

# [18]Collaborative Recommendation Model Based on Multi-modal Multi-view Attention Network Movie and .

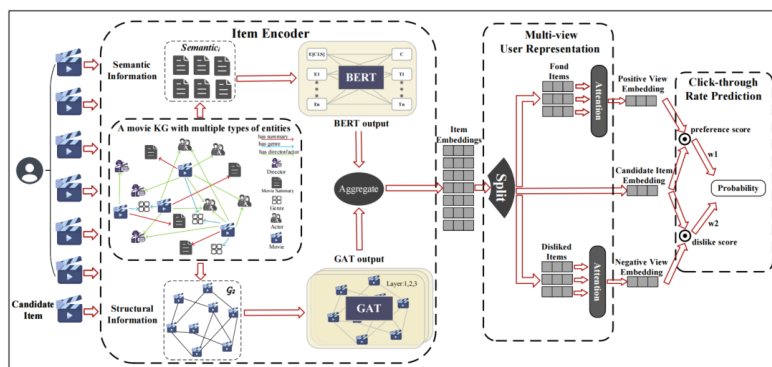
## • CRM MAN

- 基于多模态多视角注意网络的协同推荐模型
- related work
  - *knowledge graph-based recommendation*
    - 知识图可帮助解决推荐系统的可解释性和冷启动问题，许多工作将知识图谱作为额外信息加入原始推荐算法
      - Zhang等人，[28]将协同过滤与文本嵌入、知识图结构嵌入和项目的图像嵌入相结合，提出了一个端到端推荐模型。
      - Wang等人[29]利用知识感知的卷积神经网络，整合了新闻的语义级和知识级表示，生成了一个知识感知的嵌入向量。
    - 通过对消息传递机制的研究，许多工作都使用知识图中的**消息传递机制**进行推荐
      - Wang等人[16]利用Ripple的传播，将项目作为种子，并在项目知识图上进行偏好传播。他们认为，外部项也属于用户的潜在偏好，所以在表示用户时，需要考虑到它们
      - Wang等人，[20]独立地使用TransR对知识图进行了预处理。为了捕获知识图和用户-项目交互图之间的高级关系，他们建立在注意嵌入传播层的架构上，随着高阶连接递归地传播嵌入。最后，他们将这些嵌入应用于协同过滤
      - Cao等人[30]提出了基于知识增强的基于翻译的用户偏好模型，该模型联合学习用户、项目、实体和关系的表示。通过这种方式，他们从这两个任务中捕获了互补的信息，以促进它们的相互增强。
      - Wang等人，[31]旨在利用知识图嵌入任务来协助推荐任务。他们通过交叉和压缩单元明确地建模了用户和项目之间的高级交互，并自动控制了两个任务之间的知识转移
      - Sang等人[32]使用了一个传播模型来学习项目实体和用户实体在知识图中的嵌入。然后，将项目和用户嵌入输入到一个具有隐藏卷积层的交互式地图中，以显式地建模它们的嵌入维数之间的复杂成对相关性。这样，该模型就可以发现知识图所包含的高阶交互信息，从而提高推荐算法的性能。利用知识图中的辅助信息，
      - Wang等人[21]调查了用户-项目交互背后的意图。他们通过将每个意图建模为KG关系的注意组合，从而鼓励了各种意图的独立性，以获得更好的模型能力和可解释性
  - *Multi-modal Recommendation*
  - *Multi-modal Representation*

- 联合表示 (joint representations)
  - 联合表示将多种模态的信息一起投射到一个统一的多模态向量空间中。
    - 斯里瓦斯塔瓦等人[38]提出了一种用于学习深度玻尔兹曼机器的多模态数据生成模型。它通过从每个数据模态上的条件分布中采样来学习一个概率模型，从而提取出多模态数据的有意义的联合表示。
    - Tan等人[39]提出了一种基于转换器的框架，它采用了三个编码器：语言编码器、对象关系编码器和跨模态编码器来学习跨模态表示。
    - Li等人[40]使用在图像中检测到的对象标签作为锚点。以锚点作为参考物质，将自注意机制应用于文本图像对，学习联合表征
- 协调表示 (coordinated representations)
  - 协调表示将每个模态信息投射到各自的表示空间，而投影向量之间满足一定的相关约束。
    - Radford等人[41]提出了一个对比框架，该框架使用图像编码器获取图像表示，并使用文本编码器获取文本表示。这两种模式的表示被用来计算图像-文本对的相似性
    - Huo等人[42]设计了一个双塔多模态预训练模型，它隐式地模拟了图像表示和文本表示之间的跨模态相关性。
    - Duan等人[43]将不同的模态信息编码为一个联合视觉语言编码空间，该空间由集群中心的字典视图。通过聚类分配，他们比较正样本和负样本，同时优化聚类中心。

Symbol	Definition
$L$	length of text sentence
$s$	semantic information embedding vectors
$d_e$	BERT word embedding dimension
$d_h$	semantic information embedding dimension
$\mathcal{G}_1$	A KG with multiple types of entities
$\mathcal{G}_2 = (N, E)$	$\mathcal{G}_2$ is an undirected one-part graph, $N$ and $E$ are node set and edge set
$\alpha$	attention weights in graph attention mechanism
$P = \{p_1, \dots, p_M\}$	initialization vector set of graph node
$P' = \{p'_1, p'_2, \dots, p'_M\}$	intermediate result vector set of graph nodes
$P'' = \{p''_1, p''_2, \dots, p''_M\}$	structural information vector set
$d_{k''}$	dimension of structural information vector
$r$	item embedding vector
$\xi$	user history interactive item set
$\beta$	weights of self-attention mechanism
$u_{prefer}, u_{dislike}$	user preference representation and user dislike representation
$click$	item click-through rate
$c$	embedding vector of the candidate item
$w_1, w_2$	learnable parameters, representing the weights of preference and dislike click-through rate
$W$	learnable parameter matrix

## Method



- Item encoder

- 融合item的多模态信息并嵌入item

- 语义信息嵌入

- 

- $\{t_1, t_2, t_3, \dots, t_L\}$ , where  $t_1 = [CLS]$

- 

- $H = \{t'_1, \dots, t'_L\} = BERT(Input)$

- $Input = TE(tokens) + PE(tokens) + SE(tokens)$

- $tokens = \{t_1, \dots, t_L\}$

- TE: 标记嵌入

- PE: 位置嵌入

- SE: 段嵌入

- 

- We take out the word embedding  $t'_1$  corresponding to  $t_1 = [CLS]$ .

- $t'_1$ 浓缩了整个句子的语义信息，我们用它来表示这个句子，并将它输入到一个全连接的层中来进行维数变换。

- 

- $s = Wt'_1 + b$

- s: 提取的语义信息

- W: 随机初始化的可学习的投影矩阵

- b: 偏差

- 结构信息嵌入

- 使用知识图谱的信息来构造一个无向无加权单步图G2

- $G2 = (N, E)$

- 如果两个实体与同一个特定实体有关系，那么这两个实体通常在某种程度上有相似的关系。我们试图利用KG的信息来构造一个无向无加权单部分图 $G2 = (N, E)$ 来捕获这种与结构信息相似的关系

- 对于KG中对应的items的entities，如果有超过 $S_m$ 个相同的entities between 实体i和实体j，那么node i和j之间有一条边

- $E = \{e_{ij} \mid \text{if } N_i \cap N_j > S_m\}$

- $N_i$ : 实体i的邻居实体集合

- $N_j$ : 实体j的邻居实体集合

- 结构性信息反映了items之间的相关性，这有助于捕获用户对items的社区偏好

- 为捕获单步图G2中的item社区关系，使用GAT算法

- GAT算法

- 是一种依据节点的地理位置进行分簇，并对簇内的节点选择性地进行休眠的路由算法。其核心思想是：在各数据源到数据目的地之间存在有效通路的前提下，尽量减少参与数据传输的节点数，从而减少用于数据包侦听和校收的能量开销。它将无线传感器网络划分成若干个单元格(簇)，各单元格内任意一个节点都可以被选为代表，代替本单元格内所有其他节点完成数据包向相邻单元格的转发。被选中的节点成为本单元格的簇头节点其他节点都进行休眠，不发送、接收和侦听数据包
- GAT算法通常分为两个阶段
  - 第一阶段为虚拟单元格的划分。节点根据其位置信息和通信半径将网络区域划分为若干个虚拟单元格，并保证相邻单元格中的任意两个节点都可以直接通信，假设节点已知整个监测区域的位置信息和本身的位置信息，节点可以通过计算得知自己属于哪个单元格。
  - 第二阶段为虚拟单元格中的簇头节点的选择。节点周期性地进入休眠和工作状态。从休眠状态唤醒后与本单元内其他节点进行信息交换，以此确定自己是否需要成为簇头节点，每个节点处于发现、活动以及休眠三种状态。在网络初始化时，所有节点均处于发现状态，每个节点通过发送广播消息通告自己的位置和ID等信息，然后每个节点将自身的定时器设置为某个区间内的随机值  $T_d$ ，一旦定时器超时，节点发送消息声明其进入活动状态，成为簇头。节点如果在定时器超时前收到来自同一单元格内其他节点成为簇头的声明，则说明自己在这次簇头竞争中失败，从而进入休眠状态。成为簇头的节点设置定时器  $T_a$  来设置自己处于活动状态的时间。在  $T_a$  超时前，簇头节点定期广播自己处于活动状态的信息，以抑制其他处于发现状态的节点进入活动状态：当  $T_a$  超时后，簇头节点重新回到发现状态，处于活动状态的节点如果发现本单元格出现了更适合成为簇头的节点，会自动进入休眠状态。
- 使用SDNE算法对G2进行半监督学习
  - 得到初始化向量
 
$$P = \{p_1, p_2, \dots, p_M\}$$
  - 然后，我们使用一个双层多头注意机制来获得最终的节点嵌入。采用注意机制来测量不同邻居对当前节点的影响。注意机制学习到的节点表示包含项目之间的相关关系，有助于协同推荐
    - 注意力机制的权重

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [\mathbf{W} p_i \parallel \mathbf{W} p_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\vec{a}^T [\mathbf{W} p_i \parallel \mathbf{W} p_k]))}$$

- 使用基于连接策略的多头注意机制来获得一级节点表示

$$P' = \{p'_1, p'_2, \dots, p'_M\}, p'_M \in \mathbb{R}^{d_{k'}}$$

•

$$p'_i = \parallel \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k p_j \right)$$

$$\sigma(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$$

- 基于平均策略的多头注意力机制，得到最终节点嵌入

$$P'' = \{p''_1, p''_2, \dots, p''_M\}, p''_M \in \mathbb{R}^{d_{k''}}$$

•

$$p''_i = \sigma \left( \frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k p'_j \right)$$

- item嵌入

- Concatenate Aggregation

- 通过连接不同模态的表示来获得项的多模态表示

•

$$r = s_i \parallel p''_i$$

- Average Aggregation

•

When  $d_e = d_{k''}$ , we can point-wise sum  $s_i$  and  $p''_i$  and take the average

•

$$r = \frac{1}{2} (s_i + p''_i)$$

- Multi-view user representation

- 表示用户的偏好和不喜欢视图

•

$$\xi_u^{prefer} = \{v | v \in V \text{ where } y_{uv} = 1\}$$

$$\xi_u^{dislike} = \{v_{dislike} | v_{dislike} \in V \text{ where } y_{uv} = 0\}$$

$$\xi_u = \xi_u^{prefer} \cup \xi_u^{dislike}$$

- 用户u的偏好表征u<sub>prefer</sub>:

$$R' = \left( \bigparallel_{x=1}^X \left( softmax\left(\frac{RW_i^Q (RW_i^K)^T}{\sqrt{d_{hide}}}\right) RW_i^V \right) \right) W^O$$

$$u_{prefer} = Mean(R')$$

- R: 用户u喜欢的items的向量矩阵
- X: 多头自注意力机制的head number
- u<sub>dislike</sub>同理可得

- Click-through rate prediction

- 用候选item表征和用户表征来进行点击率的预测
- 应用点积法来计算点击率
- 

$$click = w1 \times c^T u_{prefer} + w2 \times c^T u_{dislike}$$

- c: 候选item C的item嵌入向量
- w1 w2是可学习的权重参数

- Model Training

- 使用负采样策略对模型进行训练
- 

$$click_i'^+ = -\log \left( \frac{\exp(click_i^+)}{\exp(click_i^+) + \sum_{r=1}^R \exp(click_{i,r}^-)} \right)$$

$$Loss = \sum_{i \in U} click_i'^+$$

- click<sub>i</sub>'<sup>+</sup>: 正则化后的第i个正例的点击概率

以上内容整理于 [幕布文档](#)