天津科技大学研究生学位论文

(申请硕士学位)

基于 MapReduce 的聚类算法并行化分析与应用研究

PARALLEL ANALYSIS AND APPLICATION OF CLUSTERING ALGORITHM BASED ON MAPREDUCE

专 业 名 称：计算机应用技术

指 导 教 师：孙志伟 副教授

研 究 生 姓 名：冯海波

申请学位级别：工学硕士

论文提交日期：2017 年 3 月

分类号：TP319 学校代码：10057
密级：（秘密、机密、绝密） 研究生学号：11834004

基于 MapReduce 的聚类算法并行化分析与应用研究

Parallel Analysis of Clustering Algorithm Based on MapReduce And Applied Research

专 业 名 称 ：计算机应用技术
指导教师姓名：孙志伟 副教授
研 究 生 姓 名 ：冯海波
申请学位级别：工学硕士
论文提交日期：2016 年 3 月
论文课题来源：学校自选项目
学位授予单位：天津科技大学

天 津 科 技 大 学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究工作所取得的成果。除文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果内容，也不包括为获得天津科技大学或其它教育机构的学位或证书而使用过的材料。对本文研究做出重要贡献的个人和集体，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后果由本人承担。

作者签名：

日期：　　年　　月　　日

知识产权和专利权保护声明

本人郑重声明：所呈交的论文是本人在导师具体指导下并得到相关研究经费支持下完成的，其数据和研究成果归属于导师和作者本人，知识产权单位属天津科技大学；所涉及的创造性发明的专利权及使用权完全归天津科技大学所有。本人保证毕业后，以本论文数据和资料发表论文或使用论文工作成果时署名第一单位仍然为天津科技大学。本人完全意识到本声明的法律后果由本人承担。

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，同意学校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，同意公布论文的全部或部分内容，允许论文被查阅和借阅。本人授权天津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。

保　密 □（请在方框内打“√”），在　　年解密后适用本授权书。

本学位论文属于

不保密 □（请在方框内打“√”）。

作者签名：　　日期：　　年　　月　　日

导师签名：　　日期：　　年　　月　　日

摘　要

聚类分析是数据挖掘领域中一种常用的技术手段，通过数据之间的距离将数据集分为多个簇，使得簇内尽可能相似，簇间尽可能不相似，又可以依据数据集的类型不同，分为基于层次、划分、密度、网格、模型等多种聚类算法。通过聚类算法可以将其应用于，气候区划，供热效果分析，文本聚类，离群点检测，生物信息等诸多领域。然而，随着信息时代的到来这些领域中的数据已呈现出爆炸性增长，因此考虑使用并行计算的方式来解决这个问题。MapReduce 是由谷歌提出的一个并行计算模型，通过 MapReduce 这种分布式的编程模式，可对传统聚类算法中具有并行性的部分进行编程，将大大加快我们对大数据集的处理能力。

本文首先对聚类算法基于划分、层次、网格、密度、模型的方法进行总结，并对经典算法中存在的并行性进行分析，将聚类算法应用在供热领域，研究出水温度曲线与目标温度曲线的关系，提出基于曲线相似度的供热过程评价方法，使用并行化的 k-means 进行聚类，这是一种基于划分的聚类算法，将其并行过程进行 Map()函数和 Reduce()函数的设计，使用 Hadoop 开源架构进行环境搭建，最后基于 React 开发了一个软件平台，便于数据的可视化展示。

实验结果表明，聚类算法的并行化可以大大加快传统串行算法在处理大数据集上的不足，并依据聚类结果完成了不同供热行为供热锅炉房的分簇，每簇中的数据又提供了趋势变化、横向平移、纵向平移、横向伸缩、纵向伸缩五个方面的度量，为锅炉管控人员在供热系统上提供了参考。

关键词：聚类算法；MapReduce；供热锅炉系统；React

ABSTRACT

Clustering analysis is a commonly used technical means in the field of data mining. The data set is divided into multiple clusters by the distance of the data, so that the clusters are as similar as possible, and the clusters are as similar as possible. According to the type of data set Different, based on hierarchical, division, density, grid, model and other clustering algorithm. It can be applied to clustering algorithm, climate zoning, heating effect analysis, text clustering, outlier detection, biological information and many other fields. However, with the advent of the information age, the data in these areas has shown explosive growth, so consider the use of parallel computing to solve this problem. MapReduce is a parallel computing model proposed by Google. Through distributed programming mode of MapReduce, it is possible to program the parallel part of the traditional clustering algorithm, which will greatly accelerate our ability to deal with large data sets.

In this paper, we first summarize the clustering algorithm, summarize the clustering algorithm based on partition, hierarchy, grid, density and model, and analyze the parallelism in the classical algorithm, and apply the clustering algorithm to heat In this paper, the relationship between the effluent temperature curve and the target temperature curve is studied. A heating process evaluation method based on curve similarity is proposed. Clustering is performed using parallel k-means. This is a clustering algorithm based on partitioning. The process of Map () function and Reduce () function design, the use of Hadoop open source architecture for environmental structures, and finally based on React developed a software platform to facilitate the visualization of data display.

The experimental results show that the parallelization of the clustering algorithm can greatly speed up the shortcomings of the traditional serial algorithm in dealing with the large data set and complete the clustering of the heating boiler rooms with different heating behaviors according to the clustering results. But also provides the trend of change, horizontal translation, vertical translation, horizontal stretching, vertical stretching five aspects of the measurement for the boiler control personnel in the heating system provides a reference.

Key words: Clustering Algorithm, MapReduce, Heating Boilers, React

目　录

1 绪论

本章主要对论文课题的背景进行介绍，对选题进行分析，阐述国内外对聚类算法并行化的研究现状及其在各个领域的应用发展，分析和证明了本课题的研究意义和应用价值，根据以上内容引出本文的结构安排和章节介绍。

1.1 课题研究背景及意义

随着计算机硬件的高速发展以及即将到来的 AI/VR 时代，网络客户端和服务器每时每刻都在产生着浩如烟海的数据，然而我们却缺乏对其充分的理解和应用，传统的数据分析方法已经不能满足海量数据分析和处理的要求。于是，数据挖掘技术应运而生。数据挖掘被大家所广泛接受的定义是，从大量的不完全的、有噪声的、模糊的疑惑是随机的数据中，提取出隐含的、事先不为人所知道的、却又是潜在有用的知识和信息的过程。它是一种在海量数据中寻找规则或者模式的过程，是一个新兴的并且具有广阔应用前景的研究领域。

聚类分析是数据挖掘技术中重要组成部分，可以有效的分析数据并从中发现有用的信息。聚类分析是依据数据中对象与对象之间其之间的关系（相似度），将数据对象分簇。聚类的最终目的是，使簇内的对象与对象之间是相似的（相关的），而不同簇中的对象是不同的（不相关的）。簇内的相似性（同质性）越大，簇间差距越大，聚类效果就越有效果。它广泛应用于多个领域，如文本聚类、模式识别、人工智能、市场分析、医疗卫生、图像分析和信息检索。

数据挖掘的任务是从大量数据中提取有价值的信息，算法运行效率成了对海量数据处理的桎梏之一，传统的单机串行算法运行效率较低；部分聚类算法中蕴涵并行的步骤，为了解决处理这种效率问题，将并行化的程序设计思想（并行处理）引入聚类算法，这样可以降低算法的时间复杂度，利用大量廉价计算机集群进行并行计算，从而有效的缩短聚类结果的时间。

Hadoop 是一个开源的分布式云计算平台，能够实现对大量的数据集高效、可靠、可伸缩的分布式并行处理。MapReduce 是 Hadoop 生态系统中对谷歌 MapReduce 的开源实现，利用 MapReduce 编程模式，我们就可以方便地把已有的算法移植到 Hadoop 平台实现算法的并行化。

当前，MapReduce 在数据挖掘领域被广泛应用，出现了很多基于 MapReduce 平台的聚类算法。然而随着数据量的进一步增加，实际应用需求的差异，以及实际项目

中数据集的不同，针对数据挖掘中的诸多问题，除了研究新的聚类算法以外，针对具体应用需求对现有聚类算法进行改进并移植到 Hadoop 平台上进行分布式实现，从而提高对大规模数据集处理的扩展性，也非常的有效且相对方便，成为当前研究的重要方向，具有十分重大的意义。

1.2 国内外发展现状

由于任何一种聚类算法都无法将各种各样的高维数据积聚成形色不同的结果簇[1]。因此在聚类分析领域依据不同的数据结构形成了多种不同的聚类算法方向，传统聚类方法中分为基于划分，基于层级，基于密度，基于网格和基于模型的五类经典算法用来处理各种不同的数据集。

1.2.1 基于划分的聚类方法

基于划分的聚类方法。k-means（k-均值、k-平均）最早是由美国科学家 MacQueen 在 1967 年提出[2]，凭借其算法简单、快速的特点当属聚类算法中最经典的算法。然而，由于要提前输入 k 值，所以 k 值的选取将对聚类的结果产生较大的影响，并且它不能发现非凸面的聚簇，或者说数据集量大小差别较大的聚簇，同时，由于少量的离群点会对平均值产生非常大的影响，所以其对噪声和离群点也极其敏感。为了处理对噪声敏感的问题，PAM 和 CLARA 算法被 L Kaufman 和 PJ Rousseeuw 在 1990 年提出[3]，由于使用每个类的接近中心的数据对象表示而被称为 k-中心的方法，当数据存在离群点和噪声时 k-中心点算法处理起来比 k-means 更具健壮性。很多人对基于划分的方法进行了改进来适应大规模的数据集合复杂结构的聚类。Z Huang 扩展了 k-means 算法提出了 k-modes 聚类算法，又被称为 k-模算法[4]，用数模将类的平均值替换掉。Lauritzen 在 EM[5]算法中根据聚簇与对象之间的从属关系发生的概率值来分配对象从而不把对象分配给一个确定的聚簇。Ng 和 Han 将取样技术和 PAM 算法结合起来随机的选择数据中的一小部分作为样本提出 CLARANS[6]算法，使其不考虑整个数据集合也同样在大规模数据集下的聚类分析取得优势。Estero，Kriegel 和 Xu 为了进一步改进 CLARANS 算法的性能采用极为高效的空间存取方法 R*树和聚焦技术[7]。FREM 算法也相应的改进了 EM 算法的运算性能，从而使其更加适合大数据的聚类分析[8]。

1.2.2 基于层次的聚类方法

基于层次的聚类方法。Kaufman 和 Rousseeuw 分别提出的凝聚和分裂方法可以作为最基本的层次聚类算法，当然其分裂点与合并点的选择困难问题导致其鲁棒性较差。Zhang 提出了一种综合的基于层次的聚类算法——BIRCH[9]，其首先将一棵 CF 树建

立在内存当中，然后再挑选一个聚类算法对其叶子节点进行聚类。为了识别出复杂的聚簇和大小不同的聚簇并且过滤离群点，Guha 提出了一种不用单个中心或对象来代表一个类的算法 CURE，它选择了数据空间中固定数目的具有代表性的点集来代表一个类[10],然后 Guha 又在 CURE 的基础上提出了 ROCK[11]方法,来适用于分类的数据。金阳和左万利提出了一种聚类算法 DNNS[12]，这是一种基于动态近邻选择模型的算法，它主要解决了 ROCK 对阈值 θ 的确定和选择困难的缺点问题。

1.2.3 基于网格的聚类方法

基于网格的聚类方法。Wang 提出了 STING[13]，Sheikholeslami 等提出了 Wavecluster[14]，这两种方法都是多分辨率的基于网格的聚类算法。它在开始的时候讲一个多为的网格结构强加在数据空间上，用来进行汇总数据，在找到密集区域之前采用小波变换的方式变换原有特征空间。为了处理更加复杂的图像数据的聚类问题，Wavecluster+[15]这种方法被 Yu Dantong 提出了一种改进的小波变换聚类方法来用解决。综合了基于密度和网格的方法，将其聚类算法进行结合，Agrawal 等提出了 CLIQUE[16]用于处理聚类的高维数据，陈宁和陈安提出的基于密度的增量式网格聚类方法[17]也是使用了此特征，而 Schikuta 等提出的 BANG[18]聚类算法是为了处理大数据集而产生的。这诸多基于网格的聚类算法皆对数据的输入次序不敏感，并不需要假设哪种规范的数据分布方式，因此对数据的维度和规模有较好的伸缩性，但是由于方法相对有所简化，导致聚类的结果准确度相对较低。

1.2.4 基于密度的聚类方法

基于密度的聚类方法。Ester 和 Kriegel 等提出了 DBSCAN[19]聚类算法，这种基于密度的聚类算法把具有极高密度的区域划分为类，并且能在具有噪声的空间数据中发现任意形状的簇。周水庚等对 DBSCAN 算法在各个阶段进行全方位的改进提出 FDBSCAN[20]算法从而大幅度提高 DBSCAN 的执行效率。Ankerst 等提出的 OPTICS[21]算法克服了 DBSCAN 参数设置复杂的缺点，这是一种基于类排序的方法。利用数据样本分布数量等密度线图的想法，赵艳广提出了一种等密度线的聚类算法[22]。Hinneburg 提出的 DENCLUE[23]和裴继法等提出的密度函数法[24]均是基于密度分布函数的聚类算法。Qiu X 等提出的方法[25]是将密度和距离综合应用起来进行聚类的簇划分。曾东海等[26]将多维数据空间依次划分成多个体积大小相等的矩形方阵单元格，并基于此方阵单元格定义密度和簇等概念模型，对其建立了一种基于空间划分的空间型索引结构树，通过这种空间划分树来寻找密集单元和与密集单元联通区域形成聚类结果。李光强等研

究出了一种适应空间局部密度变化的空间聚类算法-ADBSC[27]，其引入距离变化率概念来自动适应空间位置的局部密度变化。倪巍伟等[28]在 DBDC 的基础上针对高维空间水平数据分布的情况，给出一种局部分布式聚类算法。

1.2.5 基于模型的聚类方法

基于模型的聚类方法。统计学方法中常用方法包括 Fisher 提出的 COBWEB[29]，Gennari 等提出的 CLASSIT[30]，以及 Cheeseman 等提出的 AutoClass[31]，Pizzuti Clara 等提出的 P-AutoClass[32]等。神经网络方法中有 D.E.Rumelhart 提出的竞争学习[33]，采用几个单元的层级结构划分，以一种“胜者为王”的思想对系统目前处理的对象进行竞争。Kohonen 提出的学习矢量量化 LVQ[34]方法，是一种自适应将数据的聚类算法，其训练自基于对具有期望类别信息的数据。Kohonen 为了更有效的利用训练样本，将获胜单元和第二单元在特定条件下进行触发更新[35]，以便更有效的训练样本，Teuvo 提出自组织特征映射 SOM 算法[36]，高维空间的数据被 SOM 算法转化到二维空间，并且保证了在二维空间中输入样本间的相似度，逐步收敛到最近的类别依据数据的分布。刘铭等[37]通过压缩神经元的特征集合，为了减少聚类的时间提高效率仅仅选择与神经元代表的文档类相关的特征构造特征向量，避免了无关向量的干扰，提升了精确度。

1.2.6 基于 MapReduce 的并行聚类算法

MapReduce 编程模型最早由 Google 提出，Apache 基金会的开源项目 Hadoop 中的 MapReduce 则是对其的开源实现，Hadoop 拥有强大的生态环境，包括了 HDFS、MapReduce、Yarn、Zookeeper 等开源实现，使用 MapReduce 编程模型可以很好的让我们屏蔽到底层并行的通信等编程机制，而只关注与对 MapReduce 提供的接口进行编程。

近年来，由于 MapReduce 处理大数据集的优越性，国内外学者对聚类在 MapReduce 上的应用展开了大量研究。江小平等[38]利用 MapReduce 编程模型将 K-均值算法进行了并行化，但其算法没有解决初始聚类中心点问题。周婷等[39]将 K-means 拆分到 Map 和 Reduce 任务分别计算质心距离和完成质心更新，使算法表现出了很好的稳定性和扩展性。赵庆[40]提出了基于 Canopy 的改进 K-means 算法，针采用“最大最小原则”避免了 Cannopy 选取的盲目性，利用 MapReduce 进行并行编程，以新闻信息作为数据集验证了算法的准确率和稳定性。贾瑞玉等[41]利用遗传算法的粗粒度并行化设计思想，将各个子种群编号作为个体区分并通过 MapReduce 实验验证了算法的优越性。李兰英[42]等为了确定初始聚类中心，首先用模糊聚类的思想对数据集进行分

类，然后采用动态计算中心点的方式进行二次分类最后在 MapReduce 编程模型上进行实验使算法的收敛速度更快。张磊等[43]利用网格处理技术对数据进行预处理，用网格预处理后的单元重心点取代单元中保存的所有点，然后利用 MapReduce 将各个单元的重心点作为聚类的基本数据单元进行分析，其结果表明使用并行的聚类算法降低了计算的复杂度。

1.3 本文章节安排

本文将分为六个章节对课题进行阐述，各章的内容安排如下：

第一章：绪论。本部分作为开篇伊始，将着重带领大家了解本课题选用的价值及其背景和意义。通过介绍与课题相关的国内外最新发展现状，包括数据挖掘学科的进展，MapReduce 技术的进步与发展，以及包括本节在内的整篇文章的章节介绍，以便读者更容易的把握本文主要内容。

第二章：相关技术研究。本章节将介绍本课题的研究路线，及其中用到的关键技术，对此进行详细的阐述。分别介绍数据挖掘的分类与研究，对聚类分析模块做进一步的深入探索，对目前常用的聚类方法进行对比分析，然后解释了 MapReduce 的编程模型，对其中的主要过程进行了介绍，最后介绍了聚类算法的应用领域并且对供热锅炉系统的物理模型进行阐述，方便读者了解供热锅炉系统的运作方式。

第三章：基于曲线相似度的供热过程评价。本章提出了一个基于曲线相似度的供热过程评价模型，通过 Frechet 距离，伸缩和平移一共五个度量对出水温度和评价曲线做对比，最后加权平均为一个度量，量化的评价供热锅炉的供热过程，从而方便锅炉管控人员根据数据操作锅炉输出参数，进行指标结算，更加精确的控制热量，节省能耗。

第四章：基于 MapReduce 的并行 K 均值算法研究。本章在之前的基础上，对聚类算法进行并行化处理，研究了基于划分的常规聚类算法 K 均值，对其进行并行处理并将供热行为评价模型融入进行改进，从而提高传统串行算法的效率，在 Map 操作之前加入一个 PreMap 操作函数对属性相似度进行计算，解决在处理大规模数据集时的时间复杂度问题。

第五章：实验及结果。本章内容主要对前几章的算法进行代码实现，通过搭建 Hadoop 平台，运行于 Linux 环境中，利用集群的高并发效果编写相应的 MapReduce 函数，同时对天津地区供热锅炉系统的历史数据进行分析，进行聚类分析供热行为，从而发现各个锅炉房管理人员的管控问题，帮助管理者进行绩效管理。

第六章：供热效果实时监测系统。考虑到供热系统的即时性特征，为了方便管理着可以及时发现问题，同时可以利用数据可视化的手段将聚类的结果进行展示，方便人们查看具体的内容理解数值型数据的直观含义，帮助管控人员及时发现问题，调整控制策略。

第七章：总结和展望。最后将本篇文章所涉及的研究路线和方法进行总结，通过指出目前算法和系统中的不足之处，期待后续在各个方面进行仔细的调整。

2 相关技术研究

本章主要对本文中使用到的一些技术做相关的介绍和研究，将主要介绍数聚类分析的类别和具体算法，同时对 MapReduce 编程模型进行了研究，并对一些经典聚类算法进行了并行化分析，最后对一些应用领域进行了探索。

2.1 聚类分析

2.1.1 聚类分析的概念

聚类分析简称聚类，是一个把数据对象（或观测）划分成子集的过程。每个子集是一个簇，使得簇中的对象彼此相似，但与其他簇中的对象不相似。由聚类分析产生的簇的集合称作一个聚类。在这种语境下，在相同的数据集上，不同的聚类方法可能产生不同的聚类。划分不是通过人，而是通过聚类算法进行。聚类是有用的，因为它可能导致数据内事先未知的群组的发现。

作为一种数据挖掘功能，聚类分析也可以作为一种独立的工具，用来洞察数据的分部，观察每个簇的特征，将进一步分析集中在特定的簇集合上。另外，聚类分析可以作为其他算法的预处理步骤，之后这些算法将在检测到的簇和选择的属性或特征上进行操作。

在某些应用中，聚类又称作数据分割，因为它根据数据的相似性把大型数据集合划分成组。聚类还可以用于离群点检测，其中离群点（“远离”任何簇的值）可能比普通情况更值得注意。

2.1.2 基于层次的聚类算法

层次聚类方法将数据对象组成层次结构或簇的“树”。对给定的数据集进行逐层分解，直到满足某种条件为止。具体可分为“自底向上”和“自顶向下”两种方案。在“自底向上”方案中，初始时每个数据点组成一个单独的组，在接下来的迭代中，按一定的距离度量将相互邻近的组合并成一个组，直至所有的记录组成一个分组或者满足某个条件为止。如下图所示为自底向上的凝聚层次聚类算法示意图：

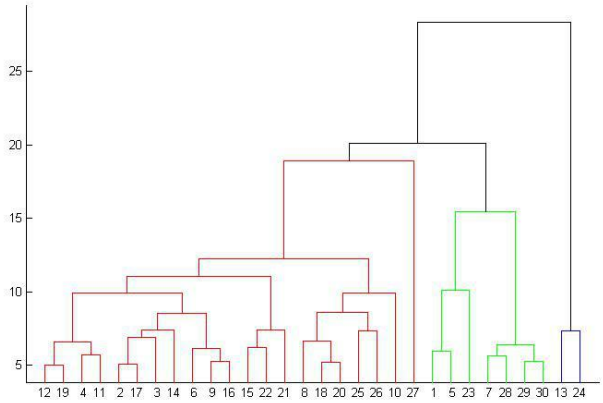


图 2-1 凝聚型层次聚类算法示意图

Fig. 2-1 Aggregation Hierarchical Clustering Algorithm Schematic

这两种路径本质上没有孰优孰劣之分，只是在实际应用的时候要根据数据特点以及自己想得到的“类”的个数，来考虑是自上而下更快还是自下而上更快。至于根据相似性度量判断“类”的方法包括最短距离法、最长距离法、中间距离法、类平均法等等（其中类平均法往往被认为是最常用也最好用的方法，一方面因为其良好的单调性，另一方面因为其空间扩张/浓缩的程度适中）。基于层次的方法中比较新的算法有 BIRCH[44]（Balanced Iterative Reducing and Clustering Using Hierarchies）主要是在数据体量很大的时候使用，而且数据类型是数字型；ROCK[45]（A Hierarchical Clustering Algorithm for Categorical Attributes）主要用在列表型的数据类型上；Chameleon[46]（A Hierarchical Clustering Algorithm Using Dynamic Modeling）里用到的相似性度量是 KNN（K-Nearest-Neighbor）算法，并以此构建一个 graph，Chameleon 的聚类效果被认为非常强大，比 BIRCH 好用，但运算复杂度很高，为 O(n^2)。以下是 Chameleon 的聚类效果图，其中一个颜色代表一类，可以看出来是可以处理非常复杂的形状的。

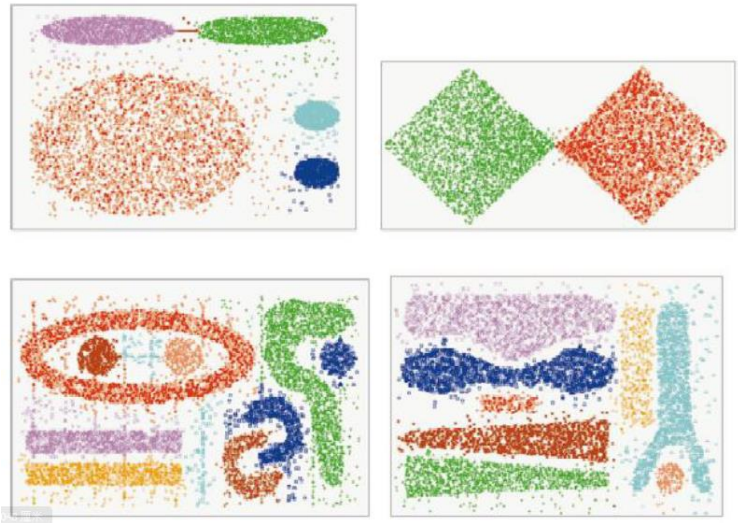


图 2-2 Chameleon 聚类效果图

Fig. 2-2 Chameleon clustering effect diagram

2.1.3 基于划分的聚类算法

给定包含 N 个点的数据集，划分法将构造 K 个分组，每个分组代表一个聚类，这里每个分组至少包含一个数据点，每个数据点属于且仅属于一个分组。对于给定的 K 值，算法先给出一个初始的分组方法，然后通过反复迭代的方法改变分组，使得每一次改进之后的分组方案较前一次好，这里好的标准在于同一组中的点越近越好，不同组中的点越远越好。首先你要确定这堆散点最后聚成几类，然后挑选几个点作为初始中心点，再然后依据预先定好的启发式算法（heuristic algorithms）给数据点做迭代重置（iterative relocation），直到最后到达“类内的点都足够近，类间的点都足够远”的目标效果。也正是根据所谓的“启发式算法”，形成了 k-means 算法及其变体包括 k-medoids、k-modes、k-medians、kernel k-means 等算法。k-means[47]对初始值的设置很敏感，所以有了 k-means++[48]、intelligent k-means[49]、genetic k-means[50]；k-means 对噪声和离群值非常敏感，所以有 k-medoids[51]和 k-medians[54]；k-means 只用于数值类型数据，不适用于类别类型数据，所以 k-modes[55]被提出；k-means 不能解决非凸（non-convex）数据，所以有了 kernel k-means[56]。另外，很多经验都告诉我们基于划分的聚类多适用于中等体量的数据集，但我们也不知道“中等”到底有多“中”，所以不妨理

解成，数据集越大，越有可能陷入局部最小。图 2-3 为 k-means 的聚类算法图解：

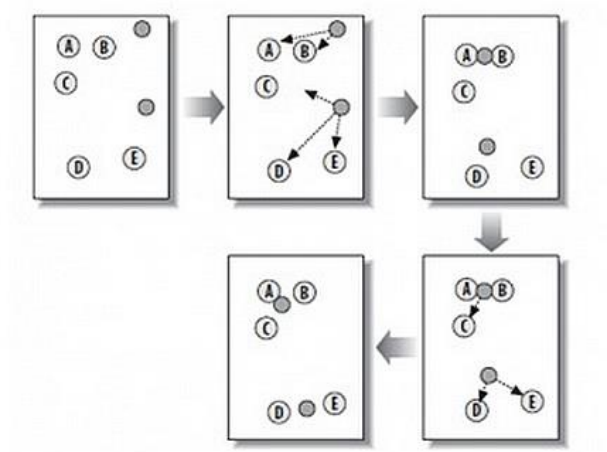


图 2-3 k-means 聚类算法图解

Fig. 2-3 K-means clustering algorithm

2.1.4 基于密度的聚类算法

基于密度的方法的特点是不依赖于距离，而是依赖于密度，从而克服基于距离的算法只能发现“球形”聚簇的缺点。其核心思想在于只要一个区域中点的密度大于某个阈值，就把它加到与之相近的聚类中去。由于 k-means 解决不了这种不规则形状的聚类。于是就有了基于密度的聚类算法来系统解决这个问题。该方法同时也对噪声数据的处理比较好。其原理简单说圆，其中要定义两个参数，一个是圆的最大半径，另外一个是一个圆里最少应容纳几个点。最后在一个圆里的，就是一个类。DBSCAN[57]（Density-Based Spatial Clustering of Applications with Noise）就是其中的典型，可惜参数设置也是个问题，对这两个参数的设置非常敏感。DBSCAN 的扩展叫 OPTICS[58]（Ordering Points To Identify Clustering Structure）通过优先对高密度（high density）进行搜索，然后根据高密度的特点设置参数，改善了 DBSCAN 的不足。图 2-4 是表现了 DBSCAN 聚簇的生成过程：

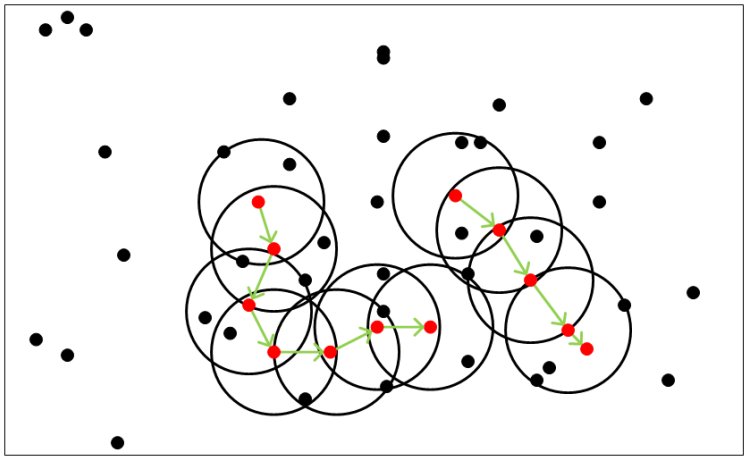


图 2-4 DBSCAN 聚类算法聚簇图解

Fig. 2-4 DBSCAN clustering algorithm clustering diagram

2.1.5 基于网格的聚类算法

这种方法通常将数据空间划分成有限个单元的网格结构，所有的处理都是以单个的单元为对象。这样做起来处理速度很快，因为这与数据点的个数无关，而只与单元个数有关。基于网格的聚类算法原理就是将数据空间划分为网格单元，将数据对象集映射到网格单元中，并计算每个单元的密度。根据预设的阈值判断每个网格单元是否为高密度单元，由邻近的稠密单元组形成“类”。该类方法的优点就是执行效率高，因为其速度与数据对象的个数无关，而只依赖于数据空间中每个维上单元的个数。但缺点也是不少，比如对参数敏感、无法处理不规则分布的数据、维数灾难等。STING[59]（Statistical Information Grid）和 CLIQUE（Clustering In QUEst）是该类方法中的代表性算法。图 2-5 是 CLIQUE 的一个例子：

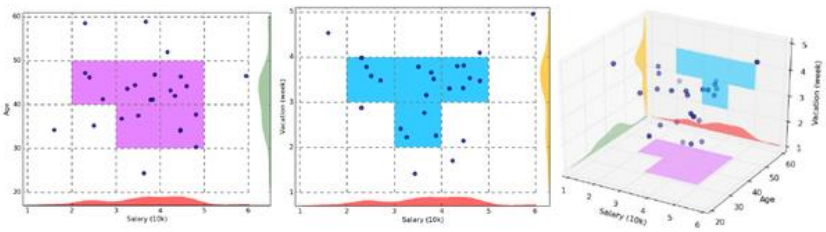


图 2-5 Clique 聚类算法结果图解

Fig. 2-5 Clique Clustering Algorithm Result Graph

2.1.6 基于模型的聚类算法

基于模型的方法给每一个聚类假定一个模型，然后去寻找能很好的拟合模型的数据集。模型可能是数据点在空间中的密度分布函数或者其它。这样的方法通常包含的

潜在假设是：数据集是由一系列的潜在概率分布生成的。通常有两种尝试思路：统计学方法和神经网络方法。基于模型的聚类算法主要是指基于概率模型的方法和基于神经网络模型的方法，尤其以基于概率模型的方法居多。这里的概率模型主要指概率生成模型（generative Model），同一“类”的数据属于同一种概率分布。这种方法的优点就是对“类”的划分不那么“坚硬”，而是以概率形式表现，每一类的特征也可以用参数来表达；但缺点就是执行效率不高，特别是分布数量很多并且数据量很少的时候。其中最典型、也最常用的方法就是高斯混合模型（GMM，Gaussian Mixture Models）。基于神经网络模型的方法主要就是指 SOM（Self Organized Maps）了，也是我们所知的唯一一个非监督学习的神经网络了。图 2-6 表现的就是 GMM 的一个例子，其中用到 EM 算法来做最大似然估计。

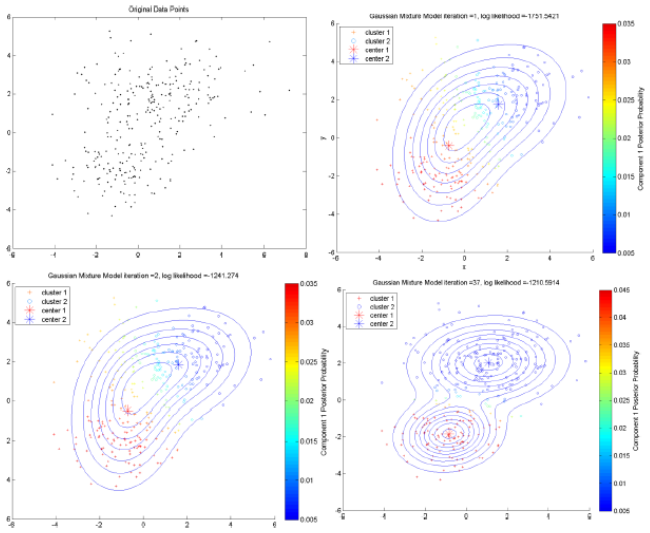


图 2-6 GMM 聚类算法结果图解

Fig. 2-6 GMM clustering algorithm

2.2 聚类算法并行化分析

(1) K-means 基本原理：首先随机的选择 K 个对象，每个对象代表一个簇的初始均值和中心；对剩余的每个对象，根据其与各个簇的均值的距离，将其指派到最相似的簇。然后计算每个簇的新均值，过程不断重复直到准则函数收敛。

效率分析：时间复杂度 O（nki）、空间复杂度 O（k）。

MapReduce 并行化分析：k-means 从逻辑上分为三部分，聚类中心初始化、迭代更新聚类中心、聚类标注，三部分都可以 MapReduce 并行化。

(2) CLARANS 基本原理：与 k-means 相似，也是以聚类中心划分聚类的，一旦 k

个聚类中心确定了，聚类马上就能完成。不同的是 k-means 算法以类簇的样本均值代表聚类中心，而 CLARANS 采用每个簇中选择一个世纪的对象代表该簇。其余的对每个对象聚类到其最相似的代表性对象所在的簇中。

效率分析：时间复杂度 O（n^2）、空间复杂度 O（ks）。

MapReduce 并行化分析：CLARANS 从逻辑上分为三部分，聚类中心和邻域样本初始化、迭代更新聚类中心、聚类标注，均可并行化处理。

(3) DBSCAN 基本原理：DBSCAN 算法一种基于密度的聚类算法，与划分和层次聚类算法不同，它将簇定义为密度相连的点的最大集合，能够将足够高的密度区域划分为簇，并可以在有噪声的空间数据中发现任意形状的聚类。

效率分析：时间复杂度 O（n^2）空间复杂度 O(n)。

MapReduce 并行化分析：DBSCAN 从逻辑上分为三部分，样本抽样、对抽样样本进行聚类、聚类标注，均可并行化计算。

(4) BIRTH 基本原理：BIRTH 算法利用层次方法的平衡迭代规约和聚类，是一个综合的层次聚类方法，它用聚类特征和聚类特诊树概括聚类特征，该算法可以通过聚类特征可以方便的进行中心、半径、直径以及类内、类间进行距离的计算。

效率分析：时间和空间的复杂度均为 O（N）。

MapReduce 并行化分析：不适合对分隔的数据进行处理，而且是增量计算的。

(5) Chameleon 基本原理：Chameleon（变色龙算法）是在一个层次聚类中采用动态模型进行聚类的方法。在它的聚类过程中，如果两个簇间的互联性和近似度与簇内部对象间的互联性和近似高度相关，则合并这两个簇。基于动态模型的合并过程中有利于自然的聚类发现，而且只要定义了相似度函数就可以应用于所有类型的数据。

效率分析：时间复杂度 O(n^2)、空间复杂度 O（n）。

MapReduce 并行化分析：不适合对分隔数据处理。

(6) STING 基本原理：STING 是一种基于网格的多分辨率聚类技术，它将空间区域划分为矩形单元，针对不同级别的分辨率，通常存在多个级别的矩形单元，这些单元形成了一个层次结构；高层的每个单元划分为多个第一层的单元。

效率分析：时间复杂度 O(n)、空间复杂度 O（1）。

MapReduce 并行化分析：算法的数据分隔不是简单的块分隔，不适合 MR 并行化处理。

2.3 聚类算法并行化应用研究

由于 MapReduce 利用大量廉价计算机搭建集群，可以大大增加算法的运行效率，减少时间复杂度，将其和聚类算法结合对大数据集进行操作分析将会提高效率。

潘吴斌[52]将聚类算法的并行实现应用于气象领域，随着气象方向信息化水平的提高，气象服务、科研和交流活动中积累了大量气象数据的信息资料，包括站台信息和遥感卫星等，其使用 K-means 算法与 MapReduce 分布式计算模型结合，对 30a 地面气象气候资料进行了全国温度带及干湿区划分，取得了较好的结果。

张睿欣[53]将聚类算法的并行实现应用于微博用户信息的数据挖掘中，微博是微型博客的简称，可以在 SNS 社交网络平台上提供信息发布分享和获取的操作，近年来诸多明星和政府机构的入驻及其商业推广，大量的活跃用户发布信息操作记录形成了大规模的数据集，其利用 K-means 与 MapReduce 结合增强了算法处理大数据的能力，从而得到了相近兴趣爱好的用户聚类结果。

本文将在供热领域展开研究，随着“节能减排”的推广，及日益恶劣的雾霾环境恶化，对供热锅炉系统的数据分析将极具意义，可以为更合理的锅炉管理提供建议。

供热锅炉是一个多参数、多变量的复杂调度对象。燃气锅炉系统其输入参数包括燃气量，热水水位，冷水水位等。在工程上常常将简化处理用在锅炉控制上，主要划分为汽包水位控制系统，燃烧系统控制和辅助设备调节系统三部分。

汽包水位控制系统。被调量是汽包水位，调节量是给水流量。它主要是考虑气包内物料的平衡，使给水量适量锅炉的蒸发量，维持气包中的水位在工艺范围内。

燃烧系统控制，使燃料燃烧所产生的能量适应锅炉负荷的需要。蒸汽锅炉其基本任务是使燃烧所产生的热量能满足蒸汽热负荷的需要，蒸汽的母管压力是锅炉供气量与蒸汽负荷所需能量是否得到满足的指标。

辅助设备调节系统，这部分的控制多为简单控制系统。对于蒸汽锅炉有过热蒸汽调节系统，热力除氧器的压力和温度调节系统，热水锅炉有补水压力调节系统等。

通过以上锅炉的输入参数将会使炉膛内的凉水加热，使用传感器对总管内的水温检测，控制其水温和水压并输入到出口总管处。其后通过分水器，将总管细分为通往各个换热站的官网输送方向，也就是我们出口总管，此时的热水温度和压力通常较大，因为在流经官网的时候会有大量的热量损耗，并且依据各个供热站供热区域的大小将适度的调整出水总管的温度和压力。随后热水经过一次官网抵达各个供热站，经过二次官网流经各个供热户的小区，一般来讲各个管网叶子节点的地方也是供热住户的所

在地，此时会经过热交换的过程通过暖气片能采暖设备将热量带走，达到住户的生活温度，由于管网是双向的，此时管网会与路由交换技术一样根据压力向来的方向流经回去，一直达到锅炉房的回水总管。和之前一样，由于供热区域的不同，带走的热量也不尽相同，将回水和新入凉水进入混水器设备，调节到一定温度再次输入炉膛，如此循环经过锅炉房、分水器、出水总管、换热站、供热站、回水总管和混水器等设备及结构节点后，完成一次热量循环，从而达到将燃气转换为热量的物流过程，最终使供热户达到采暖需求。

2.4 本章小结

本章为相关技术研究，主要对聚类算法做了具体的介绍，对基于划分、层次、密度、网格、模型的方法进行了研究，对 Google 提出的 MapReduce 编程模型进行了介绍，同时对聚类算法的并行化及其应用进行了研究。

3 基于曲线相似度的供热过程评价方法

随着供热行业自动化水平的不断提高，“煤改气”和“十三五”规划节能减排的进行，我国的城市供热过程基本上实现了自动控制，供热品质得到改善，能源利用率得到提高。很多学者已经利用数据挖掘等技术对锅炉系统历史数据分析。孙群丽等对锅炉运行数据进行关联规则挖掘，提供了几组在不同负荷及外部条件下的最优运行方式与参数控制[60]；路海昌等通过对时间序列进行相空间重构，建立了基于支持向量回归的时序数据预测模型，从而实现对锅炉输出参数的预测[61]；岳晓忠采用后向反馈 BP 神经网络理论和关联规则算法的数据挖掘方法，对锅炉实时运行数据进行分析，从而建立锅炉运行模型[62]。以上的分析都是如何去优化控制锅炉，但实际操作人员由于各种原因较难掌握这些方法的使用，而供热公司也没有合适的方法对现场人员供热过程量化分析，即缺乏切实可靠的量化评价方法，难以满足管理者实时掌握考评状况和调整运行策略的需求[63]。因此研究供热过程的评价方法对推进节能减排、降低运行成本都具有重要意义。

本章首先解释目标温度曲线的由来并分析目标温度曲线与实际出水温度曲线之间不同的多种可能情况，然后将其分解为趋势变化、平移和伸缩三个属性相似度并给出各个属性的相应计算方法，并最后融合为一个一致性度量来评价供热过程。通过实验分析，证明了此评价方法的有效性，为相关管理人员量化管理提供了一种参考依据，避免仅以燃气、水、电等能耗来衡量供热行为，而是需要区分不同的供热过程，在满足供热户室内温度的情况下尽可能节能。

3.1 目标温度曲线与出水温度曲线分析

3.1.1 目标温度曲线

早期锅炉系统管理人员根据运行经验，根据室外温度会制定一个出水温度标准作为供热锅炉出水温度的参考，通常与室外温度为线性相关。在此基础上一段连续时间的出水温度就构成了目标温度曲线。随着节能减排的规划以及物联网技术的应用，目标温度的定义需考虑各方面因素确定，主要包括天气条件（室外温度、风速、日照），供热用户不同时间段的需求，回水温度，出水提前量等。其中天气条件将直接影响供热用户的采暖需求，回水温度反映了热量的利用情况，而供热公司也需要根据供热用户的作息规律、生活习惯、上班或在家的情况进行适度的调控，尽量节约能源，出水提前量指热水热交换后到用户家里的时间。因此是一个比较复杂的过程，需要考虑较

多因素，主要因素如公式（3-1）所示。

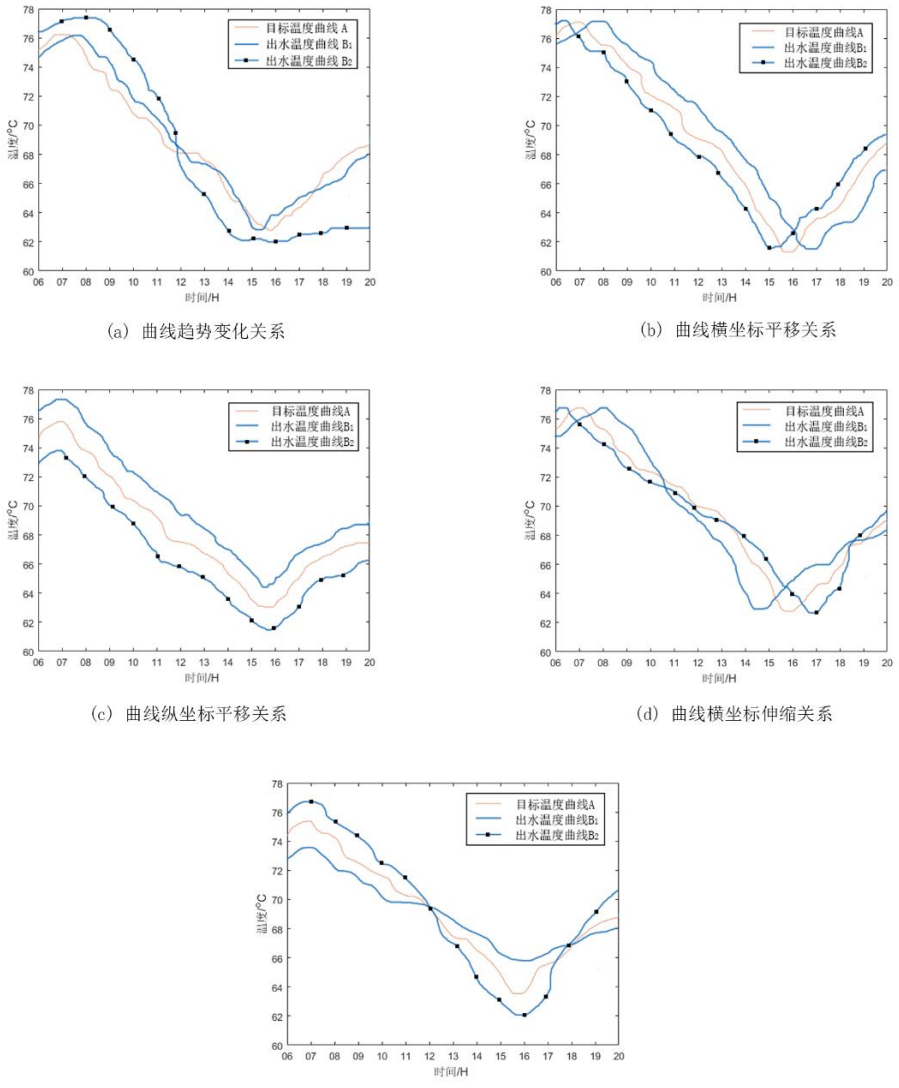
$$\mathrm{T(t)} = \mathrm{f}(T_t, W_t, S_t, O_t, B_t, E_t) \qquad (3-1)$$

式中t为时间，$T_t, W_t, S_t, O_t, B_t, E_t$分别为t时间的室外温度、风速、日照、偏移量、回水温度、出水提前时间。

目标温度曲线主要是根据专家经验综合以上因素并参考供热用户建筑计算或对大量历史采集数据分析预测得到。

3.1.2 目标温度曲线与出水温度曲线相似度

将目标温度曲线（A）与出水温度曲线（B）进行一致性分析，可以分析出锅炉房供热过程是否严格按照要求供热，及满足节能要求，为锅炉管理人员对操作人员量化



(a) 曲线趋势变化关系　(b) 曲线横坐标平移关系

(c) 曲线纵坐标平移关系　(d) 曲线横坐标伸缩关系

(e) 曲线纵坐标伸缩关系

图 3-1 出水温度与目标温度对比图

Fig. 3-1 Comparison of effluent temperature and target temperature

管理提供参考。

两条曲线的一致性存在多种情况，如图 3-1 所示。其中图 3-1-(a)中曲线 B1 基本与曲线 A 相同，而曲线 B2 的在 12 时后较曲线 A 的趋势有明显差异，表明曲线 B2 基本没有按照曲线 A 进行相应的调整，即两条曲线反映的供热调节趋势不一致；图 3-1-(b)表现了曲线间的横向平移问题，在供热数据上，横坐标的平移代表了出水温度与目标温度调整的提前和延时情况，纵坐标的平移代表了出水温度与目标温度的温差情况，即供热温度相比目标温度偏高或偏低。其中曲线 B1 在横坐标上较曲线 A 整体提前，而曲线 B2 在横坐标上较曲线 A 整体延后，表明虽然趋势调整基本正确，但调节的及时性存在问题；图 3-1-(c)表现了曲线间的纵向平移问题，曲线 B1 在纵坐标上较曲线 A 低了近 2 摄氏度，表明供热温度偏低，未能达到供热户供暖需求，而曲线 B2 在纵坐标上较曲线 A 高了近 2 摄氏度，表明供热温度偏高，造成了能源浪费；图 3-1-(d)中，曲线 B1 的最大值与最小值间横坐标差距相较于曲线 A 较小，说明该降温时间延后，而升温时间提前，造成了能源的浪费，而曲线 B2 的最大值与最小值间横坐标差距大于曲线 A，说明降温时间提前而升温时间延后，未能达到供热户需求，容易引发客诉；图 3-1-(e)中，曲线 B1 与曲线 A 相比，其最大值与最小值间的纵坐标差距更小，在高温时未能达到供热户需求，在低温时浪费热量，而曲线 B2 最大值与最小值间的纵坐标差距更大，在高温时浪费热量，在低温时未能达到供热户需求。

综合上述分析，两条曲线间主要存在趋势变化、平移和伸缩三个属性上不同。因此，出水温度曲线与目标温度曲线的一致性评价问题可以分解为对这三种属性相似度的计算问题。在供热数据中，曲线间的差异也代表了不同的供热行为：趋势变化主要反映两者在整体形态上的一致性，即操作人员能否按照要求进行合理的调控，横向时间的偏移体现了温度调整的提前或延后，纵向温度的偏移体现了供热温度的偏高或偏低；伸缩体现了曲线在升温和降温调整的不同。

3.2 供热效果评价

3.2.1 出水温度曲线与目标温度曲线的趋势变化

两条曲线趋势变化相同，可以认为锅炉系统能综合考虑各种因素对出水温度做出合理的调整，使供热户在不同的条件下得到最合适的热量。对两条曲线趋势的分析，可以认为是曲线的相似性问题[64]。

曲线的相似性测度一般有两种方法：距离测度法和相似性函数法[65]。相似性函数是用函数的方法来表征两曲线相似的程度，主要有夹角余弦和相关系数等方法，但由

于供热数据受天气和地域影响较大，很难拟合成相应的曲线函数，而距离测度法主要有 Euclidean 距离、Minkowsky 距离、Hausdorff 距离、Fréchet 距离等[66]，其中，Hausdorff 距离和 Fréchet 距离主要用来计算两个点集间的相似性，但 Hausdorff 距离忽略了点集的时间序列问题，基于供热系统时序数据的特点，本文采用 Fréchet 距离作为曲线趋势变化属性相似度的度量，其优点在于充分考虑了曲线的连续性，非常适用于曲线间的相似性比较[67]。

Fréchet 距离由 M. Fréchet 提出，描述了两质点分别沿着 2 条给定曲线以任意速度单向运动时，二者之间的最短距离。Axel Mosig 和 Michael Clausen 曾将 Fréchet 距离与变换群的交叉子集结合，应用到判别两条曲线的相似性上[68]，曹凯等引入 Fréchet 距离进行云规则推理，设计了一种智能地图匹配算法[69]。Eiter 和 Mannila 在连续 Fréchet 距离的基础上提出了离散 Fréchet 距离[70]的定义，而朱洁等考虑了离散 Fréchet 距离的关键特征峰值点，减少了算法的复杂度并将其运用到了手写签名验证上[71]，收到了一定的效果。

离散 Fréchet 距离定义如下：

（1） 给定 1 个有 n 个至高点的多边形链P = $\langle P_1, P_2, P_3, \dots, P_n, \rangle$，1 个沿着 P 的 k 步，分割 P 的峰值点成为 k 个不相交的非空子集$\{P_i\}_{i=1,\dots,k}$，使得$P_i = \langle P_{n_{i-1}+1}, \dots, P_{n_i} \rangle$和$1 = n_0 < n_1 < \cdots < n_k = n$

（2） 给定 2 个多边形链A = $\langle a_1, \dots, a_m \rangle$，B = $\langle b_1, \dots b_n \rangle$，1 个沿着 A 和 B 的组合步是 1 个沿着 A 的 k 步$\{A_i\}_{i=1,\dots,k}$和 1 个沿着 B 的 k 步$\{B_i\}_{i=1,\dots,k}$组成，使得对于 $1 \le$ i $\le$ k，A_i, B_i中有 1 个恰好包含 1 个至高点。

（3） 1 个沿着链 A 和 B 的组合步W = $\{(A_i, B_i)\}$的花费(cost)为：

$$d_F^W(A,B) = \max_i \max_{(a,b)\in A_i \times B_i} dist(a,b)$$

其中 dist()为 a,b 间的欧氏距离，则链 A 和 B 间的离散 Fréchet 距离为：

$$\mathrm{F(A,B)} = \min_W d_F^W(A,B) \qquad (3-2)$$

3.2.2 出水温度曲线与目标温度曲线的平移问题

出水温度曲线与目标温度曲线的平移表现为两者横纵坐标的差异，在温度曲线中，横坐标的度量为时间，表现为出水温度调整的提前或延后问题；而纵坐标的度量为温度，体现在两条曲线的温差问题，具体为锅炉系统是否能按需达到预定温度以及在不需要较多热量时降低负荷节约能源。

3.2.2.1 时间差异

时间差异定义为两条曲线的 n 个同一维度上特征点间的时间差均值。而特征点的确定会对度量结果产生较大影响，考虑到供热锅炉数据的特点，在同一维度上很难找到成对的特征点[72]。这里先获得两条曲线的最值，然后按照最值将每条曲线单独划分为多个单调区间，对于单调增区间最小值为初始特征点对，否则最大值为初始特征点对，然后在各区间根据初始特征点对的类型计算下一个特征点对，取其时间的差值作为时间差异。依次计算出每个时间段的差异集合，最后将这些差异的均值作为差异度量。

$$D_X = \frac{\sum_{j=1}^{k}\sum_{i=1}^{m}\left(X^{A}_{\max_{ji}} - X^{B}_{max_{ji}}\right)}{n'} \tag{3-3}$$

式中n'为特征点的个数，m 为每个单调区间的点数，A 为目标温度，B 为出水温度，则$X^{A}_{max_{ji}}$为目标温度曲线在第 j 个区间温度为第 i 大值的时间，供热控制较差时可在合理时间段取值。

3.2.2.2 温度差异

温度差异简称温差，主要表现在供热温度较低时是否能满足采暖需求，反之是否发生能源浪费。这里将分两方面考虑，曲线的最大值差和最小值差。

两条曲线最大值处的温差，能够判断供热效果，出水温度是否能按需达到采暖需求，为供热用户提供足够的热量。目标温度较高时通常意味着用户在家或者气象条件不能提供较多自然热量，所以需要锅炉系统高负荷运转提供充足热量，也可以一定程度上减少客诉。

两条曲线最小值处的温差，能够判断供热锅炉系统是否节约能源。目标温度较低时通常意味着，用户家中无人，或者室外温度等气象条件能提供较多自然热量，故锅炉系统需要降低运行负荷减少热量，节约能源降低运行成本。

综合上述两个方面将两条曲线的温差定义如下：

$$D_Y = \frac{Y^{A}_{min} - Y^{B}_{\min} + Y^{A}_{max} - Y^{B}_{max}}{2} \tag{3-4}$$

A 为目标温度，B 为出水温度，Y^{A}_{min}为目标温度曲线最小值点的温度。

3.2.3 出水温度曲线与目标温度曲线的伸缩问题

出水温度曲线与目标温度曲线的伸缩问题，在其横坐标上表现为锅炉系统在时间上，是否按统一节奏对锅炉系统进行调控，对天气情况的变化是否做出时间一致的操

作；在其纵坐标上表现为锅炉系统对温度控制的灵敏度，在高低温转换时可以及时达到预期温度。

借助离差标准化的思想，最值差可以完整的表现整体的数据跨度[73]，将横纵坐标的最值差比作为两条曲线的伸缩比，能较好的反映数据整体的特点，对其横向和纵向伸缩比的计算方式如式（3-5）和（3-6）所示.

$$E_X = 1 - \frac{X_{max}^B - X_{min}^B}{X_{max}^A - X_{min}^A} \tag{3-5}$$

$$E_Y = 1 - \frac{Y_{max}^B - Y_{min}^B}{Y_{max}^A - Y_{min}^A} \tag{3-6}$$

式中X_{max}^A为目标温度曲线最大值的横坐标， Y_{min}^B为出水温度曲线最小值的纵坐标。

3.2.4 评价结果

将三种属性相似度共五个度量加权融合为出水温度曲线和目标温度曲线的一致性度量：

$$\mathrm{sim(A,B)} = \frac{\omega_1 F}{\varepsilon_F} + \frac{\omega_2 |D_X|}{\varepsilon_{D_X}} + \frac{\omega_3 |D_Y|}{\varepsilon_{D_Y}} + \frac{\omega_4 |E_X|}{\varepsilon_{E_X}} + \frac{\omega_5 |E_Y|}{\varepsilon_{E_Y}} \tag{3-7}$$

式中ω_1、ω_2、ω_3、ω_4、ω_5分别为趋势、横向平移、纵向平移、横向伸缩、纵向伸缩属性相似度的权值，$\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 1$，可通过数据统计及最小二乘法得出[74]。$\varepsilon_F$ 、ε_{D_X} 、ε_{D_Y} 、ε_{E_X} 、ε_{E_Y}分别为五个属性相似度的阈值。

本文通过分析出水温度曲线与目标温度曲线间的一致性，分别给出曲线的趋势变化、平移、伸缩三种属性相似度的计算方式，并将其加权融合为一个评价结果，用来对锅炉供热过程进行评价。

3.3 本章小结

本章提出了一个基于曲线相似度的供热行为评价方法，通过分析室外温度和供热锅炉系统出水温度的关系，对其趋势、横向平移、纵向平移、横向伸缩、纵向伸缩做了依次的总结，最后将五种属性相似度进行加权得到一个一致性度量，将对供热锅炉系统的供热行为进行评价。

4 基于 K 均值的供热过程评价并行算法

为了研究天津区域供热锅炉系统不同的控制行为，其供热行为具有较大差异，为了分析数据采集系统中的数据价值，采用基于划分的 K 均值聚类算法，可以将相同的供热行为进行划分，不同的供热行为进行分割，且考虑到供热采集系统的历史数据较为庞大，所以将传统的 K 均值算法进行并行化并且与第三章中的评价模型进行结合，加快算法的运算速度。

4.1 K 均值算法研究

K 均值（K-means）由 MacQueen 最早提出，它是一个基于划分的聚类算法的直接实现，介于其算法思想简单，串行算法的实现相对容易的特点，在多个计算机交叉学科里面都有广泛的应用。它是聚类算法中最常见的划分方法，所谓的划分方法就是给定一个包含 n 个数据对象的数据集，将数据集划分为 k 个子集，其中每个子集均代表一个聚类，同一聚类中的对象相似度较高，而不同聚类中的对象相似度较小。聚类相似度是利用各聚类中对象的均值获得一个“中心对象”来进行计算。划分方法也就是说将数据分为 k 组，这些组满足以下要求：

1) 每组至少应该包含一个对象；

2) 一般的划分中每个对象必须且只能属于一个组；

3) 在一些模糊划分中可以允许每个对象属于多个组。

k 均值算法的工作过程如下：首先从 n 个数据对象中任意选择 k 个对象作为初始的聚类中心，而对于所剩下的其他对象，则根据它们与这些聚类中心的相似度，一般相似度是根据具体需要确定，比如对于文本的聚类多采用余弦相似度，而对于数据的聚类可以采用欧氏距离计算相似度。计算相似度之后分别将他们分配给与其最相似的聚类；然后再计算每个所获新聚类的聚类中心(该聚类中所有对象的均值)；不断重复这一过程直到相似度函数开始收敛为止。

算法：根据聚类中的均值进行聚类划分的 k 均值算法
输入：聚类个数 k，以及包含 n 个数据对象的数据集
输出：满足方差最小标准的 k 个聚类
处理流程：
1. 从 n 个数据对象中任意选择 k 个对象作为初始聚类中心。
2. 分别计算剩下的数据对象到这 k 个中心的距离或者相似度，根据实际需求来选择距离度量。
3. 将这些数据对象分别划归到距离最近或者相似度最高的聚类。

4. 根据聚类结果，重新计算 k 个聚类各自的中心，计算方法是取簇中所有元素各自维度的算术平均数。
5. 将数据集中全部数据对象按照新的中心重新聚类。
6. 重复以上操作，经过 t 次迭代计算直到聚类结果不再变化或者变化趋于指定的收敛域为止。
7. 聚类结束，将结果输出。

4.2 基于 K 均值的供热行为评价并行算法

根据 MapReduce 编程模型设计算法的流程，主要设计 Map 函数、Combine 函数以及 Reduce 函数分别实现途中的算法执行过程，为了更好的处理供热数据，首先对是否参与聚类的属性维度进行分析，然后将属性相似度作为 Map 函数的输入，因此我们在第一次 MapReduce 迭代之前加入一个 PreMap 函数对数据进行计算。

4.2.1 降维预处理

参与聚类的数据其维度的复杂和数据的大小皆会影响聚类的效果，当我们考虑高维数据的时候，合理的降维是一个很重要的方面，降维应该遵循一定的方法。降维的方法之一是主成分分析，它在分析时考虑维度的方差和维度间的相关性，然后对具有相关性的维度进行综合。而本文的曲线属性相似度具有五个度量，分别为趋势、横向平移、纵向平移、横向伸缩、纵向伸缩。本文认为五个维度的相关性不大，所以根据在关联规则算法 CLIQUE 中采用的先验规则表示，如果聚类的某一维是密集的，那么它对于整个 k 维聚类也是可用的，否则，在整个 k 维数据聚类中我们认为它仍然不起作用。此处我们把这一关联规则挖掘中的性质应用到非空间属性的降维中，提出了另外一种判断方法。

本文通过对五个属性相似度进行标准差的计算，标准差可以反映一个数据集的离散程度，使用式 4-1 对各个属性数据进行计算，使用如果标准差较小，则表示此属性数据离散程度较小，比较聚集，对聚类结果影响较小，反之则可能对聚类结果有较大影响；另一方面，有时数据比较分散，但在整个数据空间中的分布相对均匀，则对聚类结果影响也较小，因此可以通过对属性维度进行频数统计，做出频数分布直方图，如果结果符合正态分布或者较为分散，则表明本属性数据比较分散，同类的相似度较小，同样不会表达出较好的聚类结果。

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (4\text{-}1)$$

通过对属性数据标准差的计算，以及频数分布直方图的结果情况表达，可以为参与聚类的维度作参考，达到降维的目的，从而使聚类的结果更好。

4.2.2 PreMap 函数设计

PreMap 函数主要对第三章的属性相似度进行并行化的计算，由于供热锅炉系统一天当中产生的数据量极大，使用 Hadoop 的多节点集群也将大大加快这个过程。PreMap 函数的主要过程是依次计算离散 Frechet 距离、横向平移、纵向平移、横向伸缩、纵向伸缩，将一天的实时数据计算完毕后得出五个属性相似度，作为下个阶段 Map 函数的输入，其中输出格式为 key/value 的键值对形势。

算法：PreMap 函数伪代码

```
输入：n 个站点的 N 条数据
输出：n 个站点 m 天的属性相似度<key, value>
preMap(data) {
   for(i = 0; i <= data[0].length; i ++) do {
      1 得到数据集中一个单位一天的数据
      2 对缺失值进行插值处理
      3 得出目标温度曲线 T(t)
      4 依据公式依次计算目标温度曲线与出水温度曲线的 F、D_X、D_Y、E_X、E_Y
   }
}
```

4.2.3 Map 函数的设计

Map 函数的任务是先构建一组全局的初始聚类中心，Map 函数将文件的每行作为一个样本，从样本中提取出有用的数据，以 key/value 键值对形式表示，并计算每个样本到各个聚类中心的距离，选择距离最小的聚类中心，并把该数据样本分配到该聚类中，并把该数据样本标记为所属的新聚类类别，形成 key/value 键值对的输出形式。Map 过程的输入为待聚类所有数据对象和上一轮迭代后产生的聚类中心，<key, value>对是 MapReduce 框架默认的形式，输入数据以<行号, 记录行>的形式表示<key', value'>对；Map 函数根据全局的聚类中心计算得到与每个输入样本距离最近的聚类中心，并做新聚类的标记；输出中间结果以<聚类 ID, 记录属性向量>的形式表示<key, value>对。Map 函数输入的是当前记录相对于输入数据文件起始点的偏移量，value 是当前记录的各维坐标值。先从 value 中解析出当前记录各维的值；然后分别计算其与 k 个聚类中心的距离，找出距离最近的聚类 ID;最后输出<key', value'>，其中 key'是距离最近的聚簇 ID，value'是当前记录的各维坐标值。

Map 函数的输入数据存储在每一台主机的硬盘中，其文件格式为一个接一个的<key, value>对，每一个<key, value>对即为一个输入数据，另外，每台主机还拥有一个全局的聚类中心表。根据这两个信息，函数可以计算得到与每个数据对象最为相似的

聚类中心。Map 函数的输出数据同样为<key', value'>对的形式，key'表示聚类 ID; value'表示与该聚类中心最为相似的数据对象。Map 函数的伪代码如表所示：

算法：Map 函数

```
map(<key, value>,<key', value'>)
{
   定义 instance 数组，记录从 value 中解析出每个样本的各个维度值
minDis = Math.max(); 辅助变量 minDis 初始化为最大值
index = -1; index 初始化为-1；
for(i = 0; i <= k – 1; i ++) do {
   dis = instance  与第 i 个聚类中心各维度值得距离；
   if dis < minDis{
minDis = dis;
index=i;
}
}
将 index 作为 key';
将各维坐标值作为 value';
输出<key', value'>;
}
```

为了方便中间结果在本地进行预处理，以此减轻集群的时间耗费和通信压力，一般会在 Map 操作之后插入一个 Combine 操作，但 Combine 操作也不是每个程序都适用的。将函数输出数据<key', value'>对进行本地合并，相当于本地化的操作。由于 Map 的输出数据都暂存在本地的 HDFS 分布式文件系统中，所以添加 Combine 操作可以减小集群的通信量和传输所耗费的时间，同时加快下一步 Reduce 操作的执行时间。

4.2.4 Combine 函数的设计

Combine 函数的作用就是对 Map 过程产生大量中间结果进行本地化处理，可以减轻数据在节点之间的传输时间耗费和带宽占用。Combine 函数对所处节点内的 Map 结果进行预处理，对具有同一 key 值的 value 值进行处理，然后将处理得到的局部聚类结果，紧接着，再传给集群中的 Reduce 函数进行规约操作，这就是 Combine 函数的任务[63]。Combine 函数的输入形式为 key/value 键值对，key 代表聚类类别 ID，value 代表聚类为 key 的记录的各维坐标值。Combine 函数首先解析出每个记录的各维坐标值，然后，将各维坐标值相加，得到局部聚类结果的累加和，计算得到总的样本个数。输出的<key', value'>对中 key'是聚类 ID;value'是记录总数和各维坐标值的累加和。

Map 任务完成后，MapReduce 算法会启动一个 Combine 任务来合并那些具有相同聚类 ID 的中间结果数据。由于中间结果数据都是本地化存储的，该过程不会产生

集群的通信开销。在 Combine 函数中，对本地的具有相同聚类 ID 的记录向量进行求和。为了得到新的全局聚类中心，还需要统计节点中每个聚类的记录个数。Combine 函数的伪代码如表：

算法： Combine 函数

combine(<key, value>,<key', value'>)
{
初始化一个用于存储各维坐标值累加的数组，初始值都为 0；
初始化变量 num 为 0，用于统计相同聚类的记录数目；
while(value.hasNext()){
解析 value 中每个记录的各维坐标值；
将各维坐标值累加存放到数组中；
num++;
}
将 key 作为 key';
构造存储 num 和数组信息的字符串，作为 value';
输出<key', value'>;
}

4.2.5 Reduce 函数的设计

Reduce 函数通过汇总 Combine 函数得到的局部聚类结果计算出新的聚类中心，并将其用于下一轮迭代运算[64]。Reduce 函数输入数据的形式为 key/value 键值对，key 代表聚类类别 ID，value 代表各个 Combine 函数得到的中间结果(intermediate result)，Reduce 函数首先计算每个节点输出的局部聚类结果的样本个数，并解析每个样本的各维坐标值，然后将对应的各维累加值分别对应相加，再除以刚才计算得到的总样本个数，计算结果就是新的聚类中心坐标。输出结果的形式为 key/value 键值对，key 代表聚类类别 ID，value 代表计算得到的新聚类中心。

Reduce 过程的输入是 Combine 过程之后得到的局部聚类结果，将得到的局部聚类结果进行合并，生成全局聚类结果。Reduce 函数的输入是从 Combine 任务得到的每个节点的数据，该数据包括了同一类中的记录向量的累加和和以及记录个数。然后，将所有节点上同一聚类的记录向量累加并计算类中总的记录个数，得到新的聚类中心，作为下一轮迭代运算的聚类中心。Reduce 函数的伪代码如表所示：

算法： Reduce 函数
reduce(key, value),<key', value'> { 初始化一个用于存储各维坐标值累加的数组，初始值都为 0； 初始化 NUM 为 0，用于统计相同聚类的总的记录个数；

```
    while(value.hasNext()) {
    解析 value 中的各维坐标值和记录个数 num;
    将各维坐标值累加存放到数组中；
    NUM+=num；
}
将数组中的各个分量除以 NUM，得到新的聚类中心；
将 key 作为 key’;
构造一个包含新聚类中心各维坐标值的字符串，作为 value’;
输出<key’, value’>;
}
```

在执行完 Reduce 任务之后，根据 Reduce 的输出结果计算新的聚类中心，并更新到 HDFS 分布式文件系统中，并将该文件复制到集群中的所有节点上，然后，计算连续两轮 MapReduce Job 的误差平方和准则函数，若差值小于设定的值，则聚类准则函数已收敛，算法结束；否则，将新的聚类中心替换原来的聚类中心，启动新一轮迭代运算，同样是 Map 任务，Combine 任务和 Reduce 任务的流程。当迭代输出趋于稳定收敛时，就可以得到最终的聚类结果。

4.3 本章小结

本章内容在传统串行 k-均值的算法的基础上对其进行并行化，同时加入曲线相似度属性相似度的计算，从而加快 k-均值在供热领域上数据处理的速度，对 PreMap 函数、Map 函数、Reduce 函数以及 Combine 函数进行了设计，完成了 k-均值的并行化。

5 实验及结果

本章将在前几章理论支撑的基础上进行实验，本实验数据来自天津地区供热锅炉公司提供的 2013-2016 供热季共十几个供热站的历史数据共 388800 条数据。其中包含气象仪数据，风速、日照、温度、湿度等，包含锅炉系统数据，供水温度、回水温度、出口总管温度、回水总管温度、供水压力、回水压力、出口总管压力、回水总管压力。在对原始数据从 SQL Server 数据库中进行提取，筛选部分数据，经过数据清洗和数据预处理等操作，弥补供热锅炉系统在实时监测时传感器发生的异常值和缺失值，将其作为本实验的数据集。

通过本文提出的曲线相似度供热行为评价方法，对各个换热站每日的出水温度和气象条件的关系分析度量值，将度量值作为数据的各个维度，分析锅炉房的供热行为，利用 MapReduce 对此大数据集进行处理，使用并行化的 k-means 进行聚类从而分辨不同的供热行为。本章首先介绍实验环境的搭建，利用 Apache 提供的 Hadoop 搭建实验集群环境，然后使用本文提出的供热过程评价模型进行计算，通过结果进行可视化展示，分析聚类结果，帮助锅炉管控人员更好的绩效分析和对只能自动系统进行微调。

5.1 环境搭建

5.1.1 Hadoop 集群资源规划

为了充分利用有限的实验资源，将使用四台虚拟机搭建在 VMware 环境中，在宿主机中将硬盘资源和内存切分为多个虚拟机进行实验，其宿主机和虚拟机的配置分别如下：

宿主机：硬盘 700GB 内存 16GB CPU I7 4760 四核处理器 操作系统 Windows 10 企业版

虚拟机：硬盘 20GB 内存 1GB CPU I7 4760 单核处理器 操作系统 Linux Kali

根据以上配置内容，规划的拓扑结构如下图所示：

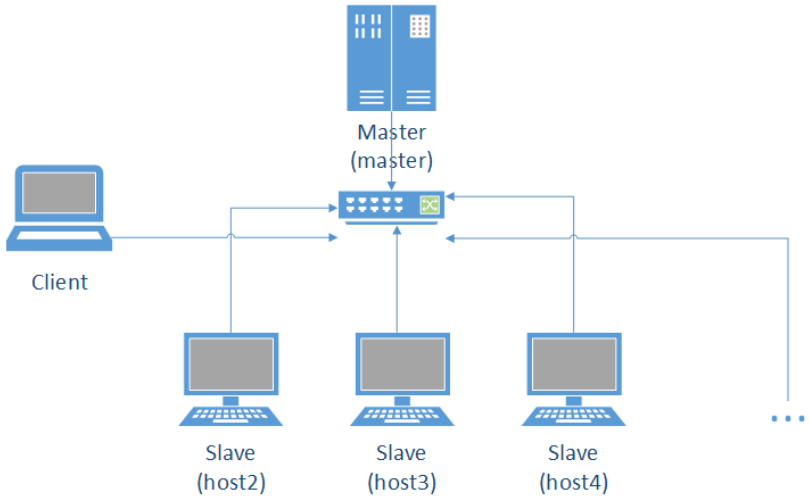


图 5-1 Hadoop 集群拓扑图及 IP 资源规划

Fig. 5-1 Hadoop Cluster Topology and IP Resource Planning

5.1.2 Hadoop 环境配置

1. 使用对 Windows 支持较好的 VMware 软件将宿主机分割，依据 IP 地址分配虚拟机地址

2. 为每台虚拟机安装 Linux 系统，这里选用 Kali，使用的版本是 kali-linux-1.0.9a-i386，Kali 的前身是 backtrack，是一个基于 Debian 的 Linux 操作系统，内置诸多易用的黑客工具，开放超级管理员权限，并裁剪内核

3. 由于实验所用宿主机为本机，为了便捷访问各个节点，需要安装 SecureCRT 连接工具，方便进行上传文件和访问节点操作

4. 使用 CRT 等软件登录虚拟机，安装常用的 vim、ssh 等软件

```
sudo apt-get install vim
sudo apt-get install ssh
```

5. 修改各个节点的主机名和网络配置，分别修改 hostname 文件和 interface 文件，依据拓扑结构填写已经规划和分配好的 IP 地址，子网掩码和 DNS 服务器及其网关

```
sudo vim /etc/hostname
sudo vim /etc/network/interface
```

6. 修改 hosts 文件，hosts 文件存储了主机名和 IP 地址的映射关系，当发生网络请求时，若请求的网络主机名存在于 hosts 文件中，则不进行 DNS 查询，优先使用

hosts 中的 IP 地址

```
sudo vim /etc/hosts
```

7. 配置 SSH，实现无密码登录。SSH 可以理解为加密的 telnet 信道，几点之间可以通过加密的 vty 通道进行数据访问，控制存储更操作，在进行 SSH 配置之后，每次数据交互时无需再次输入密码，这在 Hadoop 中的 master 节点和 slave 节点之间具有很重要的意义。

使用 ssh-keygen -t rsa 然后敲击三次回车生成默认的 SSH 握手文件，打开~/.ssh 目录，在 master 上将公钥放到 authorized_keys 中，使用命令：

```
sudo cat id_rsa.pus >> authorized_keys
```

然后将此 authorized_keys 文件拷贝至其他 slave 节点中的~/.ssh 目录下：

```
sudo scp authorized_keys hadoop@10.10.11.192:~/.ssh
```

其中 scp 的命令格式为 远程主机用户名@远程主机名或 IP：存放路径。

修改 authorized_keys 文件权限，使文件生效，完成 SSH 免密登录配置：

```
chmod 644 authorized_keys
```

8. 安装 Java 环境。由于 Hadoop 基于 Java 编写，所以需要使 Java 环境生效，通过 SecureCRT 的 rz 命令将文件上传至各个节点中，将文件放到/usr/lib/java 中，解压缩后设置环境变量，追加 PATH 路径，然后编译生效：

```
sudo vim ~/.bashrc
export JAVA_HOME=/usr/lib/java/jdk1.8.0_40
export PATH=$JAVA_HOME/bin:$PATH
source ~/.bashrc
```

9. 上传 Hadoop，并配置。还是通过 SecureCRT 的 rz 命令，将 Hadoop 上传至/usr/local/下，解压缩文件并重命名文件

```
tar -zxvf hadoop1.2.1.tar
sudo mv hadoop1.2.1 hadoop
```

修改环境变量使 hadoop 命令生效：

```
sudo vim ~/.bashrc
export HADOOP_HOME=/urs/local/hadoop
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$PATH
source ~/.bashrc
```

分别修改/usr/local/Hadoop/conf 下的配置文件，包括 hadoop-env.sh/core-

site.xml/hdfs-site.xml/mapred-site.xml/master/slave 等文件，其目的是对 HDFS 核心文件配置，MapReduce 核心站点配置以及声明 master 和 slave 主机。

hadoop-env.sh

```
export JAVA_HOME=/usr/lib/java/jdk1.8.0_40
```

core-site.xml

```
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/usr/local/hadoop/tmp</value>
</property>
```

hdfs-site.xml

```
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>heartbeat.recheckinterval</name>
<value>10</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/usr/local/hadoop/hdfs/name</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/usr/local/hadoop/hdfs/data</value>
</property>
```

mapred-site.xml

```
<property>
<name>mapred.job.tracker</name>
```

```
<value>master:9001</value>
</property>
```

masters

```
master
```

slaves

```
host1
host2
host3
host4
…
```

5.1.3 Hadoop 集群的启动

对 NameNode 进行格式化，仅格式化一次即可

```
hadoop namenode -format
```

启动 hadoop

```
cd /usr/local/hadoop/bin
./start-all.sh
```

通过 jps 命令可以查看 hadoop 在各个节点上的运行情况

图 5-2 Hadoop 集群启动成功状态图

Fig. 5-2 Hadoop Cluster Startup Successful Statechart

5.2 实验过程

依据供热习惯及节能需求，目前很多供热单位晚十点半左右开始维持在某一较低温度，早四五点左右开始升温，即在夜间会将锅炉系统的出水温度维持在较低的水平，所以只对每天 6:00 至 20:00 的数据进行分析。

依据（3-1）式专家根据运行经验确定一天的目标温度调整规律，其中某日的甲锅炉房出水温度曲线及其温度曲线的对比如图 5-3 所示。从图中可以看出，目标温度曲

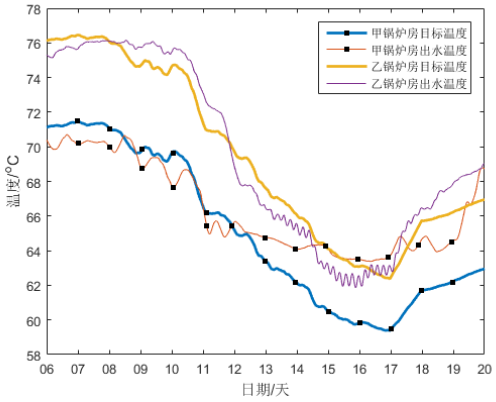


图 5-3 甲乙锅炉房出水温度及目标温度对比图

Fig. 5-3 A and B boiler room water temperature and target temperature comparison chart

线在上午 7 时左右达到最大值，为一天最冷的时间，供热户在此时间段在家居多，所以需要提供较高热量。随着室外温度、日照等气象条件的提高，出水温度将逐渐降低，而在下午 4 时开始，供热公司考虑到更多用户将会回到家中，且室外温度和日照逐渐降低，所以升高锅炉出水温度，为用户提供更多热量。

为了测试曲线相似度算法的合理性，本文首先通过较少数据进行验证，通过 SecureCRT 终端软件将数据上传至 Hadoop 中的 HDFS 分布式存储系统上，运行 MapReduce 算法，对数据进行 k-means 并行聚类，进行 k-means 聚类时需要确定要划分为几类的 K 值，依据新的 K-均值算法最佳聚类数确定方法，将 K 值设定为 K=3 或者 4，依次进行实验，其中部分聚类结果如表 5-1 和 5-2 所示。

表 5-1 K 为 3 时 k-means 聚类结果（部分）

锅炉房	F	D_X	D_Y	E_X	E_Y	类标号
乙 1	2.46	2.24	-0.21	-0.27	-0.26	1
乙 2	2.07	12.59	0.41	0.36	0.02	1
乙 3	3.64	1.71	-0.69	-0.25	0.29	1
乙 4	3.85	-2	-1.06	-0.08	0.24	1
乙 5	4.52	6.2	-0.37	-0.19	0.23	1
乙 6	4.03	6.06	-0.76	-0.32	0.21	1
乙 7	2.97	10.15	0.32	-0.74	0.06	1
甲 2	7.83	41.4	-2.62	0.97	-1.26	2
甲 7	5.04	48.4	-0.49	0.6	-2.8	2

甲 1	7.51	16	-3.54	0.87	-1.86	3
甲 3	4.84	27	-2.38	-0.94	-0.62	3
甲 4	11.61	11	-5.68	0.26	-1.45	3
甲 5	8.99	17.82	-4.3	-0.51	-1.37	3
甲 6	5.7	23.73	-1.77	0.97	-1.84	3

表 5-2 K 为 4 时 k-means 聚类结果（部分）

锅炉房	F	D_X	D_Y	E_X	E_Y	类标号
甲 4	11.61	11	-5.68	0.26	-1.45	1
乙 2	2.07	12.59	0.41	0.36	0.02	1
乙 5	4.52	6.2	-0.37	-0.19	0.23	1
乙 6	4.03	6.06	-0.76	-0.32	0.21	1
乙 7	2.97	10.15	0.32	-0.74	0.06	1
甲 2	7.83	41.4	-2.62	0.97	-1.26	2
甲 7	5.04	48.4	-0.49	0.6	-2.8	2
乙 1	2.46	2.24	-0.21	-0.27	-0.26	3
乙 3	3.64	1.71	-0.69	-0.25	0.29	3
乙 4	3.85	-2	-1.06	-0.08	0.24	3
甲 1	7.51	16	-3.54	0.87	-1.86	4
甲 3	4.84	27	-2.38	-0.94	-0.62	4
甲 5	8.99	17.82	-4.3	-0.51	-1.37	4
甲 6	5.7	23.73	-1.77	0.97	-1.84	4

可见，当 K 值为 3 时具有较好的聚类效果，可以将不同的供热锅炉房进行区分，将属性相似度中的离散 Frechet 距离和延时取出进行作图可得图 5-4，取出两簇的中心

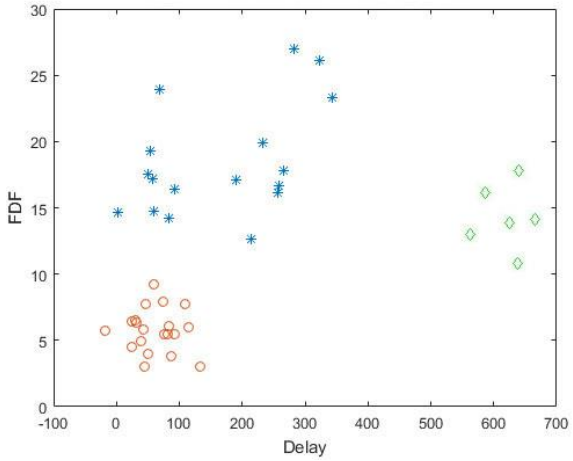


图 5-4 并行 k-means 聚类算法效果图(部分)

Fig. 5-4 Parallel k-means clustering algorithm

点进行具体分析，其中两个锅炉房站点的部分一致性度量对比如图 5-5 所示。

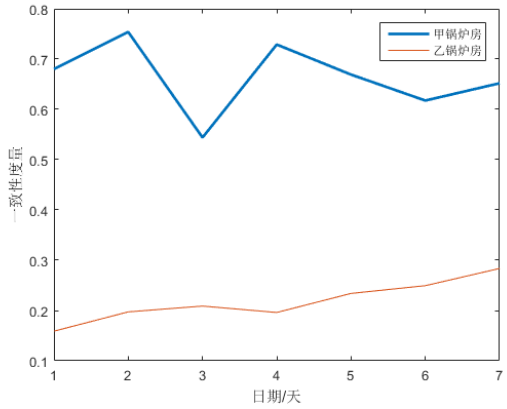


图 5-5 甲乙锅炉房一致性七天对比

Fig. 5-5A boiler room consistency seven days contrast

可以看出依据本评价方法乙锅炉房的出水温度与目标温度一致性更高，查询原始采集数据可知，甲锅炉房的单位耗气量为 $10.6m^3/m^2$，乙锅炉房的单位耗气量为 $9.2m^3/m^2$，因此算法是有效的。

由图 5-5 可知，第二天和第四天的一致性度量基本相同，进一步分析其原因，对数据按照式（5-1）进行离差标准化后，直观的对比图如图 5-6 和图 5-7 所示。

$$x^* = \frac{x - min}{max - min} \qquad (5-1)$$

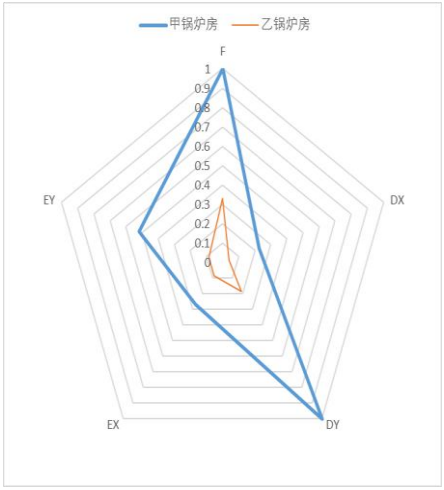


图 5-6 甲、乙锅炉房第四天属性相似度对比图

Fig. 5-6 A, B boiler room on the fourthday of similarity comparison

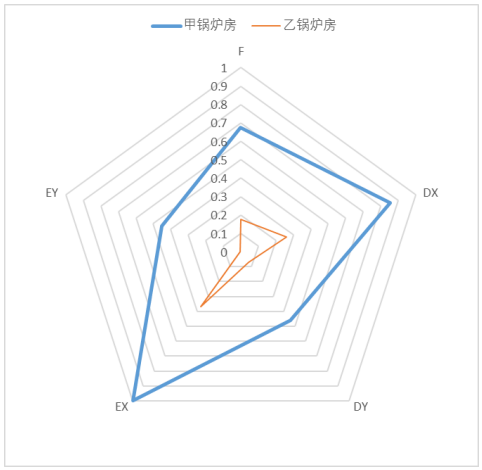


图 5-7 甲、乙 锅炉房第二天属性相似度对比

Fig. 5-7 A, B boiler room the next day similarity comparison of attributes

甲锅炉房第二天的纵向伸缩基本一致，趋势变化和纵向平移相较于第四天数据更优，横向平移和横向伸缩相对较差，表现为时间延时较小能及时为用户提供热量，但随目标温度的调控较差。乙锅炉房第二天的趋势变化、纵向平移和纵向伸缩相较于第

四天数据更优，而横向平移和横向伸缩相对较差，供热上表现为能为用户提供足够热量，随目标温度合理调控，但其时间延时相对较大。乙锅炉房五个度量皆优于甲锅炉房，供热上表现为乙锅炉房可以按照目标温度进行及时的适度调控，其时间延时较低，相较于甲锅炉房，能在高温时达到供热户需求，低温时能及时降温节约能源。

实际运行中乙锅炉房工作质量较高，根据天气、用户生活习惯、回水温度等各种因素积极调整运行参数，采取自动控制系统来调节出水温度；而甲锅炉房只是按照室外温度进行人工控制，没有精细化供热，其时效性表现较差。所以乙锅炉房的出水温度曲线在与目标温度曲线的一致性上更为接近。两个锅炉房供热户的投诉都很少，则以上数据说明在保证用户室内温度的情况下，乙锅炉房在一定程度上减少了煤气消耗，节约了能源。

在进行并行算法效果评价时通常会使用加速比（Speedup）分析。其方法是，保持数据大小不变，逐渐增加集群节点的个数，则 n 个集群节点的加速比计算公式如式 5-2 所示：

$$\text{Speedup(n)} = \frac{T_1}{T_n} \tag{5-2}$$

其中 n 为集群节点的数目，T_1为单机或单节点伪集群的算法运行时间，T_n为 n 个节点的集群处理算法的运行时间。

将数据集分别切割为：全部数据、1/2、1/4、1/8、1/16，为了图表中表示方便分别标识为 D1、D1/、D1/4、D1/8、D1/16，通过参数设置 Hadoop 集群的节点数量分别为 1-8 个节点，对不同量级的数据集进行计算，其加速比对比如图 5-8 所示。

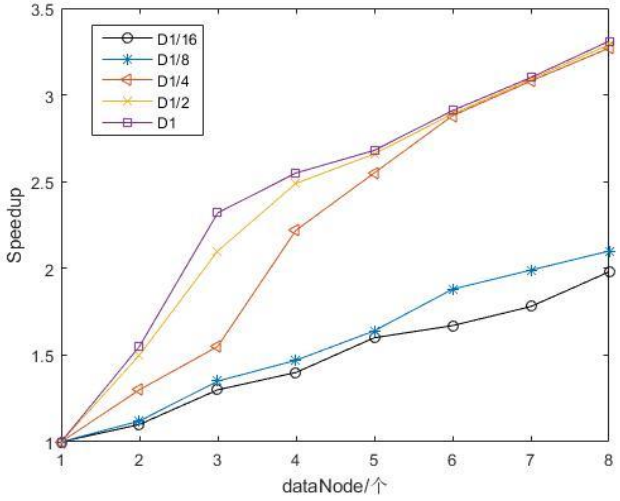


图 5-8 不同节点下的加速比分析

Fig. 5-8 Acceleration Ratio Analysis under Different Nodes

由图 5-8 可以看出，并行算法的加速比随着 Hadoop 中集群节点的个数 n 的增多，保持着线性的增长，这表示了使用 Hadoop 的集群多节点模式可以提高算法的效率，并且随着数据量级的增大，算法的加速比 Speedup 的性能也会表现更好，证明了本算法的 Map 函数与 Reduce 函数的键值设定比较合理，此外，算法添加了 Combine 的过程，这也使算法的执行在通信上没有太大的时间耗费，因此，当数据集规模越大时，算法的加速比越好。

5.3 实验结论

为了更好地量化管理，达到满足供热的前提下节能减排的目的，本文提出了一个锅炉供热过程的评价方法。对锅炉房的出水温度曲线与目标温度曲线进行一致性分析，分解为趋势变化、平移和伸缩三种属性相似度分析其一致性，给出每个属性相似度的计算方式并加权融合为一个一致性度量。然后利用天津某供热公司的供热数据，通过本文提出的供热行为评价方法将进行了对比分析，同时，为了研究不同的一致性属于何种管控方式使用聚类的方式将数据进行聚簇，并且为了加快算法的执行效率，使用 Hadoop 的 MapReduce 编程模型对聚类算法进行并行化研究，结果表明，本文提出的评价方法可以较好的区分不同的供热行为，为锅炉供热系统的相关管理人员提供量化考核及其物联网系统参数调整的参考依据，同时，采用基于 MapReduce 的并行聚类算法大大提高了传统单机聚类算法的效率，对供热这种实时大规模数据有很好的适应性。在以后的研究工作将在此基础上，将对实时采集数据进行分析，为及时调整供热行为提供实时建议和参考。

6 供热效果评价实时监测系统

众所周知，数据可视化在数据挖掘的研究过程中扮演着重要的角色，将海量复杂的数据经过数据清洗，聚类分析等过程后得出结果集，通过数值型数据集无法向我们清晰的展示其规律和有趣的知识，为此我们通过数据可视化的手段将数值型数据集表达为图形化数据集，从而让结果一目了然，尤其是在实时系统中，可更快速的发现异常问题，迅速定位故障点从而解决问题，这里将搭建一个供热效果评价实时监测系统，通过实时系统中的采集数据，进行数据同步，进行实时数据分析，并通过前端页面进行展示。

6.1 系统架构概述

系统的数据来源源自于已存在的锅炉实时监测系统，通过温度传感器、压力传感器、流量计、气象仪等物联网设备将锅炉系统中实时采集到的数据存放于数据库中。而由于大量的数据是冗余了或者不在我们的分析范围内，且为了保证原系统的稳定性，我们将另外搭建一套数据库服务，这里我们将使用非关系型数据库也就是 No SQL 数据库 MongoDB，后端语言将采用 Node.js 的中间件 Mongoose 与后端数据库 MongoDB 建立连接作为数据访问层，然后将使用 Node.js 的框架 koa 方便我们进行敏捷开发，koa 被称为下一代 web 开发框架，具有洋葱模型等分层的概念，这些部分将作为我们后端的系统设计。

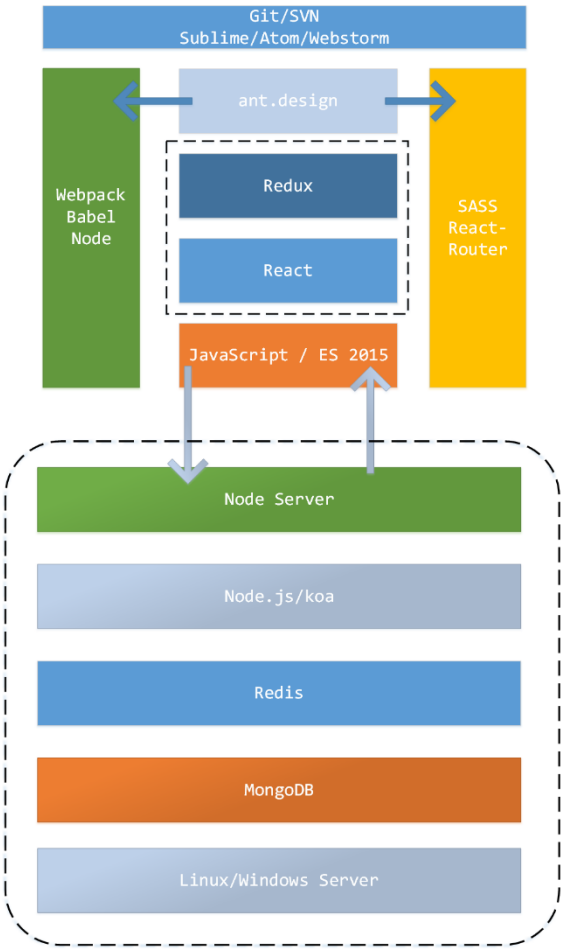


图 6-1 供热效果评价实时监测系统架构

Fig.6-1 Real - time monitoring system

由于现在的 web 开发多为前后端分离模式，且考虑到我们的供热效果评价模型需要较强的前端 UI 能力，数据渲染能力，所以本系统需要较高的前端性能。为了提高网络中的传输，使用 JSON 作为标准的通信语言格式，并规范好各个接口的格式，统一规范将大大提高开发效率和协作模式。前端的语言以 JavaScript 为基础，又将使用最新的 ECMA Script 规范作为语法，大大丰富了之前 ES5 的一些窘境。然后使用 React 作为 JavaScript 框架进行前端开发，React 具有组件化的思想，非 MVC 类的设计，并使用单向数据绑定让我们分析数据的流向；前端 UI 组件将使用阿里巴巴系蚂蚁金服开源的 ant.design 最为 UI 组件库，其中，我们的 CSS 将使用 SASS 这种 CSS 预处理器进行管理，由于 ECMA Script 2015 还未完全在各种浏览器中普及，所以我们需要使用 Babel 对代码进行转义，转义为浏览器可以解析的 ECMA Script 5 版本，保证系统

的稳定性;最后,为了自动化构建系统,方便开发,一体化设计平台的考虑使用 webpack 对前端代码进行打包，并会包括 Babel 和 SASS 编译等功能，还会为图标进行转义成 base64 格式等，还会使用压缩混淆等功能，从而全方位的提高我们的开发体验，并能保证系统上线后的稳定性及较快的访问速度；并且在对系统开发的过程中会选用 Sublime Text 3 或者 visual studio code 等编辑器进行代码书写，保证书写规范，增加开发体验，通过 Git 来对代码进行版本控制。

6.2 技术模块选用

6.2.1 MongoDB 和 Mongoose

MongoDB[76]是一个面向文档的数据库,它并不是关系型数据库,直接存取 BSON,这意味着 MongoDB 更加灵活,因为可以在文档中直接插入数组之类的复杂数据类型,并且文档的 key 和 value 不是固定的数据类型和大小,所以开发者在使用 MongoDB 时无须预定义关系型数据库中的“表”等数据库对象，设计数据库将变得非常方便，可以大大地提升开发进度。在扩展性方面，假设应用数据增长非常迅猛的话，通过不断地添加磁盘容量和内存容量往往是不现实的，而手工的分库分表又会带来非常繁重的工作量和技术复杂度。在扩展性上，MongoDB 有非常有效的，现成的解决方案。通过自带的 Mongos 集群，只需要在适当的时候继续添加 Mongo 分片，就可以实现程序段自动水平扩展和路由，一方面缓解单个节点的读写压力，另外一方面可有效地均衡磁盘容量的使用情况。整个 Mongos 集群对应用层完全透明，并可完美地做到各个 Mongos 集群组件的高可用性。这也与我们的 MapReduce 高度锲合，可以完美的搭配运行起来，同时运用于 Linux 服务器中。

Mongoose 是 Node 的一个中间件，为了更好的访问 MongoDB，同时它为我们封装了诸多常用操作，通过定义 Schema、Model 和 Entity 来对一个文档进行抽象，从而可以让开发者通过 JavaScript 操作 BSON 数据，进行传统的增删改查操作。

通过下列命令将 MongoDB 安装在本地服务中：

```
Mongod --dbpath "D:\MongoDB\data\db" --logpath "D:\MongoDB\data\log\MongoDB.log" --install --serviceName "MongoDB"
```

6.2.2 Node.js 和 Koa

Node.js[77]是一个运行在服务器端的 JavaScript，是 JavaScript 的一个运行时，Node 的出现改变了前端开发的窘境，是前端开发可以扩展到后端甚至移动端，进入大前端时代，一些公司里以 Linkendin 为代表会将整个后端架构使用 Node；而另一部分以阿

里巴巴为代表的中途岛计划，将 Node 作为他们整个架构中的视图渲染层；同时，更多的人会将 Node 作为前端开发中的工具库，从而丰富整个工程化体系，维护前端开发成本，保证代码质量。我们这里选用 Node 正因为适应这种 Node 发展的趋势，又考虑到本供热效果评价系统的业务并不是很复杂，仅仅是数据的读取与前端统计效果的展示。

Koa[78]是基于 Node.js 平台的下一代 web 开发框架，koa 由 Express 发展而来，致力于成为一个更小、更富有表现力、更健壮的 Web 框架。使用 koa 编写 web 应用，通过组合不同的 generator，可以免除重复繁琐的回调函数嵌套，并极大地提升错误处理的效率。koa 不在内核方法中绑定任何中间件，它仅仅提供了一个轻量优雅的函数库，使得编写 web 应用变得得心应手。本系统将使用 koa 的洋葱模型做中间件级联：

```
var koa = require('koa');
var app = koa();
app.use(function *(next){
var start = new Date;
yield next;
var ms = new Date - start;
this.set('X-Response-Time', ms + 'ms');
});
app.use(function *(next){
var start = new Date;
yield next;
var ms = new Date - start;
console.log('%s %s - %s', this.method, this.url, ms);
});
```

6.2.3 ECMA Script 2015

ES 2015 发布与 2015 年 6 月份，又被成为 ES 6，是 JavaScript 语言的下一代标准。它的目标是，是使得 JavaScript 语言可以用来编写复杂的大型应用程序，成为企业级开发语言。针对于 ES 5 的诸多糟粕与不足，ES 6 进行的更新，通过扩展 let/const 命令，对 var 关键词的作用域和生命提升以及全局变量都进行了详细的规范；通过引入箭头函数对之前 this 作用域问题进行的增强和限制；通过引入 import/export 关键词对 JavaScript 语言模块化的不足进行了增强，使我们不必再使用额外的 CMD/AMD 规

范下的模块加载器。本供热效果评价实时监测系统的开发，也是考虑到各大浏览器对ES 6 的规范基本支持。

6.2.4 React 和 ant.design

目前较火的前端框架有 React[79]、vue[81]、Angular[80]，其中 vue 和 Angular 都是MVVM 思想的实现，Angular 分别发布了 1.0 和 2.0 两个版本，是个大而全的解决方案，而 vue 是一个轻量级小而美的视图层解决方案，采用 mobeil first 的设计思路。而由 Facebook 发布的 React 则是另外一个划时代的 JavaScript 框架，其采用 virtual DOM 的思想，利用组件化的设计思路，将前端组件看做一个个有限状态机，当组件状态发生变化的时候执行 diff 算法，只对 DOM 变化的部分进行更新，这使得 React 在前端渲染上相较于其他框架具有得天独厚的优势，尤其在启用前端路由的基础上，页面的跳转将极为迅速。基于本供热效果评价实时监测系统并不具有复杂的业务逻辑和要求较高的视觉效果，且为了将 React 的组件化设计思想发挥到极致，我们将采用阿里巴巴旗下蚂蚁金服的产品 ant.design 作为前端 UI 的组件库，利用其组件将前端界面构建出来。

```
import React from ‘react’;
import ReactDOM from ‘react-dom’;
import 'antd/dist/antd.css';
import { DatePicker } from 'antd';
ReactDOM.render(<DatePicker />, mountNode);
```

6.2.5 SASS、Babel 以及 Webpack

由于 css 是一种图灵不完整语言，并不具备编程的元素，只是算作一门配置语言，这样导致了代码逻辑极为混乱，为了解决这个问题，本系统采用 SASS[82]这种 CSS 预处理器，来为 CSS 加入编程的元素，通过变量和函数定义相应的 mixin 区块，帮助我们规范的调整代码模块分析 CSS 代码逻辑，大大节省了我们的开发时间，使代码变得更加简单可依赖。由于之前也使用 ES 6 这种高级规范，包括 SASS 在内，在所有的浏览器并未完全支持的情况下我们只能将其算作语法糖，因此我们需要通过 Babel 进行转义，Babel 取自《圣经》中巴别塔的含义，其目的是为了让各自的语言相通，通过使用 Babel 可以讲 ES 6 和 SASS 这种高级规范和预处理语言编译为现代浏览器可以识别的原生内容。

另外，为了方便将系统各个模块的静态资源文件进行管理打包，区分模块之间的

依赖关系，同时将 Babel 集成进流程化的内容中，我们将使用 Webpack[83]这种打包工具，它将一些静态资源都视为模块，无论是 js 代码还是 css 代码，都可以通过 import 的方式进行引入，这里也更好的诠释了 React 组件化的思想，我们将供热效果实时监测系统的各个模块通过依赖关系进行管理，最终打包成为一个 bundle.js 进行全军引入，大大加快了后续的操作速度，增强了用户体验，以下为 webpack 的配置文件。

```
var path = require('path');
var ROOT_PATH = path.resolve(__dirname);
var APP_PATH = path.resolve(ROOT_PATH, 'app');
var BUILD_PATH = path.resolve(ROOT_PATH, 'build');
module.exports = {
        entry: APP_PATH,
        output: {
                path: BUILD_PATH,
                filename: 'bundle.js'
        },
       module: {
                loaders: [{
                    test: /\.jsx?$/,
                    loader: 'babel',
                    exclude: /node_modules/,
                    query: {
                            presets: ['es2015', 'react']
                    }
                }]
      },
    resolve: {
        extension: ['', '.js', '.jsx']
    }
};
```

6.3 实时监测展示

本系统主要分为三个功能。第一个功能可以对气象数据进行实时监测，可以从远端气象仪将气象数据带回，并通过 EChart 的方式将其显示，包括室外温度，风速和日照，也是影响锅炉供热出水温度的主要参考因素，通过这个页面我们可以清晰的看到

户外的天气情况，以便对锅炉控制参数及时作出调整和预测。下图为实时数据的功能展示，其将通过 AJAX 技术定时刷新数据：

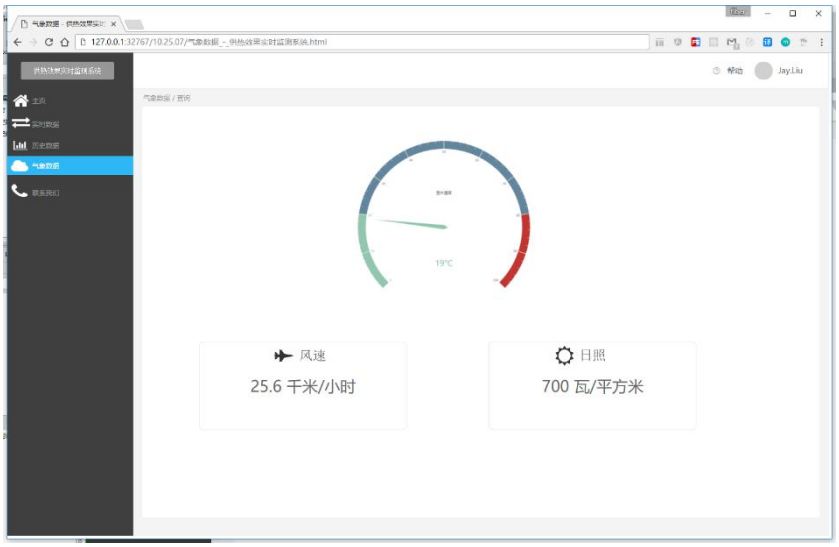


图 6-2 供热效果实时监测系统 - 气象数据

Fig. 6-2 Real - time monitoring system of heating effect - meteorological data

本系统的第二个功能为列表数据展示。依据算法进行聚类后进行列表的数据可视化，其使用表格控件，共含有八列数据，包括锅炉标号、日期、离散 Frechet 距离、横向平移、纵向平移、横向伸缩、纵向伸缩、类标号。图 6-2 为依据人员权限显示的聚类结果列表信息：

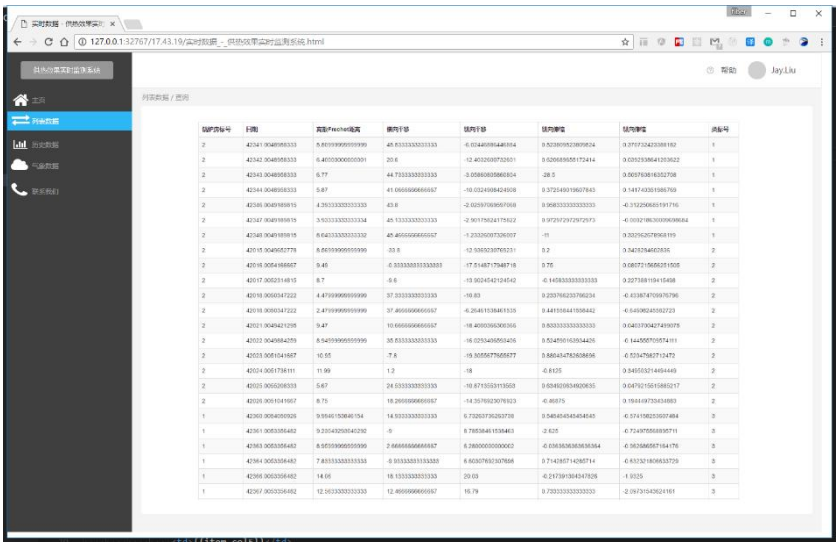


图 6-3 供热效果实时监测系统 – 列表数据

Fig. 6-3 Real - time monitoring system of heating effect - real - time data

第三个功能则是历史数据。可以根据时间段进行查询，由于底层 MongoDB 数据服务在已经在 Linux 服务器中，并且其中部分数据经过 MapReduce 进行运算的结果返回，所以在查询时并不需要进行实时运算，相当于已经具备的缓存，本设计结构参考阿里巴巴旗下阿里云的设计体系，其系统架构底层也为 Hadoop，另外在数据查询之前，搭建了一个 ODPS 系统，从而定期同步数据并进行运算，保存在数据库缓存表中，对常用的查询信息定时运算，并定时清空老旧的缓存，属于使用了最近使用原则的设计模式，因此本系统在查询时已经由 MongoDB 导入了 MapReduce 的数据，所以查询数据很快，其中会依据算法推荐出供热效果较好的锅炉房，并将异常点显示出来，从而帮助锅炉管控人员进行更加详细的查询，下图为其中一段时间的查询结果：

图 6-4 供热效果实时监测系统 - 历史数据

Fig. 6-4 Real - time monitoring system of heating effect - historical data

7 总结与展望

7.1 本文总结

数据挖掘又被称为知识发现，可以发现在海量数据中各种有趣的规律和问题，也可以应用于各种与计算机学科的交叉领域，从而完成之前未发现的问题，是一种自己非监督学习的过程。聚类分析就是数据挖掘中一种常见的分析手段，通过聚类算法自己来寻找数据中的规律，将数据分为一类一类的簇状结构。我们发现，近年来数据量的数量级暴增，已经达到 TB、PB 级别，传统聚类分析数据挖掘算法已经远远达不到我们的计算诉求，在处理海量数据时显得捉襟见肘，这些数据可能会来自于 Web 数据库，数据仓库，实时监测数据，历史归档数据等。

本文介绍了数据挖掘的常用方法，并对其中的聚类分析做了详细介绍，对基于划分，基于层次，基于密度，基于网格和基于模型的聚类算法做了研究。为了实现在聚类分析在处理大数据集上的不足，本文又研究了谷歌公司最早提出的 MapReduce 编程模型，他将并行数据通行编程层透明化，让我们只关注算法的实现，加快了我们处理大数据集的进展。

基于天津供热地区提供的供热实时监测数据，将数据挖掘的算法应用于供热领域上，分析气象条件和出水温度的关系，由于各个锅炉房自动控制系统不够智能或者管路管控人员经验不足导致供热行为的不同,这里使用聚类分析中经典的 k-means 算法，并指定 k 值为 3 和 4，通过编写 MapReduce 算法，将曲线相似度的属性相似度计算融入算法中形成改进的并行 K-means 算法，从而加强其处理大数据集的能力。

实验结果表明，MapReduce 的使用大大节省了串行算法的时间，同时对聚类簇的研究，分析出了不同锅炉房的控制能力，为其供热行为做了评价，这将方便锅炉管控人员更好的调整智能物联网设备的参数，以及对人工控制做深入的培训和精细的了解。

文章最后，又基于实验过程和结论开发了一个供热效果评价实时监测系统，通过实时系统的数据同步进行分析，利用目前最前沿的技术开发一个系统，将数据利用数据可视化的知识展现出来，帮助人们发现问题总结问题。

本文在这些工作的基础上证明了 MapReduce 并行处理的能力，通过分析的结果很好的帮助锅炉管控人员进行调整，是具有现实意义和实践能力的。

7.2 未来展望

本文分析了出水温度和气象条件的关系，然而供热锅炉系统的回水温度也对系统

起着重要作用，其反应了供热区域的面积和热量损耗度等诸多问题，接下来的工作会针对出水温度和回水温度以及气象条件进行更加耦合性的研究，探索加入回水温度后的数据会产生怎样的影响，这将更精确的为我们提供决策依据。

另外，本文分析和使用了 k-means 算法并对其做并行化处理，实际上聚类算法还有其他更多的先进算法来使用，接下来可以继续使用其他算法将其应用于供热领域来研究其关系，会不会得出更好的聚类结果。

最后，供热领域的数据具备其特殊性，与时间的气象条件有很大的关系，对时序的数据挖掘也是现在的一个热点，通过气象数据和供热数据研究其时序变化和规律也将为我们提供相应的预测机制，从而达到最大限度的节约能源帮助我们进行及时的供热锅炉系统参数调整，是预测更加及时，而让供热用户充分的享受热量而又使供热公司不再浪费资源达到节能减排的规划。

8 参考文献

[1] Hartigan J A. Clustering algorithms[M]. Wiley, 1975.

[2] Macqueen J. Some Methods for Classification and Analysis of MultiVariate Observations[C]// Proc. of, Berkeley Symposium on Mathematical Statistics and Probability. 1967:281-297.

[3] Kaufman L, Rousseeuw P J. Finding groups in data. an introduction to cluster analysis[J]. Wiley, 1990.

[4] Huang Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values[J]. Data Mining and Knowledge Discovery, 1998, 2(3):283-304.

[5] Lauritzen S L. The EM algorithm for graphical association models with missing data[J]. Computational Statistics & Data Analysis, 1995, 19(2):191-201.

[6] Tung A K H, Hou J, Han J. COE: Clustering with Obstacles Entities A Preliminary Study[M]// Knowledge Discovery and Data Mining. Current Issues and New Applications. Springer Berlin Heidelberg, 2000:165-168.

[7] Ester M, Kriegel H P, Xu X. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification[M]// Advances in Spatial Databases. Springer Berlin Heidelberg, 1995:67-82.

[8] Ordonez C, Omiecinski E. FREM: fast and robust EM clustering for large data sets[C]// Eleventh International Conference on Information and Knowledge Management. ACM, 2002:590--599.

[9] Zhang T. BIRCH: an efficient data clustering method for very large databases[J]. Acm Sigmod Record, 1996, 25(2):103-114.

[10] Guha S, Rastogi R, Shim K. CURE : An Efficient Clustering Algorithm for Large Databases[J]. Information Systems, 1998, 26(1):35-58.

[11] Guha S, Rastogi R, Shim K. ROCK: A Robust Clustering Algorithm for Categorical Attributes[J]. Information Systems, 2000, 25(5):345-366.

[12] 金阳，左万利．一种基于动态近邻选择模型的聚类算法[J]．计算机学报，2007, 30(5):756-762.

[13] Wang W, Yang J, Muntz R R. STING: A Statistical Information Grid Approach to Spatial

Data Mining[C]// International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 1997:186-195.

[14] Sheikholeslami G, Chatterjee S, Zhang A. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases[C]// Proceedings of the 24rd International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 1998:428--439.

[15] Yu D, Chatterjee S, Zhang A. Efficiently detecting arbitrary shaped clusters in image databases[J]. 1999:187-194.

[16] Agrawal R, Gehrke J E, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications:, US 6003029 A[P]. 1999.

[17] 陈宁, 陈安, 周龙骧. 基于密度的增量式网格聚类算法(英文)[J]. 软件学报, 2002, 13(1):1-7.

[18] Schikuta E，Erhart M.BANG-clustering：a novel grid-clustering algorithm for huge data sets[C]//LNCS 1451，1998.

[19] Ester M, Kriegel H P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[C]// 2008:226--231.

[20] Zhou S, Zhou A, Jin W, et al. FDBSCAN: A fast DBSCAN algorithm[J]. Journal of Software, 2000.

[21] Ankerst M. OPTICS: ordering points to identify the clustering structure[J]. Acm Sigmod Record, 1999, 28(2):49-60.

[22] 赵艳厂, 谢帆, 宋俊德. 一种新的聚类算法:等密度线算法[J]. 北京邮电大学学报, 2002, 25(2):8-13.

[23] Hinneburg A, Keim D A. An Efficient Approach to Clustering in Large Multimedia Databases with Noise[J]. KDD, 1999, 98.

[24] 裴继法, 谢维信. 聚类的密度函数方法[J]. 西安电子科技大学学报, 1997(4):463-467.

[25] Qiu X, Tang Y, Meng D, et al. A new fuzzy clustering method based on distance and density[J]. 2002, 7:282-286.

[26] 曾东海, MI Hong, 刘力丰. 一种基于网格密度与空间划分树的聚类算法[J]. 系统工程理论与实践, 2008, 28(7):125-131.

[27]李光强, 邓敏, 刘启亮,等. 一种适应局部密度变化的空间聚类方法[J]. 测绘学报, 2009, 38(3):255-263.

[28]倪巍伟, 陈耿, 吴英杰,等. 一种基于局部密度的分布式聚类挖掘算法[J]. 软件学报, 2008, 19(9):2339-2348.

[29]Fisher D.Improving inherence through conceptual clustering[C]//Proc 1987 AAAI Conf，1987：461-465.

[30]Gennari J H, Langley P, Fisher D. Models of incremental concept formation[J]. Artificial Intelligence, 1989, 40(1-3):11-61.

[31]Cheeseman P， Stutz J.Bayesian classfication （ Auto Class ）： theory and result[M]//Advances in Knowledge Discovery and Data Mining.[S.l.]：AAAI Press/MIT Press，1996：153-180.

[32]Clara P，Domenico T.P-Auto Class：scalable parallel clustering for mining large data sets[J].IEEE Trans on Knowledge and Data Engineering，2003，15（3）：629-641.

[33]Rumelhart D E， Zipser D.Feature discovery by competitive learning[J].Cognitive Science，1985，9（1）：75-112.

[34]Kohonen T.Self-organization and associate memory[M].Berlin：Springer-Verlag，1984.

[35]Kohonen T.Improved versions of learning vector quantization[C]//International Joint Conference on Networks，San Diego，1990：545-550.

[36]Teuvo K.The self-organizing map[J].Neurocomputing，1998，21（13）：1-6.

[37]刘铭，王晓龙，刘远超.一种大规模高维数据快速聚类算法[J].自动化学报，2009，35（7）：859-866.

[38]江小平, 李成华, 向文,等. k-means 聚类算法的 MapReduce 并行化实现[J]. 华中科技大学学报:自然科学版, 2011, 39(s1):120-124.

[39]周婷, 张君瑛, 罗成. 基于 Hadoop 的 K-means 聚类算法的实现[J]. 计算机技术与发展, 2013, 23(7):18-21.

[40]赵庆. 基于 Hadoop 平台下的 Canopy-Kmeans 高效算法[J]. 电子科技, 2014, 27(2):29-31.

[41]贾瑞玉, 管玉勇, 李亚龙. 基于 MapReduce 模型的并行遗传 k-means 聚类算法[J]. 计算机工程与设计, 2014, 35(2):657-660.

[42]李兰英, 董义明, 孔银,等. 改进 K-means 算法的 MapReduce 并行化研究[J]. 哈尔

滨理工大学学报, 2016, 21(1):31-35.

[43]张磊, 张公让, 张金广. 一种网格化聚类算法的 MapReduce 并行化研究[J]. 计算机技术与发展, 2013(2):60-64.

[44]Zhang B T, Ramakrishnan R, Livay M. Brich: an efficient data clustering method for very large databases[C]// Procedings of ACM Sigmod, ACM. 2010.

[45]Guha S, Rastogi R, Shim K. ROCK: A Robust Clustering Algorithm for Categorical Attributes[J]. Information Systems, 2000, 25(5):345-366.

[46]Karypis G, Han E H, Kumar V. CHAMELEON A hierarchical clustering algorithm using dynamic modeling[J]. Computer, 1999, 32(8):68-75.

[47]Macqueen J. Some Methods for Classification and Analysis of MultiVariate Observations[C]// Proc. of, Berkeley Symposium on Mathematical Statistics and Probability. 1967:281-297.

[48]Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding[C]// Eighteenth Acm-Siam Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, Usa, January. 2007:1027-1035.

[49]Chiang M T, Mirkin B. Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads[J]. Journal of Classification, 2010, 27(1):3-40.

[50]Krishna K, Narasimha M M. Genetic K-means algorithm.[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 1999, 29(3):433-9.

[51]Kaufmann L, Rousseeuw P J. Clustering by Means of Medoids[C]// Statistical Data Analysis Based on the L1-norm & Related Methods. North-Holland, 1987:405-416.

[52]潘吴斌. 基于云计算的并行 K-means 气象数据挖掘研究与应用[D]. 南京信息工程大学, 2013.

[53]张睿欣. 一种聚类算法的并行化改进及其在微博用户聚类中的应用[D]. 上海交通大学, 2014.

[54]by Jain A K. Dubes RC: Algorithms for Clustering Data[J]. 2010.

[55]Chaturvedi A, Green P E, Caroll J D. K-modes Clustering[J]. Journal of Classification, 2001, 18(1):35-55.

[56]Dhillon I S, Guan Y, Kulis B. Kernel k-means: spectral clustering and normalized cuts[C]// Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2004:551--556.

[57]Ester M, Kriegel H P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[C]// 2008:226--231.

[58]Ankerst M, Breunig M M, Kriegel H P, et al. OPTICS: ordering points to identify the clustering structure[J]. Acm Sigmod Record, 1999, 28(2):49-60.

[59]Wang W, Yang J, Muntz R R. STING: A Statistical Information Grid Approach to Spatial Data Mining[C]// International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 1997:186-195.

[60]张珊. 供热锅炉绩效评价及优化系统的研究[D]. 大连海事大学, 2013.

[61]路昌海, 刘贵松, 张明琤. 基于支持向量回归的锅炉出水温度时间序列预测[J]. 区域供热, 2014(6):18-22.

[62]岳孝忠. 基于锅炉运行优化的数据挖掘平台研究与实现[D]. 电子科技大学, 2012.

[63]江亿, 彭琛, 胡姗. 中国建筑能耗的分类[J]. 建设科技, 2015(14):22-26.

[64]高兴.基于特征信息的测井曲线相似度算法研究与应用[D]. 东北石油大学, 2013.

[65]张宇, 刘雨东, 计钊. 向量相似度测度方法[J]. 声学技术, 2009, 28(4):532-536.

[66]郑丽萍, 李光耀, 梁永全,等. 本体中概念相似度的计算[J]. 计算机工程与应用, 2006, 42(30):25-27.

[67]HELMUT ALT, MICHAEL GODAU. COMPUTING THE FRÉCHET DISTANCE BETWEEN TWO POLYGONAL CURVES[J]. International Journal of Computational Geometry & Applications, 2011, 5(1):75-91.

[68]Mosig A, Clausen M. Approximately matching polygonal curves with respect to the Fréchet distance[J]. Computational Geometry, 2005, 30(2): 113-127.

[69]曹凯, 唐进君, 刘汝成. 基于 Fréchet 距离准则的智能地图匹配算法[J]. 计算机工程与应用, 2007, 43(28):223-226.

[70]Eiter T, Mannila H. Computing discrete Fréchet distance. See Also[J]. See Also, 1994, 64(3):636-637.

[71]朱洁, 黄樟灿, 彭晓琳. 基于离散 Fréchet 距离的判别曲线相似性的算法[J]. 武汉大学学报:理学版, 2009, 55(2):227-232.

[72]蔡启林, 寿晓峰. 供暖热负荷延时曲线及其应用[J]. 区域供热, 1991(2):1-10.

[73]李光, 吴祈宗. 基于结论一致的综合评价数据标准化研究[J]. 数学的实践与认识, 2011, 41(3):72-77.

[74]王福昌, 曹慧荣, 朱红霞. 经典最小二乘与全最小二乘法及其参数估计[J]. 统计与决策, 2009(1):16-17.

[75]周世兵, 徐振源, 唐旭清. 新的 K-均值算法最佳聚类数确定方法[J]. 计算机工程与应用, 2010, 46(16):27-31.

[76]吕林. 基于 MongoDB 的应用平台的研究与实现[D]. 北京邮电大学, 2015.

[77]Zammetti F. Introducing Node.js[M]// Pro iOS and Android Apps for Business. Apress, 2013:119-141.

[78]Roden G. Neues Framework für Node.js: Koa[J]. Heise Zeitschriften Verlag, 2013.

[79]Archer R. ReactJS: For Web App Development[M]. CreateSpace Independent Publishing Platform, 2015.

[80]Ford B. Angular JS in Action[J]. Pearson Schweiz Ag, 2015.

[81]Mengesonnentag R. Die JavaScript-Bibliothek Vue.js erreicht Release-Status[J].

[82]Libby A. Instant SASS CSS how-to[J]. 2013.

[83]Speaker-Boissi, Re A. Efficient Static Assets Pipeline with Webpack[C]// Applicative. ACM, 2015.

9 攻读学位期间发表的学术论文和所做的项目

论文：

[1] 孙志伟，冯海波，马永军，王福全，董亮亮．基于曲线相似度的供热行为评价[J]．天津科技大学学报，2016．

项目：

天津港焦炭煤码头公司办公系统整合项目

天津市公路管理局后台管理系统

天津鸿觉能源燃气管网 GIS 项目

天津市供热锅炉管理项目

天津物流跟踪 GIS 项目

10 致谢

本文是在孙志伟副教授的悉心指导下完成的，同时也得益于课题组组长马永军教授的方向引导和硬件支持。非常感谢孙老师和马老师在学术上的谆谆教诲，孙老师和马老师严谨的教学姿态，诲人不倦的精神，因材施教的育人之道，深深的影响着我，也为我以后在职业生涯上的发展奠定了基调。孙老师在数据挖掘领域有很深的见解，同时孙老师善于将数据挖掘的知识应用于实践，通过横向项目进行结合，充分发挥了计算机领域数据挖掘方向多学科交叉研究的特色，非常感谢孙老师对我的悉心指导。

孙老师不仅在纵向科研学术领域对我有深入的帮助，同时带我进入了横向项目开发的领域。研究生期间跟随马老师和畅老师做了诸多项目，自己也从本科路由与交换的方向转移到了前端开发的方向，在技术领域有了深入的实际实践体会，帮助我在实际项目中锻炼动手能力，项目思维。也经过老师们的教导和自己的深入学习，在毕业之际拿到了阿里、美团等多个 offer，能让我加入到了阿里巴巴这样著名的互联网公司，这些都离不开老师给予的机会和教导。

同时感谢自己的学校，天津科技大学。我在本校从本科开始一共学习了六年半的时间，自己的美好青春都在这里尽情的挥洒，遇到了很多可爱的有趣的人，有耐心指导我的学长，帮助我的学姐，有热爱学习让我倍感压力学弟学妹，这里的每个人都是我的榜样，和他们在一起合作很快乐。感谢畅卫功老师在本科和研究生阶段的开发指导，感谢张传雷老师在学术和项目的悉心指导，感谢张强老师，陈少杰老师的帮助。

研究生阶段是我人生转折的一个过渡阶段，研究生同学们在各自知识领域深入的造诣深深的影响了我，我们在一起探讨学术，一起做项目，让自己的日常生活特别充实，每天都过得非常有意义。感谢单渊博和我一起在数据挖掘领域上的讨论，感谢陶鑫、谢迎、荚佳舍友们在生活上的交融和帮助，模式识别和大数据团队的成员万莉、刘洋、薛永浩在学术上的探讨，也非常感谢天津科技大学河西校区 309 实验室的小伙伴和天津科技大学泰达校区 301 的小伙伴。

最后感谢我的父母和家人。是你们在背后的无私支持让我有了今天的进步和成就，让我在离家千里的地方安逸的学习，认真的完善自己的职业生涯。忠厚诚信的家风也深深的影响着我的生活，让我在为人处世方面如鱼得水。感谢我的父亲冯顺昌，我的母亲戴莲芹，我的姐姐冯艳秋和姐夫张俊杨，是你们的支持才让我的研究生生涯如此的顺利，充实和辉煌。