大规模智能电网用户的精细化电能分解与用电行为分析

# 摘要

智能电网用户的用电数据隐含着丰富的信息，每个负载都有其特有的用电模式。如果我们能从家庭的总用电量中挖掘出每个负载的用电模式，这对于提高能源的利用率有很大的意义。电能分解技术可用于挖掘每个负载的用电模式，电能分解技术是指根据家庭的总用电量分解出家庭中各个负载的消耗量。目前的电能分解技术包括两类，一类是基于嵌入式的电能分解，但这种方法存在着严重的不足，不仅硬件成本高而且人工成本也很高。另一类是基于非嵌入式的电能分解。

本论文研究的是基于非嵌入式的电能分解，通过分析用电数据中总用电量和负载用电量之间的关系，从而得到用户各用电器的用电量。同时根据用户的用电数据分析用户的家庭属性，包括家庭面积、房间数和家庭人数等。

关键词：精细化电能分解；用电行为分析；矩阵分解；回归模型；深度学习

REFINED POWER DECOMPOSITION

AND POWER CONSUMPTION BEHAVIOR ANALYSIS OF LARGE SCALE SMART GRID USERS

# Abstract

The electricity grid data for smart grid users implies a wealth of information, and each load has its own specific power consumption pattern. If we can find out the electricity consumption pattern of each load from the total electricity consumption of the household, this will be of great significance for improving the utilization rate of energy. Electric energy decomposition technology can be used to mine the power consumption mode of each load. The energy decomposition technology refers to the decomposition of the consumption of each load in the household according to the total power consumption of the household. The current electric energy decomposition technology includes two types, one is based on embedded electric energy decomposition, but this method has serious drawbacks, not only high hardware cost but also high labor cost. The other type is based on non-embedded power decomposition.

The research in this dissertation is based on the non-embedded power decomposition. By analyzing the relationship between the total power consumption and the load power consumption in the power consumption data, the power consumption of each user's electrical appliances is obtained. At the same time, the user's family attributes are analyzed based on the user's electricity consumption data, including family size, number of rooms, and family size.

Keywords: refined power decomposition; power consumption behavior analysis;

matrix factorization; regression model; deep learning

目 录

[摘要 I](#_Toc517621567)

[Abstract II](#_Toc517621568)

[1绪论 1](#_Toc517621569)

[1.1课题研究背景及其意义 1](#_Toc517621570)

[1.2研究现状 1](#_Toc517621571)

[1.2.1基于嵌入式的用电分解 1](#_Toc517621572)

[1.2.2基于非嵌入式的用电分解 1](#_Toc517621573)

[1.2.3基于矩阵分解、回归模型以及神经网络的家庭属性特征预测 2](#_Toc517621574)

[1.3本文研究内容 2](#_Toc517621575)

[1.3.1分析Dataport公开数据集，对数据进行预处理 2](#_Toc517621576)

[1.3.2采用tensorflow框架实现矩阵分解进行电能分解 2](#_Toc517621577)

[1.3.3采用tensorflow框架实现矩阵分解、回归模型、神经网络进而实现属性特征预测 2](#_Toc517621578)

[2矩阵分解、回归模型以及深度学习概述 3](#_Toc517621579)

[2.1矩阵分解 3](#_Toc517621580)

[2.1.1矩阵分解 3](#_Toc517621581)

[2.1.2非负矩阵分解（NMF,Non-negative matrix factorization） 3](#_Toc517621582)

[2.2回归模型 3](#_Toc517621583)

[2.2.1线性回归 4](#_Toc517621584)

[2.2.2岭回归 4](#_Toc517621585)

[2.2.3套索回归 4](#_Toc517621586)

[2.3神经网络 4](#_Toc517621587)

[2.3.1神经网络 4](#_Toc517621588)

[2.3.2深度学习 4](#_Toc517621589)

[2.4本章小结 5](#_Toc517621590)

[3基于矩阵分解的精细化电能分解 6](#_Toc517621591)

[3.1总体思路 6](#_Toc517621592)

[3.2基于矩阵分解的精细化电能分解原理 6](#_Toc517621593)

[3.3基于矩阵分解的精细化电能分解训练 7](#_Toc517621594)

[3.3.1数据归一化处理 7](#_Toc517621595)

[3.3.2损失函数 7](#_Toc517621596)

[3.4代码实现思路 7](#_Toc517621597)

[3.5本章小结 8](#_Toc517621598)

[4基于矩阵分解、回归模型、深度学习用电行为分析 9](#_Toc517621599)

[4.1总体思路 9](#_Toc517621600)

[4.2用电行为分析的矩阵分解、回归模型以及深度学习模型结构 9](#_Toc517621601)

[4.2.1非负矩阵分解 9](#_Toc517621602)

[4.2.2回归模型结构 10](#_Toc517621603)

[4.2.3深度学习模型结构 10](#_Toc517621604)

[4.3用电行为分析的矩阵分解、回归模型以及深度学习模型训练 10](#_Toc517621605)

[4.3.1数据归一化处理 10](#_Toc517621606)

[4.3.2损失函数 11](#_Toc517621607)

[4.3代码实现思路 12](#_Toc517621608)

[4.4本章小结 13](#_Toc517621609)

[5实验结果与分析 15](#_Toc517621610)

[5.1实验环境与配置 15](#_Toc517621611)

[5.2数据集来源和预处理 16](#_Toc517621612)

[5.2.1实验数据集来源 16](#_Toc517621613)

[5.2.2实验数据预处理 16](#_Toc517621614)

[5.3实验评估指标 16](#_Toc517621615)

[5.4实验结果 16](#_Toc517621616)

[5.5实验结果分析 19](#_Toc517621617)

[5.6本章小结 20](#_Toc517621618)

[总结与展望 21](#_Toc517621619)

[参考文献 23](#_Toc517621620)

[致谢 24](#_Toc517621621)

# 1绪论

## 1.1课题研究背景及其意义

居民用电是能源消耗的一大部分，约占总能源消耗的三分之一。通过能量分解来分解出居民家庭中各种负载的消耗量，例如冰箱、洗衣机、电磁炉等，可以帮助家庭能源生态中的几个参与者做出明智的决策。例如，研究表明，拥有能源分解能力的居民可以将能源消耗降低15%。能源细分还可以帮助电力公司和政策制定者改进负荷预测和检测电力损耗或电力损失错误配置的设备。然而，尽管能量分解有许许多多的好处，但目前只有极少数的居民家里拥有能量分解的硬件。

除此之外，现在广告商投放广告都具有针对性，根据用户的特征从而投放更具有针对性的广告。例如，搜索引擎根据用户的搜索记录分析出用户的个人特征，个人特征包括性别、年龄、职业以及生活所在地，根据这些特征分析出用户的现阶段的需要甚至是将来的需求，最后针对用户的需求投放具有针对性的广告。同理，如果能通过居民的总用电量预测出居民的家庭属性特征，这里的家庭属性特征可以包括家庭人数、家庭房间数以及房屋所占面积，根据这些家庭属性特征，广告商可以设计出具有针对性的服务。

## 1.2研究现状

### 1.2.1基于嵌入式的用电分解

现在的用电分解主要是基于嵌入式的用电分解，这种方式需要人工安装传感器，需要在每个家庭用电器接电处安装传感器，每隔相同时间收集每个传感器的数据。大多数居民都没有安装能量分解的硬件是因为这种能量分解需要的硬件成本过于昂贵。硬件成本包括价格高至上千元高的频智能电表、各种传感器。除了硬件成本，还有各种硬件安装、硬件维护等成本。所以，现阶段的挑战是能研究出不用硬件的能量分解技术。虽然基于嵌入式的用电分解操作简单，但此种方法存在的严重弊端是分解的精确度不高，是因为在收集传感器数据时会存在误差。

### 1.2.2基于非嵌入式的用电分解

如今也存在基于非嵌入式的用电分解，也就是不用在每个用电器上安装传感器，通过收集一个家庭的总消耗用电量，通过分析用电数据中总用电量和用电器用电量之间的关系，并建立非嵌入式的模型对用户的总用电量进行分解，预测用户各用电器的用电量。

现有的基于非嵌入式用电分解主要是矩阵分解，矩阵分解主要是将矩阵分解成两个具有特殊含义的矩阵，但此种方法在一定意义上存在着问题，因为此种方法把数据解释为二维的，但事实上，这种解释性不强。除此之外，现有的基于非嵌入式的用电分解存在不高效和精确度低的问题，而造成这种问题的原因主要是上述所说的二维矩阵可解释性不强和对矩阵分解的优化函数不合适。

### 1.2.3基于矩阵分解、回归模型以及神经网络的家庭属性特征预测

现今，利用用户的家庭用电量来预测或者分析用户的家庭静态属性特征的相关研究尚未有研究，换句话说，本论文研究的基于矩阵分解、回归模型以及神经网络的家庭属性特征预测的研究属于首研。

## 1.3本文研究内容

### 1.3.1分析Dataport公开数据集，对数据进行预处理

由于Dataport数据集中存在一些无用的数据，也就是不符合本论文实验要求的数据，所以在做实验之前要先对数据做处理，本论文利用python语言、numpy和panda数据处理库，对Dataport公开数据集的数据进行预处理，筛选和处理成本实验需要的特定数据格式。

### 1.3.2采用tensorflow框架实现矩阵分解进行电能分解

利用tensorflow库搭建矩阵分解的模型框架，通过输入经过预处理的数据，最终通过优化损失函数达到分解矩阵也就是补全矩阵中缺失数据的目的。

### 1.3.3采用tensorflow框架实现矩阵分解、回归模型、神经网络进而实现属性特征预测

利用tensorflow库搭建回归模型、神经网络的模型框架，通过输入经过预处理的数据，最终通过优化损失函数达到预测家庭属性特征的目的。

# 2矩阵分解、回归模型以及深度学习概述

## 2.1矩阵分解

### 2.1.1矩阵分解

矩阵分解是把一个矩阵分解成两个或者多个矩阵的乘积。在矩阵运算中，把一个高维的矩阵分解成低维的或者形式比较简单又或者具有某种特性的两个或两个以上矩阵的乘积，这样的一种运算在矩阵理论的研究和应用中都具有非同一般的意义。相对于其他矩阵运算，矩阵分解有两个普遍公认的优点，一个是在分解后得到的分解矩阵能显示出原始矩阵的数值特征，例如矩阵的特征值和奇异值，另一个是矩阵分解的过程通常给我们提供了计算或者分析某些数值的根据。

# 2.1.2非负矩阵分解（NMF,Non-negative matrix factorization）

（1）定义

非负矩阵分解（NMF）算法是指把一个矩阵分解为两个非负的矩阵，换句话说，是在分解得到的两个矩阵非负数的约束条件之下进行分解的方法[1-2]。

（2）原理

传统的NMF[3-6]问题可以描述如下：

给定矩阵A∈R+n\*m，寻找非负矩阵U∈R+n\*r和非负矩阵V∈R+r\*m，其中r远小于n和m，使得A≈UV，之所以是约等于，是因为当前解法并非精确解，而只是数值上的近似解。

分解前后可以理解为：原始矩阵A的列向量是对左矩阵U中所有列向量的加权和，而权重系数就是右矩阵V对应列向量的元素，故称U为基矩阵，V为系数矩阵。NMF算法提供了基于简单迭代的求解U，V的方法，求解方法具有收敛速度快、左右非负矩阵存储空间小的特点，它能将高维的数据矩阵降维处理，适合处理大规模数据[7]。

非负矩阵分解是基于乘法更新发展的迭代更新算法，其将矩阵分解算法转化为如下的优化问题，即最小化两个矩阵之间的欧几里得距离的优化问题：

min‖A-A´‖2 =∑ij(Aij- A´)2……………………………………（2-1）

其中A为原始矩阵，A´为分解后的矩阵重构而成的矩阵，即A´=UV。

## 2.2回归模型

回归模型重要的基础或者方法就是回归分析，回归分析是研究一个变量（被解释变量）关于另一个（些）变量（解释变量）的具体依赖关系的计算方法和理论，是建模和分析数据的重要工具。

### 2.2.1线性回归

在[统计学](https://zh.wikipedia.org/wiki/%E7%BB%9F%E8%AE%A1%E5%AD%A6)中，线性回归（Linear regression）[8-9]是利用称为线性回归方程的[最小二乘](https://zh.wikipedia.org/wiki/%E6%9C%80%E5%B0%8F%E4%BA%8C%E4%B9%98%E6%B3%95" \o "最小二乘法)函数对一个或多个[自变量](https://zh.wikipedia.org/wiki/%E8%87%AA%E5%8F%98%E9%87%8F)和[因变量](https://zh.wikipedia.org/w/index.php?title=%E5%9B%A0%E5%8F%98%E9%87%8F&action=edit&redlink=1)之间关系进行建模的一种[回归分析](https://zh.wikipedia.org/wiki/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90)。

### 2.2.2岭回归

岭回归[10-11]分析是一种用于存在多重共线性（自变量高度相关）数据的技术。在多重共线性情况下，尽管最小二乘法（OLS）对每个变量很公平，但它们的差异很大，使得观测值偏移并远离真实值。岭回归通过给回归估计上增加一个偏差度，来降低标准误差。

### 2.2.3套索回归

Lasso算法（least absolute shrinkage and selection operator，又译最小绝对值收敛和选择算子、套索算法）是一种同时进行[特征选择](https://zh.wikipedia.org/wiki/%E7%89%B9%E5%BE%81%E9%80%89%E6%8B%A9" \o "特征选择)和[正则化](https://zh.wikipedia.org/wiki/%E6%AD%A3%E5%89%87%E5%8C%96)（数学）的[回归分析](https://zh.wikipedia.org/wiki/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90" \o "回归分析)方法，旨在增强[统计模型](https://zh.wikipedia.org/wiki/%E7%BB%9F%E8%AE%A1%E6%A8%A1%E5%9E%8B)的预测准确性和可解释性。虽然最早是为应用最小二乘法而定义的算法，lasso正则化可以简单直接地拓展应用于许多统计学模型上，包括[广义线性模型](https://zh.wikipedia.org/wiki/%E5%BB%A3%E7%BE%A9%E7%B7%9A%E6%80%A7%E6%A8%A1%E5%9E%8B" \o "广义线性模型)，[广义估计方程](https://zh.wikipedia.org/w/index.php?title=%E5%B9%BF%E4%B9%89%E4%BC%B0%E8%AE%A1%E6%96%B9%E7%A8%8B&action=edit&redlink=1)，[成比例灾难模型](https://zh.wikipedia.org/w/index.php?title=%E6%88%90%E6%AF%94%E4%BE%8B%E7%81%BE%E9%9A%BE%E6%A8%A1%E5%9E%8B&action=edit&redlink=1)和[M-估计](https://zh.wikipedia.org/w/index.php?title=M-%E4%BC%B0%E8%AE%A1&action=edit&redlink=1)。Lasso选择子集的能力依赖于限制条件的形式并且有多种表现形式，包括[几何学](https://zh.wikipedia.org/wiki/%E5%87%A0%E4%BD%95%E5%AD%A6" \o "几何学)，[贝叶斯统计](https://zh.wikipedia.org/w/index.php?title=%E8%B4%9D%E5%8F%B6%E6%96%AF%E7%BB%9F%E8%AE%A1&action=edit&redlink=1)，和[凸分析](https://zh.wikipedia.org/w/index.php?title=%E5%87%B8%E5%88%86%E6%9E%90&action=edit&redlink=1)。

## 2.3神经网络

### 2.3.1神经网络

神经网络[15-16]是一种网络结构，由输入层、隐藏层和输出层连接构成。每一层由多个节点构成，每一个节点代表一种特定的输出函数，我们称之为激活函数。相邻层之间通过全连接或者局部连接方式连接。

神经网络之所以能表达回归模型无法表达的复杂函数，是因为其有两个特性或者说是神经网络不同于回归模型的特点，一个是多层性，另一个是有非线性的激活函数。回归模型不管有多少层，最终也是线性的，而神经网络一个多层，另一个更重要的是有非线性的激活函数，在每一层输出前先经过非线性函数，这样一来，整个模型就是非线性的，也就可以拟合出任意的函数。

### 2.3.2深度学习

深度学习是含多隐层的[多层感知器](https://baike.baidu.com/item/%E5%A4%9A%E5%B1%82%E6%84%9F%E7%9F%A5%E5%99%A8" \t "_blank)。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示[12-14]。

深度学习为监督学习提供了一个强大的框架。通过添加更多层以及向层内添加更多单元，深度网络可以拟合复杂性不断增加的函数。给定足够大的模型和足够大的标注训练数据集，我们可以通过深度学习将输入向量映射到输出向量，完成大多数对人来说能迅速处理的任务。其他任务，比如不能被描述为将一个向量与另一个相关联的任务，或者对于一个人来说足够困难并需要时间思考和反复琢磨才能完成的任务，现在仍然超出了深度学习的能力范围。

## 2.4本章小结

本章对本论文所要用到的理论知识进行了大致的讲解。首先介绍了矩阵分解的定义和非矩阵分解的定义和原理，然后对回归模型的其中三种也就是线性回归、岭回归以及套索回归做了概念上的介绍，最后介绍了深度学习的概念。

# 3基于矩阵分解的精细化电能分解

## 3.1总体思路

本章实验的总体思路是先对原始数据做归一化处理，通过归一化处理，能使实验数据的不同向量具有可比性和使实验更容易收敛。接着对原始数据做非负矩阵分解，把原始数据矩阵分解为两个的矩阵，这两个矩阵是相对低维的矩阵，这样可以实现矩阵降维的目的。然后对分解得到的两个相对低维的矩阵做矩阵乘法就可以得出一个与原始数据矩阵相似的新矩阵，新矩阵和原始矩阵不仅维数相同，而且相同位置的元素的值是接近的。通过求出新矩阵和原始矩阵的差值平方并加上通过矩阵分解得到的两个矩阵的矩阵范数作为损失函数，最后通过梯度下降算法迭代优化该损失函数从而得到最近似原始数据矩阵的新矩阵。

## 3.2基于矩阵分解的精细化电能分解原理

电量是一个非负的量，也就是说不仅矩阵分解前的原始矩阵是非负，而且矩阵分解后的两个矩阵也是非负的，因此所用到的矩阵分解是非负矩阵分解。给定原始数据矩阵A∈R+n\*m，寻找非负矩阵U∈R+n\*r和非负矩阵V∈R+r\*m，其中r远小于n和m，使得A≈UV。A是电量矩阵，其形式如下:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mw\_1 | mw\_2 | … | mw\_12 | … | aggregate\_1 | … | aggregate\_12 |
| 0 | 5.6 | null | … | 6.5 | … | 708.8 | … | 650.4 |
| 1 | 4.0 | 8.2 | … | 3.6 | … | 609.4 | … | 780.6 |
| 2 | null | 4.5 | … | null | … | 500.0 | … | 456.2 |
| … | … | … | … | … | … | … | … | … |

A的行是每个家庭，列是12个月的总用电量和每个负载12个月的用电量，其中12个月的总用电量均有数值，而每个负载12个月的用电量有些为空。U为家庭的潜在因子或者是影响家庭用电的属性，例如可以是收入水平、房屋面积或者其他的因素。V为每个月的潜在因子或者是和某个负载使用有关的属性，例如可以是季节。UV是近似A的矩阵，而UV矩阵每个负载12个月的用电量均不为空，其形式如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mw\_1 | mw\_2 | … | mw\_12 | … | aggregate\_1 | … | aggregate\_12 |
| 0 | 5.5 | 4.6 | … | 6.4 | … | 700.4 | … | 658.7 |
| 1 | 3.7 | 8.1 | … | 3.2 | … | 607.1 | … | 760.8 |
| 2 | 3.4 | 4.6 | … | 7.0 | … | 490.6 | … | 456.0 |
| … | … | … | … | … | … | … | … | … |

从而补全了A矩阵中每个负载12个月的空缺值。所以本实验是一个最优化问题，目的在于找出最近似A的UV。

## 3.3基于矩阵分解的精细化电能分解训练

### 3.3.1数据归一化处理

在数据分析之前，我们通常需要先将数据标准化（normalization）。数据标准化处理主要包括数据同趋化处理和无量纲化处理两个方面。其中最典型的就是数据的归一化处理，即将数据统一映射到[0,1]区间上，其中本实验使用的是min-max标准化（Min-Max Normalization），具体公式如下：

………………………………………（3-1）

其中max为样本数据的最大值，min为样本数据的最小值。

### 3.3.2损失函数

损失函数（loss function）顾名思义是用量化的方式来评估模型的预测值f(x)与真实值Y的偏离程度，通常使用L(Y, f(x))来表示。根据不同的实际问题，损失函数有所不同，例如分类问题可以使用交叉熵，而回归问题使用均方差。通常来说，如果一个模型的损失函数越小，那么这个模型的拟合程度就越好。但是，模型对数据的拟合程度越好并不说明这个模型的性能越好，也就是说可能会出现过拟合的问题，所以评估一个模型的性能应该使用测试数据来评估。损失函数通常可以表示成如下式子：

+ ………………..（3-2）

其中，前面的均值函数表示的是经验风险函数，L代表的是损失函数，后面的Φ是正则化项（regularizer）。

其中本实验用的是平方损失函数（最小二乘法, Ordinary Least Squares）和L2范数的和，损失函数的具体公式如下：

……………………………..（3-3）

其中A为数据矩阵，U和V为分解得到的两个矩阵，和为系数。

## 3.4代码实现思路

代码实现需要的数据有：预处理过的训练数据data，学习率lr，训练步数steps，矩阵分解时需要引入的特征因子rank即3.2所提到的r。

1. 把训练数据data转化成tensorflow所能处理的数据格式A
2. 利用numpy的随机类随机生成两个矩阵并转成tensor格式，分别为 U和V，其中U为矩阵分解得到的第一个矩阵，U矩阵的行数为data的行数也就是训练数据的条数，列数为特征因子r的值，V为矩阵分解得到的第二个矩阵，V矩阵的行数为特征因子r的值，列数为data的列数也就是训练数据属性值的数目
3. 利用tensorflow设置U和V矩阵中每个元素的值均非负
4. 利用tensorflow的矩阵乘法函数对U和V做矩阵乘法，得到矩阵UV
5. 利用tensorflow构造损失函数loss，loss由两部分组成，一部分为训练数据A和UV的欧几里得距离，另一部分是矩阵U的Frobenius范数和矩阵V的Frobenius范数
6. 利用tensorflow设置模型的学习率为lr
7. 利用tensorflow自带的梯度下降算法迭代steps次不断优化损失函数loss也就是最小化loss
8. 程序运行完得到的矩阵U和V即为矩阵分解得到的两个矩阵， UV即为所求矩阵也就是和A近似的矩阵

## 3.5本章小结

先介绍了本章实验的总体思路，本章实验的总体思路是先对原始数据做归一化处理，通过归一化处理，能使实验数据的不同向量具有可比性和使实验更容易收敛。接着对原始数据做非负矩阵分解，把原始数据矩阵分解为两个的矩阵，这两个矩阵是相对低维的矩阵，这样可以实现矩阵降维的目的。然后对分解得到的两个相对低维的矩阵做矩阵乘法就可以得出一个与原始数据矩阵相似的新矩阵，新矩阵和原始矩阵不仅维数相同，而且相同位置的元素的值是接近的。通过求出新矩阵和原始矩阵的差值平方并加上通过矩阵分解得到的两个矩阵的矩阵范数作为损失函数，最后通过梯度下降算法迭代优化该损失函数从而得到最近似原始数据矩阵的新矩阵。

然后分别对总体思路中的每个要点进行了解释，也就是对非负矩阵分解应用在本实验上的结构做了具体的解释，介绍了数据归一化的目的和详细介绍了本实验所对应的损失函数，最后介绍了实现本实验的代码思路。

# 4基于矩阵分解、回归模型、深度学习用电行为分析

## 4.1总体思路

本章实验的总体思路是先对原始数据做归一处理，通过归一化处理，能使实验数据的不同向量具有可比性和使实验更容易收敛。接着分别用非负矩阵分解、线性回归、岭回归、套索回归以及神经网络建立模型并对训练数据进行训练，通过输入预处理过的数据并输出预测的数据，然后计算模型的预测值和真实值的差距也就是得到损失函数，通过梯度下降算法优化损失函数从而得到优化后的权重参数，然后使用同样的过程对训练得到的权重进行验证，也就是把新的数据输入到模型中得到预测的结果。

## 4.2用电行为分析的矩阵分解、回归模型以及深度学习模型结构

### 4.2.1非负矩阵分解

电量是一个非负的量，也就是说不仅矩阵分解前的原始矩阵是非负，而且矩阵分解后的两个矩阵也是非负的，因此所用到的矩阵分解是非负矩阵分解。给定原始数据矩阵A∈R+n\*m，寻找非负矩阵U∈R+n\*r和非负矩阵V∈R+r\*m，其中r远小于n和m，使得A≈UV。A是电量矩阵，其形式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | house\_num\_rooms | area | num\_occupants | aggregate\_1 | … | aggregate\_12 |
| 0 | null | 200.6 | 4 | 708.8 | … | 650.4 |
| 1 | 4.0 | 180.2 | null | 609.4 | … | 780.6 |
| 2 | null | null | 3 | 500.0 | … | 456.2 |
| … | … | … | … | … | … | … |

A的行是每个家庭，列是12个月的总用电量和家庭人数、房间数、房屋面积这三种家庭静态属性的值，其中12个月的总用电量均有数值，而家庭人数、房间数、房屋面积这三种家庭静态属性的值有些为空。U为家庭的潜在因子或者是影响家庭用电的属性，例如可以是收入水平、房屋面积或者其他的因素。V为每个月的潜在因子或者是和某个负载使用有关的属性，例如可以是季节。UV是近似A的矩阵，而UV矩阵每个家庭静态属性的值均不为空，其形式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | house\_num\_rooms | area | num\_occupants | aggregate\_1 | … | aggregate\_12 |
| 0 | 3.5 | 198.6 | 4.0 | 700.6 | … | 620.6 |
| 1 | 4.0 | 156.4 | 4.2 | 598.4 | … | 775.5 |
| 2 | 4.3 | 200.8 | 3.5 | 480.5 | … | 480.8 |
| … | … | … | … | … | … | … |

从而补全了A矩阵中每个家庭静态属性的空缺值。所以本实验是一个最优化问题，目的在于找出最近似A的UV。

### 4.2.2回归模型结构

#### 4.2.2.1基于线性回归模型结构

线性回归（Linear regression）是利用称为线性回归方程的[最小二乘](https://zh.wikipedia.org/wiki/%E6%9C%80%E5%B0%8F%E4%BA%8C%E4%B9%98%E6%B3%95" \o "最小二乘法)函数对一个或多个[自变量](https://zh.wikipedia.org/wiki/%E8%87%AA%E5%8F%98%E9%87%8F)和[因变量](https://zh.wikipedia.org/w/index.php?title=%E5%9B%A0%E5%8F%98%E9%87%8F&action=edit&redlink=1)之间关系进行建模的一种[回归分析](https://zh.wikipedia.org/wiki/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90)。这种函数是一个或多个称为回归系数的模型参数的线性组合。

#### 4.2.2.2基于岭回归模型结构

本实验的岭回归模型是在线性回归模型的基础上加入了L2正则化项。

#### 4.2.2.3基于套索回归模型结构

本实验的岭回归模型是在线性回归模型的基础上加入了L1正则化项。

### 4.2.3深度学习模型结构

本实验的神经网络结构时三层网络结构，其中前两层为隐藏层，最后一层为输出层。前两层隐藏层在输出时均使用relu函数（Rectified Linear Unit修正线性单元）作为激活函数，最后一层也就是输出层不使用激活函数。

## 4.3用电行为分析的矩阵分解、回归模型以及深度学习模型训练

### 4.3.1数据归一化处理

在数据分析之前，我们通常需要先将数据标准化（normalization）。数据标准化处理主要包括数据同趋化处理和无量纲化处理两个方面。其中最典型的就是数据的归一化处理，即将数据统一映射到[0,1]区间上，其中本实验使用的是min-max标准化（Min-Max Normalization），具体公式如下：

………………………………………（4-1）

其中max为样本数据的最大值，min为样本数据的最小值。

### 4.3.2损失函数

损失函数（loss function）顾名思义是用量化的方式来评估模型的预测值f(x)与真实值Y的偏离程度，,通常使用L(Y, f(x))来表示，通常来说，如果一个模型的损失函数越小，那么这个模型的拟合程度就越好。但是，模型对数据的拟合程度越好并不说明这个模型的性能越好，也就是说可能会出现过拟合的问题，所以评估一个模型的性能应该使用测试数据来评估。损失函数通常可以表示成如下式子：

+ …………（4-2）

其中，前面的均值函数表示的是经验风险函数，L代表的是损失函数，后面的Φ是正则化项（regularizer）。

* 非负矩阵分解

非负矩阵分解使用的是平方损失函数（最小二乘法, Ordinary Least Squares）和L2范数的和，损失函数的具体公式如下：

………………………（4-3）

其中A为数据矩阵，U和V为分解得到的两个矩阵，和为系数。

* 线性回归

线性回归使用的损失函数仅为平方损失函数，也就是真实值与预测值差的平方，具体公式如下：

……………………………..（4-4）

其中为真实值，为预测值。

* 岭回归

岭回归使用的损失函数为平方损失函数和L2正则化项的和。平方损失函数是真实值与预测值差的平方。加入L2正则化项后可以在拟合过程中倾向于让权值尽可能小，最后构造一个所有参数都比较小的模型。可以设想一下对于一个线性回归方程，若参数很大，那么只要数据偏移一点点，就会对结果造成很大的影响；也就是说，如果我们能使模型的参数都比较小，那么数据中的噪声就不会对实现结果造成比较大的影响，这样一来模型的泛化能力就会比较强。具体公式如下：

………………………..（4-5）

其中为真实值，为预测值，W为模型的权重。

* 套索回归

套索回归使用的损失函数为平方损失函数和L1正则化项的和。平方损失函数是真实值与预测值差的平方。之所以要加入L1正则化项，是因为加入L1正则化有助于生成一个稀疏权值矩阵。稀疏权值矩阵中大部分的元素都为0，这样就可以进行特征选择。而之所要进行特征选择，是因为输入数据的所有属性值不一定都是有用的，也就是不应该对结果产生影响的。具体公式如下：

………………………（4-6）

其中为真实值，为预测值，W为模型的权重。

* 神经网络

神经网络使用的损失函数仅为平方损失函数，也就是真实值与预测值差的平方，具体公式如下：

………………………………..（4-7）

其中为真实值，为预测值。

## 4.3代码实现思路

* 非负矩阵分解

代码实现需要的数据有：预处理过的训练数据data，学习率lr，训练步数steps，矩阵分解时需要引入的特征因子rank即3.2所提到的r。

1. 把训练数据data转化成tensorflow所能处理的数据格式A
2. 利用numpy的随机类随机生成两个矩阵并转成tensor格式，分别为U和V，其中U为矩阵分解得到的第一个矩阵，U矩阵的行数为data的行数也就是训练数据的条数，列数为特征因子r的值，V为矩阵分解得到的第二个矩阵，V矩阵的行数为特征因子r的值，列数为data的列数也就是训练数据属性值的数目
3. 利用tensorflow设置U和V矩阵中每个元素的值均非负
4. 利用tensorflow的矩阵乘法函数对U和V做矩阵乘法，得到矩阵UV
5. 利用tensorflow构造损失函数loss，loss由两部分组成，一部分为训练数据A和UV的欧几里得距离，另一部分是矩阵U的Frobenius范数和矩阵V的Frobenius范数
6. 利用tensorflow设置模型的学习率为lr
7. 利用tensorflow自带的梯度下降算法迭代steps次不断优化损失函数loss也就是最小化loss
8. 程序运行完得到的矩阵U和V即为矩阵分解得到的两个矩阵， UV即为所求矩阵也就是和A近似的矩阵

* 回归模型

线性回归、岭回归和套索回归的代码实现思路基本相同，唯一不同是在构造损失函数时是否要加入正则化项。

代码实现需要的数据有：预处理过的训练数据data，预处理过的标签数据data\_label，学习率lr，训练步数steps。

1. 把训练数据矩阵data和标签数据data\_label转置并转成tensorflow所能处理的数据格式tensor，分别记为X和Y
2. 利用numpy的随机类随机生成两个矩阵并转成tensor格式，分别为W和b，其中W为模型的权重矩阵，b为模型的偏置矩阵
3. 利用tensorflow的矩阵乘法函数对W和X做矩阵乘法，得到WX，然后加上偏置b，可得到训练数据的预测值pred\_Y
4. 利用tensorflow构造损失函数loss，如果是线性模型，loss只由真实值Y和预测值pred\_Y的欧几里得距离构成。如果是岭模型，则由两部分组成，一部分为真实值Y和预测值pred\_Y的欧几里得距离，另一部分是L2正则化项。如果是岭模型，则由两部分组成，一部分为真实值Y和预测值pred\_Y的欧几里得距离，另一部分是L1正则化项。
5. 利用tensorflow设置模型的学习率为lr
6. 利用tensorflow自带的梯度下降算法迭代steps次不断优化损失函数loss也就是最小化loss
7. 程序运行完得到的矩阵W和b即为模型的两个参数

* 神经网络

本实验的神经网络的层数为三层，前两层是隐藏层，后一层为输出层。

代码实现需要的数据有：预处理过的训练数据data，预处理过的标签数据data\_label，学习率lr，训练步数steps。

1. 把训练数据矩阵data和标签数据data\_label转置并转成tensorflow所能处理的数据格式tensor，分别记为X和Y
2. 利用numpy的随机类随机生成六个矩阵并转成tensor格式，分别为W1、W2、W3和b1、b2、b3，其中W为模型的权重矩阵，b为模型的偏置矩阵
3. 利用tensorflow计算第一层W1×X＋b1，并把结果输入到tensorflow的激活函数relu中得到h1，接着计算第二层W2×X＋b2，并把结果输入到tensorflow的激活函数relu中得到h2，然后计算第三层W3×X＋b3，可得到训练数据的预测值pred\_Y
4. 利用tensorflow构造损失函数loss，loss由真实值Y和预测值pred\_Y的欧几里得距离构成。
5. 利用tensorflow设置模型的学习率为lr
6. 利用tensorflow自带的梯度下降算法迭代steps次不断优化损失函数loss也就是最小化loss
7. 程序运行完得到的矩阵W和b即为模型的六个参数

## 4.4本章小结

先介绍了本章的总体思路，总体思路是先对原始数据做归一处理，通过归一化处理，能使实验数据的不同向量具有可比性和使实验更容易收敛。接着分别用非负矩阵分解、线性回归、岭回归、套索回归以及神经网络建立模型并对训练数据进行训练，通过输入预处理过的数据并输出预测的数据，然后计算模型的预测值和真实值的差距也就是得到损失函数，通过梯度下降算法优化损失函数从而得到优化后的权重参数，然后使用同样的过程对训练得到的权重进行验证，也就是把新的数据输入到模型中得到预测的结果。

然后分别对总体思路中的每个要点进行了解释，也就是对本实验的非负矩阵分解、线性回归、岭回归、套索回归以及神经网络的结构进行了解释，介绍了数据归一化的目的和详细介绍了本实验所对应的损失函数，最后介绍了实现本实验的代码思路。

# 5实验结果与分析

## 5.1实验环境与配置

本实验使用的编程语言是Python语言，Python语言相对其他编程语言，有其几个突出的优势，一个是Python代码简洁，另一个是编程效率高，编程效率高是指Python的一行代码能实现其他编程语言的数行代码，最重要的一个优势是Python有很多强大的扩展库，例如Numpy、Pandas，这些扩展库能高效率地实现很多功能。本实验使用的深度学习框架TensorFlow它相对于其他框架，它有几个突出的优势，一个是以图来组织计算节点，这样会使构建的学习网络结构更加清晰，步骤更得简单，另一个是能进行并行计算，这样会提高计算效率，最重要的一个优势是能智能选择计算环境，具体地说，就是能在CPU和GPU之间自由地选择。安装TensorFlow包括以下步骤：

（1）安装Python

从官网下载Python3.5安装包并安装，安装完成后配置环境变量即可。

（2）安装sublime

从官网下载最新版本的sublime并安装。

（3）安装TensorFlow

本实验安装的是TensorFlow-gpu1.6，之所以选择安装gpu版本的TensorFlow，是因为gpu版本的TensorFlow计算速度相比cpu版本的会快至少20倍。本实验使用的是pip安装，进入命令行窗口，输入命令pip install tensorflow-gpu==1.6即可安装。如果要使用gpu版本的TensorFlow，还需安装CUDA以及cuDNN并进行相应的配置。

（4）安装Numpy

Numpy是Python的一个科学计算扩展库，Numpy不仅给各种数据提供了一个中转接口，也就是能把非Python能使用的数据格式转成Python能使用的数据格式，还会使程序的运算速度比未使用Numpy的Python原生程序要快很多。本实验使用的是pip安装，进入命令行窗口，输入命令pip install numpy即可安装。

（5）安装Pandas

Pandas 是Python的一个数据格式扩展库。Pandas提供了两种数据格式，一种是DataFrame，另一种是Series。其中，DataFrame类似于数据库的数据表，能进行各种数据运算，例如选择，合并等。Series类似于Python的字典，但有一个不同于Python字典的特点是Series是有序的，而Python字典是无序的。Pandas为预处理数据提供了很大的便利。本实验使用的是pip安装，进入命令行窗口，输入pip install pandas即可安装。

（6）安装matplotlib

matplotlib 是Python编程语言及其数值数学扩展包 [NumPy](https://zh.wikipedia.org/wiki/NumPy" \o "NumPy)的可视化操作界面。它为利用通用的图形用户界面工具包，如Tkinter, wxPython, Qt或GTK+向应用程序嵌入式绘图提供了应用程序接口（API）。本实验使用的是pip安装，进入cmd窗口，利用命令pip install matplotlib即可安装。

## 5.2数据集来源和预处理

### 5.2.1实验数据集来源

我们使用公开的数据集（Parson et al。2015）进行评估。 数据端口是家庭能源数据最大的公共数据集。 数据端口数据集包含2015年美国德克萨斯州奥斯汀的586个家庭的数据。每一个家庭的数据包含一年12个月每个月的总用电量以及每种负载每个月的用电量。 数据集还包含静态家庭特性，如家庭区域面积、家庭人数、家庭的房间数量。

### 5.2.2实验数据预处理

根据实验研究的内容不同对数据做不同的预处理。对于用电分解实验，实验所需的数据形式应为一年12个月的总用电量均不能为空，还有可以为空的六种负载的12个月的数据，根据这个要求对数据进行筛选，筛剩的数据为105个家庭的数据。对于用户用电行为分析实验，实验所需的数据形式为一年12个月的总用电量均不能为空，还有唯一为空的3种静态家庭属性，根据这个要求对数据进行筛选，筛剩的数据为346个家庭的数据。

## 5.3实验评估指标

* 评估指标

六种负载在家庭消耗电量的占比明显不同，有些负载仅占总用电量的百分之五，而有些负载却占总用电量的百分之五十，例如oven的实际值为1个单位，而分解值为2个单位，其实际值和分解的差的绝对值为1个单位，fridge的实际值为50个单位，而分解值为50个单位，其实际值和分解值的差的绝对值为10个单位，但是从矩阵分解的角度来说，很明显对于fridge的分解效果比对于oven的分解效果更好。同样，在预测用户的家庭静态属性特征时，有些静态特征的值很小，例如家庭人数，而有些静态特征的值很大，例如房屋面积。假如家庭的实际值为3个单位，而预测值为6个单位，其实际值和分解的差的绝对值为3个单位，房屋面积的实际值为100个单位，而预测值为120个单位，其实际值和分解的差的绝对值为20个单位，所以如果用实际值和分解值或者预测值的差的绝对值作为评估指标，那将不能很好地对实验结果进行客观的分析。

因此，本文实验的评估指标为RMSE，具体的计算公式如下：

…………………………………..（5-1）

## 5.4实验结果

* 基于矩阵分解的精细化电能分解

表5-1及图5-1是使用非负矩阵分解算法对电量进行1000000次迭代分解之后求得的每一个负载的RMSE，表5-2及图5-2是使用非负矩阵分解算法对电量进行2000000次迭代分解之后求得的每一个负载的RMSE。

表5-1矩阵分解在六种负载上的误差

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| appliance | mw | oven | hvac | fridge | dw | wm |
| nnmf | 0.389364 | 0.367464 | 0.312977 | 0.052234 | 0.257017 | 0.481416 |

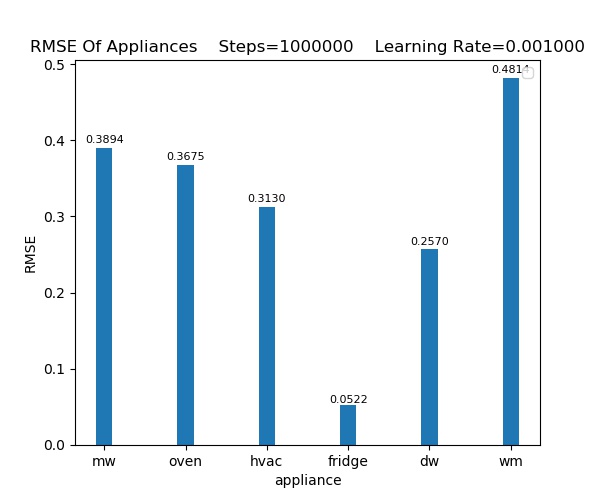


图5-1 矩阵分解在六种负载上的误差

表5-2矩阵分解在六种负载上的误差

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| appliance | mw | oven | hvac | fridge | dw | wm |
| nnmf | 0.339675 | 0.380522 | 0.167407 | 0.048613 | 0.212701 | 0.517429 |

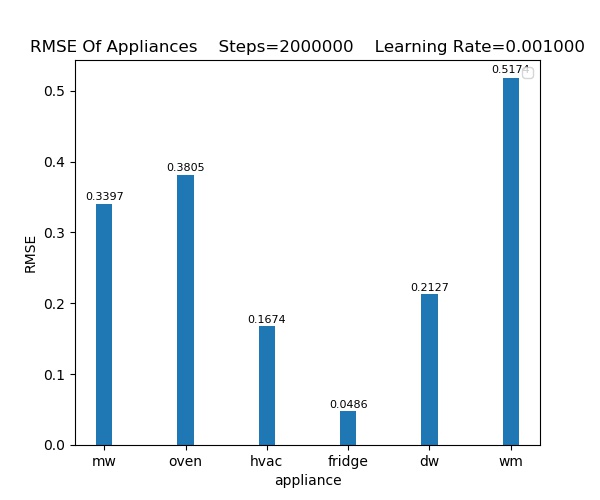


图5-2 矩阵分解在六种负载上的误差

* 基于回归模型以及深度学习的用电行为分析

表5-3及图5-3是使用线性回归、岭回归、套索回归以及神经网络算法进行1000000次训练之后预测的每一个家庭静态属性特征的RMSE，表5-4及图5-4是使用非矩阵分解、线性回归、岭回归、套索回归以及神经网络算法进行1000000次训练之后预测的每一个家庭静态属性特征的RMSE。

表5-3几种算法在在家庭静态特征上的误差

|  |  |  |  |
| --- | --- | --- | --- |
| feature | house\_num\_rooms | area | num\_occupants |
| nnmf\_feature | 0.365105 | 0.00131 | 0.3514 |
| linear\_reg | 0.299753 | 0.730144 | 0.068106 |
| ridge\_reg | 0.315264 | 0.726658 | 0.079343 |
| lasso\_reg | 0.355826 | 0.722763 | 0.10964 |
| neural\_network | 0.354138 | 0.675269 | 0.109307 |

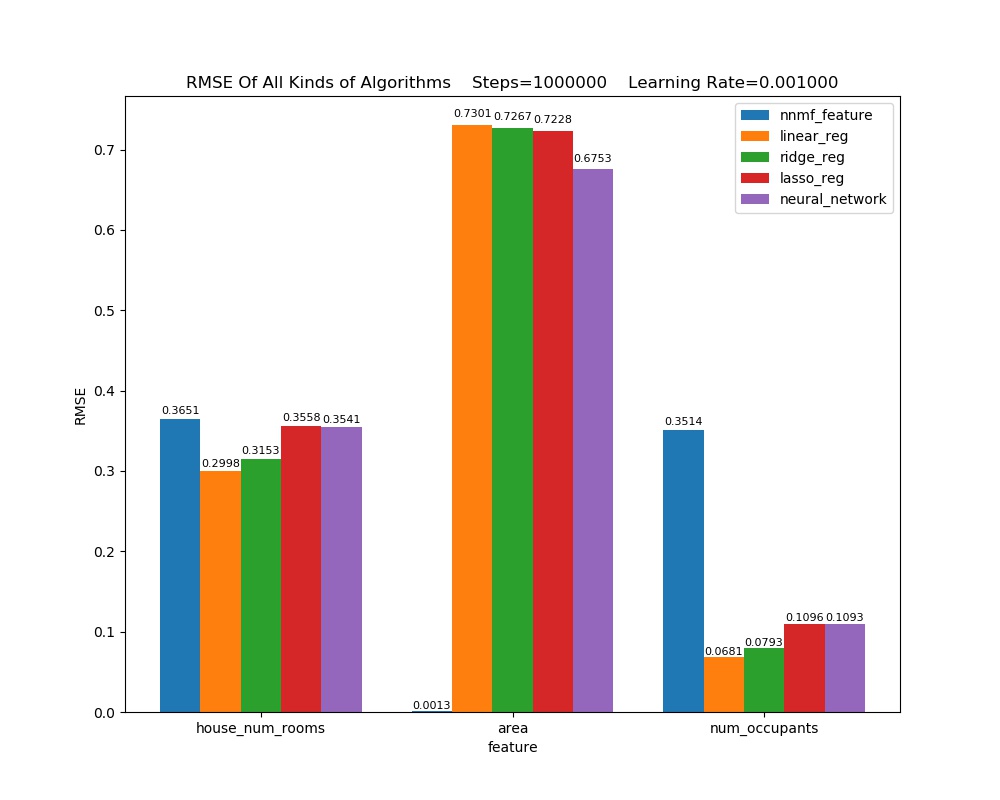


图5-3 几种算法在在家庭静态特征上的误差

表5-4几种算法在在家庭静态特征上的误差

|  |  |  |  |
| --- | --- | --- | --- |
| feature | house\_num\_rooms | area | num\_occupants |
| nnmf\_feature | 0.368271 | 0.000494 | 0.349492 |
| linear\_reg | 0.299753 | 0.730144 | 0.068125 |
| ridge\_reg | 0.315264 | 0.726658 | 0.079343 |
| lasso\_reg | 0.34701 | 0.722777 | 0.111187 |
| neural\_network | 0.354138 | 0.675269 | 0.109307 |

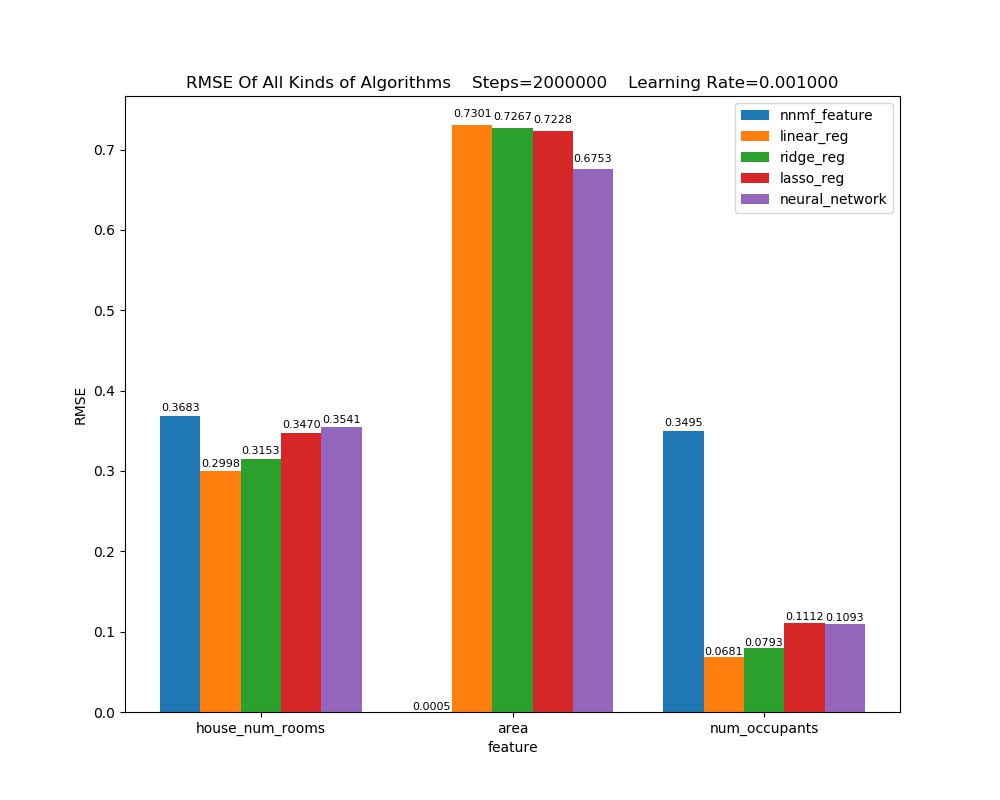


图5-4 几种算法在在家庭静态特征上的误差

## 5.5实验结果分析

* 基于矩阵分解的精细化电能分解

对电能分解分别做了两个实验，一个是学习率为0.001且迭代次数为1000000次的矩阵分解，另一个是学习率为0.001且迭代次数为2000000次的矩阵分解。从实验结果可以看出，当迭代次数为1000000次时已基本收敛，即使迭代更多的次数，实验结果也不相上下。此外，也能看出矩阵分解算法对fridge和dw的效果最好，而对oven和wm的分解效果相对其他负载较差。从以上结果可以看出，基于非嵌入式的电能分解优于基于嵌入式的电能分解，前者相对于后者不仅硬件成本还有人工成本低，而且前者的准确率比后者高，因为后者存在测量收集误差和硬件误差。

* 基于矩阵分解、回归模型以及深度学习的用电行为分析

对用电行为也分别用矩阵分解、三种回归模型算法以及神经网络算法做了一共10个实验，分别是学习率为0.001且迭代次数为1000000次的矩阵分解、学习率为0.001且迭代次数为2000000次的矩阵分解、学习率为0.001且训练迭代次数为1000000次的线性回归、学习率为0.001且训练迭代次数为2000000次的线性回归、学习率为0.001且训练迭代次数为1000000次的岭回归、学习率为0.001且训练迭代次数为2000000次的岭回归、学习率为0.001且训练迭代次数为1000000次的套索回归、学习率为0.001且训练迭代次数为2000000次的套索回归、学习率为0.001且训练迭代次数为1000000次的神经网络、学习率为0.001且训练迭代次数为2000000次的神经网络。从实验结果可以看出，当迭代次数为1000000次时已基本收敛，即使迭代更多的次数，实验结果也不相上下。此外，也能看出矩阵分解、三种回归模型以及神经网络对房间数的预测最为准确，而对房屋面积的预测相对较差。

## 5.6本章小结

本章首先介绍了实验所使用的编程语言以及所使用到扩展库，接着介绍了实验环境的配置。其次对实验所用数据的来源以及对实验所要做的不同预处理步骤做了简单介绍。再次接着介绍了实验所选用的实验评估指标和贴上了实验的核心代码。最后对本论文所提出的基于矩阵分解的精细化电能分解和基于矩阵分解、回归模型以及深度学习的用电行为分析的实验结果进行展示和分析，从实验结果可以看出，对于电量的矩阵分解效果很好，说明非负矩阵分解算法应用在电量的分解问题上是适合的，且对分解过程具有很强的解释性。另外，对于用户的用电行为分析也就是对用户家庭的静态属性特征进行预测的效果也很好，说明非负矩阵分解、回归模型和深度学习在预测用户的家庭静态特征这个问题上也是合适的。

# 总结与展望

* 本文工作总结

家庭用户消耗的用电量占总用电的三分之一以上。智能电网用户的用电数据隐含着丰富的信息，可以用来研究用户行为模式，提高能源利用率。然而，目前存在的电能分解方法需要借以在家庭中安装嵌入式传感器从而会需要消耗昂贵的硬件资源。基于上述的嵌入式问题，本论文通过分析用电数据中总用电量和用电器用电量之间的关系，并建立非嵌入式的模型对用户的总用电量进行分解，预测用户各用电器的用电量。同时根据用户的用电数据分析用户的家庭属性，包括家庭面积、房间数和家庭人数等。根据本文的研究工作，主要包括以下几个方面：

（1）分析Dataport公开数据集，对数据进行预处理

由于Dataport数据集中存在一些无用的数据，也就是不符合本论文实验要求的数据，所以在做实验之前要先对数据做处理，本论文利用python语言、numpy和panda数据处理库，对Dataport公开数据集的数据进行预处理，筛选和处理成本实验需要的特定数据格式。

（2）采用tensorflow框架实现矩阵分解进行电能分解

利用tensorflow库搭建矩阵分解的模型框架，通过输入经过预处理的数据，最终通过优化损失函数达到分解矩阵也就是补全矩阵中缺失数据的目的。

（3）采用tensorflow框架实现回归模型、神经网络进行属性特征预测

利用tensorflow库搭建回归模型、神经网络的模型框架，通过输入经过预处理的数据，最终通过优化损失函数达到预测家庭属性特征的目的。

* 未来工作展望

本论文对基于矩阵分解的精细化电能分解和基于回归模型以及深度学习的用电行为分析进行了研究，研究结果显示非负矩阵分解应用在电量分解上是合适，非负矩阵分解后得到的家庭静态属性特征时合适的，回归模型以及神经网络在预测家庭静态属性特征也是合适的，但由于个人学识和课题研究时间所限，在以下方面还存在一些不足，有待更深入的研究：

（1）回归模型以及神经网络往往需要足够的数据来进行训练，以避免过拟合而对新的测试数据不能有很好的预测效果。而本论文所进行实验所用到的数据仅500多条数据，而且在经过数据预处理也就是进行筛选之后，所剩下的可用数据仅仅是348条，这对于线性回归、岭回归、套索回归以及神经网络来说，这些数据是匮乏的。由于数据的匮乏，本实验的结果不能达到更好的效果。不过，如果将来有机会将会对训练数据进行扩展补充例如可以通过收集途径或者利用对抗神经网络生成更多的训练数据，这样将会避免由于数据的不足所造成的欠拟合也就是将会增强模型的泛化能力。

（2）由于数据的采样问题，例如采样数据是每个月的六种负载的用电量而不是每天的负载的用电量，这会使本来每天各种负载的使用量的家庭，但在每个月的总量上相同或者接近，也就是因为采样问题模糊了家庭的特殊的时间段使用不同的用电器，换句话说是因为采样问题使实验数据丢失了每条数据所蕴含的特征，所以在预测家庭静态特征方面会有一定的不足。由于客观的原因，本实验所使用的的数据从客观的角度来说是不够准确的，所以，如果将来有机会会收集或者获取更准确的用户每天的用电数据，这样将会使预测家庭静态特征的效果更好。

（3）由于计算机计算资源的限制，具体来说就是CPU、GPU和允许内存的限制，不能在较短的时间内得出实验结果，从而不能很好地根据实验分析实验结果是偏差过大还是方差过大又或者是两者都过大对程序进行更多的调参，在有限的时间只能进行超参数的粗调而不能进行微调，这就很有可能错失某些对实验结果有重要影响的超参数点，所以如果将来有更充足的计算资源，将会进行更多的调参工作，相信效果会比现在的更好。

# 参考文献

[1] 汪鹏.非负矩阵分解:数学的奇妙力量[J].计算机教育,2004(10):38-40.

[2] 栗茂林,梁霖,陈元明,徐光华,何康康.基于聚类优化的非负矩阵分解方法及其应用[J].中国机械工程,2018,29(06):720-725.

[3] 何梦娇,杨燕,王淑营.一种基于非负矩阵分解的聚类集成算法[J].计算机科学,2017,44(09):58-61.

[4] 孙静,蔡希彪,孙福明.基于特征融合的多约束非负矩阵分解算法[J].计算机应用,2017,37(10):2834-2840.

[5] 何海洋,王勇,蔡国永.基于用户类别偏好相似度和联合矩阵分解的推荐算法[J].数据采集与处理,2018,33(01):179-185.

[6] 王建芳,刘冉东,刘永利.一种带偏置的非负矩阵分解推荐算法[J].小型微型计算机系统,2018,39(01):69-73.

[7] 张勇进. 带弹性网的稀疏主成分分析[D].华中科技大学,2006.

[8] Combined model based on optimized multi-variable grey model and multiple linear regression[J].Journal of Systems Engineering and Electronics,2011,22(04):615-620.

[9] Statistical Inference for Partially Linear Regression Models with Measurement Errors[J].Chinese Annals of Mathematics,2008(02):207-222.

[10] Xiuli WANG,Mingqiu WANG.Combination of Nonconvex Penalties and Ridge Regression for High-Dimensional Linear Models[J].Journal of Mathematical Research with Applications,2014,34(06):743-753.

[11] Huan Luo. A Prediction Method Based on Improved Ridge Regression[A]. The Institute of Electrical and Electronics Engineers、IEEE Beijing Section.Proceedings of 2017 IEEE 8th International Conference on Software Engineering and Service Science[C].The Institute of Electrical and Electronics Engineers、IEEE Beijing Section:,2017:4.

[12] 熊红凯,高星,李劭辉,徐宇辉,王涌壮,余豪阳,刘昕,张云飞.可解释化、结构化、多模态化的深度神经网络[J].模式识别与人工智能,2018,31(01):1-11.

[13] 王月岭. 基于支持向量机与概率输出网络的深度学习[D].西安理工大学,2017.

[14] 刘晓曈. 基于深度学习的分类预测方法研究及应用[D].东南大学,2017.

[15] Real estate appraisal system based on GIS and BP neural network[J].Transactions of Nonferrous Metals Society of China,2011,21(S3):626-630.

[16] 焦李成,杨淑媛,刘芳,王士刚,冯志玺.神经网络七十年:回顾与展望[J].计算机学报,2016,39(08):1697-1716.

# 致谢

时光荏苒，转眼之间四年大学生活即将结束，在大学生活结束之际，毕设以及毕设论文使我在大学思念收获最大的东西，经过一学期的不断努力研究，查阅各种资料，不断的尝试，不断的失败，在完成毕业论文期间遇到了很多问题，但是经过自己不断的钻研和在老师的帮助下，最终解决了碰到的很多问题，此次的毕设论文使我对大学四年学到的很多专业课之时进行了进一步的巩固，对自己的耐心也有了大的提升，在论文过程中碰到了很多问题，在解决问题的过程中我学到了很多书本外的知识，体验到了在遇到问题时一遍又一遍的经过尝试最后解决问题的那种喜悦心情。

在即将毕业之际，我想对在我学习生活中有帮助的人进行感谢。

首先特别感谢我的导师——王洪涛老师。在论文写作的各个环节中，王老师不厌其烦地对我提供各方面的指导，从论文的选题、方案的确定、参考文献的搜集整理、实验过程的编程指导、论文的书写直到最后的定稿。在此过程中，我不仅掌握了更多的理论知识，还提高了编程实践能力以及更加熟悉编程工具的使用。除此之外，在论文的书写方面也得到了很大的提升。王老师在工作中以及对我的指导过程中体现的孜孜不倦精神使我非常敬佩，使我明白了老师在学识方面对学生无尽的付出，而且老师丰富的知识水平使我十分钦佩，让我明白知识的海洋永无止尽，我所要学习的知识还有很多。在论文完成之际，谨向王老师致以最崇高的敬意和最衷心的感谢！

其次，我要感谢我的各位专业课老师。专业课老师知识渊博，教学态度严谨。在我遇到不懂的问题时，老师会耐心引导我一步一步理解问题。在大学的四年里，他们无私奉献地传授知识，从他们的课堂上我学到了丰富的理论知识以及丰富的编程能力。正因为有了这些专业知识的基础，我才能形成扎实的计算机理论体系和，我才能在导师的指导下顺利完成毕业设计。

再次，我要感谢陪伴我大学四年的朋友们。在大学的四年里，我们一起上课、一起写程序、一起讨论问题、一起参加课余活动、一起畅聊人生。在我生活上遇到挫折的时候，我的朋友们会鼓励我坚定信心、坚信柳暗花明又一村，在我学习上遇到难题时，我的朋友们会和我一起讨论一起解决难题。

最后，我要感谢最爱我的家人，在我的求学路上，他们一直在背后默默地支持我，一直鼓励我克服一切困难、勇敢面对一切挫折。感谢父母对我二十多年的养育之恩，感谢他们通过自己的努力给我营造一个温暖的家。