西安交通大学

硕士学位论文

基于多标签分类的心血管疾病预测模型研究与应用

学位申请人：

指导教师：

类别（领域）：计算机技术

2018 年 5 月

Research and Application of Prediction Model of Cardiovascular disease based on Multi-label Classification

A thesis submitted to

Xi'an Jiaotong University

in partial fulfillment of the requirements

for the degree of

Master of Engineering

By

Supervisor:

(Computer Technology)

May 2018

论文题目：基于多标签分类的心血管疾病预测模型研究与应用
类别（领域）：计算机技术
学位申请人：
指导教师：

摘 要

心血管疾病是一种严重威胁人类健康的常见病，具有高患病率、高致残率和高死亡率的特点，所以做到早预测，早治疗，提高心血管疾病患者的生存率显得极其重要。本文根据临床数据中患者所患疾病的多标签特性，基于多标签分类技术对心血管疾病预测模型进行研究，有效预测一个病人可能存在的并发症。

本文以实际临床数据为基础，首先分析数据属性与常见心血管疾病的关系，统计分析多种心血管疾病的数据分布，确定用于多标签分类的目标标签，对目标标签的临床数据进行提取和预处理，由于数据量达到亿级，利用 spark 大数据平台完成数据加载预处理，获得具有多标签特性的心血管疾病数据集，再采用统计学方法和分类算法进行二次特征选择，选择表现效果最好的特征集。然后为解决心血管疾病数据集分布的稀疏性、不均衡性，针对心血管疾病数据集中大样本过度冗余、小样本缺乏数据表示等不均衡问题，提出了多标签双重自适应随机采样算法。最后根据心血管疾病数据集的特性，提出了一种基于混合策略的多标签分类框架来构建心血管疾病预测模型，该框架面对分类算法无法训练大数据量的问题，本文提出了基于数据均衡性的混洗方法，该方法不仅提高了混合框架的训练效果，还有效解决了现有算法的计算瓶颈。

本文解决海量医疗数据加载与预处理、心血管疾病影响因子选择、多标签数据集不均衡性、多标签分类算法应用于心血管疾病预测与评估等问题，重点在于提出的多标签双重自适应随机采样算法，该算法均衡了多标签数据集的分布，提高模型预测的可靠性、准确性，进而在此基础上结合数据局部性特点和全局标签相关性特点的优势，提出了基于混合策略的多标签分类框架来构建心血管疾病预测模型，获得了好的预测性能，展示了多标签分类算法在心血管疾病预测领域的优势。

关 键 词：心血管疾病；多标签；多标签不均衡性；重采样；混合策略
论文类型：应用研究

Title: Research and Application of Prediction Model of Cardiovascular disease based on Multi-label Classification
Professional Fields: the degree of Master of Engineering
Applicant:
Supervisor:

ABSTRACT

Cardiovascular disease has become a serious threat to human health. It has the characteristics of high morbidity, high disability and high mortality. Therefore, early prediction, early treatment and improving the survival rate of cardiovascular disease are extremely important. Based on the multi-label characteristics of the patient in clinical data, this paper studies the prediction model of cardiovascular disease based on the multi-label classification.

On the basis of actual clinical data, the relationship between data attributes and cardiovascular diseases was analyzed, and the distribution of cardiovascular diseases dataset was analyzed, and the labels were determined. Then the clinical data of the labels were extracted and preprocessed. Owing to the number of clinical data reached ten million, Spark technology completed loading and preprocessing the multi-label of cardiovascular disease dataset, and combining statistical methods with classification algorithms to select the best features. Aiming at the imbalance of multi-label dataset, such as large sample redundancy and small sample lacking of data representation, a multi-label double adaptive random sampling algorithm was proposed. Finally, according to the characteristics of the dataset of cardiovascular disease, a multi-label classification framework based on mixed strategy was proposed to construct the prediction model of cardiovascular disease. Faced with the classification algorithm cannot train large amount of data, this paper proposed a data mixing method based on equilibrium. This method not only improved the performance of the hybrid framework, but also effectively solved the bottleneck of the existing algorithms.

In this paper, we solved these problems, such as the massive data loading, cardiovascular disease pretreatment, factor selection, highly unbalanced for multi-label dataset, and the prediction and evaluation of cardiovascular disease. The paper emphasizes the multi-label double adaptive random sampling method, because the method balanced the distribution of multi-label dataset and improved reliability and accuracy. Based on that, the advantages of data locality and global correlation between labels were also discussed. Then a multi-label classification framework based on hybrid strategy was proposed to build cardiovascular disease prediction model and achieve good prediction performance.

KEY WORDS:Cardiovascular disease;Multi-label classification;Multi-label class imbalance; Resampling; Hybrid scheme
TYPE OF THESIS: Application Research

目 录

CONTENTS

1 绪论

1.1 课题研究背景和意义

近年来，我国医疗体制改革不断深化，医疗领域的信息化不断完善，每天产生出大量的医疗数据，这些数据中不仅包括电子病历、体检信息等数据，还涉及到公共卫生管理信息平台特别是疾控部门的医疗信息，这些医疗数据对防控区域性爆发的流行病、疾病间关系的发现具有很大的意义[1]。随着大数据技术的发展，对医疗数据的分析及挖掘越来越被重视。

心血管疾病是一种严重威胁人类特别是中老年人健康的常见慢性病，发病时不易察觉、极易危及生命、易导致多种并发症、疗程长且难以治愈等特点，即使应用目前最先进、完善的治疗手段，仍可有将近 50%的心脑血管意外幸存者生活不能完全自理，全世界每年死于心脑血管疾病的人数高达 1500 万人，居各种死因首位[2]。图 1-1 是近年来农村、城市各种疾病发病所占的百分比，可见心血管疾病是严重威胁人类身体健康的最主要因素，所以做到早预测，早治疗，提高心血管患者的生存率显得极其重要。

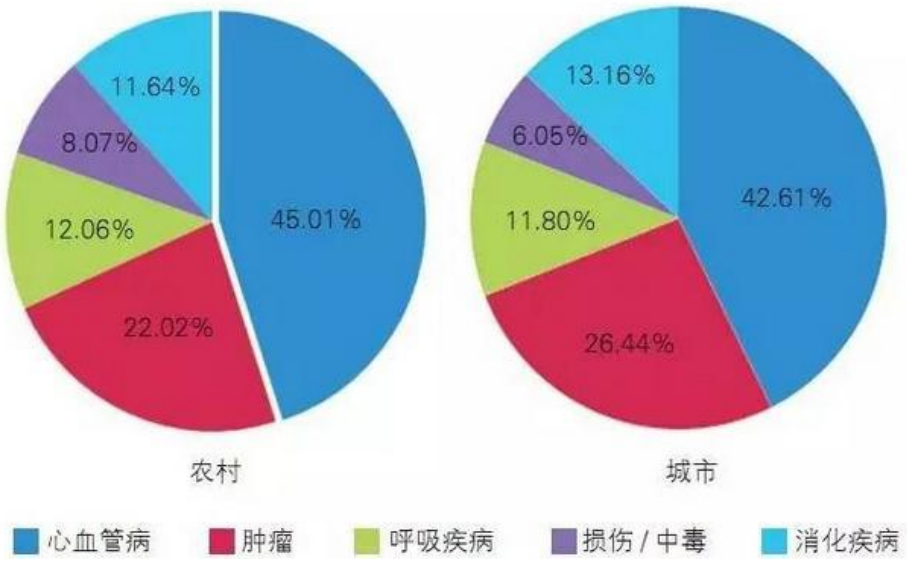


图 1-1 心血管疾病在农村、城市的发病率

通过疾病预测评估过早干预用户可能患有的疾病，提高患者生存率一直是医学领域的研究热点。以前人们利用人口统计学、医疗条件、生活常规等基本信息来计算发展某种疾病的可能性，这种计算是使用基于方程的数学方法和工具完成的，一般涉及到的很少的变量和数据[3]。随着机器学习、大数据技术的快速崛起，不仅能够处理大量医学数据而且可处理大规模的变量，充分挖掘了医学数据潜在的规律，提高疾病预测的准确性。

监督学习作为机器学习领域中最多研究、应用最广泛的方法之一，通过已有的训练样本学习一个模型，使该模型对任意输入的待预测样本，都能得到一个好的预测输出结果[4]。传统的单标签分类算法是把研究对象当作具有明确、单一的语义，对象被标注为唯一的类别标签,已经取得了巨大的研究成果。由于现实世界的很多对象往往具有多义性和模糊性，并不具有唯一的语义，例如一张风景图片包含了海洋、沙滩、人三大类标签。一位病人可能同时患有心衰、心梗、脑卒中等多种心血管疾病，此时分类算法必须能够准确的识别该对象中的多个标签，而单标签算法无法准确描述一个对象可能和多个标签相关的问题，因此多标签分类（或多标签学习）研究应运而生。在多标签分类中每个待分类对象由一个样本描述，该样本具有多个而不是唯一的类别标签，分类的目标是将所有合适的类别标签赋予未知样本。多标签分类广泛应用于图像标注、生物医学、文本分类等众多领域，不同的领域对多标签分类有不同的要求。

医学领域中真实医疗数据保密性强而不容易获取，同时这些数据内部存在极其复杂的关系且具有重大挖掘价值，为现代医疗发展起着必不可少的作用，因此医学领域的数据分析和研究一直是社会关注的焦点和相关学科的重点研究对象。近年来随着大数据技术的崛起，数据挖掘技术不断趋于成熟，在现代医疗数据研究领域得到了广泛应用。多标签分类问题最初提出的重要一部分原因就是医学领域数据挖掘的迫切需求[5]。由于各种心血管疾病间复杂的医学关系、发病的先后关系及伴随的多种并发症等难以分析的特性，多标签分类算法不仅能预测单个疾病的发展情况，而且考虑了疾病间的复杂关系，能够识别患者可能患有多种疾病的风险。生物医学领域多标签分类问题普遍存在，心血管疾病因其时间长、隐匿性强、难以完全治愈等问题，在早期不易察觉常被患者忽视，多种并发症导致患者病情越发严重，以至于影响后期治疗，因此在医学基础上利用多标签分类算法准确预测出患者早期可能存在的多种疾病，以达到早治疗早康复的目的。由于疾病间的复杂性，多标签分类技术在医学领域的应用依然存在很大挑战。

1.2 国内外研究现状

1.2.1 机器学习在生物医学领域的发展现状

随着医疗大数据以及计算机辅助医疗诊断技术的不断发展，利用机器学习、数据挖掘方法与疾病相关的特征值来预测疾病变得越来越重要，例如 Pena-Reyes 和 Sipper 等人利用模糊遗传算法预测乳腺癌，计算出超过 96%的准确率[6]，Wang 等人利用人工神经网络模型和基于多层感知器来辨别口腔癌和口腔黏膜纤维瘤，得到了非常好的效果，Al-Ammar Barnes 利用有监督的聚类算法来预测癌症。He-J 和 Gu -H 等人利用多标签算法解决蛋白质亚细胞水平上预测叶绿体蛋白的定位问题[7]。Hofer T 等人在已有的临床慢病数据上使用比较多种多标签分类算法来解决慢病预测问题[8]。W Zhang ,F Liu 提出以 ML-KNN 为基础的新多标签分类算法解决药物副作用预测问题，取得了有效成果[9]。医疗数据包括纯数据、信号、图像、文字等多种模式,其属性类型包括分类

型、数值型或二者混合,数据中可能还包含了大量无用信息，因此,对医疗数据的降噪、剃选等预处理过程会比较复杂,挖掘过程交互性强,且可能需要反复多次[10]。

总之,在医疗领域,多种数据挖掘算法都有了很好的应用,针对特定疾病问题,选择合适的挖掘方法,才能真正挖掘出符合临床实际的、有价值的知识。

1.2.2 多标签分类的研究现状

近年来，随着大数据技术的不断发展，机器学习得到空前关注及应用，为社会各领域做出来很大贡献。多年来经过学者们不断地深入研究，多标签分类问题有了许多显著的解决方案，并得到了很好的应用。根据文献[11]，总体上来说，多标签分类算法是单标签分类算法的扩展，主要分为 PT(problem transformation)和 AA(algorithm adaption)。

1) PT 方法通过将多标签问题转换为一个个单标签分类问题进行处理，常见的有 Binary Relevance(BR)[12]、Classifier Chains (CC)[13]、 Label Powerset (LP)[14]、Hierarchy Of Multi-label learners(HOMER)[15]、Random k-labelsets(RAkEL)[16]。Binary Relevance(BR)算法为每一个标签训练一个的二元分类器，测试时，依次使用每个二元分类器判断测试对象是否属于对应标签。该算法简单直接，但是未考虑标签之间的相关关系。LabelPowerset(LP)方法将标签集合中的每个标签子集进行了二进制编码,转换为了单标签多分类问题，考虑了标签间的相关性，但是随着标签集合规模的不断扩大，标签编码将以指数形式增长，算法的复杂度变大。Random k-labelset(RAKEL)算法解决 LabelPowerset (LP)算法中标签集数量过多的问题，该算法对标签随机分组，以组为限进行训练，从而大大减少了算法训练过程中的标签数量。

2) AA 方法则是扩展已有的单标签分类算法使其能够处理多标签问题。基于单标签分类算法 AdaBoost.MI,Schapire 等人提出了解决多标签分类问题的 AdaBoost.MH[17]算法，该算法使用每个多标签训练数据生成 q (标签数量)个新的单标签训练数据，该算法的主要缺点是增加了训练数据的数量，加重了训练开销。ML-KNN[18]通过改进 KNN 算法，通过统计方法得出每个标签的先验概率，当输入一个未分类数据，对标签集合中的每个标签，分别计算该未分类数据具有该标签的概率，来预测该样本是否属于该标签。此外还有改进 C4.5 算法的多标签决策树，基于支持向量机（support vector machines, SVMs）和神经网络的改进算法等等。

多标签分类中识别一种事物具有的多种特性更加切合现实世界，因此在图像、生物、文本等多领域得到了重视，例如 Zincir-Heywood 等人进行蛋白质功能分类，Li & Ogihara 利用多标签算法分类音乐类别，Boutell 则用于情感语义识别，多标签分类方向已经成为机器学习中的重要分支。

1.3 论文主要研究内容

本文基于各医院、诊所等医疗机构的门诊病历数据，采用多标签分类算法建立心血管疾病预测模型，得到可靠且符合实际意义的预测效果，论文的主要研究工作包含以下几个方面：

1) 对心血管疾病预测模型建立前的心血管疾病数据集的处理方法进行了详细设计和实现。包括原始医疗数据的特性分析、目标标签选取及心血管疾病数据集的提取、心血管疾病数据集的清洗和特征降维。对特定心血管疾病数据集的特性，利用 spearman 统计和逻辑回归算法进行了特征二次选择，获得了有效的影响因子。

2) 针对心血管疾病数据集采样中出现的多标签数据集不均衡性问题，分析了 ML-RUS、ML-SMOTE 等重采样算法带来的大类样本采样过度造成的信息丢失而小类样本过采样造成的信息冗余等不均衡问题，提出了一种多标签双重自适应采样算法 ML-DARS 来调整数据集的不均衡程度，并利用实际数据集进行了实验验证与对比分析，证明了 ML-DARS 算法能够在不改变原有数据总体分布的情况下获得更为均衡的多标签数据集。

3) 根据实际心血管疾病数据集的特性，从样本近邻的局部角度和标签间相关性的全局角度出发，采用基于混合策略的多标签学习框架，提出了 ML-KNN 算法和 RAKEL 算法相结合的多标签分类框架，建立了心血管疾病预测模型，并对模型进行了实验分析验证，结果表明该模型相比于现有的多标签分类算法取得了很好的预测效果。

1.4 论文的组织结构

本文主要工作分为四大部分：一是对心血管疾病的调研，多标签分类问题及解决方法的探讨与分析；二是本文介绍了用于研究的心血管疾病数据从提取开始、利用医学知识和数据预处理技术清洗数据、到特征选择以及最终应用于多标签分类算法等过程；三是深入理解多标签分类中的类别不均衡性问题，并在此基础上进行调研，通过学习已有的解决多标签不均衡性问题的方法，本文提出了一种多标签双重自适应性采样算法 ML-DARS；四是利用现有的多标签分类算法构建基于混合策略的心血管疾病预测模型。具体结构如下：

第一章 主要介绍心血管疾病研究的重要性，多标签分类的研究现状，以及机器学习技术在医学领域的发展情况。

第二章 介绍多标签分类的定义，多种多标签分类的评价指标，现有且常用的多标签分类算法，在此基础上详细探讨了数据不均衡性问题及多标签领域不均衡问题的研究现状，同时描述了现有的评价多标签数据集分布的指标以及不均衡度。

第三章 对心血管疾病预测模型进行了研究，考虑到医学数据复杂性以及专业性强等特点，本章对真实的医疗数据首先做了分析统计，确定了基于多标签的心血管疾病数据集，然后进行了一系列复杂的预处理过程和特征选择过程，为应用于多标签分类算法等全程的预测模型建立奠定了扎实的基础。

第四章 首先针对多标签不均衡性问题进行了分析，探讨了现有的解决多标签不均衡性的采样方法，为加快采样过程且获取更加均衡的数据集，提出了一种将欠采样和过采样相结合的多标签双重自适应随机采样算法，将其应用于心血管疾病数据集和公共多标签数据集，并在此基础上与现有的采样算法进行实验对比分析。

第五章 分析了心血管疾病数据集的自身特性，提出了基于混合策略的多标签分类框架构建心血管疾病预测模型，并对实验结果进行了分析。

第六章 对本文工作进行总结与评价，以及下一步研究方向。

2 多标签分类相关研究

2.1 多标签分类研究概述

机器学习中分类问题是典型的监督学习，通常分为两大步骤，一是训练模型，二是模型预测。训练模型时，首先将训练样本表示成模型可识别的特征向量，然后对特征属性进行特征选择，获得影响分类类别的有效特征，利用分类算法获得一定的泛化误差内尽可能拟合训练数据的分类模型，然后将测试样本表示成与训练样本相同的特征向量，通过分类模型，计算出测试样本相关的类标签。

传统的单标签分类算法将一个样本划分到唯一、特定的一个类别中，例如预测天气是否下雨、判断一个人的职业是教师还是警察等。然而现实世界中，我们通常遇到判断预测一种事物的多种语义，像一幅图片中同时包括海洋、蓝天、白云、树等等，一首歌曲通常被归到快乐、情感等多种类别当中，多标签分类问题面向于给定一个样本，将该样本通过分类算法，归类到相关的类标签中，可能是一个也可能是多个[19]。

类别不均衡性指训练过程中某一类别的样本数目远超过其他类别的样本数目，导致分类模型效果变差甚至无效[20]。现有研究中有了一些成熟的处理单标签分类中类别不均衡性方法，而多标签分类中考虑到标签间关系的复杂性，多标签类别不均衡性较单标签分类更为严重，处理该问题在多标签分类领域越来越被重视。

2.2 多标签分类方法

在单标签分类中，一个样本仅属于一个类，而多标签中一个样本可能属于多个类别，多标签问题可看成单标签分类问题扩展而得到的更加广义和复杂的分类问题。接下来介绍多标签分类的定义、相关的评估方法和已有的分类算法。

2.2.1 多标签分类的形式化描述

为了形式化描述多标签分类问题，设 $\chi = R^d$ 表示 d 维实例特征空间，$\mathrm{L} = \{\lambda_j : j = 1 \ldots M\}$ 为该空间的有限标签集合，即有 M 种可能的标签，多标签训练样本集 $D = \{(x_i, y_i), i = 1 \ldots N\}$，$x_i$ 表示 d 维特征向量，$\mathrm{Y}_i \subseteq L$ 表示对应的第 i 个样本所关联的标签集合。多标签分类的目标是根据训练得到的函数 $h = x \to 2^y$，将测试样本集中的每一个样本 x，获得与其相关的标签集合 y [21]。

在多标签分类问题中，使用分类算法前，首先要衡量该数据集是否具有多标签分类的必要，因为它最终影响多标签分类效果，例如该数据集中的标签向量大多数仅有一个标签有效，那么收到的多标签分类效果就非常微弱。该为了描述多标签数据集的特征，几种非常有用的多标签数据集衡量指标，标签基数见公式(2-1)，表示每个样本平均的有效标签个数，一般来讲该值约大于等于 2，具有较好的多标签数据集特性，对

应的标签密度如公式(2-2)所示。另一种常用的多标签测量方法为标签差异性，衡量样本空间中出现的不同标签组合的数量，公式见(2-3)[22]。

$$Lcard(D)=\frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{2-1}$$

$$LDen(D)=\frac{1}{|y|}\bullet Lcard(D) \tag{2-2}$$

$$LDiv(D)=|\{Y\mid\exists x:(x,Y)\in D\}| \tag{2-3}$$

2.2.2 多标签分类的统计评价方法

多标签分类效果评估方法不同于单标签分类，学术界目前已经有许多成熟的多标签分类评估方法，大体上分为两大类，一种是二元分类评估，另一种是基于标签相关性排序的分类评估[23]。为了定义评估方法，给一样本 x_j，标签预测集合设 Z_j，预测单个标签 λ 对应的排序函数为 $r_j(\lambda)$，与样本最相关的标签排序越靠前，最不相关的标签排序越靠后[24][25]。

1) 二元分类评估

许多基于二元分类评估方法在于计算样本集的真实标签与预测标签的差异性的平均值,常见的 Hamming loss、Subset Accuracy[26]。另一种是对单个标签进行评估，然后取所有标签评估结果的均值，例如 micro-averaged 、macro-averaged[27]。在这里可又分为基于样本的评估和基于标签的评估方法。

(1) 基于样本的评估方法

Hamming loss 的定义如公式(2-4)：

$$Hamming\text{-}Loss=\frac{1}{N}\sum_{i=1}^{N}\frac{Y_i\Delta Z_i}{M} \tag{2-4}$$

Δ表示两个标签集对应标签的差异性，即逻辑上的异或操作，代表相关标签被预测为不相关标签的比例，一定程度上值越小分类效果越好。

Subset Accuracy 的计算公式(2-5)：

$$SubsetAccuracy=\frac{1}{M}\sum_{i=1}^{M}I(Z_i=Y_i) \tag{2-5}$$

上式中 $I(true)=1$， $I(false)=0$，该式严格要求预测标签集精确匹配真实标签集，即预测标签等于真实标签的样本数所占比例，该值越大越好。

Precision, Recall, F1, Accuracy 的定义见公式(2-6)：

$$Precision=\frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i\cap Z_i|}{|Z_i|}\qquad Recall=\frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i\cap Z_i|}{|Y_i|}$$

$$F1=\frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i\cap Z_i|}{|Z_i|+|Y_i|}\qquad Accuracy=\frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i\cap Z_i|}{|Y_i\cup Z_i|} \tag{2-6}$$

(2) 基于标签的评估方法

Micro-averaged 和 macro-averaged 很类似，都是基于在单个标签上的二元评价方法。设二元评估方法为 $B(tp,tn,fp,fn)$，其中 $tp_\lambda, tn_\lambda, fp_\lambda, fn_\lambda$ 代表样本集在单个标签 λ 上预测的真正类，真负类，假正类，假负类。Micro-averaged 和 macro-averaged 的计算方法如公式(2-7)所示：

$$\begin{aligned}B_{macro}&=\frac{1}{M}\sum_{\lambda=1}^{M}B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda)\\ B_{micro}&=B(\sum_{\lambda=1}^{M}tp_\lambda,\sum_{\lambda=1}^{M}fp_\lambda,\sum_{\lambda=1}^{M}tn_\lambda,\sum_{\lambda=1}^{M}fn_\lambda)\end{aligned} \tag{2-7}$$

这些指标越大越好，最优值是 1。

2) 排序评估

One-error 评估方法表示排序最靠前的标签不属于样本的相关标签集合中的一员，对应的样本所占比例如公式(2-8)所示，该值越小越好：

$$1-Error=\frac{1}{N}\sum_{i=1}^{N}\delta(\arg\min_{\lambda\in L}r_i(\lambda)) \tag{2-8}$$

其中

$$\delta(\lambda)=\begin{cases}1 & \text{if } \lambda\notin Y_i\\ 0 & \text{otherwise}\end{cases}$$

Coverage 指标见公式(2-9),指首先取样本的所有相关标签排序的最大深度，然后在所有样本上取平均，得到样本集的平均深度[28]。该值越小说明分类时相关标签排序越靠前，分类效果越好。

$$\text{cov}=\frac{1}{N}\sum_{i=1}^{N}\max_{\lambda\in L}r_i(\lambda)-1 \tag{2-9}$$

Ranking loss 的公式定义如(2-10)：

$$R-Loss=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\bar{Y}_i|}|\{(\lambda_a,\lambda_b):r_i(\lambda_a)>r_i(\lambda_b),(\lambda_a,\lambda_b)\in Y_i\times\bar{Y}_i\}| \tag{2-10}$$

上式中 $\bar{Y}_i$ 是 y_i 的补集,即样本 x_i 的不相关标签集合。对于单个样本而言,其排序损失值就是它的所有相关标签与不相关标签对中,发生排序错误的百分比。因此 Ranking Loss 就是所有样本的平均排序损失值，该值越小越好。

2.2.3 典型的多标签分类方法

随着多标签分类技术的不断发展，许多重要的多标签分类算法被提出。这些算法总体上被分为两大类：基于问题转换的方法（Problem transformation methods）和算法适应性方法（Algorithm adaptation methods）[29][30]。问题转换方法将多标签问题转换成已有的、成熟的单标签分类问题，典型的算法有 Binary Relevance(BR)、classifier chains(CC)、Label Powerset(LP)、Random k-labelsets(RAKEL)、Hierarchy Of Multi-label learners (HOMER)。算法适应性方法则是对已有单标签分类算法进行修改来解决多标签分类问题，例如 Multi-Label k-Nearest Neighbor(ML-KNN)、AdaBoostMH。下面进行详细探讨。

1) 基于问题转换的方法

(1) Binary Relevance 方法

该算法非常流行的转换方法，对 L 中出现每个不同的标签，学习到 M 个二分类器，该二分类器通常从单标签分类器中选取，例如决策树、SVM、朴素贝叶斯等。将原始数据集划分为 M 个包含所有原始样本的数据集 $D_{\lambda_j}, j=1...M$，若原始样本中包含标签 λ_j，D_{λ_j} 中对应该样本标记为正类，否则为负类。当新样本到来时，Binary Relevance 输出被 M 分类器预测输出的类标签的组合。表 2-1 代表原始数据集，转换成 Binary Relevance 算法要求的数据集如表 2-2，将表 2-1 数据集划分为四个不同的二分类数据集。

表 2-1 原始多标签数据集

样本	属性	标签 1	标签 2	标签 3	标签 4
1	X1	1	0	0	1
2	X2	0	0	1	1
3	X3	1	0	0	0
4	X4	0	1	1	1

表 2-2 Binary Relevance 转化后的数据集

样本	标签 1	样本	标签 2	样本	标签 3	样本	标签 4
1	1	1	0	1	0	1	1
2	0	2	0	2	1	2	1
3	1	3	0	3	0	3	0
4	0	4	1	4	1	4	1

Binary Relevance 算法为每个标签独立的建立二分类器，可并行进行，具有简单高效性。然而该算法基于标签间相互独立的前提，忽略了标签间的相关性，丢失了许多的重要信息。同时由于标签集中有多个标签，导致转换后数据集正样本数目严重小于负样本，正负样本比例严重失衡，出现数据不均衡现象。

(2) Label Powerset 方法

Label Powerset 充分利用了标签间的相关性，考虑了多标签中每组可能出现的唯一标签，对标签集进行二进制编码生成新类别，然后用单标签分类中多分类任务来完成。如表 2-3，Label Powerset 中新类别的生成。

表 2-3 Label Powerset 转换后的新类别

样本	属性	标签 1	标签 2	标签 3	标签 4	新类别
1	X1	1	0	0	1	9
2	X2	0	0	1	1	3
3	X3	1	0	0	0	8
4	X4	0	1	1	1	7

对于 M 个标签集合，理论上最多可生成 2^q 种不同类别，随着标签规模的不断增大，Label Powerset 问题生成新类别的规模呈指数级别增长，不仅增加分类的难度，生成新标签时，有的标签对应的样本数目很大，有的的样本数目非常小，导致严重不均衡现象，致使预测效果变差。同时该方法无法学习到未出现的新类别，上表中将 1,2,3 样本作为训练集，4 样本作为测试集，其转换后的新类别 7 未出现在训练集中，分类器不可能将其正确分类。

(3) RAKEL 方法

考虑到 Label Powerset 算法在大规模训练样本和标签上的计算复杂性及预测性能问题，RAKEL 算法对 Label Powerset 算法进行了改进。RAKEL 将原始标签集随机划分为一些小的标签子集，每个标签子集包含 k 个标签，且标签子集之间可能会有重叠现象，为每个标签子集使用 Label Powerset 方法将多标签分类问题转换为单标签中的多分类问题。若 RAKEL 选择 m 个分类器， 对一个新样本，可得到 m 个预测标签组合，预测组合公式如(2-11)。

$$y=\{l_j \left| \mu(x,l_j)/\tau(x,l_j)>0.5, 1\le j\le c\right.\} \tag{2-11}$$

$u(x,l_j)$ 表示标签 l_j 在 k 个标签子集中实际出现的频率，$\tau(x,l_j)$ 表示标签 l_j 在所有分类器中可能出现的最大频率，当标签 l_j 出现的频率超过它所能获得的最大频率的一半时，就认为该标签是相关的。

2) 算法适应性方法 ML-KNN 算法

ML-KNN 的基本思想是采用 k 近邻技术处理多标签分类问题，利用最大化后验概率规则(MAP) 推理出待预测样本的标签信息。一未知样本 x，记 $N(x)$ 表示样本集 D 中 k 近邻样本集。一般来讲，样本间相似度使用欧氏距离来度量。对于标签 j, ML-KNN 计算样本 x 的标签 y_j 出现在近邻中的次数，公式如(2-12)：

$$C_j=\sum_{(x^*,\ Y^*)\in N(x)}\ (y_j\in Y^*) \tag{2-12}$$

H_j 表示样本 x 中出现标签 y_j 事件，$P(H_j|C_j)$ 表示样本 x 的近邻中存在 C_j 个标签

y_j时，H_j出现的后验概率，相对的H_j未出现的后验概率为$P(\neg H_j|C_j)$，根据 MAP 规则， 预测标签集合由$P(H_j|C_j)$是否大于$P(\neg H_j|C_j)$决定，如公式(2-13)所示：

$$Y=\{y_j \mid P(H_j|C_j)/P(\neg H_j|C_j) \geq 1, 1\leq j \leq m\} \tag{2-13}$$

根据贝叶斯理论，可得公式(2-14)：

$$\frac{P(H_j|C_j)}{P(\neg H_j|C_j)}=\frac{P(H_j)\bullet P(C_j|H_j)}{P(\neg H_j)\bullet P(C_j|\neg H_j)} \tag{2-14}$$

$P(H_j)$ $P(\neg H_j)$表示H_j出现的先验概率，可通过计算每个标签出现在训练样本集的频次来估计。后验概率$P(C_j|H_j)$利用似然估计计算，标签y_j出现在训练样本集中的次数和 k 个邻居中有C_j个标签y_j来决定。$P(C_j|\neg H_j)$类似，即标签y_j没有出现在训练样本集中的次数和 k 个邻居中有C_j个标签y_j来决定。

ML-KNN 继承了懒惰学习和朴素贝叶斯的优点，即决策边界可以自适应地调整，由于每个类标号是基于先验概率进行估计的，所以该算法对类别不均衡性程度不敏感，不足之处在于为每个待预测样本计算到全局已知样本的距离，计算量大[31]。

2.3 标签不均衡性处理

2.3.1 单标签分类中类别不均衡性问题

在现实中有很多类别不均衡问题，它是常见的，并且也是合理的，符合人们期望的。如在欺诈交易识别中，属于欺诈交易的应该是很少部分，即绝大部分交易是正常的，只有极少部分的交易属于欺诈交易。又如，在客户流失的数据集中，绝大部分的客户是会继续享受其服务，只有极少数部分的客户不会再继续享受其服务。在监督学习领域，因为分配给每个类标签样本数量存在差异，我们经常遇到数据集不均衡现象，为降低全局误分率，分类器偏向于大类样本，损失了小类样本，分类器会大大地因为数据不平衡性而无法满足分类要求，因此在构建分类模型之前，需要对分类不均衡性问题进行处理[32]。传统的单标签分类中对不均衡性问题进行了深入研究，如数据重采样，即在数据预处理阶段，通过删除部分大类样本或者增加小类样本来均衡数据集的分布。重采样技术独立于特定的分类算法，实践证明了其有效性。此外还有算法适应性，代价敏感性分类，它们则依赖于特定的分类算法。算法适应性方法是通过修正现有的分类算法处理数据不均衡问题，代价敏感性分类则结合算法适应性，在分类过程中采用代价敏感策略，误分类小类别样本的代价要大于误分类大类别样本的代价，算法更倾向于小类别样本[33]。

2.3.2 多标签类别不均衡性

单标签分类问题中一个样本仅属于一个标签，而多标签分类中大多数样本同时属于多个标签，这些标签数目通常在几十个到几百个之间，因此多标签分类中的类别不均衡性更为严重。尽管许多多标签数据集中出现大规模的标签集合，但是每个样本通

常仅属于该标签集合的一小部分标签。图 2-1 展示了常见的 7 种多标签公共数据集中每个标签的样本比例，很容易看出大多数数据集中存在 2 到 4 个出现频率很高的标签，剩余的标签对应的样本数目则比较少，也就是说大多数标签被少于 5%的样本所表示[34]。

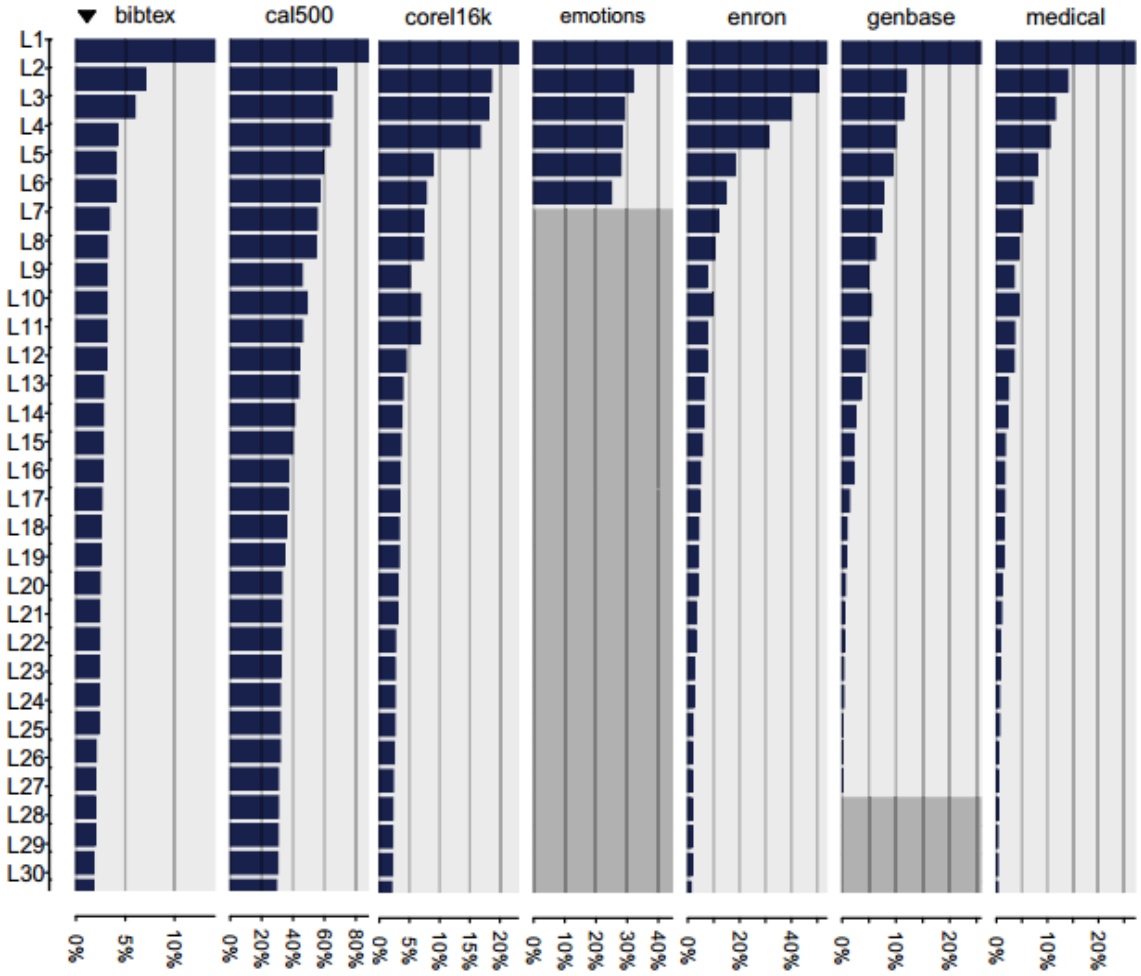


图 2-1 7 种公共数据集的多标签分布情况

目前度量多标签数据集不均衡性的常用指标有两种方法，IRLbl 度量数据集中每个标签不均衡度，定义如公式(2-15)，利用样本集中出现最多的类标签的样本数目与当前标签的样本数目之比来表示该标签的不均衡率。MeanIR 是对所有标签 IRLbl 值的加权平均，公式如(2-16)。

$$IRLbl(l)=\frac{\arg\max_{l'=L1}^{L|L|}(\sum_{i=1}^{N}h(l',Y_i))}{\sum_{i=1}^{N}h(l,Y_i)},\quad h(l,Y_i)=\begin{cases}1 & l\in Y_i\\ 0 & l\notin Y_i\end{cases} \tag{2-15}$$

$$meanIR=\frac{1}{|L|}\sum_{l=L1}^{L|L|}IRLbl(l) \tag{2-16}$$

一般说，meanIR 越大意味着数据集的不均衡程度越大，IRLbl(l)<meanIR 认为标签l为大类标签，IRLbl(l)≥meanIR 则认为l属于小类标签。多标签数据集中存在多个大类别，也存在多个小类别，所以多标签数据集和传统单分类数据集的不均衡性具有本质区别。多标签数据集中，经常出现小类标签和大类标签共同出现在同一样本中，这增加了多标签不均衡问题处理难度。

2.3.3 多标签类别不均衡性问题的处理方法

跟解决单标签分类中不均衡性问题类似，多标签类别不均衡性方法也主要有三大研究方向：算法适应性、集成方法、重采样技术[35]。下面分别介绍：

1) 算法适应性

该方法经常基于多标签分类算法为每个标签赋予不同权重，缓解不均衡性问题。如在文献[7]中人类蛋白质的定位预测问题面临高度不平衡的细粒度问题，解决方法是基于非参数概率模型，结合协方差矩阵获得标签相关性和每个标签相关联的加权系数来修复不平衡的问题。MIMLRBF[36]算法是基于径向基神经网络（RBFN）的多实例多标签细粒度的分类算法。MIMLRBF 对 RBFN 进行优化，来解决多标签分类问题及多标签数据集的不平衡问题。MIMLRBF 中通过调整连接隐层与输出层的权重来适应每个标签的单个偏置。Min－Max-Modular network 算法将分类任务划分成几个更小的任务，不同策略确保更小任务面临的数据不均衡性低于原始任务，这些子任务使用 SVM 算法处理[37]。算法适应性方法依赖于特定算法，而且它们应用于特定领域。

2) 集成方法

基于分类器集成技术在多标签学习领域已经展示了它的优势，例如 RAKEL、ECC、HOMER 算法都获取了很好的性能，同样该方法常常用于解决不均衡问题。文献[38]利用 RAKEL、ECC、MLKNN 等算法为基础构建一个混合集成框架，缓解不均衡问题。对于传统分类问题中不均衡解决方法，集成 Binary Relevance 分类器的方法解决多标签中不均衡问题，该算法名为 BR-IRUS， 思想是在每个标签上训练多个分类器，每训练一次使用所有小类样本和部分大类样本。这些方法的主要弱点是大量分类器的训练降低效率。

3) 重采样技术

重采样技术独立于多标签分类算法，基于数据预处理技术创建更为均衡的数据集，是最为广泛使用的数据均衡方法，现有的多标签重采样技术几乎都是受传统单标签学习中重采样方法影响，即欠采样、过采样、小样本合成技术。

目前多标签分类领域处理多标签不均衡性问题的采样方法被提出，最早的是欠采样 LP-RUS 和过采样 LP-ROS 算法，基于 Label Powerset 转换多标签数据集的方法，将 Label Powerset 转换后的每个标签集作为一个类别定义，LP-RUS 随机删除大类别样本，LP-ROS 随机复制小类样本增加小类样本的数目，该算法简单有效，但由于该算法中最大可生成 $2^{|L|}$ 种不同新类别，增加了算法复杂度。此外该算法不能解决这类极端问题：一个样本集中有 502 个样本，对应了 502 种不同的标签集合，也就是说样本集中的所

有标签集都是不同的，虽然 MeanIR 指标显示数据不均衡很高，Label Powerset 不能处理该问题，因为生成的新类别数目是相同的。为了解决 LP-RUS 和 LP-ROS 算法存在的问题，ML-RUS 和 ML-ROS 算法对其进行了修改，单独评估原始数据集中每个标签的不均衡度，而非 Label Powerset 中生成的新类别，降低了复杂度。鉴于 ML-ROS 算法直接复制小类标签的样本方法虽然在一定程度上均衡了数据分布，但是也造成了大量重复样本导致的过拟合问题，为解决该问题，MLSMOTE 算法利用传统单标签分类中小样本合成技术 SMOTE，对训练集中小类样本通过插值产生额外的新样本，丰富小类样本同时避免过拟合问题。

2.4 本章小结

本章首先介绍了多标签分类的定义、评价指标以及相关方法，之后就类别不均衡问题进行了探讨，先描述了类别不均衡性问题，之后对传统单标签分类中的类别不均衡性现象及处理方法做了介绍，最后引出了多标签情况下的类别不均衡性探讨，解释了多标签类别不均衡的情况，类别不均衡性评价指标，和已有的多标签类别不均衡性处理方法。通过本章对多标签分类中一些知识点的介绍，为之后多标签分类算法应用于心血管疾病预测模型奠定了基础。

3 心血管疾病预测模型研究

由于大量的原始医疗数据表现形式多样化且存在很多噪声点，不能直接用于心血管疾病预测模型。为此，本章首先给出了心血管疾病预测模型的目标以及步骤，然后研究了心血管疾病数据的特点，详细分析并实现了数据加载、确定并抽取研究对象、数据预处理、特征选择的心血管疾病预测模型前期处理过程。

3.1 心血管疾病预测模型建立的目标和步骤

本文以面向区域医疗和公共卫生的健康大数据处理分析研究及示范应用项目为背景，利用大数据和机器学习技术，按照医学规则，从 3 亿条门诊记录以及高血压、糖尿病患者医疗数据记录的电子病例、疾控、体检数据中提取与心血管疾病相关的信息，并进行处理、分析，发现影响心血管疾病的潜在因子，在此基础上应用多标签分类算法对心血管疾病进行风险预测，以辅助医疗诊断。

医疗数据的处理以其复杂性、不完整性、专业性强等多种特性，一直是各项研究中首要也是必不可少的重要环节。由于门诊数据和临床数据是经人工录入系统的，所以录入过程中难免出现一些失误或者意外情况，例如患者就医时对于某些医学参数不清楚导致该医学指标缺失，医生在录入数据过程中也可能出现失误，患者可以在多个医疗机构就诊而导致该患者的医疗数据产生冗余、不一致性。此外，这些原始医学数据指标之间存在属性多样性，例如年龄为数值型属性、患者症状描述则为文本类属性。

由于支撑本文的项目中原始医疗数据包括了所有患者的数据，其数据量大，数据复杂，而并非仅是心血管疾病数据，为了使拿到的原始医疗数据更好的应用于心血管疾病预测模型，本文对心血管疾病预测模型按照以下步骤进行了研究：

1) 数据采集：首先需要分析确定原始医疗数据中的心血管疾病人群，再通过统计这些数据以及分布，确定要用于多标签分类预测的心血管疾病即预测目标，然后从原始医疗数据中提取预测目标对应的心血管疾病数据集。本文的原始医疗数据包括门诊记录表、个人信息表、高血压表、糖尿病表中的信息，其中门诊记录表达到了 3 亿记录，其它表数据量均在 10 万以上，为了获得全部的患者数据，需要将这 4 个表关联，数据量非常庞大，由于数据量大且不规则等特性，传统 Oracle 数据库处理起来不仅麻烦、费时且需要更高的硬件成本，为此，利用 spark 平台的大数据处理优势，进行数据采集过程。

2) 数据预处理：由于心血管疾病数据集中存在数据缺失量大、数据异常等问题，如特征字段中心率的缺失值达到 70%，随访评价结果的缺失值达到了 90%等，需要对数据进行异常值检测、缺失值处理、属性转换等预处理操作。此外，一位患者根据随访时间的不同对应了多条患病记录，有的患者对应的患病记录多达几百条，而本文研究的心血管疾病预测模型需要的心血管疾病数据集针对每位患者一条记录，所以需要

根据医学规则进行特殊处理。

3) 特征选择：根据前面步骤一系列处理，除了删除机构编码、科室编码、日期等难以分析或无意义的特征外，依然存在对心血管疾病数据集影响很小的特征，为了预测结果的准确性、可靠性，对心血管疾病数据集的特征进行分析，由于现有多标签分类中特征选择方法不成熟，通过统计学方法和分类算法，提取影响心血管疾病发展的有效因子。

4) 心血管疾病预测模型：由于心血管疾病数据分布的稀疏性、不均衡性，首先利用多标签统计评价数据集的方法对目标标签集进行统计分析，评估该标签集的多标签特性，在此基础上对心血管疾病数据集极不均衡问题进行处理。将多个多标签分类算法应用于心血管疾病数据集，通过多标签分类评价指标，分析多标签分类效果的差异性及不足，并在此基础上提出一种基于混合策略的多标签分类框架提高心血管疾病预测的性能。

总之，心血管疾病预测模型需要从原始医疗数据进行分析统计出发，选取并确定要研究的心血管疾病数据集，提取所需心血管疾病数据并进行清洗预处理过程，处理完后还需要对特征分析从而选择有效的心血管疾病影响因子，最后才能应用于预测建模并进行结果分析。

3.2 心血管疾病数据采集

3.2.1 数据采集分析

本文用于研究的心血管疾病数据集针对于高血压、糖尿病人群，是一个城市所有医疗机构所拥有的全部高血压、糖尿病患者的医疗数据，其中包括了所有患者个人信息、患病症状、各种健康指标等信息。表信息描述见表 3-1，由于一个患者标识对应多条就诊记录，所以记录总数不小于患者数目。

表 3-1 原始医疗数据来源说明

含义	记录总数（万）	患者数目（万）	相关数据
门诊记录表	30000	1500	身高、体重、BMI、诊断描述等
个人信息表	1800	1500	性别、年龄、文化程度、婚姻状况、学历等
高血压表	44	12	门诊机构、医院编码、高血压级别、呼吸频率、收缩压、舒张压、心率、各种并发症等
糖尿病表	12	6	呼吸频率、收缩压、舒张压、心率、各种并发症等

为了从表 3-1 中获得心血管疾病的人群分布，需要将门诊记录表、个人信息表、高血压表、糖尿病表进行表关联操作，得到了包括所有特征在内详细的原始医疗数据表，其中表维度达到 703，表 3-2 列出来具有代表性的大部分特征，其中特征 CONFIRM_DATE 之后每个 ICD 码代表一种病症，为 0-1 离散型特征，且每个 ICD 码

后对应该病症的确诊时间，其中有的 ICD 码最终会作为心血管疾病数据集中预测目标的特征出现，有的将代表预测目标，这里限于篇幅仅列出前三个 ICD 码对应的病症及确诊日期列。ICD 码和病症的对应关系见附录。

表 3-2 原始医疗数据表的特征

特征列 1	特征列 2	特征列 3	特征列 4	特征列 5	特征列 6
MPI_PERSON_ID(患者 ID)	BIT37	E15_X00	I28_801	I66	I79_2
ORG_CODE(机构编码)	BIT38	E16_000	I28_900	I67_0	I84
NATION_CODE(民族代号)	BIT39	E16_100	I30	I67_1	I88
CAREER_CODE(职业代号)	BIT40	E16_101	I31	I67_2	I89
EDU_CODE(教育水平)	BIT41	E16_103	I32	I67_4	I95_000
ABO_CODE(血型)	BIT42	E16_107	I33	I67_5	I95_100
CITIZEN_CODE(居住城市)	BIT43	E16_108	I34	I67_700	I95_101
MARITAL_CODE(婚姻状况)	BIT44	E16_200	I35	I67_800	I95_200
AGE(年龄)	BIT45	E16_801	I37	I67_801	I95_900
SEX(性别)	BIT46	E16_803	I38	I67_803	I97_001
HEART_RATE_TIMES(心率)	BIT47	G43	I39	I67_805	I97_100
GLU(空腹血糖)	BIT48	G44	I40	I67_900	I97_801
BMI (身高体重比)	BIT49	G45	I41	I70_0	I97_806
FOLLOWUPDATE(随访日期)	E10_301	G47	I42	I70_1	I98_0
SYMPTOMNAME (症状描述)	E10_302	G80	I43	I70_2	I98_2
RISK_STRATIFY(风险等级)	E10_401	G81	I44	I70_8	I98_3
SBP(收缩压)	E10_403	G90	I45	I70_9	I99_X00
DBP(舒张压)	E10_501	G91	I47_0	I71_1	N18
CONFIRM_DATE (确诊日期)	E10_503	H34	I47_1	I71_2	R02
BIT01	E10_601	H35_001	I47_2	I71_3	R03
BIT01_DATE	E10_900	H35_002	I47_9	I71_9	R04
BIT03	E10_901	H35_003	I48_X00	I72_0	R09
BIT03_DATE	E11_000	H35_004	I48_X01	I72_2	R10
BIT05	E11_002	H35_008	I48_X02	I72_3	R11
BIT05_DATE	E11_100	H35_011	I48_X03	I72_4	R12
BIT07	E11_101	H35_100	I49_001	I72_8	R13
BIT08	E11_300	I11_900	I49_002	I72_9	R45
BIT09	E11_301	I15	I49_100	I73_000	R46
BIT10	E11_400	I20_002	I49_200	I73_001	R50
BIT11	E11_401	I20_006	I49_300	I73_100	R51
BIT12	E11_403	I20_801	I49_400	I73_802	R52
BIT13	E11_404	I20_802	I49_500	I73_803	R53
BIT14	E11_500	I20_803	I49_800	I73_804	R54
BIT15	E11_501	I20_900	I49_900	I73_900	R55
BIT16	E11_601	I24_000	I50_906	I73_901	R56
BIT17	E11_700	I24_001	I51_302	I73_902	R57
BIT18	E11_900	I25_000	I51_303	I73_903	R58
BIT20	E12_100	I25_100	I51_304	I74_0	R59
BIT21	E13_102	I25_101	I51_400	I74_2	R60
BIT23	E13_200	I25_102	I51_402	I74_3	R61
BIT24	E13_300	I25_103	I51_403	I74_8	R62
BIT25	E13_600	I25_300	I51_500	I74_9	R63
BIT27	E13_900	I25_400	I51_501	I77_0	R64
BIT28	E13_901	I25_500	I51_600	I77_1	R65
BIT29	E13_902	I25_600	I51_700	I77_2	R68
BIT30	E13_903	I25_801	I51_701	I77_5	R69
BIT31	E13_904	I25_802	I51_702	I77_6	R70
BIT32	E13_905	I25_900	I51_703	I77_8	R71
BIT33	E13_904	I25_901	I51_802	I77_9	R72
BIT34	E13_905	I27_0	I51_900	I78_1	R73
BIT35	E13_907	I27_8	I51_901	I78_8	R74
BIT36	E14_900	I27_9	I65	I78_9	

由于每位患者患病情况不同，所以高血压表、糖尿病表中存在大量疾病信息，这

些疾病种类繁多，处理起来十分麻烦，但是医学上对这些疾病都有统一的归类，鉴于本文定位于心血管病人群，所以这里仅对心血管疾病数据进行归类，我们首先根据医学规则进行 ICD 编码将所获得数据特征中属于同一大类的疾病进行合并，部分信息见表 3-3，例如下表中肺心病这一类包括了医学上的原发性肺动脉高压、慢性肺源性心脏病等多种小类疾病组成。

表 3-3 心血管疾病归类

疾病代号 ICD	疾病名称	所属类别
I25_000	被描述为动脉硬化性心血管病	稳定性冠心病
I25_100	动脉硬化性心脏病	
I25_101	冠状动脉狭窄	
I25_102	冠状动脉粥样硬化	
I27_0	原发性肺动脉高压	肺心病
I27_8	特指肺源性心脏病	
I27_9	慢性肺源性心脏病	
I11_000	高血压心脏病伴心力衰竭	心力衰竭
I13_000	高血压性心脏病肾脏病伴心力衰竭	
I50_000	充血性心力衰竭	
I50_100	左心衰竭	
I12_000	高血压性肾衰竭	严重肾病人群
I12_900	高血压性肾病	
I12_902	肾动脉硬化	
I12_904	肾小动脉硬化症	
I13_000	高血压性心脏病肾脏病伴心力衰竭	
I13_100	高血压性心脏病肾脏病伴肾功能衰竭	
I13_900	高血压性心脏病及肾脏病	
N17	急性肾衰竭	
N19	未特指的肾衰竭	
E11_200	2 型糖尿病性肾病	
I20_000	不稳定性心绞痛	急性心梗/急性冠脉综合症
I20_005	心肌梗死后心绞痛	
I20_101	变异型心绞痛	
I20_102	冠状动脉痉挛	
I21	急性心肌梗死	
I24_801	急性冠状动脉供血不足	
I24_900	急性心肌缺血	
I24_901	急性冠脉综合征	
I21	急性心肌梗死	
I23_6	心房、心耳和心室的血栓	
I24_801	急性冠状动脉供血不足	
I24_900	急性心肌缺血	
I46_000	心脏停搏复苏成功	心源性猝死
I46_100	心源性猝死	
I46_900	心脏骤停	
I46_901	呼吸心跳骤停	
I11_900	高血压性心脏病	心脏功能性病变
I50_906	心肌损害	
I51_700	心脏扩大	
I51_701	左室肥大	
I51_702	右室肥大	

I51_703	左房扩大
I51_802	全心炎

3.2.2 预测目标的选取

本文的研究定位是特定的心血管疾病类人群，而原始医疗数据表包括了心血管类、消化类、内分泌类等各种疾病人群，所以需要从原始医疗数据表中统计提取所需要的心血管类人群，经过对原始医疗数据表中的心血管疾病合并归类后，得到的心血管疾病数据统计结果如表 3-4，这些疾病的人群可重复，因为一位患者可能患有多种疾病。

根据表 3-4 的统计结果，选择特定的心血管疾病作为研究目标，有的心血管疾病对应的人数太少不考虑加入多标签集合，例如肺栓塞仅 21 人，心源性猝死仅 128 人数据量太少不具备预测能力，而严重脑神经疾病、意外死亡及后遗症该类疾病不属于心血管疾病的范畴，考虑到严重的心血管疾病危及人们的生命，所以研究预测严重心血管疾病具有重要的研究价值，最终选择用于多标签分类的预测目标标签如表 3-5 所示。

表 3-4 心血管疾病初步数据统计

疾病名称	患者人数
严重脑神经疾病、意外死亡及后遗症	60050
心衰人群	1079
心源性猝死	128
脑卒中	7809
急性心梗	1030
肺栓塞	21
肾衰竭	1168
心肌缺血	1137
心脏功能病变	3062
冠心病	16761
高血压	123019
糖尿病	65798

表 3-5 多标签分类中预测目标

心血管疾病	患者人数
脑卒中	7809
心衰	1079
心梗	1030
肾衰	1168
心肌缺血	1137
心脏功能病变	3062
冠心病	16761
高血压	123019
糖尿病	65798

3.2.3 数据采集实现

1) 数据采集平台

门诊记录表、个人信息表、高血压表、糖尿病表中由于数据量大且极不规则等特性，通过结构化组织存储在 Oracle 数据库中，如表 3-1 所示，数据量都在万级以上，特别是门诊记录表有 3 亿条记录，不论是表关联还是数据统计，数据量更大，在计算量、计算速度上难度很大。

为此，利用现在成熟的大数据存储处理平台 hdfs、spark 技术进行提取处理心血管疾病数据。Hdfs 技术作为分布式大数据存储架构，具有高可靠性。Spark 是适用于大数据的高可靠性、高性能分布式并行计算框架，支持内存技术、多迭代批量处理、流计算和图计算等多种范式。本次数据处理以 hdfs 为后台存储原始数据的地方，spark 则作为计算平台，从 hdfs 中读取数据并进行相关计算，数据采集处理的平台架构[39]如图 3-1 所示。

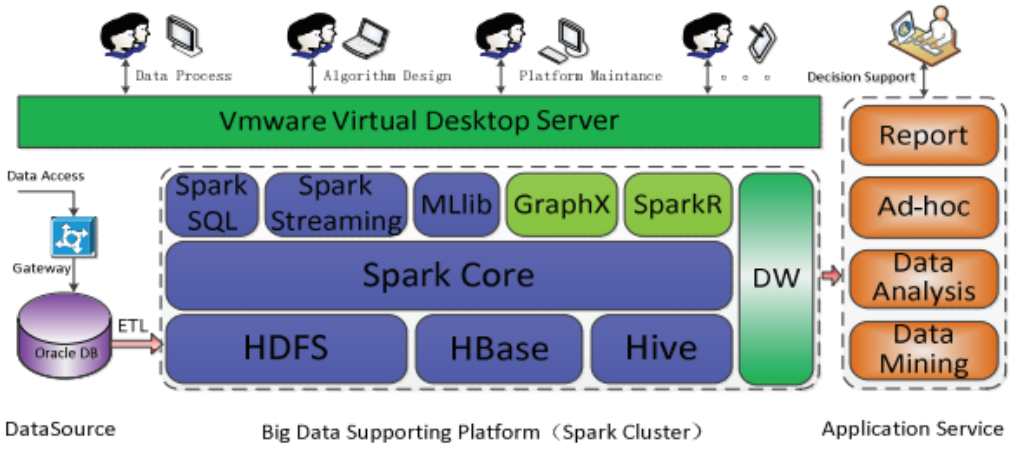


图 3-1　数据采集处理平台

本次分布式处理平台中，spark 集群配有 13 台机器，一个作为主节点，4 核 16GB 内存，500GB 磁盘，其他作为计算节点，8 核 12GB 内存，500GB 磁盘。

本次数据采集过程将原始数据从 Oracle 导出并加载到 hdfs 中存储，利用医学知识、spark 平台，提取例如脑卒中、心衰等属于严重心血管疾病的数据，再通过统计这些数据以及分布，确定要研究的对象，即用于多标签分类的各个目标标签。

2)　数据采集过程

首先关联门诊记录表、个人信息表、高血压表、糖尿病表，初步获得患者的所有患病信息，其代码如下，然后根据医学定义读取图 3-2 各文件的 ICD 码统计各心血管疾病数据量结果如图 3-3，通过分析，决定了最终心血管疾病预测的目标标签如表 3-5。

```
//关联各表，获得所有人群的相关信息
public Dataset<Row> select(Dataset<Row>... tables) {
        int i;
        String sql = "";
        for (i = 0; i < tables.length; i++)
            tables[i].createOrReplaceTempView("table_"+(i+1));
        if (2 == i) {
            sql = "SELECT *FROM    table_1 h, table_2 thr WHERE
```

```
                thr.MPI_PERSON_ID=h.MPI_PERSON_ID";
                 Dataset<Row> results = getSc().sql(sql);
                 return results;
            } else return null;
        }
```

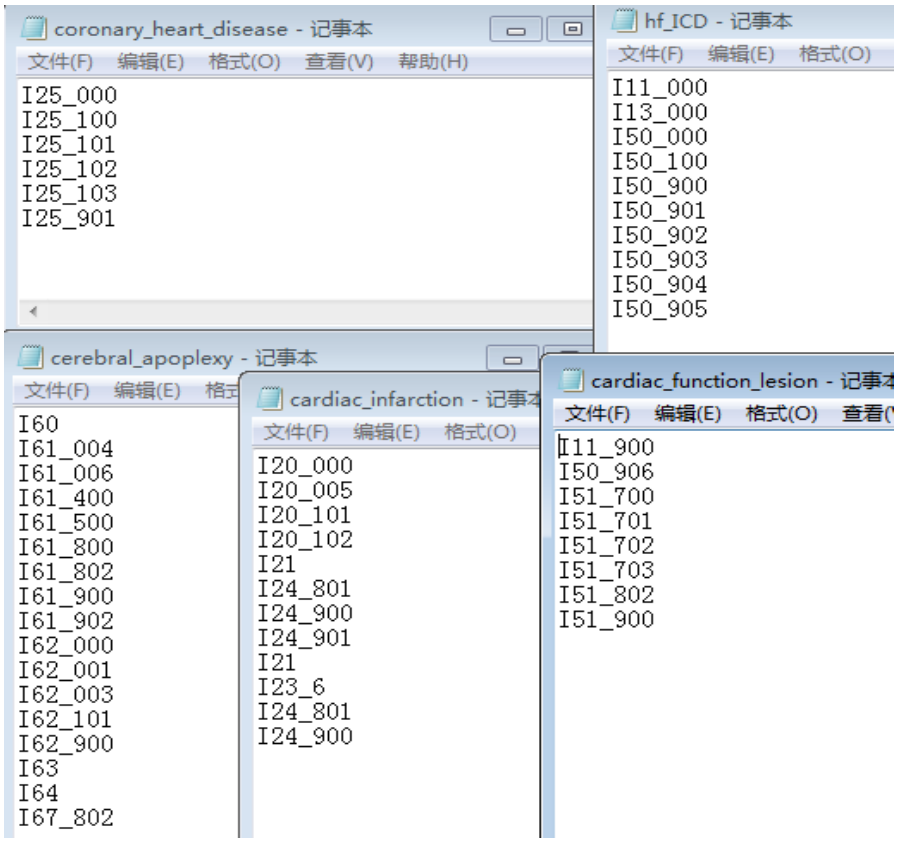


图 3-2 各心血管疾病的 ICD 码

```
cerebral_apoplexy人群处理完毕! 数据量为 7809
hf_ICD人群处理完毕! 数据量为 1079
pulmonary heart disease人群处理完毕 数据量为 87
sudden cardiac death人群处理完毕! 数据量为 128
cardiac_infarction人群处理完毕! 数据量为 757
pulmonary embolism人群处理完毕! 数据量为 21
nephropathy人群处理完毕! 数据量为 1168
ischemia_myocardial人群处理完毕! 数据量为 1137
cardiac_function_lesion人群处理完毕! 数据量为 3062
coronary_heart_disease人群处理完毕! 数据量为 16761
gaoxueya人群处理完毕! 数据量为 123019
tangniao人群处理完毕! 数据量为 65798
```

图 3-3 统计各心血管疾病数据量

然后合并心血管疾病预测的目标标签对应的各个子 ICD 码作为一大类疾病的结果值，认为只要子 ICD 中出现该病就认为患者患有该标签代表的心血管疾病。关联表过

程和提取 ICD 码对应的目标疾病人群的过程代码如下：

```
//核心代码，提取指定的 ICD 人群
public JavaRDD<Row> filterData(JavaPairRDD<String, String> dataset, final Boolean flag) {
        System.out.println("filterdata 总的 mpi_person_id 数:" + dataset.count());
        JavaRDD<Row> icd1 = changeToRow(dataset.filter(new Function<Tuple2<String,
        String>, Boolean>() {
            public Boolean call(Tuple2<String, String> value) throws Exception {
                //获取每行的 ICd 值,过滤指定的 icd
                if (value._2 != null) {
                    if (isIncluding(value._2.trim().split(",")) == flag) {
                        if (flag) count1.add(1);
                        else count2.add(1);
                        return true;
                    } else return false;
                } else {
                    if (!flag) count2.add(1);
                    return !flag;
                }
            }
        }));
        return icd1;
    }
```

3.3 心血管疾病数据预处理

3.3.1 数据清洗

本文中所用到的心血管疾病数据中，由于医疗记录存在数据缺失量大、数据异常等问题，需要对数据进行异常值检测处理、缺失值处理等操作，此外鉴于患者门诊数据中一位患者可对应多个门诊记录，不符合本文中多标签数据集的要求即一位患者对应一条记录，所以医学数据处理起来比较复杂。

心血管疾病数据中每条患病记录中的医学特征分为两大类，一是一般身体指标如年龄、性别、血糖等，这类特征数目比较少，对应的属性类型既有数值型也有文本型比较混杂，另一种则是患者对应的患病种类即 ICD 码特征如一个患者可能患有心律失常、心肌缺血等多种并发症，这类特征每个疾病 ICD 编码单独作为一列，属于 0-1 属性，即患有该病为 1，没有则为 0，因为一位患者一般来说患的疾病数很少大约 2-3 种，而所有患者患病情况千差万别，所以这类特征很稀疏。很明显，这里主要处理的特征集中于一般身体指标。

根据医学规则，首先根据数据集特性，删除对模型无意义特征例如机构编码、档案流水号、终止管理日期、随访流水号、并发症的取值全为 0 等。删除冗余性特征，例如出生日期和年龄只需保留年龄即可，许多特征既有该指标编码又有该指标名称，只需保留指标编码就可以了。大量的缺失值使得机器学习的效果不理想，无法学习到好的模型，因此需要初步统计每一个特征的非缺失值数目，删掉缺失值在 70%以上的

特征，例如婚姻状况、血型等。

3.3.2 异常值处理

对医学特征中出现不符合常识的异常数据进行检测、处理，鉴于医学数据的严谨性、标准化特性，使用人工检测表方法规定生物学特征的取值范围，不在范围内的被视为异常值，为了尽可能保留研究对象的数据量，除了日期数据异常进行删除操作之外，其他特征异常值视为缺失值处理。人工检测表部分信息如下表 3-6 所示。

表 3-6　检测表部分取值

特征名称	正常取值范围	特征类型	处理方式
性别	1 男　2 女	离散型	零填充作为缺失值处理
年龄	4 到 115 之间	数值型	零填充作为缺失值处理
SBP 收缩压	80 到 200 之间	数值型	零填充作为缺失值处理
DBP 舒张压	40 到 160 之间	数值型	零填充作为缺失值处理
GLU 空腹血糖	2 到 14 之间	数值型	零填充作为缺失值处理
BMI（身高体重比）	10 到 50 之间	数值型	零填充作为缺失值处理

由于心血管疾病确诊日期唯一但随访日期(复查日期)不唯一，所以一个患者标识对应多条记录，提取随访日期在心血管疾病确诊日期之前且最早的记录，因为我们需要获取待预测疾病可能存在的先验条件，即每个患者对应一条记录。

3.3.3 缺失值处理

首先缺失值在 70%以上的特征删除，然后对其它特征缺失值进行填充，有的特征属性为文本属性例如症状编码记录患者身体情况，经分析发现主要集中于特定的几个词，所以该属性可看做离散属性，进行维度扩充，对于每一个值，出现该特征的记为 1，否则记为 0，这样就不存在缺失值的情况。由于一位患者对应多条随访记录，所以使用该患者相关的记录数据处理缺失值相对更具有可靠性，因此数值型属性以一位患者多条记录中其他非缺失值的平均值填充，离散型属性则以一位患者多条记录中其他非缺失值出现次数最多者进行填充。因为存在一位患者的多条记录都为空值的情况，所以对剩余的缺失值处理过程根据每个特征的特性分为两部分，一是不变性特征例如体重、身高、性别等均值或者众数填充，二是易变性特征例如心率、空腹血糖等零值填充。

由于各个特征间属性值因为实际意义不同存在较大差异，需要对数据进行归一化处理，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。数据归一化方法常有 z-score 归一化，将特征处理后，数据均值为 0，标准差为 1，其公式(3-1)所示，此外还有离差标准化，将原始数据线性变换后使结果落到[0,1]区间转换函数如公式(3-2)所示。鉴于离差标准化方法有一个缺陷就是当有新数据加入时，可能导致 max 和 min 的变化，需要重新定义，所以对此医疗数据采用 Z-score 方法进行数据归一化处理。

$$X^* = \frac{X - \mu}{\sigma} \tag{3-1}$$

μ 为所有样本数据的均值，σ 为所有样本数据的标准差。

$$x^* = \frac{x - \min}{\max - \min} \tag{3-2}$$

其中 max 为样本数据的最大值，min 为样本数据的最小值。

3.3.4 数据预处理的实现

本次数据预处理实现在 3.2 节基础上，依然利用 spark 计算平台来完成。该过程的 ETL 流程代码如下，根据得到的心血管疾病数据集，为获取有效数据，首先删除明显的无效特征，然后根据人工检测表处理异常值，最后处理缺失值。

```
#提取实验数据
./sub_etl.sh ${clinical_0817} ${yjy_reg2016} ${icdfile} ${icdname} ${number} ${min}  ${titles} ${output}
number=4
#删除无用特征
spark-submit --py-files my_tools.py ICD_data.py $output$icdname/3/part* $output$icdname/$number
#异常值检测处理
spark-submit --py-files my_tools.py ExceptionValue.py $output$icdname/$number/part* $output$icdname/`expr $number + 1`
number=`expr $number + 1
#多条记录取一条及缺失值填充
spark-submit --py-files my_tools.py diff_hf_dataset_e.py $output$icdname/$number/part* $icdfile BIT01  $output$icdname/`expr $number + 1 `"_experment"
```

异常值处理过程中离散型和连续型特征选择零值填充，而异常日期数据无法预测进而删除该记录，对应代码如下：

```
//离散属性异常值检测并处理
def find_Exception_disperse(data1, data2):
    d1=data1
    flag='0'
    for j in data2:
        if j.strip()==d1.strip():
            flag='1'
            break
    if flag=='0':
        d1=flag
  return d1
//连续属性异常值检测并处理：寻找并异常值,异常值视为缺失值处理
def find_Exception_continue(data1, data2):
    d1=data1
```

```
    if len(re.findall("\d+",d1.strip()))<=0:#返回匹配的字符串列表
        d1='0'
    else:
     if not ((float(d1.strip())>=float(data2[0].strip())) and
        float(d1.strip())<=float(data2[1].strip()))) :
            d1='0'
    return d1
#日期列中的异常值删除操作
def delete_date(row):
   flag=dateCompare1(row["DATE_OF_BIRTH"],row["CONFIRM_DATE"],"%Y-%m-%d %H:%
M:%S.0")
    if flag:
flag=dateCompare1(row["CONFIRM_DATE"],row["FOLLOWUP_DATE"],"%Y-%m-%d %H:%M
:%S.0")
    return flag
```

最后考虑到缺失值需要用患者的多条记录进行处理，所以将取患者唯一一条记录和缺失值一起处理，相关代码如下，final_data 函数在取患者所患心血管疾病时确诊日期在随访日期之后且最早的随访记录时，考虑到该记录中存在的缺失值，所以同时又根据其他记录处理该缺失值，即调用了 sum_null 函数，average 函数则是对连续属性均值处理，离散属性众数处理。

```
#多条记录取一条,规则为最早的随访记录
def final_data(value1,value2):
    value=value1
    bdate1=mindate(value1)
    bdate2=mindate(value2)
     comp=dateCompare(value1[column2-1],"%Y-%m-%d  %H:%M:%S.0",value2[column2-1],"%
     Y-%m-%d %H:%M:%S.0")#最早随访日期
    if comp>0:
        value=value2
    if comp==0:
       vf1=dateCompare(bdate1,"%Y-%m-%d",value1[column2-1],"%Y-%m-%d %H:%M:%S.0")
        vf2=dateCompare(bdate2,"%Y-%m-%d",value2[column2-1],"%Y-%m-%d %H:%M:%S.0")
         if (vf1<=0) and(vf2>0):#确诊日期大于随访日期
             value=value2
         if (vf1>0) and (vf2<=0):
             value=value1
         #对出现空值的字段处理填充
    if value==value1:
          return sum_null(value,value2)
    else:
          return sum_null(value,value1)
#该患者多条记录中其他的非缺失值填充被选中记录的缺失值
def sum_null(val1,val2):
    for i in range(0,len(val1)):
          if header[i+1].strip()==icd:
```

```
                    break
        if (header[i+1].find("DATE")<0)and((i+1)!=column3)and((i+1)!=column4)and
          ((i+1)!=column5)and((i+1)!=column6):#保存当前非空值总和和非空值的数目，并用'|'分割
            if val1[i]==0 and isinstance(val2[i],float)and (val2[i]!=0):#当前值为空，新值 val2 不为空
                val1[i]=str(val2[i])+'|'+str(1)
                continue
            if val1[i]==0 and isinstance(val2[i],basestring)and(val2[i].index('|')>=0):
                val1[i]=val2[i]
                continue
            if isinstance(val1[i],basestring)and(val1[i].index('|')>=0)and
                isinstance(val2[i],float)and(val2[i]!=0):
                val1[i]=str(float(val1[i].split('|')[0])+float(val2[i]))+'|'+str(float(val1[i].split('|')[1])+1)
                continue
            if isinstance(val1[i],basestring)and(val1[i].index('|')>=0)and
                isinstance(val2[i],basestring)and (val2[i].index('|')>=0):
                val1[i]=str(float(val1[i].split('|')[0])+float(val2[i].split('|')[0]))+'|'+str(float(val1[i].split('|
                ')[1])+float(val2[i].split('|')[1]))
    return val1
    #连续属性均值处理，离散属性众数处理。
    def average(val):
        for i in range(0,len(val)):
            if header[i+1].strip()==icd:
                    break
        if (header[i+1].find("DATE")<0)and((i+1)!=column3)and((i+1)!=column4)and
          ((i+1)!=column5)and((i+1)!=column6):
            if isType(i)=="continue" and isinstance(val[i],basestring)and val[i].find('|')>=0:#对连续性
空值数据取均值
                    val[i]=float(val[i].split('|')[0])/float(val[i].split('|')[1])
            if isType(i)=="disperse" and isinstance(val[i],basestring)and val[i].find('|')>=0:#对离散性
空值数据取众数
                    val[i]=int(val[i].split('|')[0])
        return val
```

3.4 心血管疾病数据特征选择

3.4.1 特征数据提取

在数据预处理阶段，去掉了一部分很明显像用户 ID 这样的无意义特征和删除了缺失值大的特征，但是依然存在某些对标签区分度不大的特征。目前心血管疾病的特征保留情况见表 3-7，共 297 个特征，特征对应说明见附录，可以看到，大部分特征为心血管疾病可能的相关病症，而在医学领域，通常一种疾病仅和几种或十几种病症密切相关，在进行多标签分类学习时，过多的特征可能会导致训练效果下降。为了预测结果的准确性、可靠性，需要对心血管疾病数据进行进一步降维处理，提取影响预测目标发展的有效因子。

表 3-7 心血管疾病数据选择前的特征

特征列 1	特征列 2	特征列 3	特征列 4	特征列 5	特征列 6
AGE(年龄)	BIT44	F00	I39	I67_900	I95_000
SEX(性别)	BIT45	F01	I40	I69_0	I95_100
GLU(空腹血糖)	BIT46	F02_3	I41	I69_1	I95_101
BMI (身高体重比)	BIT47	F02_8	I42	I69_3	I95_200
SBP(收缩压)	BIT48	F05	I43	I69_4	I95_900
DBP(舒张压)	BIT49	F06	I44	I69_8	I97_001
MARITALCODE(婚姻)	BIT50	G20_X00	I45	I70_0	I97_100
BIT01	BIT51	G20_X02	I46_000	I70_1	I97_801
BIT02	E10_301	G20_X03	I46_100	I70_2	I97_806
BIT03	E10_302	G21_200	I46_900	I70_8	I98_0
BIT04	E10_401	G21_900	I46_901	I70_9	I98_2
BIT05	E10_403	G30	I47_0	I71_1	I98_3
BIT06	E10_501	G43	I47_1	I71_2	I99_X00
BIT07	E10_503	G44	I47_2	I71_3	N18
BIT08	E10_601	G45	I47_9	I71_9	R02
BIT09	E10_901	G47	I48_X00	I72_0	R03
BIT10	E11_000	G80	I48_X01	I72_2	R04
BIT11	E11_002	G81	I48_X02	I72_3	R09
BIT12	E11_100	G90	I48_X03	I72_4	R10
BIT13	E11_101	G91	I49_001	I72_8	R11
BIT14	E11_300	H34	I49_002	I72_9	R12
BIT15	E11_301	H35_001	I49_100	I73_000	R13
BIT16	E11_400	H35_002	I49_200	I73_001	R45
BIT17	E11_401	H35_003	I49_300	I73_100	R46
BIT18	E11_403	H35_004	I49_400	I73_802	R50
BIT19	E11_404	H35_008	I49_500	I73_803	R51
BIT20	E11_500	H35_011	I49_800	I73_804	R52
BIT21	E11_501	H35_100	I49_900	I73_900	R53
BIT22	E11_601	I24_000	I51_302	I73_901	R54
BIT23	E11_700	I24_001	I51_303	I73_902	R55
BIT24	E12_100	I24_101	I51_304	I73_903	R56
BIT25	E13_102	I25_200	I51_400	I74_0	R57
BIT26	E13_200	I25_300	I51_402	I74_2	R58
BIT27	E13_300	I25_400	I51_403	I74_3	R59
BIT28	E13_600	I25_802	I51_500	I74_8	R60
BIT29	E13_901	I26_0	I51_501	I74_9	R61
BIT30	E13_902	I26_9	I51_600	I77_0	R62
BIT31	E13_903	I27_0	I51_901	I77_1	R63
BIT32	E13_904	I27_8	I65	I77_2	R64
BIT33	E13_905	I27_9	I66	I77_5	R65
BIT34	E15_X00	I28_801	I67_0	I77_6	R68
BIT35	E16_000	I28_900	I67_1	I77_8	R69
BIT36	E16_100	I30	I67_2	I77_9	R70
BIT37	E16_101	I31	I67_4	I78_1	R71
BIT38	E16_103	I32	I67_5	I78_8	R72
BIT39	E16_107	I33	I67_700	I78_9	R73
BIT40	E16_108	I34	I67_800	I79_2	R74
BIT41	E16_200	I35	I67_801	I84	
BIT42	E16_801	I37	I67_803	I88	
BIT43	E16_803	I38	I67_805	I89	

此外，考虑到医学领域数据的严谨性、保守性、复杂性，本文选择的心血管疾病

间可能存在非常密切的相互作用，同时多标签分类领域的特征选择方法尚未成熟，所以本文采用统计检验和分类算法相结合，提出了基于多标签分类的心血管疾病数据特征二次选择策略。

3.4.2 特征数据的二次选择

由于心血管疾病的特征往往是造成该心血管疾病的重要影响因素，所以医学上对心血管疾病影响因子的分析非常重视，为了分析获取每个目标标签对应的重要特征，同时又考虑到多标签分类的特性，本文对特征进行了二次处理策略，该策略将已预处理好的基于心血管疾病的多标签数据集使用多标签分类领域常用的转化方法 Binary Relevance(BR)来对标签集合中的每个标签单独处理。处理过程分两步，如下：

1) 首先对研究对象进行特征相关性分析，去除共线性强的特征，选择与标签相关性较强的特征，常用的统计学方法有 pearson 相关系数、spearman 秩相关系数，两种方法都是度量两个随机变量的相关程度，pearson 相关系数用协方差除以两个变量的标准差得到的，介于-1 到 1 之间，当两个变量线性关系增强时，当一个变量增大，另一个变量也增大时，表明它们之间是正相关的，相关系数大于 0；如果一个变量增大，另一个变量却减小，表明它们之间是负相关的，相关系数小于 0；如果相关系数等于 0，表明它们之间不存在线性相关关系，该方法限于两变量呈线性相关关系，如果是曲线相关可能不准确，此外两变量须符合正态分布。Spearman 相关系数对原始变量不做要求，适用范围广，通常被认为是排列后的变量间的 pearson 线性相关系数。本文考虑到心血管疾病特征的复杂性及特征存在的稀疏性，选用 spearman 作为检验手段，此过程选择出与每种心血管疾病相关性强的特征，保留 $p<0.05$ 的特征。

2) 经过特征相关性选择，获得了与特定单标签疾病相关的特征，去除了共线性特征，一方面为了验证第一次特征选择的有效性，将选出的有效特征作为特定标签的属性，应用逻辑斯特回归分类算法分别评估单个心血管疾病的分类效果，分析每个特征的权重，进一步得到权重大的特征，另一方面，由于第一次处理后每个标签的特征维度平均达到了 90 多个，所有标签特征组合起来特征维度依然很高，逻辑斯特回归是机器学习领域常用的有效分类算法，其模型公式如(3-3)。数据统一分布在 0-1 之间即 $Y\in(0,1)$,可以看出模型中权重 ω 越大，表示该特征对预测结果的贡献度越大，当 $|\omega|$ 越小，说明该特征无法对预测结果产生影响，从中剔除，所以这里用逻辑斯特回归模型产生的特征权重进行进一步特征选择来获得更为干净的有效因子集合。根据特征权重由高到低排序，选择排序靠前的特征作为该疾病的度量指标，权重越大对目标疾病的贡献程度越大。

$$Y=\frac{1}{1+e^{-\omega^{T}x}} \tag{3-3}$$

根据第一次特征选择的结果，对心血管疾病数据集进行了第二次特征选择，获得每个标签对应特征的权重，根据权重对特征排序，最后获得每个特征在所有目标标签上的平均排名，根据实际情况合并所有预测目标的特征。

3.4.3 特征选择结果

首先，经过第一次特征选择，获得每个预测目标对应特征的显著性水平，以心衰为例部分结果如表 3-8，例如该表展示了心律失常的相关性，spearmanr_pval 值小于 0.05，95%的置信度认为两个随机变量相关，表明心律失常与心衰间存在强相关性，将其保留进入第二次特征选择过程，而由于慢性缺血性脑血管病和心衰 spearmanr_pval 值大于 0.05，无法证明两者间存在一定联系，故而将其去除，不进入下一轮特征选择过程。

表 3-8 心衰 Spearman 相关性分析

特征	特征名	spearmanr_pval
I67_2	大脑动脉粥样硬化	2.21536E-12
BIT16	代谢紊乱	9.32435E-58
BIT38	肺水肿	2.22117E-32
BIT33	肺炎	4.70524E-20
I49_900	心律失常	3.36162E-29
BIT46	胸痛、呼吸异常	6.92032E-27
AGE	年龄	2.99704E-18
BIT32	感冒引起的症状	0.0826336673
I67_805	慢性缺血性脑血管病	0.1247875084
BIT40	多种相关的皮肤病	1.451383E-09

经过第一次基于统计检验方法的特征选择过程后，过滤掉不相关或者相关性不明确的心血管疾病特征，保留了有确定相关关系的特征，但是该过程仅针对单个心血管疾病预测目标，要将有效特征集合应用于多标签分类预测过程中，进行了基于逻辑回归算法的第二次特征选择过程，该过程获得每个预测目标标签的特征权重。同样以心衰为例，部分结果如表 3-9，年龄、心率等特征权重绝对值较大有利于预测结果，而皮肤病类、代谢综合征等权重过小对预测结果影响因素较小。

表 3-9 逻辑回归模型的特征权重

特征	说明	模型权重
AGE	年龄	0.06502474
BMI	体重身高比	-0.09750807
BIT33	肺炎	0.299
DBP	舒张压	0.10771338
SBP	收缩压	0.084
I49_900	心律失常	0.179
BIT46	胸痛、呼吸异常	0.287
BIT40	多种相关的皮肤病	-0.00428
G90	自主神经系统疾患	0.013
E16_803	代谢综合征	0.00508

最后，根据第二次特征选择结果，得到了每个预测目标标签对应的特征权重，特征权重越大，表明该特征对预测目标标签越重要，因此，为了取得多标签分类过程中有效的特征集合，首先去掉单个标签中不相关特征或者权重绝对值小于 0.001 的特征，对每个预测目标标签的剩余特征按照权重大小降序排列，然后获得每个特征在所有目标标签上的平均排名，部分结果见表 3-10，均值是该特征排序的平均值，最后的综合

排名是根据均值而获得的特征在所有标签的排名。最后保留了效果最好的 178 个特征集合，见表 3-11，表中 178 个特征按排名结果列出。特征编码对应的特征说明见附录。

表 3-10 特征在每个标签的排名

目标标签	I47_1	年龄	BMI	I48_X02	SBP	R45	E16_200	E16_803	BIT16	G20_X00
脑卒中	12	14	26	23	56	153	67	143	74	19
心衰	33	6	75	77	58	7	26	61	62	52
心梗	3	8	54	77	32	121	16	14	26	39
肾衰	4	27	42	16	39	157	41	32	51	125
心肌缺血	36	13	9	76	27	17	28	12	47	14
心脏功能病变	10	70	27	24	31	8	103	26	55	126
冠心病	14	24	25	57	54	15	32	58	45	50
高血压	66	33	37	44	34	14	35	59	91	52
糖尿病	3	23	25	19	153	2	147	99	63	45
均值	20.11	24.22	35.56	45.89	53.78	54.89	55	56	57.11	58
综合排名	1	2	3	4	5	6	7	8	9	10

表 3-11 用于心血管疾病预测的特征集合

特征列 1↓	特征列 2↓	特征列 3↓	特征列 4↓	特征列 5↓
I47_1	BIT08	E11_501	BIT07	E11_301
AGE	BIT51	BIT03	R04	I49_800
BMI	BIT24	E11_401	GLU	E13_200
I48_X02	BIT17	R73	E15_X00	F01
SBP	R51	BIT06	I67_900	BIT44
R45	R57	E16_103	R12	BIT18
E16_200	G20_X02	I48_X01	I69_1	I67_1
E16_803	G81	I67_4	I70_2	H34
BIT16	BIT09	R65	BIT14	E11_101
G20_X00	R72	G90	R69	I67_805
BIT15	BIT34	R58	R10	I35
R54	BIT39	E12_100	I84	I27_0
BIT20	BIT46	E11_700	F02_3	I25_802
BIT22	I72_8	BIT48	R09	H35_008
R11	BIT36	I77_6	R68	I72_9
I25_200	I27_9	SEX	R63	I48_X03
BIT23	G43	G47	I49_400	H35_004
I65	BIT40	BIT28	I73_900	BIT12
I70_9	BIT30	BIT11	G80	I51_400
E16_107	BIT05	BIT10	I95_900	I71_2
BIT27	BIT04	R50	BIT49	I43
I51_600	I49_100	BIT29	BIT25	I79_2
I70_8	BIT41	I89	I69_8	R59
G30	I69_3	R56	R61	I24_000
I69_4	I66	R52	I49_300	I67_5
I49_900	G44	I45	E11_300	DBP
F00	R60	BIT50	I49_500	I88
I67_803	BIT32	I77_1	BIT35	I47_2
I42	BIT47	I31	G20_X03	R13
BIT33	N18	I26_9	BIT21	R62
I67_2	BIT42	BIT01	I74_3	E13_300

E10_401	BIT38	BIT37	E11_601	I51_402
BIT13	BIT45	I40	E10_302	H35_001
G45	BIT43	R55	H35_002	I25_300
I46_901	E13_102	R53	BIT31	
I38	I70_0	R74	BIT02	

3.5 本章小结

本章首先对心血管疾病数据预测模型的过程进行了研究并给出了预测目标及步骤，然后分析设计了数据集本身的特性、选取规则及确定研究对象，对数据集做了必要的数据清洗，最后为了更好的应用多标签分类算法，本章对数据特征进行了降维，根据心血管疾病数据集的特性，利用 spearman 统计和逻辑回归算法相结合进行了特征二次选择策略，获得有效的影响因子，此外还对各部分工作进行了实现。本章通过对心血管疾病数据集进行一系列的预处理操作，为后续工作打下了坚实的基础。

4 多标签双重自适应随机采样算法

本章针对心血管疾病数据集中出现的多标签不均衡性问题，分析了 ML-RUS、ML-SMOTE 等重采样算法带来的大类样本采样过度造成的信息丢失而小类样本过采样造成的信息冗余等不均衡问题，提出了一种多标签双重自适应采样算法 ML-DARS。

4.1 多标签数据集不均衡性问题分析

本文中心血管疾病预测模型是基于医院历史门诊数据中的诊断结果和患者自身的体检指标作为模型预测的特征集合。

由于一位病人一般并发症数目不超过 3 种，且一种疾病一般仅与少数其他疾病存在明显的相互关系，因而心血管疾病数据集中特征存在稀疏性，数据分布存在不均衡等问题，使得分类算法偏向于大类样本，导致预测效果缺乏可靠性。

为此，本文在建立预测模型之前，需要解决基于多标签的心血管疾病数据集存在的不均衡性问题。多标签分类问题中不仅涉及到多数类标签，而且还要考虑类标签间的相互关系，所以多标签分类比传统单标签分类要面临更为复杂的挑战。多标签分类过程中经常遇到类标签分布不均衡性现象，这些不均衡现象要比传统分类中的更为严重，加剧了多标签分类任务的难度。虽然多标签学习领域的不均衡性数据集到处可见，但是处理多标签不均衡性的方法近些年才被重视和研究。不论是传统的单标签分类还是多标签分类，重采样算法以独立于具体分类算法且成熟有效而得到了广泛研究和应用。

目前，解决多标签不均衡性问题的重采样方法是欠采样和过采样两种方法，这两种方法分开独立的处理数据不均衡性问题，实际上多标签数据集中大类样本和小类样本之间存在很大差距，如果仅对数据集进行欠采样，为了和小类样本达到均衡，大类样本数目变少，而小类样本过少很难学习到有效模型，导致最终的模型欠拟合。如果仅对数据集过采样，小类样本为了缩小和大类样本的差距，对小类样本进行复制或插值而产生的新样本数目超过了原始的小类样本数目，可能造成模型过拟合，有甚者产生的噪声数据对模型造成严重影响。

通过研究分析，本章提出了一种多标签双重自适应随机采样算法 ML-DARS(Multi-label double adaptive random sampling algorithm)，并通过与其他已有的数据均衡性算法比较，证明了新算法的有效性。该算法考虑整个数据集的分布，从两个层面解决多标签不均衡性问题即欠采样过程和过采样过程相结合，降低数据集中大类样本的同时也适当的增加了小类样本的数目，防止了大样本信息大量丢失和小样本数据冗余问题。通过将该算法应用于心血管疾病数据集和公共数据集，ML-DARS 算法产生的数据集符合实际的数据分布的特点，很好地解决了多标签数据不均衡性问题，

提升了分类效果。接下来通过探讨已有的多标签重采样算法，在此基础上重点介绍ML-DARS 算法，以及对应的实验结果和评价。

4.2 多标签双重自适应采样算法 ML-DARS

4.2.1 ML-DARS 算法概述

目前多标签分类领域的数据不均衡问题通常解决方法是基于 LP 方法和 ML 方法的重采样技术。其中 ML 方法在 LP 方法的基础上进行了修正，更加有效的解决了多标签不均衡性问题。基于 ML 方法对标签集合中的每个标签评估不均衡性，如公式(2-12)、(2-13)，IRLbl≥MeanIR, 将标签归到小类标签集合中，IRLbl<MeanIR 则归入大类标签集合中，在此基础上采用不同策略对数据集重采样。基于 ML 方法具体包括 ML-RUS 欠采样算法、ML-ROS 过采样算法、ML-SMOTE 小样本合成算法。

ML-RUS 算法主要思想是随机删除不在小类标签集合中的样本。包含小类标签的样本从要删除的样本集中排除出去，每次从删除样本集中随机选择一样本进行删除，该样本仅含有大类标签，之后检测大类标签样本集中每个标签的 IRLbl 是否不小于初始的 MeanIR，若是说明该标签对应的样本集变小了，将该标签从大类标签集合中删除，重复上述过程直到达到规定的阈值，完成欠采样过程，该算法在一定程度上缓解了多标签数据不均衡问题。

类似于 ML-RUS 算法，ML-ROS 算法根据公式(2-12)、(2-13)，首先将原始标签集划分为小类标签集和大类标签集，不同之处在于该算法对每个小类标签，找到小类标签对应的实例集合，遍历并在其中随机挑选一个实例进行复制，并将复制的实例添加到数据集中，当标签的 IRLbl 达到 MeanIR 说明该标签对应的样本数目接近于平均样本样本数目，停止随机挑选的过程，并将该标签从小类标签集合中除去，重复以上过程直到满足指定的阈值，过采样过程完成。这种方法若直接复制样本的数目过多会造成模型的过拟合现象。

由于 ML-ROS 算法直接复制样本而导致的过拟合现象，ML-SMOTE 算法受单标签分类中小样本合成算法 SMOTE 的启发，根据出现在小类标签集合中的每个标签对应的实例集合来合成新的样本。这个新合成的实例以相关样本和对应的近邻为基础，形成了自己的特征集合以及对应的标签集合。

已有的解决多标签不均衡性的方法要么是欠采样，要么是过采样，这种单方面采样来均衡数据集的方法虽然能够在一定程度上解决数据不均衡问题，但是采样过度可能导致数据集发生偏移，大类样本变成小类样本造成的大量信息丢失，小类样本由于产生的新样本过多而造成的过拟合问题。另一方面，目前的多标签采样算法 LP 和 ML 通过人工设定采样率来决定要删除的大样本数目，或者增加新的小类样本的数目，这种方法使用起来，首先通过多次试探了解多标签数据集的分布情况，而多标签数据集本身涉及到一个样本拥有多个类标签的这种复杂特性增加了分析难度，粗略的设置采

样率存在随机性大且未能考虑到整体数据集的分布情况，造成最终的数据集分布存在很大不确定性，使模型分类效果受到很大影响。

为了对以上问题进行处理，本节提出了多标签双重自适应重采样算法 ML-DARS 算法。该算法主要考虑到多标签数据集内部很大的差异性，将欠采样和过采样过程相结合，同时最小化人工干预对数据集造成的极大不确定性。该算法解决问题主要分为以下步骤：

1) 小类标签、大类标签的划分，即选择哪些标签作为小类标签，哪类标签需要划分到大类标签集合。

2) 采样算法选择，本算法鉴于欠采样和过采样结合的思想，首先需要确定有效的欠采样算法、过采样算法为整个过程提供更好的基础。

3) 如何确定采样过程结束，或者是如何确定数据集分布达到了较为均衡的情况。上述已讨论过，已有算法通过人为设定采样率规定需要减少或增加的样本数目存在极大的不合理性，而本算法充分利用数据集自身的内在分布信息来确定数据集的均衡性，进而作为采样结束的标准，获得更为均衡且合理的数据集。

4) 如何集成欠采样和过采样两种算法来获得更为均衡的数据集，即具体的 ML-DARS 算法设计。

4.2.2 标签集合的划分

由于 ML-DARS 算法将欠采样过程和过采样相结合，算法开始之前需要确定哪些标签属于小类标签，哪些标签属于大类标签，然后根据划分好的大类标签集合和小类标签集合进行后续的欠采样和过采样过程。ML-DARS 算法同样根据标签间的不均衡程度对标签集合进行划分。根据第二章中的公式(2-12)、(2-13)当标签 l 的不均衡度 $IRLbl \geq MeanIR$，将标签 l 归为小类标签，否则将标签 l 归为大类标签，同时一个样本可能既包含大类标签又有小类标签，对此，以尽可能保留所有小类标签的原始信息的思想，将该样本归为小类标签所在范畴，即保留该标签。随着采样过程的不断进行，需要不断更新数据集对应的不均衡度 $IRLbl$ 、 $MeanIR$ ，来评估数据集当前的状态，进而决定是否要继续采样。

4.2.3 采样算法选取

正如上文介绍，目前最为流行的采样算法基于 LP 和基于 ML 的方法，LP 方法带来标签集转化后产生的多分类标签规模大且数据量极不均衡等问题，基于 ML 方法的采样算法更为合理的均衡了数据集的分布。其中 ML-RUS 算法受 LP-RUS 算法思想的启发，对每个标签首先进行评估其不均衡度并合理的划分大小类标签，然后删除采样率所规定的样本数目，这些样本对应的标签不含有所有小类标签，既保证了小类标签对应的样本信息不会丢失，同时又删除了大类样本带来的冗余信息。

ML-SMOTE 算法既考虑了 ML-ROS 算法的优势，即复制小类样本增加小类标签的样本数目，又解决了 ML-ROS 算法存在的重复样本导致的模型过拟合问题，对小类样

本插值产生新样本，既有效解决了不均衡问题，又获得了合理且较好的模型分类效果，实际上， ML-SMOTE 普遍代替 ML-ROS 算法用来解决多标签不均衡性问题。

基于上述分析，ML-DARS 算法受 ML-RUS 算法思想和 ML-SMOTE 算法思想的启发，ML-DARS 算法欠采样过程仅删除出现在大类标签集合中的样本，而过采样过程中通常会出现样本对应的标签集合中同时出现小类标签和大类标签共存的情况，为了防止 ML-SMOTE 算法中对出现在小类标签集合中的样本进行合成的同时也造成大类标签过多的情况，ML-DARS 算法对出现在小类标签集合中的样本进行合成时，仅对小类标签投票，忽略小类标签集合外的标签投票情况。

ML_DARS 算法既解决大样本信息冗余问题，又解决了小样本缺乏信息表示问题，极大均衡了多标签数据集的分布。

4.2.4 数据集均衡性标准

上文提到已有的多标签重采样算法，人工设定采样率来决定采样是否结束，存在的问题是采样完成后的数据集可能并未达到均衡程度，虽然可以通过不断调整采样率来接近于数据的均衡程度，但随机性大且没有明确规定数据集是否达到均衡性标准。为了解决人工采样率存在的问题以及能够经过采样后数据集自适应性的达到均衡程度，ML-DARS 算法进行了修正。

受标签间不均衡度的度量指标的启发，一般来说，标签不均衡度越是接近于 MeanIR，说明标签集整体的分布越均衡，当 IRLbl(y_m)=MeanIR 时，标签 y_m 对应的样本数为整个数据集的平均样本数 MeanInstances，度量指标如公式(4-1)所示,max 为多标签数据集中最大标签对应的样本数，即最大样本数。

$$MeanInstances = \frac{max}{MeanIR} \tag{4-1}$$

很容易得出，多标签数据集经过划分后，大类标签集中每个标签对应的样本数目大于平均样本数 MeanInstances，小类标签集中的每个标签对应的样本数则小于平均样本数 MeanInstances。 在采样过程中，为防止大类样本采样过度变成小类样本，小类样本由于样本数目大量增加而变成大类样本，造成数据集分布的偏离，ML-DARS 利用 MeanInstances 保持原始数据集的分布，即大类样本欠采样后始终大于或等于 MeanInstances，小类标签过采样后始终小于 MeanInstances，ML-DARS 算法就是不断缩小大类样本、小类样本与 MeanInstances 的差距的过程，从而接近于更为均衡的多标签数据集。

此外，MeanIR 衡量多标签数据集整体的不均衡度，而最大类标签的不均衡度为 1，随着采样过程的继续，若多标签数据集越来越均衡，那么数据集平均不均衡度 MeanIR 呈不断减小的趋势，也就是说 MeanIR 越小，数据集越均衡，其值不小于 1。为了获得较为均衡的数据集，ML-DARS 算法每过一定采样间隔，更新 MeanIR，和各标签不均衡度 IRLbl, 为了防止采样过度，需提前停止采样过程，需要对 MeanIR 设置容许范围 threshold，即公式如(4-2)，当大类标签的不均衡度大于 MeanIR-threshold， 该标签对应

的样本数目达到均衡标准，不再对其欠采样，同理当小类标签的不均衡度小于等于MeanIR+threshold，不再对其过采样。直到所有标签对应的样本均达到均衡，采样过程结束。容许能力 threshold 需要手动设置，一般而言 threshold=5 最为合适。

$$IRLbl > MeanIR - threshold$$
$$IRLbl \le MeanIR + threshold \tag{4-2}$$

4.2.5 ML-DARS 算法设计

ML-DARS 算法伪代码见表 4-1，首先需要将原始的多标签数据集根据标签间的不均衡性度量指标划分为小类标签集合、大类标签集合，同时获得原始标签集中度量均衡性的平均样本数，然后算法开始了欠采样过程和过采样过程相结合的采样过程，具体是取原始多标签数据集中的一个样本，对该样本的标签集合进行分析：如果该标签集中存在小类标签集合中一个或多个标签，则认为该样本属于小类样本，进行过采样，其伪代码见表 4-2，利用新样本合成思想，第一步获取该样本中存在的每个小类标签的近邻，第二步根据每个小类标签以及它的近邻进行插值和标签投票的方式形成新的样本，注意仅对小类标签投票，第三步将新样本加入多标签数据集中；欠采样过程中，如果该标签集中的标签仅出现在大类标签集合中，则删除该样本；此外，随着采样过程的继续，有的标签既不存在于小类标签集合也不存在于大类标签集合，这类标签对应的样本数目一定程度上达到了使数据集均衡的目的。对该样本分析完后，由于多标签数据集的分布发生了变化，所以需要重新评估小类标签集合和大类标签集合。这里分为二步走策略，第一步，判断小类标签集合中的每个标签对应的样本数目是否超过平均样本数 MeanInstances，若超过，从小类标签集合中删掉对应标签，同理删除大类标签集合中小于等于平均样本数的标签。这一步主要是防止数据采样过度而偏离原始的数据分布，第二步，更新小类标签集合和大类标签集合中每个标签对应的不均衡度 IRLbl 以及新的整个数据集的平均不均衡度 MeanIR，从小类标签集合和大类标签集合中删除不均衡度 IRLbl 在平均不均衡度的容许范围内的标签，继续循环，直到小类标签集合和大类标签集合为空为止，均衡了数据集的分布。

表 4-1 ML-DARS 算法伪代码

算法：ML-DARS
Input: 训练数据集 D, 近邻数量 k
Output: 采样后的新数据集 Y
(1) **for** each label **in** L **do**
(2) IRLbllabel =calculateIRLbl(D,label)
(3) **If** IRLbllabel < meanIR **then**
(4) Label 加入 maxBag 集合中
(5) **else** label 加入 minBag 中
(6) **End for**

```
(7)  meanInstances =  最大样本数除以 MeanIR
(8)  while (!minBag.isEmpty() or !maxBag.isEmpty() ) and |D|>0
(9)      instance = getInstance(D)
(10)     If instance  中出现了 minBag 中的小类标签  then
(11)       ML-UpgradeSMOTE(D,instance, minBag) //新样本合成
(12)     End if
(13)     If instance  中除了大类标签外不存在其他标签  then
(14)       Delete instance
(15)     End if
(16)     更新采样后的数据集中 IRLbl，  MeanIR
(17)     更新 minBag, maxBag 中的标签
(18)     For each maxlabel in maxBag do
(19)       If maxlabel  的样本数目小于等于 meanInstances    then
(20)             Remove    maxlabel    from maxBag
(21)       End if
(22)     End for
(23)     类似(18)~(22)  步，将大于 meanInstances  的标签从 minBag 中删除
(24)     For each minlabel in minBag do
(25)       If    IRLblminlabel <= meanIR+threshold    then
(26)         Remove minlabel from minBag
(27)       End if
(28)     End for
(29)     类似（24）~(28)步，  将 IRLbl 大于 meanIR-threshold 的标签从 maxBag
中删除
(30) End while
```

ML-UpgradeSMOTE 算法主要为以下步骤：

(1)　根据小类标签样本即种子样本，找到该样本中每个小类标签分别对应的样本集合。

(2)　对于该样本中的每个小类标签，从对应样本集合中寻找距离该样本最近的 k 个实例样本。

(3)　根据选择的 k 个近邻实例，新样本的特征值利用小样本合成算法来生成，从邻居中任选一实例，对于数值型属性，种子样本和选取的实例的特征值之间随机生成一个值，对于离散型属性，出现在所有邻居中次数最多的值作为特征值。

(4)　新合成样本的标签集合生成方式是邻居样本通过投票的方式超过半数的小类标签作为新样本的标签集合。

ML-UpgradeSMOTE 算法过程如表 4-2 中的描述。

表 4-2 ML-UpgradeSMOTE 算法伪代码

算法：ML-UpgradeSMOTE
Input: 训练数据集 D，待插值的样本 instance，对应的小类标签集合 minBag
Output: 插值后的新样本
(1)**Function** ML-UpgradeSMOTE(D,instance, minBag)
(2) **For** each $minBag_i$ **in** instance **do**
(3) minBagSamples = 获取 $minBag_i$ 对应的所有样本
(4) **For** each sample **in** minBagSamples **do**
(5) 计算 instance 在 minBagSamples 中距其他样本的距离记为 distances
(6) **End for**
(7) 根据 distances 对样本由小到大排序
(8) 获取最靠前的 k 个实例 得 neighbors
(9) neighbors 中随机选择一个实例 refNeigh
(10) synthSmpl = newSample(sample, refNeigh, neighbors)
(11) synthSmpl 加入训练集 D
(12) **End for**
(13) **End function**
(14) **Function** newSample(sample, refNeigh, neighbors)
(15) synthSmpl =new Sample
(16) **For** each feature **in** synthSmpl **do**
(17) **If** feature is numeric **then** //数值型属性插值处理
(18) Diff = refNeigh.feature – sample.feature
(19) Offset = diff * randvalue(0,1)
(20) Value =sample.feature + offset
(21) **Else** value =mostFreVal(neighbors.feature)//频次最多
(22) synthSmpl.feature = value
(23) **End for**
(24) Lblcounts = count(sample.labels, neighbors.labels,minBag) //计算种子样本和邻居样本中小类标签频次
(25) **If** lblcounts>(k+1)/2 **then**
(26) 将超过半数的标签作为新样本 synthSmpl 的标签集合
(27) **Return** synthSmpl
(28) **End function**

4.3 ML-DARS 算法实验

4.3.1 实验数据集

为了展现 ML-DARS 算法的优势，本次实验在心血管疾病数据集和 2 个公共多标签数据集进行了实验验证，其中包括了均衡与不均衡的数据集，并与 ML-RUS、ML-SMOTE 算法进行对比分析，具体信息见表 4-3，其展示了原始数据集的样本数目、特征数目、标签数目、对应的标签基数、最大不均衡度、平均不均衡度、平均样本数度量指标等，一定程度上反映了数据的分布情况。可以看出 Cardiovascular（心血管疾病数据集）标签基数很小，最大不均衡度达到了 100 以上，样本数目在 10 万以上，而

平均样本数目才 1779，说明数据分布很不均衡，而 Chronic 相对均衡，Emotions 较为均衡。

表 4-3 数据集分布指标

Dataset	samples	features	Labels	Card	MaxIRLbl	MeanIR	meanInstances
Cardiovascular	129066	178	9	1.46	121.49	54.37	1779
Chronic	6591	310	10	2.38	13.03	4.86	847
Emotions	391	72	6	1.81	1.77	1.49	112

4.3.2 实验设置

本次实验基于开源的多标签学习库 Mulan 来进行开发， Mulan 中包含大量的、成熟的多标签学习算法，同时也提供了多种公共多标签数据集，为多标签分类领域的研究提供了极大的便利。

本次实验利用 Hamming-Loss 、SubAccuracy、F-measures$_{micro}$、F-measures$_{macro}$ 作为算法性能评价指标,其指标介绍见 2.2.2 节。F-measures$_{micro}$ 先取所有标签各指标之和，然后在所有标签的整体上求 F 值，F-measures$_{macro}$ 先求各标签的 F 值再对所有标签取平均，两者结合起来从宏观、微观两方面度量多标签分类的准确度。

为了评估多标签均衡性算法对分类效果产生的不同影响，本次实验选取了常用的多标签分类算法 BR、ML-KNN、HOMER 对采样后的数据集进行分类学习，并以 J48、SVM、Logistic 算法作为基分类器，同时，本次实验采用 ML-RUS、ML-SMOTE 算法与 ML-DARS 算法作对比，限于心血管疾病数据集原始数据量太大无法直接进行分类，将 ML-RUS 采样率设置为 0.16 以尽可能接近于 ML-DARS 算法采样的样本数，ML-SMOTE 算法在 ML-RUS 算法基础上进行的，采样率 0.1。公共数据集 chronic 、emotions，设置 ML-RUS 和 ML-SMOTE 算法采样率均为 0.8，其中涉及到求近邻算法，其默认 k 值均为 10，通过不同的性能评价指标来评价算法的好坏。

4.3.3 实验结果分析

为了验证 ML-DARS 算法的有效性，首先对心血管疾病数据集利用算法 ML-RUS、ML-SMOTE、ML-DARS 算法进行了重采样，通过不同的分类器得到最终的结果如表 4-4 所示，Hamming-Loss 指标表示相关标签被预测为不相关标签的比例，该指标越小越好。

表 4-4 ML-DARS 算法在心血管疾病数据集性能比较

Classifier	Hamming-Loss			SubAccuracy		
Algorithm	ML-RUS	ML-SMOTE	ML-DARS	ML-RUS	ML-SMOTE	ML-DARS
BR-Logistic	0.0849	0.0860	0.0675	0.5424	0.5539	0.5747
BR-J48	0.0746	0.0756	0.0743	0.5084	0.5225	0.5440
BR-SVM	0.0939	0.0925	0.0674	0.4541	0.4837	0.5695
HOMER-J48	0.1009	0.1178	0.0854	0.4100	0.4338	0.5076
MLKNN	0.1082	0.1271	0.0833	0.4401	0.4664	0.5147

可以看出，ML-DARS 算法得到的结果均小于 ML-RUS 和 ML-SMOTE 算法，相对

于 ML-RUS 和 ML-SMOTE 算法提升了约 2%的性能，而 ML-SMOTE 算法在 ML-RUS 算法获得的数据集上进行的，所以 ML-SMOTE 算法比 ML-RUS 算法的值稍低。ML-DARS 在精确度指标 SubAccuracy 上明显高于 ML-RUS 和 ML-SMOTE，性能提升了近 5%。

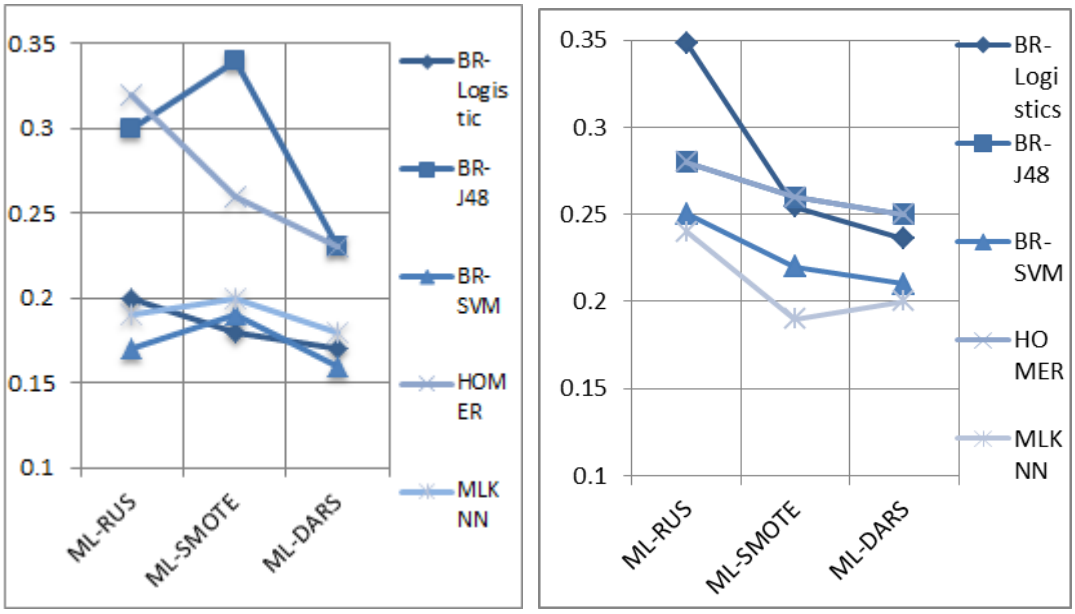
表 4-5 展示了 ML-DARS 算法在 $F\text{-measure}_{micro}$ 和 $F\text{-measure}_{macro}$ 指标上的表现，除了 ML-DARS 算法在 BR-Logistic、BR-SVM 上的准确度低了些，但在其它三种算法上有明显的提升，ML-DARS 算法性能总体上比 ML-RUS 和 ML-SMOTE 算法提升了 6%。

相比于现有的 ML-RUS 和 ML-SMOTE 算法，ML-DARS 算法有效解决了心血管疾病数据集中的类别不均衡问题。

表 4-5　ML-DARS 算法在心血管疾病数据集性能比较

Classifier Algorithm	$F\text{-measure}_{micro}$			$F\text{-measure}_{macro}$		
	ML-RUS	ML-SMOTE	ML-DARS	ML-RUS	ML-SMOTE	ML-DARS
BR-Logistic	0.7427	0.7506	0.7817	0.3174	0.3377	0.3250
BR-J48	0.6256	0.6282	0.7742	0.3016	0.3013	0.3983
BR-SVM	0.7555	0.7664	0.7802	0.3145	0.3158	0.3098
HOMER-J48	0.5871	0.5890	0.7550	0.2864	0.2888	0.4062
MLKNN	0.6178	0.6910	0.7345	0.2373	0.2428	0.2837

针对公共数据集 chronic 、emotions，Hamming-Loss 性能比较如图 4-1 所示，由于 chronic、emotions 数据集本身就较为均衡，ML-RUS、ML-SMOTE 两种算法要么欠采样过度导致信息严重丢失，要么过采样导致样本冗余，ML-DARS 算法根据均衡程度下的平均样本数和容许范围两方面保证了最大程度上保持了原有数据的分布又适当的增加了小类样本获得更为均衡的数据集，所以在 Hamming-Loss 上总体优于 ML-RUS、ML-SMOTE 算法。

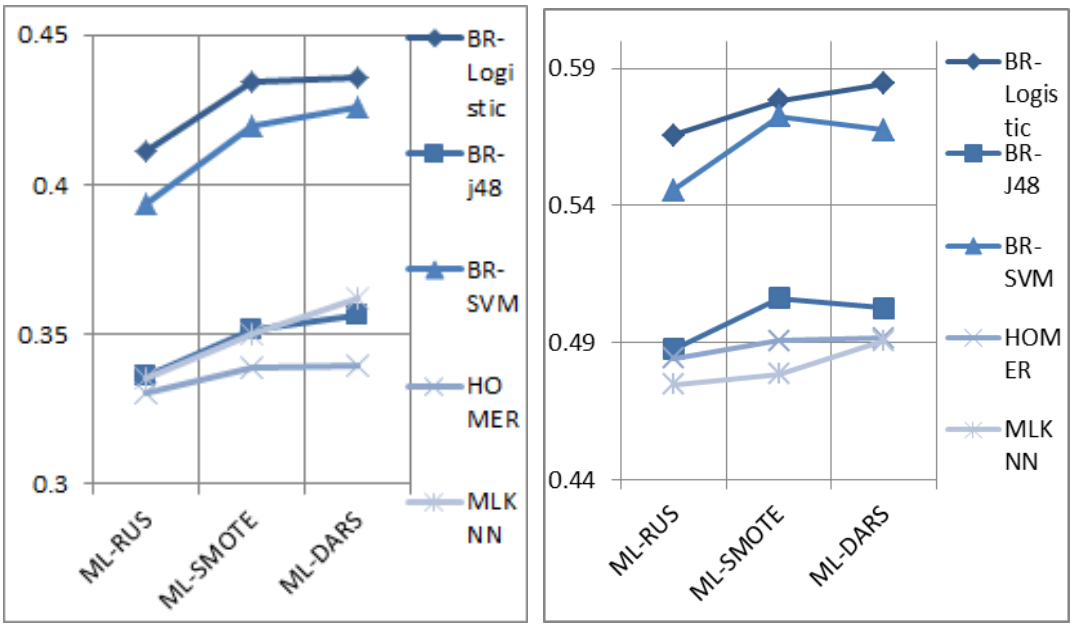


a) Chronic 数据集　　b)　emotions 数据集

图 4-1 Hamming-Loss 性能比较

SubAccuracy 指标如图 4-2 所示，ML-DARS 算法在 chronic 数据集上表现总体优于

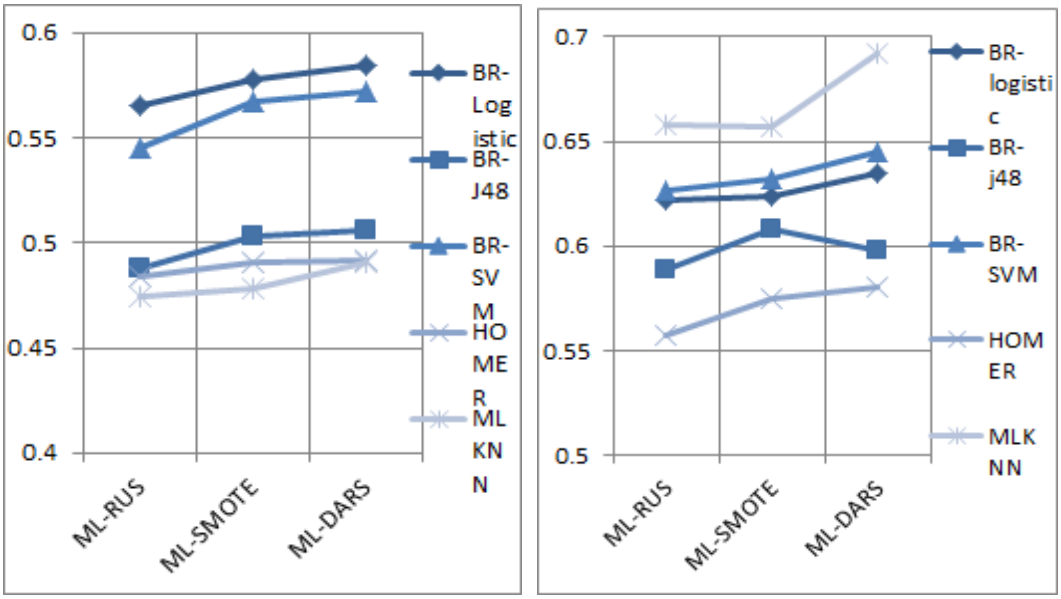
ML-RUS、ML-SMOTE 算法。对于 emotions 数据集，样本数目较小，标签数目相对较大，分类算法难以学习到有效数据，所以欠采样算法导致精确度下降，ML-SMOTE 算法则合成新样本增加数据集的数量，提高了分类精确度，ML-DARS 中欠采样过程可能导致数据集丢失小量大类样本，但是因为其过采样过程增加了小类样本，所以性能与 ML-SMOTE 算法不相上下。



a) Chronic 数据集　　b) emotions 数据集

图 4-2 SubAccuracy 性能比较

F-measure$_{micro}$ 指标如图 4-3 所示，ML-DARS 算法在 chronic 数据集上表现效果比 ML-RUS 算法、ML-SMOTE 算法更好，对 emotions 数据集而言，ML-DARS 除了 BR-J48 分类算法上略低于 ML-SMOTE 算法，其他算法均为最优。

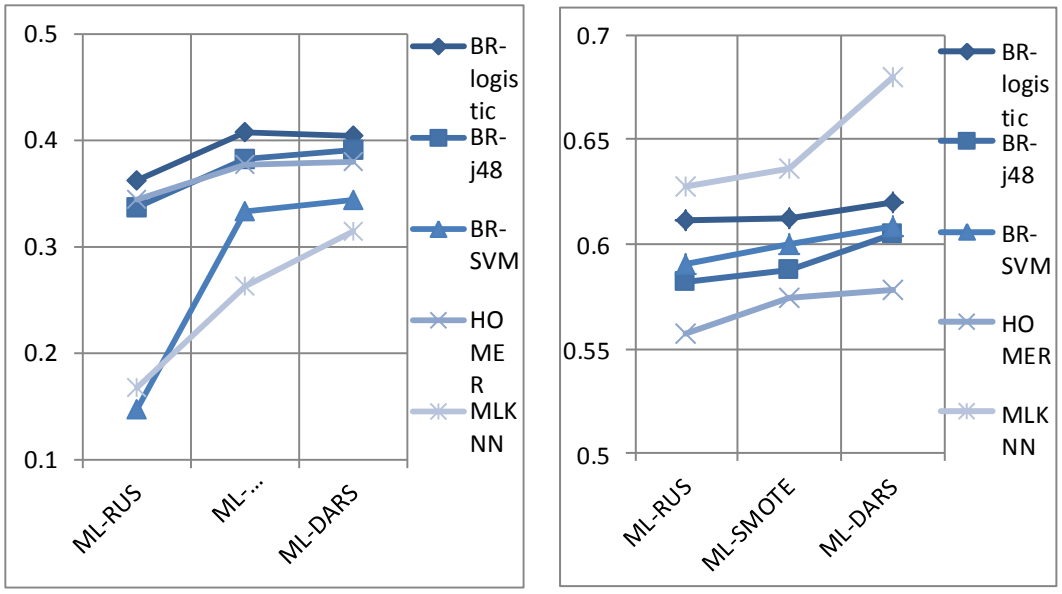


a) Chronic 数据集　　b) emotions 数据集

图 4-3　F-measure$_{micro}$ 性能比较

同理，F-measure$_{macro}$ 指标如图 4-4 所示，ML-DARS 算法总体优于 ML-RUS、

ML-SMOTE 算法。



a) Chronic 数据集　　b) emotions 数据集

图 4-4 $F\text{-measure}_{macro}$ 性能比较

总体来说，ML-DARS 算法能够获取更为均衡的数据集，比 ML-RUS 算法、ML-SMOTE 算法获得了更好的分类性能。

4.4 本章小结

本章对多标签数据集不均衡性问题展开讨论，分析了现有的且常用的多标签不均衡性方法 ML-RUS、ML-SMOTE 的优缺点。为了解决现有的多标签不均衡算法带来的大类样本采样过度造成的信息丢失而小类样本过采样造成的信息冗余，本章提出了一种多标签双重自适应采样算法 ML-DARS，详细介绍了该算法的原理，并利用实际数据集进行了实验验证，与 ML-RUS、ML-SMOTE 算法进行对比分析，证明了 ML-DARS 算法能够获得更为均衡的数据集。

5 基于混合策略的心血管疾病预测模型

本章根据算法集成思想，提出了 ML-KNN 算法和 RAKEL 算法相结合的基于混合策略的多标签分类框架，建立了心血管疾病预测模型。

5.1 模型概述

经过第 3 章和第 4 章对心血管疾病数据集的处理，获得了完整且准确的心血管疾病数据集，用于最终心血管疾病预测模型的建立与分析。

目前已有的多标签分类算法在一定程度上很好的解决了不同应用场景下的分类预测问题，而心血管疾病数据集存在两大特性：一是症状相似性，即患有相同心血管疾病的患者一般有相同的表现症状，例如所有的高血压患者一般会出现头晕、胸闷、心悸等症状，二是心血管疾病之间存在很强的互相影响关系，例如高血压患者更容易患冠心病、心衰等大多数易危及生命的心血管疾病。

基于近邻的标签信息来评估样本的隶属度，这种方法利用局部信息构造决策面，通过度量相似症状的患者患有的疾病来确定待预测患者可能患有的心血管疾病，但是这种方法缺乏表示标签间的相关信息，而且有限的局部信息也会影响分类器的分类性能。而目前充分利用标签间的相关信息的分类算法纵观数据集的全局信息，势必会忽略局部细节。

为了充分挖掘心血管疾病数据集信息获得更好的预测效果，考虑到 ML-KNN 算法能够根据近邻样本信息来预测待测样本，有效的利用了心血管疾病数据中症状相似的特性。而 RAKEL 算法既解决了 LP 计算复杂度问题，又充分了标签间的相关性特点，是心血管疾病间相互关系挖掘的一大助力。基于此，本文结合 ML-KNN 和 RAKEL 算法的优势，提出了基于混合策略的多标签分类框架。

5.2 基于混合策略的心血管疾病预测模型

5.2.1 大量数据分批处理

众所周知，RAKEL 算法是对 LP 所带来的大规模标签组合、数据极不均衡性问题的改进，在一定程度上降低了 LP 问题造成的极大的时空复杂度。然而，RAKEL 算法的本质依然是基于 LP 思想，即便通过标签子集的方式减少了标签组合的粒度进而降低了一次训练的规模，但训练样本总体的数目并未减少，而且该算法需要为每个标签子集训练一个分类模型，也会占用大量的空间资源，经常出现内存空间不足而溢出的现象，在大规模训练集面前，单凭 RAKEL 算法是无法有效训练数据的。

基于此，本文对心血管疾病数据集进行划分，在 RAKEL 算法能够处理的数据规模范围内，将心血管疾病数据集分成多个子集，对每个子集训练一个 RAKEL 模型，在测试集上进行预测，获得测试样本的一次预测结果，循环多个子集，对测试集上样

本的多次预测结果取均值，作为 RAKEL 训练过程中的测试样本预测结果。在这里划分数据子集并非随机，数据集的规模越大，目标标签对应的样本数目差异越大，数据越是不均衡，同样为了确保划分后的子集更为均衡，数据子集划分策略如下：

首先，心血管疾病数据集按照小节 2.3.2 的划分标准，获得小类标签集合和大类标签集合，出现在小类标签集合的所有样本被看成是小类样本集合，其他的样本组成大类样本集合。

然后，通过对心血管疾病数据集的试探，获得 RAKEL 算法大约一次能够处理的样本数目，即获得数据子集的大约估计。由于小类样本集中存在大类标签，而大类样本集中不存在小类标签，为得到更为均衡的数据集，将数据子集中的小类样本和大类样本比设为 1.5:1 到 2:1 之间，即均衡数据的分布，保持了原有数据分布的总体情况。据此计算，可得到数据子集中需要的小类样本数目和大类样本数目 。

最后，根据小类样本集合的大小和数据子集中需要的小类样本数目，可将小类样本集合约均分为 k1 份，同理大类样本集合均分为 k2 份，对 k1 份小类样本子集分别与 k2 份大类样本子集进行混洗，每次使用一次混洗结果进行 RAKEL 算法训练，共混洗 k1*k2 次，即为训练子集的数目。

这样，既解决了 RAKEL 算法在大规模训练样本集上的障碍，又保证了更为均衡的数据集，在多个不同的样本子集上训练而保证了和大规模样本情况下一样的预测效果。

5.2.2 模型构建

根据心血管疾病数据集的特性，本文提出了基于混合策略的多标签分类方法(MR)来构建心血管疾病预测模型。该模型分为三个步骤如下：

首先利用 ML-KNN 算法获得训练集上每个标签被误分类的概率，作为衡量该标签能够被局部信息所表示的能力，由于多标签数据集存在极不均衡性，即多个标签组成的数据集很大，而一种标签对应的样本数目占总样本数的比例则非常小，多标签分类算法在训练过程中，待预测标签对应的样本集为正类，其他样本对应于负类，预测结果通常偏向于大类，为了准确反映待预测标签的被误分类的程度，利用正类样本中被预测为负类的样本所占比例作为每个标签被误分类的概率，记为 α，公式如(5-1)所示，表示标签 j 被 ML-KNN 误分类的概率。

$$\alpha_j = FP_j \big/ L_j \tag{5-1}$$

然后在此基础上，对每一个测试样本 x，经过 ML-KNN 预测得到对应标签的预测结果记为 $f_{ML-KNN}(x, j)$。

利用 5.2.1 大量数据分批训练的思想，以 RAKEL 算法为基准训练样本，获得一次的预测结果记为 $f_{RAKEL}(x, j)$。

最后加权两种算法的预测结果作为最终的预测值，加权结果如公式(5-2)表示。

$$f(x,j)=(1-\alpha)\,f_{ML-kNN}(x,j)+\alpha\bullet\frac{1}{k1*k2}\sum_{i=1}^{k1*k2}f_{RAKEL}(x,j) \tag{5-2}$$

其中$1\le j\le M$，$0\le\alpha\le$。待预测样本通过阈值求出，见公式(5-3)，一般threshold=0.5。

$$h(x)=\{j\mid f(x,j)\ge threshold\} \tag{5-3}$$

该过程获取了 ML-KNN 算法对每个标签的误分类率、对测试数据的预测结果和 RAKEL 算法的预测结果，对两者的结果通过公式(5-2)加权作为最终的预测结果，训练、预测过程可并行进行，弥补了混合策略所带来极大的时间开销。

基于混合策略的多标签分类模型 MR 伪代码描述如下表 5-1 所示。

表 5-1 MR 模型伪代码

算法: MR 模型伪代码
Input: 训练数据集 D, 测试数据集 T
Output: 预测结果，模型评估结果
(1) L ← 为训练集 D 中对应的原始标签集合
(2) **α** ← Hybrid parameter
(3) N← number of misclassified samples about every label
(4) M ← number of instances about every label
(5) ML-KNN train dataset D
(6) **For** each instance **in** D **do**
(7) Truth ← true label of instance
(8) Predict ← ML-KNN predict instance
(9) **For** each label **in** L
(10) **If** $truth_{label}$ ==1 and $predict_{label}$ ==0 **then**
(11) N_{label} .add(1)
(12) **End if**
(13) **End for**
(14) **End for**
(15) **For** each label **in** L
(16) α_{label} ← N_{label} /M_{label}
(17) **End for**
(18) Double[][] rakelPrediction ← integrateRAKEL(D, T) //将数据集划分并分别训练，返回所有分类器对测试样本预测均值
(19) predictFinal ← finally Hybrid result for every sample
(20) **for** each instance **in** T **do**
(21) $predict_{mlknn}$ ←MLKNN .predict(instance)
(22) $predict_{rakel}$ ← rakelPrediction[instance.location]
(23) **For** each label **in** L
(24) $predictFinal_{label}$ ← (1-α_{label}) $predict_{mlknn}$ + α_{label} $predict_{rakel}$
(25) **End for**
(26) **End for**
(27) Evaluation of prediction results

数据集划分为大类子数据集和小类子数据集，大小类标签数据集混洗，RAKEL 算法分批训练，测试样本中每个标签的值取所有 RAKEL 模型的预测均值，数据划分混洗及 RAKEL 训练预测算法伪代码描述如表 5-2 所示。

表 5-2 数据混洗及训练预测算法伪代码

算法: RAKEL 分批训练数据集
Input: 训练数据集 D, 测试数据集 T
Output: 所有测试样本最终的预测结果
(1) Function integrateRAKEL(D , T)
(2) **for** each label **in** L **do**
(3) $IRLbl_{label}$ =calculateIRLbl(D,label)
(4) **If** $IRLbl_{label}$ < meanIR **then**
(5) Label 加入 maxBag 集合中
(6) **else** label 加入 minBag 中
(7) **End for**
(8) smallSamples ← 训练集中出现在 minBag 集合中的所有样本
(9) maxSamples ← 训练集中出现在 maxBag 集合中的所有样本
(10) smallSampleSplit[] ← splitInstances(smallSamples , k1);//小样本集划分为 k1 份
(11) maxSampleSplit[] ← splitInstances(maxSample, k2); //大样本集划分为 k2 份
(12) double[][] val ← 所有测试样本最终的预测结果
(13) **For** each smallSamples **in** smallSampleSplit **do**
(14) **For** each maxSamples **in** maxSampleSplit **do**
(15) trainingSet ← mergeSamples(smallSamples, maxSamples)
(16) rakelmodel← RAKEL. train (trainingSet);
(17) **For** each sample **in** T **do**
(18) output← rakelmodel.prediction(sample)
(19) Val +=output 的结果按在 val 的位置进行累加
(20) **End for**
(21) **End for**
(22) **End for**
(23) val[i][j]=val[i][j]/(smallSampleSplit.length*maxSampleSplit.length)
(24) **return val**
(25) **End function**

5.3 实验

5.3.1 数据集分析与处理

本次实验在第 4 章基础上进行的，对特征选择后的心血管疾病数据集通过 ML-DARS 算法进行处理，降低冗余数据，调整数据分布，获得合理的数据集。

原始数据集基数 card 为 1.46，密度 density 为 0.16，共存在 197 种不同的标签组合，数据分布如图 5-1 所示，由于心血管疾病数据集有 9 个目标标签，所以理论上组合情况一共有 512 种，这里我们通过对原始数据集标签规模排序，取出了 40 种标签组合的分布情况，可以看出 Label1 标签对应的样本数目要远高于其他标签，不仅难以用于分

类算法而且存在数据极不均衡问题。

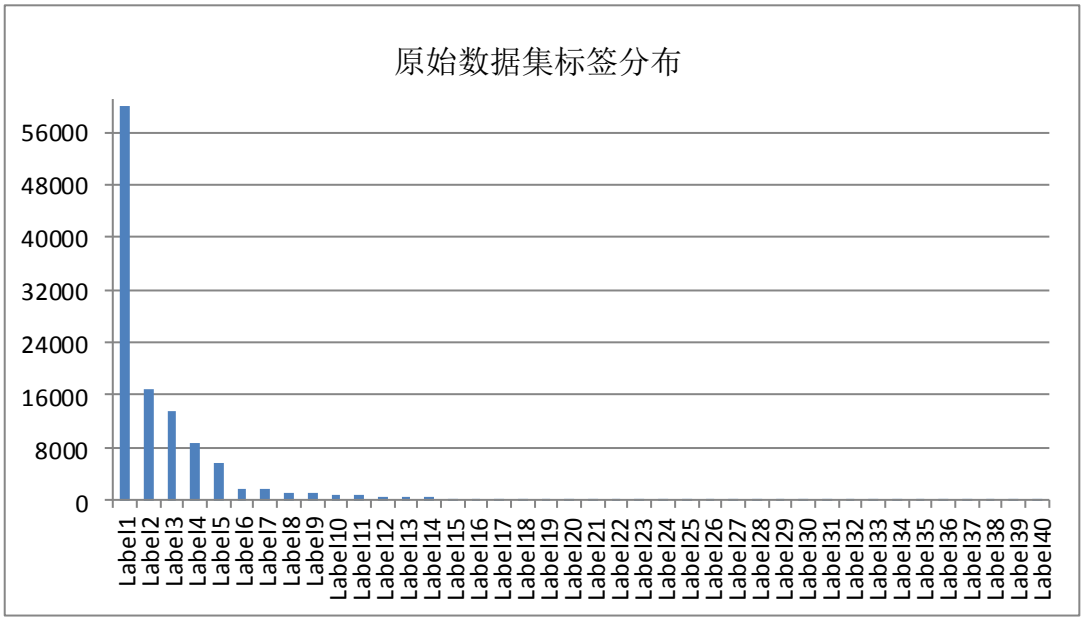


图 5-1 原始数据集部分标签分布

经过 ML-DARS 算法处理，新数据集基数 card 为 2.24，密度 density 为 0.25，共存在 199 种不同的标签组合，前 40 中标签组合分布如图 5-2，可以看出，前 5 种标签组合占数据集的大部分，特别是标签组合 Label1，在原始数据集中几乎占据了全部，经处理后，降低了该组合标签的比重，ML-DARS 算法有效均衡了数据集的分布，且在总体上保持了原有数据分布。

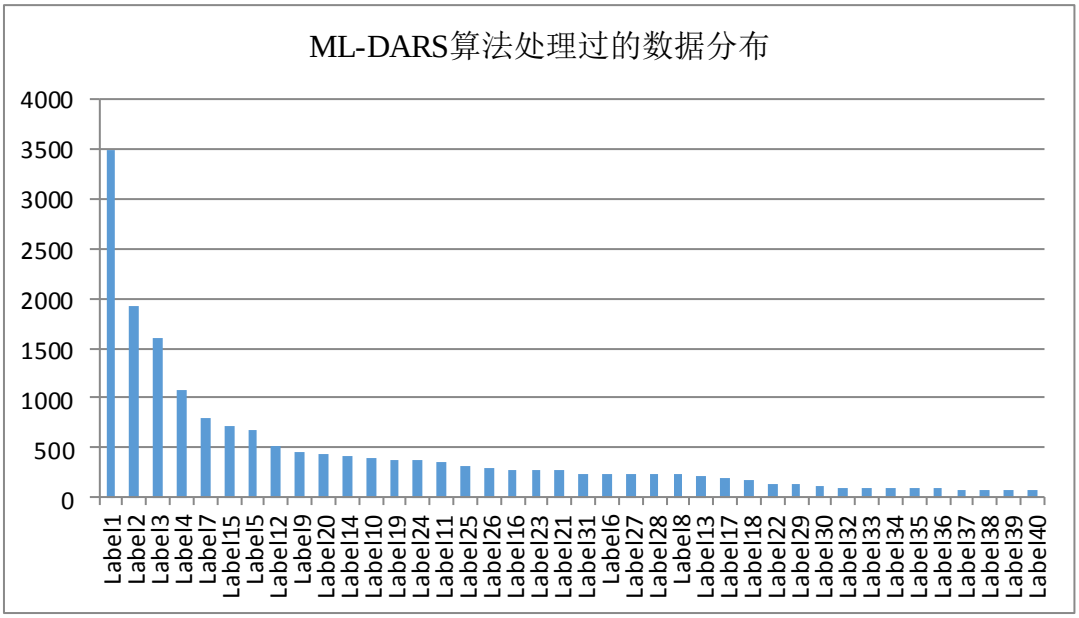


图 5-2 ML-DARS 算法均衡后的部分数据分布

经 ML-DARS 处理，将最终的心血管疾病数据集应用于模型预测过程。心血管疾病数据集包括训练集和测试集数据统计如表(5-3)所示：

表 5-3 最终的数据集统计

数据集	样本数	特征数	标签组合数	标签基数
训练集	19896	178	199	2.24
测试集	6682	178	112	1.48

5.3.2 RAKEL 算法分批训练策略

首先对心血管疾病数据集统计大小类样本占的比例，小类样本集大小为 8294 ，大类样本集大小为 11602。经过对 RAKEL 算法在心血管疾病数据集上的多次试探，在 RAKEL 算法尽可能充分学习到规律的情况下，需要将训练子集数目控制在 7000 到 7200 之间，超过该范围使用 RAKEL 算法在现有软硬件环境下发生内存溢出。

然后为了缓解数据不均衡现象，初步思路将划分后的训练子集大小样本比例尽可能接近 1.5:1，即训练子集中小类样本数目约需要 4200 到 4320 之间，大类样本约需要 2800 到 2880 左右。之后考虑到现有样本的实际情况，由此，对现有的 8294 小类样本集将其划分为 2 部分，每部分为 4147 个小类样本，而大类样本 11602 则划分为 4 部分，每部分约为 2900 个大类样本，最后每个小类样本子集分别与每个大类样本子集进行混洗，得到 8 份数据集分别用于 RAKEL 算法训练预测，获得 8 个 RAKEL 模型，每个模型对测试样本进行预测，最终的预测结果取 8 个模型预测结果的均值作为此部分的预测结果。

5.3.3 分类结果

本次基于混合策略的多标签分类框架 MR 首先涉及到了 ML-KNN 算法，利用该算法获得每个标签对应的误分类率，作为与 RAKEL 算法预测结果的加权系数。这里所有算法涉及到基分类器都默认为 J48，ML-KNN 中近邻系数 k 采用算法包中默认值 10，利用 RAKEL 算法时，将子集大小设置为 3，共训练了 18 个 RAKEL 分类器。获得的权重系数误分率如表 5-4 所示，可以看出大多数标签由于对应的样本比例非常小，利用 ML-KNN 算法小类标签被误分的概率几乎都在 50%以上，极大的降低了模型整体的性能。

表 5-4 权重系数误分率

目标标签	权重（误分率）	标签大约所占样本比例
cerebral_apoplex（脑卒中）	0.9157	17%
hf_ICD（心衰）	0.6037	8%
cardiac_infarction（心肌梗死）	0.6308	8%
Nephropathy（肾衰竭）	0.6103	11%
ischemia_myocardial（心肌缺血）	0.8307	8%
cardiac_function_lesion（心脏功能病）	0.9898	11%
coronary_heart_disease（冠心病）	0.4345	39%
Hypertension（高血压）	0.0526	80%
Diabetes（糖尿病）	0.3881	42%

这里为了更明显的表明混合策略的有效性，分别取基于混合策略分类模型 MR-J48、ML-KNN 和 RAKEL-J48 单独分类所得的 Macro-averaged F-measure 结果，如

图 5-3 所示，该结果更为细节的展示了每个标签分类精度上的变化，从整体上看，可以看出 MR-J48 中权重系数误分率的影响下，MR 方法结合了 ML-KNN 和 RAKEL 算法的优势，获得了更好的分类精度，特别是小类标签心衰疾病和肾衰竭疾病的精确度明显优于 ML-KNN 和 RAKEL 单独分类的结果，误分类权重很大的脑卒中、心肌缺血、心脏功能病变等疾病的精确度在 RAKEL 算法的影响下也结合了 ML-KNN、RAKEL 算法的优势对分类效果有所提升。

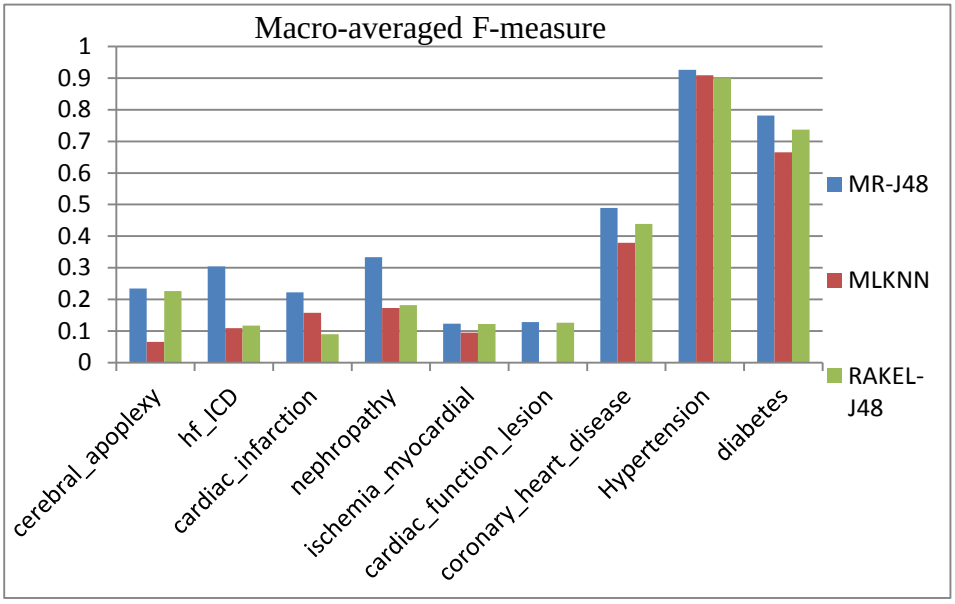


图 5-3 Macro-averaged F-measure 结果

为了进一步证明混合策略框架 MR 的有效性，本文其与其它现有的多标签分类算法进行了详细的比较，对比 MR 框架中用到的 ML-KNN、RAKEL 算法以及 BR、HOMER 算法，下面从 Hamming-Loss、RankingLoss、SubsetAccuracy、Micro-averaged F-measure 等常用的评价指标展示了 MR 的分类效果。

如图 5-4 所示，可以看出，MR 在 Hamming-Loss 表现相比于 ML-KNN、RAKEL 算法降低了约 2%，均要低于其他算法。

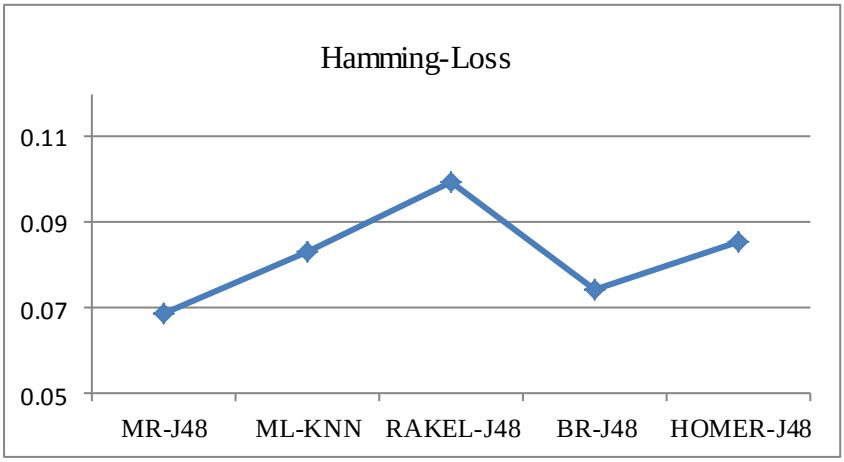


图 5-4 Hamming-Loss

图 5-5 展示了 RankingLoss 指标的表现，RankingLoss 指标通过测试样本集中相关标签与不相关标签排序错误的均值评估分类结果，该值越小，表示相关标签排序越靠前，多标签分类效果越好，其中 ML-KNN 的值比 BR-J48、HOMER-J48 都低，说明了 ML-KNN 利用局部信息预测的优势。MR 在 RankingLoss 表现相比于 ML-KNN、RAKEL 算法降低了 3%，且要低于 BR、HOMER 算法指标值。通过对训练集进行数据混洗，极大降低了 RAKEL 算法的 Hamming-Loss、RankingLoss 值，提高了整体的分类性能。

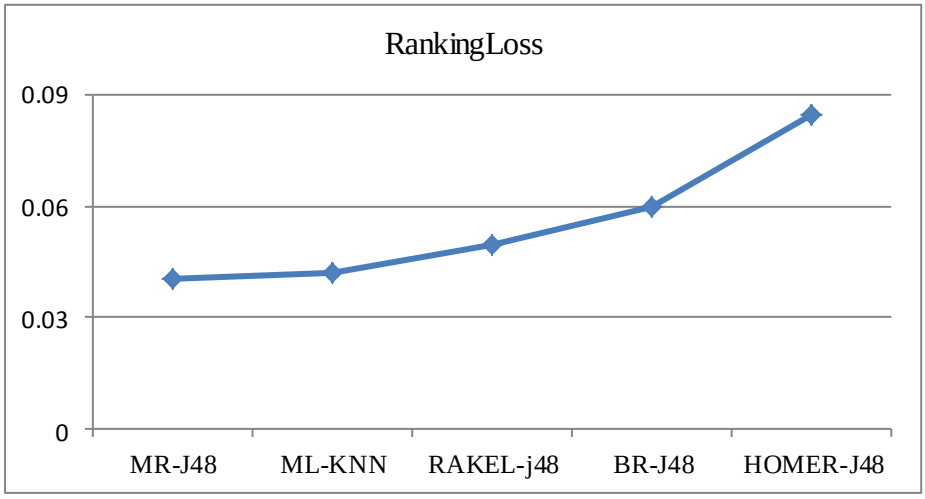


图 5-5 RankingLoss

图 5-6 可以看出，分类准确度方面，MR 方法的 SubsetAccuracye 均要高于其他算法约 5%，ML-KNN 精确度要稍低于 BR 算法但高于 HOMER 算法，RAKEL 由于数据量低于其他算法很多而显示出分类精度很低。基于 MR 框架的心血管疾病预测模型先对训练集根据大小类标签对训练集进行了混洗，获得了更好更均衡的数据子集，并对多个 RAKEL 算法预测结果进行了加权，以及对 ML-KNN 和 RAKEL 分类结果按标签误分率进行调整，极大提高了心血管疾病模型的预测效果。

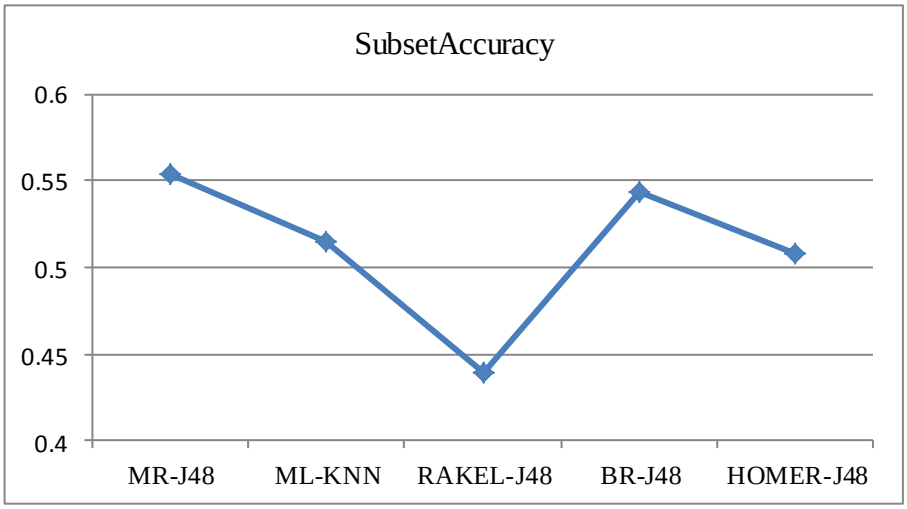


图 5-6　SubsetAccuracy

如图 5-7 所示，MR 方法的 Micro-averaged F-measure 均要高于其他算法近 4%，其中由于 RAKEL 算法直接处理大数据量只能通过减少数据量和调整其子集的大小以及要训练的子集数目来完成，这样 RAKEL 算法无法学习到全部数据带来的信息量，所以分类精度很低。MR 框架下进行的心血管疾病预测模型通过数据混洗和权重调整提高了分类效果。

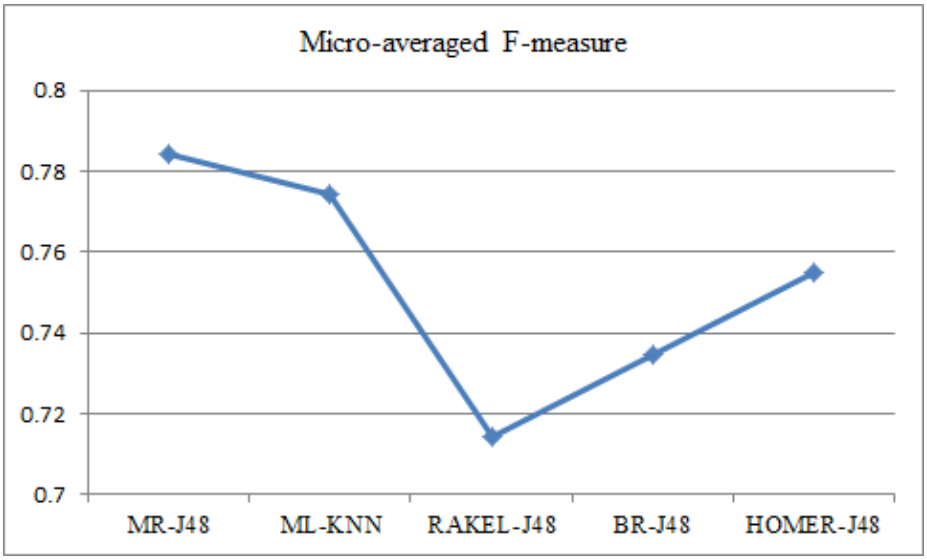


图 5-7 Micro-averaged F-measure

通过二元分类的方法评估真实标签和预测标签的差异性，表 5-5 列出了 MR 模型及其它分类器在 Hamming-Loss、SubAccuracy、Micro-averaged F-measure、acro-averaged F-measure 指标的值，从表中可以看出，除了 Macro-averaged F-measure 指标中 MR 比 BR-J48、HOMER 算法较低外，其他算法的分类效果均在 MR 之下。

表 5-5 分类结果对比 1

Classifier Algorithm	Hamming-Loss ↓	SubAccuracy ↑	Micro-averaged F-measure ↑	Macro-averaged F-measure ↑
MR-J48	0.0686	0.5538	0.7844	0.3636
BR-J48	0.0743	0.544	0.7742	0.3983
ML-KNN	0.0833	0.5147	0.7345	0.2837
HOMER-J48	0.0854	0.5076	0.755	0.4062
RAKEL-J48	0.0995	0.4386	0.7143	0.3264

表 5-6 从基于标签排序的评估方法方面展示了各分类器的分类效果，MR 的指标值均要好于其它分类器的指标，ML-KNN 算法在 Hamming-Loss、SubAccuracy、Ranking Loss、Coverage、Average Precision、OneError 指标中分类效果仅次于 MR 模型，而在 RAKEL 算法在 Micro-averaged F-measure、Macro-averaged F-measure 指标要高于 ML-KNN。所以从 ML-KNN 算法挖掘局部信息和 RAKEL 算法考虑全局标签关系两方面来入手，基于 MR 框架的心血管疾病预测模型充分学习训练集数据的规律从而很好的提高了心血管疾病的预测效果。

表 5-6 分类结果对比 2

Classifier Algorithm	Ranking Loss ↓	Coverage ↓	Average Precision ↑	OneError ↓
MR-J48	0.0342	0.8661	0.9293	0.0815
ML-KNN	0.0422	0.9446	0.9132	0.0942
RAKEL-J48	0.0548	1.0978	0.8924	0.1423
BR-J48	0.0599	1.1369	0.8966	0.1236
HOMER-J48	0.0847	1.4356	0.8661	0.1534

5.4 本章小结

本章作为心血管疾病预测模型中的最后一步，将多标签分类算法应用于心血管疾病预测模型获得并分析预测效果。首先在前面章节的基础上，为获得更好的分类效果，根据心血管疾病数据集的特性，本文采用基于混合策略的多标签分类框架，提出了 ML-KNN 算法和 RAKEL 算法相结合的多标签分类框架，建立心血管疾病预测模型，该模型对 ML-KNN、RAKEL 算法取长补短，并且在 RAKEL 算法无法处理大数据量的情况下，防止欠采样导致的信息丢失降低分类性能，同时获取更为均衡的数据集，对心血管疾病数据集按照均衡性指标进行划分，采用大类样本小类样本混洗的方式，反复多次，保证每个数据集中的每个样本参与训练，在此基础上进行了实验分析验证，结果表明该模型相比于现有的多标签分类算法取得了很好的预测效果。

6 结论与展望

6.1 论文工作结论

本文以面向区域医疗和公共卫生的健康大数据处理分析研究及示范应用项目为背景，从各医疗机构的原始门诊数据出发，建立了基于多标签分类的心血管疾病预测模型。由于原始门诊数据包含了所有疾病人群，且存在大量脏数据，特征因子分散于各个表中，因此无法直接用于心血管疾病预测模型的建立。此外多标签数据集存在严重数据不均衡问题导致基于多标签分类的心血管疾病预测模型预测效果变差，现有多标签分类算法难以充分学习到心血管疾病数据集的全部信息。为此本文从心血管疾病数据采集出发，通过以下工作建立了基于多标签分类的心血管疾病预测模型：

1) 从项目所提供真实的、可靠的原始门诊数据出发，通过统计分析各心血管疾病人群，确定了预测目标及提取对应数据，得到了用于心血管疾病预测建模的初始数据集，并通过异常值、缺失值检测处理和有效的二次特征选择，获得了干净的、完善的具有多标签分类特性的心血管疾病数据集。根据研究过程，设计实现了心血管疾病数据集获取、预处理和特征选择过程并给出了处理结果。

2) 为了能够将心血管疾病数据集应用于多标签分类算法中，提高多标签分类的可靠性，本文对心血管疾病数据集进行了数据均衡性采样，根据心血管疾病数据集的多标签特性以及确保采样后的数据总体上仍能保持原始数据的分布，提出了欠采样、过采样过程相结合的多标签双重自适应随机采样算法 ML-DARS，并在心血管疾病数据集和两种不同的公开数据集上实验验证和对比了现有多标签均衡算法。

3) 本文在现有多标签分类算法研究的基础上，结合心血管疾病数据集相同疾病的患者症状一般相同和疾病间强相关性的两大医学特性，提出了以 ML-KNN 和 RAKEL 算法为基础的基于混合策略的多标签分类框架建立心血管疾病预测模型，并对 RAKEL 无法处理大量数据的问题，提出了小类标签对应样本集和大类标签样本集分批混洗的方式应用 RAKEL 算法，通过 ML-KNN 误分类权重作为 ML-KNN 和 RAKEL 算法预测结果值的加权，获得最终的预测结果，最后以 ML-DARS 算法处理后得心血管疾病数据集作为模型输入数据进行了实验验证，并给出了预测结果的各项评估指标作为心血管疾病模型预测效果的展示。

综上所述，通过此次心血管疾病预测模型的建立，表明利用多标签分类算法能够有效解决医学领域疾病预测这样一个复杂难题，为机器学习技术应用于医学数据研究提供了新方法、新思路。

6.2 展望

本文为解决心血管疾病预测问题，同时考虑到心血管疾病间存在的复杂关系，多标签分类技术被作为此次预测的手段。本文为了将多标签分类算法应用于心血管疾病预测问题，提出了一套完整的医疗数据处理流程， 获得了干净且真实的心血管疾病数据集，提出的均衡算法解决多标签数据集不均衡问题，基于混合策略的模型获得了更高的分类效果。此外心血管疾病预测模型依然有待进一步提升，未来工作从以下两方面进行考虑，一方面，由于心血管疾病预测模型中特征大部分为病症，而这些病症种类繁多，且为 0-1 特征，导致特征维度过大也过于稀疏，所以疾病的特征选择作为今后的研究重点。另一方面，根据本文的心血管疾病预测模型处理过程，扩展到其它疾病作为预测目标进行分析。

致　谢

参考文献

[1] 刘少楠．高血压并发心衰风险预测模型的研究与应用[D]．西安交通大学，2016．

[2] 杭州臻景功能医学中心．功能医学:如何降低血管疾病的风险[J]．健康人生，2016(5):39-40．

[3] Poke Mogo 博客．疾病预测，机器学习和医疗保健．https://blog.csdn.net/pokemogo/article/details/79075269.

[4] 王普．多标记学习算法研究及生物医学数据挖掘中的应用[D]．中科院深圳先进技术研究院，2017．

[5] 董纯洁．基于实例与逻辑回归的多标签分类模型[D]．南京大学，2013．

[6] 邬 杨．基于机器学习的卵巢肿瘤预测与分析研究．吉林大学，2016．

[7] He J, Gu H, Liu W. Imbalanced Multi-Modal Multi-Label Learning for Subcellular Localization Prediction of Human Proteins with Both Single and Multiple Sites[J]. Plos One, 2012, 7(6):e37155.

[8] Hofer T, Hofer T, Schumacher M, et al. Performance comparison of multi-label learning algorithms on clinical data for chronic diseases[J]. Computers in Biology & Medicine, 2015, 65(C):34-43.

[9] Zhang W, Liu F, Luo L, et al. Predicting drug side effects by multi-label learning and ensemble learning.[J]. Bmc Bioinformatics, 2015, 16(1):365.

[10] 王宁.基于 Hadoop 平台的海量医疗数据挖掘算法的研究与实现. 北京邮电大学.2013．

[11] Tsoumakas G, Katakis I, Taniar D. Multi-Label Classification: An Overview[J]. International Journal of Data Warehousing & Mining, 2008, 3(3):1-13.

[12] Zhang M L, Zhou Z H. A Review on Multi-Label Learning Algorithms[J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(8):1819-1837.

[13] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification[J]. Machine Learning, 2011, 85(3):333-359.

[14] Read J. A pruned problem transformation method for multi-label classification[C]// Proc. 2008 New Zealand Computer Science Research Student Conference. 2008:143--150.

[15] Tsoumakas G, Katakis I, Vlahavas I. Effective and efficient multilabel classification in domains with large number of labels[J]. Ecml/pkdd Workshop on Mining Multidimensional Data, 2008.

[16] Tsoumakas G, Katakis I, Vlahavas I. Random k-Labelsets for Multilabel Classification[J]. IEEE Transactions on Knowledge & Data Engineering, 2011, 23(7):1079-1089.

[17] Schapire R E, Singer Y. BoosTexter: A Boosting-based System for Text Categorization[J]. Machine Learning, 2000, 39(2-3):135-168.

[18] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning[M]. Elsevier Science Inc. 2007.

[19] 周浩．中文多标签文本分类算法研究[D]．上海交通大学，2014．

[20] 李思豪，陈福才，黄瑞阳．一种多标签随机均衡采样算法[J]．计算机应用研究，2017，34(10):2929-2932．

[21] Boutell M R, Luo J, Shen X, et al. Learning multi-label scene classification ☆[J]. Pattern Recognition, 2004, 37(9):1757-1771.

[22] Seiffert C, Khoshgoftaar T M, Hulse J V, et al. RUSBoost: A Hybrid Approach to Alleviating

Class Imbalance[J]. IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans, 2010, 40(1):185-197. 5(C):34-43.

[23] Tsoumakas G, Katakis I, Vlahavas I. Mining Multi-label Data[J]. 2009:667-685.

[24] 何伟骏．基于层次—互斥模型的多标签分类算法的研究与应用[D]．中山大学，2015．

[25] 张居杰．多标签学习中关键问题研究[D]．西安电子科技大学，2016．

[26] 王臻．基于学习标签相关性的多标签分类算法[D]．中国科学技术大学，2015．

[27] 谷炫志．基于情感的多标签个性化音乐分类技术的研究与实现[D]．浙江大学，2016．

[28] 陈自洁．多标签分类问题的图结构描述及若干学习算法的研究[D]．华南理工大学，2015．

[29] 李熙铭．基于主题模型的多标签文本分类和流文本数据建模若干问题研究[D]．吉林大学，2015．

[30] 方铭．多标签分类算法的研究及其在中医诊断帕金森领域的应用[D]．南京大学，2015．

[31] 王刚．多标签学习及其在帕金森中医诊断中的应用[D]．南京大学，2014．

[32] 肖雨奇．多标签学习应用于中医诊断帕金森中类别不均衡问题研究[D]．南京大学，2016．

[33] 陈旭，刘鹏鹤,，孙毓忠等．面向不均衡医学数据集的疾病预测模型研究．Vol.40，在线出版号 No.155．

[34] Charte F, Rivera A J, Jesus M J D, et al. Addressing imbalance in multilabel classification: Measures and random resampling algorithms[J]. Neurocomputing, 2015, 163:3-16.

[35] Charte F, Rivera A J, Jesus M J D, et al. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation[J]. Knowledge-Based Systems, 2015, 89:385-397.

[36] Zhang M L. Ml-rbf: RBF Neural Networks for Multi-Label Learning[M]. Kluwer Academic Publishers, 2009.

[37] Wang X L, Chen Y Y, Zhao H, et al. Parallelized extreme learning machine ensemble based on min-max modular network[J]. Neurocomputing, 2014, 128(5):31-41.

[38] Tahir M A, Kittler J, Bouridane A. Multilabel classification using heterogeneous ensemble of multi-label classifiers[J]. Pattern Recognition Letters, 2012, 33(5):513-523.

[39] Liu W, Li Q, Cai Y, et al. A prototype of healthcare big data processing system based on Spark[C]// International Conference on Biomedical Engineering and Informatics.IEEE, 2016:516-520.

附 录

表附录 A-1 ICD 码和病症名称的对应关系

ICD 码	病症	ICD 码	病症
BIT01	霍乱	F01	血管性痴呆
BIT02	先天性梅毒	F02_3*	帕金森病性痴呆(G20+)
BIT04	疱疹病毒[单纯疱疹]感染	F05	谵妄,非由酒精和其他精神活性物质所致
BIT05	皮肤癣菌病	F06	脑损害和功能障碍及躯体疾病引起的其他精神障碍
BIT06	恶性疟原虫疟疾	G20_x00	帕金森病
BIT07	口腔、食管和胃原位癌	G20_x03	帕金森综合征
BIT08	缺铁性贫血	G30	阿尔茨海默病
BIT09	播散性血管内凝血[去纤维蛋白综合征]	G43	偏头痛
BIT10	先天性碘缺乏综合征	G44	其他头痛综合征
BIT11	甲状旁腺功能减退症	G45	短暂性大脑缺血性发作和相关的综合征
BIT13	夸希奥科病[恶性营养不良病]	G47	睡眠障碍
BIT14	维生素 A 缺乏病	G81	偏瘫
BIT15	局部多脂症,肥胖症	G90	自主神经系统的疾患
BIT16	芳香氨基酸代谢紊乱	H34	视网膜血管阻塞
BIT17	脑部疾病、损害和功能障碍引起的人格和行为障碍	H35_004	高血压性视网膜病变
BIT20	其他的基底核变性疾病	H35_008	视网膜血管炎
BIT24	神经根和神经丛疾患	I10	特发性(原发性)高血压
BIT27	中毒性脑病	I11_000	高血压心脏病伴心力衰竭
BIT28	睑腺炎和睑板腺囊肿	I11_900	高血压性心脏病
BIT29	外耳炎	I12_000	高血压性肾衰竭
BIT30	风湿热伴有心脏受累	I12_900	高血压性肾病
BIT31	静脉炎和血栓性静脉炎	I12_902	肾动脉硬化
BIT32	急性鼻咽炎[感冒]	I15	肾血管性高血压
BIT33	病毒性肺炎，不可归类在他处者	I15	继发于特指肾疾患高血压
BIT34	急性支气管炎	I15	Liddle 综合征
BIT35	肺气肿	I15	继发于内分泌疾患高血压
BIT36	哮喘	I15	特指继发性高血压
BIT37	支气管扩张(症)	I15	继发性高血压
BIT38	肺水肿	I20_000	不稳定性心绞痛
BIT39	口腔、涎腺和颌疾病	I20_801	稳定型心绞痛
BIT40	葡萄球菌性烫伤样皮肤综合征	I20_900	心绞痛
BIT41	化脓性关节炎	I21	急性心肌梗死
BIT42	急性肾炎综合征	I24_901	急性冠脉综合征
BIT44	无脑儿和类似畸形	I25_101	冠状动脉狭窄
BIT45	心脏搏动异常	I25_102	冠状动脉粥样硬化
BIT46	咳嗽	I25_103	冠状动脉粥样硬化性心脏病
BIT47	胃肠气胀及有关情况	I25_200	陈旧性心肌梗死
BIT48	嗜眠、木僵和昏迷	I25_500	缺血性心肌病
E10_900	1 型糖尿病	I25_801	慢性冠状动脉供血不足
E11_601	2 型糖尿病足病	I25_900	慢性缺血性心脏病
E11_700	2 型糖尿病伴多并发症	I25_901	冠状动脉性心脏病
E13_907	继发性糖尿病	I26_9	肺栓塞未提及急性肺源性心脏病
E14_900	糖尿病	I27_0	原发性肺动脉高压
E16_200	低血糖症	I27_9	慢性肺源性心脏病
E16_803	代谢综合征	I31	心包的其他疾病

表附录 A-2　ICD 码和病症名称的对应关系

ICD 码	病症	ICD 码	病症
I34	非风湿性二尖瓣疾患	I69_3	脑梗死后遗症
I35	非风湿性主动脉瓣疾患	I69_8	其他和未特指的脑血管病后遗症
I38	瓣膜未特指的心内膜炎	I70_0	主动脉的动脉粥样硬化
I40	急性心肌炎	I70_2	四肢动脉的动脉粥样硬化
I42	心肌病	I70_8	其他动脉的动脉粥样硬化
I46_901	呼吸心跳骤停	I70_9	全身性和未特指的动脉粥样硬化
I47_1	室上性心动过速	I72_9	未特指部位的动脉瘤
I47_2	室性心动过速	I73_900	周围血管病
I48_x01	心房颤动	I74_3	下肢动脉栓塞和血栓形成
I48_x02	阵发性心房纤颤	I77_1	动脉狭窄
I49_001	心室颤动	I77_6	未特指的动脉炎
I49_100	房性早搏	I79_2*	分类于他处的疾病引起的周围血管病
I49_300	室性早搏	I88	非特异性淋巴结炎
I49_400	早搏	I89	淋巴管和淋巴结的其他非感染性疾患
I49_500	病窦综合征	I95_900	低血压
I49_800	特指心律失常	I99_X00	循环系统特指疾患
I49_900	心律失常	N17	急性肾衰竭
I50_000	充血性心力衰竭	N18	慢性肾病
I50_100	左心衰竭	N19	未特指的肾衰竭
I50_900	心力衰竭	R04	呼吸道出血
I50_901	心功能不全	R09	累及循环和呼吸系统的其他症状和体征
I50_902	心功能Ⅰ级	R10	腹部和盆腔痛
I50_903	心功能Ⅱ级	R11	恶心和呕吐
I50_904	心功能Ⅲ级	R12	胃灼热
I50_905	心功能Ⅳ级	R13	吞咽困难
I50_906	心肌损害	R45	累及情绪状态的症状和体征
I51_400	心肌炎	R50	其他和原因不明的发热
I51_600	心血管疾病	R51	头痛
I51_700	心脏扩大	R52	疼痛,不可归类在他处者
I51_701	左室肥大	R53	不适和疲劳
I51_900	心脏病	R54	衰老
I61_900	脑出血	R55	晕厥和虚脱
I61_902	脑血管破裂	R56	惊厥,不可归类在他处者
I62_001	硬膜下血肿	R58	出血,不可归类在他处者
I63	脑梗死	R59	淋巴结增大
I64	脑卒中,未特指为出血或梗死	R60	水肿,不可归类在他处者
I65	入脑前动脉的闭塞和狭窄,未造成脑梗死	R61	多汗症
I66	大脑动脉的闭塞和狭窄,未造成脑梗死	R63	有关食物和液体摄取的症状和体征
I67_2	大脑动脉粥样硬化	R64	恶病质
I67_4	高血压脑病	R65	全身炎症反应综合征
I67_802	急性脑血管病	R68	其他的一般症状和体征
I67_803	脑动脉供血不足	R69	原因不知和原因未特指的发病
I67_805	慢性缺血性脑血管病	R72	白细胞异常,不可归类在他处者
I67_900	脑血管病	R73	血糖水平升高
I69_1	脑内出血后遗症	R74	血清酶水平异常

攻读学位期间取得的研究成果

学位论文独创性声明（1）

本人声明：所呈交的学位论文系在导师指导下本人独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担以下责任和后果：

1．交回学校授予的学位证书；

2．学校可在相关媒体上对作者本人的行为进行通报；

3．本人按照学校规定的方式，对因不当取得学位给学校造成的名誉损害，进行公开道歉。

4．本人负责因论文成果不实产生的法律纠纷。

论文作者（签名）：　　日期：　　年　　月　　日

学位论文独创性声明（2）

本人声明：研究生　　所提交的本篇学位论文已经本人审阅，确系在本人指导下由该生独立完成的研究成果。

本人如违反上述声明，愿意承担以下责任和后果：

1．学校可在相关媒体上对本人的失察行为进行通报；

2．本人按照学校规定的方式，对因失察给学校造成的名誉损害，进行公开道歉。

3．本人接受学校按照有关规定做出的任何处理。

指导教师（签名）：　　日期：　　年　　月　　日

学位论文知识产权权属声明

我们声明，我们提交的学位论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公开阅览、借阅以及申请专利等权利。学位论文作者离校后，或学位论文导师因故离校后，发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安交通大学。

论文作者（签名）：　　日期：　　年　　月　　日

指导教师（签名）：　　日期：　　年　　月　　日