

# 一种超图和自监督的协同训练算法

徐一波, 李昊昱

河北地质大学信息工程学院, 河北 石家庄

收稿日期: 2024年1月25日; 录用日期: 2024年2月22日; 发布日期: 2024年2月29日

## 摘要

为了针对会话推荐中节点数量与图规模不断扩增而产生的节点稀疏问题, 提出一种超图与自监督协同训练算法——DHCN-COTREC (Dual-Channel Convolutional for Hypergraphs and Collaborative Training Recommendation)。首先, 构建两个不同的图编码器, 将数据进行编码用于自监督学习, 产生正负样本信息; 其次, 引入协同训练算法, 并加入发散约束进行对比学习, 从而最大化两个编码器之间的互信息; 最后, 根据序列中最后一个项目与正样本之间的余弦距离, 向用户推荐TOP-K个项目。本文在Tmall与Diginetica真实数据集上进行实验, 结果表明, DHCN-COTREC算法的准确度和平均倒数排名比目前最优的DHCN算法分别提高了26%和1.92%。证实了DHCN-COTREC算法的有效性。

## 关键词

超图, 会话推荐, 自监督, 协同训练, 发散约束

# A Collaborative Training Algorithm Using Hypergraph and Self-Supervision

Yibo Xu, Haoyu Li

College of Information Engineering, Hebei GEO University, Shijiazhuang Hebei

Received: Jan. 25<sup>th</sup>, 2024; accepted: Feb. 22<sup>nd</sup>, 2024; published: Feb. 29<sup>th</sup>, 2024

## Abstract

In order to address the problem of sparse nodes in recommendation systems caused by the increasing number of nodes and graph size in session-based recommendation, a co-training algorithm called DHCN-COTREC using hypergraphs and self-supervised learning is proposed. Firstly, two different graph encoders are constructed to encode the data for self-supervised learning, generating positive and negative sample information. Secondly, a co-training algorithm is introduced, and divergent constraints are added for contrastive learning to maximize the mutual information between

the two encoders. Finally, TOP-K items are recommended to users based on the cosine distance between the last item in the sequence and positive samples. The experiment was conducted on Tmall and Diginetica real datasets, and the results showed that the accuracy and average reciprocal rank of DHCN-COTREC algorithm were improved by 26% and 1.92% respectively compared with the current optimal DHCN algorithm, which confirms the effectiveness of the DHCN-COTREC algorithm.

## Keywords

Hypergraph, Session-Based Recommendation, Self-Supervision, Collaborative Training, Divergent Constraints

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

会话推荐是一种分析和挖掘用户当前语境和输入文本, 并生成相应建议或回答的算法。随着在线客服、智能语音助手、聊天机器人等实时对话应用的兴起, 会话推荐算法成为帮助用户快速解决问题或获取信息的重要工具[1]。目前, 会话推荐算法已广泛应用于电子商务、社交网络、音乐、电影等领域[2], 大大提高了用户满意度并对平台的收益产生了积极影响。

会话推荐算法主要有循环神经网络(RNN)和图神经网络(GNN)两种建模方式。先前的会话推荐算法只考虑用户的最后一次点击或选择, 忽略了过去点击信息。基于 RNN 的算法 SR-RNN [3]首次将递归神经网络(Recurrent Neural Networks, RNN)应用于推荐系统, RNN 将会话数据作为单向序列进行建模。但当一个会话中, 用户行为数量十分有限时, RNN 就很难捕获用户的行为表示。因为使用 RNN 建模时, 如果前面时序的动作项较少, 最后一个输出结果的准确性就比较低。基于 GNN 的算法 SR-GNN [4]将历史会话序列构造为有向图, 捕捉项目之间的转化, 生成准确的项目嵌入向量, 无需严格按照时间顺序对项目进行排序。但是 GNN 忽略了会话数据中项目之间的复杂相关性。在真实场景下, 项目之间的转化通常是由之前各项的点击行为联合触发, 而且项目与项目之间存在高度关联, 简单的图结构无法描述这种复杂的集合关系。

超图是一种比普通图更广义的图表示方法, 传统图中的边只能连接两个节点, 而超图中的边可以连接任意数量的节点。这使得超图能够表达更高阶的数据结构, 并且更加灵活地处理数据的相关性。超图通常由顶点集和超边集构成, 其中超边集可以连接任意数量的顶点。这样可以建立高阶数据的相关性, 同时超边是集合形式的, 强调了所涉及元素的一致性而不是相对顺序, 因此使用超图可以更好地捕捉会话中各项目之间的复杂交互。

最早将图卷积应用于超图的是 HGNN [5], 提出了超图神经网络框架。使用超图卷积操作解决了复杂数据表示学习中的高阶数据结构和数据相关性问题。同时在引文网络分类和视觉对象识别任务上进行了实验, 与其他传统方法相比, HGNN 在多模态数据处理方面表现更好。但该模型存在一些局限性。例如, 该模型在处理大规模数据时可能会遇到困难, 因为它需要计算超图中所有节点之间的关系。此外, 该模型还需要对超图进行预处理, 以便将其转换为标准图形式。HyperGCN [6]提出了基于超图的半监督学习方法, 使用频谱图卷积和基于超图嵌入的思想, 更好地使用超图对复杂关系进行建模。该论文中使用真实世界的超图数据集、DBLP 数据集和组合优化数据集进行实验, 证明 HyperGCN 的有效性和优越性, 特别是在一些有大量噪声超边的超图数据集上。Jiang 等人[7]提出了一种动态超图神经网络, 该网络由动态

超图构建模块(DHG)和超图卷积模块(HGC)组成。DHG 模块会动态地更新每一层超图节点表征以适应更好的分类任务; HGC 模块包括顶点卷积和超边卷积两个阶段, 分别用于在超图中对顶点和超边特征做聚合。Bandyopadhyay 等人[8]开发了直线超图卷积网络, 提出了一种基于线图的超图卷积网络(LHCN)算法, 首次在超图学习中使用了超图的线图这一概念。该算法将超图映射到一个带权且带属性的线图里, 然后在线图上执行标准的图卷积操作。这是一种将图卷积应用于具有可变超边大小的新方法。在会话推荐系统中, 基于超图的算法能更好地挖掘用户和项目的深层次特征, 因此使用超图进行建模在会话推荐领域受到了广泛的关注。例如, Bu 等人[9]针对音乐推荐系统中存在的多种类型的社交媒体信息和音乐声学内容, 将协同过滤和基于声学的音乐推荐相结合, 构建了一个综合的推荐框架, 通过使用超图模型, 提出了一种新的音乐推荐算法, 能够很好地处理不同类型对象和关系之间的建模问题。但由于模型依赖特定的媒体信息和音乐内容, 使得模型的泛化能力不强, 如果这些信息发生变化或者缺失, 可能会影响推荐的效果。Li 等人[10]使用超图来模拟新闻数据中不同对象之间的各种高阶关系, 并将新闻推荐问题转化为对细粒度超图的排序问题, 通过传递推理来有效地缓解冷启动问题。但如何将该方法应用到实际场景中进行规模化和在线实时推荐, 需要进一步研究。HyperRec [11]将超图学习和推荐系统相结合, 使用超图来建模用户在短期内对下一项推荐的偏好。然而, 这些方法并未充分利用超边之间的信息, 并且也不是专门为基于会话的场景设计的。Xie 等人[12]提出了一种基于超图的会话推荐方法来解决这些问题。该方法采用了双通道超图卷积神经网络, 可以充分考虑项目之间的关系。这是首个将超图神经网络与会话推荐相结合的算法。尽管基于超图的方法提升了数据处理的能力, 但仍存在会话信息挖掘不充分和项目评分稀疏等问题。

与此同时, 自监督学习(Self-Supervised Learning)也受到了广泛关注, 与传统的有监督学习不同, 自监督学习不需要人工标注大量数据集, 它利用未标记数据进行训练。通过设计预训练任务(即自监督任务)从大量未标记的数据中提取有用、可迁移的知识, 以减少对手动标签的依赖。自监督学习任务通常是通过将数据中的某些信息来生成“伪标签”, 然后将这些“伪标签”作为监督信号来训练模型。这种方法可以处理大规模未标记数据, 提高模型的泛化能力和鲁棒性。

受文献[12]启发, 本文提出了一种基于超图和自监督协同训练的会话推荐算法 DHCN-COTREC。通过引入自监督信号, 改善数据稀疏的问题, 通过引入发散约束, 改善输出结果相似度过高的问题。在 Tmall 和 Diginetica 两个数据集上进行对比试验, 实验结果表明 DHCN-COTREC 算法性能优于其他会话推荐算法。

## 2. 相关算法

### 2.1. 超图卷积算法

超图卷积算法定义如下(假设包含  $N$  个节点和  $M$  个超边):

$$X_i^{(l+1)} = \sum_{j=1}^N \sum_{\epsilon=1}^M H_{i\epsilon} H_{j\epsilon} W_{\epsilon\epsilon} X_j^{(l)} \quad (1)$$

$X_i^{(l+1)}$  表示节点  $j$  在第  $l$  层的表示向量;  $\sum_{j=1}^N H_{j\epsilon}$  表示每个超边的度;  $\sum_{\epsilon=1}^M H_{i\epsilon}$  表示每个顶点的度;  $W_{\epsilon\epsilon}$  表示权重。

参照 Wu 等人[13]建议, 不使用非线性激活函数和卷积滤波器参数矩阵, 具有更简单的模型结构, 并在某些场景下可以更快地训练和推理。(1)式归一化后的形式为:

$$X_h^{(l+1)} = D^{-1} H W B^{-1} H^T X_h^{(l)} \quad (2)$$

$D$  是节点度矩阵;  $H$  是超图的邻接矩阵;  $W$  是权重矩阵;  $B$  是超边度矩阵;  $X^{(l)}$  是节点第  $L$  层的特征;  $X_h^{(l)}$  表示第  $l$  层超图卷积操作后的节点特征表示(下标  $h$  表示超图节点)。

超图卷积可以看作是对超图结构进行“节点-超边-节点”特征变换的两阶段精化。乘法操作  $H^T X_h^{(l)}$  定义了从节点到超边的信息聚合, 然后预乘  $H^T$ , 表示将信息从超边聚合到节点。将  $X^{(0)}$  通过  $L$  层超图卷积后对每一层的项求平均, 得到最终项的表示:

$$\frac{1}{L+1} \sum_{l=0}^L X_h^{(l)} \quad (3)$$

由于上述操作没有涉及序列特征, 在这里引入一个可训练的位置编码矩阵(即时间信息)。用户的兴趣和行为习惯会随着时间的推移而发生变化, 因此考虑时间因素可以更好地反映用户的兴趣演化过程。位置编码矩阵是 Transformer [14]中引入的一种有效的技术。在超图卷积中通过一个可学习的位置矩阵  $P_r = [p_1, p_2, p_3, \dots, p_m]$  来体现用户会话中项目的顺序关系, 其中  $m$  是当前会话的长度, 在会话  $s$  中嵌入的第  $t$  个项  $s = [i_{s,1}, i_{s,2}, i_{s,3}, \dots, i_{s,m}]$  为:

$$X_t^* = \tanh(W_1[X_t \parallel P_{m-i+1}] + b) \quad (4)$$

$W_1 \in \mathbb{R}^{d \times 2d}$  与  $b \in \mathbb{R}^d$  均为可学习参数,  $P_{m-i+1}$  是位置编码矩阵  $p$  中第  $m-i+1$  个位置的嵌入。

加入位置信息的主要目的是由于会话的长度是不固定的, 前向位置信息无法衡量当前项目和预测项目之间的距离。

通过聚合该会话中的项目表示来体现会话嵌入, 参照 SR-GNN 中使用的方式来细化会话  $s = [i_{s,1}, i_{s,2}, i_{s,3}, \dots, i_{s,m}]$ :

$$\alpha_t = f^T \sigma(W_2 x_s^* + W_3 x_t^* + c), \theta_h = \sum_{t=1}^m \alpha_t x_t^* \quad (5)$$

其中  $x_s^* = \frac{1}{m} \sum_{t=1}^m x_m$ ,  $x_m$  表示会话嵌入, 对每个项目嵌入求平均获得整个会话的表示(通过对会话中所有项目的嵌入进行平均, 可以得到一个代表整个会话的嵌入向量), 等价于从用户的浏览记录中获取平均兴趣,  $x_t^*$  为会话中的第  $t$  个项目的嵌入, 前半部分是根据平均嵌入和每个项目来计算相应项目的重要性, 进行加权求和后得到能体现出用户兴趣的会话嵌入  $\theta_h$ 。与直接求平均相比, 加权平均得到的结果还能够起到一定程度的降噪。

## 2.2. 自监督协同训练算法

自监督协同训练算法是一种自监督学习算法, 可以在缺乏大量有标签数据的情况下进行学习, 并且可以从大量未标注的数据中进行学习, 从而提高模型在未知数据上的泛化能力, 并减少标记数据的成本。该算法的步骤如下: 1) 从两个视角建立两个图, 分别设置两个编码器从两个图中学习嵌入; 2) 对两个编码器进行协同训练, 通过迭代更新两个编码器, 以生成更富含信息的样本; 3) 使用对比学习定义损失函数, 学习共同表示空间, 使得不同视图之间的互信息最大化。帮助学习到更加一致和有用的表示; 4) 在多次迭代之后, 两个编码器生成的嵌入会相似, 为增加对抗性扰动样本, 可以设置发散约束; 5) 同时优化推荐损失, 自监督损失和差异化损失。

## 3. 结合超图和自监督协同训练会话推荐算法

会话推荐算法是基于会话数据构建的。根据会话长度, 会话数据大致可以分为长、中、短三种类型, 每种类型都具有其特征。然而, 由于用户行为的不确定性, 一个会话中可能包含与所需交互无关的内容。

随着数据规模的扩大, 无关数据的数量也在增加, 这导致了数据稀疏程度逐渐增强。因此, 预测的评分可能偏离实际的评分, 从而降低了推荐的效果。为此, 本文提出了一种方法: 首先引入超图模型来捕捉会话间项目一致性和多对多的高阶关系, 然后使用自监督学习缓解数据稀疏问题。

### 3.1. 超图与线图建模

在超图中, 每一个会话都将视为一个超边, 学习项目间的关系。设  $G=(V, E)$  表示一个超图, 其中  $V$  是包含  $N$  个唯一顶点的集合,  $E$  是包含  $M$  个超边的集合。超边  $\varepsilon \in E$  包含两个或多个顶点, 并被分配一个正权值  $W_{\varepsilon\varepsilon}$  (初始化为 '1'), 所有权值构成一个对角矩阵  $W \in \mathbb{R}^{M \times M}$ 。关联矩阵  $H \in \mathbb{R}^{N \times M}$  表示当前点属于哪些超边, 其中  $N$  表示项目的个数、 $M$  表示超边的个数。如果该点在这条超边中则  $H_{i\varepsilon}=1$ , 否则  $H_{i\varepsilon}=0$ 。

此外, 还定义了节点度矩阵  $D$  和超边度矩阵  $B$ 。节点度矩阵  $D: D_{ii} = \sum_{\varepsilon=1}^M W_{\varepsilon\varepsilon} H_{i\varepsilon}$ , 对于每个节点  $i$ , 计算  $i$  在超图中的度数(即与  $i$  相连的所有超边的权重之和), 然后将结果存储在  $D$  的对角线上。节点度矩阵  $D$  的作用是对节点特征进行归一化, 以便在超图卷积中更好地捕捉节点之间的关系。超边度矩阵  $B: B_{\varepsilon\varepsilon} = \sum_{i=1}^N H_{i\varepsilon}$ , 对于每个超边  $\varepsilon$ , 计算  $\varepsilon$  在超图中的度数(即与  $\varepsilon$  相连的所有节点的数量), 然后将其存储在  $B$  的对角线上。超边度矩阵  $B$  的作用是对超边特征进行归一化, 以便在超图卷积中更好地捕捉超边之间的关系。之后进行卷积操作  $X_h^{(l+1)} = D^{-1} H W B^{-1} H^T X_h^{(l)}$ ,  $D^{-1}$  表示归一化,  $H W B^{-1} H^T$  部分是一个超图卷积, 其中  $H$  表示点到超边的关系;  $W B^{-1}$  是超边到超边的关系;  $H^T$  是超边到点的关系, 最后乘以上一层的状态  $X_h^{(l)}$ 。重复上述计算过程至  $L$  层, 累加每一层的结果得到项目的表示, 并加入反向的位置编码作为非线性变换得到最终的表示。

线图是一种将超图中的超边转换为节点, 将超边之间的关系转换为节点之间关系的图, 是对整个会话的表示。对于一个超图  $G=(V, E)$ , 其线图  $L(G)=(V', E')$  的节点集合  $V'$  是超边集合  $E$  中的每一个超边对应的一个节点, 即  $V' = \{e | e \in E\}$ , 而集合  $E'$  是由超边之间的交集构成的, 当超边  $e_1$  与超边  $e_2$  在超图  $G$  中至少有一个节点相交时, 在线图  $L(G)$  中对应的节点  $e_1$  和  $e_2$  之间就存在着一一条边, 边的权重定义为  $W_{p,q} = \frac{|e_p \cap e_q|}{|e_p \cup e_q|}$ , 表示相同项目的交集除以项目的并集。接下来初始化线图节点, 由于在线图中只能获得项目的嵌入, 不能获得会话的初始嵌入, 故将初始值设置为最开始节点的均值表示  $\theta_i^{(0)} = \frac{1}{n} \sum_{j=1}^m x_j^{(0)}$ 。最后加入  $\theta_i^{(l+1)} = \hat{D}^{-1} \hat{A} \theta_i^{(l)}$ , 其中  $\hat{A} = A + I$  表示线图结构的邻接矩阵,  $I$  表示单位矩阵,  $D$  表示  $\hat{D}_{p,p} = \sum_{q=1}^m \hat{A}_{p,q}$ 。

经过  $L$  层卷积操作以后,  $\theta$  能得到会话层的信息传播结果, 并对累加每层传播得到的结果求平均  $\theta_l = \frac{1}{L+1} \sum_{l=0}^L \theta_l^{(l)}$ 。

### 3.2. 超图和自监督的协同训练算法

模型图如图 1 所示。首先, 将会话数据建模为超图。超图中的每个节点代表一个项目, 超边表示多个项目之间的相关性。具体而言, 如果两个项目同时出现在一个会话中, 则它们之间存在一条重复边; 同样, 如果一个会话包含三个或更多项目, 则它们被组成一条超边。其次, 使用超图卷积网络, 由两个通道组成, 即超图卷积和线图卷积。超图卷积通道输入关于超边中节点的信息, 对会话信息和上下文信息进行建模以生成会话表示; 线图卷积通道从超边中的项目节点中提取高阶信息来加强超图的建模。最

后, 对两个卷积交替训练, 直至达到收敛状态。

节 3.1 结合超图卷积和线图卷积的信息, 相互提供互补的视角。这种协同训练可以细化自身, 通过产生自监督信号和发散约束来完成。自监督信号本质上是由正负样本构成的伪标签, 代表了用户可能交互的项目集合。这些伪标签可以作为线图卷积的输入来辅助训练。在进行卷积操作后, 所得到的两组会话嵌入不相同。因为每个卷积操作只描述了会话内或会话间的结构信息, 所以这两组嵌入可以相互补充, 能够学习到两个方向上的互补信息。在进行自监督学习时, 这两组嵌入可以被视为相互对应的 ground-truth。如果两个会话嵌入在两个视图都表示相同的会话, 则被标记为 ground-truth, 否则为 false。

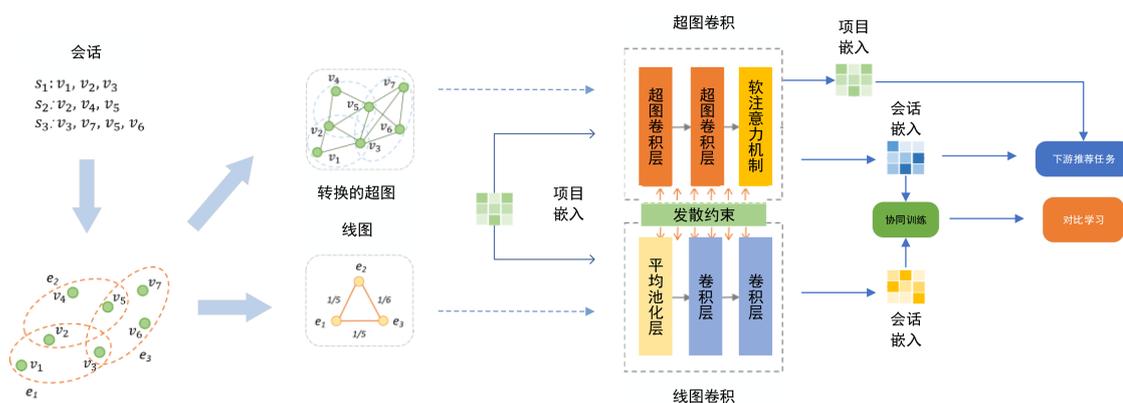


Figure 1. DHCN-COTREC Model structure diagram

图 1. DHCN-COTREC 模型结构图

为了最小化自监督信号, 需要从两个角度进行操作。第一个角度是超图通道, 将会话嵌入与项目嵌入做内积, 得到一个得分。从所有项目中选取排名靠前的  $K$  个项目, 作为正样本; 从推荐的前 10% 项目中除了正样本, 随机采样  $K$  个负样本。得到的正负样本被视为对原数据进行数据增强, 并作为超图卷积的输入来辅助训练。第二个角度是线图通道, 执行的步骤类似于超图通道, 但在计算项目分数时, 只能使用项目的最初形式, 而不是学习到的向量表征形式。

在协同训练流程中, 需要分别考虑两个通道的作用。超图卷积可产生正负样本, 而线图卷积可生成推荐预测的表示。需要注意的是, 正样本应与线图会话嵌入接近, 而负样本则应远离。这里可以使用 InfoNCE 计算距离来实现这一目标:

$$L_{SSL} = L_{SSL}^0 + L_{SSL}^1$$

$$L_{SSL}^1 = -\log \frac{\sum_{i \in c_S^{P^+}} \Psi(X_{(t)}^{last}, \theta_S^P, X_{(t)}^i)}{\sum_{c \in c_S^{P^+}} \Psi(X_{(t)}^{last}, \theta_S^P, X_{(t)}^i) + \sum_{j \in c_S^{P^-}} \Psi(X_{(t)}^{last}, \theta_S^P, X_{(t)}^j)} \quad (6)$$

其中  $I \in \{0, 1\}$ ,  $\theta_S^P$  是从线图学习到的一个会话表示,  $\theta_S^P \cdot X_{(0)}^i$  表示正样本的嵌入表示,  $X_{(0)}^{last} \cdot \theta_S^P$  表示最后一个样本的嵌入。该公式表示需要推荐的项目应该与当前会话中的最后一个项目 (last-item) 最接近。要拉近 last-item 与正样本之间的距离, 并推远与负样本之间的距离。如果最后的项目与预测的正样本更接近, 则表示此次推荐是成功的。

在协同训练中, 使用两个编码器来处理不同的数据视图, 并通过交替训练来提高模型的性能。然而, 如果这两个编码器变得过于相似, 则可能会导致模型出现模式崩溃问题, 即两个编码器生成相同的输出

结果。为了防止经过一定的学习迭代导致两个编码器变得相似并生成相同的数据, 加入“发散约束[11]”来确保两个编码器之间存在一定的差异性。具体而言, 通过引入对抗样本来增加训练数据的多样性, 并鼓励两个编码器在处理相同数据时产生不同的输出结果。这样可以使得模型更加稳健, 并提高其泛化能力和推荐结果的多样性。一共分为四步:

1) 定义 KL 散度约束: 首先, 需要定义 KL 散度约束, 用于衡量两个编码器之间的差异性。KL 散度是一种用于衡量两个概率分布之间差异的指标, 可以用来比较两个编码器生成的概率分布。

2) 引入对抗样本: 为了增加训练数据的多样性, 并鼓励两个编码器在处理相同数据时产生不同的输出结果, 需要引入对抗样本。对抗样本是指通过对原始数据进行微小扰动而生成的新数据, 这些扰动可以使得模型更加鲁棒, 并提高其泛化能力。

3) 计算 KL 散度: 接下来, 需要计算两个编码器生成的概率分布之间的 KL 散度, 并将其作为约束条件加入到模型中。这样可以确保两个编码器之间存在一定的差异性, 并避免出现模式崩溃问题。

4) 优化模型: 最后, 需要通过交替训练来优化模型。在每次迭代中, 先使用一个编码器生成对抗样本, 然后使用另一个编码器来处理这些样本, 并计算其生成的概率分布。接着, 计算两个编码器之间的 KL 散度, 并将其作为约束条件加入到模型中。最终, 通过反向传播算法来更新模型参数, 以使得模型更加稳健, 并提高其泛化能力和推荐结果的多样性。

## 4. 实验与结果分析

本节旨在对本文提出的 DHCN-COTREC 算法进行实验分析。首先介绍使用的数据集、评价指标、实验环境和参数设置, 然后进行和对比算法的实验, 最后对实验结果进行分析。

### 4.1. 实验环境实验数据集与评价指标

Table 1. Statistics of datasets

表 1. 数据集的统计信息

名称	用户数量	商品数量	交互数量	稀疏性
Tmall	963,923	2,353,207	44,528,127	99.99%
DIGINETICA	600,684	184,047	993,483	99.99%

本文的实验环境是 Windows11 专业版和 Python3.9 编译环境。使用 Tmall 和 Diginetica 数据集。如表 1 所示, Tmall 数据集包含用户在天猫平台上的行为记录, 如商品浏览、添加购物车、购买等。Diginetica 数据集包含从电商搜索引擎日志中提取的用户会话信息, 包括匿名用户 ID、查询记录、产品说明和元数据等。

本文将数据集按 8:2 的比例划分成训练集和测试集。数据集中的一段完整会话属性包含: 用户 id、项目 id、会话 id、时间戳。在数据预处理阶段, 构建会话点击和会话时间戳字典, 分别将会话 ID 映射到会话期间用户交互的项目列表和最后一次交互的时间戳。过滤单一项目和会话项数小于 5 的会话, 并统计每个项目在所有会话中出现的次数。根据每个会话中最后一次项目交互的时间戳将数据分成训练集和测试集。测试集由最后 100 秒的交互组成, 其余会话用于训练。最后将会话转换为项目 ID 序列, 并且重新编号为从 ‘1’ 开始的 ID 列表, 用于训练和测试模型。

该实验使用推荐算法中常用的评价指标平均倒数排名(Mean Reciprocal Rank, MRR)来评估算法的性能。MRR 越大表示误差越小, 推荐系统的性能越好。

$$MRR = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (7)$$

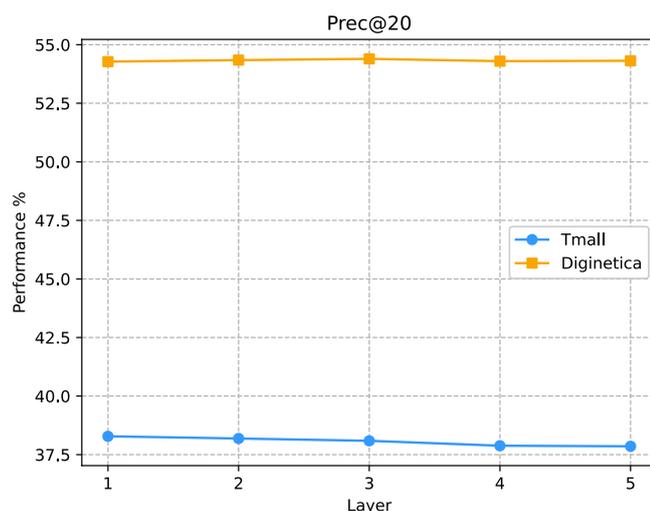
其中,  $|Q|$  是用户个数,  $rank_i$  是第  $i$  个用户。

## 4.2. 实验结果

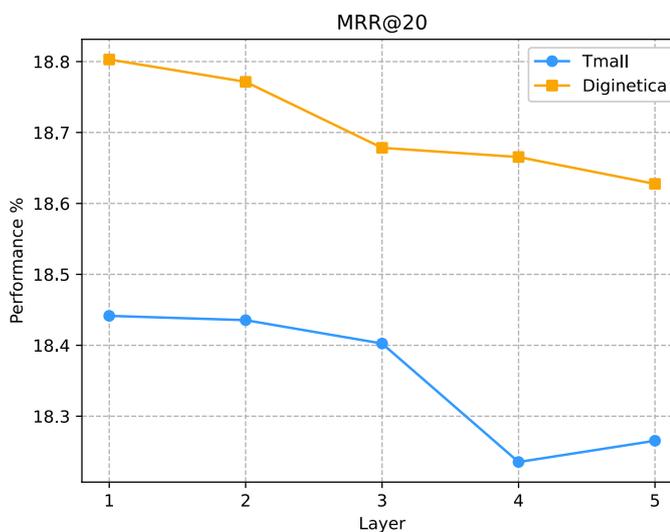
本文实验中, 设置迭代次数为 30; 学习率(lr)为 0.001; L2 系数为  $1e-5$ ; 块大小(batchSize)、嵌入大小(embSize)为 100; 发散约束(lam)为 0.005。

在使用 InfoNCE 计算项目的余弦距离时, 有一个超参数称为负样本区分度( $\tau$ ), 又称为温度系数。如果将温度系数设得过大, 对比损失会对所有的负样本一视同仁, 导致模型没有对不同类型的负样本加以区分, 进而影响模型的学习能力。相反, 如果将温度系数设得过小, 模型会更关注难负样本, 但是这些负样本很可能也是潜在的正样本, 影响模型的收敛和泛化能力。因此, 本文根据经验选择温度系数为 0.2。

为了研究图卷积网络中层数对模型性能的影响, 观察不同网络层数下 Prec@20 和 MRR@20 值的变化。实验中, Layer 层数从 1 层开始, 每次增加 1 层, 直到 5 层结束。实验结果如图 2 所示。



(a)前 20 项精确度



(b)前 20 项平均倒数排名

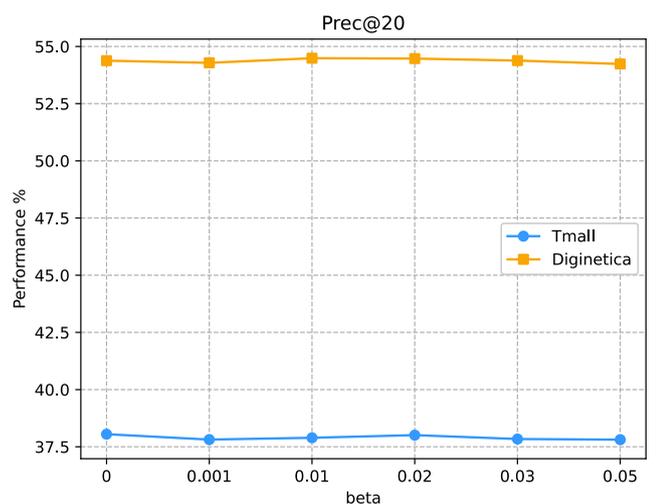
**Figure 2.** The impact of the number of layers in graph convolutional networks on prediction accuracy  
**图 2.** 图卷积网络层数对预测准确度的影响

根据图2的实验结果显示, 随着网络层数的变化, Tmall 数据集的 Prec@20 和 MRR@20 逐渐下降, 表明该数据集使用单层网络的性能最佳。

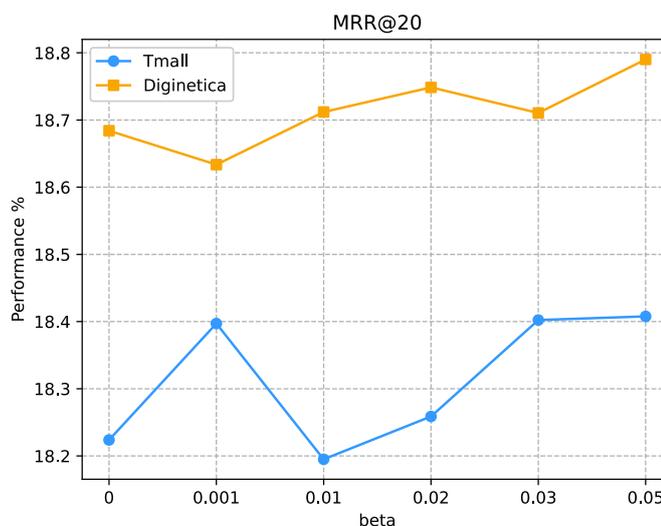
在 Tmall 数据集上, 一层网络可以获得最佳性能可能是因为该数据集中项目之间的关系比较简单, 不需要太多的层数来捕捉这些关系。此外, Tmall 数据集中的会话长度相对较短, 每个会话中包含的项目数量较少。即使只使用一层网络, 也可以保留足够的信息来区分不同项目。

对于 Diginetica 数据集, 该算法对网络层数相对不敏感, 可能是因为该数据集中项目之间的关系比较复杂, 需要更多的层数来捕捉这些关系, 实验表示最佳选择是三层。同时, 随着网络层数的增加, MRR@20 的性能会下降。这可能是因为每一次的卷积操作都会对项目表示进行平均池化等操作, 从而使得项目表示变得过于平滑化, 项目之间的差异逐渐减小, 导致缺少了区分度。

为了验证自我监督学习幅度参数对超图-自监督协同训练算法的效果, 本实验在不同的参数  $\beta$  情况下进行实验对比。  $\beta \in \{0.001, 0.01, 0.02, 0.03, 0.05\}$  在两个不同数据集的实验结果如图3所示。



(a)前 20 项精确度



(b)前 20 项平均倒数排名

**Figure 3.** The effect of self-supervised learning magnitude parameter on prediction accuracy  
**图 3.** 自我监督学习幅度参数对预测准确度的影响

根据图 3 的实验结果显示, 当推荐任务与自我监督任务联合优化时, 推荐任务能够获得较好的收益。对于这两个数据集, 较小的 $\beta$ 值可以提高 Prec@20 和 MRR@20 指标, 随着 $\beta$ 值的增加, 性能逐渐上升。

在 Tmall 和 Diginetica 数据集上, 实验结果显示最好的  $\beta$  值设置为 0.02。增加自监督参数的大小可以使模型更好地学习到项目之间的相似性关系, 这是因为自监督学习任务可以提供额外的训练信号, 帮助模型更好地学习到项目之间的相似性。 $\beta$ 值越大, 模型越注重自我监督任务, 而忽略了推荐任务本身。这可能会导致过拟合和性能下降。相反, 如果 $\beta$ 值太小, 则模型可能无法从自我监督任务中获得足够的信息来提高推荐性能, 从而导致欠拟合和性能下降。

### 4.3. 比较算法

**Table 2.** Performances of all comparison methods on two datasets

**表 2.** 比较方法在两个数据集上的性能

算法	Tmall				Diginetica			
	Prec@10	MRR@10	Prec@20	MRR@20	Prec@10	MRR@10	Prec@20	MRR@20
Item-KNN	6.65	3.11	9.15	3.31	25.07	10.77	35.75	11.57
FPMC	13.10	7.12	16.06	7.32	15.43	6.20	26.53	6.95
GRU4REC	9.47	5.78	10.93	5.89	17.93	7.33	29.45	8.33
NARM	19.17	10.42	23.30	10.70	35.44	15.13	49.70	16.17
STAMP	22.63	13.12	26.47	13.36	33.98	14.26	45.64	14.32
SR-GNN	23.41	13.45	27.57	13.72	36.86	15.52	50.73	17.59
GCE-GNN	28.01	15.08	33.42	15.42	41.16	18.15	54.22	19.04
FGNN	20.67	10.07	25.24	10.39	37.72	15.95	50.58	16.84
DHCN	25.14	13.91	30.43	14.26	39.87	17.53	53.18	18.44
DHCN-COTREC	31.41	17.81	38.01	18.26	40.83	17.76	54.49	18.71
提升(%)	24.94	28.04	24.91	28.05	2.48	1.31	2.46	1.46

为了验证本文算法在提升预测准确度方面的有效性, 将本文算法与其他已有算法进行了对比实验。超参数设置为嵌入大小(embSize)为 100, 块大小(batchSize)为 100, L2 正则化为  $1e-5$ , 初始学习率 lr 为 0.001。

总体性能实验结果如表 2 所示, 用粗体突出本文算法。使用 DHCN-COTREC 的性能与最佳基准性能 DHCN 之间的差异计算提升效率, 分析表中结果得出以下结论。

在标准为 Prec@10、MRR@10、Prec@20 和 MRR@20 情况下, 对比了 Item-KNN [15]、FPMC [16]、GRU4REC、NARM [17]、STAMP [18]、SR-GNN、GCE-GNN [19]、FGNN [20]、DHCN, 7 种算法。

**Item-KNN:** 推荐与会话中先前点击项目相似的项目, 使用会话向量之间的余弦相似度计算相似性。

**FPMC:** 一种基于马尔可夫链的序列方法, 可以同时捕捉时间信息和长期的用户喜好信息。

**GRU4REC:** 利用小批量会话并行训练, 并采用基于排序的损失函数对用户序列进行建模。

**NARM:** 基于 RNN 的模型, 对顺序行为进行建模以产生推荐。

**STAMP:** 利用自我注意机制增强基于会话的推荐。

**SR-GNN:** 使用门控图卷积层来学习项目转换。

**GCE-GNN:** 基于会话推荐的全局上下文增强图神经网络, 利用图卷积和自注意机制来捕捉项目之间

的关系, 生成推荐结果。

**FGNN:** 全图神经网络模型, 用于学习项目转换模式的固有顺序, 并计算会话级表示以生成推荐。

对比实验结果如表二所示。实验结果表明 DHCN-COTREC 在所有数据集上都显示出了较大的优势。与 SR-GNN、GCE-GNN 和 FGNN 相比, 本文算法有两个优点:

1) 本文算法采用: (1) 超图的构建。使用超图来捕获项目间的相互关系。将每一个超边建模为一个集合, 其中包含完全相互连接的项目, 可以更好地利用远距离项目之间的联系, 从而提升模型的性能。(2) 时间因素的加入。将物品序列作为输入, 使用基于超图的卷积神经网络对其进行建模, 得到物品的表示向量。同时, 将时间序列作为输入, 得到时间上下文信息的表示向量。最后, 将两个向量拼接起来作为会话的表示向量, 并通过全连接层进行推荐结果预测。同时考虑了物品之间的相似性和时间上下文信息。

2) 采用自监督协同训练策略: (1) 可以生成更准确和鲁棒的自监督信号(伪标签)并利用会话中项目之间的关系来帮助模型更好地理解项目之间的交互。(2) 能缓解由于节点数量增加带来的数据稀疏性问题, 有效地利用其它项目中的信息来提高下一个项目预测准确度, 提高推荐精度。

## 5. 结论

本文算法的核心是利用超图进行数据建模, 以充分考虑项目之间的关系, 并采用自监督协同训练算法缓解了会话推荐领域的数据稀疏性问题。通过将会话建模为“序列-图”模型, 利用自监督学习策略扩充数据, 并通过协同训练实现不同视图下的特征表示。实验结果表明, 自监督协同训练能够利用不同的连接信息递归生成 **ground-truth**, 且保留了完整的会话信息, 从而提高了推荐效果。然而, 本文算法目前仍存在一些缺点, 例如迭代速度较慢、容易过拟合以及涉及的参数较多, 需要大量的计算资源和运行时间。为了解决这些问题, 我们需要做更深入的研究。

## 参考文献

- [1] Patra, B.K., Launonen, R., Ollikainen, V., *et al.* (2014) Exploiting Bhattacharyya Similarity Measure to Diminish User Cold-Start Problem in Sparse Data. *Discovery Science: 17th International Conference*, DS 2014, Bled, 8-10 October 2014, 252-263. [https://doi.org/10.1007/978-3-319-11812-3\\_22](https://doi.org/10.1007/978-3-319-11812-3_22)
- [2] Thakkar, P., Varma, K., Ukani, V., *et al.* (2019) Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning. In: Satapathy, S. and Joshi, A., (Eds.), *Information and Communication Technology for Intelligent Systems*, Springer, Singapore, 173-180. [https://doi.org/10.1007/978-981-13-1747-7\\_17](https://doi.org/10.1007/978-981-13-1747-7_17)
- [3] Hidasi, B., Karatzoglou, A., Baltrunas, L., *et al.* (2015) Session-Based Recommendations with Recurrent Neural Networks. arXiv: 1511.06939.
- [4] Wu, S., Tang, Y., Zhu, Y., *et al.* (2019) Session-Based Recommendation with Graph Neural Networks. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, Honolulu, 27 January 2019 - 1 February 2019, 346-353. <https://doi.org/10.1609/aaai.v33i01.3301346>
- [5] Feng, Y., You, H., Zhang, Z., *et al.* (2019) Hypergraph Neural Networks. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, Honolulu, 27 January 2019 - 1 February 2019, 3558-3565. <https://doi.org/10.1609/aaai.v33i01.33013558>
- [6] Yadati, N., Nimishakavi, M., Yadav, P., *et al.* (2019) Hypergcn: A New Method for Training Graph Convolutional Networks on Hypergraphs. *Advances in Neural Information Processing Systems*, **32**, 1511-1522. [https://proceedings.neurips.cc/paper\\_files/paper/2019/hash/1efa39bcaec6f3900149160693694536-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2019/hash/1efa39bcaec6f3900149160693694536-Abstract.html)
- [7] Jiang, J., Wei, Y., Feng, Y., *et al.* (2019) Dynamic Hypergraph Neural Networks. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, 10-16 August 2019, 2635-2641. <https://doi.org/10.24963/ijcai.2019/366>
- [8] Bandyopadhyay, S., Das, K. and Murty, M.N. (2020) Line Hypergraph Convolution Network: Applying Graph Convolution for Hypergraphs. arXiv:2002.03392.
- [9] Bu, J., Tan, S., Chen, C., *et al.* (2010) Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content. *Proceedings of the 18th ACM International Conference on Multimedia*, Florence, 25-29 October 2010, 391-400. <https://doi.org/10.1145/1873951.1874005>

- 
- [10] Li, L. and Li, T. (2013) News Recommendation via Hypergraph Learning: Encapsulation of User Behavior and News Content. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, Rome, 4-8 February 2013, 305-314. <https://doi.org/10.1145/2433396.2433436>
- [11] Wang, J., Ding, K., Hong, L., *et al.* (2020) Next-Item Recommendation with Sequential Hypergraphs. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 25-30 July 2020, 1101-1110. <https://doi.org/10.1145/3397271.3401133>
- [12] Xia, X., Yin, H., Yu, J., *et al.* (2021) Self-Supervised Hypergraph Convolutional Networks for Session-Based Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 4503-4511. <https://doi.org/10.1609/aaai.v35i5.16578>
- [13] Wu, F., Souza, A., Zhang, T., *et al.* (2019) Simplifying Graph Convolutional Networks. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, 9-15 June 2019, 6861-6871.
- [14] Vaswani, A., Shazeer, N., Parmar, N., *et al.* (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, **30**, 5998-6008.
- [15] Sarwar, B., Karypis, G., Konstan, J., *et al.* (2001) Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, 1-5 May 2001, 285-295. <https://doi.org/10.1145/371920.372071>
- [16] Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L. (2010) Factorizing Personalized Markov Chains for Next-Basket Recommendation. *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, 26-30 April 2010, 811-820. <https://doi.org/10.1145/1772690.1772773>
- [17] Li, J., Ren, P., Chen, Z., *et al.* (2017) Neural Attentive Session-Based Recommendation. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 6-10 November 2017, 1419-1428. <https://doi.org/10.1145/3132847.3132926>
- [18] Liu, Q., Zeng, Y., Mokhosi, R., *et al.* (2018) STAMP: Short-Term Attention/Memory Priority Model for Session-Based Recommendation. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, 20-23 August 2018, 1831-1839. <https://doi.org/10.1145/3219819.3219950>
- [19] Wang, Z., Wei, W., Cong, G., *et al.* (2020) Global Context Enhanced Graph Neural Networks for Session-Based Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 25-30 July 2020, 169-178. <https://doi.org/10.1145/3397271.3401142>
- [20] Qiu, R., Li, J., Huang, Z., *et al.* (2019) Rethinking the Item Order in Session-Based Recommendation with Graph Neural Networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, 3-7 November 2019, 579-588. <https://doi.org/10.1145/3357384.3358010>