



南京大學

本科畢業論文

院 系 數學系

專 業 統計學

題 目 稀疏學習優化特征選擇：

算法比較與性能評估

年 級 2020 學 號 201840045

學生姓名 王佳鵬

指導教師 王征宇 職 稱 副教授

提交日期 2024 年 6 月 6 日



南京大学本科毕业论文（设计） 诚信承诺书

本人郑重承诺：所呈交的毕业论文（设计）（题目：稀疏学习优化特征选择：算法比较与性能评估）是在指导教师的指导下严格按照学校和院系有关规定由本人独立完成的。本毕业论文（设计）中引用他人观点及参考资源的内容均已标注引用，如出现侵犯他人知识产权的行为，由本人承担相应法律责任。本人承诺不存在抄袭、伪造、篡改、代写、买卖毕业论文（设计）等违纪行为。

作者签名：

学号：

日期：

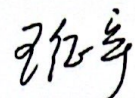
南 京 大 学

本科生毕业论文（设计、作品）指导教师评阅意见

指导教师评语：（不少于 300 字）

本论文研究稀疏学习算法在特征选择中的效果及应用。机器学习问题的训练数据极大，涉及特征极多，产生的待求解的数学问题（优化）问题维数很高，实际求解很困难。特征选择是降低问题规模的有效手段，而稀疏学习成为特征选择中解决高维数据问题的重要方法。从简单的 Lasso 算法至复杂的 $L_{2,1-2}$ 算法，本文探索稀疏学习里不同正则化格式的性能，比较它们在实验中的表现，并探讨其不同场景下的适用性。通过编程实现和数值实验，论文评估各算法的优劣，并通过可视化呈现其在特征选择任务中的实际效果。这项研究对于优化数据处理、提高决策效率具有重要意义，为实际问题的解决提供了有力支持。

该论文的研究课题是有实用价值的也有一定难度，研究内容丰富且有一定创新之处；论文写作规范、有条理、有逻辑；文献的掌握也是全面的；是一篇合格的本科毕业论文，论文体现了该学生在机器学习与优化算法方面比较扎实的理论基础、较强的编程能力与合格的论文写作能力。

指导教师签名： 

2024年5月21日

南 京 大 学

本科生毕业论文（设计、作品）评阅教师评阅意见

评阅教师评语：（不少于 300 字）

该论文以稀疏学习算法在特征选择中的效果及应用为研究对象，选题新颖且具有较高的学术和实用价值。作者通过对稀疏学习算法的深入研究，比较了不同正则化器在实验中的性能，并探讨了其在不同场景下的适用性。该研究有效应对了大数据时代下高维数据处理的挑战，对特征选择这一关键问题提出了科学且实用的解决方案。

首先，作者对稀疏学习算法的选择和分析具有很强的针对性，涵盖了稀疏学习的多个层面，充分展示了算法的广泛性和多样性。通过编程实现和数值实验，作者不仅验证了各算法在不同数据集上的性能，还通过可视化手段清晰地展示了各算法在特征选择任务中的实际效果。

其次，该论文在实验设计和结果分析上非常严谨。作者通过多次实验，比较了不同算法在多个数据集上的表现，确保了结果的可靠性和稳定性。此外，论文对实验结果的分析详尽，能够深入剖析各算法的优劣，找出其适用的具体场景和限制条件。这种细致入微的分析为后续研究和实际应用提供了宝贵的参考。

最后，该研究具有较高的应用价值。随着大数据技术的不断发展，如何高效地进行特征选择以提高模型的性能已成为亟待解决的问题。

作者的研究不仅在理论上丰富了稀疏学习算法的应用场景，也在实践中为解决问题提供了有力的支持。

总体而言，这是一篇内容充实、结构严谨、分析透彻的本科毕业论文。作者在稀疏学习算法的研究中展示了扎实的理论基础和较强的实验能力，研究结果具有较高的可信度和应用价值。

评阅教师签名：吴婷

2024年5月22日

南 京 大 学

本科生毕业论文（设计、作品）答辩记录、成绩评定

答辩记录：不同正则化引出的最优解，其稀疏度有什么差别？

回答： L_{asso} ，弹性网络， $l_{2,1}$ ， $l_{2,1-2}$ 所引出的最优解的稀疏度递增。

怎样看出准确性的差别？

回答：通过比较稀疏度可以看出复杂的方法可以更好的准确性。

答辩记录人签名：

答辩小组评语：（不少于 100 字）

论文选题新，方法实用可行，分析严密详实，在答辩过程中，该同学思路清晰，表述准确，正确详尽地回答了问题。答辩委员会经过无记名投票，一致通过答辩，建议授予学士学位。

答辩小组成员：

周菊萍 赵秋兰 吴婷、张红

成绩

89

组长签名：

周菊萍

答辩时间：2024年5月24日

南京大学本科生毕业论文（设计、作品）中文摘要

题目：稀疏学习优化特征选择: 算法比较与性能评估

院系：数学系

专业：统计学

本科生姓名：王佳鹏

指导教师（姓名、职称）：王征宇副教授

摘要：

本论文旨在研究稀疏学习算法在特征选择中的效果及应用。随着数据的爆炸性增长，稀疏学习成为特征选择中解决高维数据问题的重要方法。从简单的Lasso 算法至复杂的 $l_{2,1-2}$ 算法，我们关注稀疏学习里不同正则化器的性能，比较它们在实验中的表现，并探讨其不同场景下的适用性。通过编程实现和数值实验，我们将评估各算法的优劣，并通过可视化呈现其在特征选择任务中的实际效果。这项研究对于优化数据处理、提高决策效率具有重要意义，为实际问题的解决提供了有力支持。

关键词：特征选择；稀疏学习；机器学习；正则化器

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Sparse learning optimization feature selection: algorithm comparison and performance evaluation

DEPARTMENT: Department of Mathematics

SPECIALIZATION: Statistics

UNDERGRADUATE: Jiapeng Wang

MENTOR: Zhengyu Wang, Associate Professor

ABSTRACT:

This paper aims to investigate the effectiveness and applications of sparse learning algorithms in feature selection. With the explosive growth of data, sparse learning has become an important method for addressing high-dimensional data issues in feature selection. From simple Lasso algorithms to complex $l_{2,1-2}$ algorithms, we focus on the performance of different regularization techniques in sparse learning, compare their performance in experiments, and explore their applicability in various scenarios. Through programming implementation and numerical experiments, we will evaluate the strengths and weaknesses of each algorithm and present their practical effects in feature selection tasks through visualization. This research is of significant importance for optimizing data processing and enhancing decision efficiency, providing strong support for addressing practical problems.

KEYWORDS: Feature Selection; Sparse Learning; Machine Learning; Regularizer

目 录

第一章 引言	1
1.1 研究背景及意义	1
1.2 研究方法和框架概述	2
第二章 文献综述	3
2.1 特征选择与稀疏学习概述	3
2.2 国内外研究现状	3
2.3 算法概述	4
第三章 稀疏学习中的特征选择算法	6
3.1 基于向量的特征选择算法	6
3.1.1 Lasso 算法	6
3.1.2 弹性网络算法	8
3.2 基于矩阵的特征选择算法	8
3.2.1 $l_{2,1}$ 正则化	8
3.2.2 $l_{2,1-2}$ 正则化	10
第四章 数值实验	13
4.1 数据集介绍	13
4.2 基于 26 字母数据集的实验	14
4.2.1 数据探索和可视化	14
4.2.2 特征选择	15
4.2.3 模型评分	16
4.3 基于手写中文零到亿的数据集的实验	19
4.3.1 数据探索和可视化	19

4.3.2	特征选择	19
4.3.3	模型评分	22
4.4	实验结果分析	22
4.4.1	算法准确度排名	22
4.4.2	特征选择的分类考虑	23
第五章 结论与展望		24
5.1	研究成果总结	24
5.2	存在的问题	25
5.3	后续研究的建议和展望	26
参考文献		28
致 谢		30

第一章 引言

1.1 研究背景及意义

在当今社会和科技的快速发展中，数学知识和统计方法被广泛应用于我们的日常生活。机器学习作为一个热门话题，以其在人工智能、数据分析等领域的卓越表现，吸引了众多人的关注。在大数据时代，人们每天都会产生、接收和处理海量的数据，这些数据本身就是宝贵的财富。特征选择方法使得人们能够从这庞大的数据海洋中提取出真正具有意义的部分。通过各种机器学习算法的模拟和预测，人们往往能够优化自己的决策，创造更大的社会价值。

特征选择是一种选择数据子集的过程，旨在从原始特征中识别出相关的特征子集，这可以方便后续的分析，例如聚类 and 分类。在许多情况下，机器学习面临的原始数据集存在一系列缺陷，如维度过高和噪声数据过多。面对这些数据，稀疏学习因其简约特性和计算优势而备受关注。在使用各种算法进行特征降维的过程中，常常会遇到复杂的计算问题。然而，在具有稀疏性的条件下，这些问题可以得到有效的处理和解决。为了获得更好的结果，人们提出了许多稀疏正则方法以提高稀疏算法的性能。正则化器（regularizer）作为其中的一种重要手段，通过向模型的损失函数中添加惩罚项来约束模型的复杂度，进而有效地应对过拟合问题。在许多情况下，凸正则化器被认为是有效且有帮助的。但在特定情况下，非凸正则化器展现出了优于凸正则化器的性能，这是一个值得研究的方向。

通过研究比较不同正则化器在稀疏学习中的表现，我们可以验证各种方法的实际有效性以及它们的不同适用场景。在研究特定算法时，我们回顾了先前的数学方法，并在新的数据集上进行了实验，以检验算法的收敛性，并比较了实验结果。无论是监督学习还是无监督学习，我们都可以找到切实有效的方法进行稀疏学习，从而提高特征选择的性能，最终帮助我们解决实际生活中的问题。

1.2 研究方法和框架概述

本研究结合国内外的研究成果^[1]，系统地综合和分析了稀疏学习算法的特点^[2]，并进行了全面的评估。研究的核心集中在四种代表性的稀疏学习算法上，分别是 Lasso 算法、弹性网络、 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化。

首先，本文将详细介绍这四种算法的数学模型、优化目标和相关的数学定理。对于 Lasso 算法和弹性网络，我们会介绍它们的基本形式、目标函数以及求解方法。对于基于矩阵的正则化方法，包括 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化，我们将深入探讨它们如何通过结构化正则化来考虑特征间的关联性。

为了直观地评估这些算法在特征选择任务中的性能，本研究计划对上述算法进行编程实现，并在两个公开数据集上进行数值实验。这些数据集将涵盖不同的特征维度、样本量和特征与响应变量的关系，以确保实验结果的鲁棒性和可靠性。

在实验部分，我们将通过可视化，分数指标等方式比较各算法在不同数据集上的运行时间、模型准确度以及特征选择的效果。通过这些比较，我们旨在揭示各算法在不同数据条件下的优势和局限性，以及它们如何影响特征选择的最终效果。

通过上述研究方法和框架，本文希望在为稀疏学习算法的选择和应用提供全面而深入的理解，为特征选择任务提供有效的指导和参考。

第二章 文献综述

2.1 特征选择与稀疏学习概述

作为一种数据降维技术，特征选择旨在通过去除不相关、冗余或嘈杂的特征，从原始特征中选择一小部分相关特征。特征选择通常可以加快学习过程，带来更好的学习性能、更高的学习精度、更低的计算成本和更好的模型可解释性。就标签信息的可用性而言，特征选择技术可大致分为三类：有监督方法、半监督方法和无监督方法，本文我们将围绕有监督方法展开。标签信息的可用性允许有监督的特征选择算法有效地选择有区别的和相关的特征来区分来自不同类别的样本。有监督特征选择算法可以识别相关特征以最好地实现监督模型的目标（例如分类或回归问题），并且它们依赖于标记数据的可用性。

根据与学习模型的交互，监督特征选择方法大致分为四种类型，即过滤式、包装式、混合式和嵌入式方法。嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，即在学习器训练过程中自动地进行了特征选择。本文涉及的稀疏学习既是嵌入式特征选择的一种。基于稀疏学习的特征选择方法旨在最小化拟合误差以及一些稀疏正则化项。稀疏正则化器迫使某些特征系数变小或恰好为零，然后可以简单地消除相应的特征。

稀疏学习通过利用数据的稀疏性，在降低模型复杂度的同时保持良好的预测性能和可解释性，这使得它在处理高维数据、特征选择、信号处理等领域得到了广泛的应用。因此，近些年来，稀疏学习一直是机器学习等领域的一个研究热点。

2.2 国内外研究现状

在基于结构化稀疏的特征选择中，人们使用的算法分为两类^[3]，分别是基于向量的特征选择和基于矩阵的特征选择。这两类算法，以及其包含的各种稀疏学习正则化器，受到了国内外广泛的研究，具有不同的性质和适用范围。

基于向量的特征选择是处理稀疏数据中比较直观和容易理解的方法。我们所熟知的经典的 l_1 正则化方法 (Lasso) 便属于该类。该算法通过添加 l_1 范数惩罚项来产生大量的零系数, 促使模型产生稀疏性, 进而实现特征选择。类似的算法如弹性网络等也属于基于向量的特征选择。在特定情形下, 它们作为正则化器在稀疏学习中也有优异的表现。然而, 该类算法的一大缺陷便是其对特征之间相关性或其他结构信息的忽略, 这会导致整体而言其特征选择的效果可能会逊色于基于矩阵的正则化器。而且当我们面对更为复杂的高维数据集时, 由于算法的简单, 类似 l_1 正则化的算法可能会面临计算上的困难, 最终也会影响到输出的结果。

基于矩阵的特征选择是更为复杂的方法, 但它的优势在于能够考虑到特征之间的结构信息, 比如特征的相关性或者组关系。依靠这类正则化器实现的全局优化, 我们往往能够在模型稀疏化的同时得到更好的特征子集。一个该类正则化器的例子是 $l_{2,1}$ 正则化, 该算法中 $l_{2,1}$ 范数是矩阵中列范数和行范数的组合, 用于促使模型产生稀疏性并同时考虑特征之间的关系, 帮助挖掘数据的结构信息。通过基于矩阵的特征选择, 人们可以更有效地选择有意义的特征, 以提高模型的效果。尽管如此, 考虑到这一类算法的复杂性和计算开销, 我们并不能在任何情况下都推崇它们, 而是应该结合具体问题的特点和需求选择最合适的方法。

除了提及的算法外, 在近年来的研究中, 我们可以看到越来越多的新型正则化器在稀疏学习和特征选择领域受到广泛关注, 例如 $l_{2,1-2}$ 正则化器等。这些新方法的出现极大丰富了特征选择的方法学, 为我们提供了更多选择和可能性。总体而言, 在国内外的持续研究中, 我们不断提高特征选择的效果, 并将新的算法应用于更复杂和多样化的场景, 以满足社会对更精准可靠数据处理的需求。

2.3 算法概述

有监督的特征选择是一种在给定目标标签的情况下, 通过优化目标函数来选择最优特征子集的方法。在给定的 d 维特征和 n 个样本的训练数据集 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ 以及目标标签 y 的情况下, 特征选择的目标是找到一个最优的特征子集 \mathbf{W} , 使得损失函数和正则化器的加权和最小。

具体地，我们可以用以下数学形式描述这一目标：

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}}(L_{\mathbf{X},\mathbf{y}}(\mathbf{W}) + \alpha R(\mathbf{W})) \quad (2-1)$$

其中， $L_{\mathbf{X},\mathbf{y}}(\mathbf{W})$ 表示算法的损失函数，用于衡量模型预测与真实标签之间的差异。 $R(\mathbf{W})$ 是正则化器，它对特征子集 \mathbf{W} 的稀疏性进行惩罚或奖励，从而实现特征选择的目的。参数 α 是一个超参数，用于平衡损失函数项和正则化项之间的重要性。

这个目标函数的优化过程旨在找到一个最佳的特征子集 \mathbf{W} ，这个子集不仅能够很好地解释目标标签 y ，还具有较高的稀疏性。通过这种方式，我们能够更有效地利用特征与标签之间的相关性，从而提高模型的泛化能力和预测准确性。

接下来我们将介绍几种常见的稀疏学习算法，尽管这些算法的具体形式可能有所不同，但它们的优化目标基本上都符合上述目标函数（2-1）的形式。通过对这些算法的深入分析和比较，我们将能够更好地理解它们的工作原理、优势和局限性，为特征选择提供有力的理论和实践支持。

第三章 稀疏学习中的特征选择算法

3.1 基于向量的特征选择算法

3.1.1 Lasso 算法

Lasso 算法是非常经典的算法^[4]，所选用的是 l_1 正则化器。 l_1 范数得到的是向量中各元素的绝对值之和，即：

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (3-1)$$

进而可以得到该算法的目标函数：

$$\min_w \|y - X^T w\|_2^2 + \alpha \|w\|_1 \quad (3-2)$$

其中 $w = (w_1, w_2, \dots, w_d)$ 是要估计的未知权重的向量。注意到 w 取得稀疏解意味着初始的 d 个特征中仅有对应着 w 的非零分量的特征才会出现在最终模型中，于是，求解 l_1 范数正则化的结果是得到了仅采用一部分初始特征的模型；换言之，基于 l_1 正则化的学习方法就是一种嵌入式特征选择方法，其特征选择过程与学习器训练过程融为一体，同时完成。

依照周志华《机器学习》一书^[5]的第十一章： l_1 正则化问题的求解可以采用近端梯度下降法，近端梯度下降法是众多梯度下降 (gradient descent) 方法中的一种。与经典的梯度下降法相比，近端梯度下降法主要是想解决目标函数中存在不可微或不方便微分的部分。数学上来说，其主要解决的问题可表示为 $\min_w f(w) = \min_w \text{ming}(w) + h(w)$ 这其中， $f(w)$ 是可微的凸函数，而 $h(w)$ 则是某些地方不可微的凸函数。它们在 Lasso 算法中分别对应损失函数和 $\|w\|_1$ 项。

近端梯度下降法的大致思想是首先在一定条件下，通过泰勒展开式将目标函数可微的部分化简，然后以分段函数的形式，对 $\|w\|_1$ 求导，令导数为 0，进而

得到分段函数形式的迭代公式。现若 $f(w)$ 可导，且 $\nabla f(w)$ 满足 L-Lipschitz (利普希茨连续条件)，即存在常数 $L > 0$ 使得，

$$\frac{|\nabla f(w') - \nabla f(w)|}{|w' - w|} \leq L \quad (3-3)$$

那么在 w_k 附近可将 $f(w)$ 二阶泰勒展开并近似为：

$$\begin{aligned} f(w) &\simeq f(w_k) + \nabla f(w_k) \|w - w_k\| + \frac{L}{2} \|w - w_k\|_2^2 \\ &= \frac{L}{2} \left[\|w - w_k\|_2^2 + 2 \frac{1}{L} \nabla f(w_k) \|w - w_k\| + \left(\frac{1}{L} \nabla f(w_k) \right)^2 \right] \\ &\quad - \frac{L}{2} \left(\frac{1}{L} \nabla f(w_k) \right)^2 + f(w_k) \\ &= \frac{L}{2} \left[\left\| w - \left(w_k - \frac{1}{L} \nabla f(w_k) \right) \right\|_2^2 \right] + const \end{aligned} \quad (3-4)$$

这里若通过梯度下降法对 $f(w)$ 进行最小化，则每一步下降迭代实际上等价于最小化二次函数 $f(w)$ ，从而推广到我们最上面的优化目标，类似的可以得到每一步的迭代公式：

$$w_{k+1} = \underset{w}{\operatorname{argmin}} \frac{L}{2} \left\| w - \left(w_k - \frac{1}{L} \nabla f(w_k) \right) \right\|_2^2 + \alpha \|w\|_1 \quad (3-5)$$

令 w^i 表示 w 的第 i 个分量，将上式按分量展开可看出，其中不存在 $w^i w^j (i \neq j)$ 这样的项，也就是说各分量直间互不影响，假设只有一个分量 w^1 带入展开，求导，令导得零，便可以求得这个小范围内的最小值点，推广可得上式解为：

$$w_{k+1}^i = \begin{cases} z^i - \alpha/L, & z^i > \alpha/L \\ 0, & |z^i| \leq \alpha/L \\ z^i + \alpha/L, & z^i < -\alpha/L \end{cases} \quad (3-6)$$

其中， $z = w_k - \frac{1}{L} \nabla f(w_k)$ 。由此我们得到了针对 Lasso 算法的快速求解。

3.1.2 弹性网络算法

下面介绍的是弹性网络算法^[6]，它是一种同时使用 l_1 与 l_2 范数作为先验正则项训练的线性回归模型。这种组合允许学习到一个只有少量参数是非零稀疏的模型，就像 Lasso 一样，但是它仍然保持一些像 Ridge(l_2 范数)的正则性质^[7]。 l_2 范数是我们最常用的范数，其定义如下：

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \quad (3-7)$$

将 l_1 , l_2 范数组合，我们可以得到弹性网络的目标函数为：

$$\min_w \|y - X^T w\|_2^2 + \alpha \rho \|w\|_1 + \frac{\alpha(1-\rho)}{2} \|w\|_2^2 \quad (3-8)$$

我们这里可以看出，如果 $\rho = 1$ ，则该目标函数就为 Lasso 算法的目标函数。而该算法的求解也可以仿照 Lasso 算法的近端梯度下降法，仅需更改一下目标函数即可。

3.2 基于矩阵的特征选择算法

3.2.1 $l_{2,1}$ 正则化

与基于向量的特征选择算法相比， $l_{2,1}$ 正则化是高效且鲁棒的一个新颖的算法。其优点在于能在所有数据点中选择具有联合稀疏性的特征。接下来我们将依照^[8]，介绍该算法的具体内容。对于矩阵 $\mathbf{M} = (m_{ij})$ ，其第 i 行、 j 列分别记为 m^i 、 m_j ，我们定义其 $l_{2,1}$ 范数为：

$$\|\mathbf{M}\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^m m_{ij}^2} = \sum_{i=1}^n \|m^i\|_2 \quad (3-9)$$

然后在此也可以推广出 $l_{r,p}$ 范数的定义：

$$\|\mathbf{M}\|_{r,p} = \left(\sum_{i=1}^n \left(\sum_{j=1}^m |m_{ij}|^r \right)^{\frac{p}{r}} \right)^{\frac{1}{p}} = \left(\sum_{i=1}^n \|m^i\|_r^p \right)^{\frac{1}{p}} \quad (3-10)$$

依次得到该算法的目标函数为：

$$\min_{\mathbf{W}} \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_{2,1} + \alpha \|\mathbf{W}\|_{2,1} \quad (3-11)$$

也可以写为如下形式：

$$\min_{\mathbf{W}} \frac{1}{\gamma} \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_{2,1} + \|\mathbf{W}\|_{2,1} \quad (3-12)$$

为求解该算法，首先将上述目标函数改写为：

$$\min_{\mathbf{W}, \mathbf{E}} \|\mathbf{E}\|_{2,1} + \|\mathbf{W}\|_{2,1} \quad \text{s.t.} \quad \mathbf{X}^T \mathbf{W} + \gamma \mathbf{E} = \mathbf{Y}. \quad (3-13)$$

也即：

$$\min_{\mathbf{W}, \mathbf{E}} \left\| \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} \right\|_{2,1} \quad \text{s.t.} \quad \begin{bmatrix} \mathbf{X}^T & \gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} = \mathbf{Y} \quad (3-14)$$

进而令 \mathbf{A} 和 \mathbf{U} 如下：

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{X}^T & \gamma \mathbf{I} \end{bmatrix} \in \mathbb{R}^{n \times m} \\ \mathbf{U} &= \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} \in \mathbb{R}^{m \times c} \end{aligned} \quad (3-15)$$

则我们可以得到目标函数最终改写的式子：

$$\min_{\mathbf{U}} \|\mathbf{U}\|_{2,1} \quad \text{s.t.} \quad \mathbf{A}\mathbf{U} = \mathbf{Y} \quad (3-16)$$

人们为了解决式 (3-16) 曾尝试过许多方法，现介绍其中较为简单的一种。

首先式 (3-16) 的拉格朗日函数可以写为:

$$\mathcal{L}(\mathbf{U}) = \|\mathbf{U}\|_{2,1} - \text{Tr}(\mathbf{A}^T(\mathbf{AU} - \mathbf{Y})) \quad (3-17)$$

求上式对 \mathbf{U} 的偏导并令其为 0，得到式 (3-18):

$$\frac{\partial \mathcal{L}(\mathbf{U})}{\partial \mathbf{U}} = 2\mathbf{DU} - \mathbf{A}^T \mathbf{\Lambda} = \mathbf{0} \quad (3-18)$$

其中 \mathbf{D} 是对角矩阵，第 i 个元素为:

$$d_{ii} = \frac{1}{2 \|\mathbf{u}^i\|_2} \quad (3-19)$$

左乘 \mathbf{AD}^{-1} ，并有 $\mathbf{AU} = \mathbf{Y}$ ，可以得到如下式子:

$$\begin{aligned} 2\mathbf{AU} - \mathbf{AD}^{-1}\mathbf{A}^T\mathbf{\Lambda} &= \mathbf{0} \\ \Rightarrow 2\mathbf{Y} - \mathbf{AD}^{-1}\mathbf{A}^T\mathbf{\Lambda} &= \mathbf{0} \\ \Rightarrow \mathbf{\Lambda} &= 2(\mathbf{AD}^{-1}\mathbf{A}^T)^{-1}\mathbf{Y} \end{aligned} \quad (3-20)$$

将上式代回求式 (3-18)，得到式 (3-21):

$$\mathbf{U} = \mathbf{D}^{-1}\mathbf{A}^T(\mathbf{AD}^{-1}\mathbf{A}^T)^{-1}\mathbf{Y} \quad (3-21)$$

由于式 (3-16) 中的问题是一个凸问题，当且仅当式 (3-21) 满足时， \mathbf{U} 是该问题的全局最优解。接下来仅需使用迭代算法便可解得式 (3-21) 的 \mathbf{U} 。

3.2.2 $l_{2,1-2}$ 正则化

$l_{2,1-2}$ 正则化由^[9]提出，可以被视为对 $l_{2,1}$ 正则化的自然改进，它是矩阵的一个非凸但 Lipschitz 连续的稀疏度量。以上我们有对不同范数的定义，如对于 $\mathbf{W} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_m^T]^T$ ，我们有:

$$\|\mathbf{W}\|_{2,p} = \left(\sum_{i=1}^m \|\mathbf{w}_i\|^p \right)^{1/p} = \left(\sum_{i=1}^m \left(\sum_{j=1}^n \mathbf{w}_{ij}^2 \right)^{p/2} \right)^{1/p} \quad (3-22)$$

以及：

$$\|\mathbf{W}\|_{2,0} = \sum_{\mathbf{w}_i \neq \mathbf{0}} \|\mathbf{w}_i\|_2^0 \quad (3-23)$$

其中 \mathbf{W} 和 \mathbf{w} 分别代表矩阵和向量。

特别地，当 $p = 2$ 时， $l_{2,p}$ 为 Frobenius 范数，简称 F-范数，记为 $\|\cdot\|_F$ 。

具有相同尺度的两个矩阵之间的欧几里得内积定义为： $\langle \mathbf{W}, \mathbf{V} \rangle = \sum_{i,j} \mathbf{W}_{ij} \mathbf{V}_{ij} = \text{Tr}(\mathbf{W}^T \mathbf{V})$ ，其中 $\text{Tr}(\cdot)$ 是矩阵的迹。那么特别地， $\|\mathbf{W}\|_F^2 = \langle \mathbf{W}, \mathbf{W} \rangle = \text{Tr}(\mathbf{W}^T \mathbf{W})$ 。

接下来我们考虑矩阵的次梯度，假设 $\mathbf{W} \in \mathbb{R}^{m \times n}$ ， $\|\mathbf{W}\|_F$ 的次梯度为：

$$\partial \|\mathbf{W}\|_F = \begin{cases} \left\{ \frac{\mathbf{W}}{\|\mathbf{W}\|_F} \right\}, & \text{if } \mathbf{W} \neq \mathbf{0} \\ \left\{ \mathbf{M} \in \mathbb{R}^{m \times n} : \|\mathbf{M}\|_F \leq 1 \right\}, & \text{otherwise} \end{cases} \quad (3-24)$$

而 $\|\mathbf{W}\|_{2,1}$ 的次梯度为：

$$\partial \|\mathbf{W}\|_{2,1} = \left\{ \left[\psi(\mathbf{w}_1)^T, \psi(\mathbf{w}_2)^T, \dots, \psi(\mathbf{w}_m)^T \right]^T \right\} \quad (3-25)$$

此处定义 $\psi(\mathbf{w}_i)$ 为：

$$\psi(\mathbf{w}_i) = \begin{cases} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2}, & \text{if } \mathbf{w}_i \neq \mathbf{0} \\ \hat{\mathbf{w}}_i \in \{ \mathbf{w} \in \mathbb{R}^n : \|\mathbf{w}\|_2 \leq 1 \}, & \text{otherwise.} \end{cases} \quad (3-26)$$

最终可以在上面的基础上定义 $\|\mathbf{W}\|_{2,1-2}$ 如下：

$$\|\mathbf{W}\|_{2,1-2} = \|\mathbf{W}\|_{2,1} - \|\mathbf{W}\|_{2,2} = \|\mathbf{W}\|_{2,1} - \|\mathbf{W}\|_F. \quad (3-27)$$

以此定义得到的 $l_{2,1-2}$ 范数与上文提到的其他范数相比可以获得更稀疏的解，且其 Lipschitz 连续性质使其更容易优化。

有了范数，我们可以写出该算法的目标函数：

$$\min_{\mathbf{W}} L_{\mathbf{X},\mathbf{Y}}(\mathbf{W}) + \alpha \|\mathbf{W}\|_{2,1-2} \quad (3-28)$$

其中几个损失函数均可被用于该目标函数中，如 Frobenius 范数损失：

$$L_{\mathbf{X},\mathbf{Y}}(\mathbf{W}) = \|\mathbf{Y} - \mathbf{X}^T \mathbf{W}\|_F^2 \quad (3-29)$$

以及 $l_{2,1}$ 范数损失：

$$L_{\mathbf{X},\mathbf{Y}}(\mathbf{W}) = \|\mathbf{Y} - \mathbf{X}^T \mathbf{W}\|_{2,1} \quad (3-30)$$

那么为求解该目标函数，我们引入以下算法，该算法可以视为对 $l_{2,1}$ 正则化算法求解的一个改进。首先假设：

$$\mathbf{A}^k = \begin{cases} \|\mathbf{W}^k\|_F^{-1} \mathbf{W}^k, & \mathbf{W}^k \neq \mathbf{0} \\ \mathbf{0}, & \mathbf{W}^k = \mathbf{0} \end{cases} \quad (3-31)$$

对于有监督学习的特征选择，我们考虑式 (3-28) 的线性凸子问题，即：

$$\min_{\mathbf{W}} (L_{\mathbf{X},\mathbf{Y}}(\mathbf{W}) + \alpha \|\mathbf{W}\|_{2,1}) - \alpha \langle \mathbf{W}, \mathbf{A}^k \rangle \quad (3-32)$$

此处注意到第一次迭代时由于 $\mathbf{W}^0 = \mathbf{0}$ ，可以得到 $\mathbf{A}^0 = \mathbf{0}$ 。此时式 (3-32) 便可视为 $l_{2,1}$ 正则化问题。类似地，进行迭代我们最终可以得到 $l_{2,1-2}$ 正则化算法的解。

第四章 数值实验

4.1 数据集介绍

在本次数值实验中，主要将使用的是两个数据集，分别是一个手写 26 个英文字母的数据集和一个手写中文零到亿的数据集。这两个数据集都属于计算机视觉的研究范畴（Computer Vision），主要希望通过机器学习等方法实现对手写字母或文字的辨别。

计算机视觉的研究目标是模仿和复制人类视觉系统的功能和性能，通过计算机对图像和视频进行处理和解释。随着人工智能技术的迅速进步，计算机视觉已经在许多领域展现出了巨大的应用潜力和价值。例如，在面部识别、医疗图像分析、无人驾驶、安全监控等领域，计算机视觉技术都已经取得了令人瞩目的成果。

特别是在文字识别这一领域，计算机视觉技术在自动化、高效性和准确性方面的要求都非常高。手写文字的识别涉及到复杂的形状和结构分析，而且还需要处理不同人的书写风格和书写质量不一的问题。因此，有效的特征选择和稀疏学习算法在手写文字识别任务中尤为关键。

通过本次数值实验，我们希望能够验证和比较不同的稀疏学习算法在手写字符数据集上的性能，进一步探索如何通过特征选择来优化手写文字识别的效果。在本次实验中，将使用准确率作为评估指标，对最终的结果进行打分。准确率（Accuracy）是我们在机器学习中极为常用的指标，它指的是所有的预测正确样本（正类负类）的占总的比重。具体数学上来看，在一次预测中，我们给出以下定义：

N: 样本总数;

TP (True Positive): 正确的正例，一个实例是正类并且也被判定成正类;

TN (True Negative): 正确的反例，一个实例是假类并且也被判定成假类。

则随之即可得到具体准确率的数学表达:

$$Accuracy = \frac{TP + TN}{N} \tag{4-1}$$

接下来的数据实验中,我们将根据准确率的高低来判断预测模型的好坏,进而判断特征选择算法的优劣。

4.2 基于 26 字母数据集的实验

4.2.1 数据探索和可视化

我们第一个面对的数据集为手写 26 个英文字母的数据集^[10],该数据集中包含大概 370000 张像素为 28×28 的手写字母图像。在接下来的数据处理中,出于对电脑性能和运算时间的考虑,仅选择从中随机抽取的 80000 张图进行实验。

以下为摘取其中四张图片的展示:

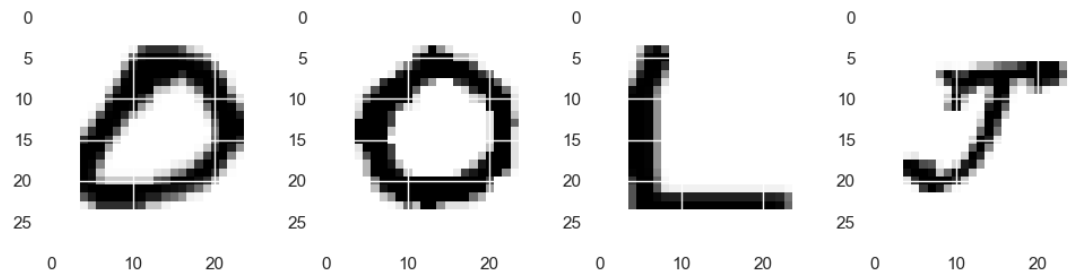


图 4-1 26 英文字母数据集样本图片

在接下来的具体实验中,我们的方法均是基于数据,所以我们进而要将 28×28 的图像转化为 784 列的表格(在 Python 中为其特有的数据格式:数据帧),转化后表格的前五行如下:

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.10	...	0.639	0.640	0.641	0.642	0.643	0.644	0.645	0.646	0.647	0.648
305828	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
185295	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
293268	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
163128	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
157156	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 784 columns

图 4-2 原始数据帧前五

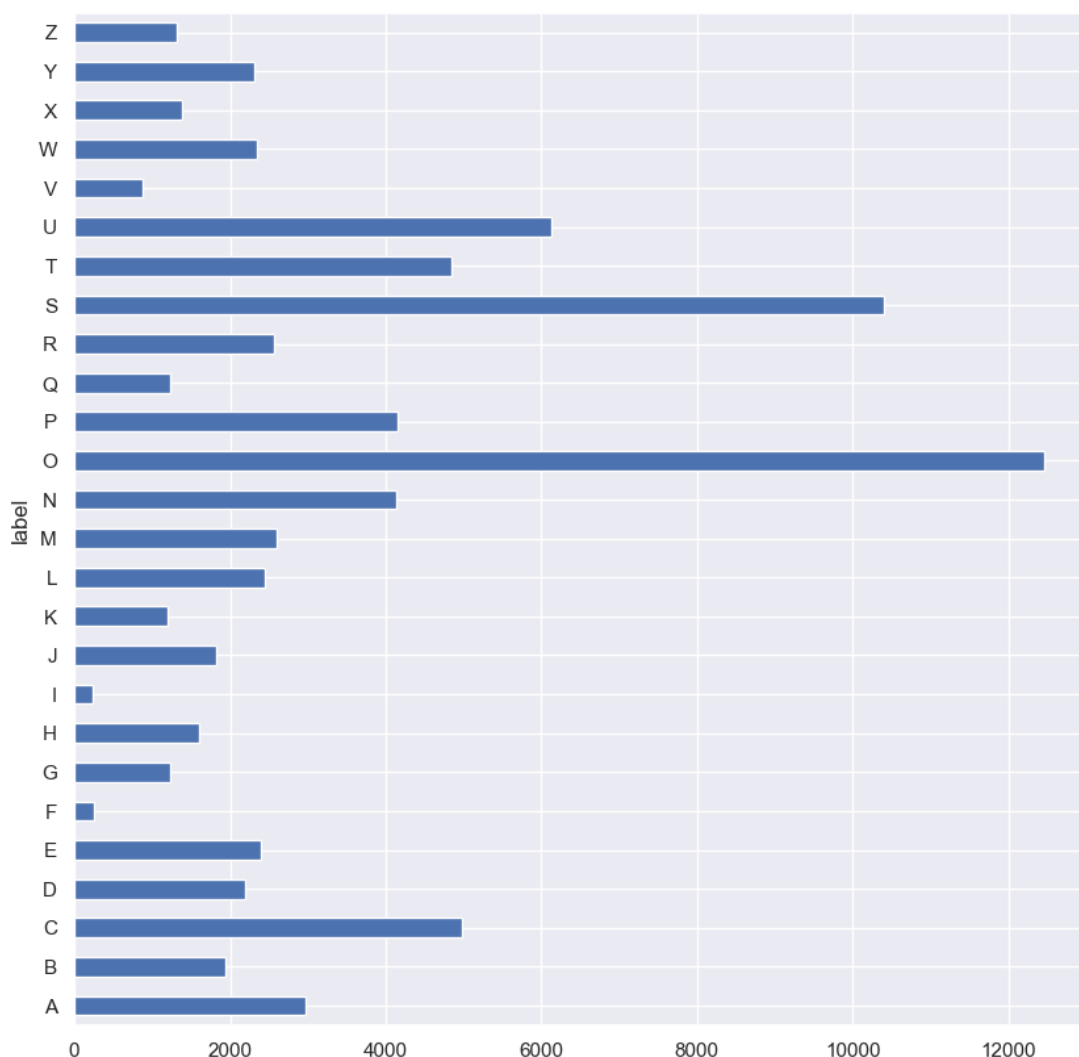


图 4-3 标签列分布状况

同时，作为有监督学习，此处也要考虑标签列的情况，图 4-3 是柱状图所体现的标签列的分布情况。从分布图中可以看出，在这次实验中我们面对的是一个不平衡的数据集，样本数据中 O 和 S 较多，而 F 和 I 较少，对于我们预测得到的数据，也应该有大致分布情况。

4.2.2 特征选择

接下来进入到数值实验的关键部分，即四种正则化方法分别进行的特征选择，首先在下面给出四种特征选择后得到的矩阵列数。（矩阵列数在数据分析中又被称为特征数或维度，以下图 4-4 所展示的列数也可以理解为四种算法筛选后得到的最优特征数）由此看出四种方法均实现了特征选择的效果，而所得的特征数也相差一百以内。这说明数据整体的复杂度不高，维度也较少，在下面的

	无特征筛选	Lasso	弹性网络	$l_{2,1}$	$l_{2,1-2}$
矩阵列数	784	352	384	412	380

图 4-4 算法得到的最优特征数

实验中也可以佐证这一点。

下面的实验将给出随着特征增加，每个算法得到的准确度的提升情况。由于我们所选用的算法都能够对特征的重要性进行排序，因此我们可以根据这些排序结果逐步向模型中添加特征，从而研究模型的特征选择效果以及对特征重要性排序的准确性。

我们将通过线型图展示这些算法在不同特征数量下的准确度表现。具体来说，图 4-5 将展示 Lasso 算法、弹性网络算法、 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化在所选数据集上的准确度随特征数量变化的情况。在图中， X_1, X_2, X_3, X_4 分别代表这四种算法所对应的准确度数据。

从图中我们可以观察到，这四种算法在该数据集上的准确度整体上相当，存在一定的波动。特别值得注意的是， $l_{2,1-2}$ 正则化算法随着特征数量的增加，准确度的提升更为平稳，这可能表明该算法在处理高维数据时具有更好的鲁棒性和稳定性。

总体而言，尽管这四种算法在特定的数据集上可能存在细微的差异，但都表现出了良好的特征选择和模型训练能力。仅就该数据集而言，这四种算法均可以被应用并取得很好的效果。

4.2.3 模型评分

当今时代面对计算机视觉数据集，人们往往会选择使用深度学习模型。深度学习过程省略了传统机器学习的特征工程，尝试通过算法直接从数据中学习高级特征，通常会得到更高的分数。但深度学习的缺点为其可解释性较差，很多时候预测出的结果无法被解释，也就很难被信任。尤其在金融，安全等领域，人们会更偏向于原理较为简单的传统机器学习模型。

在对该数据集的模型评分中，为统一标准，将对四种算法分别筛选得到的四个最优特征集运用支持向量机方法建模（SVM）。具体方法为，首先将数据集整

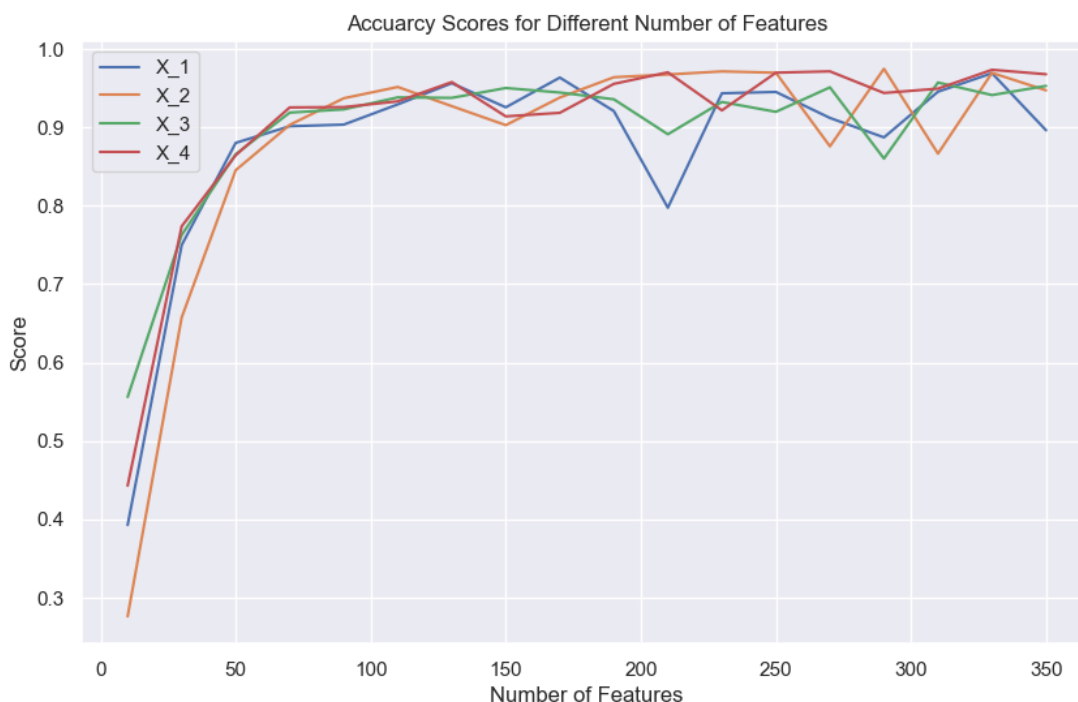


图 4-5 随特征增加的准确度变化图

体的 75% 记为训练集，25% 记为测试集。然后对四个最优特征集分别使用 SVM 算法拟合，用于预测测试集的算法，此处我们选择准确率（Accuracy）作为评估指标，最终得到每个算法的分数。

针对每个算法预测的标签值，同样可以在此观察其每个字母的分布，如图 4-6。图中得以看出四种算法预测的标签分布与实际标签分布并无大差异，粗略的展现出模型预测的大体成功性。

接下来将比较具体算法的分数结果。除四个算法特征选择得到的分数以及无特征筛选直接机器学习得到的分数外，为了展示传统机器学习算法与深度学习算法的比较，此处也新增了一个基于 CNN 深度学习算法得到的准确度。具体得到的分数如图 4-7 所示：

由准确度可以清晰的看出，相比于传统特征选择与机器学习方法，深度学习算法得到的准确度是更高的。但若是专注于该数据集，由于本身数据集的行数与列数并不大（尤其是列数），另外四种算法之间差异不大，得到的准确度也属于可以接受的范畴。这种现象的产生可能与图片的像素仅为 28×28 有关。针对这种类型的数据，也许传统的那四种正则化实现特征选择的算法也可以有用武之地，这主要取决于在业务角度决策者是否愿意牺牲部分的准确度以换取更高的可解释性。而关于这四种算法内部的比较，由之前的实验结果，可以得出这样

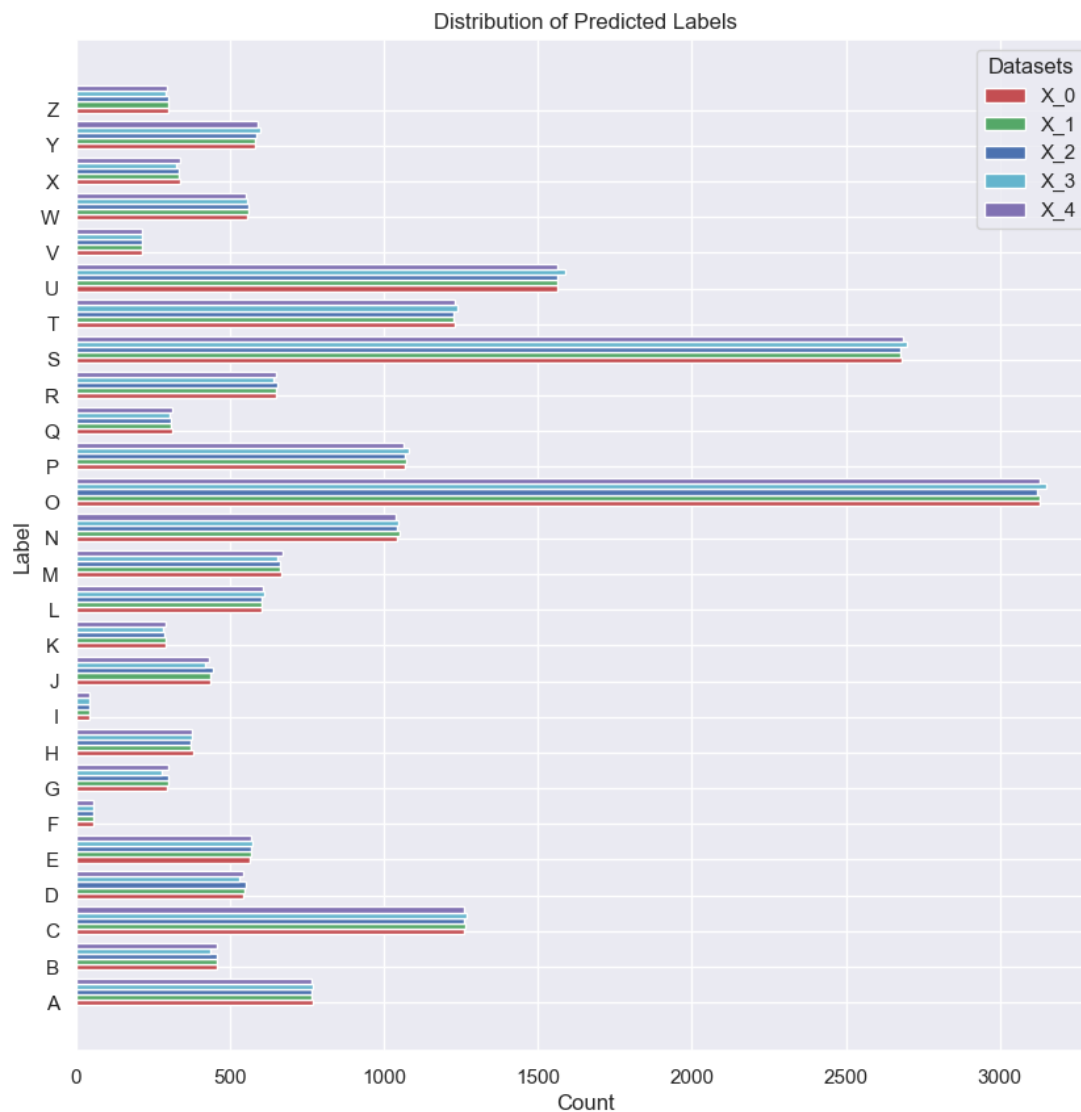


图 4-6 预测标签列分布情况

	无特征筛选	Lasso	弹性网络	l2, 1	l2, 1-2	CNN
准确率	0.95855	0.9628	0.96325	0.96135	0.9593	0.97755

图 4-7 算法准确率

的结论：除 $l_{2,1-2}$ 正则化算法外，其余三种算法的特征选择效果大致相同。而对于 $l_{2,1-2}$ 正则化而言，尽管其在准确度方面并未脱颖而出，但由图 4-5，它对特征重要性的排序更加准确，这使得该算法相比较起来更值得被选取。

在下面的数据集中，由于维度的增加，我们可以更好的比较四种正则化算法，辨别它们各自的优劣势。

4.3 基于手写中文零到亿的数据集的实验

4.3.1 数据探索和可视化

对于手写中文数字的数据集^[11]，其大致样式与上面一个数据集区别不大。但此处要注意这个数据集的图片像素为 64×64 ，所以其特征维度有约 4000 维，这决定了这个数据集是更为复杂的数据集。这个数据集中共有 15000 张图片，图 4-8 将摘取每个数字一张图片样本作为演示。

此处也一样考虑标签列的分布，由图 4-9，发现在这个数据集中并没有数据分布不平衡的现象，所有数据都是均匀分布的，即 15 个标签，每个标签包含 1000 个样本。

由于已知我们将要面对的原始特征集为 15000×4096 ，表格规模过于庞大，这里就不具体展现了。接下来直接进入四种算法的特征选择阶段。

4.3.2 特征选择

首先图 4-10 给出在该数据集上四个算法得到的特征数：

由此可以看出在面对高维数据集时。基于向量的特征选择方法和基于矩阵的特征选择方法出现了显著的区别。基于矩阵的两个特征选择方法由于能够考虑到特征之间的相关性，所以会筛掉更多的无用特征，使得模型拟合速度更快而准确度更高。

在针对这个数据集的具体模型拟合过程中，由于数据维度过大，所以采用上一个数据集的支持向量机（SVM）方法所涉及的计算过于复杂，耗费的时间非常多。所以以下针对最优特征集的机器学习算法，将采用较为新颖且速度较快的 LightGBM 集成学习方法^[12]。接下来首先简单介绍一下该机器学习算法。

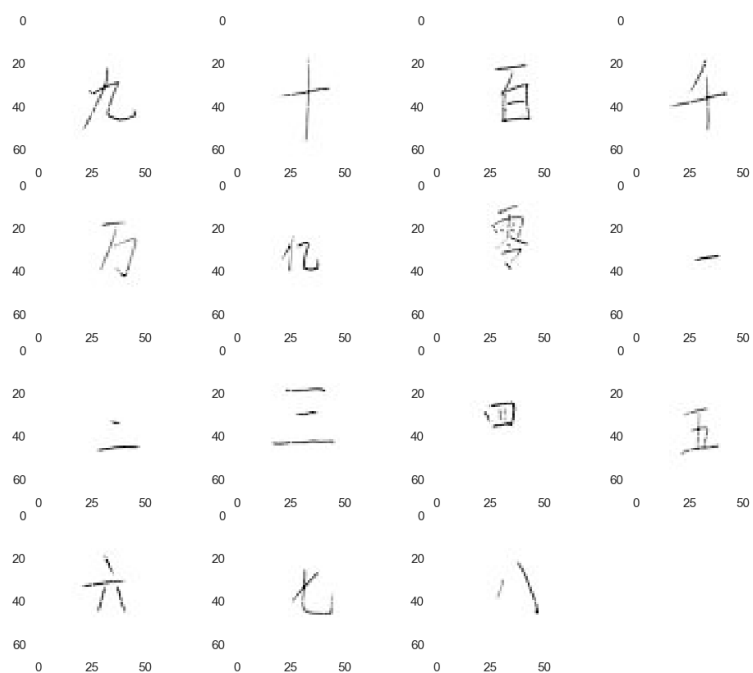


图 4-8 中文数字数据集样本图片

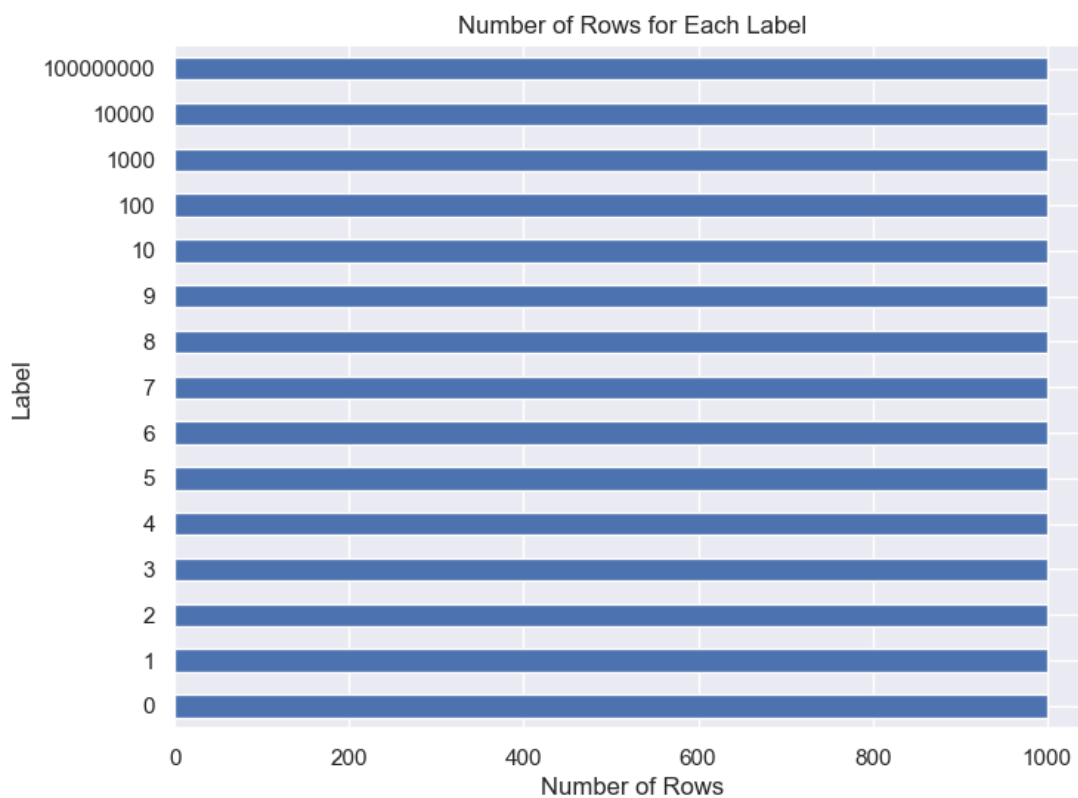


图 4-9 标签列分布状况

	无特征筛选	Lasso	弹性网络	$l_{2,1}$	$l_{2,1-2}$
矩阵列数	4096	3721	3766	2811	2937

图 4-10 算法得到的最优特征数

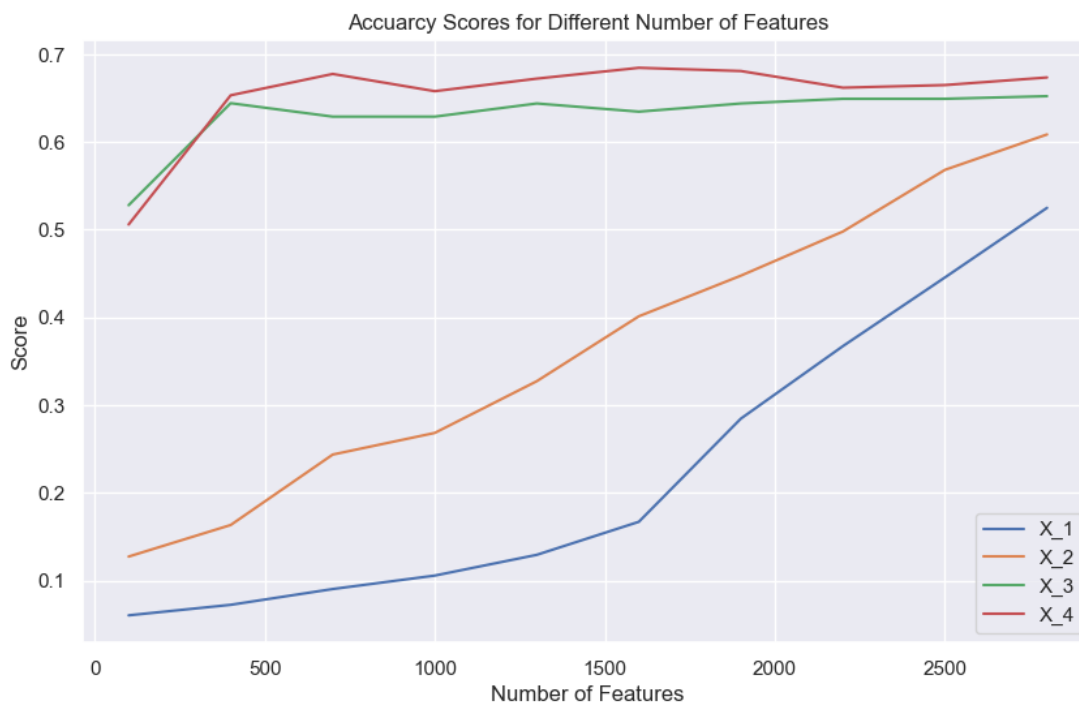


图 4-11 随特征增加的准确度变化图

GBDT（梯度提升决策树）是机器学习中一个十分常用的算法，它采用了 boosting 的集成方法，核心思想是通过迭代优化一个累加的预测函数，每一步都针对前一轮的残差（即真实值与预测值之差）构建一个新的弱学习器。通过这种方式，梯度提升树逐步减小残差，从而提升模型的整体性能。LightGBM（Light Gradient Boosting Machine）则是一个实现 GBDT 算法的框架，其优点在于支持高效率的并行训练，并且具有更快的训练速度、更低的内存消耗、更好的准确率。

接下来与上个数据集同样情况，给出图 4-11 展现随着特征增加，模型准确度的提升效果：

这里从图中也明显可以看出随着矩阵维度增加， $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化这两个基于矩阵的特征选择方法获得了更好的效果。这两个算法得到的最优特征集抓住了对准确度提升最大的特征，并且准确度明显的高于另外两种算法。而

	无特征筛选	Lasso	弹性网络	$l_{2,1}$	$l_{2,1-2}$
准确率	0.65147	0.65147	0.65147	0.6532	0.656

图 4-12 算法准确率

单比较 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化这两者，我们也可以得出， $l_{2,1-2}$ 正则化的准确度是略优于 $l_{2,1}$ 正则化的，这是其优势和提升所在。

4.3.3 模型评分

要注意到针对该数据集，由于本身特征数较为庞大，且此处选择的机器学习算法为了运算速度而牺牲了一部分的准确度，所以接下来的准确度并不高。图 4-12 给出各个算法的准确度用以比较：

据此，以及上面各图，可以清晰地看出四种算法得到准确度的显著差异。基于向量的特征选择方法，即 Lasso 和弹性网络，对分数的提升非常少 ($<10^{-5}$)。而 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化会对模型有一定的优化。同时，由于 $l_{2,1-2}$ 正则化可看作是 $l_{2,1}$ 正则化的优化，其准确度要更高一些。

4.4 实验结果分析

在本节中，我们对实验结果进行详细的分析，以深入理解各种稀疏学习方法在特征选择任务上的表现。我们首先基于准确度对所用算法进行了粗略的排名，然后从特征选择的角度进行了更深入的分析考虑。

4.4.1 算法准确度排名

根据我们的实验结果，Lasso 方法的表现最差，其次是弹性网络，然后是 $l_{2,1}$ 正则化，最后是 $l_{2,1-2}$ 正则化。这个排名与这些算法的复杂性和发展历程是一致的。Lasso 和弹性网络是最早被广泛研究和应用的稀疏学习方法，它们的数学模型相对简单，因此在某些复杂数据集上可能无法捕捉到特征间的复杂关系。而 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化是基于矩阵的特征选择方法，能够更好地考虑特征间的相互影响，因此在一些特定的数据集上表现更好。

4.4.2 特征选择的分类考虑

从特征选择的角度来看，基于矩阵的特征选择方法整体上优于基于向量的特征选择方法。这一点在特征维度较大的数据集上表现得尤为明显。基于向量的特征选择方法仅考虑单个特征与目标列的关系，而忽略了特征与特征之间可能存在的相互影响。相比之下，基于矩阵的方法能够全面考虑特征间的关系，从而更准确地选择有用的特征。

在考虑时间效率时，当数据集的特征较少时，Lasso 和弹性网络是合适的选择。这两种方法具有较快的速度和较强的可解释性，同时在准确度上与其他方法相差不大。但当处理特征较多的复杂数据集时，基于矩阵的正则化方法，特别是效果更好的 $l_{2,1-2}$ 正则化，更为合适。这是因为这些方法能够更全面地考虑特征间的相互影响，通过选择维数较少的数据子集来减少内存使用，同时提高模型预测的准确度。

综上所述，选择合适的稀疏学习方法和特征选择策略需要根据具体的数据集特性和需求来确定。在特征维度较大或特征间相互关联度高的情况下，基于矩阵的正则化方法可能是更优的选择。而在特征较少或对模型可解释性有要求的情况下，Lasso 和弹性网络则是更为合适的选项。

第五章 结论与展望

5.1 研究成果总结

在本文中，首先提出的是利用稀疏学习算法进行特征选择的研究背景，方法及意义。总体而言，特征选择是统计预测中至关重要的一环，而接下来介绍的稀疏学习算法可以优化该过程，得到最优特征集。在对稀疏学习的研究中，人们提出了许多算法（正则化器），而它们各有自己的优劣之处和适用场景。本文主要将重点放在由浅入深的四种正则化器上，比较它们的性能差别。本文希望通过比较研究得出在不同的场景下什么样的稀疏学习算法做特征选择得到的效果最好，这样就可以进而节省挑选算法的时间，提高特征选择的性能，优化模型的模拟和预测功效，对实际生活方方面面的策略选择提供帮助。无论是在医疗诊断、金融风控还是工业生产等领域，选择合适的特征和模型都将对策略制定和问题解决产生深远的影响。因此，本文的研究不仅有助于学术界对稀疏学习算法特征选择的理解，也具有广泛的实际应用价值。

在稀疏学习过程中，我们详细介绍了四种正则化器在特征选择中的原理和特点。首先是 Lasso 算法，这是最早被广泛研究和应用的正则化器之一。Lasso 通过对目标函数添加 L1 正则化项，实现了特征的稀疏性，从而使得一些不相关或冗余的特征的系数被压缩至零。Lasso 算法的简单性和可解释性使其成为特征选择领域的一个重要工具。

然后是弹性网络算法，这是对 Lasso 算法的改进。弹性网络结合了 L1 和 L2 正则化，综合了两者的优点，既能实现特征选择，又能处理特征之间的相关性，从而提高了模型的稳定性。然而，当特征之间存在非线性关系时，弹性网络算法可能会面临准确度下降的问题。

为了解决这一问题，我们引入了基于矩阵的特征选择方法，其中 $l_{2,1}$ 正则化是一个典型的代表。与基于向量的方法不同， $l_{2,1}$ 正则化能够考虑特征之间的结构关系，如特征间的相关性和组关系。这种方法虽然计算复杂度较高，但通常能

够提供更为精确的特征选择结果。

最后，我们介绍了 $l_{2,1-2}$ 正则化，这是对 $l_{2,1}$ 正则化的进一步优化和改进。通过改变正则化器的形式和参数设置， $l_{2,1-2}$ 正则化能够增强模型的准确度和鲁棒性，特别是在处理复杂数据和非线性关系时表现更为出色。

综上所述，这四种正则化器各有特点，适用于不同类型和复杂度的数据。选择合适的正则化器对于特征选择的准确性和模型性能至关重要。通过深入比较和分析这些正则化器，我们不仅可以更好地理解它们的工作原理，还能为实际应用提供有针对性的建议，从而优化模型的性能和效果。

在数值实验部分，我们选取了与日常生活紧密相关的数据集进行验证，以更好地评估所提稀疏学习算法在实际应用中的性能。具体而言，我们选择了手写 26 个英文字母的数据集和手写中文数字从零到亿的数据集，这两个数据集在文档转换、数字化和自动化数据录入等日常应用中具有广泛的实用价值。尽管计算机视觉听起来可能与我们的日常生活有些远离，但实际上，它在多个应用场景中都得到了广泛的应用。特别是在需要进行文字或字母识别的场景中，如自动文档处理、手写数字识别等，对特征选择的准确性要求极高。因此，对这些数据集进行特征选择以提高预测性能显得尤为重要。所以对这两个数据集作特征选择以提高预测性能就尤为重要。那么经过数值实验，文中也已给出得到的结论。总结来说，即在维度较少的数据集，四种稀疏学习算法得到的准确度差别不大，可以权衡准确度，速率和可解释性多方面决定；但对于维数较多的数据集，可能基于矩阵的特征选择算法得到的准确度有显著性的提高，所以 $l_{2,1}$ 正则化和 $l_{2,1-2}$ 正则化可能在这种场景中更适合被选择。

5.2 存在的问题

在对稀疏学习算法进行全面评估和比较后，我们可以明确地看到其在特征选择中具有高效的应用潜力。然而，这并不意味着这些算法没有局限性或可以完全替代其他方法。在四种正则化方法的对比中，我们发现很难在运算速率和准确度之间找到一个完美的平衡。这意味着在实际应用中，可能需要根据具体情况和需求来权衡这两个方面。

与其他方法，尤其是深度学习技术如 CNN（卷积神经网络）相比，稀疏学

习算法在特征选择中有其独特的优势和劣势。其中，最显著的优势是其强大的可解释性。稀疏学习算法能够直接输出特征的重要性，帮助我们理解哪些特征对预测结果有更大的影响。这种可解释性对于许多实际应用场景，如金融和医疗领域，是非常重要的。

然而，与深度学习等算法相比，稀疏学习在准确度上可能存在一定的差距。这一点在我们的数值实验中也得到了验证。因此，在选择特征选择方法时，业务需求和目标是非常关键的考虑因素。在某些场景下，例如金融^[13]和医疗领域^[14]，从业者可能更倾向于选择能够提供高可解释性的稀疏学习算法，即使这意味着需要牺牲一部分的准确度。

综上所述，稀疏学习算法在特征选择中确实有其独特的优势，尤其是在可解释性方面。然而，在实际应用中，选择最合适的算法需要综合考虑多种因素，包括但不限于准确度、速度、可解释性以及具体的业务需求。本文介绍的稀疏学习算法为特征选择提供了一个有力的工具，但其是否适合特定应用还需根据实际情况进行综合评估和选择。

5.3 后续研究的建议和展望

随着这几年深度学习的发展，很多场景下人们普遍更关注神经网络等算法的应用。神经网络的强大表示能力和自动特征学习能力使其在许多任务上表现出色。然而，这也使得特征选择这一传统机器学习中的关键步骤在神经网络中变得不那么显著，因为神经网络能够自动从数据中学习到有用的特征表示。

尽管如此，传统机器学习算法，特别是在特征选择方面表现出色的稀疏学习算法，仍然在许多实际应用场景中有其独特的价值。这些算法不仅能够提供高度可解释的特征选择结果，还能在模型训练过程中实现特征选择，从而降低模型复杂性，提高模型的泛化能力和计算效率。稀疏学习算法作为一种嵌入式的特征选择方法，其与深度学习不同的设计理念和应用场景为其赋予了独特的优势。在一些需要模型可解释性、计算效率或者特征选择能力的应用场景中，稀疏学习算法可能会更为适合。例如，在医疗领域，由于需要解释和理解模型的预测结果，因此可解释性和准确性同等重要。在这种情况下，稀疏学习算法的优势就会更加明显。

因此，本文介绍的稀疏学习算法不仅具有实际应用的价值，而且在当前机器学习领域的发展趋势下，其未来的发展空间和适用场景仍然广泛。特别是在需要特征选择的复杂应用中，稀疏学习算法作为一种有效和可靠的方法，将继续发挥其重要的作用。

对于后续的研究，目前而言的重心仍放在较为复杂的基于矩阵的特征选择算法上。通过本文的对比来看，两个基于矩阵的特征选择算法在准确度上有着自己的优势。尤其随着数据的爆炸式增长，人们必定将要求模型可以处理更多的特征数，人们也有足够强大的算力去应用一些更复杂，更准确的模型。所以主要发展基于矩阵的特征选择算法是必然的。

而在对未来后续研究的展望上，仍要关注稀疏学习中正则化器的选择。目前，矩阵范数，特别是 $l_{2,1}$ 范数和 $l_{2,1-2}$ 范数，在特征选择中已经显示出很好的性能和效果。 $l_{2,1-2}$ 范数的组合特性使其在一些应用场景中表现得更为出色，这也暗示了寻找更多有效的范数或范数组合的潜在价值。

未来研究的一个重要方向可以是探索和发展新的范数或范数组合，以进一步优化稀疏学习算法的性能和鲁棒性。这需要结合数学理论和实际应用需求，通过理论分析和实验验证来评估新的正则化器的有效性。此外，随着数据规模的增大和应用场景的复杂性，大规模数据实验的重要性也日益凸显。通过大规模数据实验，研究人员可以更全面地评估新的正则化器在不同数据集和应用场景中的性能，从而为其优化和改进提供有力的支持。在此过程中，研究人员可以将稀疏学习与深度学习结合起来，各取所长^[15]。

总之，未来的稀疏学习研究将需要跨学科的合作，结合数学、计算机科学和应用领域的知识，以实现算法的持续优化和应用扩展。这将是一个充满挑战和机遇的研究领域，也将为解决问题和推动技术进步提供有力的支持。

参考文献

- [1] 张红英, 董珂臻. 稀疏统计学习及其最新研究进展综述[J]. 西南师范大学学报 (自然科学版), 2023, 48(04): 1-12.
- [2] QIAO L B, ZHANG B F, SU J S, LU X C. Review: A systematic review of structured sparse learning[J]. Frontiers of Information Technology & Electronic Engineering, 2017, 18.
- [3] GUI J, SUN Z, JI S, TAO D, TAN T. Feature Selection Based on Structured Sparsity: A Comprehensive Study[J]. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 2017, 28.
- [4] TIBSHIRANI R. Regression Shrinkage and Selection Via the Lasso[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1996, 58.
- [5] 周志华. 机器学习[M]. 清华大学出版社, 2016.
- [6] ZOU H, HASTIE T. Regularization and Variable Selection Via the Elastic Net[J]. Journal of the Royal Statistical Society: Series B (Methodological), 2005, 67.
- [7] 李玮琢. 基于回归和分类的混合线性模型研究[J]. 信息记录材料, 2022, 23(08): 7-11.
- [8] NIE F, HUANG H, CAI X, DING C. Efficient and Robust Feature Selection via Joint $2,1$ -Norms Minimization[J]. International Conference on Neural Information Processing Systems, 2010.
- [9] SHI Y, MIAO J, WANG Z, ZHANG P, NIU L. Feature Selection With $l_{2,1-2}$ Regularization[J]. IEEE, 2018, 29.
- [10] <https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format?rvi=1>, Last accessed on 2024-5-12. 2018.

- [11] <https://www.kaggle.com/datasets/fedesoriano/chinese-mnist-digit-recognizer/data>, Last accessed on 2024-5-12. 2021.
- [12] 李占山, 姚鑫, 刘兆赓, 张家晨. 基于 LightGBM 的特征选择算法[J]. 东北大学学报 (自然科学版), 2021, 42(12): 1688-1695.
- [13] 王岭. 稀疏逻辑回归在银行信贷业务中的应用[D]. 大连理工大学, 2021.
- [14] 赵雨佳. 基于稀疏学习的神经退行性疾病早期诊断方法研究[D]. 深圳大学, 2020.
- [15] 袁正鹏. 结构稀疏深度学习算法研究[D]. 北京交通大学, 2021.

致 谢

感谢LUG@NJU提供的 LaTeX 编译环境支持, jupyter notebook 和 Python 为我提供了强大的代码编译环境, 使得我的研究和写作变得更加高效和便捷。此外, Kaggle平台上的数据集为我的研究提供了宝贵的资源和参考, 为我完成论文研究提供了坚实的基础。

短短四年南大时光渐进尾声, 似乎青春也要在跌跌撞撞中慢慢走向终点。人生路上又一次征途开始, 而我只得背起行囊, 回首间似乎还能看见那个稚嫩的, 兴奋地拿着南京大学录取通知书的自己。在南大的四年里, 我深入数学系的学习, 经历了无数的学术探索和生活冒险。这段时间, 我曾多次迷茫, 质疑自己的选择, 面对学业的压力和挑战时, 也曾经历过痛苦和困惑。但无论何时, 我都没有怀疑过南京大学这个平台所带给我的宝贵成长机会。它不仅为我提供了广阔的学术视野和严谨的学术氛围, 更让我有机会结交志同道合的朋友, 与优秀的教授们共同学习、共同进步。未来的日子里, 无论身在何处, 我都会时常回想起这个我称之为母校的地方, 回想起在这里度过的美好时光和接受的深刻教诲。它将永远成为我成长路上的坐标, 指引我前行的方向。谢谢你, 南京大学。

同时, 在这里我也要深深地感谢我的家人和朋友们。他们对我的支持和帮助不仅仅是物质上的, 更是精神上的支柱, 是我前行路上最坚实的后盾。每当我遇到困难和挑战, 感到疲惫和无助时, 家人总是第一时间站在我的身边, 给我无私的关心和鼓励。他们的关心和爱意像一道明亮的灯塔, 指引我走出困境, 重拾信心。更为珍贵的是, 有一群同学们与我一同走过这段人生旅程, 我们共同学习、共同进步, 互相激励, 相互支持。每当我遇到困难和挑战, 看到他们努力拼搏, 我就会被他们的精神所感染, 激发出我内心的斗志和坚持。我们一起分享喜悦, 一起面对挑战, 一起努力追求梦想, 这段美好的时光将成为我人生中最宝贵的回忆。

最后, 我要向所有教授过我的老师们致以最深厚的感谢。他们是我学术道路上的灯塔, 每一位都充满智慧和慈爱, 教会我不仅是专业知识, 更是如何成为一

个真正有用之人的道理。他们的教诲不仅丰富了我的学识，更塑造了我为人处世的态度和价值观，成为我人生中不可或缺的一部分。特别要提及的是，我的毕业论文的指导老师，王征宇副教授。他耐心地指导我完成开题报告，提供了宝贵的建议和批评，让我能够不断地完善和深化我的研究方向。在整个论文写作的过程中，王老师始终如一地给予我鼓励和支持，为我答疑解惑，让我顺利完成了毕业论文，更让我在这个过程中收获了无尽的成长和宝贵的经验。

回想起那个年少轻狂的自己，总是眼里闪烁着对未知世界的好奇与向往。而如今未来的路就在眼前，却又不禁迟疑起来。与小时候的憧憬相比，现实中的世界似乎更加复杂、更加多变，充满了未知和挑战。社会的竞争激烈，环境的变迁，都让我感到有些迷茫和不安。然而，在这样的迷茫和挣扎中，我深知自己最应该做的是坚守本心，相信自己的能力，相信自己的选择，找到适合自己发展的方向。

走在遍地都是六便士的大道上，至少也偶尔抬起头来看看月亮吧。