

Sequential Recommendation with Graph Neural Networks

• ABSTRACT

- 顺序推荐旨在利用用户的历史行为来预测他们的下一次交互。现有工作尚未解决顺序推荐中的两个主要挑战。首先，用户在其丰富的历史序列中的行为往往是隐含的、嘈杂的偏好信号，它们不能充分反映用户的实际偏好。此外，用户的动态偏好往往会随着时间的推移而迅速变化，因此很难在他们的历史序列中捕捉用户模式。在这项工作中，我们提出了一个名为 SURGE (SeqUential Recommendation with Graph neural nEtworks 的缩写) 的图神经网络模型来解决这两个问题。具体来说，SURGE 通过基于度量学习将松散的项目序列重构为紧密的项目-项目兴趣图，将长期用户行为中不同类型的偏好整合到图中的集群中。通过在兴趣图中形成密集的集群，这有助于明确区分用户的核心兴趣。然后，我们在构建的图上执行集群感知和查询感知图卷积传播和图池化。它从嘈杂的用户行为序列中动态融合并提取用户当前激活的核心兴趣。我们对公共和专有工业数据集进行了广泛的实验。实验结果表明，与最先进的方法相比，我们提出的方法具有显著的性能提升。对序列长度的进一步研究证实，我们的方法可以有效地模拟长行为序列。

• INTRODUCTION

- 顺序推荐试图通过利用用户的历史行为序列来预测用户的下一个行为，这已被现代在线信息系统广泛采用，如新闻、视频、广告等。不同于传统的以静态方式对用户偏好建模的推荐任务，顺序推荐能够捕捉用户的进化和动态偏好。例如，用户可能只喜欢在世界杯期间观看足球新闻，这可以看作是一种短期偏好。
- 现有的工作通过从三个角度解决问题，已经意识到建模快速变化的短期偏好的重要性。具体来说，早期的努力采用人工设计的规则或注意力机制来为历史交互的项目分配时间衰减的权重。第二类作品利用递归神经网络来总结行为序列，但由于难以对长期依赖关系建模，它们在捕捉用户动态兴趣方面存在短期瓶颈。最近的解决方案将长期和短期利益联合建模以避免忘记长期利益，但长期/短期利益的划分和整合仍然具有挑战性。简而言之，上述工作通常更关注近期的用户行为，无法充分挖掘旧的行为序列来准确估计他们当前的兴趣。因此，顺序推荐中存在以下两个尚未得到很好解决的主要挑战。
 - 长序列中的用户行为反映了隐含的、有噪声的偏好信号。
 - 用户可能会与许多带有隐式反馈的项目进行交互，例如点击和观看。与可以推断用户偏好（例如喜欢和收藏）的显式反馈不同，单个隐式反馈不能反映用户偏好。用户大部分时间可能会点击他们不感兴趣的项目，之后不会选择相似的项目进行交互。然而，这些记录将成为用户行为历史中的噪音，使他们真实兴趣的建模恶化。
 - 由于其多样性，用户偏好总是随着时间的推移而变化。
 - 正如我们所提到的，用户偏好正在发生变化，无论是慢是快。给定一个时间点，一些偏好可能仍然被激活，而另一些可能已经被停用。因此，即使我们从隐含和嘈杂的行为中提取了用户偏好，对它们在历史中的变化进行建模并估计当前时间激活的偏好仍然具有挑战性，这是推荐模型的核心。

- 为了解决这两个挑战，我们提出了一种基于图的方法，使用图卷积网络来提取隐式偏好信号。然后使用动态图池来捕获偏好的动态。具体来说，我们首先将松散的项目序列转换为紧密的项目-项目图，并设计一个细心的图卷积网络，将弱信号收集到能够准确反映用户偏好的强信号。然后，我们提出了一种动态图池化技术，该技术自适应地保留激活的核心偏好以预测用户的下一个行为。
- 综上所述，本文的贡献如下：
 - 我们通过考虑隐含的信号行为和快速变化的偏好，从另一个角度来分析序列建议。
 - 我们建议通过在构建的项目兴趣图上设计基于图神经网络的模型，将用户行为中的隐式信号聚合为显式信号。然后，我们设计了动态池来过滤和保留已激活的核心优先级以供推荐。
 - 我们对从现实应用中收集的两个大规模数据集进行了广泛的实验。实验结果表明，与最先进的顺序推荐方法相比，性能有了显著提高。进一步的研究也验证了我们的方法能够有效地模拟长时间的行为序列。

• PROBLEM FORMULATION

- 在这里，我们提供了顺序推荐的正式定义。假设我们有一组项目，用 X 表示，其中 $x \in X$ 表示一个项目。项目数表示为 $|X|$ 。通常，用户与项目有一个顺序交互序列： $\{x_1, x_2, \dots, x_n\}$ ，其中 n 是交互次数， x_i 是用户与之交互的第 i 项。顺序推荐旨在预测与用户偏好匹配的下一个项目 x_{n+1} 。用户的偏好可以从按时间顺序排列的用户项目隐式反馈中推断出来。基于上述定义，顺序推荐的任务可以表述为：
 - 输入：每个用户的交互历史 $\{x_1, x_2, \dots, x_n\}$ 。
 - 输出：一个推荐模型，用于估计具有交互历史的用户 $\{x_1, x_2, \dots, x_n\}$ 将在第 $(n+1)$ 步与目标项目 x_t 交互。

• METHODOLOGY

- 图 1 说明了我们提出的 SURGE 模型，该模型由以下四个部分组成，我们将一一详述。
 - 兴趣图构造。
 - 通过基于度量学习将松散的项目序列重建为紧密的项目兴趣图，我们明确地整合和区分长期用户行为中的不同类型的偏好。
 - 兴趣融合图卷积层。
 - 在构建的兴趣图上进行图卷积传播，动态融合用户兴趣，强化重要行为，弱化噪声行为。
 - 兴趣提取图池化层。
 - 考虑用户在不同时刻的不同偏好，进行动态图池化操作，自适应地保留动态激活的核心偏好。
 - 预测层。
 - 在将池化图展平为简化序列后，我们对增强兴趣信号的演变进行建模，并预测用户很有可能与之交互的下一个项目。

• Interest Graph Construction

- 为了在用户丰富的历史行为中整合和区分不同类型的偏好，我们可以将松散的项目序列转换为紧密的项目-项目兴趣图。两个项目之间的共现关系是一个合理的构建标准，但挑战在于共现关系的稀疏性不足以为每个用户生成一个连通图。在本节中，我们提出了一种基于度量学习的新方法，可以为每个交互序列自动构建图结构，以探索其兴趣分布。
- 原始图形构造。
 - 这个新颖的模块尝试为每个交互序列构建一个无向图 $G = \{V, E, A\}$ ，其中 E 是要学习的图边集， $A \in R^{n \times n}$ 表示相应的邻接矩阵。每个顶点 $v \in V$ 与 $|V| = n$ 对应于一个交互项（并且相关的嵌入向量表示为 $h_v \in R^d$ ）。我们的目标是学习邻接矩阵 A ，其中每条边 $(i, j, A_{i,j}) \in E$ 表示项目 i 是否与项目 j 相关。
 - 通过将每个用户的交互历史表示为图表，更容易区分他/她的核心兴趣和外围兴趣。核心兴趣节点由于连接的相似兴趣较多，其度数高于外围兴趣节点，相似兴趣的频率越高，子图越密集越大。这样就构建了一个先验框架，即邻居节点相似，密集子图是用户的核心利益。
- 节点相似度度量学习。
 - 由于我们需要一个相邻节点相似的先验图，因此可以将图学习问题转化为节点相似度度量学习，与下游推荐任务联合训练。这种图构建方法通用，易于实现，能够完美应对归纳学习（测试时有新项目）。度量学习可以分为基于内核的方法和基于注意力的方法。基于核的方法的常见选项包括余弦距离、欧几里得距离和马氏距离。一个好的相似度度量函数应该是可学习的，以提高表达能力并具有可接受的复杂性。为了平衡表现力和复杂性，我们采用加权余弦相似度作为我们的度量函数，公式如下：（1）其中 \odot 表示Hadamard产品， α 是一个可训练的权重向量，以自适应地突出项目嵌入 h_i 和 h_j 的不同维度。请注意，学习到的图结构会随着项目嵌入的更新而不断变化。
 - 为了增加表达能力和稳定学习过程，相似度度量函数可以扩展到多头度量。具体来说，我们使用 ϕ （正面的数量）权重向量来计算 ϕ 独立的相似度矩阵（每个代表一个视角），使用上述相似度度量函数，并将它们的平均值作为最终相似度：（2）其中， M_{ij} 计算第 δ 头的两个项目嵌入 h_i 和 h_j 之间的相似性度量，每个头隐含地捕获不同的语义视角。
- 通过 ϵ -稀疏实现图的稀疏化。
 - 通常，邻接矩阵元素应该是非负的，但余弦值 M_{ij} 是根据 $[-1, 1]$ 之间的度量范围计算得出的。简单地对其进行归一化不会对图稀疏性施加任何约束，并且可以产生一个完全连接的邻接矩阵。这在计算上是昂贵的，并且可能会引入噪声（即不重要的边），并且它不够稀疏，以至于后续的图卷积不能专注于图的最相关方面。
 - 因此，我们只考虑具有最重要连接的节点对，从 M 中提取对称稀疏非负邻接矩阵 A 。为了使提取阈值的超参数不敏感并且不破坏图的稀疏分布，我们采用了整个图的相对排序策略。具体来说，我们屏蔽掉（即设置为零） M 中小于非负阈值的元素，该阈值是通过将 M 中的度量值进行排名而获得的。（3）其中 $\text{Rank}_{\epsilon n^2}(M)$ 返回 ϵn^2 -度量矩阵 M 中的最大值。 n 是节点数， ϵ 控制生成图的整体稀疏性。

- 它不同于整个图的绝对阈值策略和节点邻域的相对排序策略。前者设置了一个绝对阈值来删除邻接矩阵中的较小元素。当超参数设置不当时，随着embedding的不断更新，度量值分布也会发生变化，可能无法生成图或生成完整图。后者返回邻接矩阵中每一行的固定数量的最大值的索引，这将使生成的图的每个节点具有相同的度数。强制均匀稀疏分布会使下游的 GCN 无法充分利用图的密集或稀疏结构信息。
- 兴趣融合图卷积层
 - 如上所述，我们有可学习的兴趣图来区分不同的兴趣。核心利益和周边利益分别形成大集群和小集群，不同类型的利益形成不同的集群。此外，为了将弱信号收集到能够准确反映用户偏好的强信号，我们需要在构建的图中聚合信息。
 - 通过图形注意卷积进行兴趣融合。
 - 我们提出了一个集群和查询感知的图注意力卷积层，可以在信息聚合过程中感知用户的核心兴趣（即位于集群中心的项目）和与查询兴趣相关的兴趣（即当前目标项目）。输入是一个节点嵌入矩阵 $\{h_1, h_2, \dots, h_n\}, h_i \in \mathbb{R}^d$ ，其中 n 是节点数（即用户交互序列的长度）， d 是每个节点中嵌入的维度。该层产生一个新的节点嵌入矩阵 $\{h'_1, h'_2, \dots, h'_n\}, h'_i \in \mathbb{R}^{d'}$ ，作为其可能具有不同维度 d' 的输出。
 - 计算对齐分数 E_{ij} 以映射目标节点 v_i 在其邻居节点 v_j 上的重要性。一旦获得，归一化的注意力系数用于执行与其对应的嵌入的加权组合，在应用残差连接和非线性函数 σ 后作为每个节点的精细输出嵌入：(4)
 - 注意聚合函数可以是 Mean、Sum、Max、GRU 等函数，这里我们使用简单的 sum 函数，其他函数留待以后探索。为了稳定注意力机制的学习过程，我们采用类似于 [25, 26] 的多头注意力。准确地说， ϕ 独立注意力机制执行上述转换，然后将它们的嵌入连接为以下输出表示：(5) 其中 \parallel 表示连接操作， E_{δ} 是第 δ 注意力头得到的归一化注意力系数， W_{δ} 是对应的线性变换的权重矩阵。值得注意的是，最终返回的输出 h' 将对应于每个节点的 $\phi d'$ 维嵌入（而不是 d' ）。
 - 集群和查询感知注意力。
 - 为了在整合兴趣时增强重要信号并削弱噪声信号，我们提出了一种集群和查询感知注意机制。我们使用注意系数在消息传递过程中重新分配边缘信息的权重。注意机制考虑以下两个方面。
 - 首先，我们假设目标节点 v_i 的邻域将形成一个簇，并将图中的目标节点 v_i 视为簇 $c(v_i)$ 的中心点。我们将目标节点 v_i 的 k -hop 邻域定义为集群的感受野。集群中所有节点嵌入的平均值 $h_{c,c}$ 代表集群的平均信息。为了识别目标节点是否是集群的中心，使用目标节点嵌入及其集群嵌入来计算以下注意力分数，(6) 其中 W_c 是变换矩阵， \parallel 是连接算子， \odot 表示 Hadamard 积。在我们的实验中，注意力机制 Attention_c 是一个以 LeakyReLU 作为激活函数的两层前馈神经网络。
 - 其次，为了服务于下游动态池化方法，学习用户兴趣对不同目标兴趣的独立演化，还应考虑源节点嵌入 $h_{s,s}$ 和目标项目嵌入 $h_{t,t}$ 之间的相关性。如果源节点与查询项的相关性更高，则其在向目标节点的聚合中的权重会更显

着，反之亦然。由于只有相关的行为才能在最终的预测中发挥作用，所以我们只保留相关的信息，不相关的信息会在聚合时被丢弃。（7）其中 W_q 是变换矩阵， \parallel 是连接算子， \odot 表示 Hadamard 积。在我们的实验中，注意力机制 $Attention_q$ 是一个应用 LeakyReLU 非线性的两层前馈神经网络。其中 W_q 是变换矩阵， \parallel 是连接算子， \odot 表示 Hadamard 积。在我们的实验中，注意力机制 $Attention_q$ 是一个应用 LeakyReLU 非线性的两层前馈神经网络。

- 我们遵循附加注意机制同时考虑集群和查询的因素。我们将目标节点的集群分数和源节点的查询分数相加，作为源节点 j 对目标节点 i 的更新权重。为了使系数在不同节点之间易于比较，我们使用 softmax 函数在 j 的所有选择中对它们进行归一化。注意系数 E_{ij} 计算如下：（8）其中节点 i 的邻域 N_i 包括节点 i 本身。在包含自循环传播的上下文中（当 j 等于 i 时）， α_i 控制目标节点可以接收多少信息，而 β_j 控制源节点可以发送多少信息。

- 兴趣提取图池化层

- 隐式兴趣信号与显式兴趣信号的融合是通过对兴趣图进行信息聚合来完成的。在本节中，我们使用图池化方法进一步提取融合信息。类似于 CNN 中 Pooling 中特征图的下采样，图池化旨在合理缩小图的大小。通过对所构建的图结构进行粗化，将松散的兴趣转化为紧的兴趣并保持其分布。
- 通过图池进行兴趣提取。
 - 为了获得池化图，需要一个集群分配矩阵。假设存在一个软簇分配矩阵 $S \in R^{n \times m}$ ，它可以将节点信息池化为簇信息。 m 是反映池化程度的预定义模型超参数，其中 $m < n$ 。给定节点嵌入 $\{h_1, h_2, \dots, h_n\}$ 和节点得分 $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$ 的原始图，聚类嵌入和粗化图的分可以如下生成，（9）
（10）其中 γ_i 是通过对 β_i 应用 softmax 获得的，表示第 i 节点的重要性得分。分配矩阵 S 的每一行对应于一个 n 节点，每一列对应于一个 m 簇。它提供了每个节点到相应集群的软分配。上述方程根据集群分配 S 聚合节点嵌入和分数，从而为每个 m 集群生成新的嵌入和分数。
 - 接下来，我们讨论如何学习节点的可微软簇分配 S 。我们使用 GNN 架构来生成分配矩阵。基于邻接矩阵和节点嵌入，通过标准消息传递和 softmax 函数获得分配映射的概率矩阵。（11）其中权重矩阵 W_p 的输出维度对应于最大簇数 m 。softmax 函数用于获取第 i 节点被划分为 m 簇之一的概率。值得注意的是，我们可以通过执行 $STAS$ 获得池化图的邻接矩阵 A^* ，确保集群之间的连通性。然后，上述方程的重复可以执行多层池化，以实现感兴趣的分层压缩。
- 任务正规化。
 - 然而，仅使用来自下游推荐任务的梯度信号很难训练集群分配矩阵 S 。非凸优化问题容易在训练初期陷入局部最优。此外，每个节点嵌入的相对位置 $\{h_{\square^1}, h_{\square^2}, \dots, h_{\square^m}\}$ 对应于交互的时间顺序。但是在池化的聚类嵌入矩阵 $\{h_{\square^1}^*, h_{\square^2}^*, \dots, h_{\square^m}^*\}$ 中，反映用户兴趣的聚类之间的时间顺序很难得到保证。因此，我们使用三个正则化项来缓解上述问题。
 - 同样的映射正则化。

- 为了使连接强度较大的两个节点更容易映射到同一个簇，使用第一个正则化如下，（12）其中 $\|\cdot\|_F$ 表示 Frobenius 范数。邻接矩阵 A 中的每个元素代表两个节点之间的连接强度， SST 中的每个元素代表两个节点被划分到同一个簇的概率。
 - 单一隶属正规化。
 - 为了清楚地定义每个簇的从属关系，我们使每一行 S_i ：在分配矩阵中通过正则化熵来逼近 one-hot 向量，如下所示，（13），其中 $H(\cdot)$ 是可以减少不确定性的熵函数映射分布。最优情况是第 i 节点只映射到一个簇，此时熵 $H(S)$ 为 0。
 - 相对位置正则化。
 - 下游兴趣演化建模需要维护池化前后用户兴趣的时间顺序。然而，在池化簇嵌入矩阵 $\{h, h, \dots, h\}$ 不可微。因此，我们设计了一个位置正则化来确保池化期间集群之间的时间顺序，如下所示，（14），其中 P_n 是位置编码向量 $\{1, 2, \dots, n\}$ 和 P_m 是位置编码向量 $\{1, 2, \dots, m\}$ 。最小化 L2 范数使得 S 中非零元素的位置更靠近主对角线元素。直观地说，对于原始序列中位置靠前的节点，其所分配到的簇的位置索引往往在靠前。
- 图表读数。
 - 至此，我们得到了一个紧密粗化的图 G^* ，代表了用户更强的兴趣信号。同时，我们对原始图 G 执行加权读出以约束每个节点的重要性，在传播层的前向计算之后聚合所有节点嵌入以生成图级表示 hg ：（15）其中权重是池化前每个节点的得分 γ_i ，Readout 函数可以是 Mean、Sum、Max 等函数。我们这里使用简单的 sum 函数来保证置换不变，其他函数留待以后探索。我们将此图级表示馈送到最终预测层，以更好地提取池化层中每个集群的信息。
- 预测层
 - 兴趣演化建模。
 - 在外部环境和内部认知的共同作用下，用户的核心利益不断演变。用户可能会在一段时间内对各种运动产生兴趣，而在另一时间需要书籍。但是，仅仅使用上面提到的读出操作并没有考虑核心利益之间的演变，这无疑会造成时间顺序的偏差。为了给最终的利益表示提供更多相关的历史信息，还需要考虑利益之间的时间关系。
 - 在外部环境和内部认知的共同作用下，用户的核心利益不断演变。用户可能会在一段时间内对各种运动产生兴趣，而在另一时间需要书籍。但是，仅仅使用上面提到的读出操作并没有考虑核心利益之间的演变，这无疑会造成时间顺序的偏差。为了给最终的利益表示提供更多相关的历史信息，还需要考虑利益之间的时间关系。（16）众所周知，GRU 克服了 RNN 的梯度消失问题，并且比 LSTM 更快。此外，为了更好地利用兴趣提取层中融合兴趣的重要性权重 γ_i^* ，我们采用带有注意力更新门（AU-GRU）的 GRU 将注意力机制和 GRU 无缝结合。AUGRU 使用注意力

分数 γ^* 来缩放所有更新门的维度，这导致较少的相关兴趣对隐藏状态的影响较小。更有效地避免了兴趣漂移带来的干扰，推动相对兴趣平稳演化。

- 预测

- 我们将兴趣提取层的图级表示和兴趣演化层的演化输出作为用户当前的兴趣，并将它们与目标项目嵌入连接起来。给定连接的密集表示向量，完全连接的层用于自动学习嵌入的组合。我们使用两层前馈神经网络作为预测函数来估计用户在下一刻与物品交互的概率，实验部分的所有比较模型都将共享这个流行的设计，（17）
- 根据现实行业中的 CTR（点击率）预测，我们使用负对数似然函数作为损失函数，并与所有比较模型共享此设置。优化过程是最小化损失函数以及 $L2$ 正则化项以防止过度拟合，（18）其中 O 是训练集， $|O|$ 是训练实例的数量。 Θ 表示可训练参数的集合， λ 控制惩罚强度。标签 $y_o = 1$ 表示正例， $y_o = 0$ 表示负例。而 y^o 代表网络在 softmax 层之后的输出，代表下一个项目被点击的预测概率。此外，将第 3.3.2 节中的三个正则化项添加到最终推荐目标函数中，以获得更好的性能和更可解释的集群分配。

- EXPERIMENT

- 在本节中，我们对两个真实世界的数据集进行序列推荐实验，以评估我们提出的方法，目的是回答以下三个问题。
 - 与最先进的顺序推荐器相比，所提出的方法表现如何？
 - 所提出的方法能否有效且高效地处理各种长度的序列？
 - 方法中不同成分的作用是什么？

- 总结

- 模型架构

- Interest Graph Construction（兴趣图架构）
- Interest-fusion Graph Convolutional Layer（兴趣融合图卷积层）
- Interest-extraction Graph Pooling Layer（兴趣提取图池化层）
 - 隐式兴趣信号与显式兴趣信号的融合是通过对兴趣图进行信息聚合来完成的。
 - Interest extraction via graph pooling（通过图池化提取兴趣）
 - Assignment regularization（赋值正则化）
- Graph readout（图形读数）
 - 至此，我们得到了一个紧密粗化的图 G^* ，代表了用户更强的兴趣信号。
 - 同时，我们对原始图 G 执行加权读出以约束每个节点的重要性，在传播层的前向计算之后聚合所有节点嵌入以生成图级表示 hg ：（15）其中权重是池化前每个节点的得分 γ_i ，Readout 函数可以是 Mean、Sum、Max 等函数。我们这里使用简单的 **sum 函数** 来保证置换不变，其他函数留待以后探索。我们将此图级表示馈送到最终预测层，以更好地提取池化层中每个集群的信息。



Prediction Layer (预测层)

幕布 - 极简大纲笔记 | 一键生成思维导图