ICDM 2022：大规模电商图上的风险商品检测——技术报告

目录

1 赛题理解

在电商平台上的风险商品检测场景中，黑灰产和风控系统之间存在着激烈的对抗，黑灰产为了躲避平台管控，会蓄意掩饰风险信息，恶意用户会通过伪造设备、伪造地址等方式，伪造较为“干净”的关联关系。我们认为本赛题主要的目的通过充分挖掘这种存在大量噪声的图结构数据，来发现“黑样本”产品。以下是我们对赛题的分析和数据的探索。

1.1 赛题分析

主办方给定了部分带有 0 和 1 标签的 item 类型节点数据（1 表示节点异常，0 表示正常），并且给定了所有节点的 256 维度的特征信息，通过设计合理的模型对这部分数据的特征信息进行学习，去推理其他未知标签的 item 节点，本质是一个二分类任务。

比赛的图网络中有 7 种类型节点，是一个异构图网络，因此我们可以优先考虑传统的处理异构图的模型如 HAN，HGT 等，与主办方给出 baseline 的 RGCN 模型做实验对比。而再好的模型在实际处理数据时都容易出现过拟合现象，特别模型越复杂过拟合情况越严重，一般我们会想到设计某种无监督的方式去对模型先进行预训练，让模型能够充分学习到训练集和测试集的特征分布，再进行下游 finetune，这样可以极好地避免过拟合现象。

1.2 数据分析

初赛和复赛图的统计信息

表 1: 统计信息

阶段	节点类型	边类型	节点总数	边总数
初赛	7	7	13,806,619	157,814,864
复赛	7	7	10,284,026	120,691,444

初赛和复赛图的黑白样本比例

表 2: 黑白样本比例

阶段	白样本数	黑样本数	比例
初赛	77,198	8,364	约 9:1
复赛	?	?	?

通过可视化分析初赛和复赛图数据发现，赛题所给出的两个图的节点数量分布几乎一致。

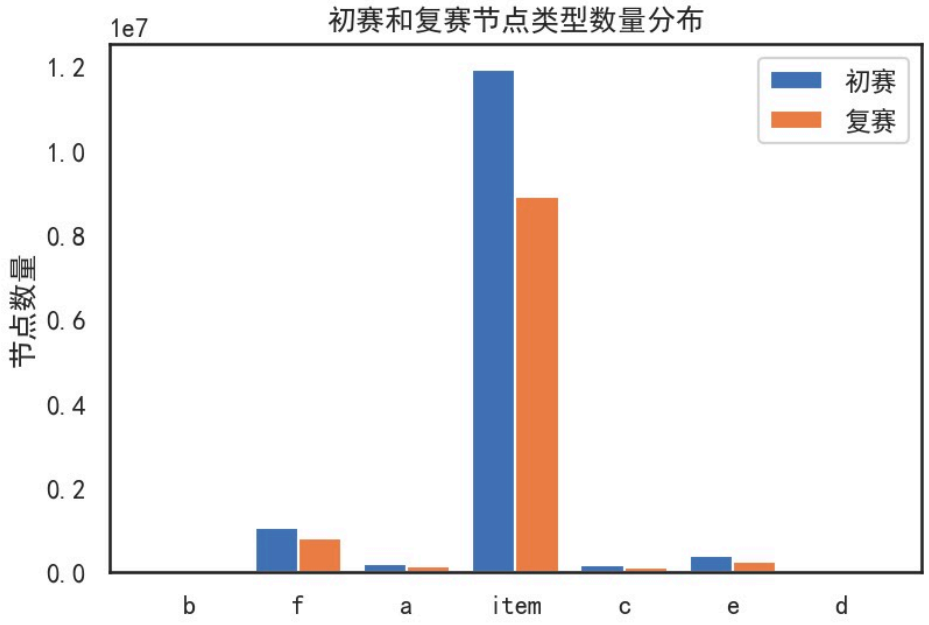

图 1.1: 各类节点分布

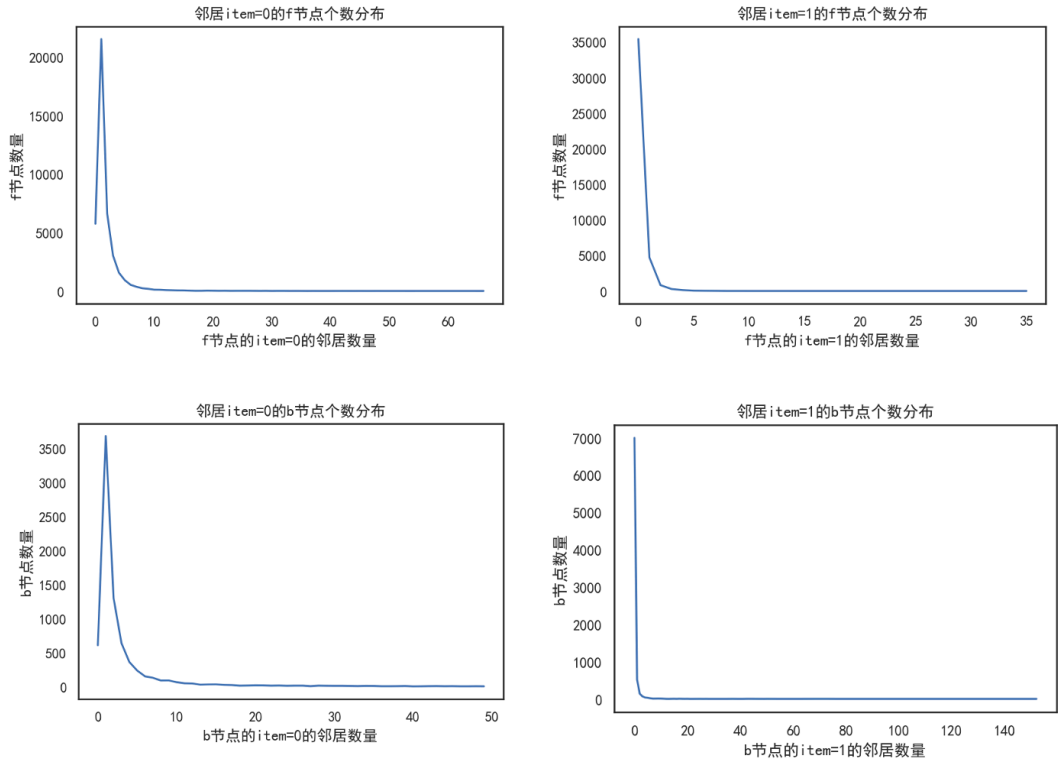

图 1.2: b 和 f 节点的 item 标签分布

2 模型方案

通过对赛题的分析，本节将介绍我们的模型方案，主要包括：(1). 先进行简单实验快速确定一个综合效果不错的模型，这个模型能够优于 baseline 的 RGCN 模型；(2). 将选择好的模型进行层数堆叠使得模型复杂化得到一个大模型，并设计合理的无监督训练方式将该复杂模型进行预训练；(3). 将预训练好的模型结合赛题给定的 0 和 1 标签的数据进行下游 finetune，并进行测试集推理，此时预期已经能得到一个很不错的效果；(4). 设计合理的数据增强方式，继续提升模型的性能。

下面将详细介绍我们的模型设计，初赛方案和复赛方案，复赛方案是在初赛的基础上进一步深入和完善的。

2.1 模型设计

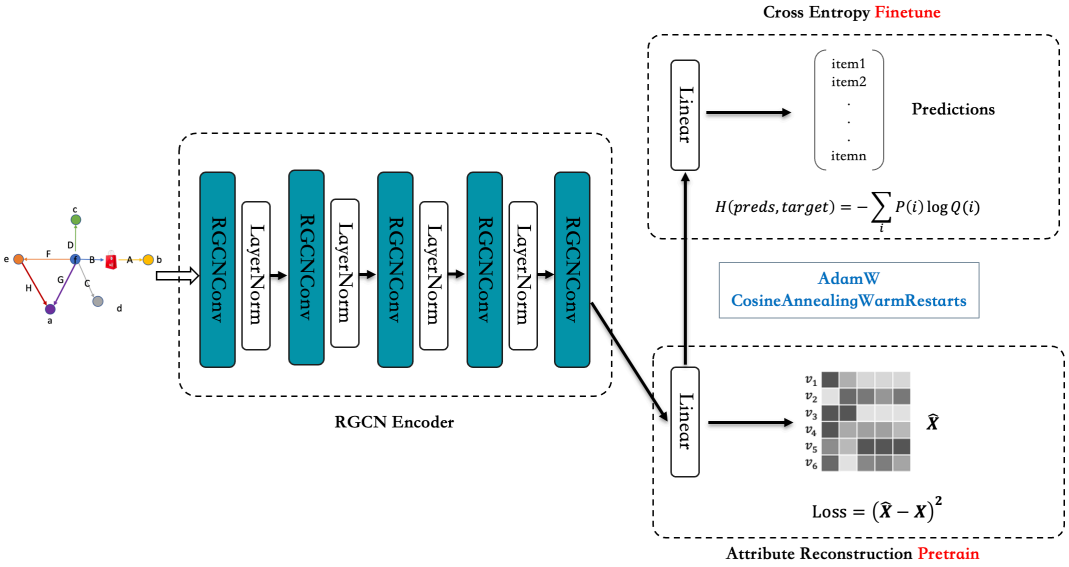

图 2.1: 模型框架

我们的模型主要由三部分组成，包括 RGCN Encoder，预训练部分和微调部分。

RGCN Encoder

RGCN Encoder 的设计我们是从 Bert[1] 预训练模型中获得灵感，在保证基础模型有一个好的效果的前提下，将基础模型进行层数堆叠构建大模型进行预训练，因此我们堆叠了 5 层 RGCNConv[5] 模块，为了避免层数太深导致梯度爆炸，我们在每层 RGCNConv 之间用 LayerNormalization 进行数据的标准化处理。编码器网络可以用公式表示如下：

$$H^l = RGCNConv(H^{l-1}, edge\_index, edge\_type) \tag{1}$$

$$Z^l = LayerNorm(H^l) \tag{2}$$

$$Z = Dropout(\alpha(Z^l)) \tag{3}$$

公式 3 中的 α 我们用的是 relu 激活函数。

预训练部分

模型预训练部分主要是用一个 MLP 将 RGCN 编码器得到的节点 embedding 映射到和原始节点属性相同纬度，从而实现节点特征重构。公式表示如下：

$$\hat{X} = MLP(Z) \tag{4}$$

预训练部分使用的节点特征重构损失函数为平方误差损失

$$Loss_{rec} = (\hat{X} - X)^2 \tag{5}$$

微调部分

模型微调训练部分同样使用一个 MLP 将模型预训练的输出接一个 MLP，将节点 embedding 映射到 2 维，得到每个节点属于 0 或 1 的概率。公式表示如下

$$preds = MLP(\hat{X}) \tag{6}$$

微调部分使用交叉熵损失函数，对预训练的模型进一步微调，让模型能够更好的挖掘图网络中的异常节点。

$$H(preds, target) = -\sum_i P(i) \log Q(i) \tag{7}$$

模型具体参数和训练细节我们将在3.1小节进行详细介绍。

2.2 初赛方案

1. 搭建 5 层 RGCN 加载初赛训练集和测试集数据，以特征重构的方式进行无监督预训练，训练了 500 轮。

2. 使用预训练了 500 轮的模型进行下游 finetune，发现下游训练到 8 轮左右验证集达到最优往后开始过拟合, 进一步采用不划分验证集，用所有带标签数据进行全量训练 7 轮，初赛线上分数达到 94.4（但存在一定波动）。

2.3 复赛方案

1. 初赛结束做进一步数据分析发现和 item 直接连接的 f 和 b 节点存在一定规律(分析原因见)，因此我们认为如果 f 和 b 连接的所有 item 中只要有一个异常，那么就认为该 f 和 b 节点异常，赋予标签为 1，否则认为 f 和 b 节点正常，赋标签为 0。

2. 因为 RGCN 模型认为图中所有节点都是同一类型，只是边类型不同，因此赋予 b 和 f 节点标签后引入和 item 节点一起训练，相当于扩充了训练集数量，发现带来了极高的正向收益。

3. 在 500 轮基础上加入了初赛的 b 和 f 节点预训练 300 轮，然后带入赋予标签后的 b 和 f 和 item 一起下游全量训练 7 轮。

4. 继续引入了复赛的测试集 item、b 和 f 节点同初赛节点一起预训练然后下游 fine-tune。

2.4 其他方案尝试

此外，我们还尝试了大量其他方案，但是均未取得较显著的提升，因此最终选择放弃。

1. 节点特征扩充策略，使用 node2vec 给 item 节点扩充 128 维特征，由于总特征变多，模型训练非常慢，而且服务器资源使用过大，最终效果没有明显提升，最后放弃。

2. 在原始 256 维节点特征基础上添加，给 item 节点添加节点度特征，以及周围邻居节点度特征的统计值 (max，min，std，mean)，效果也不好。

3. 使用 DGLD[6] 开源库中纯无监督模型进行预训练，比如 CoLA[4] 和 Dominant[2]，效果无提升。

4. 设计对比学习模型，参考 MVGRL[3] 和 GRACE[7] 模型，引入数据增强策略进行对比学习自监督预训练，最后效果也没提升。

3 实验部分

本节主要介绍我们详细的实验设置，模型训练参数，以及针对性的对比消融实验。

3.1 实验设置

本小节主要介绍我们复赛最终方案的实验设置。(1) 模型参数设置。我们采用 5 层 RGCNConv，每层的神经元数目为 768，预训练模型的 MLP 输出纬度为 256，和赛题中所给的节点特征纬度保证一致；我们在每层 RGCN 模型之间使用 Dropout 的概率为 0.4。(2) 训练设置。由于赛题所给的图数据非常庞大，因为我们使用 batch 化训练，batch-size 为 256，使用邻居采样器，固定采样邻居节点层数为 2，每层采样邻居节点

数目为 300 个，将采样后的异构子图，转为同构子图，输入模型进行训练；我们一共进行了 3 次预训练和 1 次全量微调训练，第 1 次预训练，初赛阶段的预训练模型，仅仅只训练了带标签的 item 类型节点 500 轮；第 2 次预训练，与带标签的 item 相连的 f 和 b 节点带入预训练了 300 轮；第 3 次预训练，加载复赛的 item、b 和 f 节点一起预训练了 85 轮；最后进行全量微调训练 7 轮。

3.2 消融实验

3.2.1 简单模型实验结果

因为是异构图网络，我们优先考虑传统的 HGT 模型，以及针对 baseline 的 RGCN 模型添加注意力的 RGAT 模型一起进行实验对比。考虑到主办方提示的数据噪声较大，节点特征的同质性不强（即不同类型节点之间的特征信息区分度不高），因此我们也引入了针对同构图的 GAT、TransformerConv 等模型。因为没有进行预训练模型容易出现波动，因此我们平均每种模型统一训练 100 轮，固定 batch-size=1024，取验证集分数最高的结果去推理测试集，每种模型实验 3 次，实验结果如表3所示。

表 3: 简单模型实验结果

模型	每层采样数目	隐层大小	初赛验证集	初赛测试集
RGCN	300	768	0.9352	0.9253
RGAT	64	128	0.9273	0.9151
HGT	300	768	0.8991	0.8703
GAT	300	768	0.9293	0.9182
TransformerConv	300	768	0.9155	0.8812

实验结果超出我们的预期，因为节点特征的同质性很弱，利用专门的异构图模型 HGT 处理，把每种类型节点分散到不同的特征空间，影响到了所有节点之间的信息交互，而 RGCNConv 模型则是把网络中所有节点看成同一类型，只是边的类型不同进行区分，很好的将所有类型节点映射到同一个空间中，实现了更好的信息交互，同时用边的不同间接区分了节点的不同，而 RGAT 模型因为引入了注意力，会对所有的边都进行注意力计算，导致不能采样过多的节点，不然容易超显存，隐层大小也受到了影响，受这两个条件影响导致总体效果略差于 RGCNConv。

同时，因为我们实验室的图异常检测开源库 DGLD[6] 是基于 dgl 的，所以在比赛初期我们打算使用基于 dgl 的 baseline，我们做了以下实验，如表4。

后来我们在 PyG 框架的基础上引入预训练模型后，线上分数突破 0.94+，但是使用 dgl 并复现同样的预训练模型，却达不到这个分数，因此，我们最终考虑使用基于 PyG 的 RGCNConv 作为基础模型搭建复杂的大模型。

表 4: 基于 dgl 的模型实验结果

方案	初赛验证集	初赛测试集
使用 2 层 RelGraphConv	0.928	0.92263
使用 2 层 RelGraphConv，增加神经元个数	0.93281	0.923319
使用 3 层 RelGraphConv，增加神经元个数	0.938824	0.925188
2 到 3 层之间加类似残差连接，并添加 BatchNorm	0.94065	0.929349

3.2.2 预训练模型有效性分析

在此次比赛中我们已知节点的特征信息，因此从模型的实现角度，最简单的方法就是“节点属性重构法”，“节点属性重构法”的想法更多来源于无监督的图神经网络异常检测任务中，一般认为，在图网络中通过无监督训练进行节点重构，越异常的节点其模型输出的节点表征和原始的节点特征误差越大，我们也尝试用无监督的方式去处理本次比赛的数据，但效果较差，于是我们借鉴这种用法去做模型的预训练，从而更好的提取节点表征。

我们将搭建好的 5 层 RGCN 模型设置了两个输出接口，一个输出接口映射到 256 维，用于和节点的特征进行损失计算，另外一个接口映射到 2 维，用于预训练结束后的下游 finetune。随着不断预训练，我们记录了不同预训练轮数下进行下游 finetune 的结果，如表5所示。

表 5: 预训练模型下游 finetune 结果

预训练轮数	微调最优轮数	初赛验证集	初赛测试集
200	9	0.9432	0.9357
300	8	0.9443	0.9398
500	8	0.9450	0.9402

预训练模型后的结果超出我们预期，预训练后进行下游 finetune，验证集分数会呈现稳步上升的情况，如果上升到某个 epoch 后突然下降，线上便开始过拟合，我们记录了这个 epoch 数目，平均训练到 8-9 轮线下验证集分数就最优，此时线上也很轻松就突破了 0.94。同时我们在后续的实验中都采用了预训练 500 轮的模型结果。

结合了下游 finetune 的实验结果，我们注意到划分验证集去训练大多 8 轮时最优，我们为了让模型学习到更多的带标签的数据信息，不划分验证集，而将所有带标签的数据直接输入到模型训练，因为引入了验证集，相当于模型训练的 batch 数目变多，我们将全量训练的轮数直接设置为 7 轮，并且与直接设置为 8 轮做对比，考虑到模型训练存在波动性，我们进行了多次实验，实验结果如表6所示。

表 6: 全量训练轮数与初赛测试集分数对比

实验次数	全量训练 6 轮	全量训练 7 轮	全量训练 8 轮
1	0.9367	0.9436	0.9412
2	0.9353	0.9444	0.9399
3	0.9372	0.9438	0.9420
4	0.9361	0.9431	0.9421

3.2.3 引入 b 和 f 节点有效性分析

理论上我们觉得 f 和 b 可能为用户节点，也可能是其他和 item 密切相关的节点，我们进一步结合现实考虑到只有异常的用户才会产生异常的 item。然后我们做了可视化分析 (如图1.2)，找出所有和带标签的 item 节点相连的 f 节点，然后统计和每个 f 节点相连的 item 节点，其标签为 0 和 1 的个数，并进行可视化展示，我们发现，和大多数 f 节点大概有 2 到 3 个与之相连的 item 节点标签为 0，同时大部分 f 节点相连的 item 标签为 1 的节点个数为 0，非常少，这说明 f 节点和 b 节点如果也有黑白样本之分的话，他们之间的比例应该和 item 节点之间存在一定相关性；进一步，由于我们的模型是先将节点转为同构图，也就是把节点类型看成同一类进行训练，所以我们可以考虑引入 b 和 f 带标签节点（这时候，如何给 b 和 f 节点打上标签就成为一个值得考虑的问题!)，增加训练集样本数量，带标签节点样本数量的增加，在一定程度上可以提高模型的拟合能力。我们做了大量实验，进行验证，实验证明确实这样处理呈现正相关。

我们从两个角度去做数据增强，先训练一个模型给 item 打上伪标签和异常传播的方式给 item 的一跳邻居节点 f 和 b 打上标签，如果 f 和 b 的所有 item 邻居节点存在一个异常，我们就认为该 f 和 b 节点异常，否则认为正常，引入 f 和 b 节点需要加入 f 和 b 节点预训练，这里我们在初赛预训练 500 轮基础上引入 f 和 b 继续预训练了 300 轮。两种策略的实验结果如表7所示。

表 7: 伪标签数据增强实验结果

实验设置	初赛验证集	全量训练 7 轮复赛测试集
无增强	0.9450	0.9189
item 伪标签增强	0.9421	0.9170
b 和 f 伪标签增强	0.9462	0.9212

在得知了引入 f 和 b 带来了显著的效果后，我们继续引入复赛的测试集 item 节点，以及其一跳邻居 f 和 b 节点加入初赛已经引入 f 和 b 预训练了 300 轮的模型继续预训练。其实验结果超出了我们的预期，变得很不稳定。

表 8: 引入复赛预训练实验结果

继续预训练轮数	全量训练 7 轮复赛测试集
50	0.9222
85	0.9243(最终结果)
100	0.9221
150	0.9191

4 总结与思考

本节主要介绍我们打完这个比赛的一些收获和总结，以及我们对这个比赛的后续问题的思考和展望等。

4.1 总结

历时一个半月的大规模电商图上风险商品检测算法赛已经结束了，在这个比赛中我们学到了很多知识，锻炼了实践动手能力，提高了团队合作能力。非常感谢主办方提供的真实业务场景数据，让我们对图上的异常检测算法有更深的认识!

4.2 思考

1. 如何设计更有效的预训练模型?
2. 论文中的深度图嵌入算法为什么效果不好?
3. b 和 f 节点打标签策略?

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018.

[2] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In Proceedings of the 2019 SIAM International Conference on Data Mining, pages 594–602. SIAM, 2019.

[3] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. CoRR, abs/2006.05582, 2020.

[4] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. CoRR, abs/2103.00113, 2021.

[5] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks, 2017.

[6] Sheng Zhou. https://github.com/EagleLab-ZJU/DGLD, 2022.

[7] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. CoRR, abs/2006.04131, 2020.