分 类 号_____ 学号 D201477747

学校代码 10487 密级 _____

华中科技大学

博士学位论文

基于张量的大数据多聚类及其安全和高效方法研究

学位申请人： 赵雅靓

学 科 专 业： 计算机系统结构

指 导 教 师： 杨天若

答 辩 日 期： 2019 年 5 月 20 日

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Engineering

Tensor-Based Big Data Multiple Clusterings with Their Secure and Efficient Implementations

Ph. D. Candidate : Yaliang Zhao

Major : Computer Architecture

Supervisor : Prof. Laurence T. Yang

Huazhong University of Science & Technology

Wuhan 430074，P. R. China

May. 20, 2019

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本文属于　保　密□，在_____年解密后适用本授权书。

　　不保密□。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　　年　月　　日　　日期：　　年　月　日

摘 要

随着云计算、物联网、社交网络和社会新媒体等高新信息技术的飞速发展，现实世界大量的感知设备、智能产品、网络通信，以及人类知识、思维能力、社会关系和文化元素，从多个维度产生了大规模的多源异构数据，这些数据具有特征混杂、模态多样、类型复杂等特点，并在不同的视图下蕴含着不同的知识和价值。多聚类可以从不同观点产生多个不同的聚类结果，有利于从多方面揭示隐藏在数据中的不同结构，是解决网络舆情分析、重大疾病分析、资源推荐和金融风险预测等众多应用问题的关键技术，在我国社会、工业和经济领域有着迫切的需求，具有广阔的应用前景。

现有多聚类研究大多针对小规模、单领域数据集，其聚类结果难以解释，无法根据上下文情境变化实现多模态的聚类，且算法大多面向具体应用，难以扩展到其他领域，缺乏通用性。此外，在大数据时代，大数据所呈现的类型多样、数据规模大、价值密度不均、增长速度快等特征也对大数据环境下的多聚类研究提出了新的挑战。论文选定大数据环境下面向多源异构数据的多聚类作为主要研究对象，围绕基于张量的大数据多聚类及其安全和高效实现两大方面展开理论、技术及方法的系列研究。主要研究内容和创新成果如下：

首先，针对大数据环境下的多聚类，为了度量所有特征空间中属性组合的重要性，提出了一种基于多线性属性排名的权重学习方法，进而提出基于可选择加权张量距离的张量多聚类方法。此外，为了提高多聚类质量，一方面在计算距离时能保证所选择的特征与未选择的特征完全分离；另一方面针对如何去除数据中的噪声和冗余，提出基于张量分解的多聚类方法。同时为了提高基于张量分解的多聚类方法的性能，提出一种用于各特征空间属性重要性度量的多关系属性排名方法。实验表明提出的多聚类方法具有较高的聚类准确率和较低的冗余度。

其次，在云计算环境中，为保护用户隐私，提出云端安全的张量多聚类方法。通过研究多聚类算法的云端安全计算模式，设计混合云模型下云端安全的多聚类分析和服务框架，提出一种安全高阶密度峰值聚类方法，进而提出了安全张量多聚类方法以及相关的多种安全子协议，并给出了安全性证明。实验结果表明这些方法完全可以保

证用户的隐私安全、100%的聚类准确率和较高的伸缩性与数据可用性，且客户端十分轻量级，算法具有高可扩展性。

再次，针对维度灾难和高效计算问题，提出基于张量链分解的张量多聚类及其并行计算方法。基于张量链分解形式下各张量基本运算的计算规律，分别提出基于张量链分解的多线性属性组合权重学习算法和基于张量链分解的可选择加权张量距离，进而提出基于张量链分解的张量多聚类方法。该方法能够实现在张量链分解的形式下完整的张量多聚类过程，并能保证甚至提高聚类结果的准确性。此外，在云计算分布式环境中，依据节点计算能力和通信能力设计高效的分布式并行计算框架，通过研究张量链核分配机制、核调度以及核运算的并行策略，提出了基于张量链核的分布式并行策略，以充分利用张量网络并行计算优势来提高张量多聚类算法的并行效率。

最后，针对大数据动态增长带来的大量重复计算问题，提出张量多聚类的增量式更新方法，包括增量式密度峰值聚类和增量式张量多聚类。针对张量多聚类方法，分别提出了基于迭代的属性权重增量学习方法和基于微分的属性权重增量学习方法，并基于一种简单快速的 K-medoids 算法设计相应的增量式 K-medoids 算法，使得在多聚类增量时不需要计算全部距离，从而有效提高张量多聚类的增量更新算法效率。实验结果表明，所提出的增量式密度峰值聚类与同类方法相比具有较高的聚类准确率和效率，而提出的增量式张量多聚类方法，不仅能保证增量更新的较高准确率，而且能够极大程度提高多聚类分析中数据动态增量更新维护的效率。

论文所提出的基于张量的大数据多聚类及其安全和高效方法可为多聚类理论研究提供新的有益思路，同时也将促进多聚类分析在大数据时代的应用与发展。

关键词：大数据，张量，多聚类，安全计算，分布和并行计算，增量更新。

Abstract

With the rapid development of high-tech information technologies such as cloud computing, Internet of Things, social networking and social new media, there are a large number of sensing devices, intelligent products, network communications, and human knowledge, thinking skills, social relations and cultural elements in the real world. These produce large-scale multi-source heterogeneous data, which are characterized by mixed features, diverse modalities, and complex types, and contain different knowledge and values in different views. Multiple clusterings can generate multiple different clustering results from different perspectives, which is beneficial to reveal different structures hidden in the data from many aspects, and it is known as the key technology to solve many problems such as network public opinion analysis, major disease analysis, resource recommendation and financial risk prediction. This technology has urgent needs in social, industrial and economic fields, and has broad application prospect.

Most of the existing multiple clustering researches are aimed at small-scale, single-domain datasets. The clustering results are difficult to interpret, and multi-modal clustering cannot be realized according to contextual changes. Most of the algorithms are specific to specific applications, and it is difficult to extend to other fields, even it is lack of versatility. In addition, in the era of big data, the characteristics of big data such as diverse types, large data size, uneven value density and fast growth rate also pose new challenges to the multiple clustering research in the big data environment. This thesis selects the multi-source and heterogeneous data clustering in the big data environment as the main research object, and carries out a series of theoretical, technical and method studies focusing on tense-based big data multiple clusterings and their secure and efficient implementation. The main research contents and innovations are as follows:

Firstly, for multiple clusterings in big data environment, in order to measure the importance of attribute combinations in all feature spaces, a weight learning method based on multi-linear attribute ranking is proposed, and then a multiple clustering method based on selective weighted tensor distance is proposed. Besides, in order to improve the quality of clustering, on the one hand, the selected features can be completely separated from the unselected features when calculating the distance, and on the other hand, On the other hand,

how to remove noise and redundancy in the data, so a tensor decomposition-based multiple clustering method is proposed. At the same time, in order to improve the performance of tensor decomposition-based multiple clustering method, a multi-relational attribute ranking method for each feature space attribute importance measures is proposed. Experiments show that the proposed multiple clustering methods have higher clustering accuracy and lower redundancy.

Secondly, in the cloud computing environment, for the purpose of preserving user privacy, a secure tensor-based multiple clustering method on cloud is proposed. By researching the cloud secure computing mode of multiple clustering algorithm, designing multiple clustering analysis and service framework of cloud security under hybrid cloud model, a secure high-order density peak clustering method is proposed. Furthermore, a secure tensor-based multiple clustering method and related secure sub-protocols are proposed, and the proofs of security are provided. The experimental results show that these methods can guarantee the user's privacy security, 100% clustering accuracy and high scalability and data availability, and the client is very lightweight and the algorithm is highly scalable.

Thirdly, for the dimension disaster and efficient computing problems, a tensor-based multiple clustering based on tensor train decomposition and its parallel computing method are proposed. Based on the calculation rules of the basic operations of tensor in the tensor train decomposition form, the multi-linear attribute combination weight learning algorithm based on tensor train decomposition and the selective weighted tensor distance based on tensor train decomposition are proposed respectively, and then the tensor-based clustering method based on tensor train decomposition is proposed. This method can realize the complete the multiple clustering process in the form of tensor train decomposition and can guarantee or even improve the accuracy of clustering results. Moreover, in the cloud computing distributed environment, an efficient distributed parallel computing framework is designed according to the computing power and communication ability of the nodes. By studying the parallel strategy of tensor train core allocation mechanism, nuclear scheduling strategy and the parallel strategy of core operations, a distributed parallel strategy based on tensor train core is proposed to fully utilize the tensor network parallel computing advantage to improve the parallel efficiency of the tensor-based multiple clustering

algorithm.

Finally, in view of the large number of repeated calculations caused by the dynamic growth of big data, an incremental update method for tensor-based multiple clusterings is proposed, including incremental density peak clustering and incremental tensor-based multiple clusterings. For the tensor-based multiple clustering method, the iterative-based attribute weight increment learning method and the differential-based attribute weight increment learning method are proposed respectively. A simple and fast K-medoids algorithm is used to design the corresponding incremental K-medoids algorithm, so that it is not necessary to calculate all distances, thus effectively improving the efficiency of the tensor-based multiple clustering incremental update algorithm. The experimental results show that the proposed incremental density peak clustering has higher clustering accuracy and efficiency than the similar methods, and the proposed incremental tensor-based multiple clustering method can not only ensure incremental update, but also greatly improve the efficiency of dynamic incremental update maintenance of data in multiple clustering analysis.

The proposed tensor-based big data multiple clusterings and their secure and efficient methods in the thesis can provide new useful ideas for multiple clustering theory research, and also promote the application and development of multiple clustering analysis in the era of big data.

Keywords: Big Data, Tensor, Multiple Clusterings, Secure Computing, Distributed and Parallel Computing, Incremental Updating.

目 录

1 绪 论

1.1 研究背景及意义

1.1.1 研究背景

多聚类（multiple clusterings）作为数据挖掘的一个新兴研究领域，近年来受到各领域学者的极大关注[1,2]。随着云计算、物联网、社交网络和社会新媒体等高新信息技术的飞速发展，现实世界大量的感知设备、智能产品、网络通信，以及人类知识、思维能力、社会关系和文化元素，从多个维度产生了大规模的多源异构数据，这些数据具有特征混杂、模态多样、类型复杂等特点，并在不同的视图下蕴含着不同的知识和价值[3]。同时，在许多实际应用中，从多个分析任务[2] 收集大量数据，需要根据不同需求对数据聚类从而产生不同的分组[4]。例如，在智慧公交大数据应用中，数据包含车载视频数据、IC卡刷卡记录（线路、站点、乘客、车辆、时间）、车载GPS轨迹数据以及公交信息化系统中的XML文档数据、天气等信息，对车辆一个可能的分组是根据车载视频数据、线路、站点、时间、天气等特征，公交集团可以据此预测公交客流量；同时，另一个分组可以根据车载GPS轨迹数据、线路、站点、乘客、时间和天气等特征对其聚类，聚类结果可为公交乘车推荐系统提供高效服务。相对于只关注发现对象单一分组的传统单聚类，多聚类可以从数据的不同观点产生多个不同的聚类结果，多方面揭示隐藏在数据中的不同结构，更好地满足当今大数据多分析任务的需求。多聚类分析理论是当前数据挖掘和知识发现研究的最新进展，也是最活跃的领域，国际顶级会议KDD、ICDE、ICDM等最近几年发表了多篇论文报导这方面的研究成果，尤其在2015年1月《Machine Learning》杂志出版了一辑关于多聚类分析的特刊。通过多聚类理论研究可以促使人们更加全面挖掘现实世界对象的复杂关系，为开展高效准确的多分析任务提供理论依据。多聚类是解决网络舆情分析、重大疾病分析、资源推荐和金融风险预测等众多应用问题的关键技术，在我国社会、工业和经济领域有着迫切的需求，具有广阔的应用前景。

目前国内外在多聚类分析方面的研究主要集中在多视图聚类、可选择聚类和子空

间聚类[1]。多视图聚类能够通过融合多源信息来挖掘数据的内在结构，比单一视图聚类具有更好的聚类性能，但不能从多角度选择不同特征的组合产生不同的聚类结果；可选择聚类能够挖掘数据的不同模式，提供多个不同的聚类结果供用户选择，更加符合人类看待世界的多样性特点，但主要针对小规模、单领域数据集，不能结合多视图信息来提高聚类性能；子空间聚类主要用于高维数据聚类，能够通过提取的子空间发现良好的类簇，但不能从不同的观点产生多个不同的聚类结果。近年来，也有一些研究结合上述方法来改进多聚类，但仍都主要针对低维、小规模、单领域数据集（如图像数据），其聚类结果难以解释，且无法根据上下文灵活变化的聚类对象为不同的应用提供按需服务；而且算法大多面向具体应用，难以扩展到其他领域，缺乏通用性。

此外，由于大数据具有“4V”特征，即规模庞大（Volume）、类型多样（Variety）、增长快速（Velocity）和价值密度不均（Value），这些特征对大数据环境下的多聚类提出了新的挑战，并将对现有多聚类分析的计算模式、理论和方法产生深远的影响。首先，随着数据量的增长和数据维度的不断增大，有限的计算和存储资源很难实现实时的多聚类分析，借助强大的云计算能力来进行多聚类可以有效提高其计算和存储效率。但将聚类算法部署在云平台为用户提供聚类服务时，用户需将数据外包给云进行分析，这也给用户稳私带来了严重的威胁，因此隐私保护数据挖掘技术成为当前研究热点[36]。其次，在大数据环境下，随着数据规模的不断增大，数据存储、计算负荷、内存开销等都将呈指数级增长，从而引起维度灾难问题。数据压缩技术和分布式并行处理方式被广泛应用于大数据处理中，聚类算法也大多采用这些技术来提高聚类过程的效率[50]。最后，数据对象、数据特征空间实时动态变化，聚类结构也需要动态更新，针对流式增量数据的更新技术成为解决大数据快速增长的有效手段[60]。但目前国内外关于聚类算法的隐私保护技术、分布式并行计算模型和增量更新模式，仍主要针对传统的单聚类算法，缺乏面向多聚类分析领域的扩展，从而制约了大数据环境下多聚类分析的应用与发展。

因此，为了应对大数据的来源多样、特征高维、关系复杂、规模庞大和生成快速等问题，充分发挥现有多聚类方法的优势，同时对多聚类过程进行实时安全和高效处理，创新研究一种面向大规模多源异构数据的多聚类方法，并探索其可行的安全和高

效计算方法，是大数据时代多聚类研究的核心问题。

1.1.2 研究意义

大数据环境下的多聚类可以全方位挖掘数据内在关联结构，多视角获取蕴含知识的本质特征。然而，针对大规模多源异构数据的多聚类理论和方法研究，需要考量多聚类模型的通用性和有效性以及多聚类算法的安全性、时效性和动态性，将主要面临如下难点：

首先，实际应用的需求随时间、空间和情境而变化，不同情境下看待数据的观点也不尽相同，用户希望可以根据上下文情境变化动态选择多源异构数据的不同特征来聚类，并获取期望的聚类结果。在构建面向实际应用需求的多聚类方法时，需要考虑动态选择特征空间组合的灵活性以及聚类结果的可解释性、非冗余性、高质量性，以满足大数据多分析任务的要求。因此，构建面向大数据的多聚类方法是实现多聚类分析及其安全和高效计算的关键前提。

其次，在云端进行多聚类分析时，保护用户的隐私成为亟待解决的关键问题，而隐私保护的聚类算法设计难点在于寻求隐私保护、数据可用性、算法准确率和效率之间的平衡。在设计云端安全的多聚类方法时，需要考虑保护信息的完整性，即聚类过程中没有任何额外信息泄露，聚类结果的可用性，并且在保证多聚类结果高准确率同时能够极大程度的提高安全算法效率。因此，设计实现有效的安全多聚类方法，其技术难点的解决仍需要深入研究。

再次，当采集的多源异构数据集较大、数据特征维度较高时，数据存储、计算负荷、内存开销等都将呈指数级增长，从而引起维度灾难问题。此外，由于这些数据大多随机动态产生，包含大量的噪声数据或冗余数据，从而极大影响了多聚类分析的效率和质量。因此，在大数据环境下，为了提高多聚类算法的效率，需在降低数据存储的同时求取高质量核心集，并研究设计高效的分布式并行策略以满足实际应用需求，其技术方法的实现面临新的挑战。

最后，实际应用中数据对象、数据特征空间实时动态变化，特别是随着时间的增长，数据将会不断累积，从而引起聚类结构的不断动态更新。然而，在已有多个聚类结果基础上同时对新增对象进行多聚类时，由于原有对象之间的相异性结构并未发生

改变，若对所有对象完全重新聚类将会导致产生大量的重复计算。因此，为解决大数据环境下针对动态实时更新数据的多聚类高效计算问题，研究设计多聚类算法的增量更新机制，尚需新的技术手段。

张量模型是一种新型的大数据表示模型，它的优势在于能够融合多源异构数据，能更加自然的表示多维度的数据，且基于张量的大数据分析和处理方法能够考虑数据内部的关联，因此它通常在大数据表示上具有更好的性能，可广泛应用于社区探测、个性化推荐、图像识别、交通预测等大数据应用领域。

因此，为了解决前述难点问题，本文首次采用张量代数理论，主要围绕如下关键问题进行深入和系统研究：基于张量代数理论、矩阵理论、多线性空间理论，研究构建基于张量的多聚类方法，通过多方面挖掘数据的隐含类簇，实现大数据的多模态聚类，为后续研究奠定理论基础；基于混合云模型和同态加密体系，研究张量多聚类的云端安全计算模式，在实现大数据多模态聚类的同时能更好的保护用户隐私；设计基于张量链分解的张量多聚类算法，以期在降低存储规模的同时提取高质量数据，进而研究其分布式并行策略，为大数据的多聚类高效计算提供理论支撑；针对动态实时更新的数据，研究设计高效的张量多聚类增量更新机制，并在增量计算的同时保证多聚类结果的准确性。

本文提出的基于张量的大数据多聚类方法，可以满足不同情境下的不同需求，实现多模态聚类。而在此基础上提出的安全的、分布式并行的以及增量更新的张量多聚类方法，可以提供云端隐私保护和实时高效的多聚类服务，对大数据时代的数据分析和智能应用具有重要的实际意义。

1.2 国内外研究现状

面向大数据环境下的多聚类理论，需要考虑构建满足大数据不同应用需求的灵活、有效的通用模型并进行算法设计，并对聚类方法的安全性、高效性和动态性进行深入分析。下面将从数据对象的简约表示、多聚类分析方法、隐私保护聚类方法、聚类算法的并行化及增量式聚类更新方法等方面介绍国内外研究现状。

1.2.1 数据对象的简约表示

为了便于对大数据环境下多聚类对象进行有效分析及安全和高效计算，需要对多源异构复杂数据进行简约表示。近年来，很多学者做了大量的工作。目前与本论文最为相关的研究集中在基于张量、张量分解和张量网络分解模型的数据表示等方面。

很多学者利用三维张量模型对异构数据进行表示，在[5]中，Symeonidis 等利用三维张量模型将地理位置信息与社会网络信息结合，通过位置跟踪服务和用户提交的个性化评价信息为用户提供推荐服务，包括朋友、位置和活动等。对于大量时间序列类型的数据处理，Sun 等[6]提出了一种采用动态三维张量对数据进行分析的方法，将数据统一到动态三维张量模型中进行描述。为了解决社会化标签推荐系统中数据极度稀疏的问题，在[7]中，Symeonidis 等提出了一种能够完整表示社会网络中用户、标签、资源三维数据主要特征的张量模型。国内也有学者通过三维张量来解决三部图转换过程中元组丢失的问题[8]。张量模型能够很好的描述现实世界中事物之间的关系，但是随着张量维度的增加，数据存储、计算负荷、内存开销等都将呈指数增长，从而引起维度灾难问题。

所以，也有学者提出张量分解模型数据表示。CP 分解[9]是把一个张量分解成多个向量的外积之和，可以把原来的指数关系降到线性关系，但当面对高阶张量时，其最优秩是个 NP 问题，同时其分解算法也不稳定。Tucker 分解[10]是把原始张量分解成一个核心张量和多个因子矩阵，这种分解对于高阶张量而言，算法是稳定的，但分解后的核心张量，维度依然是在指数上，仍存在维度灾难问题。近年来，张量网络作为一种用于分析处理高维大数据的新兴技术，成为目前国际上的一个研究热点。不同于张量分解，它是把原始张量分解成多个低阶低维的核心张量。HT（层次 Tucker）分解[11]，其分解算法需要递归，所以实现开销较大。TT（张量链）分解[12]作为 HT 的一种最简单方式，它是把原始张量分解成一个链的方式，分解后的张量都是低阶低维的，一般都是三阶的。其优点是参数较少，接近于 CP 分解，且算法稳定不需要递归。但是如果针对低阶高维张量，例如大规模的向量或矩阵，由于阶数很低，TT 的优势就明显降低，所以出现了 QTT（量化张量链）分解[13]。这种方法首先是把低阶的向量或者矩阵量化为一个张量，然后再使用 TT 分解。这样，可以把许多针对向量、矩阵的大规

模问题变成小规模的问题来进行求解。张量网络由于其极好的压缩能力和分布式并行处理方式而被认为是分析大数据的非常有前途的工具，为本论文的研究奠定了良好的理论基础。

总之，数据对象的简约表示是：数据的适当张量化，或进一步以分布式张量网络近似表示它们，并以张量或张量网络形式执行所有运算操作。但目前针对张量和张量网络的研究主要是面向具体分解模型和算法，鲜有提出与多聚类结合的理论成果发表。因此借鉴现有成果的有益思路，研究构建基于张量理论的对象模型及其相似度度量方法，是进行大数据环境下多聚类分析及其安全和高效计算的关键前提。

1.2.2 多聚类方法

多视图聚类、可选择聚类和子空间聚类是多聚类分析的重要研究方向，目前相关的国内外研究主要集中在结合子空间理论、随机过程、矩阵论、图谱理论和信息论等学科理论。

在多视图聚类方面，Zhang等[14]提出了针对低秩张量的多视图子空间聚类算法LT-MSC，该算法将不同视图的子空间表示矩阵表示在一个具有低秩限制条件的张量中，从而极好地为不同视图之间的交叉信息建立模型。Xia等[15]结合马尔可夫链提出了一种鲁棒的多视图聚类算法，该算法首先为每个单一视图构造转移概率矩阵，然后用这些矩阵来重新获得一个共享的低秩转移概率矩阵作为标准马尔可夫链的关键输入，并依此来进行聚类。针对现实数据中可能不存在视图间的映射这一问题，Zhang等[16]提出了基于NMF聚类框架的受限多视图聚类算法。Liu等[17]给出了一种基于NMF的多视图聚类算法，通过构造一个有约束的联合矩阵分解过程，给出在多个视图中兼容的聚类解决方案。通过引入联合结构化稀疏诱导准则学习不同聚类中特征的权重，Wang等[18]提出了一个集成所有异构特征的多视图学习模型，该模型不仅可以用于多视图聚类，也可以用于处理分类任务。Nie等[19]假设所有视图有相同的潜在类，但是每个视图包含不完整的信息，基于此假设提出一种新的自动加权多视图学习框架（AMGL），该框架可以自动学习所有图的权重且无需任何参数。Pradhan等[20]根据多种形式的信息，利用多视图聚类来对用户或者商品进行聚类，并将其应用到基于协同过滤（CF）的评级预测系统中，提高了该系统的准确度。

在可选择聚类方面，基于谱聚类理论，[21]等通过寻找拉普拉斯矩阵不同特征向量发现多个稳定且具有最大相异性的可选择聚类结果。结合信息论思想，Truong等[22]提出基于进化算法的多目标优化方法同时聚类，生成一个Pareto最优聚类结果集。将寻找可选择聚类结果作为一个迭代优化问题，Kontonasios等[23]用最大熵的概念来挖掘多个不同的可选择聚类结果。结合子空间聚类思想，Günnemann[24]等提出一种贝叶斯框架，通过对数据建立在子空间投影中具有不同贝塔分布的多个混合模型来产生多个相异的聚类结果；并在此基础上通过引入用户先验知识，进一步提出一种半监督可选择聚类方法，以增加可选择聚类结果的数量[25]。为使可选择聚类更有意义，Tatu等[26]采用导入兴趣度子空间搜索算法选择一组候选子空间，基于定义的子空间相似度函数和提供的可视化导航交互式地探索对大数据集的子空间进行聚类。采用特征选择方法，Tao等[27]将原始数据集转换到具有加权特征的数据子空间，以便产生不同的聚类结果。Yang等[28]基于NMF提出了一种可替代聚类算法，利用其非负特性来保证聚类结果间的非冗余性，并且他们还设计一个二次项来度量参考聚类和新聚类间的冗余度。

在子空间聚类方面，CLIQUE[29]、MAFIA[30]和DBSCAN[31]是一些经典的子空间聚类方法，主要基于网格单元探索所有潜在的子空间，找到密集的类簇。近年来也有一些新的子空间聚类方法相继提出。Elghazel[32]集成监督和无监督特征选择方法，并通过从不同的特征集得到多个聚类找到一个共识来强化该方法。Wang等[33]研究了低维子空间中加入敌对噪音或随机噪音的稀疏子空间聚类（SSC）的特性并改进了SSC算法，使其更有效地识别底层的子空间。You等[34]提出一种基于自表现的子空间聚类算法SSC-OMP，它采用正交匹配追踪（OMP）来找到稀疏表示，替代基于L1的BP方法。针对传统聚类算法在对时间序列数据聚类时较难发现平滑子空间问题，Huang等[35]提出一种新的K-means类型平滑子空间聚类算法TSKmeans，该算法可以有效利用时间序列数据集的固有子空间信息以增强聚类性能。

总的来说，现有多视图聚类能够通过融合多源信息提高聚类性能，但也只产生对象的单一分组，无法从多角度选择不同特征的组合产生不同的聚类结果；而可选择聚类可以产生多个聚类结果，但主要针对小规模、单领域数据集，且仅以多个聚类结果之间相异性最大化作为唯一评价标准，聚类结果难以解释；子空间聚类用于高维聚类，

能够通过提取的子空间发现良好的类簇，但也不能从数据不同的观点产生多个不同的聚类结果。因此，充分发挥现有多聚类方法的优势，创新研究一种大数据环境下的多聚类方法，其技术难点的解决需进行深入研究。

1.2.3 隐私保护聚类方法

近年来，支持任意背景知识下可以保护用户隐私信息的海量数据挖掘算法已成为研究热点。聚类作为重要的数据挖掘技术，针对隐私保护聚类的研究一直受到学术界的广泛关注。流行的用于隐私保护聚类的技术主要有两种：数据失真和数据加密。

数据失真技术[36,37]是通过添加噪声等手段将数据扰乱或随机化以保护数据隐私的，是以牺牲数据的部分准确性和真实性为代价的。差分隐私是一种基于数据失真的隐私保护技术，对敏感数据添加噪声的同时保持数据的统计特征的性质。差分隐私聚类[38-41]中定义了一个极为严格的攻击模型，可以抵抗具有任意背景知识的攻击，并对隐私泄露风险给出了严谨、定量化的表示和证明，能够在大大降低隐私泄露风险的同时，极大地保证数据的可用性，但目前差分隐私保护技术大都基于可信的数据挖掘者，并不适合外包环境下的聚类算法。

数据加密技术是通过对数据加密来保护数据隐私的，并给出正式的安全性证明，在准确率和安全性方面优于数据失真技术。目前基于加密技术的聚类算法研究主要集中在K-means聚类中，Vaidya[42]提出了基于垂直分布数据的隐私保护K-means聚类方法，在三方半诚实模型下使用基于加密的安全协议实现垂直分割数据的隐私保护K-means聚类。在聚类过程的每次迭代中，进一步通过乱码电路协议进行数据点之间不同距离的安全大小比较，从而安全地实现了将数据点分配到相应的中心点的过程。但该算法只适用于数据垂直分布情况下的简单K-means聚类。在此基础上，Jagannathan等[43]提出一种针对任意分布数据的隐私保护的分布式K-means聚类算法，两方分别计算出2k个簇，然后依照距离相加最小的原则对这些簇进行归并，进而得到k个簇，其中寻找最小距离的安全比较协议通过乱码电路实现，但是该方法只适用于两方计算的K-means聚类，且通信成本较高。基于此工作，该团队[44]又提出一种高效通信的隐私保护聚类算法，基于一种新设计的K-means聚类算法提出了其隐私保护方法，其中乱码电路仍然用于安全地寻找数据点之间的最小距离，但该方法只能用于水平分割的数

据集。Beye等[45]重新针对K-means聚类算法提出了一种基于三方计算的高效隐私保护算法，其中除使用了安全比较乱码电路外，还提出了实现安全除法的乱码电路。Jha[46]开发了两个协议，仅需两方来实现在水平分割集上的隐私保护K-means聚类。Bunn[47]使用Paillier密码系统和安全标量乘实现两方的K-means聚类。Sakuma等[48]为一种应用于大规模K-means聚类的新协议，但需要多个非共谋方。尽管上述方法使用相对简单的半诚实模型，但所提出的协议执行过程中都有额外信息的泄露，如中间结果的分配、每个类簇中对象的数量、迭代的次数等。此外，Zhang等[49]提出了一种BGV全同态加密的隐私保护高阶密度峰值聚类算法。然而，由于一些运算（如安全比较和除法等）无法实现，除了相似度可在密文上计算外，密度峰值聚类仍然在明文上执行。

总之，目前隐私保护的聚类方法研究主要集中在K-means、密度峰值聚类等传统单聚类方法上，均无法提供完整的信息保护，鲜见面向多聚类相关隐私保护算法的成果发表。而在云端进行安全的多聚类计算时，需要考虑在保护用户隐私的同时，还要保证数据的可用性和聚类结果的准确率并能极大程度的提高算法效率。因此，借鉴现有隐私保护聚类的有益思路，实现多聚类的云端安全计算模式，仍需进行深入研究。

1.2.4 聚类算法并行化

并行化是大数据环境下提高聚类算法效率的有效技术手段，国内外学者在聚类算法并行化方面已取得了较大进展。

Kim等[50]提出了针对大数据集的基于密度的并行聚类算法DBCURE，可以根据不同密度找到聚类且适用于MapReduce并行编程，并提出了适合MapReduce框架且可以并行找到一些类簇的DBCURE-MR算法。针对大规模数据流对象，Wu等[51]讨论了GPU和CUDA平台的动态并行性特点对BRICH的益处，通过GPU加速减少聚类时间并能够适应数据集的可扩展性。Chen等[52]通过保留最近邻的方式来稀疏化稠密矩阵并得到其近似表示，针对该策略他们提出了并行谱聚类PSC算法，对稀疏化过程中的内存使用和数据计算采用分布式并行处理，并给出了系统的解决方案。Ferreira Cordeiro等[53]采用MapReduce对大型高维数据的子空间聚类进行分布式处理，并针对I/O成本、进程节点间网络成本的最小化问题提出了BoW算法，以实现两者的均衡。Yan等[54]提出了并行快速迭代聚类p-PIC算法，他们将数据分割成小块并分布存储在多台机器上，进

而解决PIC算法在存储大型数据的相似度矩阵时所面临的内存问题。鲁伟明等[55]提出了基于MapReduce的分布式近邻传播聚类算法DisAP，该算法先将数据点随机划分为规模相近的子集，进而并行地对各子集进行AP聚类，最后融合各子集结果再次进行AP聚类得到最终聚类结果。结合细粒度并行思想，朱红等[56]提出了基于改进属性约简的细粒度并行AP聚类算法IRPAP，先用改进的基于差别矩阵的属性约简算法对属性进行约简，然后用缓冲区同步机制并行处理数据点之间的消息计算，最终实现AP聚类。赵卫中等[57]基于云计算平台Hadoop实现了并行K-means聚类算法。Lee等[58]提出了一个运行在Pregel上的图聚类算法，Pregel是由Google提出的适用于处理大规模图数据的分布式并行框架。王韬等[59]基于弹性分布式数据集（Resilient Distributed Datasets, RDDs）提出了分布式聚类集成算法DisCE。

综上所述，当采集的多源异构数据集较大、数据特征维度较高时，分布式并行策略能有效提高算法效率，而目前针对聚类的并行计算模型大都是面向单视图、单聚类结果的，尚少见面向多聚类相关算法的成果发表。因此，针对多聚类的分布式并行计算模型的研究有待做进一步探索。

1.2.5 聚类算法增量式更新

为了解决数据动态实时更新带来的重复计算问题，国内外学者在增量式聚类方面进行了相关研究，目前已取得了一些初步进展。

Sun等[60]利用K-Medoids和NNA的思想改进AP聚类算法，提出增量AP聚类算法。Zhang等[61]提出了一个增量子空间聚类算法，首先使用子空间聚类模型CC-Cluster对时间点子集上的数据流子群的一致性变化模式进行捕捉，然后将当前模式和潜在的未来模式组织成一个有向无环图pDAG，通过动态更新pDAG实现增量子空间聚类算法。Wang等[62]将增量层次聚类算法应用到文献综述技术中，在新的文档出现时实时更新文档摘要。Zhou等[63]对图聚类算法SA-Cluster进行改进，提出了增量算法Inc-Cluster，该算法在聚类开始时就计算出完整的随机游走距离矩阵，然后在每次迭代过程中，根据所给的属性权重增量，用Inc-Cluster算法更新原始随机游走距离矩阵，而不是从头开始重新计算。Ning等[64]提出了增量谱聚类算法，通过引入发生率矢量/矩阵，将两个动态行为（数据点的插入/删除、现有数据点的相似性改变）以相同的框架表示，并

且增量地更新特征系统。Wang等[65]提出了基于多中心的增量模糊聚类算法IMMFC，它采用多个中心点而不是一个中心点来表示数据块中的类，同时根据这些中心点自动生成一些成对约束，用来帮助最终的聚类过程。

总的来说，随着时间的增长，数据会不断的累积，聚类结构也会随之发生变化。当对象实时动态增量时，增量更新机制能有效提高聚类算法的效率和聚类结果的可扩展性，而目前针对聚类的增量更新算法尚处于初步研究阶段，目前的成果也都是面向单视图、单聚类结果的，尚少见面向多聚类相关算法的成果发表。因此，针对多聚类的动态增量更新机制研究仍需新的方法和技术手段。

1.3 目前存在的问题

综上所述，尽管有很多学者在多聚类以及安全聚类、高效聚类方面开展了广泛的研究，但是这些方法无法应对大数据环境下的多聚类分析和高效计算所面对的挑战，目前尚缺乏灵活有效的多聚类方法，并且在大数据多聚类过程中安全、分布式并行以及增量式更新计算方面还存在很多亟待解决的问题：

（1）如何构建基于张量的大数据多聚类方法？

在大数据环境下的多聚类分析中，通常存在以下两个问题：第一，大数据来源多样，比如设备、传感器、网络、通信、人类行为等，其中包含结构化、半结构化和非结构化数据，如图像、音频、视频和文本等。不同类型的数据具有不同的结构、不同的分布、不同的度量方式以及复杂的关系，极大影响了多聚类分析的效率和质量；第二，用户的需求随时间、空间和情境而变化，不同情境下看待数据的观点也不尽相同，用户希望的是可以根据需求任意选择多源异构数据的不同特征获取期望的聚类结果，且多个聚类结果之间允许具有部分相似性。因此，如何利用张量代数理论，结合现有多聚类方法优势，构建面向大数据的多聚类方法是本文需要解决的前提。

（2）如何实现云端安全的张量多聚类？

近年来，云计算安全受到了学术界和业界的广泛关注，出现了大量的隐私保护方法。加密方法不仅可以提供形式化的隐私保证，而且在准确性上优于其它方法。在使用加密方法时，一种有效的做法是将数据外包给云端之前对其进行加密，然后在云端

对加密的数据执行所有计算，直到加密的最终结果返回给客户端进行解密。在这个过程中，云端不会学习到任何敏感数据和中间结果，保障了用户隐私的安全。然而，在加密数据上实现隐私保护的张量多聚类会带来一些问题和挑战：第一，为了保护用户的隐私和所有中间结果，与多聚类相关的各种安全运算是必不可少的，包括加法、乘法、除法、比较、取幂等，但是现有的加密方法只提供了有限的安全运算；第二，为保证聚类结果的准确性，在密文上计算的对象张量距离与明文的精度要尽可能相同，需要高效的同态加密体系和浮点数的处理，但是现有的方法中两者往往不能同时兼顾；第三，在云端聚类时应尽可能降低客户端的计算成本，但现有加密方法往往需要客户端进行解密操作，或者协助在明文上执行一些无法实现的安全运算；第四，为了提高聚类算法的效率和可扩展性，需要合理利用云资源，以满足计算成本高、数据量不断增长的需求。因此，如何利用加密方法实现云端安全、高效、准确的张量多聚类，同时尽可能减轻客户端的负担，是实现云端安全多聚类需要解决的难题。

（3）如何实现基于张量链分解的张量多聚类？

在大数据环境下，基于张量的多聚类方法通常会遇到以下两个问题：第一，随着张量规模的增大，在一个对象张量上进行运算所需的时间、空间会呈指数级增长，从而带来维度灾难问题，必将大大降低多聚类的效率；第二，通过融合多源信息构成的原始对象张量含有大量的冗余和噪声数据，这将极大影响多聚类的准确性。而近年来提出的张量网络理论，由于其极好的压缩能力和分布式并行处理方式而被认为是分析大数据的非常有前途的工具。因此，假设所有的对象张量均以张量链分解的形式存储在云端或者数据中心上，那么如何在张量链分解的形式上，通过各种张量链操作或运算，实现完整的张量多聚类，同时实现高效的分布式并行计算，是实现张量多聚类高效计算的又一关键问题。

（4）如何实现张量多聚类的增量式更新？

在大数据时代，数据对象、数据特征空间实时动态变化，聚类结构需要动态更新。传统方法是定期更新数据集，重新执行聚类算法以生成新的聚类结构，而原有对象之间的相异性结构并未发生改变，因此将产生大量的重复计算。所以，为了解决大数据环境下多聚类的高效计算问题，研究张量多聚类的增量更新方法具有重要实际意义。

在现有聚类结果的基础上，对新增对象进行聚类，传统思想是如何将新增对象融入已有结果中或独立构成新的类簇等，所以这类方法更加关注的是从已有聚类结果入手，考虑新增对象及它们的关系。但是，这样做容易产生增量聚类结果不准确、效率低等问题，主要是考虑关系不全面以及反复的类簇合并、拆分导致。因此，基于已有聚类结果进行增量计算时，如何最大限度地避免重复计算，保证及时响应用户的查询需求并高效、准确的反馈最新聚类结果，是实现增量式张量多聚类所要解决的核心问题。

1.4 研究内容与目标

1.4.1 研究内容

针对大数据环境下面向多源异构数据的多聚类理论，本文从张量多聚类方法构建、云端安全的、分布式并行计算与动态增量更新的张量多聚类等方面展开理论、方法及技术研究。主要研究内容如图 1.1 所示：

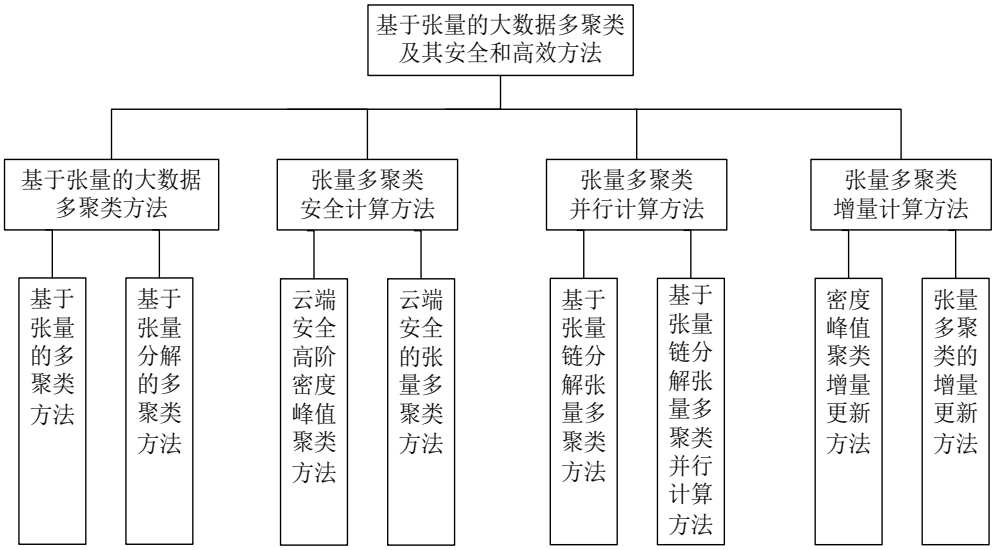


图 1.1 论文研究内容框架结构和相互关系图

研究内容一：基于张量的大数据多聚类方法

（1）基于张量的多聚类方法

从大数据应用需求出发，研究基于上下文动态选择特征空间组合产生多个不同情境下的聚类结果，构建大数据环境下基于张量的多聚类方法。基于对象张量表示模型，通过设计不同特征空间属性组合的权重学习算法，寻求提高多聚类质量的途径；基于对象张量相似度度量数学模型，通过分析寻找元素位置度量系数矩阵投影规律，设计

引入特征空间选择系数，从而构建基于可选择加权张量距离的张量多聚类方法。

（2）基于张量分解的多聚类方法

为了在计算距离时可以将所选择的特征与未选择的特征完全分离来提高多聚类质量，研究基于相似度矩阵加权平均的多聚类方法以及基于可选择加权欧式距离的多聚类方法。然而，由于这两种方法仍没有考虑不同属性的影响，没有考虑如何去除噪声和冗余，特别是对于高维数据。因此，结合这两种方法的优势进一步研究基于张量分解的多聚类方法；同时，为了提高该方法的性能，研究设计一种用于各特征空间属性重要性度量的多关系属性排名方法。

研究内容二：张量多聚类安全计算

（1）云端安全的高阶密度峰值聚类方法

为了实现云环境中计算张量多聚类的同时保护用户的隐私，首先研究密度峰值聚类算法的云端安全计算模式。设计混合云模式下云端安全的聚类分析和服务框架，进而研究云端安全的聚类分析方法；分析聚类算法中的基本运算，根据现有同态加密运算协议，研究设计同态加密体系下的高阶密度峰值聚类算法所需的安全运算协议，包括安全排序、安全平方张量距离等安全子协议，从而构建云端安全的高阶密度峰值聚类分析方法并给出半诚实模型下的安全性证明。

（2）云端安全的张量多聚类方法

在前面研究的基础上，设计混合云模式下云端安全的多聚类分析和服务框架，进而研究云端安全的多聚类分析方法。分析基于张量的多聚类算法基本运算，根据现有同态加密运算协议，结合同态加密体系，研究设计张量多聚类算法所需的安全运算协议，包括安全指数、安全权重学习、安全可选择加权张量距离等安全子协议，从而实现云端安全的张量多聚类分析方法并给出半诚实模型下的安全性证明。

研究内容三：张量多聚类并行计算

（1）基于张量链分解的张量多聚类方法

结合张量链分解和张量链计算理论，研究张量链分解形式下各张量基本运算的计算规律。在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量链，研究实现基于张量链分解的多线性属性组合权重学习算法和可选择

加权张量距离，进而构建基于张量链分解的张量多聚类方法。

（2）基于张量链分解的张量多聚类及并行计算方法

在云计算分布式环境下，根据张量链分解形式下各张量基本运算的计算规律，依据节点计算能力和通信能力设计高效的分布式并行计算框架，研究张量链的核分配机制、核调度策略及核运算并行策略，设计基于张量链核的分布式并行策略，充分利用张量网络并行计算优势提高张量多聚类算法的并行效率。

研究内容四：张量多聚类增量计算

（1）密度峰值聚类的增量更新方法

针对大数据动态增长给大数据分析带来的大量重复计算问题，首先研究密度峰值聚类的增量更新方法。在原始密度峰值聚类算法的基础上，分析研究与其相关的三个主要组成部分的更新算法：局部密度更新方法、基于红黑树的对象依赖重连方法、中心更新方法和类簇分割更新方法，从而对原有聚类结果和中间结果直接进行调整。

（2）张量多聚类的增量更新方法

针对基于张量的多聚类方法，研究基于迭代的属性权重学习方法和基于微分的属性权重学习方法，并基于一种简单快速的 K-medoids 算法研究设计相应的增量式 K-medoids 算法，使得在多聚类增量时不用计算全部距离，从而有效提高张量多聚类的增量更新算法效率。

1.4.2 研究目标

在大数据环境下，研究满足实际应用需求的多聚类理论及应用，解决针对大规模多源异构数据的张量多聚类方法构建及其安全和高效计算的关键技术难题。主要研究目标：

（1）设计实现在对象张量表示模型基础上不同特征空间属性组合的权重学习算法，提出基于可选择加权张量距离的张量多聚类方法，设计并提出用于各特征空间属性重要性度量的多关系属性排名方法以及基于张量分解的多聚类方法。

（2）设计安全的张量多聚类分析和服务框架，提出高阶密度峰值聚类和张量多聚类方法所需的基本安全运算协议，以及多聚类过程中的相关子算法安全协议，设计

实现完整的云端安全的密度峰值聚类方法和张量多聚类方法并给出相关安全性证明。

（3）提出基于张量链分解的张量多聚类方法，并设计一套高效的分布式并行计算框架，提出张量链核分配机制、核调度策略及基于张量链核的分布式并行策略。

（4）提出针对动态数据对象的密度峰值聚类的增量更新方法，设计一套完整的张量多聚类的增量更新方法，进而提高张量多聚类分析方法的计算效率及多聚类结果可扩展性。

1.5 研究创新

本文选定大数据环境下的多聚类及其安全高效计算作为主要研究对象，围绕基于张量的大数据多聚类及其安全和高效方法两大方面展开理论、技术及方法的系列研究，主要具有以下四个方面创新：

（1）基于张量的大数据多聚类方法

针对大数据的来源多样、特征高维、关系复杂、规模庞大和生成快速等问题，充分发挥现有多聚类方法的优势，首先提出了一种基于多线性属性排名的权重学习方法，用来度量所有特征空间中属性组合的重要性，进而提出可选择加权张量距离，基于此提出了基于张量的多聚类方法；其次，为了在计算距离时可以将所选择的特征与未选择的特征完全分离来提高多聚类质量，提出了基于相似度矩阵加权平均的多聚类方法和基于可选择加权欧式距离的多聚类方法。最后又针对这两种方法没有考虑如何去除数据中的噪声和冗余而影响聚类质量的问题，提出了基于张量分解的多聚类方法。同时为了提高基于张量分解的多聚类方法的性能，提出一种用于各特征空间属性重要性度量的多关系属性排名方法。实验表明提出的方法具有较高的聚类准确率和较低的冗余度。本文所提出基于张量的多聚类方法可为大数据环境下的多聚类理论及应用研究提供新的有益思路。

（2）云端安全的张量多聚类方法

在云计算环境中，为保护用户隐私，提出了张量多聚类算法的云端安全计算模式。设计混合云模型下云端安全的张量多聚类分析和服务框架，提出一种安全高阶密度峰

值聚类方法，进而提出了安全张量多聚类方法以及相关的多种安全子协议，包括安全指数、安全排序、安全属性排名、安全平方张量距离以及安全可选择加权张量距离协议。上述两种方法中所有的聚类计算任务都是在云端实现的，而云不会公开或推断出任何机密信息，这不仅提高了聚类的效率，而且保护了用户的隐私。并且客户端无需参与任何聚类计算，对用户来说是非常轻量级的，同时这两种安全聚类方法都可以在半诚实模型下实现基于Paillier加密体系的完整安全协议，保护的信息包括聚类的中间结果以及最终的聚类结果并能够保证结果具有较高的可用性。此外，该方法可以保证100%的聚类准确率，且算法具有较高的扩展性。本文所提出的云端安全的张量多聚类方法亦可为其它面向大数据的安全聚类方法研究提供借鉴参考。

（3）基于张量链分解的张量多聚类及其并行计算方法

针对维度灾难和高效计算问题，提出一套基于张量链分解的张量多聚类及其并行计算方法。首先，基于张量链分解形式下各张量基本运算的计算规律，在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量链，提出基于张量链分解的多线性属性组合权重学习算法和可选择加权张量距离，进而提出基于张量链分解的张量多聚类方法。从而实现在张量链分解的形式下完整的张量多聚类过程，并能保证甚至提高聚类结果的准确性；其次，在云计算分布式环境中，依据节点计算能力和通信能力设计高效的分布式并行计算框架，通过研究张量链核分配机制、核调度策略及核运算的并行策略，提出了基于张量链核的分布式并行策略，以充分利用张量网络并行计算优势来提高张量多聚类算法的并行效率。在此所提出的并行策略可为面向大数据的张量多聚类方法的高效计算提供新的有效技术途径。

（4）张量多聚类的增量式更新方法

针对大数据动态增长带来的大量重复计算问题，本文提出了增量式密度峰值聚类和增量式张量多聚类。首先，在原始密度峰值聚类算法的基础上，提出了与其相关的三个主要组成部分的更新算法：局部密度更新方法、基于红黑树的对象依赖重连方法、聚类中心更新方法和类簇分割更新方法，从而对原有聚类结果和中间结果直接进行调整；其次，针对张量多聚类方法，分别提出了基于迭代的属性权重学习方法和基于微

分的属性权重学习方法，并基于一种简单快速的K-medoids算法设计相应的增量式K-medoids算法，使得在多聚类增量时不需要计算全部距离，从而有效提高张量多聚类的增量更新算法效率。实验结果表明，提出的增量式密度峰值聚类与同类方法相比具有较高的聚类准确率和效率，而提出的增量式张量多聚类方法不仅能保证增量更新的聚类准确率，而且能够极大程度地高多聚类分析中数据动态增量更新维护的效率。

1.6 论文组织结构

本文围绕基于张量的大数据多聚类及其安全和高效实现展开理论与方法的研究，组织结构如下：

第 1 章介绍本文的研究背景和研究意义，介绍国内外相关研究工作，分析目前在多聚类以及安全聚类、高效聚类方面存在的问题，引出论文的研究内容和研究目标，并对本文的创新点进行总结。

第2章首先介绍张量相关理论，包括张量代数、张量距离、张量分解和张量链分解；然后，介绍两种基于马尔科夫理论的数据排名方法；接着，介绍目前流行的密度峰值聚类方法，最后，介绍同态加密中较为典型的Paillier加密体系。

第3章基于对象张量表示模型，设计不同特征空间属性组合的权重学习算法，构建基于可选择加权张量距离的张量多聚类方法。此外，为了在计算距离时可以将所选择的特征与未选择的特征完全分离来提高多聚类质量，研究基于相似度矩阵加权平均的张量多聚类方法和基于可选择加权欧式距离的多聚类方法。然后又针对这两种方法没有考虑如何去除数据中的噪声和冗余而影响聚类质量的问题，通过扩展两者的优势进一步研究基于张量分解的多聚类方法。同时，为了提高基于张量分解的多聚类方法的性能，研究设计一种可用于各特征空间属性重要性度量的多关系属性排名方法。

第4章研究张量多聚类方法的云端安全计算模式。设计混合云模型下云端安全的多聚类分析和服务框架，基于此研究一种安全高阶密度峰值聚类方法，进而研究云端安全的张量多聚类方法以及相关的各种安全子协议，包括安全指数、安全排序、安全属性排名、安全平方张量距离以及安全可选择加权张量距离协议，并对提出的方法给出安全性证明。

第5章研究张量链分解形式下各张量基本运算的计算规律，在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量链，研究基于张量链分解的多线性属性组合权重学习算法和可选择加权张量距离，构建基于张量链分解的张量多聚类方法。在云计算分布式环境中，依据节点计算能力和通信能力设计高效的分布式并行计算框架，研究张量链核分配机制、核调度策略及核运算并行策略，以及基于张量链核的分布式并行策略。

第6章首先研究增量式密度峰值聚类方法，包括与其相关的三个主要组成部分的更新算法：局部密度更新方法、基于红黑树的对象依赖重连方法、聚类中心的更新方法和类簇分割更新方法，从而对原有聚类结果和中间结果直接进行调整。其次，针对增量式张量多聚类方法，研究基于迭代的属性权重学习方法和基于微分的属性权重学习方法，并基于一种简单快速的K-medoids算法，设计相应的增量式K-medoids算法，从而有效提高张量多聚类的增量更新算法效率。

第7章总结本文的主要工作和研究成果，并进一步探讨未来的研究方向。

2 多聚类相关理论基础

本章是全文的理论基础。首先介绍张量相关理论，包括张量代数、张量距离、张量分解和张量链分解；然后介绍两种基于马尔科夫理论的数据排名方法；接着介绍流行的密度峰值聚类方法；最后介绍同态加密中较为典型的 Paillier 加密体系。

2.1 张量相关理论

本节给出张量代数的相关背景知识，介绍张量模型及张量相关运算，张量距离、张量分解以及张量链分解及相关运算。

2.1.1 张量代数

张量是高阶矩阵的推广并成功地应用于数据挖掘、图分析、信号处理和计算机视觉[66]等多个领域。实际上，张量可以看成是一个多维数组。张量的阶数即模的数量，向量是一阶张量，矩阵就是二阶张量，三阶或更高阶的张量称为高阶张量。

假设 $\mathcal{T} \in R^{I_1 \times I_2 \times \ldots \times I_N}$ 表示一个 N 阶张量，那么沿张量第 n 模展开得到的矩阵为 $T_{(n)} \in \Re^{I_n \times I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1}}$，张量元素 $t_{i_1 i_2 \cdots i_N}$ 对应矩阵元素 $e_{i_n j}$，其中 j 为：

$$\begin{aligned} j = &(i_{n+1}-1) I_{n+2} \cdots I_N I_1 \cdots I_{n-1} + (i_{n+2}-1) I_{n+3} \cdots I_N I_1 \cdots I_{n-1} \\ &+ \cdots + (i_2 - 1) I_3 I_4 \cdots I_{n-1} + \cdots + i_{n-1} \quad . \end{aligned} \tag{2.1}$$

张量与矩阵的单模乘运算：张量 $\mathcal{T}$ 与矩阵 $U \in \Re^{J \times I_n}$ 的 n-*mode* 积定义如下：

$$\left(T \times_n U\right)_{i_1 i_2 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} \left(t_{i_1 i_2 \cdots i_N} u_{j i_n}\right). \tag{2.2}$$

秩-1 张量：一个 N 阶张量 $\mathcal{T} \in R^{I_1 \times I_2 \times \ldots \times I_N}$ 是一个秩-1 张量，那么它可以表示为 N 个向量的外积，即

$$\mathcal{T} = \mathrm{t}^{(1)} \circ \mathrm{t}^{(2)} \circ \cdots \circ \mathrm{t}^{(N)}, \tag{2.3}$$

其中，$\mathrm{t}^{(i)} (1 < i < N)$ 是向量，也就是张量的每一个元素对应向量的元素如下：

$$\mathcal{T}_{i_1 i_2 \cdots i_N} = t_{i_1}^{(1)} t_{i_2}^{(2)} \ldots t_{i_N}^{(N)} (1 \le i_n \le I_N). \tag{2.4}$$

2.1.2 张量距离

张量距离[67]是一种衡量高阶数据对象之间相似度的度量方式。与具有正交性假设约束的传统欧式距离（Euclidean Metric）不同，张量距离可以通过考虑不同坐标之间的复杂关系来有效地度量高阶张量空间中数据对象之间的距离。设有张量 $\mathcal{X} \in \Re^{I_1 \times I_2 \times \cdots \times I_N} (N > 1)$，x是它的向量化形式。对于元素 $\mathcal{X}_{i_1 i_2 \ldots i_N} (1 \le i_j \le I_j, 1 \le j \le N)$，$x_l$ 是其向量化形式x的第 l 个元素，其中 $l = i_1 + \sum_{j=2}^{N} (i_j - 1) \prod_{k=1}^{j-1} I_k$ 。设有相同大小张量 $\mathcal{Y}$，那么 $\mathcal{X}$ 与 $\mathcal{Y}$ 之间的张量距离可由下式计算：

$$\begin{aligned} d_{TD} &= \sqrt{\sum_{l,m=1}^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_N}} g_{lm} (x_l - y_l)(x_m - y_m)} \\ &= \sqrt{(\mathrm{x\text{-}y})^T G (\mathrm{x\text{-}y})}, \end{aligned} \tag{2.5}$$

其中，g_{lm} 是元素位置度量系数，G 是与元素位置距离相关的度量矩阵，可以反映高阶数据对象之间不同坐标的内在关系。G 中的元素 g_{lm} 定义如下：

$$g_{lm} = \frac{1}{2\pi\sigma^2} \exp\left\{ \frac{-\|p_l - p_m\|_2^2}{2\sigma^2} \right\}, \tag{2.6}$$

其中，σ 是一个正则化参数，$\|p_l - p_m\|_2$ 是元素 $\mathcal{X}_{i_1 i_2 \ldots i_N}$（对应 x_l）和元素 $\mathcal{X}_{i_1' i_2' \ldots i_N'}$（对应 x_m）位置之间的距离，定义如下：

$$\|p_l - p_m\|_2 = \sqrt{(i_1 - i_1')^2 + (i_2 - i_2')^2 + \cdots (i_N - i_N')^2}. \tag{2.7}$$

特别地，当 $G = I$ 时，张量距离退化成欧氏距离。

2.1.3 张量分解

张量分解从本质上来说是矩阵分解的高阶泛化，现在已经出现多种应用相对广泛的相关算法。最经典的算法是CP分解[68]和Tucker分解[69]，它们都可以看作是奇异值分解（SVD算法）在某个角度上的推广。

CP 分解可以看作将一个 N 阶张量表示为若干秩-1 张量的和：

$$\mathcal{T}=\sum_{i=1}^{r}u_1^{(i)}\circ u_2^{(i)}\circ\cdots\circ u_N^{(i)}. \tag{2.8}$$

CP 分解可以把原来的指数关系降到线性关系，但当面对高阶张量时，其最优秩是个 NP 难问题，同时其分解算法也不稳定。

Tucker 分解也是一种常用的张量分解形式，它可以看作是主成分分析（PCA）的高阶推广。对于一个原始张量 $\mathcal{T}$ 可以被分解如下形式：

$$\mathcal{S}=\mathcal{T}\times_1 U_1^{T}\times_2 U_2^{T}\cdots\times_N U_N^{T}, \tag{2.9}$$

其中，$\mathcal{S}$ 是核心张量，它可以看作原始张量 $\mathcal{T}$ 的压缩版，表示不同模上不同分量之间的相互作用关系。$U_1,U_2,\cdots,U_N$ 是因子矩阵，它们保留了各个模上的主成分。采用交替最小二乘（ALS）思想，高阶正交迭代（HOOI）[70]方法可用来计算优化因子矩阵 $U_1,U_2,\cdots,U_N$。

此外，当需要分解得到的各主成分进行还原时，可以利用如下公式进行近似还原：

$$\widehat{\mathcal{T}}=\mathcal{S}\times_1 U_1\times_2 U_2\cdots\times_N U_N, \tag{2.10}$$

其中 $\widehat{\mathcal{T}}$ 是核心张量和因子矩阵做单模乘还原得到的近似张量。

2.1.4 张量链分解

近年来，张量网络作为一种用于分析处理高维大数据的新兴技术，已成为目前国际上的一个研究热点。不同于张量分解，它把原始张量分解成多个低阶低维的核心张量。张量网络的主要分解形式包括：层次 Tucker（Hierarchical Tucker，HT）分解[71]，其分解算法需要递归，所以实现开销较大；张量链（Tensor Train，TT）分解[72]作为 HT 的一种最简单方式，它把原始张量分解成一个链的方式，分解后的张量都是低阶低维的，一般都是三阶的，其优点是参数较少，接近于 CP 分解且算法稳定不需要递归；但若针对低阶高维张量，例如大规模的向量或矩阵，由于其阶数很低，TT 的优势就明显降低，所以出现了量化张量链（Quantitaty Tensor Train，QTT）分解[73]。QTT 首先是把低阶的向量或者矩阵量化为一个张量，然后再使用 TT 分解；由此，可以把许多针对向量、矩阵的大规模问题变成小规模问题来进行求解。因此张量网络由于其

极好的压缩能力和分布式并行处理方式而被认为是分析大数据的非常有前途的工具。

（1）张量链分解格式

给定张量 B，通过张量 $A \approx B$，元素近似为：

$$A(i_1,i_2,\ldots,i_d) = G_1(i_1)G_2(i_2)\ldots G_d(i_d), \tag{2.11}$$

这里 $G_k(i_k)$ 是一个 $r_{k-1} \times r_k$ 的矩阵。这些参数相关矩阵的乘积是大小为 $r_0 \times r_d$ 的矩阵，因此必须施加“边界条件” $r_0 = r_d = 1$。矩阵 $G_k(i_k)$ 实际上是一个三维数组，它可以看作一个具有元素 $G_k(\alpha_{k-1}, n_k, \alpha_k) = G_k(i_k)\alpha_{k-1}\alpha_k$ 的 $r_{k-1} \times n_k \times r_k$ 数组。

在索引形式中，上式可写为：

$$A(i_1,\ldots,i_d) = \sum_{\alpha_0,\ldots,\alpha_{d-1},\alpha_d} G_1(\alpha_0,i_1,\alpha_1)G_2(\alpha_1,i_2,\alpha_2)\ldots G_d(\alpha_{d-1},i_d,\alpha_d), \tag{2.12}$$

由于 $r_0 = r_d = 1$，这种分解也可以通过线性张量网络来表示，如图 2.1 所示（以 d=5 的情形为例）：

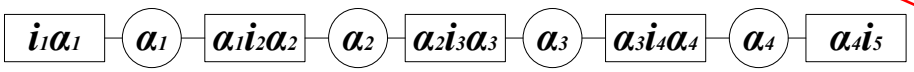


图 2.1　张量链分解网络

在图 2.1 中有两种类型的节点：矩形包含空间索引（即原始张量的索引 i_k）和一些辅助索引 α_k，并且具有这些索引的张量与这种类型的节点相关联。圆形仅包含辅助索引 α_k，用于表示一个链：如果辅助索引存在于两个核中，则连接它。辅助索引的和是给定的，即为了估计一个张量的一个元素，必须将矩形中的所有张量相乘，然后对所有辅助索引执行求和。由于整个图看起来像一列火车的车厢以及车厢之间的链接，所以用张量链分解或简称 TT 分解来解释其名字。秩 r_k 称为压缩秩或 TT 秩，而 TT 分解的三阶张量 G_k-核称为 TT 核（类似于 Tucker 分解的核心）。

（2）Tensor Train 分解算法

算法 2.1： TT 分解算法

输入： d 维张量 **A**，规定精度 ε。

输出： 得到具有 TT 形式的 A 的 TT 近似 B 的核 $G_1, G_2, \ldots, G_d$，其中 A 的 TT 秩 $\hat{r}_k$ 等于 A 的展开矩阵 A_k 的 δ 秩。这里 $\delta = \frac{\varepsilon}{\sqrt{d-1}}\|A\|_F$，计算出的近似满足 $\|\mathbf{A}-\mathbf{B}\|_F \le \varepsilon\|\mathbf{A}\|_F$

1：计算截断参数 $\delta \leftarrow \frac{\varepsilon}{\sqrt{d-1}}\|A\|_F$

2：临时张量：$\mathbf{C}=\mathbf{A}$，　$r_0=1$。

3：for k=1 to d-1 do

4：　　$C \leftarrow reshape\left(C,\left[r_{k-1}n_k,\frac{numel(C)}{r_{k-1}n_k}\right]\right)$

5：　　计算 δ 截断的 SVD，$C \leftarrow USV+E,\|E\|_F \le \delta, r_k = rank_\delta(C)$

6：　　新的核，$G_k \leftarrow reshape\left(U,\left[r_{k-1},n_k,r_k\right]\right)$

7：　　$C \leftarrow SV^T$

8：　end for

9：　$G_d \leftarrow C$.

10：返回具有核 $G_1, G_2, \ldots, G_d$ 的 TT 形式的张量 **B**。

（3）Tensor Train 基本运算

加法运算：设有 TT 形式下的两个张量 $\mathbf{A}=A_1(i_1)\ldots A_d(i_d)$，$\mathbf{B}=B_1(i_1)\ldots B_d(i_d)$，和 $\mathbf{C}=\mathbf{A}+\mathbf{B}$ 的核定义为

$$C_k(i_k)=\begin{pmatrix} A_k(i_k) & 0 \\ 0 & B_k(i_k) \end{pmatrix}, \quad k=2,\ldots,d-1, \tag{2.13}$$

对于边界核有

$$C_1(i_1)=\begin{pmatrix} A_1(i_1) & B_1(i_1) \end{pmatrix}, \quad C_d(i_d)=\begin{pmatrix} A_d(i_d) \\ B_d(i_d) \end{pmatrix}, \tag{2.14}$$

实际上，通过直接相乘有

$$C_1(i_1)C_2(i_2)\ldots C_d(i_d)=A_1(i_1)A_2(i_2)\ldots A_d(i_d)+B_1(i_1)B_2(i_2)\ldots B_d(i_d). \tag{2.15}$$

如果将以 TT 格式给出的向量 t 与自身相加，则秩加倍，但是结果应该被压缩为具有与 t 相同的秩的 2t，实际上两个向量的加法不需要操作，但它增加了 TT 秩。

数乘运算：TT 形式下的数乘运算非常简单。假设与数字 α 进行数乘运算，则只需要在一个核上乘以 α。

多维压缩：以如下形式表示要计算的多维压缩：

$$W = \sum_{i_1,\ldots,i_d} A(i_1,\ldots i_d)u_1(i_1)\ldots u_d(i_d), \tag{2.16}$$

这里 $u_k(i_k)$是长度为 n_k 的向量。这是 A 与规范秩-1 张量的标量积：

$$W = \left\langle \mathbf{A}, \otimes_{i=1}^{d} u_j \right\rangle \tag{2.17}$$

在这种情况下，张量 A 由张量网格上的函数值组成且 u_k 是（一维）正交权重。令 A 处于 TT 格式，则

$$\mathbf{A} = G_1(i_1)\ldots G_d(i_d) \tag{2.18}$$

然后有

$$W = \left(\sum_{i_1} u_1(i_1)G_1(i_1)\right)\left(\sum_{i_2} u_2(i_2)G_2(i_2)\right)\ldots\left(\sum_{i_d} u_d(i_d)G_d(i_d)\right) \tag{2.19}$$

Hadamard 乘积：计算两个张量 **A** 和 **B** 的元素（Hadamard）乘积 **C**：

$$\mathrm{C=A*B} \tag{2.20}$$

其中 **C** 的元素被定义为：

$$C(i_1,\ldots,i_d) = A(i_1,\ldots,i_d)B(i_1,\ldots,i_d) \tag{2.21}$$

结果仍然是 TT 格式的且为 **A** 和 **B** 的 TT-秩相乘的积：

$$\begin{aligned} C(i_1,\ldots,i_d) &= A_1(i_1)\ldots A_d(i_d)B_1(i_1)\ldots B_d(i_d) \\ &= \left(A_1(i_1)\ldots A_d(i_d)\right)\otimes\left(B_1(i_1)\ldots B_d(i_d)\right) \\ &= \left(A_1(i_1)\otimes B_1(i_1)\right)\left(A_2(i_2)\otimes B_2(i_2)\right)\ldots\left(A_d(i_d)\otimes B_d(i_d)\right) \end{aligned} \tag{2.22}$$

这意味着 **C** 的核为：

$$C_k(i_k) = A_k(i_k)\otimes B_k(i_k), \quad k = 1,\ldots,d \tag{2.23}$$

标量积：使用 Hadamard 乘积可以计算两个张量的标量积。对两个张量 **A**、**B**，标量积可定义为：

$$\langle \mathbf{A},\mathbf{B}\rangle = \sum_{i_1,\ldots,i_d} A(i_1,\ldots,i_d)B(i_1,\ldots,i_d) = \sum_{i_1,\ldots,i_d} C(i_1,\ldots,i_d) \tag{2.24}$$

这里C=A∗B。

矩阵和向量的乘积：如果矩阵 M 的元素被定义如下形式

$$M\left(i_1,\ldots,i_d,j_1,\ldots,j_d\right)=M_1\left(i_1,j_1\right)\ldots M_d\left(i_d,j_d\right), \tag{2.25}$$

则将其称为 TT 格式，这里的 $M_k(i_k,j_k)$是一个 $r_k-1\times r_k$ 矩阵。如果所有的 TT 秩等于 1，则 M 被表示为矩阵的一个 Kronecker 乘积：

$$M=M_1\otimes M_2\otimes\cdots M_d \tag{2.26}$$

假设具有矩阵 M 和具有 TT-核心 X_k 和元素 $X(j_1,...,j_d)$.的 TT 格式的向量 x，在此情况下的矩阵乘向量乘积是如下和的计算：

$$Y\left(i_1,\ldots,i_d\right)=\sum_{j_1,\ldots,j_d}M\left(i_1,\ldots,i_d,j_1,\ldots,j_d\right)X\left(j_1,\ldots,j_d\right) \tag{2.27}$$

其结果张量仍然是 TT 格式的。实际上，

$$\begin{aligned}Y\left(i_1,\ldots,i_d\right)&=\sum_{j_1,\ldots,j_d}M_1\left(i_1,j_1\right)\ldots M_d\left(i_d,j_d\right)X_1\left(j_1\right)X_d\left(j_d\right)\\&=\sum_{j_1,\ldots,j_d}\left(M_1\left(i_1,j_1\right)\otimes X_1\left(j_1\right)\right)\ldots\left(M_d\left(i_d,j_d\right)\otimes X_d\left(j_d\right)\right)\\&=Y_1\left(i_1\right)\ldots Y_d\left(i_d\right)\end{aligned} \tag{2.28}$$

其中，

$$Y_k\left(i_k\right)=\sum_{j_k}\left(M_k\left(i_k,j_k\right)\otimes X_k\left(j_k\right)\right). \tag{2.29}$$

2.2 基于马尔科夫理论的数据排名算法

2.2.1 PageRank 算法

PageRank 是 Google 公司的核心技术之一，它通过考虑链接的数量和质量来识别网站的重要性，现在众多搜索引擎将其广泛应用于对网站进行排名[74]。PageRank 的主要思想是基于这样的假设：如果网站从其他网站接收到更多高质量的链接，那么网站将变得更加重要。PageRank 构造一个新的随机游走模型来模拟一个“随机冲浪者”，根据马尔可夫链采用概率 α 的步长冲浪并以概率 1-α 随机跳跃，根据马尔可夫链，它总能收敛达到一个唯一的平稳分布。假设 M 是表示随机游走的列随机矩阵，那么 PageRank 向量 x 是唯一的并且解决了以下线性系统：

$$\mathrm{x}=\alpha M x+(1-\alpha)\mathrm{v}\,, \tag{2.30}$$

其中 v 是随机向量，素性修正概率参数 α 满足 $0<\alpha<1$。

幂方法[75]主要用于计算最大特征值和相应特征向量的矩阵，在这里可以用来迭代求解 PageRank 的解向量 x。令 v_1 为对应于特征值 $\delta_1=\delta_{\max}(A)$ 的特征向量，其中 A 表示矩阵，v_1 与初始向量 z 之间的夹角 $\angle(\mathrm{z},\mathrm{v}_1)$ 定义如下：

$$cos\angle(\mathrm{z},\mathrm{v}_1)=\frac{\mathrm{z}\mathrm{v}_1}{\|\mathrm{z}\|_2\|\mathrm{v}_1\|_2}. \tag{2.31}$$

如果向量 z 和 v_1 彼此不垂直，则幂方法可以生成与 v_1 变得越来越平行的向量序列。此外为了计算张量的最佳秩-1 近似，文献[70]提出了一种高阶幂法。

2.2.2 HAR 算法

HAR（Hub，Authority 和 Relevance）算法是一种通过计算对象的中心、权威和相关性得分来识别多关系数据中对象和关系的重要性的方法[76]，其基本思想是考虑相互加强的中心、权威和相关性之间的关系。例如，如果一个对象通过高相关性分数的关系指向具有高权威分数的多个对象，则该对象将获得较高的中心得分。HAR 在多关系张量中构造新的随机游走模型，并计算到达对象作为中心或权威以及分别使用相关性的限制概率。根据具有有限状态空间的不可约马尔可夫链，该模型总是具有唯一的平稳分布[77]，如果 H，A 和 R 分别是表示随机行走的列随机转移概率张量，那么 HAR 的得分向量 x，y 和 z 是唯一的并可通过以下多元多项式方程求解：

$$\begin{aligned}\mathrm{x}&=(1-\alpha)\mathcal{H}\mathrm{yz}+\alpha\mathrm{o},\\ \mathrm{y}&=(1-\beta)\mathcal{A}\mathrm{xz}+\beta\mathrm{o},\\ \mathrm{z}&=(1-\gamma)\mathcal{R}\mathrm{xy}+\gamma\mathrm{r},\end{aligned} \tag{2.32}$$

其中向量 o,r 随机分配的概率分布，素性修正概率参数 α，β，γ 满足 $0<\alpha,\beta,\gamma<1$。

2.3 密度峰值聚类

密度峰值聚类算法（Clustering by Fast Search and Find of Density Peaks，CFS）[78]是由Alex Rodriguez和Alessandro Laio于2014年提出并发表在Science上的一种基于密度的聚类方法，其主要思想是通过快速搜索密度峰值来发现类簇中心，从而快速找出被低密度区域分离的高密度区域。CFS算法基于这样两个假设：（1）类簇中心点的密

度大于周围邻居点的密度；（2）类簇中心点与更高密度点之间的距离相对较大。因此若 x_i 为待聚类数据点，对于数据集 $S=\{x_i\}_{i=1}^{N}$ 而言，CFS算法中主要需要计算两个值：一个是局部密度 ρ_i，另一个是相对于最近的S高密度点的距离 δ_i。

局部密度 ρ_i 的计算公式为：

$$\rho_i=\sum_j \chi(d_{ij}-d_c), \tag{2.33}$$

其中函数

$$\chi(x)=\begin{cases}1, x<0;\\ 0, x\geq 0.\end{cases} \tag{2.34}$$

公式(2.33)中，d_{ij} 为数据点 x_i 和 x_j 之间的距离，d_c 为截断距离，局部密度 ρ_i 表示数据集中与 x_i 之间的距离小于 d_c 的数据点的个数。因此，截断距离对计算每个点的局部密度是非常重要的因素，故CFS算法的有效性主要取决于对截断距离 d_c 的选择。根据经验，通过选择一个 d_c，可使每个点的平均邻居数量大约是数据点总数的1-2%。

距离 δ_i 的计算公式定义如下

$$\delta_i=\min_{j:\rho_j>\rho_i}\{d_{ij}\}, \tag{2.35}$$

即 δ_i 表示在局部密度大于 ρ_i 的点中与 x_i 距离最小的那个（些）点与 x_{q_i} 之间的距离；而对于具有最高局部密度的点 $\delta_i=\max_j\{d_{ij}\}$，只要同时保证 ρ 和 δ 取得最大就可以将 x_i 选为聚类中心。

这样，聚类中心是那些具有高局部密度 ρ 和较大距离 δ 的点，而具有低局部密度 ρ 和较大距离 δ 的点可被视作噪声和异常点，根据局部密度和距离构成的坐标 (ρ,δ) 可以得到决策图，但该方法需要人工干预选取聚类中心。因此，原文作者给出另一种可以定量选取中心点个数的方法，即计算一个综合考虑 ρ 和 δ 的值 γ_i，定义如下：

$$\gamma_i=\rho_i\times\delta_i. \tag{2.36}$$

显然，γ_i 值越大，x_i 越有可能成为中心点。因此，只需对 $\{\gamma_i\}_{i=1}^{N}$ 升序排列，从后往前截取若干个数据点作为聚类中心即可。

根据上述定义和分析，密度峰值聚类算法的伪代码如下：

算法2.2：密度峰值聚类算法

输入：对象 $x_1, x_2, \cdots, x_n$。

输出：聚类结果 *cl*。

1：for *i*=1 to *n* do

2：　　for *j*=*i*+1 to *n* do

3：　　计算所有不同对象之间的距离 d_{ij}；

4：　　end for

5：end for

6：计算用于确定截断距离的参数*k*，$k \leftarrow \lceil factor*(n(n-1)/2) \rceil$；

7:根据距离数组中第 *k* 个小值确定截断距离，$d_c \leftarrow min_{kth}\left(\{d_{11}, d_{12}, \ldots, d_{1n(n-1)/2}\}, k\right)$；

8：for *i*=1 to *n* do

9：　　计算第 *i* 个对象的局部密度，$\rho_i \leftarrow \sum_j \chi(d_{ij} - d_c)$；

10：end for

11：找到具有最高局部密度的对象 x_h，计算其距离并记录该最大距离 d_{ij} 的下标*j*，

$x_h = (\delta_h,\ cl_h) \leftarrow \max_j \{d_{ij}\}$；

12：for *i*=1 to *n* do

13：　　计算第*i*个对象的距离，并记录最小距离 d_{ij} 的下标*j*，$(\delta_i,\ cl_i) \leftarrow \min_{j:\rho_j > \rho_i} \{d_{ij}\}$；

14：　　计算 $\gamma_i \leftarrow \rho_i \times \delta_i$；

15：end for

16：对所有的 γ 值按升降序排序 $\{\gamma_1', \gamma_2', \cdots, \gamma_n'\}$；

17：初始化记录聚类中心标记的数组，$(e_1, e_2, \ldots, e_n) \leftarrow 0$；

18：for *i*=*n*-1 downto 1 do

19：　　根据斜率确定聚类中心，

if $(\gamma_{i+1}' - \gamma_i') / (\gamma_i' - \gamma_{i-1}') \geq slope$ then $e_i \leftarrow 1$；

20： end if

21：end for

22：for i=n downto 1 do

23： for j=1 to n do

24： 判断第 j 个对象是否为中心，如果是则标记为 1，否则为 0，

if $(\gamma_i^{'}==\gamma_j)$ and $(e_i==1)$ then

$cl_j \leftarrow n+1$；

25： end if

26： end for

27：end for

算法第 18 步从 n-1 开始是因为考虑到聚类结果至少包含一个类簇，所以 γ 值最大的对象一定是聚类中心。第 22 到 27 步是对所有对象进行归类，主要是判断若该对象如果是中心，则将其记录指向另一个点的下标（即对象依赖关系）直接改为最大值 n+1，而其他非中心对象则根据其记录的下标指向的对象进行归类。

2.4 同态加密

Paillier密码系统[79]是一种语义上安全的加性同态和概率非对称加密方案，其安全性基于复合数残差类问题的确定。令p和q为两个随机大素数，公钥pk由(N，g)给出，其中 $N = pq$ 和 $g \in Z_{N^2}^*$ 以及密钥 $sk = lcm((p-1),(q-1))$，则对于给定明文 $a,b,c \in Z_N$，Paillier密码系统具有以下属性：

（1）同态加法：$[\![a+b]\!]=[\![a]\!]*[\![b]\!] \mod N^2$；

（2）同态乘法：$[\![a*b]\!]=[\![a]\!]^b \mod N^2$。

这里，由于 N 是唯一的，为了简单起见，本文其余部分将省略模 N^2，且根据模周期属性：$N\text{-}x=-x \mod N$，$-x$ 可以表示为 $N\text{-}x$。因此结合同态乘法，$-c$ 的密文是 $[\![\text{-}c]\!]=[\![(-1)*c]\!]=[\![c]\!]^{-1}=[\![c]\!]^{N-1}$。

2.5 本章小结

本章主要介绍了后续章节用到的关键理论与基础，包括张量相关概念、张量距离以及张量分解中具有代表性的 Tucker 分解和张量链分解；并介绍了两种基于马尔科夫转移理论的数据排名算法、流行的密度峰值聚类算法以及可以用于隐私保护的 Pariilar 同态加密体系，旨在为后面章节的研究提供理论基础。

3 基于张量的大数据多聚类方法

多聚类有利于从不同角度发现大数据中隐藏的不同数据模式，如在社团发现、资源推荐、基因表达等领域都具有重要应用价值。而现有研究主要针对低维、小规模、单领域数据集，且聚类结果难以解释，无法根据上下文情境灵活的聚类对象并为不同的应用提供按需服务，主要原因在于现有算法大多面向具体应用，难以扩展到其他领域，缺乏通用性。为了应对大数据的来源多样、特征高维、关系复杂等问题，充分发挥现有多聚类方法的优势，创新研究一种面向大规模多源异构数据的多聚类方法是大数据时代多聚类研究的核心问题。本章利用张量代数理论，在构建对象张量的基础上，重点研究一种基于张量的多聚类方法；进而利用 Tucker 分解理论，研究一种基于张量分解的多聚类方法，挖掘数据在不同情境下隐含的不同类簇，实现大数据环境下的多模态聚类。

3.1 问题定义

在大数据环境下进行多聚类分析，通常会遇到以下两个问题：第一，大数据来源多样，比如设备、传感器、网络、通信、人类行为等，其中包含结构化、半结构化和非结构化数据，如图像、音频、视频和文本等。不同类型的数据具有不同的结构、不同的分布、不同的度量方式以及复杂的关系，极大影响了多聚类分析的效率和质量；第二，用户的需求随时间、空间和情境而变化，不同情境下看待数据的观点也不尽相同，用户希望的是可以根据需求任意选择多源异构数据的不同特征并获取期望的聚类结果，且多个聚类结果之间允许具有部分相似性；而现有方法大多关注产生的多个聚类结果之间是一种完全的替代，聚类结果的可解释性较差。根据第 2 章对张量代数理论的介绍可知，张量作为一种分析和处理的工具，已广泛应用于数据挖掘、计算机视觉、信号处理、科学计算和图像识别等领域，因此利用张量可以有效融合多源信息，并通过张量分解对大数据进行降维和去冗余、噪声，提取高质量数据集，有利于大数据环境下数据分析和挖掘。

综上，如何利用张量代数理论，结合现有多聚类方法的优势，构建面向大数据的多聚类方法是本章需要解决的问题。具体来说，第一，如何将数据对象表示为高效、简洁、统一的模型，对具有不同特征的数据对象进行统一表达为有效进行大数据多聚类分析的基础；第二，如何度量高阶空间异构数据对象之间的相似性，保证多个聚类结果之间的高质量和低冗余性；第三，如何评估不同特征空间属性的重要性，确保噪声属性不会降低最终结果的聚类质量；第四，如何根据大数据应用的不同需求灵活地聚类数据对象，满足大数据多分析任务的需求。

3.2 基于张量的多聚类方法

为了解决上述问题，首先需要构建大数据环境下的多聚类分析、处理和服务平台。因此，本节提出一种基于张量的多聚类分析和服务框架，其总体框架图如图 3.1 所示。

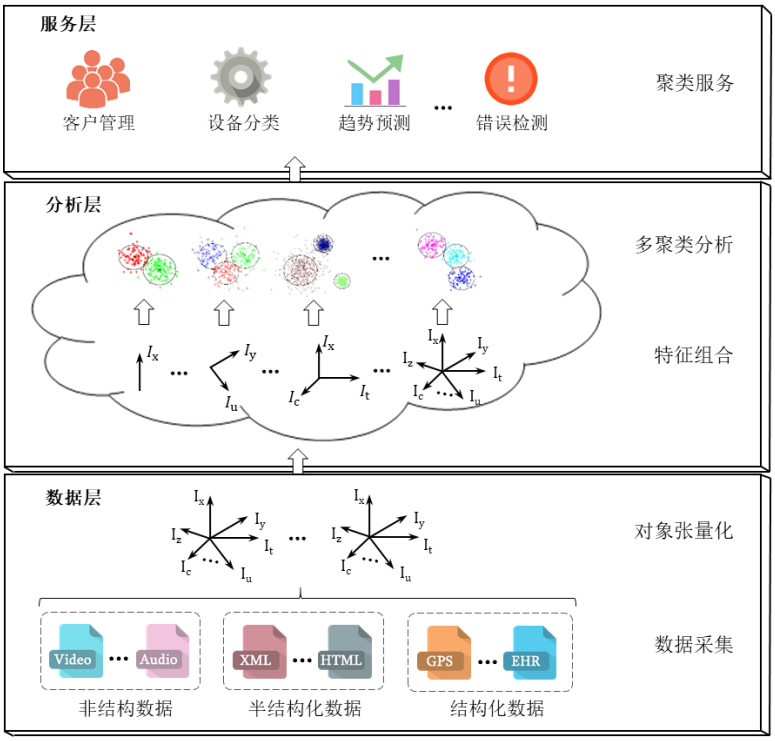


图 3.1 基于张量的多聚类分析和服务框架

该框架自底向上包括数据层、分析层和服务层，下面简单说明每层的功能：

（1）数据层：该层首先从物理空间、网络空间和社会空间采集大规模多源异构数据，这些数据通常包含具有不同类型的特征的相同的对象集；然后对多源异构数据

的融合，主要通过张量化过程将数据对象统一表示为对象张量模型。

（2）分析层：该层主要根据上下文情境变化及实际应用需求对大数据进行多模态聚类分析。利用所提出的基于张量的多聚类方法提取增强的知识，并根据所选择的不同特征组合，进一步生成多个不同的聚类结果，从而挖掘不同情境下数据隐含的不同类簇。

（3）服务层：根据大数据不同应用的需求，服务层将根据分析层产生的不同聚类结果为其提供相应的服务。例如，设备智能维护应用可以根据设备运行时的不同参数产生不同的聚类结果，为用户提供不同的设备健康预测服务，便于监控各种影响设备运转因素的状态。

上述框架中的三个层面相辅相成，体现了信息化技术与大数据应用的有机集成。本节主要研究分析层中基于张量的多聚类分析方法，下面将对该方法进行详细阐述。

3.2.1 张量空间数据对象表示模型

为了构建大数据环境下灵活的多聚类方法，首先需要对多源异构数据对象构建统一表示模型，以便对异构数据对象的相似性进行统一的度量。本文利用[80]提出的张量化方法将异构数据对象转化为统一的对象张量模型。例如，GPS 数据可以表示为一个三阶子张量 $\Re^{I_{tim}\times I_x\times I_y}$，其中 I_{tim},I_x,I_y 分别表示时间、纬度和经度；视频数据可以表示一个四阶子张量 $\Re^{I_{tim}\times I_f\times I_w\times I_h}$，其中 I_{tim},I_f,I_w,I_h 分别表示时间、帧、宽度和高度。如图 3.2 所示，通过张量扩展运算，将这两个子张量扩展为统一的六阶张量。

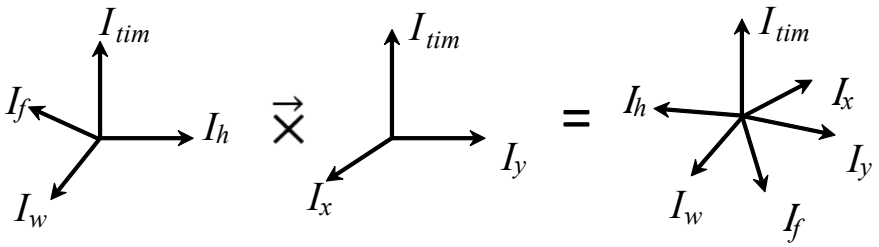


图 3.2 子张量扩展为统一张量过程

3.2.2 多线性属性组合权重学习方法

为了评估不同特征空间属性的重要性，确保噪声属性不会降低最终结果的聚类质

量，本文研究一种权重张量学习方法来计算不同属性组合的权重。本小节首先介绍如何构建关联张量；然后提出一种多线性属性权重排名方法，为每个特征空间的属性生成排名向量；最后，研究将各个特征空间属性权重排名向量融合成属性组合权重张量。

3.2.2.1 关联张量

假设数据集包含 n 个张量化的数据对象：$\mathcal{X}_1,\mathcal{X}_2,\ldots,\mathcal{X}_n$，每个数据对象均可表示为 k 阶张量 $\Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$，其中 $I_{f_1},I_{f_2},\ldots,I_{f_k}$ 分别对应由不同属性描述的 k 个特征空间，则将每个对象张量的非零元素转换为 1 并进行累加可获得关联张量 $\mathcal{T}_a\in\Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$。关联张量中具有非负整数值的元素 $t_{i_1i_2\cdots i_k}$ 表示对应的属性组合 $i_1,i_2,\ldots,i_k$ 共同出现在所有对象中的次数。例如，图 3.3 表示一个由 7 个对象张量计算得到的三阶关联张量，其中坐标（2,3,1）对应的三个空间属性共关联 3 次；而坐标（3,4,1）的元素值为 0，表示对应的三个空间属性不相关。

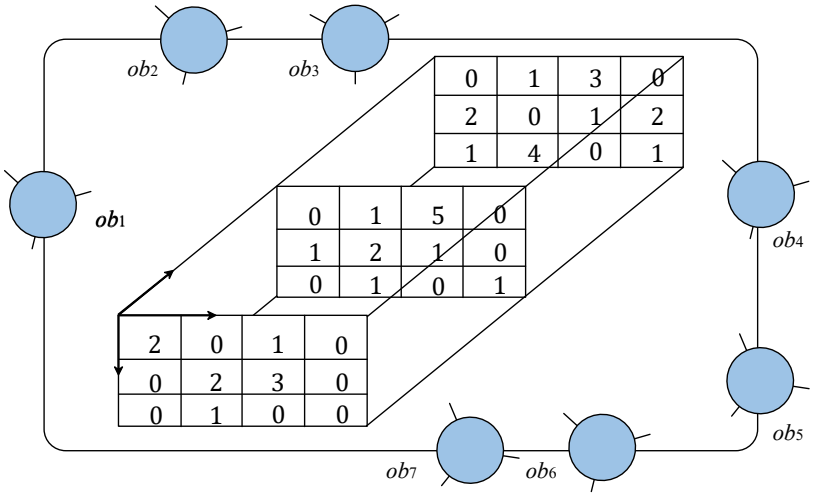


图 3.3 三阶关联张量

3.2.2.2 多线性属性权重排名方法

根据多线性数据排名方法[81,82]，本文利用多线性 PageRank 理论和高阶幂法计算属性排名向量 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_k$，它们分别表示各个特征空间每个属性的重要性。

首先，将前面得到的关联张量 $\mathcal{T}_a$ 转换为转移张量 $\mathcal{T}_{tr}^{(l)}(l=1,2,\ldots,k)$，其元素计算公式为

$$t^{tr(l)}_{i_1\cdots i_l\cdots i_k}=\frac{t^{a}_{i_1\cdots i_l\cdots i_k}}{\sum_{i_l=1}^{z}t^{a}_{i_1\cdots i_l\cdots i_k}}, \tag{3.1}$$

其中，z 表示关联张量 $\mathcal{T}_a$ 所有阶的最大维度。例如，图 3.4 展示一个从三阶关联张量转换得到的三阶转移张量。因为转移张量要求为一个超对称张量[83]，所以需要先将 3×4×3 的关联张量通过补零进行扩维，共加入 28 个零元素到 3×4×3 的关联张量中，然后沿第 1 阶的各个纤进行归一化，得到 4×4×4 的转移张量模型 $\mathcal{T}^{(1)}_{tr}$。对于所有的元素为零的纤，称为悬挂点[74]，除了沿第 1 阶扩展的 0 以外，其余值可以设置为 $1/I_f$。同样方法可以构造第二、三阶对应的转移张量 $\mathcal{T}^{(2)}_{tr}$ 和 $\mathcal{T}^{(3)}_{tr}$。

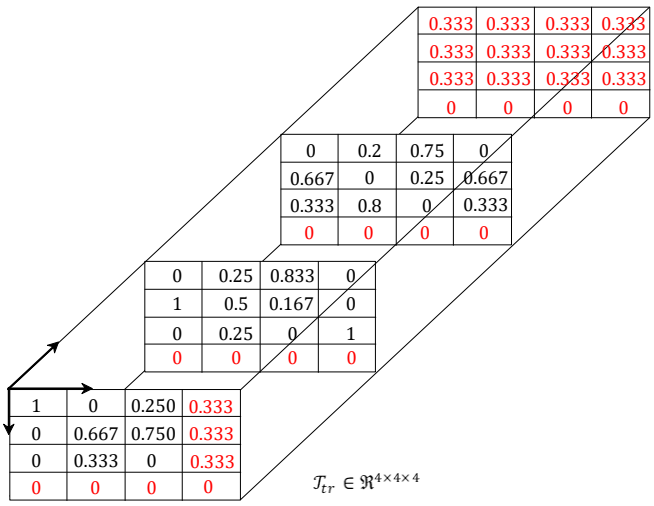


图 3.4 三阶转移张量

其次，对于每一个排名向量 $\mathrm{w}_l(l=1,2,\ldots,k)$，其计算公式如下

$$\mathrm{w}_l=\mathcal{T}^{(l)}_{tr}\times_1\mathrm{w}_l\cdots\times_{l-1}\mathrm{w}_l\times_{l+1}\mathrm{w}_l\cdots\times_k\mathrm{w}_l, \tag{3.2}$$

其中 w_l 是第 l 个 k 阶转移张量 $\mathcal{T}^{(l)}_{tr}$ 的主特征值 1 对应的特征向量，进而为了保证收敛[81]，在公式（3.2）中引入素性修正参数 α 后得到如下公式

$$\mathrm{w}_l=\alpha\mathcal{T}^{(l)}_{tr}\times_1\mathrm{w}_l\cdots\times_{l-1}\mathrm{w}_l\times_{l+1}\mathrm{w}_l\cdots\times_k\mathrm{w}_l+(1-\alpha)\mathrm{u}, \tag{3.3}$$

其中 α 是一个满足 $0<\alpha<1$ 的概率，u 是一个随机向量。

基于公式（3.3）并扩展多线性 PageRank 理论和高阶幂法，从而给出迭代求解属性排名向量 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_k$ 的算法如下：

算法 3.1：多线性属性权重排名算法

输入：k 阶转移张量 $\mathcal{T}_{tr}^{(1)},\mathcal{T}_{tr}^{(2)},\ldots,\mathcal{T}_{tr}^{(k)} \in \Re^{\overbrace{m\times m\times\cdots\times m}^{k}}$ ，各特征空间的属性维数分别为 $I_{f_1},I_{f_2},\ldots,I_{f_k}$。

输出：属性权重排名向量 $\mathrm{w}_1 \in \Re^{I_{f1}},\mathrm{w}_2 \in \Re^{I_{f2}},\ldots,\mathrm{w}_k \in \Re^{I_{fk}}$ 。

1：设置素性修正概率参数 $0<\alpha<1$；

2：选择阈值 ε；

3：for l=1 to k do

4： 初始化向量 w_0，满足 $\sum_{i=0}^{m}\left[\mathrm{w}_0\right]_i=1$ ；

5： 设置随机向量 u，满足 $\sum_{i=1}^{m}\left[\mathrm{u}\right]_i=1$ ；

6： 初始化变量 $j=0$；

7： 重复执行下列操作

8： $j=j+1$；

9： 连续单模乘，$\mathrm{w}_j \leftarrow \alpha\mathcal{T}_{tr}^{(l)}\times_1 \mathrm{w}_{j-1}\cdots\times_{l-1}\mathrm{w}_{j-1}\times_{l+1}\mathrm{w}_{j-1}\cdots\times_k \mathrm{w}_{j-1}+(1-\alpha)\mathrm{u}$；

11： 直到满足 $\left\|\mathrm{w}_j-\mathrm{w}_{j-1}\right\|<\varepsilon$ ；

12： 截取 w_j 的前 I_{f_l} 个元素作为排名向量 w_l；

13：end for

14：返回 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_k$。

3.2.2.3 权重张量

根据算法 3.1 计算得到各特征空间属性权重排名向量 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_k$，利用向量外积运算计算属性组合权重张量 $\mathcal{T}_w \in \Re^{I_{f1}\times I_{f2}\times\cdots\times I_{fk}}$，公式如下

$$\mathcal{T}_w=\mathrm{w}_1\circ\mathrm{w}_2\circ\cdots\circ\mathrm{w}_k. \qquad (3.4)$$

权重张量 $\mathcal{T}_w$ 的元素表示所有特征空间中各种属性组合的重要性。运用该方法得到的权重张量可以有效地平衡属性的相对贡献，从而保证噪声属性不会降低最终的多聚类结果质量。

3.2.3 可选择加权张量距离

根据第 2 章的介绍，在[67]中提出的张量距离能够有效的度量两个高阶空间对象之间的距离。然而，现有的张量距离定义认为对于坐标对应的所有属性组合都是同等重要的，且与元素位置距离相关的度量系数是固定的。本文提出了一种可选择加权张量距离，通过在原有张量距离中引入特征选择系数和权重因子，不仅可以灵活地根据上下情境选择属性组合，还能提高最终结果的聚类质量。

给定一个对象 $\mathcal{X}$，$\mathcal{Y} \in \Re^{I_{f1} \times I_{f2} \times \cdots \times I_{fk}}$，$\mathcal{X}$，$\mathcal{Y}$ 分别表示它们的向量化形式。类似的，w 表示权重张量 $\mathcal{T}_w \in \Re^{I_{f1} \times I_{f2} \times \cdots \times I_{fk}}$ 的向量化形式。在公式(2.5)的基础上，将权重张量 $\mathcal{T}_w$ 以向量化形式加入张量距离中，可以得到两个对象张量 $\mathcal{X}$和$\mathcal{Y}$ 的可选择加权张量距离（Selective Weighted Tensor Distance，SWTD），计算公式如下：

$$\begin{aligned} d_{SWTD} &= \sqrt{\sum_{l,m=1}^{I_{f1} \times I_{f2} \times \cdots \times I_{fk}} g_{lm} w_l (x_l - y_l) w_m (x_m - y_m)} \\ &= \sqrt{(\mathrm{w} * (\mathrm{x\text{-}y}))^T G (\mathrm{w} * (\mathrm{x\text{-}y}))}, \end{aligned} \tag{3.5}$$

其中 w_l，w_m 分别是位置 $i_1 i_2 \ldots i_k$（对应 l）和 $i_1^{'} i_2^{'} \ldots i_k^{'}$（对应 m）的权重因子，表示该位置对应的属性组合的重要性。

此外，为了能够根据上下文情境灵活地选择不同的特征空间组合，产生一些有意义的多聚类结果，为大数据应用提供有价值的服务，故在张量距离中需引入特征选择系数。通过对张量距离中各参数的意义及相互关系做理论分析，结合实验评测选择最优的归一化参数设置，选择不同的张量元素计算混合距离得到不同的相似度，分析其与选择不同的特征空间组合的相互关系及普适规律性，对度量矩阵 G 定义如下

$$g_{lm} = \frac{1}{2\pi\sigma^2} \exp\left\{ \frac{-\left\| p_l - p_m \right\|_2^2}{2\sigma^2} \right\}, \tag{3.6}$$

其中，位置距离 $\left\| p_l - p_m \right\|_2$ 定义如下

$$\left\| p_l - p_m \right\|_2 = \sqrt{v_1 (i_1 - i_1^{'})^2 + v_2 (i_2 - i_2^{'})^2 + \cdots + v_k (i_k - i_k^{'})^2}, \tag{3.7}$$

其中，对应张量空间第 j 阶的 $v_j (1 \le j \le k)$ 是特征空间组合向量 $\mathrm{v} \in \{0,1\}^k$ 的元素，表示

第 j 个特征空间是否被选择，如果选择了第 j 个特征空间，则其值为 1，否则为 0。

3.2.4 多视图张量

根据前面方法得到的权重张量 $\mathcal{T}_w$ 和所选特征空间组合向量 v，利用可选择加权张量距离度量张量空间对象之间的距离，计算得到视图矩阵 S_V，集成视图矩阵 $S_V^{(1)},S_V^{(2)},\cdots,S_V^{(r)}$（对应特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r$）可得到多视图张量 $\mathcal{T}_{mv}\in\Re^{I_{ob}\times I_{ob}\times I_{vi}}$。因为 k 个特征空间有 2^k 个不同的组合，r 的取值范围从 1 到 2^k。例如，图 3.5 给出了一个由 10 个具有 5 个特征空间的对象张量构造的三阶多视图张量。在多视图张量 $\mathcal{T}_{mv}$ 中，视图矩阵 $S_V^{(2)}$ 对应特征空间选择向量 v_2=（0,1,0,1,0）对应的相似矩阵，表示选取了第二、第四个特征空间。

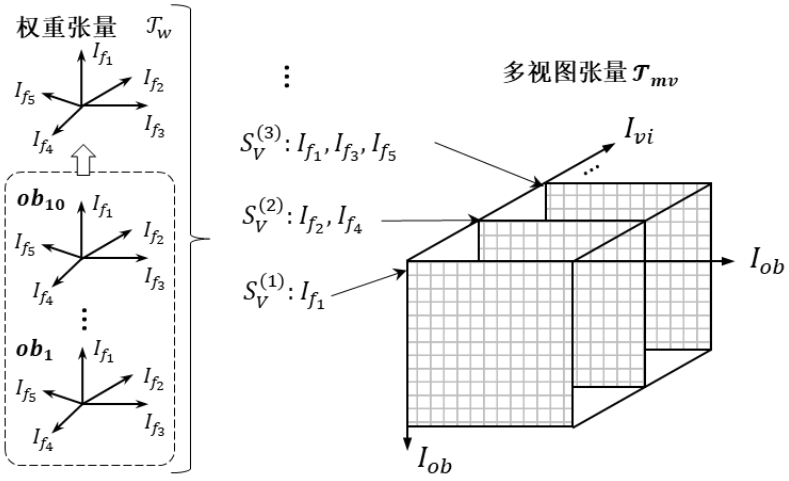


图 3.5 三阶多视图张量

3.2.5 基于张量的多聚类算法

在前面的模型和方法基础上，本文提出了基于张量的多聚类算法，具体描述如下：

算法 3.2：基于张量的多聚类算法

输入：对象张量 $\mathcal{X}_1,\mathcal{X}_2,\ldots,\mathcal{X}_n\in\Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$，

特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r\in\{0,1\}^k$。

输出：多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

1：将对象张量向量化 $\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_n$；

2：转换原始对象张量非零元素为 1，得到新的对象张量 $\mathcal{X}_1^{'},\mathcal{X}_2^{'},\ldots,\mathcal{X}_n^{'}$；

3：计算关联张量 $\mathcal{T}_a \leftarrow \sum_{d=1}^{n}\mathcal{X}_d^{'}$；

4：设置 $z \leftarrow \max\left\{I_{f_1},I_{f_2},\ldots,I_{f_k}\right\}$；

5：for l=1 to k do

6：　　计算转移张量 $\mathcal{T}_{tr}^{(l)}$；

7：end for

8：根据算法 3.1 计算属性权重排名向量 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_k$；

9：计算权重张量 $\mathcal{T}_w \leftarrow \mathrm{w}_1 \circ \mathrm{w}_2 \circ \cdots \circ \mathrm{w}_k$；

10：for q=1 to r do

11：　　for l=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

12：　　for m=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

13：　　计算位置距离 $\left\|p_l-p_m\right\|_2 \leftarrow \sqrt{\sum_{t=1}^{k}v_{q_t}(i_t-i_t^{'})^2}$；

14：　　计算距离矩阵 $g(l,m) \leftarrow \frac{1}{2\pi\sigma^2}\exp\left\{\frac{-\left\|p_l-p_m\right\|_2^2}{2\sigma^2}\right\}$；

15：　　end for

16：　　end for

17：　　for j=1 to n do

18：　　for h= j+1 to n do

19：　　计算视图矩阵 $S_V^{(q)}(j,h) \leftarrow \sqrt{(\mathrm{w}*(\mathrm{x}_j\text{-}\mathrm{x}_h))^T\,\mathrm{G}\,(\mathrm{w}*(\mathrm{x}_j\text{-}\mathrm{x}_h))}$；

20：　　end for

21：　　end for

22：end for

23：利用得到的 $S_V^{(1)},S_V^{(2)},\ldots,S_V^{(r)}$ 构建多视图张量 $\mathcal{T}_{mv}$；

24：将多视图张量$\mathcal{T}_{mv}$作为典型聚类算法的输入并行产生多聚类结果；

25：返回多聚类结果$cl_1, cl_2, \ldots, cl_r$。

在算法的第 24 步，可以选择任意以距离作为输入的典型聚类算法进行聚类。

3.3 基于张量分解的多聚类方法

对于上一节介绍的基于张量的多聚类方法，其主要优点是：（1）引入特征选择系数可以根据不同的应用需求灵活地聚类对象；（2）利用加权张量距离可以提高聚类性能。然而，由于所选择的特征子空间不能与融合后的张量空间完全分离，导致未选择的特征在计算张量距离时会对其他特征产生影响，尤其是在选择较少的特征空间时。另外，由于张量距离中的度量矩阵计算会消耗大量的内存，因此对高阶高维对象的聚类效率不高。

为了解决上述问题，本文在前期工作的基础上，进一步提出了三种新的多聚类方法，以保证所选特征空间不受未选择的特征空间的影响，且适用于高阶高维对象。常规想法自然是首先考虑计算每个特征空间的相似度矩阵，然后计算所选择的特征空间相似度矩阵的加权平均，称为基于相似度矩阵的多聚类（similarity matrices-based multiple clusterings，SMMC）。但后续却发现仅通过简单的将相似度矩阵叠加实际上忽略了多源信息的融合。因此，本文提出将选取的特征空间进行组合，然后利用加权欧式距离计算相似度矩阵，称为基于欧式距离的多聚类（Euclidean distance-based multiple clusterings，EMC）。但是，这种简单的信息组合又没有考虑不同特征空间的属性之间的交互和噪声的消除。因此，本文进一步提出了一种基于张量分解的多聚类（tensor decomposition-based multiple clusterings，TDMC）。这三种多聚类方法的关系是逐步发展的。同时，为了提高聚类性能，本文还提出一种多关系属性权重排名方法并将其应用于 TDMC 中。本节将对提出的三种多聚类方法进行详细阐述。

3.3.1 基于相似度矩阵的多聚类方法

首先提出一种基于相似度矩阵的多聚类方法，可以根据不同应用的需求对异构数据对象进行聚类。本小节先介绍相似度张量和特征空间权重向量，然后给出多视图张

量和 SMMC 算法。

3.3.1.1 相似度张量和特征空间权重向量

一般来说，异构数据源包含不同的特征空间，它们有不同的属性维度和相似性度量方式，当然这些特征空间的属性在聚类过程中的重要性也是不同的。如何协调不同来源的各种信息是获得良好聚类性能的关键问题之一。因此，本文扩展[84]中提出的方法来解决这个问题，整个过程的示意图如图 3.6 所示。首先，构造一个相似度张量 $\mathcal{T}_s \in \Re^{I_{ob}\times I_{ob}\times I_{fe}}$，其中 I_{ob} 和 I_{fe} 分别表示对象和特征空间，沿阶 I_{fe} 的切片是归一化以后相似度矩阵 $S_N^{(1)}, S_N^{(2)}, \ldots, S_N^{(k)}$，对应 k 个不同的具有独立属性维数和相似性度量方式的特征空间；然后，为了给每个特征空间分配合适的权重来提高聚类性能，对相似度张量采用 HOOI 分解方法获得权重向量 w，其元素 w_i 表示第 i 个特征空间的重要性。

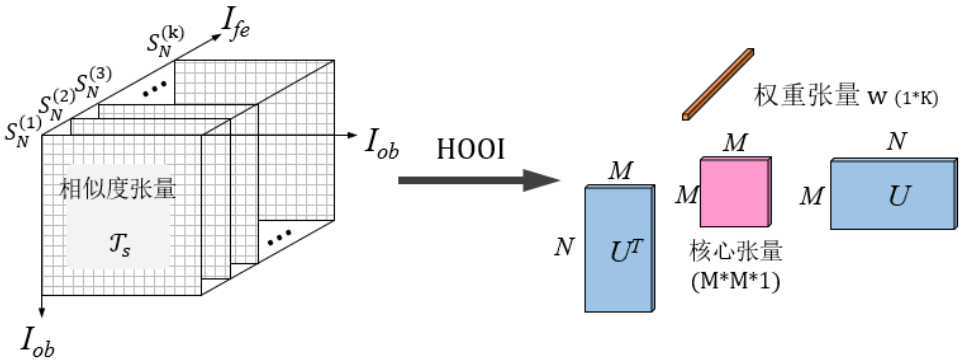


图 3.6 权重向量计算过程

3.3.1.2 多视图张量

因为不同的应用需要不同的数据特征，因此在根据不同的应用需求进行聚类时，应该允许用户选择不同的特征空间组合。受[85]的多视图聚类方法启发，本文提出一种多视图张量模型。首先，根据特征空间选择向量 $\mathrm{v} \in \{0,1\}^k$，在相似度张量 $\mathcal{T}_s$ 中选择所需的相似度矩阵，进行加权平均构成视图矩阵 S_V，其计算公式如下

$$S_V = \frac{\sum_{i=1}^{k} v_i w_i S_N^{(i)}}{\sum_{i=1}^{k} v_i w_i}, \tag{3.8}$$

其中 v_i 为特征空间选择系数，取 1 表示选择第 i 个特征空间，在这里，为了评估不同

数据源的贡献，对所选特征空间对应的权重向量 w 的元素进行归一化；然后，集成视图矩阵 $S_V^{(1)},S_V^{(2)},\ldots,S_V^{(r)}$（对应特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r$）得到多视图张量 $\mathcal{T}_{mv}\in\Re^{I_{ob}\times I_{ob}\times I_{vi}}$。因为 k 个特征空间有 2^k 个不同的组合，r 的取值范围从 1 到 2^k。例如，图 3.7 给出了一个三阶多视图张量，其中视图矩阵 $S_V^{(1)},S_V^{(2)},S_V^{(3)}$ 分别对应于 v_1=(0,1,0)、v_2=(1,0,1)和 v_3=(1,1,1)。

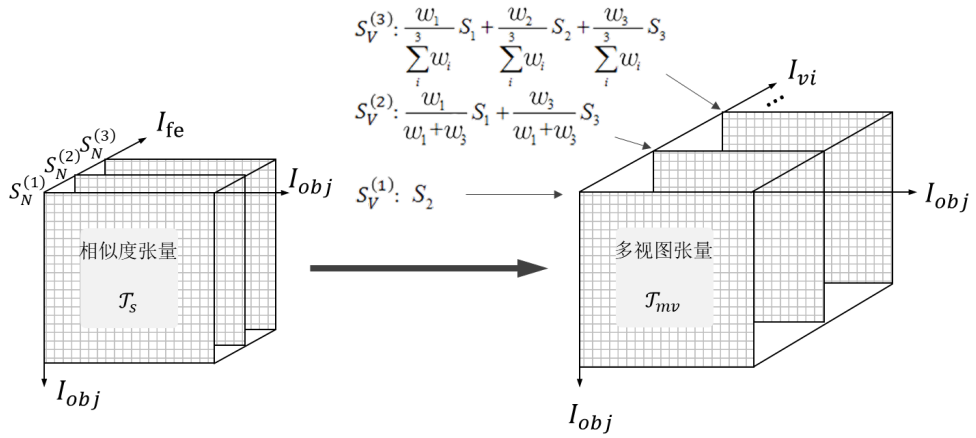


图 3.7 三阶多视图示例

3.3.1.3 基于相似度矩阵的多聚类算法

在前面的方法和模型基础上，基于相似度矩阵的多聚类算法可归纳如下：

算法 3.3：基于相似度矩阵的多聚类算法

输入：相似度矩阵 $S_N^{(1)},S_N^{(2)},\ldots,S_N^{(k)}\in\Re^{n\times n}$，特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r\in\{0,1\}^k$。

输出：多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

1：用相似度矩阵 $S_N^{(1)},S_N^{(2)},\ldots,S_N^{(k)}$ 构建相似度张量 $\mathcal{T}_s$；

2：调用 HOOI 算法获得权重向量 w；

3：for l=1 to r do

4：　　计算视图矩阵 $S_V^{(l)}\leftarrow\dfrac{\sum_{i=1}^{k}v_i^l w_i S_N^{(i)}}{\sum_{i=1}^{k}v_i^l w_i}$；

5：end for

6：利用得到的 $S_V^{(1)}, S_V^{(2)}, \ldots, S_V^{(r)}$ 构建多视图张量 $\mathcal{T}_{mv}$；

7：将多视图张量 $\mathcal{T}_{mv}$ 作为典型聚类算法的输入并行产生多聚类结果；

8：返回多聚类结果 $cl_1, cl_2, \ldots, cl_r$。

3.3.2 基于欧式距离的多聚类方法

虽然基于相似度矩阵的多聚类可以对不同的特征空间使用不同的相似性度量来计算相似性，但是这种方法忽略了所选特征空间的融合。因此，本方法根据不同的应用组合所选择的特征空间并用可选择加权欧式距离计算相似度。下面给出了多视图张量模型和相应的基于欧式距离的多聚类算法。

3.3.2.1 EMC 的多视图张量

为了能够更好的融合多源信息，EMC 首先对不同特征空间的所有属性进行归一化。假设归一化后的对象向量 $\mathrm{x}, \mathrm{y} \in \Re^{I_{f_1}+I_{f_2}+\cdots+I_{f_k}}$，其中 $I_1, I_2, \ldots, I_k$ 分别表示 k 个特征空间的维度，引入权重向量 w 和特征空间组合向量 v，可用如下公式计算 x 和 y 之间的可选择加权欧式距离（Selective Weighted Euclidean distance，SWED）

$$d_{SWED} = \sqrt{\sum_{i=1}^{k} (\mathrm{v}_i \mathrm{w}_i^{\,2} (\mathrm{x}_{f_i} - \mathrm{y}_{f_i})^T (\mathrm{x}_{f_i} - \mathrm{y}_{f_i}))}\,. \tag{3.9}$$

然后，本文用 SWED 来计算对象之间的相似度，并构造视图矩阵 $S_V^{(1)}, S_V^{(2)}, \cdots, S_V^{(r)}$，$r$ 的取值范围类似于 SMMC。最后，集成视图矩阵构建多视图张量 $\mathcal{T}_{mv} \in \Re^{I_{ob} \times I_{ob} \times I_{vi}}$。例如，图 3.8 展示了一个由四个特征空间构成的三阶多视图张量模型。在多视图张量

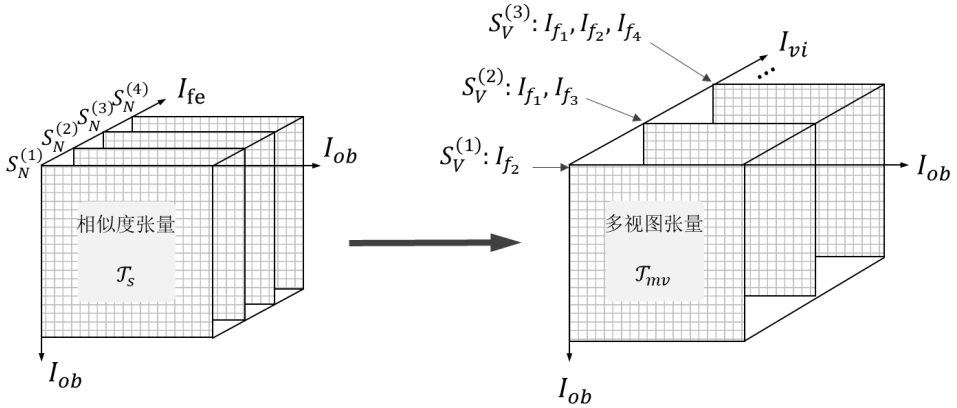


图 3.8 EMC 的三阶多视图张量

$\mathcal{T}_{mv}$ 中，视图矩阵 $S_V^{(3)}$ 是 v3=（1, 1, 0, 1）对应的相似度矩阵，表示选取的是第 1、2、4 个特征空间。

3.3.2.2 EMC 算法

根据前述模型和方法，基于欧式距离的多聚类算法可归纳如下：

算法 3.4：基于欧式距离的多聚类算法

输入：对象向量 $\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_n \in \Re^{I_{f_1}+I_{f_2}+\ldots+I_{f_k}}$，特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r \in \{0,1\}^k$。

输出：多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

1：标准化对象 $\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_n$；

2：调用 HOOI 算法获得权重向量 w；

3：for *j*=1 to *r* do

4：　for *l*=1 to *n* do

5：　　for *m*=*l*+1 to *n* do

6：　　计算视图矩阵 $S_V^{(j)}{}_{(l,m)} \leftarrow \sqrt{\sum_{i=1}^{k}(\mathrm{v}_i^j \mathrm{w}_i{}^2(\mathrm{x}_{f_i}^l-\mathrm{x}_{f_i}^m)^T(\mathrm{x}_{f_i}^l-\mathrm{x}_{f_i}^m))}$；

7：　　end for

8：　end for

9：end for

10：利用得到的 $S_V^{(1)},S_V^{(2)},\ldots,S_V^{(r)}$ 构建多视图张量 $\mathcal{T}_{mv}$；

11：将多视图张量 $\mathcal{T}_{mv}$ 作为典型聚类算法的输入并行产生多聚类结果；

12：返回多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

3.3.3 基于张量分解的多聚类方法

虽然 EMC 在计算相似度时首先融合所选择的特征空间，但仍然没有考虑属性之间的交互以及如何去除噪声。因此，本文进一步提出基于张量分解的多聚类方法。类似于基于张量的多聚类方法，本节首先给出基于张量分解的多聚类分析和服务框架，总体框架图如图 3.9 所示，同样自底向上由数据层、分析层和服务层三层组成。各层

次功能类似于基于张量的多聚类方法，明显不同在于特征组合之前需先对对象张量进行张量分解。

在接下来的小节中，首先介绍基于张量的异构数据对象表示模型；然后，研究一种度量属性相关重要性的权重张量构造方法，并提出权重张量距离，再给出 TDMC 的多视图张量模型；最后，对 TDMC 算法进行总结。

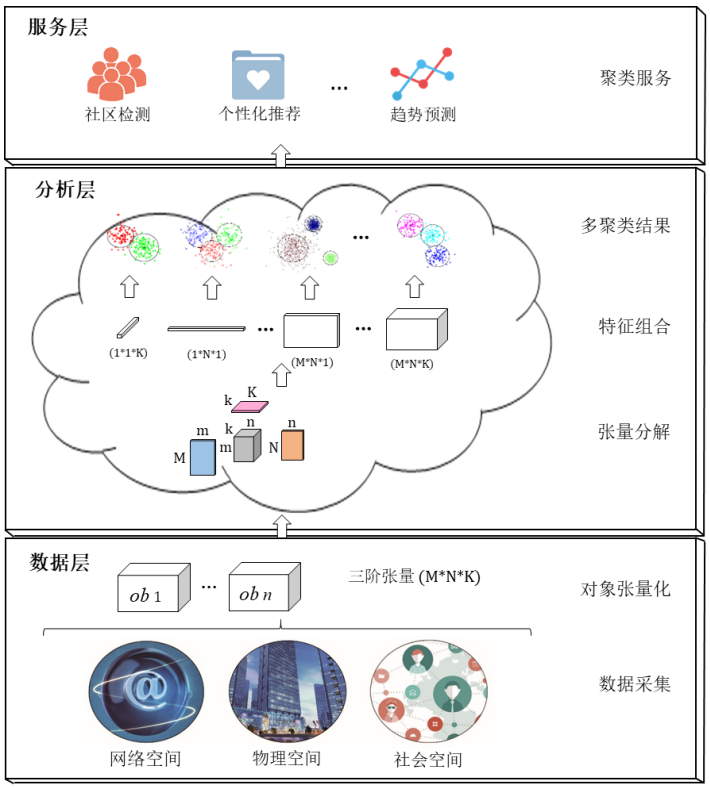


图 3.9 基于张量分解的多聚类分析和服务框架

3.3.3.1 对象张量化

TDMC 扩展了 SMMC 和 EMC 的优点，首先将多源信息融合成统一的张量表示模型，然后对这些具有异构数据的对象应用统一的相似度度量。与基于张量的多聚类方法相同，在此仍然采用张量化方法将多源数据转化为统一的张量模型。例如，当想要挖掘用户之间的不同关系时，可以考虑分别从物理空间、社交空间和网络空间来选择时间、通话对象和通话持续时间等特征，并将这些特征可以被表示成一个三阶子张量 $\Re^{I_{tim}\times I_{obj}\times I_{dur}}$，其中 I_{tim}, I_{ob}, I_{dur} 分别表示时间、通话对象和通话持续时间；此外，从物理空间采集的全球定位系统（GPS）数据可以表示一个四阶子张量 $\Re^{I_{tim}\times I_x\times I_y\times I_z}$，其中

I_{tim}, I_x, I_y, I_z 分别表示时间、经度、纬度和海拔。如图 3.10 所示，通过张量扩展运算，将这两个子张量扩展为统一的六阶张量。

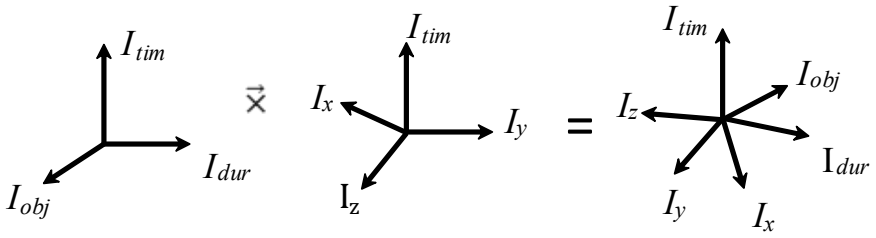


图 3.10 张量化过程示例

3.3.3.2 权重张量构造方法

同样，为了评估不同特征空间属性对聚类的重要性，同时确保噪声属性不会降低最终结果的聚类质量，本文研究一种权重张量学习方法来计算不同属性组合的权重。本小节首先介绍如何构建关联张量，然后提出一种多关系属性排名方法，为每个特征空间的属性生成排名向量；最后，融合属性排名向量构建属性组合权重张量。

（1）关联张量

与 3.2.2 中的关联张量相同，它主要表示不同特征空间的属性相关的程度。假设一个数据集包含有 n 个张量化的对象：$\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$，其中 $I_{f_1}, I_{f_2}, \ldots, I_{f_k}$ 分别对应由不同属性描述的 k 个特征空间。将所有对象的非零元素转换为 1 后进行累加获得关联张量 $\mathcal{T}_a \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$，关联张量中具有非负整数值的元素 $t^a_{i_1 i_2 \cdots i_k}$ 表示对应的属性组合 $i_1, i_2, \ldots, i_k$ 的共同出现在所有对象中的次数。例如，图 3.11 表示一个由 6 个对象张量计算得到的三阶关联张量。在三个特征空间中，第 3、第 2、第 1 个属性分别与 6 个对象共 4 次相关，第 1、第 2、第 1 个属性不相关。这里省略了元素值 0，意味着属性不相关。

（2）多关系属性排名方法

为了评估每个属性的重要性，可以根据相关性计算它们的得分。在多关系数据分析中的中心、重要性和相关性得分的启发下，可认为不同特征空间中的属性得分也具有相互增强的关系。因此，根据[76, 86]中介绍的定理和算法，本节提出的多关系属性排名方法可以看作是对 HAR 算法的一种推广。

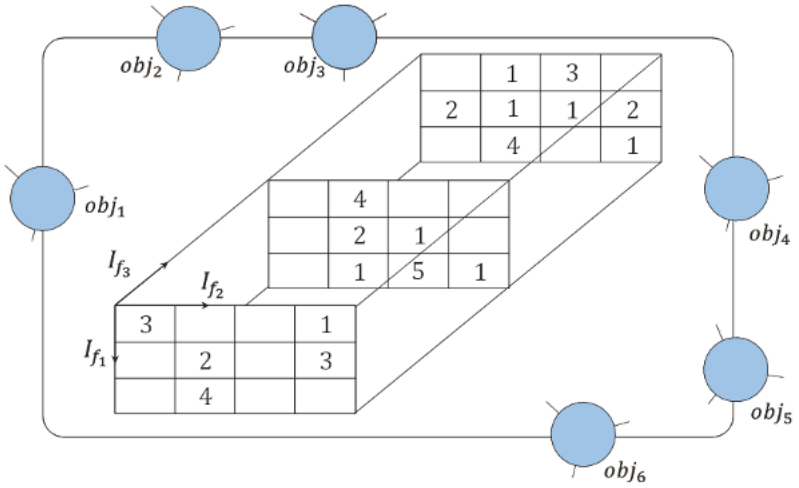


图 3.11 三阶关联张量

首先，将前面得到的关联张量 $\mathcal{T}_a$ 通过归一化转换为转移张量 $\mathcal{T}_{tr}^{(l)}(l=1,2,\ldots,k)$，其元素计算公式为

$$t_{i_1\ldots i_l\ldots i_k}^{tr(l)}=\frac{t_{i_1\ldots i_l\ldots i_k}^{a}}{\sum_{i_l=1}^{I_{fl}} t_{i_1\ldots i_l\ldots i_k}^{a}}. \tag{3.10}$$

不同于多线性属性排名方法，这里不需要先做补零操作，只需转移张量和关联张量以保持同阶同维，其中转移张量的元素 $t_{i_1\ldots i_l\ldots i_k}^{tr(l)}$ 表示当给定其它特征空间的属性时，所有对象具有第 l 个特征空间的第 i_l 个属性值的概率。类似于 HAR 算法，对于所有 $1\le i_l\le I_{f_l}$，元素 $t_{i_1\ldots i_l\ldots i_k}^{tr(l)}$ 的值都为零，则将它们的值都设为 $1/z$。例如，图 3.12 展示一个从图 3.11 的三阶关联张量转换得到的三阶转移张量。因此，直接沿第一阶的各个纤进行归一化即可得到 3×4×3 的转移张量 $\mathcal{T}_{tr}^{(1)}$。在图中，t_{131},t_{231},t_{331} 的值都设置为 1/3。类似地，可以构造第二、三阶对应的转移张量 $\mathcal{T}_{tr}^{(2)},\mathcal{T}_{tr}^{(3)}$。

其次，这里的权重向量 $\mathrm{w}_l(l=1,2,\ldots,k)$ 表示每个特征空间属性的得分，可以用如下多元多项式进行求解

$$\mathrm{w}_l=\alpha_l\mathcal{T}_{tr}^{(l)}\times_1\mathrm{w}_1\cdots\times_{l-1}\mathrm{w}_{l-1}\times_{l+1}\mathrm{w}_{l+1}\cdots\times_k\mathrm{w}_k+(1-\alpha_l)\mathrm{e}_l, \tag{3.11}$$

其中，每个权重向量满足元素和为 1。

进一步，为了保证算法收敛[81]，在公式(3.10)中引入素性修正参数 α，得到如下

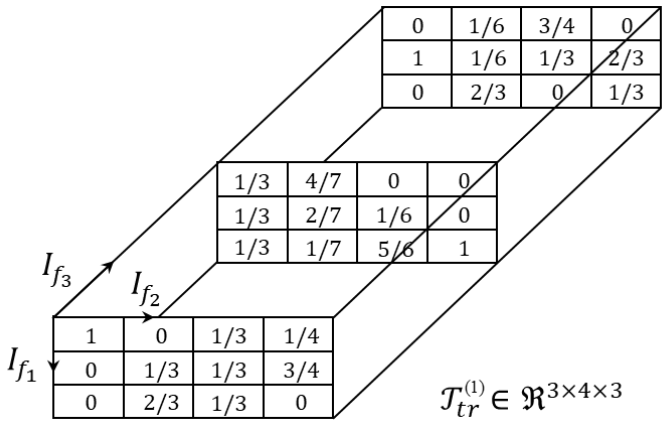


图 3.12 三阶转移张量

公式

$$\mathrm{w}_l = \alpha_l \mathcal{T}_{tr}^{(l)} \times_1 \mathrm{w}_1 \cdots \times_{l-1} \mathrm{w}_{l-1} \times_{l+1} \mathrm{w}_{l+1} \cdots \times_k \mathrm{w}_k + (1-\alpha_l)\mathrm{e}_l , \tag{3.12}$$

其中，α 是一个满足 $0<\alpha<1$ 的概率，$e_l \in \Re^{I_l}$ 是对第 l 阶随机分配且满足元素和为 1 的概率分布。

命题 1 令 $\mathcal{T}_{tr}^{(1)}, \mathcal{T}_{tr}^{(2)}, \ldots, \mathcal{T}_{tr}^{(k)}$ 为 k 阶转移张量，$0<\alpha_1, \alpha_2, \ldots, \alpha_k<1$，$e_1, e_2, \ldots, e_k$ 是给定的修正参数和分配的概率分布。如果对于 $l=1,2,\ldots,k$，$\mathcal{T}_{tr}^{(l)}$ 是不可约的，那么存在向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k > 0$ 满足公式(3.12)且解向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k$ 都是唯一的。

通过扩展 HAR 算法，迭代求解公式(3.12)得到权重向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k$ 作为属性得分，多关系属性权重排名算法归纳如下。

算法 3.5：多关系属性权重排名算法

输入：k 阶转移张量 $\mathcal{T}_{tr}^{(1)}, \mathcal{T}_{tr}^{(2)}, \ldots, \mathcal{T}_{tr}^{(k)} \in \Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$，各特征空间的属性维数分别为 $I_{f_1}, I_{f_2}, \ldots, I_{f_k}$。

输出：属性权重排名向量 $\mathrm{w}_1 \in \Re^{I_{f_1}}, \mathrm{w}_2 \in \Re^{I_{f_2}}, \ldots, \mathrm{w}_k \in \Re^{I_{f_k}}$。

1：for l=1 to k do

2：　初始化向量 $(\mathrm{w}_l)_0$，满足 $\sum_{i=1}^{I_l} (\mathrm{w}_l)_0^{(i)} = 1$；

3：　设置分配的概率分布 $\mathrm{e}_l \leftarrow 1$；

4：　设置不可约参数 $0<\alpha_l<1$；

5：end for

6：选择阈值 ε；

7：初始化变量 $j \leftarrow 1$；

8：for l=1 to k do

9：　　计算第 j 次迭代得到的权重向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k$，

$$(\mathrm{w}_l)_j \leftarrow \alpha_l \mathcal{T}_{tr}^{(l)} \times_1 (\mathrm{w}_1)_j \cdots \times_{l-1} (\mathrm{w}_{l-1})_j \times_{l+1} (\mathrm{w}_{l+1})_{j-1} \cdots \times_k (\mathrm{w}_k)_{j-1} + (1-\alpha_l)\mathrm{e}_l ;$$

10：end for

11：如果满足 $\sum_{l=1}^{k} \left\| (\mathrm{w}_l)_j - (\mathrm{w}_l)_{j-1} \right\| < \varepsilon$，则停止；否则，设置 $j = j+1$ 并且转 8 步；

12：for l=1 to k do

13：　　赋值属性排名向量 $\mathrm{w}_l \leftarrow (\mathrm{w}_l)_j$；

14：end for

15：返回属性权重排名向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k$。

（3）权重张量

权重张量的构建和 3.2.2.3 节介绍的方法相同，利用公式(3.4)的向量外积运算去计算属性组合权重张量 $\mathcal{T}_w \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$。同样地，权重张量 $\mathcal{T}_w$ 的元素表示所有特征空间中各种属性组合的重要性，利用它可以有效地平衡属性的相对贡献，从而保证噪声属性不会降低最终的多聚类结果质量。

3.3.3.3 加权张量距离

加权张量距离是将 3.2.3 节介绍的可选择加权张量距离中去掉了选择系数，只保留了权重因子。因此，给定对象张量 $\mathcal{X}$，$\mathcal{Y} \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$，x，y 分别表示它们的向量化形式。类似地，w 表示权重张量 $\mathcal{T}_w \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$ 的向量化形式。两个对象张量 $\mathcal{X}$和$\mathcal{Y}$ 的加权张量距离（selective weighted tensor distance，WTD），计算公式如下

$$\begin{aligned} d_{WTD} &= \sqrt{\sum_{l,m=1}^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}} g_{lm} w_l (x_l - y_l) w_m (x_m - y_m)} \\ &= \sqrt{(\mathrm{w} * (\mathrm{x}\text{-}\mathrm{y}))^T G (\mathrm{w} * (\mathrm{x}\text{-}\mathrm{y}))}, \end{aligned} \tag{3.13}$$

其中度量 G 为公式(2.6)。

3.3.3.4 TDMC 的多视图张量

在 TDMC 方法中，利用张量分解的方法获取原始对象张量和权重张量各特征空间的主成分及其核心张量，并根据所选特征组合构造近似对象张量和近似权重张量；然后应用 WTD 计算对象之间的相似度得到视图矩阵，通过集成这些矩阵得到多视图张量。

多视图张量构造的关键步骤如下：

步骤 1：张量分解

对于原始对象张量和权重张量，本文采用 HOOI 张量分解方法进行维度约简和核心数据提取。因此，它只存储包含每个特征空间的主成分的截断因子矩阵和决定因子矩阵之间关系的核心张量。这里，权重张量和对象张量在截断时保持每个阶的维度相同。

步骤 2：近似张量构建

根据不同的应用，选取所需特征空间的因子矩阵，结合核心张量计算得到仅包含所需特征空间主成分的近似对象张量和近似权重张量。

步骤 3：视图矩阵计算

利用 WTD 度量近似对象张量之间的距离，构造视图矩阵 S_V。如果有 r 个选择的特征空间组合向量，则可以得到视图矩阵 $S_V^{(1)}, S_V^{(2)}, \ldots, S_V^{(r)}$， r 的取值范围类似于 SMMC。

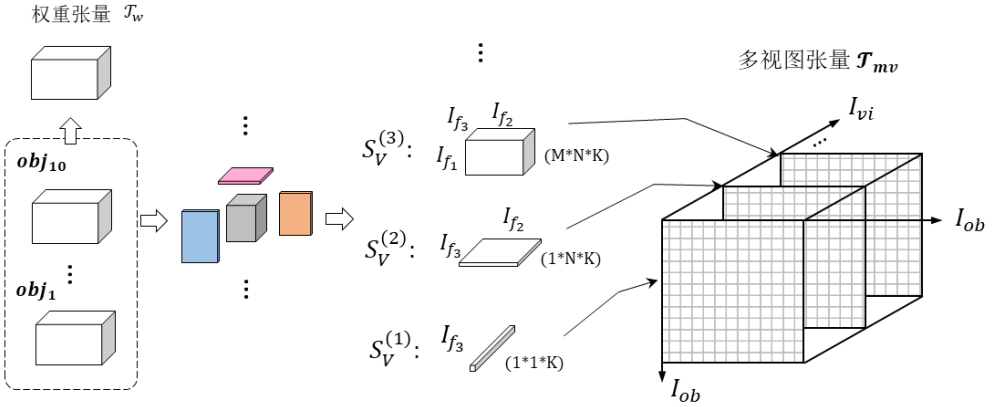


图 3.13 一个 TDMC 的多视图张量

步骤 4：多视图张量构建

集成视图矩阵构建多视图张量 $\mathcal{T}_{mv} \in \Re^{I_{ob} \times I_{ob} \times I_{vi}}$。例如，图 3.13 展示了一个三阶多视图张量模型。在多视图张量 $\mathcal{T}_{mv}$ 中，以视图矩阵 $S_V^{(2)}$ 为例，它是 $\mathrm{v}_2 = (0,1,1)$对应的相似度矩阵，表示选取的是第 1、2、4 个特征空间。

3.3.3.5 TDMC 算法

在前面的模型和方法基础上，本文提出的基于张量分解的多聚类算法可具体描述如下：

算法 3.6：基于张量分解的多聚类算法

输入：对象张量 $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$，特征空间选择向量 $\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_r \in \{0,1\}^k$。

输出：多聚类结果 $cl_1, cl_2, \ldots, cl_r$。

1：转换原始对象张量的非零元素为 1，得到新的对象张量 $\mathcal{X}_1', \mathcal{X}_2', \ldots, \mathcal{X}_n'$；

2：计算关联张量 $\mathcal{T}_a \leftarrow \sum_{d=1}^{n} \mathcal{X}_d'$；

3：for l=1 to k do

4：　　计算转移张量 $\mathcal{T}_{tr}^{(l)}$；

5：end for

6：根据算法 3.5 计算属性权重排名向量 $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k$；

7：计算权重张量 $\mathcal{T}_w \leftarrow \mathrm{w}_1 \circ \mathrm{w}_2 \circ \cdots \circ \mathrm{w}_k$；

8：调用 HOOI 算法分解权重张量，

$(S_w, U_{w(1)}, U_{w(2)}, \ldots, U_{w(k)}) \leftarrow HOOI(\mathcal{T}_w, z_1, z_2, \ldots, z_k)$；

9：for h=1 to n do

10：　　调用 HOOI 算法分解原始对象张量，

$(S_h, U_{h(1)}, U_{h(2)}, \ldots, U_{h(k)}) \leftarrow HOOI(X_h, z_1, z_2, \ldots, z_k)$；

11：end for

12：for j=1 to r do

13： 取 v_j 中值为 1 的索引并放入向量 d；

14： 设置 q 记录向量 d 的元素个数；

15： 构建近似权重张量，$\widehat{\mathcal{T}_w} \leftarrow S_w \times_{d_1} U_{w(d_1)} \times_{d_2} U_{w(d_2)} \cdots \times_{d_q} U_{w(d_q)}$；

16： 构建近似权重张量，$\widehat{\mathcal{T}_w} \leftarrow S_w \times_{d_1} U_{w(d_1)} \times_{d_2} U_{w(d_2)} \cdots \times_{d_q} U_{w(d_q)}$；

17： 将 $\widehat{\mathcal{T}_w}$ 向量化为 w；

18： for h=1 to n do

19： 构建近似对象张量，$\widehat{X_h} \leftarrow S_h \times_{d_1} U_{h(d_1)} \times_{d_2} U_{h(d_2)} \cdots \times_{d_q} U_{h(d_q)}$；

20： 将 $\widehat{X_h}$ 向量化为 x_h；

21： end for

22： for a=1 to $I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}$ do

23： for b=1 to $I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}$ do

24： 计算位置距离 $\|p_a - p_b\|_2 \leftarrow \sqrt{\sum_{t=1}^{k} (i_t - i_t^{'})^2}$；

25： 计算距离矩阵 $g(a,b) \leftarrow \frac{1}{2\pi\sigma^2} \exp\left\{\frac{-\|p_a - p_b\|_2^2}{2\sigma^2}\right\}$；

26： end for

27： end for

28： for l=1 to n do

29： for m= j+1 to n do

30： 计算视图矩阵 $S_V^{(q)}(l,m) \leftarrow \sqrt{(\mathrm{w} * (\mathrm{x}_l - \mathrm{x}_m))^T \mathrm{G} (\mathrm{w} * (\mathrm{x}_l - \mathrm{x}_m))}$；

31： end for

32： end for

33：end for

34：利用得到的 $S_V^{(1)}, S_V^{(2)}, \ldots, S_V^{(r)}$ 构建多视图张量 $\mathcal{T}_{mv}$；

35：将多视图张量$\mathcal{T}_{mv}$作为典型聚类算法的输入并行产生多聚类结果；

36：返回多聚类结果$cl_1, cl_2, \ldots, cl_r$。

在算法的第 35 步，可以选择任意以距离作为输入的典型聚类算法进行聚类。

3.4 实验分析

本节分别对基于张量的多聚类算法和基于张量分解的相关多聚类算法进行实验验证，实验是在 3.20 Ghz Intel Core i5 3470 CPU 和 8 GB RAM 的 PC 上实现的，软件环境是 Windows 10。典型聚类算法选用的是 2007 发表在 Science 上仿射传播（affinity propagation, AP）聚类算法[87]，它将所有对象都看作聚类中心，这样聚类质量不用依赖初始聚类中心的选择。最后给出基于张量的和基于张量分解的多聚类方法有代表性的实验结果。

3.4.1 基于张量的多聚类方法性能分析

本节将提出的基于张量的多聚类方法应用于共享自行车维护系统，首先介绍评价方法和数据集，最后给出实验结果和分析。本实验将 TMC 与另外两种方法进行比较，分别用不加权的张量距离（TD）和欧式距离（ED）代替 TMC 中的 SWTD 来测量数据对象的相似性，归一化参数σ和概率参数α分别设置为 0.2 和 0.3。

3.4.1.1 复杂度分析

本小节从理论上分析 TMC 方法的计算成本。假设数据集中有 n 个对象张量，每个对象张量的元素个数为 m。

在 TMC 方法中，总的计算成本T_{total}包括计算转移张量的成本T_{tr}、构建权重张量的成本T_{sawr}、计算可选择加权张量距离矩阵的成本T_{sswtd}、AP 聚类算法的成本T_{AP}，算法整体计算复杂度定义为

$$T_{total} = T_{tr} + T_{sawr} + T_{sswtd} + T_{AP}, \tag{3.14}$$

其中，转移张量的计算主要是关联张量，即将所有原始张量进行累加并对每个非零元素做除法的时间，其时间复杂度T_{tr}为$O(mn)$；构造权重张量T_{sawr}的时间复杂度为$O(m)$，因为转移张量的每个元素都涉及一系列的数乘运算，但数乘的次数是一个常

数；对于计算可选择加权张量距离矩阵的成本T_{sswtd}，因为计算一个距离，对于每一个g_{lm}必须执行四个乘法和两个减法，共需计算m^2个g_{lm}，总共需要计算$n(n-1)/2$个距离，因此，T_{sswtd}是$O(m^2n^2)$。此外，AP 算法的时间复杂度T_{AP}为$O(n^2\log n)$。综上所述，TMC 方法的总体时间复杂度T_{total}是$O(m^2n^2)$。

3.4.1.2 评价方法

（1）杰卡德指数（Jaccard Index，JI）

两个不同聚类结果之间的相似性度量，用来衡量聚类结果[88]之间的冗余度。假设 T 和 L 表示两种不同的聚类结果，JI 的定义如下

$$JI(T;L)=\frac{n_{11}}{n_{11}+n_{10}+n_{01}}, \tag{3.15}$$

其中n_{11}为 T 和 L 中属于相同类簇的对象对总数，n_{10}和n_{01}分别为只在 T 和只在 L 中同一类簇的对象对总数。根据公式(3.15)，较低的 JI 值比较好，因为它们表示生成的聚类结果之间的冗余较低，意味着可以产生更丰富的知识。

（2）邓恩指数（Dunn Index，DI）

最小的类间间距与最大的类内间距的比值[89]，是用来衡量聚类质量的一个广泛使用的标准，是一种关于聚类质量内部评价方法。假设聚类结果$C=\{c_1,c_2\cdots,c_k\}$，DI 的定义如下

$$DI(T)=\frac{\min_{i\neq j}\delta(c_i,c_j)}{\max_{1\le l\le k}\Delta(c_l)}, \tag{3.16}$$

其中$\delta(c_i,c_j)$是一种类内距离度量，$\Delta(c_l)$是在类簇c_l中所有对象两两之间的距离。根据公式(3.16)，DI 值越大，表示类内越紧凑，类与类之间越能区分开，聚类质量越好。

3.4.1.3 数据集

近年来被广泛部署在许多城市的自行车共享系统，为人们提供了一种绿色、便捷的出行方式，对于一个存储在云端的自行车共享系统数据集，记录了自行车从制造、使用到维护的多个特征。一个可能的分组是依据每个站点的历史记录和天气特征，制造商可以据此预测易受损组件和数量，从而提前准备以保证自行车维护系统的平稳运

行；同时，另一个分组可以根据站点位置、时间和使用记录等特征，运营商利用该聚类结果可以为自行车推荐系统提供高效服务。

表 3.1 纽约自行车共享系统数据集

自行车	#站点		325
	#记录数		473620
气象	#天气	#晴	377
		#雾/霾	28
		#小雨	33
		#中雨	11
	温度/° F		[61,88]
	风速/mi/h		[0,13.8]
	#记录数		449

本文以纽约市共享自行车系统[1]的真实数据集[90, 91]（自行车数据来自自行车共享系统，气象数据来自气象系统）进行实验，其详细统计数据如表 3.1 所示。其中，自行车数据包含 473620 条自行车共享记录，这些记录包括以下信息：开始时间、停止时间、起点站（站名、站名、站点纬度、经度）、终点站（站名、站名、站点纬度、经度）等；气象数据包含 449 条记录，每小时至少有一条记录，每条记录包含四个不同的特征：时间、天气、温度和风速。

3.4.1.4 实验结果和分析

首先对原始数据进行量化、清洗和组合等预处理，为每个站点生成预处理之后的记录，每个记录包含 4 个特征空间：交通模式、天气、温度和风速，一条记录对应一个对象张量，每个对象张量的规模是 7×4×28×14；根据获得的数据集，本次实验随机选取 72 号站点 40 条、150 条、800 条记录进行实验；对于每组记录，任意组合 4 个特征空间，得到 15 个聚类结果，分析结果如图 3.14 和图 3.15 所示。

图 3.14 为 JI 值统计图，JI 值用来测量聚类结果之间的冗余度。在每个饼图中，一种颜色表示 JI 的一个值范围，其百分比表示 JI 值在相应值范围内的比值，取不同的对象数时，JI 值在 0-0.2 范围内的分别占到 76%、86%和 70%，说明基于张量的多聚类方法生成的多聚类结果之间冗余度较低，可以产生更丰富的知识，从而为不同的

[1] https://nycopendata.socrata.com/

大数据应用提供不同的有意义的服务。

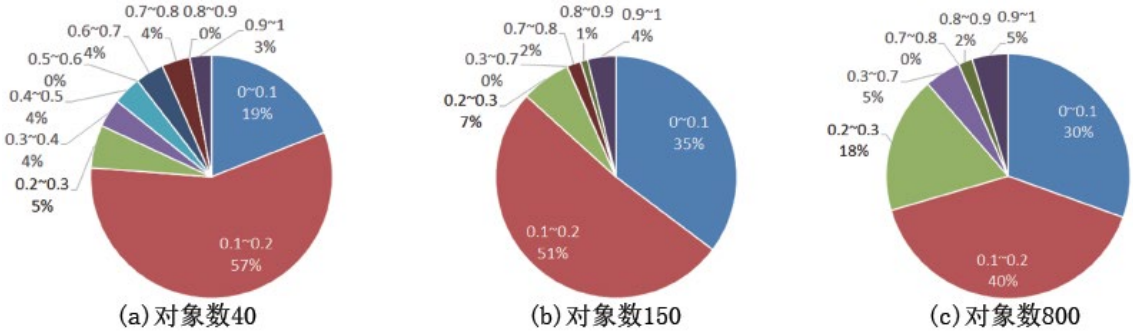
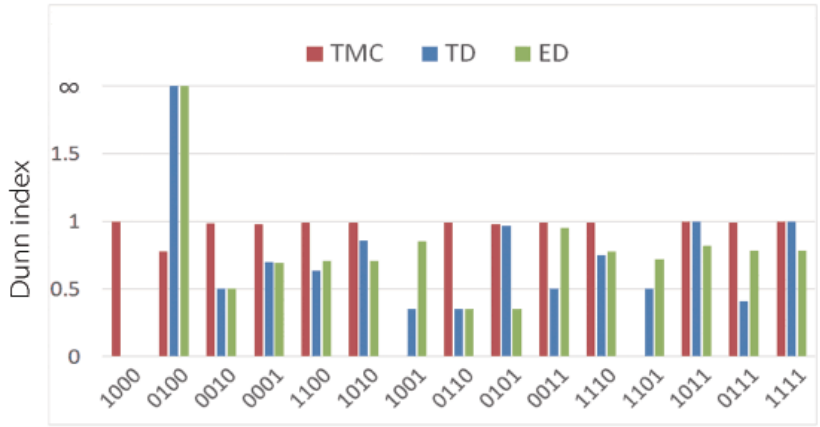


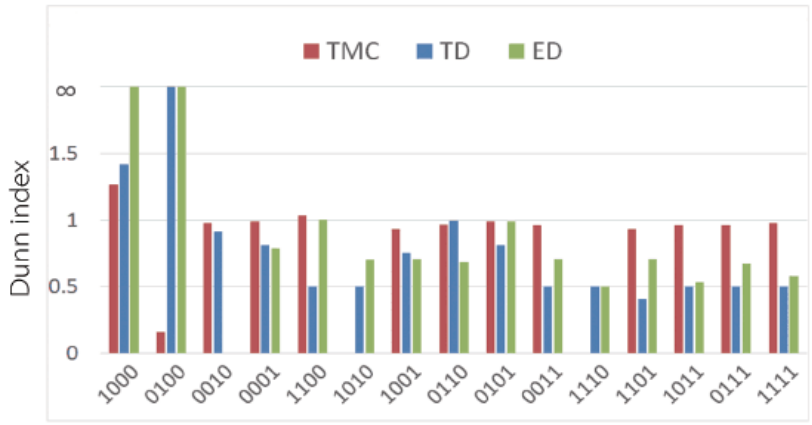
图 3.14 不同对象数的 JI 值

图 3.15 为 DI 统计值，用来测量聚类结果的质量，其中横轴上的各种索引号表示特征空间的组合。例如，“1100”的前两个数字为 1，表示选择特征空间交通模式和天气来进行聚类；相反，后面两个数字“00”表示没有选择温度和风速。图中还存在一些异常的 DI 值，这说明 AP 聚类算法的结果存在异常情况，其中出现 0 的情况：有相似度为 0 的对象被聚到了错误的类中，导致两个类簇的类间间距为 0，因而 DI 值等于 0；出现无穷的情况：可能会出现每一个类中都只有一个元素的情况，则最大的类内间距就是 0，导致 DI 值等于无穷；除上述情况外，代表聚类结果好。由前两个直方图可以看出，TMC 的 DI 值大多大于 TD 和 ED，这说明 TMC 在对象数量为 40 和 150 时可以产生更高质量的聚类。然而，随着对象数量增加到 800，三种方法的 DI 值都略有下降，但 TMC 的 DI 值仍相对较高。产生这种结果有两个原因：一方面，可选择加强张量距离反映了高阶空间对象不同坐标之间的内在关系，而欧式距离不能利用这种相互影响；另一方面，通过添加权重因子，在可选择加权张量距离中考虑了不同属性组合的贡献，提高了聚类质量，而在张量距离中它们被认为是同样重要。

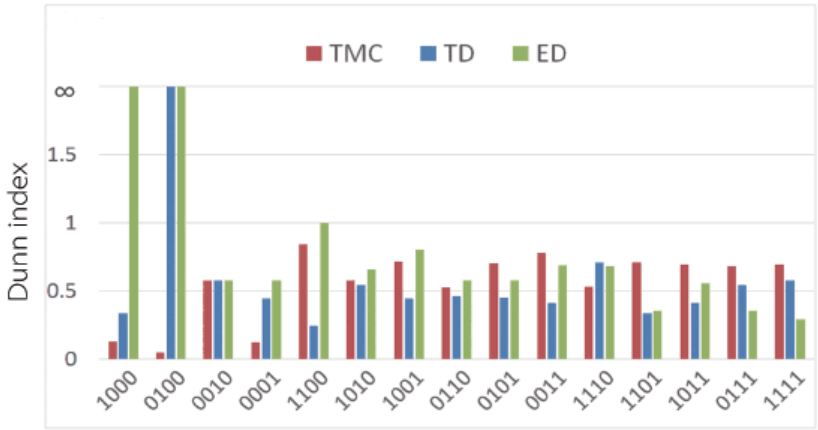
因此，基于张量的多聚类方法可以根据不同的需求灵活选择特征空间并发现高质量聚类结果的同时，保证了低冗余性。它结合了三种新颖的思想来进行大数据环境下的多聚类：第一，通过对象张量表示模型来融合多源信息空间；第二，根据大数据的不同应用，通过可选择加权张量距离灵活地度量高阶空间对象张量之间的真实距离；第三，通过多线性属性组合权重学习方法得到的权重张量来降低噪声属性的影响。



(a) 对象数40



(b) 对象数150



(c) 对象数800

图 3.15 不同对象数的 DI 值

3.4.2 基于张量分解的多聚类方法性能分析

本实验将提出的基于张量分解的多聚类方法应用于电子商务系统，首先介绍评价方法和数据集，最后给出实验结果和分析。本实验分别对提出的三种多聚类方法进行对比，归一化参数σ和素性修正参数$\alpha_1,\alpha_2,\ldots,\alpha_k$分别设置为 0.2 和 0.3。

3.4.2.1 评价方法

同 3.4.1.2，本节仍使用 JI 值衡量聚类结果之间的冗余度，DI 值衡量聚类质量，在此不再赘述。

3.4.2.2 数据集

电子商务近年来发展迅速，以 2015 年 11 月 11 日的天猫全球购物节为例，商品交易总额（GMV）为 912.1 亿元，超过了 2014 年中国社会消费品零售总额的日均水平。与传统商业相比，电子商务具有购物方便、成本低廉等明显优势。此外，在电子商务中，从用户的位置信息、搜索习惯、行为模式等都可以获得更丰富的背景数据。通过利用这些海量数据，电商服务商可以根据用户的不同需求对商品进行聚类，从而进行个性化推荐。

本文使用真实的电子商务数据集进行实验，数据集详细信息见表 3.2 和表 3.3。该数据集来自 2015 年天猫数据集 [2]。

表 3.2 用户行为日志

数据域	描述
用户_*id*	用户唯一 ID
商品_*id*	商品 ID 号
分类_*id*	商品所属分类 ID
卖家_*id*	卖家 ID
品牌_*id*	品牌 ID
时间	用户交易时间
行为类型	用户交易类型

表 3.3 用户信息日志

数据域	描述
用户_*id*	用户唯一 ID
年龄范围	用户年龄范围： -1 <18:-2 for [18-24]； -3 for [25,29]; -4 for [30,34] -5 for [35-29];-6 for [40,49] -7 and 8 for >=50; -And NULL for unknown;
性别	-用户性别 -0 表示女性:-1 表示男性; --2 NULL 表示未知

数据集由两部分组成：第一部分包含 1048576 个用户行为日志，数据格式为：（用

[2] http://dblab.xmu.edu.cn/blog/2335/

户_*id*、商品_*id*、分类_*id*、卖家_*id*、品牌_*id*、时间、动作类型)，其中时间来自物理空间，其余特征来自网络空间；第二部分包含来自社交空间的 424171 个用户信息日志，数据格式为：（用户_*id*、年龄范围、性别）。

3.4.2.3 聚类质量和冗余度

首先对原始数据进行量化、清洗和组合等预处理，以商品 ID、商品类别 ID、商家 ID、商品品牌 ID、商品目标人群年龄范围为主键将两部分数据转化为一个数据集。为了简单起见，本文将得到的数据集记为 U，类别集记为 C，商家集记为 S，品牌集记为 B，年龄范围集记为 A。

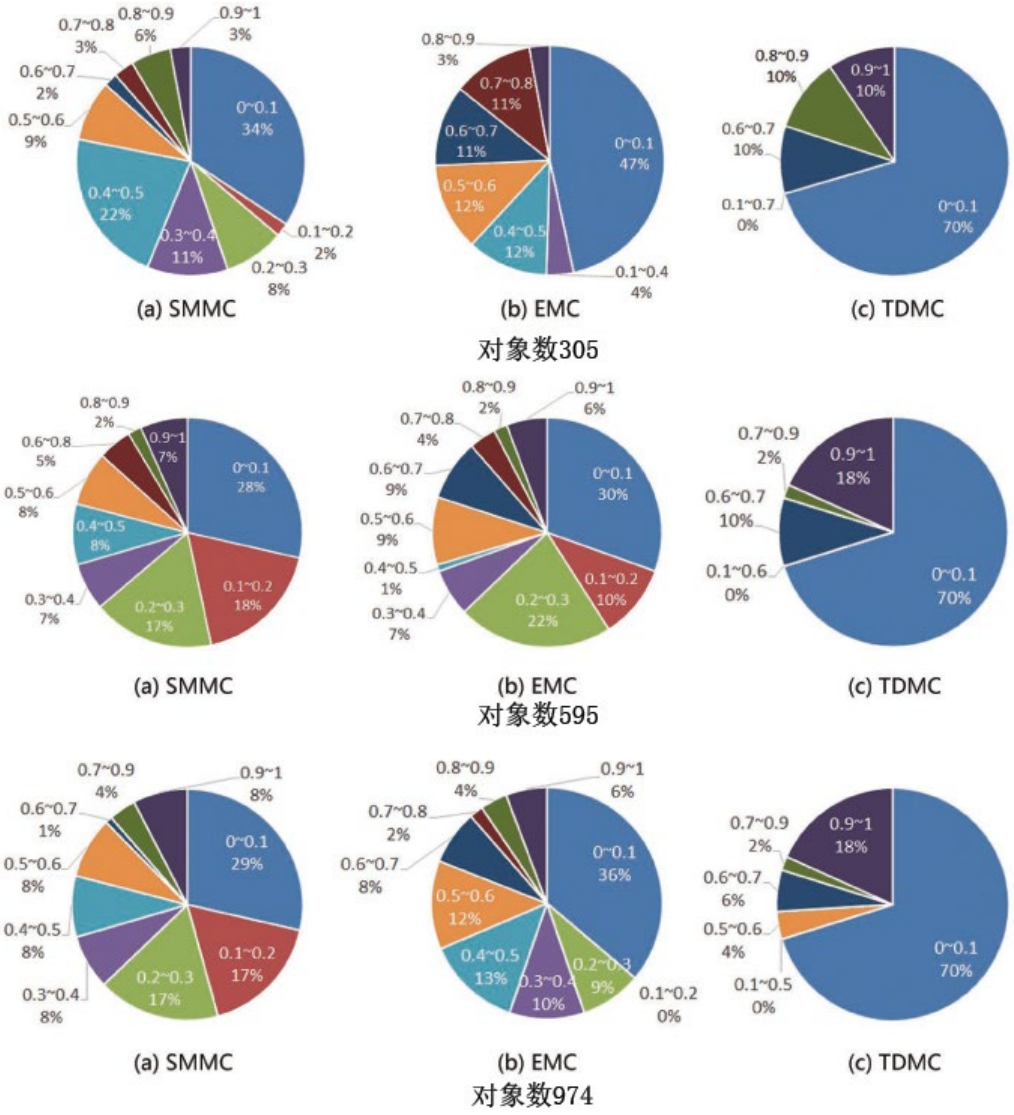


图 3.16 不同对象数的 JI 值

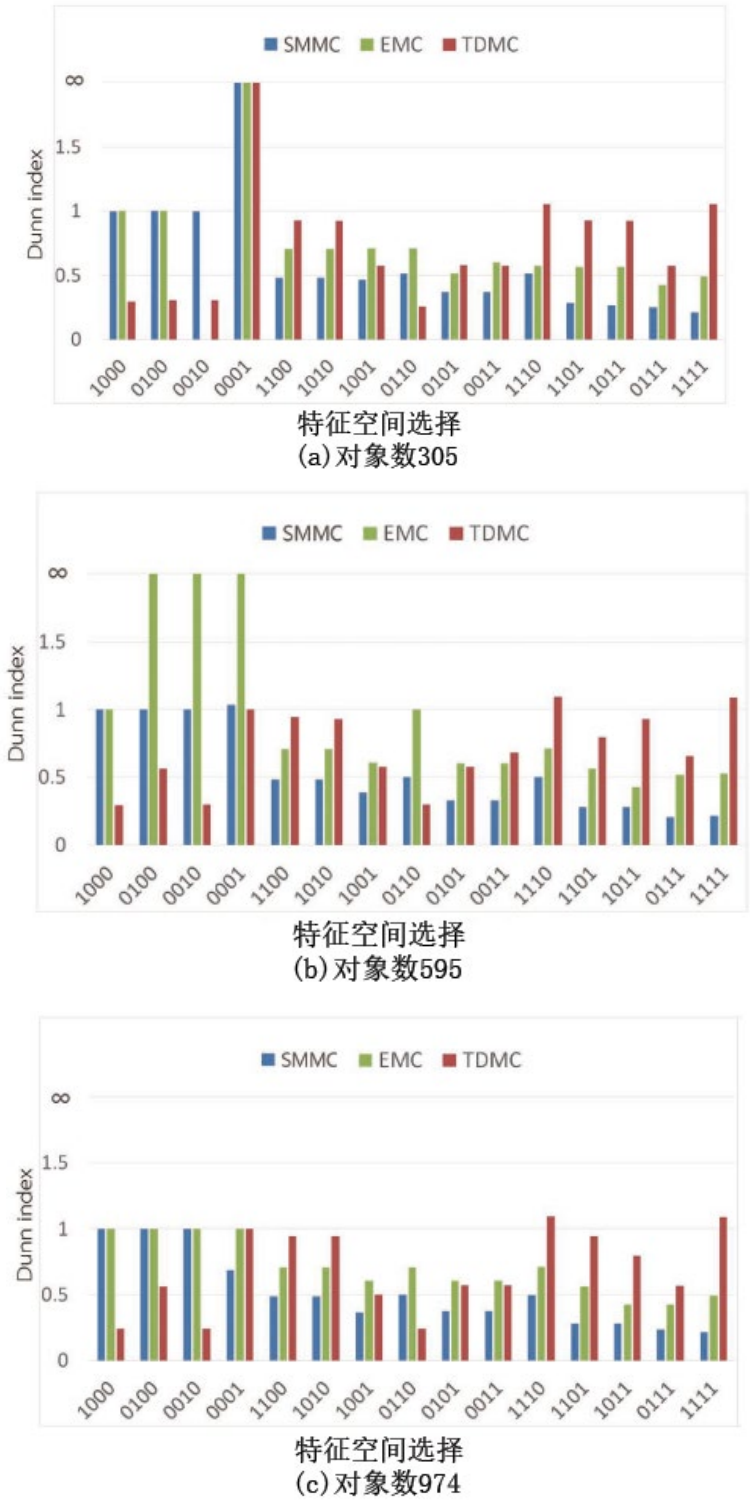


图 3.17 不同对象数的 DI 值

在此从 U 中选取一些样本子集进行实验，其详细过程如下：(1)从 C 中选择 c 个类别，从 S 中选择 s 个商家，从 B 中选择 b 个品牌，从 A 中选择 a 个年龄范围；

(2)找出所有这些属性的项；(3)增加 c，s，b 和 a，然后重做(2)，以获得不同对象数数据子集。

图 3.16 是 JI 值的统计数据，从图中可以看出 TDMC 的冗余度是最好的。在三个子图中，TDMC 的 JI 在 0-0.1 之间的值占到 70%以上，均高于 SMMC 和 EMC。此外，随着对象数量的增加，这一比例几乎没有变化。结果表明，TDMC 能够发现更多不同的隐藏数据模式，更适合根据用户的不同需求进行个性化推荐。

图 3.17 通过 DI 对聚类结果进行评价。当只选择一个特征空间时，TDMC 的性能较差，但是如果选择多个特征空间，TDMC 的性能最好，EMC 的性能优于 SMMC。如前所述，SMMC 可以在不同的特征空间中利用不同的相似性度量，但缺乏对特征空间融合的考虑；EMC 将信息融合考虑在内，在多个特征空间中实现了更好的聚类质量；TDMC 通过将对象数据集成到张量中，利用 WTD 计算相似度，更加细致地考虑了相互影响，反映了高阶数据对象不同坐标之间的内在关系。因此，TDMC 具有良好的性能，特别是在选择多个特征空间时表现较好。

综上所述，本文提出的方法可以根据不同的需求灵活选择特征空间以发现高质量的聚类结果，同时保证低冗余。具体来说，可通过在物理、网络、社会空间中集成多源信息，允许用户根据应用的需求选择不同的特征空间组合，确保得到的多个聚类结果满足多个分析任务的需求。同时，本文的方法可以保证聚类结果的质量和差异性。在三种方法的效率方面，SMMC 和 EMC 优于 TDMC，但与聚类的质量相反。通常，在大多数应用中，用户对聚类的质量更感兴趣；因此若用户更关心效率，SMMC 和 EMC 均可作为首选方法。

3.5 本章小结

本章旨在对大规模多源异构数据进行灵活地聚类并高效地产生多个聚类结果，为大数据应用提供有意义的服务。首先提出了一种基于多线性属性排名的权重学习方法，用来度量所有特征空间中属性组合的重要性。研究了基于不同应用的数据对象聚类，并总结了TMC算法。其次，又介绍了两种简单的多聚类方法，一种是基于所选特征空间相似度矩阵加权平均的SMMC方法，另一种是利用SWED计算相似度的EMC方

法。然而，它们没有考虑不同属性的影响，也没有考虑如何去除噪声和冗余，特别是对于高维数据。因此，本文进一步扩展了SMMC和EMC的优势，提出了以HOOI和WTD进行相似性度量的TDMC方法。同时，为了提高TDMC的性能，研究了一种用于各特征空间属性重要性度量的多关系属性排名方法。最后本章分别通过一个自行车共享系统和一个个性化推荐系统对提出的方法进行评价。实验结果表明，所提出的多聚类方法，特别是TMC和TDMC方法，能够获得较高质量多聚类结果，同时冗余度较低，能够满足大数据应用的不同需求，可为大数据分析和应用提供增强的知识提取和服务。

4 云端安全的张量多聚类方法

随着云计算日益普及，拥有强大的计算资源和数据存储的云可以实时处理和分析数据，也可以通过软件自动管理数据。此外，由于云集成了大量相关数据和先进的数据挖掘技术，整合相关信息和数据分析算法，可为企业提供更准确的信息和更智能的服务，所以越来越多的企业为了节约成本而将内部数据外包到云端进行计算。而在大数据环境下，随着数据量的增长和数据规模的不断增大，有限的计算和存储资源很难实现实时的多聚类分析，因此借助强大的云计算能力来进行基于张量的多聚类可以有效提高其计算效率。然而将聚类算法部署在云平台为用户提供服务时，用户直接将数据外包给云平台来进行多聚类会暴露用户的敏感或隐私信息，因为云服务提供商可能是好奇或者恶意的。而信息一旦泄露，就可能危及企业的安全生产，甚至危及人民群众的生命安全。因此，本章利用同态加密理论并结合先进的混合云思想，首先研究一种可以作为安全的张量多聚类计算的基础协议，即混合云模型下的安全高阶密度峰值聚类协议；进而研究一种隐私保护的张量多聚类方法，从而实现云端安全地大数据多模态聚类。

4.1 问题定义

近年来，云计算安全受到了学术界和业界的广泛关注，出现了大量的隐私保护方法。隐私保护的聚类方法有两种重要的技术：扰动法和加密法。前者利用数据失真技术满足聚类分析隐私保护的要求[36,92]；后者是利用密码学方法对聚类[93]进行隐私保护。加密方法不仅可以提供形式化的隐私保证，而且在准确性上优于扰动法。在使用加密方法时，一种有效的做法是将数据外包给云端之前对其进行加密，然后在云端对加密的数据执行所有计算，直到加密的最终结果返回给客户端进行解密。在这个过程中，云端不会学习到任何敏感数据和中间结果，从而保障了用户隐私的安全。然而，这却给在加密数据上实现安全的张量多聚类带来一些新的问题和挑战。本文主要考虑四个方面：第一，为了保护用户的隐私和所有中间结果，与多聚类相关的各种安全运

算是必不可少的，包括加法、乘法、除法、比较、取幂等，但是现有的加密方法只提供了有限的安全运算；第二，为保证聚类结果的准确性，在密文上计算的对象张量距离与明文的精度要尽可能相同，需要高效的同态密码系统和浮点数的处理，但是现有的方法中两者往往不能同时兼顾；第三，在云端聚类时应尽可能降低客户端的计算成本，但现有加密方法往往需要客户端进行解密操作，或者协助在明文上执行一些没有实现的安全运算；第四，为了提高聚类算法的效率和可扩展性，需要合理利用云资源，以满足计算成本高、数据量不断增长的需求。

因此，如何利用加密的方法实现云端安全、高效、准确的张量多聚类，同时尽可能减轻客户端的成本，是本章需要解决的问题。具体来说，第一，如何保证张量多聚类计算的安全性，即云端在不暴露任何用户隐私和中间结果的情况下，安全地实现完整的多聚类过程；第二，如何保证聚类结果的准确性，重点是如何安全高效地处理各种相关的浮点运算；第三，如何使客户端的运算成本达到最低；第四，如何提高云端安全张量多聚类算法的效率和可扩展性。

4.2 云端安全的高阶密度峰值聚类方法

作为张量多聚类的基础算法，本文提出一种高阶密度峰值聚类的隐私保护实现方法。本节首先介绍经典的密度峰值聚类算法，然后描述提出的云端安全聚类分析框架，进而提出一系列相关的基本安全协议，最后给出云端安全的高阶密度峰值聚类方法。

4.2.1 云端安全聚类分析框架

本小节主要介绍安全高阶密度峰值聚类方法的安全模型和安全分析框架。

4.2.1.1 安全模型

为了最大限度地保护用户隐私，该方案在安全多方计算中采用了半诚实模型。半诚实模型[94]假设任何参与方以其正确的输入忠实地执行协议，但在协议执行过程中，各参与方总是试图推断出其他参与方的机密信息，其中各参与方指的是公有云和私有云。公有云和私有云可以构成一个混合云[95]，它既具有公有云计算能力，又具有私有云安全性，图 4.1 给出了混合云模型的示意图，混合云之间以及与客户端的具体交互

过程将在下面的安全分析框架中进行详细介绍。在实际应用中，公有云提供商通常是像 Microsoft 和谷歌这样成熟的 IT 公司，而私有云提供商通常是政府监管下的特殊机构。一般来说，考虑到名誉、商业利益和承担的法律风险，他们之间不会相互串通，更不会恶意窃取用户隐私信息，所以混合云这个假设是合理的。

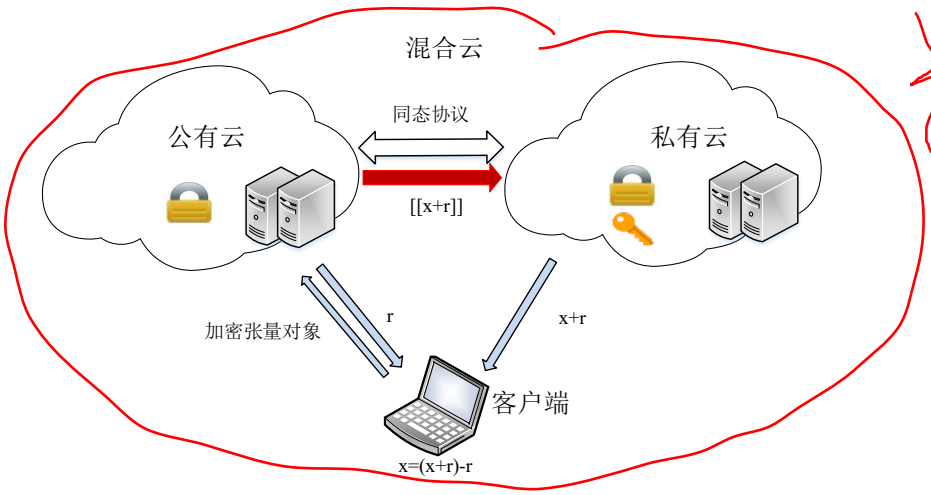


图 4.1 混合云安全模型

如图 4.1 所示，就公有云和私有云的关系而言，在混合云模型上的每个安全协议都需要这两方一起工作。这些协议执行的主要思想是：公有云通过同态加法对密文进行扰动并将其发送到私有云；然后私有云解密受扰动的密文，对其进行特定操作，再次加密结果并将其发送回公有云；最后，公有云通过同态加法去除加密结果的扰动。此过程中私有云虽然可以解密，但是只能得到受扰动的明文；而公有云没有私钥，只能获取无法解密的密文。因此，整体来说混合云模型下执行安全协议是不会泄露用户隐私的。

4.2.1.2 安全聚类分析和服务框架

图 4.2 概述了本文提出的一种安全聚类分析与服务框架，包括数据外包、安全分析、聚类服务共三层，各层的功能描述如下：

（1）数据外包层：开始私有云使用 Paillier 加密体系生成公钥 *pk* 和私钥 *sk*，并将 *pk* 发布给客户端和公有云。客户端使用 *pk* 加密对象张量，然后将密文对象张量 $[\![\mathcal{X}_1]\!],[\![\mathcal{X}_2]\!],\ldots,[\![\mathcal{X}_n]\!]$ 发送到公有云，此后不参与任何聚类计算。

（2）安全分析层：利用所提出的安全协议，在加密的对象张量上，公有云与私

有云协同完成整个聚类过程。在公有云上生成最终的加密聚类结果$[\![cl]\!]$，再通过加随机数 r 得到扰动的密文结果$[\![cl+r]\!]$并发送到私有云，最后私有云解密发送给客户端。

（3）聚类服务层：客户端从公有云接收到 r 后，从私有云接收到扰动的聚类结果 $cl+r$，通过做明文减法运算即可得到明文聚类结果 cl。例如在工业物联网中，安全高阶密度峰值聚类可以在设计、调度、预测维护等方面提供有关安全聚类服务。

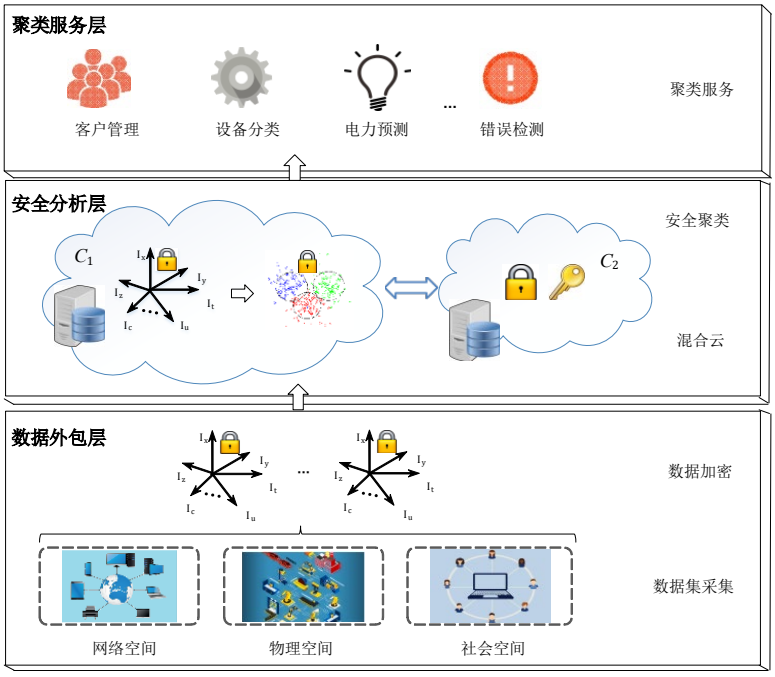


图 4.2 安全聚类分析和服务框架

在该框架中，各方能看到的中间结果只是加密数据或者扰动的明文，却无法获取任何其他敏感信息，因此运用该框架可以保护用户的隐私。此外，对于返回的聚类结果，客户端只需要执行明文减法操作，因而是非常轻量级的。下面将重点讨论安全分析层中所提出的安全高阶密度峰值聚类协议及其相关安全子协议。

4.2.2 基本安全协议

本节首先介绍相关的一些现有安全协议；然后给出一组通用协议作为构造所提出的安全高阶密度峰值聚类方法的子协议。这里，本文假设所有的安全协议都符合上述安全模型和框架，其中 C_1 和 C_2 分别表示公有云和私有云。

4.2.2.1 现有安全协议

在安全的高阶密度峰值聚类中，涉及到的现有安全协议描述如下：

（1）安全乘法协议（Secure Multiplicatin Protocol，SM）

在该协议中，公有云C_1输入隐私数据$[\![a]\!]$和$[\![b]\!]$，私有云C_2拥有私钥sk，C_1和C_2在不泄露a，b或中间结果的情况下，交互协作的计算输出结果$[\![a*b]\!]$且该结果只有C_1知道。协议的细节详见[96]，该协议可以形式化表示为

$$[\![a*b]\!] \leftarrow SM\left([\![a]\!],[\![b]\!]\right). \tag{4.1}$$

（2）安全比较协议（Secure Comparison Protocol，SC）

在该协议中，公有云C_1输入隐私数据$[\![a]\!]$和$[\![b]\!]$，私有云C_2拥有私钥sk，C_1和C_2在不泄露a，b和中间结果的情况下，交互协作的比较a和b的大小，如果$a \geqslant b$，输出是$[\![1]\!]$；否则，输出是$[\![0]\!]$。输出结果只有C_1知道。协议的细节详见[97]，该协议可以形式化表示为

$$[\![a \geq b]\!] \leftarrow SC\left([\![a]\!],[\![b]\!]\right). \tag{4.2}$$

（3）安全相等协议（Secure Equation Protocol，SEQ）

在该协议中，公有云C_1输入隐私数据$[\![a]\!]$和$[\![b]\!]$，私有云C_2拥有私钥sk，C_1和C_2在不泄露a，b和中间结果的情况下，交互协作的比较a和b是否相等，如果$a=b$，输出是$[\![1]\!]$；否则，输出是$[\![0]\!]$。输出结果只有C_1知道。协议的细节详见[98]，该协议可以形式化表示为

$$[\![a = b]\!] \leftarrow SEQ\left([\![a]\!],[\![b]\!]\right). \tag{4.3}$$

（4）安全除法协议1（Secure Division Protocol，SD1）

在该协议中，公有云C_1输入隐私数据$[\![a]\!]$，明文数据b，其中$a/b = q.r$，a、b、q和r分别为被除数、除数、商和余数，私有云C_2拥有私钥sk。C_1和C_2在不泄露a，b和中间结果的情况下，交互协作的计算出商$[\![q]\!]$，且该输出结果只有C_1知道。协议的细节详见[99]，该协议可以形式化表示为

$$[\![q]\!] \leftarrow SD1\left([\![a]\!], b\right). \tag{4.4}$$

（5）安全除法协议2（Secure Division Protocol，SD2）

在该协议中，公有云C_1输入隐私数据$[\![a]\!]$和$[\![b]\!]$，其中$a/b=q.r$，a、b、q和r分别为被除数、除数、商和余数，私有云C_2拥有私钥sk。C_1和C_2在不泄露a，b和中间结果的情况下，交互协作的计算出商$[\![q]\!]$，且该输出结果只有C_1知道。协议的细节详见[100]，该协议可形式化表示为

$$[\![q]\!] \leftarrow SD2\left([\![a]\!], [\![b]\!]\right). \tag{4.5}$$

在上述两个安全除法协议中，由于 SD1 效率远高于 SD2，但是只有 SD2 能够保护除数，所以将视情况分别使用这两种协议来实现安全的除法。

4.2.2.2 提出的安全协议

为了实现完整的安全高阶密度峰值聚类，本文研究一组相关的通用协议，并在这些协议中采用扩倍和缩倍因子来控制浮点数精度。

（1）安全平方张量距离（Secure Squared Tensor Distance，SSTD）

在该协议中，公有云C_1输入加密的隐私对象张量$[\![\lambda_o\mathcal{X}]\!]$和$[\![\lambda_o\mathcal{Y}]\!]$以及明文度量矩阵$G$，$C_2$持有私钥$sk$，这里$\mathcal{X}$，$\mathcal{Y}\in\Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$，其中$I_{f_1},I_{f_2},\ldots,I_{f_k}$分别对应由不同属性描述的$k$个特征空间，$\lambda_o$是扩倍因子。该协议的目标是：$C_1$和$C_2$在不泄露$\lambda_o\mathcal{X}$、$\lambda_o\mathcal{Y}$和中间结果的情况下，协同计算出$[\![\lambda_o d_{SSTD}]\!]$且该输出结果只有$C_1$知道。因为度量对象之间的距离等于度量距离的平方，它们都保持相对顺序，所以为了提升效率，在这里计算距离的平方。安全平方张量距离协议的基本思想描述如下：

算法 4.1：安全平方张量距离协议

输入：云C_1拥有加密的对象张量$[\![\lambda_o\mathcal{X}]\!]$和$[\![\lambda_o\mathcal{Y}]\!]$，明文度量矩阵$G$，云$C_2$拥有私钥$sk$。

输出：加密平方张量距离$[\![\lambda_o d_{SSTD}]\!]$，该结果只有云$C_1$知道。

1：云C_1,C_2交互执行下列步骤：

2：for l=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

3： for m=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

4： 利用同态减法安全计算(x_l-y_l)，$[\![\lambda_o(x_l-y_l)]\!]\leftarrow[\![\lambda_o x_l]\!]*[\![\lambda_o y_l]\!]^{N-1}$；

5： 利用同态乘法安全计算$g_{lm}(x_l-y_l)$，$[\![\lambda_o^2 d_{lm}]\!]\leftarrow([\![\lambda_o(x_l-y_l)]\!])^{\lambda_o g_{lm}}$；

6： 利用 SD1 进行缩倍，$[\![\lambda_o d_{lm}]\!]\leftarrow SD1([\![\lambda_o^2 d_{lm}]\!],\lambda_o)$；

7： 利用同态减法安全计算(x_m-y_m)，$[\![\lambda_o(x_m-y_m)]\!]\leftarrow[\![\lambda_o x_m]\!]*[\![\lambda_o y_m]\!]^{N-1}$；

8： 利用同态乘法安全计算$g_{lm}(x_l-y_l)$，$[\![\lambda_o^2 d_{lm}]\!]\leftarrow([\![\lambda_o(x_m-y_m)]\!])^{\lambda_o d_{lm}}$；

9： 利用 SD1 进行缩倍，$[\![\lambda_o d_{lm}]\!]\leftarrow SD1([\![\lambda_o^2 d_{lm}]\!],\lambda_o)$；

10： end for

11：end for

12：云C_1利用同态加法累加所有的$[\![\lambda_o d_{lm}]\!]$，得到加密的平方张量距离，

$$[\![\lambda_o d_{SSTD}]\!]\leftarrow\prod_{l,m=1}^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}[\![\lambda_o d_{lm}]\!]。$$

（2）安全排序Top*k*（Secure Soat Top*k*，SSOAT*k*）

在该协议中，公有云C_1拥有加密的隐私数组$[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_n]\!]$以及明文整数$k$，$C_2$持有私钥$sk$。该协议的目标是：$C_1$和$C_2$在不泄露$d_1,d_2,\ldots,d_n$和中间结果的情况下，交互协作的按升序排序Top$k$个数据并记录其对应在原始数组中的下标数组，且该输出结果只有C_1知道。安全排序Top$k$协议的基本思想是k趟冒泡排序，具体描述如下：

算法 4.2：安全排序 Top$k$ 协议

输入：云C_1拥有加密的数组$\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_n]\!]\}$，明文整数 k，云C_2拥有私钥 sk。

输出：加密的升序排列 Top$k$ 数组$\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_k]\!]\}$

和下标数组$\{[\![h_1]\!],[\![h_2]\!],\ldots,[\![h_k]\!]\}$，该结果只有云$C_1$知道。

1：云C_1执行下列步骤：

2：for i=1 to n do

3： 初始化下标数组，$[\![h_i]\!] \leftarrow [\![i]\!]$；

4：end for

5：云C_1，C_2交互执行下列协议：

6：for i=1 to k do

7： for j=n downto i+1 do

8： 利用 SC 对数组中从后向前相邻两个数进行比较，$[\![c]\!] \leftarrow SC([\![d_j]\!],[\![d_{j-1}]\!])$；

9： 记录d_j到临时变量，$[\![d_{temp}]\!] \leftarrow [\![d_j]\!]$；

10： 根据第 8 步比较结果，如果d_j小于d_{j-1}，将d_{j-1}赋给d_j，否则值不变，

$$[\![d_j]\!] \leftarrow SM\left([\![c]\!],[\![d_j]\!]\right) * SM\left([\![1]\!] * [\![c]\!]^{N-1},[\![d_{j-1}]\!]\right);$$

11： 根据第 8 步比较结果，如果d_j小于d_{j-1}，将d_{temp}赋给d_{j-1}，否则值不变，

$$[\![d_{j-1}]\!] \leftarrow SM\left([\![c]\!],[\![d_{j-1}]\!]\right) * SM\left([\![1]\!] * [\![c]\!]^{N-1},[\![d_{temp}]\!]\right);$$

12： 记录d_j的下标到临时变量，$[\![h_{temp}]\!] \leftarrow [\![h_j]\!]$；

13： 根据第 8 步比较结果，如果d_j小于d_{j-1}，将d_{j-1}的下标赋给d_j的下标，否则值不变，$[\![h_j]\!] \leftarrow SM\left([\![c]\!],[\![h_j]\!]\right) * SM\left([\![1]\!] * [\![c]\!]^{N-1},[\![h_{j-1}]\!]\right)$；

14： 根据第 8 步比较结果，如果d_j小于d_{j-1}，将d_{temp}赋给d_{j-1}的下标，否则值不变，$[\![h_{j-1}]\!] \leftarrow SM\left([\![c]\!],[\![h_{j-1}]\!]\right) * SM\left([\![1]\!] * [\![c]\!]^{N-1},[\![h_{temp}]\!]\right)$；

15： end for

16：end for

对安全排序Top$k$协议的输出进行修改，结果只有$[\![d_k]\!]$和$[\![h_k]\!]$，则SSOATk协议就是求第k小的值，表示为$\mathrm{SMIN}_{k\mathrm{th}}(\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_n]\!]\},k)$。特别地，当$k$=1时，$\mathrm{SMIN}_{k\mathrm{th}}$就是最小值协议，表示为$\mathrm{SMIN}_n(\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_n]\!]\})$；而当$k=n$时，$\mathrm{SMIN}_{k\mathrm{th}}$就是最大值协议，表示为$\mathrm{SMAX}_n(\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_n]\!]\})$。

4.2.3 安全的高阶密度峰值聚类

根据前面的基本安全协议和同态加密理论，本节描述提出的云端安全高阶密度峰值聚类协议。

安全高阶密度峰值聚类（Secure HOCFS，SHOCFS）：公有云C_1输入加密的对象张量$[\![\mathcal{X}_1]\!],[\![\mathcal{X}_2]\!],\ldots,[\![\mathcal{X}_n]\!]$，$C_2$持有私钥$sk$。该协议的目标是：$C_1$和$C_2$在不泄露$\mathcal{X}_1,\mathcal{X}_2,\ldots,\mathcal{X}_n$和中间结果的情况下，协同计算出聚类结果，且该输出结果只有C_1知道。安全排序Top$k$协议的基本思想是k次冒泡排序，具体描述如下：

算法 4.3：安全高阶密度峰值聚类协议

输入：云C_1拥有加密的对象张量$[\![\mathcal{X}_1]\!],[\![\mathcal{X}_2]\!],\ldots,[\![\mathcal{X}_n]\!]$，云$C_2$拥有私钥$sk$。

输出：加密的聚类结果$[\![cl]\!]$，该结果只有云C_1知道。

1：云C_1，C_2交互执行下列步骤：

2：for i=1 to n do

3：　for j=i+1 to n do

4：　　安全计算对象之间的平方张量距离，$[\![d_{ij}]\!] \leftarrow SSTD(\mathcal{X}_i,\mathcal{X}_j)$；

5：　end for

6：end for

7：计算用于确定截断距离的参数 k，$k \leftarrow \lceil factor*(n(n-1)/2) \rceil$；

8：展开距离矩阵$[\![d_{ij}]\!]$为一维数组；

9：安全的计算截断距离，$[\![d_c]\!] \leftarrow SMIN_{kth}(\{[\![d_1]\!],[\![d_2]\!],\ldots,[\![d_{n(n-1)/2}]\!]\},k)$；

10：for i=1 to n do

11：　for j=1 to n do

12：　　利用 SC 和同态加安全计算第 i 个对象的局部密度，

$$[\![\rho_i]\!] \leftarrow \prod_{j=1}^{n}([\![1]\!]*SC([\![d_{ij}]\!],[\![d_c]\!])^{N-1})；$$

13：　end for

14：end for

15：for i=1 to n do

16：　安全的求出第 i 个对象到其他对象距离的最大值，

$$[\![a_i]\!] \leftarrow SMAX_n([\![d_{i1}]\!],[\![d_{i2}]\!],\ldots,[\![d_{in}]\!])；$$

17：　for j=1 to n do

18：　　利用 SC 对第 i 个对象和第 j 个对象的局部密度进行安全比较，

$$[\![c_j]\!] \leftarrow SC([\![\rho_i]\!],[\![\rho_j]\!])；$$

19：　　如果比较的第 j 个对象比第 i 个对象的局部密度大，则将其之间的距离 d_{ij} 记录下来，否则赋值第 i 个对象到其他对象距离，

$$[\![g_j]\!] \leftarrow SM([\![1]\!]*[\![c_j]\!]^{N-1},[\![d_{ij}]\!])*SM([\![c_j]\!],[\![a_i]\!])；$$

20：　end for

21：　在所有密度大于第 i 个对象的距离中安全的找到最小距离及其下标

$$([\![\delta_i]\!],[\![cl_i]\!]) \leftarrow SMIN_n([\![g_1]\!],[\![g_2]\!],\ldots,[\![g_n]\!])；$$

21：　利用 SM 安全计算第 i 个对象的 γ 值，$[\![\gamma_i]\!] \leftarrow SM([\![\rho_i]\!],[\![\delta_i]\!])$；

22：end for

23：对所有的 γ 值安全的按升序排序，

$$([\![\gamma_1']\!],[\![\gamma_2']\!],\ldots,[\![\gamma_n']\!]) \leftarrow SSOAT_k(\{[\![\gamma_1]\!],[\![\gamma_2]\!],\ldots,[\![\gamma_n]\!]\},n)；$$

24：初始化记录聚类中心标记的数组，$[\![e_n]\!] \leftarrow [\![1]\!]$；

25：for i=n-1 downto 1 do

26：　利用 SD2 安全的计算每个 γ 值的斜率，

$$[\![t]\!] \leftarrow SD2([\![\gamma_{i+1}']\!]*[\![\gamma_i']\!]^{N-1},[\![\gamma_i']\!]*[\![\gamma_{i-1}']\!]^{N-1})；$$

27：　如果给定的 γ 的斜率小于该对象点的斜率，则标记为 1，

$$[\![e_i]\!] \leftarrow SM(SC([\![t]\!],[\![slope]\!]),[\![e_{i+1}]\!])；$$

28：end for

29：for i=n downto 1 do

30：　for j=1 to n do

31： 利用 SEQ 判断第 j 个对象是否为中心，如果是聚类中心则利用 SM 标记为 1，否则为 0，$[\![c]\!] \leftarrow SM(SEQ([\![\gamma_i']\!],[\![\gamma_j]\!]),[\![e_i]\!])$；

32： 如果第 j 个对象是聚类中心，则标记为 n+1；否则，值保持不变，$[\![cl_j]\!] \leftarrow SM([\![c]\!],[\![n+1]\!]) * SM([\![1]\!] * [\![c]\!]^{N-1},[\![cl_j]\!])$。

33： end for

34：end for

4.3 云端安全的张量多聚类方法

在基于张量的多聚类方法基础上，结合同态加密技术，本文研究一种在混合云模型上的张量多聚类的隐私保护实现方法。本节首先描述云端安全的张量多聚类分析框架，进而提出一系列相关的基本安全协议，最后给出云端安全的张量多聚类方法。

4.3.1 云端安全多聚类分析框架

本小节主要介绍安全的张量多聚类方法的安全模型和安全的多聚类分析框架。

4.3.1.1 安全模型

和 4.2.1.1 介绍的安全模型相同，仍使用基于半诚实模型的混合云安全模型，其定义、组成和云的关系在此不再赘述。

4.3.1.2 安全多聚类分析和服务框架

图 4.3 描述了一种安全多聚类分析与服务框架，包括数据外包、安全分析、聚类服务共三层，各层的功能描述如下：

（1）数据外包层：开始私有云使用 Paillier 加密体系生成公钥 *pk* 和私钥 *sk*，并将 *pk* 发布给客户端和公有云，客户端使用 *pk* 加密对象张量，然后将密文对象张量 $[\![\mathcal{X}_1]\!],[\![\mathcal{X}_2]\!],\ldots,[\![\mathcal{X}_n]\!]$ 发送到公有云，此后不参与任何聚类计算。

（2）安全分析层：利用所提出的安全协议，在加密的对象张量上，公有云与私有云利用同态加密协同完成整个多聚类过程。在公有云上生成最终加密的多聚类结果 $[\![cl_1]\!],[\![cl_2]\!],\ldots,[\![cl_b]\!]$，然后公有云生成随机数 r 并加到这些结果上，得到扰动的密文多

聚类结果$[\![cl_1+r]\!],[\![cl_2+r]\!],\ldots,[\![cl_b+r]\!]$并发送到私有云，最后由私有云解密再发送给客户端。

（3）聚类服务层：客户端从公有云接收到 r 后，从私有云接收到扰动的多聚类结果$[\![cl_1+r]\!],[\![cl_2+r]\!],\ldots,[\![cl_b+r]\!]$，再通过做明文减法运算即可得到明文多聚类结果$[\![cl_1]\!],[\![cl_2]\!],\ldots,[\![cl_b]\!]$。客户端可以利用这些多聚类结果为大数据多分析任务提供服务，如基因分类、资源推荐、电量预测等。

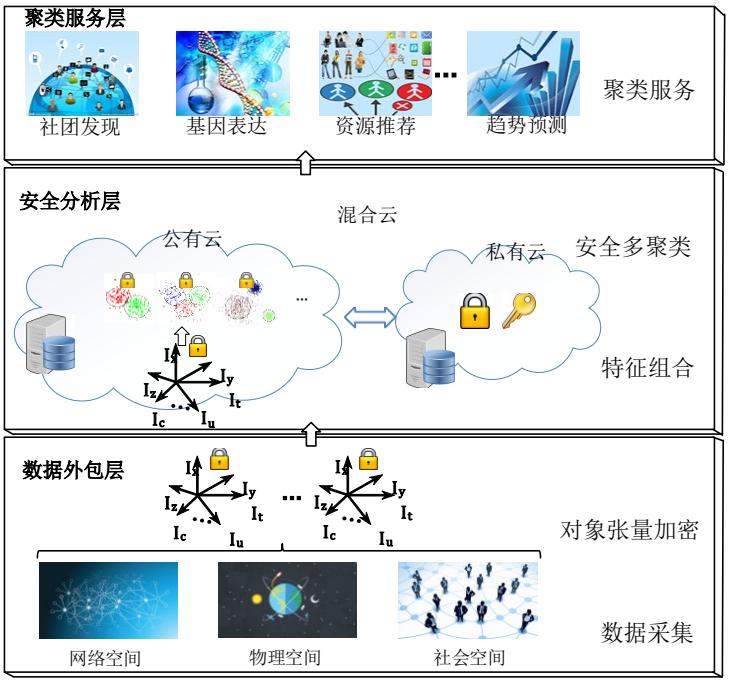


图 4.3 安全多聚类分析和服务框架

该框架与 4.2.2.2 节中介绍的框架的不同在于安全分析层，这里采用的基于张量的多聚类方法。因此，下面将重点研究安全分析层中，安全张量多聚类协议及其相关安全子协议。

4.3.2 基本安全协议

本节首先介绍相关的一些现有安全协议；然后给出一组通用协议作为构造安全的张量多聚类方法的子协议。这里，假设所有的安全协议都符合上述安全模型和框架，其中 C_1 和 C_2 分别表示公有云和私有云。

4.3.2.1 现有安全协议

安全张量多聚类中涉及的现有安全协议，包括安全乘法（SM）、安全比较（SC）、安全除法1（SD1）、安全除法2（SD2）和安全高阶密度峰值聚类（SHOCFS）。

4.3.2.2 提出的安全协议

为了在混合云模型上实现完整的安全张量多聚类过程，本文构建了一组相关的通用协议。在这些协议中同样采用扩倍和缩倍因子来控制浮点数精度。

（1）安全指数（Secure Exponentiation，SE）

在该协议中，公有云C_1输入加密的指数$[\![\lambda_d x]\!]$，私有云C_2持有私钥sk。该协议的目标是：C_1和C_2在不泄露$[\![\lambda_d x]\!]$和中间结果的情况下，交互协作的计算加密的指数结果$[\![\lambda_e e^x]\!]$，且该结果只有C_1知道。

由于Paillier加密体系没有同态指数运算，无法直接支持对象间可选择加权张量距离的安全计算。目前常用泰勒级数展开法将指数运算转化为多项式函数，仅涉及到加法和乘法运算[101]，然而，该方法有两个主要的局限性。首先，泰勒展开的低效率不能满足安全计算的要求；其次，需要通过放大来保持浮点精度，而Paillier加密体系受明文大小的限制，很难达到可接受的精度范围。

观察公式(2.6)和(2.7)，指数的记数器为0，-1，-2，-3，…，根据对象张量的大小，一般来说是指数越小对应的指数结果就越小。所以，在可选择加权张量距离中的所有可能的指数是一系列离散值，由于其部分结果非常接近0，故可以用0来代替。针对这些特点，本文提出一种基于判别式方法的安全指数运算协议，算法4.4总结如下

算法 4.4：安全指数协议

输入：云C_1拥有加密的指数$[\![\lambda_d x]\!]$，云C_2拥有私钥 sk。

输出：加密的指数结果$[\![\lambda_e e^x]\!]$，该结果只有云C_1知道。

1：云C_1执行下列步骤：

2：从所有 x 可能的值中选择前 k 个最大值$m_1, m_2, \ldots, m_k$；

3：for i=1 to k do

4：　　对选择的 k 个最大值及其指数值扩倍并加密作为判别式准则，

$[\![\lambda_d m_i]\!]$，$\left[\!\left[\lambda_e e^{m_i}\right]\!\right]$；

5：end for

6：云C_1, C_2安全的交互执行下列步骤：

7：for i=1 to k do

8：　　利用 SC 比较输入的 x 和选择的最大值，$[\![c_i]\!] \leftarrow SC([\![\lambda_d x]\!],[\![\lambda_d m_i]\!])$；

9：　　根据第 8 步判断结果，如果输入的 x 是选择的最大值之一，则记录其对应的指数值，否则为零，$[\![\lambda_e s_i]\!] \leftarrow SM([\![c_i]\!],\left[\!\left[\lambda_e e^{m_i}\right]\!\right])$；

10：end for

11：云C_1用同态加法累加第9步记录的值并返回加密指数结果$\left[\!\left[\lambda_e e^{x}\right]\!\right] \leftarrow \prod_{i=1}^{k} [\![\lambda_e s_i]\!]$。

在算法第2步中，根据张量对象的大小，C_1从所有可能的x值中选择k个最大值，丢弃指数结果非常小的剩余值，由于这些值已经很难通过扩倍来保持它们的精度，因此丢弃这些值并不会影响结果的准确性，反而可以大大提高运算效率。一般情况下，在可选择加权张量距离中选择前三个最大值已经可以达到足够高的准确率。

定理1：提出的SE协议在半诚实模型上是安全的。

证明：第2步中明文的可能的指数值是可以公开的值，因为它们除了表示张量的大小，并不代表任何关于数据的隐私信息。而由于大小实际上并不属于隐私的范畴，因此第2步是安全的。第4，11步使用的是语义安全的Paillier加密体系。此外，第2、4、11步不与C_1和C_2交互。第8步和第9步分别采用SC协议和SM协议，它们都有正式的安全性证明，能够在半诚实模型下保证它们的安全性。因此，提出的SE协议是安全的。

值得一提的是，SE协议不仅适用于可选择加权张量距离的指数计算，还适用于任何具有离散值指数的指数运算。因此，本文提出的SE协议具有一定的通用性。

（2）安全属性权重排名（Secure Attribute Weight Ranking，SAWR）

在该协议中，公有云C_1输入加密的K阶转移张量$[\![\lambda_w \mathcal{T}_{tr}^{(1)}]\!],[\![\lambda_w \mathcal{T}_{tr}^{(2)}]\!],\ldots,[\![\lambda_w \mathcal{T}_{tr}^{(k)}]\!]$，私有云$C_2$持有私钥$sk$。该协议的目标是：$C_1$和$C_2$在不泄露$\lambda_w \mathcal{T}_{tr}^{(1)}$，$\lambda_w \mathcal{T}_{tr}^{(2)},\ldots,\lambda_w \mathcal{T}_{tr}^{(k)}$和

中间结果的情况下，交互协作地计算加密的属性权重排名向量$[\![\lambda_w \mathrm{w}_1]\!],[\![\lambda_w \mathrm{w}_2]\!],\ldots,[\![\lambda_w \mathrm{w}_k]\!]$，且该结果只有$C_1$知道。安全属性权重排名协议的基本思想是基于3.2.2节提出的多线性属性组合权重学习算法，具体描述如下：

算法 4.5：安全属性权重排名协议

输入：公有云C_1输入加密的K阶转移张量$[\![\lambda_w \mathcal{T}_{tr}^{(1)}]\!],[\![\lambda_w \mathcal{T}_{tr}^{(2)}]\!],\ldots,[\![\lambda_w \mathcal{T}_{tr}^{(k)}]\!]$，私有云$C_2$拥有私钥$sk$。

输出：加密的属性权重排名向量$[\![\lambda_w \mathrm{w}_1]\!],[\![\lambda_w \mathrm{w}_2]\!],\ldots,[\![\lambda_w \mathrm{w}_k]\!]$，结果只有云$C_1$知道。

1：云C_1执行下列步骤：

2：设置修正概率参数$0<\alpha<1$，对其扩倍并进行同态加密，$[\![\lambda_w \alpha]\!]$；

3：for l=1 to n do

4：　初始化向量w_0，满足$\sum_{i=1}^{m}[\mathrm{w}_0]_i=1$，对其扩倍并进行同态加密，$[\![\lambda_w \mathrm{w}_0]\!]$；

5：　设置随机向量u，满足$\sum_{i=1}^{m}[\mathrm{u}]_i=1$，对其扩倍并进行同态加密，$[\![\lambda_w \mathrm{u}]\!]$；

6：　初始化$[\![AR]\!]\leftarrow[\![\lambda_w \mathcal{T}_{tr}^{(l)}]\!]$；

7：　云C_1，C_2交互执行下列步骤：

8：　for j =1 to c do

9：　　for i =1 to l-1 do

10：　　安全的计算模 i 积，$[\![AR]\!]\leftarrow[\![AR\times_i(\lambda_w \mathrm{w}_{j-1})]\!]$；

11：　　利用 SD1 缩倍$[\![AR]\!]\leftarrow SD1([\![AR]\!],\lambda_w)$；

12：　　end for

13：　　for i = l+1 to k do

14：　　安全的计算模 i 积，$[\![AR]\!]\leftarrow[\![AR\times_i(\lambda_w \mathrm{w}_{j-1})]\!]$；

15：　　利用 SD1 缩倍，$[\![AR]\!]\leftarrow SD1([\![AR]\!],\lambda_w)$；

16：　　end for

17：　　赋值$[\![\lambda_w \mathrm{w}_j]\!]\leftarrow[\![AR]\!]$；

18： for t=1 to m do

19： 利用 SM 对 w_j 的每个元素安全的乘上修正参数 α，

$$[\![\lambda_w{}^2(\mathrm{w}_j)_t]\!] \leftarrow SM([\![\lambda_w\alpha]\!],[\![\lambda_w(\mathrm{w}_j)_t]\!])$$；

20： 利用 SM 对 u_j 的每个元素安全的乘上修正参数1-α，并同态加上前一步结果，$[\![\lambda_w{}^2(\mathrm{w}_j)_t]\!] \leftarrow [\![\lambda_w{}^2(\mathrm{w}_j)_t]\!] * SM([\![\lambda_w]\!] * [\![\lambda_w\alpha]\!]^{N-1},[\![\lambda_w u_t]\!])$；

21： 利用 SD1 缩倍，$[\![\lambda_w(\mathrm{w}_j)_t]\!] \leftarrow SD1([\![\lambda_w{}^2(\mathrm{w}_j)_t]\!],\lambda_w)$；

22： end for

23： end for

24：云 C_1 截取 $[\![\lambda_w \mathrm{w}_j]\!]$ 的前 I_{f_l} 个元素作为加密的排名向量 $[\![\lambda_w \mathrm{w}_l]\!]$。

25：end for

（3）安全可选择加权张量距离（Secure Selective Weighted Tensor Distance，SSWTD）

在该协议中，公有云 C_1 输入加密的对象张量 $[\![\lambda_o\mathcal{X}]\!],[\![\lambda_o\mathcal{Y}]\!]$，加密的度量矩阵 $[\![\lambda_e G]\!]$，加密的权重张量 $[\![\lambda_w\mathcal{T}_w]\!]$，私有云 C_2 持有私钥*sk*。该协议的目标是：C_1 和 C_2 在不泄露 $\lambda_o\mathcal{X}$、$\lambda_o\mathcal{Y}$、$\lambda_e G$、$\lambda_w\mathcal{T}_w$ 和中间结果的情况下，协同计算加密的平方可选择加权张量距离 $[\![\lambda_o d_{SWTD}]\!]$，且该结果只有 C_1 知道。安全可选择加权张量距离协议的基本思想是基于3.2.3节提出的可选择加权张量距离公式(3.5)、(3.6)和(3.7)，这里和4.2.2.2节中的安全平方张量距离一样，为了提高算法效率，最终返回的也是距离的平方。算法具体描述如下：

算法 4.6：安全可选择加权张量距离协议

输入：公有云 C_1 输入加密的对象张量 $[\![\lambda_o\mathcal{X}]\!],[\![\lambda_o\mathcal{Y}]\!]$，加密的度量矩阵 $[\![\lambda_e G]\!]$，加密的权重张量 $[\![\lambda_w\mathcal{T}_w]\!]$，私有云 C_2 拥有私钥 *sk*。

输出：加密的平方可选择加权张量距离 $[\![\lambda_o d_{SWTD}]\!]$，结果只有云 C_1 知道。

1：云 C_1，C_2 交互执行下列步骤：

2：for l=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

3：　for m=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

4：　利用 SM 安全计算 $g_{lm}w_l$，$[\![\lambda_e\lambda_w d_{lm}]\!]\leftarrow SM([\![\lambda_e g_{lm}]\!],[\![\lambda_w w_l]\!])$；

5：　利用 SD1 缩倍，$[\![\lambda_e d_{lm}]\!]\leftarrow SD1([\![d_{lm}]\!],\lambda_w)$；

6：　利用同态减安全计算 (x_l-y_l)，$[\![\lambda_o(x_l-y_l)]\!]\leftarrow[\![\lambda_o x_l]\!]*[\![\lambda_o y_l]\!]^{N-1}$；

7：　利用 SM 安全累乘 (x_l-y_l)，$[\![\lambda_e\lambda_o d_{lm}]\!]\leftarrow SM([\![\lambda_e d_{lm}]\!],[\![\lambda_o(x_l-y_l)]\!])$；

8：　利用 SD1 缩倍，$[\![\lambda_o d_{lm}]\!]\leftarrow SD1([\![\lambda_e\lambda_o d_{lm}]\!],\lambda_e)$；

9：　利用 SM 安全累乘 w_m，$[\![\lambda_o\lambda_w d_{lm}]\!]\leftarrow SM([\![\lambda_o d_{lm}]\!],[\![\lambda_w w_m]\!])$；

10：　利用 SD1 缩倍，$[\![\lambda_o d_{lm}]\!]\leftarrow SD1([\![\lambda_o\lambda_w d_{lm}]\!],\lambda_w)$；

11：　利用同态减安全计算 (x_m-y_m)，$[\![\lambda_o(x_m-y_m)]\!]\leftarrow[\![\lambda_o x_m]\!]*[\![\lambda_o y_m]\!]^{N-1}$；

12：　利用 SM 安全累乘 (x_m-y_m)，$[\![\lambda_o^2 d_{lm}]\!]\leftarrow SM([\![\lambda_o d_{lm}]\!],[\![\lambda_o(x_m-y_m)]\!])$；

13：　利用 SD1 缩倍，$[\![\lambda_o d_{lm}]\!]\leftarrow SD1([\![\lambda_o^2 d_{lm}]\!],\lambda_o)$；

14：　end for

15：end for

16：云 C_1 利用同态加累加所有的 $[\![\lambda_o d_{lm}]\!]$，得到加密的平方可选择加权张量距离，

$$[\![\lambda_o d_{SWTD}]\!]\leftarrow\prod_{l,m=1}^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}[\![\lambda_o d_{lm}]\!]。$$

4.3.3 安全的张量多聚类

根据前面的基本安全协议和同态加密理论，本节重点描述本文所提出的云端安全张量多聚类协议。

安全张量多聚类（Secure TMC，STMC）：公有云 C_1 输入n个加密的对象张量 $[\![\lambda_o\mathcal{X}_1]\!],[\![\lambda_o\mathcal{X}_2]\!],\ldots,[\![\lambda_o\mathcal{X}_n]\!]$，以及加密的特征空间组合向量$[\![\mathrm{v}_1]\!],[\![\mathrm{v}_2]\!],\ldots,[\![\mathrm{v}_b]\!]$，私有云 C_2 持有私钥sk。该协议的目标是：C_1 和 C_2 在不泄露对象张量 $\lambda_o\mathcal{X}_1,\lambda_o\mathcal{X}_2,\ldots,\lambda_o\mathcal{X}_n$、特

征空间组合向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_b$ 和中间结果的情况下，协同计算出加密的多聚类结果，且该输出结果只有 C_1 知道。安全张量多聚类的基本思想是基于3.2.5节提出的基于张量的多聚类算法，具体描述如下：

算法 4.7：安全张量多聚类协议

输入：云 C_1 拥有加密的对象张量 $[\![\lambda_o\mathcal{X}_1]\!],[\![\lambda_o\mathcal{X}_2]\!],\ldots,[\![\lambda_o\mathcal{X}_n]\!]$，加密的特征空间组合向量 $[\![\mathrm{v}_1]\!],[\![\mathrm{v}_2]\!],\ldots,[\![\mathrm{v}_b]\!]$，云 C_2 拥有私钥 sk。

输出：加密的多聚类结果 $[\![cl_1]\!],[\![cl_2]\!],\ldots,[\![cl_b]\!]$，该结果只有云 C_1 知道。

1：云 C_1 利用同态加法 SD1 对所有的对象张量的元素做加法，计算关联张量 $[\![\lambda_o\mathcal{T}_a]\!]$，其元素为 $\left[\!\!\left[\lambda_o t^{a}_{i_1 i_2\ldots i_k}\right]\!\!\right] \leftarrow \prod_{d=1}^{n}\left[\!\!\left[\lambda_o t^{ob(d)}_{i_1 i_2\ldots i_k}\right]\!\!\right]$；

2：云 C_1，C_2 交互执行下列步骤：

3：设置 $z = max\{I_{f_1}, I_{f_2},\ldots,I_{f_k}\}$；

4：for l=1 to k do

5：　利用同态加法和 SD2 安全计算转移张量 $[\![\lambda_o\mathcal{T}^{(l)}_{tr}]\!]$，其元素为

$$\left[\!\!\left[\lambda_w t^{tr(l)}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right] \leftarrow SD2\left(\left[\!\!\left[\lambda_o t^{a}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right]^{\lambda_w}, \prod_{i_l=1}^{z}\left[\!\!\left[\lambda_o t^{a}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right]\right);$$

6：end for

7：利用 SAWR 协议安全计算加密的权重向量，

$$[\![\lambda_w\mathrm{w}_1]\!],[\![\lambda_w\mathrm{w}_2]\!],\ldots,[\![\lambda_w\mathrm{w}_k]\!] \leftarrow SAWR\left(\left[\!\!\left[\lambda_w\mathcal{T}^{(1)}_{tr}\right]\!\!\right],\left[\!\!\left[\lambda_w\mathcal{T}^{(2)}_{tr}\right]\!\!\right],\ldots,\left[\!\!\left[\lambda_w\mathcal{T}^{(k)}_{tr}\right]\!\!\right]\right);$$

8：初始化权重向量 $[\![\lambda_w\mathcal{T}_w]\!]$，其元素为 $\left[\!\!\left[\lambda_w t^{w}_{i_1 i_2\ldots i_k}\right]\!\!\right] \leftarrow [\![\lambda_w]\!]$；

9：for l=1 to k do

10：　利用 SM 更新权重张量 $[\![\lambda_w\mathcal{T}_w]\!]$，其元素为

$$\left[\!\!\left[\lambda_w^2 t^{w}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right] \leftarrow SM\left(\left[\!\!\left[\lambda_w t^{w}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right],\left[\!\!\left[\lambda_w(\mathrm{w}_l)_{i_l}\right]\!\!\right]\right);$$

11：　为避免乘法产生溢出，利用 SD1 缩倍，$\left[\!\!\left[\lambda_w t^{w}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right] \leftarrow SD1\left(\left[\!\!\left[\lambda_w^2 t^{w}_{i_1\ldots i_l\ldots i_k}\right]\!\!\right],\lambda_w\right)$；

12：end for

13：for q=1 to b do

14：　for l=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

15：　　for m=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

16：　　利用同态加法计算$\|p_l-p_m\|_2^2$，$\left[\!\left[\|p_l-p_m\|_2^2\right]\!\right]\leftarrow\prod_{t=1}^{k}\left[\!\left[(v_q)_t\right]\!\right]^{(i_t-i_t')^2}$；

17：　　利用 SD1 缩倍，$\left[\!\left[\frac{\lambda_d\|p_l-p_m\|_2^2}{2\sigma^2}\right]\!\right]\leftarrow SD1(\left[\!\left[\|p_l-p_m\|_2^2\right]\!\right]^{\lambda_d\lambda_\sigma},\lambda_\sigma 2\sigma^2)$；

18：　　利用 SE 计算度量矩阵的元素$(g_q)_{lm}$，

$$\left[\!\left[\lambda_\sigma\lambda_e(g_q)_{lm}\right]\!\right]\leftarrow SE(\left[\!\left[\frac{\lambda_d\|p_l-p_m\|_2^2}{2\sigma^2}\right]\!\right]^{N-1})^{\frac{\lambda_\sigma}{2\pi\sigma^2}};$$

19：　　利用 SD1 缩倍，$\left[\!\left[\lambda_e(g_q)_{lm}\right]\!\right]\leftarrow SD1(\left[\!\left[\lambda_\sigma\lambda_e(g_q)_{lm}\right]\!\right],\lambda_\sigma)$；

20：　　end for

21：　end for

22：　for j=1 to n do

23：　　for h= j+1 to n do

24：　　利用 SSWTD 计算视图矩阵,

$$\left[\!\left[\lambda_o(S_V{}^{(q)})_{j,h}\right]\!\right]\leftarrow SSWTD(\left[\!\left[\lambda_o\mathcal{X}_j\right]\!\right],\left[\!\left[\lambda_o\mathcal{X}_h\right]\!\right],\left[\!\left[\lambda_eG_q\right]\!\right],\left[\!\left[\lambda_w\mathcal{T}_w\right]\!\right]);$$

25：　　end for

26：　end for

27：end for

28：利用得到的加密的$\left[\!\left[S_V^{(1)}\right]\!\right],\left[\!\left[S_V^{(2)}\right]\!\right],\ldots,\left[\!\left[S_V^{(b)}\right]\!\right]$构建加密的多视图张量$\left[\!\left[\mathcal{T}_{mv}\right]\!\right]$；

29：将多视图张量$\left[\!\left[\mathcal{T}_{mv}\right]\!\right]$作为安全的 CFS 的输入并行产生加密的多聚类结果。

在第 29 步中，安全的 CFS 是 SHOCFS 的一部分，主要将其中的安全平方张量距离换成这里的安全可选择加权张量距离即可，在此不再另写算法。

4.4 实验分析

本节分别对安全高阶密度峰值聚类和安全的张量多聚类方法进行理论分析和实验验证。本节所有实验都在实验室构建的云平台上进行，包括 8 台 Intel Core i5 3470 3.2Ghz CPU, 16 gb RAM, 4 核计算机；最后给出安全高阶密度峰值聚类和安全张量多聚类方法有代表性的实验结果。

4.4.1 安全高阶密度峰值聚类性能评价

本节首先对安全高阶密度峰值聚类方法提供了安全性分析；其次，从计算成本和通信成本两个方面对该方法进行理论分析；然后，介绍数据集和算法评价指标；最后给出相关实验结果和分析。

4.4.1.1 安全性分析

在安全的两方计算中，本文利用半诚实模型证明安全高阶密度峰值聚类方法的安全性。在混合云模型中，用户不参与安全高阶密度峰值聚类的计算。这里的两方指的是公有云 C_1 和私有云 C_2 且各自遵循每个协议的规则，但与此同时，它们又会在协议执行过程中积极尝试推断隐私信息。由于所有中间结果和最终结果都是使用正式的 Paillier 加密体系来进行保护的，所以 C_1 学习不到任何信息。同时，每个协议的输出也是密文，只有 C_1 知道。此外，尽管 C_2 可以使用私钥 sk 解密中间结果，但它只能看到随机值或加扰动的用户隐私数据。而且，由于本文的协议每一步都使用同态性质或已被正式证明安全的基本协议，如 SM[96]、SC[97]、SD1[99]等，根据组合定理[102]，可认为本文所提出的安全高阶密度峰值聚类方法是可以完全保护用户隐私的。

4.4.1.2 复杂度分析

考虑云计算成本的组成和半诚实混合云安全框架的特点，本小节从理论上分析该方案的计算成本和通信成本。假设数据集中有 n 个对象张量，每个对象张量的元素个数为 m，其中零元素个数为 m_0，非零元素个数为 m_1。

（1）计算成本

客户端先对所有非零数据进行加密再上传到云 C_1，之后就不再参与任何安全的高

阶密度峰值聚类的计算。因此，客户端的计算成本取决于同态加密的时间和所有对象张量的非零元素的数量，假设同态加密是单位时间的，则客户端的计算复杂度为 $O(m_1 n)$。

根据安全高阶密度峰值聚类协议，云端计算成本 T_c 包括安全计算安全平方张量距离矩阵的成本 $T_{c_{sstd}}$ 和安全密度峰值聚类的成本 $T_{c_{scfs}}$，定义为

$$T_c = T_{c_{sstd}} + T_{c_{scfs}}, \tag{4.6}$$

其中，时间复杂度 $T_{c_{sstd}}$ 是 $O((m^2 - m_0^2)n^2)$，时间复杂度 $T_{c_{scfs}}$ 是 $O(kn^2)$。因为要计算一个张量距离，云服务器需要执行 $3m^2$ 次 SM 协议和 $2m^2$ 同态加法操作，由于零元素的运算时间可以忽略，所以复杂度可以降到 $O((m^2 - m_0^2))$；一共要计算 $0.5n(n-1)$ 个张量距离，所以 $0.5n(n-1)$ 时间复杂度是 $O((m^2 - m_0^2)n^2)$；因为安全密度峰值聚类协议使用 $\mathrm{SMIN}_{k\mathrm{th}}$ 协议安全的处理 n^2 个加密的距离，而且它是基于 k 次冒泡排序，即安全密度峰值聚类的时间复杂度为 $O(kn^2)$。因此，总计算时间复杂度 T_c 是 $O((m^2 - m_0^2)n^2 + kn^2)$。

（2）通信成本

假设 Paillier 加密密钥长度为 s，因为客户端需要向云端发送 n 个加密的对象张量，其中每个对象张量包含 m_1 个非零元素，所以在聚类开始之前，客户端到云 C_1 的通信复杂度为 $nm_1 s$。整个聚类算法完成后，由于云 C_1 需要发送到客户端一个随机数，云 C_2 需要发送 n 个对象张量的扰动聚类结果到客户端，因此从云到客户的通信复杂度为 $(n+1)s$。

4.4.1.3 数据集

智能电网的出现给电力系统带来了巨大的变化。例如，聚类企业用电量已被广泛的应用于精准的用电需求预测方法和有效的异常行为分析模型中，其预测分析结果可以提高电网运行效率，同时提升企业的市场竞争力。然而，电力数据采集和云计算也带来了新的隐私威胁。例如，当企业用电情况公布给第三方时，检测企业的生产活动等无意或恶意侵犯隐私的可能性会非常高。

在本实验中，基于真实的智能电网数据集（大航控股集团收集的企业用电消耗数

据[3]、气象系统收集的气象数据[4]、经济网收集的经济数据[5]）对提出的方案进行评估。

该数据集包含了 2015 年江苏省扬中市高新区 1000 多家企业的用电量。气象数据包括天气（晴、雨、雪、阴、多云）和一天的平均气温。经济数据是指反映一定时期内生产领域价格变化的生产者价格指数。本文融合日期、天气、温度、经济四个特征空间，为每个企业构建一个对象张量。对象张量的元素就是需要保护的企业用电数据。

4.4.1.4 评价方法

（1）E_*：计算生成的聚类中心和真实聚类中心之间的距离，它是一种广泛使用的评价聚类中心质量的评价标准[49]，其定义为：

$$E_* = \sqrt{\sum_{i=1}^{c} \left\| v_{ideal}^{i} - v_{*}^{i} \right\|^2}, \tag{4.7}$$

其中，v_{ideal}^{i} 是第 i 个真实聚类中心和 v_{*}^{i} 表明由某个具体算法*产生的第 i 个聚类中心。E_* 值越低，产生的聚类中心更准确。

（2）兰德指数（Rand Index，RI）：用于度量聚类算法聚类结果与实际聚类结果的一致性[49]，计算公式为

$$RI = \frac{TP + TN}{TP + FP + TN + FN}, \tag{4.8}$$

其中，TP 表示正确地将一对相似的对象分组到一个类簇中；TN 表示一对不相似的对象被正确地分到两个类簇中；FP 表示将一对不相似的对象错误地分组到一个类簇中；FN 表示一对相似的对象被错误地分到两个类簇。RI 值越高，聚类结果的准确率越高。

4.4.1.5 加密时间

在将数据外包给云端进行智能电网数据集的聚类之前，客户端需要先使用 Paillier 加密体系对对象张量进行加密。为了评估数据集大小对加密时间的影响，本文在客户端分别加密了 40、80、120、160 和 200 个对象。客户端加密是在一个 4 GB 内存的 CPU 内核中进行的，加密时间如图 4.4 所示。加密时间随对象数量增加呈线性增长，从 1.486 s 到 6.764 s 不等。

[3] https://tianchi.aliyun.com/competition/entrance/231602/information
[4] http://lishi.tianqi.com/yangzhong/201501.html
[5] https://www.ceicdata.com/zh-hans/china/producer-price-index-monthly/ppi-ip-producer-goods-excavation

客户端只需要对每个对象张量执行一次加密，而且加密操作也可以离线预执行；此外，该方案不需要客户端对结果进行解密，只需要使用明文减法去除扰动即可，其执行时间可以忽略。因此，该方案对用户来说是非常轻量级的。

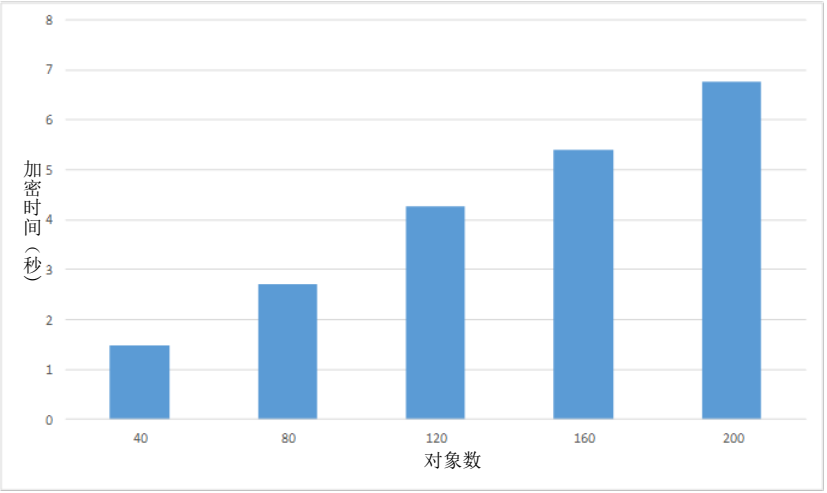


图 4.4 智能电网数据集加密时间

4.4.1.6 执行时间

本节将分别使用不同对象数和不同节点数的加速比来评估使用云计算的安全密度峰值聚类的效率和可扩展性。要实现混合云框架，公有云中至少需要一个节点，私有云中至少需要一个节点，为方便起见，实验部分使用 k 来表示每个云有 k 个节点。在实际的实验中，公有云中的每个节点在私有云中都有对应的节点，这两个节点都有一个套接字连接。在云模拟中，一个公有云节点使用一个 CPU 核，三个私有云节点使用一个 CPU 核，因为私有云节点的负载要比公有云节点轻得多。考虑到截断距离 d_c 的计算成本较高，但是 d_c 也可以根据经验值直接给出，因此本节分别用图 4.5 和图 4.6 来展示方案中是否计算 d_c 的加速比。

图 4.5 为给定因子为 0.02 时计算 d_c 的加速比变化情况。在图 4.5 (a)中，对于 40、80、120、160 和 200 个对象，共用 20 个节点，加速比从 18.62 到 19.84 不等，说明该算法随着数据集大小的增加适合并行执行。从图 4.5 (b)可以看出，加速比随着使用 1 个节点、5 个节点、10 个节点、15 个节点和 20 个节点而增大。当使用节点数较少，如只有 5 个节点时，加速比仅为 4.5，而使用 20 个节点时，加速比为 18.62。这样的

实验结果表明，该方案具有较高的可扩展性，因此可以通过使用更多的云节点实现大数据的安全多聚类可扩展性。

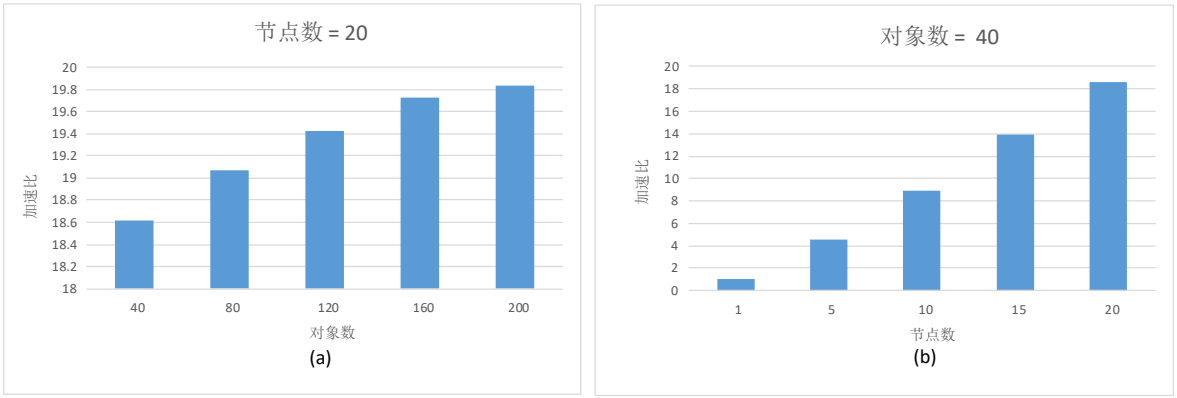


图 4.5 d_c = 0.02 时的加速比

图 4.6 为不计算 d_c 时加速比的变化情况。与图 4.5 相同，图 4.6(a)和图 4.6(b)展示了不同对象和节点个数下的加速比，分别从 19.03 增加到 19.71 和从 1 增加到 19.03，加速比优于计算 d_c 的情况，这是因为大部分执行时间都花费在计算 d_c 上，降低了整体算法的性能。因此，可以将经验值 d_c 作为安全高阶密度峰值聚类算法的输入参数以节省计算成本。但是，从图 4.5 和 4.6 可以看出，计算 d_c 与否的总体趋势都是相同的。综上所述，该方案具有较高的可扩展性，可以通过添加云节点进一步提高其性能。

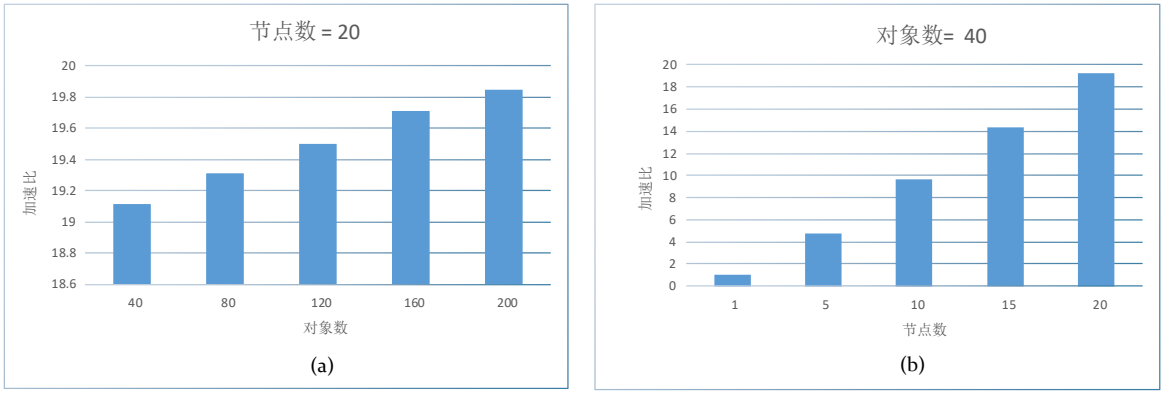


图 4.6 不需要计算 d_c 的加速比

4.4.1.7 聚类准确率

在本小节中，安全高阶密度峰值聚类的准确率评估使用 E_* 和 RI。由于该方案的目标是实现完整的安全的高阶密度峰值聚类算法，因此本文将现有明文算法的结果作

为聚类准确率评价的基准；同时，考虑到所提方法的鲁棒性，分别对 40、80、120、160 和 200 个对象执行了 5 次安全高阶密度峰值聚类方法。E_* 的值都是 0，RI 都是 1。这些结果表明安全高阶密度峰值聚类方法的聚类结果与明文的高阶密度峰值聚类方法算法完全一致，这主要是由具有正规的 Paillier 加密体系和浮点精度控制的保证。因此，就 E_* 和 RI 值来说，该方法可以达到最佳性能的聚类准确率和鲁棒性。

4.4.2 安全的张量多聚类性能评价

本小节将评估安全张量多聚类的性能。首先，对安全张量多聚类的安全性进行验证；然后从理论上对安全张量多聚类的计算复杂度和通信成本进行评价；接下来介绍数据集和评价指标；最后给出了仿真实验结果和相应的分析。所有实验均在模拟云平台上进行，模拟云平台由 Simgrid 工具实验室计算机组成，每台 PC 机采用 3.20GHz Intel Core i5 3470 CPU(4 核)和 16Gb RAM。

4.4.2.1 安全性分析

在安全张量多聚类中仍然遵循半诚实模型下的安全两方计算原则。这里的两方仍指的是云 C_1 和云 C_2，它们共同构成了混合云，能够正确地遵循协议的执行步骤，但在执行协议过程中各方又尽可能的获取更多的额外隐私信息，而客户端在将加密的对象张量上传到给云端之后，不再参与任何张量多聚类的计算。由正式的 Paillier 加密体系保证，C_1 只能获得中间结果和密文多聚类结果；同时，由于 C_2 拥有私钥 sk，根据协议允许其对中间结果进行解密，但它只得到受扰动的明文中间结果和最终多聚类结果。此外，在每一步计算中，所提出的协议都是利用同态加密性质或一些安全性已得到形式化证明的基本安全协议，例如 SM[96]、SC[97]、SD1[99]等，根据复合定理[102]，本文所提出的安全张量多聚类协议是完全安全的。因此，在整个协议的执行过程中，没有任何用户隐私数据泄露给 C_1 和 C_2。

4.4.2.2 复杂度分析

考虑云计算成本的组成和半诚实混合云安全框架的特点，本小节从理论上分析该方案的计算成本和通信成本。假设数据集中包含 n 个对象张量，每个对象张量的元素

个数为 m，其中零元素个数为 m_0，非零元素个数为 m_1。

（1）计算成本

为了减轻客户端的负担，在提出的安全张量多聚类方法中，客户端先加密所有对象张量，然后再将其上传到云 C_1，之后不需要参与任何多聚类的计算。因此，客户端的计算成本由单个元素的加密时间和所有对象张量的非零元素的总数决定，所以客户端的计算复杂度为 $O(m_1 n)$。

在安全的张量多聚类协议中，云计算成本 T_c 包括安全计算转移张量的成本 $T_{c_{tr}}$、安全构建权重张量的成本 $T_{c_{sawr}}$、安全计算可选择加权张量距离矩阵的成本 $T_{c_{sswtd}}$、安全密度峰值聚类的成本 $T_{c_{scfs}}$，算法整体计算复杂度定义为

$$T_c = T_{c_{tr}} + T_{c_{sawr}} + T_{c_{sswtd}} + T_{c_{scfs}}, \tag{4.9}$$

其中，转移张量的计算主要是关联张量，即将所有原始张量进行累加并对每个非零元素做除法的时间，其时间复杂度 $T_{c_{tr}}$ 为 $O(m_0 n + m_1 n)$；安全构造权重张量 $T_{c_{sawr}}$ 的时间复杂度为 $O(m)$，因为转移张量的每个元素都涉及一系列的安全乘法，但安全乘法的次数是一个常数；对于安全计算可选择加权张量距离矩阵的成本 $T_{c_{sswtd}}$，因为计算一个距离，对于每一个 g_{lm} 必须执行四个安全乘法和两个同态减法，共需计算 m^2 个 g_{lm}，总共需要计算 $n(n-1)/2$ 个距离，然而 $T_{c_{sswtd}}$ 不是 $O(m^2 n^2)$ 的原因是零元素没有加密，如果两个零做减法，相应的计算成本可以忽略，因此，$T_{c_{sswtd}}$ 是 $O((m^2 - m_0^2)n^2)$。在[14]中，证明了 SCFS 算法的时间复杂度 $T_{c_{scfs}}$ 为 $O(n^2)$。综上所述，安全张量多聚类协议的总体时间复杂度 T_c 是 $O((m^2 - m_0^2)n^2)$。

（2）通信成本

假设 Paillier 加密密钥长度为 s，执行安全张量多聚类之前，客户端将 $(nm_1 + bk)s$ 个消息上传到云 C_1；整个算法执行完成后，客户端需要从云端下载 $(bn+1)s$ 个消息，其中包括云 C_2 发送的扰动多聚类结果和云 C_1 发送的随机数。

4.4.2.3 数据集

在本节的实验和仿真中，将本文提出的方法应用于两个真实数据集。第一个数据集是 4.4.1.3 中关于智能电网的数据集，包括 2015 年江苏扬中高新区 1000 多家企业的用电量、经济数据和气象数据。在智能电网数据集中，每个对象张量都有四个阶：日期、经济指数（生产者价格指数 PPI）、天气（多云、晴天、阴天、雨雪）和当天的平均气温。一个企业对应一个对象张量，每个对象张量中的元素是该企业在张量空间坐标（日期、天气、温度和 PPI）上的用电量。每个对象张量的规模是 24×24×5×11，分别对应于 24 个不同日期，24 种不同温度，5 种不同天气和 11 个不同的 PPI 数据。第二个数据集是 3.4.1.3 节中的纽约自行车共享系统，共有 473620 条自行车共享记录，包括以下信息：开始时间、停止时间、起点站、终点站等。在自行车的数据集上，一条记录对应于一个对象张量，每个对象张量的规模是 7×4×28×14，四个阶分别对应：交通模式、天气、温度和风速。两个数据集相比，智能电网数据集的对象张量具有更高的维度、更多的非零元素以及值更大的元素。

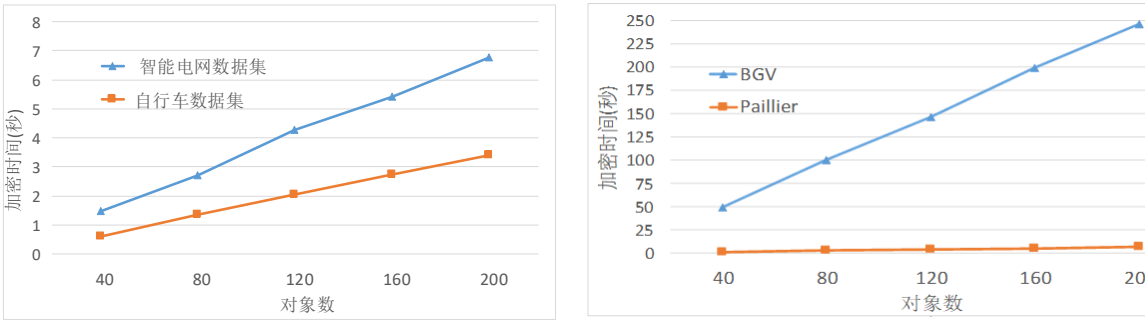
4.4.2.4 评价方法

本节的评价方法同 4.4.1.4，仍然采用 E∗和 RI 来评价聚类质量，在此不再赘述。

4.4.2.5 加密时间

在本方案中，客户端先使用 Paillier 加密体系对对象张量进行加密，然后再将它们上传到云端进行安全的多聚类。因为加密是在客户端进行的，因此需要评估该方案给客户端带来的负担。为了评估数据集规模对加密时间的影响，在两个数据集中分别选取 40、80、120、160 和 200 个对象张量由客户端进行加密。图 4.7(a)为两个数据集中不同对象数的加密时间，其中智能电网数据集分别由 1.486s 增加到 6.764s，自行车数据集由 0.618s 增加到 3.415s，可以看出加密时间随对象数量增加线性增加。此外，自行车数据集的加密时间总是小于智能电网数据集，这说明不同张量的规模、稀疏性以及元素值的大小对加密时间有重要影响。

加密操作在整个方法中只需要执行一次，而且可以离线预执行。另外，在提出的方案中客户端从云端下载多聚类结果后，不需要执行解密操作，只需从得到的明文结果中做明文减法去除扰动即可，其时间可以忽略。



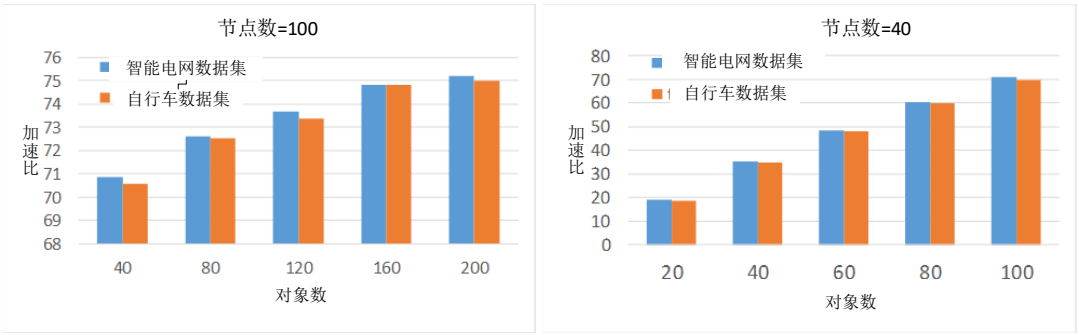
(a) (b)

图 4.7 两个数据集上两种加密方案的加密时间对比

此外，本文还和隐私保护高阶 CFS 方法[49]中使用的 BGV 加密技术进行比较，图 4.7 (b)为分别使用 BGV 和 Paillier 加密体系对智能电网数据集进行加密的时间对比，使用 Paillier 加密的时间只是 BGV 加密时间的 1/40-1/30。总之，所提出的安全张量多聚类方法对于客户端来说是非常轻量级的。

4.4.2.6 执行时间

本节采用延迟比率作为加速比，其定义为并行执行时间与串行执行时间的比率，可以用来评估算法的可扩展性和效率。安全张量多聚类的加速比是在模拟云平台上实验计算得到的。串行执行时间来自在分别云 C_1 和 C_2 上仅使用一个节点的实验，而并行执行时间来自分别在两个云上使用多个节点的实验。仿真实验分别在智能电网数据集和自行车数据集上进行，针对不同的对象和节点数量，仿真结果如图 4.8 所示。



(a) (b)

图 4.8 两个数据集的加速比

图 4.8(a)为使用 100 个节点的加速比变化情况，针对对象数 40、80、120、160 到 200，加速比从 70.85 缓慢增加到 75.19，说明安全张量多聚类方法具有很强的并行可

扩展性。图 4.8 (b)为 40 个对象的加速比变化情况，随着节点数从 20 个增加到 100 个，智能电网数据集和自行车数据集的加速比分别从 18.93 到 70.85 和 18.79 到 69.56，几乎呈线性增加。这样的结果表明，当在云上使用更多的节点时，安全张量多聚类方法在大数据环境下会具有较高的可扩展性。此外，数据集越大通常会产生更高的加速比，但差异并不明显。

4.4.2.7 聚类准确率

在本小节中，安全张量多聚类的准确率使用 E_* 和 RI 进行评价。考虑到所提出的方案是为了实现完整的隐私保护的张量多聚类算法，因此本文将原始明文的张量多聚类算法的结果作为聚类准确率评价的基准。同时，为了评估所提方法的鲁棒性以及扩倍因子对方法准确率的影响，本文对两个数据集各选取 200 个对象，使用了不同的扩倍因子进行实验，结果如表 4.1 所示。

从表 4.1 可以看出，当扩倍因子低于 10^5 时，E_* 的值分别从 0.027 和 0.031 逐渐减小接近 0，而 RI 的值分别从 0.901 和 0.915 逐渐增大接近 1，这样的变化意味着安全的多聚类结果几乎与明文算法一致，但也稍存在一些误差。根据分析，这些误差主要是由 FN 错误导致，即存在个别相似的对象被错误地分到两个类簇。而通过进一步调整精度，即当扩倍因子达到 10^5 后，E_* 和 RI 值分别达到 0 和 1 并稳定。该结果表明，安全的张量多聚类的聚类结果与原始的明文张量多聚类算法完全一致，即隐私保护下的张量多聚类的准确率相对于不考虑安全的原始张量多聚类算法可以达到 100%，这主要是由 Paillier 加密体系和浮点精度扩倍因子保证的。因此，该算法在聚类准确率和鲁棒性方面都具有较好的性能。

表 4.1 聚类准确率评价结果

数据集	评价标准	10^3	10^4	10^5	10^6	10^7
智能电网	E_*	0.027	0.01	0	0	0
	RI	0.901	0.988	1	1	1
自行车	E_*	0.031	0.015	0	0	0
	RI	0.915	0.976	1	1	1

4.5 本章小结

针对大数据环境下不同的应用，为了能够对其提供安全和高效地聚类服务，本章提出了一种安全高阶密度峰值聚类方法，基于此提出了安全张量多聚类方法以及相关的多种安全子协议。在提出的方案中，所有的聚类计算任务都是在云端实现的，而云端不会公开或推断出任何机密信息，这不仅提高了聚类的效率，而且保护了用户的隐私。两种方法都是利用混合云模型下的 Paillier 加密体系来实现安全聚类计算的。最后，本章分别从安全性、计算和通信成本等方面对两种方法进行了理论分析，并在电网和自行车两个真实数据集上进行实验验证，主要从加密时间、执行时间和聚类准确率等方面对提出的方法进行评价。

实验结果表明：（1）两种安全聚类方法都可以在半诚实模型下实现基于 Paillier 加密体系的完整安全协议，保护的信息包括距离值、算法中间结果、聚类中心、类簇数量、每个类簇中的对象数量、每个类簇中的对象；（2）客户端无需参与任何聚类计算，只需进行加密时间仅是 BGV 加密的 1/40 -1/30 的快速加密，以及利用明文减法去除云返回的聚类结果的扰动，这些操作对用户来说是非常轻量级的；（3）通过正式的 Paillier 同态加密体系的保证和利用扩倍因子控制浮点精度，两种方法的聚类结果与明文算法一致，均可达到 100%的聚类准确率；（4）随着节点数的增加，加速比几乎都是呈线性增长，说明当使用更多的云服务器时，算法具有较高的扩展性，而这对于大数据分析和处理是非常重要的。

5 基于张量链分解的张量多聚类及其并行计算方法

张量模型能够很好的描述现实世界中事物之间的关系，但是在大数据环境下，随着数据规模的不断增大，张量的阶数也不断增加，数据存储、计算负荷、内存开销等都将呈指数级增长，从而引起维度灾难问题。因此，有学者提出利用张量分解模型进行数据表示、存储和分析。本文在第 3 章中提出了一种基于张量分解的多聚类方法，它不仅可以通过 Tucker 分解提取高质量核心数据，而且在计算距离时可以将所选择的特征与未选择的特征完全分离，从而提高聚类质量，但是在处理高阶高维数据时仍然存在困难。因为 Tucker 分解是把原始张量分解成一个核心张量和多个因子矩阵，这种分解对于高阶张量而言，分解后的核心张量，其维度依然是指数级，因此仍存在维度灾难问题。另外，由于无法直接在 Tucker 分解的形式上做各种运算，因此需要先还原再进行多聚类，这大大降低了算法的效率。而近年来提出的张量网络理论，因其具有极好的压缩能力和分布式并行处理方式而被认为是分析大数据的非常有前途的工具。因此，本章利用张量网络理论，重点研究一种基于张量链分解的张量多聚类方法并研究其并行计算方法，从而在解决维度灾难的同时，实现张量多聚类的高效计算。

5.1 问题定义

在大数据环境下，基于张量的多聚类方法通常会遇到以下两个问题：第一，随着张量规模的增大，在一个对象张量上进行运算所需的时间、空间会呈指数级增长，从而带来维度灾难问题，将会大大降低多聚类的效率。第二，通过融合多源信息构成的原始对象张量含有大量的冗余和噪声数据，这将极大影响多聚类的准确性。虽然前面本文提出了一种基于张量分解的多聚类方法，但它仍然不能解决上述问题，因此需要一种更加有效的分解方法用于张量多聚类中。近年来，张量网络作为一种用于分析处理高维大数据的新兴技术，成为目前国际上的一个研究热点。不同于张量分解，它是把原始张量分解成多个低阶低维的核心张量。张量链分解[72]作为张量网络的一种重要分解方法，它把原始张量分解成一个链的方式，而分解后的张量都是低阶低维的，一

般都是三阶的，其优点是参数较少，接近于 CP 分解[68]，且算法稳定不需要递归；而且对于天然低秩的张量，通过张量链分解得到的核心数据占用的存储空间更少，或者通过保留一定精度的张量链分解，因保存了更高质量的核心数据，还能进一步提高数据分析的准确率。此外，由于张量链分解得到核心数据可以分布式的存储在云端或者数据中心，更重要的是可以直接在张量链分解的形式上做运算，其结果仍然是张量链的形式，这非常有利于分布式并行计算。

因此，本文假设所有的对象张量均以张量链分解的形式存储在云端或者数据中心上，那么如何在张量链分解的形式上，通过各种张量链的操作或运算，实现完整的张量多聚类以及高效的分布式并行计算是本章需要解决的问题。具体来说，第一，如何尽可能在张量链分解的形式下实现完整的张量多聚类过程并保证甚至提高聚类结果的准确性；第二，如何设计高效的分布式并行计算框架以及张量链核的并行机制，从而极大提高张量多聚类的效率。

5.2 基于张量链分解的张量多聚类方法

在 3.2 节基于张量多聚类方法的基础上，本文研究其在张量链分解形式下的多聚类方法。本节首先介绍基于张量链分解的多线性属性组合权重学习算法，然后描述基于张量链分解的可选择加权张量距离，进而提出基于张量链分解的张量多聚类方法。

5.2.1 基于张量链分解的多线性属性组合权重学习算法

在张量多聚类方法中，多线性属性组合权重学习算法是其重要组成部分，主要目的是度量特征空间属性组合的重要性，从而提高多聚类质量。因此，本小节主要研究如何在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量。

5.2.1.1 关联张量链

根据 3.2.2.1 节中关联张量的构建过程，在张量链分解形式下不再做将非零元素转换为 1 的操作，这样就不用先将张量链形式做还原，而是直接利用张量链加法将所有的对象张量链做累加，相当于将所有原始对象张量直接相加。实际上，这样得到的关

联张量不仅能够体现不同特征空间属性的关联性，而且还能够进一步体现关联的程度。最后，由于连续做张量链的加法操作会带来秩增大的问题，因此为了降秩，本文利用[72]中介绍的基于近似理论的 Rounding 操作来实现。通过以上基于张量链的操作可以得到张量链形式的关联张量链 $\overline{\mathcal{T}_a}$，该过程示意图如图 5.1 所示。

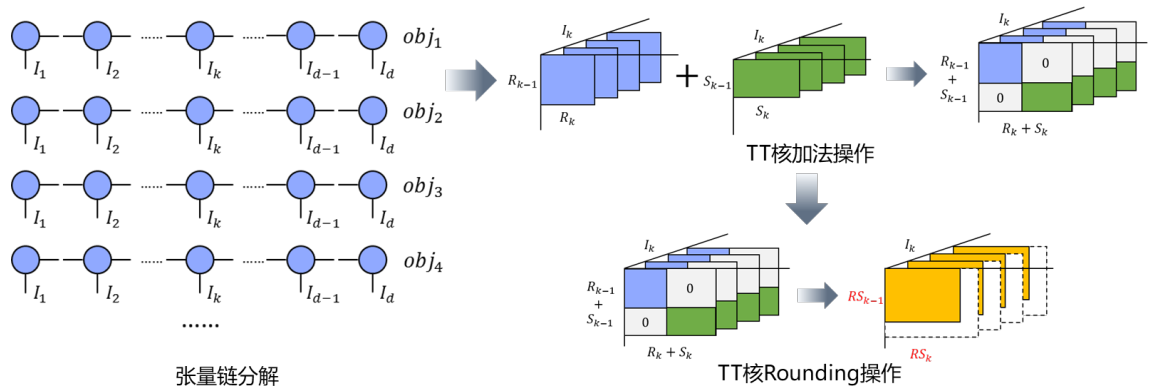


图 5.1 推导关联张量链的过程示意图

5.2.1.2 转移张量链

根据 3.2.2.2，考虑到转移张量是一个超对称张量，需要先将关联张量扩成各阶同维，然后再沿每一阶做归一化。因此需要对关联张量链做相应的操作，首先对关联张量链通过补 0 操作进行扩维，然后对张量链形式下的纤求和，再分别进行非 0 纤和全 0 纤的归一化。

（1）关联张量链的扩维

要将关联张量链扩成各阶同维，需先求出关联张量链所有 TT 核第二阶（即对应原始张量的阶）的最大维数；然后根据得到的最大维数进行补 0 操作，将每一阶都补为最大维数。由张量链分解理论的元素对应关系可知，对于张量链形式下的张量，对原始张量沿某一阶做补 0 操作，就是对该阶所在的 TT 核上沿该阶补元素为 0 的切片。

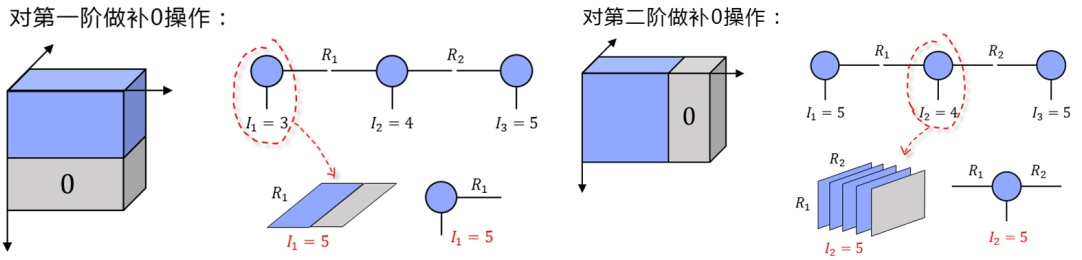


图 5.2 TT 形式下的补 0 操作

例如，图 5.2 展示了在张量链形式下将一个 3×4×5 的张量扩维成 5×5×5 的过程示意图，先对第一个 TT 核做补 0 操作，再对第二个 TT 核做补 0 操作。

（2）张量链形式下的张量纤元素求和

在沿张量某个阶的每个纤做归一化之前，需要先求该纤的和。根据公式（2.11）中张量的元素与 TT 核的一一对应关系，推导后发现对原始张量沿某一阶的纤求和，相当于先把该阶所对应的 TT 核的切片相加求和，再根据张量的纤的坐标中固定值选择对应 TT 核中的切片，这些矩阵相乘即得到一个纤的和，记为 m。例如，本文在一个 4 阶张量中要求纤（2, 3, :, 1）的和，假设该纤的元素分别为 $a_1, a_2, \ldots, a_n$，则它们对应的张量链形式为：第 1 个 TT 核中固定对应第 2 个切片，第 2 个 TT 核中固定对应第 3 个切片，第 3 个 TT 核则分别对应第 $1, 2, \ldots, n$ 个切片，第 4 个 TT 核中固定对应第 1 个切片；而对 $a_1, a_2, \ldots, a_n$ 求和相当于先对第 3 个 TT 核的 n 个切片求和，再与其它切片按阶的顺序相乘，图 5.3 是该求和过程的示意图。

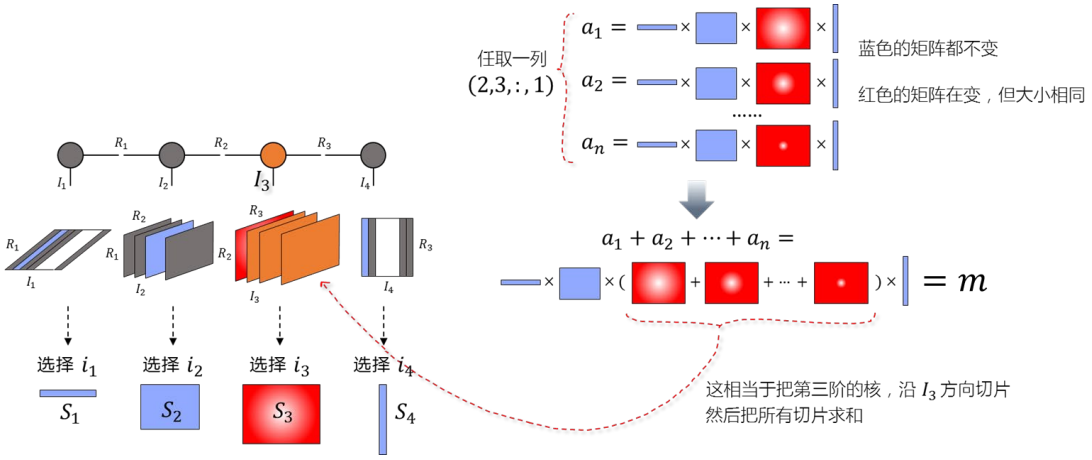


图 5.3 TT 形式下纤元素求和

（3）非 0 纤的归一化

对于张量的元素和不为 0 的纤进行归一化，在原始张量中是直接用元素除以该纤的和，而在张量链形式下由于无法直接做除法，因此为了保证结果一致，本文采用和 m 的倒数 $1/m$ 和元素相乘的做法。这里本文需要设置非 0 纤归一化辅助张量链来存储和的倒数，即首先建立辅助张量，其中和非 0 的纤的元素均放置 $1/m$，其它填 0；然后对辅助张量做张量链分解，得到辅助张量链；最后与扩维后关联张量链做 Hadamard

积，即可得到非 0 纤的归一化后的结果，该结果仍然保持张量链形式。图 5.4 是非 0 纤归一化过程的示意图。

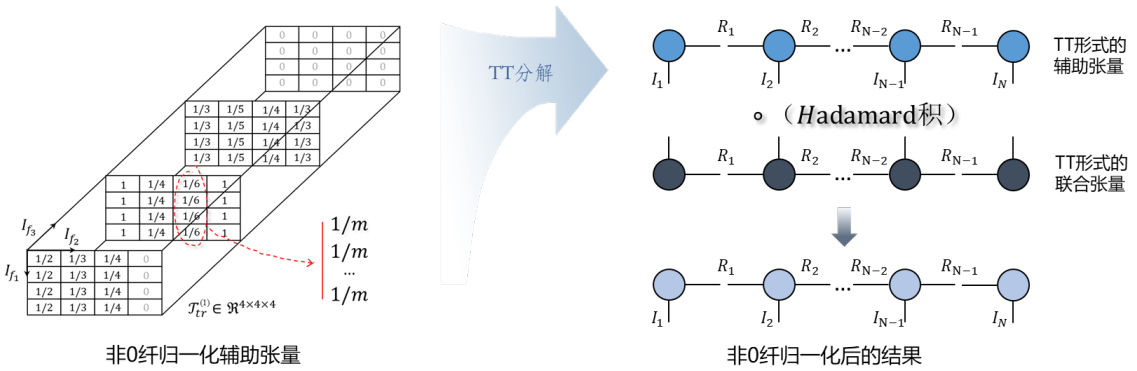


图 5.4 非 0 纤归一化过程示意图

（4）0 纤的归一化

对于和为 0 的纤，其对应的点为悬挂点，根据 3.2.2.2 节中对悬挂点的处理，需要对该纤中非扩维得到的 0 填为$1/dim$，其中 dim 表示扩维之前该阶的维度；而对于扩维填充的 0 则保持不变。因此，本文首先建立 0 纤归一化辅助张量，其中和为 0 的纤按照上述操作分别填$1/dim$ 和 0，而对于和为 0 的纤则全部填 0；然后对辅助张量做张量链分解，得到 0 纤归一化的张量链形式。图 5.5 是 0 纤归一化过程的示意图。

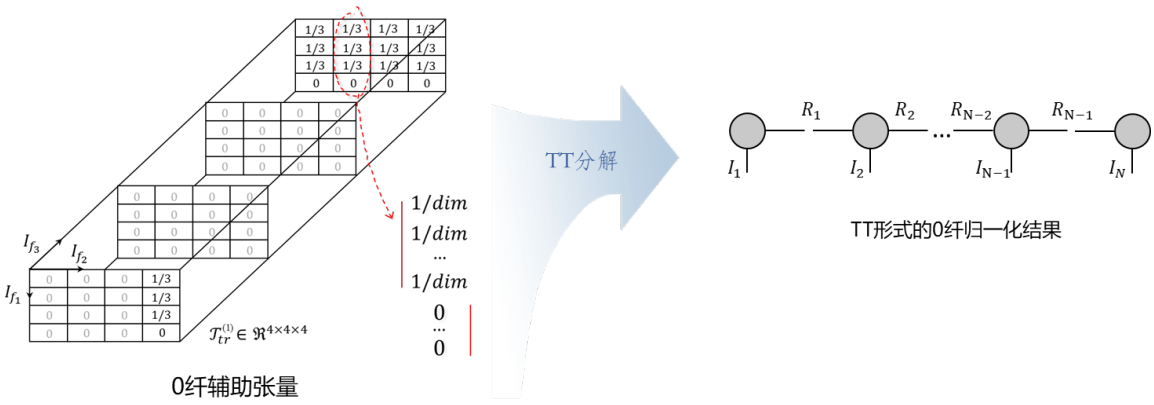


图 5.5 0 纤归一化过程示意图

（5）转移张量链

最后，因为对第（3）和（4）步建立的辅助张量相加即可得到转移张量，因此相应的将第（3）和（4）步得到的张量链形式的非 0 纤和 0 纤的归一化结果，直接做张

量链加法操作，即可得到转移张量链。图 5.6 是最后求转移张量链过程示意图。

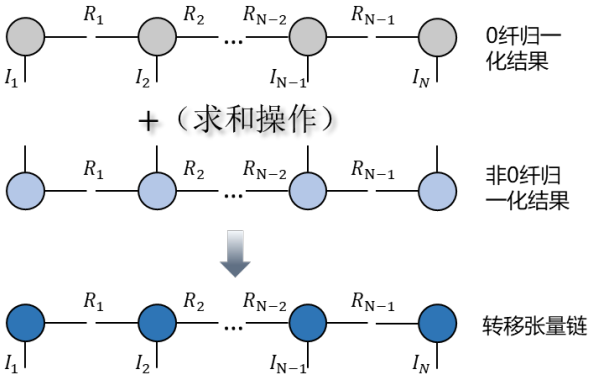


图 5.6　0 纤归一化过程示意图

5.2.1.3　属性组合权重张量链

根据多线性属性权重排名学习算法 3.1，因为输入的转移张量是张量链的形式，修改相应的步骤即可得到张量链形式下的多线性属性权重排名算法，从而得到各阶的属性排名向量，最后做外积即得到权重张量，算法总结如下：

算法 5.1：属性组合权重学习算法

输入：k 阶转移张量链 $\overline{\mathcal{T}_{tr}^{(1)}},\overline{\mathcal{T}_{tr}^{(2)}},\ldots,\overline{\mathcal{T}_{tr}^{(k)}}$ 。

输出：属性组合权重张量链 $\overline{\mathcal{T}_{w}}$ 。

1：设置素性修正概率参数 $0<\alpha<1$；

2：选择阈值 ε；

3：for l=1 to k do

4：　初始化向量 w_0，满足 $\sum_{i=1}^{m}[\mathrm{w}_0]_i=1$；

5：　设置随机向量 u，满足 $\sum_{i=1}^{m}[\mathrm{u}]_i=1$；

6：　初始化变量 $j\leftarrow 0$；

7：　重复执行下列操作

8：　　$j \leftarrow j+1$；

9：　　转移张量做连续单模乘，即对转移张量链 $\overline{\mathcal{T}_{tr}^{(l)}}$ 的 TT 核分别做

$$TT_{core_1}\times_2 \mathrm{w}_{j-1},\ldots,TT_{core_{l-1}}\times_2 \mathrm{w}_{j-1},TT_{core_{l+1}}\times_2 \mathrm{w}_{j-1},\ldots,TT_{core_k}\times_2 \mathrm{w}_{j-1};$$

10： 进行素性修正，$\mathrm{w}_j \leftarrow \alpha \times \mathrm{w}_j + (1-\alpha)\mathrm{u}$；

11： 直到满足排名向量的二范数小于阈值$\left\|\mathrm{w}_j - \mathrm{w}_{j-1}\right\| < \varepsilon$；

12： 截取w_j的前I_{f_l}个元素作为排名向量w_l；

14：end for

15：各阶排名向量做外积获得权重张量，$\mathcal{T}_w \leftarrow \mathrm{w}_1 \circ \mathrm{w}_2 \circ \cdots \circ \mathrm{w}_k$。

16：对权重张量进行张量链分解得到属性权重张量链$\overline{\mathcal{T}_w}$并返回。

5.2.2 基于张量链分解的可选择加权张量距离

在获得权重张量以后，本小节研究张量多聚类的另一个重要组成部分，即如何在张量链分解形式下计算两个对象的可选择加权张量距离。

根据 3.2.3 节，设有对象张量$\mathcal{X}$，$\mathcal{Y} \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$，则$\mathcal{X}$与$\mathcal{Y}$之间的可选择加权张量距离可由公式(3.5)，(3.6)，(3.7)计算得到。因此，要想得到两个对象之间的可选择加权张量距离，必须要遍历所有元素，即首先对对象张量所有对应位置的元素都做减法运算，并将其与权重张量元素w_l、w_m和度量矩阵元素g_{lm}相乘后再相加。而一个张量中含有的元素数量是$I_{f_1} I_{f_2} \cdots I_{f_k}$，如果遍历张量的每个元素来运算，那么需要做的运算次数为$(I_{f_1} I_{f_2} \cdots I_{f_k})^2$。当数据规模大时，需要做的运算量将十分庞大，效率可能会非常低。而在张量链分解的形式下，直接将张量链核对应做相应运算，并通过一定的并行优化使运算过程变得更简单、更高效。

首先，根据公式(3.5)，需要得到对象差张量链核、权重张量链核和度量系数张量链核。因为张量链运算不支持减法操作，所以将张量链形式下的对象张量链$\overline{\mathcal{X}}$加上负的$\overline{\mathcal{Y}}$，即可得到$\mathcal{X}$-$\mathcal{Y}$的张量链形式表示为$\overline{\mathcal{T}_{x-y}}$。权重张量链核$\overline{\mathcal{T}_w}$可由算法 5.1 得到。对于度量系数张量链核，需对系数矩阵$G \in \Re^{I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k} \times I_{f_1} \times I_{f_2} \times \cdots \times I_{f_k}}$做 Rashape 操作，使其先变成度量系数张量$\mathcal{G} \in \Re^{I_{f_1} \times I_{f_1} \times I_{f_2} \times I_{f_2} \cdots \times I_{f_k} \times I_{f_k}}$，再对$\mathcal{G}$做 TT 分解即可得到度量系数张量链核$\overline{\mathcal{G}}$，其中每个核是规模为$R_{n-1} \times I_{f_i} \times I_{f_i} \times R_n$的四阶张量。

其次，在上述得到的张量链核的基础上，将对象差张量链$\overline{\mathcal{X}-\mathcal{Y}}$和权重张量链$\overline{\mathcal{T}_w}$

做 Hadamard 积，得到的结果仍然是张量链的形式。再将 $\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w}$ 与度量系数张量链 $\overline{\mathcal{G}}$ 对应阶分别做两次单模乘运算，得到

$$\overline{d_{SWTD}}=(\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w})\overline{\mathcal{G}}(\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w}). \tag{5.1}$$

图 5.7 是在张量链核的形式下求可选择加权张量距离的过程示意图。

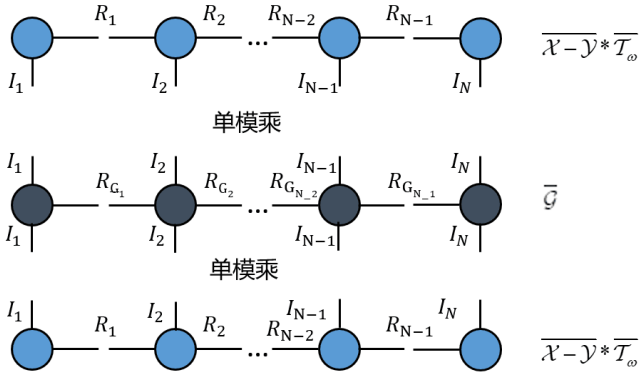


图 5.7　张量链分解形式下可选择加权张量距离过程示意图

最后，将得到的张量链核形式的 $\overline{d_{SWTD}}$ 用 TT 还原操作并对得到的结果进行开方即可得到可选择加权张量距离 d_{SWTD}。

5.2.3 基于张量链分解的张量多聚类方法

根据前面的分析，本文提出张量链形式下的张量多聚类算法，具体描述如下：

算法 5.2：基于张量链分解的张量多聚类算法

输入：对象张量链 $\overline{\mathcal{X}_1},\overline{\mathcal{X}_2},\ldots,\overline{\mathcal{X}_n}\in\Re^{I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}}$，

特征空间选择向量 $\mathrm{v}_1,\mathrm{v}_2,\ldots,\mathrm{v}_r\in\{0,1\}^k$。

输出：多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

1：将所有对象张量链做加法得到关联张量链 $\overline{\mathcal{T}_a}$；

2：求最大维度 $z\leftarrow\max\left\{I_{f_1},I_{f_2},\ldots,I_{f_k}\right\}$；

3：for l=1 to k do

4：　　计算转移张量链 $\overline{\mathcal{T}_{tr}^{(l)}}$；

5：end for

6：根据算法 5.1 计算属性权重张量链 $\overline{\mathcal{T}_w}$；

7：for q=1 to r do

8：　for l=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

9：　　for m=1 to $I_{f_1}\times I_{f_2}\times\cdots\times I_{f_k}$ do

10：　　计算位置距离 $\|p_l-p_m\|_2\leftarrow\sqrt{\sum_{t=1}^{k}v_{q_t}(i_t-i_t^{'})^2}$；

11：　　计算距离矩阵 $g(l,m)\leftarrow\frac{1}{2\pi\sigma^2}\exp\left\{\frac{-\|p_l-p_m\|_2^2}{2\sigma^2}\right\}$；

12：　　end for

13：　end for

14：　Rashape 度量系数矩阵 G 并做 TT 分解得到度量系数张量链 $\overline{\mathcal{G}}$；

15：　for j=1 to n do

16：　　for h= j+1 to n do

17：　　在张量链形式下计算对象张量链相减 $\overline{\mathcal{T}_{x_j-y_h}}$；

18：　　对象差张量链和权重张量链做 Hadamard 积 $\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w}$；

19：　　计算可选择加权张量距离张量链

$$\overline{d_{jh}}=(\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w})\overline{\mathcal{G}}(\overline{\mathcal{X}-\mathcal{Y}}*\overline{\mathcal{T}_w})；$$

20：　　将 $\overline{d_{jh}}$ 还原并开方得到视图矩阵 $S_V^{(q)}(j,h)$；

21：　　end for

22：　end for

23：end for

24：利用得到的 $S_V^{(1)},S_V^{(2)},\ldots,S_V^{(r)}$ 构建多视图张量 $\mathcal{T}_{mv}$；

25：将多视图张量 $\mathcal{T}_{mv}$ 作为典型聚类算法的输入并行产生多聚类结果；

26：返回多聚类结果 $cl_1,cl_2,\ldots,cl_r$。

在算法的第 25 步，本文可以选择任意以距离作为输入的典型聚类算法进行聚类。

5.3 基于张量链分解的张量多聚类并行计算方法

在 5.2 节基于张量链分解的张量多聚类方法的基础上，本文研究其在张量链分解形式下的并行计算方法。本节首先介绍基于张量链分解的张量多聚类并行分析处理框架，进而给出在张量链分解形式下的张量多聚类方法并行策略。

5.3.1 基于张量链分解的张量多聚类并行分析处理框架

本节给出基于张量链分解的张量多聚类并行分析处理框架，其总体框架如图 5.8 所示。

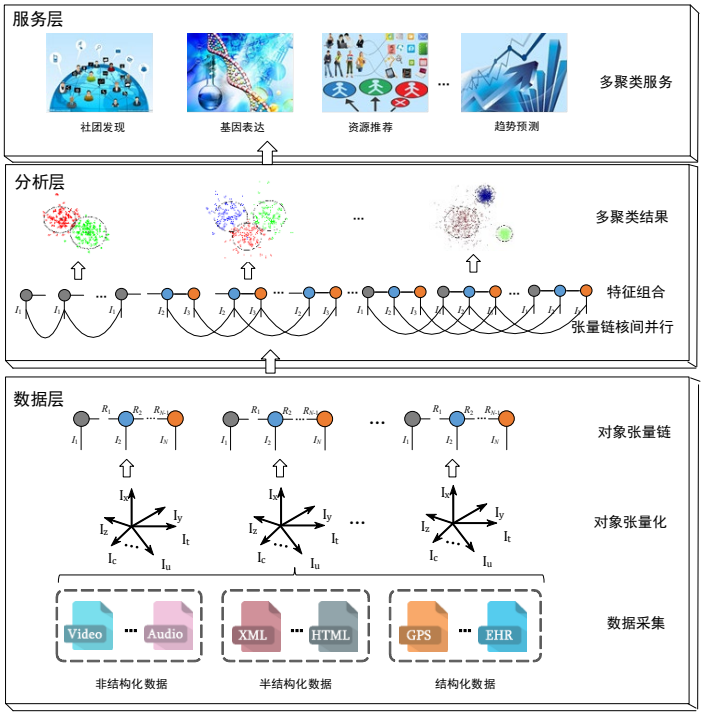


图 5.8 基于张量链分解的张量多聚类并行分析处理框架

该框架自底向上包括数据层、分析层和服务层，下面简单说明每层的功能：

（1）数据层：该层首先对采集的大规模多源异构数据进行表示，与张量多聚类方法不同的是，这里在数据对象表示成张量模型的基础上，对它们进行张量链分解，将分解得到张量链核分布式的存储在云平台上。

（2）分析层：该层主要根据上下文情境变化选择不同的特征组合，即每个对象张量链选择相同阶的张量链核，然后在这些张量链核上直接进行张量多聚类。在多聚类过程中，可对选择的张量链核进行分布式并行计算，以提高算法执行效率。

（3）服务层：类似于张量多聚类方法，该层根据大数据不同应用的需求，服务层将根据分析层产生的不同聚类结果为其提供相应的服务，例如可用于社团发现、基因表达、资源推荐、趋势预测等系统。

5.3.2 基于张量链分解的张量多聚类并行策略

根据张量链分解形式下各张量基本运算的计算规律，利用张量链分解的分布式并行优势，在对象张量链进行张量多聚类时，不仅可以对算法本身进行并行，而且可以对分布式存储的张量链核之间进行并行计算，以提高算法执行效率。例如，在计算属性权重张量时，首先根据选择的特征组合来选择每个对象张量链的核，对相同阶的核之间可以并行做计算，如执行求和、单模乘运算。再如，将$\overline{\mathcal{G}_l}$，$\overline{\mathcal{T}_w}$与$\overline{\mathcal{T}_{x-y}}$在张量链的形式下直接做求和、Hadamard 积、单模乘的时候，也可以对对应阶之间的核进行分布式并行计算。

为了充分利用云节点计算能力来提高算法执行效率，同时保证张量链核之间并行计算的逻辑关系，本文使用了 Map-Reduce 分布式并行思想来实现张量多聚类方法的张量链形式下的计算。依靠分布式消息队列，可以实现根据计算时各节点的计算和通信能力灵活地进行张量链核分配、核调度及核运算，对各节点进行充分利用，从而保证节点之间的负载均衡。并行算法实验的实现可采用 python.multiProcessing 并行处理工具，将存储在云端的张量链核分发给各 Worker 节点进行分布式并行处理，当计算完成后，对得到的张量链核形式的结果进行分组和归约，还原成原始结果，最后收集回 Master 节点进行合并。最终对合并的结果进行多次聚类，计算生成多个不同的聚类结果。

5.4 实验分析

本节对提出的基于张量链分解的张量多聚类算法及其分布式并行计算方法的性能进行实验验证。串行实验是在 3.20 Ghz Intel Core i5 3470 CPU 和 8 GB RAM 的 PC 上实现的，软件环境是 Window 10 和 Python。分布式并行实验在模拟云平台上进行，模拟的云平台由实验室多台 PC 机搭建组成，每台 PC 机采用 3.20GHz Intel Core i5 3470

CPU(4 核)和 16Gb RAM，软件环境为 Ubuntu 18 和 Python，分布式并行处理工具采用 python.multiProcessing，实验中典型聚类算法选用的是 AP 聚类算法。

5.4.1 基于张量链分解的张量多聚类方法性能评估

本节主要对提出的基于张量链分解的张量多聚类方法的性能进行评估。首先给出本实验使用的评价方法和数据集，然后分别从聚类质量、TT 分解精度对算法执行时间的影响、以及 TT 分解精度对内存效率的影响等方面进行评估。

5.4.1.1 评价方法

同 3.4.1.2，本节仍使用 JI 值衡量基于张量链分解的张量多聚类结果之间的冗余度，使用 DI 值来衡量基于张量链分解的张量多聚类结果的聚类质量。在此不再赘述。

5.4.1.2 数据集

同 3.4.1.3，实验使用的测试数据集是来自纽约市的真实世界数据集（从自行车共享系统收集的自行车数据、气象系统收集的气象数据）。自行车数据包含 473620 辆自行车共享记录，包括以下信息：起始时间、停车时间、起点站（车站 ID、车站名称、车站纬度和经度）、目的地车站（车站号、车站名称、车站纬度和经度）等等。气象数据包含 449 条记录，每小时至少有一条记录，每条记录包括四个不同的特征：时间，天气，温度和风速。

实验首先对原始数据进行了一系列预处理步骤，如量化、提取和组合，为每个站生成混合记录，每项记录包括 4 个特征：交通模式，天气，温度和风速。一条记录对应于一个对象张量，每个对象张量的规模是 7×4×28×14。利用获得的数据集，本节随机选择 No.72 站中的 40 条记录进行实验，通过任意组合 4 个特征空间，得到 15 个聚类结果并对其进行分析。

5.4.1.3 聚类质量

图 5.9 展示了使用基于张量链分解的张量多聚类产生的聚类结果得到的 DI 值。水平轴索引值表示不同的特征组合，例如，索引值“1100”前两位为 1，表示选择数据集的交通模式和天气来产生聚类结果，而另外两个特征属性温度和风速则没有被选择。从图中可以看出，大多数情况下，DI 值都接近 1，而且不存在不使用张量链分解

的张量多聚类时会出现 0 值的情况，这样的结果表明算法产生的聚类结果具有较高的聚类质量。通过分析可知，因为在张量链分解形式下进行张量多聚类，需要首先对原始对象张量、权重张量和度量系数矩阵进行 TT 分解，从而可以去除原始数据中的冗余和噪声，提取了高质量的数据集，因此产生的结果具有较高的 DI 值。

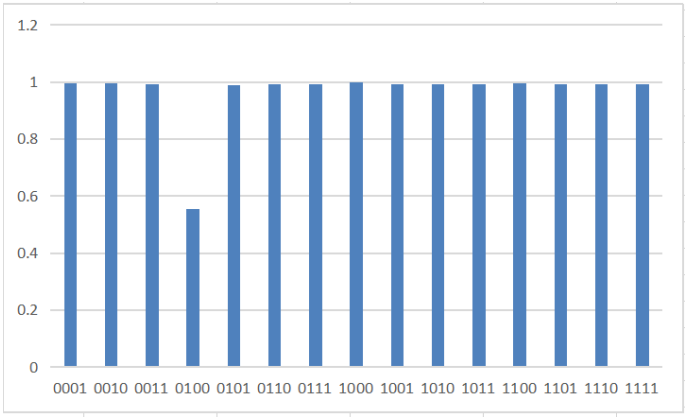


图 5.9 DI 值结果图

图 5.10 展示了从基于张量链的张量多聚类方法产生的多聚类结果得到的 JI 值。JI 值测量聚类结果之间的冗余度，JI 值越低，表示产生的多个聚类结果越不相似。图中每种颜色表示 JI 值的一个范围，百分比表示分布这个范围内的 JI 值的比例。从图中可以看出，87%的 JI 值都分布在 0~0.2 区间范围内，该值也好于不用张量链分解的张量多聚类。通过分析可知，因为聚类结果是在使用了 TT 分解提取高质量数据上得到的，因此该方法产生的 JI 值相对较低，说明利用基于张量链分解的张量多聚类方法能够产生更多不同的聚类结果，可以发现更多不相似的聚类结构，有利于从不同角度对数据进行分析。

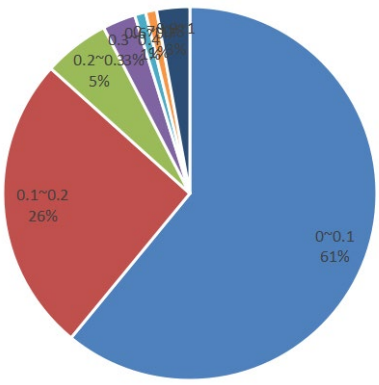


图 5.10 JI 值分布图

5.4.1.4 TT 分解精度对算法执行时间的影响分析

在实验中，分别在 TT 分解精度为 e^{-10}，e^{-20}，e^{-30}，e^{-40}，e^{-50} 的条件下执行基于张量链分解的张量多聚类，结果如图 5.11 所示。从图中可以看出，TT 分解精度对算法执行时间并没有明显的影响，这表明 TT 分解不是本算法提升计算效率的瓶颈，后面如果对算法进行优化，可能的优化方向应该是 TT 形式下的 Hadamard 积以及 TT 形式下的加法运算。

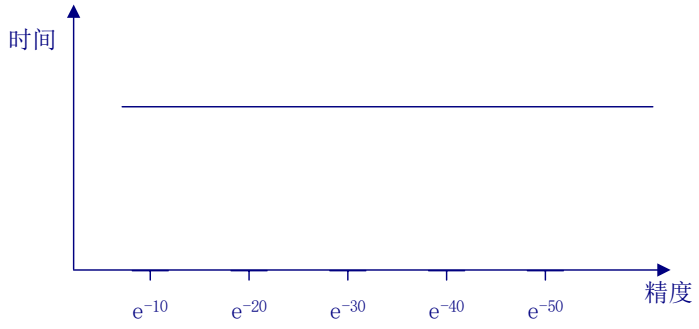


图 5.11 算法执行时间与 TT 分解精度的关系

5.4.1.5 TT 分解精度对内存效率的影响分析

TT 的优势在于可以解决维度灾难问题，对数据提取高质量核心集后压缩并分布式存储，在算法执行过程中，可以在 TT 核上直接做运算，因此在算法执行过程中，通过 TT 分解可以节约内存，有助于提升算法执行效率。图 5.12 展示了在不用张量链分解的张量多聚类算法中，占内存较大的张量和矩阵经过张量链分解后的数据压缩比。从图 5.12(a)中可以看出，使用 TT 分解对于对象张量和权重张量可以获得较大的压缩比，但是随着对象数量的增长，对象张量压缩比逐渐降低并接近稳定在 200，对象数量对于压缩比的影响还是较大的。因此，数据量的增长将是算法效率提升的瓶颈。此外，从图 5.12(b)中可以看出，使用 TT 分解的度量系数矩阵的压缩比是 115284，说明对于原始规模较大的度量系数矩阵，使用 TT 分解后可以获得相当好的数据压缩，这对于算法的效率提升非常关键。

5.4.2 基于张量链分解的张量多聚类算法并行计算性能

本节主要对提出的基于张量链分解的张量多聚类算法分布式并行计算的性能进行评估。首先给出本实验使用的数据集，然后分别从不同对象数和不同节点数的加速

比两个方面进行评估。

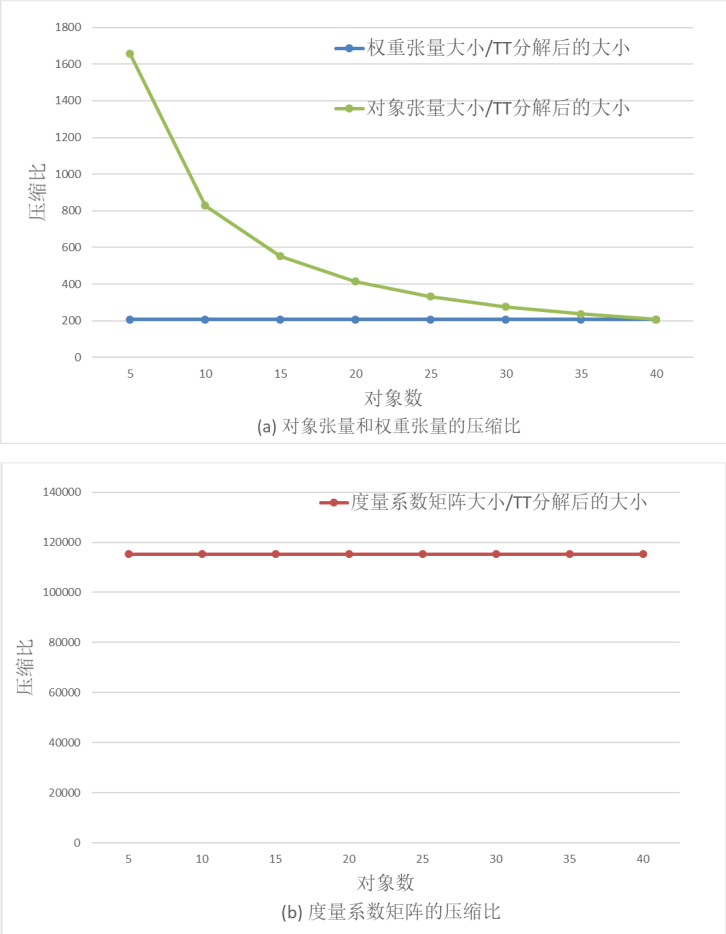


图 5.12 TT 分解对存储效率的影响

5.4.2.1 数据集

在本节的实验和仿真中，仍然使用 3.4.1.3 中实验使用的测试数据集，来自纽约市的真实世界数据集（从自行车共享系统收集的自行车数据、气象系统收集的气象数据）。自行车数据包含 473620 辆自行车共享记录，包括以下信息：起始时间、停车时间、起点站（车站 ID、车站名称、车站纬度和经度）、目的地车站（车站号、车站名称、车站纬度和经度）等等。气象数据包含 449 条记录，每小时至少有一条记录，每条记录包括四个不同的特征：时间，天气，温度和风速。

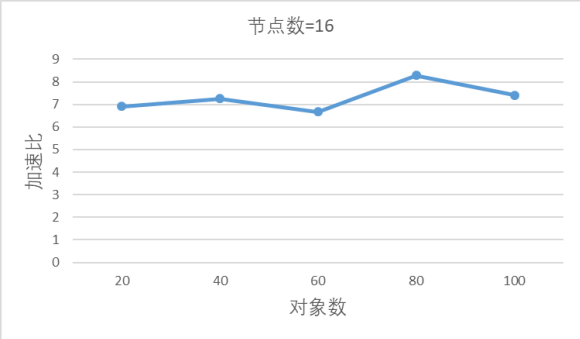
实验首先对原始数据进行了一系列预处理步骤，如量化、提取和组合，为每个站生成混合记录，每项记录包括 4 个特征：交通模式，天气，温度和风速。一条记录对应于一个对象张量，每个对象张量的规模是 7×4×28×14。利用获得的数据集，本节

分别随机选择 No.72 站中的 20、40、60、80、100 条记录进行实验。

5.4.2.2 基于张量链分解的张量多聚类并行计算性能分析

本节采用延迟比率作为加速比，其定义为并行执行时间与串行执行时间的比率，可以用来评估算法的可扩展性和效率。基于张量链分解的张量多聚类的加速比是在模拟云平台上实验计算得到的。串行执行时间来自仅使用一个节点的实验，而并行执行时间来自在模拟云上使用多个节点的实验。仿真实验分别针对不同的对象数和节点数的算法分布式并行和串行执行的加速比，仿真结果如图 5.13 和 5.14 所示。

图 5.13 为使用 16 个节点的加速比变化情况。针对对象数 20、40、60、80 到 100，加速比保持在 7 左右，说明算法随数据量的增长加速比的值较稳定，设计的分布式并行策略很好的保持了节点的负载均衡。图 5.14 为 40 个对象的加速比变化情况。随着节点数从 1 个增加到 16 个，自行车数据集的加速比从 1 到 7，几乎呈线性增加。这样的结果表明，当在云端使用更多的节点时，基于张量链分解的张量多聚类分布式并行计算方法在大数据环境下具有较高的可扩展性。



5.13 不同对象数的加速比

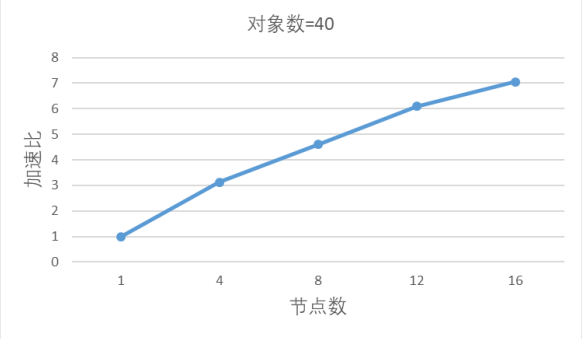


图 5.14 不同节点数的加速比

5.5 本章小结

在云计算分布式环境下，本章重点研究了张量链分解形式下各张量基本运算的计算规律。在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量，研究并实现基于张量链分解的多线性属性组合权重学习算法和可选择加权张量距离，进而构建基于张量链分解的张量多聚类方法。并依据节点计算能力和通信能力设计高效的分布式并行计算框架，研究了张量链核分配机制、核调度策略及

核运算的并行策略，设计了基于张量链核的分布式并行策略，充分利用张量网络并行计算优势提高张量多聚类算法的并行效率。

经过真实数据集的验证，表明基于张量链分解的张量多聚类方法确实能够为多维多源异构数据提供满足不同需求的不同聚类结果，且多聚类结果之间的冗余度较低，聚类质量较好，其并行计算加速比较好，这说明基于张量链分解的张量多聚类方法及其并行策略确实是有效的、可行的。

6 张量多聚类的增量更新方法

在大数据时代，数据对象、数据特征空间实时动态变化，聚类结构需要动态更新。传统方法是定期更新数据集，重新执行聚类算法以生成新的聚类结构，但是原有对象之间的相异性结构并未发生改变，因此完全重新聚类将导致产生大量的重复计算。所以，为了解决大数据环境下多聚类的高效计算问题，研究张量多聚类的增量更新方法具有重要实际意义。本章首先研究经典密度峰值聚类的增量更新方法，进而研究张量多聚类的增量更新方法，最后通过在不同数据集上的真实实验以及不同的评价标准，对提出方法进行全面评估，从而证明方法的有效性。

6.1 问题定义

在现有聚类结果的基础上，对新增对象进行聚类，传统思想是如何将新增对象融入已有结果中，或者独立构成新的类簇等，所以这类方法更加关注的是从已有聚类结果入手，考虑新增对象与它们的关系。但是，这样做容易产生增量聚类结果不准确、效率低等问题，主要是考虑关系不全面以及反复的类簇合并、拆分导致。因此，基于已有聚类结果进行增量计算时，如何最大限度地避免重复计算，并保证增量聚类结果的准确性是本章所要解决的核心问题。

给定一组对象集 $\{T_i\}(i=1,2,3,\ldots,)$，其中对象数量以流方式增长，首先，假设初始数据集有 m_1 个对象，对它们进行聚类并生成初步聚类结果；然后，新增对象或者异类流入，计算出它们与原有对象之间的距离或相似度，连续地输入到增量聚类算法；接下来，增量聚类算法不断地将它们正确的添加到之前的聚类结果中，以更新聚类结构；最后，当用户需要查询最新的聚类结果时，增量聚类算法应该以尽可能小的延迟提供包含所有当前对象的结果。需要注意的是，用户的查询行为是不可预测的，因此在每一次增量聚类结果中，对象的数量 $m_t(t=2,3,\ldots,)$ 也应该是不固定的值。

所以，本章研究的问题是针对流式数据，如何设计增量聚类算法，不仅能够对聚类结果实时更新，并能及时响应用户的查询需求，高效、准确地反馈最新聚类结果。

6.2 密度峰值聚类的增量更新方法

作为张量多聚类算法的基础，需要首先对一些典型聚类算法的增量更新方法进行研究。本节以密度峰值聚类算法作为张量多聚类的基础聚类算法，从算法本身入手，首先研究局部密度的增量更新及对象之间依赖关系的重新连接，然后阐述如何增量式更新中心点及类簇，进而提出增量式密度峰值聚类算法。

6.2.1 局部密度的增量更新

根据对象的局部密度公式(2.33)和距离公式(2.35)可知，当新增对象加入时，原有对象的密度可能会增加。然而，计算δ时需要排好序的局部密度值，新密度和原有密度的变化必然会打乱原有ρ的顺序。针对这一问题，本文提出了基于红黑树的密度更新方法[103]。红黑树是维护动态排序数组的有效工具，它是一种自平衡的二叉搜索树，对于几乎所有需要的操作，包括插入、删除、选择单个或某个值范围内的元素等，它都能保证$O(logn)$的复杂度，其空间复杂度为$O(n)$；同时红黑树提供了高效的按值访问，其中元素按值排序，其原始索引也存储在元素中；另外为了提供高效的索引访问，还保留了原有无序的动态数组。

对于局部密度更新的主要思想是：当一个新增对象到来时，首先计算它与所有已有对象之间的距离，这一步是必须的；然后，将这些所有距离和d_c来计算新增对象的ρ，每有一个距离小于d_c，则新增对象的ρ值加1，同时也要将该距离对应的已有对象的ρ值加1，因为它们现在有一个新的邻居点；最后，计算新增对象的局部密度并将其插入有序数组T_ρ和无序的动态数组A_ρ中。对于算法的实现，本文利用红黑树来实现局部密度的更新操作，在原有排好序的局部密度数组T_ρ中删除旧的局部密度ρ_i，并插入新的局部密度值ρ_i+1。关于对象的局部密度的增量更新算法描述如下：

算法6.1：局部密度增量更新算法

输入：原有以连接依赖表示的聚类结果R，原有局部密度数组T_ρ，原有无序的局部密度动态数组A_ρ，原有无序的依赖距离动态数组A_δ，距离d_{ij}（$0<i<j<=m$，其

中 $m-1$ 为已有对象的个数，新增对象为第 m 个对象），截断距离 d_c 。

输出：更新后的 $T_\rho^{'}, A_\rho^{'}, R^{'}, A_\delta^{'}$ 。

1：初始化新增对象的局部密度 $\rho_m \leftarrow 0$；

2：for i=1 to m-1 do

3：　判断，如果 $d_{im} < d_c$，那么

4：　　新增对象的局部密度加1，$\rho_m \leftarrow \rho_m + 1$；

5：　　更新已有对象的局部密度 $\rho_i \leftarrow \rho_i + 1$，并更新它在 T_ρ 和 A_ρ 的位置；

6：　　利用算法6.2检测并重新连接密度超过局部密度刚更新的对象；

7：　end if

8：　在 T_ρ 和 A_ρ 插入 ρ_m；

9：　从 T_ρ 中选择局部密度大于新增对象的对象，找到它的连接依赖对象，将结果插入到R和 A_δ 中。

10：end for

11：返回更新后的 $T_\rho^{'}, A_\rho^{'}, R^{'}, A_\delta^{'}$ 。

6.2.2 对象之间依赖的重新连接

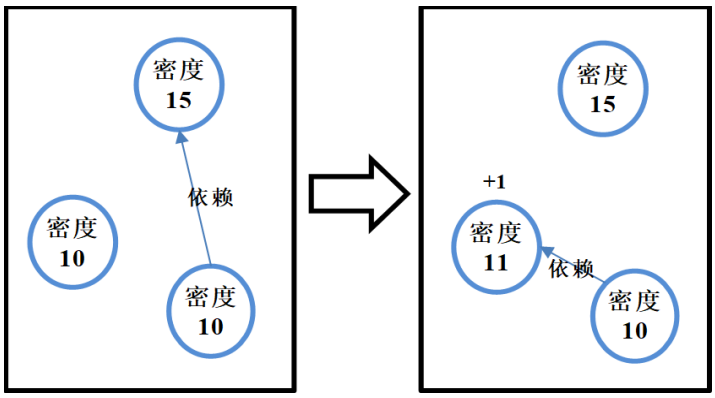


图 6.1 连接依赖更新示意图

由公式局部密度公式(2.33)和距离公式(2.35)可知，当对象的局部密度变化时，对象之间的顺序连接的连接依赖关系以及距离 δ 都是需要改变的。图6.1给出了一个简单的连接依赖更新的示意图，局部密度为10的对象A连接到局部密度为15的对象B，局部

密度为10的点*C*经过增量计算后变成了11。此时点*C*的局部密度比点*A*大，且d_{ac}显然小于d_{ab}。显然，*A*点需要重新连接到*C*点，而不是*B*点。为了高效的检测和重新连接所有像*A*一样的点，本文设计了基于红黑树的重新连接方法，其主要步骤描述如下：

算法 6.2：依赖关系重新连接算法

输入：原有以连接依赖表示的聚类结果R，原有局部密度数组T_ρ，原有无序的局部密度动态数组A_ρ，原有无序的依赖距离动态数组A_δ，距离d_{ij}（$0<i<j<=m$，其中$m-1$为已有对象的个数，新增对象为第m个对象），截断距离d_c，局部密度刚更新过的对象*I*。

输出：更新后的$R^{'}, A_\delta^{'}$。

1：从T_ρ中，选择在密度区间$[\rho_I-1,\rho_I)$内的对象，获得这些对象的ρ值和索引；

2：对于每一个选定的对象，执行下列操作

3：　使用索引得到选择的对象J和I的距离d_{IJ}；

4：　判断，如果$d_{IJ}<\delta_J$，那么

5：　　重新连接J指向I，更新距离$\delta_J \leftarrow d_{IJ}$；

6：　　更新数组R和A_δ；

7：　end if

8：返回更新后的$R^{'}, A_\delta^{'}$。

在算法的第 1 步中，根据红黑树的性质，该算法通过在T_ρ中选择局部密度在区间$[\rho_I-1,\rho_I)$的对象，从而找到那些局部密度被局部密度增加的对象所超过的对象；然后从第 2 步到第 8 步，对于每个局部密度被超越的对象，检查超越它的对象是否比以前的连接依赖的对象更近，如果更近，那么应该更新连接依赖关系与δ值；最后将这类更新记录在R和A_δ中。

6.2.3 聚类中心及类簇的增量更新

对于聚类中心和类簇的更新，本文研究设计一种轻量级的更新方法。因此，这里将进行批量的聚类中心和类簇更新，也就是说在算法接收到最新的查询后，再执行更

新聚类中心和类簇的步骤，这样可以避免很多冗余的重复执行。但是，批量更新可能会稍微延迟查询后聚类结果的反馈。因此，如果该算法应用于反馈延迟敏感的场景，那么这一步应该定期执行，而不是等待最新的查询。

如第2.3节所述，密度峰值聚类方法需要有序的γ值来选择中心点，然后把整个类簇的连接依赖通过将中心点的连接依赖赋最大值n+1来切断。针对这一思想，中心点和类簇更新方法的目标是如何高效地重新计算和重新排序ρ或δ值有更新的对象的γ值。在此为了减少重复计算，当一个对象的ρ或δ值在前面的步骤中有更新时，本文通过将对象的索引插入到一个哈希表结构来记录变化的对象。

算法6.3描述了所提出的中心点和类簇更新算法，对于所有ρ或δ值改变的对象，该方法更新它们的γ值，同时保持它们在红黑树中的顺序。

算法6.3：中心点及类簇更新算法

输入：原有以连接依赖表示的聚类结果R，原有判断中心的指标数组T_γ，原有无序的局部密度动态数组A_ρ，原有无序的动态距离数组A_δ，原有无序的判断中心的指标数组A_γ，原有聚类中心集合c，ρ或δ值有变化对象集合A_{cp}。

输出：更新后的$T_\gamma^{'}, A_\gamma^{'}, c^{'}$和一个$R$的副本。

1：复制T_γ的$top(c+1)$个元素到集合$A_{top\gamma}$；

2：对于$A_{top\gamma}$中的每个对象i执行下列操作

3：　　得到这个对象的ρ_i, δ_i；

4：　　从A_γ得到旧的γ_i，并使用其值在T_γ中删除它自身；

5：　　计算新值$\gamma_i \leftarrow \rho_i \delta_i$，然后插入到$T_\gamma$并更新$A_\gamma$；

6：结束循环；

7：判断，如果T_γ的$(c+1)-top$的元素都和$A_{top\gamma}$一样，那么

8：返回更新后的$T_\gamma^{'}, A_\gamma^{'}, c^{'}$和一个$R$的副本；

9：end if

10：遍历T_γ，按降序选择中心点，并像原始密度峰值聚类方法那样分割类簇；

11：返回更新后的$T_\gamma^{'}, A_\gamma^{'}, c^{'}$和一个$R$的副本。

上述算法中从第3到5步，对象的索引取自给定的对象集合，并从相应的数组取出更新后的ρ和δ值以及原有的γ值，获取原有的γ值是为了高效地在红黑树中定位和删除它们；然后，计算新的γ值，插入到T_γ并更新A_γ；最后，更新所有的γ值，从第7步到第9步，该算法检查原有中心点，如果所有的原有中心依然存在于$A_{top\gamma}$，这意味着中心点没有变化，中心点和类簇更新完成；否则，算法将遍历T_γ的每个值，按降序选择新的中心点，最后像经典的密度峰值聚类方法一样分割类簇。

6.2.4 增量式密度峰值聚类算法

根据前面对密度峰值聚类算法主要组成部分的增量更新方法的研究，本文提出完整的增量式密度峰值聚类算法（Incremental CFS，ICFS），具体描述如下：

算法6.4：增量式密度峰值聚类算法

输入：原有对象之间的距离$d_{ij}(0<i<j\le m_1)$，截断距离d_c。

增量输入：在新增对象和原有对象之间的距离d_{ij}，对聚类结果的查询*query*。

输出：最新的聚类结果$cl^{'}$。

1：初始化各相关数组 $(R_0 T_\rho T_\gamma, A_\rho, A_\delta, A_\gamma) \leftarrow \text{initial\_clustering}(d_{ij}, d_c)$；

2：设置对象的个数$m \leftarrow m_1$；

3：初始化$R \leftarrow R_0$；

4：执行循环：

5：　新增一个对象，$m \leftarrow m+1$；

6：　计算新的距离$d_{jm}\left(0<j<=m\right)$；

7：　利用算法6.1更新局部密度，在密度更新过程中利用算法6.2重新连接对象；

8：　利用算法6.3更新中心点和类簇；

10：结束循环

11：返回最新的聚类结果$cl^{'}$。

6.3 张量多聚类的增量更新方法

根据 3.2 节介绍的张量多聚类方法可知，其主要包括关联张量构建、多线性属性组合权重学习算法、可选择加权张量距离与典型聚类算法四部分。其中关联张量的增量计算相对简单，只需对新增对象求和即可，此外因为距离天然以相异度矩阵的形式进行增量，因此，本节重点研究其它两个部分的增量计算方法。首先研究两种不同的属性权重学习增量更新方法，然后介绍提出的增量式 K-medoids 聚类算法，最后阐述提出的增量式张量多聚类算法。

6.3.1 权重学习迭代增量更新方法

考虑非增量的多线性属性组合权重学习算法，首先生成一个随机向量作为初始的权重向量；然后用该向量依次单模乘转移张量的各个阶，只跳过需要训练权重的那一阶，再进行素性修正；最后得到一个新的权重向量，之后依次循环迭代更新该权重向量，直到算法收敛，即权重向量几乎不再变化为止（用新旧向量之间的二范数小于误差阈值来判定）。因此，在做增量权重学习时，一个自然的想法是直接将当前权重向量作为属性权重向量学习算法下一次循环的初始向量，在此基础上迭代计算更新权重向量，即以当前权重向量代替原始算法的初始随机向量，从而增量学习属性权重向量。迭代增量更新方法的算法描述如下：

算法 6.5：权重排名迭代增量更新算法

输入：N 阶增量转移张量 $\mathcal{T}_{tr}^{(1)},\mathcal{T}_{tr}^{(2)},\ldots,\mathcal{T}_{tr}^{(k)}\in\Re^{\overbrace{m\times m\times\cdots\times m}^{k}}$，各特征空间的属性维数分别为 $I_{f_1},I_{f_2},\ldots,I_{f_N}$，当前已有权重向量 $\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_N$。

输出：更新的属性权重排名向量 $\mathrm{w}_1^{'}\in\Re^{I_{f1}},\mathrm{w}_2^{'}\in\Re^{I_{f2}},\ldots,\mathrm{w}_N^{'}\in\Re^{I_{fN}}$。

1：设置素性修正概率参数 $0<\alpha<1$；

2：选择阈值 ε；

3：for l=1 to k do

4：　将当前已有第 l 阶的权重向量作为初始化向量 $\mathrm{w}_0^{'}\leftarrow\mathrm{w}_l$；

5：　设置随机向量 u，满足 $\sum_{i=1}^{m}[\mathrm{u}]_i=1$；

6： 初始化变量 $j=0$；

7： 重复执行下列操作

8： $j=j+1$；

9： 连续单模乘，$\mathrm{w}_j^{'} \leftarrow \alpha \mathcal{T}_{tr}^{(l)} \times_1 \mathrm{w}_{j-1}^{'} \cdots \times_{l-1} \mathrm{w}_{j-1}^{'} \times_{l+1} \mathrm{w}_{j-1}^{'} \cdots \times_N \mathrm{w}_{j-1}^{'} + (1-\alpha)\mathrm{u}$；

11： 直到满足 $\left\| \mathrm{w}_j^{'} - \mathrm{w}_{j-1}^{'} \right\| < \varepsilon$；

12： 截取 $\mathrm{w}_j^{'}$ 的前 I_{f_l} 个元素作为排名向量 $w_l^{'}$；

13：end for

14：返回 $\mathrm{w}_1^{'}, \mathrm{w}_2^{'}, \ldots, \mathrm{w}_N^{'}$。

通过对该方法进行分析，其复杂度为 $O(kn^N)(N \geq 2)$，因为算法是在当前权重向量基础上开始训练的，因此收敛速度快，而且可以保证增量学习结果的高准确率，该方法同时适用于流式和批量增量学习，但可能存在不收敛情况。因此，本文研究另一种基于微分理论的权重学习增量更新方法。

6.3.2 权重学习微分增量更新方法

受增量式谱聚类方法[104]以及数据排名增量更新方法[105]的启发，本文提出一种权重学习的微分增量更新方法。本小节首先提出两个定理，并根据相关理论给出其证明，然后介绍特征空间属性权重学习基于微分理论的增量更新方法。

定理 6.1： 假设已知公式 $\mathrm{w} = \mathcal{T}_{tr} \times_2 \mathrm{w} \times_3 \mathrm{w} \times \cdots \times_N \mathrm{w}$，其中 $\mathcal{T}_{tr}$ 和 w 分别表示 N 阶转移张量和属性权重排名向量。令 $\Delta\mathcal{T}_{tr}$ 表示转移张量 $\mathcal{T}_{tr}$ 的增量部分，$\Delta\mathrm{w}$ 表示权重排名向量 w 的增量部分，则权重排名向量的增量与转移张量的增量满足如下公式：

$$\begin{aligned}&(\mathcal{T}_{tr} \times_2 \mathrm{w} \times_3 \mathrm{w} \cdots \times_{N-1} \mathrm{w} + \mathcal{T}_{tr} \times_2 \mathrm{w} \times_3 \mathrm{w} \cdots \times_{N-2} \mathrm{w} \times_N \mathrm{w} + \mathcal{T}_{tr} \times_3 \mathrm{w} \cdots \times_N \mathrm{w} - I)\Delta\mathrm{w} \\ &= -\Delta\mathcal{T}_{tr} \times_2 \mathrm{w} \times_3 \mathrm{w} \cdots \times_N \mathrm{w}. \end{aligned} \tag{6.1}$$

其中，I 是单位矩阵。

证明： 在给定的公式两边取微分，可以得到

$$(\mathrm{w}+\Delta\mathrm{w}) = (\mathcal{T}_{tr} + \Delta\mathcal{T}_{tr}) \times_2 (\mathrm{w}+\Delta\mathrm{w}) \times_3 (\mathrm{w}+\Delta\mathrm{w}) \cdots \times_N (\mathrm{w}+\Delta\mathrm{w}). \tag{6.2}$$

将上述公式展开，可以得到如下表的微分项，

表 6.1 在公式(6.1)右边执行微分操作得到的 2^N 项

	1	2	3	…	2^{N-1}	2^N
	$\mathcal{T}_{tr}$	$\mathcal{T}_{tr}$	$\mathcal{T}_{tr}$	…	$\Delta\mathcal{T}_{tr}$	$\Delta\mathcal{T}_{tr}$
1	w	w	w	…	Δw	Δw
2	w	w	w	…	Δw	Δw
⋮	⋮	⋮	⋮		⋮	⋮
N-2	w	w	Δw	…	w	Δw
N-1	w	Δw	w	…	Δw	Δw

在表 6.1 中，对二阶微分项如 $\mathcal{T}_{tr}\underbrace{\text{w}\cdots\text{w}}_{N-3}\Delta\text{w}\Delta\text{w}$，三阶微分项如 $\mathcal{T}_{tr}\underbrace{\text{w}\cdots\text{w}}_{N-4}\Delta\text{w}\Delta\text{w}\Delta\text{w}$，…，$N$ 阶微分项 $\Delta\mathcal{T}_{tr}\underbrace{\Delta\text{w}\cdots\Delta\text{w}}_{N-1}$，相对于一阶微分项，这些二阶微分项、三阶微分项、…、N 阶微分项的变化对最终的增量更新结果影响不大，因此在计算过程中可以忽略不计。

因此，移去二阶微分项、三阶微分项、…、N 阶微分项，可得

$$\begin{aligned}\Delta\text{w} = {} & \mathcal{T}_{tr}\times_2\text{w}\times_3\text{w}\cdots\times_{N-1}\text{w}\times_N\Delta\text{w} + \mathcal{T}_{tr}\times_2\text{w}\times_3\text{w}\cdots\times_{N-2}\text{w}\times_{N-1}\Delta\text{w}\times_N\text{w} \\ & +\cdots+\mathcal{T}_{tr}\times_2\Delta\text{w}\times_3\text{w}\cdots\times_N\text{w} + \Delta\mathcal{T}_{tr}\times_2\text{w}\times_3\text{w}\cdots\times_N\text{w}. \end{aligned} \tag{6.3}$$

再通过一系列等式变换，整理后可以得到公式(6.1)。

为了表达和计算方便，将排名向量的计算过程通过矩阵进行重新表示，采用如下公式计算排名向量

$$\text{w} = M\underbrace{(\text{w}\otimes\text{w}\cdots\otimes\text{w})}_{N=1}. \tag{6.4}$$

上边的定理 6.1 也随之改变为如下定理 6.2。

定理 6.2: 给定公式 $\text{w} = M\underbrace{(\text{w}\otimes\text{w}\cdots\otimes\text{w})}_{N=1}$，其中 M 表示由转移张量展开得到的矩阵。令 ΔM 和 Δw 分别代表转移张量展开矩阵增量和排名向量增量部分，则它们满足如下公式：

$$\Delta\text{w} = \mathcal{T}_{tr}(\underbrace{\text{w}\otimes\text{w}\otimes\cdots\otimes\text{w}}_{N-2}\otimes\Delta\text{w}) + \mathcal{T}_{tr}(\underbrace{\text{w}\otimes\text{w}\otimes\cdots\otimes\text{w}}_{N-3}\otimes\Delta\text{w}\otimes\text{w})$$

$$+\cdots+\mathcal{T}_{tr}(\Delta\mathrm{w}\otimes\underbrace{\mathrm{w}\otimes\cdots\otimes\mathrm{w}}_{N-2})+\Delta\mathcal{T}_{tr}(\underbrace{\mathrm{w}\otimes\cdots\otimes\mathrm{w}}_{N-1}). \tag{6.5}$$

证明： 对给定公式的两边分别求微分，可得

$$(\mathrm{w}+\Delta\mathrm{w})=(M+\Delta M)\underbrace{((\mathrm{w}+\Delta\mathrm{w})\otimes(\mathrm{w}+\Delta\mathrm{w})\otimes\cdots\otimes(\mathrm{w}+\Delta\mathrm{w}))}_{N=1}. \tag{6.6}$$

展开公式右边的直积(Kronecker 积)，可以得到

$$\begin{aligned}&\underbrace{(\mathrm{w}+\Delta\mathrm{w})\otimes(\mathrm{w}+\Delta\mathrm{w})\otimes\cdots\otimes(\mathrm{w}+\Delta\mathrm{w})}_{N-1}\\&=\underbrace{\underbrace{\mathrm{w}\otimes\cdots\otimes\mathrm{w}}_{N-1}+\underbrace{\mathrm{w}\otimes\cdots\otimes\mathrm{w}}_{N-2}\otimes\Delta\mathrm{w}+\cdots+\underbrace{\Delta\mathrm{w}\otimes\cdots\otimes\Delta\mathrm{w}}_{N-2}\otimes\mathrm{w}+\underbrace{\Delta\mathrm{w}\otimes\cdots\otimes\Delta\mathrm{w}}_{N-1}}_{2^N}.\end{aligned} \tag{6.7}$$

将上述公式展开，移去较小影响的二阶、三阶、…、N 阶微分项，则可得到公式(6.1)。

为了能够直接计算数据排名向量的增量结果，对上述公式进行变换。假设矩阵 $A\in R^{n\times n}$，矩阵中的元素表示为

$$a_{s,t}=\sum_{i=1}^{n^{N-2}}(m_{s,i+(t-1)n^{N-2}}p_i)+\sum_{k=1}^{N-3}\sum_{i=0}^{n^{N-(k+2)}-1}\sum_{j=1}^{n^k}(m_{s,in^{k+1}+j+(t-1)n^k}p_{in^k+j})+\sum_{i=0}^{n^{N-2}-1}(m_{s,in+t}p_{i+1}) \tag{6.8}$$

其中，

$$p_i=\prod_{l=1}^{N-2}(w_{i\%n^{N-l}/n^{l-1}+1}) \tag{6.9}$$

令 $b\in\Re^n$ 表示一个 n 维向量，其元素定义如下

$$b_i=-\sum_{x=1}^{n^{N-1}}(\Delta m_{i,x}q_x) \tag{6.10}$$

其中 q_x 是矩阵 $Q=\mathrm{p}\otimes\mathrm{w}$ 的元素。

定义权重排名向量的增量为 $\Delta\mathrm{w}=\left[\Delta w_1,\Delta w_2,\cdots,\Delta w_n\right]^{\mathrm{T}}$，则公式(6.5)可以表示成

$$\begin{bmatrix}a_{1,1}&a_{1,2}&\cdots&a_{1,n}\\\vdots&\vdots&&\vdots\\a_{n,1}&a_{n,2}&&a_{n,n}\end{bmatrix}\begin{bmatrix}\Delta w_1\\\Delta w_2\\\vdots\\\Delta w_n\end{bmatrix}=\begin{bmatrix}b_1\\b_2\\\vdots\\b_n\end{bmatrix}. \tag{6.11}$$

根据公式(6.11)，可以将权重排名向量的增量部分表示为 $\Delta\mathrm{w}=A^{-1}b$。假定矩阵 A 非奇异，则可以通过采用追赶法、高斯消元法等求解方程组求得权重排名向量的增量。

在这里，更新权重排名向量时只处理变化的数据。首先根据先求得权重排名增量，然后与已有的权重排名向量结合，就能求得更新后的排名向量。算法如下：

算法 6.6: 权重排名微分增量更新算法

输入：增量转移张量展开矩阵 ΔM，原有转移张量展开矩阵 M，原有排名向量 w；

输出：更新后的属性权重排名向量 w'。

1: 用转移张量展开矩阵 ΔM，原有转移张量展开矩阵 M，原有排名向量 w 计算系数矩阵 A；

2: 用转移张量展开矩阵 ΔM，原有转移张量展开矩阵 M，原有排名向量 w 计算向量 b；

3: 假定矩阵 A 非奇异，计算矩阵 A 的逆矩阵；

4: 通过公式 $\Delta w = A^{-1}b$ 求得属性权重排名向量增量 Δw；

5: 将向量增量 Δw 和原有的排名向量 w 进行加法合并；

6: 返回合并得到的更新后的属性权重排名向量 w。

通过对上述算法的分析可知，因为没有迭代过程，可以避免高阶幂法的不收敛情况，其复杂度为 $O(n^3)(N=2)$，以及 $O(n^N)(N\geq 3)$。但是由于舍去影响小的微分项可能存在误差，因此由于误差累积无法确保高准确率，且微分项在实际应用中必须足够小，该方法只适用于流式增量学习。

6.3.3 增量式 K-medoids 聚类算法

在 6.2 节中本文研究了密度峰值聚类的增量更新方法，然而，根据 3.4.1.1 节的张量多聚类算法的复杂度分析，假设 m 为单个对象张量元素数，n 为对象张量个数，该算法复杂度分布如表 6.2 所示：

表 6.2 张量多聚类算法的复杂度

TMC 的步骤	关联张量	权重学习	距离计算	聚类
复杂度	$O(mn)$	$O(m)$	$O(m^2n^2)$	$O(n^2)$

由表 6.2 可以看出，张量多聚类的耗时主要在距离的计算，而距离天然以相异度矩阵的形式进行增量，图 6.2 是相异度矩阵的增量示意图，如果使用普通的聚类算法

或增量聚类算法，则相异度矩阵必须用距离填满才能进行聚类，那么如何提高效率呢？本文考虑在增量计算中尽可能减少距离的计算数量。而 6.1 节提出的增量密度峰值聚类增量更新方法，虽然在同类增量聚类方法中可以获得较好的性能，但是它要求新增对象必须与所有原有对象计算距离，因此需要寻求可以减少距离计算数量的其它增量式聚类方法。这里，本文改用增量式 K-medoids 聚类方法，根据文献[106]中一种简单快速的 K-medoids 算法，本文研究设计了相应的增量更新 K-medoids 算法，在多聚类增量时不需要计算全部距离，从而有效提高张量多聚类的增量更新算法效率。

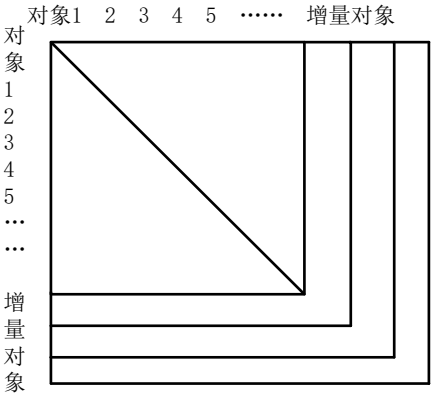


图 6.2 距离增量示意图

非增量式 K-medoids 聚类方法的中心思想是通过迭代更新中心点的位置，最终使每个类簇内部的距离和达到最小。

算法 6.7: 增量式 K-medoids 聚类算法

输入：原有对象之间的距离 $d_{ij}(0<i<j\le m_1)$。

增量输入：新增对象和原有中心点之间的距离 d_{ij}。

输出：最新的聚类结果 *cl*。

1：选择中心，对原有对象按照 $v_j=\sum_{i=1}^{n}\frac{d_{ij}}{\sum_{l=1}^{n}d_{il}}, j=1,\ldots,n$，计算每个对象的 v，挑选出前 k 个 v 最小的对象作为中心。

2：归类，按照就近原则，每个非中心的对象（包括原有对象以及新增对象）归入离其最近的中心点所在类簇。

3：更新中心点，对于每个类簇内的每个对象（包括原有对象以及新增对象），计算其与同类簇其它中心点的距离和，每个类簇内拥有最小距离和的点成为类簇的新中心点。

4：判断迭代是否终止，如果每个类簇内的新中心点的距离和都与老中心点一致，则迭代终止。否则返回第 2 步继续迭代。

5：返回最新的聚类结果 *cl*。

算法在增量过程中如果需要对 *k* 进行增大或缩小，只需要把在第 2 步中产生的总的距离归一化后排序，按照第 1 步的方法重新产生中心点继续迭代即可。该算法的复杂度为 $O(nk+(\frac{n}{k})^2)$，对于新增对象，只需要计算和所有中心的距离以及被分配类簇内与其它对象的距离即可，可以有效提高增量张量多聚类算法的效率。

6.3.4 增量式张量多聚类算法

根据前面对张量多聚类算法主要组成部分的增量更新方法的研究与分析，本文提出完整的增量式张量多聚类算法（Incremental TMC，ITMC），具体描述如下：

算法6.8：增量式张量多聚类算法

输入：m_1 个原有对象张量。

增量输入：新增对象张量，对多聚类结果的查询*query*。

输出：最新的多聚类结果 $cl_1^{'},cl_2^{'},\ldots,cl_r^{'}$。

1：设置对象的个数 $m \leftarrow m_1$；

2：执行循环：

3：　新增一个对象，$m \leftarrow m+1$；

4：　根据情况选择算法6.6或者算法6.7计算增量更新的属性排名向量 $\mathrm{w}_1^{'},\mathrm{w}_2^{'},\ldots,\mathrm{w}_N^{'}$；

5：　增量更新的属性排名向量做外积得到增量更新的权重张量 $\mathcal{T}_w^{'}$；

6：　for *l*=1 to *k* do

7：　　计算新增对象与原有第*l*个多聚类结果中心点的可选择加权张量距离；

8： 利用算法6.8进行增量式聚类，得到更新的第l个聚类结果$cl_l^{'}$；

9： end for

10：结束循环

11：返回最新的聚类结果$cl_1^{'},cl_2^{'},\ldots,cl_r^{'}$。

6.4 实验分析

为了评价所提出的增量式密度峰值聚类方法和增量式多聚类方法的性能，本文在多个数据集上对两种算法进行了实验评估。所有实验都是在 3.4GHz Intel Core i5-7500 CPU, 16GB 内存，1TB 硬盘的 Win10 操作系统上进行的。

6.4.1 密度峰值聚类的增量更新方法性能评价

针对所提出的增量式密度峰值聚类方法进行性能评估，主要通过在多个数据集上对提出的方法和以前的两种增量密度峰值聚类方法（ICFSKM[107]和 ICFSMR[108]）进行实验对比。本节首先介绍数据集和评价指标，然后从聚类质量和时间成本两个方面对提出方法进行了验证。

6.4.1.1 数据集

这些数据集包括密度峰值聚类原始论文[78]中用于可视化的 1 个人工数据集和 3 个真实数据集（Iris[6]、Wireless[6] 和 Olivetti faces[7]）。

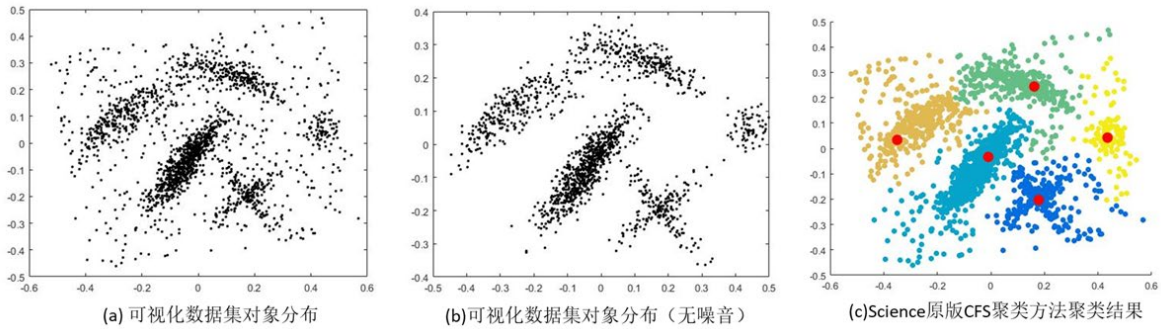


图 6.3 人工数据集可视化图

人工数据集的可视化如图 6.3 所示，图(a)对应的是人工数据集的一些噪声分布，

[6] http://archive.ics.uci.edu/ml/
[7] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

而无噪声的版本如图(b)所示，5 个类簇非常清晰。最后，原始密度峰值聚类方法带噪声的聚类结果如图(c)所示，可以作为数据集的官方标签。

Iris 也称鸢尾花卉数据集，包含 150 个数据，分为 3 类，每类 50 个数据，每个数据包含花萼长度，花萼宽度，花瓣长度，花瓣宽度 4 个属性。Wireless 数据集包含 2000 个数据，每个数据包含 7 个属性，每个属性是智能手机上观察到的 wifi 信号强度。Olivetti faces 数据集包含 40 个不同的主题，每一个都有 10 张不同的图片。其中有些受试者在不同的时间拍摄照片，主要改变了灯光、面部表情（睁开/闭上眼睛、微笑/不微笑）和面部细节（戴眼镜/不戴眼镜）。所有的照片都是在黑暗的均匀背景下拍摄的，受试者都是直立的，正面的姿势（容忍一些侧移）。

此外，为了模拟实际应用中数据增量到达场景，在实验中将每个数据集分成 5 份，初始对象 20%，每次到达对象数 20%。表 6.3 描述了实验中使用的增量数据集的详细信息。

表 6.3 实验数据集

数据集	对象数	属性数	初始对象数	到达对象数
可视化数据集	2000	2	400	400 * 4
Iris	150	4	30.	30 * 4
Olivetti Faces	400	92 * 112	80	80 * 4
Wireless	2000	7	400	400 * 4

6.4.1.2 评价方法

对人工数据集的可视化结果可以进行直观判断，对于真实数据集本实验采用了两种常用的评价指标归一化互信息（Normalmutual Information, NMI）[107]和时间成本，来衡量所提出的增量密度峰值聚类方法和其他两种增量方法的性能。其中 NMI 的定义如下：

$$NMI = \frac{\sum_{c=1}^{k}\sum_{p=1}^{m} n_c^p \log(\frac{n \cdot n_c^p}{n_c n_p})}{\sqrt{(\sum_{c=1}^{k} n_c \log(\frac{n_c}{n}))(\sum_{p=1}^{m} n_p \log(\frac{n_p}{n}))}}, \tag{6.12}$$

其中，n 是总的对象数，n_c 和 n_p 分别第 c 个类和第 p 个类的对象数，n_c^p 表示在类 c

和 p 中共同出现的对象数。NMI 主要用来度量两个类的相似度，分子表示互信息（MI），分母用来做标准化，因此 NMI 值越大聚类质量越好。

6.4.1.3 聚类结果可视化

本小节展示了直观可视化的聚类质量。为了模拟增量计算场景，需将所有数据集分成五个相等的部分，依次输入三种算法。每个增量算法连续产生 4 个批量增量对应的 5 个聚类结果。在可视化图中，实验用不同的颜色表示不同的类簇，红色圆点表示该点是类簇的中心。

图 6.4 展示了 ICFSKM 方法在人工数据集的可视化结果。其中，图(a)展示了与原始 CFS 不同的全局聚类结果，这是因为 ICFSKM 方法提出了一种新的初始化方法来代替原来的 CFS，然而，观察这幅图，新的初始化方法比原来的 CFS 效果更差。接下来的图(b)到图(f)对应的是 4 次批量增的聚类结果，显然，ICFSKM 方法的聚类结果并不理想，主要是因为后面的聚类结果继承了前面的错误。在图(f)中，噪声被简单地分成了两个类簇，而这些类簇附近的点和它们并不相似。综上所述，从人工数据集的可视化效果来看，ICFSKM 方法聚类的效果较差。

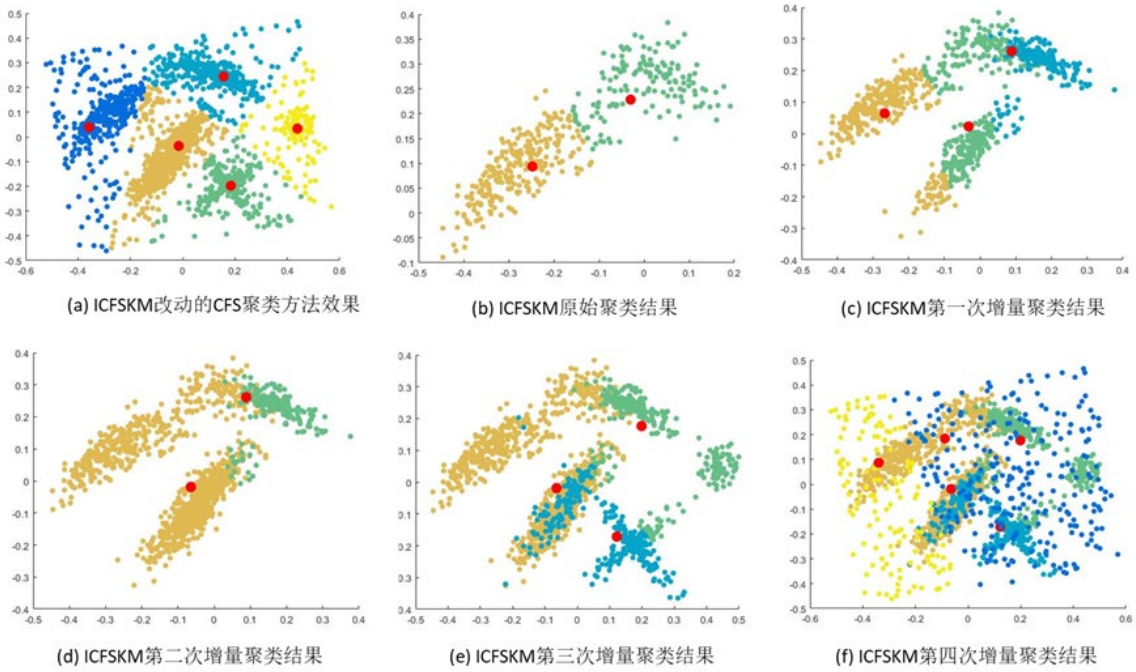


图 6.4 ICFSKM 方法在人工数据集的可视化结果

图 6.5 展示了 ICFSMR 方法在人工数据集的可视化结果。因为 ICFSMR 方法引入

了代表点的概念，这意味着它们可以代表自己的类簇，在可视化时用方形点来表示代表点。如图(a)到图(f)所示，ICFSMR 的增量聚类结果略好于 ICFSKM，因为在增量过程总有些错误被修正而不是被继承下去，而且 ICFSMR 对噪声的容忍度明显更好。但是，在该方法的可视化结果中仍然存在很多错误，比如将不同的类簇识别为一个类簇，将一个类簇中的点划分为多个类簇以及代表点的表示效果不佳等。综上所述，在人工数据集上，ICFSMR 方法的可视化效果优于 ICFSKM 方法，但仍不完善。

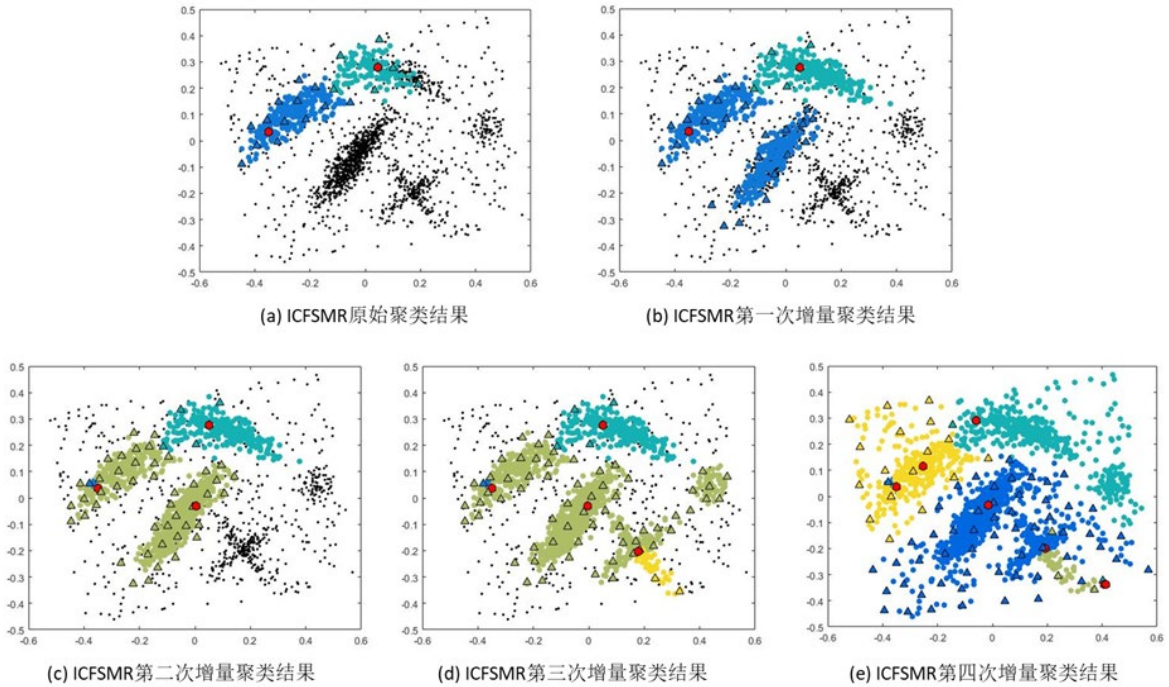


图 6.5 ICFSMR 方法在人工数据集的可视化结果

最后，本文提出的 ICFS 方法的结果如图 6.6 所示。从图(a)到图(e)可以明显看出，增量聚类结果几乎是完美的。增量聚类过程中唯一的错误出现在图(c)中，其中一个类簇被错误的分成了两个类簇，但是在图(d)中它被立即修复。事实上，在原始的 CFS 方法中也无法避免出现同样的错误。

6.4.1.4 聚类准确率

除了人工数据集的可视化结果外，利用这三种方法分别对三个数据集进行处理，其结果的 NMI 值如表 6.4 所示。每个实验给出 5 个 NMI 值对应 5 次聚类结果。通过对 NMI 值的观察可以看出，几乎在所有的情况下，ICFS 方法的 NMI 都是最高，即使在少数不是最好的情况下，其 NMI 值也非常接近最高。这样的结果表明，ICFS 方法

能够产生高质量的聚类结果，具有更好的鲁棒性。

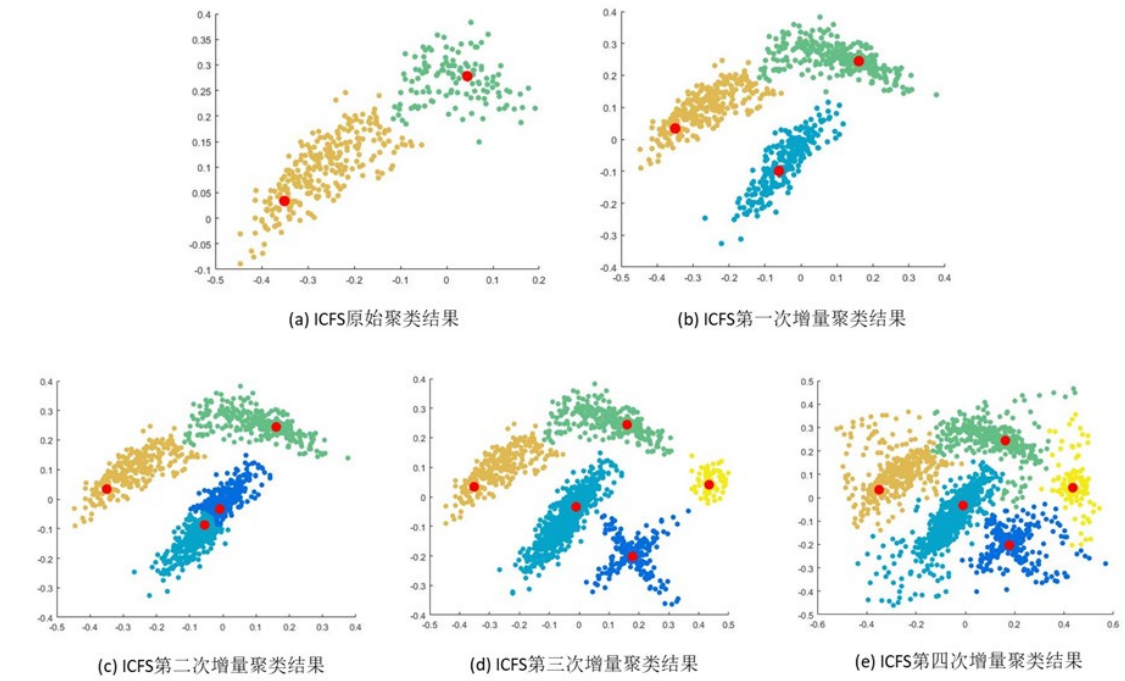


图 6.6 ICFS 方法在人工数据集的可视化结果

表 6.4 三种方法在不同数据集的 NMI 对比

数据集	三种方法	原始	第一次	第二次	第三次	第四次
Iris	ICFSKM	0.59	0.7	0.724	0.737	0.75
	ICFSMR	0.9	0.76	0.74	0.72	0.63
	ICFS	**0.9**	**0.88**	**0.76**	**0.76**	**0.87**
Olivetti Faces	ICFSKM	0.513	0.526	0.559	0.545	0.509
	ICFSMR	0.6	0.55	0.6	0.56	0.55
	ICFS	**0.6**	**0.58**	**0.56**	**0.57**	**0.6122**
Wireless	ICFSKM	0.8	0.624	0.579	0.593	0.749
	ICFSMR	0.954	0.708	0.724	0.498	0.495
	ICFS	**0.954**	**0.81**	**0.956**	**0.742**	**0.882**

6.4.1.5 执行时间

一般来说，针对同一问题，不同的方法在结果的质量和成本之间需要存在权衡。然而，ICFS 方法的成本并不比其他两种方法高。

如表 6.5 所示，在相同的场景列出所有的时间成本，表中均为累加时间，例如第

四批的时间是从第一批开始到第四批结束的全部时间。在大多数情况下，ICFS 方法的时间开销最小。随着数据量的增加，ICFS 在所有实验中时间成本的趋势也都较低。

表 6.5 三种方法在不同数据集的运行时间(ms)对比

数据集	三种方法	原始	第一次	第二次	第三次	第四次
Iris	ICFSKM	24	28.5	33.6	38.5	44
	ICFSMR	35	39.1	43.7	47.5	52.5
	ICFS	**12.8**	**22.9**	**32.7**	**41.2**	**47.2**
Olivetti Faces	ICFSKM	144	313	532	671	1840
	ICFSMR	178	274	388	580	700
	ICFS	**69**	**169**	**245**	**320.6**	**427.4**
Wireless	ICFSKM	193	618	1342	2308	3526
	ICFSMR	166	526	949	1498	2173
	ICFS	**102**	**358**	**768**	**1308**	**1991**

6.4.2 张量多聚类的增量更新方法性能评价

本节主要对所提出的增量式张量多聚类方法进行性能评估，通过在多个数据集上对提出的方法进行实验验证。接下来首先介绍数据集和评价指标，然后主要从聚类质量和时间成本两个方面对提出的方法进行了评估。

6.4.2.1 数据集

在本小节的实验中，本文分别用 CMUPIE Face[8]和第 4.4.1.3 节中的智能电网数据集对提出的方法进行评估，表 6.6 描述了实验中使用的增量数据集的详细信息。

表 6.6 实验数据集

数据集	对象集	属性集	初始对象数	到达对象数
CMUPIE Face	11554	64*64	48	48
智能电网	1454	24*24*5*11	50	50

6.4.2.2 评价方法

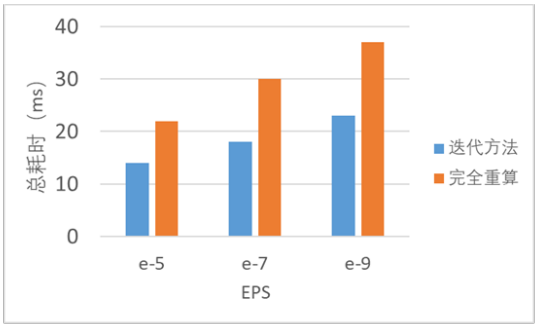
同 6.4.1.2，本小节本实验仍然使用 NMI 来评价聚类质量，在此不再赘述。

6.4.2.3 权重学习迭代增量更新方法性能评估

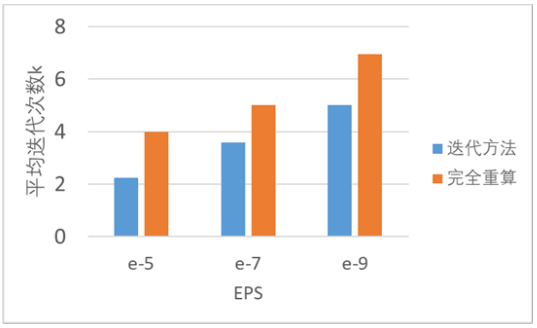
本节实验首先对权重学习迭代增量更新方法进行性能评估，分别从总耗时和平均

[8] https://www.cs.cmu.edu/~face/database.htm

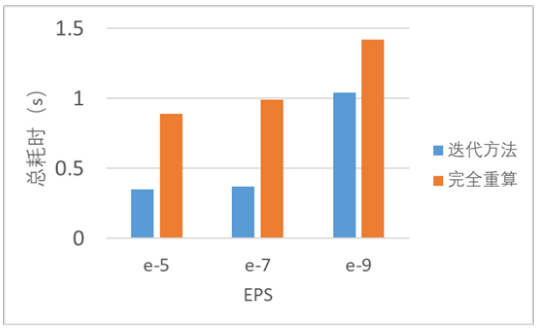
迭代次数来展示实验结果。通过将该方法分别用于 CMUPIE Face 和智能电网数据集，素性修正参数α取 0.5。从图 6.7 中可以看出，在两个数据集上，权重学习迭代增量更新方法可以使用比完全重算更少的迭代次数收敛到 EPS 误差，因此总耗时也大幅度低于增量时算法完全重算。



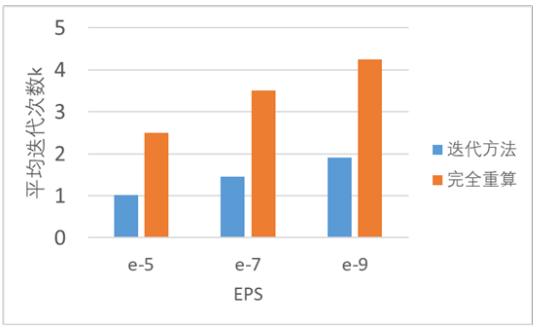
(a) CMUPIE Face上迭代方法和重算总耗时比较(α=0.5)



(b) CMUPIE Face上迭代方法和重算迭代次数比较(α =0.5)



(c)智能电网上迭代方法和重算的总耗时比较(α =0.5)



(d) 智能电网上迭代方法和重算的迭代次数比较(α =0.5)

图 6.7 迭代增量学习方法的性能

6.4.2.4 两种权重学习增量更新方法性能对比

本节对两种权重学习方法分别从精确度和执行时间上进行对比。首先，将两种权重学习增量更新方法在结果的精确度上进行实验对比，两种方法分别用于智能电网数据集，素性修正参数α取 0.5。

表 6.7 智能电网数据集上不同 EPS 下两种方法的误差比较(α =0.5)

EPS	e^{-5}	e^{-7}	e^{-9}	e^{-10}
迭代方法	$1.27e^{-7}$	$7.27e^{-10}$	$1.92e^{-13}$	0
微分方法				$2.69e^{-4}$

从表 6.7 中可以看出，微分方法只需在初始化时使用一次高阶幂法，因此给这次高阶幂法分配了更精确的 EPS(e^{-10})，然而其结果误差仍然远大于使用更不精确 EPS 的迭代方法的结果。

其次，图 6.8 展示了两种权重学习增量更新方法在执行时间上的对比。本文将两种方法分别用于智能电网数据集，素性修正参数α取 0.5。从图中可以看出，虽然迭代的方法耗时随迭代次数 k 呈线性增长，而微分方法执行一次，但是迭代方法在 k 接近 50 时耗时才超越微分方法，而如此高的 k 一般是不现实的。

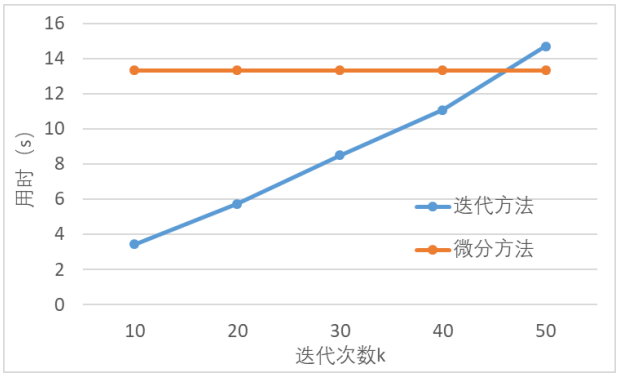
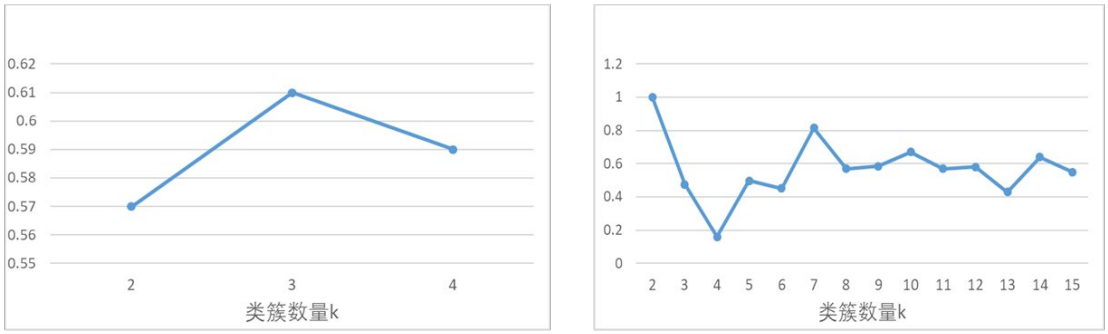


图 6.8 迭代增量学习方法的性能

综上所述，迭代方法在精确度和效率上均胜过微分方法，但微分能够避免迭代不收敛等问题，微分方法可以作为迭代方法的补充，在特定场景下仍然适用。

6.4.2.5 聚类准确率

图 6.9 展示了增量式张量多聚类方法 ITMC 和非增量式张量多聚类 TMC 在 NMI 上的对比。本文将两种方法分别用于 CMUPIE Face 和智能电网数据集，素性修正参数α取 0.5。从图中可以看出，ITMC 产生的结果与 TMC 之间的 NMI 均在 0.6 左右，较为相似但有一定区别。



(a) CMUPIE Face上ITMC和TMC结果间的NMI(α =0.5)　(b) 智能电网上ITMC和TMC结果间的NMI (α =0.5)

图 6.9 增量与非增量张量多聚类方法 NMI 对比

图 6.10 是以 CMUPIE Face 数据集的正确结果为基准分别计算的 NMI，可以看

出 ITMC 结果质量略高于 TMC，这主要是由于增量过程中 ITMC 遍历了更多聚类的解，比 TMC 更能避免陷入局部最优解。需要说明的一点是，这里所有的实验本文都做过三次，分别加入的是迭代方法、微分方法和完全重算得到的权重，但是三次实验的聚类结果几乎没有区别，相互之间的 NMI 都为 1，这也说明了权重部分的增量方法误差已经足够小，不足以引起后续聚类的误差。

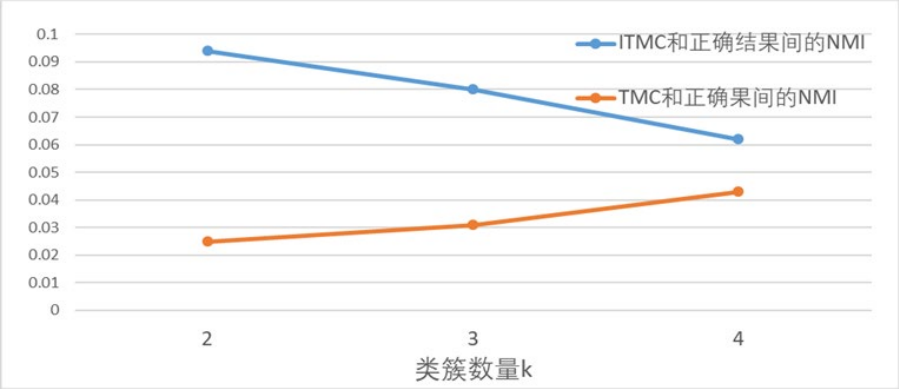
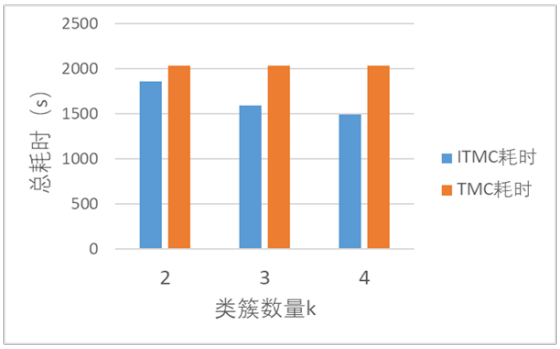


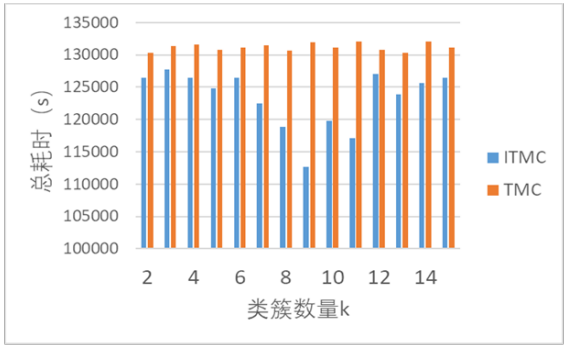
图 6.10 CMUPIE Face 上 ITMC 和 TMC 相对于正确结果的 NMI (α =0.5)

6.4.2.6 执行时间

最后，在增量式张量多聚类方法 ITMC 和非增量式张量多聚类 TMC 在执行时间上进行实验对比。同样，本文将两种方法分别用于 CMUPIE Face 和智能电网数据集来对比执行时间，素性修正参数 α 取 0.5。从图 6.11 可以看出，ITMC 总耗时明显低于 TMC，这主要是由于 ITMC 方法计算的可选择加权张量距离的数量少于 TMC。



(a) CMUPIE Face上ITMC和TMC的时间比较(α =0.5)



(b) 智能电网上ITMC和TMC的时间比较(α =0.5)

图 6.11 ITMC 和 TMC 执行时间比较(α =0.5)

6.5 本章小结

本章主要针对大数据数量动态增长的特点，研究了增量密度峰值聚类方法和增量式张量多聚类方法。

首先，在原始 CFS 算法的基础上，研究其三个主要组成部分的更新算法：局部密度更新方法、基于红黑树的对象依赖重连方法、中心更新方法和类簇分割更新方法，对原有聚类结果和中间结果进行调整。为了评价所提出的 ICFS 方法的性能，本文在一个人工数据集和三个真实数据集上进行了实验，并与其他两种方法进行了比较。在直观的可视化中，ICFS 方法产生的结果最好，其他两种算法的结果都存在明显的错误继承和噪声容忍度差的缺陷。在 NMI 指数中，ICFS 在大多数情况下 NMI 值最高，在其余情况下，其 NMI 值也非常接近于最高的 NMI 值，这证明 ICFS 具有较好的鲁棒性结果。从时间成本方面看，ICFS 的时间成本在大多数情况下是最小的，在其他情况下，它的时间成本也非常接近最小的。随着数据量的增加，ICFS 在所有实验中时间成本的趋势也都较低。

最后，本章研究增量式多聚类方法，提出两种权重学习方法以及设计增量式 K-medoids 聚类方法来减少距离的计算，从而提高整体算法的效率。从实验结果来看，权重学习迭代增量更新方法总体优于权重学习微分增量方法，但是作为迭代方法的补充，基于微分的方法还是有一定的应用场景。

7 总结与展望

多聚类可以从数据的不同观点产生多个不同的聚类结果，有利于从多方面揭示隐藏在数据中的不同结构。但目前的研究大多针对小规模、单领域数据集，并且聚类结果难以解释，无法根据上下文情境变化实现多模态的聚类，且算法大多面向具体应用，难以扩展到其他领域，缺乏通用性。同时，随着大数据时代的到来，大数据所具有的“4V”特征，即规模庞大(Volume)、类型多样(Variety)、增长快速(Velocity)和价值密度不均(Value)等，对大数据环境下的多聚类方法研究提出了新的挑战，并将对现有多聚类分析的计算模式、理论和方法产生深远的影响。

本文从大数据的四大特点出发，针对大数据多聚类理论研究，在对象张量表示模型的基础上，提出了基于张量的大数据多聚类方法，并围绕张量多聚类方法，提出了云端隐私保护的安全计算方法、基于张量链分解的并行计算方法、针对动态数据的增量更新方法，实现云端安全、高效的多模态聚类，从而挖掘不同情境下数据隐含的不同类簇。本文所提出基于张量的大数据多聚类及其安全和高效方法可为多聚类理论研究提供新的有益思路，同时也将促进多聚类分析在大数据时代的应用及发展。

7.1 主要成果

本文提出基于张量的大数据多聚类方法，并在此基础上实现云端安全的、分布式并行、增量式的大数据多聚类，具体研究成果包括：

1.基于张量的大数据多聚类方法

针对大数据的来源多样、特征高维、关系复杂、规模庞大和生成快速等问题，并充分发挥现有多聚类方法的优势，开展了大数据环境下面向大规模多源异构数据的多聚类理论研究。首先，提出了一种基于多线性属性排名的权重学习方法，用来度量所有特征空间中属性组合的重要性，进而在张量对象统一表示模型的基础上，提出了基于张量的多聚类方法；其次，为了在计算距离时能够通过将所选择的特征与未选择的特征完全分离来提高多聚类质量，提出了基于相似度矩阵加权平均的多聚类方法和基

于可选择加权欧式距离的多聚类方法。然而，这两种方法没有考虑不同属性的影响，也没有考虑如何去除噪声和冗余，特别是对于高维数据。因此，本文进一步通过扩展它们的优势，提出了基于张量分解的多聚类方法。同时，为了提高基于张量分解的多聚类方法的性能，设计了一种用于各特征空间属性重要性度量的多关系属性排名方法。本文所提出的多聚类方法，特别是基于张量和张量分解的多聚类方法，能够获得较高质量的多聚类结果，同时冗余度较低，能够满足大数据应用的不同需求，为大数据分析和应用提供增强的知识提取和服务。

2.云端安全的张量多聚类方法

在云计算环境中，为保护用户隐私，研究张量多聚类算法的云端安全计算模式。设计了混合云模型下云端安全的张量多聚类分析和服务框架，提出了一种云端安全的高阶密度峰值聚类方法，进而提出了云端安全的张量多聚类方法，以及相关的多种安全子协议，包括安全指数、安全排序、安全属性排名、安全平方张量距离以及安全可选择加权张量距离协议。在提出的安全计算方案中，所有的聚类计算任务都是在云端实现的，而云端不会公开或推断出任何机密信息，这不仅提高了聚类的效率，而且保护了用户的隐私。客户端无需参与任何聚类计算，对用户来说是非常轻量级的。两种安全聚类方法都可以在半诚实模型下实现基于 Paillier 加密体系的完整安全协议，同时可以保证 100%的聚类准确率，且算法具有较高的扩展性，这些对于大数据分析和处理都是非常重要的。

3.基于张量链分解的张量多聚类及其并行计算方法

针对维度灾难和高效计算问题，提出一套基于张量链分解的张量多聚类及其并行计算方法。首先，基于张量链分解形式下各张量基本运算的计算规律，在对象张量的张量链分解形式下，直接构建关联张量链、转移张量链、属性组合权重张量链，提出基于张量链分解的多线性属性组合权重学习算法和可选择加权张量距离，进而提出基于张量链分解的张量多聚类方法，从而实现在张量链分解的形式下完整的张量多聚类过程，并能保证甚至提高聚类结果的准确性；其次，在云计算分布式环境中，依据节点计算能力和通信能力设计高效的分布式并行计算框架，通过研究张量链核分配机制、核调度策略及核运算的并行策略，提出基于张量链核的分布式并行策略，充分利

用张量网络并行计算优势提高张量多聚类算法的并行效率。

4.张量多聚类的增量式更新方法

针对大数据动态增长带来的大量重复计算问题，开展了增量式张量多聚类研究。首先，在原始密度峰值聚类算法的基础上，提出了与其相关的三个主要组成部分的更新算法：局部密度更新方法、基于红黑树的对象依赖重连方法、聚类中心更新方法和类簇分割更新方法，从而对原有聚类结果和中间结果直接进行调整；其次，针对张量多聚类方法，分别提出了基于迭代的属性权重学习方法和基于微分的属性权重学习方法，并基于一种简单快速的 K-medoids 算法，设计了相应的增量式 K-medoids 算法，使得在多聚类增量时不需要计算全部距离，从而有效提高张量多聚类的增量更新算法效率。从实验结果来看，本文提出的增量式密度峰值聚类和张量多聚类方法不仅能保证增量更新的聚类准确率，而且能够极大程度地提高聚类分析中数据动态增量更新维护的效率。

7.2 研究展望

本文基于张量模型构建了大数据多聚类方法并实现了云端安全以及分布式、增量式的张量多聚类高效计算。然而，在基于张量的大数据多聚类及其安全和高效方法的研究中，仍然存在许多亟待解决的问题，张量多聚类在各领域大数据分析中的进一步应用还需要新的理论与方法。下面探讨未来的一些研究工作，概括如下：

1. 基于张量的多聚类建模和相似性度量

张量的优势在于能够融合多源异构数据，在更高阶和更高维表示各种各样的数据，能够实现大数据的多模态分析。但同时使用张量又会带来一些亟待解决的问题，如给融合后的特征空间的选择带来不能完全分离导致无法保证聚类结果高准确率的问题，以及使用目前的张量分解方法及张量距离相似度度量方式带来的计算效率的问题。因此，在进一步的研究中，需要寻找其他更有效的基于张量代数理论的多聚类模型和方法，并将这些多聚类方法在更多真实的大数据应用中进行评估；其次对现有方法进行改进也是十分必要的。

2. 隐私保护的安全张量多聚类方法

云端安全的张量多聚类方法在数据可用性、数据隐私保护程度、聚类准确率和可扩展性方面都有很好的表现，但是，从目前安全多聚类方法的性能评估来看，效率方面还存在很大的提升空间。因此，考虑聚类算法中比较运算较多，未来可以结合其他有效的隐私保护方法，而不是仅采用同态加密的方法，比如乱码电路技术比较运算方面具有一定优势，可以设计同态加密和乱码技术结合的安全多聚类方法。

3. 张量多聚类分布式并行计算方法

张量链分解形式的优势在于解决维度灾难问题，数据可以分布式压缩存储并且在该分解形式下直接做各种运算，但这种直接运算同时又会带来大量更加复杂的计算。因此，如何极大程度的提高该分解形式下的并行计算效率是需要重点解决的问题。未来，在当前研究的基础上，可以进一步研究核内并行与核间并行的混合并行策略，并研究如何通过减少同步点数量的方法最大限度减少通信量。此外，可以结合 QTT 对高维的阶继续进行扩阶降维，并基于此设计更深层次的并行策略，从而进一步提高并行计算的效率。

4. 张量多聚类增量式动态更新方法

增量式聚类的关键是动态更新的效率和准确率，就目前的增量式密度峰值聚类和多聚类方法来看，还可以进一步寻找新的方法来改进效率和准确率。比如改进聚类中心的选择、改进截断距离的选择等，从而修正增量密度峰值聚类中偶尔出现的中间误差。此外，可以结合马尔可夫链预测模型，在已产生多聚类结果的基础上，结合对象流的历史数据序列，为当前对象建立转移张量，从而在未来对象流更新之前预测多聚类结果，进一步提高张量多聚类的增量更新算法效率。

综上所述，为解决大数据环境下多聚类方法构建及其安全和高效计算等方面的关键问题，需要基于高阶张量代数理论提出新的方法，更进一步地解决大数据环境下张量多聚类的安全和高效计算难题，奠定后续研究的理论基础，同时也为促进张量多聚类在大数据时代的应用及发展提供理论支撑和有效保障。

致 谢

喻家山，东湖畔，喻园五年，感悟颇多。在我博士研究生求学阶段，回想当初的迷茫、困惑、不知所措，到后来的从容、开心、略有小作，才发现原来自己是那么的幸运……幸得师长的指导、同学的帮助、亲人的关爱和朋友的鼓励，让我坚守学术，坚持到底，也让我最终体会到学术进步后的喜悦。此时内心油然而生无限的感激之情！

感谢！发自肺腑！！

衷心感谢我的恩师杨天若教授悉心指导和亲切关怀。五年如一日，我的成长离不开杨老师的循循善诱和鞭策鼓励；杨老师不分昼夜、节日无休地多次与我就学术中许多核心问题作深入细致地探讨，给我提出切实可行的指导性意见，并细致全面地修改我的小论文。在本文选题、科学研究、实验设计和论文撰写等方面，杨老师都给予了高屋建瓴的指导和真知灼见的建议，也饱含了杨老师无数的心血和汗水。同时杨老师在治学上严谨细致、一丝不苟的态度，在工作上大胆创新、敢为人先的作风，一直是我学习的榜样，也使我受益终生。

感谢河南大学计算机与信息工程学院的领导与同事，在我博士研究生的学习阶段前后，郑逢斌教授、沈夏炯教授、马骏教授等都给予了最大程度的支持，让我得以安心在学校完成博士学业和博士科研课题。

特别感谢林曼教授，谢谢您利用宝贵的时间在论文组织、写作、语言等方面事无巨细地给我指导。感谢丘锡生教授、Jamal Deen教授、董冕雄教授、李瑞轩教授、邱美康教授、朱大开教授等，谢谢你们在我博士论文开题时提出诸多宝贵建议，并提供许多可行的科研方法，谢谢你们经常为我们带来精彩的前沿学术报告，为我们的学术难题释疑解惑，我学术上的进步与您们的鼓励、关怀和帮助分不开。

感谢实验室罗常青老师、郝飞老师、王蔚老师、莫益军老师、崔金华老师、余辰老师、谢夏老师、韩建军老师、骆杨老师在科研和生活中给予的帮助。感谢研究生孙佳宇同学和张荣皓同学在我论文完成过程中给予的无私帮助，一次次修改、调试程序，正是你们默默的付出才有了理想的实验结果。感谢实验室曾晶、匡立伟、王晓康、刘

华中、王普明、冯君、张顺利、杨静、高源、尹德祥、朱进、李锦涛以及其他所有的师弟师妹们，我们一起学习、讨论、争辩、成长，在一个美好的氛围中度过我们最为重要的人生阶段。

感谢家人和朋友们一直以来对我的陪伴和照顾，在漫长而又艰辛的求学生涯中，你们的理解和支持为我缓解了很多压力，给我带来了前进的动力。

特别感谢评阅专家对本文的批评与指正!

最后向所有给予我关心、支持、帮助和鼓励的师长、同学、亲朋好友表示深深的感谢，祝你们工作顺利，身体健康!

赵雅靓

于武汉·华科·喻家山

2019年5月30日

参考文献

[1] J. Bailey. Alternative Clustering Analysis: A Review, Data Clustering: Algorithms and Applications, CRC Press, 2013: 533–548.

[2] E. Müller, I. Assent, S. Günnemann, T. Seidl, and J. Dy. MultiClust Special Issue on Discovering, Summarizing and Using Multiple Clusterings, Machine Learning, 2015, 98(1-2): 1.

[3] Z. Liu, D. Yang, D. Wen, W. Zhang, and W. Mao, Cyber-Physical-Social Systems for Command and Control, IEEE Intelligent Systems, 2011, 26(4): 92–96.

[4] E. Müller, S. Günnemann, I. Färber, and T. Seidl. Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data, in Proceedings of the 28th International Conference on Data Engineering (ICDE'12), 2012: 1207-1210.

[5] P. Symeonidis, A. Papadimitriou, and Y. Manolopoulos. Geo-Social Recommendations Based on Incremental Tensor Reduction and Local Path Traversal, in Proceedings of the 3rd ACM SIG Spatial International Workshop on Location-Based Social Networks (LBSN'11), 2011: 89-96.

[6] J. Sun, D. Tao, and C. Faloutsos. Beyond Streams and Graphs: Dynamic Tensor Analysis, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), 2006: 374-383.

[7] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos.Tag Recommendations Based on Tensor Dimensionality Reduction, in Proceedings of the 2nd ACM International Conference on Recommender Systems (RecSys'08), 2008: 43-50.

[8] 廖志芳, 李玲, 刘丽敏, 李永周. 三部图张量分解标签推荐算法. 计算机学报, 2012, 35(12):2625-2632.

[9] J. D. Carroll and J. J. Chang. Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of “Eckart-Young” Decomposition, Psychometrika, 1970, 35: 283-319.

[10] L. R. Tucker. Some Mathematical Notes on Three-mode Factor Analysis, Psychometrika, 1966, 31: 279-311.

[11] L. Grasedyck. Hierarchical Singular Value Decomposition of Tensors, SIAM Journal on Matrix Analysis and Applications, 2010, 31(4): 2029-2054.

[12] I. V. Oseledets. Tensor-train Decomposition, SIAM Journal on Scientific Computing, 2011, 33(5): 2295-2317.

[13] I. V. Oseledets. Approximation of 2d×2d Matrices Using Tensor Decomposition, SIAM Journal on Matrix Analysis and Applications, 2010, 31(4): 2130-2145.

[14] C. Zhang, H. Fu, S. Liu, G. Liu and X. Cao. Low-Rank Tensor Constrained Multiview Subspace Clustering, in Proceedings of the IEEE International Conference on Computer Vision (ICCV'15), 2015: 1582-1590.

[15] R. Xia, Y. Pan, L. Du and J. Yin. Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition, in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'14), 2014: 2149-2155.

[16] X. Zhang, L. Zong, X. Liu and H. Yu. Constrained NMF-Based Multi-View Clustering on Unmapped Data, in Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), 2015: 3174-3180.

[17] J. Liu, C. Wang, J. Gao and J. Han. Multi-view Clustering via Joint Nonnegative Matrix Factorization, in Proceedings of the SIAM International Conference on Data Mining (SDM'13), 2013: 252-260.

[18] H. Wang, F. Nie and H. Huang. Multi-View Clustering and Feature Learning via Structured Sparsity, in Proceedings of the 30th International Conference on Machine Learning (ICML'13), 2013: 352-360.

[19] F. Nie, J. Li and X. Li. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification , in Proceedings of the International Joint Conferences on Artificial Intelligence, 2016.

[20] L. Pradhan, C. Zhang and P. Chitrakar. Multi-view Clustering in Collaborative Filtering Based Rating Prediction, in Proceedings of the 10th IEEE International Conference on Semantic Computing (ICSC'16), 2016:250-253.

[21] J. Hu, Q. Qian, J. Pei, R. Jin and S. Zhu. Finding Multiple Stable Clusterings, in Proceeding of the 15th IEEE International Conference on in Data Mining (ICDM'15), 2015: 171-180.

[22] D.T. Truong and R. Battiti. A Flexible Cluster-Oriented Alternative Clustering Algorithm for Choosing from the Pareto front of Solutions, Machine Learning, 2015, 98(1-2): 57-91.

[23] K.N. Kontonasios and T.D. Bie. Subjectively Interesting Alternative Clusterings, Machine Learning, 2015, 98(1-2): 31-56.

[24] S. Günnemann, I. Färber and T. Seidl. Multi-view Clustering Using Mixture Models in Subspace Projections, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12), 2012: 132-140.

[25] S. Günnemann, I. Färber, M. Rüdiger and T. Seidl. SMVC: Semi-Supervised Multi-View Clustering in Subspace Projections, in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14), 2014: 253-262.

[26] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl and D. Keim. Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data, in Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST'12), 2012: 63-72.

[27] V. T. Tao and J. H. Lee. A Novel Approach for Finding Alternative Clusterings Using Feature Selection, in Proceedings of the 17th International Conference on Database Systems for Advanced Applications (DASFAA'12), 2012: 482-493.

[28] S. Yang and L. Zhang. Non-redundant Multiple Clustering by Nonnegative Matrix Factorization, Machine Learning, 2016: 1-18.

[29] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, in Proceedings of the International Conference on Management of Data, 1998: 94–105.

[30] H. Nagesh, S. Goil, and A. Choudhary. Mafia: Efficient and Scalable Subspace Clustering for Very Large Data Sets. Technical Report 9906-010, Northwestern University, 1999.

[31] A. Hinneburg and D. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise, in Proceedings of the 4th ACM International Conference on Knowledge Discovery and Data Mining (KDD'98), 1998: 58–65.

[32] H. Elghazel and A. Aussem. Unsupervised Feature Selection with Ensemble Learning，Machine Learning, 2015, 98(1-2): 157-180.

[33] Y. X. Wang and H. Xu. Noisy Sparse Subspace Clustering, in Proceedings of the 30th International Conference on Machine Learning (ICML'13), 2013: 1-8.

[34] C. You, D. P. Robinson and R.Vidal. Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit, in Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016:3918-3927.

[35] X. Huang, Y. Ye, L. Xiong, R. Y. K. Lau, N. Jiang, and S. Wang. Time Series K-means: A New K-means Type Smooth Subspace Clustering for Time Series Data, Information Sciences, 2016, 367–368: 1-13.

[36] S. R. M. Oliveira and O. R. Zaïane. Privacy Preserving Clustering by Data Transformation, in Proceedings of the 18th Brazilian Symposium on Databases, 2003: 304-318.

[37] M. N. Lakshmi and D. K. S. Rani. Privacy Preserving Clustering by Hybrid Data Transformation approach, International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, 2013, 3(8): 696-700.

[38] D. Su, J. Cao, N. Li, E. Bertino and H. Jin. Differentially Private K-means Clustering, in Proceedings of the Sixth ACM Conference on Data and Application Secure and Privacy, 2016: 26-37.

[39] Y. Wang, Y. X. Wang and A. Singh. Differentially Private Subspace Clustering, in Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS'15), 2015: 1-9.

[40] Y. Wang, Y. Wang and A. Singh. A Theoretical Analysis of Noisy Sparse Subspace Clustering on Dimensionality-reduced Data, CoRR, vol. abs/1610.07650, arXiv: 1610.07650v1, 2016

[41] S. Guo and X. Meng. Density Peaks Clustering with Differential Privacy, in Proceedings of the 8th Biennial Conference on Innovative Data Systems Research (CIDR'17), 2017: 1-6.

[42] J. Vaidya and C. Clifton. Privacy-preserving K-means Clustering over Vertically Partitioned Data, in Proceedings of the 9th ACM ACM SIGKDD International

Conference on Knowledge Discovery and Data Mining (KDD'03), 2003: 206–215.

[43] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A New Privacy-preserving Distributed k-clustering Algorithm, in Proceedings of the 2006 SIAM International Conference on Data Mining, 2006: 494-498.

[44] G. Jagannathan, K. Pillaipakkamnatt, R. N. Wright and D. Umano. Communication-efficient Privacy-preserving Clustering, Transactions on Data Privacy, 2010, 3(1): 1-25.

[45] M. Beye, Z. Erkin and R. L. Lagendijk. Efficient Privacy Preserving K-means Clustering in A Three-party Setting, in Proceedings of the 2011 IEEE International Workshop on Information Forensics and Secure (WIFS), 2011: 1-6.

[46] S. Jha, L. Kruger and P. Mcdaniel. Privacy Preserving Clustering, in Proceedings of the 10th European Symposium on Research in Computer Secure, 2005: 397-417.

[47] P. Bunn and R. Ostrovsky. Secure Two-party K-means Clustering, in Proceedings of the 14th ACM Conference on Computer Communication Secure, 2007: 486-497.

[48] J. Sakuma and S. Kobayashi. Large-scale K-means Clustering with User-centric Privacy-preservation, Knowledge. Information Systems, 2010, 25(2): 253-279.

[49] Q. Zhang, L. T. Yang, Z. Chen and B. Fanyu. PPHOCFS: Privacy Preserving High-order CFS Algorithm on The Cloud for Clustering Multimedia Data, ACM Transactions on Multimedia Computing Communication & Applications, 2016,12(4): 66:1-15

[50] Y. Kim, K. Shim, M.S. Kim and J.S. Lee. DBCURE-MR: An Efficient Density-Based Clustering Algorithm for Large Data Using MapReduce, Information Systems, 2014, 42(15): 15-35.

[51] X. Wu, X. Zhu, G.Q. Wu and W. Ding. Data Mining with Big Data, IEEE Transactions on Knowledge and Data Engineering, 2014, 26(1): 97-107.

[52] W.Y. Chen, Y. Song, H. Bai, C.J. Lin and E.Y. Chang. Parallel Spectral Clustering in Distributed Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568-586.

[53] R.L. Ferreira Cordeiro, C. Traina Junior, A.J. Machado Traina, J. López, U. Kang and C. Faloutsos. Clustering Very Large Multi-Dimensional Datasets With MapReduce, in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge

Discovery and Data Mining (KDD'11), 2011: 690-698.

[54] W. Yan, U. Brahmakshatriya, Y. Xue, M. Gilder and B. Wise. p-PIC: Parallel Power Iteration Clustering for Big Data, Journal of Parallel and Distributed Computing, 2013, 73(3): 352-359.

[55] 鲁伟明, 杜晨阳, 魏宝刚, 沈春辉, 叶振超. 基于MapReduce的分布式近邻传播聚类算法, 计算机研究与发展, 2012, 49(8): 1762-1772.

[56] 朱红, 丁世飞, 许新征. 基于改进属性约简的细粒度并行 AP 聚类算法, 计算机研究与发展, 2012, 49(12): 2638-2644.

[57] 赵卫中, 马慧芳, 傅燕翔, 史忠植. 基于云计算平台 Hadoop 的并行 k-means 聚类算法设计研究, 计算机科学, 2011, 38(10): 166-168.

[58] S. J. Lee, S. Kang, H. Kim and J. K. Min. An Efficient Parallel Graph Clustering Technique Using Pregel, in Proceedings of the 3rd IEEE International Conference on Big Data and Smart Computing (BigComp'16), 2016: 370-373.

[59] 王韬, 杨燕, 滕飞, 冯晨菲. 基于 RDDs 的分布式聚类集成算法, 小型微型计算机系统, 2016, 37(7): 1434-1439.

[60] L. Sun and C. Guo. Incremental Affinity Propagation Clustering Based on Message Passing, IEEE Transactions on Knowledge and Data Engineering, 2014, 26(11): 2731-2744.

[61] Q. Zhang, J. Liu and W. Wang. Incremental Subspace Clustering over Multiple Data Streams, in Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07), 2007: 727-732.

[62] D. Wang and T. Li. Document Update Summarization Using Incremental Hierarchical Clustering, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10), 2010: 279-288.

[63] Y. Zhou, H. Cheng and J.X. Yu. Clustering Large Attributed Graphs: An Efficient Incremental Approach, in Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10), 2010: 689-698.

[64] H. Ning, W. Xu, Y. Chi, Y. Gong and T.S. Huang. Incremental Spectral Clustering by Efficiently Updating the Eigen-System, Pattern Recognition, 2010, 43(1): 113-127.

[65] Y. Wang, L. Chen and J.P. Mei. Incremental Fuzzy Clustering With Multiple Medoids for Large Data, IEEE Transactions on Fuzzy Systems, 2014, 22(6): 1557-1568.

[66] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications, Society for Industrial and Applied Mathematics, 2009, 51(6): 455-500.

[67] Y. Liu, Y. Liu, and K. C. Chan. Tensor Distance Based Multilinear Locality-Preserved Maximum Information Embedding, IEEE Transactions on Neural Networks, 2010, 21(11): 1848-1854.

[68] J. D. Carroll and J. J. Chang. Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition, Psychometrika, 1970, 35: 283-319.

[69] L. R. Tucker. Some Mathematical Notes on Three-Mode Factor Analysis, Psychometrika, 1966, 31(3): 279-311.

[70] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the Best Rank-1 and Rank-(r_1, r_2, . . ., r_n) Approximation of Higher-Order Tensors, SIAM Journal on Matrix Analysis and Applications, 2000, 21(4): 1324-1342.

[71] L. Grasedyck. Hierarchical Singular Value Decomposition of Tensors, SIAM Journal on Matrix Analysis and Applications, 2010, 31(4): 2029-2054.

[72] I. V. Oseledets. Tensor-train Decomposition, SIAM Journal on Scientific Computing, 2011, 33(5): 2295-2317.

[73] I. V. Oseledets. Approximation of 2d×2d Matrices Using Tensor Decomposition, SIAM Journal on Matrix Analysis and Applications, 2010, 31(4): 2130-2145.

[74] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to The Web, Stanford, CA, USA: Stanford University, Tech. Rep., 1999.

[75] T. E. Booth. Power Iteration Method for The Several Largest Eigenvalues and Eigenfunctions, Nuclear science and engineering, 2006, 154(1): 48-62.

[76] X. Li, M. K. Ng, and Y. Ye. HAR: Hub. Authority and Relevance Scores in Multi-relational Data for Query Search, in Proceedings of the 12th SIAM International Conference on Data Mining Data Mining, Apr. 26–28 2012: 141-152.

[77] S. Ross. Introduction to Probability Models, Cambridge, MA, USA: Academic Press, 2014.

[78] A. Rodriguez and A. Laio. Clustering by Fast Search and Find of Density Peaks, Science, 2014, 344(6191): 1492-1496.

[79] P. Paillier. Public-key Cryptosystems Based on Composite Degree Residuosity Classes, in Proceedings of the 17th International Conference Theory Applications Cryptograph. Technology, Prague, Czech Republic, May. 2–6 , 1999: 223–238.

[80] L. Kuang, F. Hao, L. T. Yang, M. Lin, C. Luo, and G.Min. A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction, IEEE Transactions on Emerging Topics Computing, Sep. 2014, 2(3): 280-291.

[81] D. F. Gleich, L. H. Lim, and Y. Yu. Multilinear PageRank, SIAM Journal on Matrix Analysis and Applications, 2015, 36(4): 1507-1541.

[82] W. Li and M. K. Ng. On The Limiting Probability Distribution of a Transition Probability Tensor, Linear Multilinear Algebra, 2014, 62(3): 362-385.

[83] L. Qi. Eigenvalues of a Real Supersymmetric Tensor, Journal of Symbolic Computation, 2005, 40(6): 1302-1324.

[84] X. Liu, S. Ji, W. Glanzel, and B. De Moor. Multiview Partitioning via Tensor Methods, IEEE Transactions on Knowledge and Data Engineering, 2013, 25(5): 1056-901.

[85] X. Meng, X. Liu, Y. Tong, W. Gl€anzel, and S. Tan. Multi-view Clustering with Exemplars for Scientific Mapping, Scientometrics, 2015, 105(3): 1527-1552.

[86] X. Li, M. K. Ng, and Y. Ye. Multicomm: Finding Community Structure in Multi-dimensional Networks, IEEE Transactions on Knowledge and Data Engineering, 2014, 26(4): 929-941.

[87] B. J. Frey and D. Dueck. Clustering by Passing Messages between Data Points, Science, 2007, 315(5814): 972-976.

[88] P. N. Tan, M. Steinbach, and V. Kumar. Introduction to Data Mining. Boston, MA, USA: Pearson Addison Wesley, 2006.

[89] J. C. Dunn. A fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-separated Clusters, Cybernet 3, 1973, 3: 32-57.

[90] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban Computing: Concepts, Methodologies, and Applications, ACM Transactions on Intelligent Systems and Technology, 2014, 5(3): 38.

[91] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic Prediction in a Bikesharing System, in

Proceedings of the 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2015, 3-6: 1-10.

[92] M. N. Lakshmi and D. K. S. Rani. Privacy Preserving Clustering by Hybrid Data Transformation Approach, International Journal of Emerging Technology and AdvancedEngineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, 2013, 3(8): 696-700.

[93] Y. Lindell and B. Pinkas, Privacy-Preserving Data Mining, in Proceedings of the International Cryptology Conference on Advances in Cryptology, 2000: 36–54.

[94] C. Esposito, A. Castiglione, B. Martini, and K. K. R. Choo, Cloud Manufacturing: Secure, Privacy, and Forensic Concerns, in Proceedings of the IEEE Cloud Computing, 2016, 3(4): 16–22.

[95] X. Huang and X. Du, Achieving Big Data Privacy via Hybrid Cloud, in Proceedings of the 33rd Annual IEEE International Conference on Computer Communications Workshops, 2014: 512–517.

[96] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, K-nearest Neighbor Classification over Semantically Secure Encrypted Relational Data, in Proceedings of the IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1261–1273.

[97] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, Machine Learning Classification over Encrypted Data, in Proceedings of the 22nd Annual Network & Distributed System Secure Symposium, 2015.

[98] X. Liu, R. H. Deng,W. Ding, R. Lu, and B. Qin, Privacy-Preserving Outsourced Calculation on Floating Point Numbers, in Proceedings of the IEEE Transactions on Information Forensics and Secure, 2016, 11(11): 2513–2527.

[99] T. Veugen, Encrypted Integer Division and Secure Comparison, in Proceedings of the International Journal of Applied Cryptography, 2014, 3(2): 166-180.

[100] J. Feng, L. T. Yang, Q. Zhu, and K. K. R. Choo, Privacy-Preserving Tensor Decomposition over Encrypted Data in A Federated Cloud Environment, in Proceedings of the IEEE Transactions on Dependable and Secure Computing(TDSC'18), 2018.

[101] Q. Zhang, L. T. Yang, and Z. Chen, Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning, in Proceedings of the IEEE Transactions on

Computers, 2016, 65(5): 1351–1362.

[102] O. Goldreich, Foundations of Cryptography, vol. 2. basic applications, in Proceedings of the Cambridge Univ. Press, 2004.

[103] L. J. Guibas and R. Sedgewick. A Dichromatic Framework for Balanced Trees, in Proceedings of the Symposium on Foundations of Computer Science, 1978.

[104] H. Ning, W. Xu and Y. Chi. Incremental Spectral Clustering by Efficiently Updating the Eigen-system, in Proceedings of the Pattern Recognition, 2010, 43(1): 113–127.

[105] L. Kuang, L. T. Yang and Y. Liao. An Integration Framework on Cloud for Cyber-Physical-Socail Systems Big Data, in Proceedings of the IEEE Transactions on Cloud Computing (TCC'15), 2015.

[106] H. S. Park and C. H. Jun, A Simple and Fast Algorithm for K-medoids Clustering, in Proceedings of the Expert Systems with Applications, 2009, 36(2-part-P2): 3336-3341.

[107] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao and Li, P. An Incremental CFS Algorithm for Clustering Large Data in Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 2017:1-1.

[108] L. Zhao, Z. Chen and Y. Yang, Incremental CFS Clustering on Large Data, IEEE Global Conference on Signal and Information Processing, Montreal, Canada, 2017.

附录 1 攻读博士学位期间发表的学术论文

[1] **Yaliang Zhao**, Laurence T. Yang, Ronghao Zhang, A Tensor-Based Multiple Clustering Approach with Its Applications in Automation Systems, IEEE Transactions on Industrial Informatics, vol. 14, no. 1, pp. 283–291, 2018. （第一作者，SCI 1 区，署名单位：华中科技大学计算机科学与技术学院）

[2] **Yaliang Zhao**, Laurence T. Yang, Jiayu Sun, A Secure High-Order CFS Algorithm on Clouds for Industrial Internet of Things, IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3766–3774, 2018. （第一作者，SCI 1 区，署名单位：华中科技大学计算机科学与技术学院）

[3] **Yaliang Zhao**, Laurence T. Yang, Jiayu Sun, Privacy-Preserving Tensor-Based Multiple Clusterings on Cloud for Industrial IoT, IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2372–2381, 2019. （第一作者，SCI 1 区，署名单位：华中科技大学计算机科学与技术学院）

[4] **Yaliang Zhao**, Laurence T. Yang, Ronghao Zhang, Tensor-Based Multiple Clustering Approaches for Cyber-Physical-Social Applications, IEEE Transactions on Emerging Topics in Computing, DOI: 10.1109/TETC.2018.2801464, 2018. （第一作者，SCI 2 区，署名单位：华中科技大学计算机科学与技术学院）

[5] **Yaliang Zhao**, Samwel K. Tarus, Laurence T. Yang, Jiayu Sun, Yunfei Ge, Jinke Wang, Privacy-Preserving Clustering for Big Data in Cyber-Physical-Social Systems: Survey and Perspectives, Information Science, Accepted, 2019. （第一作者，SCI 2 区，署名单位：华中科技大学计算机科学与技术学院）

[6] Liwei Kuang，Laurence T. Yang，Xiaokang Wang，Puming Wang，**Yaliang Zhao**，A Tensor-Based Big Data Model for QoS Improvement in Software Defined Networks, IEEE Network, vol. 30, no. 1, pp. 30-35, 2016. （第五作者，SCI 1 区，署名单位：华中科技大学计算机科学与技术学院）

[7] Xia Xie, Shuwen Luo, Hai Jin, **Yaliang Zhao**, Xijiang Ke, Laurence T. Yang, A Pre-Processing Mechanism for MapReduce-based Inequality-Join, IEEE Systems

Journal, 2015, Accepted. （第四作者，SCI 2 区，署名单位：华中科技大学计算机科学与技术学院）

[8] Xia Xie, Yanzan Wu, Hai Jin, **Yaliang Zhao**, Xijiang Ke, Laurence T. Yang, A Partitioning Algorithm Based on Vertex-Cut and Community Detection, IEEE Systems Journal, 2014, Accepted. （第四作者，SCI 2 区，署名单位：华中科技大学计算机科学与技术学院）

附录 2 攻读博士学位期间参加的科研项目

[1] 国家自然科学基金项目，61802112，面向多源异构数据的多聚类通用模型及安全高效算法研究，2019/01-2021/12，在研，主持。

[2] 国家重点研发计划“网络空间安全”重点专项, 2017YFB0801804，云数据中心威胁预警与精准防御关键技术与系统, 2017/07-2020/12，在研，参与。

[3] 人-机-物三元空间的大数据表达、管理、处理、服务的体系设计，国家“千人计划”项目（No. HUSTQR201102），2011/12-2016/12，结题，参与。

[4] 华中科技大学自主创新研究基金资助项目，2018KFYXKJC046，高效安全的信息-物理-社会大数据分析与处理研究，2018/01-2019/12，在研，参与。

[5] 华中科技大学计算机科学与技术学院自主创新研究基金重点项目，2016YXZD017, 基于张量的大数据统一表示与高效分析方法，2016/01-2017/12，结题，参与。

[6] 深圳市科技计划项目，JCYJ20170307172200714，云上高效安全的张量大数据分析与处理研究，2017/06-2019/06，在研，参与。

附录 3 攻读博士学位期间申请的专利

[1] 杨天若, **赵雅靓**，张荣皓，一种基于张量的跨域异构大数据多视角聚类方法和装置，中国发明专利，申请号：201810970444.6。

[2] 杨天若, **赵雅靓**，孙佳宇，一种多源异构数据的聚类方法及装置，中国发明专利，申请号：201811593400.2。

附录 4 攻读博士学位期间获得的奖励

[1] 第三届中国计算机学会武汉分部优秀博士生论坛荣获一等奖，并授予中国计算机学会武汉分部优秀博士生称号，2019.01。