基于图模型的电影推荐系统

作者姓名
(电子科技大学，计算机科学与工程学院)

摘　要: dsasdadasda
关键词: dsasdadasda
摘　要: dsasdadasda
关键词: dsasdadasda

1　引言

随着移动互联网的快速发展，我们进入了信息爆炸时代。当前通过互联网提供服务的平台越来越多，相应的提供的服务种类 (购物，视频，新闻，音乐，社交等) 层出不穷，为了更好的为用户提供服务，在为用户提供服务的同时赚取更多的利润，越来越多的公司通过采用个性化推荐技术，辅助用户更快地发现自己喜欢的东西。公司根据用户在产品上的行为记录，结合用户自身和物品的信息，利用推荐技术 (机器学习的一个分支) 来为用户推荐可能感兴趣的物品。推荐系统的主要思想是分析用户的历史行为和偏好进行建模，并自动向用户推荐感兴趣的物品，然后为用户获得个性化推荐列表 [**?**]。相比于传统的信息检索通过分类和搜索引擎去获取信息，推荐系统不需要用户提供明确的需求，甚至当用户自己都不明确自己的需求时，推荐系统通过分析用户的历史行为给用户建模，为用户推荐他感兴趣的物品或信息。

图嵌入是解决图分析问题的一种有效方法。图嵌入将图形数据转换为一个低维向量空间，在该空间中，图形结构信息和图形属性得到最大程度保留，图相关算法也能更简单地使用低维向量（嵌入）进行更有效率的计算。使用深度学习技术来进行图嵌入也是目前热门的研究领域，一些基于深度学习的图嵌入技术同时也属于图神经网络（Graph Neural Network, GNN），例如图卷积神经网络（Graph Convolutional Network, GCN）等。基于 GNN 的方法以其处理结构数据和探索结构信息的优势，已成为推荐系统中最新的方法。因此，基于图嵌入的推荐算法研究有着极大的研究意义。

2 系统架构

3 基于图注意力的归纳式图嵌入推荐算法

3.1 基于图注意力的归纳式图嵌入推荐模型

本文提出的基于图注意力的归纳式图嵌入推荐模型由以下几个部分组成：

1. 封闭子图提取：不采用在原始图即原始邻接矩阵上操作的方式，而是对每个用户/项目对进行封闭子图提取。训练好的模型在每个子图上独立进行预测。
2. RGAT 注意力层：使用四层堆叠的 RGAT 图神经网络层进行节点特征表征。
3. 池化层：将各层的隐向量拼接成为最终的子图表示。
4. MLP 全连接层：将子图嵌入作为输入预测评分值。

3.1.1 封闭子图提取

第一部分是封闭子图提取。封闭子图指的是基于图原始邻接矩阵所提取的子图，即包括原始图部分节点的子图。我们使用 G 来表示由评分矩阵构造而来的无向用户/项目二部图。在图 G 中，用户类型节点用 u 表示，项目类型节点用 v 来表示，该图中只存在用户与项目之间的连边，即用户 u 对项目 v 有过历史评分 r，不存在用户与用户以及项目与项目之间的连边。对于观察到的某一用户 u 对某一项目 v 的评分 $r = R_{u,v}$，我们从 G 中提取一个围绕 (u, v) 的一跳封闭子图。我们将提取到的一跳封闭子图馈送到 GNN，利用已知的 (u, v) 对的评分训练 GNN 模型。然后，测试阶段对于每个测试 (u, v) 对，我们再次从 (u, v) 中提取其一跳封闭子图，并使用经过训练后的 GNN 模型预测其评分。需要注意的是，在提取 (u, v) 的训练封闭子图之后，我们应该在封闭子图中删除边 (u, v)，因为它是要预测的目标。

3.1.2 图注意力消息传递层 RGAT

这一部分是训练一个多层附加注意力机制的图神经网络模型。为了考虑到不同边类型以及使用注意力机制关注不同的邻居。我们采用 RGAT 作为 GNN 的消息传递层，并对其做改进成为多层网络堆叠形式，以提高网络

对图特征的提取能力。同时进行多次实验测试选择合适的注意力系数运算方案来适应本课题的推荐任务。

- 网络输入与输出：

  这里介绍每个消息传递层的输入与输出形式，输入到某神经网络层 l 的形式为一个具有 $R = \mathfrak{R}$ 个边类型数量和总共 N 个节点的二部图。第 l 层的第 i 个节点被表示成一个特征向量 $\mathbf{x}_i^l \in \mathbb{R}^F$，全部节点的特征向量可以聚合表示为一个矩阵，即 $\mathbf{X}^l = [\mathbf{x}_1^l \mathbf{x}_2^l \cdots \mathbf{x}_N^l] \in \mathbb{R}^{N*F}$。输出则是该层经过一系列变换后的矩阵 $\mathbf{X}^{l+1} = [\mathbf{x}_1^{l+1} \mathbf{x}_2^{l+1} \cdots \mathbf{x}_N^{l+1}] \in \mathbb{R}^{N*F'}$ 其中 $\mathbf{x}_i^{l+1} \in \mathbb{R}^{F'}$ 是第 i 个节点初始特征在第 l 层经过变换后的输出，即该神经网络层计算后的节点向量表征。

- 节点中间表征：

  不同边类型 r 应该传递不同的信息，对应每种边类型 r 应使用相应可学习的权重矩阵进行一次中间特征变换，对于节点中间表征 $d_i^{l(r)}$ 更新规则如下

$$\mathbf{D}^{l(r)} = \mathbf{X}^l \mathbf{W}^{l(r)} \tag{1}$$

  其中 $\mathbf{X}^l$ 是第 l 层的输入矩阵，$\mathbf{W}^{l(r)} \in \mathbb{R}^{F*F'}$ 是第 l 层中特定边类型 r 的一个可学习的参数矩阵，$\mathbf{D}^{l(r)} = [\mathbf{d}_1^{l(r)} \mathbf{d}_2^{l(r)} \cdots \mathbf{d}_N^{l(r)}] \in \mathbb{R}^{F*F'}$ 是该层初始输入 $\mathbf{X}^l$ 在边类型 r 下的经变换后的中间表征。该步骤主要是对于不同边类型 r 的输入特征向量进行特征变换，获得在不同边类型 r 下的节点特征向量。

- 节点关联度计算统一形式：

  节点关联度指的是两个节点的关联程度大小，作为节点之间注意力系数的中间计算形式，最后还需进行 softmax 函数归一化后获得最终的节点之间的注意力系数。在两个节点之间的节点关联度仅仅是基于这些节点的特征进行计算，只收集节点领域信息，该节点关联度不同于最终节点之间的注意力系数，节点关联度 $E_{i,j}^{l(r)}$ 计算形式如下

$$E_{i,j}^{l(r)} = a(\mathbf{x}_i^{l(r)}, \mathbf{x}_j^{l(r)}) \tag{2}$$

  其中 $E_{i,j}^{l(r)}$ 对于每个边类型其中是独立的。其中，$j \in \eta_i^{(r)}$ 表示节点 j 属于节点 i 的邻域（即在用户/项目二部图中用户节点 i 与项目节点 j

有连边），表示某种对于节点的节点关联度计算形式，在下面部分给出具体计算方法

- 查询向量 Q，键向量 K，权重值向量 V：

  现我们给出节点关联度的具体形式表示:

$$q_i^{l(r)} = \mathbf{h}_i^{l(r)} \mathbf{Q}^{l(r)} \in \mathbb{R} \tag{3}$$

$$k_i^{l(r)} = \mathbf{h}_i^{l(r)} \mathbf{K}^{l(r)} \in \mathbb{R} \tag{4}$$

  节点关联度计算需要由查询向量以及键向量计算得到，其中 $\mathbf{Q}^{l(r)} \in \mathbb{R}^{F'}$ 为可学习的查询向量，$\mathbf{K}^{l(r)} \in \mathbb{R}^{F'}$ 为可学习的键向量，查询向量与键向量分别与 $h_i^{l(r)}$ 做向量乘积，得到标量值 $q_i^{l(r)}$ 与标量值 $k_i^{l(r)}$ 作为节点的查询标量值与键标量值。

- 节点关联度具体计算：

  将节点 i 的查询标量与节点 j 的键标量值相加后，即表示这两个节点之间的匹配程度，用来描述 i，j 节点之间的关联程度大小，我们称之为节点关联度。

  这里的节点关联度采用线性加和进行计算，也可以采取如乘积、拼接等方法计算得出节点 i，j 之间的关联度。

$$E_{i,j}^{l(r)} = LeakyRelu(q_i^{l(r)} + k_i^{l(r)}) \tag{5}$$

  ，其中 $LeakyRelu$ 函数具体形式如下：

$$LeakyRelu(x) = \{\, x \,, x > 0 \lambda x, x \leq 0 \tag{6}$$

  ，其中 λ 参数是预先设定的，在 0 到 1 的范围之内。

- 注意力系数：

  采用跨关系传播机制，计算第 l 层中边类型 r 下节点 i 与节点 j 之间的注意力系数 $\alpha_{i,j}^{l(r)}$:

$$\alpha_{i,j}^{l(r)} = \frac{exp(E_{i,j}^{l(r)})}{\Sigma_{r' \in \mathfrak{R}} \Sigma_{k \in \eta_i^{(r')}} exp(E_{i,j}^{l(r')})}, \forall i : \Sigma_{r \in \mathfrak{R}} \Sigma_{k \in \eta_i^{(r)}} \alpha_{i,j}^{l(r)} = 1 \tag{7}$$

  此步骤计算方法为常使用的 $softmax$ 函数，归一化计算节点 i 的邻域中某一特定节点对 (i, j) 之间的注意力系数大小

- 传播规则：

  给出最终的传播规则:

$$\mathbf{x}_i^{l+1} = \sigma(\Sigma_{r\in\mathfrak{R}}\Sigma_{j\in\eta_i^{(r)}}\alpha_{i,j}^{l(r)}\mathbf{h}_j^{l(r)}) \tag{8}$$

  其中为出现在式（？）中的中间计算表示，$\alpha_{i,j}^{l(r)}$ 为式（）计算的到的节点 i，j 之间的注意力系数，σ 为非线性激活函数，这里我们选择使用 $ReLu$ 函数

$$ReLu(x) = \{\, x\ , x > 00, x \leq 0 \tag{9}$$

3.1.3 池化层

将节点表示池化到整个封闭子图级别的特征向量中。常规的池化操作由求和、求平均、排序、DiffPooling 等。我们使用将目标用户与目标项目的最终嵌入表示作拼接，增强节点特征的信息，作为子图的表示。

节点 i 的最终向量表示：

$$\mathbf{h}_i = concat(\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^l) \tag{10}$$

然后，将处于同一封闭子图中的目标用户节点表征与目标项目节点表征做拼接池化成该子图的嵌入：

$$\mathbf{g} = concat(\mathbf{h}_u, \mathbf{h}_v) \tag{11}$$

我们使用 $\mathbf{h}_u$ 与 $\mathbf{h}_v$ 来分别表示最终的目标用户和目标项目，将 $\mathbf{h}_u$ 与 $\mathbf{h}_u$ 向量作拼接形成该子图的嵌入表示 $\mathbf{g}$。

3.1.4 MLP 全连接层

使用两层全连接神经网络，维度形状分别为 $[128, 32]$ 以及 $[32, 1]$

$$\hat{r} = \mathbf{w}^T\sigma(\mathbf{W}\mathbf{g}) \tag{12}$$

其中 $\mathbf{W}$ 和 $\mathbf{w}$ 是 MLP 的参数，用来将图表征 $\mathbf{g}$ 转换为一个标量评分值 $\hat{h}$，其中 σ 为非线性激活函数，我们使用 $ReLu$ 函数。

3.2 模型训练

我们设置最小化模型预测评分与数据集真实评分之间的均方根误差（MSE），作为损失函数，公式如下：

$$\mathcal{L} = \frac{1}{|\{(u,v) \mid \Omega_{u,v} = 1\}|_{(u,v):\Omega_{u,},v}} \sum_{u,1} \left(R_{u,v} - \widehat{R}_{u,v} \right)^2 \tag{13}$$

4 实验与分析