知识图谱上的双塔召回：阿里的IntentGC模型

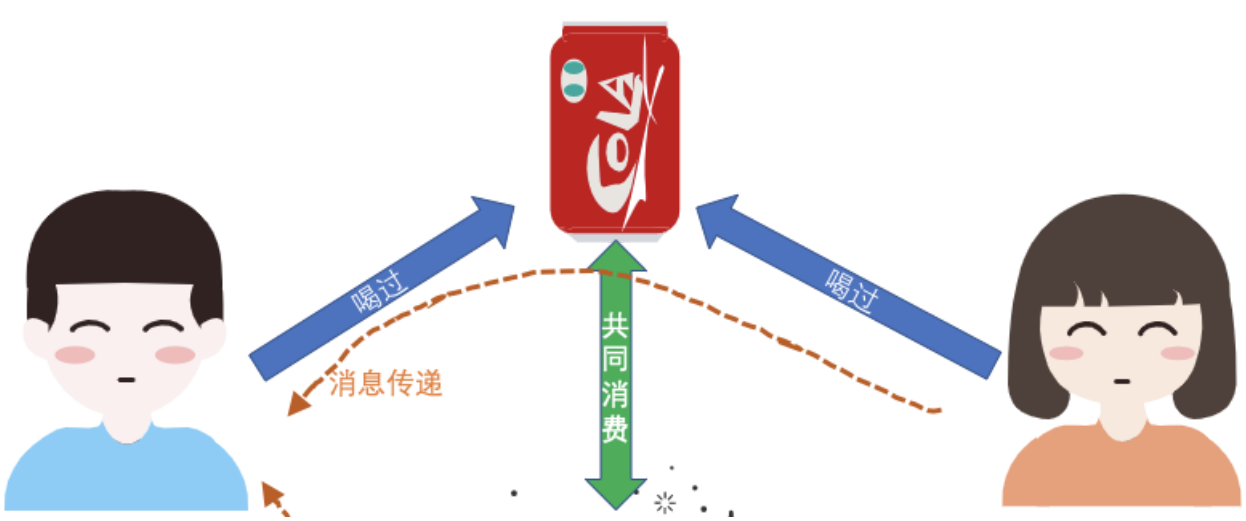
原创 石塔西 推荐道 11月25日

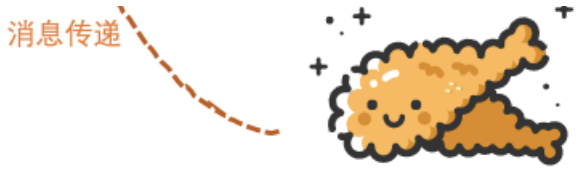
收录于话题

#图神经网络 5 #推荐算法 10 #知识图谱 1 #召回算法 2 #图卷积神经网络 6

关注本人的同学可能发现，我最近点评的文章都是关于"GNN在推荐系统应用"方向的。这当然与现如今这个方向非常火有关，但是作为一个合格的炼丹师+调参侠，总要搞清楚一门技术为什么火？这么火的技术对于自己是否有用？根据我的理解，由“传统机器学习→深度学习→图计算或知识图谱”这一路下来的发展脉络如下：

1. 一切技术的目标都是为了更好地“伺候”好“**推荐系统的一等公民 — ID类特征**”。用户购买过的商品、光顾过的店铺、搜索过的关键词、商品的分类与标签，都是这样的ID类特征
2. 传统的机器学习只会“**严格匹配**”。用户喜欢喝可口可乐，算法不会给他推百事可乐，因为“可口可乐”与“百事可乐”是两个不同的概念，占据了两个不同的ID。这时的推荐算法只有“**记忆**”功能。
3. 深度学习的特点是，**一切皆可embedding**。通过将“可口可乐”与“百事可乐”都扩展成embedding向量，发现这两个“概念”不是正交的，反而在向量空间里非常相近，从而推荐系统有机会给一个只喝过可口可乐的用户推荐百事可乐。这时的推荐算法不再只能记忆，而是有了举一反三的“**扩展**”功能。
4. 而到了“图计算”或“知识图谱”的阶段，ID类特征换了个名字，变成图上的节点或者知识图谱中的entity。换名字是小事，关键是**这些ID不再是孤立的，而是彼此关联，从而带来了信息的传递**。之前，小明喝过“可口可乐”，只有“可口可乐”这一个概念为推荐算法刻画小明贡献信息。如今，因为小红也喝过“可口可乐”，小红的信息也能传递给小明；因为“可口可乐”与“炸鸡”经常一起消费，所以“炸鸡”的信息也能够传递到小明身上。所以，**图计算或知识图谱的引入，使推荐算法能够利用的信息更加丰富，有利于缓解让人头疼的“冷启动”问题**。



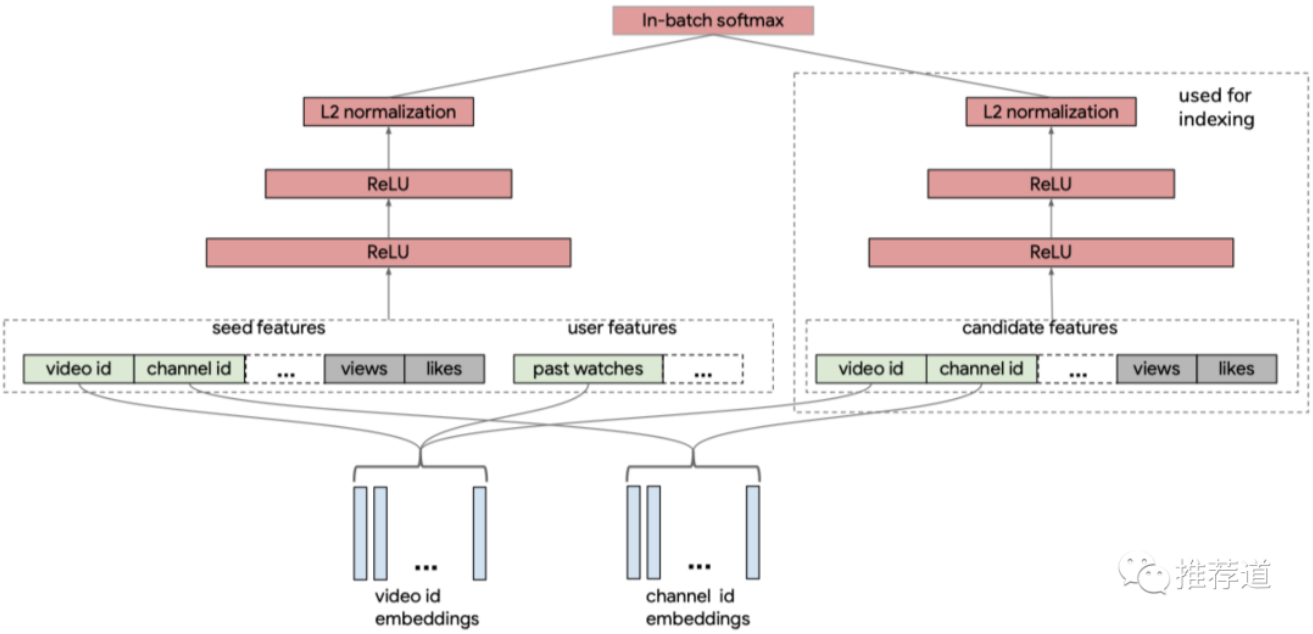


今天点评的文章是阿里发表在KDD 2019的论文《IntentGC: a Scalable Graph Convolution Framework Fusing Heterogeneous Information for Recommendation》，而此文正是上述第4点思路的典型代表。

读完整篇文章，发现**IntentGC的整体架构其实就是一个双塔模型**，只不过换了个马甲，改叫Dual Graph Convolution。而与DSSM这种传统双塔不同之处就在于，user与item特征在接入各自的塔之前，已经通过异质信息网络（Heterogeneous Information Network, HIN，实际上就是知识图谱）**聚合过各自邻居的信息**。所以，我才在文章的标题里将IntentGC形容为“**建立在知识图谱之上的双塔模型**”。

回顾双塔模型

做召回算法的，哪个能不知道大名鼎鼎的“双塔”模型？以电商推荐为例



经典双塔结构

- 把用户信息，比如用户购买过的商品、访问过的店铺、关注过的品牌、...这些ID类特征，先经过embedding，再接入user-tower，得到user embedding “z_u”
- 将商品信息，比如商品所属的一级分类、二级分类、文字描述、...这些ID类特征，先经过embedding，再接入item-tower，得到item embedding
  - 送入item-tower的，既有用户真正购买的商品v，得到其向量z_v；
  - 也有随机采样得到商品neg，得到其向量z_{neg}

    - 由于一般的随机负采样得到的商品，与用户的兴趣相差太远，对于训练算法太容易。所以也要考虑增加负采样的难度，即所谓的**hard negative**。IntentGC对这个问题也有考虑，即从正例v所属的相同类别下再采样一个item，作为hard negative。关于召回时的采样策略，见我的另一篇文章《负样本为王：评Facebook的向量化召回算法》。

- 得到用户向量z_u，用户购买过的商品的向量z_v, 随机采样得到的商品的向量z_{neg}之后，接下来就是如何设计loss。其基本思路就是，z_u与z_v应该足够近（e.g., 点积大），而z_u与z_{neg}足够远。常见的loss有hinge loss或BPR loss。IntentGC这里采用的就是hinge loss=max{0, $z_u \cdot z_{neg} - z_u \cdot z_v + \delta$}
- 训练完毕之后，将上百万的候选商品都经过训练好的item-tower生成item embedding，存入FAISS。
- 线上召回时，来访的user取得user embedding（线上实时生成，或者，离线计算好），到FAISS里面进行近邻搜索，得到与user embedding最相近的top-N个商品，作为召回结果返回。

IntentGC对双塔的改进

到目前为止，就是传统的双塔模型，so far so good。而IntentGC看到的改进点在于：

- 像用户访问过的店铺、商品所属分类这样的ID类信息，只是单纯地为刻画user和item贡献了自己本身的信息，但是它们背后的“社交”功能还未被开发和利用。
- 与当前用户逛同一家商店的其他用户的信息，对于刻画当前用户也非常有帮助。同理还有与当前用户喜欢同一品牌的其他用户的信息、与当前用户使用相同搜索词的其他用户的信息、……。正所谓“**人以群分**”，这种类似于**User Collaborative Filtering**的思想被实践证明是非常有效的。
- 与当前商品同属一个类别的其他商品的信息，对于刻画当前商品也非常有帮助。同理还有与当前商品属于一个品牌的其他商品的信息，与当前商品使用类似文字描述的其他商品的信息、……。正所谓“**物以类聚**”，这种类似于**Item Collaborative Filtering**的思想同样被实践证明是相当有效的。

IntentGC解决的思路也非常清晰，

- **双塔还保持不变**，只是喂入塔的user和item特征，需要扩展
- 不仅包括当前user和当前item自身的特征
- 还要融合当前**user**在各种“关系”（**e.g.,** 逛过同一店铺、喜欢同一品牌、使用过相同的搜索词）下的相邻**user**的特征。
- 还要融合当前**item**在各种“关系”（**e.g.,** 属于同一类别、属于同一品牌、相似的文字描述）下的相邻**item**的特征。

问题的难点在于：

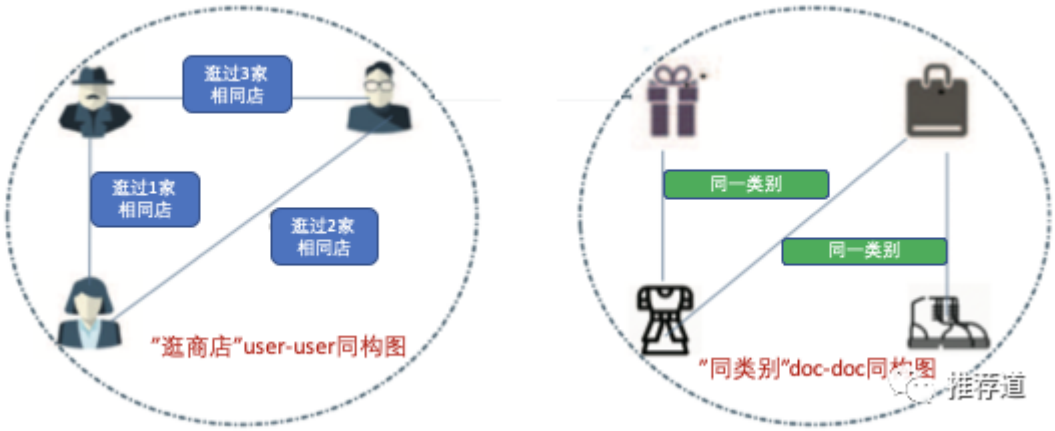
- user的邻居user，自己也有邻居user；item的邻居item，自己也有邻居item。所以这是一个多层图卷积的问题。
- 图上，存在多种节点（user和item）、多种关系（user之间逛过同一店铺、item之间属于同一品牌），因此属于异构信息网络，也即知识图谱。众所周知，处理异构图要比同构图更有难度。
- 毕竟是要在一个web-scale级别的推荐系统中投入运行，性能也是必须要考虑的问题。

接下来，让我们看看IntentGC是如何解决这些问题的。

异构图转化为多张同构图

IntentGC处理含有这么多关系的知识图谱的方法是将其转化为“**两类多张**”同构图

- 两类是指user类和item类。user类图中包含多张只有user节点的图，doc类图包含多张只有doc节点的图。**user与doc之间没有建立边，都是待预测的，user/doc的信息不能相互传递**。
- user类下，每种“关系”单独建成一张同构图。以“逛过同一家店铺”为例，
  - 建成的图上只有user节点，
  - user与user之间的边上有权重，和“两个用户共同逛过的店铺的数目”有关。这个权重越大，代表两个用户对“店铺”的爱好越相同
  - 为了降低计算规模，IntentGC在建模时，将边上权重比较小的边都删除
  - 为了避免个别“网红店”导致大量用户之间出现关联，建图时要将这些超热门的“网红店”排除在外
- doc类下，每种“关系”单独建成一张同构图，图上只有doc节点。方法与建立user-user同构图类似。



两类多张同构图

如何整合多张图上的信息

这么多张图，但是我们只需要一个user embedding，怎么融合多张图上的信息得到唯一的user embedding呢？IntentGC的作法非常简单，就是加权平均（权重待学习）

$$\mathbf{g}_u^{k-1}(i) = \sigma(w_u^{k-1}(i,1) \cdot \mathbf{h}_u^{k-1} + \sum_{r=1}^{R-2} w_u^{k-1}(i,r+1) \cdot \mathbf{h}_{\mathcal{N}(u)}^{k-1})$$

IntentGC聚合多种关系时的公式

- k代表卷积的层数，R代表“关系”总数。
- 以上函数是在用k-1层的user embedding "h_u^{k-1}"生成h_u^k，公式左边的g_u^{k-1}是生成h_u^k过程中的一个中间结果。生成h_u^k的完整流程下一节会讲到。
- $h_{N^{(r)}(u)}^{k-1}$代表第k-1层中，第r种“关系”下，用户"u"的邻居向量的聚合(pooling)结果
- $w_u^{k-1}(i,r)$代表第k-1层中，融合“第 r 种关系的邻居向量”时的权重(r=1，代表融合自身时的权重)。至于"i"，因为我们要**学习多套融合权重（类似multi-head attention）**，'i'代表第'i'套融合权重。

快速图卷积

看到上边的公式，细心的同学可能会疑惑，因为这个公式和我们常见的GCN/GraphSAGE在聚合”自身“与”邻居“时采用的公式不一样。

$$\mathbf{h}_{\mathcal{N}(u)}^{k-1} = \text{AGGREGATE}(\mathbf{h}_a^{k-1}, \forall a \in \mathcal{N}(u))$$
$$\mathbf{h}_u^k = \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_u^{k-1}, \mathbf{h}_{\mathcal{N}(u)}^{k-1}))$$

常见GCN聚合公式

与常见GCN公式相比，我们发现IntentGC的聚合公式，**只有向量之间element-wise sum，缺少了常见的“拼接+FC”**。作者给出的原因是：

- 常见的"拼接+FC"方式，涉及h_u^{k-1}向量的每一位，与$h_{N^{(r)}(u)}^{k-1}$向量的每一位，之间的交叉，占据了前代+回代中大部分的计算资源。
- 但是作者认为，**不同特征之间的高阶交叉，是“塔”的职责。既然最后要由“塔”完成不同特征之间的高阶交叉，在这里再做交叉就多余了**，被作者视为unnecessary interaction而摒弃。

为此，作者在进行图卷积时，

- 聚合自身h_u^{k-1}和邻居$h_{N^{(r)}(u)}^{k-1}$时，**只有element-wise的加权和，抛弃了复杂**$W^k \cdot CONCAT(...)$，既节省计算时间，又减少了待优化的参数而防止过拟合；

- 而且要学习多套（以下公式中的L套）加权和，类似于CNN中的多channel，或者Attention中的muliti-head，以增强模型的表达能力；
- 多套加权方式下学习到的向量，再通过另一套权重相加起来（以下公式中的θ_i），得到第k层各节点的最终向量表示；

$$\mathbf{g}_u^{k-1}(i) = \sigma(w_u^{k-1}(i,1) \cdot \mathbf{h}_u^{k-1} + \sum_{r=1}^{R-2} w_u^{k-1}(i,r+1) \cdot \mathbf{h}_{\mathcal{N}^{(r)}(u)}^{k-1})$$

$$\mathbf{h}_u^k = \sigma(\sum_{i=1}^{L} \theta_i^{k-1} \cdot \mathbf{g}_u^{k-1}(i))$$

IntentGC信息聚合公式

困惑

我们在计算大规模图卷积时，肯定不能一次性让全图参与计算，必然是采取mini-batch的方式。而mini-batch训练时，必不可少的一个步骤就是，**以mini-batch中的节点作为最后一层的target节点，逐层向下进行neighbor sampling，每层都得到一个subgraph，然后卷积只在每层的subgraph上完成。**

mini-batch中逐层邻居采样

但是IntentGC的论文里宣称，这种**采集mini-subgraph的方式低效，而IntentGC中的快速图卷积抛弃了这种方式，只是做ordinary node sampling**。

> However, for user-item HIN G, it is **difficult to generate such clustered subgraphs** for representation reusing. This is because the user-item preference links are **quite sparse**. If we follow the producer in their method for sampling, we would get a **very huge subgraph, or even the whole graph**. Hence, in order to apply our approach on large scale graphs, we develop a faster convolution operation which allows **ordinary node sampling**.

> It is worth to note that, this is a highly flexible implementation in that we **remove the limitation of training on clustered mini-graph batches**. Instead of producing clustered mini-graphs for every batch, we **sample random nodes** and fetch their neighborhoods from the graph indexing engine by hash keys in the run time of training. **The inference component is much like the training component** except without backward propagation

论文中的这些论述让我非常困惑

- 做mini-batch训练时，逐层采集邻居形成sub-graph是必不可少的呀，第3层的目标节点依赖于它们在第2层的邻居，而这些邻居又依赖于它们在第1层的邻居
- 为什么论文里说，这种方式不适用于user-item HIN，是因为sparse？那为什么又说得到的图是huge甚至是全图？关键一点，**实际上图卷积是在user-user/doc-doc同构图上进行的，根本不涉及user-item异构图**

唯一给我一点线索的就是“**The inference component is much like the training component**”这一句话。看过PinSAGE论文的同学都知道，PinSAGE在训练时和预测时，代码运行方式是完全不同的

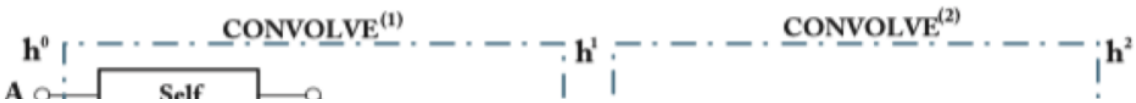
- 训练时两层循环，**外层循环每个batch，内层循环各层卷积**。
- 预测时两层循环，**外层循环每一层，内层循环各个batch。在本层将所有节点的embedding都计算出来之后，才开始下一层**。因为如果还照搬训练时的计算方式，有些节点的embedding会被重复计算。

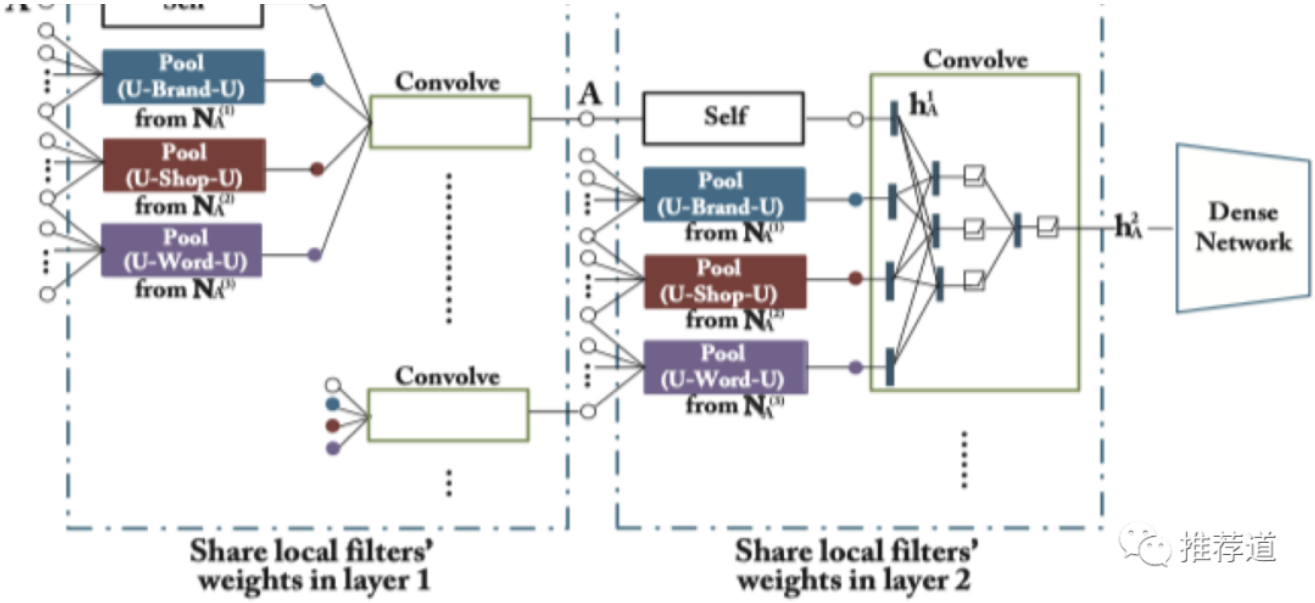
如果像IntentGC据说，训练与预测一样，我唯一能想到的就是，IntentGC在训练与预测时，遵循“**外层循环每一层，内层循环各batch；先把当前层所有节点的embedding都计算出来，才开始下一层**”的方式。但是这样一来，每个batch的回代就被推迟了，反而影响训练速度。

实在想不明白，IntentGC这种“**remove the limitation of training on clustered mini-graph batches**”的方式到底是什么样的，还请知道详情的高人，不吝赐教。或者找个机会，我再重读一遍作者的代码。

总结

经过快速图卷积，得到user节点embedding喂入user-tower，得到正负item节点的embedding喂入item-tower，由两侧的tower将各个维度的输入特征进行高阶交叉。得到最终user embedding和item embedding，喂入margin-based hinge loss。这些就属于双塔模型的常规操作了，一笔带过。整个模型结构如下所示：





IntentGC结构

至此，IntentGC就介绍完毕。总结一下它的优点：

- 在传统双塔模型中，大量的ID类特征（用户逛过的店铺、商品所属类别、.....）只是单纯为模型贡献了本身的特征，但是它们背后的“社交”功能还未被开发和利用。
- IntentGC通过图的方式，对ID类特征的“社交”信息加以利用。喂入塔的**user/item特征，不仅包含了其自身的信息，同时也融合了与其类似的user/item的信息，类似User CF或Item CF**。喂入双塔的信息大大丰富，有助于模型学到更复杂的模式，同时也缓解了对低活用户、冷门商品的“冷启动”问题。

尽管“**知识图谱+双塔**”是一个不小的创新，但是IntentGC还不能取代传统双塔模型，起码现在不能。

- 目前用于推荐系统的GNN，几乎都是**静态图**。一个user/item只能由一个节点来表示，**无法承载user/item的动态信息**。
- 而传统模型则没有这方面的限制，同一个用户可以贡献多条训练样本，而每条样本中该用户的动态信息（比如最近1小时/6小时/1天的点击分布）可以截然不同。

喜欢此内容的人还喜欢

无中生有：论推荐算法中的Embedding思想
推荐道

负样本为王：评Facebook的向量化召回算法
推荐道