

推荐融合GNN，图谱、多模态竟取得了如此惊艳的效果

原创 十方 炼丹笔记 3天前

收录于话题

#搜索推荐前沿算法 59 #必读论文系列 9

↑↑↑关注后"星标"炼丹笔记

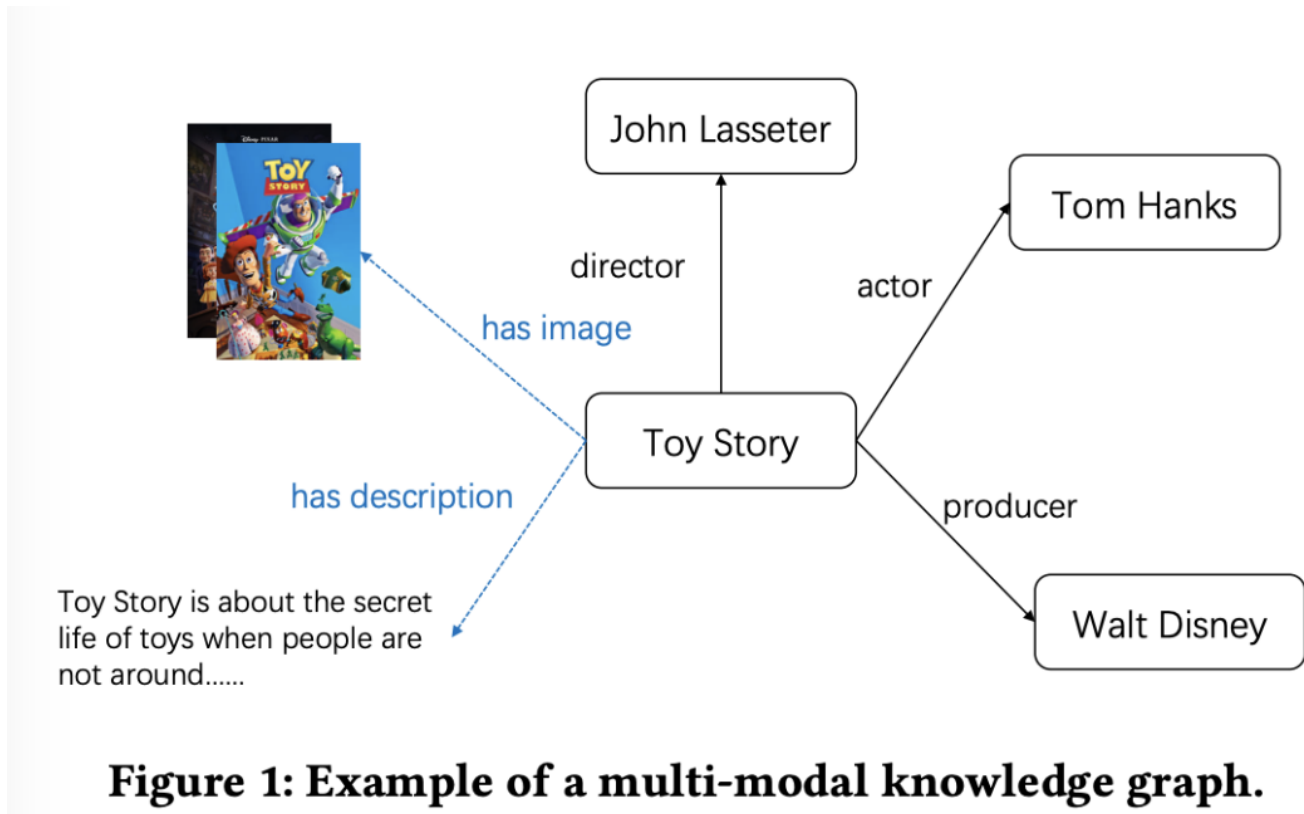
炼丹笔记干货

作者：十方，三品炼丹师

说到推荐系统，就不得不面对数据稀疏和冷启动问题，怎么解决呢？美团这篇论文《Multi-Modal Knowledge Graphs for Recommender Systems》说，我们不仅要加数据，而且是各种类型的都加。很多论文提出了用知识图谱作为推荐系统的辅助信息，而忽视了文本和图像，美团这篇论文，是第一个把知识图谱，多模态，attention都用来构建推荐系统的。模型模型也很霸气，叫MKGAT。GNN，知识图谱，多模态，推荐体系统在这篇论文里应有尽有。

知识图谱与推荐系统

知识图谱也算是火了很久，也已经广泛应用于学术界，工业界。知识图谱能给推荐系统提供丰富的特征和信息，有效的缓解了数据稀疏和冷启动。然而图片和文本会对推荐效果产生重大影响。举个🌰，在看一部电影前，用户倾向于看海报和影评。在选择吃什么前，都会在大众点评上浏览各种美食图片。所以美团觉得，把这些多模态数据融入知识图谱，会对推荐系统产生很大的正面影响。一个简单的多模态知识图谱如下图：



这张图，其实是很难作为推荐系统的输入的，所以往往是通过GNN等方式，学习图上每个实体的embedding，然后把和推荐上下文相关实体的embedding发送给下游辅助模型学习。学习多模态知识图谱的表达，有两种方法：

- 基于特征的方法：把多模态信息作为实体额外的信息，然而这需要数据源中每个实体都有多模态相关信息。
- 基于实体的方法：把各种多模态信息，都当作一个单独的实体，和其他实体建立联系。如上图"Toy Story" 和"电影海报"的关系是有一张图片。基于实体的方法就是分别用迁移模型和CNN等方法学习3元组(h,r,t)的embedding，h为head，r为relation，t为tail。上图就有很多三元组(Toy Story, has image, 海报)，(Toy Story, director, John Lasseter)等等。

论文提到，基于实体的方法关注点是实体直接的联系，而非多模态数据，因此需要更显示的表达多模态信息相关实体，这时候MKGAT隆重登场。

问题定义

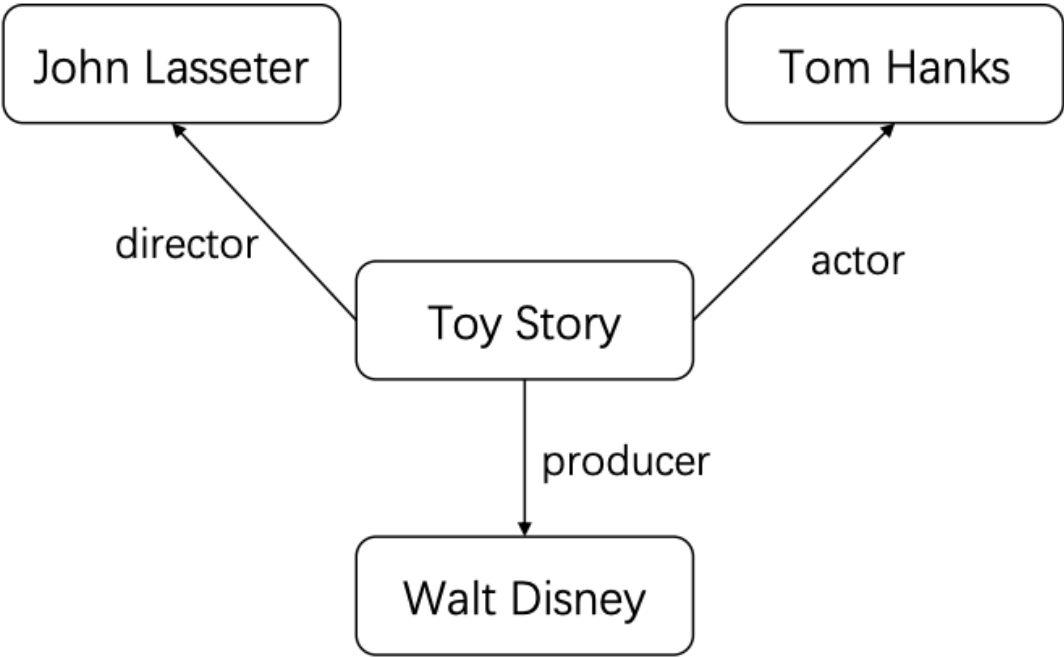


Figure 2: Example of a knowledge graph.

一个简单的知识图谱，就是由节点和边构成， $G=(V, E)$ 。从上图可知，这是个有向图，一条边 r 表达了 $head$ 和 $tail$ 的关系，如(Toy Story, actor, Tom Hanks)。多模态知识图谱中，图片文本就成了一等公民，成为了图中的实体。

联合知识图谱(collaborative knowledge graph)，如下图：

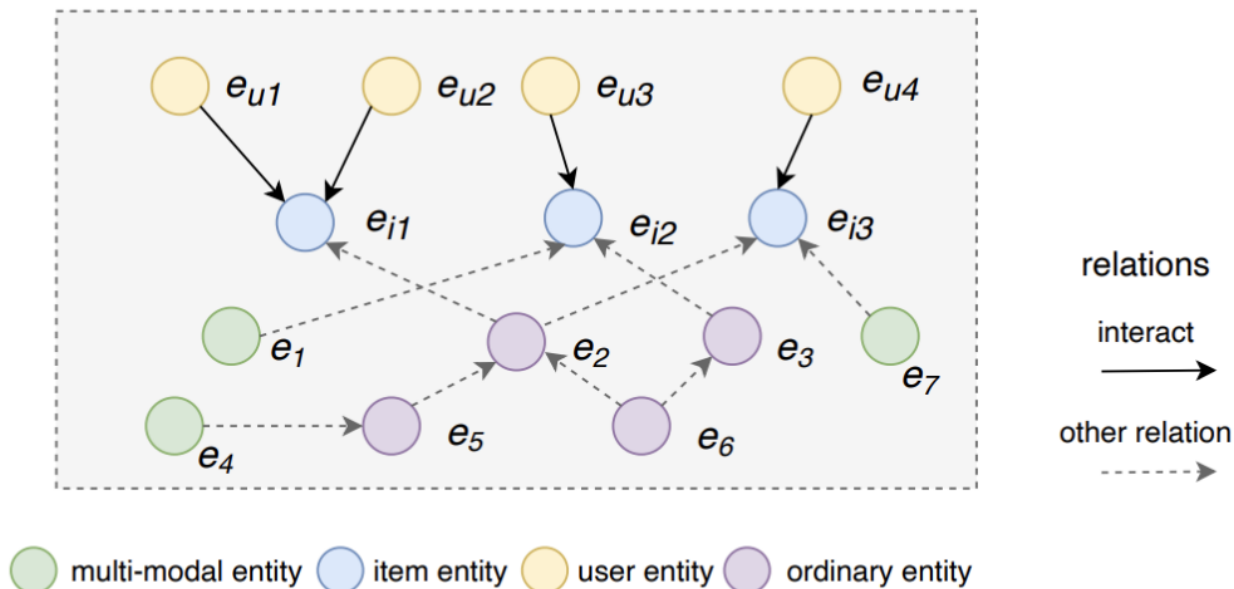
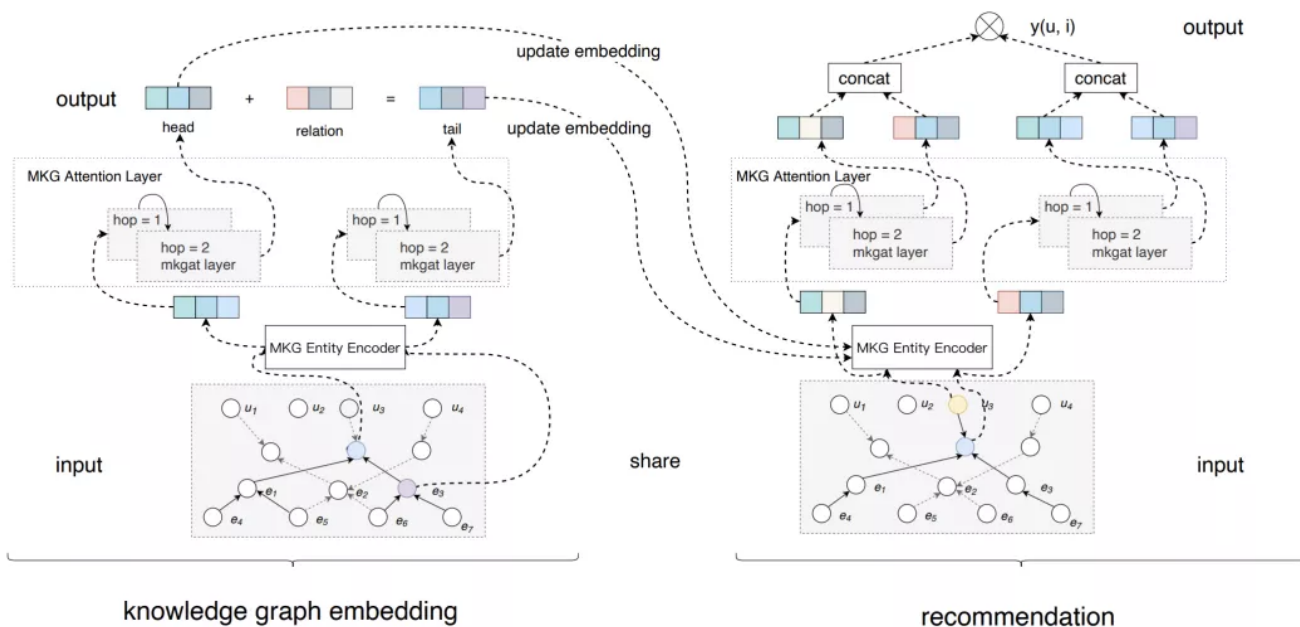


Figure 3: Example of a collaborative knowledge graph.

eu表示用户实体，ei表示item实体，绿色点就是多模态的实体，紫色点就是一般的实体。这张图箭头有实线，有虚线。实线表示某个用户对某个实体产生了某种交互(点击，购买等)，虚线表示各种非人实体之间其他关系。这篇论文的输入就是上图，输出是用户会和某个item交互的概率。

MKGAT

先看下整体模型架构：

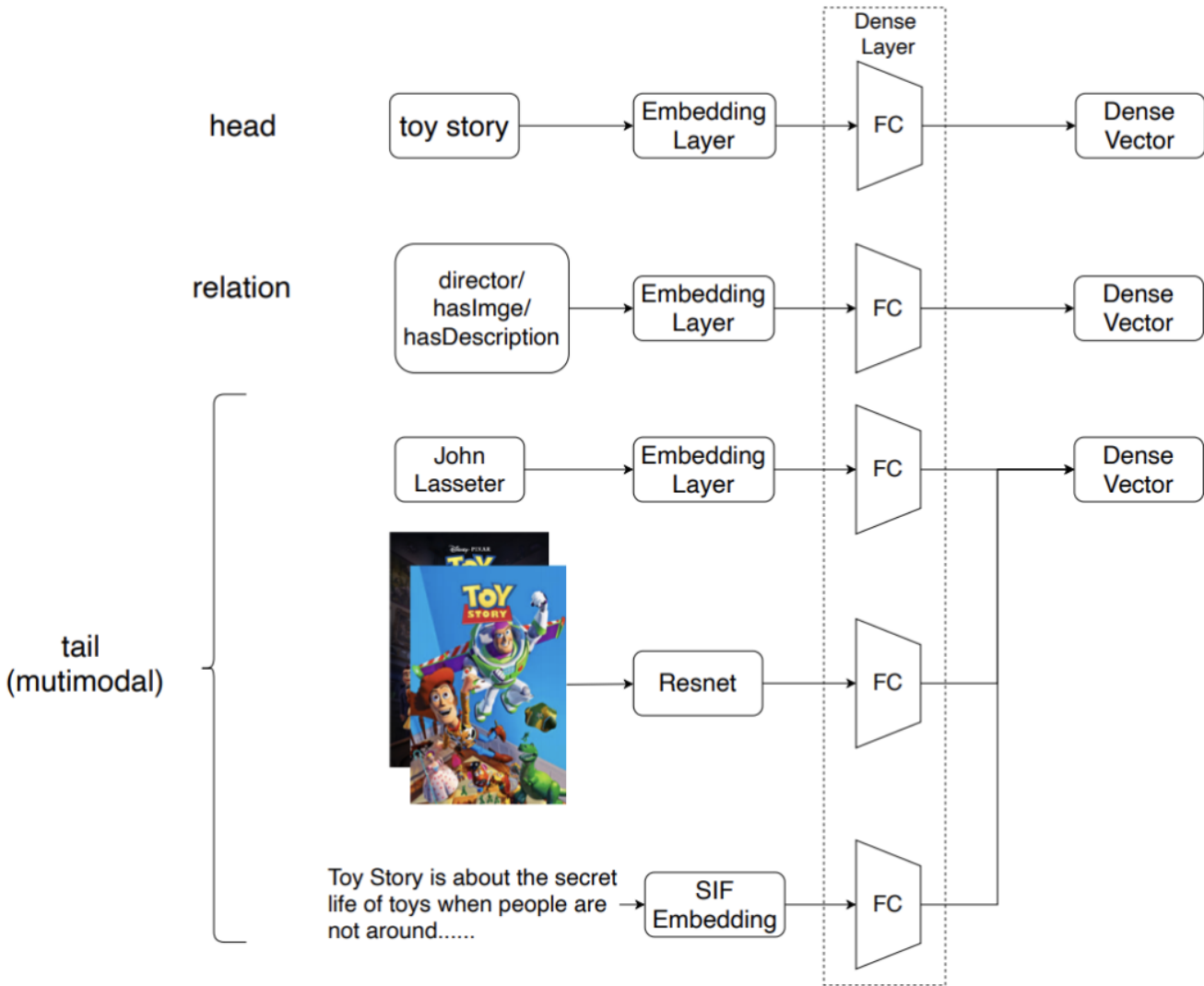


MKGAT可以拆解为两个子模块，多模态embedding模块和推荐模块。在介绍各个子模块前，我们先介绍两个小的模块：

- 多模态图谱实体编码器: 给不同类型实体编码
- 多模态图谱注意力层:用注意力机制，融合所有邻居节点的信息，学习新实体的embedding。

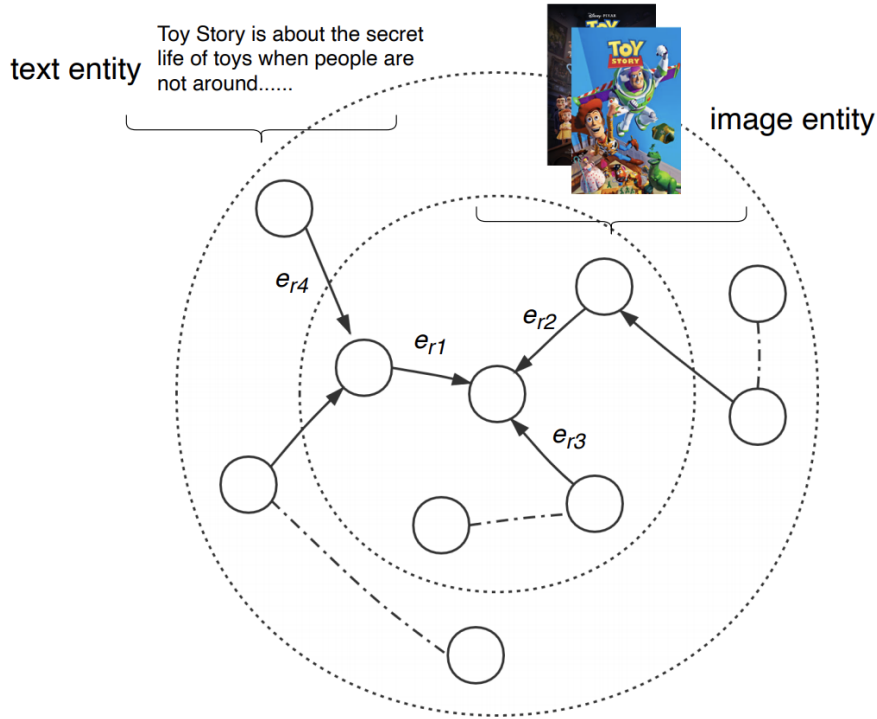
多模态embedding把联合知识图谱作为输入，充分利用上面提到的两个小模块，去学习各个entity的表达。再用各个实体embedding的表达，去学习图谱之间的关系。**推荐模块**充分利用知识图谱学到的embedding，和联合知识图谱去丰富用户和items的表达，从而提升推荐效果。

多模态图谱实体编码器其实很简单，如下图所示：



不同类型的数据用不同的 embedding，一般的实体就是直接 embedding+FC，图像就是 CNN(Resnet)+FC，文本就是 SIF+FC。

多模态图谱注意力层，看过GAT论文应该很熟悉，每个实体的embedding按照不同权重，聚合邻居节点。然而GAT忽视了KG节点直接关系是不同的，所以论文对此做了改进。如下图所示：



假设有个实体 h ，我们要学习 (h, r, t) ，在transE模型中认为 $\mathbf{h} + \mathbf{r} = \mathbf{t}$ ，但是实体 h 连接了很多多模态邻居节点，我们可以融合这些邻居信息去丰富 h ，最终得到 \mathbf{e}_{agg} ，如下公式所示：

$$\mathbf{e}_{agg} = \sum_{(h, r, t) \in \mathcal{N}_h} \pi(h, r, t) \mathbf{e}(h, r, t)$$

传播层： \mathcal{N}_h 就是 h 和所有邻居节点形成的三元组， $\mathbf{e}(h, r, t)$ 是三元组的embedding，前面乘的是该三元组的attention分数。如下定义：

$$\mathbf{e}(h, r, t) = \mathbf{W}_1(\mathbf{e}_h \parallel \mathbf{e}_r \parallel \mathbf{e}_t)$$

$$\pi(h, r, t) = \text{LeakyReLU}(\mathbf{W}_2 \mathbf{e}(h, r, t))$$

$$\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{(h, r', t') \in \mathcal{N}_h} \exp(\pi(h, r', t'))}$$

融合层：得到 \mathbf{e}_{agg} 后如何融合原本实体的向量 \mathbf{e}_h 呢？论文给出了两种方法：

加法融合：先对 \mathbf{e}_h 做线性变换到公共空间，然后直接相加（借鉴残差网络）

$$\mathbf{f}_{add} = \mathbf{W}_3 \mathbf{e}_h + \mathbf{e}_{agg}$$

concat融合：

$$\mathbf{f}_{concat} = \mathbf{W}_4(\mathbf{e}_h \parallel \mathbf{e}_{agg})$$

传播层和融合层在知识图谱中可以多次操作，以挖掘更深的信息，比如我们可以对邻居的邻居做传播融合，得到邻居的embedding后在做一次传播融合，得到该实体的embedding。

得到embedding后，就要确定目标函数了，该文用了一个pairwise ranking loss，其中t'是随机采样的实体，不满足(h,r,t')的关系，如下式所示：

$$\text{score}(h, r, t) = \|\mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t\|_2^2$$

$$\mathcal{L}_{\text{KG}} = \sum_{(h,r,t,t') \in \mathcal{T}} -\ln \sigma(\text{score}(h, r, t') - \text{score}(h, r, t))$$

推荐模块

上一节只是把embedding学好了，但我们最终目标是给用户推荐商品。在推荐模块中，attention层依然可以复用，去融合邻居节点的信息。

对于一个用户(知识图谱中的一个实体)，他已经有一个向量 \mathbf{e}_0 ，我们可以通过他的邻居，用attention的方式，再给他生成另一个向量 \mathbf{e}_1 ，用他邻居和他邻居的邻居再生成向量 \mathbf{e}_2 ，依此类推，最后把这些向量concat在一起，就是用户最终向量。推荐的item同理，如下式：

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \parallel \dots \parallel \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \parallel \dots \parallel \mathbf{e}_i^{(L)}$$

最终得到 \mathbf{e}_u^* 和 \mathbf{e}_i^* 后，直接点积再接BPR loss即可，如下式子：

$$\hat{y}(u, i) = \mathbf{e}_u^{*\top} \mathbf{e}_i^*$$

$$\mathcal{L}_{\text{recsys}} = \sum_{(u,i,j) \in \mathcal{O}} -\ln \sigma(\hat{y}(u, i) - \hat{y}(u, j)) + \lambda \|\Theta\|_2^2$$

实验

对比了目前一些算法，该MKGAT表现无论是在recall指标还是ndcg，都优于现有算法。

Models	MovieLens		Dianping	
	recall	ndcg	recall	ndcg
NFM	0.3591	0.4698	0.1163	0.0724
CKE	0.3600	0.4723	0.1321	0.0895
KGAT	0.3778	0.4827	0.1522	0.1301
MMGCN	0.3966	0.5023	0.1424	0.1255
MKGAT	0.4134	0.5181	0.1646	0.1433
%Improv.	4.2%	3.1%	8.1%	10.1%

论文还提到，多模态相比于没有多模态的图谱，对推荐效果的提升也是显而易见的。

Models	KGAT		MKGAT	
	recall	ndcg	recall	ndcg
base	0.1522	0.1301	0.1542	0.1341
base + text	0.1544	0.1343	0.1589	0.1389
%Improv.	1.5%	3.2%	3.1%	3.5%
base + image	0.1572	0.1352	0.1612	0.1396
%Improv.	3.3%	3.9%	4.5%	4.1%
base + text + image	0.1598	0.1361	0.1646	0.1433
%Improv.	4.9%	4.6%	6.7%	6.8%



炼丹笔记 | 叁



该二维码 7 天内 (1月31日前) 有效，重新进入将更新



知识星球

炼丹笔记

合伙人：九羽



长按扫码预览社群内容
和星主关系更进一步

风里雨里，我们等你！