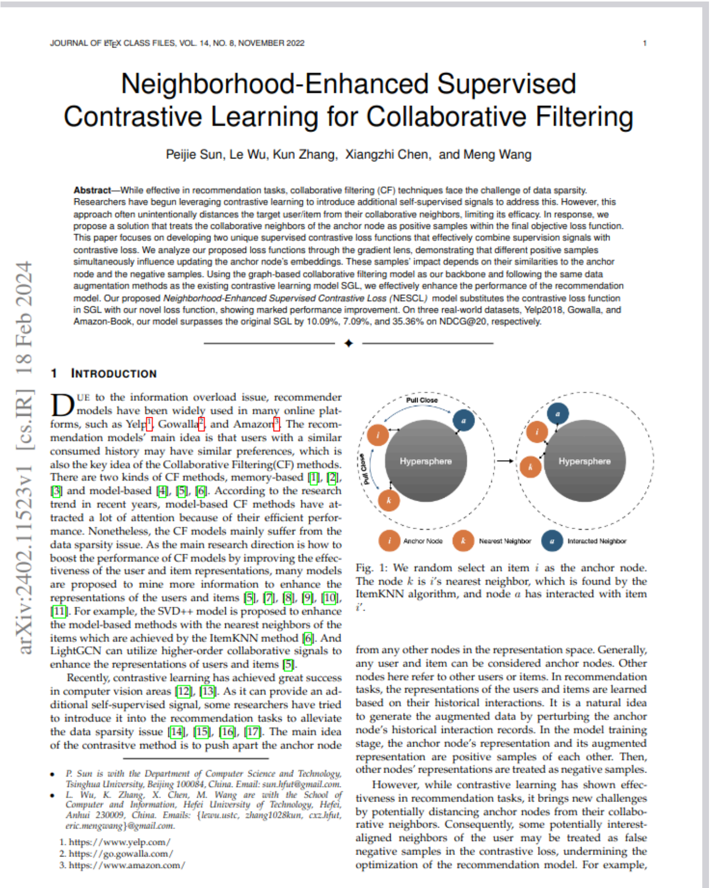
基于邻域增强的监督对比学习推荐方法

汇报人

谭盟

2024 年 12 月

基于邻域增强的监督对比学习推荐方法



Neighborhood-Enhanced Supervised Contrastive Learning for Collaborative Filtering

Peijie Sun, Le Wu, Kun Zhang, Xiangzhi Chen, and Meng Wang

Abstract—While effective in recommendation tasks, collaborative filtering (CF) techniques face the challenge of data sparsity. Researchers have begun leveraging contrastive learning to introduce additional self-supervised signals to address this. However, this approach often unintentionally distances the target user/item from their collaborative neighbors, limiting its efficacy. In response, we propose a solution that treats the collaborative neighbors of the anchor node as positive samples within the final objective loss function. This paper focuses on developing two unique supervised contrastive loss functions that effectively combine supervision signals with contrastive loss. We analyze our proposed loss functions through the gradient lens, demonstrating that different positive samples simultaneously influence updating the anchor node's embeddings. These samples' impact depends on their similarities to the anchor node and the negative samples. Using the graph-based collaborative filtering model as our backbone and following the same data augmentation methods as the existing contrastive learning model SGL, we effectively enhance the performance of the recommendation model. Our proposed *Neighborhood-Enhanced Supervised Contrastive Loss* (NESCL) model substitutes the contrastive loss function in SGL with our novel loss function, showing marked performance improvement. On three real-world datasets, Yelp2018, Gowalla, and Amazon-Book, our model surpasses the original SGL by 10.09%, 7.09%, and 35.36% on NDCG@20, respectively.

1 Introduction

Due to the information overload issue, recommender models have been widely used in many online platforms, such as Yelp[1], Gowalla[2], and Amazon[3]. The recommendation models' main idea is that users with a similar consumed history may have similar preferences, which is also the key idea of the Collaborative Filtering(CF) methods. There are two kinds of CF methods, memory-based [1], [2], [3] and model-based [4], [5], [6]. According to the research trend in recent years, model-based CF methods have attracted a lot of attention because of their efficient performance. Nonetheless, the CF models mainly suffer from the data sparsity issue. As the main research direction is how to boost the performance of CF models by improving the effectiveness of the user and item representations, many models are proposed to mine more information to enhance the representations of the users and items [5], [7], [8], [9], [10], [11]. For example, the SVD++ model is proposed to enhance the model-based methods with the nearest neighbors of the items which are achieved by the ItemKNN method [6]. And LightGCN can utilize higher-order collaborative signals to enhance the representations of users and items [5].

Recently, contrastive learning has achieved great success in computer vision areas [12], [13]. As it can provide an additional self-supervised signal, some researchers have tried to introduce it into the recommendation tasks to alleviate the data sparsity issue [14], [15], [16], [17]. The main idea of the contrasitve method is to push apart the anchor node from any other nodes in the representation space. Generally, any user and item can be considered anchor nodes. Other nodes here refer to other users or items. In recommendation tasks, the representations of the users and items are learned based on their historical interactions. It is a natural idea to generate the augmented data by perturbing the anchor node's historical interaction records. In the model training stage, the anchor node's representation and its augmented representation are positive samples of each other. Then, other nodes' representations are treated as negative samples.

Fig. 1: We random select an item i as the anchor node. The node k is i's nearest neighbor, which is found by the ItemKNN algorithm, and node a has interacted with item i'.

However, while contrastive learning has shown effectiveness in recommendation tasks, it brings new challenges by potentially distancing anchor nodes from their collaborative neighbors. Consequently, some potentially interest-aligned neighbors of the user may be treated as false negative samples in the contrastive loss, undermining the optimization of the recommendation model. For example,

- *P. Sun is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. Email: sun.hfut@gmail.com.*
- *L. Wu, K. Zhang, X. Chen, M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China. Emails: {lewu.ustc, zhang1028kun, cxz.hfut, eric.mengwang}@gmail.com.*

1. https://www.yelp.com/
2. https://go.gowalla.com/
3. https://www.amazon.com/

- 提出了两种新的监督对比学习损失函数(NESCL),分别是“in”版本和“out”版本,用于推荐系统中的协同过滤任务。
- 在三个真实数据集(Yelp2018、Gowalla和Amazon-Book)上进行了大量实验,验证了所提方法的有效性。相比现有最好的对比学习方法SGL,在NDCG20指标上分别提升了10.09%、7.09%和35.36%。
- 进行了详细的理论分析,从梯度的角度分析了不同正样本对锚节点表示学习的影响。

GCN 和推荐系统

- GCN (Graph Convolutional Network) 是一种处理图数据的神经网络模型
- 类似于 CNN 在图像上的卷积操作，GCN 在图结构数据上进行信息传递
- 每个节点会聚合其邻居节点的信息来更新自己的表示
- 通过多层传递可以捕获高阶的图结构信息

LightGCN

- 在论文中使用的 LightGCN 是一种简化的 GCN 模型
- 去掉了特征变换和非线性激活
- 只保留了邻居信息聚合操作
- 在推荐任务上表现更好

LightGCN

$h_i^{l+1} = \sum\left(\frac{1}{\sqrt{N_i * N_j}} * h_j^l\right)$

- h_i^{l+1} 是节点 i 在第 l+1 层的表示
- h_j^l 是节点 i 的邻居节点 j 在第 l 层的表示
- N_i 是节点 i 的邻居数量(度数)
- N_j 是节点 j 的邻居数量(度数)
- $\sum$ 表示对所有邻居节点 j 求和

在推荐系统中,用户和物品的表示主要依赖于它们之间的交互关系,而不是节点的特征。

实验——数据集

- Yelp2018:用户对商家的评价数据
- Gowalla:基于位置的社交网络数据
- Amazon-Book:亚马逊图书评价数据

实验——评估指标

- Recall@K

关注推荐系统找到相关物品的能力,但不考虑物品的排序位置

- NDCG@K

考虑推荐物品的位置顺序的一个指标,排在前面的正确推荐会得到更高的权重

- K 设置为{20}

表示只评估推荐列表的前 20 个物品，因为在实际应用中,用户通常只会关注推荐列表的前几项

- 基线模型:LightGCN
- 主要对比方法:SGL(当前最好的对比学习推荐方法)

实验结果

- 在三个数据集上均优于 SGL
- NDCG@20 提升显著:
- Yelp2018: +10.09%
- Gowalla: +7.09%
- Amazon-Book: +35.36%

消融实验

- 不同温度系数 τ 的影响:
  - 发现较小的 τ 值(0.1)效果最好
  - 说明引入多个正样本可以抵消假负样本的影响
- 不同正样本选择策略的对比:
  - Identity Weights:将相似度设为 1
  - Similarity Weights:使用基于记忆的方法计算相似度
  - Random Sampling:随机采样一个最近邻
  - Weighted Sampling:根据相似度加权采样

超参数分析

- GNN 层数的影响(1-3 层)
- 数据增强策略的选择(Node Dropout/Edge Dropout/Random Walk)
- 数据增强比例 ρ 的影响(0.1-0.9)
- 正则化系数 α 的影响(0.1-0.5)

结论

- 较小的温度系数效果更好
- Random Sampling 策略表现最佳
- 不同数据集最优超参数设置:
  - Yelp2018:2 层 GNN,Node Dropout,ρ=0.3,α=0.3
  - Gowalla:3 层 GNN,Node Dropout,ρ=0.3,α=0.1
  - Amazon-Book:2 层 GNN,Edge Dropout,ρ=0.3,α=0.3
- 代码基于 RecBole 框架实现，开源。

主要创新点

- 首次提出将协同邻居作为正样本纳入对比学习的损失函数中,有效解决了现有对比学习方法可能会不当地拉远锚节点与其协同邻居距离的问题。
- 通过理论分析证明了所提损失函数能够根据正样本与锚节点和负样本的相似度,自适应地调整不同正样本的权重。
- 发现在较小的温度系数下模型表现更好,这说明通过引入多个正样本可以抵消假负样本的不利影响,同时放大真负样本的有益影响。
- 提出了多种策略来选择和整合不同类型的正样本,增强了模型的鲁棒性和性能。

附录

1	NESCL	0.1917	0.1617	Neighborhood-Enhanced Supervised Contrastive Learning for Collaborative Filtering	2024
2	BSPM-EM	0.192	0.1597	Blurring-Sharpening Process Models for Collaborative Filtering	2022
3	BSPM-LM	0.1901	0.157	Blurring-Sharpening Process Models for Collaborative Filtering	2022
4	LT-OCF	0.1875	0.1574	LT-OCF: Learnable-Time ODE-based Collaborative Filtering	2021
5	SimpleX	0.1872	0.1557	SimpleX: A Simple and Strong Baseline for Collaborative Filtering	2021

6	UltraGCN	0.1862	0.158	UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation	2021
7	Emb-GCN	0.1862	-	UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation	2021
8	GF-CF	0.1849	0.1518	How Powerful is Graph Convolution for Recommendation?	2021
9	LightGCN	0.183	0.1554	LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation	2020
10	NGCF	0.157	-	Neural Graph Collaborative Filtering	2019

Thank You!

The End

Typst vs. LaTeX Beamer

- 分析了不同正样本对锚节点表示学习的影响
- 分析了不同超参数(温度系数、GNN 层数、数据增强策略等)的影响
- 验证了不同邻居选择策略的效果

Typst vs. LaTeX Beamer

- 提出了一种新的思路来解决对比学习中的问题:

Typst vs. LaTeX Beamer

- 提出了一种新的思路来解决对比学习中的问题:
- 将协作邻居作为正样本加入损失函数
- 有效缓解了传统对比学习可能会将相似节点推远的问题

Typst vs. LaTeX Beamer

- 设计了两种新颖的监督对比损失函数:
- 能够自适应地调整不同正样本的权重
- 在理论上证明了其有效性

Customization

- Change colors using RGB values or predefined color names
- Modify fonts and sizes using the `set text()` function
- Create custom layouts with grids and columns
- Use functions for repeated elements or styles

Adding Images

Adding images in Typst:

```
#image("path/to/image.png", width:
80%)
```

Columns and Grids

This is the first column

- Point 1
- Point 2

And this is the second column

- Point A
- Point B

Special Slides

- Create custom slide layouts for chapter slides or side-picture slides
- Use functions to generate these special slides
- Example of a chapter slide:

```
#let chapter-slide(title, image) = {
  set page(fill: rgb("#0047BA"))
  align(center + horizon)[
    #text(fill: white, size: 40pt, weight: "bold")[#title]
  ]
  place(bottom + right, image(image, width: 30%))
}
```

Typography

- Use readable fonts:
  - Sans-serif fonts are generally preferred for presentations
  - Serif fonts can be used with high-definition projectors
- Avoid using:
  - Monospace fonts for normal text
  - Decorative or hard-to-read fonts
- Maintain consistent font usage throughout the presentation

Best Practices

- Keep slides simple and focused
- Use consistent styling throughout the presentation
- Limit the amount of text on each slide
- Use visuals to support your message
- Practice your presentation to ensure good timing and flow

Summary

- Typst offers a powerful alternative to LaTeX for creating presentations
- Easy to learn and use, with good customization options
- Produces high-quality PDF output
- Supports math typesetting and complex layouts
- Remember to focus on content and clear communication

Thank You!

Any questions?