基于注意力模型的多模态特征融合雷达知识推荐

摘要

为了能够在数量庞大的雷达技术资料中快速准确地找到科研人员感兴趣的雷达知识信息并进行推荐，提出了一种基于注意力模型的文本多模态特征融合雷达知识推荐方法，学习高层次的雷达知识的多模态融合特征表示，进而实现雷达知识推荐。该方法主要包括数据预处理、多模态特征提取、多模态特征融合和雷达知识推荐 4 个阶段。

该方法的核心是通过注意力模型提升特征向量的性能，并且利用多模态特征之间优势互补的特性，提取知识的多模态特征，学习一种高层次的融合特征表示。分别提取雷达知识的词向量特征（Ｗｏｒｄ２ｖｅｃ）特征和词频 -逆文档率特征（ＴＦ－ＩＤＦ），并将雷达知识的Ｗｏｒｄ２ｖｅｃ特征输入到注意力模型中，得到Ｗｏｒｄ２ｖｅｃ 特征经过处理后的特征向量。为了实现多模态特征的融合，还设计了一种基于深度神经网络的多模态深度融合方法，结合分类交叉熵损失学习多模态特征的高阶融合特征。在雷达知识推荐阶段，采用所学习的雷达知识的融合特征计算相似度，将相似度最高的Ｎ个（Ｔｏｐ－Ｎ）推荐给用户。

基于注意力机制的雷达知识推荐模型

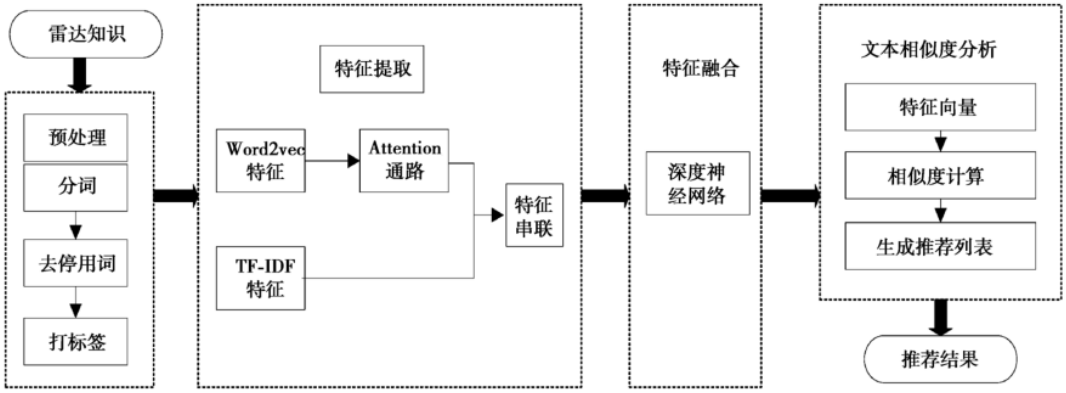


图 1　雷达知识推荐流程

1. 雷达知识预处理

   (1)分词

   (2)去停用词

2. 注意力模型

   (１)将提取的雷达知识的Ｗｏｒｄ２ｖｅｃ特征向量作为模型的输入层；

(2)使用 LSTM 作为编码器，对输入的文本特征向量进行编码，获得文本的语义编码；

(3)将语义编码输入到注意力机制层，放大重要特征的权重，得到整个网络的权重分配，充分挖掘文本的深层语义信息，得到表征能力更强的文本表示向量 ；

(4)将注意力权重与潜在语义表示进行加权平均，得到最终的文本表示。

3. 多模态特征提取

(1)提取并通过注意力模型处理雷达知识的 Word2vec 特征。

(2)提取雷达知识的基于词频的特征权重（TF - IDF）算法特征

(3)多模态特征串联

4. 多模态特征融合

为了实现多模态特征融合，提出了一种基于深度神经网络的多模态特征融合方法，结合分类交叉熵损失学习多模态特征的高阶融合特征，具体的网络结构如图 3 所示。其中，多层神经网络负责融合多模态特征，softmax 层则用于度量分类损失，从而评价多模态融合特征的性能。网络的输入为雷达知识经过注意力模型处理之后的 Word2vec特征和 TF - IDF特征的串联特征。在网络的隐藏层中，使用2个全连接层，用于学习雷达知识的高阶特征，进行多模态特征融合。通过softmax层的输出计算分类损失，使损失函数最小化，提高融合后特征的分类精度。最终输出适用于目标任务的雷达知识的高阶融合特征。

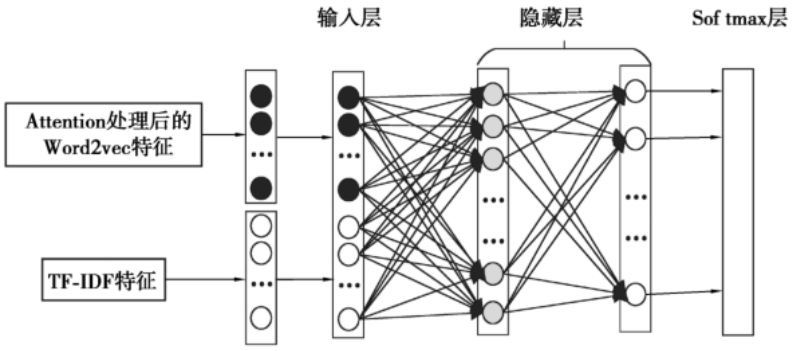


图 3　深度神经网络结构

损失函数由两部分组成。第一部分为交叉熵损失：

$$L_1 = -\sum_i y_i \log(p_i),$$

yi 为雷达知识的实际类别； pi 为 softmax 判别器输出的类别。

除此之外，还考虑了不同模态之间的相似度保持，在损失函数中引入了拉普拉斯图正则化项：

$$L_2 = \mathrm{tr}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{y}),$$

L = D - S， S 由标签信息计算得来，$S = yy^T$， D 是一个对角阵， D 的对角元为 S 的列和$\boldsymbol{D} = \mathrm{diag}\left(\sum_j S_{\mathrm{ij}}\right)$， t r 表示矩阵的迹运算。

综上所述，深度神经网络的损失函数为

$$\mathrm{Loss} = L_1 + \alpha L_2,$$

5. 雷达知识推荐

对雷达知识进行余弦相似度计算，生成推荐列表

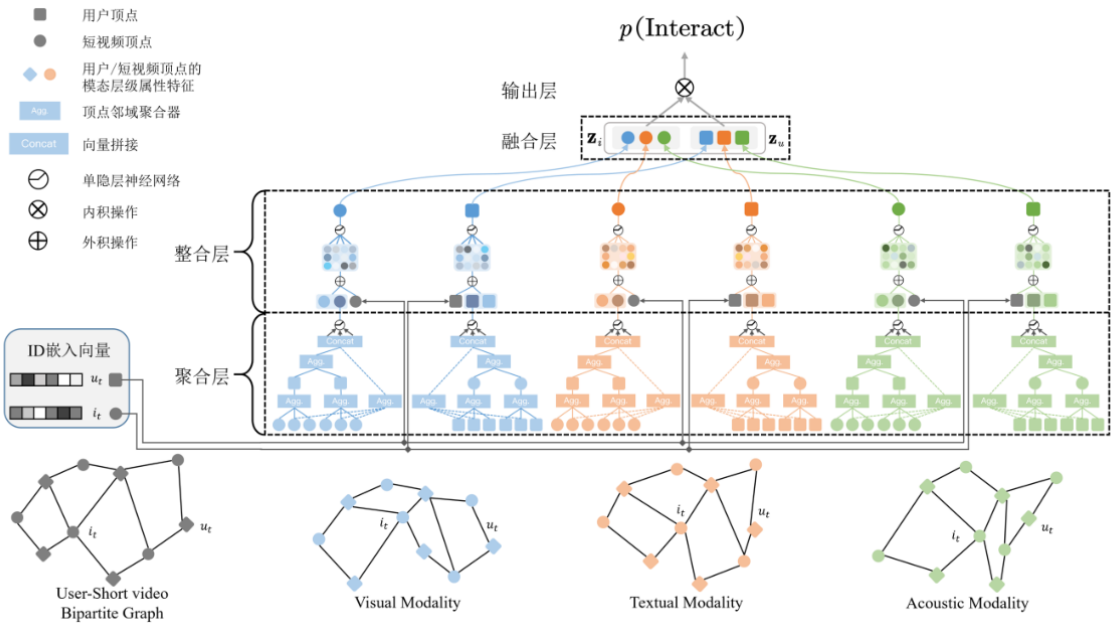
$$\mathrm{sim}(a,b) = \frac{\sum_{i=1}^{256}(v_{ai} * v_{bi})}{\sqrt{\sum_{i=1}^{256}(v_{ai})^2} * \sqrt{\sum_{i=1}^{256}(v_{bi})^2}},$$

基于多模态图卷积网络的短视频推荐算法

现有的推荐方法难以从短视频的模态层级进行用户兴趣建模，衡量短视频模态信息之间的差异性对用户偏好的影响。因此，本文提出结合短视频数据多模态的特点和图卷积网络(Graph Convolutional Network, GCN)的模型框架设计了一种基于多模态图卷积网络的短视频推荐算法，捕捉不同模态下用户的兴趣表达，进而为用户产生推荐。

本文首先根据用户对短视频的交互行为建立“用户-短视频”二部图(bipartite graph)，并依照短视频的模态种类不同分别构造对应的模态二部图；在传统 GCN 的邻域聚合方法上进行改进，提出了两级邻域聚合(Bi-level aggregation)策略以及基于注意力机制的邻域聚合器；根据 GCN中目标顶点的信息来源不同，设计阶层整合和外积整合两种整合层设计方法，实现目标顶点信息与邻域信息的整合；设计融合层实现用户以及短视频不同模态间信息的融合，输出用户和短视频的表征向量，体现短视频不同模态包含的信息差异对用户偏好的影响；最后通过输出层计算用户向量与短视频向量间的相似程度，为用户输出推荐。

多模态图卷积网络的短视频推荐算法，框架图。推荐模型主要由“用户-短视频”二部图、聚合层、整合层、融合层和输出层构成。



1.二部图

根据短视频的视觉 (visual)、文本 (textual)和听觉 (acoustic) 三种模态类型不同，将交互行为二部图转化为特定模态下的二部图。每个二部图的拓扑结构相同，图上顶点的属性信息为对应的模态信息，不同模态图中顶点之间的距离远近代表相同顶点在不同模态情况下包含信息的差异。

2.聚合层

随着 GCN 深度 的增加，位于拓扑图中距离目标顶点跳数(hop)越多的高阶邻居如二阶、三阶邻居，其属性信息对于目标顶点表示学习的影响容易随着 GCN 层的传播而逐渐平滑直到出现梯度消失的现象，出现过平滑的问题。

本文采用两级聚合策略，使用初态跳接得方法缓解过平滑问题。

聚合层由**邻域聚合**和**非线性处理**两部分组成，邻域聚合步骤将目标顶点的邻域信息通过聚合函数进行聚合，产生初步的邻域表示；

$$\mathbf{h}_{m,\mathcal{N}^k(v)}^{\ell} = f_{\text{agg}}\left(\{\mathbf{h}_{m,u}^{\ell}, \forall u \in \mathcal{N}^k(v)\}\right), \quad k \in \{1,2\}$$

将目标顶点 v 的一阶邻域信息和二阶邻域信息结合，输出目标顶点 的邻域表示向量

$$\mathbf{h}_{m,\mathcal{N}(v)}^{\ell}=\left[\mathbf{h}_{m,\mathcal{N}^1(v)}^{\ell},\mathbf{h}_{m,\mathcal{N}^2(v)}^{\ell}\right]$$

非线性处理通过将目标顶点低阶特征与其邻域特征进行拼接，输入到单层神经网络中获取目标顶点的高阶特征：

$$\mathbf{h}_{m,v}^{\ell+1}=\sigma\left(\mathbf{W}^{\ell}\cdot\left[\mathbf{h}_{m,v}^{\ell},\mathbf{h}_{m,\mathcal{N}(v)}^{\ell}\right]\right)$$

本文中，我们在传统的平均聚合以及最大池化聚合方法的基础上提出创新的**注意力聚合**方法：

2.1注意力聚合方法

通过逐顶点 (node-wise) 的方法，在目标顶点与其邻居顶点之间引入注意力分数，衡量目标顶点与邻居顶点的相似程度，从而根据邻居注意力分数大小不同，引导目标顶点进行表示学习。顶点 i 为顶点 v 的邻居，则两者的相似度定义为

$$\mathrm{sim}_{v,i}=a(\mathbf{W}_v\mathbf{h}_v^{\ell},\mathbf{W}_i\mathbf{h}_i^{\ell}),\ i\in\{\mathcal{N}^1(v),\mathcal{N}^2(v)\}$$

再通过函数LeakeyReLu 非线性转换和 Softmax 函数进行归一化得到顶点 v 和 i 之间的注意力分数：

$$\alpha_{v,i}=\frac{\exp\left(\mathrm{LeakyReLU}\left(\mathrm{sim}_{v,i}\right)\right)}{\sum\limits_{u\in\{\mathcal{N}^1(v),\mathcal{N}^2(v)\}}\exp\left(\mathrm{LeakyReLU}\left(\mathrm{sim}_{v,u}\right)\right)}$$

使用目标顶点与邻域之间的注意力分数对自身进行逐顶点聚合：

$$\mathbf{h}_{m,\mathcal{N}^k(v)}^{\ell}=\sigma\left(\sum_{u\in\mathcal{N}^k(v)}\alpha_{u,v}\mathbf{W}_u\mathbf{h}_{m,u}^{\ell}\right)$$

为了使聚合结果更加健壮，我们将多头注意力机制

$$\mathbf{h}_{m,\mathcal{N}^k(v)}^{\ell} = \Big\|_{p=1}^{P} \sigma\left(\sum_{u\in\mathcal{N}^k(v)} \alpha_{v,u}^{p}\mathbf{W}_u\mathbf{h}_{m,u}^{\ell}\right) \tag{3-13}$$

3.整合层

整合层在模型中的功能是将特定模态下目标顶点的自身属性信息与其高阶邻域信息进行整合.本文设计了阶层整合和外积整合两种整合方法用于顶点不同来源信息的整合：

$$\mathbf{H}_{m,v} = f_{\text{merge}}\left(\mathbf{h}_{m,v},\mathbf{x}_{m,v},\mathbf{h}_{v,id}\right), v\in\mathcal{V}, m\in\mathcal{M}$$

$H_{m,v}$ 为顶点 v 在模态 m 下的表示向量; $X_{m,v}$为顶点在模态 包含的原始信息，可视为第零阶信息; $h_{v,id}$为在“用户-短视频”二部图通过图嵌入方法得到的顶点 的嵌入向量，可以等效为顶点结构信息的表示向量。

3.1阶层整合

顶点的原始信息$X_{m,v}$ 和 ID 嵌入信息 $h_{v,id}$定义为顶点的低阶信息；顶点 v 经过聚合层的输出 $H_{m,v}$ 定义为高阶信息。

首先将顶点原始信息和 ID 嵌入信息两者按元素进行拼接，再通过一层前馈神经网络

$$\mathbf{h}_{m,v,low} = \text{LeakyReLU}\left(\mathbf{W}_{\text{merge}}\left[\mathbf{x}_{m,v},\mathbf{h}_{v,id}\right] + \mathbf{b}\right)$$

随后将顶点的低阶表示与高阶信息 进行级联作为整合层的输出

$$\mathbf{H}_{m,v} = \left[\mathbf{h}_{m,v,low},\mathbf{h}_{m,v}\right]$$

在本节中，我们提出了一个新的组合层，它将结构信息 hm、内在信息um和模态连接uid集成到一个统一的表示中，其公式如下：

$$\mathbf{u}_m^{(1)} = g(\mathbf{h}_m, \mathbf{u}_m, \mathbf{u}_{id}), \quad (4)$$

其中um∈ Rdm是模态m中用户u的表示；uid∈ Rd是用户ID的d维嵌入，保持不变，并用作跨模态的连接。

受先前关于多模态表示的工作的启发，我们首先应用协调方式的思想，即分开投射$\mathbf{u}_m$，进入与uid相同的潜在空间：

$$\hat{\mathbf{u}}_m = \text{LeakyReLU}(\mathbf{W}_{2,m}\mathbf{u}_m) + \mathbf{u}_{id}, \quad (5)$$

其中$\mathbf{W}_{2,m} \in \mathbb{R}^{d\times d_m}$是将um转移到ID嵌入空间的可训练权重矩阵。因此，来自不同模态的表示在同一超平面中是可比较的。同时，ID嵌入uid本质上弥补了模态特定表示之间的差距，并在梯度反向传播过程中跨模态传播信息。在这项工作中，我们通过以下两种方法实现了组合函数g（·）：

•串联组合，使用非线性变换将两种表示串联起来：

$$g_{\text{co}}(\mathbf{h}_m, \mathbf{u}_m, \mathbf{u}_{id}) = \text{LeakyReLU}\left(\mathbf{W}_{3,m}(\mathbf{h}_m||\hat{\mathbf{u}}_m)\right), \quad (6)$$

其中||是连接操作，W3,m是可训练的模型参数。

•元素组合，考虑两种表示之间的元素特征交互：

$$g_{\text{ele}}(\mathbf{h}_m, \mathbf{u}_m, \mathbf{u}_{id}) = \text{LeakyReLU}\left(\mathbf{W}_{3,m}\mathbf{h}_m + \hat{\mathbf{u}}_m\right), \quad (7)$$

其中W3,m表示将当前表示转移到公共空间的权重矩阵。在元素组合中，考虑了两个表示之间的相互作用，而在串联组合中假设两个表示是独立的。

3.2外积整合

外积整合的思路是将顶点在特定模态下信息分为**内容信息** (content information) 和**结构信息** (structural information) 两部分。$H_{m,c}$ 为内容信息，由顶点 v 经过聚合层的输出 $H_{m,v}$ 和其原始属性信息 $X_{m,v}$ 进行拼接得到；

$H_{m,s}$为结构信息由顶点的 ID 嵌入信息构成，其包含了顶点在图中的拓扑结构信息。我们通过外积的方法对两类信息的向量进行交叉，最后经过一层前馈神经网络输出：

$$\mathbf{H}_{m,v} = \text{LeakyReLU}\left(\mathbf{W}_{\text{merge}}\left(\mathbf{h}_{m,c} \otimes \mathbf{h}_{m,s}\right) + \mathbf{b}\right)$$

4.融合层和输出层

融合层将顶点（用户顶点和短视频顶点）的多个模态表示向量进行融合：

$$\begin{cases} \mathbf{z}_u = [\mathbf{H}_{V,u}, \mathbf{H}_{T,u}, \mathbf{H}_{A,u}], \ u \in \mathcal{U} \\ \mathbf{z}_i = [\mathbf{H}_{V,i}, \mathbf{H}_{T,i}, \mathbf{H}_{A,i}], \ i \in \mathcal{I} \end{cases}$$

定义“用户-短视频”二部图中与用户顶点 u 有直接相连边的短视频顶点 i_p 为正样本；负样本定义为“用户-短视频”二部图中度数较高，且与目标用户顶点没有直连边的短视频顶点i_n

损失函数

$$J(\mathbf{z}_u) = \sum_{u \in \mathcal{U}} -Q\left[\log\left(\sigma(\mathbf{z}_u^\top \mathbf{z}_{i_p})\right) + \log\left(\sigma(-\mathbf{z}_u^\top \mathbf{z}_{i_n})\right)\right],\ (u, i_p) \in \mathcal{G}, (u, i_n) \notin \mathcal{G}$$