合用户的感知行为数据与社会热点信息,构建一个细粒度的场景识别与推荐系统,能够实现端侧的高效部署与实时响应。项目目标不仅包括高质量的场

景识别能力，还涵盖对场景与服务之间关系的理解与建模，以实现真正的“智能场景化推荐”。

为实现上述场景化推荐的目标，本项目采用了基于 Transformer 并结合对比学习的序列推荐模型。近年来，基于注意力机制的序列推荐模型在学术界取得了显著发展，逐步成为主流研究范式。传统的序列推荐方法如 Markov 模型或 RNN 受限于短期依赖建模和信息压缩问题，难以捕捉复杂的长期用户兴趣动态。注意力机制的引入有效缓解了这些限制，允许模型在不依赖固定状态转移的前提下对历史行为进行全局建模，从而提升了推荐的表达能力与个性化水平。

在学术界，代表性工作 SASRec 将 Transformer 结构引入推荐系统，利用自注意力机制学习用户行为序列中各位置之间的依赖关系，实现了动态兴趣表达与高效序列建模。该模型避免了 RNN 在长序列中存在的信息遗失问题，并通过位置编码保持序列顺序信息，显著优于传统方法。随后，BERT4Rec 进一步拓展了建模视角，采用双向注意力机制对序列进行掩码建模，通过重构被遮蔽物品的方式进行训练，不仅增强了上下文理解能力，也提升了模型的稳健性和泛化能力。与之相对，GPTRec 等模型则采用自回归式注意力建模方式，聚焦未来行为预测，适用于线上实时推荐。

随着大数据与人工智能技术的飞速发展，个性化推荐已成为电商、社交媒体和内容平台提升用户体验与商业价值的关键支撑系统。近年来，越来越多的推荐系统开始融合上下文信息、行为特征及社会因素等多维信号，以实现更加精准和动态的个性建模。特别是在推荐架构

的模型层面，Transformer 及其变体已逐步取代传统的协同过滤与浅层神经网络，成为工业界主流的技术选择。

以字节跳动为例，其基于 Transformer 架构结合多兴趣建模的推荐系统，成功应用于今日头条、抖音等产品的资讯与短视频推送，在捕捉用户多样化兴趣方面展现出强大性能。阿里巴巴提出的 DIN 模型则通过引入注意力机制，实现用户历史行为的动态加权，有效提升了淘宝广告推荐的点击率。腾讯广告则部署了 Target-aware SASRec 模型，将 Transformer 用于建模长序列用户行为，结合目标感知机制提升粗排阶段的表现，而其使用于精排阶段的 TIN 模型，进一步将时间编码与注意力机制结合，在捕捉用户行为与推荐目标的时序关系中表现优越。

综上所述，目前业界推荐模型正逐步从通用规则匹配向深度学习驱动的复杂建模过渡。在基于注意力机制的序列建模基础上，结合知识图谱以及多模态信号融合等手段，推动推荐系统实现更深层次的上下文感知与个性化推荐，代表了智能推荐技术发展的未来方向。

3.2 技术方案

本项目针对应用/服务推荐任务，提出了一种结合用户画像和用户场景的推荐模型，旨在通过有效利用用户画像和用户场景知识图谱来提升应用/服务推荐的性能。过去的序列推荐模型通常基于用户-项目交互顺序，通过捕捉用户历史行为序列的时间动态和语义特征，挖掘用户的兴趣变化并预测其未来偏好。然而，这些方法在冷启动场景或候选集规模较大时，常常面临推荐效果下降和计算效率不足的问题。

(1) 知识图谱构建

为提升推荐效果和计算效率，本框架首先引入场景知识图谱对候选集进行剪枝优化，利用知识图谱中的场景和应用/服务节点的关联关系筛选出与目标场景相关的候选集，从而有效缩小推荐范围，提升推荐的准确性和效率，如图 3.1 所示。

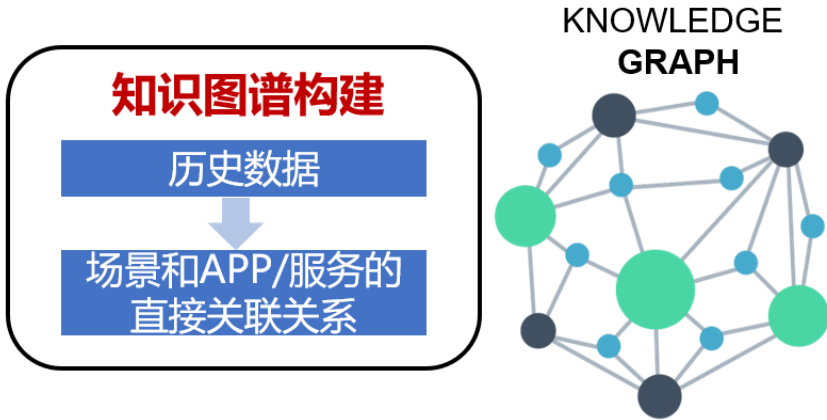


图 3.1：知识图谱构建

(2) 应用/服务推荐框架

通过设计不同的融合策略，在行为序列建模的不同阶段将用户画像和用户行为序列有机结合，支持从全局上下文到细粒度动态建模的多层次推荐需求。通过用户画像和用户场景知识图谱的引入，不仅能够根据场景信息精准限定候选应用/服务，还能够有效提升序列推荐系统在冷启动场景或数据稀疏情况下的推荐准确性，进一步增强推荐系统的适应性和鲁棒性。具体地，如图 3.2 所示，用户画像驱动序列推荐融合框架包括 4 个主要模块：1) **前缀融合**、2) **逐项融合**、3) **序列建模**、4) **后端融合**。其中前缀融合模块通过提示生成器将用户画像信息转换为个性化提示向量，并将这些提示作为行为序列的上下文前缀嵌入推荐模型，从而优化行为建模的表现。

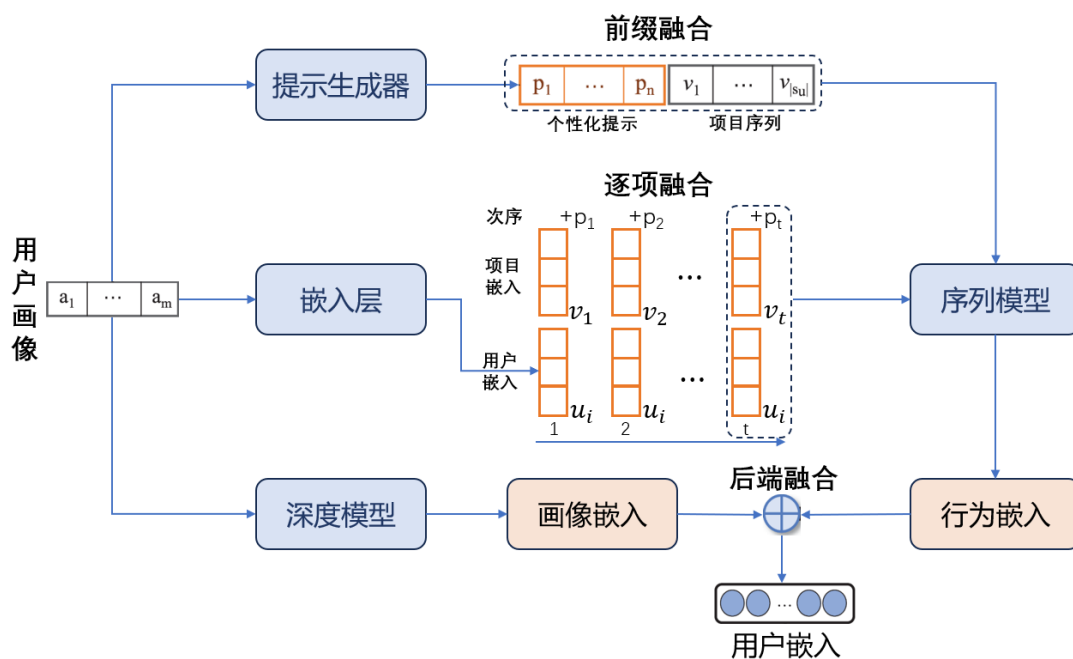


图 3.2：用户画像驱动序列推荐融合框架

(3) 数据处理与表征

数据定义：

● 用户画像数据

用户画像数据主要包括静态属性和长期行为特征，用于描述用户的固有特征和偏好。以下是属于用户画像的数据字段：

1. 设备基本信息：

- ✧ DEVICE_MODEL:设备型号
- ✧ OS_VERSION:操作系统版本
- ✧ BRAND:手机品牌
- ✧ MANUFACTURER:制造商
- ✧ CONNECT_DEVICE_LIST:已连接设备列表
- ✧ WIFI_SWITCH:WiFi 开关状态
- ✧ BT_SWITCH:蓝牙开关状态
- ✧ WIFI_CONNECT_STATUS:WiFi 连接状态
- ✧ BT_CONNECT_STATUS:蓝牙连接状态

✧ POWER_STATUS:电源状态（如充电中、电量低）

2. 长期使用统计：

✧ APP_USAGE_STATISTIC_7:过去 7 天的应用使用统计

✧ APP_USAGE_STATISTIC_7_LENGTH:过去 7 天的应用使用统计长度

✧ INTENTS_USAGE_STATISTIC_7:过去 7 天的意图使用统计

✧ INTENTS_USAGE_STATISTIC_7_LENGTH:过去 7 天的意图使用统计长度

✧ DOMAINS_USAGE_STATISTIC_7:过去 7 天的域使用统计

✧ SUBDOMAINS_USAGE_STATISTIC_7:过去 7 天的子域使用统计

✧ APP_USAGE_STATISTIC_30:过去 30 天的应用使用统计

✧ APP_USAGE_STATISTIC_30_LENGTH: 过去 30 天的应用使用统计长度

✧ INTENTS_USAGE_STATISTIC_30:过去 30 天的意图使用统计

✧ INTENTS_USAGE_STATISTIC_30_LENGTH:过去 30 天的意图使用统计长度

✧ DOMAINS_USAGE_STATISTIC_30:过去 30 天的域使用统计

✧ SUBDOMAINS_USAGE_STATISTIC_30:过去 30 天的子域使用统计

✧ LONG_TERM_SERVICE_STATISTIC:长期服务统计数据

✧ SERVICE_DOMAIN:服务所属域

✧ SERVICE_SUBDOMAIN:服务所属子域

3. 其他画像属性:

✧ SERVICE_ID:服务 ID

✧ MOTION_STATUS:运动状态 (如静止、步行)

✧ SCENES:场景(如学校、地铁)

✧ MODULE_NAME:模块名称

✧ ABILITY_NAME:能力名称

✧ SERVICE_LEVEL:服务级别

✧ INTENT:意图

✧ HOUR:当前小时

✧ DAY:当前日期

● 交互序列数据

交互序列数据主要涉及用户在不同时间段内的行为序列,反映用户的动态偏好和实时互动情况。以下是属于交互序列的数据字段:

1. 短期使用统计 (最近 0-5 分钟):

✧ UNION_USING_APPS_0_5:最近 0-5 分钟使用的应用

✧ UNION_USING_APPS_0_5_LENGTH:最近 0-5 分钟使用的
应用数量

✧ UNION_USING_INTENTS_0_5:最近 0-5 分钟使用的意图

✧ UNION_USING_INTENTS_0_5_LENGTH:最近 0-5 分钟使
用的意图数量

✧ UNION_USING_DOMAINS_0_5:最近 0-5 分钟使用的域

✧ UNION_USING_SUBDOMAINS_0_5_:最近 0-5 分钟使用
的子域

2. 中期使用统计 (最近 5-10 分钟):

- ✧ UNION_USING_APPS_5_10:最近 0-5 分钟使用的应用
- ✧ UNION_USING_APPS_5_10_LENGTH:最近 0-5 分钟使用的应用数量
- ✧ UNION_USING_INTENTS_5_10:最近 0-5 分钟使用的意图
- ✧ UNION_USING_INTENTS_5_10_LENGTH:最近 0-5 分钟使用的意图数量
- ✧ UNION_USING_DOMAINS_5_10:最近 0-5 分钟使用的域
- ✧ UNION_USING_SUBDOMAINS_5_10_:最近 0-5 分钟使用的子域

● 当前应用数据:

- ✧ CANDIDATE_APP:当前点击的应用
- ✧ LABEL: 用户点击应用的标签, 值为 0 或者 1

数据处理:

● 共享嵌入层

在推荐系统中, 为了高效地处理大量离散特征 (如服务 ID、意图 ID 等), 通常会使用嵌入层将这些高维稀疏特征映射到低维稠密向量空间。这样做的目的是捕捉特征之间的语义关联, 并减少模型的参数数量。共享嵌入层是指针对某些具有相似语义或关联紧密的特征, 使用同一个嵌入矩阵来生成它们的向量表示。例如, 服务 ID 和意图 ID 可能来源于同一业务逻辑或数据源, 因此它们的嵌入可以共享同一个权重矩阵, 从而提高参数的利用效率, 避免过拟合, 并加快训练速度。共享嵌入层的设计基于以下几个考量:

语义相关性: 当不同特征之间存在内在联系时, 共享嵌入层可以捕捉这种关系。例如, 服务 ID 和意图 ID 可能共同决定了用户的行为偏好, 共享嵌入层能够在这种情况下更好地建模它们的交互效应。

资源节约：多个特征共用同一个嵌入矩阵减少了模型的参数总量，特别是在特征数量庞大时，这一点尤为重要。这不仅降低了存储开销，还提高了训练和推理的效率。

泛化能力提升：通过共享学习到的嵌入矩阵，模型能够在不同特征间传递知识，增强对未见过数据的泛化能力。例如，某个服务 ID 在训练数据中出现较少，但通过共享嵌入层，它可以从其他高频服务 ID 中学到有用的特征表示。的领域偏好，并在推荐时优先考虑相关领域的服务，从而提高推荐的精准度和用户体验。

● 用户、服务、意图向量构建

特征嵌入与拼接：用户、服务、意图特征通常包含多种类型的数据，例如用户的历史行为、偏好、基本信息、服务 ID、用户在过去一段时间内的搜索关键词、点击行为或偏好标签等。这些特征往往是稀疏的、高维度的，直接使用这些原始数据会导致模型难以有效学习。为此，我们首先将这些特征通过嵌入层转换为低维度的稠密向量（嵌入向量）。嵌入层的作用是将离散的特征值映射到连续的向量空间，使得相近的特征值在向量空间中具有相似

密集层与正则化处理：拼接后的服务向量维度较高，不利于高效训练。通过密集层（全连接层）对其进行降维处理，提取关键特征并压缩到较低维度。密集层通过学习权重矩阵，将高维度输入转换为低维度输出，同时保留最重要的信息。为了防止过拟合，我们应用 L2 正则化，限制权重的大小。最后，通过批量归一化来标准化数据分布，确保每一层的输入具有稳定的统计特性，加速训练收敛并提高模型的稳定性。最终得到的服务嵌入向量能够在保持语义信息的同时，具备良好的泛化能力和计算效率，支持与用户嵌入向量的交互计算，如点积运算，用于推荐系统的召回或排序任务。

● 特征表示

1. 时间特征

在推荐系统和分类任务中，时间特征（如小时、天）是重要的上下文信息，能够反映用户行为的周期性和时效性。例如，用户在特定时间段内的活跃度或偏好可能会有所不同。为了有效地利用这些时间特征，模型需要将其转换为机器学习算法可处理的形式，并结合其他特征进行联合建模。

时间特征通常为离散的类别型变量，例如小时（0 到 23）和天（周几）。直接使用数值形式可能导致模型错误地推断出顺序关系（如小时 23 比小时 1 大），而实际上这些特征是分类性质的。因此，采用 one-hot 编码将每个时间点转换为独立的二进制向量，消除了数值大小带来的误导，使模型能够独立学习每个时间点的权重。

在某些应用场景中，时间特征的处理不仅仅是为了捕捉其本身的特征分布，还需要与其他特征进行交互，例如过滤或加权。例如，在处理用户的小时级别活动数据时，生成的时间掩码可以用来指示特定时间段内的行为是否活跃，或者在注意力机制中作为权重矩阵的一部分，强调某一时间段内的特征重要性。这有助于模型在不同的时间窗口内动态调整关注点，提升预测的准确性。

通过 one-hot 编码和掩码生成，时间特征被转换为结构化的输入，使得模型能够在训练过程中捕捉到时间相关的模式和规律，从而更好地应用用户行为的周期性变化，提高模型的时效性和准确性。

2. One-Hot 类别特征

在推荐系统中，One-Hot 类别特征是指那些具有离散、互斥取值的特征，例如用户的设备型号（`DEVICE_MODEL`）、操作系统版本（`OS_VERSION`）、品牌（`BRAND`）、设备状态（`MOTION_SYATUS`）

等。每个特征只有一个具体的取值，比如设备型号可能是“MATE 40”或“P60”，操作系统版本可能是“HarmonyOS 4”或“HarmonyOS 5”。这些特征的原始形式通常是字符串或分类变量，但机器学习模型无法直接处理这种形式的数据，因此需要进行编码和转换。

对于每个 One-Hot 类别特征（如 DEVICE_MODEL），首先通过 Input 层定义一个形状为 (1,) 的输入层，对应单个特征标量输入值。为了将离散的类别特征转换成模型可以处理的数值形式，我们使用嵌入层将每个类别映射到一个固定维度的稠密向量空间。例如，如果 DEVICE_MODEL 可能有 100 种型号，嵌入层会将每个类别编码成一个 8 维的向量 (EMBEDDING_LENGTH=8)。这个过程不仅减少了特征的维度，还能够在一定程度上捕捉类别之间的语义关系，比如相似的设备型号可能会在嵌入空间中靠近。

另外，为了避免模型对某些特征（如用户唯一标识 UDID）的过度依赖导致过拟合，我们在嵌入层之后应用 Dropout 技术。Dropout 会在训练阶段随机丢弃一部分嵌入向量的维度，强制模型学习到更多依赖于其他特征的泛化能力。例如，当处理 UDID 特征时，即使某个用户的标识被丢弃，模型仍然需要依靠设备型号、使用习惯等其他特征来进行预测，从而提高了模型的鲁棒性和泛化能力。

3. Multi-Hot 类别特征

在推荐系统中，Multi-Hot 类别特征是指那些具有多个可能取值的特征，每个取值代表一类标签，并且一个样本可以同时属于多个类别。例如，用户的兴趣标签可能包括“科技”、“娱乐”、“体育”等多个类别，或者用户当前使用的应用程序列表（如 UNION_USING_APPS_0_5、UNION_USING_APPS_0_10）可能包含多个应用。这些特征的原始形式通常是集合或多值列表，需要转换为

模型可处理的格式。

为了将这些多值类别特征转换为稠密向量，我们首先将每个类别标签通过嵌入层映射到一个固定维度的向量空间。每个用户的兴趣标签列表中的每个标签都会被转换成对应的嵌入向量，形成一个二维矩阵（标签数 \times 嵌入维度）。例如，对于特征列 `UNION_USING_0_5`（用户在过去 0-5 分钟内使用的应用程序列表），每个应用程序都被映射到一个嵌入向量，生成一个矩阵。

由于一个样本可能有多个标签，我们需要将这些嵌入向量聚合为一个单一的向量来表示整个特征。常用的聚合方法之一是求均值，即对每个样本的所有嵌入向量在标签维度上取平均值。这样做是为了将多值特征压缩成一个固定长度的向量，便于后续的特征交互和模型训练。例如，用户的意图使用统计（如 `INTENTS_USAGE_STATISTIC_7`）转换为多个嵌入向量后，通过求平均得到一个综合的向量，反映了该用户的整体兴趣偏好。这种方法既保留了所有标签的信息，又避免了维度爆炸的问题。

对于某些噪声较大的特征（如 `WIFI_SWITCH` 或 `BT_SWITCH`），在池化后的向量上应用 Dropout，减少过拟合风险。

4. 标签特征

在推荐系统中，多层次标签特征（如 `PRE__TWO__LEVEL__TAGS`、`PRE__THREE__LEVEL__TAGS` 和 `PRE__INTENTS`）用于刻画服务调用行为的层级化语义信息。这些标签通过服务元数据与服务调用行为关联，具体包括：

1) 服务领域标签：如 `SERVICE__DOMAIN` 表示服务的主领域分类（如“智能家居控制”、“媒体播放”），对应代码中的 `SECOND__LEVEL__TAG` 字典

2) 服务子领域标签: 如 SERVICE__SUBDOMAIN 细化到具体功能维度(如“灯光控制”、“视频流媒体”), 对应 THIRD__LEVEL__TAG 字典

3) 意图语义标签: 如 INTENT 描述服务调用的具体意图(如 "adjust__brightness" 、 "play__video__stream"), 对应 INTENTS 字典
针对这些层级化标签特征的处理流程如下:

(a) 分层字典映射:

对 PRE__TWO__LEVEL__TAGS 特征使用 SECOND__LEVEL__TAG 字典进行编码, 字典覆盖所有服务主领域类型

对 PRE__THREE__LEVEL__TAGS 特征加载 THIRD__LEVEL__TAG 字典, 包含细粒度子领域分类

PRE__INTENTS 特征关联 INTENTS 字典, 映射意图字符串到唯一索引

(b) 动态嵌入构建:

每个标签类型创建独立嵌入层, 维度为 LONG__EMBEDDING__LENGTH

示例: PRE__TWO__LEVEL__TAGS 的嵌入矩阵大小为 $len(SECOND_LEVEL_TAG) + 2$, 保留 0 值作为 padding mask

(c) 多标签聚合:

对服务调用记录中的多个标签执行均值池化 (tf. reduce__mean (axis=1))

例如: 某个服务调用涉及 ["media" , "streaming"] 两个子领域标签时, 其嵌入向量在序列维度取平均生成综合表征。

每个标签特征都有一个预定义的词汇表, 例如, 二级标签的词汇表包含了所有可能的二级类别。每个类别在词汇表中有唯一的索引。例如, “新闻”可能对应索引 1, “视频”对应索引 2, 依此类推。通过这

种方式，标签特征被转换为数值化的索引，便于嵌入层处理。

每个标签类别通过嵌入层映射到一个固定维度的稠密向量。例如，二级标签嵌入层将每个索引转换为 8 维向量。如果一个用户的历史行为包含了多个标签（如“新闻”、“娱乐”），这些标签对应的嵌入向量会被聚合成一个单一向量。计算这些嵌入

向量的均值，得到一个综合的向量表示。这个聚合向量反映了用户在该层级标签上的整体特征，有助于模型理解用户的多维兴趣偏好。

通过这种方式，标签特征被有效地转化为模型可处理的形式，同时保留了多标签信息的综合效应，增强了推荐系统的个性化能力和准确性。

5. 数值特征

数值特征作为高维向量输入到神经网络的第一层——全连接层。全连接层通过权重矩阵（维度为 $n \times 8$ ，其中 n 为输入特征数）和偏置向量，执行线性变换。例如，对于设备型号 `DEVICE_MODEL`、操作系统版本 `OS_VERSION` 等数值特征，每个特征通过权重进行加权求和，并叠加偏置，生成 8 维中间向量。该过程捕捉输入特征间的线性组合关系，例如，`DEVICE_MODEL` 与 `OS_VERSION` 对用户偏好可能存在的联合影响权重。

线性变换后的结果通过 ReLU 激活函数，引入非线性特性。ReLU 将负值设为 0，正值保持不变，使输出向量具有稠密性（大部分维度非零），增强模型表达复杂模式的能力。例如，某个输出维度可能表示年龄和收入的正相关组合，而另一维度则反映教育水平的独立贡献。

最终得到的 8 维稠密向量作为高层次特征表示，可用于下游任务（如分类、回归）。这些向量融合了原始特征的线性和非线性交互信息，提高了模型的预测精度和泛化能力。输出维度的选择（如 8 维）

通常基于实验调优或业务需求，旨在平衡信息保留与计算效率。

6. Top APP Usage 特征

Top App Usage 特征(如 UNION__USING__APPS__0__5、UNION__USING__APPS__5__10 等)代表用户最常使用的应用程序，可能是稀疏的类别型数据(例如，应用 ID 列表)。为了与其他特征(如数值特征)在同一模型中进行联合处理，需要将这类特征转换为固定维度的稠密向量。

1) 特征编码与维度扩展

每个 Top App Usage 特征首先被编码为独热向量或嵌入向量。假设其他特征处理后的维度为 D，这些特征通过嵌入层(Dense 层)扩展至相同维度 D。例如，使用 Dense(8) 层将每个特征的输入维度 M(由 max__len 决定)转换为 8 维稠密向量，并通过 ReLU 激活函数引入非线性。

2) 捕获潜在关联

处理后的稠密向量不仅扩展了特征维度，还捕捉了应用间的潜在关联。例如，不同应用的使用频率可能共同反映用户的消费习惯或社交活跃度。通过嵌入层的学习，这些关联被结构化地融入模型，提升预测精度。

3) 应用场景示例

短期行为分析: UNION__USING__APPS__0__5(0-5 分钟内的应用使用记录)可用于捕捉用户的即时需求，如导航应用的高频使用可能指示出行场景。

中期偏好追踪: UNION__USING__APPS__5__10(5-10 分钟内的使用记录)反映用户的阶段性偏好，如办公软件的持续使用可能指示工作状态。

长期画像构建：APP_USAGE_STATISTIC_30（过去 30 天的应用统计）用于构建稳定的用户画像，如娱乐类应用的高频使用指示休闲偏好。

4) 模型融合策略

所有处理后的 Top App Usage 特征被扩展为 8 维向量并通过 `expand_dims` 操作统一维度，便于与设备属性、时间特征等其他特征融合，共同输入到下游模型（如 CNN 或 Transformer），实现多维度用户行为的综合分析。

在推荐系统或用户画像建模中，Top App Usage 特征的维度扩展确保了模型能够综合考虑用户的多方面偏好。例如，一个用户频繁使用的社交媒体和购物应用可能被映射到同一维度的不同权重，反映出其消费习惯和社交活跃度。通过全连接层的处理，这些信息得以结构化地融入模型，提高推荐的精准度和个性化程度。

（4）前缀融合

基于上述计算得到的用户画像向量 x_u ，我们将其送入提示生成器中，以生成用户相关的提示信息。提示生成器可使用多层感知机 (MLP) 或大语言模型 (LLM) 生成嵌入，其作用是将用户画像信息转换为一组连续的提示向量 $P_u = \{p_1, p_2, \dots, p_n\}$ ，作为用户行为建模的全局上下文输入。

MLP 具体的生成过程如下：

$$P_u = W_2 \cdot \sigma(W_1 \cdot x_u + b_1) + b_2$$

其中， W_1 和 W_2 为可学习的权重矩阵， b_1 和 b_2 为偏置项， σ 为激活函数。生成的提示向量 P_u 被插入到用户的行为序列 $S_u = \{v_1, v_2, \dots, v_{|S_u|}\}$ 的最前端，形成扩展后的输入序列：

$$\hat{S}_u^{prefix} = \{p_1, p_2, \dots, p_n, v_1, v_2, \dots, v_{(|S_u|)}\}$$

LLM 具体的生成过程如下：

将用户画像转换为自然语言描述的形式作为 LLM 的提示词，输入到大模型中。例如：

"用户使用的是 HUAWE 品牌的手机, 操作系统是 HamonyOS_3.0。当前时间是 2025 年 4 月 28 日下午 3 点, 用户位于上海市杨浦区..."

提取其生成的隐藏层并经过平均池化操作得到固定维度的用户画像表示 P_u , 插入到用户行为前端 S_u 获得增强的序列表示 \hat{S}_u^{prefix} 。

通过这种方式, 提示向量在行为序列建模中起到了引导作用, 使得模型在捕捉用户动态行为的同时, 也能够综合考虑用户画像的全局信息, 为推荐系统提供了更丰富的输入表达。这种前缀融合机制能够在冷启动场景下表现出更强的泛化能力, 显著提升推荐性能。

(5) 逐项融合

逐项融合模块通过在行为序列的每个时间步, 直接结合用户画像嵌入与物品嵌入, 完成对用户静态属性与行为动态特征的细粒度融合。对于用户 u 的画像嵌入 x_u , 我们在行为序列中的每个位置 t 将其与对应的物品嵌入 v_t 拼接, 生成时间步 t 的融合嵌入表示:

$$e_t = [v_t || x_u] + p_t$$

其中, v_t 是序列中第 t 个物品的嵌入表示, x_u 为用户画像嵌入表示, p_t 为可学习的位置编码, 用于表征物品在序列中的顺序关系。通过将用户画像信息嵌入到每个时间步的特征表示中, 逐项融合模块在序列建模的输入阶段即实现了对静态属性和动态行为的统一建模。最终, 融合后的行为序列嵌入表示为:

$$\hat{S}_u^{item} = \{e_1, e_2, \dots, e_{|S_u|}\}$$

逐项融合强调在行为序列的每一个时间步都显式地引入用户画像信息, 这种机制在增强序列中每个物品表示的个性化程度的同时,

还能后续的序列建模模块提供更丰富的输入信号，捕捉用户画像与行为动态之间的复杂交互关系。

(6) 序列建模

序列建模是用户画像驱动序列推荐融合框架的核心模块，负责捕捉用户行为序列的时间动态和语义特征，并结合用户画像信息和数据增强技术生成用户的行为偏好表示。该模块以扩展后的行为序列为输入，通过深度序列建模方法提取序列中的隐含模式，为推荐任务提供高质量的特征表示。本项目中，序列建模模块采用基于 Transformer 的序列模型，该模型以自注意力机制为基础，结合数据增强模块，采用对比学习框架，能够高效捕捉序列中长距离和非连续的依赖关系，从而对用户的兴趣变化进行全面建模。

具体而言，序列建模模块以融合后的行为序列嵌入 \hat{S}_u 作为输入，其中 \hat{S}_u 为前缀融合生成的 \hat{S}_n^{prefix} 或逐项融合生成的 \hat{S}_n^{item} 。由行为序列嵌入，可以得到第 l 层的行为矩阵为：

$$H_u^l = \{h_{u,1}^l, h_{u,2}^l, \dots, h_{u,|S_u|}^l\}$$

其中 $h_{u,i}^l$ 表示用户 u 在 l 层的第 i 个行为特征。那么 $l+1$ 层的行为矩阵 H_u^{l+1} 可以如下学习得到：

$$H_u^{l+1} = \text{Transformer}^l(H_u^l)$$

每一层通过自注意力机制和前馈神经网络对序列中的时间步表示进行更新，捕捉序列中不同时间步之间的依赖关系。经过 2 层 Transformer 的堆叠后，生成序列的最终表示矩阵 H_u^N ，其中每一行表示序列中对应时间步的嵌入。用户的行为偏好表示 u_s 是基于序列表示矩阵 H_u^N 的最后一个时间步嵌入 $H_{u,|S_u|}^N$ 生成的，该向量可以总结整个序列的上下文信息，作为用户的行为表示：

$$u_s = h_{u,|S_u|}^N$$

(7) 后端融合

后端融合模块是用户画像驱动序列推荐融合框架的关键环节，负责将用户画像生成的属性偏好 u_a 和用户行为建模生成的动态偏好 u_s 结合起来，以生成完整的用户嵌入表示 u_p 。 u_a 是通过用户画像嵌入 x_u 输入多层感知机得来，表示用户的静态偏好特征；而 u_s 是通过序列建模模块对用户行为序列嵌入 \hat{S}_u 进行处理后得到的，表示用户的动态行为偏好。后端融合设计多种结合策略，全面整合用户的静态和动态特征，构建精准的用户表示。具体而言，包含向量拼接和逐元素相乘和加权求和的操作：

$$u_p = [u_a | u_s], \quad u_p = u_a \odot u_s, \quad u_p = \alpha u_a + (1 - \alpha) u_s$$

其中， \odot 表示逐元素相乘， α 为超参数，用于平衡用户画像和行为序列的贡献。通过这些方法，后端融合模块能够在综合用户画像的长期偏好和行为序列的短期动态偏好的同时，增强用户表示的表达能力。

(8) 训练阶段

在训练过程中，模型对行为序列 \hat{S}_u 应用三种数据增强策略，裁剪(Crop)：随机选择序列的连续部分；掩码(Mask)：随机将某些项替换为特殊标记；重排(Reorder)：随机打乱序列中某一段的顺序。通过这些增强方法，模型随机选择增强策略生成两个不同视角的序列表示，并将它们视为正样本对，同时把批次中其他序列的表示作为负样本，利用 InfoNCE 损失函数最大化正样本之间的相似度，最小化与负样本的相似度。形式化表示为：

$$L_{CL} = -\log \frac{\exp(\frac{\text{sim}(z_u^{aug1}, z_v^{aug2})}{\tau})}{\sum_{v \in B} \exp(\frac{\text{sim}(z_u^{aug1}, z_u^{aug2})}{\tau})}$$

其中 z_u^{aug1} 和 z_v^{aug2} 是序列 S_u 两个增强视图的表示, sim 是相似度函数, τ 是温度参数, B 是批次中的样本集合。最终, 通过联合优化序列推荐任务损失和对比学习损失:

$$L_{rec} = - \sum_k y_k \log(\widehat{y}_k) + (1 - y_k) \log(1 - \widehat{y}_k)$$

$$\widehat{y}_k = \sigma(c_k \cdot u_p)$$

$$L = L_{Rec} + \lambda L_{CL}$$

其中, y_k 是样本 k 的真实标签, 通常为 0 或 1, 表示用户是否点击。 \widehat{y}_k 是模型预测的兴趣值, u_p 为用户嵌入, c_k 为候选项, σ 为 sigmoid 函数。结合传统序列推荐损失 L_{Rec} 和对比学习损失, 模型能够学习到更具判别性和更稳健的序列表示。

(9) 评估阶段

在评估过程中, 首先对每个候选项 c_k 和用户嵌入 u_p , 计算它们之间的内积:

$$\text{score}(c_k, u_p) = c_k \cdot u_p$$

其中, c_k 为候选项的向量表示, u_p 为用户的嵌入向量。内积操作衡量了候选项和用户偏好的匹配程度。

接着, 将内积结果输入到 Sigmoid 函数中, 得到每个候选项的点击概率:

$$p(c_k | u) = \frac{1}{1 + \exp(-\text{score}(c_k, u_p))}$$

这个概率值表示候选项 c_k 被用户点击的可能性。Sigmoid 函数的输出范围在 $[0,1]$ 之间，能够转换为概率形式。根据计算出的点击概率 $p(c_k | u_p)$ ，对所有候选项进行排序，选择点击概率最高的前 4 个候选项，在此基础上，将选择的 Top-4 候选项与真实的用户点击数据进行比较。若真实点击项在 Top-4 候选项中，则认为推荐系统预测正确；否则，预测错误。通过对所有样本执行此操作，可以得到每个样本的评估结果，从而对所有样本计算 Top-4 候选项的准确率。

拟提出的基于用户画像和用户场景的应用/服务推荐模型，充分融合了用户静态属性与场景知识图谱，通过多策略的协同建模显著提升推荐性能与系统鲁棒性。在整个框架中，用户画像信息与行为序列信息在前缀融合和逐项融合模块中以不同方式深入结合，使得模型能够同时捕捉用户的长期偏好和短期行为特征。场景知识图谱用于候选集剪枝，通过挖掘场景与服务之间的结构化关联关系，有效压缩推荐范围，降低计算复杂度，同时提升推荐相关性。此外，在序列建模阶段，模型引入了对比学习框架与多种数据增强策略，包括裁剪、掩码和重排等，生成多个不同视角的序列表示，并通过 InfoNCE 损失函数进行训练。这些增强方法提升了模型对用户行为序列中噪声与变化的鲁棒性，使得序列表示更具判别性和泛化能力，特别适用于冷启动和数据稀疏场景。在后端融合阶段，模型进一步整合由用户画像生成的静态偏好向量与由序列建模模块获得的动态行为偏好表示，全面提升用户表示的表达能力。综合而言，本模型通过用户画像的细粒度融合与场景知识图谱的高效剪枝，显著提升了应用/服务推荐的推荐精度和计算效率。模型的多层次设计支持从全局上下文到动态行为的全面建模，展现出良好的适应性和鲁棒性，为应用/服务推荐任务提供了强

有力的技术支持。

3.3 模型评估与结果分析

为了确保项目技术路径的合理性与先进性，我们引入了主流推荐模型（双塔模型及 deepFM & DIN 模型）的对比分析。以下是对各模型的详细对比分析及本模型可行性的深入论证：

1. 双塔模型

双塔模型广泛应用于召回阶段，通过分别编码用户和物品特征，利用余弦相似度高效筛候选集。然而，该模型的核心局限在于静态特征表示，难以捕捉用户动态偏好与时序行变化。在出行场景中，用户兴趣随时间和地点快速变化，双塔模型的静态编码机制导致荐结果滞后，难以满足即时性需求。

2. deepFM & DIN 模型

DeepFM 模型通过结合因子分解机与深度神经网络有效学习特征间的低阶和高阶交互，但在捕捉用户行为的动态时序性、序列依赖以及复杂时空情境（如出行场景常见的通勤规律、即时需求变化）方面能力有限，难以充分适应出行推荐的动态特性。而 DIN 模型虽引入注意力机制来增强历史行为与目标物品的相关性，提升了个性化精度，但其注意力机制仍侧重于局部交互，对于用户行为序列中更宏观的全局模式、长距离依赖以及复杂的时空演变（如早晚高峰、周末休闲的周期性行为）的建模能力不足，导致在深度理解用户长期习惯与短期意图相结合的复杂出行情境下，推荐效果可能受限。两者在充分挖掘和利用出行场景下用户行为序列的深层语义和动态演化方面均存在一定的局限性。

3. 基于用户画像和用户场景的应用/服务推荐模型

本模型依托 Transformer 架构，通过自注意力机制深刻挖掘行为序列的时间依赖与上下文关联，尤其擅长处理稀疏高维数据。结合项目特有的数据增强与对比学习策略，本模型在以下几个方面展现出优势：

(1) 长期依赖建模：Transformer 结构有效捕捉用户历史行为中的长期模式，如节假日出行习惯、工作日通勤规律等，提供更加精准的场景感知。

(2) 数据增强提升鲁棒性：通过对行为序列进行扰动、掩码与重排等处理，本模型增强对异常行为（如偶发性偏离常规路线）的适应能力，提高模型泛化性能。

(3) 对比学习优化特征空间：构造正负样本对，引导模型聚焦判别性特征，改善推荐列表的多样性与相关性，降低过拟合风险。

本模型在出行推荐场景中的可行性体现在其能够深刻理解用户静态偏好与动态行为，并结合上下文情境进行精准预测：

(1) 全面的用户偏好建模：通过前缀融合、逐项融合与后端融合的多阶段策略，模型能够从不同粒度和层面整合用户画像信息与行为序列，形成对用户长期偏好和短期意图的完整认知。

(2) 精准的动态行为预测：基于 Transformer 的序列建模模块，结合数据增强和对比学习，能够准确捕捉复杂出行场景下用户行为的时间动态和深层语义，有效预测用户未来的出行需求。

(3) 强大的鲁棒性与泛化能力：数据增强和对比学习的应用，使得模型在面对数据稀疏、用户行为多变或存在异常（如偶发性偏离常规路线）时，仍能保持较好的推荐性能，并能有效应对冷启动问题。

(4) 细粒度个性化与场景感知：逐项融合机制确保了用户画像对序列中每个行为的影响,结合场景知识图谱(如提及的候选集剪枝),能够提供高度个性化且符合当前场景的推荐。

评估结果具体数值如下表所示：

模型	exist_click@1	exist_click@2	exist_click@3	exist_click@4	AUC
two_towers	0.53	0.73	0.78	0.81	0.88
DIN&deepFM	0.37	0.58	0.68	0.74	0.92

对于此次评估结果我们对交互序列进行了分析,交互序列详情如下：

不存在交互序列的占比	0.19067423240353987
存在交互序列的占比	0.8093257675964601
平均交互序列长度	2.7226045658023637

经分析发现,在当前数据集中,有 19%的样本不存在用户行为序列,且用户过去 10min 的交互序列的平均长度仅为 2.7。远低于本模型正常运作所需的阈值。在实际训练过程中,本模型的性能受到用户行为序列长度的显著影响。由于该模型依赖自注意力机制建模用户的动态偏好,其效果在序列较短或缺失严重的样本中显著下降。这类样本在训练集中缺乏足够的历史上下文信息,导致模型难以捕捉用户的偏好迁移模式,进而影响其对下一个交互的预测能力。此外,序列缺失也降低了训练样本的有效性,使得模型难以学习稳定的序列表示,从而整体性能下滑。

下一步计划在数据集收集中,将用户行为序列的统计时间增加至 1 天或更长,从而扩充交互序列长度时模型能更好地捕捉用户的偏好迁移模式,从而实现更加有效的推荐。

3.4 可行性分析及项目周期安排

本项目在技术路径上围绕用户画像与用户场景的联合建模展开,

通过数据增强、特征融合和序列建模等多层设计，有效提升了推荐系统在复杂出行情境下的感知能力与推荐性能。在数据预处理环节，方案充分考虑移动环境中数据的不完整性与异构性问题，通过清洗低活跃行为、归一化数值特征、POI 语义增强等策略，缓解了位置信息缺失对场景识别准确性的影响。特别是在空间信息稀疏的条件下，通过将坐标转化为带有地理语义的标注，实现了对上下文环境的有效补全。

为增强模型的泛化能力与鲁棒性，项目引入对比学习机制，对行为序列进行扰动、掩码与重排等处理，生成语义相近但形式不同的多视角表示，从而提高模型对用户兴趣变化与行为偏差的适应能力。在特征表示方面，设计了多类型融合机制，不同类型特征根据其属性采用差异化编码方式，最终统一映射到共享语义空间进行融合。多源信息的集成不仅为模型提供了更全面的行为背景，也提升了对动态场景的感知粒度。

在行为建模过程中，模型通过前缀提示或逐项嵌入两种策略，将静态用户画像信息灵活引入序列模型，从全局上下文到局部行为细节实现多尺度融合。Transformer 结构在建模用户行为序列时捕捉到了长期依赖与序列模式变化，结合引导式提示和局部交互增强策略，使得模型不仅能理解历史行为，还能识别其背后的行为动因。在训练阶段，加入对比学习损失函数，通过构造正负样本对，引导模型聚焦于序列间的判别性特征，进一步提升了建模的稳定性与表达能力。

整体来看，该方案在场景感知、用户建模与推荐预测三方面形成了有机协同，通过知识图谱剪枝机制有效控制候选空间、提升响应效率；通过多模态特征融合与行为序列建模准确刻画用户偏好与意图；通过数据增强与对比学习机制增强模型鲁棒性与冷启动表现。系统具备良

好的可扩展性与端侧部署潜力，可适配不同数据规模与业务场景，为个性化服务推荐提供了稳健且具落地价值的解决方案。

为确保项目顺利推进并高效交付，制定如下详细项目计划安排：第 1 周为项目准备与数据处理阶段，熟悉整体技术方案，完成用户画像数据和交互序列数据的收集、清洗与格式标准化，搭建基础工程框架，包括数据解析、特征提取脚本及共享嵌入层骨架，确保所有原始数据可以按需转换为模型可用输入格式；第 2 周为知识图谱构建与候选集剪枝阶段，完成场景知识图谱初版搭建，梳理场景与应用/服务的关联关系，开发基于知识图谱的候选集筛选模块并进行剪枝效果测试，明确剪枝策略及效果评估标准；第 3 周进入特征表征与用户服务向量构建阶段，完成用户画像、服务、意图、时间、标签等各类特征的编码、嵌入及向量化处理，实现 One-hot、Multi-hot 及数值特征映射标准，同时完成 Top App Usage 特征扩展与归一化，确保特征表征统一且稳定；第 4 周为融合模块开发阶段，完成前缀融合模块的 MLP 提示生成器实现与接口预留，完成逐项融合模块中用户画像与物品嵌入的细粒度拼接处理，确保融合后的行为序列顺利对接后续建模流程，并完成初步联调验证；第 5 周进入序列建模与后端融合阶段，基于 Transformer 架构完成序列建模模块开发，引入数据增强策略包括裁剪、掩码与重排，同时搭建对比学习框架并实现后端融合模块，综合用户静态画像与动态行为生成最终用户表示，确保模块接口完整、特征流通顺畅；第 6 周为模型训练、评估与调优阶段，全面搭建训练流程与评估流程，执行小规模数据集的初步训练与验证，联合优化推荐任务损失与对比学习损失，进行 Top-4 准确率评估，整理实验结果并基于验证集效果进行超参数微调与训练策略优化；第 7 周为最终验收与文档整理阶段，完成整体项目的代码梳理与模块注释，编写开发文

档、使用手册及模型结构流程图，整理各阶段实验与性能对比报告，确保文档齐全、内容规范，同时根据验收标准完成最终功能测试、演示与交付。整体安排保证各阶段任务分解细致、衔接合理，进度可控，并留有适当机动时间，以应对项目实施过程中可能出现的调整或问题，确保最终按时保质完成交付。