同济大学
TONGJI UNIVERSITY

硕士学位论文

基于数据挖掘的住宅建筑能耗预测模型研究

姓　　名：沈 寿 鹏

学　　号：1531650

所在院系：电子与信息工程学院

学科门类：工 学

学　　科：控制科学与工程

指导教师：肖 辉 教授

二〇一八年三月

同濟大學
TONGJI UNIVERSITY

A dissertation submitted to

Tongji University in conformity with the requirements for

the degree of Master of Science in Engineering

Research on Energy Consumption Prediction Model of Residential Building Based on Data Mining

Candidate: Shen Shoupeng

Student Number: 1531650

School/Department: College of Electronics and Information Engineering

Discipline: Engineering

Major: Control Science and Engineering

Supervisor: Prof. Xiao Hui

March, 2018

基于数据挖掘的住宅建筑能耗预测模型研究

沈寿鹏

同济大学

学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年　　月　　日

同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

摘要

近年来，随着计算机信息网络技术的迅猛发展，各行各业存储了大量的数据信息，日益精进的数据挖掘技术已经能从海量数据中发现隐藏价值并支持人类的决策。

对建筑领域实现数据合理化，评价科学化、预测信息准确化对构建低碳建筑有重要的作用。同时，以住宅建筑能耗特征为背景,采用数据挖掘技术预测住宅建筑电力能耗和能耗等级。系统以 SVM 为基本优化框架，克服了能耗数据的稀疏高维度特性，提出适用于住宅建筑能耗数据的高效分类、回归算法，并且设计实验验证了算法的可行性。本文为住宅建筑划分住宅建筑能耗等级，辅助电力部门掌握居民的用电行为动态，合理分配电力能源；此课题为研究住宅建筑能耗行为提供了技术方法，具有重要的实践价值。

本文的主要工作和创新点如下：

（1）针对住宅数据处理过程中的高维度问题，分别引入主成分分析（Principle Component Analysis, PCA）、奇异值分解（Singular Value Decomposition, SVD）、随机森林（Random Forest，RF）等经典特征工程算法，通过仿真分析比较各算法的性能。考虑到不同的特征工程算法适用不同数据集，通过优化 SVM 核函数来解决数据的稀疏性，提出了基于 PCA、SVD 和 RF 估计结合的 PSR-SVM 算法。

（2）针对聚类过程中存在的初始中心点随机化选择问题，本文结合 PSO 优化算法提出 PSO-Kmeans 算法。实验中以 UCI 数据为算法比较测试集，设计实验比较优化算法与传统 Kmeans 算法在不同数据集上的性能差异，实验引入聚类评价参数评价聚类效果并且验证了优化聚类算法在聚类场景中有较好的实验性能。

（3）针对实际场景中的住宅建筑能耗数据预测问题，以美国佛罗里达州的住宅信息、天气、电力和天然气等信息组成的数据集为研究样例。本文采用优化聚类算法针对住宅建筑能耗按季度划分能耗提取聚类中心并且划分能耗等级。采用提出的组合特征工程算法解决住宅建筑能耗数据中的高维数据特性，提升实验效果，并且比较神经网络、逻辑斯特、GBDT 在住宅能耗数据分类、回归预测问题场景上的表现效果。针对能耗数据等级分类中存在类不平衡现象，本文引入采样算法解决住宅建筑能耗等级分布不平衡问题。

最后对全文工作进行了总结，本文使用的数据挖掘算法框架能满足住宅建筑能耗预测准确性的要求，并给出下一步工作展望。

关键词：数据挖掘，住宅建筑，稀疏性，不平衡类，聚类

ABSTRACT

In recent years, with the rapid development of computer information network technology, all walks of life have storage a large amount of data and information. Increasingly sophisticated intelligent data analysis technology has been able to find hidden value from massive data and support human's decision.

Rationalization of data in the field of architecyure ,scientific rating,and accurate prediction of information play an important role in building low carbon society. At the same time, based on the characteristics of energy consumption of residential buildings, data mining technology is used to predict the power consumption and energy consumption levels of residential buildings. The system uses SVM as the basic optimization framework, which overcomes the sparse high-dimensional characteristics of energy consumption data, proposes an efficient classification and regression algorithm for energy consumption data of residential buildings, and design experiments verify the feasibility of the algorithm. This article divides the residential building energy consumption level for the residential building, assists the relevant department to grasp the dynamic behavior of the residents' electricity consumption, and reasonably distributes the electric energy; this topic provides a technical method for studying the energy consumption behavior of the residential building and has important practical value.

The main work and innovation of this paper are as follows:

(1)Aiming at the problem of high dimension and sparsity in residential data, this paper lead some classic dimensianality reduction algorithem.For example, Principle Component Analysis(PCA), Singular Value Decomposition(SVD) and Random Forest(RF) and so on,into SVM .which compare the performance of each algorithm through design variable experiments. Considering that different feature reduction algorithms apply different data sets and solve the sparsity of data by optimizing SVM kernel function. Verifing the conclusion through experiments, propose P+S+RF-SVM based on PCA, SVD and RF.

(2)Aiming at the randomization of the initial center point in the clustering process,this paper propose PSO-Kmeans algorithem which base on the PSO optimization algorithm. Taking the UCI data as the experimental test set, the

experimental design was used to analyze the performance differences between the optimized algorithm and the traditional Kmeans algorithm on different datasets. The clustering evaluation parameters were introduced to evaluate the clustering effect and the optimized algorithm was proved to be good in the clustering scenario Experimental performance

(3) For the actual scenario of energy consumption data, the article deal with the data which is consist of residential information,weather,electricity and natural gas from Florida,USA. In this paper, the optimal clustering algorithm for residential building energy consumption are divided by quarter. The combined feature is used to solve the high-dimensional data characteristics of residential building energy consumption data to improve the experimental results, and to compare the performance of neural network, logistic and GBDT in residential energy consumption data classification and regression forecasting scenarios. In order to solve the imbalance of energy consumption in data classification, this paper introduces a sampling algorithm to solve the imbalance of energy consumption in residential buildings. The experiment's result shows that the introduction of sampling algorithm significantly improves the prediction accuracy of the model and validates the validity of the model.

Finally, the paper summarizes the work, and gives the next work outlook.

Key Words: Data mining, Residential building, Sparsity,Imbalanced class, Cluster

目录

插图索引

表格索引

第1章 绪论

1.1 研究背景

随着信息化社会的不断推荐，各行业都储存了海量的、多样化数据。如电商网站的顾客浏览记录；超市顾客的交易记录和商品交易记录；医院病人的临床医疗信息。城市道路交通监管部门的监控记录；学校学生日常消费记录。各行各业的工作人员都尝试着从海量数据中挖掘出数据背后隐藏的价值以支持各个行业的商业行为。例如电商会精准投放顾客所感兴趣的商品信息，提高效益。超市会根据商品的交易信息调整物品摆放位置提高交易额。医院收集大量的临床信息，降低早产儿的夭折率。交通部门会根据历史流量信息，竞逐预测未来的车流量，及时实行交通管控，减少城市的拥堵情况，提高城市交通的管理能力，提高人民的幸福感。

在信息化社会里，建筑业也在探索新的管理方式达到节能减排的。运用计算机技术将建筑信息和建筑能耗等信息进行深度整合。实现信息共享、业务联动和服务整合。2011 年 5 月，住房和城乡部联合财政部下发《关于进一步推进公共建筑节能工作的通知》,要求全国范围内对各大国家机关建筑和大型公共建筑都需要进行能耗统计的相关工作，对城市中的重点建筑实行分项计量和实时动态监测，并指定能耗标准限制能耗行为，加强公共建筑物技能管理。因此如今的建筑行业已经有很多专家学者提出了针对办公建筑的节能控制策略。

针对建筑能耗数据的分析研究一直广受各专家学者的关注。专家学者一直致力于利用分析技术研究建筑的能耗规律，做到实时动态掌握建筑能耗规律。而住宅建筑又是建筑体系中一个不可缺少的部分[1]。加强对大型住宅建筑能耗管理以及对动态实时地监测住宅建筑的能耗，此种措施也能很好的落实国家关于建筑节能的节能政策。如今的监控平台主要是实时显示各个时间段建筑物的能耗情况，并没有深入挖掘能耗信息中隐藏的信息，更无法预测未来时间内建筑物的能耗趋势，进而提供对能耗的控制策略。随着现代化的技术手段不断提高，对现有建筑能耗数据进行分析，挖掘其潜在价值，不仅对住宅建筑的能耗行为的引导有指导意义，而且能对掌握建筑的能源动态行为和对整个社会的节能减排工作有着举足轻重的作用。

1.2 研究目的及意义

本课题研究的目的是采用数据挖掘先进的技术和强大的数据分析能力，对住宅建筑能耗数据进行深层的分析、研究和预测。本课题的研究过程中比较了不同算法在住宅建筑能耗预测场景中的适用性。相比较之前的所用的单一的统计学方法，采用数据挖掘的方式分析能耗数据，可靠性强和有更强的泛化性。

首先，采用回归算法对住宅建筑能耗值精确预测，此方法的提出能为政府职能部门提供的节能减排政策提供技术支持。另一方面，对住宅建筑群划分电力能耗等级，并且采用算法达到准确预测建筑能耗等级。为了能达到上述两个方面的目的。本文采用的是美国佛罗里达州公开的建筑数据集，此数据集具有覆盖范围广、数据特征维度高等特点（数据及的具体来源在下文具体指出）并且能后支持数据挖掘工作的开展。

建筑物在运行过程中需要消耗一定的能源消耗，人们对室内热环境的需求基本上是恒定的，因此，起先认为建筑节能的根本途径就在于设计建造低能能耗的建筑物。但是随着社会发展和建筑节能技术的提高，我国新建建筑得到单位面积耗热量已经达到了建筑节能设计标准要求。所以只能从住宅用户层面解决住宅建筑能耗问题。而由于针对住宅建筑研究方法的缺失，因此急需一种具有数据支持的技术手段为政府职能部门政策的制定提供支持，如电价按能耗等级收费。

由于长期以来住宅建筑能耗只能宏观统计，建筑节能的提升与发展路径等方面的研究非常有限，且存在诸多不足，致使国家层面的相关政策和规划缺少依据，国家总量约束节能的科学拆解与落地实施也缺乏支持，提供一种有效的技术手段也会对整个社会的建筑节能领域的节能工作的开展显得必不可少。本课题的研究是基于数据挖掘技术研究具有高维特征属性和特征信息多等特点的住宅建筑能耗数据。本课题的算法框架能够精确预测住宅建筑未来的电力能耗掌握用户的用电趋势；政府职能部门可以根据算法模型预测结果制订相关法律法规，干预用户的用电行为，将节能减排落实到实处，如对电价按能耗等级收费。

1.3 数据挖掘分析技术以及建筑节能领域研究综述

1.3.1 数据挖掘技术发展研究现状

在信息爆炸的时代，最典型的时代特点就是各种各样的数据呈现海量性。但是研究然预案并不能将直接将海量的数据应用到现实场景中。比如常用的数据仓

库中存在着很多未被加工的结构化、非结构化的原始数据，但是这些数据又对社会的建筑有着潜在价值。统计学的方式已经很难应对这些挑战了。因此，基于计算机技术的分析方法开始受到各行各业研究者的关注。特别是智能数据分析技术[2-3]中的数据挖掘技术[4-5]研究最近几年呈现火爆的局面。

金融、建筑能耗分析、市场开发、现代化教、智慧城市的建设[6]和医疗诊断的等众多领域中的数据挖掘技术应用已经开展的如火如荼。在商业领域中挖掘某些商品的关联属性，为决策合理的安排商品摆布提高企业的竞争力有着很大的作用。数据挖掘技术是在一堆看似不相关的特征中发现他们内在联系。数据提取主要围绕着数据分布特点，如对相关规则进行总结、关联、分类和聚类。同时按照这些规则对数据预测、趋势分析。这些经常运用统计学、数据库以及人工智能方法。如基于距离算法（K 均值算法、支持向量机）、基于概率的算法（朴素贝叶斯、决策树算法）以及人工智算法（人工神经网络、遗传算法），以上所列举的算法都已经在工业场景中得到广泛应用并且其良好的实验性能等到了验证。

决策树是预测各种情况发生的概率，是一种基于概率分布表示方法[7]。常见的决策树的算法有 ID3、C4.5、CART 等。在特征选择过程中，研究者大多使用随机森林。随机森林[8]是一种基于随机化选择特征的统计学习理论的组合分类器。它将 Bootsrap 重抽样方法和决策树算法相结合，计算各个特征与结果值的信息增益，挑选对结果影响较大的特征向量。目前很多学者和研究人员已经大随机森林模型应用到实际场景中并且取得了很大量的研究成果。。例如马玥[9]等人使用随机算法用于土地分类研究并且取得了良好效果。M Kohansal 等人运用随机森林预测运营商电力价格，找出了历史价格、时间以及新的辅助功能对目标电价之间的关系[10]。

人工神经网络是模仿人类大脑和功能而构建的网络，人工神经网络中有多个结构简单的处理单元按照某种连接方式而成的信息处理系统。神经网络中的每一个节点类似于人神经中枢中的一个神经元。因此人工神经网络凭借具有分析存储学习、大规模并行处理、较高的鲁棒性和容错能力以及联想记忆等诸多优点[11]成为数据分析领域中最强大工具之一。Wu J[12]等人对 IGBT 功率模块节点温度预测建立模型，采用 BP 神经网络模型的结果优于简单的多项式拟合算法。李晶[13]等人利用神经网络和 SVR 建立了大气溶光学厚度预测模型。实验对比结果表明，SVM 和神经网络均具有较强的非线性拟合能力，易于实现，并且 BP 神经网络模型比 SVM 模型训练时间更短、更准确。

针对实际应用场景，很多学者将不同的算法进行整合，实现算法间的取长补短并且去得了不错的效果例如山东大学的曹贵宝[14]等人融合随机森林算法和卷积神经网络分割神经细胞显微图像并且取得了良好的效果；谢肇庆[15]等人使用BP

神经网络和K均值聚类算法反映起重机与起重机起升机构中电机（起升电机和开闭电机）瞬时输出功率的关系，并且实际数据证明了该方法的有效性；Burton S H等[16]用聚类算法对所研究问题做划分，将相似度高的 问题归为一类，以减少目标问题个数，从而减少搜索空间，提高关联规则算法产生候选项集的效率;

1.3.2 数据挖掘技术在建筑节能领域的应用

传统的建筑能耗分析方法是通过构建模型模拟建筑物的能耗水平，所有的计算方法都是在一个理想的设定条件下进行，实验数据并不能真实的反映建筑物的能耗情况和用户的能耗行为具有一定的片面性。

随着现代技术的飞速发展，海量数据背后隐藏这丰富的信息，数据集具有数据量大、特征维度高等特点，这些结构化和非结构化数据中包含着噪声、数据缺失等不确定性因素。常规的分析方法难以发现和总结这些数据中隐藏的价值。Star.Ammar Ahmess[17]等人通过数据挖掘技术建立了一种能耗预测模型和房间可用日光模型，对样本数据应用贝叶斯分类算法、决策树方法和支持向量机方法，然后比较各个分类方法在能耗预测问题上的准确性和可靠性。Zhijian Hou[18]等人在使用多个参数对传感器操作过程中，提出了一种结合粗糙集和人工神经网络方法也取得了不错的预测效果。Yang Gao[19] 等人将 C4.5 分类方法应用于确定合适的约束条件和外部条件在何种条件下对用户的舒适度影响最大。

住宅建筑能耗研究是建筑节能领域中很重要的分支，很多专家学者已经展开了广泛的研究。Alberto Hemandez Neto 和 Flavio Augusto Sanzovo Fiorelli 在建筑能耗数据集上，比较人工神经网络和 EnergyPlus 预测软件的预测模拟结果，表明两种方法都适用于建筑能耗的预测场景[20]。Regis Signor 等学者对巴西利亚、纳塔尔和福塔雷萨等 14 个城市进行办公建筑能后模拟比对，对能耗数据进行多元回归，由此构建办公建筑电力能耗的预测方法，该方法被广泛应用于巴西诸多城市的电耗预测[21]。S,Karatasou 和 M.Santamouris [22]对实际工程案例中应用改进的人工神经网络方法构建能耗预测模型，验证了神经网络模型具有很强的可靠性和精度。陈文凭和杨昌志[23]针对建筑类别以及特征，对 BP 神经网路进行改进和创新，并利用其变体对商业建筑负荷进行短期预测，其将结果与 DeST 模拟软件相比较，验证此模型的可行性。雷娅蓉[24]等基于灰度神经网络系统模型构建民用建筑的生活能耗预测系统并且取得了很好的预测结果。华南理工大学的蒋毅[25]利用数据库存储了大量的结构化数据，以这些数据为基础，利用 MATLAB 自带工具箱进行能耗预测，发现机遇面积的的能耗电力预测和以建筑物能耗指标进行精确预测存在着一定的差异。

目前建筑能耗分析方法主要集中于分类方法中线性回归算法和神经网络算法，主要是实现基于建筑能耗数据实现对未来某个时间点的预测功能，而针对建筑物的能耗评级并且提出合理管理方式是最近刚刚兴起的研究方向，近年来越来越多的学者开始转移到通过聚类分析技术和关联规则分析手段监测数据中存在的异常点或者在大量数据中发现其中的分布规律。通过关联规则分析手段分析结果和各个维度之间的关系，从中挑选影响因此最大的特征，为后续管理者指定节能措施提供更好的节能方案，达到更好的节能目的。

1.4 数据挖掘中的稀疏不平衡问题

机器学习、数据挖掘过程中，数据工程师通常假设训练数据样本中的各个类别是分布均衡的并且样本数量大致相当。但是在工程问题或者学术问题中，在数据集样本中，某些类别样本数量远远多于其他几个或者个别类别的情况，例如垃圾邮件过滤、卫星雷达照片对海洋表面石油油污监测、银行系统信贷检测以及医疗系统中的疾病监测、网络入侵等问题。数据集中某种情况出现比较少的的数据集被称为不平衡数据集。

数据集中的数据不平衡问题是有监督学习中常见的问题，因为在少数类实例中通常包含着重要信息，此时错分一个实例都将付出很大的代价，为了提高模型算法的泛化能力。在使用不平衡数据集训练时，由于分类器会注重准确率，对多数类会存在过拟合情况，因而就对少数类的实例存在欠拟合的现象。因此在实际预测时，少数类别的预测准确性才决定了模型算法的泛化能力。在学术界，不平衡分类问题作为十大数据挖掘难题在被科研工作者研究。

目前对于机器学习问题中类别不平衡问题的改进主要分为两个方面，分别是基于数据层面和基于算法分类器层面讲不平衡数据集转化成平衡数据集。数据层面主要是通过增加少数类和减少多数类来降低样本数的不平衡程度。数据层面的方法主要的是对多数类进行欠采样和少数类进行过采样两类方法。根据采样比率设置降低原始各个类别样本数量的不平衡程度。提升模型准确性。模型算法层面主要是针对各个分类器的损失函数、核函数进行改进。改进型重采样方法主要是欠采样为主，例如高效的算法有 SMOTE[26]、EasyEnsemble[27]等采样算法。基于算法层面的改进涉及面则广泛的多，例如对 SVM 算法的惩罚参数重选或者对损失函数（Loss function）重定义[28]，或者是利用集成类算法（Bagging，Boosting）进行投票表决。极限学习机（Extreme learning machine， ELM）是最近几年发展起来的处理类别不平衡问题的算法框架[29-30]，特别是在稀疏在线不平衡类问题中，有着良好的表现。2010 年，Batuwita[31]等人提出 FSVM-CIL 算法在处理类别不平

衡问题上有着更进一步的改进，实验证明其和 C4.5 等传统分类器相比有着更高的精确度。以上这些算法的目的都是为了提高分类准确性，评价标准上反映出算法或者数据层面的改进效果，例如准确率(Precision)、查全率(Recall)、F-measure、G-mean 等等常规的评价标准。

1.5 聚类中心随机初始化问题

聚类分析技术是数据挖掘中最常用的无监督学习手段，该算法能找出数据的分布规律，便于工作者针对数据的分布规律进行分析。聚类分析技术属于典型的无监督学习，算法的实现不存在任何标签导致最终的分簇结果。目前存在很多种聚类分析方法如：划分法、层次法、基于模型的方法。K 均值算法（Kmeans 算法[32]）是基于划分法的经典聚类算法，此算法具有容易复现、收敛速度快和应用范围广等诸多特点。但是另一方面，聚类算法也存在着对聚类初始中心点敏感、聚类中心数无法确定和在高维度数据中迭代次数较多等诸多不足之处。

粒子群算法(Particle Swarm Optimization PSO)[33]是一种基于群体智能的随机启发式搜索算法，该算法已经广泛应用于非线性、不可微和多峰值的复杂问题的解决方案中，主要是利用了该算法具有高效并行和很强的全局搜索能力的特点。此优化算法具有程序简单，调节参数少等诸多优点，因此得到了很快速的发张。此算法已经广泛应用于多个学科和工程领域。但是优化算法也存在着算法粒子容易早熟、全局收敛性较差的问题。很多专家学者在 kmeans 算法中引入 PSO 算法提高聚类效果，缓解了 Kmean 算法对聚类中心初始点的依赖问题[34-35]。

1.6 主要内容及结构安排

1.6.1 主要内容

本文旨在研究住宅建筑能耗预测模型系统以及采用数据挖掘技术提出住宅建筑能耗分级标准，根据模型精确预测住宅建筑下一个月的电力能耗以及能耗等级，依据预测的建筑能耗等级对住户用电行为提出预警以及合理化建议。

本文采用的实验数据为佛罗里达州的住宅能耗数据，数据融合了气温变化、天然气消耗和建筑物自身特性信息。另外，本文的特征向量还囊括了上一个月的电力能耗，该特征向量覆盖信息广、具有高维度稀疏特性。将这些融合了多种维度的数据作为住宅建筑预测的特征向量，预测未来住宅建筑的能耗等级以及电力

消耗值。上述提到的能耗等级的划分时基于过去全体住宅建筑的能耗数据划分，以此构建能耗等级的标准划分能耗等级。

本文主要内容如下几个部分，针对住宅建筑能耗数据具有高维特征数据特性，通过实验比较典型的几种特征工程算法，并且结合 PCA、SVD 和 RF 提出了组合特征工程算法。针对无监督的聚类问题，由于聚类中存在初始中心点随机化问题，并且初始点的选择会对最终的聚类效果有显著影响，本文提出 PSO-Kmeans 算法解决该问题。设计实验比较改进的聚类算法性能优于原始算法。并将该改进型聚类算法划分住宅建筑电力能耗等级。由于能耗等级分类属于不平衡分类问题，后期并引入采样算法提高分类准确性。

1.6.2 章节安排

基于以上内容，本文共有六章，具体章节安排如下：

第一章,论述本课题研究的意义。讨论数据挖掘技术在建筑能耗预测的国内外研究现状以及应用现状，引出研究能耗预测系统的必要性，进一步指出此课题的目的及意义，并且指出在本课题研究中，数据挖掘过程中存在的问题，并且描述本文的主要内容和论文结构。

第二章，本章主要介绍在解决住宅建筑能耗问题预测时，需要采用的算法;在研究中，采用了多种回归和分类算法解决住宅建筑的能耗预测预测。并且具体介绍采样算法，解决后期预测过程中存在的数据平衡问题。

第三章，本章主要解决住宅建筑能耗预测问题。在解决实际场景问题时，如何构造特征向量，采用哪些特征工程算法。并且介绍各个算法在住宅建筑预测问题上的实验性能。

第四章，本文利用 K 均值算法对住宅能耗数据按季度做聚类分析，根据聚类算的中心点作为能耗分级的标准。本章提出了 PSO-Kmeans 的算法解决了 K 均值算法初始中心点随机化问题以及聚类中心数无法确认的问题。并通过实验仿真验证优化算法性能明显优于原始算法。在本章中，也将优化算法应用于美国佛罗里达州住宅建筑的能耗等级划分上，在将住宅能耗数据分级后，引出接下来的类不平衡问题。

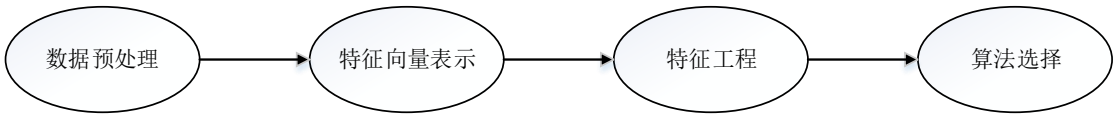
第五章，本章引入 SMOTEENN 采样算法解决住宅建筑物的能耗等级预测场景中不平衡分类问题。通过仿真实验，比较各个回归算法和分类器在住宅建筑能耗数据上的效果，最后提出了最适合住宅建筑术能耗预测的算法流程。

第六章，总结本文所做的工作，并提出进一步完善的方向。

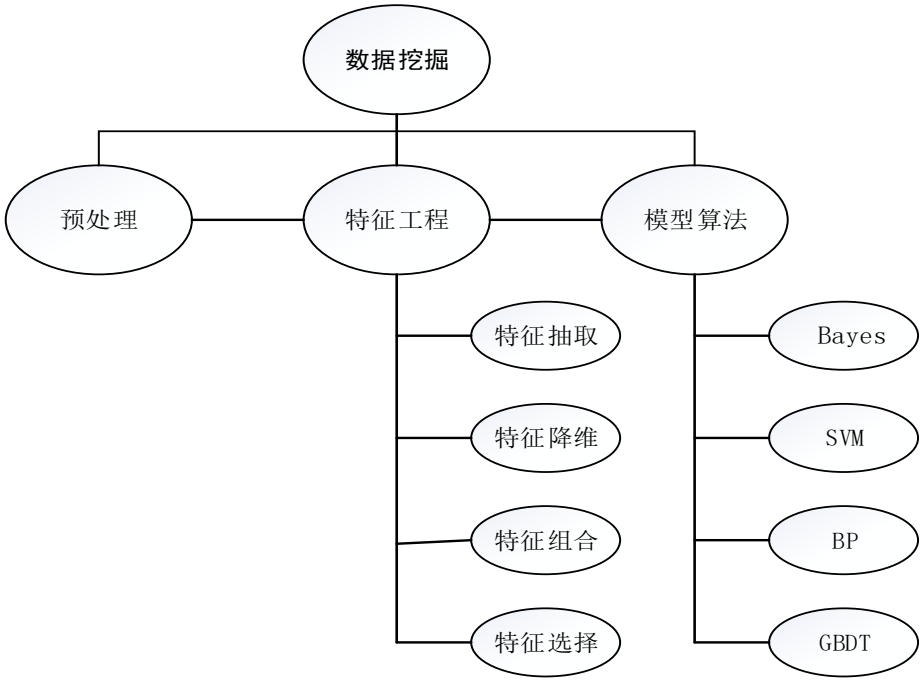
第2章 研究方法与技术

2.1 引言

本文主要采用数据挖掘手段解决住宅建筑能耗数据预测问题，研究的目的是预测住宅建筑下一个月的能耗值和下一个月的能耗等级，数据挖掘的主要流程图如 2.1 所示。数据挖掘的流程大致划分为数据预处理、特征工程和模型算法选择。本章的主要内容是介绍本文采用的数据挖掘中的一些技术手段如何解决实际场景中的问题。数据挖掘过程中，工作量最大的任务是特征向量的构建和，特征工程算法的选择，模型最后的结果很大程度上取决于以上两者的效果。



(a) 简要流程



(b) 详细步骤

图 2.1 数据挖掘步骤图

2.2 数据预处理

在数据挖掘任务中，首先需要处理海量的结构化和非结构化数据，从海量数据中选取何时的特征构造特征向量。数据挖掘任务将此过程称之为数据预处理[30-31]。数据预处理过程一般要解决包含弥补缺失值、人为噪声数据等问题。另一方面，由于每个维度的量纲不同，数据之间量纲不一样，数值差异较大会造成模型算法收敛速度较慢，结果准确度降低。数据预处理过程将各个特征转化为同一编码形式，如归一化、正则化等实用的技术手段已经在工业界得到了普遍的使用。
标准化：

$$x' = \frac{x-\mu}{\delta} \tag{2.1}$$

其中μ和σ分别为平均数、方差。
归一化:

$$x' = \frac{x-x_{min}}{x_{max}-x_{min}} \tag{2.2}$$

x_{max}, x_{min}数据中的最大、最小值，x 为归一化数据。

2.3 特征工程

数据预处理是数据挖掘任务中的首要步骤，其次特征工程[32] 也在数据挖掘任务中有着举足轻重的作用。算法模型的性能很大程度取决于特征工程和数据预处理步骤的处理。特征工程在工业界占据了数据挖掘任务很大的任务量，模型最后的准确性很大程度取决于这一步骤。这一步骤可能还需要进行特征向量的抽取、特征向量的组合等技术手段。由于数据挖掘任务的特征数据融合了多个维度的信息，这些数据杂乱并且没有任何规律，冗余的信息会降低算法的准确性；另一方面，算法模型的算法复杂即模型的建立时间是与输入的特征维度成正比，数据维度越高，算法运行时间越长，模型越难训练；因此特征工程是数据挖掘过程中不可缺少的步骤[33]。

特征选择是数据挖掘过程中一个重要的实验环节，在构造对象的基本特征向量之后，通常特征维度会很高，常规的分类器在训练过程中汇消耗时间，而且对实验效果影响较小或者无影响的无关特征会影响实验效果。特征选择主要包括信息增益(Information Gain)、开放拟合检验方法(CHI 估计)、互估计(Mutaul Information)，潜在语义分析(Latent Semantic Analysis,LSA)、期望交叉熵、相关系数等等，以及一些机器相关算法，例如主成分分析(Priciple Componente Analysis，PCA)、奇异值分解(Singular Value Decomposition,SVD)、随机森林(Random

Forest,RF) [36]等等。接下来介绍本文使用的降维算法原理。

2.3.1 主成分分析

主成分分析（Principle Component Analysis, PCA）是一种高效的统计方法[37-39]，算法通过正交变换将可能相关性很高的特征转化成一组线性不相关的特征向量，转变后的这组变量叫原来向量的主要成分。在数据挖掘中，为了准确、全面的分析某现实场景问题，往往事先需要先提取很多和结果相关的特征变量，这些特征信息在不同程度上反映了此种场景中的某些信息，但是大量的特征向量又带来维度灾难问题。所以 PCA 广泛的应用在各个场景下的机器学习和数据挖掘领域的科研实验中。每个特征所携带信息的大小通常使用离差平方和或者方差衡量，特征向量山的方法越大说明携带的信息量越大。如图 2.2 为 PCA 算法示意图。

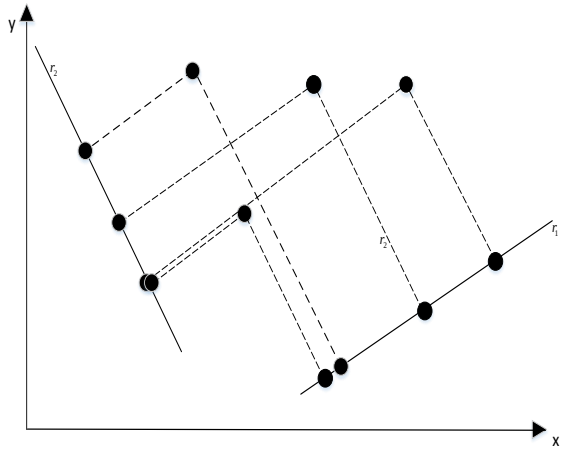


图 2.2 PCA 降维算法示意图

PCA 降维的目的是找到数据差异最大的维度，如上图所示，r_1, r_2为特征向量，将原始点映射到特征向量空间上，找到带有信息量最大的维度，上图所示应为r_1方向。数据特征降维是为了降低算法开销、提取有效信息和去除噪声。PCA 算法步骤如表 2.1 描述如下：

表 2.1 PCA 降维算法步骤

输入：样本集 D=$\{x_1, x_2, x_3 \ldots\ldots x_n\}$ 低维空间维数 d
对样本进行中心化：$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$，并用原数据减去均值：$x_i' = x_i - \bar{x}$
计算样本协方差矩阵XX^T
计算协方差矩阵XX^T的特征和特征向量
由大到小排列特征值及对应的特征向量，选取最大的 d 个最大的特征值对应的特征向量组成变换矩阵 W。

计算降维之后的矩阵$W^T X$

2.3.2 奇异值分解

奇异值分解（Singular Value Decomposition, SVD）[40-42] 又称为特征值分解，该算法能将高维矩阵转化成低维矩阵，起到特征降维的算法效果。奇异值分解可以将不需要的奇异值去除，利用有效的奇异值进行矩阵重构后达到有用信息的目的。矩阵奇异值分解算法在数据挖掘场景中有着很广泛的应用。实验中生成的大部分矩阵都不是方阵，如何提取有效的信息描述这样的矩阵。SVD 是解决这种问题的常用手段。

$$(A^T \mathrm{A})\mathrm{x} = \lambda \mathrm{x} \tag{2.3}$$

求得的特征值就对应奇异值的平方，求得的特征向量 v 被称为右奇异向量，另一个有：

$$\sigma_i = \sqrt{\lambda_i} \tag{2.4}$$

$$u_i = \frac{Av_i}{\sigma_i} \tag{2.5}$$

其中，所求的u_i就是左奇异向量，σ_i为奇异值。

对于一个$\mathrm{n} * \mathrm{n}$的相互垂直的网格，SVD 方法可以将其变换到另一个形式的垂直网格,SVD 的变换示意如图 2.3 所示。

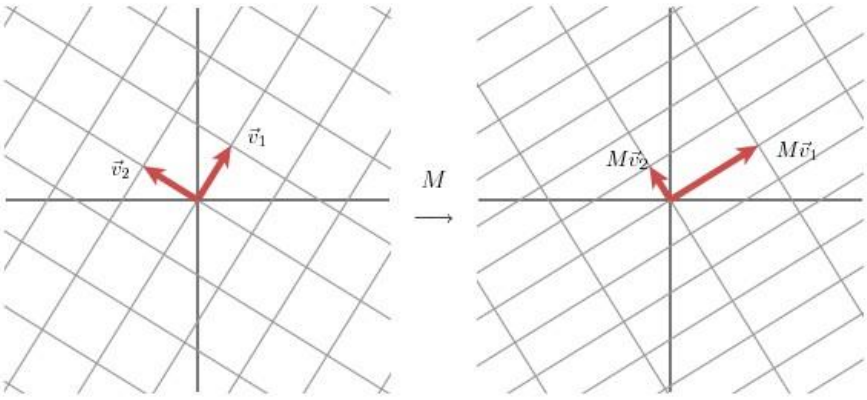


图 2.3 SVD 中垂直向量的变换图

如图 2.3 所示，SVD 的变化就是找到两个相互正交的单位向量v_1和v_2，则mv_1和mv_2正交。$\mathrm{M}v_1$和$\mathrm{M}v_2$的单位向量可以表示为u_1和u_2，$\sigma_1 \cdot u_1 = \mathrm{M}v_1$和$\sigma_2 \cdot u_2 = \mathrm{M}v_2$。$\sigma_1$和$\sigma_2$，被称作矩阵 M 的奇异值，分别表示不同方向向量上的模。

这样就有如下关系式：

$$\begin{cases} \mathrm{M}v_1 = \sigma_1 u_1 \\ \mathrm{M}v_2 = \sigma_2 u_2 \end{cases} \tag{2.6}$$

本文对线性变化公式进行简单推导,由于向量v_1和v_2是正交的单位向量，我们可以得到如下的式子：

$$\mathrm{x} = (v_1 \cdot x)v_1 + (v_2 \cdot x)v_2 \tag{2.7}$$

最终的式子为：

$$\mathrm{M}x = u_1\sigma_1 v_1^T x + u_2\sigma_2 v_2^T x \tag{2.8}$$

$$\mathrm{M} = u_1\sigma_1 v_1^T + u_2\sigma_2 v_2^T \tag{2.9}$$

上述关系式可表示成：

$$\mathrm{M} = \mathrm{U}\Sigma v^T \tag{2.10}$$

其中 u 矩阵的列向量分别是u_1, u_2，Σ是一个对角矩阵，对角元素分别是对应的σ_1和σ_2，v 矩阵的列向量分别是v_1, v_2。

任意矩阵都可以分解成三个矩阵的相乘，虽然 SVD 的效果可以通过 PCA 的降维获得，但是 SVD 更具有稳定性，应用范围更广的特点。

2.3.3 随机森林

随机森林（Random Forest ,RF）是由 Brieman[43]于 2001 年提出的一个集成学习算法框架，以决策树为最小单位，随机从特征空间中选择节点作为分裂节点，采用 Bagging 算法思想将多个训练集合构建决策树。

随机森林的随机性是体现在实例和特征的选择上以确保基分类器的多样性；虽然在构建决策树时，需要在特征空间的子空间中选择最佳的信息增益点即最佳分裂点，虽然选取时随机化的，但是属性的贡献值并不是完全随机计算的，二是根据一定的规则选取，从而保证基分类器的准确性。

显然，随机森林是由多个树组成的，随机森林的构成过程即是多个决策树的构建过程，因此随机森林也相应的继承了决策树选择的“重要”特征的能力。当某一个重要特征出现噪声时，模型算法的预测性能会显著降低。

随机森林也适用于对重要特征进行挑选的场景中，为了达到这一目的，随机森林采取了对特征重要程度进行量化的方式选择又有特征组合。常见的技术方法包括：统计特征作为分割特征的频度表明它的重要性程度，分割的次数越多表明越重要；构建决策树的时候，节点的分割效果度量采用基尼指数（Gini Index）的方法，计算特征的“Gini Importance[44]”和在 OOB 样本上计算特征的“Permutation importance”[45]等都是衡量特征重要性的手段。

基尼指数（Gini Index）作为另一种常用的分割节点效果的度量指标，同信息增益一样长期占据着信息度量的榜首位置。使用随机森林对特征重要性度量的时候，经常应用基尼系数作为衡量指标。每一个特征在决定分割节点的时候，是计

算该特征的作为分割特征的效果，上文提到的基尼系数越小，说明该特征越重要，分割效果越好反之，分割效果则比较差。

随机森林作为一种全局特征挑选方式，只适用于分类情况场景中，通过改变某些特征数值，改变分类结果，从而计算特征的重要程度。在回归场景中，由于输出是一组连续值，所以无法通过改变特征值，来求特征的重要程度。

2.4 模型算法选择

在机器学习和数据挖掘领域中，一般将任务分成有监督和无监督学习两种任务形式。分类和回归问题在数据挖掘过程中属于典型的有监督学习问题。传统的有监督学习(Supervised learning),例如朴素贝叶斯算法(Native Bayes)[46]、SVM 算法、最邻近算法[47](K-Nearst neighbor)、逻辑斯特回归算法（Logistic Regression）[48]，以及集成算法如 Adaboost、随机森林算法(Random Forest)、GBDT[49]算法、神经网络算法(Backpropagation)[50]等等。通常意义上讲，分类问题也是回归问题中的一个子问题，因此分类算法也同样适用于回归问题。在现实生活场景中，每个算法凭借自己独特的理论基础，都有着自己的应用场景，所以在不同的数据挖掘场景中，每个场景的适应性也不尽相同。因为在数据挖掘中，不同的特征属性需要应用不同的算法模型。在本文的后续实验中，采用不同的实验设计比较结果，选取不同的算法组合模型期望得到最优的实验结果。

2.4.1 支持向量机

支持向量机(Support Vector Machine ,SVM)是 Cortes [51]和 Vapnik 于 1995 年发表的，该算法一经提出就在分类任务中显示了卓越性能，此算法很快被研究人员应用到数据挖掘的各个现实场景中，本文的回归和分类预测问题中也采用这一算法。SVM 是以统计学习为基础的一种新型机器学习算法，此算法克服了神经网络和传统机器学习算法中存在的局部最优点和维数灾难等问题。支持向量机是基于有限的数据信息，既要降低算法模型复杂性也要维护模型的泛华能力，模型在寻求一种最佳的折中方式。在充分拟合现有数据的同时，并且还能对测试样本数据集有很强的预测泛化能力。支持向量机的求解借助于凸函数求解全局最优解，此种计算方式可以最大限度保持对未知预测样本的推广泛化能力。支持向量机通过独有的核函数的机制将原始特征空间中的非线性分类界面映射到高维特征空间中，在高维特征空间找出最佳划分面。正因为如此，SVM 算法凭借其特有的算法框架，一开始就引起了学术界的强烈反响。时至今日，在机器学习、数据挖掘和

模式识别等领域都依然可以见到支持向量机的身影；另外，该算法作为典型的有监督学习模型，依然可以用于分析数据、模式识别、分类以及回归分析等工业场景。SVM 在工业界广泛应用于二分类或者多分类场景中。SVM 是一种以结构经验最小原理和 VC 维理论文基础[52]的统计学习理论。VC 维理论简单来说就是一个问题的复杂程度取决于 VC 维，VC 维越高，应用场景就越复杂。算法处理数据的本质是和维度无关或者关联度较小。

机器学习中主要是针对数据中的输入和输出建立依赖关系，能够达到很好的泛化能力，准确的预测未来某个样本的输出值。线性支持向量机的目标函数为：

$$f(x) = w^T \cdot x + b \tag{2.11}$$

是为了能够找到一个截距使得式中的γ最大。

$$\gamma = \frac{1}{||w||}(w \cdot x + b) \tag{2.12}$$

在划分之后可以将平面划分为两个部分，并且此时的划分效果最强，泛化能力最佳。在训练模型时，在保证模型泛化性的同时，也需要尽量对现有数据进行充分利用。大量的训练数据引申出凸二次规划问题，此问题转化成一个 NP-hard 问题[53]。在模型的训练过程中，通常对目标函数引入拉格朗日乘子，原始目标函数问题即转换成拉格朗日对偶问题。拉格朗日对偶函数一般表示为：

$$\mathrm{L(w, b, \alpha)} = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i(y_i(w^T x_i + b) - 1) \tag{2.13}$$

在训练分类器的同时，原始问题及转化成对偶问题，要求的 w 和 b 通过引入的α的计算得到。原始问题的对偶问题转化为目标函数的极大极小问题：

$$\max_{\alpha} \min_{w,b} L(w, b, \alpha) \tag{2.14}$$

SVM 通过非线性变化φ(x),将输入空间的低维数据映射到高特征空间（希尔伯特空间）[54]，如果支持向量机的求解只用到内积运算，而在训练模型过程从而中，在低维度的欧式空间，有存在着$(x_i \cdot x_j)$这样的内积运算，即可转换成表达式：

$$\mathrm{k(x, x')} = \beta(x \cdot x') \tag{2.15}$$

也可写成：

$$\mathrm{k(x, x')} = \varphi(\mathrm{x}) \cdot \varphi'(x') \tag{2.16}$$

在式（2.16）中，函数$\mathrm{k(x, x')}$ 被称为核函数[55]。核函数的引入为支持向量机。因此 SVM 在非线性数据集下的应用奠定了良好的基础，核函数如图 2.4 所示，经过高斯核变换后可以直接转化成线性分类。

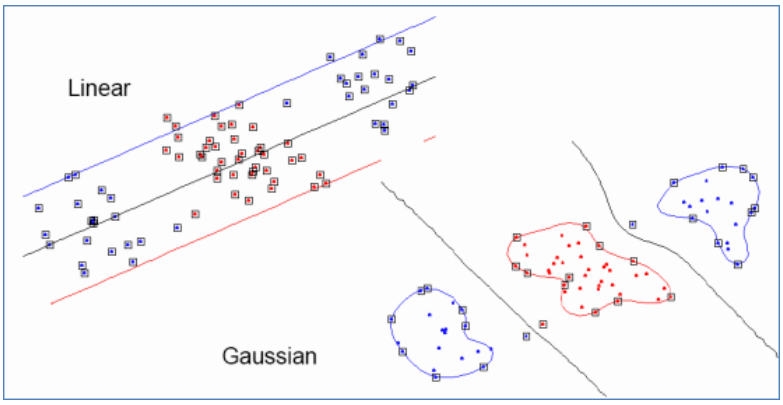


图 2.4 核函数机制示意图

综上所述，核函数是利用映射函数，将低维数据映射到高维数据中，并且转化成线性分布。也就是说，在事先选定核函数k(x, x')的条件下，可以将非线性问题转化为线性的分类问题的支持向量机[56]。在实际应用场景中，核函数的选择常常需要工程经验为基础。在工业以及学术领域中，常用的核函数有：

（1）线性核（Linear Kernel）：

$$K(x,y) = x^T y + c \tag{2.17}$$

（2）多项式核（Polynomial Kernel）：

$$K(x,y) = (ax^T y + c)^d \tag{2.18}$$

（3）高斯径向基核（Gaussian Kernel）：

$$k(x,y) = \exp(-\frac{||x-y||^2}{2\sigma^2}) \tag{2.19}$$

径向基核函数是指取值仅仅依赖于特定点距离的实值函数，也就是$\varphi(x,y) = \varphi(||x-y||)$。任意一个满足这个特性的函数φ都叫做径向基核函数，标准的核函数的距离计算都是使用欧氏距离[57]。

（4）幂函数（Exponential Kernel）：

$$k(x,y) = \exp(-\frac{||x-y||}{2\sigma^2}) \tag{2.20}$$

（5）拉普拉斯核（Laplacian Kernel）：

$$k(x,y) = \exp(-\frac{||x-y||}{\sigma}) \tag{2.21}$$

核函数的广泛使用，不仅解决了“维数灾难”问题，减少了算法模型的计算时间,并且可以在不需要知道任何参数的前提下，可以将低维数据映射到高维数据，从而将低维空间中的非线性问题转化成高维特征空间中的线性可分问题。这样在高维特征空间中可以应用一个平面或者为降一维平面划分数据集的正负类。并且所需要调节的参数指标非常少，很有利于工程人员的算法调节工作。

2.4.2 梯度提升树

梯度提升树(gradient boosting decision tree,GBDT)是 2001 年 Friedman[58] 提出的一种将 boosting 算法与决策树构建思想相融合的产物。它是一种迭代的决策树算法，该算法由多颗决策树组成，该模型的建立是以所有树的误差总和相加作为目标函数。

GBDT 中 Boosting 的算法思想体现在每颗决策树训练的是历史决策树分类结果中的结果，以此结果为导向重新训练一颗新树。具体算法如图 2.5 所示。

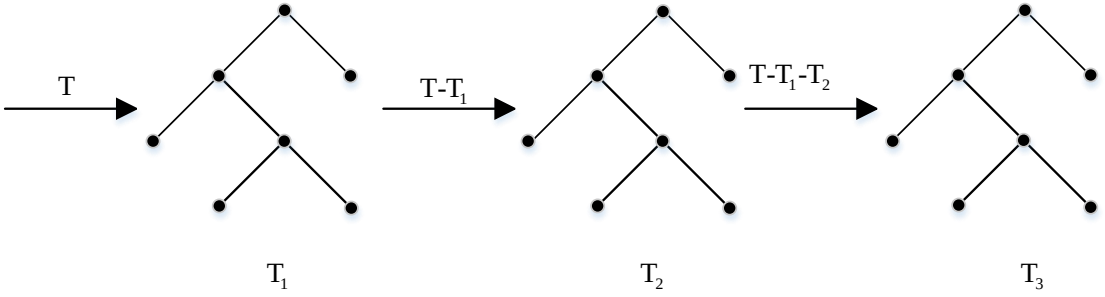


图 2.5 GBDT 算法思路示意图

如图 2.5 所示，GBDT 的训练过程串行的，每次只能训练一颗树，第二颗树的训练目标是减小第一课树 T_1 的训练结果与真实值的残差，接下来树的训练目标是将将历史训练过程中的每一个树的结果残差相加作为目标，即：

$$T=T_1+T_2+T_3 \tag{2.22}$$

对于损失函数的迭代优化选择一般是直接对残差进行优化或者是选取梯度下降值进行优化。如图 2.5 所示，GBDT 与传统的 boosting 在本质上有所区别，他们两者的优化目标一个以迭代为目标，一个十一重新采样为目标。以二元分类为例，如图 2.6 所示：

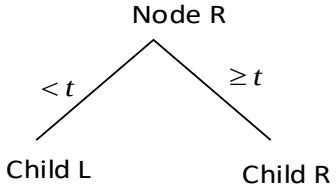


图 2.6 二元分类示意图

对于一个待分裂的节点 R，其输出值以不同样本 y 的平均值 μ 作为节点的输出值，即：

$$\mu=\frac{\sum_{i=1}^{n} y_i}{n} \tag{2.23}$$

于是图中的节点误差可以表示为：

$$Error = \sum (y_i - \mu)^2 \tag{2.24}$$

节点的分裂过程中，需要选择分裂增益最大的属性进行划分，分裂增益 G 的计算方法如：

$$G = S - S_j \tag{2.25}$$

在训练的过程中一般采用方差作为损失函数，可以得到S_j，如公式：

$$S_j = \sum_{m \in L} (y_m - \mu_L)^2 + \sum_{n \in R} (y_n - \mu_{\mathrm{R}})^2 \tag{2.26}$$

分别将 S，S_j展开，如公式：

$$\mathrm{G}=(\frac{\mathrm{Sum}_L^{\ 2}}{\|L\|}+\frac{\mathrm{Sum}_R^{\ 2}}{\|L\|})-\frac{\mathrm{S}um^2}{\|\mathrm{Total}\|} \tag{2.27}$$

其中，$\mathrm{Sum}_L^{\ 2}$和$\mathrm{Sum}_R^{\ 2}$分别表示字数中所有样本的平方和，$\mathrm{S}um^2$表示所有样本的平方和。于是，只需要对 G 求取最优解即可。

2.4.3 神经网络

Meculloch 和 Pitts 对神经网络进行了简化，如图 2.7 所示，在人工神经元结构中，如果共有 n 个输入，其中$x_i\ (i = 1,2 \cdots n)$为来自与其相连的第 i 个神经元为输入。$\omega_i\ (i = 1,2 \cdots n)$代表输入第$i$个神经元与当前神经元的连接权重，神经网络中上一层的每一个神经元都会对下一层的每一个神经元产生权重影响。神经网络的算法算法推导就是解决每个神经元对其他神经元的权重问题。设当前神经元的阈值为θ，激活函数为f，激活函数的就是选择阈值，在何种情况下会激活该层的神经元对下一层的神经元产生影响。则当前神经元的输出计算公式如下。

$$y = f(\sum_{i=1}^{n} \omega_i x_i - \theta) \tag{2.28}$$

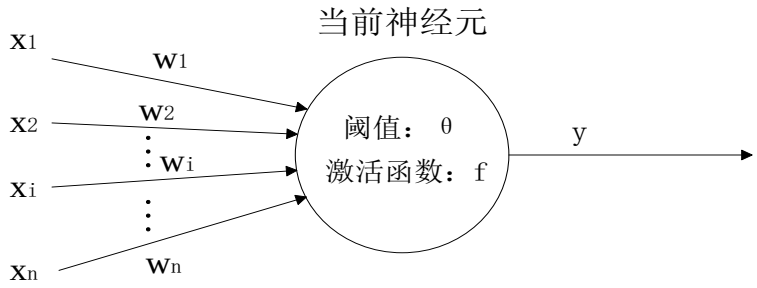


图 2.7 人工神经元结构示意图

通常，常用的神经网络激活函数主要有三种：

① 阶跃函数： $sgn(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0. \end{cases}$

② Sigmoid 函数: $sigmoid(x) = \frac{1}{1+e^{-x}}$

③ 双曲正切函数： $g(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

如图 2.7 所示为神经网络中最基本的神经单元，每个神经网络中的许多个人工神经元进行互相连接后组成的网络称为神经网络。图 2.8 为一种典型的多层前馈神经网络示意图，该种神经网络结构中，层与层直接按的神经元连接都是采用全连接的方式，而且连接是单向的，相隔两层之间不存在连接关系。如图 2.8 是一个较为简单的前馈神经网络结构，共包含三层：输入层，一层隐含神经元层以及输出层，期望通过输入层数据$x_1, x_2 \cdots x_d$预测输出层$y_1, y_2 \cdots y_l$的值。目前，在多层前馈神经网络的参数求解过程中通常采用误差逆传播算法（Error Back Propagation Algorithm，简称 BP 算法）。误差传播就是以实际值和预测值的均方差为导向，以最大速率减少均方差的大小。为了达到这种目的，通常是是采用梯度下降的原理，以斜率最大的方向为下降点求取下一次迭代时每个神经元的权重，通过这样不停的迭代，是均方差值，趋于稳定时的神经元参数就是最后每个神经元的权重。下面以图 2.8 所示的神经网络结构为例，介绍利用神经网络进行预测的原理。

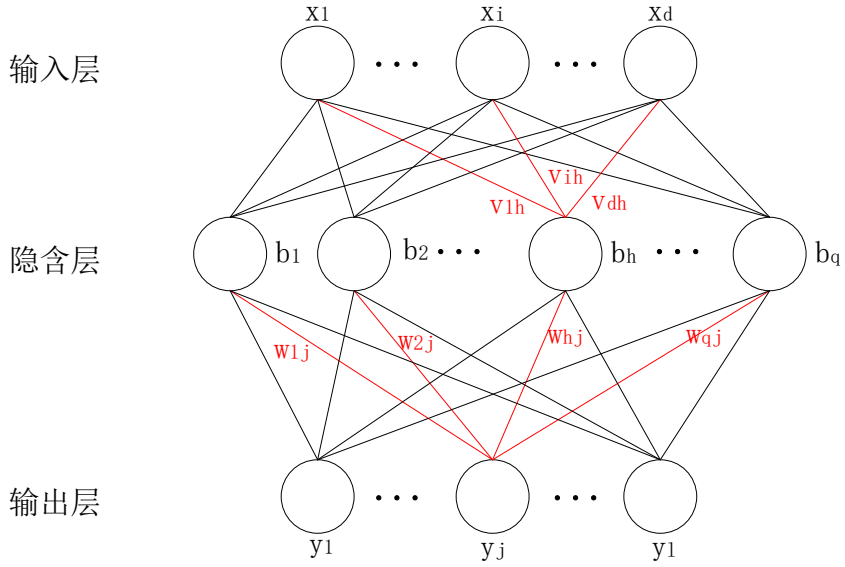


图 2.8 BP 神经网络结构图

根据定义，隐含层神经元h的输入α_h的计算公式如下：

$$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i \tag{2.29}$$

同理，输出层神经元j的输入β_j可由如下公式计算：

$$\beta_j = \sum_{h=1}^{q} w_{hj}\, b_h \tag{2.30}$$

此时，输出层神经元j的输出为

$$y_j = f(\beta_j - \theta_j) \tag{2.31}$$

其中，v_{ih}——上一层的神经元i对隐含层节点h的权重；

γ_h——隐含层神经元h的阈值；

w_{hj}——隐含层的神经元h对输出层神经元j的权重；

b_h——隐含层神经元h的输出；

θ_j——输出层神经元j的阈值；

f——激活函数。

假设误差逆传播算法的最终目标为最小化均方误差。假设现有训练样例$(\boldsymbol{x_k}, \boldsymbol{y_k})$，则相对应的神经网络输出为$\hat{\mathbf{y}}_{\boldsymbol{k}} = (\hat{y}_1^k, \hat{y}_2^k \cdots \hat{y}_l^k)$，且$\hat{\mathbf{y}}_{\boldsymbol{k}}$的属性值$\hat{y}_j^k = f(\beta_j - \theta_j)$.

从而，训练样例$(\boldsymbol{x_k}, \boldsymbol{y_k})$的均方误差为

$$E_k = \frac{1}{2}\sum_{j=1}^{l}(y_j^k - \hat{y}_j^k)^2 \tag{2.32}$$

整个训练数据集上的累积误差为：

$$E = \sum_{k=1}^{m} E_k \tag{2.33}$$

如**错误!未找到引用源。**的示意图所示的神经网络中共有$(d + l + 1)q + l$个参

数需要确定，分别为输入层到隐含层共dq个连接权重，隐含层q个阈值，隐含层到输出层ql个连接权重以及输出层l个阈值，神经网络即通过梯度下降法求得$(d+l+1)q+l$参数的值，由此构建整个神经网络传递图。参数的求取即是 2.32 的均方误差最小。可采用梯度下降法求解参数值。由于篇幅原因，本节不介绍具体的参数不作详细的推导，感兴趣的读者可以阅读文献[59]。

上文提到的梯度下降法的参数更新公式如下所示：

$$w_{hj} \leftarrow w_{hj} + \Delta w_{hj} \quad (2.34)$$

$$\theta_j \leftarrow \theta_j + \Delta\theta_j \quad (2.35)$$

$$v_{ih} \leftarrow v_{ih} + \Delta v_{ih} \quad (2.36)$$

$$\gamma_h \leftarrow \gamma_h + \Delta\gamma_h \quad (2.37)$$

其中，$\Delta w_{hj} = \eta g_j b_h$；$\Delta\theta_j = -\eta g_j$；$\Delta v_{ih} = \eta e_h x_i$；$\Delta\gamma_h = -\eta e_h$；$b_h = f(\alpha_h - \gamma_h)$；$g_j = \hat{y}_j^k(1-\hat{y}_j^k)(\mathrm{y}_j^k - \hat{y}_j^k)$；$e_h = b_h(1-b_h)\sum_{j=1}^{l} w_{hj} g_j$；$\eta$为学习率。

基于均方误差最小化以及梯度下降法更新权重，BP 神经网络的求解方法步骤见表 2.2。

表 2.2 BP 神经网络算法步骤

输入： 训练数据集D = $\{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}) \cdots (\boldsymbol{x_m}, \boldsymbol{y_m})\}$；

学习率η；

停止条件阈值δ。

过程：

随机初始化网络所有的参数，取值范围为(0,1);

while 累积误差$E \geq \delta$

for 训练数据集中的每个样本 $(\boldsymbol{x_k}, \boldsymbol{y_k})$ do

根据当前网络参数以及公式 2.31 计算当前样本的估计值$\hat{\boldsymbol{y}}_{\boldsymbol{k}}$；

根据公式 2.34、2.35、2.36 以及 2.37 更新网络参数；

end for

end

输出： 连接权值与阈值确定的多层前馈神经网络。

2.5 稀疏不平衡分类问题

在后期对住宅建筑能耗等级预测过程中，存在类不平衡问题，即每个等级中的数据量不相等。如果直接对不平衡数据集进行住宅建筑能耗等级预测会存在错分少数类和对多数类过拟合问题。而本文后期的实例求取处理过程中，对住宅建

筑能耗数据进行少类的过采样和多类的欠采样方法解决类不平衡问题。采用的技术手段如下面两节所示。

2.5.1 SMOTE 算法

SMOTE 算法是 Chawla[60]等人于 2002 年首次在人工智能杂志上提出。SMOTE 算法的提出是针对不平衡数据集中的少类样本进行过采样，该算法是至今为止最经典的过采样算法，很多学者由此引申出很多基于边界分类的采样算法，但是依然不能降低此经典算法在不平衡分类问题中的应用。该采样算法的过采样思想主要是通过在少数类的数据点之间随机插入新样本以达到平衡数据集样本的目的。SMOTE 算法的特点是不复制原有数据，而是通过新增数据点的方式平衡数据集，因此此种方法在一定程度上可以避免分类器的过渡拟合，提升模型算法的分类能力和预测的准确性。

算法流程是：假设过采样倍数为 n,对每个属于少数类中 P 的样本x_i,首先找到它在 P 中的 k 个临近样本，再这 k 个临近样本中选择 n 个样本$x_{ij}(j = 1,2,\dots,n)$,然后按照如下公式合成新的少数样本点$\mathrm{y}_j(j = 1,2,\dots n)$:

$$\mathrm{y}_j = \mathrm{x}_i + rand(0,1) * (\mathrm{x}_i - \mathrm{x}_{ij}) \tag{2.38}$$

其中$rand(0,1)$表示去见(0,1)之间的任一随机数。将 新产生的数据样本添加到数据集中达到平平衡数据集的作用。在实际应用过程中，需要设置的参数就是采样比率，根据数据集不平衡的程度来设定，即决定最后的采样样本容量。算法流程如下表 2.3 所示。

表 2.3 SMOTE 算法步骤

输入原数据集 T，N 为生成样本百分比，K 为最邻近个数
置 i=0
每次迭代将 i 加 1，直至训练样本中的样本个数，并进行如下操作：
对每个样本$x_i \in$ T，计算样本 x_i的 K 个邻近值，生成数组 sample
置 index=0 ,每次循环 index 加 1 直至 N：
在（1，k）随机生成一个整数 j:
y=$\mathrm{x}_i + rand(0,1) * (\mathrm{x}_i - \mathrm{sample}_i)$
将生成的 y 加入到生成的样本T_{smote}
返回T_{smote}和 T 的集合

SMOTE 过采样过程如下图 2.9 所示,图中的r_1、r_2、r_3、r_4即为增加的少数类样本。

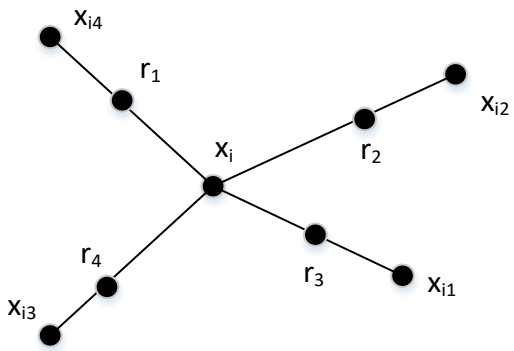


图 2.9 SMOTE 算法示意图

2.5.2 ENN 算法

Wilson 等人提出的编辑最近邻规则（Edit Nearst Neighbor, ENN）算法[61]是基于多类样本的数据处理算法。ENN 算法的目的是消除导致错误分类的样本数据集。ENN 寻找特定多类数据集周围的临近样本，如果临近样本中的数据大多数样本类别与自身类别不同，此数据可能对后期的实验产生影响，因此将其从原数据集中删除，降低不平衡比例。算法示意图如图 2.10 所示，，▲代表少数类 ●代表多类左图为原始样本，右图为 ENN 算法之后的样本。ENN 算法步骤如下表 2.4 所示，其中训练集定义为 T，最终的集合为T_{enn}。

表 2.4 ENN 算法步骤

1）置 i=0, $T_{enn} = T$
2）每次对迭代计数器 i 加 1，直至训练样本中的样本个数，：
比较数据集中的x_i在 T 中的 K 个近邻中的多数样本与x_i的类别，如果不同，则将x_i从原数据集中删除
如果迭代计数器 i 小于训练集 T 中的样本个数，返回步骤 2
得到T_{enn}为最终原型样本集，结束。

图 2.10 ENN 算法示意图

2.6 本章小结

本章着重介绍机器学习算法的发展现状，指出了数据挖掘任务中的主要流程框架，以及提出了数据挖掘中一些常见的问题，例如数据挖掘任务中高维度稀疏性问题和类别不平衡问题的解决算法方案，并且详细介绍了算法原理，为接下来的住宅建筑能耗预测问题的解决作准备。在本小节中针对本课题的研究内容绘制了该研究的算法流程图，如图 2.11 所示。

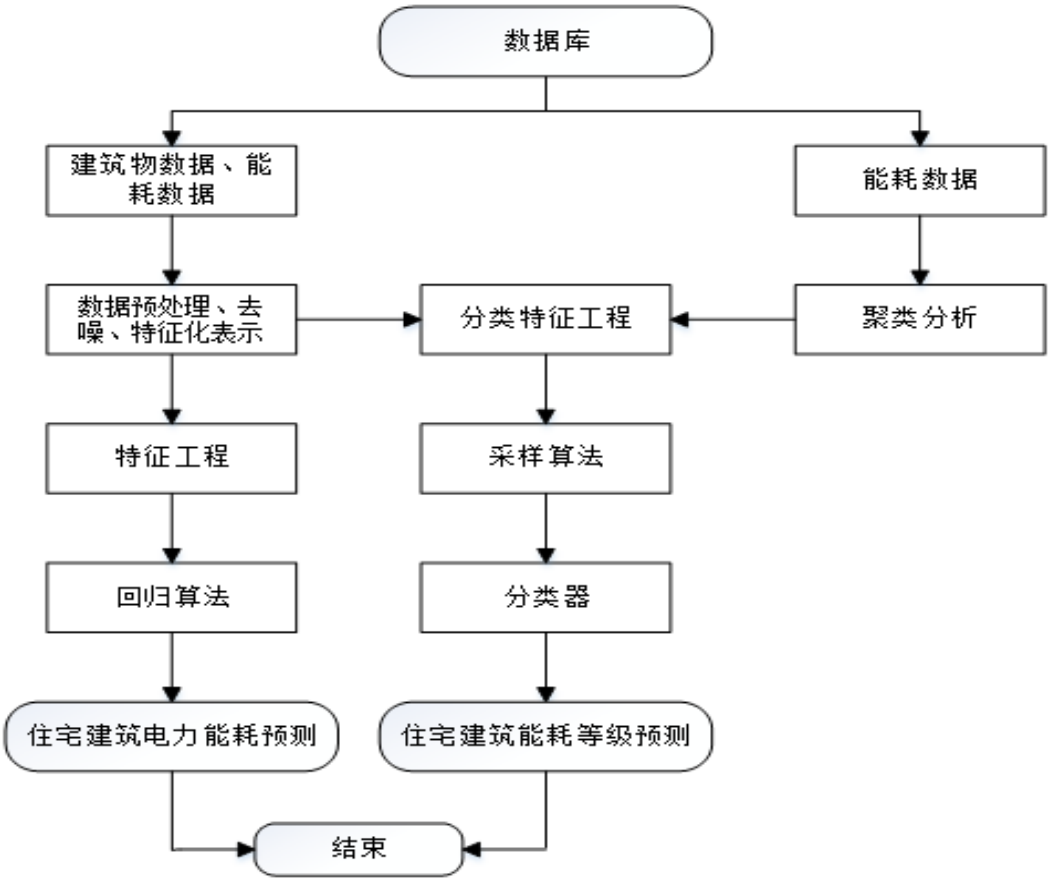


图 2.11 研究框架图

第3章 住宅建筑能耗预测

3.1 引言

本章内容为提出住宅建筑能耗预测问题的解决方案，在解决该问题时，主要是如何构建住宅建筑的特征向量。在构建特征向量时，要剔除数据中的脏数据和冗余数据，避免对预测结果产生影响。在特征工程处理中，主要解决特征向量中存在的冗余信息和维数灾难问题，利用算法提取特征性向量中的主要成分和变换特征空间；在后续的实验设计中，比较各个算法在解决住宅建筑能耗预测问题时的实验性能，并选取最优的算法组合解决该实际问题。

3.2 住宅建筑能耗数据预处理

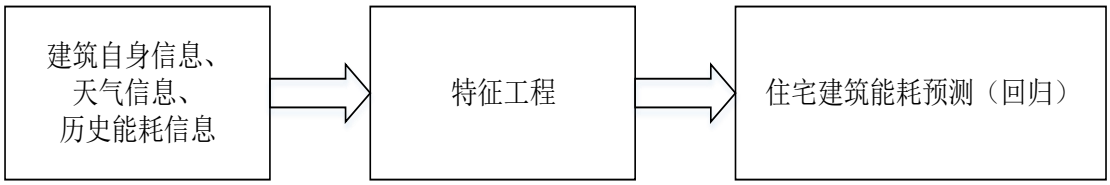


图 3.1 住宅建筑能耗预测步骤图

本章解决的是住宅建筑能耗预测问题，采用的佛罗里达州的住宅建筑数据，数据来自 https://openei.org/datasets/dataset/doe-buildings-performance-database-sample-residential-data 网站。根据建筑物的邮编收集天气信息。输入是建筑自身的建筑属性、天气信息和历史能耗信息，其中建筑物自身信息为 14 维；天气信息共包括了气温、气压、湿度、降雨量、降雪量、阴天数、雷雨天气数，这些信息被拆分成了 28 维数据；历史能耗信息包括住宅建筑上一个月的天然气和电力能耗值。预测的输出为住宅建筑下一个月电力能耗值。建筑物自身信息如下表 3.1 所示。

表 3.1 建筑物自身属性

建造完成时间	人口调查区域	面积	窗户玻璃类型	墙体类型
暖气装置	墙体密度	层数	降温系统	邮编号码
辅助供暖燃料	降温燃料标号	降温系统标志位	气候区标号	

3.3 住宅建筑能耗预测实验结果及其分析

评价标准

在回归问题中一般采用R^2作为回归问题的评级标准，该参数表明在多元回归问题中的回归平方在平方和中的比例即算法的拟合程度。

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2} \tag{3.1}$$

均方差

$$\delta = \frac{1}{n}\sum_{i=1}^{n}(x_{pre\,i} - x_i)^2 \tag{3.2}$$

均方差是个数据预测值偏离真实值的距离平方和的平均数，也就是常说的误差平方和的平均数，计算公式接近方差。δ越小说明预测越准确，不过当实际值较小时，即是预测误差较大，δ也会很小，所以均方差并不适用于所有的回归标准，在回归评价中，一般采用R^2作为评价参数评价算法性能。。

3.3.1 实验策略

在本实验中主要采用比较算法的形式，选取最优的算法组合。本实验中，由于是回归问题，因此在特征工程选择方面不能采用随机森林选取最优的特征子集合，本章在实验方面主要采用 PCA 和 SVD 为降维方法。并且设计实验比较算法在住宅建筑数据上的实验性能。

实验策略一，本策略为不采用任何特征工程，直接将住宅能耗数据进行归一化处理，即采用回归算法预测住宅建筑电力能耗值。每个的实验结果为每个回归算法在住宅建筑能耗数据的测试集上运行 20 次，计算 R^2、均方差和代码运行复杂度。

实验策略二，由于 RF 不适用于回归问题的特征工程，此策略以 SVM 为回归算法，比较 PCA、SVD 和 PCA+SVD 在住宅建筑能耗数据场景的性能。实验结果对各个特征工程算法运行 20 次，比较 R^2 和均方差、运行时间三个指标的性能差异。

实验策略三，此次实验，以 PCA+SVD 的组合算法为特征工程的主要方法，引入 BP、GBDT、Logistic 和 SVM 作为回归算法，住宅能耗数据经过数据归一化预处理和特征降维步骤，比较四种回归算法在住宅建筑数据上的性能差异。此次实验对各个回归算法运行 20 次，算出这 20 次实验评价参数的平均值，比较 R^2、均方差和运行时间三个指标的性能差异以便挑选最好算法组合框架解决住宅建筑预测问题。

3.3.2 实验结果

根据实验策略一设计实验的结果如下表 3.2 所示。实验结果表明，在没有任何降维算法的情况下，SVM 算法有很好的实验结果，SVM 的核函数将低维特征映射到高维特征空间中，在高维特征上对住宅建筑数据进行回归运算。因此 SVM 的高性能是以牺牲 CPU 运行时间为代价。

表 3.2 不采用特征工程算法的实验结果图

算法	参数	结果	CPU 运行时间
BP	R^2	0.569	2.253
	均方差	0.062	
Logistic	R^2	0.551	0.297
	均方差	0.065	
GBDT	R^2	0.569	375.9
	均方差	0.061	
SVM	R^2	0.751	1869
	均方差	0.036	

根据实验策略二的设定，此次实验以 SVM 以回归算法，比较各个降维算法在住宅建筑能耗数据的性能。实验结果如下表 3.3 所示。实验结果表明 PCA 的降维时间多于 SVD 的降维时间，组合算法的运行时间最长，但是组合算法的R^2和均方差指标却是最优的。较好的实验效果是以牺牲运行时间为代价。因此在后续的实验中，组合算法是特征工程的首选算法。

表 3.3 特征工程算法在能耗数据实验结果

降维算法	参数	结果	CPU 运行时间
PCA	R^2	0.626	761
	均方差	0.052	
SVD	R^2	0.619	819
	均方差	1.6e-5	
P+S	R^2	0.85	712
	均方差	0.039	

根据实验策略三，以组合算法为特征工程降维算法；此次实验主要是比较不同的回归算法在住宅简述能耗数据上的实验效果，选取最好的算法组合解决住宅建筑能耗预测问题。本次实验如表 3.4 所示。实验统计结果表明，Logistic 和 GBDT 在住宅建筑能耗预测场景上的效果并不理想，算法的优势是复杂度不高，模型运

行时间不长。而 BP 和 SVM 算法针对回归问题需要消耗大量的时间。高质量预测结果是以牺牲算法复杂度为代价，但是模型的预测时间在可接受范围之内并且算法最后的预测结果也能达到课题开始的目标。。

表 3.4 能耗预测实验结果

算法	参数	结果	CPU 运行时间
BP	R^2	0.586	2964.3
	均方差	0.060	
Logistic	R^2	0.303	0.039
	均方差	0.091	
GBDT	R^2	-0.073	681
	均方差	0.173	
SVM	R^2	0.85	712
	均方差	0.039	

最终本课题基于住宅建筑能耗数据的回归问题，采用以 SVM 算法为基础，PCA+SVD 为特征工程方法，数据特征经过归一化处理，采用十字交叉的方法，预先留下十分之一的数据集作为测试集。所得的预测结果如表 3.5 所示。提取其中 3000 个预测点作展示，以实际值为横坐标，预测值为纵坐标绘制结果示意图如图 3.2 所示，图中虚线为y = x。如图表明，算法的泛化能力比较强，预测值在y = x上下波动，但是数据集中存在一些噪声点，导致实验效果不理想。因此在后期的研究过程中应该将更多的精力花在数据预处理阶段，以达到提高住宅建筑能耗预测的性能。

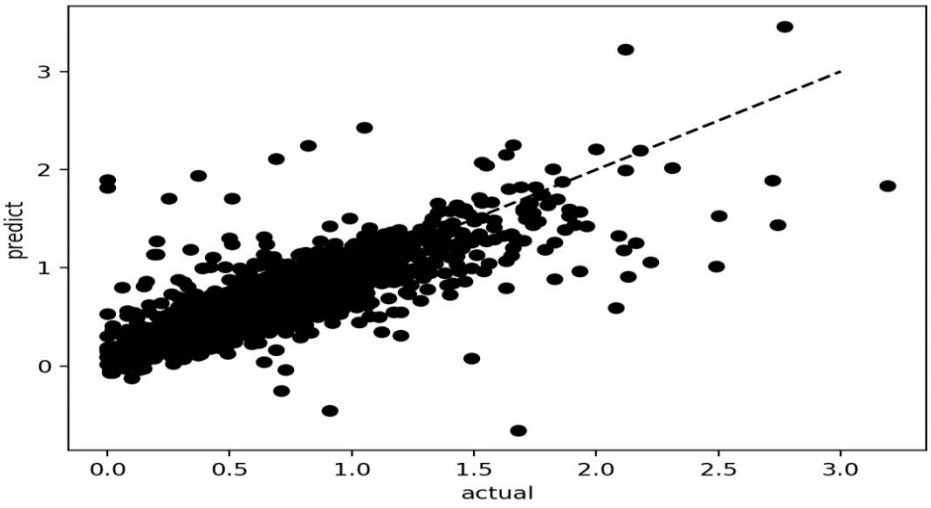


图 3.2 住宅建筑能耗预测图

表 3.5 住宅建筑能耗预测结果

实际值	预测值	实际值	预测值	实际值	预测值	实际值	预测
0.78	0.75	1.76	1.47	1.04	1.13	0.82	0.92
0.63	0.58	0.6	0.61	0.89	0.88	0.26	0.40
0.54	0.52	0.91	0.87	1	1.2	1.55	1.27
0.76	0.76	0.73	0.46	0.76	0.58	0.11	0.53
1.31	1.31	1.07	1.10	0.54	0.55	0.51	0.62
0.45	0.37	0.6	0.53	0.43	0.39	0.47	0.63
0.96	1.05	0.41	0.32	0.96	1.05	0.57	0.65
1.15	0.86	0.41	0.44	1.09	0.89	0.24	0.31
0.88	0.93	0.88	0.93	0.64	0.71	0.96	1.10
0.09	0.18	0.64	0.51	2.12	1.99	1.52	1.71
0.31	0.27	0.26	0.30	0.84	0.89	1.12	1.08
0.3	0.71	0.36	0.40	0.35	0.36	1.52	1.71
0.73	0.66	0.92	0.93	1.15	0.89	0.57	0.65
0.78	0.77	0.85	0.88	0.23	0.36	0.56	0.69
1.03	1.05	0.39	0.31	0.77	0.78	1.13	1.08
0.32	0.13	1.01	0.77	0.3	0.21	0.3	0.34
0.43	0.34	0.27	0.35	0.1	0.17	0.38	0.41
0.58	0.41	0.19	0.21	0.69	0.69	0.96	0.57
0.17	0.22	0.44	0.51	1.27	0.95	0.78	0.82
0.31	0.40	0.62	0.90	0.49	0.32	0.65	0.91
0.59	0.37	0.68	0.76	0.67	0.61	0.62	0.87
0.54	0.53	0.42	0.35	0.67	0.61	0.73	0.88
0.45	0.50	0.8	0.93	0.59	0.34	0.79	0.76
0.4	0.25	0.02	0.07	0.91	0.95	0.12	0.17
0.63	0.64	1.38	1.86	1.03	1.22	0.36	0.35
0.72	0.58	0.85	0.78	0.5	0.097	0.43	0.29
0.85	0.87	0.34	0.26	0.68	0.59	0.94	0.85
0.28	0.24	0.23	0.33	0.81	0.29	0.46	0.62
0.72	0.60	0.15	0.41	1.84	1.36	0.95	0.70
0.37	0.46	0.32	1.21	0.42	0.45	0.52	0.63
0.51	0.51	1.13	1.20	0.51	1.55	0.18	0.26

0.72	0.79	0.33	0.38	0.35	0.42	0.37	0.33
0.58	0.70	0.29	0.33	0.25	0.16	0.42	0.43
0.5	0.52	0.8	0.88	0.72	0.53	0.58	0.54
0.55	0.56	0.51	0.58	0.19	0.35	0.65	0.68

3.4 本章小结

本章针对住宅建筑能耗预测实例求解问题，交代了数据挖掘中特征向量的构成，特征向量中融合了历史能耗信息、建筑物信息和天气信息，此住宅建筑能耗数据具有高维稀疏特性，因此本章采用组合的特征降维方式提取数据中的有效信息、降低回归算法的复杂度。利用数据挖掘技术解决住宅建筑能耗预测问题，实验比较了各个特征工程算法在住宅建筑能耗数据预测场景中性能差异，并且以组合降维方法为特征工程方法，比较不同的回归模型在住宅建筑能耗数据预测问题的算法性能。实验最后选取最优的算法组合解决住宅建筑能耗预测问题，最终实验结果R^2维持在 0.85 左右，实验误差在可接受范围之内。

第4章 住宅建筑能耗等级划分

4.1 引言

本课题在对住宅建筑能耗数据划分等级时，即是采用聚类算法解决能耗数据等级划分问题。本文在分析以上算法的基础上，本章提出了一种改进型 PSO 算法不停的迭代搜索全局最优的聚类中心，然后选取适应度值最小的全局最优粒子解为聚类中心，后续聚类中的初始聚类中心即为初始聚类中心点来获得理想的聚类划分标准，最终引入聚类评价参数评判最后聚类的效果。该算法充分利用 PSO 算法全局搜索能力，从一开始就找出适应度值最小的聚类中心点，以此来避免聚类算法结果对初始点的依赖。在提取聚类中心后，以聚类中心的平均值作为划分能耗等级标准。划分等级之后，统计各个等级中的数据量，引出下一章中采用采样算法解决类不平衡问题。

4.2 Kmeans 算法

聚类算法的划分即是使是 4.1 最小，C=$\{C_1，C_2, C_3, \ldots\ldots C_n\}$使得最小化平方误差

$$\mathrm{E} = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||_2^2 \qquad (4.1)$$

其中$\mu_i = \frac{1}{C_i} \sum_{x \epsilon C_i} x$是簇$C_i$的均值向量，为了是目标韩式式 4.1 达到最小化，Kmeans 采用贪心策略，通过不断迭代优化求解稳定的聚类中心。算法流程表 4.1 所示：

表 4.1 Kmeans 算法步骤

S 输入：样本集D = $\{x_1, x_2, \ldots\ldots x_m\}$,聚类簇数，聚类簇数 k。

过程：

从数据集中随机选取 k 个数据点作为初始聚类中心

Repeat

计算样本点与各个聚类中心的欧式距离，将每个数据点划分到最近的簇中

更新簇均值，重新计算每个簇的平均值做下一次的迭代中心，对中心点变标记

util 中心点不发生变化

输出： 簇划分 C={C_1，C_2，……C_k}

距离公式选择

设 X=$\{x_1, x_2, x_3, \ldots\ldots x_n\}$,为 m 维空间中的一组对象，其中$x_i$，$x_j$为 X 中的两个对象，d($x_i, x_j$)表示$x_i$与$x_j$之间的距离。d($x_i, x_j$)虚满足如下性质：

常用的距离公式如下公式，本文在数据处理和模型建立时，所采用的距离公式都是以欧式距离为标准。

(1)明科夫斯基距离

$$\mathrm{d}(x_i, x_j) = \sqrt{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^{\lambda}} \tag{4.2}$$

(2)曼哈顿距离

$$\mathrm{d}(x_i, x_j) = \sqrt{\sum_{k=1}^{n} |x_{ik} - x_{jk}|} \tag{4.3}$$

(3) 欧式距离

$$\mathrm{d}(x_i, x_j) = \sqrt{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^{2}} \tag{4.4}$$

4.3 聚类评价指标

聚类是典型的无监督学习，大多数聚类算法依赖于某些假设，以便定义数据集中存在的子群，如聚类中心数和初始聚类中心。因此，在大多数应用中，由于无法确定聚类的中心个数，本文引入评价参数评估聚类实验的实验效果。

在评价聚类性能的标准中，类之间的分离程度以及类内的紧凑成是作为评价聚类性能的两个主要参数。然而算法的目标是基于初始阶段假设（聚类初始中心）或者输入参数值（聚类的数目、最小直径或者类中的点的个数）。如 DBSCAN[62] 的聚类指标是基于密度变化定义集群，该算法考虑的是类的基数和半径值。Kmeans 聚类的结果取决于聚类个数设定以及初始聚类中心选取。结合以上两个标准的使用，本文采用 S_dbw 作为衡量聚类性能的参数。S_dbw 不仅考虑了类内的紧凑度，而且参考了两个簇之间的密度问题。

将数据集 S 划分到 D=$\{V_i | i = 1,2,3 \ldots\ldots c\}$类，其中$V_i$作为各个类的中心

$$\mathrm{stdev} = \frac{1}{c}\sqrt{\sum_{i=1}^{c} ||\delta_i||} \tag{4.5}$$

stdev 作为簇中心距离远点的平均距离,c 为聚类个数。为了评估在数据集中的平均分布密度与类中的密度之间的关系，引入$\mathrm{Dens_{bw(c)}}$:

$$\mathrm{Dens_{bw(c)}} = \frac{1}{c(c-1)}\sum_{i=1}^{c}\sum_{\substack{i=1 \\ i\neq j}}^{c}\frac{density(u_{ij})}{\max(density(v_i),\mathrm{density}(v_j))} \tag{4.6}$$

v_i, v_j 为c_i, c_j的中心。u_{ij}为c_j, c_i的中心点。密度函数定义为：

$$\mathrm{density(u)}=\sum_{i=1}^{n_{ij}} f(x_i, \mathrm{u}),\ x_i, \in\ c_i \cup c_j \subseteq S \tag{4.7}$$

n_{ij}为c_i, c_j两个类中的点。f(x,u)定义为：

$$f(x,u)=\begin{cases}0 & \text{if } d(x,u)> stedv\\ 1\end{cases} \quad (4.8)$$

如上述的公式可以表明，如果一个点与 u 的距离小于集群的平均标准差，那么就属于 u 的邻域。鉴于某些聚类算法需要限定聚类的个数，引入 Scat(c)参数，此参数是用来定义类的平均散射程度。

$$Scat(c)=\frac{1}{c}\sum_{i=1}^{c}||\delta(v_i)||/||\delta(S)|| \quad (4.9)$$

$\delta(S)$为数据集的方差，它的第 p 维方差定义为：

$$\delta_x^p=\frac{1}{n}\sum_{k=1}^{n}(x_k^p-\bar{x}^p)^2 \quad (4.10)$$

其中$\bar{x}^p$ 是第 p 维的平均值，定义为：

$$\bar{X}=\frac{1}{n}\sum_{k=1}^{n}x_k\ \forall x_k\in S \quad (4.11)$$

$\delta(v_i)$为类c_i的方差并且第 p 维的方差定义为：

$$\delta_{v_i}^p=\sum_{k=l}^{n_i}(x_k^p-v_i^p)^2/n_i \quad (4.12)$$

参数 S_dbw 定义如下：

$$S\_dbw(c)=Scat(c)+Dens\_bw(c); \quad (4.13)$$

因此，S_dbw 参数能从两个方面评估聚类效果，一个是类内的紧密程度，另一方面是类之间的分离程度。因此聚类的效果应该是在类之间比较紧密，类之间较分散。通过 S_dbw 参数数值反映，应该是该适应度值越小，聚类性能越佳，越能体现数据的分布规律。本文基于 S_DBW 参数评价能耗等级分级效果并且以此为评价指标比较优化算法性能。。

4.4 PSO 算法

粒子群算法(Particle Swarm Optimization ,PSO) 算法是 Kennedy 和 Eberhart 提出一种模拟鸟群觅食过程中群体行为的新型群体算法。本文利用此算法优化解决聚类中心的随机初始化问题。PSO 算法的构成是根据整个粒子群众的最优位置和单个粒子轨迹中的最优位置来改变速度和位置向全局最优解的方向飞行，其中最重要的是引入一个适应度函数对整个粒子群的性能进行评价。假设种群规模为 m 的粒子群在 n 维空间搜索时，若目前为止的个体最优位置为$pbest_i$,群体最优位置为 Gbest，则每个粒子的飞行速度V_i和位置X_i 可以根据下面的式子进行调整：

$$V_i(t+1)=w*V_i(t)+c_1*r_1*(pbest_i-X_i)+c_2*r_2*(Gbest-X_i) \quad (4.14)$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (4.15)$$

其中，t 是表示迭代次数的变量；w 称为惯性权重系数；系数c_1, c_2为学习因子；r_1, r_2均匀分布在[0,1]范围内的随机数。

4.5 基于改进 PSO 的 Kmeans 算法

聚类场景中的 Kmeans 算法具有收敛速度快、算法稳定等特点，然而聚类的过程中也存在这聚类初始中心点随机化选取和无法确定聚类中心个数的问题，因此算法的不确定性比较大；另一方面，聚类算法没有一个很好的指标衡量最后的聚类效果，因此本文引入适应度值作为最后聚类个数选取的评价指标。为了解决聚类中心随机初始化问题和对初始中心敏感的特点，本文利用 PSO 强大的全局搜索能力，但是优化算法在迭代后期存在着搜索速度非常缓慢，适应度值趋于稳定，也就是常说的粒子群早熟问题。针以上两个问题,本文引入 PSO 和聚类效果评价指标提高聚类效果和决定最终的聚类个数。从另一方面，本文也对优化算法进行改进，提高优化算法的搜索能力。算法根据迭代次数动态调整粒子的惯性参数设定，当前的迭代次数作为惯性权重的影响因子，根据迭代次数动态的调节权重因子，增强 PSO 算法中粒子群的全局搜索性能；在使用优化算法时，很容易出现局部极值为题，本文引入适应度的方差阈值解决粒子群的早熟问题，提前结束粒子群的迭代。该算法充分利用了 PSO 的全局搜索能力解决 Kmeans 聚类中心随机初始化问题以及选择合适聚类中心数。

动态调整惯性权重

为了增强算法的全局搜索能力，本文对 PSO 算法参数结构进行了调整。前期由于数据量少，为了加快迭代效果，因此前期 PSO 算法中的 w 值应该比较大来加强全局搜索能力而后期整个粒子群的适应度值趋于稳定，因此搜索能力变差，减小权重来加强粒子群的局部搜索能力。本文使用的算 w 可采用以下线性调整策略:

$$\mathrm{w(t)} = w_{max} - (w_{max} - w_{min})\mathrm{t}/iter_{max} \qquad (4.16)$$

其中$iter_{max}$为粒子群的最大迭代次数。w_{max}为最大惯性权重，w_{min}为最小惯性权重，t 为当前迭代次数。则粒子位置的调整公式可改为：

$$X_i(t+1) = X_i(\mathrm{t}) + H_0(1 - t/t_{max})V_i(t+1) \qquad (4.17)$$

因为 kmeans 聚类是取类中每个维度的中心值作为聚类的中心点，限定位置边界$psositon_{max}$、$psosition_{min}$为数据集中每个维度的最大值和最小值。当$V_{max} > 0.4psositon_{max}$,$V_{max}$更新为$psosition_{max}/2$。当$position_i$>$psosition_{max}$,或者$position_i$<$psosition_{min}$,更新为数据的平均值。

粒子适应度函数

粒子群中的每个粒子都是代表了空间中一个可能解，即后期聚类算法中初始聚类中心。若数据特征是 q 维向量，则每个粒子的位置和速度都是 q*k 维向量。最终粒子群找到的适应度函数最小的粒子的向量表示即是聚类中心的初始中心点。

本文引入值评估在粒子群集之间的区域内的平均密度与群集密度的关系。目标是聚类中的密度与所考虑的聚类中的密度相比都较低。本文所采用的适应度值为 S_dbw。粒子群在迭代次数内找到适应度值最小的解，以此解作为后期聚类算法的初始聚类中心。

加入优化算法的聚类算法是寻找到初始适应度值最小的聚类中心点。为了防止优化时陷入局部极值，本文计算粒子群方差解决此问题。当整个粒子的方法区域稳定时，即是陷入局部最优解了。整个粒子群的适应度值趋于稳定，方差也会趋于变小也即陷入局部最优解中。群体适应度方差δ^2可定义为：

$$\delta^2 = \frac{1}{m}\sum_{i=1}^{m}[f(x_i) - f_{avg}]^2 \tag{4.18}$$

其中 m 为粒子群中粒子个数，$f(x_i)$为单个粒子的适应度值；f_{avg}为粒子群的适应度均值。当粒子群的适应度方差趋于稳定时，表明已经进入局部极值正，则终止执行 PSO 算法，转而执行 Kmeans 算法。

表 4.2 优化算法步骤

输入：聚类数据集 T，数据集聚类中心数 k,粒子群规模 m,粒子群最大迭代次数t_{max}
输出：稳定的 k 个聚类中心。
初始化操作，数据集中选取 k 个初始中心。初始化速度、全局最优位置和粒子的最优位置和全局适应度极值。
Repeat
动态调整惯性权重，当前位置和速度与边界速度和位置比较，更新粒子位置
依照欧式距离将每个数据点划分到最近的簇中
若粒子适应度值小于个体极值，更新个体极值和粒子最优轨迹位置和判断是否为全局最优位置，否则迭代次数加 1；
Util 群体适应度方差小于设定阈值或者迭代次数达到了最大迭代次数。否则继续执行
执行 kmeans 算法。

4.6 实验及结果分析

本文收集 UCI 开放的数据集，这些数据集一般用作分类测试数据集，实验

以正确率和适应度值为评价聚类效果指标，比较原始 Kmeans 算法和 PSO-Kmeans 算法的性能差异。数据集信息如下表 4.3 所示，target 代表样本中类的个数。

表 4.3 测试集样本

数据集	样本数量	维度(特征数)	Target
Abalone	4177	8	3
Iris	150	4	3
Wine	178	13	3
breastcancer	568	30	2

为了从各个维度比较优化算法和原始算法的实验性能，本文引入聚类正确率、适应度值和迭代次数作为比较算法性能参数。实验结果如下表 4.4 所示。

表 4.4 聚类实验结果

数据集	算法	最高正确率	Fitness	平均迭代次数	平均正确率
Breastcancer	Kmeans	0.905	0.2	7.65	0.905
	PSO-Kmeans	0.928	0.189	5.25	0.928
Iris	Kmeans	0.887	0.079	7.7	0.84
	PSO-Kmeans	0.887	0.079	6.55	0.867
Wine	Kmeans	0.955	0.173	6.65	0.943
	PSO-Kmeans	0.966	0.1723	9.95	0.946
abalone	Kmeans	0.514	0.109	41.3	0.513
	PSO-Kmeans	0.538	0.0756	15.25	0.527

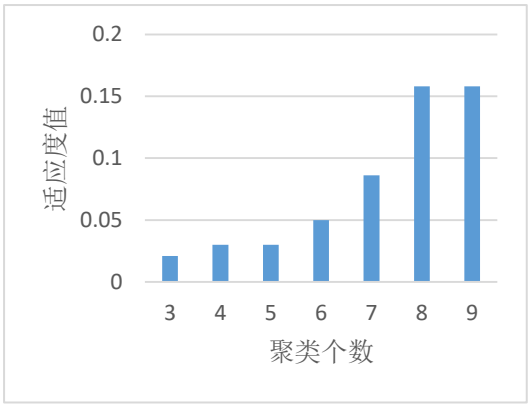
聚类算法的聚类效果极其依赖初始聚类中心的选择和聚类中心的个数选择，针对这一问题本文加入了 PSO 优化算法的聚类算法作为聚类算法的前一步骤选择全局适应度值较小点作为聚类初始中心点。根据表 4.4 结果表明加入了优化算法的聚类算法比未加入优化算法的适应度值小，说明加入优化算法的聚类效果较好，更能将具有相似特性的点合成一簇。在正确率方面，优化算法对正确率有小部分提升，大约提高了 2 个百分点左右；另一方面，引入优化算法可以减少聚类算法的迭代次数。数据维度较高导致迭代次数越多，算法运行时间越长。实验结果表明优化算法的迭代次数明显减少，算法计算性能得到提升。该优化算法的不足之处就是加入的 PSO 优化算法中的粒子运算的迭代次数较多，运行时间较长，虽然加少了 kmeans 的得带次数，但是此种迭代次数的降低时以离子群算法中大量的迭代为代价的。

4.7 住宅建筑能耗等级划分

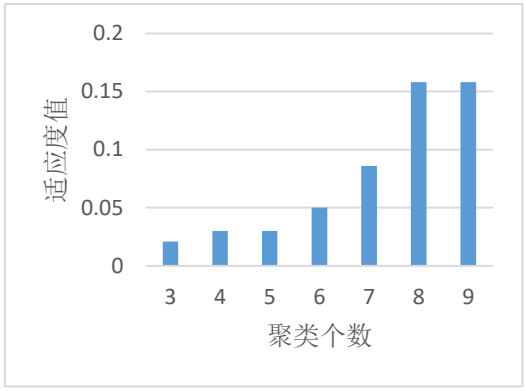
国家已经对电价进行分级收费，但是在分级的过程中缺少实验数据支撑。本文提出对住宅建筑能耗护划分能耗等级讨论。现如今很多学者采用的统计学方式对住宅建筑能耗数据进行讨论，统计结果无法令人信服，不能很好地描述数据的分布规律和维护数据的整体性。本文采用无监督学习算法，利用其强大的数据分析能力分析住宅能耗数据的分布规律。如果对全年的数据进行整体分析，以一年的能耗数据组成特征向量进行整体分析，不能反映能耗的季节性变化。因此，本实验的能耗等级划分以季度为划分单位，提取每个季度中三个月的能耗值组成特征向量，特征向量在做图像显示的时候也是可以作为住宅建筑这个季度的坐标点。如第一季度的组成向量为{一月能耗，二月能耗，三月能耗}。本文首先利用聚类算法提取聚类中心，以聚类中心的平均值该季度划分能耗等级。因为聚类属于无监督学习算法，聚类的个数无法事先确定。本文引入 S_dbw 适应度值作为评价聚类效果参数，适应度值越小，说明聚类的效果越好，簇内连接越紧密，簇之间分离程度越大。由于传统的聚类算法存在聚类中心初始随机化的原因，本文引入 PSO 优化算法解决原始聚类中心初始随机的问题。由于上文已经证明了优化算法在一些场景中的聚类效果优于原始聚类算法，因为本文采用优化的 PSO-Kmeans 算法对住宅建筑能耗数据进行聚类分析，根据聚类的个数提取聚类中心，以聚类中心的平均值作为划分能耗等级的参考点。实验统计结果如表 4.5 所示以适应度值为评价参数；并且绘制图显示不同聚类个数的适应度值，如图 4.1 所示，表中的适应度值就是以 S_dbw 为评价标准。

表 4.5 聚类结果

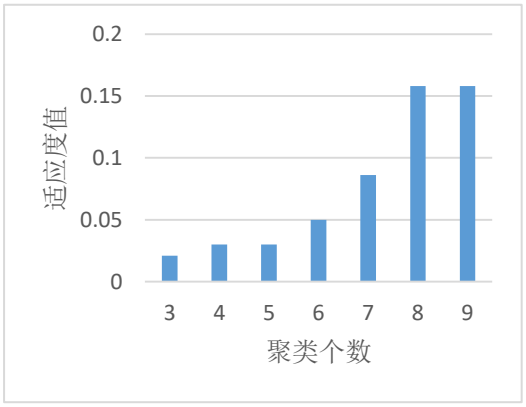
聚类个数	第一季度适应度值	第二季度适应度值	第三季度适应度值	第四季度适应度值
3	0.416	0.274	0.123	0.021
4	0.394	0.397	0.153	0.030
5	0.472	0.679	0.153	0.030
6	0.444	1.130	3.020	0.050
7	0.529	2.013	2.582	0.086
8	0.546	2.391	2.336	0.158
9	0.514	7.272	8.171	0.158



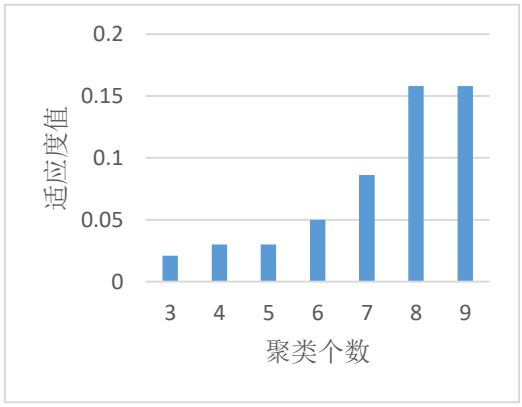
(a) 第一季度聚类统计图



(b) 第二季度聚类统计图



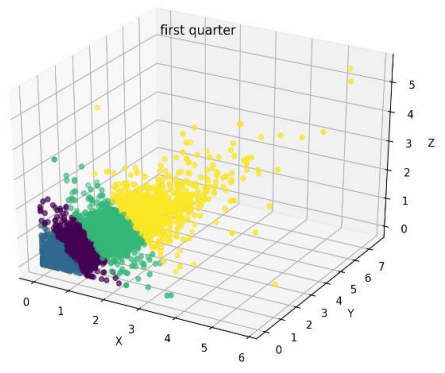
(c) 第三季度聚类统计图



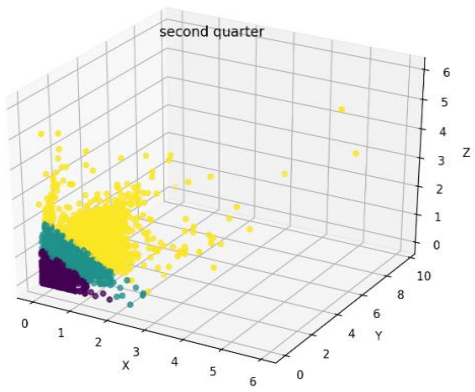
(d) 第四季度聚类统计图

图 4.1 住宅数据聚类统计图

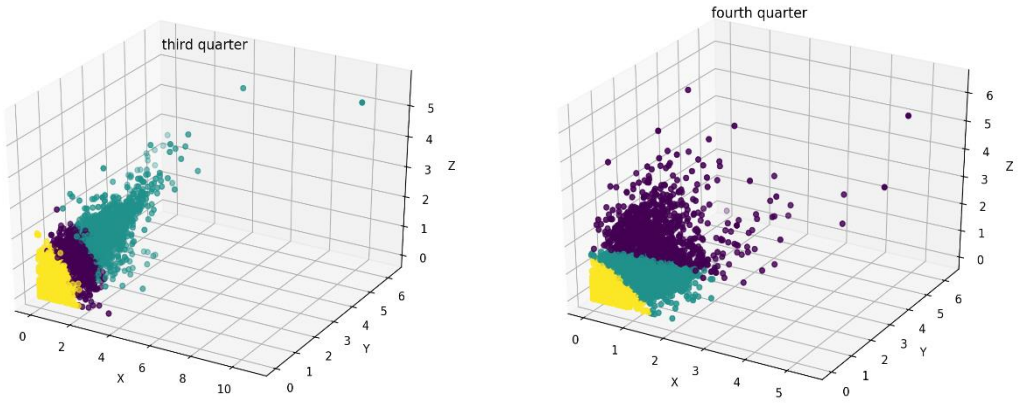
当适应度值最小时的聚类个数应为最终聚类个数。如表 4.5 所示，第一季度聚成四类、第二季度、第三季度、第四季度聚为三类，聚类统计结果如 4.1 图所示。则第一季度可分为五个等级，第二，三四季度可以分为四个等级。如图 4.2 所示为各个季度的聚类图，坐标点为季度中三个月的能耗值组成向量,以第一季度为例，特征向量的构成既是 1、2、3 月份的能耗值组成特征向量，此向量就是代表该住宅建筑的点坐标，图中不同颜色代表不同的簇。图中的 x、y、z 坐标分别代表此季度中三个月的能耗值，所有坐标轴的单位都是(w/m^2)。



(a) 第一季度聚类图



(b) 第二季度聚类图



(c) 第三季度聚类图　　(d) 第四季度聚类图

图 4.2 聚类效果图

根据计算最优适应度值，计算聚类中心点，根据中心点的平均值划分每个住宅建筑每月的能耗等级，每个等级中点的个数统计如下表 4.6 所示。

表 4.6 各个季度等级统计结果

季度	0	1	2	3	4
第一季度	3901	5582	1256	248	16
第二季度	4999	7259	3583	719	0
第三季度	3104	7738	4795	923	0
第四季度	6015	8101	2271	173	0

如表 4.6 所示 第一季度可以分为低、略低、 中等，略高，高五个等级。第二、三、四季度分为低、中等、略高、高四个等级。根据聚类中心的平均值将每个月的能耗等级进行划分。如上表所示为每个季度中，每个能耗等级统计结果。统计结果表明每个季度中的等级存在类不平衡问题分布，在不平衡分布的数据集中训练数据会影响结果的准确性。

4.8 本章小结

本章首先对数据挖掘中的 Kmeans 聚类算法进行了分析以及在使用过程中存在初始中心随机初始化和聚类中心个数无法去顶的问题。本文引入了聚类评价指标作为适应度值，提升聚类算法的性能。设计实验验证优化聚类算法的优越性，达到了聚类算法的性能要求。在本章中，采用优化的聚类算法对住宅建筑能耗数据提取中心，以中心点的平均值作为划分能耗等级的标准，并且统计了各个季度中不同能耗等级的个数，由此引出下一章将采用的采样算法解决不平衡分类问题。

第5章 住宅能耗等级预测

5.1 引言

本章主要解决的问题是建筑能耗等级的预测，第四章中已经将住宅建筑能耗等级进行了划分，此划分是基于佛罗里达州的建筑群，等级的划分都是以建筑能耗数据为划分点。在前文的叙述中，能耗等级的划分后，存在数据不平衡问题。在使用数据挖掘技术处理分类问题时，默认数据集是平衡的，可不平衡的数据集会影响最后分类器的实验效果。本文的主要内容是采用采样算法解决类不平衡以提高等级预测的准确性。在特征工程处理时，提出组合算法解决特征数据中的噪音和维数灾难问题，并且在高维稀疏数据集验证算法性能。

5.2 数据预处理

数据预处理阶段，输入的待处理数据一般为带有噪声和异常值的数据集。数据预处理阶段需要采用一定的数据处理手段，将所有类型的数据集中的特征量纲划分到同一标准。特征工程、归一化为数据预处理阶段的主要步骤。特征选择是基于标签计算每个特征的相关系数，选取相关系数最大的某些特征。由于特征选择是基于全局计算特征的关联度，因此特征选择只能是分类问题中使用，本章的特种工程引入随机森林算法挑选信息量最大的特征。特征向量依然延续第三章所述，以建筑信息、天气信息和历史能耗信息作为特征向量，但是以划分的住宅建筑能耗等级为输出，以此预测下一个月的能耗等级。如图 5.1 为本次住宅建筑简要流程图。

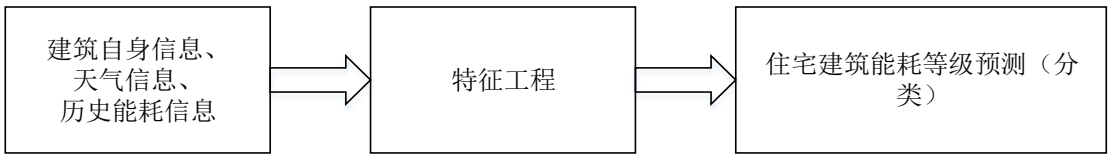


图 5.1 住宅建筑能耗等级预测流程图

5.3 特征工程算法试验比较

5.3.1 实验概述

用 PCA、SVD 和 RF 进行几种特征工程算法的对比实验。实验数据集选取 5

个 UCI 标准数据集，均具有高维度稀疏特性。本文基于 SVM 算法，将 PCA、SVD 和 RF 相结合的组合特征工程算法，将此种算法和三个单个特征工程算法以 ROC 曲线的面积值作为评价标注比较算法性能。

5.3.2 实验评价标准

为了对预测分类结果进行统计，一般的分类结果采用混淆矩阵（Confusion Matrix）作为描述。该表很好的反映了分类结果的各种情形，并对不同的预测性情作出了区分。在典型的二分类情形中，是一个 2*2 的矩阵。划分结果的统计表如表 5.1 所示，第一列是“positive”的个数，第二列是预测为“negative”的个数。

表 5.1 混淆矩阵

	Positive	Negative
Positive	True Positive（TP）	False Negative（FN）
Negative	False Positive（FP）	True Negative（TN）

如表 5.2 所示，分类可以将结果分为四类：

表 5.2 混淆矩阵注解

真正（True Positive, TP）	被模型分类正确的正类样本
假负（False Negative, FN）	被模型分类错误的负类样本
假正（False Positive， FP）	被模型分类错误的正类样本
真负（True Negative， TN）	被模型分类正确的负类样本

进一步可以推出表 5.3 中的指标。

表 5.3 TPR,FNR,FPPR,TNR 表达式

真正率（True Positive Rate, TPR）	正样本被预测正确的比例 $\mathrm{TPR}=\frac{TP}{TP+FN}$
假负率（False Negative Rate, FNR）	正样本被预测错误的比例 $\mathrm{FNR}=\frac{FN}{TP+FN}$
假正率（False Positive Rate, FPR）	负样本被预测错误的比例 $\mathrm{FPR}=\frac{FP}{FP+TN}$
真负率（True Negative Rate, TNR）	负样本呗预测正确的样本比例 $\mathrm{TNR}=\frac{TN}{FP+TN}$

进一步，由混淆矩阵可以计算如下评价指标，如表 5.4 所示,

表 5.4 评价标准公式

准确率（Accuracy）	预测正确的样本比例 $\text{accuracy} = \frac{TP+TN}{TP+FN+FP+TN}$
精确率（Precision）	正样本被预测正确的比例 $\text{Precision} = \frac{TP}{TP+FP}$
召回率（Recall）	正样本被预测为正样本的比例 $\text{Recall} = \frac{TP}{TP+FN}$
ROC 曲线	ROC 曲线的 x 轴便是 FPR，y 轴便是 TPR

用于不平衡分类评价的指标定义如下：

F-measure

在实验中，在需要得到不同的精确率和召回率时，可以通过改变参数值或者核函数。与 ROC 曲线一样，实验中也需要一个定量的指标对混淆矩阵中的参数进行联合比较。其中，F-measure 为精确率和召回率的调和平均值，将两个指标联合起来，使评价标准更具鲁棒性。

$$\text{F}-\text{measure} = \frac{2*precision*recall}{presicion+recall} \tag{5.1}$$

F-measure 被广泛应用在信息检索领域，用来衡量检索分类和文本分类的性能。F-measure 也被广泛应用在更深层次的自然语言处理领域，比如命名实体识别、分词等，用来衡量算法或系统的性能。本文中，F-measure 用来评估类别不平衡分类问题中的算法效率。

ROC

ROC 曲线是指描述二分类问题效果的一种参数，即将样本预测为正类(positive)和负类(negative)。在一个二分类问题中，一个预测问题主要会被划分为四种类型情况，将正类准确划分为正类，即为真正类；反之，如果样本为负类而被划分为正类，则称之为假真正类；如果负类被预测称为负类，则是真负类，正类呗预测为负类则为假负类。以上四种情形即为预测常见的几种类型。

ROC 曲线与坐标轴所围成的面积越大，则实验的准确率越高。所以，ROC 曲线经常被用来作为类别不平衡问题中的评价指标。AUC 参数则为 ROC 曲线在坐标轴上覆盖的面积，面积值越大则最终的实验结果越好，接下来比较特征工程算法的实验性能统计即是以 AUC 面积值为主要性能指标比较各个算法在住宅建筑能耗数据集上的实验效果。

5.3.3 特征工程算法实验比较

本文的能耗等级预测属于不平衡的多分类问题，但是多分类问题可以转化成多个二分类问题，可以跨多个分类计算每个二分类的矩阵得分的均值。本文采用的所有指标都是基于不平衡数据集的评价指标。在机器学习、数据挖掘任务中，经常会使用基于 python 的 sklearn 包，sklearn 包中所有基于不平衡分类任务的参数评价指标都会根据每个类的分布加权重求取平均值。

本实验选用 UCI 公开的具有高维度、稀疏性的数据集为算法的测试集，数据可通过 http://archive.ics.uci.edu/ml/下载，数据特性如下表 5.5 所示

表 5.5 测试数据集

数据集	样本数量	维度(特征数)
Madelon	2600	500
Arcene	200	10000
Gisette	1500	5000
Dorothea	280	5495
Derex	300	659

因为 SVM 在分类问题中有很好的实验性能，因此本实验基于 SVM 算法，将 PCA、SVD、RF 和 RF+PCA+SVD 四种算法作为特征工程方法进行实验仿真。本文的数据预处理步骤全都采用归一化解决不同维度之间的量纲问题，采用十字交叉的方式，事先预留出一份的测试集，以测试集的实验结果作为最后的统计数据。每个特征工程算法都运行 20 次，结果取 AUC 的平均值（M）和方差（V），数据从两个方面分析各个算法在不同数据集上的实验性能以及实验结果如下表 5.6 所示。

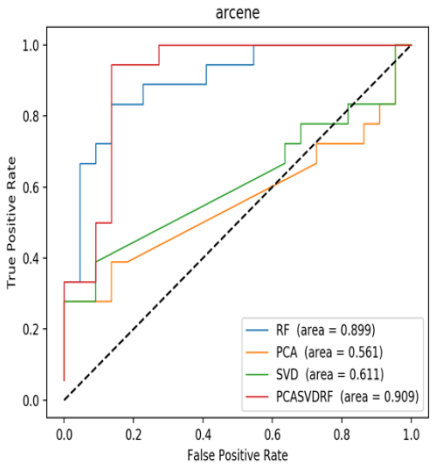
表 5.6 特征工程算法比较结果

数据集	评价标准	RF	PCA	SVD	RFPCASVD
Arcene	M	0.915	0.353	0.355	0.910
	V	0.029	0.030	0.0315	0.029
Madelon	M	0.627	0.628	0.626	0.804
	V	0.013	0.014	0.11	0.016
Gisette	M	0.984	0.984	0.985	0.983
	V	0.008	0.007	0.007	0.007
Dorthea	M	0.608	0.593	0.606	0.717
	V	0.102	0.097	0.094	0.097

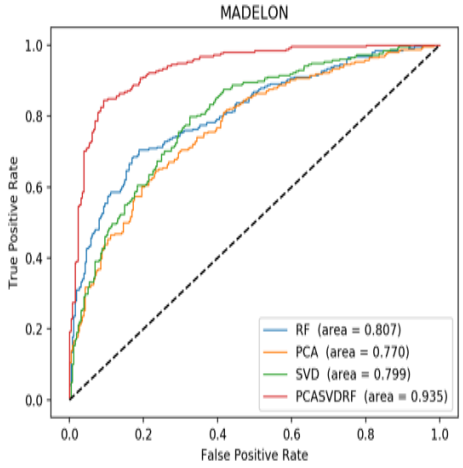
Derex	M	0.42	0.44	0.44	0.754
	V	0.11	0.10	0.11	0.117

上表的实验统计结果表明，以 SVM 为分类框架，组合特征工程算法，有更好的实验性能和更好的实验稳定性。

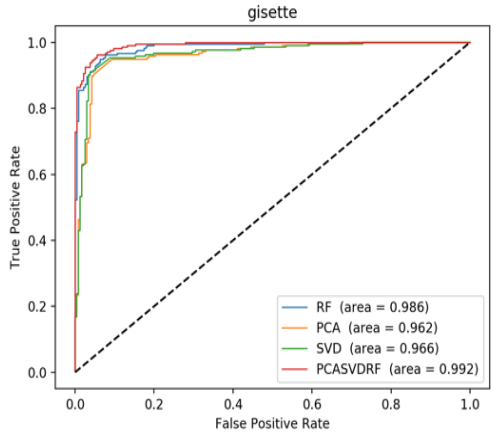
本实验采用对各个数据集中预留一部分数据做测试数据，根据测试集绘制四种特征工程算法的 AUC 曲线，正如上文所述，ROC 的面积值越大说明准确性越高。对各个数据集绘制 AUC 曲线如下图 5.2 所示。



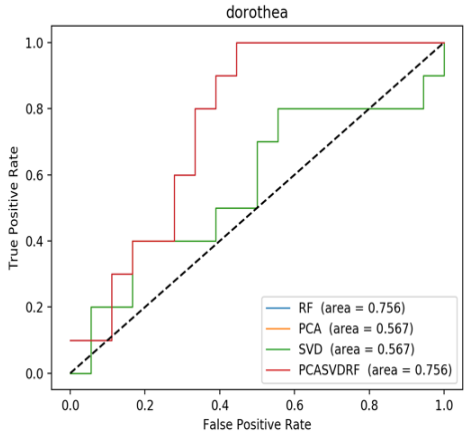
(a) Arcene



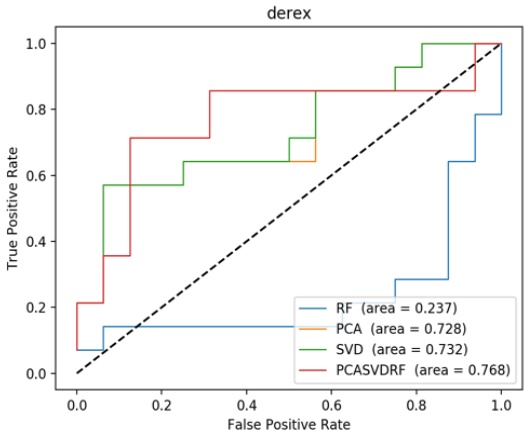
(b) madelon



(c) gisette



(d) dorothea



(e) derex

图 5.2 特征工程算法结果对比

本文收集了 UCI 中的五组数据集，通过调节参数得到了各组数据集的 ROC 曲线，如上图 3.3 所示，基于 RF+PCA+SVD 的 ROC 曲线面积明显大于采用 PCA、SVD 和 RF 特征工程算法所围成的面积。实验数据表明 RF+PCA+SVD 的准确率比其他三个算法的准确率要高，实验效果更好，算法性能提高。

本实验为了从各个维度比较几种特征工程算法的特点，本文引入复杂度实验比较各个算法的 CPU 运行时间，实验对各个数据集运行 50 次取平均值，实验结果如下表 5.7 所示。

表 5.7 CPU 时间比较（单位，秒）

数据集	RF	PCA	SVD	RFPCASVD
Arcene	0.1214607	0.01022	0.01052	0.11806
Madelon	0.5128251	0.48866	0.49272492	0.814553
Dorothea	0.11482364	0.0173337	0.01510	0.1200479
Gisette	0.34439982	0.2241	0.21987	0.3857
Derex	10.8487	8.2965	8.9507	12.005

在对四个特征工程算法的 CPU 运行时间实验比较发现，在分类实验中 RFPCASVD 组合特征工程降维在运行时间上比其他三种单独运行的算法所需要的时间更长。在所花费的时间 RF 所占的比例比较大，也即意味着算法复杂度更高。

5.3.4 住宅能耗数据实例

由于上文的实验数据表明，SVM 在不经过处理的住宅建筑能耗数据上具有

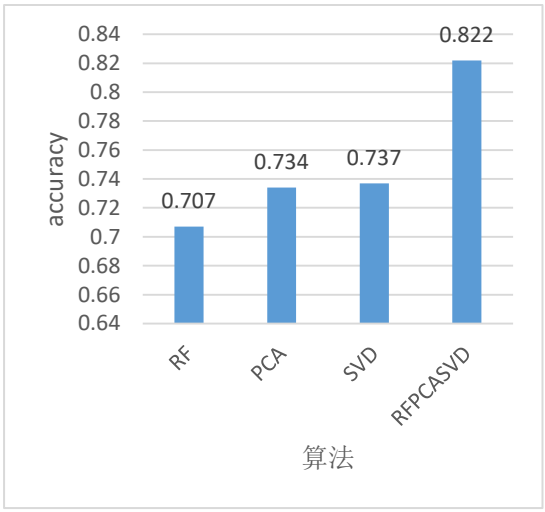
良好的而算法性能，因此本实验基于 SVM 算法，研究 PCA、SVD、RF 以及组合算法(RD+PCA+SVD)在住宅建筑能耗数据上的性能，提出基于三种降维算法的 RF+P+S-SVM 的组合特征工程方法。并对得出的结果进行分析和总结。实验结果如下表 5.8 所示，由于使用的住宅建筑能耗等级分类是多典型的分类场景，所以不能绘制 AUC 曲线评价标准，而是使用 Accuracy、Precision、Recall 和 F-measure 参数作为评价指标，并且在 20 次实验中取平均值以及最大值作为评价指标,这样能从多个维度比较算法的性能。

表 5.8 特征工程算法实验结果

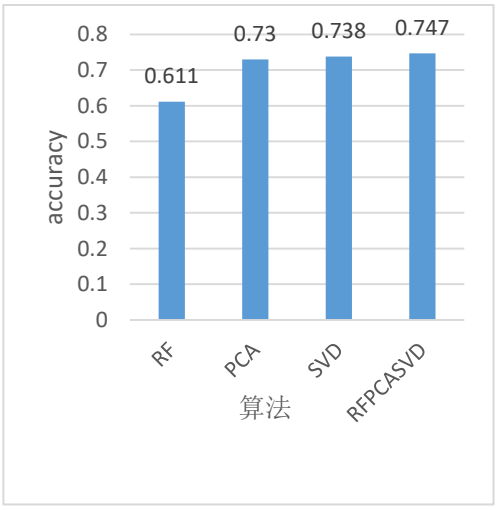
	评价指标		RF	PCA	SVD	RFPCASVD
第一季度	Accuracy	Mean	0.707	0.734	0.737	0.822
		Max	0.737	0.748	0.747	0.822
	Precision	Mean	0.673	0.736	0.74	0.818
		Max	0.708	0.75	0.75	0.818
	Recall	Mean	0.707	0.734	0.737	0.822
		Max	0.737	0.748	0.747	0.822
	F-measure	Mean	0.66	0.731	0.734	0.818
		Max	0.694	0.745	0.744	0.818
第二季度	Accuracy	Mean	0.611	0.73	0.738	0.747
		Max	0.627	0.749	0.74	0.759
	Precision	Mean	0.594	0.739	0.741	0.749
		Max	0.609	0.75	0.757	0.761
	Recall	Mean	0.611	0.736	0.738	0.747
		Max	0.627	0.749	0.753	0.759
	F-measure	Mean	0.593	0.733	0.736	0.746
		Max	0.608	0.747	0.75	0.758
第三季度	Accuracy	Mean	0.629	0.722	0.739	0.774
		Max	0.655	0.722	0.752	0.790
	Precision	Mean	0.608	0.730	0.742	0.775
		Max	0.636	0.730	0.753	0.792
	Recall	Mean	0.629	0.722	0.739	0.774
		Max	0.655	0.722	0.752	0.790
	F-measure	Mean	0.586	0.714	0.729	0.772
		Max	0.611	0.714	0.743	0.788

第四季度	Accuracy	Mean	0.739	0.767	0.77	0.777
		Max	0.752	0.767	0.78	0.784
	Precision	Mean	0.734	0.765	0.77	0.774
		Max	0.745	0.765	0.78	0.783
	Recall	Mean	0.740	0.768	0.776	0.777
		Max	0.751	0.768	0.88	0.784
	F-measure	Mean	0.723	0.761	0.769	0.773
		Max	0.738	0.761	0.781	0.780

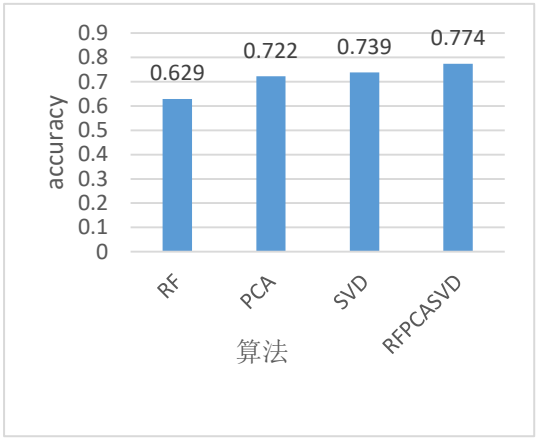
如图 5.3 所示，选取 Accuracy 作为评价指标，绘制每个季度的准确率图，图 5.3 统计结果表明采用 RFPCASVD 组合特征工程算法的准确率明显高于单个特征工程算法，虽然在某些季度上的特征工程算法的性能只比单个算法的性能稍高一点，但是并不妨碍算法的整体性能指标。所以接下来的实验都是以组合算法为特征工程算法为采样算法的特征工程算法以提高住宅建筑能耗等级预测的准确性。在表 5.8 中，从各个维度比较特征工程算法在住宅建筑能耗数据集上的等级预测性能指标。实验的统计结果表明，组合特征工程算法在住宅建筑能耗数据上的能耗等级预测有很好的实验效果。



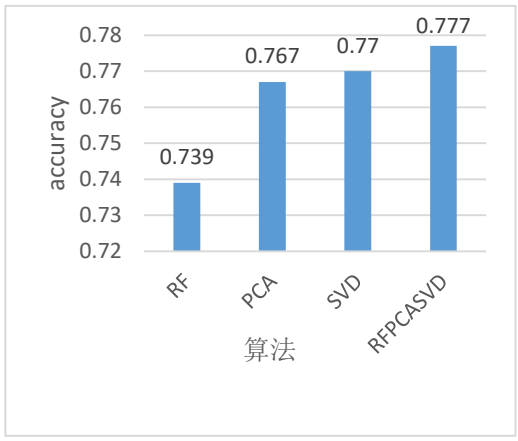
(a) 第一季度特征算法结果图



(b) 第二季度特征算法结果图



(c) 第三季度特征算法结果图



(d) 第四季度特征算法结果图

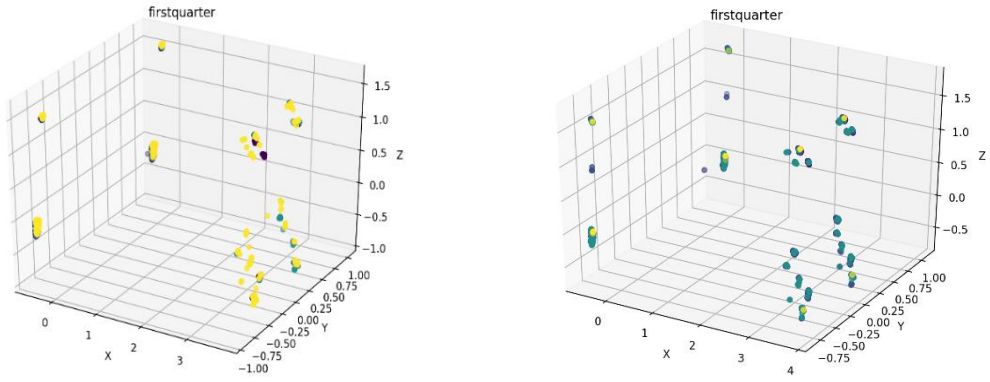
图 5.3 住宅建筑数据特征工程算法比较图

5.4 住宅建筑能耗等级预测

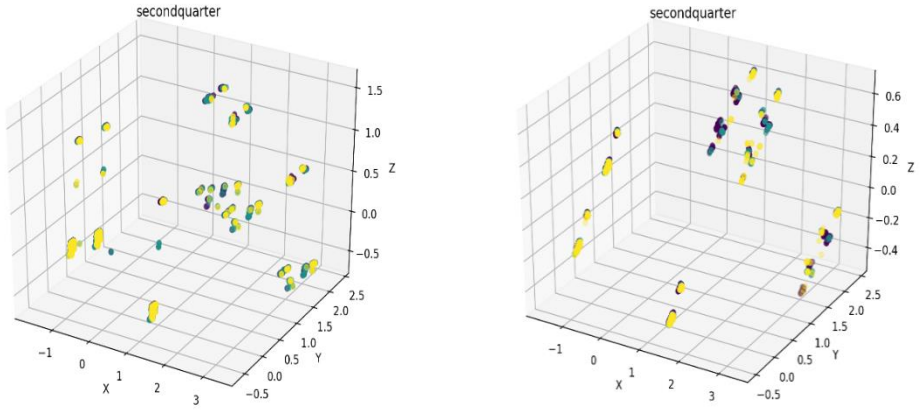
5.4.1 采样算法实验结果

在数据挖掘领域，分类场景中普遍存在类不平衡情况，少数类中通常包含重要信息。不平衡的数据集中的少数类会被错分到多类中，会降低分类准确性。本文的住宅建筑能耗数据已经在第四章中采用优化算法进行等级划分，并且在能耗数据等级划分中出现了类不平衡问题。

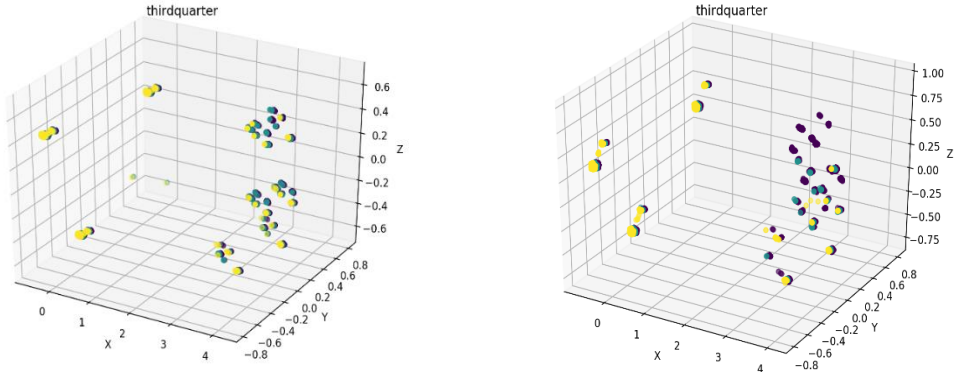
本文的能耗等级分类也存在类不平衡问题。过采样和欠采样是处理数据不平衡问题的常用方法。本文采用据聚类算法解决了住宅建筑能耗的等级划分问题，但是根据上文的介绍，类不平衡问题呼之欲出。本文采用过采样和欠采样方法结合处理能耗数据不平衡问题。分别采用 SMOTE 算法和 ENN 算法对不平衡类采样。为了便于展示采样后点，原始数据集合和采样后的数据均利用 PCA 降维到三维以便于展示效果(坐标轴没有任何物理意义)。实验结果如下图 5.4 所示。



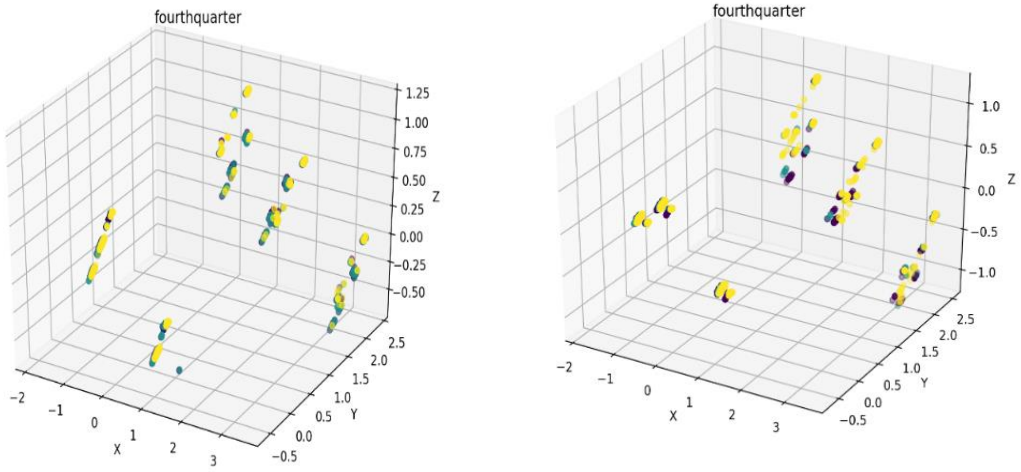
(a) 第一季度采样对比图



(b) 第二季度采样对比图



(c) 第三季度采样对比图



(d) 第四季度采样对比图

图 5.4 采样结果图，

如上图 5.4 所示，左侧图为原图，右侧图为经过算法处理后的采样图。本文对能耗等级数据使用采样算法，左侧图为原始数据，右侧图为采样后的数据。采样结果表明多类数据进行了欠采样（删除了少量点），对少类数据进行了重采样（新增了数据点），此种算法可以平衡多类和少类中数据集个数，以提高后续能耗等级预测结果。

5.4.2 住宅建筑能耗等级预测实验结果及分析

策略 1 针对能耗分类的流程，建立了一套以 PCA+SVD+RF 的组合特征工程方法，以 GBDT、随机森林等算法作为比较算法。根据仿真结果，针对能耗数据比较各个算法性能。研究采样算法在能耗数据上的性能。

策略 2 本实验针对不平衡的分类问题以 SMOTERENN+SVM 算法为架构，比较未使用采样算法和使用采样算法的准确率。

本文的实验策略，本文比较采样算法以及不使用采用算法的准确率比较

根据实验策略一，住宅能耗数据经过归一化处理，以 SVM 算法为基础，采用四种特征工程方法的实验结果如表 5.9 所示，以 Accuracy、Precision、Recall、F-measure 四个指标为依据，实验结果表明以 RF+PCA+SVD 为特征工程算法在住宅能耗数据上的性能较高。以平均值（mean）和最大值(max)作为各个参数分支。

表 5.9 分类算法实验结果

	评价指标		SVM	GBDT	BP
第一季度	Accuracy	Mean	0.822	0.799	0.799
		Max	0.822	0.813	0.813
	Precision	Mean	0.818	0.797	0.81
		Max	0.818	0.812	0.832
	Recall	Mean	0.822	0.815	0.816
		Max	0.822	0.838	0.838
	F-measure	Mean	0.818	0.797	0.812
		Max	0.818	0.813	0.833
第二季度	Accuracy	Mean	0.747	0.707	0.707
		Max	0.759	0.721	0.721
	Precision	Mean	0.749	0.716	0.752
		Max	0.761	0.732	0.764
	Recall	Mean	0.747	0.747	0.747
		Max	0.759	0.759	0.759
	F-measure	Mean	0.746	0.703	0.747
		Max	0.758	0.717	0.757
第三季度	Accuracy	Mean	0.774	0.723	0.723
		Max	0.790	0.744	0.743
	Precision	Mean	0.775	0.724	0.771

		Max	0.792	0.745	0.794
	Recall	Mean	0.774	0.774	0.774
		Max	0.790	0.790	0.790
	F-measure	Mean	0.772	0.719	0.768
		Max	0.788	0.740	0.790
第四季度	Accuracy	Mean	0.777	0.757	0.765
		Max	0.784	0.768	0.760
	Precision	Mean	0.774	0.756	0.776
		Max	0.783	0.765	0.790
	Recall	Mean	0.777	0.777	0.77
		Max	0.784	0.784	0.780
	F-measure	Mean	0.773	0.752	0.774
		Max	0.780	0.763	0.787

根据实验策略二，住宅建筑数据经过归一化数据预处理，采用 RFPCASVD 特征工程方法，比较 GBDT、SVM 和 BP 算法在住宅建筑耗能数据在性能差异。实验表明 SVM 的实验效果明显高于其他两个算法。为了从多维度比较各个算法在住宅建筑数据上的表现，SVM、BP 和 GBDT 算法在住宅建筑分类场景中算法复杂度结果如表 5.10 所示，结果为 20 次十字交叉的算法运行平均值。

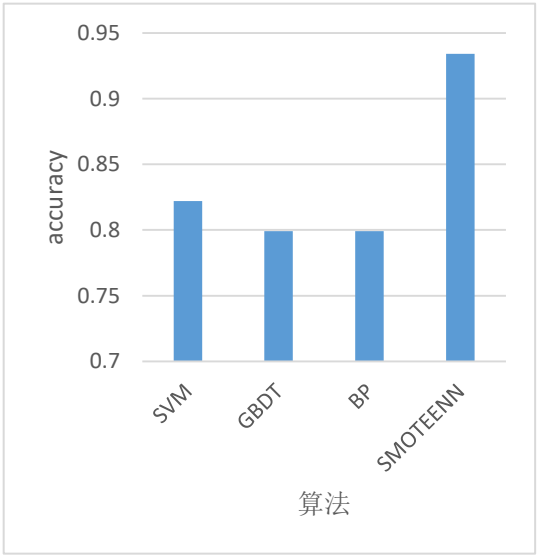
表 5.10 CPU 时间比较（单位，秒）

季度	SVM	GBDT	BP
第一季度	201.3	8.5	2.6
第二季度	798.1	10.5	4.1
第三季度	657.2	15.8	4.6
第四季度	640.7	14.4	2.7

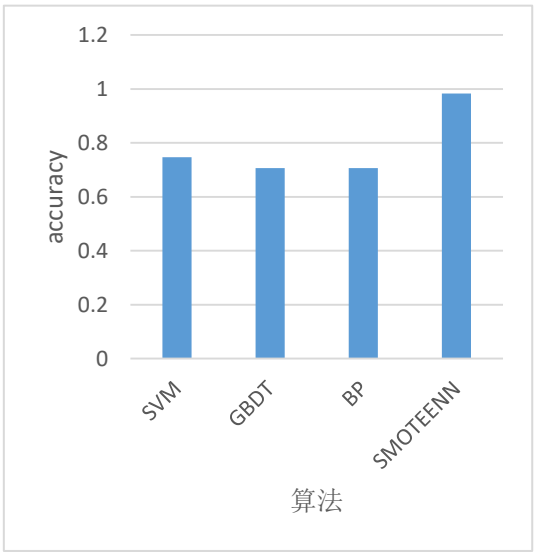
如表 5.10 所示，在各个季度住宅能耗等级预测中，SVM 算法所消耗的 CPU 时间最长，即算法复杂度较高，但是算法效果也最好，算法复杂度高的原因应该是通过核函数的作用，将低维的数据集映射到更高维的特征空间中，也就是无形中的升维，经验表明，维度越高，算法的复杂度越大，模型算法的训练越困难，所需要的时间也就越多，占用 CPU 资源的时间也就越长。。

根据以上两个策略，实验结果表明 RF+PCA+SVD-SVM 算法框架更适合于住宅能耗数据分类场景。由于上章统计结果表明中，由于对住宅建筑能耗的分级情况中的每个等级数目不相等，由此产生了不平衡分类的问题，而不平衡的数据集会对模型算法的实验结果产生影响，因此实验策略二引入 SMOTEENN 复合的采样算法提升预测模型分类的准确性，实验结果如下表 5.11 和图 5.5 所示。实验

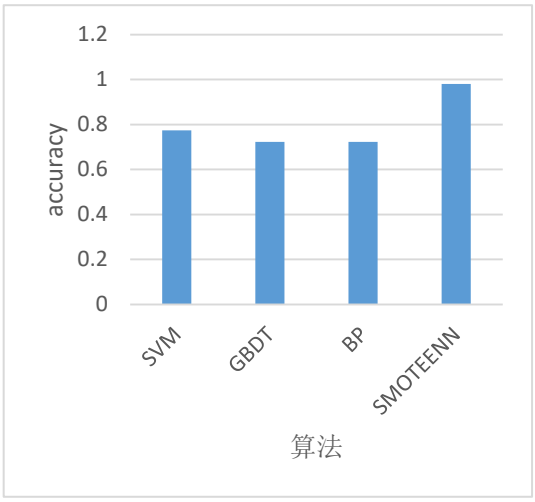
表明引入 SMOTEENN 算法大幅度提高了分类的准确性，而且算法的稳定性以及各个分类指标都远远高于其他未采用采样算法的分类结果。因此。本文提出的基于住宅建筑能耗预测算法流程能够满足预测的准确性要求，能准确的预测出下一个月住宅建筑的能耗等级，为政府职能部门的宏观调控提供数据支持。



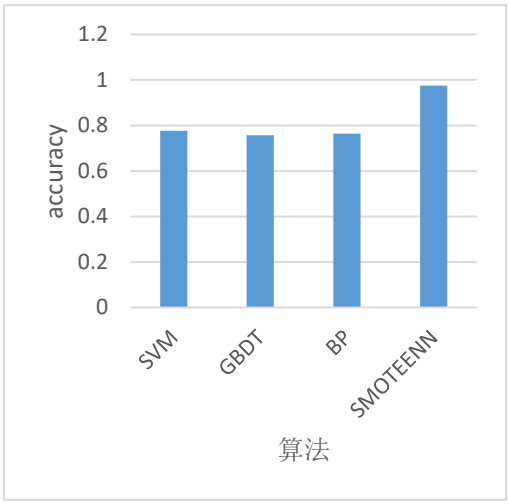
(a) 第一季度算法 accuracy 比较结果图



(b) 第二季度算法 accuracy 比较结果图



(c) 第三季度算法 accuracy 比较结果图



(d) 第四季度算法 accuracy 比较结果图

图 5.5 算法分类结果比较图

表 5.11 采样算法实验结果

	评价指标	实验指标	SMOTEENN	无采样
第一季度	Accuracy	Mean	0.934	0.822
		Max	0.938	0.822
	Precision	Mean	0.959	0.818
		Max	0.967	0.818
	Recall	Mean	0.934	0.822

		Max	0.938	0.822
	F-measure	Mean	0.945	0.818
		Max	0.951	0.818
第二季度	Accuracy	Mean	0.983	0.747
		Max	0.987	0.759
	Precision	Mean	0.990	0.749
		Max	0.994	0.761
	Recall	Mean	0.983	0.747
		Max	0.987	0.759
	F-measure	Mean	0.987	0.746
		Max	0.991	0.758
第三季度	Accuracy	Mean	0.98	0.774
		Max	0.980	0.790
	Precision	Mean	0.989	0.775
		Max	0.991	0.792
	Recall	Mean	0.978	0.774
		Max	0.980	0.790
	F-measure	Mean	0.983	0.772
		Max	0.985	0.788
第四季度	Accuracy	Mean	0.975	0.777
		Max	0.980	0.784
	Precision	Mean	0.989	0.774
		Max	0.996	0.783
	Recall	Mean	0.974	0.777
		Max	0.980	0.784
	F-measure	Mean	0.982	0.773
		Max	0.988	0.780

5.5 本章小结

本章将改进型聚类算法应用到能耗数据分析的实际场景中，针对能耗数据进行仿真。本文针对能耗数据的稀疏高维度特性，对 PCA、SVD、RF 特征工程进行仿真比较。并将三种算法相结合，基于 SVM 算法提出组合算法。对住宅建筑能耗预测问题进行深入研究，并将误差控制在可接受范围内。采用改进型聚类算

法对住宅建筑能耗进行分级。引入不平衡类分类处理方法，提高预测住宅建筑能耗等级准确率。在能耗等价预测时，引入 SVM、GBDT、BP 等做比较算法，实验表明，基于 SMOTEENN 框架的 SVM 算法在住宅建筑能耗等级分类场景中有很好的性能，能够满足预测住宅建筑能耗等级预测要求。

第6章 总结与展望

6.1 全文总结

本文主要是针对能耗数据提出一种准确预测住宅建筑电力能耗的框架。并且能利用大量数据对住宅建筑进行电力能耗分类，并采用机器学习方法预测建筑物的电力能耗等级。本文取得研究成果如下：

（1）针对能耗数据的高纬度系数特性，本文通过实验讨论了 PCA、SVD、RF 等几种特征工程方法，并且结合 SVM 算法，提出了 RF+PCA+SVD 算法。

（2）在对住宅建筑赣能耗等价划分的同时提出了一种 PSO+Kmeans 算法提高聚类准确性，并且引用经典的 UCI 数据集，引入聚类评估参数评价算法改进效果。

（3）本文采用数据挖掘手段分析研究住宅建筑能耗数据，根据住宅建筑能耗数据特性，提出适用于该场景的能耗预测算法和能耗等级预测流程

（4）在对住宅建筑能耗等级预测时，采用机器学习方法对数据不平衡进行重采样，提高预测等级的准确性。并且通过实验验证了这一算法的可行性。在引入处理不平衡分类的同时，通过大量实验验证 RF-PCA-SVD+SVM 在能耗数据分类上有很好的效果。

本课题的研究主要是针对住宅建筑能耗数据精准预测未来住宅建筑的电力能耗值以及建筑能耗等级。根据住宅建筑高维度特性，本课题提出了解决该问题的算法流程。采用数据挖掘技术解决住宅建筑能耗预测问题的方法，并为后来者提供了一种解决方案，并为国家职能部门的能源调配提供了依据，达到节约能源的目的；基于一个区域的住宅建筑能耗等级划分可以为以后根据住宅电力能耗等级进行收费。

6.2 下一步展望

（1）针对不平衡算法的采样问题，在以后的学习生活中，再不影响精度的情况下，提出一种更加高效的采样算法和使用深度学习算法，提高分类准确性。

（2）本文采用建筑物的建筑数据存在缺失现象，并且住宅数据大多集中在佛罗里达州，希望能找到丢失的建筑数据以及能广泛的收集分布在各地的能耗电力能耗数据。并且能源数据能够进行分项计量，根据数据研究用户行为。

致谢

白云苍狗，转眼间三年的硕士学生生涯即将画上圆满的句号。回想整个研究生学习阶段，总有一种不枉此生的感觉。在此我要感谢我的母校同济大学，给予我如此优秀的汲取知识平台，你的内涵呢和包容让我在步入社会前不断成长；感谢所有给予我知识营养的老师们，不仅带给我新的知识，还交给我为人处世的方法 ，让我从一个青涩懵懂的少年蜕变成了能独立科研研究能力和思想的研究生。在此，我由衷的感谢同济各位优秀老师、同学在我的生活以及学习上的帮助。你们的陪伴是我人生中最宝贵的财富，是你们的陪伴让我的学生生涯更加丰富多彩。。

首先，我由衷的感谢肖辉教授。您严谨求实的学术态度以及宽厚待人的人格魅力，一直是我们前进的标杆。三年的研究生生涯，您扎实的理论知识以及丰富的经验，由浅入深的指导我们进行科研工作。当我们在学术中遇到挫折时，您就像一个航标灯指引我们攻克科研上的难题。从论文选题、研究方法、方案论证以及文章结构等方面给予我悉心指导和帮助，直至论文完成。此外，感谢课题组岳继光老师、苏永清老师、何斌老师、董延超老师等老师在研究生期间给予我的指导和帮助。同时，还要感谢电子与信息工程学院控制系的所有教职工，感谢你们为我们提供了良好的研究环境与氛围。

我所在的实验室是一个温暖的集体。这里我要感谢张凯以及王雪松学弟，是你们在我孤独、无助、焦虑的时候站在我的身旁，给予我无私的帮助。你们在论文实验方面给出好的建议，在论文思想方面碰撞出火花。感谢陈小双师姐和殷文杰师兄，每当我遇到难题时，总是积极地帮助我、耐心的指导我；感谢同实验室的陆正飞、李爽、彭玲和罗彩姿学妹，我们一起学习、一起组织聚餐建设实验室氛围，我由衷的感谢们这三年的陪伴，也衷心的祝愿你们在以后的工作、生活中一帆风顺。同时感谢朱应昶学习、朱蒙蒙学弟、李金运学弟以及曹巍学弟，正因为你们的陪伴，实验室显得倍加温馨。

最后把我最诚挚的感谢留给我的家人，正是由于你们的无私支持，我才能生活的如此幸福快乐。再华美的辞藻也无法表达我对你们的感激之情。无论遇到再大的困难，你们永远是我停靠的港湾；你们的鼓励、陪伴始终伴随着我的前进的道路;你们的包容、理解与支持能让我在没有后顾之忧的情况下坚持自己的选择，我会继续努力前进，以更大的成绩回报你们。

2018.03.10

参考文献

[1] 苏晓峰. 大型公共建筑能耗监测、模型及管理信息系统研究[D]. 西安建筑科技大学, 2013.

[2] Corchado E, Woźniak M, Abraham A, et al. Recent trends in intelligent data analysis[J]. Neurocomputing, 2014, 126(3):1-2.

[3] Snasel V. Intelligent Data Analysis and its Applications: Volume 2[J]. Advances in Intelligent Systems & Computing, 2014, 298(4):549-563.

[4] 黄解军, 潘和平, 等. 数据挖掘技术的应用研究[J]. 计算机工程与应用, 2003, 39(2):45-48.

[5] Wang L, Tong Y F. Application Research of Data Mining Technology in Power Dispatching Management System[C] International Conference on Smart Grid and Electrical Automation. IEEE, 2016:1-4.

[6] 顾佳跃, 赵晓静, 肖筱华. 面向智慧城市的大数据处理技术研究与实现[C] 中国计算机用户协会网络应用分会2014年网络新技术与应用年会. 2014.

[7] Lin C F, Yeh Y C, Yu H H, et al. Data mining for providing a personalized learning path in creativity: An application of decision trees[J]. Computers & Education, 2013, 68(4):199-210.

[8] Gu Y X, Wang Q R, Suen C Y. Application of a multilayer decision tree in computer recognition of chinese characters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1983, 5(1):83.

[9] 马玥, 姜琦刚, 孟治国,等. 基于随机森林算法的农耕区土地利用分类研究[J]. 农业机械学报, 2016, 47(1):297-303.

[10] Sadeghi-Mobarakeh A, Kohansal M, Papalexakis E E, et al. Data Mining based on Random Forest Model to Predict the California ISO Day-ahead Market Prices[C] Isgt. 2017.

[11] Mu H, Qi D, Zhang M. Edge Detection of Wood Image with Rot Based on BP Neural Network[J]. Journal of Convergence Information Technology, 2013, 8(2):506-513.

[12] Wu J, Zhou L, Du X, et al. Junction Temperature Prediction of IGBT Power Module Based on BP Neural Network[J]. Journal of Electrical Engineering & Technology, 2014, 9(3):970-977.

[13] Li J, Xiong F, Li J, et al. Contrastive Research of SVM and BP Neural Network in AOD Prediction[C] China Control Conference. 2017.

[14] 曹贵宝. 随机森林和卷积神经网络在神经细胞图像分割中的应用研究[D]. 山东大学, 2014.

[15] Zhu X, Guo J, Xie Z. A Dynamic Weighing Method for Portal Crane in Bulk Port: Based on Clustering and BP Neural Network[C] International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration. IEEE, 2017:22-25.

[16] Burton S H, Morris R G, Giraud-Carrier C G, et al. Mining useful association rules from questionnaire data[J]. Intelligent Data Analysis, 2014, 18(3):479-494.

[17] Ahmed A, Korres N E, Ploennigs J, et al. Mining building performance data for energy-efficient operation[J]. Advanced Engineering Informatics, 2011, 25(2):341-354.

[18] Hou Z, Lian Z, Ye Y, et al. Data mining based sensor fault diagnosis and validation for building

air conditioning system[J]. Energy Conversion & Management, 2006, 47(15–16):2479-2490.

[19] Gao Y, Tumwesigye E, Cahill B, et al. Using data mining in optimisation of building energy consumption and thermal comfort management[C] International Conference on Software Engineering and Data Mining. IEEE, 2010:434-439.

[20] Chirarattananon S, Taveekun J. An OTTV-based energy estimation model for commercial buildings in Thailand[J]. Energy & Buildings, 2004, 36(7):680-689.

[21] Westphal F S, Lamberts R. Regression analysis of electric energy consumption of commercial buildings in Brazil[C], 清华大学, 2007.

[22] S.Karatasou ,M,Santamouris,V.Geros,Modeling and predicting building's energy use with artifitial neural networks；Methods anf results[J].Energy anf Buildings,2006,38(8):949~958

[23] 陈文凭, 杨昌智. 基于人工神经网络的商场建筑冷负荷预测研究[C] 湖南省暖通空调制冷学术年会. 2007.

[24] 雷娅蓉. 重庆市居住建筑能耗预测方法研究[D]. 重庆大学, 2008.

[25] 蒋毅. 建筑能耗的统计平台及其基于BP神经网络预测方法的研究[D]. 华南理工大学, 2012.

[26] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16(1):321-357.

[27] Truccolo W, Eden U T, Fellows M R, et al. A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects[J]. Journal of Neurophysiology, 2005, 93(2):1074-1089.

[28] Donaldson I, Martin J, De B B, et al. PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine[J]. Bmc Bioinformatics, 2003, 4(1):11.

[29] Wilson D L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data[J]. Systems Man & Cybernetics IEEE Transactions on, 1972, SMC-2(3):408-421.

[30] Tien D X, Lim K W, Jun L. Comparative study of PCA approaches in process monitoring and fault detection[C] Industrial Electronics Society, 2004. IECON 2004. Conference of IEEE. IEEE, 2004:2594-2599 Vol. 3.

[31] Batuwita R, Palade V. FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning[J]. IEEE Transactions on Fuzzy Systems, 2010, 18(3):558-571.

[32] Xu J H, Liu H. Web user clustering analysis based on KMeans algorithm[M]. IEEE, 2010.

[33] Kennedy J, Eberhart R. Particle swarm optimization[C] IEEE International Conference on Neural Networks, 1995. Proceedings. IEEE, 2002:1942-1948 vol.4.

[34] 刘悦婷, 李岚. 基于自适应权重的粒子群和K均值混合聚类算法研究[J]. 甘肃科学学报, 2010, 22(4):106-109.

[35] 吕奕清, 林锦贤. 基于MPI的并行PSO混合K均值聚类算法[J]. 计算机应用, 2011, 31(2):428-431

[36] 荣盘祥, 曾凡永, 黄金杰. 数据挖掘中特征选择算法研究[J]. 哈尔滨理工大学学报, 2016, 21(1):106-109.

[37] Tien D X, Lim K W, Jun L. Comparative study of PCA approaches in process monitoring and fault detection[C] Industrial Electronics Society, 2004. IECON 2004. Conference of IEEE. IEEE, 2004:2594-2599 Vol. 3.

[38] Dunia R, Qin S J, Edgar T F, et al. Identification of faulty sensors using principal component analysis[J]. Aiche Journal, 2010, 42(10):2797-2812.

[39] Kaistha N, Upadhyaya B R. Incipient Fault Detection and Isolation of Field Devices in Nuclear Power Systems Using Principal Component Analysis[J]. Nuclear Technology, 2001, 136(2):221-230.

[40] Colace F, Santo M D, Moscato V, et al. Data Management in Pervasive Systems[C] Springer Publishing Company, Incorporated, 2015.

[41] Gardin J C. Annual Review of Information Science and Technology[M]. John Wiley & Sons, Inc. 2008.

[42] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8):30-37.

[43] Breiman L. Random Forest[J]. Machine Learning, 2001, 45:5-32.

[44] Menze B H, Kelm B M, Masuch R, et al. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data[J]. Bmc Bioinformatics, 2009, 10(1):1-16.

[45] Strobl C, Boulesteix A L, Zeileis A, et al. Bias in random forest variable importance measures: illustrations, sources and a solution[J]. Bmc Bioinformatics, 2007, 8(1):25.

[46] Androutsopoulos I, Koutsias J, Chandrinos K V, et al. An evaluation of Naive Bayesian anti-spam filtering[J]. Tetsu-to-Hagane, 2000, cs.cl 0006013(2):9--17.

[47] Guo G, Wang H, Bell D, et al. KNN Model-Based Approach in Classification[C] Otm Confederated International Conferences "on the Move To Meaningful Internet Systems. Springer, Berlin, Heidelberg, 2003:986-996.

[48] Menard S. Applied Logistic Regression Analysis[J]. Technometrics, 2002, 38(2):192-192.

[49] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5):1189-1232.

[50] Hecht-Nielsen. Theory of the backpropagation neural network[J]. Neural Networks, 1988, 1(1):445-445.

[51] Cortes C, Vapnik V. Support-vector networks[C] Machine Learning. 1995:273-297.

[52] Vapnik V, Levin E, Cun Y L. Measuring the VC-dimension of a learning machine[M]. MIT Press, 1994.

[53] Truccolo W, Eden U T, Fellows M R, et al. A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects[J]. Journal of Neurophysiology, 2005, 93(2):1074-1089.

[54] 张浩然, 汪晓东. 回归最小二乘支持向量机的增量和在线式学习算法[J]. 计算机学报, 2006, 29(3):400-406.

[55] 奉国和. SVM分类核函数及参数选择比较[J]. 计算机工程与应用, 2011, 47(3):123-124.

[56] Kannan S R, Ramathilagam S A, Pandiyarajan R. Effective fuzzy c-means based kernel function in segmenting medical images [J]. Computers in Biology & Medicine, 2010, 40(6):572-579.

[57] Donaldson I, Martin J, De B B, et al. PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine[J]. Bmc Bioinformatics, 2003, 4(1):11.

[58] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5):1189-1232.

[59] 周志华. 机器学习 : Machine learning[M]. 清华大学出版社, 2016.

[60] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2011, 16(1):321-357.

[61] Wilson D L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data[J]. Systems Man & Cybernetics IEEE Transactions on, 1972, SMC-2(3):408-421.

[62] Ester M, Kriegel H P, Sander J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[J]. 1996.

[63] P Stone，G Kuhlmann，ME Taylor，Y Liu. "Keepaway Soccer: From Machine Learning Testbed to Benchmark". Robocup: Robot Soccer World Cup IX, 2005,4020:93-105.

[64] U Brefeld，T Scheffer. "AUC maximizing support vector learning". Icml Workshop on Roc Analysis in Machine Learning, 2005, pp.377 – 380.

附录A 论文简要代码

```
def
calculateinnercluster(self,clusterdictionary,co
mbinationNumber):
    dataset=self.dataSet
    row,colum=dataset.shape
    averagecenter=np.mean(dataset, axis=0)
    sumindex = 0
    datareslist = []
    centerNumbers=self.ikmeans# 聚类中心
数
    dataaverge = np.zeros(colum).reshape(1,
-1)
    for data in dataset:
        temp = data - averagecenter
        temp = temp.reshape(1, -1)
        dataaverge =
np.concatenate((dataaverge, temp), axis=0)
    dataaverge = dataaverge[1:, ]
    for data in range(colum):
        temp =
np.squeeze(np.dot(dataaverge[0:, data].T,
dataaverge[0:, data])) / row
        datareslist.append(temp)
    for data in datareslist:
        sumindex = sumindex + data ** 2
    sumindexscore = math.sqrt(sumindex)
    sumclusterScore = 0
    for i in range(centerNumbers):
        clusterMember =
clusterdictionary[i]

def updataNewFieness(self,clusterdictionary):
    number=self.ikmeans
    colum=self.n_attrs
    randomdatalist=[]
    for i in range(number):
        randomdatalist.append(i)

randomindexList=list(permutations(randomda
talist,2))
        if len(clusterMember)>0:
            clusterMember =
np.array(clusterMember)

row,colum=clusterMember.shape
            clusterOnememberAverage =
np.mean(clusterMember)
            dataaverge =
np.zeros(colum).reshape(1, -1)
            for data in clusterMember:
                temp = data -
clusterOnememberAverage
                temp = temp.reshape(1, -
1)
                dataaverge =
np.concatenate((dataaverge, temp), axis=0)
            dataaverge = dataaverge[1:, ]
            datareslist.clear()
            for data in range(colum):
                temp =
np.squeeze(np.dot(dataaverge[0:, data].T,
dataaverge[0:, data])) / row
                datareslist.append(temp)
            for data in datareslist:
                sumclusterScore =
sumclusterScore + data ** 2
    clusterInnerScore = sumclusterScore /
centerNumbers / sumindexscore
    return clusterInnerScore


    center=self.position
    sumOuterScore=0
    for data in randomindexList:

average=np.zeros(colum).reshape(1,-1)
        for index in data:
            if (index>=self.ikmeans):
                print (index)
            average=average+center[index]
```

```
            average=average/2#两个类之间的
平均值
            clusterOneCenter=center[data[0]]
            clusterTwoCenter=center[data[1]]

clusterone=clusterdictionary[data[0]]

clustertwo=clusterdictionary[data[1]]
            stedv =
self.calculateaveragedistance()
            unionIndexScore=0
            clusterOneScore=0
            for member in clusterone:
                distance =
np.sqrt(np.sum(np.square(member - average)))
                if (distance <= stedv):
                    unionIndexScore =
unionIndexScore + 1
distance=np.sqrt(np.sum(np.square(member -
clusterOneCenter)))
                if (distance<=stedv):

clusterOneScore=clusterOneScore+1
            clusterTwoScore=0
            for member in clustertwo:

distance=np.sqrt(np.sum(np.square(member -
average)))
                if(distance <=stedv):

unionIndexScore=unionIndexScore+1

distance=np.sqrt(np.sum(np.square(member -
#计算适应度值

def
calculateinnerdistance(clustermap,dataset,cent
erNumbers):
    averagecenter = np.mean(dataset)
    row ,colum=dataset.shape
    sumindex = 0
    datareslist=[]
    dataaverge=np.zeros(colum).reshape(1,-
1)
clusterTwoCenter)))
                if(distance<=stedv):

clusterTwoScore=clusterTwoScore+1
            TwoclusterScore=0
            if
(clusterTwoScore<clusterOneScore):

TwoclusterScore=clusterOneScore
            else :

TwoclusterScore=clusterTwoScore

            if (TwoclusterScore!=0):
                outClusterScore =
unionIndexScore / TwoclusterScore
            else :
                outClusterScore=0

sumOuterScore=sumOuterScore+outClusterSc
ore/self.ikmeans/(self.ikmeans-1)
            # 计算类间的距离

scat=self.calculateinnercluster(clusterdictionar
y,len(randomindexList))
    sdbw=sumOuterScore+scat
    self.sdbw = sdbw
    if (sdbw<self.fitness):
        self.fitness = sdbw
        self.bestposition = self.position

    return randomindexList


    for data in dataset:
        temp=data - averagecenter
        temp=temp.reshape(1,-1)
        dataaverge =
np.concatenate((dataaverge, temp), axis=0)

    dataaverge=dataaverge[1:,]
    for data in range(colum):
        temp=
```

```
np.squeeze(np.dot(dataaverge[0:,data].T,
dataaverge[0:,data]))/row
            datareslist.append(temp)
        for data in datareslist:
            sumindex=sumindex+data**2
        sumindexscore=math.sqrt(sumindex)
        clusterInnerScore = 0
        sumclusterScore=0
        for i in range(centerNumbers):
            clusterMember = clustermap[i]
            if(len(clusterMember)>0):
                clusterMember =
np.array(clusterMember)
row,colum=clusterMember.shape
                clusterOnememberAverage =
np.mean(clusterMember)
                dataaverge =
np.zeros(colum).reshape(1, -1)
                for data in clusterMember:
                    temp =data -
clusterOnememberAverage
                    temp=temp.reshape(1,-1)
                    dataaverge =
np.concatenate((dataaverge, temp), axis=0)
                dataaverge = dataaverge[1:, ]
                datareslist.clear()
                for data in range(colum):
                    temp =
np.squeeze(np.dot(dataaverge[0:, data].T,
dataaverge[0:, data])) / row
                    datareslist.append(temp)
                for data in datareslist:
    sumclusterScore=sumclusterScore+data**2
        clusterInnerScore = sumclusterScore /
centerNumbers / sumindexscore
        return clusterInnerScore

        # kmeans+PSO 算法
def kMeansPso(dataSet, k,
distMeas=distEclud, createCent=0):
        m = shape(dataSet)[0]
        clusterAssment = mat(zeros((m, 2)))    #
create mat to assign data points
        # to a centroid, also holds SE of each
point
        centroids=createCent
        clusterChanged = True
        number=0
        reslist=[]
        minfitness=sys.maxsize
        bestcenterdata=0
        bestlabellist=[]
        while clusterChanged:
            reslist.clear()
            clusterChanged = False
            number=number+1
            for i in range(m):    # for each data
point assign it to the closest centroid
                minDist = inf
                minIndex = -1
                for j in range(k):
                    distJI =
distMeas(centroids[j, :], dataSet[i, :])
                    if distJI < minDist:
                        minDist = distJI;
                        minIndex = j
                reslist.append(minIndex)
                if clusterAssment[i, 0] !=
minIndex:
                    clusterChanged = True
                clusterAssment[i, :] =
minIndex, minDist ** 2
            for cent in range(k):    # recalculate
centroids
                ptsInClust =
dataSet[nonzero(clusterAssment[:, 0].A ==
cent)[0]]    # get all the point in this cluster
                centroids[cent, :] =
mean(ptsInClust, axis=0)    # assign centroid
to mean

            sdbw = calculatefitness(centroids,
reslist, dataSet)
            if sdbw<minfitness:
                bestlabellist.clear()
                minfitness=sdbw
```

```
                bestcenterdata=centroids
                bestlabellist=reslist
        sdbw = calculatefitness(centroids, reslist,
dataSet)
    # 数据预处部分代码
    df =
pd.read_csv('/Users/shenshoupeng/Desktop/da
ta/fourthquarter.csv', header=-1).values
naturaldata = np.array(df)
print (naturaldata.shape)
row ,colum=naturaldata.shape
        temp = np.array(data).reshape(1, -1)
    if(data[0]==0):

array0=np.concatenate((array0,temp),axis=0)
    elif(data[0]==1):
        array1 = np.concatenate((array1, temp),
axis=0)
    elif (data[0] == 2):
        array2 = np.concatenate((array2, temp),
axis=0)
    elif (data[0] == 3):
        probas = np.concatenate((array3,
temp), axis=0)
    elif (data[0] == 4):
        array4 = np.concatenate((array4, temp),
axis=0)
array0=array0[1:,]
array1=array1[1:,]
array2=array2[1:,]
array3=array3[1:,]
array4=array4[1:,]
trainingdata=np.zeros(colum).reshape(1,-1)
testdata=np.zeros(colum).reshape(1,-1)
row,colum=array0.shape
rowtemp=(int)(row*0.9)
trainingdata=np.concatenate((trainingdata,
array0[0:rowtemp,]), axis=0)
testdata=np.concatenate((testdata,
array0[rowtemp:,]), axis=0)
row,colum=array1.shape
rowtemp=(int)(row*0.9)
trainingdata=np.concatenate((trainingdata,
        print("pso kmeans"+str(sdbw))
        return centroids,
clusterAssment,number,reslist,bestcent

array0=np.zeros(colum).reshape(1,-1)
array1=np.zeros(colum).reshape(1,-1)
array2=np.zeros(colum).reshape(1,-1)
array3=np.zeros(colum).reshape(1,-1)
array4=np.zeros(colum).reshape(1,-1)
for data in naturaldata:

array1[0:rowtemp,]), axis=0)
testdata=np.concatenate((testdata,
array1[rowtemp:,]), axis=0)
row, colum = array2.shape
trainingdata = np.concatenate((trainingdata,
array2[0:rowtemp, ]), axis=0)
testdata=np.concatenate((testdata,
array2[rowtemp:,]), axis=0)
row, colum = array3.shape
rowtemp = (int)(row * 0.9)
trainingdata = np.concatenate((trainingdata,
array3[0:rowtemp, ]), axis=0)
testdata=np.concatenate((testdata,
array3[rowtemp:,]), axis=0)
row, colum = array4.shape
rowtemp = (int)(row * 0.9)
trainingdata = np.concatenate((trainingdata,
array4[0:rowtemp, ]), axis=0)
testdata=np.concatenate((testdata,
array4[rowtemp:,]), axis=0)
trainingdata=trainingdata[1:,]
testdata=testdata[1:,]
print (trainingdata.shape)
trainlabel=trainingdata[0:,0]
trainingdata=trainingdata[0:,1:]
testlabel=testdata[0:,0]
testdata=testdata[0:,1:]
min_max_scaler =
preprocessing.MinMaxScaler()
trainingdata =
min_max_scaler.fit_transform(trainingdata)
testdata=min_max_scaler.transform(testdata)
```

个人简历、在读期间发表的学术论文与研究

个人简历：

沈寿鹏，男，1993 年 8 月生

2015 年 9 月 入同济大学控制科学与工程系，攻读硕士学位。

2015 年 6 月毕业于常熟理工学院电气与自动化工程学院自动化专业，获学士学位。