|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **分类号** | **TP181** | **密级** | **公开** | |
|  | **UDC** |  | **学位论文编号** |  | |
|  | | | | | |
| **重庆邮电大学硕士学位论文** | | | | | |
|  | | | | | |
|  | **中文题目** | **核逻辑回归算法研究及其应用** | | |  |
| **英文题目** | **Research and Application of Kernel** | | |
|  | **Logistic Regression Algorithm** | | |
| **学 号** | **S170231053** | | |
| **姓 名** | **唐建烊** | | |
| **学位类别** | **工程硕士** | | |
| **学科专业** | **计算机技术** | | |
| **指导教师** | **雷大江 副教授** | | |
| **完成日期** | **2020年3月6日** | | |
|  | | | |

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含他人已经发表或撰写过的研究成果，也不包含为获得重庆邮电大学或其他单位的学位或证书而使用过的材料。与我一同工作的人员对本文研究做出的贡献均已在论文中作了明确的说明并致以谢意。

|  |  |
| --- | --- |
| 作者签名： | 日期： 年 月 日 |

学位论文版权使用授权书

本人完全了解重庆邮电大学有权保留、使用学位论文纸质版和电子版的规定，即学校有权向国家有关部门或机构送交论文，允许论文被查阅和借阅等。本人授权重庆邮电大学可以公布本学位论文的全部或部分内容，可编入有关数据库或信息系统进行检索、分析或评价，可以采用影印、缩印、扫描或拷贝等复制手段保存、汇编本学位论文。

（注：保密的学位论文在解密后适用本授权书。）

|  |  |
| --- | --- |
| 作者签名： | 导师签名： |
| 日期： 年 月 日 | 日期： 年 月 日 |

**摘要**

**关键词：**

**Abstract**

**Keywords**:

目录

[图录 VII](#_Toc10641577)

[表录 VIII](#_Toc10641578)

[注释表 IX](#_Toc10641579)

[第1章 引言 1](#_Toc10641580)

[1.1 选题背景与研究意义 1](#_Toc10641581)

[1.2 国内外研究现状 3](#_Toc10641582)

[1.2.1 逻辑回归的研究现状 3](#_Toc10641583)

[1.2.2 交替方向乘子法的研究现状 4](#_Toc10641584)

[1.3 论文主要研究内容 5](#_Toc10641585)

[1.4 论文的结构安排 5](#_Toc10641586)

[第2章 相关技术分析 7](#_Toc10641587)

[2.1 逻辑回归算法 7](#_Toc10641588)

[2.2 多元逻辑回归算法 8](#_Toc10641589)

[2.3 多元逻辑回归正则化算法 8](#_Toc10641590)

[2.3.1 权重衰减多元逻辑回归 8](#_Toc10641591)

[2.3.2 稀疏多元逻辑回归 9](#_Toc10641592)

[2.4 自适应矩估计算法 10](#_Toc10641593)

[2.5 交替方向乘子算法 10](#_Toc10641594)

[2.6 本章小结 11](#_Toc10641595)

[第3章 极大不相关多元逻辑回归算法 13](#_Toc10641596)

[3.1 问题提出与研究思路 13](#_Toc10641597)

[3.2 极大不相关多元逻辑回归算法 14](#_Toc10641598)

[3.2.1 基于多元逻辑回归算法的改进 14](#_Toc10641599)

[3.2.2 求解算法时间复杂度分析 17](#_Toc10641600)

[3.3 算法扩展 17](#_Toc10641601)

[3.3.1 多元逻辑回归与神经网络的关系 17](#_Toc10641602)

[3.3.2 极大不相关多元逻辑回归扩展的神经网络 18](#_Toc10641603)

[3.4 实验与结果分析 18](#_Toc10641604)

[3.4.1 数据集介绍 18](#_Toc10641605)

[3.4.2 极大不相关多元逻辑回归算法实验 19](#_Toc10641606)

[3.4.3 极大不相关神经网络算法实验 23](#_Toc10641607)

[3.5 本章小结 24](#_Toc10641608)

[第4章 大规模极大不相关多元逻辑回归算法 26](#_Toc10641609)

[4.1 问题提出与研究思路 26](#_Toc10641610)

[4.2 大规模场景下的极大不相关多元逻辑回归 27](#_Toc10641611)

[4.2.1 极大不相关多元逻辑回归的一致性求解算法 27](#_Toc10641612)

[4.2.2 极大不相关多元逻辑回归的共享求解算法 29](#_Toc10641613)

[4.2.3 求解算法时间复杂度分析 30](#_Toc10641614)

[4.3 实验与结果分析 31](#_Toc10641615)

[4.3.1 运行环境与数据集介绍 31](#_Toc10641616)

[4.3.2 一致性求解算法的实验对比 32](#_Toc10641617)

[4.3.3 共享求解算法的实验对比 34](#_Toc10641618)

[4.4 本章小结 35](#_Toc10641619)

[第5章 大规模文本分类系统的设计与实现 36](#_Toc10641620)

[5.1 大规模文本分类中存在的问题与解决方法 36](#_Toc10641621)

[5.2 文本分类的一般过程 36](#_Toc10641622)

[5.3 仿真系统的设计与搭建 39](#_Toc10641623)

[5.3.1 软硬件环境要求 39](#_Toc10641624)

[5.3.2 文本表示与特征提取 39](#_Toc10641625)

[5.3.3 分类器训练与预测 41](#_Toc10641626)

[5.4 系统展示 42](#_Toc10641627)

[5.4.1 文本分类系统的界面展示 42](#_Toc10641628)

[5.4.2 文本分类系统的结果分析 45](#_Toc10641629)

[5.5 本章小结 47](#_Toc10641630)

[第6章 总结与展望 48](#_Toc10641631)

[6.1 研究工作总结 48](#_Toc10641632)

[6.2 未来研究展望 48](#_Toc10641633)

[参考文献 49](#_Toc10641634)

[致谢 53](#_Toc10641635)

[攻读硕士学位期间从事的科研工作及取得的成果 54](#_Toc10641636)

# 图录

[图3.1 MNIST数据集中手写体1和7 14](#_Toc4955098)

[图3.2 MUMLR算法在人工数据集上的识别率 20](#_Toc4955099)

[图3.3 不同算法在各公开数据集上的收敛性 22](#_Toc4955100)

[图3.4 不同算法模型参数的范数大小 23](#_Toc4955101)

[图4.1 一致性极大不相关多元逻辑回归识别率 33](#_Toc4955102)

[图4.2 一致性极大不相关多元逻辑回归计算时间 34](#_Toc4955103)

[图4.3 共享极大不相关多元逻辑回归识别率 35](#_Toc4955104)

[图4.4 共享极大不相关多元逻辑回归计算时间 35](#_Toc4955105)

[图5.1 文本分类器功能模块 37](#_Toc4955106)

[图5.2 文本表示流程图 37](#_Toc4955107)

[图5.3 文本分类器系统流程 38](#_Toc4955108)

[图5.4 集群架构图 42](#_Toc4955109)

[图5.5 分词界面 43](#_Toc4955110)

[图5.6 训练界面 43](#_Toc4955111)

[图5.7 参数调节对话框 44](#_Toc4955112)

[图5.8 预测界面 44](#_Toc4955113)

[图5.9 MUMLR算法预测展示 46](#_Toc4955114)

# 表录

[表3.1 不同算法在人工数据集上的识别率 19](#_Toc9779202)

[表3.2 参数在不同相关度数据集上的近似最优取值 21](#_Toc9779203)

[表3.3 不同算法在公开数据集上的识别率 21](#_Toc9779204)

[表3.4 不同算法在SinaNews文本分类数据集上的分类效果 21](#_Toc9779205)

[表3.5 不同算法在MNIST数据集手写体1和7上的误分率 21](#_Toc9779206)

[表3.6 DNN模型的神经网络结构 24](#_Toc9779207)

[表3.7 不同DNN神经网络算法的识别率 24](#_Toc9779208)

[表4.1 数据集信息 31](#_Toc4955121)

[表5.1 基本词性标记集 40](#_Toc4955122)

[表5.2 分词及词性标注样例 40](#_Toc4955123)

[表5.3 MUMLR算法在不同并行度下的效果 45](#_Toc4955124)

[表5.4 不同并行化分类算法的比较 46](#_Toc4955125)

# 注释表

|  |  |
| --- | --- |
| LR | Logistic Regression，逻辑回归 |
| MLR | Multinomial Logistic Regression，多元逻辑回归 |
| SLR | Sparse Logistic Regression，稀疏逻辑回归 |
| ELR | Elastic-net Logistic Regression，弹性网逻辑回归 |
| ALLR | Adaptive Lasso Logistic Regression，自适应稀疏逻辑回归 |
| SMLR | Sparse Multinomial Logistic Regression，稀疏多元逻辑回归 |
| WDMLR | Weight-decay Multinomial Logistic Regression，权重衰减多元逻辑回归 |
| MUMLR | Maximal Uncorrelated Multinomial Logistic Regression，极大不相关多元逻辑回归 |
| ADMM | Alternating Direction Method of Multipliers，交替方向乘子法 |

# 第1章 引言

## 1.1 选题背景与研究意义

随着信息化时代的发展，网络中的数据规模不断的扩大，并且呈几何增长。随之带来的就是技术人员需要对大数据进行收集、整理和分析。大数据时代引领的技术变革正在悄无声息地改变着各行各业。随着技术的发展与革新，现在人们可以使用大数据技术来处理海量的数据了，这使得很多之前只能停留在理论研究层面的算法和思想现在都能够付诸行动。然而，大量的数据必然存在着一些冗余信息，信息的价值密度也较低。因此，我们需要对数据进行高效的处理，并从中挖掘出潜在的价值。

在常规的大数据处理流程中，对数据进行有效的分类是其中重要的一个部分。常规的机器学习分类算法都可以将数据根据不同的特征去判别所属的类别，例如SVM、逻辑回归算法、决策树和朴素贝叶斯算法等。但是，在大规模数据的工业级别应用中，考虑到算法的稳定性、时间复杂度以及分类效果等各种因素，逻辑回归是最为广泛和常见的分类算法。它通过最大化似然函数来生成决策边界，最终将样本数据进行有效的分类，它训练起来非常的高效，不需要太久的计算，并且其数据特征输入也很好调整。但是，逻辑回归它生成的决策边界是线性的，因此不能解决非线性分类问题。

对于上面提到的非线性分类问题，常用的解决办法就是将非线性数据特征映射到高维空间上，使其在这个高维特征空间上可以线性可分。这种方法就被称为核方法。目前，核方法被广泛的与其他传统机器学习算法相结合去解决非线性分类的问题，最为常见的就是核SVM算法。但是，将核方法于逻辑回归算法相结合的核逻辑回归算法相比与核SVM应用场景就少很多。其主要原因是因为加核以后的逻辑回归算法计算复杂度太高，其每个样本之间都需要去计算一次核函数，而支持向量机则只有支持向量起作用，其自带稀疏性，因此只需计算少量的支持向量的核函数即可。综上，如果要将核方法应用到逻辑回归算法中，就应该避免将核矩阵参与运算，在优化算法上应该选择适当的高效算法。另外，由于核矩阵的大小是跟数据样本成正相关，随着数据规模的不断扩大，核矩阵的计算也会消耗大量的时间，为了提高核逻辑回归算法整体的效率，也可以考虑对核矩阵进行近似求解，减少时间复杂度。

另一方面，核函数的选择也存在许多的问题。单一的核函数虽然能解决非线性问题，但是不同的样本数据具有不同的分布概率，如果只采用一种核函数，就很难让数据在高维空间上充分的表达。针对以上问题，我们可以利用多核学习来组合不同类型或不同参数的核函数，从而让样本数据在特种空间上更加的合理。

随着遥感技术的发展，各种传感器每天都能采集到大量的遥感图像，人们可以从中获得大量的有用数据和信息。高分辨率的遥感图像能提供丰富的光谱信息和空间纹理信息，在资源调查、自然灾害检测、天气气象预告、土地利用分类和国防安全上有着广泛的应用。但是由于遥感图像数据量大，目标类型多样且图像会受天气、光线等因素的影响有大量的噪点，传统的依靠人工来对大量遥感图像中的地物进行识别检测和分类显然是不可行的。

目前对遥感图像地物分类存在训练样本特征维度高，异构信息冗杂，含有大量的噪声波段和非线性结构数据，传统的分类算法很难达到精度要求。因此，通过多核学习将不同类型的遥感图像特征映射到不同的维度空间中，然后再进行有效且准确的分类。

## 1.2 国内外研究现状

### 1.2.1 核方法的研究现状

近十多年来，由Vapnik等提出的支持向量机(Support Vector Machine, SVM)被广泛的应用于分类任务中，核方法最早就是应用在SVM上，通过引入核函数来解决非线性分类问题。在SVM获得巨大的成功中，核方法扮演了非常重要的角色。进一步的，核方法被应用到了其他传统的学习算法中去，可以很方便的把线性算法转化为非线性算法。例如：核主成分分析(Kernel PCA)、核Fisher判别分析(Kernel FDA)、核聚类分析等等，这些基于核的学习算法都简称为核方法。只采用一个基本核的学习方法称之为单核学习，而相比于单个核函数，多核模型具有更高的灵活性。多核学习方法根据不同的分类标准有不同的分类方式，按照多核函数的构造方法和特点的不同，可以将多核学习方法大致分成三大类别：合成核方法、多尺度核方法、无限核方法。而多核学习理论早期大体上分为：基于Boosting的多核组合模型学习方法，基于半定规划(Semi-definite programming, SDP)的多核学习方法, 基于二次约束型二次规划(Quadratically constrained quadratic program, QCQP)的学习方法，基于半无限线性规划(Semi-definite linear program, SILP)的学习方法和基于超核(Hyper kernels)的学习方法等。接着为了达到更好的收敛效果和得到更好的解，2012年Vishwanathan又提出SPG-GMKL算法，同时提出了多核的Product组合。而近年来出现的简单多核学习(Simple MKL)方法以及基于中心对齐的学习核算法(Algorithms for Learning Kernels Based on Centered Alignment)都是比较流行的多核学习方法。而在视觉识别任务中，通过在统一的内核空间中加入稀疏模型往往能取得很好的效果。

### 1.2.2 核逻辑回归的研究现状

逻辑回归(Logistic Regression, LR)算法是统计分析、机器学习和数据挖掘领域的一个经典分类算法，用于估计事件发生的可能性，输出等价于模型预测某个数据点属于正类的概率估计。它是一种线性的分类算法，具有求解速度快、预测结果可解释性强的特点，它为探索性解释数据提供了一个有用的概率模型。但是，一般的LR只能用于线性分类，对于非线性的特征数据分类就十分的困难。Logistic回归经过核化扩充之后的分类器称之为核Logistic回归（Kernel logistic regression, KLR）,这样就变成了可用于非线性特征数据分类的分类器。通过将原数据的样本映射到一个高维或无穷维的特征空间，使得样本在这个特征空间内线性可分。虽然此过程的映射规则通常是不可知的，但是可以用核函数(Kernel function)去代替特征空间上的内积运算，因此并不需要知道此特征空间里特征向量的具体表达形式。

但是由于引入了核矩阵，迭代求解KLR变得非常的缓慢。受到Platt的序列最小最优算法(Sequential Minimal Optimization, SMO)的启发，Keerthi基于对偶理论提出了求解KLR的快速对偶算法，避免了将核矩阵带入到迭代过程中去运算，从而极大的减少了计算开销。但是随着数据量的不断增大，核矩阵的计算也会影响整个算法的开销，并且将整个数据样本都通过核函数映射到高维空间上去，会带入一些冗余信息。因此，矩阵的近似求解方法被广泛的应用在了求解核矩阵上，如：Nystrom方法，Fastfood方法等。通过这些方法可以加速核矩阵的求解，利用低秩近似又可以去掉原始数据中的一些冗余信息。

## 1.3 论文主要研究内容

本文首先介绍了逻辑回归算法，随后又引入了核方法去解决非线性分类的问题，然后考虑到核方法引入的核矩阵会影响算法的效率，于是又利用了低秩近似方法去加速求解核矩阵，并采用快速对偶算法进行优化求解，于是提出了低秩近似核逻辑回归算法。接着从分类的角度将逻辑回归算法扩充到了可用于多分类和稀疏模型的稀疏多元逻辑回归算法，考虑到单核学习的局限性，于是又结合了多核学习，提出了基于中心对齐多核学习的稀疏多元逻辑回归算法。最后针对遥感图像数据的异构特征，我们将多核稀疏多元逻辑回归算法应用到遥感图像分类中去，并取得了较好的分类效果。

本文主要创新点如下：

1. 联合低秩近似方法提出了低秩近似核逻辑回归算法，在解决了非线性分类问题的同时还加速了核逻辑回归算法的求解过程，并且通过低秩近似方法去掉了数据中的冗余信息，提高了算法的分类准确率。
2. 结合多核学习提出了基于中心对齐多核学习的稀疏多元逻辑回归算法，通过多个核函数的线性组合解决了单核学习中核函数单一的问题，并且引入了稀疏正则化，使其适用于稀疏特征的非线性多分类问题。
3. 将多核稀疏多元逻辑回归算法应用到了遥感图像分类上。在多分类公共数据集和遥感图像地物分类数据集上都取得了较好的分类效果。

## 1.4 论文的结构安排

本文对核逻辑回归算法进行研究，并将其应用到遥感图像分类上，论文章节安排如下：

第1章：引言。主要介绍了论文的研究背景及发展现状，通过分析核方法以及逻辑回归算法的研究进展，针对目前研究中的问题和难点，确定了本文的主要研究内容。

第2章：相关技术分析。本章首先介绍了逻辑回归算法和稀疏多元逻辑回归算法，接着又介绍了优化算法，其中主要包括基于一阶迭代的梯度下降算法和基于二阶迭代的牛顿迭代法以及为了加速算法求解效率提出的快速对偶算法。然后又介绍了用于解决非线性问题的核方法，以及用于加速求解核矩阵的低秩近似方法，最后为了解决核函数单一的问题又介绍了多核学习算法。

第3章：低秩近似核逻辑回归算法。本章首先引出了用于求解非线性问题的核方法和逻辑回归算法相结合后利用常见的迭代优化算法求解存在时间复杂度太高的问题，然后提出了一种快速对偶优化算法去解决上述问题。接着，考虑到核矩阵计算开销大的问题，又引入了低秩近似算法，通过将低秩近似算法与快速对偶优化算法相结合，从而提出了低秩近似核逻辑回归算法，并且证明了其收敛性。最后，通过不同类型和不同规模的数据集与其他算法进行了对比，证明了本章节的算法能取得更好的分类效果。

第4章：基于中心对齐多核学习的稀疏多元逻辑回归算法。本章总结了前面核逻辑回归算法存在的问题，将其扩展成了适应于稀疏数据集的多分类算法。同时还通过将多个不同类型和不同参数的核函数进行线性组合，解决了核函数单一，不能对样本特征进行充分表达的问题。另外，由于目标函数中引入了稀疏正则项，于是我们采用了稀疏优化算法进行求解。最后通过实验证明了多核稀疏多元逻辑回归算法在多分类数据集上能够取得更好的分类效果。

第5章：遥感图像分类系统的设计与实现。 本章将第4章的算法应用到了遥感图像分类中，实现了一个简易的遥感图像分类系统，从实际应用中证明了前面所提出的算法的实际应用价值。

第6章：总结与展望。首先对本文的研究内容进行了总结，简述了低秩近似核逻辑回归算法以及多核稀疏多元逻辑回归算法解决的问题。然后针对本文研究工作的不足和研究工作中发现的问题进行了解释和展望。

# 第2章 相关技术

## 2.1 逻辑回归算法

对于逻辑回归,其假设有数据集，输入向量为，类标签为二值函数。由于二分类结果是0或者1，这与数学的阶跃函数很类似，但是阶跃函数在的位置会发生突变，这个突变在数学上很难处理。所以一般使用sigmoid函数来拟合：

|  |  |
| --- | --- |
|  | (2.1) |

上式表示正类的预测函数为。在这里，假设服从伯努利分布，取值为0和1，那么得到下面两个式子：

|  |  |
| --- | --- |
|  | (2.2) |

对于上面的两个表达式，可以将其合并为以下表达式：

|  |  |
| --- | --- |
|  | (2.3) |

假定给的样本是独立的，便可以构造出似然函数，然后使用极大似然估计(Maximum Likelihood, ML)的思想来求解参数。但是，为了满足最小化风险理论，可以将MLE的思想转化为最小化风险理论，最大化似然函数其实就等价于最小化负的似然函数。使用MLE推导的过程如下:

|  |  |
| --- | --- |
|  | (2.4) |

直接对上面的式子求导会不方便，为了便于计算，对似然函数取对数：

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

损失函数便可以通过最小化负的似然函数得到，即：

|  |  |
| --- | --- |
|  | (2.6) |

因此，我们最终通过最小化损失函数就可以得到我们所需要的最优参数，然后通过预测函数来对每一个样本计算其所属类别的概率，通过比较正负类别的概率值大小来确定样本最终所属的类别。但是，逻辑回归算法只能解决二分类问题，不能直接用于多类别数据分类。在多类别分类任务中，二分类算法通常可以采用one-vs-rest或者one-vs-one的策略进行多类别分类，但是会受到样本不平衡的影响。

## 2.2 稀疏多元逻辑回归算法

为了解决多分类任务，我们需要把逻辑回归算法扩充成多元逻辑回归算法。同时，考虑到某些特定领域的数据具有高纬度的特征，所以我们将拉普拉斯先验引入多元逻辑回归中可以使其解具有稀疏性，这让它可以在进行分类的过程中嵌入特征选择，所以在高维数据和稀疏数据集的处理上都具有很大的优势。

假设数据集包含个样本，个特征，其中，，为类别数。为One-Hot编码后的标签矩阵，即每个标签对应一个维向量，该向量中仅有1个元素取值为1，其他元素取值均为0。对于给定一个样本和标签，SMLR的损失函数：

|  |  |
| --- | --- |
|  | (2.7) |

这里为示性函数，且有，。

一般而言，解的稀疏性通常通过引入范数来获得，但求解问题是NP难的组合优化问题。因此，通常的做法是对问题进行松弛优化，即求解凸优化问题。

于是对于给定一个样本和标签，样本属于类别的概率可以表示为

|  |  |
| --- | --- |
|  | (2.8) |

也一定有

|  |  |
| --- | --- |
|  | (2.9) |

同理，我们可以通过优化算法求解损失函数，得到最优参数，从而得到每个样本的类别概率。

## 2.3 优化算法

针对上面的损失函数，我们需要采用优化算法去求解最优参数，从而才能得到最终的分类结果。一般地，如果损失函数是凸函数，我们通常可以采用一阶梯度下降算法、二阶牛顿迭代算法或快速对偶算法；如果损失函数是含有正则项的非凸函数，则可以采用迭代软阈值收缩算法(Iterative Shrinkage Thresholding Algorithm, ISTA)，也称作近端梯度下降法(Proximal Gradient Method, PGM)来求解，除此之外，针对大规模的样本也可以采用快速迭代软阈值收缩算法(Fast Iterative Shrinkage-Thresholding Algorithm, FISTA)去加速求解。

### 2.3.1 梯度下降法

梯度下降算法是机器学习中应用最为广泛的优化算法，它的主要目的是通过迭代运算找到目标损失函数的最小值，或收敛到最小值。

如果我们有以下损失函数

|  |  |
| --- | --- |
|  | (2.10) |

其中，拟合函数

|  |  |
| --- | --- |
|  | (2.11) |

针对以上的损失函数，我们需要将其负梯度方向作为迭代的下降方向，所以对

式（2.10）进行求导得到迭代方向

|  |  |
| --- | --- |
|  | (2.12) |

这里表示第个样本点的第个分量。接着我们便可以按照的负梯度方向来更新每一个，再规定每一次迭代的步长为，且，从而有以下迭代公式

|  |  |
| --- | --- |
|  | (2.13) |

以上算法每次迭代只选取了一个样本，所以被称之为随机梯度下降算法。但是由于样本中存在噪音的影响，所以并不是每一次迭代都能向着整体的优化方向，从而会导致最终求得的最优解不是全局最优解，而只是局部最优解。因此，我们通常在迭代的时候会随机选择一批样本来决定梯度的方向，这种方法被称之为随机批量梯度下降算法。这样更容易得到全局最优解，同时也易于并行化。

对于梯度下降算法，每一次的迭代更新都是以负梯度作为搜索方法前进一个步长，因此最终一定是线性收敛的。当迭代次数趋于无穷时，代价函数收敛到最优解，其收敛速率为。

### 2.3.1 牛顿迭代法

牛顿迭代法是利用迭代点处的一阶导数（梯度）和二阶导数（Hessian矩阵）对目标函数进行二次函数近似，然后把二次模型的极小点作为新的迭代点，并不断重复这一过程，直至求得满足精度的近似最小值。

假设优化函数是二阶连续可微函数，。并且为函数的极小点的一个近似，于是将在附近做泰勒展开，

|  |  |
| --- | --- |
|  | (2.14) |

这里，表示函数的梯度向量，表示函数的Hessian矩阵。假若矩阵是正定的，则有唯一极小值点，于是就可以将作为迭代的下一次近似值。于是，最终就有牛顿迭代法的迭代公式

|  |  |
| --- | --- |
|  | (2.15) |

从本质上去看，牛顿迭代法则是采用Newton步径为搜索方向，而Newton步径也是x处采用Hessian矩阵定义的二次范数。因此，一般情况下Newton方法收敛很快，在附近二次收敛，一旦进入二次收敛阶段，只需要再经过少量的迭代就可以产生具有很高精度的解。所以二阶的牛顿迭代法比一阶的梯度下降法在优化求解速度上更快。

### 2.3.1 快速对偶算法

前面我们提到的优化算法都是通过迭代运算最终达到损失函数的最小值。但是随着样本的不断增大，每次参与迭代运算的矩阵都会消耗大量的时间。因此为了在某些特定的应用环境中避免上述问题，我们可以采用快速对偶算法进行优化求解。其主要思想是利用朗格朗日乘子法将其原问题转化成对偶问题，然后在对偶问题上进行优化求解。在求解过程中，每次迭代只需要选取其中两个违反约束条件的参数进行更新，直到所有的参数都满足约束条件或者达到某个终止条件时就结束更新。这样就避免了将整个样本矩阵参与迭代运算，从而减少了算法的时间复杂度。

## 2.4 核方法

一般的逻辑回归算法无法处理非线性特征数据的分类，因此我们需要将逻辑回归与核方法相结合去处理非线性分类的问题。核方法在近年来机器学习领域十分流行，尤其是基于统计学习理论的支持向量机的成功，由此产生了多种基于核函数的方法，并且被广泛地应用于模式识别、文本分类、信号处理等多个领域。核方法通过将原数据的样本映射到一个高维或无穷维的特征空间，使得样本在这个特征空间内线性可分。虽然此过程的映射规则通常是不可知的，但是可以用核函数(Kernel function)去代替特征空间上的内积运算，因此并不需要知道此特征空间里特征向量的具体表达形式。

假设有数据集输入向量为, 为类别标签。 根据表达定理：，此处的为在特征空间的投影。于是将特征投影到高维空间有：

|  |  |  |
| --- | --- | --- |
|  | = | (2.5) |

这里的是核矩阵的第i行向量。如果我们的损失函数中还引入了正则项，则还有

|  |  |
| --- | --- |
|  | (11) |

这样，我们通过核方法就可以将逻辑回归算法扩充成核逻辑回归算法，于是便可以解决非线性数据分类问题了。

## 2.5 低秩近似方法

随着互联网技术的不断发展，数据规模的不断扩大，核方法的应用越来越受限制。其中一个关键的问题是核矩阵通常都是稠密矩阵，其存储和计算代价都非常的高，存储稠密矩阵需要的空间，而计算这样的矩阵则需要的代价，这里和分别代表了样本的个数和维度。最常用的解决方法是用有限的内存来计算得到一个近似的核矩阵，这种方法不仅解决了内存问题，也加快了核矩阵的计算。目前已有多种核近似方法被提出，其中Nystrom方法的应用最为广泛，也是之后很多求解近似矩阵算法的基础。Nystrom方法的主要思想是通过降低矩阵的秩来得到原始矩阵的近似矩阵，相当于从中随机取行列，则,其中是原始矩阵K中的块矩阵，而。而最终得到的近似矩阵是低秩的，这样不仅保留了原矩阵的主要特征，还降低了核矩阵的存储空间和计算时间的复杂度。

当输入样本映射到高纬特征空间，为了计算特征空间里的内积，假定在原空间中用协方差核去代替：

|  |  |
| --- | --- |
|  | (11) |

这里表示核矩阵的特征值，表示算子的特征函数，于是有：

|  |  |
| --- | --- |
|  | (11) |

表示原空间输入向量的概率密度，而特征函数是正交的，令：

|  |  |
| --- | --- |
|  | (11) |

从中取独立同分布的样本然后用均值去代替上的积分：

|  |  |
| --- | --- |
|  | (11) |

再由正交的特征函数得到约束条件 ,因此，就有：

|  |  |
| --- | --- |
|  | (11) |

这里是的核矩阵，列是正交的，是元素为的对角矩阵，其中，于是可以得到相关近似：

|  |  |
| --- | --- |
|  | (11) |

综合以上式子，最终得到Nystrom方法对第i个特征函数的近似：

|  |  |
| --- | --- |
|  | (11) |

其中，是矩阵中的第i列。上式也可以看成是在特征空间中将一个新的输入投影到了第i个特征向量上。

## 2.6 多核学习算法

核方法通过将原数据的样本映射到1个或多个高维甚至无穷维的特征空间，使得样本在这个特征空间内线性可分。但是这种单核学习的核函数只有1个，其结构单一, 为了将数据映射到不同的高维空间中从而使得这些数据特征能够更好地表达，于是我们在单核学习的基础上扩充成了多核学习。通过将多个不同类型或不同参数的核函数的线性组合来形成新的核函数，从而使得数据样本能更加充分的表达。目前，多核学习的经典方法有SimpleMKL和SPF-GMKL等，而基于中心对齐的学习核算法(Algorithms for Learning Kernels Based on Centered Alignment)是当前比较流行的多核学习方法。该方法通过中心对齐思想学习能得到一组核函数的权重系数，由此可以更灵活的组合出新的核函数。

## 2.7 本章小结

在本章中，我们首先从二分类的逻辑回归入手，将其扩展成为了适应高维特征的稀疏多元逻辑回归，并且根据不同类型的损失函数和数据量的大小，分别引出了不同的优化算法进行优化求解。接着又考虑到非线性数据的分类问题，于是我们又引入了核方法，通过将两者相结合解决了非线性多分类问题。最后，我们还考虑到了单核学习的局限性，于是又将其扩充成了多核学习。

# 第3章 低秩近似核逻辑回归算法

## 3.1 问题提出与研究思路

逻辑回归(Logistic Regression, LR)算法是统计分析、机器学习和数据挖掘领域的一个经典分类算法，主要针对二分类问题提出，用于估计事件发生的可能性，输出等价于模型预测某个数据点属于正类的概率估计。它是一种线性的分类算法，具有求解速度快、预测结果可解释性强的特点，它为探索性解释数据提供了一个有用的概率模型。但是，一般的LR只能用于线性分类，对于非线性的特征数据分类就十分的困难。Logistic回归经过核化扩充之后的分类器称之为核Logistic回归（Kernel logistic regression, KLR）, 这样就变成了可用于非线性特征数据分类的分类器。通过将原数据的样本映射到一个高维或无穷维的特征空间，使得样本在这个特征空间内线性可分。虽然此过程的映射规则通常是不可知的，但是可以用核函数(Kernel function)去代替特征空间上的内积运算，因此并不需要知道此特征空间里特征向量的具体表达形式。

相比于支持向量机，KLR的目标函数不是风险最小函数，而是最大似然值。因此KLR的解会产生类别分类的后验概率值，同时KLR也是一个凸优化问题，局部最优解一定是全局最优解。而对于一个凸优化问题，可以运用梯度法或牛顿迭代法进行求解，但是该两种方法在每次迭代过程中都需要对一个的核矩阵进行求逆等运算，其中表示样本个数，通常的数量达到几千个时，计算时间代价就变得非常的大，甚至难以接受，其时间复杂度为。在支持向量机的凸二次规划问题求解中，Platt等人提出了序列最小最优化算法（Sequential minimal optimization, SMO），基于此算法思想的启发，Keerthi等人又在2005年给出了一种用于计算KLR的快速对偶算法，该算法不需要将整个核矩阵带入迭代步骤里进行计算，每一次迭代只优化序列里的两个值，避免了对核矩阵进行求逆等操作，因此迭代的计算代价十分的小。

随着数据规模的不断扩大，核方法的应用越来越受限制。因为当引入核方法后，算法优化目标损失函数中就会引入一个跟样本成正相关的核矩阵，这个核矩阵的生成和求逆都需要非常大的时间开销。所以考虑到算法时间复杂度的问题，于是我们又在此基础上引入了Nystrom低秩近似方法。通过使用Nystron方法去近似求解我们需要的核矩阵，同时还能保证最后生成的核矩阵是低秩的，最后再结合我们前面提到的快速对偶算法进行优化求解，这样就不仅降低了算法的时间复杂度，还通过低秩去掉了数据中的冗余信息提高了分类精确度。

## 3.2 低秩近似核逻辑回归算法

针对第2章中逻辑回归算法中的损失函数式（2.6）我们再引入核方法中的表达定理式（2.5），于是最终就能得到KLR的损失函数

|  |  |
| --- | --- |
|  | (3.1) |

### 3.2.1 核逻辑回归的梯度下降法求解

对于上式（3.1）损失函数，需要求解的变量只有，于是对其求一阶导得到梯度：

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

所以，迭代公式则为： ，其中为步长，为惩罚项。根据迭代公式，我们最终就可以得到优化参数**。**

算法3.1描述了梯度下降法求解核逻辑回归的伪代码。

|  |
| --- |
| **Algorithm 3.1：Gradient descent method for KLR** |
| **Input**：D， , parameter: ，  **Output**：  **While not converged do**   1. Generating a kernel matrix **K** from data set D 2. Calculate the gradient update direction： 3. update ：   **Return** |

### 3.2.2 核逻辑回归的牛顿迭代法求解

对于损失函数式（3.1）的求解，也可以采用牛顿迭代法进行求解。基于式（3.2）的一阶导，再对该式求二阶导，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |

其中，

所以，

因此，其二阶导数则为

|  |  |
| --- | --- |
|  | (3.4) |

这里的***V***是一个对角矩阵，其元素是 。

于是，迭代公式则为：

|  |  |
| --- | --- |
|  | (3.5) |

根据迭代公式，我们最终就可以得到优化参数**。**

算法3.2描述了牛顿迭代法求解核逻辑回归的伪代码。

|  |
| --- |
| **Algorithm 3.2：Newton iteration method for KLR** |
| **Input**：D， , parameter:，  **Output**：  **While** not converged **do**   1. Generating a kernel matrix K from data set D   update ：  **Return** |

### 3.2.3 快速对偶算法

如果KLR中，样本的类标签y是1或-1，即,则有

|  |  |
| --- | --- |
|  | (3.6) |

将其合并为以下表达式：

|  |  |
| --- | --- |
|  | (3.7) |

同样，使用MLE作估计，得到似然函数：

|  |  |
| --- | --- |
|  | (3.8) |

对式（3.8）取负对数便能得到KLR的另一种形式损失函数：

|  |  |
| --- | --- |
|  | (3.9) |

同理，再根据表达定理：，并引入正则项

得到KLR的另一种损失函数形式：

|  |  |
| --- | --- |
|  | (3.10) |

这里的，C表示惩罚项。

对于上式的损失函数，其本身是一个凸二次规划问题，如果直接求解是比较繁琐的，于是可以使用拉格朗日乘子法得到其对偶问题，然后在对偶形式上进行求解。

首先，我们定义

|  |  |
| --- | --- |
|  | (3.11) |

于是原问题则为

|  |  |
| --- | --- |
|  | (3.12) |

接着引入广义拉格朗日函数（Generalized Lagrange function）:

|  |  |
| --- | --- |
|  | (3.13) |

此处的表示拉格朗日乘子，。令分别对的偏导为零：

|  |  |
| --- | --- |
|  | (3.14) |

于是便有，

|  |  |
| --- | --- |
|  | (3.15) |

接着再将式（3.15）带入到拉格朗日函数中，并令 ,因为是的函数，所以令：

|  |  |
| --- | --- |
|  | (3.16) |

根据的形式，于是可以对上式求导得到：

|  |  |
| --- | --- |
|  | (3.17) |

由于，因此 ，此处是单调、可微的，而的反函数的定义域是。通过检查二阶导数的非负性很容易验证如果是一个凸函数，那么也是一个凸函数，综合前面的式子，可以得到以下函数：

|  |  |
| --- | --- |
|  | (3.18) |

再利用Wolfe 对偶理论并使原问题最大化来得到其对偶形式：

|  |  |
| --- | --- |
|  | (3.19) |

带入式（3.16）和 , 并去掉负号得到：

|  |  |
| --- | --- |
|  | (3.20) |

考虑对偶问题的优化条件，为了确定阈值参数和算法的停止条件，对式（3.20）再次引入拉格朗日乘子则有：

|  |  |
| --- | --- |
|  | (3.21) |

这里的为惩罚项，是拉格朗日乘子。

同理，对式（3.21）求导则有:

|  |  |
| --- | --- |
|  | (3.22) |

为了方便描述，我们有以下定义：

|  |  |  |
| --- | --- | --- |
|  |  | (3.23) |

因此，对偶问题的最优条件可以改写为：

|  |  |
| --- | --- |
|  | (3.24) |

最后，我们再令：

|  |  |
| --- | --- |
| , 其中  , 其中 | (3.25) |

此处， 都是关于的函数，通常有

于是便可以得到该对偶问题的终止条件：。在对向量更新的时候只要还存在一对满足时，就可以认为还存在不满足最优解条件的，就会持续更新下去，直到达到终止条件。基于坐标下降的思路，当向量中的处满足并保持等式约束的时候，要让函数减小，于是有以下定义：

|  |  |
| --- | --- |
| ，  ， | (3.26) |

这里， ,可以使用牛顿迭代法来求解:

|  |  |
| --- | --- |
|  | (3.27) |

所以，我们可以得到参数更新的迭代公式为

|  |  |
| --- | --- |
|  | (3.28) |

最后，便有 ， 。对于终止条件，在实际操作中通常不会达到绝对的最优精确度。因此，可以使用近似的最优条件：, 这里的是精度参数。

### 3.2.4 快速对偶算法联合低秩近似方法

通过使用快速对偶优化算法对KLR算法进行优化求解，区别与常规的一阶梯度下降算法和二阶牛顿迭代算法，在算法优化求解的过程中，通过将其转化为对偶问题，从而能够避免核矩阵参与迭代运算，相比于常规的优化算法，能很大程度的降低算法的时间复杂度。但是，随着样本的增加，核矩阵的生成也会花费大量的时间，考虑到算法最终的时间效率，我们利用低秩近似方法去近似求解核矩阵，然后将其与快速对偶算法相结合去求解KLR。

首先，我们使用特征分解可以将核矩阵分解成如下形式：

|  |  |
| --- | --- |
|  | (3.29) |

其中，是一个正交矩阵，是对角线元素为降序排列的对角矩阵。如果，取矩阵中的前p列构建，且。于是，就可以用来近似核矩阵。引入是为了避免产生奇异矩阵，增加稳定性。而是很小的一个正数[26]。

如果能进行特征分解的话，上述的近似方法能够为核方法的计算节约很大的开销。但是矩阵特征分解的时间复杂度通常是，所以通常的做法是在计算核矩阵的时候，只需要先计算的前个特征值和特征向量，当时，时间复杂度就有明显的降低。此时再通过上面描述的Nystrom方法来近似得到所有样本点的特征方程，而的低秩近似矩阵就可以写成：

|  |  |
| --- | --- |
|  | (3.30) |

此处的分别表示用Nystrom方法得到的全部样本矩阵的特征值和特征向量。带入式（3.30）中，则有：

|  |  |
| --- | --- |
| ， | (3.31) |

这里的是矩阵的第个特征向量，而是核矩阵中的块矩阵。

假定对称半正定核矩阵 , 从中随机选取列组成矩阵，其形式如下：

|  |  |
| --- | --- |
|  | (3.32) |

其中。如果对进行SVD分解得到，这里是正交矩阵，而是块矩阵的奇异值按降序排列的对角矩阵。如果有，那么原核矩阵的低秩近似矩阵则为：

|  |  |
| --- | --- |
|  | (3.33) |

这里, 是矩阵的第i列。

为了减少KLR算法中核矩阵的计算时间，可以利用Nystrom方法将近似求得到的带入快速对偶算法中，从而节省了整体的计算开销。

算法3.3描述了快速对偶算法联合低秩近似方法求解核逻辑回归的伪代码。

|  |
| --- |
| **Algorithm 3.3：Fast dual algorithm combined with Nystrom** |
| **Input**：D， , parameter:，，Rank  **Output**：**,** b  **While not converged do**   1. Selecting m vectors from the Rank value to calculate the approximate kernel matrix 2. initialization，select 3. Update and 4. ,   **Return**  **,** b |

### 3.2.5 收敛性分析

对于梯度下降算法，每一次的迭代更新都是以负梯度作为搜索方法前进一个步长，因此最终一定是线性收敛的。当迭代次数趋于无穷时，代价函数收敛到最优解，收敛速率为。

牛顿迭代法则是采用Newton步径为搜索方向，而Newton步径也是处采用Hessian矩阵定义的二次范数。因此，一般情况下Newton方法收敛很快，在附近二次收敛，一旦进入二次收敛阶段，只需要再经过少量的迭代就可以产生具有很高精度的解。

在证明快速对偶算法收敛性之前，先给出以下说明：

首先让,因为是连续的，所以意味着，也是有边界的。另外，对于任意的序列,当时就会接近A的边界点，而，这里是变量的个数。由于快速对偶算法是基于坐标下降的算法，所以每一次的迭代都是十分明确的。

引理1：当的时候：

|  |  |
| --- | --- |
|  | (3.34) |

proof:

首先，将在附近进行有限二阶泰勒级数展开：

|  |  |
| --- | --- |
|  | (3.35) |

这里，介于t和之间，但是不等于t或，对求二阶导，则有：

|  |  |
| --- | --- |
|  | (3.36) |

又因为有 ，所以，我们就能得到这个边界：。

再根据式（3.35），令，于是有：

|  |  |
| --- | --- |
|  | (3.37) |

因此，引理1得证。

定理1：

1. 在中至少有一个极值点。
2. 中的里的每一个极值点都是快速对偶算法的解。

proof:根据引理1，由于，在中有极值点，且中的每一个极值点也都属于，又由于，所以的每一个极值点也都属于。

由于算法在每一次迭代的时候都在减小，并且是下边界，所以是一个收敛的序列。由引理1，可以立即得到收敛至0。

令 为收敛子序列，表示在中收敛了的极值点。对于任意的，让,,i的坐标在第r步迭代优化的时候选择。当时，由，得到：

|  |  |
| --- | --- |
|  | (3.38) |

由于是有限的，在无限的中，至少存在一对,其中，,为了便于书写，重新定义这个子序列：

|  |  |
| --- | --- |
|  | (3.39) |

和是关于的连续函数，以此可以得到：

|  |  |
| --- | --- |
|  | (3.40) |

这里，有：

|  |  |
| --- | --- |
|  | (3.41) |

当收敛至0的时候，有=0，，再由式（3.38）可得，。

所以，只要当的时候，就是SMO所求的解序列，而是收敛了的极值点。

## 3.3 实验与结果分析

### 3.3.1 数据集介绍

为了比较一般优化算法和低秩近似快速对偶算法在求解KLR上的计算时间花费和分类精确度的差异，我们选取了10个不同的来自UCI上的公共数据集，其中banana数据集的样本特征是非线性的。表1列出了各种数据集的详细信息：

表1 数据集信息

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集名 | 特征数 | 训练样本数 | 测试样本数 |
| ionosphere | 34 | 245 | 105 |
| Australian | 14 | 483 | 207 |
| diabetes | 8 | 537 | 230 |
| banknote | 4 | 960 | 411 |
| titanic | 3 | 1540 | 660 |
| cancer | 9 | 1939 | 831 |
| waveform | 40 | 2341 | 1003 |
| banana | 2 | 3710 | 1590 |
| mushroom | 22 | 3950 | 1693 |
| image | 18 | 3465 | 8085 |

表2给出了实验的参数设置。在以下的所有实验中，我们主要采用的是高斯核

,收敛条件参数。对于，我们将其初始值取为，在SMO算法中，必须要让,所以将训练集中class1和class2的数量统计为和,最终被初始化为和 。除此之外，也可以将全部初始化为0 。

表2 实验参数

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | SMO | NSMO-KLR | | SGD | NSGD-KLR | | | NI | NNI-KLR | | |
| 参数  Dataset |  |  |  |  |  |  |  |  |  |  |  |
| ionosphere | 2 | 2 | 15 | 3 | 2 | 0.2 | 25 | 0.05 | 15 | 0.05 | 25 |
| Australian | 2.8 | 3 | 55 | 5 | 8 | 2 | 55 | 0.2 | 3 | 0.5 | 100 |
| diabetes | 3.8 | 1.8 | 35 | 6 | 10 | 0.5 | 55 | 5 | 3 | 5 | 55 |
| banknote | 2 | 1.5 | 15 | 8 | 10 | 0.1 | 25 | 0.2 | 3 | 10 | 100 |
| titanic | 0.2 | 0.2 | 15 | 3 | 2.5 | 0.01 | 15 | 0.05 | 15 | 0.05 | 25 |
| cancer | 0.2 | 0.2 | 15 | 3 | 2.3 | 0.5 | 25 | 0.05 | 25 | 0.5 | 25 |
| waveform | 10 | 10 | 15 | 10 | 40 | 0.5 | 15 | 0.05 | 25 | 0.5 | 25 |
| banana | 0.2 | 0.2 | 80 | 1.9 | 1.9 | 0.5 | 55 | 0.5 | 2 | 0.5 | 55 |
| mushroom | 2 | 1.8 | 100 | 10 | 10 | 0.5 | 100 | 0.5 | 3 | 5 | 100 |
| image | 0.2 | 0.5 | 200 | 3 | 4.5 | 0.1 | 35 | 0.5 | 2 | 0.5 | 55 |

### 3.3.2 不同数据集实验结果对比

表3列举出了上述6种算法在不同数据集上达到同等收敛条件时的计算开销(in seconds)。当计算代价超过5000 seconds时，统一用“-”代替。

表3 计算开销

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method  Dataset | SMO | NSMO-KLR | SGD | NSGD-KLR | NI | NNI-KLR |
| ionosphere | 1.2 | 3.5 | 14.3 | 27.3 | 20.1 | 4.7 |
| Australian | 4.9 | 3.1 | 22.2 | 18.3 | 30.5 | 16.2 |
| diabetes | 5.9 | 3.4 | 18.1 | 18.3 | 18.1 | 6.4 |
| banknote | 16.3 | 5.9 | 41.7 | 25.9 | 231.9 | 103.1 |
| titanic | 61.3 | 30.8 | 76.9 | 45.9 | 2264.2 | 502.7 |
| cancer | 101 | 75.8 | 155.1 | 109 | 1875.9 | 896.4 |
| waveform | 113 | 19.4 | 220.3 | 106.8 | 610.7 | 263.5 |
| banana | 325 | 63.4 | 439.5 | 376 | 4907.2 | - |
| mushroom | 348.5 | 124.7 | 599.6 | 293 | 170.5 | 43.3 |
| image | 287.7 | 255.4 | 2167 | 1548.9 | - | - |

表4列举了8种算法在不同数据集上的分类精确度。对于不能用某方法进行分类或者计算时间花销超过5000 seconds的数据集，其精确度用“-”代替。

表4 分类精确度

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | SMO | NSMO-KLR | SGD | NSGD-KLR | NI | NNI-KLR | SVM | LR |
| ionosphere | 0.9622 | 0.9622 | 0.9339 | 0.9339 | 0.9339 | 0.9057 | 0.9433 | 0.8801 |
| Australian | 0.8599 | 0.8647 | 0.8599 | 0.8696 | 0.8406 | 0.8454 | 0.8599 | 0.8550 |
| diabetes | 0.7965 | 0.7922 | 0.7835 | 0.7835 | 0.7835 | 0.7878 | 0.6797 | 0.7835 |
| banknote | 0.9976 | 1.0 | 0.9976 | 0.9976 | 1.0 | 0.9951 | 1.0 | 0.9879 |
| titanic | 0.7927 | 0.7943 | 0.7912 | 0.8167 | 0.7837 | 0.7791 | 0.7836 | 0.7791 |
| cancer | 0.9747 | 0.9819 | 0.8351 | 0.7942 | 0.9278 | 0.8122 | 0.9747 | 0.7701 |
| waveform | 0.9143 | 0.9193 | 0.9123 | 0.9143 | 0.9173 | 0.9163 | 0.9213 | 0.9277 |
| banana | 0.9132 | 0.9138 | 0.8943 | 0.8949 | 0.9006 | 0.9006 | 0.9034 | - |
| mushroom | 0.9994 | 0.9982 | 0.9947 | 0.9929 | 0.9947 | 0.9548 | 1.0 | 0.9728 |
| image | 0.9856 | 0.9873 | 0.9431 | 0.8883 | - | - | 0.9612 | 0.8355 |

表5列出了快速对偶算法与SVM和LR的分类准确度的变化，以及NSMO-KLR和NNI-KLR算法与SMO和NI方法相比较的精度变化。其中，精度有所提高的部分已经用黑体标注出来。

表5精度提高

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | SMO | NSMO-KLR | NSGD-KLR | NNI-KLR |
| ionosphere | 0.0189 | 0 | 0 | -0.0282 |
| Australian | 0 | **0.0048** | **0.0097** | **0.0048** |
| diabetes | 0.013 | -0.0043 | 0 | **0.0043** |
| banknote | -0.0024 | **0.0024** | 0 | -0.0049 |
| titanic | 0.0091 | **0.0016** | **0.0255** | -0.0046 |
| cancer | 0 | **0.0072** | -0.0409 | -0.1156 |
| waveform | -0.0134 | **0.005** | **0.002** | -0.001 |
| banana | 0.0098 | **0.0006** | **0.0006** | 0 |
| mushroom | -0.0006 | -0.0012 | -0.0018 | -0.0399 |

### 3.3.3 实验结果分析

对于KLR的求解，采用快速对偶SMO算法在不同规模的数据集上表现都十分的优秀，通过联合核矩阵的近似求解方法，能够在此基础上发挥更好的效果，不仅在求解速度上有很大的提升，精确度也超越了梯度下降法、牛顿迭代法以及SVM和LR等常用算法。对于部分本身就是线性可分的数据集，加了核函数不一定能提高其分类的精确度，反而会增加其训练时间，但是对于例如banana这种非线性可分的数据集就是很必要的了。

在某些例如ionosphere数据集上，采用Nystrom方法近似求解核矩阵的全局开销反而会变大，这是因为在样本较少的情况下为了达到与原核矩阵求解时同样的精度误差，会导致迭代的次数增加，虽然单次迭代的时间花销有所下降，但是整体的时间花销反而增加了。由于近似核矩阵的秩是需要Rank参数去控制的，如果Rank值设置得不合理，也会导致迭代次数的增加。

而梯度下降和牛顿迭代这两种算法在迭代的过程中整个核矩阵都参与了运算，其中包括了对核矩阵进行求逆等操作，当使用近似核矩阵去代替的时候，计算的开销有明显的下降，但是精准度也会下降。例如cancer和image数据集，在使用近似核矩阵的时候，对近似核矩阵再次进行求逆和求Hessian矩阵会损失掉大部分的相关信息，所以对该两种方法采用核矩阵近似加速求解的时候是会以牺牲分类精确度为代价的。同时，由于近似核矩阵是原核矩阵的低秩近似，所以减少了冗余的无关分类信息，对于大多数公共数据集都提高分类的精确度。

## 3.4 本章小结

针对核逻辑回归算法的优化求解，我们对比了常规的诸如一阶梯度下降算法和二阶牛顿迭代优化算法，然后考虑到核方法在损失函数中引入了核矩阵，在优化求解的时候，核矩阵会参与到迭代运算中去，这样会导致在样本数量增加的情况下算法的时间开销非常的大。因此，我们采用了快速对偶算法对核逻辑回归算法进行优化求解，从而避免了核矩阵参与迭代运算。最后，再考虑到核矩阵的计算开销也非常的大，于是我们将Nystrom低秩近似方法与快速对偶算法相结合，通过Nystrom方法去低秩近似样本中的核矩阵，从而缩减了生成核矩阵的时间开销，另外由于生成的核矩阵是低秩的，只保留了样本中最主要的特征，所以也去掉了数据的冗余信息，从实验结果中也可以看出，基于低秩近似核矩阵的优化算法，不仅在算法时间复杂度上降低了，而且由于去掉了数据中的冗余信息，算法分类的准确度也有了一定程度的提升。

# 第4章 基于中心对齐多核学习的稀疏多元逻辑回归算法

上一章介绍了基于低秩近似和快速对偶算法求解的核逻辑回归算法，但是该算法只能应用于二分类问题。本章主要介绍多核稀疏多元逻辑回归算法，不仅将二算法从二分类扩充成了多分类，同时还引入了多核学习，解决了核函数单一的问题。

## 4.1 问题提出与研究思路

多元逻辑回归(Multinomial Logistic Regression, MLR) 算法是统计分析、机器学习和数据挖掘领域的一个经典多分类算法。它相对其他分类算法来说模型较简单、容易理解，也适用于大规模的分类问题，被广泛应用于诸如医学检测、地质测量、文本分类等领域。引入了正则项的多元逻辑回归称作稀疏多元逻辑回归(Sparse Multinomial Logistic Regression, SMLR),它通过将拉普拉斯先验引入多元逻辑回归中可以使其解具有稀疏性，这让它可以在进行分类的过程中嵌入特征选择，所以在高维数据和稀疏数据集的处理上都具有很大的优势。在多类别分类任务中，二分类算法采用one-vs-rest或者one-vs-one的策略进行多类别分类时会受到在样本不平衡的影响，而SMLR只需要训练1次即可，它继承了稀疏逻辑回归稀疏解的同时也较好地处理了多分类问题。因此，SMLR被广泛应用在图像中的多类物体识别、高光谱图像分类、生物信息学等领域。

为了解决非线性的特征数据分类的问题，SMLR经过核技巧核化扩充之后的分类器称为核稀疏多元逻辑回归(Kernel Sparse Multinomial Logistic Regression, KSMLR)。核方法通过将原数据的样本映射到1个或多个高维甚至无穷维的特征空间，使得样本在这个特征空间内线性可分。但是这种单核学习的核函数只有1个，其结构单一, 为了将数据映射到不同的高维空间中从而使得这些数据特征能够更好地表达，本文将多核学习与SMLR算法结合起来，通过融合多个核函数形成多核稀疏多元逻辑回归算法(Multiple Kernels Sparse Multinomial Logistic Regression Algorithm, MKSMLR)。目前，多核学习的经典方法有SimpleMKL和SPF-GMKL等，而基于中心对齐的学习核算法(Algorithms for Learning Kernels Based on Centered Alignment)是当前比较流行的多核学习方法。该方法通过中心对齐思想学习能得到一组核函数的权重系数，由此可以更灵活的组合出新的核函数。

针对非线性分类问题，除了本文使用到的核方法外，还可以采用核PCA，跟核方法类似，都是将数据特征从低纬度空间转化到高纬度空间从而达到线性可分的目的。一般这种操作都是在分类器进行学习前，先对数据进行的一种预处理方法。而本文则是将核方法嵌套到了算法的学习过程中，在每次优化损失函数的时候都会引入核技巧，这样能更好地对数据进行映射和充分地表达。除此之外，神经网络由于引入了非线性的激活函数，因此也能很好地解决非线性问题，但是训练一个神经网络需要大量的样本数据，所以对于小批量的数据集，训练效果往往差强人意。综上，对于小批量的甚至是稀疏的数据集，多核学习是目前为止较为有效可行的解决方法。而对于稀疏优化问题，本文采用快速迭代软阈值收缩算法(Fast Iterative Shrinkage-Thresholding Algorithm, FISTA)去优化求解。

## 4.2 核稀疏多元逻辑回归算法

根据前面第2章提到的稀疏多元逻辑回归算法，我们再引入核方法去解决非线性多分类的问题。根据表达定理, 此处的为在特征空间的投影，于是将特征投影到高维空间有

|  |  |
| --- | --- |
|  | (4.1) |

这里的是核矩阵的第行向量，其中，关于是非线性的，关于是线性的。和之间也是线性关系。核函数相当于使用预处理所有的输入，然后在高维或无穷维的特征空间中学习线性模型。

于是对于给定一个样本和标签，样本属于类别的概率可以表示为

|  |  |
| --- | --- |
|  | (4.2) |

也一定有

|  |  |
| --- | --- |
|  | (4.3) |

同理，将SMLR的损失函数中的所有内积计算都用核函数去替代，于是可以得到KSMLR的损失函数

|  |  |
| --- | --- |
|  | (4.4) |

通过核技巧，可以在保证有效收敛的条件下用凸优化技术来学习关于的非线性模型。因是固定不变的，需要求解的变量只有，且有。

### 4.2.1 稀疏优化算法

一般而言，解的稀疏性通常通过引入范数来获得，但求解问题是NP难的组合优化问题。因此，通常的做法是对问题进行松弛优化，即求解凸优化问题。本文采用迭代软阈值收缩算法(Iterative Shrinkage Thresholding Algorithm, ISTA)，也称作近端梯度下降法(Proximal Gradient Method, PGM)来求解稀疏优化问题。对于SMLR，优化目标式(2.7)可以写成：

|  |  |
| --- | --- |
|  | (4.5) |

其中。直接对式(4.5)进行求解并不容易，因此可以先将其转换为比较容易求解的形式。本文将在处做二阶泰勒展开，并取海森矩阵，其中的单位矩阵，变量为步长。则有

|  |  |
| --- | --- |
|  | (4.6) |

那么SMLR的目标函数就可以重新写成

|  |  |
| --- | --- |
|  | (4.6) |

于是，最小化问题变为

|  |  |
| --- | --- |
|  | (4.7) |

通过简单的代数变换，最小化式(4.7)可以被重写为

|  |  |
| --- | --- |
|  | (4.8) |

其中，式(4.8)忽略了与最小化无关的常量。对于最小化问题式(4.8)，可以快速的进行求解。

令：

|  |  |
| --- | --- |
|  | (4.9) |

则式(4.8)的解可以表示为。其中，软阈值操作被定义为

|  |  |
| --- | --- |
|  | (4.10) |

这里下标表示逐元素执行软阈值操作。

一般而言，步长通常可以取固定值，其中为利普希茨常数(Lipschitz Constant)。但使用固定步长的一个缺点是利普希茨常数不总是已知的，在大规模问题下其值也不易计算。因此，Beck等人给出了一个简单的线性搜索策略，步长的取值可以通过线性搜索(line search)的方式确定。

### 4.2.1 核稀疏多元逻辑回归优化算法

针对KSMLR问题，采用回溯ISTA算法进行优化求解，其迭代步骤如算法4.1所示。

|  |
| --- |
| 算法4.1：KSMLR问题的回溯ISTA算法 |
| 输入： |
| 初始化步长：， ， |
| 初始化参数：，  初始化核函数参数：， |
| 最大迭代次数： = 500， |
| 回溯参数： |
| 输出： |
| 算法最终的参数： |
| 迭代步骤：  1： 由样本计算得到核矩阵； |
| 2: 初始化计数器 ； |
| 3: 初始化参数； |
| 4: ； |
| 5: ； |
| 6: 当满足 或迭代到指定次数时算法终止，执行步骤7。否则，令，并返回到步骤4； |
| 7: 返回更新完成的算法参数 。 |

## 4.3 基于中心对齐的多核学习算法

### 4.3.1 中心对齐定义

如果将样本映射到高维的特征空间上，表示为，那么中心核函数可以定义为

|  |  |
| --- | --- |
|  | (4.11) |
|  |

从上式定义可以看出，核矩阵不依赖于特征映射规则。

根据中心核函数的定义，可以给出类似的中心核矩阵的定义，假定有样本，且在高维空间上的特征向量为,通过从中减去经验期望来对齐核矩阵，于是可以写作，这里。于是用来替代原来的核矩阵，使得样本是居中的。定义所有的，于是有

|  |  |
| --- | --- |
|  | (4.12) |

接着，令,且，于是可以得到，这里是一个半正定的矩阵。

另外，跟核函数一样，让表示内积操作，表示范数，于是有

并且有 。

定义一个所有元素都为1的向量，，和一个单位矩阵，于是对于任意一个核矩阵,这个中心核矩阵都可以写成

|  |  |
| --- | --- |
|  | (4.13) |

并且对于任意的两个核矩阵和，都有

|  |  |
| --- | --- |
|  | (4.14) |

### 4.3.2 内核的线性组合

我们通过中心对齐方法来学习内核的最佳线性组合，假定学习得到了个不同的核矩阵, 于是最终希望得到的多核线性组合为：

|  |  |
| --- | --- |
|  | (4.15) |

这里是中心核矩阵，其中有

|  |  |
| --- | --- |
|  | (4.16) |

通过以最大化居中对齐核矩阵来优化这个，于是优化目标为

|  |  |
| --- | --- |
|  | (4.17) |

用Align方法求解得到,这里是基于目标向量的理想内核，是矩阵的内积操作。这里设置的每一个都是独立的，, 而是用户定义的超参数，并且有且。

## 4.4 多核稀疏多元逻辑回归算法

本文通过使用样本的全部或者部分特征并以不同的核函数或同种类型核函数的不同参数来构建个不同的核矩阵,然后利用中心对齐思想和Align方法去求得这个核函数基于中心核函数的最优组合系数,其中。于是最后可以得到一个基于多核学习的核矩阵，其形式为：，这里的是中心核矩阵，并有且。

于是，对于样本和标签，样本属于类别的概率可以表示为

|  |  |
| --- | --- |
|  | (4.18) |

自然地可以得到基于多核学习的MKSMLR损失函数

|  |  |
| --- | --- |
|  | (4.19) |

由于引入了多个核矩阵，线性搜索迭代速度会非常的缓慢，为了解决MKSMLR求解速度缓慢的问题，本文采用快速迭代软阈值收缩算法(Fast Iterative Shrinkage Thresholding Algorithm, FISTA)对其进行优化求解。FISTA通常也被称为快速近端梯度法(Fast Proximal Gradient Method, FPGM)，它使用了Nesterov加速策略对原始ISTA算法进行加速，能够将ISTA算法在最坏情况的收敛率由优化为，其中为迭代次数。ISTA算法使用来近似估计海森矩阵，而FISTA算法则利用了梯度的最小利普希茨常数来近似估计海森矩阵。FISTA与ISTA的另一个区别在于，FISTA在最小化式(5)时并不是只使用了，而是使用了前两次参数的线性组合。基于同样的原因，FISTA也给出了带有回溯的版本。于是，我们便可以应用基于广泛的快速一阶训练方法FISTA来快速优化求解具有多个核矩阵的MKSMLR，其迭代步骤如算法4.2所示。

|  |
| --- |
| 算法4.2：MKSMLR问题的回溯FISTA算法 |
| 输入： |
| 初始化步长：， ， |
| 初始化参数：，  初始化核函数参数：， |
| 最大迭代次数： = 500， |
| 回溯参数： |
| 输出： |
| 算法最终的参数： |
| 迭代步骤：  1: 由样本计算得到个不同的核矩阵；  2: 用Align方法计算得到多核学习参数并生成新的核矩阵； |
| 3: 初始化计数器 ； |
| 4: 初始化参数,,；  5: ；  6: ； |
| 7: ； |
| 8: ；  9: 当满足 或迭代到指定次数时算法终止，执行步骤10。否则，令，并返回到步骤5； |
| 10: 返回更新完成的算法参数 。 |

## 4.5 实验与结果分析

本节将通过使用不同领域和不同规模的数据集来比较SMLR算法和KSMLR算法以及MKSMLR算法在分类上的效果差异。

### 4.5.1 实验设置

本文所有的实验都采用同一台服务器，中央处理器为12核、主频为2.0GHZ的Intel(R) Xeon(R) E5-2620，并且具有64GB的随机存取存储器，编程实验环境为Python 2.7。为了有效地比较不同优化算法的性能，实验选取了多个领域的不同大小的数据集，包括非线性特征数据集Banana; 多类物体识别数据集COIL20；人脸识别数据集ORL和GT-32；小规模的手写体数字识别数据集MNIST-S; 肺部基因数据集Lung;高光谱图像分类数据集Indian-pines以及图像分割数据集Segment。

在以下分类实验中，引入了其他多分类算法与本文的算法进行对比，其中包括：SVM、用于稀疏的线性可分数据集上的稀疏逻辑回归(Sparse Logistic Regression, SLR)算法、采用L2正则的权重衰减多元逻辑回归(Weight-Decay Multinomial Logistic Regression ,WDMLR)算法、采用ISTA优化算法求解的稀疏多元逻辑回归(Sparse Multinomial Logistic Regression-ISTA, SML-ISTA)算法、采用FISTA优化算法求解的稀疏多元逻辑回归(Sparse Multinomial Logistic Regression-FISTA, SML-FISTA)算法和基于单核学习的核稀疏多元逻辑回归(Kernel Sparse Multinomial Logistic Regression, KSMLR)算法。

核函数主要使用径向基函数(Radial Basis Function, RBF)， ，收敛条件参数。对于优化参数，将其初始值取为。另外，超参数*t*和表示线性搜索参数，表示正则惩罚项，为高斯核函数中的参数。其他算法如SVM，SLR和WDMLR的参数均为默认超参数。

### 4.5.2 不同数据集上的实验对比

实验结果中的分类准确率如表6所示，算法运行时间如表7所示。

表6 分类准确率

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | SVM | SLR | WDMLR | SML-ISTA | SML-FISTA | KSMLR | MKSMLR |
| Banana | 0.9069 | - | - | - | - | 0.9069 | **0.9107** |
| COIL20 | 0.8032 | 0.9676 | 0.9832 | 0.9895 | 0.9958 | 0.9977 | **1** |
| ORL | 0.9507 | 0.9420 | **0.9545** | 0.9242 | **0.9545** | 0.9000 | 0.9167 |
| GT-32 | - | - | 0.7823 | 0.7580 | 0.7621 | **0.8044** | **0.8044** |
| MNIST-S | 0.9113 | 0.9001 | 0.9109 | 0.9036 | 0.9048 | 0.9360 | **0.9400** |
| Lung | 0.7705 | 0.9344 | 0.9104 | 0.9104 | 0.9254 | 0.9180 | **0.9344** |
| Indian-pines | 0.7980 | 0.8182 | 0.7599 | 0.8120 | 0.8120 | 0.8218 | **0.8237** |
| Segment | 0.5989 | 0.9235 | 0.8268 | 0.8925 | 0.9253 | 0.9538 | **0.9567** |

表7 算法运行时间(s)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | SML-ISTA | SML-FISTA | KSMLR | MKSMLR |
| Banana | - | - | 0.78 | **1.19** |
| COIL20 | 1.71 | **0.39** | 7.61 | 13.46 |
| ORL | 142.05 | 7.5 | 10.43 | **2.73** |
| GT-32 | 88.19 | **2.03** | 37.94 | 10.77 |
| MNIST-S | **0.12** | 0.14 | 0.14 | 22.98 |
| Lung | 42.71 | **1.4** | 2.12 | 3.08 |
| Indian-pines | 427.62 | **18.58** | 68.31 | 909.1 |
| Segment | 21.33 | 20.71 | **13.68** | 33.35 |

表中的”- ”符号表示未能正确分类或分类效果接近于随机选择。

### 4.5.3 实验结果分析

从表6的实验结果中我们可以得出以下结论，对于非线性数据集Banana，常规的分类算法如SLR和WDMLR都无法将其正确分类，只有通过核技巧扩充后的算法诸如带高斯核的SVM、KSMLR和MKSMLR算法才能进行正常分类。另外，对于大多数的线性数据集，使用核技巧以后都能提高其分类器的精度，而对于数据集ORL和肺部基因数据集Lung，单核学习就表现得不如其它常规算法，但是在剩余的数据集上都能一定程度地提高分类效果。原因是数据集ORL和Lung都是特征数远大于样本数的数据集，因此对于这类数据集并不适用于核方法。采用多核学习的MKSMLR算法由于在核函数的选择上更加灵活，所以几乎在所有数据集上的表现都优于其他常规算法以及基于单核学习的KSMLR算法。对于某些具有稀疏特性的数据集诸如人脸识别数据集ORL和GT-32以及小规模的手写体数字识别数据集MNIST-S，采用稀疏优化算法ISTA求解相比如其他优化算法能得到更好的分类效果。

在核方法中，通常会引入一个稠密的核矩阵，其存储和计算代价都非常的高，存储稠密矩阵需要的空间，而计算这样的矩阵则需要的代价，这里*m*和*n*分别代表了样本的个数和维度。稀疏优化算法ISTA在最坏的情况下的收敛率为，而FISTA优化算法采用了Nesterov加速策略并利用了梯度的最小利普希茨常数来近似估计海森矩阵，因此FISTA在最坏情况下的收敛率则为，这里*T*为迭代次数。对于多核学习，在单核学习的基础上，由于引入了多个核矩阵，于是为了在保证能得到稀疏解的同时减少算法的时间开销，所以对多核学习算法KSMLR采用的是FISTA优化算法。从实验结果表6和表7都能看出，FISTA优化求解算法运行效率都要高过于ISTA优化求解算法。而对于多核学习，由于需要求解多个核矩阵，所以相比于单核学习需要花费更多的运行时间，但是由于多核学习能学习到更多有用的特征，可能导致收敛得更快，因此迭代次数要少很多，最终在某些如ORL，GT-32等数据集上，总的运行时间相比于单核学习反而更少了。

## 4.6 本章小结

本章节在前面章节的基础上提出了一种基于中心对齐多核学习的稀疏多元逻辑回归算法。通过核技巧解决了稀疏多元逻辑回归不能用于非线性数据分类的问题。然后在核函数的选择上，利用了中心对齐多核学习算法去灵活的选取核函数的权重系数，用不同核函数的线性组合去生成新的核函数，从而将数据映射到不同维度的高维空间中。在优化算法中，考虑到多核学习需要计算多个核矩阵，存在一定的时间开销，于是采用FISTA优化算法进行快速求解。最后在稀疏的人脸识别数据集以及其他领域的公共数据集上，本文提出的MKSMLR算法在分类准确率指标上都优于目前常规的分类算法。

# 第5章 遥感图像分类系统的设计与实现

本文第4章的基于中心对齐多核学习的稀疏多元逻辑回归算法一方面能解决非线性数据分类问题，另一方面由于利用了多核学习，能针对不同类型的特征使用不同的核函数让数据更加充分的表达，同时还引入了稀疏正则项，适用于稀疏类型的数据。

随着卫星遥感图像和航空遥感图片分辨率的不断提高，人们可以从遥感图像中获得更多的有用的数据和信息。遥感技术应用的核心问题是根据地物辐射电磁辐射强弱在遥感图像上表现的特征，判读识别地面物体的类属及其分布特征。遥感图像特征取决于遥感探测通道、地物光谱特征、大气传播特征及传感器的响应特征等因素。 只要了解这些因素对遥感图像特征的影响，则可按图像特征判读地面物体的属性及其分布范围，实现遥感图像的分类识别。目前图像识别和分类方法通常都是采用神经网络。但是在实际应用中,遥感图像的数据的采集和标定是较困难的,获取的已标定的训练样本集较小, 卷积神经网络需要动辄百万级的训练样本，小样本训练效果往往不能满足需求，如何从小量的数据中生成良好的神经网络或者对小样本异构特征图像进行识别和分类将是未来的研究方向[32]。此外迁移学习也可以解决遥感数据样本少的问题。而多核学习则可以将不同类型的异构特征进行不同的映射处理，充分利用了遥感图像的光谱特征和空间结构特征，能在有效的时间内得到可观的分类准确率。对于遥感图像的分类，主要有基于像素点的分类方法和面向对象的分类方法，目前有Dr. J. Li提出的基于多核学习的高光谱图像分类框架MLR-GCK和基于超像素的多核高光谱图像分类算法都取得了很好的分类效果。

通过多核学习，我们可以整合这两种不同类型的特征，从而实现更好的分类效果。本章将设计一个遥感图像分类系统，并用前面提到的算法进行仿真验证。

## 5.1 遥感图像分类的一般过程

遥感图像分类是遥感图像信息提取的一种方法。其主要是根据同类地物在相同的光照、地形等条件下具有相同或相似的光谱特征和空间信息特征。我们则主要通过不同类地物之间的特征差异性将遥感图像中的所有像素点进行类别判定，然后再用不同的颜色标记出来。

遥感图像是遥感卫星拍摄的图像，一般都含有成百上千个不同的波段，而遥感图像分类是针对遥感图像中的所有的像素点，每一个像素点如果单独抽取出来都是一条连续的光谱曲线。我们将每个像素点上的不同波段的灰度值构成一个光谱特征，于是便可以利用这些特征对遥感图像中的地物进行分类识别。

## 5.2 遥感图像分类中存在的问题与解决方法

由于遥感图像是通过遥感卫星拍摄生成的，所以通常会受到不同的拍摄角度，光照和地形变化等环境的干扰给图像分类造成难度。其分类方法主要分为无监督和有监督两个大类。无监督主要是基于相似性度量的聚类算法，如谱聚类算法、K-means算法和均值漂移算法等。而有监督的算法则主要包括SVM、多元逻辑回归算法和基于神经网络的分类算法等。

在实际应用中，我们获得的遥感图像中的像素点通常是没有先验信息的，这样的数据就非常的适合无监督的分类算法。但是，由于非监督的算法在分类的精确度上难以达到我们想要的精确度，所以我们通常会利用机器或者人为的给遥感图像打上部分标签以便于适用于有监督的分类方法。对于基于深度学习的分类算法，相比于有监督的常规机器学习算法，能够在很大程度上提高图像的分类准确率，但是由于遥感图像中带有先验信息的样本点很少，而深度学习又需要大量的训练样本才足以支撑起最终的算法模型，因此在实际应用中，遥感图像分类并不适合于深度学习框架。于是我们折中选择有监督的分类算法对遥感图像中的地物进行分类。

## 5.3 仿真系统的设计与搭建

### 5.3.1 软硬件环境要求

本文中的实验均使用Python语言编写，其系统配置如表5.1所示。

表5.1 系统配置表

|  |  |  |
| --- | --- | --- |
| 硬件 | 计算机 | 硬盘：256GB及以上  内存：8GB及以上  处理器：2.0GHz及以上 |
| 软件 | 操作系统 | 支持Windows7，Windows8，Windows8.1，Windows10 |
| 开发工具 | Python 3.x、Pycharm |

### 5.3.2 系统功能需求

根据本章节遥感图像分类的一般过程，遥感图像分类原型系统需求功能模块主要由以下部分组成：

1. 遥感图像选择功能。图像选择功能需要完成选择待分类的遥感图像并展示其中的一个波段成像，同时将遥感图像大小进行修正然后再匹配到对应的遥感数据集。

2. 应用逻辑回归算法进行遥感图像分类功能。通过选择不同类型的逻辑回归算法对遥感图像进行分类计算，然后将图像中的地物分类结果和分类准确率展示出来。

3. 遥感图像地物类别展示功能。通过遥感图像选择功能选择遥感图像后将其对应的地物标签展示出来，方便与算法分类结果进行对比。

4. 遥感图像分类结果展示功能。展示算法分类过后的遥感图像地物类别颜色标签。

5. 遥感图像分类系统参数调节功能。在算法进行分类的时候可以通过该模块进行算法的参数调节，参数会影响算法最终的分类结果，从而对地物的分类识别造成影响。

### 5.3.3 遥感图像地物表示与特征提取

本章节主要用到了2个遥感图像分类数据集，我们通过将遥感图像中的每一个像素点作为一个分类的样本，每一个像素点的不同波段就作为这个样本点的特征，而这个像素点所表示的地物则作为其类别标签，以此来进行模型的训练和预测分类。

1. Indian Pines 数据集

该数据集是最早的用于高光谱图像分类的测试数据，由机载可视红外成像光谱仪（AVIRIS）于1992 年对美国印第安纳州一块印度松树进行成像，然后截取尺寸为的大小进行标注作为高光谱图像分类测试用途。该数据总共有21025个像素，但是其中只有10249个像素是地物像素，其余10776个像素均为背景像素，在实际分类中，这些像素是需要剔除的，由于截取的这块区域均是庄稼，总共有16类，因此不同的地物具有较为相似的光谱曲线，而且这16类中，样本的分布极不均匀。

1. Pavia University 数据集

该数据集是由德国的机载反射光学光谱成像仪（Reflective Optics Spectrographic Imaging System，ROSIS-03）在2003年对意大利的帕维亚城所成的像的一部分高光谱数据。该光谱成像仪对0.43-0.86波长范围内的115个波段连续成像，所成图像的空间分辨率为1.3。其中12个波段由于受噪声影响被剔除，因此一般使用的是剩下103个光谱波段所成的图像。该数据的尺寸为 ，因此共包含2207400个像素，但是其中包含大量的背景像素，包含地物的像素总共只有42776个，这些像素中共包含9类地物，包括树、沥青道路（Asphalt）、砖块（Bricks）、牧场（Meadows）等。

### 5.3.4 分类器训练与预测



图5.1 遥感图像分类流程图

## 5.4 系统展示

### 5.4.1 遥感图像分类系统的界面展示

遥感图像分类系统的设计目标是对同一类型的遥感卫星图像进行正确的地物分类，并且用不同颜色将不同的地物标识出来。系统界面展示部分主要是根据系统功能需求来展示。

1. 仿真系统初始界面

图5.2是遥感图像分类系统的初始界面示意图。



图5.2系统初始界面

1. 遥感图像选择模块

通过选择“上一张”和“下一张”切换遥感图像数据集，同时也可以通过“路径选择”来加载指定路径下的数据集。图5.3是遥感图像数据集中其中某一波段的预览图像。

|  |  |
| --- | --- |
|  |  |

图5.3 选择遥感图像

1. 算法模块

图5.4是算法分类运行控件，同时下面还可以根据需要选择对应的合理的算法。

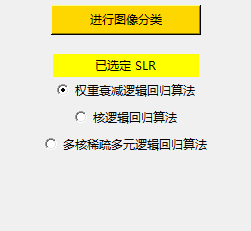


图5.4 算法运行控件

1. 参数调节模块

图5.5是算法参数调节控件，通过拉动滑条修改算法参数。

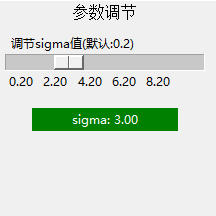


图5.5 算法参数控件

1. 分类结果和准确率模块

图5.6是核逻辑回归算法对Indian Pines数据集进行分类的地物展示图像和分类准确率。

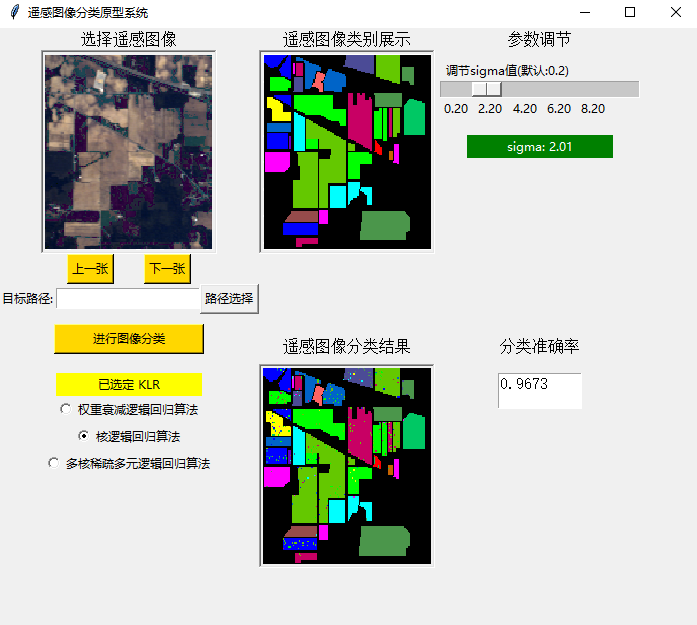


图5.6 算法分类结果

### 遥感图像分类结果及地物类别展示

通过将核逻辑回归算法应用到遥感图像分类中，我们对图像中的地物进行了分类，并用不同的颜色标注区分。图5.7和图5.8分别展示了Indian Pines数据集中16个地物和Pavia University数据集中9个地物的分类结果及其分类的精确度，其中最好的分类结果用粗体黑框标记出。



图5.7 Indian Pines地物分类结果



图5.8 Pavia University地物分类结果

## 5.5 本章小结

本章主要介绍了遥感图像分类系统的设计和实现。首先，本章介绍了遥感图像分类的一般过程，然后将前两章的算法应用到了遥感图像分类中去。接着，本章分析了系统需求，实现了6个基本功能需求并通过图片直观地展现了出来。本章通过遥感图像分类的例子，展现了核逻辑回归算法在实际工程中的应用。

# 第6章 总结与展望

## 6.1 研究工作总结

本文针对逻辑回归算法不能用于非线性数据分类的问题，通过核方法将其扩充成了可以用于非线性数据分类的核逻辑回归算法，然后考虑到核矩阵求解缓慢的问题，又结合了低秩近似方法和快速对偶算法进行优化求解。在解决了算法开销大的同时，还通过低秩降低了数据中的冗余信息，提高了算法分类的准确率。另外，考虑到逻辑回归只能应用于二分类的问题，我们又将核方法与稀疏多元逻辑回归相结合，然后再引入多核学习，提出了多核稀疏多元逻辑回归算法，解决了核函数单一和多类别非线性分类的问题。最后通过实验表明了本文提出的算法相比于其他传统分类算法，在时间开销和分类准确率上都有一定的提升。另外考虑到本文算法的实际场景，我们还讲其应用到了遥感图像分类任务上，提高了异构特征的遥感图像地物类别的识别率。

## 6.2 未来研究展望

本文第4章提出的多核稀疏多元逻辑回归算法，虽然通过多核学习解决了核函数单一的问题，但是该算法是通过将不同类型的核函数进行线性组合得到的新的核矩阵，虽然这样能让数据特征得到更加充分的表达进而提高算法的分类效果，但是算法的时间开销也会增大，在特定的大规模场景中适应性不高。针对上述问题，我们未来的研究工作将从以下几个方面展开：

1. 将算法改成并行化算法，使其适用于大规模的数据集，在时间开销和分类效果上进行合理有效的平衡。
2. 改进多核学习算法，减少算法中的超参数，并使其具有更高的泛化能力。

# 参考文献

# 致谢

时间如水，总是无言。在重庆邮电大学三年的求学生涯即将画上句号，回顾在校园里度过的每个日日夜夜，感受颇多。回想我的硕士研究生的学业时光，收获颇丰，取得了很大的进步，同时也发现了自己许多不足之处。我取得的所有成绩都离不开师长们的悉心指导和传道受业解惑。对于你们的谆谆教诲，我会永远铭记在心，并且对我今后的学业和事业产生巨大的推动作用。

路漫漫其修远兮， 日月窗间过马，上下求索略有所得，承蒙诸公教诲，生之幸矣。

首先，我要

# 攻读硕士学位期间从事的科研工作及取得的成果

**参与科研项目：**

1. IVF临床循证医学数据分析引擎 (E2019-03), 一般横向项目, 2019.01-2019.06, 已结题.
2. 面向大数据的鲁棒性监督学习并行化框架研究与应用 (cx2018120), 2018年重庆市留学归国人员创新创业项目支持人选, 2018.06-2020.12, 在研.

**完成论文及专利：**

1. Lei Dajiang, **Tang Jianyang**, Li Zhixing, Wu Yu. Using low-rank approximations to speed up kernel logistic regression algorithm[J]. IEEE Access, 2019, 7: 84242-84252. DOI:10.1109/ACCESS.2019.2924542, Accepted.
2. 雷大江，**唐建烊**，李智星，吴渝，基于中心对齐多核学习的稀疏多元逻辑回归算法，电子与信息学报，DOI：10.11999/JEIT190426，接受

**获奖：**

1. **唐建烊**, 硕士研究生学业奖学金(三等), 重庆邮电大学, 2018.09.
2. **唐建烊**, 硕士研究生国家奖学金, 重庆邮电大学, 2019.10.