

Symmetry Oracle-guided Dynamic User Preference Modeling for Sequential Recommendation

Mingyu Ge

Nankai University

Tianjin, China

2312388@nankai.edu.cn

1 背景与意义

顺序推荐方法可以从用户历史交互中捕获动态用户偏好，以实现更好的性能。oracle 引导的动态用户偏好建模方法 (Oracle4Sec)，除了过去的信息，训练期间还可以获得未来的信息，其中包含未来的“预言机”用户偏好，这将有利于对动态用户偏好进行建模。但是，过去信息和未来信息的关系仍然不够平衡，提出了一种对称式 oracle 引导的动态用户偏好建模方法 (SymmetryOracle)，用于顺序推荐，该方法利用未来信息来指导对过去信息以及过去信息指导未来信息预测的模型训练，旨在学习“前瞻性”模型以及平衡过去信息与未来信息之间的指导与被指导关系。

2 现状分析

推荐系统可以根据用户的历史互动向用户推荐潜在感兴趣的项目，已被广泛应用于各个领域，例如广告、电影推荐和电子商务。本文中，我提出了一种对称式 oracle 引导的动态用户偏好建模方法 (SymmetryOracle)。

具体来说，SymmetryOracle 包括：首先通过两个单独的编码器提取过去和未来的信息，然后通过 InverseOracleGuide 模块，用过去信息规范未来信息的获取。再通过 OracleGuide 模块学习前瞻性模型，该模块将过去和未来信息之间的差异降至最低。还定制了一个 3PTraining 训练策略，使指导更加有效。

3 方案设计

如图 1 所示，SymmetryOracle 由四部分组成：过去信息编码器、未来信息编码器、InverseOracle 指导模块和 Oracle 引导模块。前两个编码器均由嵌入查找层、噪声滤波模块、因果自关注模块和交互预测层组成。为了便于演示，Oracle 引导模块被设置为在 3D 坐标系中最小化过去信息和未来信息之间的距离。这里的过去信息编码器将后文中的过去信息经验编码器和过去信息结果编码器合在一起绘制。

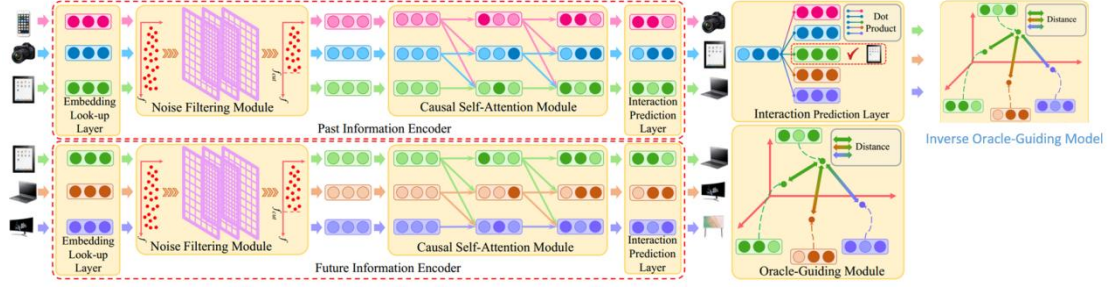


图 1:SymmetryOracle 的架构

4 技术路线

4.1 符号

设用户集和项目集分别为 U 和 V , $|U| = n$ 和 $|V| = m$ 。用户 u 的交互序列按时间顺序可表示为 $S^u = \{v_i | v_i \in V, i = 1, 2, \dots, |S^u|\}$ 。我们的目标是根据用户的交互序列 S^u 预测用户将与之交互的下一个项目 $v_{|S^u|+1}$ 。为了便于下面的描述，我们给出了两种用户交互序列的定义。对于需要为用户 u 预测的目标交互 $v_t \in V$ ，他/她的历史相互作用序列可以定义为 $H_t^u = \{v_1, v_2, \dots, v_{t-1}\}$ 并且他/她的全局相互作用序列被定义为 $G_t^u = \{v_1, v_2, \dots, v_{t-1}, v_t, v_{t+1}, \dots, v_{t+p}\}$ 与用户历史交互序列相比，它还包含 $p + 1$ 个项目。

为了在训练和推理过程中提高效率，我们通常采用固定长度的历史交互序列和全局交互序列，因此我们需要截断或填充序列 H_t^u 和 G_t^u ，以确保它们的长度等于正整数 L 。如果序列的长度大于 L ，我们通过选择最近的交互将其长度截断为 L ，而如果序列的长度小于 L ，我们在序列的左侧重复添加一个填充项，直到它的长度为 L 。因此，我们将历史交互序列重新定义为 $H_t^u = \{v_{t-L}, v_{t-L+1}, \dots, v_{t-1}\}$ ，并将全局交互序列定义为 $G_t^u = \{v_{t+P-L+1}, v_{t+P-L+1}, \dots, v_{t+P}\}$ 。

4.2 Past Information Experience Encoder

4.2.1 Embedding Look-up Layer、Noise Filtering Module 与 Causal Self-Attention Module

在论文[1]中均有详细的介绍与公式的解析。

4.2.2 Interaction Prediction Layer

在获得预测的项目嵌入 S 后，我们可以根据 H_t^u 计算目标交互 v_t 的概率：

$$P(v_t | H_t^u) = \text{Sigmoid}(Q_L^T T_{v_t}) \quad (1)$$

其中 Q_L 是目标相互作用的预测嵌入，对应于 Q 的最后一行。 T_{v_t} 是嵌入表 T 中目标交互的真实嵌入。过去信息经验编码器的损失函数：

$$\mathcal{L}_{p_1} = - \sum_{u=1}^{|U|} \sum_{t=1}^{|S^u|} \sum_{l=1}^L [\log(P(v_{m+l} | H_{t,l}^u)) + \log(1 - P(j_l | H_{t,l}^u))] \quad (2)$$

其中 S^u 是用户的交互序列, $J = \{j_1, j_2, \dots, j_L\}$ 是 L 个用户从未与之交互过的负面项目。 $H_{t,l}^u$ 是 H_t^u 的一个子序列 $m = t - L$ 。

4.3 Past Information Conclusion Encoder

4.3.1 Embedding Look-up Layer、Noise Filtering Module 与 Causal Self-Attention Module

在论文[1]中均有详细的介绍与公式的解析。

4.3.2 Interaction Prediction Layer

在获得预测的项目嵌入 S 后, 我们可以根据 H_t^u 计算目标交互 v_t 的概率:

$$P(v_t|H_t^u) = \text{Sigmoid}(Q_L^T T_{v_t}) \quad (3)$$

其中 Q_L 是目标相互作用的预测嵌入, 对应于 Q 的最后一行。 T_{v_t} 是嵌入表 T 中目标交互的真实嵌入。过去信息结果编码器的损失函数:

$$\mathcal{L}_{p_2} = - \sum_{u=1}^{|V|} \sum_{t=1}^{|S^u|} [\log(P(v_t|H_t^u)) + \log(1 - P(j|H_t^u))] \quad (4)$$

其中 S^u 是用户的交互序列, $j \notin S^u$ 是一个用户从未与之交互过的负面项目。

4.4 Future Information Encoder

4.4.1 Embedding Look-up Layer、Noise Filtering Module 与 Causal Self-Attention Module

在论文[1]中均有详细的介绍与公式的解析。

4.4.2 Interaction Prediction Layer

在获得预测的项目嵌入 R 后, 我们可以根据 G_t^u 计算目标交互 v_t 的概率:

$$P(v_{t+P+1}|G_t^u) = \text{Sigmoid}(R_L^T T_{v_{t+P+1}}) \quad (5)$$

其中 R_L 是目标相互作用的预测嵌入, 对应于 R 的最后一行。 T_{v_t} 是嵌入表 T 中目标交互的真实嵌入。因此, 未来信息编码器的损失函数为:

$$\mathcal{L}_f = - \sum_{u=1}^{|V|} \sum_{t=1}^{|S^u|} \sum_{l=1}^{P+2} [\log(P(v_{m+l}|G_{t,l}^u)) + \log(1 - P(j_l|G_{t,l}^u))] \quad (6)$$

其中 S^u 是用户的交互序列, $J = \{j_1, j_2, \dots, j_L\}$ 是 L 个用户从未与之交互过的负面项目。 $G_{t,l}^u$ 是 G_t^u 的一个子序列 $m = t - 1$ 。这里对论文[1]中的损失函数做了修改, 将 $l = 1, \dots, L$ 改为了 $l = 1, \dots, P + 2$ 。

4.5 Oracle-Guiding Module

在论文[1]中均有详细的介绍与公式的解析。

α_i 改为 α_{2i} , γ 改为 γ_2 。

4.6 Inverse Oracle-Guiding Module

这里的思路来源于 **Oracle-Guiding Module**。未来信息预测的 v_{t+P+1} 同样可以借助过去信息预测积累的经验帮助未来信息进行预测。

给定两个用户交互序列 H_t^u 和 G_t^u ，我们可以分别从过去和未来的信息编码器中获得两个项目嵌入 Q 和 R 。我们的目标是尽量减少过去信息编码器（即 Q_L （过去信息））预测的目标交互 v_t 嵌入与未来信息编码器（如 $R_{L-P-2+i}$ （ $i = 1, 2, \dots, P + 2$ ）（未来信息）），前一种嵌入可以通过后一种嵌入捕捉用户偏好在未来的变化。因此，oracle 引导模块的损失函数为：

$$\mathcal{L}_{Ig} = \sum_{i=1}^{P+2} \alpha_{li} \cdot f(Q_L, R_{L-P-2+i}) \quad (7)$$

其中 $\alpha_{li} \in (0,1]$ 是用于区分 $P+2$ 差异重要性的权重。一般来说，引导者应该更多地关注目标交互附近的未来信息，而较少关注远离目标交互的未来信息，因此，我们使用指数衰减来建模：

$$\alpha_{li} = e^{-\gamma_1(i-1)} \quad (8)$$

其中 γ_1 是衰减系数。由于 Q_L 和 R_{L-P-1i} 对应于同一项 v_t ，差异 $f(Q_L, R_{L-P-2+i})$ 的重要性最高，因此权重 α_{l1} 为 1。 $f(\cdot)$ 是一个差异测量函数。

4.7 3PTraining: A Tailored Training Strategy

我定制了一种有效的三阶段模型训练策略 3PTraining，该策略首先训练具有损失的过去信息编码器 \mathcal{L}_{p_1} 为了获取过去信息预测的经验，然后联合训练未来信息编码器和 Inverse Oracle 引导模块，损失 $\mathcal{L}_f + \beta_l \mathcal{L}_{Ig}$ ，以获取未来信息，实现从过去信息推导未来信息对未来信息预测的引导。再利用获取的未来信息指导重新审视过去信息的提取，即联合训练过去信息编码器和 Oracle 引导模块，重新修改过去编码器的各层权重，损失 $\mathcal{L}_{p_2} + \beta_g \mathcal{L}_g$ 。

4.8 Inference

在论文[1]中均有详细的介绍与公式的解析。

4.9 Discussion

4.9.1 Past Information Encoder

本文中将 Past Information Encoder 拆成两个模块：过去信息预测经验和过去信息预测结果。过去信息预测经验采用 \mathcal{L}_{p_1} ，是为了更细致的积累经验，充分利用过去信息，使对

未来信息获取的指导更合理。过去信息预测结果采用 \mathcal{L}_{p_2} ，是因为我们已经得到了由过去信息指导获取的未来信息，此时只需要由未来信息指导过去信息进行对 v_t 的预测。

4.9.2 Future Information Encoder

关于在 Future Information Encoder 中的损失函数是论文[1]中的损失函数做了修改，将 $l = 1, \dots, L$ 改为了 $l = 1, \dots, P + 2$ 。原因是过去信息预测经验 \mathcal{L}_{p_1} 利用 v_t 之前的信息进行经验的积累，再利用这个经验指导未来 v_t, \dots, v_{P+1} 的提取。所以 \mathcal{L}_f 的损失函数只需要对 v_t, \dots, v_{P+1} 部分进行计算。因此做出了上述的修改。

5 总结

针对论文[1]中的 Oracle4sec 架构，提出了一种更对称的 oracle 引导的动态用户偏好建模方法（SymmetryOracle），用于顺序推荐，该方法利用未来信息来指导对过去信息以及过去信息指导未来信息预测的模型训练，旨在学习“前瞻性”模型以及平衡过去信息与未来信息之间的指导与被指导关系。

参考文献

[1]Jiafeng Xia, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang and Ning Gu. 2024 Oracle-guided Dynamic User Preference Modeling for Sequential Recommendation. arXiv:2412.00813v1 [cs. IR] 1 Dec 2024.