第 10 卷 第 3 期
2021 年 5 月
集 成 技 术
JOURNAL OF INTEGRATION TECHNOLOGY
Vol. 10 No. 3
May 2021

引文格式：
吴嘉澍, 饶华枭, 范小朋, 等. 基于用户特征聚类联合情境特征的多维度应用推荐系统 [J]. 集成技术, 2021, 10(3): 22-34.
Wu JS, Rao HX, Fan XP, et al. A multidimensional application recommender system based on user feature hierarchical clustering with user behaviour information [J]. Journal of Integration Technology, 2021, 10(3): 22-34.

基于用户特征聚类联合情境特征的多维度应用推荐系统

吴嘉澍[1,2] 饶华枭[1,3] 范小朋[1] 王 洋[1*]

[1](中国科学院深圳先进技术研究院 深圳 518055)
[2](墨尔本大学计算与信息系统学院 墨尔本 3052)
[3](中国科学院大学 北京 100049)

摘 要 随着移动设备的普及、同时大数据时代数据过载问题的日益严重，如何更准确地根据用户的兴趣及行为向用户推荐其可能感兴趣的应用软件成为亟待解决的问题。现有的推荐系统方法大多面临着推荐内容较为单一乏味等问题，且在推荐时没有将用户所处情境加以考虑，导致推荐效果欠佳。该文提出一种基于用户特征聚类联合情境特征的多维度应用推荐系统。经奇异值分解降维并去噪后的用户特征数据会被层次聚类为多个用户组，之后与用户所处情境信息联合输入至贝叶斯模型，得到应用推荐准确概率的降序推荐列表，从而实现更加准确的应用推荐。该文在分布式框架下实现了所提出的推荐系统，使其运行高效可靠。经验证，经过奇异值分解处理后的数据组内平均差值降低至 0.4，聚类后得到的应用推荐召回率提升至 73%，较基于用户协同过滤与基于关联规则的方法有 5%～16% 的显著提升，且贝叶斯模型的量化指标均有约 10% 的提升，充分验证了所提出算法及系统的有效性。

关键词 推荐系统；应用推荐；奇异值分解；层次聚类；用户情境信息
中图分类号 TP 302 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20201215001

A Multidimensional Application Recommender System Based on User Feature Hierarchical Clustering with User Behaviour Information

WU Jiashu[1,2] RAO Huaxiao[1,3] FAN Xiaopeng[1] WANG Yang[1*]

[1](*Shenzhen Institute of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[2](*School of Computing and Information Systems*, *University of Melbourne*, *Melbourne* 3052, *Australia*)
[3](*University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)
[*]*Corresponding Author*: *yang.wang1@ siat.ac.cn*

Abstract With the prevalence of mobile devices and the gradually severe information overloading in

收稿日期：2020-12-15 **修回日期**：2021-02-25
基金项目：中国科学院中亚生态与环境研究中心项目(RCEECA-2018-001)；广东省重点领域研发项目(2020B010164003, 2019B010137002)；国家自然科学基金面上项目(61672513)
作者简介：吴嘉澍，硕士研究生，研究方向为大数据与云计算；饶华枭，硕士研究生，研究方向为大数据；范小朋，副研究员，研究方向为大数据分析处理；王洋(通讯作者)，博士，研究员，研究方向为云计算与并行分布式计算，E-mail：yang.wang1@ siat.ac.cn。

big data era, how to accurately recommend apps to users based on their interests and their user behaviours becomes an important problem. Most existing recommender systems suffer from common weaknesses such as making recommendations that are too dull and lacking diversity, and they also rarely consider the user behaviours when making recommendations, resulting in sub-optimal recommendation performance. In this paper, we propose a multi-dimensional app recommender system that takes both the hierarchically clustered user features and user behaviours into account when making recommendations. After being processed by singular value decomposition (SVD) that performs denoising and dimensionality reduction, the user features are then clustered into several user groups by the hierarchical clustering. The user behaviour information is subsequently leveraged together with the user features and inputted into the Bayesian model to produce an app recommendation list in the descending order of the probability of being accurate. Hence, the accuracy of the app recommendation is improved. This paper finally implements the proposed recommender system under a distributed framework, which ensures the efficiency and the reliability of the system. The experiments demonstrate an average within-group deviation of 0.4 after the SVD is applied. The recall of the app recommender system after the hierarchical clustering raises to 73%, which shows a 5%～16% performance boost compared with collaborative-filtering-based and association-rule-based methods. All evaluation metrics of the Bayesian model also presents a performance gain of around 10%, which verifies the effectiveness of the proposed recommender system.

Keywords　recommender system; application recommendation; singular value decomposition; hierarchical clustering; user behaviour information

Funding　This work is supported by Research Center for Ecology and Environment of Central Asia, Chinese Academy of Sciences (RCEECA-2018-001), Key-Area Research and Development Program of Guangdong Province (2020B010164003, 2019B010137002), and National Natural Science Foundation of China (61672513)

1　引　言

随着数字时代的到来以及智能移动设备的广泛应用，人们可获取的信息量急剧增多，信息过载问题也因此变得日趋严重[1]。为应对信息过载带来的弊端，将真正有用的信息以个性化的形式推荐给用户，推荐系统[2]成为解决信息过载问题的有力工具之一[3-4]，其应用也变得愈发广泛，且受到了众多的关注和研究。面对海量的信息，推荐系统可以根据用户的信息需求、兴趣爱好等，将用户可能感兴趣的信息、产品等进行个性化推荐。与传统的搜索引擎相比，推荐系统可以通过收集用户的兴趣偏好[5]，进行个性化的学习，从而自动化地发现用户的兴趣点，无需用户人为输入其兴趣点。这使得推荐系统在引导用户发现其信息需求的同时更加简洁易用。

虽然经过了长期的发展与演化，但推荐系统算法仍存在一些不足[6]。例如，有限的数据收集能力无法应对当前海量的信息，以及数据收集过程较差的连续性造成较严重的数据稀疏性等问题[7-8]。这势必会给推荐系统算法对用户行为的分析、用户兴趣的挖掘带来困难。同时，现有

的绝大多数推荐系统算法着重于对用户和所推荐的商品属性进行分析并辅助推荐，在推荐过程中往往缺失对用户所处情境[9-10]的考虑，如用户上一个浏览的应用类型、用户在某类应用中的启动次数、停留时间等，从而使得推荐系统的推荐效果欠佳。此外，推荐系统算法还面临着推荐同质化的问题[11]。绝大多数的推荐系统更善于推荐与用户历史兴趣相似的资源，推荐的内容缺乏新颖性，同类资源的重复推荐无法引导用户发现更多新的潜在兴趣点，从而使得推荐系统存在较大的局限性，其推荐准确性也因此而受损[12]。

针对上述推荐系统面临的局限性，以及推荐过程中对用户自身所处情境考虑的缺失，本文旨在对传统推荐系统算法做进一步优化，提出基于用户特征聚类联合场景特征的多维度应用推荐系统。该推荐系统将根据不同用户群体、不同应用软件特征以及用户所处情境信息为用户进行个性化的应用程序推荐。具体地，首先对用户群体的特征矩阵进行奇异值分解(Singular Value Decomposition)操作，得到降维并且去噪的用户向量表示。其中，对用户群体特征进行的奇异值分解可以去除数据中重要性较低的成分，从而对用户特征的表示进行精简。同时，这种精简同样可以使得所提出推荐系统的运行更加高效。其次，对用户群体的特征向量进行层次聚类分析，从而将具有相似用户特征的用户聚类，并利用各用户类的均值向量作为其特征表示。在此基础上，算法结合用户所处情境因素，以用户为出发点、情境为补充，依据贝叶斯模型预测最佳的应用推荐列表，从而实现为用户进行更为优化的应用软件推荐，提高用户对海量信息的使用精准度与使用效率。

在推荐系统运行性能及拓展性能方面，为满足推荐系统的高并发处理海量数据的需求，本文使用分布式集群服务器运行所提出的系统，使得数据存储、模型运算等方面的性能有了极大提升。此外，计算集群使得本文推荐系统具备良好的拓展性，对推荐系统在运行中的稳定性起到了重要的作用。

为验证所提出算法的有效性，本文将所提出的基于奇异值分解的用户特征层次聚类结合情境特征信息，并利用贝叶斯模型的推荐系统算法与传统的推荐系统算法进行推荐质量及模型量化指标的效果对比，从而验证系统的有效性与优势。

本文将在第 2 小节介绍所涉及算法的相关理论研究；第 3 小节陈述所提出算法的细节以及其在现有研究基础上的创新性；第 4 小节介绍推荐系统的架构与具体实现；第 5 小节展示本文推荐系统的实验验证结果；最后小节对本文进行概括与总结。

2 算法介绍及相关理论研究

2.1 聚类算法在推荐系统中的应用

聚类算法是一种探索性的分析方法，在聚类过程中，无需事先给出聚类的具体标准。聚类算法能够从样本数据出发，自动发现空间实体的属性间所具有的关联关系并进行聚类[13]，从而将本身没有类别的样本聚集成不同的类(簇)，其目的在于使属于同一个类的样本之间能够彼此相似，而属于不同类的样本之间在特征上应该足够疏远。同时，聚类算法对大型数据集具有良好的可拓展性，这对于数据需求量大的推荐系统而言是极其适用的。此外，聚类结果的可解释性以及运行的高效性也为其能够应用在推荐系统中打下了良好的基础。Shinde 和 Kulkarni[14]提出了基于用户评价聚类的个性化推荐系统算法。该算法首先将用户根据其对商品的评价进行分类，之后，对商品具有类似评价、类似兴趣与偏好的用户组进行个性化的商品推荐，从而使得推荐系统的推荐更加精准。经过验证，聚类算法有效地提升了推

荐系统对于商品的推荐质量。Hsu[15]则在为学生推荐英语学习课程的推荐系统中运用聚类算法，先将具有不同学习行为习惯的学生聚类为不同的学生组，随后优化推荐系统算法，为具有不同学习习惯的学生组推荐适合他们的英语课程，从而使得推荐系统更加人性化。

然而，上述聚类算法应用于推荐系统中的工作时，存在两个缺点。首先，这些方法均未在聚类前使用矩阵分解的方法对用户特征信息进行降维并去噪，这使得特征信息中不仅存在冗余以及表示能力弱的特征，而且具有较高的维度，从而损害推荐系统的运行效率。其次，这些方法并未充分利用用户自身所处的情境信息，使得推荐系统在进行推荐时具有局限性。

2.2　情境信息在推荐系统中的应用

情境信息[16]可以辅助推荐系统分析用户的背景、兴趣爱好等个性化信息。在推荐系统模型中，情境可以是用户打开某类应用的次数、用户在某类应用中的停留时长以及用户上一个使用的应用种类等能够反映用户决策的额外信息。例如，通过用户上一个使用的应用可以辅助预测用户下一个可能会使用的应用，通过实时地获取用户所处的情境信息，进而在原有用户数据和应用数据的基础上，更为人性化、智能化地分析用户的信息需求，从而提高应用推荐的效果。

然而，当前基于情境因素的推荐系统算法均利用用户情境信息来分析所推荐产品的特征，而并非对用户自身喜好特征进行分析。Hu 等[17]在为用户推荐电视节目的推荐系统中借助了用户浏览历史、购买历史、搜索历史甚至是用户鼠标移动等用户情境信息，对电视节目特征进行分析，并将电视节目更好地推荐给用户。而 Abbas 等[18]则利用 YouTube 视频的用户评分与评价信息，获得对视频更好的特征分析与定位，从而辅助推荐系统为用户推荐更加准确、合适的 YouTube 视频。

然而，上述算法均利用用户情境信息分析产品的特征，而现有的推荐系统算法较少有利用用户情境信息对用户喜好进行分析。此外，用户评分、评价等信息不仅需要用户的额外参与，而且不同用户拥有不同的评分、评价标准，这导致此类用户情境信息收集困难，且利用难度大，从而对推荐系统的推荐效果产生影响。

3　算法设计

为了更加准确高效地为用户进行应用推荐，本文提出了基于用户特征聚类联合情境特征的多维度推荐系统。首先，对用户特征进行奇异值分解；然后，对分解后的用户特征进行层次聚类操作，并用每一个用户类的均值向量作为代表该类用户的特征信息；最后，结合用户所处情境信息，利用贝叶斯模型，对用户可能感兴趣的应用进行预测并按照概率降序排序，形成应用推荐列表。

3.1　用户特征奇异值分解

首先，推荐系统所需的用户特征将以矩阵形式进行初始化。每一个用户都使用 T 维的向量表示该用户的 T 个特征（A_1—A_T），如年龄、性别、爱好、所在地区等。因此，N 个用户（U_1—U_N）将构成一个 N 行 T 列的用户特征矩阵，如图 1 所示。

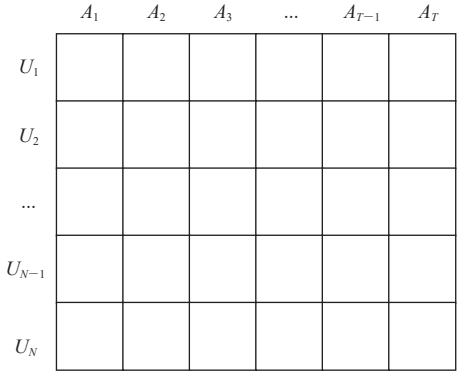

图 1　用户-特征矩阵 M 示意图

Fig. 1　Illustration of user-feature matrix M

然而，对于表示用户特征的 T 个特征维度而言，并非每个特征维度都能有效地用来表示用

户信息，其中部分特征包含噪声、或是大量的冗余信息，从而使得其不具备良好的表示能力。同时，不同特征也拥有不同的重要程度，并非所有的特征维度在表示用户时都具有相同的重要程度。此外，初始时的特征维度 T 可能较大，从而造成用户特征矩阵维度较高，若直接将庞大的用户特征矩阵输入至推荐系统算法，将使得算法运行效率较低。为解决上述问题，本文算法使用奇异值分解算法对用户特征矩阵进行分解，从而达到降维和去噪的效果。

奇异值分解算法可以将用户-特征矩阵 $\boldsymbol{M}$ 分解为用户-精简特征空间矩阵 $\boldsymbol{U}$、奇异值对角矩阵 $\boldsymbol{\Sigma}$ 和精简特征空间-特征矩阵 $\boldsymbol{V}$ 的矩阵乘积，如公式(1)所示。通过奇异值分解操作，用户特征表示能力最强的一部分特征将会被保留，而特征中影响程度不大或表示能力不强的次要特征将会被剔除[19]。通过将主要的影响因素保留，可以使得特征具有对用户更强的表示能力。此外，在经过奇异值分解后，特征维度的下降也使得所生成的用户-精简特征空间矩阵 $\boldsymbol{U}$ 拥有较低的维数，从而使得后续的推荐运算变得更加高效。

$$\boldsymbol{M}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{*} \tag{1}$$

3.2 用户特征层次聚类

在对用户特征矩阵进行奇异值分解之后，算法基于用户特征使用层次聚类算法(Hierarchical Clustering)[20]对用户进行聚类，从而将用户们聚类到特征相似、兴趣爱好相仿的用户组中。从下至上的层次聚类算法的算法流程如图 2 所示。首先，算法将所有用户数据都初始化为一个叶子节点。在每一次迭代中，算法会计算叶子节点之间的相似性(如欧氏距离)；随后，叶子节点之间的这种相似性信息将决定它们是否会被合并至新的聚类簇。层次聚类算法会不断地进行迭代合并，直到迭代终止条件达成，比如算法已经生成指定数目的聚类簇，或聚类簇之间的距离满足一定的条件。

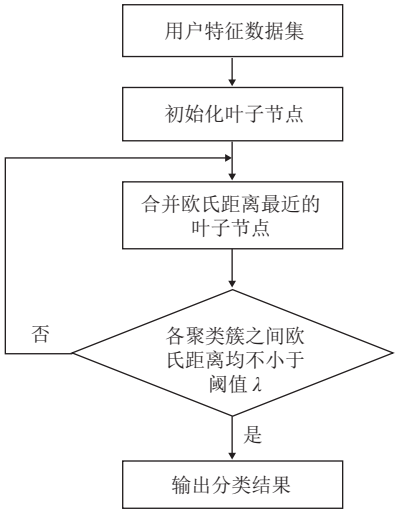

图 2 层次聚类训练流程图

Fig. 2 Description of the training process of the hierarchical clustering

本文算法将奇异值分解后的用户特征信息输入至层次聚类算法中进行迭代，直到每组聚类簇的均值向量之间的距离都大于或等于一个距离阈值 λ 时，停止聚类迭代。每一次迭代后，各个聚类簇的均值向量之间的欧式距离均会被计算。在图 3 中，图左侧的 λ 值表示当前聚类状态下各个类的均值向量之间的最小距离。若预先设定的聚类迭代停止阈值 λ 为 1.5，则聚类算法会在生成 4 个聚类簇时停止迭代，因为此时聚类簇之间的欧式距离均已大于预先设定的阈值 λ。此时，用户 G1、G2、G8 会被分为一类，用户 G3、G4、G9 会被分为一类，用户 G5 和 G7 会被分为一类，用户 G6 自成一类。迭代停止条件的满足意味着

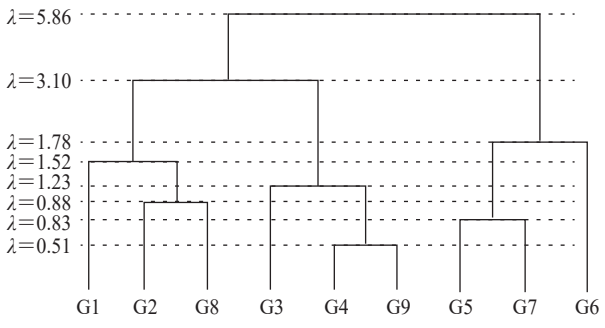

图 3 聚类过程的谱系图

Fig. 3 Dendrogram of the hierarchical clustering process

每个用户聚类簇之间已经拥有一定的差异性，从而能够避免用户被模棱两可地错分入其本不应属于的用户组。

通过层次聚类算法，具有较相似特征的用户会被聚合至同一类中，之后，每一类的均值向量将会被用来表示该类用户的特征，进行后续的推荐运算，如公式(2)所示。第 c 类用户的用户特征表示 U^c 为被分至该类的所有 I 个用户的用户特征向量的均值。

$$U^c=\frac{\sum_{i=1}^{I}U_i^c}{I} \tag{2}$$

通过这种方式，虽然同处一类的用户具有大体相似的特征，但处于同类的用户与用户之间也多少存在一定的差异性与多样性，并非完全相同。对每一类用户的特征取均值向量可以很好地将同类用户的特征进行融合，在不破坏用户原有兴趣取向的同时适当地引入同类用户带来的多样性，从而使得算法可以借助与某一用户兴趣大体相同的同类用户的平均特征，来帮助该用户发现同类用户具有的其他兴趣点，从而引导该用户发现同类用户可能拥有的潜在兴趣，使得推荐算法更加有效。

3.3　用户特征与情境因素的联合及贝叶斯预测模型的使用

与传统的推荐系统算法不同的是，本文算法在利用数据特征进行分析与应用推荐时，不再局限于用户自身特征(如年龄、性别、爱好等)。在传统的用户特征与应用特征的基础上，通过联合实时用户情境信息，实现利用多个维度的信息进行推荐的应用推荐系统。通过联合用户情境因子信息，推荐系统可以将实时获得的用户情境与原有的用户特征融合为多维度的特征空间，从而使得推荐系统更人性化、智能化地分析用户所需，优化应用推荐效果。本文算法所使用的部分用户情境特征及其表示示例见表 1。

表 1　情境特征及其表示示例

Table 1　Some user behavioural features and their corresponding examples

情境要素	表示示例
上一个使用的应用类型	应用类别 ID #7 (新闻类应用)
在第 #7 类应用（新闻类应用）停留的时间	98 min
在第 #8 类应用（游戏类应用）停留的时间	60 min
一周内打开第 #9 类应用（音乐类应用）的次数	20 次
……	……

在获得了通过奇异值分解与层次聚类得到的用户组的用户特征，以及用户情境特征与应用特征后，本文算法对原有的协同过滤推荐算法进行改进——采用用户特征、情境特征与应用特征联合成的多维特征，利用贝叶斯模型进行应用推荐[21]。在算法中，用户属性集合表示为 $U=\{U_1,U_2,U_3,\cdots,U_N\}$，情境集合表示为 $S=\{S_1,S_2,S_3,\cdots,S_N\}$，推荐应用列表采用 A 表示。算法期望通过计算在不同用户属性、不同情境下推荐不同应用的概率，从而为具有该用户特征与情境特征的用户推荐恰当的应用，如公式(3)所示：

$$P(A|U,S)=\frac{P(U|A)*P(S|A)}{P(U,S)} \tag{3}$$

根据公式(3)，算法可以计算出对于一个给出的用户特征及其所处情境特征，不同的应用被使用的概率。公式(3)中，按照 A 被使用条件下 U 的概率、A 被使用条件下 S 的概率、以及 U 和 S 同时发生的概率，可计算得到在给定用户特征 U 与情境特征 S 的情况下，不同应用 A 被使用的概率值。其中，所得到的概率越高，该应用就越值得被推荐。故概率最大的前 K 个应用，即是最值得被推荐的 K 个应用。由此，算法得到一个以概率降序排列的应用推荐列表。

同时，所提出算法可以接受并处理用户所处情境信息容易随时间发生变化的特点，算法

可以通过用户反馈的实时用户行为信息，如直接点击、收藏、停留时长长短、点击次数等，更新公式(3)中的概率信息，从而达到算法的实时更新，以使得应用推荐具有更好的实时性。

3.4 算法整体流程

本文算法的整体流程如图 4 中花括号所指部分所示。首先，算法对用户数据进行奇异值分解，并在分解得到的降维并去噪的用户特征上进行层次聚类迭代，直到满足终止条件，得到距离较远的各组用户聚类簇，并用各簇的均值向量作为该聚类簇中的用户特征；然后，利用贝叶斯模型，联合用户特征与情境特征，对应用被使用的概率进行预测并排序，最终形成应用推荐列表。

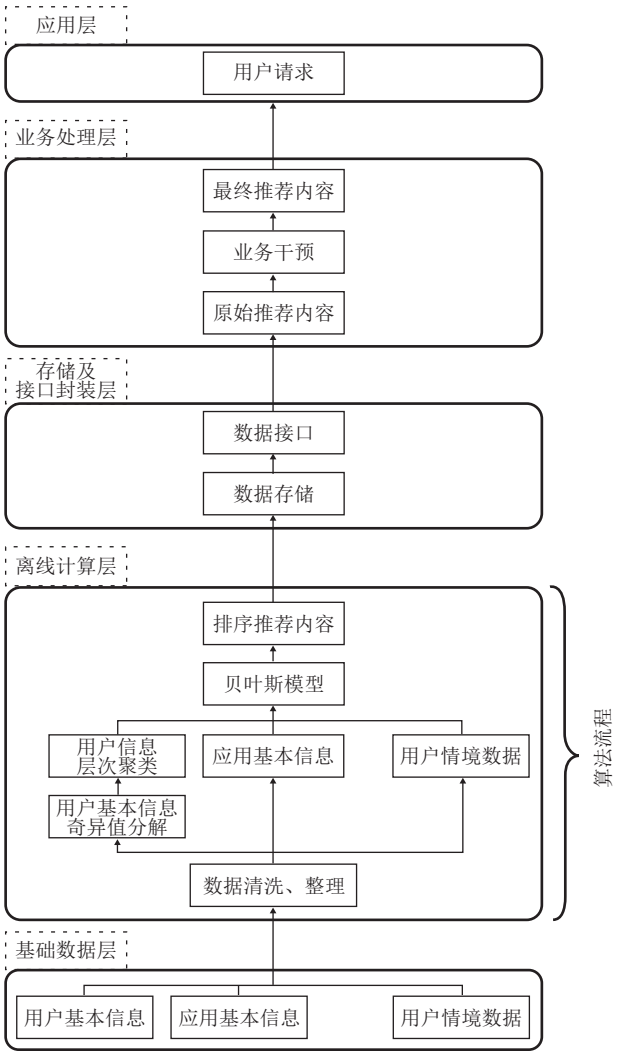

图 4 系统架构

Fig. 4 The architecture of the proposed system

4 系统架构与实现

4.1 系统总体架构

本文系统的总体架构分为 5 个部分，如图 4 所示。位于最底层的基础数据层主要包含用户基础属性数据，如年龄、性别、所在地、手机设备属性等基础数据；应用的基本信息，如某款应用属于某类、评分高低等，以及用户访问应用的数据，即算法所需要用到的情境信息。为获取所需数据信息，需要进行数据采集，其主要目标是将业务系统的数据同步至数据存储系统中进行保存，供下一步处理和使用。

本文推荐系统的第 2 层为离线计算层，其职责是进行数据处理与模型训练。首先，对所收集到的数据进行数据清洗与整理；其次，按照本文第 3 小节的算法流程进行模型训练；最后，形成可以为用户推荐应用的模型。算法流程如图 4 中圆括号所指部分所示。

推荐系统的第 3～5 层分别为存储及接口封装层、业务处理层和应用层。首先，系统会在推荐的排序内容基础上做数据接口的封装；然后，如有必要，业务处理层将按照市场变化等信息对推荐内容信息做必要的微调；最后，将业务干预的推荐数据传送至前端用户。整套系统需满足以下几点要求：

(1)建立及时、全面的互联网信息处理能力，通过探索网络资源对各种不同类型的常用应用软件信息进行基础抓取；

(2)基于云计算框架，建立分布式存储[22]、大规模数据并行计算水平拓展的技术能力，实现对系统横向扩展，方便进行大量数据的流式处理；

(3)建立高内聚低耦合的架构，复用基础能力，完善系统之间的接口设计，减少抓取、分析、可视化以及应用之间的强关联；

(4) 在增强推荐的准确度时，不断提高用户体验，利用所提出算法，提高推荐的精确度和召回率。

本文系统通过在传统推荐系统的基础上做出优化，将奇异值分解后的用户特征进行聚类，并与情境信息联合，通过贝叶斯模型分析预测用户所需要的应用，实现应用推荐效果的增强。同时，系统充分考虑了运行性能及扩展性，以及系统应对实时数据反馈补充、快速适应用户的特征属性变化的实时处理能力，通过在分布式架构上构建应用推荐系统，实现海量数据并行化处理，从而有效提高系统的吞吐量性能。

4.2　系统构建

在本文所提出的多维度应用推荐系统的构建过程中，使用了如下技术：本文系统采用 MySQL[23] 存储用户属性数据，采用 HBase[24] 存储用户画像数据。存储数据时使用 HDFS[25] 分布式文件系统。系统采用了基于 Spark[26] 搭建的底层平台，利用 Spark Streaming 进行实时计算、特征向量计算与排序列表的生成。系统在转发数据时使用 Kafka[27] 实现实时数据的集群转发。因为所构建的系统需要对数据库进行大量的复杂数据检索，所以，使用 ElasticSearch[28] 分布式系统架构成为一个良好的解决方案。此外，本文系统的可视化平台使用流行的 J2EE[29] 应用架构，平台层采用 JAVA 标准库以及 J2EE 服务器，J2EE 项目在 Tomcat[30] 服务上运行，通过 SpringMVC 框架集成实现网页开发。通过利用上述技术并合理协调配合，使得本文系统可以高效地满足所需的要求。本文所构建的系统可以支持系统实时获取并处理数据，系统在采集与处理数据时，可以支持至少百万级的数据处理吞吐量。在进行应用推荐时，本系统可以支持至少百级别的并发操作，并保持系统可用。在数据实时查询时，时延不会超过 4 s。与此同时，为满足系统拓展性的要求，增加系统处理数据的能力，本文系统额外准备了备用服务器，以避免出现数据处理量过大而导致的负荷超限。

5　实验结果分析

本文对所实现的应用推荐系统进行了充分且系统的测试，首先验证了所采用的奇异值分解与层次聚类具有良好的效果，随后从推荐结果与度量指标两个层面验证了所提出算法整体的有效性。同时，本文还对系统性能进行了测试，从而验证了系统的高效性与稳定性。

5.1　奇异值分解前后聚类有效性对比

为验证在层次聚类前进行奇异值分解的有效性，本文分别对进行与未进行奇异值分解的用户特征进行层次聚类，并随机抽取用户 #1056 所在的用户聚类簇，计算同组用户的属性平均差值。经过对比，层次聚类前未进行奇异值分解的数据平均差值可高达 0.6，而经过奇异值分解处理后数据的差值范围最大为 0.4。这表明用户特征数据在经过奇异值分解消除冗余信息之后进行层次聚类，其得到的聚类簇内保留了原始数据的主要信息，且数据更加精炼，聚类效果更好。图 5 则进一步地对算法的聚类效果进行了可视化：经过奇异值分解后进行层次聚类的聚类簇中出现错分的次数得到了明显的降低，这表明经过奇异值分解后进行聚类的有效性。

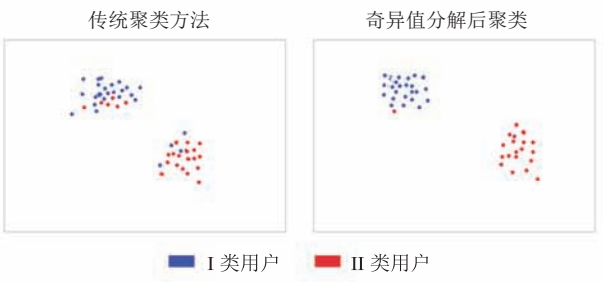

图 5　进行与不进行奇异值分解后聚类的效果对比

Fig. 5　Comparison between clustering quality with and without performing singular value decomposition before clustering

5.2 基于奇异值分解的层次聚类与传统聚类算法召回率效果对比

为进一步验证基于奇异值分解的层次聚类算法的有效性，将本文方法与传统的机器学习聚类算法进行对比，并以用户聚类结果召回率作为衡量不同算法有效性的指标。本文在基于 Hadoop 生态系统的 Mahout[31]上做用户聚类，如图 6 所示，对比试验中所采用的对比方法分别为决策树模型、K-means 算法以及数据未经过奇异值分解的传统层次聚类方法。通过将进行了去除错误数据等预处理步骤之后的用户数据集 M' 输入到不同聚类算法中，使用不同聚类算法得到 R1、R2、R3 与 R4 四种不同的聚类结果。其中，聚类结果 R4 是将数据 M' 先通过奇异值分解算法进行数据分解，之后输入至层次聚类算法中得到的聚类结果。

在验证过程中，本文采用 10 份交叉验证的方式，分别对上述 3 种传统聚类算法以及本文算法进行验证，并计算召回率。

如图 6～7 所示，采用传统的 K-means 聚类方式，平均召回率约为 57%，决策树算法的平均召回率则达 60% 左右。层次聚类前未进行奇异值分解的方法的平均召回率约为 68%，而进行奇异值分解后再进行层次聚类的方法的平均召回率可达 73% 左右，比前者提高约 5%。这充分表明本文方法相较于传统聚类算法的有效性。

5.3 推荐算法推荐结果质量对比与分析

如图 4 算法流程图所示，在用户特征数据通

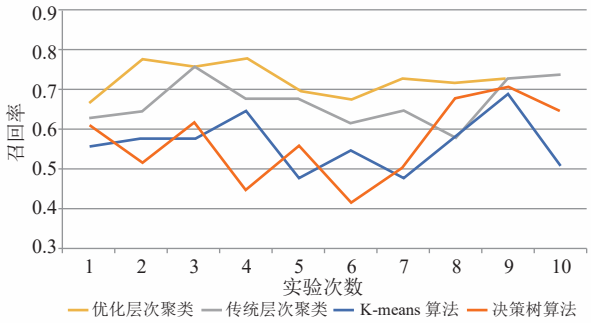

图 7 不同聚类算法的召回率对比

Fig. 7 Comparison of clustering recall of different clustering algorithms

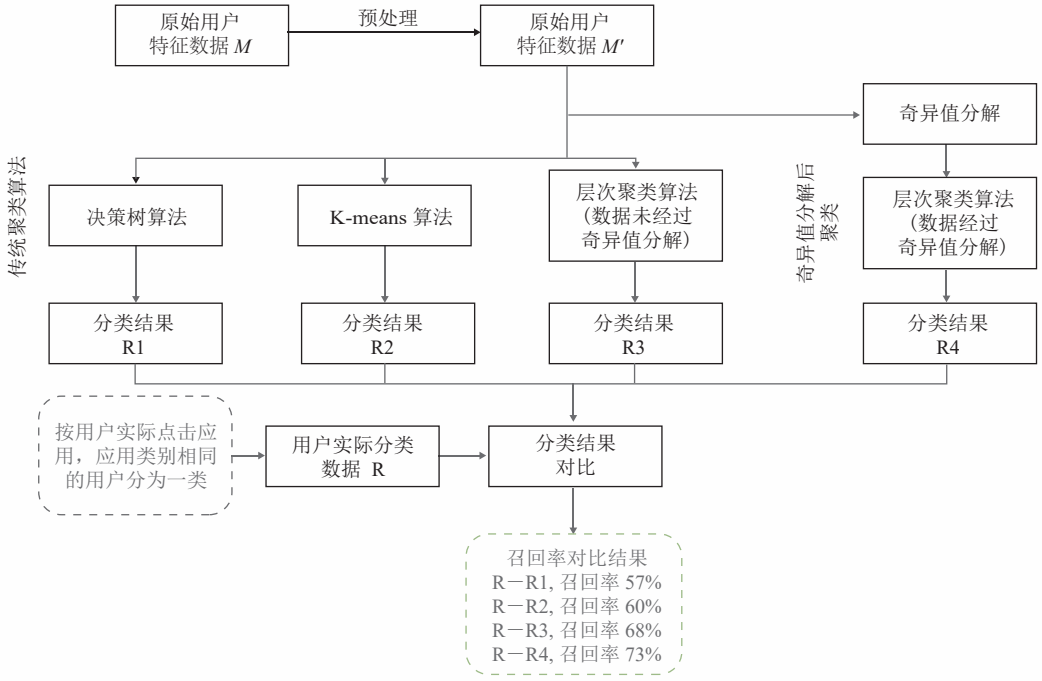

图 6 不同聚类算法得到用户聚类结果的算法流程及召回率结果

Fig. 6 User clustering algorithm procedures and their corresponding recall

过奇异值分解并进行层次聚类后，算法将采用每一个用户聚类簇的均值向量代表该用户聚类簇中的用户特征。随后，算法将联合情境特征，通过贝叶斯模型输出最后为用户推荐的应用结果列表。

为将本文算法与传统算法推荐结果的质量进行对比分析，本文采用基于协同过滤的推荐算法以及基于关联规则的推荐算法作为对比方法。在基于协同过滤的推荐算法[32]中，算法通过分析其他具有相似特征的用户的喜好，为用户推荐可能喜欢的应用。而基于关联规则的推荐算法[32]则利用发掘到的应用之间的关联规则，根据用户使用的部分应用信息为用户推荐可能感兴趣的应用。在所对比的两种方法中，均未利用奇异值分解处理特征表示、层次聚类以及用户所处的情境信息。

表 2 为不同算法针对同一用户的推荐效果。以所收集的数据集中随机抽取的用户 #0013 作为示例，在分布式框架上运行不同算法的预测应用结果中，概率值最高的前 5 个推荐结果的数据如表 2 所示。在 3 种方法中，只有本文方法为用户推荐的应用与用户实际所使用的应用相匹配，且相比其他算法而言，本文算法所推荐的其他应用也与用户实际所使用的应用具有较高的相关性，表明本文方法在推荐效果上的优越性。

5.4　推荐算法推荐结果质量量化指标对比与分析

为进一步对本文算法与现有传统的推荐算法进行推荐结果质量的量化对比，本文根据应用推荐的相关性计算了平均绝对误差(Mean Absolute Error，MAE)，即实际使用应用与推荐应用的匹配程度。推荐应用与实际使用应用类别一致时的 MAE 值低于类别不同时的 MAE 值，故更小的 MAE 值意味着更高的搜索结果质量。此外，贝叶斯模型产生的均方根误差(Root Mean Squared Error，RMSE)与推荐结果的 Recall 值也被用作算法推荐结果质量的量化对比指标。其中，更小的 RMSE 值与更大的 Recall 值都表明模型具有更理想的效果。实验对比结果如图 8 所示。

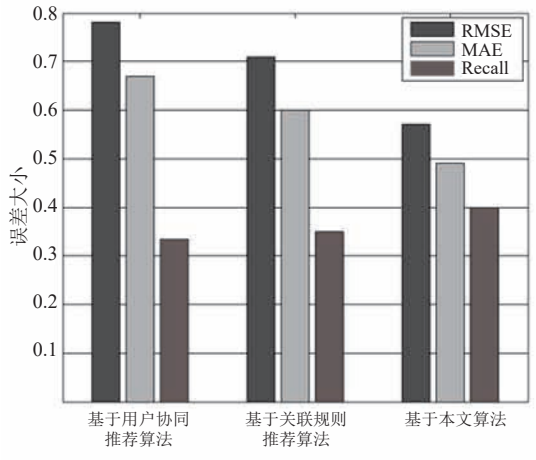

图 8　推荐算法结果量化指标对比

Fig. 8　Quantitative analysis of evaluation metrics of recommendation results

在验证过程中，本文采用了常用的 10 份交叉验证方法，并对 10 组结果求平均结果。从图 8 可明显看出，与传统推荐算法对比，本文算法拥有最小的 MAE 与 RMSE 值，且拥有最大的 Recall 值。本文方法的 MAE、RMSE 值都小于 0.6，基于关联规则算法的 RMSE 值则约为 0.7，而对于基于协同过滤的算法而言，其 RMSE 值则接近 0.8。与此同时，本文算法所产生的 Recall 值较两种对比方法而言也有较大提升。因此，这充分地从量化指标的角度表明了本文算法推荐结果的

表 2　使用不同应用算法为用户推荐应用的效果对比

Table 2　Comparison between the quality of the recommendation results produced by different algorithms

算法类别	推荐应用列表（k=5）	实际使用应用
本文算法推荐应用列表	{微博，微信，抖音，爱奇艺，拼多多}	爱奇艺
协同过滤算法推荐应用列表	{微博，微信阅读，淘宝，优酷，网易云音乐}	爱奇艺
关联规则推荐应用列表	{7973 游戏，抖音，虎牙直播，微信，QQ}	爱奇艺

有效性与准确性。

5.5 推荐系统运行性能分析

在实际测试过程中，本文还对所构建系统的性能进行了充分的测试与分析，以检测其在真实运行环境下的系统负载能力。

在性能检测过程中，通过添加 2 000 个虚拟用户，以此测试系统的平均响应时间。测试单页面用户点击时的事务平均响应时间情况如图 9 所示。经检测，系统功能的平均响应时间约为 1.4 s，最长响应时间约为 3.8 s，最短响应时间约 0.1 s，符合系统的并发要求。

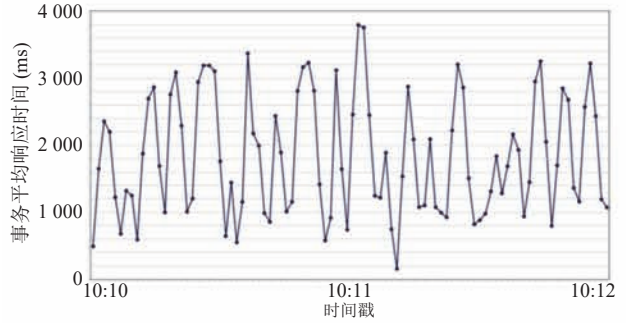

图 9 事务平均响应时间

Fig. 9 Transaction average response time

与此同时，本文还对所构建系统进行了性能测试并将结果汇总在表 3 中。测试结果充分表明了所构建系统的高效性与可靠性。

表 3 系统性能记录

Table 3 System performance records

系统性能指标	测试结果
时段	高峰时段 8 h 内
数据量级	亿级别的文本数据
吞吐量	最高 4×10^5 TPS
查询速度	1 000 条/s
处理数据速度	3×10^5 条/s
访问量并发	最高达 4 000 次/s
事务响应时间	平均请求响应时间延迟约为 1 s
最长工作时间	2 500 h

综合来看，本文推荐系统在应用推荐方面相较于所对比方法而言具有较大的提升，推荐有效性较高，且在系统性能方面高效可靠。通过实验验证与结果分析，相比于对比方法而言，本文系统在算法上的优化使得应用推荐更为准确，从而为用户更好地推荐适合的应用。同时，系统还可以接受实时更新的数据，智能化程度提升，从而减少运营复杂度。最后，本文所使用的系统架构确保了系统运行的高效性、可靠性与高扩展性，使得系统可以被更广泛的应用。

5.6 讨论与分析

本文通过在应用推荐系统中利用奇异值分解算法对用户特征信息进行降维去噪、利用层次聚类算法对用户信息进行聚类，同时考虑用户所处情境信息，从而提升应用推荐系统为用户推荐应用的准确性。相较于文献[14-15]中未在聚类前使用矩阵分解对用户特征信息进行降维并去噪的方法而言，本文方法通过奇异值分解算法，既去除了用户特征信息中的冗余成分，增强了特征信息的表示能力，又提升了算法的运行效率。此外，对用户特征信息进行层次聚类，并利用聚类均值向量作为用户类的表示有效地在不破坏用户组大体特征的情况下提升了特征的多样性，从而提升了推荐系统的推荐有效性。相较于文献[17-18]中利用用户情境信息分析产品特征的方法，本文方法利用用户情境信息分析用户的喜好特征，从而更好地辅助推荐系统为用户推荐更加适合的应用。综上所述，本文推荐系统算法相较于上文所述对比方法具有更好的应用推荐能力。

6 总 结

本文针对传统的推荐系统方法在为用户推荐应用时所面临的对用户所处情境考虑的缺失等问题，提出一种新的基于用户特征聚类联合情境特征的多维度推荐系统。所提出推荐系统使用的算法的关键技术在于，首先利用奇异值分解对用户特征进行降维、去噪，使得分解后的用户数据不再冗余，具有更强的表示能力，且更低的维度

使得所提出推荐系统运行更加高效。随后，基于分解过的用户特征对用户进行层次聚类，并用聚类簇的均值向量作为代表该聚类簇中用户的特征信息。这种做法在不破坏聚类簇内用户相似性的基础上通过求聚类簇的均值将簇内用户的特征融合，适当为用户特征引入多样性，从而避免推荐过于乏味单一导致长久以来用户对自己感兴趣的事物产生疲劳。之后，算法利用得到的用户特征联合情境特征信息组成多维度的特征并利用贝叶斯模型对推荐应用的概率进行预测，从而得到应用关于推荐概率降序排列的推荐应用列表，实现向用户进行更加高质量的应用推荐。与此同时，算法可以应对随时间发生改变的用户情境特征，从而使得推荐具有实时性。

本文基于分布式架构实现了所提出的推荐系统，并通过充分的测试数据验证，表明系统在具备准确的应用推荐能力的同时，同样具备运行的高效性与稳定性，为所提出推荐系统的应用打下坚实基础。

参 考 文 献

[1] Zhou XJ, Xu Y, Li YF, et al. The state-of-the-art in personalized recommender systems for social networking [J]. Artificial Intelligence Review, 2012, 37(2): 119-132.

[2] Sridevi M, Rao RR, Rao MV. A survey on recommender system [J]. International Journal of Computer Science and Information Security, 2016, 14(5): 265-272.

[3] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: a survey and new perspectives [J]. ACM Computing Surveys, 2019, 52(1): 1-38.

[4] Alhijawi B, Kilani Y. The recommender system: a survey [J]. International Journal of Advanced Intelligence Paradigms, 2020, 15(3): 229-251.

[5] Livinus UT, Chelouah R. Recommender system in big data environment [J]. International Journal of Computer Science, 2016, 13(5): 1-9.

[6] Khusro S, Ali Z, Ullah I. Recommender systems: issues, challenges, and research opportunities [C] // Information Science and Applications, 2016: 1179-1189.

[7] Da Silva JFG, de Moura Junior NN, Caloba LP. Effects of data sparsity on recommender systems based on collaborative filtering [C] // 2018 International Joint Conference on Neural Networks, 2018, DOI: 10.1109/IJCNN.2018.8489095.

[8] Natarajan S, Vairavasundaram S, Natarajan S, et al. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data [J]. Expert Systems with Applications, 2020, 149: 113248.

[9] Jawaheer G, Szomszor M, Kostkova P. Comparison of implicit and explicit feedback from an online music recommendation service [C] // Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, 2010: 47-51.

[10] Lerato M, Esan OA, Ebunoluwa AD, et al. A survey of recommender system feedback techniques, comparison and evaluation metrics [C] // 2015 International Conference on Computing, Communication and Security, 2015, DOI: 10.1109/CCCS.2015.7374146.

[11] Vargas S, Baltrunas L, Karatzoglou A, et al. Coverage, redundancy and size-awareness in genre diversity for recommender systems [C] // Proceedings of the 8th ACM Conference on Recommender Systems, 2014: 209-216.

[12] 王国霞, 刘贺平. 个性化推荐系统综述 [J]. 计算机工程与应用, 2012, 48(7): 66-76.
Wang GX, Liu HP. Survey of personalized recommendation system [J]. Computer Engineering and Applications, 2012, 48(7): 66-76.

[13] Kobren A, Monath N, Krishnamurthy A, et al. A hierarchical algorithm for extreme clustering [C] // Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017: 255-264.

[14] Shinde SK, Kulkarni U. Hybrid personalized

recommender system using centering-bunching based clustering algorithm [J]. Expert Systems with Applications, 2012, 39(1): 1381-1387.

[15] Hsu MH. A personalized English learning recommender system for ESL students [J]. Expert Systems with Applications, 2008, 34(1): 683-688.

[16] Kim JH, Lee D, Chung KY. Item recommendation based on context-aware model for personalized u-healthcare service [J]. Multimedia Tools and Applications, 2014, 71(2): 855-872.

[17] Hu YF, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets [C] // 2008 Eighth IEEE International Conference on Data Mining, 2008: 263-272.

[18] Abbas SM, Khan T, Khalid S. Improved context-aware YouTube recommender system with user feedback analysis [J]. Bahria University Journal of Information & Communication Technologies, 2017, 10(2): 1-8.

[19] Yang B, Tong KK, Zhao XQ, et al. Multilabel classification using low-rank decomposition [J]. Discrete Dynamics in Nature and Society, 2020, DOI: 10.1155/2020/1279253.

[20] Rui X, Wunsch D. Survey of clustering algorithms [J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645-678.

[21] Pavlov D, Balasubramanyan R, Dom B, et al. Document preprocessing for naive Bayes classification and clustering with mixture of multinomials [C] // Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004: 829-834.

[22] Elbanoby Y, Aborizka M, Maghraby F. Real-time data management for IoT in cloud environment [C] // 2019 IEEE Global Conference on Internet of Things, 2019, DOI: 10.1109/GCIoT47977.2019.9058394.

[23] MySQL. MySQL 8.0 Reference Manual [EB/OL]. 2020[2021-02-24]. https://dev.mysql.com/doc/refman/8.0/en/.

[24] Apache HBase. Apache HBase™ Reference Guide [EB/OL]. 2020[2021-02-24]. https://hbase.apache.org/book.html.

[25] Jiang L, Li B, Song MN. The optimization of HDFS based on small files [C] // 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology, 2010.

[26] Apache Spark. Apache Spark Documentation [EB/OL]. 2020[2021-02-24]. https://spark.apache.org/docs/3.0.0/.

[27] Apache Kafka. Apache Kafka Documentation [EB/OL]. 2020[2021-02-24]. https://kafka.apache.org/documentation/.

[28] ElasticSearch. Elasticsearch Guide [EB/OL]. 2020[2021-02-24]. https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html.

[29] Java 2 Platform. Enterprise Edition (J2EE) Overview [EB/OL]. 2020[2021-02-24]. https://www.oracle.com/java/technol ogies/appmodel.html.

[30] Apache Tomcat. Apache Tomcat Documentation Index [EB/OL]. 2020[2021-02-24]. http://tomcat.apache.org/tomcat-8.5-doc/.

[31] Apache Mahout. Apache Mahout Documentation [EB/OL]. 2020[2021-02-24]. https://mahout.apache.org/docs/latest/.

[32] Nagarnaik P, Thomas A. Survey on recommendation system methods [C] // 2015 2nd International Conference on Electronics and Communication Systems, 2015: 1603-1608.