

# 基于动态卷积概率矩阵分解的潜在群组推荐

王海艳<sup>1,2,3</sup> 董茂伟<sup>1</sup>

<sup>1</sup>(南京邮电大学计算机学院 南京 210023)

<sup>2</sup>(江苏省无线传感网高技术研究重点实验室 南京 210003)

<sup>3</sup>(江苏省大数据安全与智能处理重点实验室 南京 210023)

(wanghy@njupt.edu.cn)

## Latent Group Recommendation Based on Dynamic Probabilistic Matrix Factorization Model Integrated with CNN

Wang Haiyan<sup>1,2,3</sup> and Dong Maowei<sup>1</sup>

<sup>1</sup>(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023)

<sup>2</sup>(Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003)

<sup>3</sup>(Jiangsu Key Laboratory of Big Data Security & Intelligence Processing, Nanjing 210023)

**Abstract** Group recommendation has recently received great attention in the academic sector due to its significant utility in real applications. However, the available group recommendation methods mainly aggregate individual recommendation results or personal preferences directly based on an analysis of rating matrix. The relationship among users, groups, and services has not been taken into comprehensive consideration during group recommendation, which will interfere with the accuracy of recommendation results. Inspired by latent factor model and state space model, we propose a latent group recommendation (LGR) based on dynamic probabilistic matrix factorization model integrated with convolutional neural network (DPMFM-CNN), which comprehensively investigates rating matrix, service description documents and time factor and makes a joint analysis of the relationship among those three entities. The proposed LGR method firstly obtains a prior distribution for service latent factor model with the employment of text representation method based on convolutional neural network (CNN). Secondly, it integrates state space model with probabilistic matrix factorization model and draws user latent vector together with service latent vector. Thirdly, latent groups are detected through the use of multiple clustering algorithms on user latent vectors. Finally, group latent vectors are aggregated with average strategy and group rating can be generated. In addition, simulation on MovieLens is performed and comparison results demonstrate that LGR has better performance in efficiency and accuracy for group recommendation.

**Key words** convolutional neural network; probabilistic matrix factorization; state space model; clustering algorithms; group recommendation

**摘要** 近年来,群组推荐由于其良好的实用价值得到了广泛关注.然而,已有的群组推荐方法大多都是根据分析用户对服务的评分矩阵直接将个体用户的推荐结果或个体用户偏好进行聚合,没有综合地考虑用户-群组-服务这三者间的联系,导致群组推荐效果欠佳.受潜在因子模型与状态空间模型启发,结合评分矩阵、服务描述文档以及时间因素,共同分析用户-群组-服务间的联系,提出了一种基于动态

收稿日期:2017-05-23;修回日期:2017-06-21

基金项目:国家自然科学基金项目(61201163,61373138)

This work was supported by the National Natural Science Foundation of China (61201163, 61373138).

卷积概率矩阵分解的群组推荐方法. 该方法首先利用基于卷积神经网络的文本表示方法获取服务潜在特征模型的先验分布; 然后, 将状态空间模型与概率矩阵分解模型相结合, 获得用户潜在偏好向量与服务特征向量; 之后, 对用户偏好向量运用聚类算法来发现潜在的群组; 最终, 对群组中的用户偏好采取均值策略融合成群组偏好向量, 与服务特征向量共同生成群组对服务的评分, 实现群组推荐. 通过在 MovieLens 数据集上与同类方法进行对比实验, 发现所提方法的推荐有效性与精确性上更具有优势.

**关键词** 卷积神经网络; 概率矩阵分解; 状态空间模型; 聚类算法; 群组推荐

**中图法分类号** TP181; TP183

随着科学、技术和工程的迅猛发展, 近 20 年来, 在诸多领域(如交通旅游、健康医护、互联网和电子商务)都存在海量数据, “信息爆炸”现象日益严峻. 据统计: 2016 年全球 Wechat 月活跃用户数量已突破 8 亿, 2017 年最新公布的天猫“双十一购物狂欢节”产生 2.78 亿个订单. 如何确保用户在海量信息中快速获取所需服务, 有效解决“信息过载”问题成为了计算机学术界与产业界的一个热点研究问题. 个性化推荐系统作为主动为用户推送信息的一类解决方案, 是目前缓解这一问题较为成功的工具之一.

然而, 传统的服务推荐系统(如基于协同过滤技术的推荐系统)普遍侧重于向单个用户进行推荐, 但在现实生活的许多场景中, 用户是以群组形式出现的, 例如出行旅游、网上团购、饭店点餐等<sup>[1]</sup>. 此外, 在一些特殊的应用场景如移动 IPTV 推送服务, 由于带宽等各种各校因素的限制, 需要通过临时生成随机群组, 降低通讯开销, 才能更加有效地实现服务推荐的功能<sup>[2]</sup>. 面向群组的推荐研究受到越来越多的关注<sup>[3]</sup>. 现有的群组推荐方法大多都是根据用户对服务的评分信息简单地将个体用户的推荐结果或个体用户偏好进行聚合, 从而实现群组推荐, 较少关注到时间因素对推荐结果的影响. 这类方法在实际应用中是不合理的, 因为用户的偏好程度很有可能随时间推移发生改变; 此外, 尽管服务的一些描述信息并不能直接揭示用户、群组、服务间的潜在联系, 但这些附属信息对于充分挖掘三者间的关系, 提升推荐结果的准确度有很大帮助, 而这类附属信息却很少在群组推荐中被关注.

针对上述问题, 我们结合服务描述文档、用户配置文件以及时间因素, 深入挖掘用户-群组-服务三者间的潜在联系, 提出一种融合卷积神经网络的动态概率矩阵分解模型(dynamic probabilistic matrix factorization model integrated with convolutional neural network, DPMFM-CNN), 并将 DPMFM-CNN 融合到群组推荐框架, 提出了潜在群组推荐

方法(latent group recommendation, LGR), 主要工作有:

1) 将卷积神经网络(convolutional neural network, CNN)的文本表示方法融合到潜在因子模型中, 提出基于 CNN 的服务特征模型, 使得服务描述文档可以约束服务特征模型的生成.

2) 将状态空间模型融合到潜在因子模型中, 提出基于状态空间模型的动态概率矩阵分解模型(dynamic probabilistic matrix factorization model, DPMFM), 以此发现用户与服务的潜在关系.

3) 运用多种聚类算法来发现用户偏好空间中的潜在群组, 结合均值策略对群组偏好进行融合, 根据群组偏好来预测群组对服务的评分, 提出一种潜在群组推荐方法 LGR, 实现群组推荐.

4) 在 MovieLens1M 数据集上对所提出的方法进行测试, 并与现有的群组推荐方法进行对比, 验证我们方法的有效性与精确性, 同时, 我们分析了影响 LGR 群组推荐性能的重要参数.

## 1 相关工作

本节将介绍群组推荐相关的研究成果, 以及基于卷积神经网络的文本表示方法.

### 1.1 群组推荐

群组推荐研究是建立在个性化推荐系统基础上的, 它们将个性化推荐方法获得的输出, 采用一定的群组模型融合为群组推荐结果. 根据融合对象的不同, 群组推荐方法一般分为 2 类: 1) 基于个体推荐模型的融合; 2) 基于个体推荐结果的融合. 前者先将组内用户的偏好聚合为群组偏好, 再向用户推荐偏好服务; 后者先获取群组每个个体用户个性化服务列表, 再将推荐列表进行聚合后推送给组内用户. Baltrunas 等人<sup>[4]</sup>将个体用户的推荐列表聚合成群组推荐列表, 指出当个体推荐效果不佳时, 群组推荐列表更加有效, 此外, 组内用户偏好越接近, 群组推

荐效果越好. Boratto 等人<sup>[5]</sup>通过用户-项目评分矩阵建立用户相关性网络,用特定的聚类算法将相似用户聚类为群组,然后采用均值策略融合用户偏好为群组偏好,从而进行群组推荐. Feng 等人<sup>[2]</sup>结合概率主题模型,更加综合地去分析项目特征与用户偏好,运用重启随机游走算法分析用户与群组的关系,采用一定的策略将组内用户偏好融合成群组偏好,进行群组推荐. Boratto 等人<sup>[6]</sup>提出了一种解决群组发现过程中因聚类数据稀疏的而导致维度灾难的群组推荐方法 Predict&Cluster,该方法通过传统的协同过滤方法预测用户的评分矩阵的缺失项,对预测补全后的评分矩阵采用 K-means 方法进行聚类得到潜在群组,并将群组内所有用户评分采用均值策略融合为群组对项目的评分. Zeng 等人<sup>[7]</sup>基于矩阵因子模型,提出了并行化的潜在群组模型 PLGM,该方法运用矩阵分解方法获取用户的潜在因素偏好,通过 K-means 算法将邻近用户聚合为群组,采取均值策略将用户偏好聚合为群体偏好,通过拟合群组对服务评分进行群组推荐.

## 1.2 基于卷积神经网络的文本表示

卷积神经网络<sup>[8]</sup>(CNN)是前馈神经网络的一种,除了输入层与输出层,其所包括的基本结构有2种:卷积层进行特征提取;池化层进行特征映射.两者按照先卷积再池化的成对顺序出现.它早期被用来进行模式分类,在计算机视觉邻域得到了广泛的应用.

近年来,得益于词向量工作的成果<sup>[9-10]</sup>,一些学者将 CNN 灵活地运用到了自然语言处理. Shen 等人<sup>[11]</sup>提出了一种基于 CNN 的语义表示的信息检索方法,将需要查询的目标语句与文档表示为向量,经过该方法处理后的语句将被表示成一个向量,拥有相似语义的向量在向量空间中位置相近,以此对用户的查询语句做出快速的反馈. He 等人<sup>[12]</sup>提出了一种基于 CNN 的多角度句子语义相似度建模的方法,该方法使用词向量将句子表示成矩阵作为输入,卷积后采用不同的池化方法进行池化,以此达到多角度获取句子特征,将句子表示为向量后进行相似度比较. Kim 等人<sup>[13]</sup>在 word2vec<sup>[9]</sup>的工作基础上,提出了一种基于单隐层 CNN 的句子分类方法,并且该方法使用多通道的思想,设置多组不同卷积窗口,灵活地获取句子多种上下文特征,将文本表示成向量之后再进行分类.

我们将对 Kim 等人的工作进行改进,使其可以融合到群组推荐系统中去,帮助提高推荐的精确度.

## 2 基于 DPMFM-CNN 的潜在群组推荐方法 LGR

本节首先基于 CNN 的文本表示方法、状态空间模型与概率矩阵分解模型,提出动态卷积概率矩阵分解模型(DPMFM-CNN);然后提出一种基于 DPMFM-CNN 的潜在群组推荐方法(LGR).

### 2.1 动态卷积概率矩阵分解模型

动态卷积概率矩阵分解模型(DPMFM-CNN)的生成模型的图模型如图1所示,其中, $U_i$ 为用户偏好矩阵, $V$ 为服务特征矩阵, $R_i$ 为评分矩阵, $\theta$ 为服务文档特征矩阵, $W$ 为 CNN 的模型参数, $X$ 为服务描述文档集.首先,我们给出 CNN 服务特征模型怎样将服务描述文档表示服务特征向量(右侧部分)的方法,然后给出 CNN 服务特征模型结合 DPMFM 模型(左侧部分)生成的 DPMFM-CNN 模型,最后给出 DPMFM-CNN 模型参数学习方法.

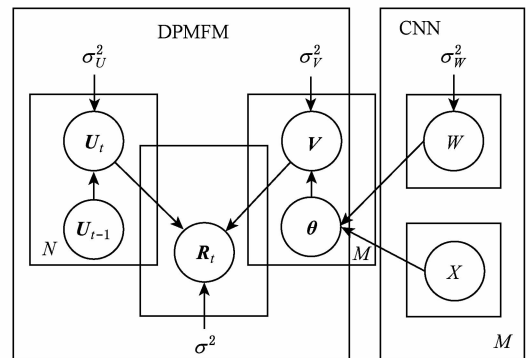


Fig. 1 DPMFM-CNN model

图1 DPMFM-CNN 模型

#### 2.1.1 CNN 服务特征模型

CNN 服务特征模型的作用与 CTR<sup>[14]</sup>中服务特征模型类似,结合 Kim 等人<sup>[13]</sup>的句子分类方法将服务描述文档表示为服务特征分布向量,并作为 DPMFM-CNN 中服务特征元素生成的条件概率,提高整个预测模型的精度. CNN 服务特征模型如图2所示,整个 CNN 服务特征模型总共4层:Embedding 层、卷积层、池化层和输出层.

##### 1) Embedding 层

Embedding 层的任务是将服务描述文档转换为一个由词向量组成的服务描述文档矩阵.第  $j$  个服务描述文档可以看做是由  $n_j$  个词组成的句子,词向量可以对服务描述语料库使用 word2vec<sup>[9]</sup>或 glove<sup>[10]</sup>等工具训练随机初始化的词向量获得,也可以使用

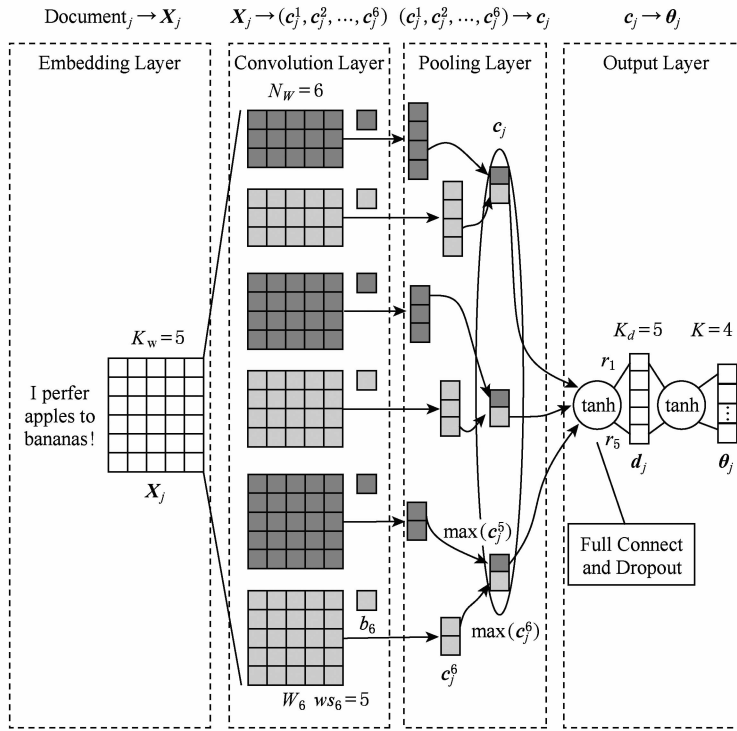


Fig. 2 CNN service feature model

图2 CNN 服务特征模型

它们预先训练好的词向量表进行查表获取,未出现在表中的词向量先随机初始化,然后进行训练获得.经过 Embedding 层的处理,文档就可以用带顺序的单词向量构成的矩阵表示为

$$\mathbf{X}_j = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_i, \dots, \mathbf{w}_{n_j}), \quad (1)$$

其中,  $j \in [1, M], n_j$  为第  $j$  个服务描述文档的总词数,  $\mathbf{w}_i \in \mathbb{R}^{K_w \times 1}, K_w$  是词量化维数.

### 2) 卷积层

卷积层的任务是提取出服务描述矩阵的上下文特征.我们采用<sup>[13]</sup>中的卷积架构对  $\mathbf{X}_j$  进行文档的上下文特征提取.假设第  $l$  个卷积窗口大小为  $\omega_{s_l}$ ,意味着每一个上下文特征由连续的  $\omega_{s_l}$  个词中提取,提取通过过滤这个窗口的权值矩阵  $\mathbf{W}_l \in \mathbb{R}^{\omega_{s_l} \times K_w}$  与其偏置项  $b_l \in \mathbb{R}$  卷积进行.提取的卷积映射属性值  $c_{i,j}^l \in \mathbb{R}$  是通过卷积窗口对  $\mathbf{X}_j$  在第  $i$  步时窗口内的内容  $\mathbf{D}_{i:(i+\omega_{s_l}-1)} \in \mathbb{R}^{K_w \times \omega_{s_l}}$  进行卷积提取出的,  $i \in [1, n_j - \omega_{s_l} + 1]$ ,其生成过程如下:

$$c_{i,j}^l = f(\mathbf{W}_l \otimes \mathbf{D}_{i:(i+\omega_{s_l}-1)} + b_l), \quad (2)$$

其中,  $\otimes$  为卷积操作符,  $f$  是一个非线性激活函数,常用的有 sigmoid, tanh 以及 ReLU,我们使用 ReLU,相比于其他函数,它可以避免梯度缺失导致收敛速度过慢或者局部最大值问题.

这样由文档矩阵  $\mathbf{X}_j$  经过第  $l$  个卷积窗口提取

出的全部卷积映射属性值构成的卷积映射属性向量可以表示为

$$\mathbf{c}_j^l = (c_{1,j}^l, c_{2,j}^l, \dots, c_{i,j}^l, \dots, c_{n_j - \omega_{s_l} + 1, j}^l). \quad (3)$$

然而,一个卷积窗口中的权值矩阵  $\mathbf{W}_l$  与其偏置项  $b_l$  只能提取一种形式的卷积映射属性向量,采用文献<sup>[13]</sup>中的多通道方法,设置多组不同的卷积窗口,这里的不同包括窗口大小  $\omega_{s_l}$  或者权值矩阵  $\mathbf{W}_l$  与其偏置项  $b_l$ .这样就可以提取不同形式的卷积映射属性向量,这里  $l \in [1, N_w], N_w$  为卷积窗口的个数.

### 3) 池化层

池化层的任务是生成定长的服务上下文特征向量.经过上面的工作,我们提取了  $N_w$  组  $\mathbf{X}_j$  的卷积映射属性向量  $\mathbf{c}_j^l$ ,但是它们都是不定长度的,并且每个卷积映射属性向量都包括了  $\mathbf{X}_j$  的全部映射属性,并不能代表上下文的明显特征.因此,我们采用最大池化,充分提取不同卷积映射属性向量的显著特征.最大池化与全连接后的结果如下:

$$\mathbf{c}_j = (\max(\mathbf{c}_j^1), \max(\mathbf{c}_j^2), \dots, \max(\mathbf{c}_j^l), \dots, \max(\mathbf{c}_j^{N_w})). \quad (4)$$

到这时,第  $j$  个服务描述文档可以被表示为维度为  $N_w$  的服务上下文特征向量  $\mathbf{c}_j$ .

### 4) 输出层

输出层的任务是将长度为  $N_w$  的服务上下文特

征向量  $\mathbf{c}_j$  映射到  $K$  维空间中去. 为了防止过拟合, 首先为模型添加 Dropout 层, 令  $\mathbf{d}_j$  为 Dropout 层的输出, 则:

$$\mathbf{d}_j = \tanh(\mathbf{W}_{\text{dropout}} \cdot (\mathbf{c}_j \cdot \mathbf{r}_{\text{dropout}}) + \mathbf{b}_{\text{dropout}}), \quad (5)$$

其中,  $\mathbf{W}_{\text{dropout}} \in \mathbb{R}^{K_{\text{dropout}} \times N_{\text{w}}}$  为非线性映射矩阵,  $\mathbf{b}_{\text{dropout}} \in \mathbb{R}^{K_{\text{dropout}} \times 1}$  为偏置向量,  $K_{\text{dropout}}$  为 Dropout 后的向量维度,  $\mathbf{r}_{\text{dropout}} \in \mathbb{R}^{N_{\text{w}} \times 1}$  为 Dropout 丢弃控制向量,  $\mathbf{r}_{\text{dropout}}$  中每一个元素  $r_p \sim \text{Bernoulli}(\text{Dropout}_{\text{rate}})$ ,  $p \in [1, K_{\text{dropout}}]$ . 从维度为  $K_{\text{dropout}}$  的向量  $\mathbf{d}_j$  映射到维度为  $K$  的服务特征向量  $\boldsymbol{\theta}_j$ , 常用的方法是进行非线性映射, 整个映射过程如下:

$$\boldsymbol{\theta}_j = \tanh(\mathbf{W}_{\boldsymbol{\theta}} \cdot \mathbf{d}_j + \mathbf{b}_{\boldsymbol{\theta}}), \quad (6)$$

其中,  $\mathbf{W}_{\boldsymbol{\theta}} \in \mathbb{R}^{K \times K_{\text{dropout}}}$  是映射矩阵,  $\mathbf{b}_{\boldsymbol{\theta}} \in \mathbb{R}^{K \times 1}$  是它的偏置向量.

CNN 服务特征模型完成了从服务描述文档  $\mathbf{X}_j$  到服务特征向量  $\boldsymbol{\theta}_j$  的转换, 整个  $\boldsymbol{\theta}_j$  的生成过程可以看成函数:  $\boldsymbol{\theta}_j = f_{\text{CNN}}(\mathbf{W}, \mathbf{X}_j)$ . 这样  $\boldsymbol{\theta}_j$  就可以作为 DPMFM-CNN 的服务特征向量  $\mathbf{v}_j$  生成的条件概率, 转换公式为

$$\mathbf{v}_j = \boldsymbol{\theta}_j + \boldsymbol{\varepsilon}, \quad (7)$$

其中,  $\boldsymbol{\varepsilon}$  是一个均值为 0 方差为  $\sigma_{\boldsymbol{\varepsilon}}^2$  的高斯噪声的向量, 可以建立起 DPMFM 与 CNN 服务特征模型的桥梁.

对于 CNN 服务特征模型中的参数集  $W$  中的每一个参数  $w_l$ , 我们假设它符合均值为 0, 方差为  $\sigma_w^2$  的高斯分布, 这样参数集  $W$  的生成概率模型的条件概率为

$$P(W | \sigma_w^2) = \prod_{w_l \in W} N(w_l | 0, \sigma_w^2). \quad (8)$$

整个 CNN 服务特征模型的条件概率为

$$P(\mathbf{V} | W, X, \sigma_v^2) = \prod_{j=1}^M N(\mathbf{v}_j | \boldsymbol{\theta}_j, \sigma_v^2). \quad (9)$$

### 2.1.2 DPMFM-CNN 的生成模型

DPMFM-CNN 在除了将 CNN 服务特征模型融入 PMF<sup>[15-16]</sup>, 还引入时间信息分析用户偏好, 时间信息是学习用户偏好演变的一个重要因素. 直接使用物理时间点对用户的偏好进行分析, 无论从计算量还是反馈信息数量都不切实际. 因此, 将时间离散化的工作必不可少, 一种直接的方法就是将时间窗口化, 窗口的大小可以运用领域知识设定, 如周、月、年等, 也可以根据用户产生数据量的流量进行自适应. 在引入时间窗口后, 我们就可以分析用户在不同窗口下的偏好了.

我们使用文献[17]中的高斯时间序列模型对用

户不同时段下的偏好进行建模. 高斯时间序列模型是一种状态空间模型, 假设用户在  $t$  状态的观察序列  $x_t$  与  $t-1$  状态的误差服从均值为  $x_{t-1}$ , 方差为  $\sigma_x^2$  的高斯分布:

$$x_t | x_{t-1} \sim N(x_{t-1}, \sigma_x^2). \quad (10)$$

我们运用高斯时间序列模型作为生成 DPMFM 模型中用户偏好的动态部分的先验概率. 根据式(10), 用户  $i$  在  $t$  时间段的偏好向量的第  $k$  个元素  $u_{ik,t}$  可以这样生成:

$$u_{ik,t} | u_{ik,t-1} \sim N(u_{ik,t-1}, \sigma_u^2), \quad (11)$$

其中,  $k \in [1, K]$ ,  $K$  为潜在特征的维数.

对于 DPMFM-CNN 中用户偏好模型的静态部分, 实际上指用户的全局偏好, 我们直接采用用户在全部时段的数据来学习全局偏好, 工作类似动态部分, 唯一不同在于其生成的先验概率为

$$\bar{u}_{ik} \sim N(0, \sigma_{\bar{u}}^2), \quad (12)$$

其中,  $\bar{u}_{ik}$  为用户  $i$  的偏好向量的第  $k$  个元素且  $k \in [1, K]$ , 全局偏好与时间是不相关的.

在完成用户的动态与静态偏好的表示后, DPMFM-CNN 模型中用户偏好的每一个元素可以被表示为  $\bar{u}_{ik,t} = \gamma \bar{u}_{ik} + (1-\gamma) u_{ik,t}$ ,  $\gamma \in [0, 1]$ , 这样用户在当前状态下产生的偏好部分将会在全局偏好中被放大, 用户动态偏好可以被认为是用户静态偏好当前状态下的修正部分, 共同决定着用户当前状态下的用户偏好.

根据上面的工作, 用户动态偏好的生成概率模型  $\mathbf{U}_t$  的条件概率

$$P(\mathbf{U}_t | \mathbf{U}_{t-1}, \sigma_u^2) = \prod_{i=1}^N N(u_{i,t} | u_{i,t-1}, \sigma_u^2). \quad (13)$$

用户  $i$  在时间段  $t$  内对服务  $j$  的评分的相对期望  $r_{ij,t}$  可以通过用户动态偏好向量与服务特征向量内积生成, 且误差服从均值为 0, 方差  $\sigma^2$  的高斯分布:

$$r_{ij,t} \sim N\left(\sum_{k=1}^K u_{ik,t} v_{jk}, \sigma^2\right). \quad (14)$$

因此整个时间段  $t$  内的评分矩阵  $\mathbf{R}_t$  生成模型的条件概率为

$$P(\mathbf{R}_t | \mathbf{U}_t, \mathbf{V}) = \prod_{i=1}^N \prod_{j=1}^M N(r_{ij,t} | \mathbf{u}_{i,t}^T \mathbf{v}_j, \sigma^2)^{I_{ij}}, \quad (15)$$

其中,  $\mathbf{I}$  为指示矩阵, 表示用户与服务是否有评分交互, 若  $R_{ij} > 0$ ,  $I_{ij} = 1$ , 否则  $I_{ij} = 0$ .

### 2.1.3 参数学习

整个模型最重要的模型参数分别为: 用户动态偏好模型  $\mathbf{U}_t$ , 用户静态偏好模型  $\bar{\mathbf{U}}$ , 服务特征模型  $\mathbf{V}$  以及 CNN 模型中的模型参数  $W$ , 本节将分别叙述

这些参数的学习过程. 由于用户的静态偏好模型与动态部分类似, 我们将仅叙述动态部分的参数学习过程.

参数学习的过程就是使生成模型可以很好地解释已观察的评分数据的过程, 我们使用最大后验 (MAP) 估计<sup>[15]</sup> 模型中的参数用户动态偏好模型  $U_t$ 、服务特征模型  $\mathbf{V}$ , 其最大后验概率的对数表达为

$$\begin{aligned} \max_{U_t, \mathbf{V}, \mathbf{W}} \ln P(\mathbf{U}_t, \mathbf{V}, \mathbf{W} | \mathbf{R}_t, X, U_{t-1}, \sigma_U^2, \sigma_V^2, \sigma_W^2, \sigma^2) \propto \\ \max_{U_t, \mathbf{V}, \mathbf{W}} [\ln(P(\mathbf{R}_t | \mathbf{U}_t, \mathbf{V}, \sigma^2) P(\mathbf{U}_t | U_{t-1}, \sigma_U^2) \times \\ P(\mathbf{V} | \mathbf{W}, X, \sigma_V^2) P(\mathbf{W} | \sigma_W^2))] = \\ \max_{U_t, \mathbf{V}, \mathbf{W}} \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij,t} (r_{ij,t} - \mathbf{u}_{i,t}^\top \mathbf{v}_j)^2 - \right. \\ \left. \frac{1}{2(\sigma_U^2)} \sum_{i=1}^N (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1})^\top (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1}) - \right. \\ \left. \frac{1}{2\sigma_V^2} \sum_{j=1}^M (\mathbf{v}_j - \boldsymbol{\theta}_j)^\top (\mathbf{v}_j - \boldsymbol{\theta}_j) - \frac{1}{2\sigma_W^2} \sum_{w_l \in \mathbf{W}} \omega_l^2 - C \right], \end{aligned} \quad (16)$$

其中,  $C = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij,t} \ln(2\pi\sigma) + \frac{MK}{2} \ln(2\pi\sigma_U) + \frac{NK}{2} \times \ln(2\pi\sigma_V) + \frac{|W|}{2} \ln(2\pi\sigma_W)$ , 是一个常数,  $K$ ,  $\mathbf{R}_t, X, U_{t-1}, \sigma_U^2, \sigma_V^2, \sigma_W^2, \sigma^2$  均为已知变量或给定常量,  $\boldsymbol{\theta}_j = f_{\text{CNN}}(\mathbf{W}, \mathbf{X}_j)$ ,  $|W|$  为整个 CNN 服务特征模型中的参数个数.

将式(16)进行处理后会得到一个关于  $U_t, \mathbf{V}, \mathbf{W}$  的函数, 并作为损失函数:

$$\begin{aligned} L(\mathbf{U}_t, \mathbf{V}, \mathbf{W}) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij,t} (r_{ij,t} - \mathbf{u}_{i,t}^\top \mathbf{v}_j)^2 + \\ \frac{\lambda_U}{2} \sum_{i=1}^N (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1})^\top (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1}) + \\ \frac{\lambda_V}{2} \sum_{j=1}^M (\mathbf{v}_j - \boldsymbol{\theta}_j)^\top (\mathbf{v}_j - \boldsymbol{\theta}_j) + \frac{\lambda_W}{2} \sum_{w_l \in \mathbf{W}} \omega_l^2, \end{aligned} \quad (17)$$

其中,  $\lambda_U = \frac{\sigma^2}{\sigma_U^2}$ ,  $\lambda_V = \frac{\sigma^2}{\sigma_V^2}$ ,  $\lambda_W = \frac{\sigma^2}{\sigma_W^2}$ .

用最优化方法求解, 以此获得用户动态偏好模型  $U_t$ 、服务特征模型  $\mathbf{V}$ . 这里, 我们采用 SGD<sup>[18]</sup> 方法来增量学习模型参数. 因此, 当前模型进行训练的每一个样本  $(i, j, r_{ij,t})$  相关的损失函数可以简化为

$$\begin{aligned} L(\mathbf{u}_{i,t}, \mathbf{v}_j) = \frac{1}{2} (r_{ij,t} - \mathbf{u}_{i,t}^\top \mathbf{v}_j)^2 + \frac{\lambda_U}{2} (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1})^\top \cdot \\ (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1}) + \frac{\lambda_V}{2} (\mathbf{v}_j - \boldsymbol{\theta}_j)^\top (\mathbf{v}_j - \boldsymbol{\theta}_j). \end{aligned} \quad (18)$$

DPMFM-CNN 的用户动态偏好模型  $U_t$ , 服务特征模型  $\mathbf{V}$  的参数学习算法步骤是:

- 1) 初始化  $\lambda_U, \lambda_V, \eta, Iter_{\max}, C_{\text{endure}}, \mathbf{U}_t \leftarrow \mathbf{U}_{t-1}, \mathbf{V} \leftarrow \boldsymbol{\theta}, \mathbf{W}$ ;
- 2) 输入  $(i, j, r_{ij,t}) \in \mathbf{R}_t, X$ ;
- 3) 更新  $\mathbf{u}_{i,t}$  与  $\mathbf{v}_j$ :  
While  $Iter_{\max} \geq iter$  且  $C_{\text{endure}} > C_B$   
 $\mathbf{u}_{i,t} \leftarrow \mathbf{u}_{i,t} - \eta ((\mathbf{u}_{i,t}^\top \mathbf{v}_j - r_{ij,t}) \mathbf{v}_j + \lambda_U (\mathbf{u}_{i,t} - \mathbf{u}_{i,t-1}))$ ,  
 $\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta ((\mathbf{u}_{i,t}^\top \mathbf{v}_j - r_{ij,t}) \mathbf{u}_{i,t} + \lambda_V (\mathbf{v}_j - \boldsymbol{\theta}_j))$ ,  
Update  $\mathbf{W}$ ,  
 $\boldsymbol{\theta}_j \leftarrow f_{\text{CNN}}(\mathbf{W}, \mathbf{X}_j)$ ;
- 4) 输出  $\mathbf{U}_t \in \mathbb{R}^{K \times N}$  和  $\mathbf{V} \in \mathbb{R}^{K \times N}$ .

其中,  $\lambda_U$  与  $\lambda_V$  为正则项系数,  $K$  为潜在特征维数,  $\eta$  为学习率,  $Iter_{\max}$  为最大迭代次数,  $C_{\text{endure}}$  与  $C$  分别为允许的误差回退次数与误差回退次数.

不同于更新用户动态偏好模型  $U_t$  与服务特征模型  $\mathbf{V}, \mathbf{W}$  是与 CNN 模型中每一层相关的参数, 包括: 每个卷积窗口的权值矩阵  $\mathbf{W}_l$  与其偏置项  $b_l$ , 以及非线性映射的权值矩阵  $\mathbf{W}_{\text{dropout}}, \mathbf{W}_\theta$  与偏置向量  $\mathbf{b}_{\text{dropout}}, \mathbf{b}_\theta$ . 这里我们采用反向传播算法训练模型参数, 并将

$$\begin{aligned} \varepsilon(W) = \frac{\lambda_V}{2} \sum_{j=1}^M (\mathbf{v}_j - \boldsymbol{\theta}_j)^\top (\mathbf{v}_j - \boldsymbol{\theta}_j) + \\ \frac{\lambda_W}{2} \sum_{w_l \in \mathbf{W}} \omega_l^2, \end{aligned} \quad (19)$$

看作是  $W$  的带  $l_2$  正则项的损失函数, 其中,  $\boldsymbol{\theta}_j = f_{\text{CNN}}(\mathbf{W}, \mathbf{X}_j)$ ,  $j \in [1, M]$ .

## 2.2 基于 DPMFM-CNN 的潜在群组推荐框架

在 2.1 节中, 我们已经通过 DPMFM-CNN 模型获取到了用户的动态与静态偏好, 与其他基于个体推荐模型融合的群组推荐方法 PLGM<sup>[7]</sup> 类似, 我们将采用多种聚类算法对 DPMFM-CNN 的输出的用户偏好向量进行聚类, 以此来发现用户集中的潜在群组, 结合均值策略进行群组偏好融合, 提出一种潜在群组推荐方法 (LGR), 其结构如图 3 所示.

整个群组推荐过程主要分为 3 步, 由上到下分别为潜在群组发现、群组偏好融合和群组评分预测.

### 2.2.1 潜在群组发现

群组可以被认为是服务有相似特征偏好的用户集合, 而从个体用户到群组的过程, 我们认为是一个聚类任务. 我们将采用 4 种适合于我们的聚类算法进行潜在群组发现:

- 1)  $K$ -means<sup>[19]</sup>.  $K$ -means 基本思想是以空间中  $k$  个点为中心进行聚类, 对最靠近他们的对象归类.

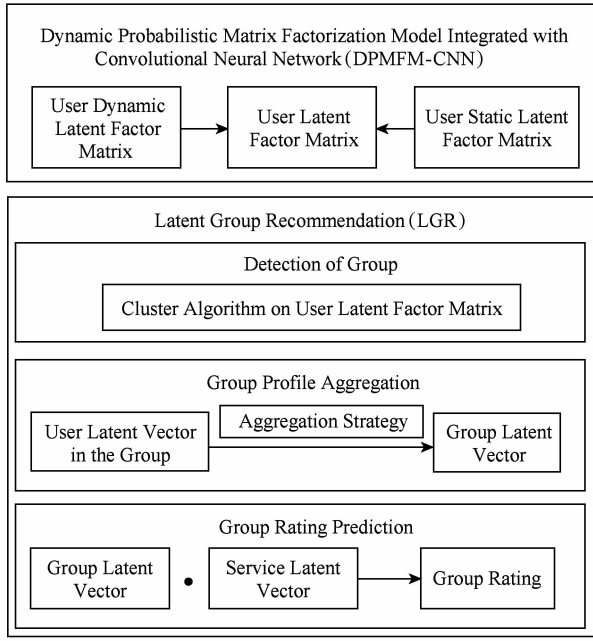


Fig. 3 Structure of latent group recommendation

图3 潜在群组推荐结构

通过迭代的方法,逐次更新各聚类中心的值,直至得到最好的聚类结果。

2)  $K$ -means++<sup>[19]</sup>.  $K$ -means++ 算法是  $K$ -means 的演化算法,不同于  $K$ -means 算法的是,初始中心点不再随机,而是符合一定原则,基本原则是使得各个种子点之间的距离尽可能的大,但是又得排除噪声的影响。

3) Birch<sup>[20]</sup>. Birch 算法是层次聚类算法之一,基本思想主要是引入了聚类特征(clustering feature, CF)和聚类特征树(CF Tree),首先通过其他的聚类方法将其聚类成小的簇,然后再在簇间采用 CF 聚类特征对簇聚类。

4) AffinityPropagation<sup>[21]</sup>. 该算法的基本思想是将全部样本看作网络的节点,然后通过网络中各条边的消息传递,计算出各样本的聚类中心. 聚类过程中,共有 2 种消息在各节点间传递,分别是吸引度(responsibility)和归属感(availability). 聚类的结果取决于样本间的相似性大小和消息传递。

希望通过不同聚类方法对群组推荐的性能表现的对比,找出最适合于发现用户偏好向量空间中具有内在相似性的用户集合的聚类方法来发现潜在群组。

## 2.2.2 群组偏好融合

通过发现用户间的内在联系并将用户聚类为群组之后,一个重要的工作就是将群组内用户的偏好

融合为群组偏好. 均值策略<sup>[22]</sup>被认为是大规模数据环境下产生群组偏好最精确的融合策略,我们将采用均值策略进行群组偏好  $\{g_j\}_{j=1}^{N_g}$ :

$$g_j = \frac{\sum_{c_i=j} \bar{u}_{i,t}}{N_{g_j}}, \quad (20)$$

其中,用户  $i$  的潜在偏好向量  $\bar{u}_{i,t} = \gamma \bar{u}_i + (1-\gamma) u_{i,t}$ ,  $\gamma \in [0,1]$ ,  $c_i$  为用户  $i$  所属的群组编号,  $N_g$  为潜在群组发现所发现的群组数目,  $N_{g_j}$  为第  $j$  个群组中用户的个数。

## 2.2.3 群组评分预测

在获得用户的群组归属后,将群组的偏好向量与服务特征向量进行内积,拟合群组对服务的评分. 不同于传统个体用户评分的拟合,这里的群组评分的拟合是群组中具有内在相似性用户的共性偏好与服务特征向量拟合的结果,整个用户对服务的拟合评分为

$$\bar{r}_{ij,t} \leftarrow g_{c_i}^T v_j, \quad (21)$$

其中,  $g_{c_i}$  为用户  $i$  所在群组的偏好向量,  $i=1,2,\dots,N$ ,  $j=1,2,\dots,M$ .

## 3 实验与分析

为了验证 LGR 的有效性 with 准确性,我们首先介绍整个实验的实验准备,然后展示实验结果并对其进行分析。

### 3.1 实验准备

本节主要对数据集、实验环境与具体设置、对比方法与评价指标这 3 个方面进行叙述。

#### 3.1.1 数据集

为了说明我们提出的基于 LGR 的群组推荐方法的有效性 with 准确性,我们选取了公开的真实数据集 MovieLens1M 作为实验数据集,此数据集主要包括用户的显式评分相关的信息. 由于 MovieLens1M 数据集并不包括电影的描述文档,因此,我们将对 MovieLens1M 数据集中的电影描述信息进行补充,根据 MovieLens1M 中每部电影名到 IMDB 中的获取其关联信息,并将每部电影的总结作为电影的描述文档。

类似 CTR<sup>[14]</sup> 中对数据的预处理工作,数据预处理主要步骤如下:1) 去除没有描述文档的电影记录;2) 去除评分次数少于 3 次的用户信息;3) 设置最大描述文档长度为 300;4) 计算每个词的 tf-idf,

去除停顿词(tf-idf 得分大于 0.5);5)选取剩余词中 tf-idf 得分前 8000 个单词作为词汇表;6)将描述文档中不在词汇表中的单词移除. 尽管 MovieLens1M 数据集的一些记录被移除,但并没有对原始数据集的稀疏度产生明显改变.

预处理后的 MovieLens1M 数据集的数据统计信息如表 1 所示:

**Table 1 Data Statistics of the MovieLens1M Data Set After Preprocessing**

表 1 预处理后的 MovieLens1M 数据集的数据统计信息

Field	Value
# users	6 040
# movies	3 544
# ratings	993 482
# timestamps	993 482
Density/%	4. 641 2

### 3.1.2 实验环境与具体设置

实验环境是 CPU 为 Intel 酷睿 i7 6700HQ 处理器, GPU 为 Nvidia GTX960M 独立显卡, 内存 8 GB, 系统为 Ubuntu 16.04 LTS.

为了实现 DPMFM-CNN, 我们主要用 python 及其一种深度学习框架 keras 与科学运算库 numpy、scipy 实现. LGR 的聚类算法采用 python 机器学习库 sklearn 中的聚类算法.

为了分析时间信息对用户偏好模型的影响, 我们对数据集做设置: 将 MovieLens1M 数据集前 80% 的数据分别按半年为时间窗口分段作为训练集, 将最后 20% 的数据作为测试集.

为了训练 CNN 的参数  $W$ , 我们使用基于 mini-batch 的 RMSprop 方法, 并且使每一个 mini-batch 包括 128 条数据. 更具体地做了设置: 1) 服务描述文档最大长度设置为 300; 2) 每一个词向量的维度设置为 200, 并且采用 glove<sup>[10]</sup> 预训练结果作为词向量的初始化输入; 3) 设置 3 组不同宽度的卷积窗口(宽度分别为 3, 4 和 5); 4) 每一种宽度的卷积窗口在卷积层的个数都设置为 100; 5) 经过多次实验的调试, 设置  $Dropout_{rate} = 0.2$ .

为了训练用户动态偏好模型  $U_t$ , 静态偏好模型  $\bar{U}$  与服务特征模型  $V$ , 我们使用 SGD<sup>[18]</sup> 方法来获得, 具体设置为: 1) 设置 MovieLens1M 数据集上训练时的  $\lambda_U = 100, \lambda_V = 10$ ; 2) 设置潜在特征维度  $K =$

50; 3) 设置 SGD 学习率  $\eta = 0.01$ ; 4) 设置最大迭代次数为 200; 5) 设置允许误差回退次数阈值  $C_{endure} = 5$ .

### 3.1.3 对比方法与评价指标

我们将在预处理后的 MovieLens1M 数据集上与 2 个基准方法进行对比.

1) Predict&Cluster<sup>[6]</sup> 方法通过传统的协同过滤方法预测用户的评分矩阵的缺失项, 对预测补全后的评分矩阵采用  $K$ -means 方法进行聚类得到潜在群组, 并将群组内所有用户评分采用均值策略融合为群组对项目的评分.

2) PLGM<sup>[7]</sup> 方法通过潜在因子模型获取用户隐式偏好, 对用户隐式偏好采用  $K$ -means 方法进行聚类得到潜在群组, 并将群组内所有用户偏好采用均值策略融合为群组偏好, 最终通过潜在因子模型预测群组对项目评分.

我们选取了均方根误差 (root mean squared error, RMSE) 来对群组推荐结果的精确性进行分析. 其在群组推荐场景下的表达形式为:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_{ij,t} - \tilde{r}_{ij,t})^2}, \quad (22)$$

其中,  $r_{ij,t}$  表示观察的数据集中用户  $i$  在  $t$  时段对服务  $j$  的评分,  $\tilde{r}_{ij,t}$  是用户  $i$  所在群组在  $t$  时段给出的服务  $j$  的预测评分,  $N$  是作为测试集的评分数量. 由其定义可知, 指标 RMSE 越小, 则所预测的推荐结果越准确.

## 3.2 实验结果与分析

本节将会从 2 个方向对实验结果进行分析: 首先在对实验结果中分析基于 DPMFM-CNN 的 LGR 的精确性与有效性; 然后分析模型中影响推荐方法性能的主要参数.

### 3.2.1 精确性与有效性分析

首先给出 LGR 与所有预采用的聚类方法在用户偏好拟合系数  $\gamma = 0.6$ 、正则化系数  $\lambda_U = 100$  与  $\lambda_V = 10$ 、潜在特征维数  $K = 50$ 、词向量维数  $K_w = 200$ 、偏好融合策略为均值策略、群组数量  $N_g = 1, 20, 50, 200, 500, 6040$  所展示的 RMSE 值. LGR 随群组数量变化的 RMSE 值如表 2 所示.

通过实验结果发现,  $K$ -means++ 算法在 LGR 推荐结果中的精度较其他算法有着一定的优势, 因此, 我们也将选取它作为 LGR 中潜在群组发现算法, 并与基准方法进行对比.



Table 2 RMSE of LGR in Different Group Size

表 2 LGR 在不同群组数量的 RMSE 值

Clustering Method	$N_g$					
	1	20	50	200	500	6 040
K-means	0.984 54	0.940 29	0.924 57	0.905 63	0.891 65	0.852 30
K-means++	0.984 54	0.933 92	0.921 02	0.904 88	0.888 89	0.852 30
Birch	0.984 54	0.948 15	0.935 04	0.910 81	0.892 26	0.852 30
AffinityPropagation	0.984 54		0.889 00( $N_g=465$ )			0.852 30

Predict&Cluster, PLGM 以及 LGR 在 MovieLens1M 数据集上的 RMSE 随群组数量  $N_g$  变化的效果对比如图 4 所示:

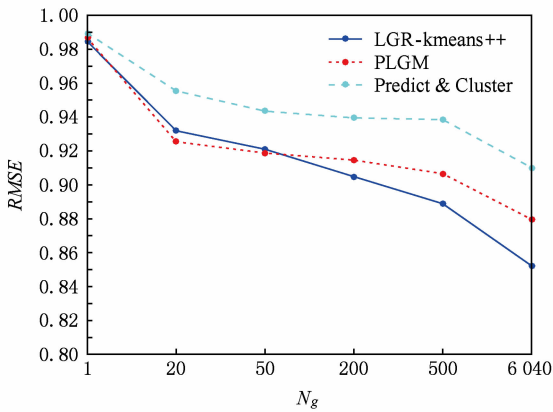


Fig. 4 RMSE in three methods ( $K=50$ , Average Strategy)

图 4  $K=50$  时 3 种方法均值策略的 RMSE

从图 4 中可以发现, LGR 不仅在个体推荐情景下的推荐准确度明显高于 Predict&Cluster 与 PLGM, 在群组推荐场景下, 也具有一定优势. 相对于 Predict&Cluster, LGR 的精度具有着较大优势, 这是因为 Predict&Cluster 聚合的是个体推荐结果, 这些预测的推荐结果维度较大, 偏好分布不明显. 在与同样是融合个体偏好模型的 PLGM 的对比过程中, 当群组数目  $N_g \leq 50$  时, LGR 的精度以微弱的劣势略逊于 PLGM, 但总体精度上依然存在着一定的优势.

### 3.2.2 影响推荐方法性能的主要参数分析

根据 3.2.1 节各聚类算法对 LGR 的性能表现的对比, 选取群组个数  $N_g=500$ , 并在上述设置的基础上分析用户偏好拟合系数  $\gamma$ 、正则化系数  $\lambda_u$  与  $\lambda_v$ 、潜在特征维数  $K$ 、词向量维数的变化对 RMSE 值的影响, 在展示一个影响因素的变化趋势时, 其他参数为最佳设置.

1) 用户偏好拟合系数  $\gamma$ . 设置  $\gamma=0, 0.1, 0.2, \dots, 1$ , 观察其变化对 RMSE 值的影响如图 5 所示:

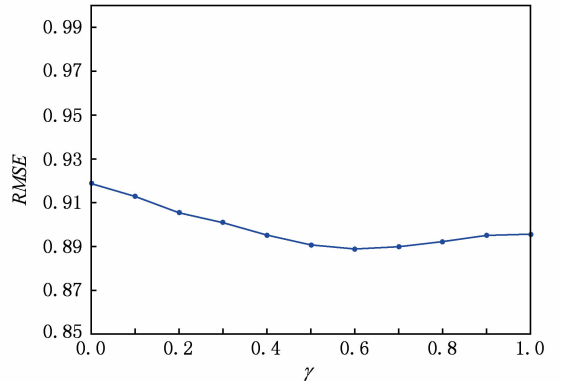


Fig. 5 RMSE in different  $\gamma$

图 5 不同  $\gamma$  下的 RMSE

2) 正则化系数  $\lambda_u$  与  $\lambda_v$ . 设置了  $\lambda_u=1, 10, 100, 1000, \lambda_v=1, 10, 100, 1000$ , 观察其变化对 RMSE 值的影响如图 6 所示:

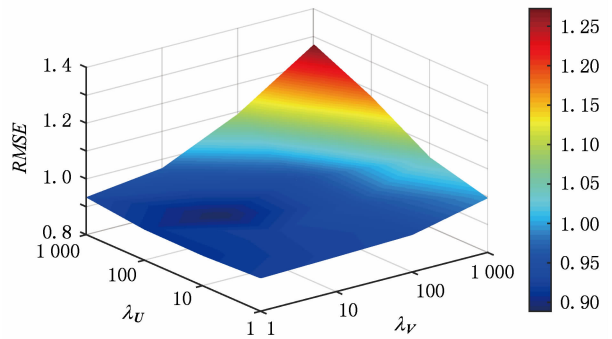


Fig. 6 RMSE in different  $\lambda_u$  and  $\lambda_v$

图 6 不同  $\lambda_u$  与  $\lambda_v$  下的 RMSE

3) 潜在特征维数  $K$ . 设置  $K=10, 20, 50, 100$ , 观察其变化对 RMSE 值的影响如图 7 所示.

4) 词向量维数  $K_w$ . 设置  $K_w=100, 200, 300$ , 观察其变化对 RMSE 值的影响如图 8 所示.

通过实验结果可以发现, 潜在特征维数  $K$  与词向量维数  $K_w$  在到达一定的数值以后对 RMSE 值几乎没有影响, 用户偏好拟合系数  $\gamma$  取值在  $0.5 \sim 0.7$  时 RMSE 有最小值, 正则化系数  $\lambda_u$  与  $\lambda_v$  分别

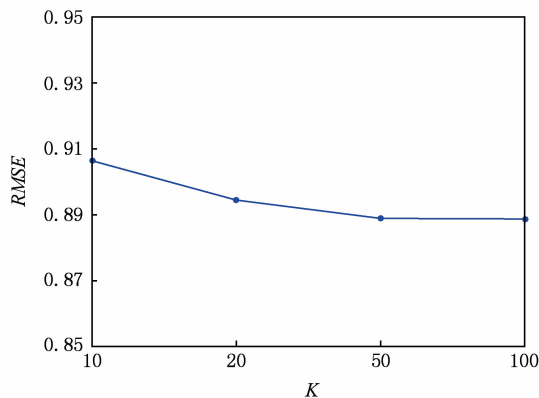


Fig. 7 RMSE in different K

图 7 不同 K 下的 RMSE

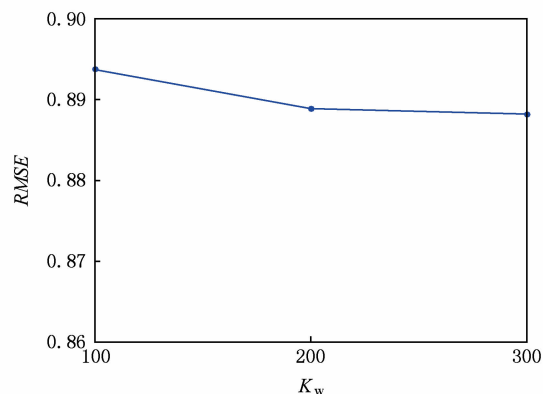


Fig. 8 RMSE in different K\_w

图 8 不同 K\_w 下的 RMSE

为 100 与 10 时 RMSE 出现最小值. 根据分析得出: 我们采用的最佳设置为  $\gamma=0.6$ ,  $\lambda_U=100$ ,  $\lambda_V=10$ ,  $K=50$ ,  $K_w=200$ .

## 4 总 结

我们提出了一种基于动态卷积概率矩阵分解模型(DPMFM-CNN), 以及将其融合于群组推荐框架的潜在群组推荐方法(LGR). DPMFM-CNN 模型结合服务描述文档、用户配置文件以及时间因素, 综合的分析了用户的静态与动态偏好; LGR 通过聚类用户偏好, 采用多种聚类算法发现用户集合中的潜在群组, 并采取均值策略进行群组偏好融合, 完成群组对服务评分的任务.

目前, 我们的工作主要关注用户的显式评分数据, 而这类数据并不是所有场景都可以获得的, 相比于用户的显式数据, 用户的隐式反馈如: 点击、评论、转发等行为更为容易收集. 因此运用用户隐式反馈数据分析用户兴趣将是一个值得关注的方向. 此外,

用户的兴趣迁移可以通过时间直观地反映出. 然而, 在这背后可能潜藏着更多的影响因素如: 地点、天气、交通、节假日等, 因此, 将用户的表示扩展到更高维度来分析用户的偏好也是一个值得进一步探讨的研究点.

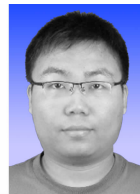
## 参 考 文 献

- [1] Zhuang Yujie, Du Yulu. Research on group recommendation system and their application [J]. Chinese Journal of Computers, 2016, 39(4): 745-764 (in Chinese) (张玉洁, 杜雨露. 组推荐系统及其应用与研究[J]. 计算机学报, 2016, 39(4): 745-764)
- [2] Feng Shanshan, Cao Jian, Wang Jie, et al. Group recommendations based on comprehensive latent relationship discovery [C] //Proc of IEEE Int on Web Services (ICWS 2016). Piscataway, NJ: IEEE, 2016: 9-16
- [3] Boratto L. Group recommender systems [C] //Proc of ACM Confon Recommender Systems (RecSys 2016). New York: ACM, 2016: 427-428
- [4] Baltrunas L, Makcinskas T, Ricci F. Group recommendations with rank aggregation and collaborative filtering [C] //Proc of ACM Confon Recommender Systems (RecSys 2010). New York: ACM, 2010: 119-126
- [5] Boratto L, Carta S, Chessa A, et al. Group recommendation with automatic identification of users communities [C] //Proc of 2009 IEEE/wic/ACM Int Joint Conf on Web Intelligence and Intelligent Agent Technologies. Piscataway, NJ: IEEE, 2009: 547-550
- [6] Boratto L, Carta S. Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender system [C] //Proc of the 16th Int Conf on Enterprise Information Systems (ICEIS 2014). Setubal: Scitepress, 2014: 564-572
- [7] Zeng Xuelin, Wu Bin, Shi Jing, et al. Parallelization of latent group model for group recommendation algorithm [C] //Proc of the 1st 2016 IEEE Int Conf on Data Science in Cyberspace (DSC 2016). Piscataway, NJ: IEEE, 2016: 80-89
- [8] Lecun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521(7553): 436-444
- [9] Mikolov T, Sutskever I, Chen Kai, et al. Distributed representations of words and phrases and their compositionality [C] //Proc of Neural Information Processing Systems (NIPS 2013). Cambridge, MA: MIT Press, 2013: 3111-3119
- [10] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation [C] //Proc of Conf on Empirical Methods in Natural Language Processing (EMNLP 2014). New York: ACM, 2014: 1532-1543
- [11] Shen Yelong, He Xiaodong, Gao Jianfeng, et al. Learning semantic representations using convolutional neural networks for Web search [C] //Proc of the 23rd Int Conf on World Wide Web (WWW 2014). New York: ACM, 2014: 373-374

- [12] He Hua, Gimpel K, Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks [C] // Proc of Conf on Empirical Methods in Natural Language Processing (EMNLP 2015). Stroudsburg, PA: ACL, 2015: 1576-1586
- [13] Kim Y. Convolutional neural networks for sentence classification [C] // Proc of the 2014 Conf on Empirical Methods in Natural Language Processing (EMNLP 2014). Stroudsburg, PA: ACL, 2014: 1474-1480
- [14] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles [C] // Proc of ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (SIGKDD 2011). New York: ACM, 2011: 448-456
- [15] Salakhutdinov R, Mnih A. Probabilistic matrix factorization [C] // Proc of the 20th Int Conf on Neural Information Processing Systems (NIPS 2007). Cambridge, MA: MIT Press, 2007: 880-887
- [16] Zhang Wei, Han Linyu, Zhang Dianlei, et al. GeoPMF: A distance-aware tour recommendation model [J]. Journal of Computer Research and Development, 2017, 54(2): 405-414 (in Chinese)  
(张伟, 韩林玉, 张佃磊, 等. GeoPMF: 距离敏感的旅游推荐模型[J]. 计算机研究与发展, 2017, 54(2): 405-414)
- [17] Roweis S, Ghahramani Z. A unifying review of linear gaussian models [J]. Neural Computation, 1999, 11(2): 305-345
- [18] Guang Ling, Yang Haiqin, Irwin K, et al. Online learning for collaborative filtering [C] // Proc of the 2012 Int Joint Conf on Neural Networks (IJCNN 2012). Piscataway, NJ: IEEE, 2012, 20: 1-8
- [19] Arthur D, Vassilvitskii S.  $k$ -means++: The advantages of careful seeding [C] // Proc of the 18th Annual ACM-SIAM Symp on Discrete Algorithms (SODA 2007). Philadelphia, PA: SIAM, 2007: 1027-1035
- [20] Zhang Tian, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases [J]. AcmSigmod Record, 1996, 25(2): 103-114
- [21] Frey B J, Dueck D. Clustering by passing messages between data points [J]. Science, 2007, 315(5814): 972-976
- [22] Boratto L, Carta S. Modeling the preferences of a group of users detected by clustering: A group recommendation case-study [C] // Proc of Int Conf on Web Intelligence, Mining and Semantics (WIMS 2014). New York: ACM, 2014



**Wang Haiyan**, born in 1974. PhD. Professor. Senior member of CCF. Her main research interests include service computing, trusted computing and big data intelligent processing technology.



**Dong Maowei**, born in 1993. Master candidate. His main research interests include service computing, deep learning (dongmaowei@gmail.com).