**软化纠错输出编码多分类算法 郭宏洲 指导教师 刘昆宏 厦门大学**

**2029320381452802095400**

**本 科 毕 业 论 文 （设 计）**

**（主修专业）**

**软化纠错输出编码多分类算法**

**Soft Coding ECOC Multiclass Classification Algorithm**

姓 名： 郭宏洲

学 号： 24320142202424

学　 院： 软件学院

专 业： 软件工程

年 级： 2014级

指导教师： 刘昆宏 副教授

**二〇一八年五月**

**厦门大学本科学位论文诚信承诺书**

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（ ）课题（组）的研究成果，获得（ ）课题（组）经费或实验室的资助，在（ ）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年 月 日

致 谢

值此论文完成之际，我谨向所有关心和支持我的人们致以诚挚的谢意！

首先，我要衷心的感谢我的导师刘昆宏副教授。在完成本文内容的过程中，刘老师给予了我许多的帮助和指导，使我获益匪浅。在我遇到问题时，刘老师总是耐心的和我讨论，帮助我分析问题、解决问题，并给予我鼓励。刘老师用他渊博的学识，多次为我点明思路，帮助我顺利完成了本文的研究。在此，谨向刘老师致以最诚挚的感谢！

其次，还要感谢所有指导过我的老师，如果没有他们在这四年里所教授的相关知识和研究的基本方法，就不会有我现在的成果，在此衷心的感谢厦门大学软件学院的所有老师们。

最后，我还要感谢所有关心我的亲人、朋友，感谢一直以来对我的关心和帮助。

摘 要

分类问题是通过分析已知数据的特征和数据所属类别的关系，从而对未知数据的所属的类别进行预测的问题。对包含两个类别以上的数据进行分类的问题称为多分类问题。分类器就是用于解决分类问题的工具。集成学习方法通过使用多个不同的分类器，从而提高分类的准确性。纠错输出编码(Error Correcting Output Codes，简称ECOC）是一种用于解决多分类问题的有效的集成学习方法。ECOC通过将多分类问题分解为多个二分类问题，对每个问题分别使用一个二分类器解决，从而实现使用二分类器解决多分类问题，并通过共同决策的方式增强分类器自身的纠错能力，提高分类结果的准确率。

现有的ECOC中的编码矩阵都是由整数（+1、-1、0）组成，本文提出一种对现有ECOC算法的改进方法——软编码ECOC(Soft Coding ECOC)，包括软值编码 ECOC(Soft Value Coding ECOC)和区间编码(Soft Interval Coding ECOC)。软编码ECOC通过将现有ECOC方法中的编码矩阵值由整数替换为实数值或区间，从而增强分类器对数据的识别能力，提高分类的准确率。

本文将软编码ECOC与现有硬编码ECOC方法在一些常用UCI数据集上的分类准确率进行对比，证明改进后的软编码ECOC在所选的UCI数据集上效果比硬编码ECOC有一定提升。

关键词：纠错输出编码；软编码；集成学习

# Abstract

Classification is to predict the class of unknown data basing on the knowledge of known data. The tools used to classify data are called classifiers. Classifiers analyze the features of known data and establish learning models to predict the class of unknown data. Classification problems that involve more than two classes are called multiclass classification problems. Ensemble learning uses several separate classifiers to perform classification together so as to improve the accuracy of prediction. Error correcting output codes(ECOC) is one of these ensemble learning methods. ECOC decomposes a multiclass problem into several binary classification problems. Then ECOC deals with the binary problems each with a separate binary classifier. ECOC enhances the ability of correcting errors by combining the results of the binary classifiers.

The coding matrices of current ECOC methods are consist of integers (+1, -1, 0). This paper proposes an approach called soft coding ECOC. Soft coding ECOC includes soft value coding ECOC and soft interval coding ECOC. Soft value coding ECOC replaces the integers in coding matrices with real numbers while soft interval coding ECOC replaces them with intervals. Soft coding ECOC has a better ability to distinguish between classes of data and has a better performance on classification.

This paper compares soft coding ECOC with current hard coding ECOC by testing their accuracy on some well-known UCI datasets. The results of experiments have proved that soft coding ECOC has a better performance than current hard coding ECOC.

**Key words:** ECOC; Soft Coding; Ensemble Learning

目 录

[第一章 绪论 1](#_Toc514264496)

[1.1 引言 1](#_Toc514264497)

[1.2 论文组织结构 3](#_Toc514264498)

[第二章 相关概念介绍 5](#_Toc514264499)

[2.1 多分类问题 5](#_Toc514264500)

[2.2 集成学习 5](#_Toc514264501)

[2.3 常用分类器 7](#_Toc514264502)

[2.3.1 支持向量机 7](#_Toc514264503)

[2.3.2 神经网络 8](#_Toc514264504)

[2.4 本章小结 9](#_Toc514264505)

[第三章 ECOC算法介绍 11](#_Toc514264506)

[3.1 算法概述 11](#_Toc514264507)

[3.2 算法描述 11](#_Toc514264508)

[3.3 算法特点 14](#_Toc514264509)

[3.4 软编码 15](#_Toc514264510)

[3.5 本章小结 15](#_Toc514264511)

[第四章 软编码ECOC 17](#_Toc514264512)

[4.1 算法概述 17](#_Toc514264513)

[4.2 算法实现 18](#_Toc514264514)

[4.2.1 软值编码ECOC 18](#_Toc514264515)

[4.2.2 区间编码ECOC 18](#_Toc514264516)

[4.3 算法分析 21](#_Toc514264517)

[4.4 本章小结 23](#_Toc514264518)

[第五章 实验设计与结果分析 25](#_Toc514264519)

[5.1 实验设计 25](#_Toc514264520)

[5.2 实验结果及分析 27](#_Toc514264521)

[5.3 本章小结 38](#_Toc514264522)

[第六章 总结与展望 39](#_Toc514264523)

[6.1 总结 39](#_Toc514264524)

[6.2 工作展望 39](#_Toc514264525)

[参考文献 41](#_Toc514264526)

# Contents

[**Chapter 1 Preface 1**](#_Toc514264552)

[**1.1** **Introduction 1**](#_Toc514264553)

[**1.2** **The Structure of This Dissertation 3**](#_Toc514264554)

[**Chapter 2 Related Concepts 5**](#_Toc514264555)

[**2.1** **Multiclass Classification 5**](#_Toc514264556)

[**2.2** **Ensemble Learning 5**](#_Toc514264557)

[**2.3** **Commom Clssifiers 7**](#_Toc514264558)

[2.3.1 Support Vector Machine 7](#_Toc514264559)

[2.3.2 Neural Network 8](#_Toc514264560)

[**2.4 Summary 9**](#_Toc514264561)

[**Chapter 3 An Introduction on ECOC 11**](#_Toc514264562)

[**3.1 Summary on ECOC 11**](#_Toc514264563)

[**3.2** **Introduction of ECOC 11**](#_Toc514264564)

[**3.3 Features of ECOC 14**](#_Toc514264565)

[**3.4** **Soft Coding 15**](#_Toc514264566)

[**3.5 Summary 15**](#_Toc514264567)

[**Chapter 4 Soft Coding ECOC 17**](#_Toc514264568)

[**4.1** **Summary on the Algorithm 17**](#_Toc514264569)

[**4.2 Realization of Soft Coding ECOC 18**](#_Toc514264570)

[4.2.1 Soft Value Coding ECOC 18](#_Toc514264571)

[4.2.2 Soft Interval Coding ECOC 18](#_Toc514264572)

[**4.3** **Analysis on Soft Coding ECOC 21**](#_Toc514264573)

[**4.4 Analysis on Soft Coding ECOC 23**](#_Toc514264574)

[**Chapter 5 Experiments and Results 25**](#_Toc514264575)

[**5.1 Experiments Designing 25**](#_Toc514264576)

[**5.2 Experiments Results And Analysys 27**](#_Toc514264577)

[**5.3** **Summary 38**](#_Toc514264578)

[**Chapter 6 Conclusions and Feature Work 39**](#_Toc514264579)

[**6.1 Conclusions 39**](#_Toc514264580)

[**6.2 Feature Work 39**](#_Toc514264581)

[**References 41**](#_Toc514264582)

# 第一章 绪论

## 1.1 引言

分类问题作为数据挖掘(Data Mining)的主要研究内容之一，其相关研究一直是数据挖掘研究领域的一个重点。分类问题实际上就是要在空间上找到一个最优的数据划分边界，将各类数据划分开，在预测时，根据数据所在的区域给出数据所属的类别。分类器便是实现这种划分、进行预测的工具。对于二分类问题，找到两类数据之间的划分边界相对比较容易，而对于多分类问题，因为多个类别的数据分布相较二分类问题具有更高的复杂性，单一分类器要区分所有类别的数据有较大的难度。依靠单个分类器进行预测有其局限性，因此集成学习作为一种有效的方法受到了关注。

集成学习通过结合多个弱分类器，从而形成一个强分类器。集成学习算法中，每个分类器负责决策的一部分，而最终输出的结果由多个弱分类器通过共同决策决定。集成学习方法相当于通过多次划分，可以更精细的划分出每一类数据的边界，从而增加多分类问题的预测准确率。纠错输出编码（Error Correcting Output Codes，简称ECOC）是集成学习方法中的一种有效的方法。ECOC作为一种多类分类问题处理框架，能有效地将多分类问题分解成一系列二分类问题，然后为每个二分类问题构造二分类器（或称作基分类器）[1]。ECOC通过编码矩阵（Coding Matrix）将多分类问题分解为多个二分类问题。通过将问题分解，ECOC中每个基分类器只需解决一个二分类问题，简化了建立模型的难度，并增加了基分类器输出结果的可靠性。ECOC编码矩阵中的每一行称为码字（codeword），记为**c**。每个码字分别对应数据中的一类。编码矩阵中的每一列分别对应ECOC的一个基分类器。预测时，将每个基分类器对样本的输出值收集后得到该样本的输出向量**o**。ECOC计算输出向量到每一类对应的码字的距离，将该样本分到距离最近的一类。

作为一种集成学习方法，目前已经有许多学者对ECOC进行了相关的研究。E.L. Allwein等人为ECOC引入了不确定的编码值[2]，为ECOC编码矩阵提供了一种三进制（+1，0，-1）的编码方式，一定程度上减轻了由于强制将原本特征不同的数据分为同一类而导致部分类别易被错分类的问题。O. Pujol等人提出了使用二叉树实现对数据进行层次划分，从而优化编码矩阵的方法DECOC[3]。DECOC以互信息为依据，根据数据的特征进行划分，从而构造基于数据的编码矩阵，使各类数据在基分类器上的输出差别最大化。S. Escalera等人在DECOC的基础上提出了ECOC-ONE[4][5]，基于验证集数据对编码矩阵进行扩展优化，为难划分的类别增加新的分类器以提升其泛化能力。S. Escalera等人还提出了一种新的Online ECOC算法，针对后续新增加的数据类别进行增量的学习[6]。J. Zhou等人提出一种根据混淆矩阵构建编码矩阵的CMSECOC[7]。N. Hatami提出了Thinned-ECOC[8]，根据编码矩阵中每一列对结果的贡献对编码矩阵进行优化，删减贡献较小的列，从而得到较小的编码矩阵。DECOC、ECOC-ONE等上述编码矩阵的构造方法均属于基于数据的编码方法。为了减少分类过程中对错误进行纠正所需的成本，P. Simeone等人提出了ECOC的多类拒绝框架[9][10][11]，拒绝不可靠的结果，通过尽可能多的将错误转变为拒绝，从而减少进行分类的代价，并提出了相应的衡量拒绝效果测度E-R曲线[12]。

目前ECOC已经有了很多应用，如在基因微阵列中的应用[13]、面部动作单元分类[14]等。ECOC算法自身纠错能力强，可靠性、鲁棒性高，泛化能力强，因此在各类多分类问题的应用中，ECOC算法均取得了较好的结果。未来ECOC算法可能被应用在更多领域。

ECOC算法预测结果的准确率很大程度上由编码矩阵的构造方法决定，但是现有的ECOC算法中的编码矩阵使用的都是整数。ECOC在编码时强制将原本数据中的若干类重新划分为正类或负类，而被归为同一类的数据类别中，可能有一些类别的数据特征和其它类别的数据差别很大，因此将这些数据作为同一类数据对基分类器进行训练可能会导致其中有些数据的输出结果与对应的编码值不匹配。另一方面，实际很多分类器输出的值是代表结果可靠性的实数，因此实际某些类输出结果可能距离对应的编码值，即基分类器在该类数据上输出的期望值很远，导致在计算距离、进行分类时该样本与目标类对应码字之间的距离增大，进而增加误分类的可能性。虽然引入不缺定编码值后的三进制编码ECOC方法以及基于数据的编码矩阵构造方法可以在一定程度上避免强制划分类别的问题，但并不能完全避免使用中基分类器的输出值偏离样本所属类别的对应编码值的问题。如果编码矩阵中所有的编码值不再是固定的三进制的整数值，而是实数值，并且某个类别对应的编码值恰好为基分类器对该类别数据结果的集中值，即采用“软编码”的方法，那么样本的输出向量与样本所属类别对应的码字的距离将极大减小，ECOC算法的效果将得到提升。

## 1.2 论文组织结构

本文共分成六章。论文首先对多分类问题、集成学习方法以及支持向量机、神经网络等常用的分类器等相关概念进行了介绍，引出了本文进行算法改进的对象——纠错输出编码（ECOC），介绍其算法的基本方法和算法特点。接着本文提出了对于ECOC算法的改进方法：软编码ECOC，介绍了其两种子方法：软值编码ECOC和区间编码ECOC的原理及实现，分析了软编码ECOC的算法特点，并通过实验数据说明了软编码ECOC对于现有ECOC算法效果的提升，结合数据分析了软编码ECOC算法相较于现有ECOC算法的优缺点以及软编码ECOC算法优于现有ECOC算法的原因。最后进行对本文内容进行总结，并对后续工作进行展望。

论文具体安排如下：

第一章 概述了多分类问题对于数据挖掘研究的重要性，简单介绍了集成学习方法和ECOC算法，罗列了一些目前ECOC算法相关的研究成果，指出目前的ECOC算法存在不足，介绍了本文的背景、目标和意义。

第二章 介绍了本文的一些相关概念，包括多分类问题、集成学习方法以支持向量机、神经网络等常用分类器。

第三章 介绍了ECOC算法的概念和实现，通过具体事例介绍了ECOC算法的编码、解码过程，分析了二进制和三进制编码矩阵的优缺点，并介绍了软编码的概念。

第四章 提出了软编码ECOC算法，详细介绍了软编码ECOC两种子方法：软值编码ECOC和区间编码ECOC，介绍了其概念和算法实现，并对软编码ECOC算法进行了具体分析，以软值编码ECOC算法为例分析了软编码ECOC算法对于现有ECOC算法中所产生的分类错误的纠正。

第五章 介绍了实验设计和实验结果，并对实验结果进行分析。通过对准确率、召回率等数据结果的具体分析，比较了硬编码ECOC与软编码ECOC的优劣，证明软编码ECOC算法的实际效果相较于硬编码ECOC有一定的提升。

第六章 总结了本文的主要工作，并提出下一步的算法改进研究方向。

# 第二章 相关概念介绍

## 2.1 多分类问题

分类问题的处理目标是将输入的数据正确的分到它实际所属的类别中。对于给定的一组形如（**xi**，*yi*）的数据集，其中 为第*i*个样本， 是第i个样本的类别标签，分类器的目标就是找到一个形如h(**xi**)=*yi*的学习模型，使得对于新的样本**x**，能够正确预测其类别y [15]。对于二分类问题，模型的建立比较简单。而对于多分类问题，很难直接使用二分类问题中所使用的模型一次性实现对所有类别的划分。对于多分类问题的解决，一些二分类器因其自身的特点，具有可以直接用于多分类问题的拓展性，如决策树（Decision Tree），一些算法通过将问题划分为多个二分类问题，使用多个二分类器解决，如支持向量机（Support Vector Machine, SVM）。本文所述的ECOC算法所属的集成学习方法也是通过结合多个分类器实现对多分类问题的处理。

## 2.2 集成学习

集成学习方法是一种训练多个分类器以解决同一个问题的机器学习方法。相较于通过训练数据建立一个模型的普通机器学习方法，集成学习方法建立一系列模型，并将它们结合使用[16]。多分类问题因其自身的复杂性，单个分类器的分类效果往往是有限的。而集成学习方法通过使用一组分类器（称为基分类器），结合多个分类器的分类结果，通过共同决策完成最终的分类预测。集成学习的关键是各个基分类器的集成策略，文献[17]中就介绍了一种有效的集成策略。集成学习方法相较于它所使用的基分类器可以具有更高的准确率。集成学习方法已经被证明为一种有效的分类方法[18]。

集成学习方法要达到较好的效果，要求每个基分类器之间具有差异性，并且基分类器的分类准确率不能过低。如果基分类器之间没有差异性，通过集成得到的结果和单个分类器的结果是一样的，集成学习的结果对于单个分类器的效果没有提升。集成学习方法要达到较好的效果，还要求每个基分类器的精度必须大于0.5。如果基分类器的准确率小于0.5，综合这些基分类器的结果将没有意义。当每个基分类器的准确率都大于0.5时，理论上，当基分类器的个数足够多时且均为相互独立时，集成学习方法的准确率将趋近于1。

构建具有差异性的基分类器，主要通过以下三种方法：

1、选取不同的数据特征对基分类器进行训练。在对基分类器进行训练时，分别选择训练集的不同特征子集，每个基分类器关注数据的一部分特征。通过训练数据特征的差异构造具有差异性的基分类器。

2、选取不同的数据样本对基分类器进行训练。在训练时从训练集中抽取不同的子集，每个基分类器分别使用数据的一个子集进行训练，通过数据集上的差异构造有差异性的基分类器。

3、选取不同种类的分类器或不同参数的同种分类器作为基分类器。为了使基分类器存在差异，从一开始就选择不同的分类器作为基分类器，或是选择使用不同参数的同种分类器，如具有不同拓扑结构的神经网络，来进行训练和预测，从而保证基分类器的差异性。

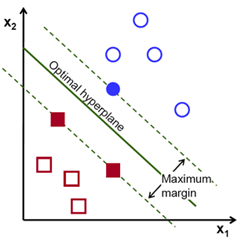
集成学习方法主要分为三种：Bagging、Boosting、Stacking。Bagging方法使用重采样的方法，对每个分类器使用不同的数据子集进行训练，决策时进行多数投票决策。Bagging方法的代表有随机森林（Random Forest）。当基分类器不稳定时，Bagging方法带来的性能提升比较明显。Boosting方法通过阶梯状的迭代训练，在训练一个分类器时都强化对上一个分类器的错误数据的纠正，其典型代表为AdaBoost。Stacking方法使用在训练集上使用不同的学习算法得到第一层相互独立的分类器，再由第二层分类器将其结合起来。

通过集成学习方法，多个弱分类器被组合形成一个强分类器，从而提高预测结果的准确率。在对各类多分类问题的应用中，集成学习方法通常具有较好的效果。

## 2.3 常用分类器

### 2.3.1 支持向量机

支持向量机（Support Vector Machine，简称SVM）在20世纪90年代由C. Cortes和V. Vapnik提出[19]。SVM是一种二分类器。SVM将输入的特征向量非线性的映射到高维特征空间后，在高维特征空间中构造一个具有最大间隔的决线性策平面，使得该超平面能在高维特征空间中将两类数据完全分离。在二维空间中，线性决策平面实际是一条直线。如图2-1所示。这条直线将两类样本分开，并且具有与两类数据的最大间隔。

****

**图2-1 SVM寻找具有最大间隔的决策边界[20]**

实际情况下，在数据原本的低维特征空间有时很难找到合适的直线或线性平面将数据分开。这时SVM将输入向量非线性的映射到一个高维特征空间，如图2-2所示。在高维特征空间中，两类样本可以被如图所示的平面分开，该平面即使所寻找的划分边界。由于维数很高时，向量的计算尤其是内积的计算复杂度很高。SVM通过使用核函数（Kernel Function）避免复杂的计算，减少计算量。常用的核函数有线性核函数、高斯核函数等。

****

**图2-2 SVM将输入向量映射到高维特征空间[20]**

使用SVM处理多分类问题时，则需要使用多个二分类SVM，并为其建立决策函数[21]，如投票决策，或One-against-the-Rest方式等。通过决策函数对多个二分类SVM的输出进行决策，从而实现多分类SVM。

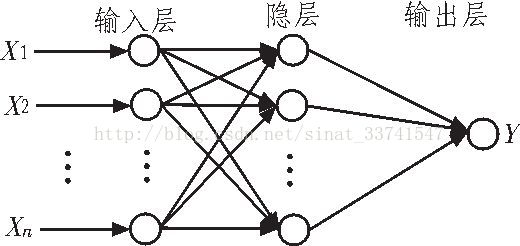
### 2.3.2 神经网络

神经网络（Neural Network）是对人脑神经元网络的结构抽象而形成的复杂网络结构。神经网络的基本单位是神经元，一个神经元的结构如图2-3 所示。一个神经元接受一个或多个输入，计算得到输出值**。**其中**w**为权向量，其中每一位的值*wj*分别对应该层第*j*个神经元的权值，b为偏置，f(**x**)称为激活函数。常用的激活函数有sigmoid函数、tanh函数等。图中的输入为**x1**、**x2**、**x3**，偏置为1。

****

**图2-3 神经元结构[22]**

神经网络由多个神经元结构连接而成。一个神经元的输出可以是下一个神经元的输入。图2-4是一个简单的神经网络结构。神经网络中包括输入层（Input Layer）、输出层（Output Layer）以及若干隐层（Hidden Layer）。每层神经元与下一层神经元连接，上一层神经元的输出也是下一层神经元的输入。输入层直接从输入向量获取输入，输出层的输出即为神经网络的输出结果。

**  
图2-4 一个简单的神经网络[23]**

BP（Back Propagation）神经网络[24]是目前应用最广泛的神经网络。BP网络的训练过程分为向前传播得到预测数据以及反向传播更新权重两步。首先数据特征向量从输入层输入，经过隐层、输出层的计算得到预测值，接着根据样本的真实标签计算预测结果与真实结果的误差，然后将该结果反向传播到各个隐层，计算出各层的误差，并根据误差更新权重。神经网络的训练过程实际上就是调整权向量**w**的过程[25]。

## 2.4 本章小结

本章主要介绍了本文相关的一些概念。本章首先简单介绍了多分类问题的概念，接着介绍了一类有效的处理多分类问题的方法——集成学习方法，介绍了集成学习方法的三类算法Bagging、Boosting、Stacking的概念，并介绍了要保证集成学习方法的泛化能力，需要基分类器具有独立性，并且准确率不能过低。最后本章还介绍了两种常用的分类器：支持向量机和神经网络。

# 第三章 ECOC算法介绍

## 3.1 算法概述

ECOC是一种用于解决多分类问题的集成学习方法。ECOC算法通过构造编码矩阵，将多分类问题分解为多个二分类问题，从而实现使用多个二分类器（基分类器）综合决策解决问题。

因为每个基分类器都只专注解决一个二分类问题，因此每个基分类器对其解决的二分类问题都有较高的可靠性。通过多个基分类器的合作，结合编码矩阵的纠错能力，ECOC分类器可以纠正一部分基分类器的错误，从而保证了ECOC分类器的可靠性和鲁棒性。另一方面，ECOC可以根据不同的数据场合选择不同的基分类器、编码矩阵、解码方法等，因而ECOC具有较强的适用性，可以适应不同的数据。

但是，ECOC的决策需要依靠基分类器，如果基分类器本身的准确率过低，综合其结果的ECOC也将不具有可靠性。同时ECOC算法还存在着对于类别过多的数据处理效率很低的问题。

目前ECOC算法的研究主要集中在对编码矩阵的优化上。研究的内容主要包括优化数据集重新划分的方式、使数据的编码值更合理，简化编码矩阵、减少算法所使用基分类器个数，以及在有新类别的数据是对编码矩阵进行重编码以识别新数据等。除了对于编码矩阵的优化（即对编码过程的研究），还有针对解码过程的研究，即如何根据编码矩阵更准确的计算样本属于哪一类别。

## 3.2 算法描述

ECOC算法分为编码和解码两个阶段。在编码阶段，算法主要进行编码矩阵的生成和对基分类器的训练。对于包含有*n*个样本、*N*类数据的数据集，ECOC算法根据所使用的编码矩阵构造方法，生成一个*N×M*的编码矩阵**T**以及对应的*M*个基分类器*f1*, *f2*, …, *fM*。常用的编码矩阵构造方法有OVO（One-versus-one）、OVA（One-versus-all）、密集随机(Dense Random)、稀疏随机(Sparse Random)等。基分类器*fj*(*j*=1,2, …, *M*)对样本****的输出记为hj(**x**)。编码矩阵中的每一行称为码字(Codeword)，记为**ci**，其中的每一位。每个码字**ci**对应数据中的一类Ci，即编码矩阵中的每行都分别对应数据中的一类。编码矩阵的每一列通过将数据重新划分从而定义一个新的二分类问题，其中原本数据类别中的若干类组成正类，其余的组成负类，并在该列对应的位置分别记+1、-1,称为每类对应的编码值。每个基分类器分别对应编码矩阵的一列，并根据新的数据划分在训练集上进行训练。如果某类数据Ci对某个基分类器*j*的结果不产生影响，则在编码矩阵的对应位置*Tij*记为0，表示对应的分类器*fj*在训练时不考虑该类数据。图3-1为对一个4类数据创建的三种常见的ECOC编码矩阵。其中白色区域表示对应编码值为+1，黑色区域表示对应编码值为-1，灰色区域表示编码值为0。其中OVA和密集随机矩阵为二进制编码矩阵，稀疏随机矩阵为三进制编码矩阵。

****

**图3-1 三种常见的编码矩阵：**

**（a）密集随机（Dense Random）。（b）OVA（One-versus-all）。（c）稀疏随机（Sparse Random）。**

在解码阶段，对于样本，每个基分类器对其都有一个输出hj(**x**)，收集所有基分类器的输出，组成一个输出向量，记为**o**=H(**x**)，其中的每一位*oj=*hj(**x**)。接着将输出向量**o**分别与每一类数据Ci对应的码字**ci**进行对比，计算两向量之间的距离*di*,即输出向量的每一位*oj*到对应编码值*cij*的距离之和。*di*=D(H(**x**), **ci**)，距离函数D可采用欧式距离（Euclidean Distance）、汉明距离（Hamming Distance）或损失函数（Loss Function）[26]等其它距离计公式。样本**x**将被分类到具有最小距离的类，即该样本的输出结果。

图3-2为ECOC编码矩阵解码过程的一个实例。图中白色区域表示对应的编码值为+1，黑色区域表示对应的编码值为-1。该编码矩阵**T**4x6表示其对应的数据集有4类数据，使用6个基分类器*f1*,…,*f6*对该问题进行分类预测。矩阵中的第一列表示*f1*将C2~C4作为正类，将C1作为负类，即*f1*用于区分C1与C2、C3、C4，*f2*用于区分C1、C2与C3、C4，其余列类似。图中的输出向量o= [1, -1, 1, -1, -1, -1]，计算得到该样本的输出向量到4个类的汉明距离分别为d1=1，d2=3，d3=2，d4=3。，因此该样本的预测类别为C1。

****

**图3-2 ECOC编码矩阵解码实例**

## 3.3 算法特点

ECOC是一种集成学习方法，可以有效的把复杂的多分类问题分解为一系列较简单的二分类问题，从而避免了构造复杂的多分类学习模型。ECOC算法使用多个二分类器作为基分类器，结合编码矩阵进行总体决策，可以有效地纠正部分基分类器的错误，提高准确率，因此ECOC具有较强的鲁棒性和可靠性。但是由于编码矩阵对于数据中的每一类都要创建对应的一行码字，并且需要相应的增加基分类器的个数，因此对于具包含过多类别的数据，ECOC方法可能会生成过于庞大的编码矩阵和过多的基分类器，导致算法的效率降低，因此ECOC算法不适合于对包含过多类别的数据的分类。除此之外，如果ECOC所使用的基分类器输出的结果准确率过低，则会导致ECOC解码后得到的预测结果不可靠，算法的准确率也会较低。

现有ECOC的编码方式分为二进制编码和三进制编码。二进制编码矩阵由+1，-1构成，即数据只被分为正、负两类。使用二进制编码矩阵时，所有类别的数据都被强制划分为正类或负类。而同被划分为正类或负类的数据中，有些类别的数据的特征可能与同类的其它数据的特征不同，不适合将其与其它数据合并为同一类，因此会出现被划分为正类的数据实际得到的输出偏向负类的情况。例如某个类的数据被强制划分为正类，而基分类器实际的输出集中在0附近，甚至接近-1。这类数据就容易被错分。三进制编码矩阵由+1，-1，0构成。对于不适合强制归为正类或负类中的某一类的数据类别，编码矩阵将其归为新的一类，其编码值为0，表示其输出值未确定。基分类器在训练时忽略对应编码值为0的类别。三进制编码矩阵避免了将数据强制分为两类的问题。但是编码值为0的类别的数据由于没有参与对应基分类器的训练，也意味着基分类器在这些类上的输出不可控。虽然编码值为0，但实际的基分类器的输出值可能接近±1。这些都可能导致分类错误的出现。

总体来说，ECOC是一种有效的集成学习方法，可以有效的解决多分类问题，但是仍有改进的空间。

## 3.4 软编码

软编码（soft coding）在计算机编码领域是指从某些外部资源获得值或函数，例如预处理器宏、外部常量、配置文件、命令行参数或数据库表，它与硬编码相反[27]。而本文中所指的软编码的含义和该定义不同。本文中的软编码指通过某种方法，将原本固定的整数值用更加合理的实数值进行替换，使编码更加灵活有效。

ECOC所实际使用的基分类器大多输出的都是表示可靠性的实数值。对于这些分类器的输出值，使用整数值的二进制编码矩阵或三进制编码矩阵有时不能很好的适应数据。本文借用软编码的概念，对使用整数值的编码矩阵进行改变，用实数值或实数区间替代整数值。

本文中，软编码是指在ECOC编码矩阵的创建过程中，不使用原本固定的整数值作为编码矩阵中的元素，而是根据ECOC基分类器的在数据集上的实际输出分布对编码矩阵进行软化编码，使用实数值或实数区间替代原本的固定的整数值，形成软编码矩阵，从而提高准确率的方法。

## 3.5 本章小结

本章主要介绍了ECOC算法的概念和算法实现，并介绍了软编码的概念。

本章首先介绍了ECOC的概念，介绍作为一种集成学习方法的ECOC的基本原理，描述了ECOC的算法实现，介绍了算法的编码和解码算法，并举例说明了ECOC的解码过程。接着，本章分析了ECOC算法的优缺点以及适用范围，比较了二进制编码矩阵与三进制编码矩阵的特点，指出硬编码ECOC的不足。最后提出软编码的概念，为后续章节内容做铺垫。

# 第四章 软编码ECOC

## 4.1 算法概述

在目前已有的ECOC算法中，构成编码矩阵的编码值都是固定的与数据无关的整数（+1、-1，0），称为硬编码（Hard Coding）。使用硬编码方法的ECOC算法称为硬编码ECOC算法。在实际应用中，很多分类器（如神经网络）的输出都是表示概率的实数，其输出结果。对于样本，基分类器的输出结果hj(x)总是在分布在其对应的编码值*cij*的一侧。另一方面，由于编码矩阵进行的是强制的划分，将多个类强制组合成一个统一的正类或负类，实际分类器输出的结果hj(**x**)与对应的编码值*cij*的距离可能很远。这种情况下硬编码ECOC在解码时由于该基分类器输出结果的异常，输出的值与其对应的编码值的距离d(hj(**x**), *cij*)有较大偏差，就可能产生错误。如果有多个分类器出现这种错误，就会产生无法被纠正错误结果。软编码ECOC（Soft Coding ECOC）就是为解决这种问题的改进的ECOC算法。软编码ECOC使用基分类器在验证集上输出结果的均值或区间替代原本的编码矩阵中对应的整数编码值，得到软化的编码矩阵，使基分类器输出结果分布集中在其目标类对应的编码值附近，缩短输出向量o到其实际所属的类别对应的码字**ci**的距离，从而提高分类的准确率。

本文提出两种软化编码方法，一种为软值编码ECOC，使用实数值替代原编码矩阵中的整数值，另一种为区间编码ECOC，使用区间替代原编码矩阵中的整数值。两种算法的核心都是软化原有的编码矩阵，使编码矩阵中的编码值更好的贴近基分类器输出结果的实际分布。软编码ECOC使用的基分类器的输出 。

## 4.2 算法实现

### 4.2.1 软值编码ECOC

软值编码ECOC使用实数值替代原本编码矩阵中的整数值。软值编码ECOC首先使用现有的编码矩阵构造方法进行编码，构造硬编码矩阵**T**，并训练基分类器。对于某一类数据，假设基分类器*fi*的输出的值集中在值*mij*附近，则*mij*作为数据的中心值更能体现基分类器*fi*在该类数据上输出结果的分布情况，因此软值编码ECOC使用*mij*替换硬编码矩阵中对应的元素*Tij*（即*cij*），得到对应的软编码矩阵SM（Soft Matrix），新的编码值 。原编码矩阵中的硬编码值记为*cij’*。输出向量**o**到**ci**的距离，距离函数D可以为欧氏距离或损失函数等（不再使用汉明距离）。软值编码ECOC的解码过程与硬编码ECOC相同，计算样本**x**的输出向量**o**到每类对应的码字之间的距离，将其分类到具有最小距离的类别。输出的结果。

对于数据的集中值*mij*的确定，本文采用的一种简单而较有效的方法是使用*fj*在验证集上对所有的输出hj(**x**)的均值，即

 (4.1)

### 4.2.2 区间编码ECOC

区间编码ECOC使用基分类器输出值集中分布的区间替换编码矩阵中的数值。在编码阶段，首先使用训练集构造硬编码矩阵**T**并训练基分类器。收集基分类器在验证集上所有的输出向量**o**，组成输出结果矩阵**PM**(Predicted Matrix)，根据**PM**分别计算每个基分类器*fj*对每一类数据Ci输出结果的均值和方差。认为*fj*对Ci类数据的输出结果集中分布在区间内或区间附近，*k*为常数。使用区间替换原硬编码矩阵中的*cij’*，得到新的区间编码矩阵**IM**(Interval Matrix)。向量码字变为一组区间数组，仍将其记为**ci**，其每一位*cij*为区间，称为编码区间。区间的两端点, 。原硬编码矩阵中的码字记为**ci**’。

解码阶段，对于样本**x**的输出向量**o**，分别将其与每一类对应的区间数组**ci**进行比较，根据其中每一位的值与对应编码区间的关系计算该样本在每一类上的得分，最终将其分类到得分最高的一类。在将样本的输出与Ci类比较时，将向量**o**中的每一位分别与矩阵IM对应的区间进行比较，如果*oj*的值在区间内，则认为这一位的值与当前的Ci类是匹配的，这一位的得分为1。如果*oj*的值不在区间内，则这一位的值与Ci类不匹配，这一位的得分为0。*oj*在Ci类上的得分。

 (4.2)

但是对于某些分类器*fj*，其输出结果的方差可能较大，区间宽度也相应增加，虽然基分类器的输出值在该区间范围内的概率变大，但是该结果比在宽度较小的区间内的结果的可信度低。方差过大也意味着结果较不稳定，即该基分类器的输出结果本身不够可靠。同时，如果*fj*的输出值更接近0而不是±1，同样该输出值的可靠性也会降低。另一方面，可能*fj*对于的输出值*oj*不在类Ci对应的编码区间*cij*内，但*oj*距离区间边界很近。因此，在公式（4.2）的基础上，使用分类器输出结果的方差以及样本输出值到±1的距离确定权值，对样本的得分进行惩罚。对于在基分类器的输出值在编码区间外的情况，根据*oj*到区间中点的距离对得分进行补偿。对公式（4.2）修改得到如下公式：

 (4.3)

其中**w**为记录分类器权重的数组，数组中每一位的值*wj*为对应分类器*fj*的权值，

 (4.4)

最终样本对应每一类的得分，样本**x**的分类结果为。

**算法4-1 区间软编码**

输入：验证集上的分类器输出的结果矩阵**PM**

输出：软编码区间**IM**，记录分类器的权重的数组**w**

Begin

for *i* = 1 to *N*:

获取**PM**中实际标签为i的结果，存为矩阵**PM\_i**

for *j* = 1 to *M*:

计算矩阵**PM\_i**第*j*列的均值和标准差





end for

end for

for *j*=1 to *M*:



end for

end

**算法4-2 区间编码解码过程**

输入：区间编码矩阵**IM**，分类器对样本**x**的输出向量**o**，分类器的权重数组**w**

输出：样本**x**所属的类别预测结果result

Begin

for *i* = 1 to *N*:

for *j* = 0 to *M*:

if *oj* between *IMij.l* and *IMij.r* then



else





end if

end for



end for

end



## 4.3 算法分析

软编码ECOC方法的优点在于对于样本, 软编码ECOC算法可以有效缩短其输出向量H(**x**)到对应码字**ci**的距离D(H(**x**), **ci**)，从而在一定程度上减少样本被错分到其它类别的可能性。尤其是对于某些类的样本在基分类器上输出的值hj(**x**)总是离对应编码值*cij*距离较远时，软编码ECOC可以起到很好的纠正作用。



**图4-1 分类器*fj*对样本x1、x2的输出结果示意图**

以软值编码ECOC为例。图4-1为某个基分类器*fj*对两个样本的输出结果的示意图。分类器*fj*的输出区间，其中hj(**x1**)=0.5, hj(**x2**)=0.7。硬编码矩阵中对应的码字中的位*cij* = 1, 计算得到该分类器输出结果的均值*mij* = 0.6。使用硬编码时**x1**、**x2**到*cij*的平均距离为0.4，而使用软值编码时到*mij*的平均距离为0.1，由此可以看出，通过软化编码矩阵，软编码方法缩短了样本输出值到其所属类别对应编码值的平均距离。因为使用了数据的均值*mij*作为新的编码值替代原本的硬编码值，数据不再是只分布在编码值*cij*的一侧，而是分布在其两侧，并且在大部分情况下总是集中在其附近。



**图4-2 软编码与硬编码的对比实例**

如果*mij*的值与硬编码矩阵中对应的硬编码值*cij’*距离很远，在使用硬编码ECOC时预测时，对于样本，这一位基分类器在该类上的输出值到对应编码值的距离的值将很大，导致最终得到的距离*di*增大，同时也导致hj(**x**)到某个特定的类Ck的距离*dk*减小，属于类Ci的样本**x**将很容易被错分到类Ck。图4-2是软编码ECOC与硬编码ECOC对比的一个实例。图右上为该数据集上训练得到的硬编码矩阵，右下为根据硬编码矩阵软化后得到的软编码矩阵。根据软编码矩阵可以看出，C1类数据的输出结果对比硬编码矩阵存在异常：*f1*、*f4*、*f5*的输出值与硬编码矩阵中的编码值差距较大。因为该输出向量**o**相较于硬编码ECOC中对应的码字**c**’距离比较远，对图中的样本，使用欧式距离作为距离计算方式，计算得到硬编码矩阵中D(**o**,**c1**’)=2.81, D(**o**,**c3**’)=1.8，输出向量**o**到码字**c3**的距离比到**c1**更近，因此在硬编码ECOC中该样本被错分到类C3。而在软编码方法中，编码矩阵正确的反映了基分类器输出值的集中情况，计算得到D(**o**,**c1)**=1.29, D(**o**,**c3)**=1.55,该样本被正确分类到C1。

但是，由于根据数据的分布改变了原本的编码矩阵，原本编码矩阵中保持码字之间距离尽量最大化的条件被弱化了。仍以图4-2为例，图中的硬编码矩阵中，**c1**与**c3**间的距离D(**c1**, **c3**)=4。在使用硬编码矩阵时，由于**c1**与**c3**间的距离较远，基分类器对C3类数据的实际输出分布也较符合编码矩阵中的编码值，因此对于样本 ，结果几乎不会被误分到C1类。而在软值编码矩阵中，**c1**与**c3**的距离D(**c1**, **c3**)=2.18。可以看出，软化后的矩阵码字之间的距离减小了很多。对于图中的软编码矩阵，虽然可以纠正一部分因为C1类的异常而导致的C1类被误分类到C3类的情况，但是由于**c1**与**c3**的距离减小，可能会产生将样本误分类到C1类的新问题。

## 4.4 本章小结

本章主要介绍了两种软编码ECOC算法——软值编码ECOC和区间编码ECOC，并对软编码ECOC算法进行了具体分析。软值编码ECOC使用基分类器输出的均值、区间编码ECOC使用基分类器输出的集中区间替代硬编码矩阵中的整数编码值。软编码ECOC算法可以有效缩短样本的输出向量到样本所属类别对应码字的距离，从而减少错误。

# 第五章 实验设计与结果分析

## 5.1 实验设计

实验选择了9个较常用的UCI数据集[28]进行实验，使用的数据集信息如表5-1所示。每次实验对每个数据集进行10次验证，每次验证将数据集随机划为训练集（包含各类数据的50%的样本）、验证集和测试集（分别包含各类数据的25%的样本）。每次验证分别使用硬编码ECOC、软值编码ECOC、区间编码ECOC三种算法测试，最终对10次结果取均值进行比较。对硬编码ECOC算法，使用训练集进行编码和训练，使用测试集进行测试，不使用验证集；对软值编码ECOC和区间编码ECOC，使用训练集进行编码和训练，使用验证集对编码矩阵进行软化，使用测试集进行测试。

**表5-1 实验使用的数据集及部分信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 样本数 | 特征数 | 类别数 |
| balance | 625 | 4 | 3 |
| car | 1728 | 6 | 4 |
| dermatology | 366 | 33 | 6 |
| satimage | 6435 | 36 | 6 |
| segment | 2310 | 19 | 7 |
| thyroid | 215 | 5 | 3 |
| vehicle | 946 | 18 | 4 |
| vowel | 990 | 13 | 11 |
| zoo | 101 | 17 | 7 |

为了更好的验证软编码ECOC的效果，实验分别采用了OVA(One-versus-all)、密集随机(Dense Random)、稀疏随机(Sparse Random) 3种矩阵构造方式，以及支持向量机(Support Vector Machine, 简称SVM)、神经网络(Neural Network) 2种分类器作为ECOC的基分类器分别进行实验。实验中，使用区间编码时，区间中的*k*值均设为1。由于欧氏距离在解码过程中具有更高的准确率和更强的纠错能力，因此在实验中距离的计算使用欧氏距离。

对于每个类Ci，实际为正类、预测结果为正类的数量记为TPi（真正例）；实际为负类、预测结果为正类记为FPi（伪正例）；实际为负类、预测结果为负类记为TNi（真负例）；实际为正类、预测结果为负类记为FNi（伪负例）。如表5-2所示。

**表5-2 预测结果分类**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 预测类别 | |
|  |  | +1 | -1 |
| 实际类别 | +1 | TP | FN |
| -1 | FP | TN |

对分类结果分别采用以下测度进行评估。

1、准确率（accuracy），总体评估分类结果的准确率，评估识别出的样本中分类正确的数量所占的百分比。

 (5.1)

2、精确率（precision），根据每一类的准确率，最终得到所有类别准确率的均值。精确率主要关注真正例和伪正例，即被预测为正类的数据中有多少实际为正例，以及有多少负例被错分为正例。

 (5.2)

3、召回率（recall），主要关注真正例和伪负例，即所有样本中有多少被正确识别出的正例，以及有多少正例没有被识别出而被错分为负例。

 (5.3)

4、综合评价指标（f-measure）。精确率和召回率都只反映了数据的一个方面，综合评价指标综合精确率和召回率结果，对预测结果进行综合评估。本文中主要使用f1分数（f1-score）作为综合评价指标。f1分数中精确率和召回率的权重相同。

 (5.4)

对于实验结果，本文还使用混淆矩阵（Confusion Matrix）对数据集中每一类的分类结果进行分析。混淆矩阵中的每一行表示数据实际所属的类别，每一列表示数据被预测的类别[29]。混淆矩阵第*i*行第*j*列的数值表示实际属于类Ci的数据的预测结果为Cj的比例。混淆矩阵中，对角线上的值为该类数据中被正确分类的样本所占的比例。本文将混淆矩阵转化为热力图，颜色越深表示对应的值越大、越接近1。黑色表示数值为1，白色表示数值为0。

## 5.2 实验结果及分析

表5-3至表5-4为分别使用不同的编码矩阵（OVA、稀疏随机，密集随机）以及不同的基分类器（SVM、神经网络），硬编码、区间编码、软值编码三种方法在数据集car和vowel上得到的实验结果，包括在数据集上的准确率accuracy、精确率、召回率以及f1分数。加粗值为三种方法中的最优结果。

由表5-3可以看出，对于car数据集的分类，在使用SVM作为基分类器时，区间编码ECOC的准确率、精确率和f1分数最高，而软值编码ECOC的召回率最高。虽然使用稀疏随机矩阵时硬编码ECOC算法的精确率最高，但结果与软编码ECOC相差不多。使用神经网络作为分类器时，在使用稀疏随机矩阵和密集随机矩阵时区间编码ECOC的精确率最高，其余指标均是软值编码ECOC为最优。不论使何哪种编码矩阵的构造方法、何种基分类器，软编码ECOC的效果都优于硬编码ECOC。软编码ECOC与硬编码ECOC的f1分数最大相差达到了0.13，召回率的最大差值甚至达到了0.18。在该数据集上，软编码ECOC算法明显优于硬编码ECOC算法，而软值编码ECOC算法和区间编码ECOC算法效果相近。

**表5-3 硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵构造方法和不同的基分类器时在car数据集上的结果对比**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 基分类器： SVM | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| accuracy | 0.8557 | **0.8749** | 0.8731 | 0.8636 | **0.8715** | 0.8536 | 0.8636 | **0.8701** | 0.8673 |
| precision | 0.6865 | **0.7703** | 0.7431 | **0.7261** | 0.6990 | 0.6814 | 0.7397 | **0.7712** | 0.7566 |
| recall | 0.5920 | 0.7521 | **0.7851** | 0.6512 | 0.7787 | **0.7881** | 0.6506 | 0.7989 | **0.8025** |
| f1-score | 0.6229 | **0.7548** | 0.7515 | 0.6713 | **0.7171** | 0.7057 | 0.6726 | **0.7710** | 0.7589 |
|  |  |  |  |  |  |  |  |  |  |
| 基分类器： 神经网络 | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| accuracy | 0.9339 | 0.9408 | **0.9476** | 0.9165 | 0.9422 | **0.9441** | 0.9339 | 0.9508 | **0.9548** |
| precision | 0.8802 | 0.8820 | **0.8888** | 0.8647 | **0.8657** | 0.8590 | 0.8792 | **0.9076** | 0.9049 |
| recall | 0.7970 | 0.8753 | **0.9061** | 0.7417 | 0.8817 | **0.9247** | 0.7999 | 0.9061 | **0.9180** |
| f1-score | 0.8281 | 0.8744 | **0.8934** | 0.7791 | 0.8666 | **0.8852** | 0.8304 | 0.9046 | **0.9090** |

图5-1、图5-2为硬编码ECOC、区间编码ECOC、软值编码ECOC三种算法在car数据集上对各个类别的预测结果的精确率、召回率、f1分数结果的雷达图以及混淆矩阵热图。



**图5-1 硬编码ECOC、区间编码ECOC、软值编码ECOC对数据集car中各类数据的精确率、召回率、综合评价指标雷达图**

**（a）精确率结果。（b）召回率结果。（c）综合评价指标结果。**

从图5-1中可以看出，对于car数据集，虽然软值编码ECOC和区间编码ECOC对vgood类数据分类的精确率比硬编码ECOC低，但是硬编码ECOC对vgood和good两类数据的召回率很低，而软值编码ECOC和区间编码ECOC对各类数据的召回率都很高，在vgood和good两类数据上远超硬编码ECOC。在综合评价指标上，软值编码ECOC和区间编码ECOC对各类数据的效果都很好，而硬编码ECOC在good类的数据上由于召回率太低因此综合评价指标分数较低。总体效果上区间编码ECOC和软值编码ECOC要优于硬编码ECOC。



**图5-2 硬编码ECOC、区间编码ECOC、软值编码ECOC在数据集car上的混淆矩阵热图**

**（a）使用硬编码ECOC的结果。（b）使用区间编码ECOC的结果。（c）使用软值编码ECOC的结果。**

混淆矩阵中第*i*行第*j*列的值记为*Mij*。从图5-2的结果可知，使用硬编码ECOC时，某些类的数据可能很容易与其距离相近的类混淆，而软编码ECOC可以有效的减少某类数据被错分到其它类的情况（减小混淆矩阵中除对角线以外的数据的值），但是同时可能会新增少量的某些新的错分的情况。例如，硬编码ECOC结果的混淆矩阵热图中的*M21*颜色较深而*M22*颜色较浅，说明使用硬编码ECOC时，car数据集上第2类数据的样本有很多被错分类为第1类，被正确分类的样本较少。在软值编码ECOC和区间编码ECOC的结果中，*M21*颜色很浅而*M22*颜色接近黑色，表明软编码ECOC中有效的减少了这种第2类的样本被错分为第1类的情况。但硬编码矩阵中*M24*的值为0，而软值编码ECOC和区间编码ECOC的结果中*M24*为灰色，即有少数第2类的样本被错分为第4类。因为软编码ECOC在缩短样本的输出向量到样本所属类别的对应类别码字间的距离的同时，也缩短了编码矩阵中码字之间的距离，所以软编码ECOC容易产生新的错误，但是这种错误对结果一般影响较小。

**表5-4 硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵构造方法和不同的基分类器时在vowel数据集上的结果对比**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 基分类器： SVM | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| accuracy | 0.8492 | **0.8707** | 0.8533 | 0.7566 | 0.8351 | **0.8446** | 0.7095 | 0.7864 | **0.8124** |
| precision | 0.8570 | **0.8776** | 0.8650 | 0.7873 | 0.8450 | **0.8548** | 0.7298 | 0.7973 | **0.8232** |
| recall | 0.8492 | **0.8707** | 0.8533 | 0.7566 | 0.8351 | **0.8446** | 0.7095 | 0.7864 | **0.8124** |
| f1-score | 0.8477 | **0.8704** | 0.8541 | 0.7509 | **0.8313** | 0.8428 | 0.6972 | 0.7836 | **0.8092** |
|  |  |  |  |  |  |  |  |  |  |
| 基分类器： 神经网络 | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| accuracy | 0.8888 | 0.8946 | **0.8963** | 0.7483 | 0.8066 | **0.8236** | 0.7607 | 0.8219 | **0.8450** |
| precision | 0.8961 | 0.9005 | **0.9046** | 0.7748 | 0.8190 | **0.8308** | 0.7849 | 0.8283 | **0.8524** |
| recall | 0.8888 | 0.8946 | **0.8963** | 0.7483 | 0.8066 | **0.8236** | 0.7607 | 0.8219 | **0.8450** |
| f1-score | 0.8868 | 0.8943 | **0.8974** | 0.7368 | 0.8020 | **0.8203** | 0.7553 | 0.8189 | **0.8426** |

根据表5-4的结果，对于vowel数据集的分类，在使用OVA编码矩阵、基分类器使用SVM时，区间编码ECOC的结果最优，而其余结果均为软值编码ECOC最优。在该数据集上区间编码ECOC和软值编码ECOC均优于硬编码ECOC，而软值编码ECOC的效果为最佳。

图5-3、图5-4为硬编码ECOC、区间编码ECOC及软值编码ECOC三种算法在vowel数据集上对各个类别的预测结果的精确率、召回率、f1分数结果的雷达图以及混淆矩阵热图。

从图5-3中可以看出，在vowel数据集上，软值编码ECOC和软编码ECOC对各类数据预测结果的精确率、召回率、综合评价指标几乎均优于硬编码ECOC的结果。使用硬编码ECOC得到的结果中各类数据的精确率、召回率、综合评价指标差异很大，部分类别的精确率、召回率、f1分数均较低，而在软值编码ECOC和区间编码ECOC得到的结果中，各类数据的结果差异较小，各类的精确率、召回率、综合评价指标比较接近，各类数据的分类水平较平均。



**图5-3 硬编码ECOC、区间编码ECOC、软值编码ECOC对数据集vowel中各类数据的精确率、召回率、综合评价指标雷达图**

**（a）精确率结果。（b）召回率结果。（c）综合评价指标结果。**



**图5-4 硬编码ECOC、区间编码ECOC、软值编码ECOC在数据集vowel上的混淆矩阵热图**

**（a）使用硬编码ECOC的结果。（b）使用区间编码ECOC的结果。（c）使用软值编码ECOC的结果。**

从图5-4可以看出，在vowel数据集上，硬编码ECOC的结果仍会产生很多较严重的误分类情况，例如第2类数据样本大部分被错分为第1类和第4类。第2类的样本在硬编码ECOC的分类结果中正确的比例很少，而这类错误分类情况在软值编码ECOC和区间编码ECOC的结果中有明显改善，某类数据样本大量被错分为其它类数据的情况在软编码ECOC算法的结果中几乎没有。

根据表5-3、表5-4以及图5-1至5-4的结果，软编码ECOC算法相较硬编码ECOC算法效果的提升，主要在于软编码ECOC算法对硬编码ECOC中召回率较低的数据类别的错误所进行的纠正。尤其是当某个类的样本在硬编码ECOC中被大量被错分到其它类别时，软编码ECOC算法具有很好的纠错能力，可以比较有效的纠正硬编码ECOC可能发生的错误，提高召回率较低的类别的召回率，从而得到更好的分类结果。



**图5-5 car数据集中部分易被错分的样本的输出向量以及相对应的软值编码矩阵、硬编码矩阵的热图对比**

图5-5热图中，红色代表值为1，白色代表值为0，蓝色代表值为-1。左侧为易被错分的样本的输出向量组成的矩阵，其中的每一行为一个样本的输出向量。该矩阵中的样本均为C1类数据的样本。在使用硬编码矩阵解码时，根据右侧的硬编码矩阵热图可以看出，图中的样本输出向量与硬编码矩阵中对应的码字c1距离很远，特别是第3位的输出值，硬编码矩阵中第3位的值为-1，而输出向量中的值更接近1。由于这一位输出的异常，图中的输出向量更接近硬编码矩阵中的c3，因此在使用硬编码矩阵解码时，这些样本被错分为C3类。而在软编码矩阵中，由于使用均值替代了整数值，对应码字c1更好的反映了该类数据的输出结果，虽然输出向量的第三位存在异常，但是在软编码矩阵中的c1的第三位的值到样本对应码字的距离比硬编码方法中减小了，因此软编码矩阵仍能将这种错误纠正，样本最终被正确分类为C1。



**图5-6 car数据集中部分易被错分的样本的输出向量以及相对应的区间编码矩阵、硬编码矩阵**

图5-6左侧为易被错分的样本的输出向量组成的矩阵，其中的每一行为一个样本的输出向量。该矩阵中的样本均为C1类数据的样本。在使用硬编码矩阵解码时，由于输出向量中第1位和第3位的异常，样本输出向量更接近c3。因此，在使用硬编码矩阵解码时，这些样本被错分为C3类。而在区间编码矩阵中，由于使用实数区间替代了整数值，对应码字更好的反映了该类数据的输出结果。，虽然输出向量中第1位的值存在异常，区间编码矩阵中c1第一位的区间包含了对应基分类器在该类上的大部分输出值。编码值第3位异常，虽然有些值未被包含在区间中，但通过距离补偿的计算，仍可以算作正确匹配。并且由于输出向量中第3位的值大多数接近0，且对应的区间宽度较大（即基分类器输出结果的标准差较大），因此该位受到惩罚较大，在计算分数时的权重很低，其错误可以被其它位纠正。在区间编码ECOC中图中的易错样本最终被正确分类为C1。

表5-5至表5-7为在其余7个数据集上硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵和不同的基分类器时的精确率、召回率、f1分数结果。

**表5-5 硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵构造方法和不同的基分类器时在不同数据集上的精确率结果**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类器： SVM | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.6918 | **0.8510** | 0.8256 | 0.6438 | **0.8140** | 0.8135 | 0.6629 | **0.8331** | 0.8027 |
| dermatology | **0.9743** | 0.9741 | 0.9739 | 0.9447 | 0.9619 | **0.9729** | 0.9724 | **0.9771** | 0.9753 |
| satimage | **0.8725** | 0.8690 | 0.8701 | 0.8246 | 0.8475 | **0.8516** | **0.8691** | 0.8592 | 0.8606 |
| segment | 0.9267 | **0.9275** | 0.9265 | 0.8940 | 0.8954 | **0.9153** | 0.9097 | 0.9141 | **0.9174** |
| thyroid | **0.9587** | 0.9550 | 0.9567 | 0.9593 | 0.9417 | **0.9642** | **0.9417** | 0.9330 | 0.9389 |
| vehicle | 0.7395 | **0.7565** | 0.7490 | **0.7528** | 0.7235 | 0.7421 | 0.7354 | **0.7452** | 0.7432 |
| zoo | **0.7748** | 0.7630 | 0.7690 | **0.8487** | 0.7987 | 0.8388 | 0.8286 | 0.8036 | **0.8714** |
|  |  |  |  |  |  |  |  |  |  |
| 基分类器： 神经网络 | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.8091 | **0.8426** | 0.8393 | 0.7326 | **0.8498** | 0.8317 | 0.8119 | 0.8707 | **0.8710** |
| dermatology | 0.9558 | **0.9602** | 0.9579 | 0.9527 | **0.9566** | 0.9536 | 0.9645 | 0.9673 | **0.9740** |
| satimage | **0.8827** | 0.8771 | 0.8776 | 0.8385 | **0.8569** | 0.8481 | **0.8700** | 0.8671 | 0.8685 |
| segment | 0.9565 | 0.9563 | **0.9566** | 0.8968 | **0.9268** | 0.9177 | 0.9534 | 0.9539 | **0.9540** |
| thyroid | 0.9598 | **0.9629** | 0.9573 | **0.9672** | 0.9479 | 0.9615 | 0.9711 | 0.9675 | **0.9731** |
| vehicle | 0.7508 | 0.7653 | **0.7702** | 0.7673 | 0.7829 | **0.7843** | 0.7494 | 0.7542 | **0.7647** |
| zoo | **0.8833** | 0.8357 | 0.7905 | 0.7548 | 0.7011 | **0.8024** | **0.8762** | 0.8464 | 0.8118 |

根据表5-5中的精确率结果，软值编码ECOC、区间编码ECOC和硬编码ECOC的精确率在大部分数据集上相差不大。使用SVM作为基分类器时，在balance数据集上区间编码ECOC和软值编码ECOC方法的精确率明显高于硬编码ECOC。使用神经网络作为基分类器时，硬编码ECOC在zoo数据集上结果要优于软编码ECOC和软值编码ECOC。在balance数据集上，区间编码ECOC和软值编码ECOC的结果对比硬编码ECOC有较大提升。在表5-5中，硬编码ECOC共胜出13次，区间编码ECOC胜出14次，软值编码胜出15次。根据精确率结果，硬编码ECOC、区间编码ECOC、软值编码ECOC三种方法的效果相近。

**表5-6 硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵构造方法和不同的基分类器时在不同数据集上的召回率结果**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类器： SVM | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.6653 | 0.8778 | **0.8949** | 0.6671 | 0.8514 | **0.8870** | 0.6606 | 0.8579 | **0.8630** |
| dermatology | 0.9725 | 0.9728 | **0.9736** | 0.9408 | 0.9568 | **0.9689** | 0.9718 | 0.9749 | **0.9752** |
| satimage | 0.8717 | 0.8747 | **0.8748** | 0.8068 | 0.8493 | **0.8504** | 0.8590 | **0.8640** | 0.8615 |
| segment | 0.9239 | **0.9242** | 0.9225 | 0.8740 | 0.8908 | **0.9118** | 0.9045 | 0.9092 | **0.9148** |
| thyroid | 0.9113 | 0.9104 | **0.9199** | 0.9047 | 0.9038 | **0.9134** | 0.8839 | 0.8868 | **0.8871** |
| vehicle | 0.7506 | **0.7566** | 0.7490 | **0.7564** | 0.7352 | 0.7507 | 0.7403 | **0.7479** | 0.7418 |
| zoo | 0.8133 | 0.8048 | **0.8138** | 0.8581 | 0.8143 | **0.8624** | 0.8476 | 0.8429 | **0.8881** |
|  |  |  |  |  |  |  |  |  |  |
| 基分类器： 神经网络 | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.6759 | **0.9338** | 0.9319 | 0.6810 | **0.9199** | 0.9176 | 0.6824 | 0.9199 | **0.9403** |
| dermatology | 0.9541 | **0.9593** | 0.9576 | 0.9505 | 0.9523 | **0.9568** | 0.9623 | 0.9649 | **0.9722** |
| satimage | 0.8708 | **0.8775** | 0.8769 | 0.8255 | **0.8553** | 0.8489 | 0.8595 | **0.8722** | 0.8706 |
| segment | **0.9563** | 0.9557 | 0.9561 | 0.8911 | **0.9226** | 0.9105 | 0.9524 | 0.9530 | **0.9531** |
| thyroid | 0.8711 | **0.8884** | 0.8833 | 0.9131 | **0.9612** | **0.9612** | 0.9208 | 0.9452 | **0.9509** |
| vehicle | 0.7610 | 0.7723 | **0.7759** | 0.7657 | 0.7853 | **0.7899** | 0.7555 | 0.7616 | **0.7702** |
| zoo | **0.8929** | 0.8643 | 0.8310 | 0.8000 | 0.7400 | **0.8357** | **0.9057** | 0.8771 | 0.8486 |

根据表5-6中召回率的结果，区间编码ECOC和软值编码ECOC对硬编码ECOC的召回率在大部分数据集上可以提高0.01~0.02，在car数据集上的提高达到了0.2以上。可以发现，在使用稀疏随机编码矩阵和密集随机编码矩阵时，软值编码ECOC具有比较明显的优势，尤其是使用SVM作为基分类器时，不仅在大多数结果的比较中都胜出，并且对于硬编码ECOC召回率的提升较明显。在所有结果中，硬编码ECOC胜出的结果只有4个，区间编码ECOC胜出的结果有13个，而软值编码ECOC胜出的结果有26个。可以认为软值编码ECOC有较好的效果，其次是区间编码ECOC，两种软编码ECOC算法效果均优于硬编码ECOC。

**表5-7 硬编码ECOC、区间编码ECOC、软值编码ECOC在使用不同的编码矩阵构造方法和不同的基分类器时在不同数据集上的f1分数结果**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类器： SVM | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.6514 | **0.8587** | 0.8464 | 0.6524 | 0.8269 | **0.8322** | 0.6428 | **0.8370** | 0.8200 |
| dermatology | 0.9723 | 0.9724 | **0.9727** | 0.9396 | 0.9588 | **0.9697** | 0.9710 | **0.9748** | 0.9742 |
| satimage | **0.8715** | 0.8700 | 0.8699 | 0.7936 | 0.8456 | **0.8467** | **0.8627** | 0.8603 | 0.8600 |
| segment | 0.9237 | **0.9247** | 0.9232 | 0.8699 | 0.8898 | **0.9091** | 0.9033 | 0.9103 | **0.9149** |
| thyroid | 0.9287 | 0.9270 | **0.9323** | 0.9229 | 0.9155 | **0.9328** | 0.9062 | 0.9047 | **0.9068** |
| vehicle | 0.7403 | **0.7498** | 0.7447 | **0.7443** | 0.7187 | 0.7424 | 0.7348 | **0.7435** | 0.7410 |
| zoo | **0.7785** | 0.7665 | 0.7709 | **0.8407** | 0.7965 | 0.8348 | 0.8232 | 0.8089 | **0.8671** |
|  |  |  |  |  |  |  |  |  |  |
| 基分类器： 神经网络 | OVA | | | Sparse Random | | | Dense Random | | |
| 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 | 硬编码 | 区间编码 | 软值编码 |
| balance | 0.6670 | **0.8693** | 0.8652 | 0.6734 | **0.8722** | 0.8567 | 0.6773 | 0.8896 | **0.8956** |
| dermatology | 0.9534 | **0.9587** | 0.9566 | 0.9488 | 0.9513 | **0.9543** | 0.9620 | 0.9651 | **0.9724** |
| satimage | 0.8755 | **0.8769** | 0.8766 | 0.8153 | **0.8541** | 0.8457 | 0.8626 | **0.8686** | **0.8686** |
| segment | 0.9560 | 0.9557 | **0.9561** | 0.8799 | **0.9218** | 0.9068 | 0.9522 | 0.9531 | **0.9532** |
| thyroid | 0.9038 | **0.9178** | 0.9122 | 0.9333 | 0.9515 | **0.9592** | 0.9394 | 0.9530 | **0.9584** |
| vehicle | 0.7534 | 0.7642 | **0.7714** | 0.7530 | 0.7765 | **0.7844** | 0.7464 | 0.7526 | **0.7665** |
| zoo | **0.8815** | 0.8430 | 0.7996 | 0.7558 | 0.7047 | **0.8070** | **0.8850** | 0.8556 | 0.8259 |

表5-7中f1分数为为对精确率和召回率的综合评价指标。根据f1分数的结果，实验中硬编码ECOC胜出的结果有7个，区间编码ECOC胜出的结果有14个，而软值编码ECOC胜出的结果有21个。根据综合评分胜出的次数看，软值编码ECOC具有最好的总体效果，区间编码ECOC其次。硬编码ECOC算法的综合效果最差。

根据表5-5到表5-7不同数据集上的结果分析，对硬编码ECOC，当各种类别的数据分布不均衡时，硬编码ECOC的选择将倾向于数量多的类别。在将多分类问题重新划分为二分类问题后，数量少的类别可能会因为在新的类别中占的比重太小，从而基分类器对其缺乏训练，导致该基分类器对比重较小的类的输出结果与对应码字不匹配。在硬编码ECOC中，该类数据的召回率将低于其它样本数量多的数据，很容易被误分类到其它类别。而软编码ECOC可以在一定程度上纠正这种倾向，使各类数据都分类效果都保持在较高的水平，提高平均的准确率。

图5-5为硬编码ECOC、区间编码ECOC、软值编码ECOC在不同数据集、使用不同编码矩阵、不同基分类器的所有结果的准确率、精确率、召回率、以及f1分数的均值的比较。

**图5-5 硬编码ECOC、区间编码ECOC、软值编码ECOC的所有实验结果的准确率、精确率、召回率、f1分数的均值比较**

图5-5中的结果中，而区间编码ECOC和软值编码ECOC的平均准确率略高于硬编码ECOC，但三种算法的平均准确率相差不多，而区间编码ECOC和软值编码ECOC的精确率、召回率、f1分数的均值明显高于硬编码ECOC。其中，软值编码ECOC的所有测试指标的平均值均为三种算法中最高的。通过图5-5中的均值结果可以看出，在实验数据集中，软值编码ECOC算法具有最好的效果。区间编码ECOC和软编码ECOC对现有硬编码ECOC算法的效果提升主要体现在精确率和召回率上。

总体而言，软编码算法ECOC的效果要优于硬编码ECOC，但是软编码ECOC并不是在任何情况下都优于硬编码ECOC。软编码ECOC算法在缩短输出向量到对于类别码字的距离的同时，也减小了各码字间的距离，因此在解码时可能会产生硬编码ECOC所没有的新的错误分类情况。码字之间的距离过小会导致编码矩阵对基分类器错误的纠正能力降低，影响分类结果的准确率，软编码ECOC算法的结果将受到比较严重的影响。在硬编码ECOC的准确率很高的数据集上，软编码ECOC相较于硬编码ECOC准确率的提升并不明显，在部分数据集上可能因为离群点的影响导致软化后编码矩阵中的值偏离数据真实的集中值，从而影响预测结果的准确率。

## 5.3 本章小结

本章首先介绍了实验设计，简单介绍了实验中所使用的UCI数据集，并介绍了实验中所使用的四种测度。接着详细分析了car数据集和vowel数据结的实验结果，并结合其它数据集的实验数据对结果进行了分析，比较了硬编码ECOC、区间编码ECOC、软值编码ECOC三种算法的分类效果。实验结果表明，软值编码ECOC算法在所选数据集上的分类效果最佳，其次是区间编码ECOC算法。两种软编码ECOC算法的分类效果在大多数情况下均优于目前的硬编码ECOC算法。

# 第六章 总结与展望

## 6.1 总结

本文首先介绍了多分类问题、集成学习以及一些常用的分类器，接着介绍了ECOC算法的概念及实现，比较了二进制和三进制编码矩阵的，分析了其优缺点，总结了现有ECOC方法中出现数据实际分布和编码矩阵中对应类别码字距离可能较远的问题，针对该问题提出了一种改进的软编码ECOC方法，包括软值编码ECOC和区间编码ECOC，并通过实验进行了验证。

本文使用软编码的方法，使用ECOC中基分类器输出值的集中值或区间对编码矩阵中原来的整数值进行替换，得到软化的编码矩阵，使得解码过程中输出向量与样本实际所属类别的对应码字的距离减小，从而提升了分类的准确率。本文通过实验比较了软编码ECOC和硬编码ECOC在一些UCI数据集上的效果，验证了软编码ECOC对现有ECOC算法有一定的提升效果。但是由于软编码ECOC的编码矩阵中的值由数据的分布确定，因此在数据量较小的数据集上容易受到离群点的印象从而导致准确率的下降，并且在数据分布不稳定的数据集上软编码ECOC的效果还有待提升。软编码ECOC算法在减小样本输出向量到其所属类别对应码字之间距离的同时，也会减小编码矩阵码字之间的距离，从而可能产生新的错误分类情况，对分类的准确率产生一定影响。

## 6.2 工作展望

作为一种有效的集成学习方法，ECOC方法具有较广泛的实用性。软编码ECOC方法可以有效的提升ECOC方法的准确率。区间ECOC方法还有更多适用场景，例如正在研究中的开集数据识别算法中，可以使用区间ECOC对未知类别的数据进行识别。但是目前的软区间ECOC方法对于ECOC方法效果的提升还不大，目前的算法还有许多值得优化的地方：

1、软值编码ECOC中，目前数据集中值的确定使用了验证集上输出的平均值，这在某些数据集上可能会受离群点的影响而无法正确的反应数据的真实分布。基分类器输出值的方差过大时，使用均值对编码矩阵进行软化的效果不尽如人意。

2、目前区间编码ECOC使用标准差来确定区间宽度，由于大部分基分类器的输出结果不符合正态分布特征，使用标准差确定区间宽度的策略并不能完全体现数据的集中分布情况，有待更优的区间选择策略。

3、区间编码ECOC，根据基分类器输出结果的方差确定的惩罚权值还有待进一步优化以获得更准确的结果。

4、软编码ECOC算法在减小样本输出向量到其所属类别对应码字之间距离的同时，也会减小编码矩阵码字之间的距离，从而可能产生新的错误分类情况。

本文主要的意义在于提出将ECOC算法中的编码矩阵值进行软化从而提高算法准确率的算法思路。

# 参考文献

1. T. G. Dietterich, G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes[J]. Journal of Artificial Intelligence Research, 1995, 2:263-286.
2. E. L. Allwein, R. E. Schapire, Y Singer. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers[M] Journal of Machine Learning Research 2000, 113-141
3. O. Pujol, P. Radeva, J. Vitria. Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes [J]. IEEE Computer Society, 2006, 28 (6) :1007-12.
4. S. Escalera, O. Pujo, P. Radeva. ECOC-ONE: A Novel Coding and Decoding Strategy [J]. International Conference on Pattern Recognition, 2006, 3 (3):578-581
5. S. Escalera, O. Pujo, P. Radeva. Problem-dependent designs for Error Correcting Output Codes. <https://www.researchgate.net/publication/228849387>.
6. S. Escalera, D. Masip, E. Puertas, et al. Online error correcting output codes [J]. Pattern Recognition Letters, 2011, 32:458-467.
7. J. Zhou, Y. Yang, M. Zhang, H. Xing. et al. Constructing ECOC based on confusion matrix for multiclass learning problems [J]. China Inf. Sci. 2016, 59: 1.
8. N. Hatami. Thinned-ECOC ensemble based on sequential code shrinking [J]. Expert Systems with Applications, 2012,39:936-947.
9. P. Simeone, C. Marrocco, F. Tortorella. A Framework for Multiclass Reject in ECOC Classification Systems[M]. Computer Science, 2007.
10. P. Simeone, C. Marrocco, F. Tortorella. Two Stage Reject Rule for ECOC Classification Systems [M]. Computer Science, 2011.
11. P. Simeone, C. Marrocco, F. Tortorella. Design of reject rules for ECOC classification systems[J]. Pattern Recognition, 2012, 45: 863-875.
12. P. Simeone, C. Marrocco, F. Tortorella. Shaping the Error-Reject Curve of Error Correcting Output Coding Systems. Computer Science, 2011.
13. KH Liu, ZH Zeng, VTY Ng Hierarchical Ensemble of ECOC for cancer classification based on multi-class microarray data [J]. Information Sciences, 2016, 349-350:102-118.
14. P. Simeone, C.Marrocco, F. Tortorella. Weighted Decoding ECOC for Facial Action Unit Classification[M]. Applications of Supervised and Unsupervised Ensemble Methods, 2009.
15. Mohamed Aly. Survey on Multiclass Classification Methods, CiteSeer, 2005.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.175.107&rep=rep1&type=pdf>.

1. Zhi-Hua Zhou. Ensemble Learning,

<https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/springerEBR09.pdf>.

1. B Krawczyk, M Woźniak. Untrained weighted classifier combination with embedded ensemble pruning. Neurocomputing, 2016, 196:14-22
2. Matteo Re , Giorgio Valentini. Ensemble methods: a review.

https://homes.di.unimi.it/~valentini/papers/ens.review.revised.pdf.

1. C. Cortes, V. Vapnik. Support-vector networks [M]. Machine Learning, 1995.
2. <https://www.cnblogs.com/spoorer/p/6551220.html>.
3. J. Weston, C. Watkins. Multi-class Support Vector Machines.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.9594&rep=rep1&type=pdf.

1. <http://ufldl.stanford.edu/wiki/index.php/Neural_Networks>.
2. <https://blog.csdn.net/sinat_33741547/article/details/76652611>.
3. Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
4. <https://www.cnblogs.com/hesi/p/7218602.html>.
5. L I Kuncheva. Using diversity measures for generating error-correcting output codes in classifier ensembles[J]. Pattern Recognition Letters, 2005, 26(1):83-90.
6. Wikipedia, <https://en.wikipedia.org/wiki/Softcoding>.
7. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>.
8. Wikipedia, <https://en.wikipedia.org/wiki/Confusion_matrix>