

基于隐含上下文支持向量机的服务推荐方法

赵晨阳¹, 王俊岭²

(1. 河南工业大学信息科学与工程学院, 河南 郑州 450001; 2. 河南工业大学理学院, 河南 郑州 450001)

摘 要: 结合上下文信息和支持向量机 (SVM), 提出了一种基于隐含上下文支持向量机的服务推荐方法。首先, 根据用户所处的不同上下文信息对用户评分矩阵进行修正, 使其带有隐含的上下文信息; 其次, 将带有隐含上下文信息的服务评分向量作为服务的特征向量, 构建训练集, 上下文信息的引入并没有增加服务特征向量的维数; 然后, 根据训练集使用 SVM 获得目标用户的分类超平面, 构建 SVM 预测模型; 最后, 计算目标用户未使用服务的特征向量点与超平面的距离, 综合考虑该距离以及相似用户的推荐, 做出服务推荐。实验结果表明, 所提推荐方法在不同的评分矩阵密度下均具有较好的推荐精度, 并且能够缩短推荐时间。

关键词: 服务推荐; 支持向量机; 隐含上下文; 评分矩阵; 推荐精度

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019190

Service recommendation method based on context-embedded support vector machine

ZHAO Chenyang¹, WANG Junling²

1. College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

2. College of Science, Henan University of Technology, Zhengzhou 450001, China

Abstract: Combined with contexts and SVM, a service recommendation method based on context-embedded support vector machine (SRM-CESVM) was proposed. Firstly, according to the different contexts, the user rating matrix was modified to make it with embedded contexts. Secondly, the rating vectors with embedded contexts were used as service feature vectors to construct training set, meanwhile the dimension of service feature vector were not increased by the introduction of contexts. Thirdly, a separation hyperplane for active user was acquired based on training set using SVM, and then the SVM prediction model was built. Finally, the distances between the feature vector points representing the active users' unused services and the hyperplane were calculated. Considering the distances and the recommendation of similar users, the service list was recommended. The experimental results further demonstrate that the proposed method has better recommendation accuracy under different rating matrix densities and can reduce recommendation time.

Key words: service recommendation, support vector machine, embedded context, rating matrix, recommendation accuracy

1 引言

Web 服务是自描述的软件实体, 可以通过互联网发布、定位和使用, 服务计算和云计算技术的快

速发展使 Web 服务成为 IT 资源共享的重要手段之一。然而, 随着具有相同功能而服务质量各异的服务在互联网上井喷式涌现, 用户很难找到他们需要的服务, 从而造成“信息过载”问题。推荐系统被

收稿日期: 2018-08-28; 修回日期: 2019-07-07

基金项目: 河南省科技厅自然科学基金资助项目 (No. 182102210388); 河南省高校科技创新团队支持计划基金资助项目 (No. 17IRTSTHN011)

Foundation Items: The Natural Science Project of Henan Science and Technology Department (No. 182102210388), The Program for Innovative Research Team in University of Henan Province (No. 17IRTSTHN011)

认为是解决该方法之一。

目前,国内外研究者对服务推荐问题已经做了大量的工作,并取得了一些重要的成果。根据推荐算法工作机制的不同,可以将当前主流的服务推荐方法分为基于内容的服务推荐、基于协同过滤的服务推荐、混合服务推荐、上下文感知的服务推荐及其他相关的服务推荐方法。

早期的服务推荐方法研究主要集中于基于内容的服务推荐。它的基本思想是主动提取用户已使用服务的特征,利用这些特征建立用户偏好模型,然后将待推荐服务的特性与用户偏好模型比较,找出符合用户偏好的服务向用户推荐。Shu 等^[1]根据用户对多媒体学习资源的个性化需求,提出一种基于内容的服务推荐方法。基于内容的推荐方法抽取某些服务的内容特征比较困难,因此其应用范围具有局限性。基于协同过滤的服务推荐方法不存在该问题,它是目前应用最为广泛和成功的方法。该推荐方法主要分为基于内存的方法和基于模型的方法。基于内存的方法包括基于用户的方法和基于服务的方法,两者都使用近邻技术预测用户未使用服务的评分,然后为用户推荐具有较高评分的服务。Li 等^[2]使用 MapReduce 大数据处理平台实现了一种基于服务的协同推荐算法,该算法与传统的协同过滤算法相比,在处理大数据集时具有较好的效率。Li 等^[3]提出了一种带有隐私保护的基于服务的协同过滤方法。Al-Hassan 等^[4]结合本体相似性和基于服务的协同过滤技术,提出了一种语义增强的混合旅游服务推荐方法。孟恒羽等^[5]提出一种基于改进 K 近邻模型的协同过滤推荐算法,将传统的协同过滤算法和改进的 K 近邻算法相结合。基于模型的服务推荐方法是另外一种类型的协同过滤推荐方法。该方法首先建立了一个基于用户历史数据的预测模型,然后利用该模型预测用户对未使用服务的偏好程度,从而为用户做出推荐。该方法采用的预测模型主要包括线性回归、支持向量机(SVM, support vector machine)、神经网络、贝叶斯模型、矩阵分解、深度学习等。李改等^[6]提出了一种基于评分预测与排序预测的协同过滤算法,该算法采用概率矩阵分解技术。Kushwaha 等^[7]提出一种基于概率矩阵分解的语义推荐系统。Ren 等^[8]提出了一种基于 SVM 的协同过滤服务推荐算法。Oku 等^[9]结合 SVM,提出了一种上下文感知的服务推荐方法。Ma 等^[10]利用用户偏好信息的差异提出了一种偏好

增强的 SVM,用于分类问题。Wei 等^[11]提出一种结合协同过滤方法和深度神经网络的电影服务推荐方法,用于解决新服务问题。黄立威等^[12]对基于深度学习的推荐系统研究进行了综述和比较。

为了发挥多种推荐方法的优势,近年来,许多研究者关注混合推荐方法,该方法使用 2 种或多种不同推荐技术,或者与其他技术相结合进行服务推荐。Yao 等^[13]考虑到用户相似性和服务语义内容的相似性,提出了一种混合服务推荐方法。Afify 等^[14]结合基于模型和基于内存的协同过滤方法,提出了一种混合的云服务推荐方法。同时,为了提高服务推荐的个性化和精度,近年来,出现了上下文感知服务推荐方法,这些上下文主要包括时间、位置、同伴、信任关系等信息。Zhao 等^[15]基于上下文和用户信任关系,提出一种视频推荐算法。Yin 等^[16]基于用户的信任模型,提出一种大数据云服务推荐方法。

虽然上述服务推荐方法在服务推荐领域做出了重要贡献,但仍然存在一些问题,主要表现在以下 3 个方面。第一,目前多数推荐方法对目标用户所有未使用服务的偏好值进行整体预测,这种预测包含了对目标用户不感兴趣服务的预测,这是不必要的,因此这种基于预测的推荐方法往往比较耗时。第二,多数推荐方法基于用户的历史数据进行推荐,事实上,用户的历史数据与特定的上下文信息存在一定的关联,因此,推荐方法需要引入上下文信息以提高推荐精度。第三,多数推荐方法在计算用户相似度时,仅使用用户的历史行为数据,很少使用用户的人口统计学信息,因此,在推荐方法中需要引入用户人口统计学信息以提高相似度计算精度,并缓解冷启动问题。

为了解决上述问题,本文借鉴现有的研究成果,提出了一种基于隐含上下文支持向量机的服务推荐方法(SRM-CESVM, service recommendation method based on context-embedded support vector machine)。与其他推荐方法相比,本文所提方法具有以下特点。第一,根据上下文信息,修正用户评分矩阵,使其带有隐含的上下文信息;将带有上下文信息的服务评分向量作为服务的特征向量,使用 SVM 构造用户分类超平面对服务进行分类。采用隐含上下文信息的服务特征向量不仅使上下文信息在服务分类中起到指导作用,而且不会增加服务特征向量的维数,因此不会增加支持向量机的计算复杂度。第二,分类超平面将用户不感兴趣的服务从

待推荐服务集合中分离出来，在计算用户对未评分服务的偏好程度，即预测未评分服务的评分时，可以只考虑用户感兴趣的未评分服务，从而极大地缩短了算法的计算时间。第三，考虑从目标用户的历史评分数据和相似用户 2 个方面为目标用户做出服务推荐。首先，根据 SVM 预测模型选出目标用户较为感兴趣的候选推荐服务集合；然后，根据目标用户的相似用户的服务推荐得出另一个候选推荐服务集合；最后，综合 2 个候选推荐服务集合得出最终的服务推荐集合。

2 SRM-CESVM 模型

SRM-CESVM 模型如图 1 所示。用户使用服务后，通常会显式地为服务给出一个评价分数，以此表达该用户对已使用服务的满意程度，从而形成用户评分数据。对于某目标用户，服务集合被分为两部分：用户使用过的服务集合（已评分服务集合）和用户未使用的服务集合（未评分服务集合）。

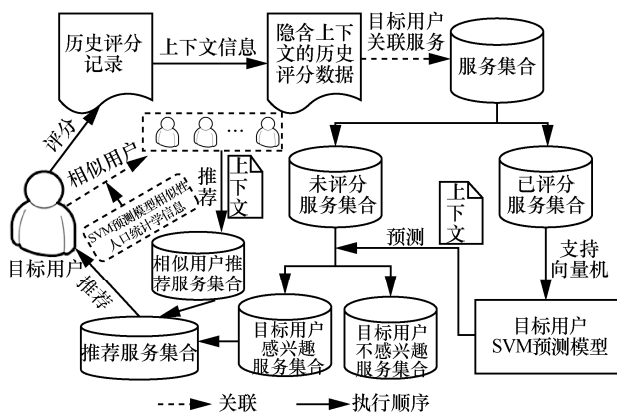


图 1 SRM-CESVM 模型

一方面，将目标用户已评分服务集合作为训练集，利用 SVM 构造目标用户的 SVM 预测模型。然后，结合目标用户推荐时所处的上下文信息，根据分类超平面，将未评分服务集合分为 2 个部分：用户感兴趣的服务集合和用户不感兴趣的服务集合。对于目标用户不感兴趣的服务集合直接过滤，不需要预测目标用户对这些服务的评分值。而对于用户感兴趣的服务集合，则结合服务所对应的服务特征向量点与超平面的距离预测目标用户对这些服务的偏好程度，给出预测值，并根据预测值的大小排序，排名靠前的服务组成一个候选推荐服务集合。

另一方面，根据目标用户与其他用户的 SVM 预测模型的相似性及目标用户与其他用户的人口统

计学信息，计算目标用户与其他用户的相似度，得出目标用户的最近邻相似用户集合。结合目标用户推荐时所处的上下文信息，将相似用户感兴趣且已使用的，同时目标用户未使用的服务组成目标用户的另一个候选推荐服务集合。最后，综合两类候选推荐服务集合，为目标用户给出合理的服务推荐。

3 SRM-CESVM 服务推荐方法的实现

本节详细描述 SRM-CESVM。首先，按照执行的先后顺序介绍方法的具体步骤；然后，对推荐方法进行整体描述。

3.1 用户评分矩阵

用户评分矩阵记录了用户对服务的评分数据。设推荐系统中存在 m 个用户和 n 个服务，则用户评分矩阵的定义如式(1)所示。

$$R_{m \times n} = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & r_{2,3} & \cdots & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & r_{m,3} & \cdots & r_{m,n} \end{pmatrix} \quad (1)$$

其中， $r_{i,j}$ 为用户 u_i 对服务 s_j 的评分， $0 \leq i \leq m$ ， $0 \leq j \leq n$ 。 $r_{i,j} = 0$ ，表示用户 u_i 没有使用过服务 s_j ，即对服务 s_j 没有进行过评分，服务 s_j 属于用户 u_i 的待推荐服务集合； $r_{i,j} > 0$ ，表示用户 u_i 使用过服务 s_j ，即对服务 s_j 进行过评分，其值越大说明用户对该服务越满意，服务 s_j 属于用户 u_i 的训练服务集合。

由于用户评分的服务数量一般比较稀少，用户评分矩阵中存在大量的零值，因此用户评分矩阵通常十分稀疏，其稀疏程度用矩阵中非零值所占比例即用户评分矩阵密度来度量。从而，用户评分数据越稀疏，零值越多，用户评分矩阵密度越小。

3.2 上下文信息与相似度计算

在用户评分矩阵中，用户对服务的评分是在一定的上下文环境下做出的^[15]，例如，使用服务的时间、用户所处的地点、当时的天气情况和用户的人口统计学信息（性别、年龄、职业、教育背景等），本文称这些上下文信息的集合为“评分上下文”。对于目标用户，如果相似用户对某个待推荐服务的评分较高，并且评分上下文信息与目标用户推荐时所处的上下文信息十分相似，那么推荐系统为目标用户推荐该服务更加具有说服力。推荐系统为目标用户推荐服务时用户所处的上下文信息集合，本文

称之为“推荐上下文”或“实时上下文”。因此，需要修正用户评分数据，使其带有隐含的上下文信息，使上下文信息在服务分类中起到指导作用，从而进一步提高推荐精度。

上下文信息可以通过各种传感器、公共服务平台或推荐服务系统本身提供。在本文实验部分，主要采用 MovieLens 数据集已收集的上下文信息进行相关实验。根据其取值的数据类型，上下文信息主要分为离散型上下文信息（例如教育背景、职业等）和连续型上下文信息（例如时间、年龄、温度等）。本节对上下文信息进行形式化描述，并提出上下文信息相似度的计算方法。

假设推荐系统中有 L 种不同类型的上下文信息，记 $c_{i,j} = (c_{i,j}^1, c_{i,j}^2, \dots, c_{i,j}^L)$ ($1 \leq i \leq m, 1 \leq j \leq n$) 为用户 u_i 使用服务 s_j 时的评分上下文信息，其中， $c_{i,j}^k$ ($1 \leq k \leq L$) 为第 k 种类型的上下文信息。记 $c_{\text{current}} = (c_{\text{current}}^1, c_{\text{current}}^2, \dots, c_{\text{current}}^L)$ 是系统为目标用户推荐服务时目标用户所处的推荐上下文信息，其中， c_{current}^k ($1 \leq k \leq L$) 为第 k 种类型的上下文信息。定义评分上下文信息 $c_{i,j}$ 和推荐上下文信息 c_{current} 的相似度计算方法如式(2)所示。

$$\text{sim}(c_{i,j}, c_{\text{current}}) = \frac{\sum_{k=1}^L \text{sim}_k(c_{i,j}^k, c_{\text{current}}^k)}{L} \quad (2)$$

其中， $\text{sim}_k(c_{i,j}^k, c_{\text{current}}^k)$ 是第 k 种类型的上下文信息 $c_{i,j}^k$ 和 c_{current}^k 的相似度。

如果第 k 种类型的上下文信息是离散型数据，根据上下文信息的特点，有 2 种相似度的计算方法。如果上下文信息的所有取值具有一定的相关性，则上下文信息相似度计算方法如式(3)所示。

$$\text{sim}_k(c_{i,j}^k, c_{\text{current}}^k) = 1 - \frac{|D(c_{i,j}^k) - D(c_{\text{current}}^k)|}{\max(D)} \quad (3)$$

其中， $D(c_{i,j}^k)$ 是 $c_{i,j}^k$ 的上下文信息值， $D(c_{\text{current}}^k)$ 是 c_{current}^k 的上下文信息值， $\max(D)$ 是所有上下文信息值的最大值。如果上下文信息的所有取值没有一定的相关性，则上下文信息相似度的计算方法如式(4)所示。

$$\text{sim}_k(c_{i,j}^k, c_{\text{current}}^k) = \begin{cases} 1, & D(c_{i,j}^k) = D(c_{\text{current}}^k) \\ 0, & D(c_{i,j}^k) \neq D(c_{\text{current}}^k) \end{cases} \quad (4)$$

例如，对于离散型上下文信息“教育背景”，规定 $D=(1,2,3,4,5,6)$ 分别对应教育程度（小学，初中，

高中，本科，硕士，博士），这些值之间具有相关性，因此采用式(3)的计算方法，例如，博士和小学的上下文相似度为 $\text{sim}_k(\text{博士}, \text{小学}) = 1 - \frac{|6-1|}{6} = \frac{1}{6}$ ，而

$\text{sim}_k(\text{博士}, \text{硕士}) = 1 - \frac{|6-5|}{6} = \frac{5}{6}$ 。其他类似的上下文信息（如天气情况等），相似度计算方法与此类似。

对于另一种离散型上下文信息“职业”，规定 $D=(1,2,3,4,5,6,7,8)$ 分别对应国家职业大类划分的 8 个类，这些值之间没有相关性，因此采用式(4)的计算方法，例如，专业技术人员与商业人员的上下文相似度为 0。对其他类似上下文信息（如性别等），相似度计算方法与此类似。

如果第 k 种类型的上下文信息是连续型数据，首先将连续数据离散化，然后采用离散型上下文信息相似度的计算方法进行计算。离散化方法如式(5)所示。

$$D(c_{i,j}^k) = \begin{cases} 1, & 0 \leq c_{i,j}^k < T_1 \\ 2, & T_1 \leq c_{i,j}^k < T_2 \\ 3, & T_2 \leq c_{i,j}^k < T_3 \\ \vdots & \vdots \\ N, & T_{N-1} \leq c_{i,j}^k < T_N \\ N+1, & T_N \leq c_{i,j}^k \end{cases} \quad (5)$$

其中， T_i ($1 \leq i \leq N$) 为分段阈值点，离散化后，上下文信息相似度的计算方法如式(3)所示。

例如，连续型上下文信息“时间”，根据距离当前时间的间隔，以 6 个月为段，将时间离散化为 $D=(1, 2,3,4,5,6,7)$ ，分别对应时间段(1~6 个月，7~12 个月，13~18 个月，19~24 个月，25~30 个月，31~36 个月，37 个月及以上)，则 $\text{sim}_k(25\sim30\text{个月}, 13\sim18\text{个月}) = 1 - \frac{|5-3|}{7} = \frac{5}{7}$ ，而 $\text{sim}_k(37\text{个月及以上}, 1\sim6\text{个月}) = 1 - \frac{|7-1|}{7} = \frac{1}{7}$ 。对其他类似连续型上下文信息（如温度等），相似度计算方法与此类似。

3.3 服务评分向量

设用户 u_i 为目标用户，为便于描述，不失一般性，设推荐系统中所有服务组成的集合中前 K 个服务属于目标用户已评分服务集合，记为 $S_r = \{s_j | 1 \leq j \leq K\}$ ，相应地， $S_{ur} = \{s_j | K+1 \leq j \leq n\}$ 为目标用户未评分服务集合。

对于任意服务 $s_j \in S_r$ ，所有用户对该服务的评分形成该服务的评分向量 $(r_{1,j}, r_{2,j}, \dots, r_{k,j}, \dots, r_{m,j})$ ，

其分量 $r_{k,j} (1 \leq k \leq m)$ 是用户 u_k 对服务 s_j 的评分，由于目标用户对该服务的评分 $r_{i,j}$ 的值是确定的，即偏好确定，因此将该服务作为目标用户的训练服务。

对于任意服务 $s_j \in S_{ur}$ ，所有用户对该服务的评分形成服务的评分向量 $(r_{1,j}, r_{2,j}, \dots, r_{k,j}, \dots, r_{m,j})$ ，目标用户对该服务的评分 $r_{i,j}$ 的值是未知的，即没有使用过该服务，因此，将该服务作为目标用户的待推荐服务。

3.4 隐含上下文用户评分矩阵

利用上下文信息的相似度修正目标用户评分矩阵，使其具有隐含的上下文信息，具体方法如下。

对于目标用户 u_i ，从目标用户已评分服务和未评分服务 2 个方面对用户评分矩阵进行修正。对于任意的已评分服务 $s_j \in S_r$ ，其对应的评分向量修正时，上下文相似度计算中采用的上下文信息是目标用户使用该服务的评分上下文和其他用户使用该服务的上下文。对于任意的未评分服务 $s_j \in S_{ur}$ ，其对应的评分向量修正时，上下文相似度计算中采用的上下文信息是为目标用户推荐该服务时的推荐上下文和其他用户使用该服务时的上下文。本文实验中使用的上下文（评分上下文和推荐上下文）来自公开的 MovieLens 数据集。本文提出的上下文修正方法的思想借鉴于蚁群算法中路径信息素的更新规则，即保留原始评分数据信息的同时，在评分数据中增加上下文信息，这样既保留了原始评分信息，又融入了新的上下文因素用于推荐。

目标用户已评分服务 $s_j \in S_r$ 对应的评分向量的修正过程如下。对于目标用户 u_i ，如果服务 s_j 是训练服务，即 $s_j \in S_r$ ，说明目标用户对该服务做出过评分，该评分具有评分上下文信息。因此，以目标用户 u_i 使用服务 s_j 时的评分上下文信息 $c_{i,j}$ 为参照，修改服务 s_j 对应的评分向量 $(r_{1,j}, r_{2,j}, \dots, r_{k,j}, \dots, r_{m,j})$ ，其修正规则如式(6)所示。

$$r_{k,j}^* = \begin{cases} r_{k,j}, & r_{k,j} \neq 0, u_k = u_i \\ \rho r_{k,j} + (1 - \rho) r_{k,j} \text{sim}(c_{k,j}, c_{i,j}), & r_{k,j} \neq 0, u_k \neq u_i \\ 0, & r_{k,j} = 0, u_k \neq u_i \end{cases} \quad (6)$$

其中， $\rho \in [0,1]$ 为遗留因子，表示原始评分信息的保留程度； $\text{sim}(c_{k,j}, c_{i,j})$ 为用户 u_k 评分服务 s_j 时的评分上下文信息 $c_{k,j}$ 和目标用户 u_i 评分服务 s_j 时的评分上下文信息 $c_{i,j}$ 的相似度，其计算方法如式(2)

所示。修正后，目标用户的已评分服务 $s_j \in S_r$ 的修正评分向量记为 $(r_{1,j}^*, r_{2,j}^*, \dots, r_{k,j}^*, \dots, r_{m,j}^*)$ 。

目标用户未评分服务 $s_j \in S_{ur}$ 对应的评分向量的修正过程如下。对于目标用户 u_i ，如果服务 s_j 是待推荐服务，即 $s_j \in S_{ur}$ ，说明目标用户对该服务未做出过评分。应该比较其他用户使用服务 s_j 时的评分上下文信息 $c_{i,j}$ 与目标用户 u_i 推荐时所处的上下文信息 $c_{current}$ 之间的相似性，因此，以目标用户 u_i 推荐时所处的上下文信息 $c_{current}$ 为参照，修改服务 s_j 对应的评分向量 $(r_{1,j}, r_{2,j}, \dots, r_{k,j}, \dots, r_{m,j})$ ，其修正规则如式(7)所示。

$$r_{k,j}^* = \begin{cases} 0, & r_{k,j} = 0, u_k = u_i \\ \rho r_{k,j} + (1 - \rho) r_{k,j} \text{sim}(c_{k,j}, c_{current}), & r_{k,j} \neq 0, u_k \neq u_i \\ 0, & r_{k,j} = 0, u_k \neq u_i \end{cases} \quad (7)$$

其中， $\rho \in [0,1]$ 为遗留因子， $\text{sim}(c_{k,j}, c_{current})$ 为用户 u_k 评分服务 s_j 时的评分上下文信息 $c_{k,j}$ 和目标用户 u_i 推荐时所处的上下文信息 $c_{current}$ 的相似度，其计算方法如式(2)所示。修正后，目标用户的未评分服务 $s_j \in S_{ur}$ 的修正评分向量记为 $(r_{1,j}^*, r_{2,j}^*, \dots, r_{k,j}^*, \dots, r_{m,j}^*)$ 。

通过上述 2 个方面对用户评分矩阵的修正，使用户对服务的评分隐含了上下文信息，有助于推荐精度的提高。修正后的隐含上下文用户评分矩阵如式(8)所示。

$$R_{m \times n}^* = \begin{pmatrix} (r_{1,1}^* | c_{1,1}) & (r_{1,2}^* | c_{1,2}) & (r_{1,3}^* | c_{1,3}) & \dots & (r_{1,n}^* | c_{1,n}) \\ (r_{2,1}^* | c_{2,1}) & (r_{2,2}^* | c_{2,2}) & (r_{2,3}^* | c_{2,3}) & \dots & (r_{2,n}^* | c_{2,n}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (r_{m,1}^* | c_{m,1}) & (r_{m,2}^* | c_{m,2}) & (r_{m,3}^* | c_{m,3}) & \dots & (r_{m,n}^* | c_{m,n}) \end{pmatrix} \quad (8)$$

其中， $(r_{i,j}^* | c_{i,j})$ 为在上下文信息 $c_{i,j}$ 条件下，用户 u_i 对服务 s_j 的修正评分 $r_{i,j}^*$ ，即隐含上下文信息的评分。

隐含上下文用户评分矩阵之所以称为“隐含”的，其目的是不仅使上下文信息在服务分类时起到指导作用，而且将修正后的服务评分向量作为服务特征向量用于训练和推荐时，不增加服务特征向量的维数和支持向量机的计算复杂度。

3.5 训练服务集合和待推荐服务集合

如上文所述，对于目标用户 u_i ，为便于描述，不失一般性，将服务集合中前 K 个服务设为目标用户已评分服务，它们属于集合 S_r ；后 $n-K$ 个服务

设为目标用户未评分服务，它们属于集合 S_{ur} 。

如图 2 所示，将所有属于 S_r 的目标用户已评分的服务组成训练服务集。对于任意服务 $s_j \in S_r$ ，采用 3.4 节所述的修正方法得到修正评分向量 $(r_{1,j}^*, r_{2,j}^*, \dots, r_{k,j}^*, \dots, r_{m,j}^*)$ ，该修正评分向量作为该训练服务的特征向量，记为 \mathbf{sv}_j^r ，用于训练目标用户的 SVM 预测模型。于是，对于目标用户，记 $TS_i = \{(\mathbf{sv}_j^r, y_{i,j}) | 1 \leq j \leq K\}$ 为训练服务集合，其中， $y_{i,j} \in \{1, -1\}$ 是服务 s_j 是否被目标用户感兴趣的标签，“1”表示感兴趣，“-1”表示不感兴趣。设 ε 为推荐系统设定的用户偏好阈值，如果目标用户 u_i 对服务 s_j 的评分 $r_{i,j}^* \geq \varepsilon$ ，则 $y_{i,j} = 1$ ；否则 $y_{i,j} = -1$ 。

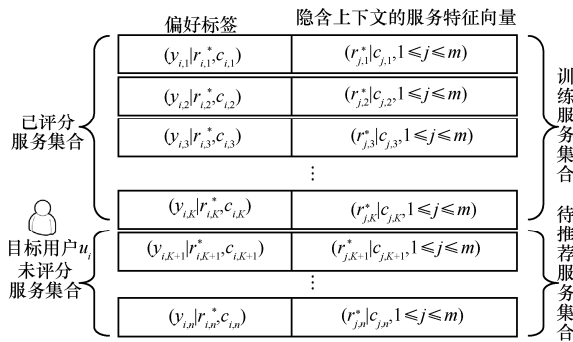


图 2 训练服务集合和待推荐服务集合

相应地，将所有属于 S_{ur} 的目标用户未评分的服务组成待推荐服务集合。对于任意服务 $s_j \in S_{ur}$ ，采用 3.4 节所述的修正方法得到修正评分向量 $(r_{1,j}^*, r_{2,j}^*, \dots, r_{k,j}^*, \dots, r_{m,j}^*)$ ，该修正评分向量作为该待推荐服务的特征向量，记为 \mathbf{sv}_j^u 。于是，对于目标用户，记 $RS_i = \{(\mathbf{sv}_j^u, y_{i,j}) | K+1 \leq j \leq n\}$ 为待推荐服务集合，由于 $r_{i,j}^* = 0$ ，故不能判断 $y_{i,j}$ 的值，需要根据训练好的 SVM 预测模型评估目标用户 u_i 对该服务的偏好程度。

3.6 隐含上下文支持向量机

对于目标用户 u_i ，获得训练服务集 TS_i 后，SRM-CESVM 服务推荐方法使用 SVM 为目标用户构建分类超平面，求解过程本质上是求解凸二次规划问题，如式(9)所示。

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{j=1}^K \xi_j \\ \text{s.t.} \quad & y_{i,j} (w \mathbf{sv}_j^r + b) \geq 1 - \xi_j \\ & \xi_j \geq 0, j = 1, 2, \dots, K \end{aligned} \quad (9)$$

其中，惩罚参数 $C > 0$ ， C 越大对误分类的惩罚越大； $\xi_j > 0$ 是松弛变量； K 是训练服务集合 TS_i 中服务的数量； w 和 b 为分类超平面的法向量和截距。

通常，SVM 采用核函数的方法，将数据从低维的数据空间映射到高维的数据空间，最终求解得到如式(10)所示的分类超平面。

$$f_i(\mathbf{sv}) = \sum_{p=1}^{n_{spv}} \alpha_p y_{i,p} \varphi(\mathbf{sv}, \mathbf{sv}_p) + b \quad (10)$$

其中， $f_i(\mathbf{sv})$ 是目标用户 u_i 的分类超平面， $\varphi(\mathbf{sv}, \mathbf{sv}_p)$ 是核函数， α_p 是支持向量 \mathbf{sv}_p 对应的拉格朗日乘子， n_{spv} 是支持向量的数量。

这样，通过分类超平面 $f_i(\mathbf{sv})$ 得到了目标用户的 SVM 预测模型。

3.7 基于 SVM 预测模型的服务推荐

在 SVM 预测模型中，根据目标用户未使用服务的特征向量，通过 $f_i(\mathbf{sv})$ 可以将该服务归属于目标用户感兴趣的服务集合 PRS_i 或不感兴趣的服务集合 NRS_i 。

具体地，对于目标用户 u_i 的任意一个未评分服务 s_j ，其对应的服务特征向量为 \mathbf{sv}_j^u 。如果 $f_i(\mathbf{sv}_j^u) \leq 0$ ，则说明目标用户对该服务不感兴趣，将该服务从待推荐服务集合中删除，不再计算其用户偏好值，这样可以极大地缩减算法的执行时间。如果 $f_i(\mathbf{sv}_j^u) > 0$ ，说明目标用户对该服务感兴趣。

目标用户对该服务的偏好程度可以用代表该服务的特征向量点到分类超平面的距离表示。服务的评分向量点离超平面距离越远，表示目标用户对服务越感兴趣。根据距离的远近对服务特征向量点进行排序，然后选出距离超平面距离较远的 N 个服务特征向量点对应的服务组成目标用户基于 SVM 预测模型的候选推荐服务集合，记为 CS_{svm}^i 。

3.8 基于相似用户的服务推荐

目标用户 u_i 与相似用户具有相似的偏好，因此考虑相似用户的推荐也很重要，其过程如下：首先，计算目标用户与其他用户之间的相似度，找出比较相似的用户集合；然后，将相似用户感兴趣且目标用户没有使用过的服务集合推荐给目标用户。在本文所提推荐方法中，用户相似度的计算通过 SVM 预测模型和用户人口统计学信息加权获得。用户人口统计学信息是计算用户相似度时需要考虑的重要因素，并且在用户评分数据稀疏或新用户问题

中，这些信息起到十分关键的作用，能够一定程度上提高推荐精度和有效地缓解算法的冷启动问题。具体步骤如下。

首先，介绍基于 SVM 预测模型的用户相似度计算方法。Oku 等^[9]提出用户相似度可以通过计算用户的 SVM 模型的相似性获得，本文借鉴该方法并做出一些改进，提出目标用户 u_i 和其他用户 u_j 关于 SVM 预测模型的用户相似度 $\text{sim}_{\text{svm}}(u_i, u_j)$ 的计算方法，如式(11)所示。

$$\text{sim}_{\text{svm}}(u_i, u_j) = \frac{1}{2} \left(\frac{m_1^+ + m_2^-}{m_1 + m_2} \times \frac{1}{\lambda_1} + \frac{n_1^+ + n_2^-}{n_1 + n_2} \times \frac{1}{\lambda_2} \right) \quad (11)$$

其中， $\lambda_1 = 1 + e^{-\frac{m_1^+ + m_2^-}{2}}$ ， $\lambda_2 = 1 + e^{-\frac{n_1^+ + n_2^-}{2}}$ ， m_1 和 m_2 分别是目标用户 u_i 的 SVM 预测模型中该用户感兴趣的服务个数和不感兴趣的服务个数， m_1^+ 是目标用户 u_i 的 SVM 预测模型中该用户感兴趣的服务对应的特征向量映射到用户 u_j 的 SVM 预测模型中同样是用户 u_j 感兴趣的服务的个数， m_2^- 是目标用户 u_i 的 SVM 预测模型中该用户不感兴趣的服务对应的特征向量映射到用户 u_j 的 SVM 预测模型中同样是用户 u_j 不感兴趣的服务的个数。同理， n_1 、 n_2 、 n_1^+ 、 n_2^- 是相对于用户 u_j 而言的。另外，该相似度计算方法在利用 SVM 预测模型计算目标用户 u_i 和用户 u_j 相似度时，还考虑了两者共同感兴趣和共同不感兴趣的服务数量。原因在于，在计算用户相似度时，不仅要考虑 2 个用户共同偏好服务数量的比例关系，还需要考虑服务数量，较多数量的共同偏好服务意味着用户相似度的计算更精确。

然后，介绍基于人口统计学信息用户相似度计算方法。人口统计学信息是指用户的年龄、性别、教育程度、职业等一系列自然与社会属性。这些信息分为离散型信息和连续型信息。首先需要数值化，具体方法类似于式(3)~式(5)的处理方法，最后采用式(12)所示的计算方法得出目标用户 u_i 和用户 u_j 关于人口统计学信息的相似度 $\text{sim}_{\text{dem}}(u_i, u_j)$ 。

$$\text{sim}_{\text{dem}}(u_i, u_j) = \frac{\sum_{k=1}^H \text{sim}_k(u_i, u_j)}{H} \quad (12)$$

其中， H 为用户人口统计学信息的个数， $\text{sim}_k(u_i, u_j)$ 为用户 u_i 与用户 u_j 的第 k 种类型的人口统计学信息的相似度。

通过上述方法，获得基于 SVM 预测模型的用户相似度和基于用户人口统计学信息的相似度，根据式(13)得到目标用户 u_i 和用户 u_j 的相似度。特殊情况下，当目标用户的训练服务数量极少或为零时，目标用户与其他用户的相似度计算式退化为基于用户人口统计学信息的相似度。

$$\text{sim}(u_i, u_j) = \frac{2\text{sim}_{\text{svm}}(u_i, u_j)\text{sim}_{\text{dem}}(u_i, u_j)}{\text{sim}_{\text{svm}}(u_i, u_j) + \text{sim}_{\text{dem}}(u_i, u_j)} \quad (13)$$

根据式(13)得到目标用户 u_i 的最近邻用户，组成目标用户的相似用户集合。对于每一个相似用户 u_j ，在其 SVM 预测模型中，首先，选取服务特征向量点距离分类超平面距离大于零且该用户感兴趣同时评分过的服务。其次，从最近邻用户选取的服务集合中删除重复的服务，形成一个临时服务集合。然后，对于属于该临时集合的任何一个服务 s_j ，如果该服务的评分上下文和目标用户当前所处上下文相似且该服务不属于集合 CS_{svm}^i ，则将该服务映射到目标用户 u_i 的 SVM 预测模型中，如果目标用户也对该服务感兴趣，则计算服务 s_j 的特征向量点到超平面 $f_i(\mathbf{sv})$ 的距离。最后，根据距离的远近进行排序，选出前 N 个服务，得到目标用户 u_i 的相似用户推荐的 N 个服务，这些服务组成目标用户 u_i 基于相似用户的候选推荐服务集合，记为 CS_{sim}^i 。

3.9 最终服务推荐

SRM-CESVM 服务推荐方法从 2 个方面为目标用户做出最终服务推荐。这 2 个方面分别来自基于 SVM 预测模型的服务推荐和基于相似用户的服务推荐。

基于上述 2 个候选推荐服务集合 CS_{svm}^i 和 CS_{sim}^i ，最终的候选推荐服务集合 CS^i 中的 N 个服务的选取方法如式(14)所示。

$$N(\text{CS}^i) = \delta N(\text{CS}_{\text{svm}}^i) + (1 - \delta)N(\text{CS}_{\text{sim}}^i) \quad (14)$$

其中， $N(\text{CS}^i)$ 、 $N(\text{CS}_{\text{svm}}^i)$ 和 $N(\text{CS}_{\text{sim}}^i)$ 分别是 3 个集合中服务的个数， δ 是选择权重参数。 CS^i 中的服务一部分来源于 CS_{svm}^i 中的前 $\delta N(\text{CS}_{\text{svm}}^i)$ 个服务，另一部分来源于 CS_{sim}^i 中的前 $(1 - \delta)N(\text{CS}_{\text{sim}}^i)$ 个服务。

最终的推荐结果来源于两方面。一方面，基于目标用户的历史评分数据，利用 SVM 预测模型为目标用户进行推荐，具有很好的解释性，用户更加

信服。另一方面，目标用户与相似用户具有相似的偏好，因此，根据相似用户的偏好为目标用户进行推荐，同样具有很好的解释性。值得注意的是，在特别情况下，只采用基于人口统计学信息的相似用户的推荐，这是因为当目标用户的训练服务数量极少或为零时，基于 SVM 预测模型的服务推荐精度严重下降或者不起作用，为了缓解数据稀疏性和冷启动，最终候选推荐服务集合全部来源于基于相似用户的推荐。

3.10 SRM-CESVM 算法

SRM-CESVM 算法描述如算法 1 所示。

算法 1 SRM-CESVM 推荐方法

输入 评分矩阵 $R_{m \times n}$ ，上下文 CT，偏好阈值 ε ，遗留因子 ρ ，选择权重 δ ，推荐服务数量 N

输出 Top- N 推荐服务集合 CS^i

- 1) 初始化 ε 、 ρ 、 δ 和 N
- 2) for 服务 s_j in 已评分服务集合 S_r do
- 3) if $r_{k,j} \neq 0$ and $u_k = u_i$ then
- 4) $r_{k,j}^* \leftarrow r_{k,j}$
- 5) else if $r_{k,j} \neq 0$ and $u_k \neq u_i$ then
- 6) $r_{k,j}^* \leftarrow \rho r_{k,j} + (1 - \rho)r_{k,j} \text{sim}(c_{k,j}, c_{i,j})$
- 7) else $r_{k,j}^* \leftarrow 0$
- 8) end if
- 9) end for
- 10) for 服务 s_j in 未评分服务集合 S_{ur} do
- 11) if $r_{k,j} \neq 0$ and $u_k \neq u_i$ then
- 12) $r_{k,j}^* \leftarrow \rho r_{k,j} + (1 - \rho)r_{k,j} \text{sim}(c_{k,j}, c_{\text{current}})$
- 13) else
- 14) $r_{k,j}^* \leftarrow 0$
- 15) end if
- 16) end for
- 17) 创建训练服务集合 TS_i 和推荐服务集合 RS_i
- 18) for 服务 s_j in 训练服务集合 TS_i do
- 19) if $r_{i,j}^* \geq \varepsilon$ then
- 20) $y_{i,j} = 1$
- 21) else
- 22) $y_{i,j} = -1$
- 23) end if
- 24) end for

$$25) \text{ 获得分类超平面 } f_i(\mathbf{sv}) \leftarrow \sum_{p=1}^{n_{\text{spv}}} \alpha_p \mathcal{Y}_{i,p}$$

$$\varphi(\mathbf{sv}, \mathbf{sv}_p) + b$$

- 26) for 服务 s_j in 推荐服务集合 RS_i do
- 27) if $f_i(\mathbf{sv}_j^{\text{ur}}) > 0$ then
- 28) $PRS_i \leftarrow PRS_i \cup \{s_j\}$
- 29) else
- 30) $NRS_i \leftarrow NRS_i \cup \{s_j\}$
- 31) end if
- 32) end for
- 33) for 服务 s_j in 感兴趣的服务集合 PRS_i do
- 34) 计算 s_j 对应的特征向量点到 $f_i(\mathbf{sv})$ 的距离
- 35) end for
- 36) 距离排序并选出前 N 个服务加入候选推荐服务集合 CS_{svm}^i 中
- 37) for 用户 $u_j (j \neq i)$ in 系统用户 do
- 38) $\text{sim}(u_i, u_j) \leftarrow \frac{2\text{sim}_{\text{svm}}(u_i, u_j)\text{sim}_{\text{dem}}(u_i, u_j)}{\text{sim}_{\text{svm}}(u_i, u_j) + \text{sim}_{\text{dem}}(u_i, u_j)}$
- 39) end for
- 40) 根据相似用户得到 N 个服务并加入候选推荐服务集合 CS_{sim}^i 中
- 41) $N(CS^i) \leftarrow \delta N(CS_{\text{svm}}^i) + (1 - \delta)N(CS_{\text{sim}}^i)$
- 42) 返回 Top- N 推荐服务集合 CS^i

4 实验结果及分析

4.1 实验数据集

实验采用 MovieLens 100K 数据集，其包括 943 个用户对 1 682 部电影的评分数据（评分数据为 1~5 分）。实验中，随机选择 80% 的数据构建训练集，剩余 20% 的数据构建测试集。该数据集还提供了一些额外的信息，包括用户人口统计学信息和用户评论电影时的时间戳，这些信息作为上下文信息用于验证本文提出的推荐方法。具体地，在本文实验中涉及的上下文信息包括用户人口统计学上下文信息（年龄、性别、职业等）和用户评分上下文信息（评论时间以及由其推导出的季节、星期等）。在模型训练中，训练集中的上下文信息作为“评分上下文”应用于训练集中服务的评分向量的修正，而在模型测试中，测试集中的上下文信息作为“推荐上下文”应用于测试集中服务的评分向量

的修正。实验环境如下：Matlab2016b 软件，系统为 Windows 7，Intel core i7，CPU 为 3.6 GHz，内存为 8 GB。

4.2 实验的评价标准

分类准确率通常作为衡量推荐方法性能的一种评价标准。其中，分类准确率 A 用于衡量推荐方法的整体分类能力，如式(15)所示；正例分类准确率 A^+ 用于衡量推荐方法对正例样本分类的准确度，如式(16)所示；负例分类准确率 A^- 用于衡量推荐方法对负例样本分类的准确度，如式(17)所示。

$$A = \frac{1}{m} \sum_{i=1}^m A_i = \frac{1}{m} \sum_{i=1}^m \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i} \quad (15)$$

$$A^+ = \frac{1}{m} \sum_{i=1}^m A_i^+ = \frac{1}{m} \sum_{i=1}^m \frac{TP_i}{TP_i + FP_i} \quad (16)$$

$$A^- = \frac{1}{m} \sum_{i=1}^m A_i^- = \frac{1}{m} \sum_{i=1}^m \frac{TN_i}{TN_i + FN_i} \quad (17)$$

对于目标用户 u_i ， TP_i 表示正例样本被正确分为正例的样本数量， FP_i 表示负例样本被错误分为正例的样本数量， TN_i 表示负例样本被正确分为负例的样本数量， FN_i 表示正例样本被错误分为负例的样本数量。

在服务推荐中，通常还采用准确率 P 、召回率 R 和 F 值衡量推荐方法的性能，分别如式(18)~式(20)所示。 P 衡量推荐列表中预测到目标用户感兴趣服务的概率。 R 衡量推荐列表中目标用户感兴趣的服务占目标用户所有感兴趣服务的比重。 P 和 R 分别表示模型的预测和覆盖能力，但有时准确率和召回率是相互影响的，即准确率高而召回率低，或召回率高而准确率低，需要综合考虑两者。而 F 值是 P 和 R 的加权调和均值，可以很好地衡量推荐算法整体推荐性能，在推荐系统中经常被采用。

$$P = \frac{1}{m} \sum_{i=1}^m P_i = \frac{1}{m} \sum_{i=1}^m \frac{|RL_i \cap TL_i|}{|RL_i|} \quad (18)$$

$$R = \frac{1}{m} \sum_{i=1}^m R_i = \frac{1}{m} \sum_{i=1}^m \frac{|RL_i \cap TL_i|}{|TL_i|} \quad (19)$$

$$F = \frac{1}{m} \sum_{i=1}^m \frac{2P_i R_i}{P_i + R_i} \quad (20)$$

其中， RL_i 表示为目标用户 u_i 推荐的服务集合， TL_i 表示测试集中目标用户 u_i 感兴趣的服务集合。

4.3 核函数的选取

在核函数选取实验中，根据文献[8,17]的相关分

析，核函数在用户评分矩阵密度 $d=0.06$ 下进行，同时根据文献和常识习惯，设置用户偏好阈值 $\varepsilon = 3$ 。由于线性核函数 $\varphi(x_i, x_j) = x_i x_j$ 在之前实验中效果不太理想，故实验中不考虑线性核函数，只比较采用多项式核函数 $\varphi(x_i, x_j) = (x_i x_j + 1)^p$ ($p \geq 1$) 和高斯核函数 $\varphi(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ ($\sigma > 0$) 的分类准确率。

图3是参数 p 取不同值时采用多项式核函数的分类准确率。随着 p 值增大， A^+ 值缓慢上升， A^- 值快速下降， A 值缓慢下降。可以看出， $p=2$ 时， A 值达到最大。因此，采用多项式核函数时， $p=2$ 是最佳参数。

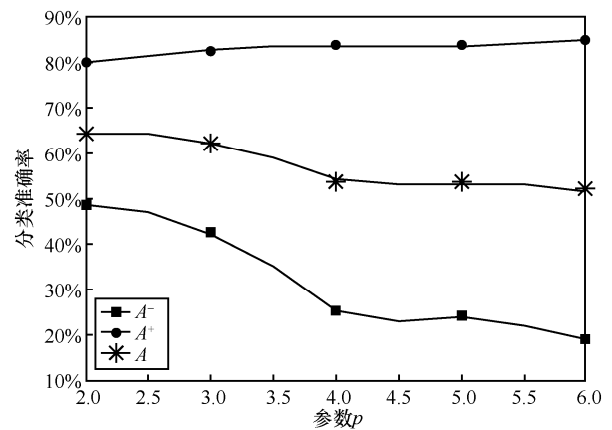


图3 不同参数值时多项式核函数的分类准确率

图4是参数 σ 取不同值时采用高斯核函数的分类准确率。随着 σ 值增大， A^+ 值前期缓慢下降，后期快速下降； A^- 值前期缓慢上升，后期快速上升； A 值缓慢到达顶峰后又缓慢下降。可以看出， $\sigma = 7$ 时， A 值达到最大，因此，采用高斯核函数时， $\sigma = 7$ 是最佳参数。

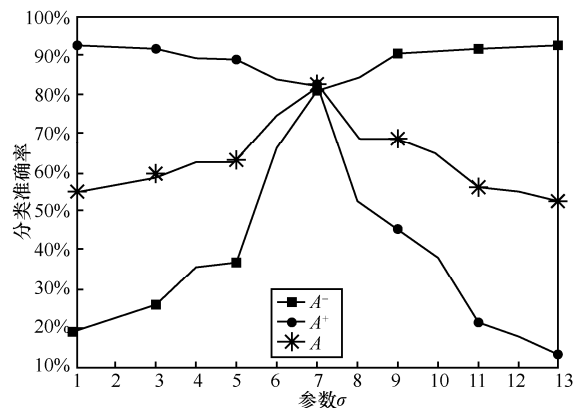


图4 不同参数值时高斯核函数的分类准确率

图 5 是多项式核函数和高斯核函数取最佳参数时, A 、 A^+ 和 A^- 值的比较。在负例分类准确率方面, 多项式核函数比高斯核函数相比效果较差。在正例分类准确率方面, 多项式核函数比高斯核函数稍低。从两者整体分类准确率而言, 高斯核函数比多项式核函数相比效果更好, 且正例和负例分类准确率比较均衡, 因此, 在后面的实验中, 选取高斯核函数作为 SVM 分类的核函数。

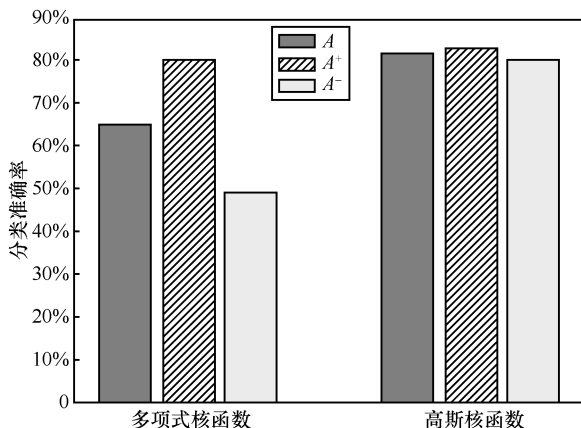


图 5 不同核函数时分类准确率比较

4.4 原始评分信息保留程度参数的选取

在不同的用户评分矩阵密度下, 本文对原始评分信息保留程度参数 ρ 对推荐结果的影响进行实验, 找出 F 的最优值。图 6 是 ρ 取不同值时的 F 值。从图 6 可以看出, 在评分矩阵密度 $d=0.06$ 和 $d=0.05$, $\rho=0.7$ 时, F 达到最优值, 在评分矩阵密度 $d=0.04$, $\rho=0.8$ 时, F 达到最优值。

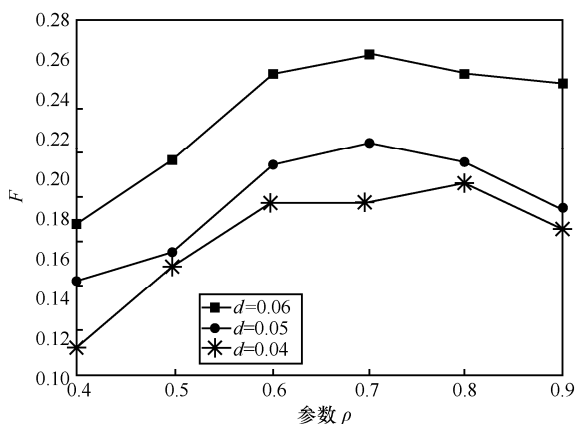


图 6 不同 ρ 值时推荐的 F 值

4.5 服务选择权重参数的选取

在不同的用户评分矩阵密度下, 本文对服务选择权重参数 δ 对推荐结果的影响进行实验。图 7 是 δ 取不同值时的 F 值。从图可以看出, 在不同的评

分矩阵密度下, $\delta=0.7$ 时, F 值达到最优, 原因如下: 一方面, 目标用户倾向于选择基于 SVM 预测模型的服务推荐, 这是因为该推荐基于目标用户的历史评分数据得出, 具有很好的解释性, 用户更加信服, 因此, 从 CS_{svm}^i 中选取较多的服务个数; 另一方面, 目标用户与相似用户具有相似的偏好, 因此, 从 CS_{sim}^i 中选取剩余的部分。

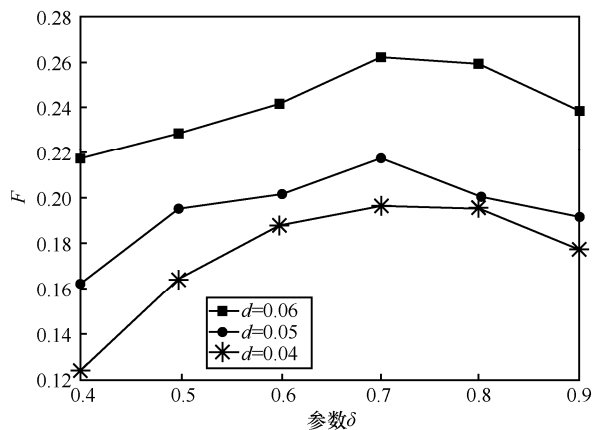


图 7 不同 δ 值时推荐的 F 值

4.6 SRM-CESVM 方法的性能

图 8~图 10 分别显示了在评分矩阵密度 $d=0.06$ 、 0.05 和 0.04 下, 该方法的推荐性能。在不同的密度下, 随着推荐服务数量 N 的增加, P 值不断下降, R 值不断上升, F 值缓慢上升后缓慢下降。当 $d=0.06$, $N=15$ 时, F 值达到最高值 0.26; N 为 10~25 时, F 值保持在 0.25 左右。当 $d=0.05$ 时, $N=20$ 时, F 值达到最高值 0.22; N 为 15~30 时, F 值保持在 0.2 左右。当 $d=0.04$, $N=25$ 时, F 值达到最高值 0.19; N 为 15~25 时, F 值保持在 0.18 左右。实验结果说明, 随着密度的下降, 评分数据变得越来越稀疏, 特别地, 当 $d=0.04$ 时, 数据极度稀疏, 但该推荐方法仍保持了不错的性能。

4.7 SRM-CESVM 方法的冷启动

为了验证本文提出的方法对冷启动问题的解决能力。本文从数据集中随机选择 100 个用户和 300 部电影作为新用户和新服务, 即人工删除评分信息。实验中, 为每个新用户推荐 20 个服务, 同时将一个新服务推荐给 20 个用户。实验中, 对于每个新用户记录用户满意的服务 (即服务评分大于等于用户偏好阈值) 占推荐服务数的比例。同时, 对于每个新服务, 记录推荐用户中用户满意的用

数占推荐用户数的比例。图 11 显示了不同矩阵评分密度时, 本文方法与文献[6]提出 URA-CF 方法的比较结果, 其值是针对所有用户或所有服务的平均比例值。从实验结果可以得出本文提出的方法由于加入了上下文信息, 特别是人口统计学信息, 在解决冷启动问题方面是可行的, 同时结果优于文献[6]中的推荐方法。

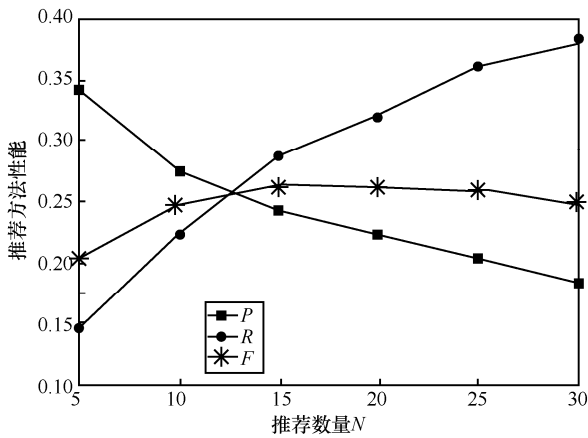


图 8 d=0.06 时推荐方法的性能

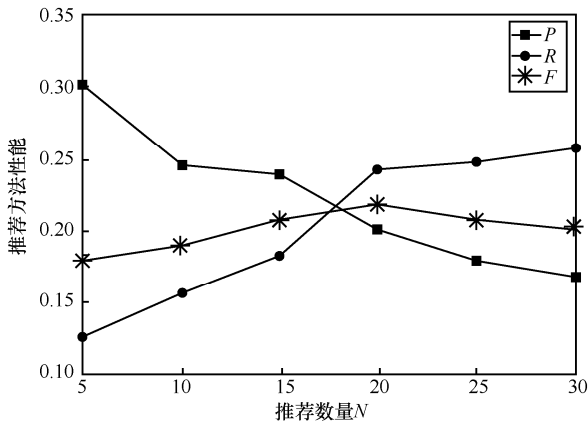


图 9 d=0.05 时推荐方法的性能

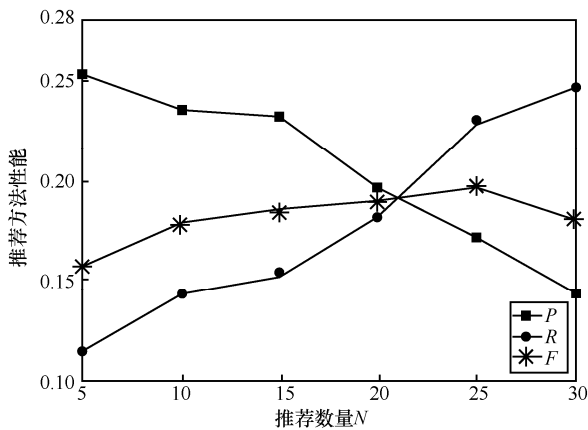


图 10 d=0.04 时推荐方法的性能

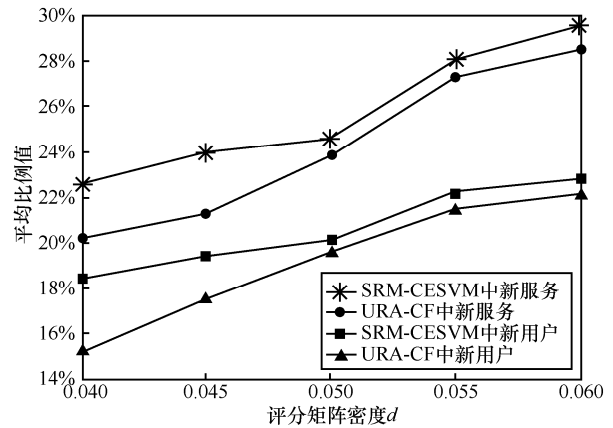


图 11 解决冷启动问题的性能比较

4.8 与其他推荐方法的性能比较

将 SRM-CESVM 方法与目前最新的协同过滤推荐方法进行比较, 对比方法具体如下。

URA-CF^[6], 一种最新的基于评分预测与排序预测的协同过滤推荐方法。该方法是基于概率矩阵分解的改进协同过滤算法。

SVM-CF^[8,17], 一种最新的基于 SVM 的协同过滤推荐算法。

UI-CF^[18], 一种最新的基于用户和项目的混合改进协同过滤方法。

图 12~图 14 分别显示了在评分矩阵密度 $d=0.06$ 、 0.05 和 0.04 下 4 种推荐方法的性能比较。不同评分矩阵密度下, SRM-CESVM 方法的性能均优于其他 3 种推荐方法, 特别是在数据极其稀疏的情况下, SRM-CESVM 方法明显优于其他方法。这是由于 SRM-CESVM 推荐方法使用了高效的 SVM 预测模型, 并引入隐含的上下文信息以提高精度, 同时使用用户人口统计学信息缓解数据稀疏性带来的推荐算法效率不高的问题。

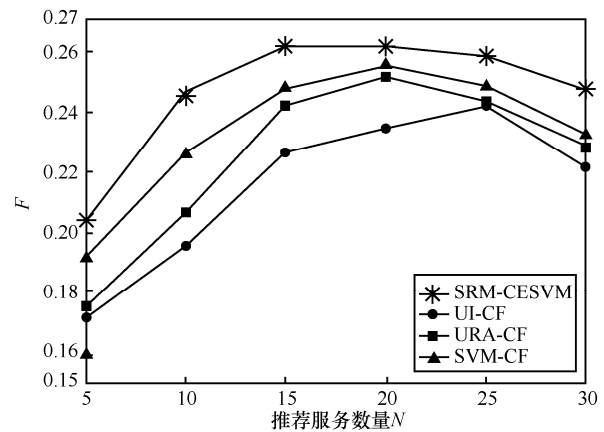


图 12 d=0.06 时 4 种推荐方法的性能比较

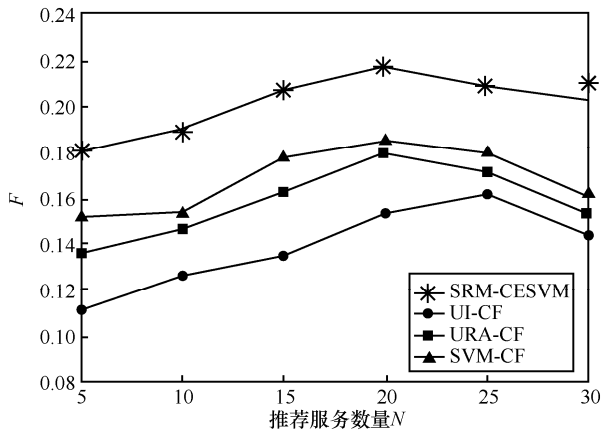


图 13 $d=0.05$ 时 4 种推荐方法的性能比较

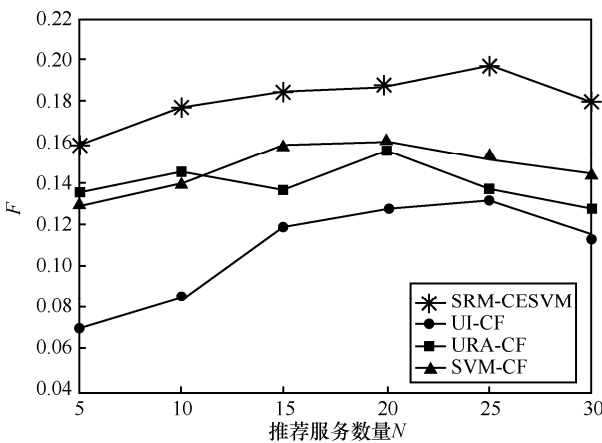


图 14 $d=0.04$ 时 4 种推荐方法的性能比较

为了比较 4 种推荐方法的时间效率，在 3 种评分矩阵密度 $d=0.06$ 、 0.05 和 0.04 下分别统计 4 种推荐方法的推荐时间消耗，结果如图 15 所示。一方面，随着数据密度的降低，推荐时间减少，这是由于数据密度降低表示推荐服务的数量减少，因此推荐时间消耗减少。另一方面，本文提出的方法的时间消耗比 UI-CF 方法减少了约 40%，比 URA-CF 方法减少了约 35%，表明 SRM-CESVM 方法具有很高的时间效率，这是由于采用 SVM 的方法过滤了用户不感兴趣的服务，节约了大量的时间。同时，SRM-CESVM 方法的时间消耗比 SVM-CF 方法增加了约 10%，这是由于引入了上下文信息及相似用户的推荐，因此时间消耗略有增加，但推荐精度有所提高。

5 结束语

本文提出了基于隐含上下文支持向量机的服务推荐方法。该方法首先利用上下文信息修正用户评分矩阵，然后利用 SVM 训练出用户的分类超平

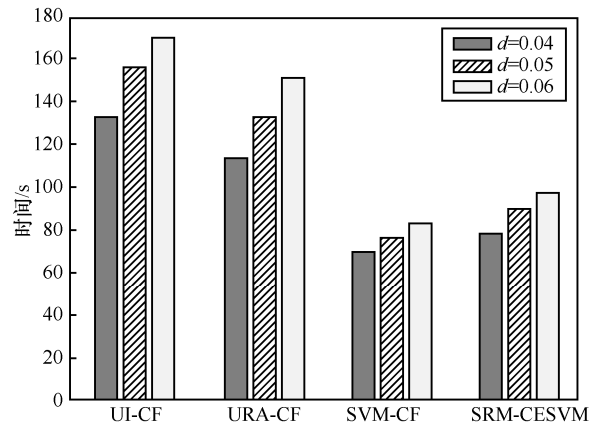


图 15 不同矩阵密度下 4 种推荐方法的时间消耗

面，最后根据 SVM 预测模型和相似用户为目标用户做出服务推荐。与其他基于预测的方法相比，该方法使用隐含上下文的服务特征向量，即不增加 SVM 的计算复杂度，又使上下文信息用于指导推荐。实验结果表明，提出的方法在推荐精度和推荐时间消耗方面均得到有效提升。

参考文献:

- [1] SHU J B, SHEN X X, LIU H, et al. A content-based recommendation algorithm for learning resources[J]. *Multimedia Systems*, 2018, 24(2): 163-173.
- [2] LI C Y, HE K J. CBMR: an optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis[J]. *Concurrency and Computation: Practice and Experience*, 2017, 29(10): e4092.
- [3] LI D S, CHEN C, LV Q, et al. An algorithm for efficient privacy-preserving item-based collaborative filtering[J]. *Future Generation Computer Systems*, 2016, 55: 311-320.
- [4] AI-HASSAN M, LU H Y, LU J. A semantic enhanced hybrid recommendation approach: a case study of e-Government tourism service recommendation system[J]. *Decision Support Systems*, 2015(72): 97-109.
- [5] 孟桓羽, 刘真, 王芳, 等. 基于图和改进 K 近邻模型的高效协同过滤推荐算法[J]. *计算机研究与发展*, 2017, 54(7): 1426-1438.
- [6] MENG H Y, LIU Z, WANG F, et al. An efficient collaborative filtering algorithm based on graph model and improved KNN[J]. *Journal of Computer Research and Development*, 2017, 54(7): 1426-1438.
- [7] 李改, 陈强, 李磊. 基于评分预测与排序预测的协同过滤推荐算法[J]. *电子学报*, 2017, 45(12):3070-3075.
- [8] LI G, CHEN Q, LI L. Collaborative filtering recommendation algorithm based on rating prediction and ranking prediction[J]. *ACTA Electronica Sinica*, 2017, 45(12): 3070-3075.
- [9] KUSHWAHA N, SUN X D, SINGH B, et al. A lesson learned from PMF based approach for semantic recommender system[J]. *Journal of Intelligent Information Systems*, 2018, 50(3): 441-453.

- [8] REN L F, WANG W J. An SVM-based collaborative filtering approach for Top-N Web services recommendation[J]. Future Generation Computer Systems, 2018(78): 531-543.
- [9] OKU K, NAKAJIMA S, MIYAZAKI J, et al. Context-aware SVM for context-dependent information recommendation[C]//The 7th International Conference on Mobile Data Management. IEEE, 2006: 109-112.
- [10] MA L R, SONG D D, LIAO L J, et al. PSVM: a preference-enhanced SVM model using preference data for classification[J]. Science China Information Sciences, 2017, 60(12): 122-103.
- [11] WEI J, HE J H, CHEN K, et al. Collaborative filtering and deep learning based recommendation system for cold start items[J]. Expert Systems with Applications, 2017(69): 29-39.
- [12] 黄立威, 刘艳博, 李德毅. 基于深度学习的推荐系统[J]. 计算机学报, 2017, 40(156): 1-30.
HUANG L W, LI Y B, LI D Y. Deep learning based recommender systems[J]. Chinese Journal of Computers, 2017, 40(156): 1-30.
- [13] YAO L N, SHENG Q Z, NGU A H H, et al. Unified collaborative and content-based Web service recommendation[J]. IEEE Transactions on Service Computing, 2015, 8(3): 453-466.
- [14] AFIFY Y M, MOAWAD I F, BADR N L, et al. Enhanced similarity measure for personalized cloud services recommendation[J]. Concurrency and Computation: Practice and Experience, 2017, 29(8): e4020.
- [15] ZHAO C Y, WANG J L. Network education video recommendation algorithm based on context and trust relationship[C]//The 4th IEEE International Conference on software Engineering and Service Science. IEEE, 2013: 537-540.
- [16] YIN C Y, WANG J, PARK J H. An improved recommendation algorithm for big data cloud service based on the trust in sociology[J]. Neurocomputing, 2017(256): 49-55.
- [17] 张宇, 王文剑, 赵胜男. 基于正负反馈的 SVM 协同过滤 Top-N 推荐算法[J]. 小型微型计算机系统, 2017, 38(5): 961-966.
ZHANG Y, WANG W J, ZHAO S N. SVM collaborative filtering Top-N recommendation algorithm based on positive and negative feedback[J]. Journal of Chinese Computer Systems, 2017, 38(5): 961-966.
- [18] CHENG S L, ZHANG B F, ZOU G B. A rating-based integrated recommendation framework with improved collaborative filtering approaches [J]. International Journal of Computers, Communications and Control, 2017, 12(3): 307-322.

[作者简介]



赵晨阳 (1982-), 男, 河南郸城人, 博士, 河南工业大学讲师, 主要研究方向为人工智能、深度学习、智能推荐等。



王俊岭 (1983-), 男, 河南郑州人, 博士, 河南工业大学讲师, 主要研究方向为人工智能、组合优化、智能推荐等。