**硕士学位论文**

损失最小化的不平衡样本分类方法的研究

**RESEARCH ON APPROACH FOR CLASSIFICATION OF INTRA-CLASS IMBALANCED DATA SETS**

**姜嘉尧**

**哈尔滨工业大学**

**2015年12月**

国内图书分类号：TP399 学校代码：10213

国际图书分类号：621.3 密级：公开

**工学硕士学位论文**

针对类内不平衡样本分类方法的研究

|  |  |  |
| --- | --- | --- |
| 硕士研究生 | ： | 石凤兴 |
| 导 师 | ： | 张春慨副教授 |
| 申请学位 | ： | 工学硕士 |
| 学科 | ： | 计算机科学与技术 |
| 所 在 单 位 | ： | 深圳研究生院 |
| 答 辩 日 期 | ： | 2015年12月 |
| 授予学位单位 | ： | 哈尔滨工业大学 |

Classified Index: TP399

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

**RESEARCH ON APPROACH FOR CLASSIFICATION OF INTRA-CLASS IMBALANCED DATA SETS**

|  |  |
| --- | --- |
| **Candidate：** | Jiayao Jiang |
| **Supervisor：** | Associate Prof. Chunkai Zhang |
| **Academic Degree Applied for：** | Master Degree in Engineering |
| **Speciality：** | Computer Science and Technology |
| **Affiliation：** | Shenzhen Graduate School |
| **Date of Defence：** | Dec., 2015 |
| **Degree-Conferring-Institution：** | Harbin Institute of Technology |

# 摘 要

随着移动互联网技术的兴起，越来越多的原始数据被收集起来用于分析与挖掘，其中很多领域内的数据是严重不平衡的，既属于不同类别的样本数量相差极为悬殊。传统的机器学习方法通常采用全局分类准确率作为训练目标，在不平衡数据集上表现欠佳，所以不平衡数据分类算法也逐渐成为一个受人关注的课题。

目前主要的不平衡数据集分类方法分为两种，分别是数据重采样法以及提升算法，他们的主要思路是通过改变原始数据集中样本的权重或者比例来使原始数据达到一种“平衡”状态，从而适用于传统的分类算法，这类算法的结果与数据分布情况相关密切，往往需要一个复杂的调节过程才能得到一个较好的结果而且泛化能力不是很强。

本文首先介绍了不平衡数据分类中存在的难点和主要问题所在，以及不平衡数据分类的评价标准；然后简述了现有的经典不平衡分类算法，并从问题入手分析他们的主要思想，分析他们的优缺点；接着本文提出了以不平衡样本分类评价标准为最终训练目标的分类思路，通过评价标准构造一个可优化的函数作为训练目标进行训练，采用了整体数据集优化分类器的思路，使用神经网络模型，反向传播算法来使目标函数迅速收敛于一个最优解以达到不平衡样本训练的目的；最后通过与其他算法进行对比试验来证明了此算法的有效性。

关键词：不平衡样本；神经网络；最小化损失学

# Abstract

目 录

[摘 要 II](#_Toc466244340)

[Abstract II](#_Toc466244341)

[第 1 章 绪 论 2](#_Toc466244342)

[1.1 研究背景与意义 2](#_Toc466244343)

[1.2 国内外研究现状 2](#_Toc466244344)

[1.2.1 经典不平衡处理方法 2](#_Toc466244345)

[1.2.2 传统分类器优化法 2](#_Toc466244346)

[1.3 主要研究内容 2](#_Toc466244347)

[1.4 章节结构安排 2](#_Toc466244348)

[第 2 章 相关研究内容介绍 2](#_Toc466244349)

[2.1 不平衡样本分类评价标准 2](#_Toc466244350)

[2.1.1 G-Measure与A-Mean 2](#_Toc466244351)

[2.1.2 F1值 2](#_Toc466244352)

[2.1.3 Roc曲线与Auc面积 2](#_Toc466244353)

[2.2 最大化F1期望的分类过程 2](#_Toc466244354)

[2.2.1 问题转化过程 2](#_Toc466244355)

[2.2.2 F1最大分类过程推理 2](#_Toc466244356)

[2.2.3 F值期望的求解过程 2](#_Toc466244357)

[2.3 损失最小化学习 2](#_Toc466244358)

[2.3.1 引言 2](#_Toc466244359)

[2.3.2 结构化支持向量机 2](#_Toc466244360)

[2.3.3 最小化全局损失函数 2](#_Toc466244361)

[2.3.4 训练过程介绍 2](#_Toc466244362)

[2.4 本章小结 2](#_Toc466244363)

[第 3 章 最大化F1值的不平衡样本数据集训练算法 2](#_Toc466244364)

[3.1 引言 2](#_Toc466244365)

[3.2 算法基本思想 2](#_Toc466244366)

[3.3 基于最大化F1值分类过程的神经网络 2](#_Toc466244367)

[3.3.1 算法思想与概述 2](#_Toc466244368)

[3.3.2 最大化F1值分类过程的神经网络 2](#_Toc466244369)

[3.3.3 算法分析 2](#_Toc466244370)

[3.4 最小化损失学习神经网络 2](#_Toc466244371)

[3.4.1 最小化全局损失学习与算法基本思想 2](#_Toc466244372)

[3.4.2 损失函数的设计 2](#_Toc466244373)

[3.4.3 训练过程 2](#_Toc466244374)

[3.5 本章小结 2](#_Toc466244375)

[第 4 章 实验结果与分析 2](#_Toc466244376)

[4.1 实验环境及数据集介绍 2](#_Toc466244377)

[4.1.1 实验环境介绍 2](#_Toc466244378)

[4.1.2 实验数据预处理 2](#_Toc466244379)

[4.1.3 实验数据集介绍 2](#_Toc466244380)

[4.2 算法基础参数设置 2](#_Toc466244381)

[4.3 算法有效性验证试验数据 2](#_Toc466244382)

[4.3.1 算法分类结果对比试验 2](#_Toc466244383)

[4.3.2 算法性能对比试验 2](#_Toc466244384)

[4.4 本章小结 2](#_Toc466244385)

[结 论 2](#_Toc466244386)

[参考文献 2](#_Toc466244387)

[攻读硕士学位期间发表的论文及其它成果 2](#_Toc466244388)

[哈尔滨工业大学学位论文原创性声明和使用权限 2](#_Toc466244389)

[致 谢 2](#_Toc466244390)

1. 绪 论

## 研究背景与意义

随着大数据时代的到来以及各种网络化的系统变得规模越来越大，结构越来越复杂，各种各样的监视、安全、金融等系统每天都获取海量的原始数据，所以原始数据的分析与潜在信息的挖掘对于分类与决策过程起着至关重要的作用。尽管现有的方法在数据挖掘领域已经获得了巨大的成功，但是在许多的实际问题中还存在着各种各样的问题，不平衡数据便是其中一个相对比较新而且比较艰巨的挑战[[1]](#endnote-2)。

不平衡数据分类问题是指不同类别样本数目相差悬殊情况下的分类学习问题。 以二分类为例，若其中有一类（正类、多数类）的学习样本比另一类（负类、少数类）的学习样本多很多，那么就称这样的分类问题为不平衡样本分类问题[[2]](#endnote-3)。随着数据挖掘与机器学习从研究性课题逐渐变为了应用技术，以及更多的分类或预测等实际需求的出现，人们发现在某些领域，不平衡数据集往往比平衡数据集更常见，并且人们对不平衡数据集中的少数类样本的重视程度通常也高于多数类，如文本分类[[3]](#endnote-4)，欺诈识别[[4]](#endnote-5)，生物信息学[[5]](#endnote-6)等，对于这些问题，人们尝试使用传统的分类方法进行分析，不过效果并不理想。

分类是机器学习和数据挖掘中重要的知识获取手段之一。常见的分类算法如决策树、贝叶斯网络、支持向量机和神经网络等已经被广泛的应用在了各类分类问题当中。然而经典的分类算法在面对不平衡数据时通常会出现少数类样本难以召回的情况，这种现象是由于现有的分类算法大多假定面对的训练数据集是大致均衡的，既各类所包含的样本数量大致相等[[6]](#endnote-7)。在此前提下，通常经典的分类器算法的训练目标是取得最高的整体分类准确率，这也就导致了当数据集的分布情况不均匀并且多数类与少数类的分布存在边缘略微重叠时，传统分类器算法会将边缘部分的样本直接归为多数类以提高全局准确率，但是这样会大大降低了少数类的召回率，尤其面对一些实际问题如疾病监测时[[7]](#endnote-8)，在数据批量处理环节，将一位患者误诊为无病的损失要远大于将一位没有患病的人误诊。而对于之前的全局准确率这一评价标准，也因为不同类别的样本分类错误的损失有所不同而无法应用在这一类分类问题当中。所以，传统的经典分类器算法和普通的基于准确率的分类评价标准在用于不平衡数据集时，由于存在上述各种各样的明显缺陷，已经无法直接应用[[8]](#endnote-9)。

综上，随着人们对于预测与分类的要求越来越高，不平衡数据集出现的越来越多，以及传统分类方法的效果无法被接收，所以不平衡样本分类算法的研究具有着重要的理论意义和实用价值。

## 国内外研究现状

不平衡样本分类问题从被研究学者们关注至今约有十余年[[9]](#endnote-10)，期间有很多的相关研究，在这里大体可以将所有常见的解决方法其分为两大类：

a）经典不平衡分类处理方法：这一类算法的主要思想是将不平衡数据集经过一系列处理或者分类过程中采取区别对待数据集中的实例，从而将原始数据转变为一个相对的“平衡态”，进而解决不平衡问题；

b）传统分类器优化法，这一类算法并没有经典不平衡处理方法那么主流，并且没有一个普遍的处理规律，他们通常的思路是将传统分类算法的训练结果进行进一步的针对不平衡数据的调整，从而使训练结果可以适应或者处理不平衡数据集。

### 经典不平衡处理方法

根据现有的研究成果，经典的解决不平衡数据集的主要方法可以总结为以下几种：

a）原始训练数据重构：通过重采样技术改变多数类或少数类样本的数目，从而使原始数据变得相对平衡；

b）通过集成学习的方法，通过训练多个弱分类器，最后通过投票或权值相加的方法得出分类结果，通常在训练过程中通过权值改变或原始数据集分割来解决不平衡问题；

c）敏感代价学习方法，也就是改变原始数据在评判标准中的权值，通常使用敏感代价矩阵来协助计算分类损失。

重采样方法是一种思想较为简单并且相对行之有效的方法，按照采样过程重采样方法又可以分为三类，分别是过采样法，欠采样法和混合采样法[[10]](#endnote-11)。其中过采样法是通过使用现有的少数类样本点生成一些新的少数类样本，可以是简单的复制，也可是进行一些插值，比较有代表性的方法为Chaudhuri等人提出的随机过采样法[[11]](#endnote-12)，Chawla等人提出的smote法[[12]](#endnote-13)，以及一些在此基础上优化的过采样方法[[13]](#endnote-14)[[14]](#endnote-15)，然而仅仅通过原始的样本点生成新的样本可能存在一些问题，比如最常见的问题就是过拟合，并且会一定程度上的改变原始样本空间分布等等。既然有过采样法，那么很容易想到与之相反的欠采样法，欠采样法便是通过去掉一部分多数类样本点以达到数据的相对平衡态，欠采样法相对于过采样法稍微复杂并且在许多场景下也更合理一些[[15]](#endnote-16)，除了早期简单的随机欠采样法之外，又有多项研究分别设置了不同的欠采样规则，并且在相应数据集上以及应用于相应的经典分类算法均取得了一定的效果。如Zhang等人提出的基于聚类的欠采样方法[[16]](#endnote-17)，Kubat等人提出的基于距离的单边欠采样法[[17]](#endnote-18)，以及去除临近点的Tomek-link法[[18]](#endnote-19)等等。然而当多数类与少数类样本相差过于悬殊的时候，单独使用过采样方法和欠采样方法可能已经很难满足需求了，所以这时需要把两种采样方法加以结合，许多研究表明[[19]](#endnote-20)[[20]](#endnote-21)，对于某些严重不平衡的数据集，混合使用两种采样方法通常效果会好于单独使用一种方法。

集成学习方法主要分为两类，分别是训练集分割方法以及提升算法。训练集分割方法是指将多数类数据集分割为多个子集，每一个子集的样本数目与少数类样本数目相当，然后用同一种或多种不同的多个分类器分别训练，每个分类器的训练集为少数类和一个多数类子集的组合，这类方法[[21]](#endnote-22)[[22]](#endnote-23)与普通的重采样法相比的优点是使用了多个分类器投票机制，而也具有共同的缺点，便是改变了。另一类提升算法也是通过训练多个分类器通过使用权重的机制来控制分类结果，不过与分割法的区别是他每次训练的时候都是采用整个训练集，不过每次训练之后根据当前分类器的分类结果来更新训练集中每个样本对下次训练时的影响权重，其中比较有代表性的方法为bagging算法和Adaboos算法[[23]](#endnote-24)。

敏感代价方法的思想是通过人为设置分类错误的损失值来调整不平衡数据分类过程，以二分类为例可以调高正例（少数类）样本的分类错误损失比例，降低负例（多数类）的分类错误损失比例来解决不平衡问题，虽然使用敏感代价方法或者基于敏感代价方法的研究也有很多，不过在不平衡样本分类问题中，其本质和效果与重采样法并无二致。

以上方法本文中称之为经典不平衡处理方法，可以发现他们都是从不平衡数据分类问题的数据角度入手，采用直接设法改变样本数目的方法或者改变样本权重的方法，旨在一次性或逐渐降低原始的中的不平衡问题，来达到一种相对的“平衡”状态，然后在构建模型期间采用的都是传统的分类算法，并来按照普通问题进行处理。

### 传统分类器优化法

传统分类器优化算法有一个共同的特点，就是针对不平衡样本分类问题，他们不再是想经典不平衡处理方法那样针对不平衡的数据集进行一些平衡化的调整，他们的主要思想是通过修改分类器的训练过程或者分类过程来适应不平衡的数据集，既通过优化算法的训练过程来减轻不平衡分布对训练过程的影响，或者采用正常的训练思路训练模型后，通过一系列其他过程进行对模型的调整，又或是得到普通的模型之后在分类阶段采用与经典分类阶段不同的方法来解决不平衡问题。

多数的传统分类器优化法并没有一个统一的思路和方案，他们通常采用不同的原理但是殊途同归，都可以解决不平衡样本分类问题。如Cheng等人提出的基于极限学习机（Extreme Learning Machine）的边界移动算法[[24]](#endnote-25)，就是通过极限学习机把原始问题当做普通问题训练之后在采用核密度估计以及分类边界移动的方法来解决不平衡问题；Jansche等人提出的最大化F1值期望的逻辑回归训练法[[25]](#endnote-26)，则是通过采用一种在训练过程中求解F1值的期望与当前模型的关系，并通过此关系不断的迭代去优化F1值得期望，进而去生成一个可以适应不平衡样本的模型；Dembczynski等人提出的F1值优化算法的思路则是在分类阶段与传统分类算法有所不同，这种算法可以适用于各种概率模型，通过先求解当把多少个实例划分为少数类时可以获取最大的F1值期望，之后在把所有实例按照可能属于正例的概率进行排序并分类，从而可以在近似时间内完成整体分类过程[[26]](#endnote-27)。

## 主要研究内容

本文主要尝试进行不平衡样本分类算法的研究并完成一种基于最小化损失的不平衡样本分类算法的设计。

首先将通过对现有的机器学习传统方法与经典的不平衡样本分类算法进行分析，得到不平衡数据集分类时的主要问题与难点。

其次尝试对F1值优化法进行改进，由于该方法更倾向于得到了一个对空间拟合相对准确的模型，而直接对不平衡样本数据进行训练并无法得到。所以本文将采用神经网络模型，并设计一个迭代过程，在每一步通过F1值优化算法求解的当前步骤最优解，然后通过最优解与目标解之间的偏差去重新优化模型，进而得到一个相对样本空间拟合相对准确的模型。

最后，由于上一算法中最大化F1值分类过程的时间复杂度过高，每次迭代过程需要使用规模的时间进行求解。所以将再次对上一算法进行优化，从不平衡样本分类的评价标准入手，其中不平衡样本的评价标准根据不同的问题有很多种，本文中预计采用最常用的F1值作为优化目标，并采用了整体优化的思路，根据F1值的计算过程，尝试建立一种F1值的期望与训练过程中的分类器的数学关系，并利用此关系通过梯度下降的思路、神经网络模型以及反向传播算法来设计模型参数更新过程，使得可以在训练集得到最大的F1值。

## 章节结构安排

本文主要结构包含如下内容：

（1）绪论：首先简要的介绍了本课题的研究背景与意义，然后分类讨论了国内外的研究现状，并结合不平衡样本分类问题的关键点分析了各种算法的主要思想与方法，最后介绍了本文的主要研究内容与章节安排。

（2）相关理论及算法介绍：这一部分首先介绍了不平衡样本分类问题中的一些常见的评价标准，然后介绍了分类过程中求解最大F1值以及结构化支持向量机这两种算法，为第三章中用到的一些理论进行一定的铺垫。

（3）算法介绍：首先介绍介绍了基于最大化F1期望的神经网络训练算法，然后通过分析该方法主要存在的问题，并从直接改进传统分类算法的训练过程的以适应不平衡数据集的角度，提出了基于损失最小化参数学习的F1值神经网络训练法。

（4）实验结果与分析：首先对以上算法进行验证性实验来证明算法的有效性，然后与其他不平衡样本分类算法在相同数据集上分别进行对比试验以验证算法的可靠性。

最后得出结论，总结全部的工作并提出接下来的展望。

1. 相关研究内容介绍

通常，机器学习是一种通过数学模型来拟合一个样本空间的问题，由于不可能获得到样本空间中所有可能出现的点集，所以通常采用该空间中的一部分点作为模型训练的集合，对于不同算法来说，他们采用不同的训练方法，但都是为了从训练集中求解出归纳偏置从而对真实空间中没有被采集到的未知的样本进行预测或者分类[[27]](#endnote-28)。然而现有的传统的分类方法大都是以训练集整体分类错误样本个数最少为训练目标的，所以他们大都默认各个类别的样本数目大致均衡，或者处于两类相交边界部分的样本点密度相差不大。而不平衡数据集恰恰不满足这个条件，如下图2-1便是一个二维的不平衡数据集，可以看到其中存在多数类与少数类分布交叠的情况，并且交叠部分多数类样本较多，所以这一部分的少数类样本点通常会被传统分类方法判断错误，并且在评价的时候由于这一部分样本点数目很少，通常不会对传统评价方法的结果有很大影响。

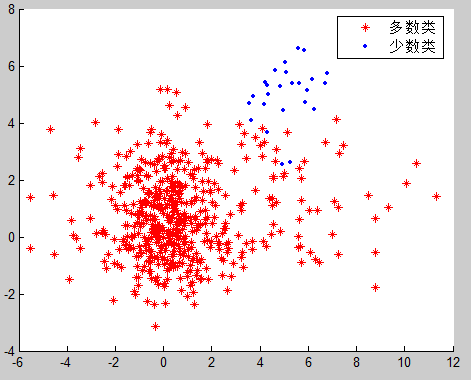


图2-1 不平衡数据集分布样例

所以无论是评价标准又或hi传统分类方法均已不能完美适用于不平衡数据集，需要有针对不平衡数据集专有的分类方法以及评价标准。

本章首先对了不平衡样本分类评价标准进行讨论，然后选取其中最常用的F1值作为本文的优化目标，最后介绍了最大化F1期望分类过程以及结构化支持向量机，并对这两种算法的流程和思想进行一定的分析和总结。

## 不平衡样本分类评价标准

不平衡样本分类问题通常为二分类问题（多数类和少数类），并且多分类的不平衡样本问题通常可以简化为多个二分类问题，所以评价标准通常是针对二分类问题所设计的。下表2-1为二分类问题的结果混淆矩阵（Confusion Matrix）。

表2-1 混淆矩阵

|  |  |  |
| --- | --- | --- |
| 真实类标 | 预测类标 | |
| 正例 | 负例 |
| 正例 | TP | FN |
| 负例 | FP | TN |

其中TP表示分类器正确预测的正例样本个数，FP表示分类器把负例预测为正例的样本个数，FN表示分类器把正例预测为负例的样本个数，TN表示分类器正确预测得到负例的样本个数，在不平衡的二分类问题中，我们通常将少数类视为正例，多数类视为负例。

对于二分类问题，无论是传统的分类问题或者是不平衡样本分类问题，他们的评价标准通常都是通过混淆矩阵中TP、FN、FP、TN四个参数进行数学计算求得的。传统的分类方法通常使用的是全局准确率作为评价标准，其计算方法如公式2-1所示。



通过上式可以看出，如果负例与正例的比例相差很悬殊，那么正例的分类正确个数TP对分子的影响很小，并且正例的总体个数TP+FP对于分母的影响也非常小，所以整体来说正例对于全局准确率的影响远小于负例，这也就是为什么此传统的分类方法不再适用于不平衡数据集分类问题，因此研究者们设计了一些其他的评价标准来评价不平衡数据集或者一些其他类型的数据集，如F1值，G-measure，A-mean，以及Roc曲线，Auc面积等，它们都是针对不同的问题被设计出来的，不过均可用于评价不平衡数据集分类效果。

由于全局准确率无法满足不平衡问题，所以最直观的思路便是将评价标准细化，所以召回率、伪阳率等评价标准应运而生。下列公式2-2至2-6为常用的细化的评价标准：

真阳率（True Positive Rate）即正例召回率，查全率：



伪阳率（False Positive Rate）：



真阴率（True Negative Rate）：



伪阴率（False Negative Rate）：



正例准确率，也是查准率：



### G-Measure与A-Mean

*G-Measure*与*A-Mean*都是通过使用*TPR*与*TNR*计算得来的，所以他们都均衡考虑了正负两类样本的分类结果[[28]](#endnote-29)[[29]](#endnote-30)。他们的计算方法如公式2-7与2-8所示：

*G-Measure*：



*A-Mean*：



由上述两个公式可以看出，如果想要得到较高*G-Measure*或*A-Mean*，需要细化评价结果中的*TPR*与*TNR*均达到较高的水平，而对于这两种评价标准，他们的区别是*G-Measure*需求两个结果更要均衡，由于其计算方法是*TPR*与*TNR*的乘积，所以如果其中一个参数较低的话会严重影响最终结果，而A-Mean相对来说没有那么的严格，由于他采取的是加和的策略，所以其中一个参数较低的话，并不会对结果影响那么严重。由于这两种评价标准分离了正例与负例的准确率，所以他们都可以用于评价不平衡数据集分类结果过，而对于这两种标准的选择需要从实际问题出发考虑。

### F1值

*F1*值为*F-Measure*或者*F-Score*的特殊情况，常用于评价机器学习相关领域内多种问题的分类结果，*F-Measure*是通过正类的准确率与召回率求得，具体求法如下公式2-9[[30]](#endnote-31)：



上式中*Recall*与*Precision*的求法如公式2-2与2-9，为人为设定的参数，从上式中可以看出，他的作用主要是设定一个放大比例，对于相对结果并无影响，当取1的时候就是我们最常用的*F1*值。从上式也可以看出，*Recall*与*Precision*的分子均为*TP*，所以正例分类正确个数对整个评分的影响最大，负例分类正确个数并不出现在表达式中，所以这一评价标准常用于评价不平衡数据集分类结果，并且通常用于正例相对重要一些的实际问题。

### Roc曲线与Auc面积

受试者工作特征曲线（receiver operating characteristic curve，简称ROC曲线），是二分类问题中一个常用的评价指标，它是以真阳率(*TPR*)为纵坐标，假阳率(*FPR*)为横坐标所绘制的一条曲线，如图2-2所示。图中曲线便是一条ROC曲线，而其意义是达到某真阳率时假阳率的值的话情况，而虚线是连接(0,0)与(1,1)的一条直线，代表完全随机过程的Roc曲线情况。所以Roc曲线只能定性的认为越接近左上角效果越好，并不能定量评价分类器的性能。为了解决这一问题，Provost等人提出了Auc面积[[31]](#endnote-32)，Auc面积是指Roc曲线与X轴以及直线X=1所围成的面积，所以这一评价标准既继承了Roc曲线的鲁棒性，又可以定量的评价分类器的结果，常用于评价分类结果，并且也可以用于评价不平衡数据集分类结果。

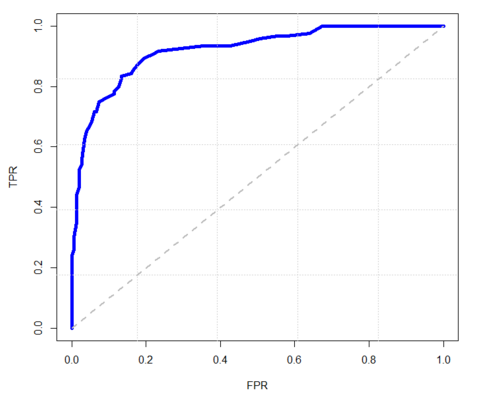


图2-2 ROC曲线示意图

以上便是细化分类标准以及各种组合不平衡数据集分类评价标准，他们各有各的特点以及最适用的问题，比如*G-measure*和*A-mean*是的思想是分离正负例并均衡考虑，而*F1*的思想是均衡考虑正例的准确率与召回率，所以对极为不平衡的分类问题*F1*评价标准可能更客观一些，而不是非常严重的不平衡数据集可能*G-measure*和*A-mean*更合理一些，由于*F1*值是研究者最认可的评价标准之一，所以本文中主要以*F1*值作为评价标准及优化目标。

## 最大化F1期望的分类过程

最大化F1期望的分类过程由Jansche提出[[32]](#endnote-33)，主要用于二分类概率模型，他的主要作用是在分类器训练过程结束之后，在分类过程中，如果待分类样本是一个待分类数据集，而不是一个单一样本的时候，该算法可以根据现有的概率模型把每个样本分为每一类的概率，求解出在该集合上的最大的分类*F1*值期望，以及可以得到这个期望的每个样本的类标。这一思想打破了传统分类算法的常用的简单分类过程(如公式2-10)，并不再是通过简单的设定一个阈值，或者是采用简单的规则来动态调整阈值，并对比输出和阈值的大小关系进行分类，而是采用了整体分类的思路，一次求出整个待分类数据集上的最佳分类结果，以达到使用分类器分类的目的。



### 问题转化过程

由公式2-9可以看出，*F1*值是通过召回率与精确率的调和平均数求得的，所以可以认为*F1*的结果只与真实类标及分类器预测类标相关，在这里我们用*z*来表示真实类标序列，用*y*来表示分类器预测类标序列，既分类器输出，他们的形式均为。所以此算法的目标是求解出的期望最大值，由于*z*是真实类标的序列，我们可以认为它是固定的，所以*F1*值的大小其实只与*y*相关，可以认为是如下公式2-11的映射关系，在后文中将简写为。



上述过程是一次对于分类器结果的评价过程，而本算法是针对于分类过程的，其主要思想是由于空间中所有可能的分类结果的概率都是固定的，当假设任意一个结果为真实的类标序列*z*时，都会对应一个的期望。根据期望的计算原理，我们可以得到值的期望如公式2-12所示，其中*y*代表的是任意一个形如的输出序列，而Pr(*y*)表示该序列在当前模型下出现的概率，虽然对于不同概率模型分类器，其表现形式可能不完全相同，但是均可以求出*y*与Pr(*y*)的映射关系，所以在这里直接简记成了Pr(*y*)的形式。对于所有的期望，其中一定有一个最大值，使其取得最大值的类标序列*z*便可以被认为是当前分类模型下最合理的输出结果。



### F1最大分类过程推理

我们假设是可以使F(*z*,)的期望达到最大的长度为待分类样本个数n的假设，所以可以表示如下的公式2-13，其中*y*，*z*如上文中介绍，均为长度为n的序列，每一位有两种可能分别为0和1，在该问题中，由于每个样本的分类概率均相互独立，所以Pr(*y*)可以通过零阶马尔可夫链求解，既公式2-14的形式，其中为分类器认为第i个样本的类标为1的概率。根据本算法对分类器概率模型的要求，我们可以认为分类器的原始输出为的形式，代表着分类器认为每个样本的类标为1的概率，这也是本最大期望求解算法的输入，而输出便是可以在训练集上获取最大分类*F1*值期望的分类类标。





从上式2-13中可以看出这是一个离散优化问题，这也是次算法解决的第一个主要难点。由于序列*z*的长度为n，每一位有两种可能，所以*z*的可能性空间为，如果想朴素的枚举出所有的可能，并分别计算出对应的期望，然后得到期望的最大值，通常情况下是不可能的。所以这里采取了内部与外部最大化求解方法，其思想如公式2-15所示，就是将可能性空间进行划分，保证这个划分包含了该空间中的所有可能，然后求解并比较每个划分中的最大值，最后得到全局的最大值。式中是*X*的一个划分，也可以写成，其中均为*X*的一个子集，并且。



在此问题中*X*为所有可能性的集合，我们将*X*划分为，其中满足公式2-16，所以原始问题2-13可以表示为公式2-17与2-18的内部外部最大值求解过程。







对于外部求解过程，其实很容易计算，只要比较每个内部解的大小，找出其中最大的即可，而内部最大值求解过程相对复杂，在这里本文省略了求解的证明过程，直接给出结论，对任意的，如果其满足充分条件2-19，其中i，k为任意两个实例（维度），那么可以推出将满足2-20。





### F值期望的求解过程

以上我们认为可以在的搜索空间内求解出使F1值获得最大期望的假设，并可以把作为该算法及分类器的最终输出。但是想求出2-13式的最终结果还需要可以在较短的时间内求解出每一个*z*所对应的*F1*值的期望，也就是求解出式2-12的结果，通过观察式2-12可以发现这也是一个复杂度为的问题，但是我们可以根据离散函数的一些特性，以及上文中的划分过程总结为以下求解过程。

当求解式2-17中的时，m为定值，也就是混淆矩阵中的*TP+FN*为定值，我们上文中已经分析过，求解F1值只需要混淆矩阵中的三个值，*TP、FP*及*TN*，当*TP+FN*固定的时候，我们只需要再确定上述三个值中的两个便可以求解出*F1*值的大小，所以F1值的可能性空间直接被缩减到了。那么我们可以将公式2-12有效的转化为公式2-21的形式。



由于*TP+FN*固定，所以FP，TP，TN中再确定两个关系或值的话便可以求解出F1值的大小，该算法中分别设定*FP+TP*与*TP*为另外两个关系和值，分别用*A*与*B*代表，那么可以认为{*A*，*B*}代表一个唯一状态，对应了一个*F1*值。所以求出所有可能的状态{*A*，*B*}出现的概率以及其对应的*F1*值便可求出整体期望。

为了求解所有状态{*A*，*B*}出现的概率，这里当把所有的待分类样本作为一个序列输入，那么便可以通过状态转移自动机来求解出所有可能的状态以及其对应的，自动机如下图2-3。

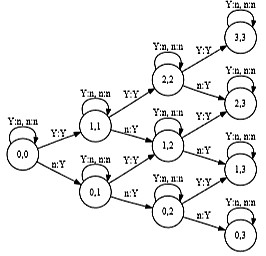


图2-3 状态转移自动机

如上图所示，当处于任意一状态时，如果预测类标为n，则下一状态还为当前状态，而当预测为Y的时候，如果的类标也恰好为Y，则状态将由{*A*，*B*}变成{*A+*1，*B+*1}，否则变为{*A*，*B+*1}。而状态概率通过当前概率乘以和来计算，最后当所有样本都通过该自动机后，所有的状态对应着一个F1值，和一个该状态的概率，将他们相乘并求和便可以得到该下的F1期望，具体求解期望的算法流程如下算法2-1。

算法2-1 F1值期望求解过程

|  |
| --- |
| 输入：分类器认为每个样本输入正例的概率*P*：，需要求解F1值期望的序列  输出：F1值期望 |
| 1. Def getMaxF1(*P*,)： 2. 初始化状态概率集合，加入初始状态 3. For i = 1 To n 4. For each State 5. If 7. Else 9. End if 11. End For 12. End For 14. For each State 15. +=*P\** 16. End For 17. Return |

至此该算法完全解决两个有于规模而带来的时间复杂度过高的问题，其整体算法如下算法2-2所示。

上文中已经提到，该算法的前提是要先获取该空间相对较为准确的概率模型，如果模型偏离真实空间分布过大，也就是分类器的输出与实际概率相差过大的话，即使通过该算法也可能无法得到很高的F1值。不幸的是不平衡样本分类问题恰恰可能造成该状况，由于不平衡数据集的分布对传统分类器的影响过大，已经超过了该算法后期优化的能力，从而造成效果降低，该问题可以使用本文第三章中的迭代算法加以改善。

算法2-2 最大化F1值期望的分类算法

|  |
| --- |
| 输入：分类器认为每个样本输入正例的概率*P*：  输出：使F1值期望达到最大的序列，既分类输出 |
| 1. 对序列*P*进行排序 2. 初始化变量， 3. For i = 1 To n 4. 构造序列*zi*满足式2-19，并且保证*zi*中为”1”的项的数目为i个 5. = getMaxF1(*P*, *zi*) 6. If ()  9. End if 10. End for 11. Return |

## 损失最小化学习

### 引言

损失最小化学习的思想最先由Joachims于2005年提出[[33]](#endnote-34)，并将其应用到他在前一年提出的结构化支持向量机中[[34]](#endnote-35)(Structured Support Vector Machine，SSVM)，设计了一种可以设定任意支持向量机输出与真实类标之间损失关系的训练方法，该文章评获05年ICML会议(International Conference on Machine Learning)的最佳论文。

基于损失最小化学习的结构化支持向量机的核心思想是自定义损失函数，并以此函数来代替传统支持向量机中的合页损失函数。只要自定义损失函数可以衡量分类器输出与目标输出之间的误差，或者为任意一个合理的映射关系，那么此算法便可以使该损失函数达到最小值以满足不同的训练需求。不过如果损失函数过为复杂或者为离散函数的话，可能导致此算法的时间复杂度大幅上升以至于可能无法求出最终结果。

### 结构化支持向量机

结构化支持向量机的思想并不是很复杂，作者提出他主要是由于传统的支持向量机的输入通常都是向量，而面对一些图像处理、文本关系、语音识别等问题，在当时的研究情况下，可能并无法有效的将结构化输入转化为向量模型。并且当使用者把自定义的结构化模型作为输入时，传统的序列最小优化算法(Sequential minimal optimization，SMO)[[35]](#endnote-36)将无法正常的使用，所以作者提出了一套对应该种输入的训练方案。其主要思想是首先随机在假设空间中构建一个初始假的假设条件，然后通过结构化输入模型以及自定义损失函数计算在该假设条件下的输出，然后找出违反该假设条件最严重的点作为支持向量，将其加入支持向量集，最后根据新的支持向量集来缩减可能的假设空间，并重新构建新的假设的。不断地进行以上的这个迭代过程，直到所有的点均满足假设条件时，或者相对于目标输出的误差小于松弛量的时候，便认为最后一个假设为当前训练样本的分类超平面，作为最后结构化支持向量机的训练结果。

### 最小化全局损失函数

对于现有的大多数的机器学习算法，我们通常认为训练数据集与测试数据集满足独立同分布，这个分布也被认为是真实的样本空间分布，也就是我们通过训练数据与测试数据总结出归纳偏置来对真实空间中的未知样本进行分类。而机器学习算法的训练目标便是在所有的假设空间中找到一个特殊的假设来使大小为的测试数据集达到设定的损失最小值，既公式2-22达到最小值。对于普通传统的损失函数，由于其通常认为空间中各点是相互独立的的，所以可以将式损失函数转化为2-23的形式，也就是把整体损失转化为了所有样本单独损失的加和。所以根据式2-22与2-23，并且根据训练数据与真实数据间的独立同分布假设，或者说训练数据是真实空间的采样假设，最终可以将2-22式转化为2-24，这也就是为什么通过训练数据集可以对真实空间进行训练或者拟合。并且这也是对于传统的机器学习方法，可以通过分别计算训练数据集中每一个点的损失，然后通过加和求解总体训练数据集的损失的原因。并且对于大多数的训练方法，一定是使式2-22达到最小值，也就是希望分类模型的输出与目标输出间的偏差变小，也就是对于不同算法，使其损失函数达到最小。







由上述式中可以看出，普通的机器学习方法利用了同分布的假设条件，在单个样本输出与其真实值之间构建偏差损失函数进行评估，尤其在支持向量机中，之前的算法通常只能这样处理，而对于许多问题中，该类型的损失函数不能准确的表达训练目标。所以基于结构化支持向量机的特定损失函数最小化的学习方法被设计了出来，其主要功能便是对于任意的损失函数提供了一个通用模型来进行训练使其达到最小。

### 训练过程介绍

对于结构化支持向量机的训练方法，起初每次迭代采用的是复杂的的搜索过程，之后作者又在多篇文章中对该算法进行优化和补充[[36]](#endnote-37)[[37]](#endnote-38)[[38]](#endnote-39)，在此我们不对训练过程本身做过多的详尽介绍，后文中我们将使用代表其求解过程。

最小化全局损失函数训练过程主要是采用不断的迭代，并通过最小化全局损失函数的额外条件来进一步对该过程进行限制，从而找到一个同时满足两个条件的可行解。延续的原始分类思想，表述为公式2-25所示，既分类过程中找到当前参数下可以使目标结果最大的序列作为输出。而最小化全局损失的需要满足式2-26取得最小值，或在训练集上取的最小期望。





经过带入以及训练样本对空间的替换以及加入松弛项，正则项之后，最后可以化为式2-27的形式，也就是该算法的求解问题。对于该问题，作者采用以下算法2-3进行解决。



算法2-3 结构化支持向量机求解最小化全局损失

|  |
| --- |
| 输入：样本集合，松弛量与正则量比例系数C，类标空间，松弛量训练调节参数  输出：支持向量机参数 |
| 2. Do  5. If  那么 7. 在集合上的结果 8. End if 9. Until不再变化 10. Return |

对于该算法，主要有两部耗时分别是第三行和第七行，其中第七行可以通过之前介绍的几篇文献中提到的算法进行优化，而第三行需要根据实际的损失函数进行参考，对于F1值损失函数，由于其本身的特性，该算法中通过类似于2.2节中思路将其降低为空间内进行求解。

然而该算法由于使用了支持向量机的思路而无法使用SMO算法，虽然在文献中提到的算法可以在线性时间内对线性核的支持向量机进行训练，然而众所周知线性核主要适用于高纬度并且低秩的输入空间，通常纬度与样本数相当的时候效果较好，对于一些普通的输入，通常要使用RBF核进行优化。然而对该算法使用RBF核时经常出现由于复杂度过高而无法训练出结果的情况，所以此算法在信息检索等领域表现更好一些。

## 本章小结

本章首先介绍了不平衡样本分类评价标准，以及各种评价标准的常用环境以及优劣情况。然后介绍了直接将评价标准参与到算法设计中的思路，介绍了两种直接作用于分类问题评价标准的算法，他们分别可以在训练阶段以及分类阶段对模型进行优化或者改善，从而直接的适应特殊的评价标准。虽然说两种算法并不能完美地解决不平衡数据集分类问题，但是他们的思想可以说都是可以从根本上解决不平衡样本数据集的分类难问题，并且巧妙的躲避开了经典算法中改变样本空间分布或训练过程受原始数据影响严重等问题。

1. 最大化F1值的不平衡样本数据集训练算法

## 引言

根据上文所述，传统分类算法是无法直接用于不平衡样本分类问题的，而对于经典的不平衡样本分类算法，其通产的思路都很直观，大都是直接作用于数据集上，无论是直接改变数据集中样本的权值分布，还是通过某种特殊的方法来增删数据集中的各类的样本，最后都是为了让原始的不平衡数据集可以达到一个相对平衡的状态，最后便可以应用传统的分类方法来解决原始问题。

众所周知，传统的分类方法其根本思想都是通过对训练样本空间进行一致性假设的归纳，既归纳偏置，从而可以对整个样本空间中未出现过的样本进行预测[[39]](#endnote-40)，对于不同的分类器来说，只不过是他们采用的VC维(Vapnik-Chervonenkis Dimension)[[40]](#endnote-41)函数，以及偏置条件不同。

不幸的是经典不平衡分类算法恰好与传统机器学习的思想相左，由于经典不平衡分类算法大多是通过改变原始样本分布来解决不平衡问题的，而传统机器学习算法是建立在训练数据与真实数据是独立同分布的基础上的，如果改变了训练数据的分布，完全可能对结果产生未知的影响。虽然这一种影响对于一些判别模型的影响还达不到可以破坏整个模型效果的级别，但是可以肯定的是，这一定会影响模型的决策过程。而且这种影响对真实空间的最终预测过程是偏向好的方向还是坏的方向通常是无法判断的，尤其是对一些有随机过程参与的算法，如SMOAT算法等，其改变分布的情况更为严重，即使使用交叉验证的方法来训练模型，也会造成多次试验中平均的分类准确率或分类F1值都会波动相对较大。所以为了解决上述问题，本文设计了一种通过以评价标准为目标从而直接对模型训练的方法，并且在多数数据集上取得了与经典不平衡分类算法相当或是更好的结果。

## 算法基本思想

本小节阐述了最大化F1值训练法的基本思想以及该方法为何可以应用于不平衡数据集分类问题。

首先假设当前的数据集是一维不平衡数据集，包含多数类与少数类两种样本，其概率密度曲线如下图3-1所示，并假设多数类与少数类的样本比例为，其中。显然以传统分类器的基本思想既以最大化全局准确率为最终训练目标，对于两类的边界部分，即使其概率密度大小相似，但是因为多数类与少数类样本基数不同，所以边界部分的多数类样本个数会远多于少数类的样本个数，最终分类界线极有可能在图中线b的位置附近，处于偏向少数类的一侧。

而经典的不平衡数据集分类算法的思想是直接采用某种方法强行的缩小多数类与少数类之间的样本比值，通常会使两类的样本点数目达到相同或十分接近，然后应用传统分类算法时，如果数据集的概率密度曲线还是如图3-1所示，由于已经不存在两类样本基数不同这一问题，所以使全局分类准确率最高的分类界限应该是图中的线a。这条线是以两类概率密度曲线交点的横坐标作为分界阈值，分界线左边的少数类与右边的多数类为错分样本，由面积法易证，此时分类错误样本数最小。

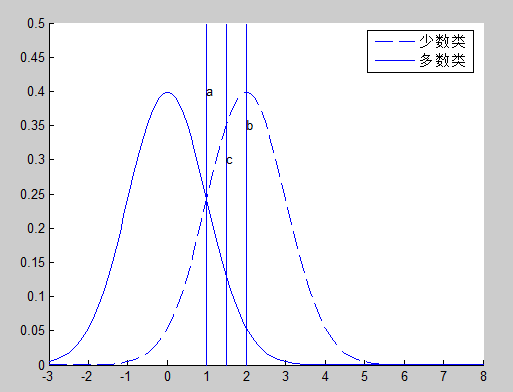


图3-1 数据概率密度曲线示意图

然而由于样本空间的改变，我们只能认为该解（线a）是当前改变后的样本空间中最佳分类线，并不能确定该分界点是否也是原数据集上的最佳点，对于不平衡样本分类问题，在原始数据集上我们通常采用2.1节中介绍的评价标准来评价最终分类效果的好坏，而经典不平衡样本分类方法通常只能使原始空间的不平衡分类评价指标有所上升，即使不采用正则项降低过拟合，通常也无法证明该解一定是该原始样本空间中的最优解或者一个极优解。

对于图3-1的数据集来说，由于它是一维数据，如果比例n固定的话，我们可以知道使该数据集的最终F1值达到最大的分界点一定只有一个，并且是可解的。本文提出的算法便是基于该思路，跳过了经典算法中改变原始数据集分布的这一过程，直接寻找可以使原始数据集最终分类结果达到最大F1值的分类界限。然而对于一维问题，其求解方法显然很容易，而对于超过一维的问题并且需要基于传统分类算法设计求解该归纳偏置的时候，该问题的规模将指数上升，因此本课题设计了基于第二章介绍算法思路的解决方法来解决此问题。

## 基于最大化F1值分类过程的神经网络

### 算法思想与概述

本算法基于传统的神经网络模型以及2.2节中介绍的最大化F1值期望的分类过程两个算法。这里选用神经网络模型主要是由于其分类函数的VC维是与神经网络的参数个数也就是模型结构是相关的[[41]](#endnote-42)，完全可以通过问题规模来进行调整，而对于普通的线性分类模型来说，其VC维固定为样本维度加一，对于一些维度较低，样本数量较多并且分布较为复杂的数据集来说可能无法完美解决分类问题。而采取其他非线性模型，如支持向量机核函数的话，通常训练时间会高于神经网络模型很多，甚至可能出现无法求解的情况。并且更为主要的一点是人工神经网络的训练本身就是一个迭代的过程，其间包含一步分类过程，对于不平衡数据集来说，产生问题的环节恰为这一个分类以及评价当前分类器分类结果的过程，可以设计一种算法对这一环节进行了针对于不平衡数据集的调整，从而缩小数据的不平衡性对于整个训练过程的影响。综上两点，本算法优先考虑使用人工神经网络最为基础分类模型。

根据上节中阐述的算法原始思想，为了使神经网络模型的训练结果可以使训练集分类结果与目标分类结果之间达到最高的F1值效果，本算法结合了最大化F1值的分类过程。上文中已经提到过，最大化F1分类过程需要一个已经在样本空间中训练好归纳偏置的分类器，也就是可以求出每个样本属于每一类相对较为准确的概率。但是直接使用不平衡数据集训练得到的神经网络不满足该要求，所以本算法对传统神经网络模型做了针对于不平衡数据集的调整，通过变换传统神经网络训练过程中的分类过程，并重新设计训练过程中分类结果的评价标准以及参数更新策略，从而适应不平衡数据集。最终本算法在多数不平衡数据集上取得了不错的分类结果，通常可以达到一个与结构化支持向量机以及经典不平衡分类算法相当或略佳的结果。

### 最大化F1值分类过程的神经网络

本算法选取普通的三层神经网络，既输入层，输出层以及单隐藏层作为整体神经网络结构，如图3-2所示，隐藏层节点数目通常根据问题规模选择或者采用缩紧策略进行调整。假设数据集的维度为n，并且神经网络隐藏层节点数目为m，所以对于这一神经网络，需要训练的参数为输入层到隐藏层的连接系数以及隐藏层到输出层的连接系数，对于每一个样本的输出，其分类器输出为式3-1至3-4的形式，而由于sigmod函数的良好0-1映射关系以及根据广义线性模型的原理，我们可以将分类器对于每个样本的输出与该样本属于正类的概率之间建立形如式3-5的关系，从而使传统意义上与概率无关的神经网络模型满足最大化F1值分类过程的概率模型要求。











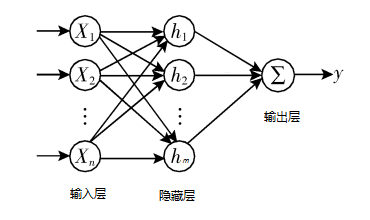


图3-2 神经网络结构示意图

而上文中已经提到采用传统的神经网络训练方法在不平衡数据集上进行训练的时候，得到的输出假设空间是偏离真实空间的，所以如何使输出空间不断的接近真实空间，或者说如何使分类器的输出与真实的不断的接近，才是这一算法的本质作用。

我们知道神经网络训练的思路都是采用一个迭代的过程，对于经典的神经网络模型，如果采用反向传播算法，通常损失函数都设置下式3-6与3-7的形式，可以看到3-7式中包含了一步求解输出的过程，之后针对所有样本的输出与期望输出的间误差，通过梯度来更新神经网络参数，从而使每一个样本误差都有减小的趋势，最终整体损失不断的缩小。而针对不平衡数据集时，根据前文中的理论，不该再以每个样本的输出与其期望输出的误差加和作为训练目标，而是应该让模型的输出与目标输出间可以达到最大的F1值，所以可以通过引入2.2节中的算法来解决这一问题。





对于2.2节中介绍的算法，我们已知该算法可以在可接受的时间内求解出当前假设空间下可以使F1值期望达到最大的分类解。所以可以利用这一算法对传统神经网络中训练过程中的式3-7中的进行进一步的调整，使评价过程以及更新过程适应不平衡数据集。由于该算法总是可以得到当前参数下的最优解，而对于当前最优解来说可能与全局最优解还具有一定的差距。其中的差距可能是由于其数据集本身不可分造成的，也可能是由于正则项造成的，还有一点便是可能由于模型求解的概率不准造成的，对于前两项，他们已经被数据集或模型参数所固定，无法大幅优化，而对于第三项来说完全可以通过计算当前输出类标与目标输出类标间的误差以及每个样本的误差贡献来对总体误差进行缩小，使最终F1值达到全局最优或局部极优，最终收敛于一个可承受的误差范围或者收敛于一个无法继续缩小的误差解。

然而使用2.2节中的算法带来的问题是会把原本的从连续函数变成了离散的，已经无法正常的使用反向传播算法求解参数的更新量。所以我们在这里将定量的损失函数贡献转化为了定性或者半定量的损失函数贡献进行求解。对于最大化F1值期望分类过程的，其输出为一个n维的由0与1组成的向量，既。我们这里可以参照支持向量机的合页损失函数的思想，如果单个样本的分类器输出达到了某一条件时，就认为这一个样本的损失为0，而没达到这一条件的时候才认为这个样本拥有一定的损失。在本问题中，我们认为神经网络的输出经过了最大化F1值分类过程后，如果某一样本最终输出与目标输出相同，那么该样本在当前状态下的模型下是没有任何损失的，而最终输出与目标输出不相同，那么就正常采用来计算损失，这里之所以不采用0与1计算损失是因为该输出是最大化F1值分类过程的输出，由于这一过程的特殊性，无法定量的求解出该步骤对分类器输出的直接影响，所以损失的求解过程继续采用正常的误差平方求解方法。根据上述描述，可以将3-7式的样本损失更新为下式3-8的形式，其中MFC代表最大化F1值分类过程(Maximize F1 classification)，其输入是所有样本的分类器输出，所以式中代表经过MFC后第i个样本的输出，可见该式并不影响普通的反向传播算法更新参数。



最终，最大化F1值分类过程的神经网络算法的全过程如下算法3-1所示。

算法3-1 最大化F1值分类过程的神经网络

|  |
| --- |
| 输入：学习率，最大迭代次数，样本集合，隐藏节点数目，目标F1值  输出：输入—隐藏层连接系数矩阵，隐藏—输出层连接系数向量 |
| 1. 对与进行初始化，每一个分量的范围为(-0.1,0.1) 2. ,, 3. For i = 1 To 4. 对样本集合求解 5. 采用MFC过程求解当前输出，并求解当前模型下F1值 6. If(>) Return 当前, 7. Else if (>) , 8. End if 9. 根据损失3-8采取反向传播算法更新与 10. End For 11. Return , |

### 算法分析

由前两节的描述我们可以知道，神经网络模型训练过程分为两步分别是使用当前模型对训练数据进行分类操作以及通过分类结果对之前模型优化，并不断迭代这一过程。对于分类过程中，模型参数W(这里W代表所有网络层之间的所有关系系数)唯一对应着训练集的最优F1值期望，并且唯一对应着一个分类解和最终训练集上的F1值，如关系3-9所示。因此式中第二部分与第四部分建立了联系，而根据上文中的训练过程可以确认当效果较差时，由于分类错误占据了误差函数主导，所以分类模型一定是向一个较好的方向发展，而当效果较好的时候根据3-9式可以发现其第二项是第四项F1值期望，并且其可能性空间为，是一个十分大的样本空间，所以可以认为第四项的结果通常会在第二项的结果附近，并且通常情况下越接近期望的结果出现的概率越高。所以本算法中即使我们在优化过程中主要优化的是第二项，每次迭代都使第二项达到一个更高的值，其最终也会推动训练集的F1值达到更高，或者有更大的概率获取更高的值。



然而对于此算法的原理与细节进一步分析，可以发现其中还存在两点有待解决的问题：

A）算法迭代过程其实并没有将与建立一个实际的数值关系，而还是通过每个样本的最终结果与目标结果间的误差来迭代优化问题。这里仔细分析此迭代过程可以发现其实其根本原理是无视掉高置信率的样本的分类损失，换言之也就是提高边界样本的分类错误损失，并且通过对边界样本的分类情况不断迭代来提高获取全局最高F1值的可能性。其实这里与经典的不平衡样本分类算法在某种程度上可以认为是不谋而合，但是其区别是经典不平衡分类算法通常采用启发式的方法或者聚类等方法来确认数据集中每个样本在训练过程中的地位，而在本节的算法中，具体哪些样本为高置信率样本，哪些样本为边界样本，哪些样本会被分类正确，哪些样本会被分类错误都是通过这一过程来确认的。虽然是一个完整的自调节过程，但是终究会因为过程与无法建立具体数值关系而在训练后期损失掉一部分样本对模型的影响。

B）由于这一过程的时间复杂度为，当n稍微大一些的时候，每次迭代过程都将耗费巨量时间，虽然神经网络模型可以通过batch等方法[[42]](#endnote-43)来一定程度上降低每次训练的样本空间，但由于本问题中主要采取的是使整体F1值最大的求解思路所以这一类算法有悖于本算法的核心思想，导致这一问题也很难得以完美的解决。

## 最小化损失学习神经网络

本节介绍了使用基于最小化损失学习的神经网络来解决不平衡样本分类问题的方案，这一算法的初始思路也是如同3.1节中介绍的思路，也是在假设空间中寻找一个特殊的假设来使当前的训练空间达到最高的F1值而不是传统算法中的最小分类误差，以此来解决不平衡问题。而为了解决上一算法中3.2.3节末尾阐述的两个问题，本节中的算法避开了产生问题的这一过程，而是通过设计当前训练空间的F1值期望与之间的直接近似关系，然后通过此关系，直接用F1值的近似期望来求解梯度，进而迭代更新神经网络的权值，使训练样本空间有更大的概率获取更高的F1值。

### 最小化全局损失学习与算法基本思想

传统机器学习分类方法通常都会涉及损失函数以及缩小损失的过程，对于大部分传统机器学习方法，他们的训练思想通常是利用数据集的独立同分布性，通过构建单个样本的输出与目标输出间的误差函数，并将所有样本的误差相加来作为训练集的总体损失，通常为下式3-10的形式。最理想的损失函数为0-1损失函数，既如果输出与目标输出属于同类别那么损失为0，不同则为1，如式3-11所示。





由于0-1损失函数是一个离散函数，通常很难对其进行优化，所以不同的机器学习分类算法通过不同的思路重新构建了损失函数，他们通常是采用连续的或可以通过其他特定方法优化的函数，如逻辑回归的对数损失，支持向量机的合页损失，Adaboost的指数损失等等，他们的图像如下图3-3所示。可以看出这些损失函数均是0-1损失的不同上界，优化这些损失函数的时候也会同时使训练集的0-1损失降低以达到训练的目的。

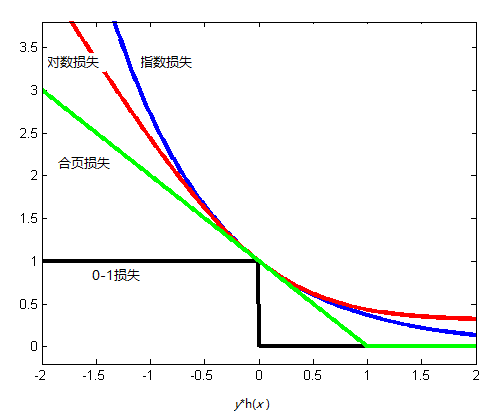


图3-3 常见损失函数图像

上文中已经提到过，对于不平衡样本分类问题，由于两类样本的数目相差悬殊，所有训练数据0-1损失的平均值或者加和已经无法有效的代表整个训练集的损失。最好的方式便是通过不平衡数据集分类评价标准来构造一个合适的损失函数用来在训练过程中使用。为了采用不平衡样本分类评价标准，需要放弃传统分类方法中利用所有样本相互独立这一特性解决问题的思想。需要将单一的样本与单一的输出间如式3-12的这种假设形式转化为式3-13的形式，在真实训练问题中我们使用训练样本的特征集合和目标输出集合来求解问题，所以式3-13中的训练空间为，目标空间为，假设*h*变为了整体假设，对应了所有样本输入以及所有的分类器输出。而整体损失由每个样本损失加和的式3-10的形式转化为了式3-14的形式，求假设3-13便等价为求解式3-15的结果。至此便将传统的机器学习分类算法利用独立性求解损失的思路转换为最小化全局损失学习的思路，以此来通过不平衡数据集分类评价标准建立损失，适应不平衡样本分类问题。









### 损失函数的设计

本算法中选取了F1这一最经典的复合评价标准作为优化目标，所以可以将损失函数设置为(1-F1)值，对于神经网络的训练，损失值最小和F1值最大其实是一样的，只要将梯度前的符号进行改变即可，所以这里将最小化损失的概念扩充到了最大化目标函数，既将3-15式转化为3-16式的形式。



对于本节中的神经网络模型，我们依然使用如式3-17中所示传统神经网络的sgn函数作为最终分类标准，对于整个训练集上上的输出我们用来表示，而目标输出依然用来表示。为了用与表示最终的F1值，我们首先需要求出召回率(Recall)与精确率(precision)，根据观察混淆矩阵与、之间的关系，可以看出混淆矩阵中的参数*TP*满足公式3-18，并且可以将公式2-3转化为公式3-19的形式，而式2-6便可以变为式3-20的形式。所以最终的F1值可以表示为式3-21的形式。











然而观察公式3-21我们可以发现因为与都是0与1组成的序列，虽然可以通过与表示出整个训练集的训练目标，但是由于求解的过程中包含了一步的操作，所以该F1值仍然是离散的，并无法与我们的神经网络输出以及各个节点间的连接层建立直接的数值联系。在结构化神经网络中求解任意目标函数时采用的是空间遍历以及双重优化的思路，也因此需要耗费巨量的时间，而在本文的算法中，我们放弃了这一思路，而是采用神经网络输出层sigmod函数特有的式3-5的性质来将与建立起联系。

本算法利用神经网络训练过程是采取先将使用当前状态的网络进行分类，然后求解损失并对此损失进行优化已达到下一更优状态这一思想，将训练中的评价过程进行变换，不再使用当前状态下神经网络的具体分类结果，而是采用当前输出的形如式3-5的概率求解损失的期望值，并对期望值进行优化，这样既可以建立输出和参数与目标间的直接联系，也可以通过优化期望值来增加目标获取更高值的概率，这样不失训练的意义。

然而建立期望联系我们并不能采用2.2.3节中的精确求解过程，首先是因为精确求解需要耗费的时间，这有悖于本节设计快速算法的初衷，其次可以看到2.2.3节中介绍的算法最后求解出的状态—概率空间已经无法存储每一样本对其的贡献，也就是无法直接建立期望与样本之间的关系，所以也无法直接建立期望与神经网络权值之间的关系，为了解决这两个主要问题我们采用下式3-22中的近似关系。



由于有期望与协方差的关系3-23，所以对于式3-22中的分子分母分别取期望，分子的期望便如式中右侧的分子的形式，而分母的期望由于后续反向传播算法中希望可以对有所样本都可以带入，而不是对于一部分样本会舍去分类器输出，所以继续采用的平方的形式而不是直接期望的形式，但是即使取平方由于最终将会接近于0或1，所以其整体值是接近的。而对于式3-23中的后一项两个变量的协方差，我们可以发现*X*与*Y*均为，他们分别在分子与分母，所以*X*与*Y*的变化趋势相反，根绝协方差的性质，可以认为协方差将一直0，所以有关系3-24，当算法不断优化右方的近似值的同时，由于期望值为近似值的上界并且相对接近，所以也可以不断使期望扩大，收敛于一个全局最优解或者局部极优解，已达到训练目的。





综上，我们便将原始的神经网络训练目标3-12完全转变为了式3-13的形式，并根据不平衡样本分类评价标准中常用的F1值进行了目标函数的设计，然后将离散函数近似为一个连续函数，使该函数可以完美的与每个训练集样本输出建立联系，进而与神经网络参数建立联系，并且证明了神经网络的训练过程中可以在优化该近似函数的同时优化整个训练集上的最终F1值，满足最初的设计思想。

### 训练过程

上文中已将介绍了本算法解决不平衡问题的特定目标函数最大化的思想，并利用不平衡样本分类的评价标准构建了应用于训练集的关于整体F1值的目标函数，本节将介绍使用该目标函数的神经网络权值具体更新方案以及整个算法的算法流程。

对于神经网络模型，其最常用的有效的权值更新策略便是反向传播算法，由于本算法的最终结果将目标函数训练到最大值所以，其更新过程如式3-24与3-25所示。其中代表了学习率，其大小影响了神经网络的收敛速度以及收敛精度，偶尔也可能会影响到最终收敛于哪一个极优解附近。其设置方案通常有人为设定法也有退火法既设定一个相对较大的初值并随着训练过程不断的缩小。





所以为了求解出每一个权值的更新量，我们需要求解出整个目标函数F1对于每一个节点中每一个参数的偏微分，其求法如下式3-26。其中代表节点的第个权值，代表节点的内积结果既通过函数(式3-2)前的结果，而代表节点对应第个权值的输入，这一项在训练过程中永远是已知第，只要通过当前状态的节点权值和输入便可以求出。接下来要求的便是式3-26结果中的前半部分既最终输出对于每个节点输出的偏微分。



为了求解各个节点的偏微分，我们需要对所有节点分为两类，一类为输出节点，另一类为隐藏层节点，由于前文中介绍过的最终目标函数是直接与输出建立了关系，所以输出节点可以直接通过当前状态下的目标函数值求解出偏微分，而对于隐藏层节点，则需要通过该节点的下游节点来求解偏微分，这也是方向传播算法链式法则的主要思想。

对于输出节点，我们采用链式法则，用函数将所求分解为式3-27的形式。其中为经过函数处理后的结果，既式3-22中的的一个分量，所以式3-27中的最后一项可以通过3-28求解，而前一项可以直接求导，其求解过程如式3-29，为了简洁该式中省略了上一节目标函数前面的常数系数，这对优化过程没有影响。







对于隐藏节点更新量，这里并没有办法直接使用目标函数求解，所以要根据每个隐藏节点的下游节点来更新其连接参数，其更新方案如式3-30所示。由于本算法为二分类问题，所以输出节点也就是只有一个。并且将求解输出节点的方法以及函数的导数带入等式3-30中之后，便可以得到式3-31的形式，其中为输出节点的偏微分，为节点的第个分量，也就是输出节点与第个隐藏节点之间的连接权值。





以上便是针对不平衡数据集的最小化损失学习算法的原始思想推演过程以及最终的训练时更新权值的方案，其算法过程如下算法3-2所示。

算法3-2 最小化损失神经网络

|  |
| --- |
| 输入：学习率，最大迭代次数，样本集合，隐藏节点数目，目标F1值  输出：输入—隐藏层连接系数矩阵，隐藏—输出层连接系数向量 |
| 1. 对与进行初始化，每一个分量的范围为(-0.1,0.1) 2. ,, 3. For i = 1 To 4. 对样本集合求解，并求解当前F1值 5. If(>) 6. Return 当前, 7. Else if (>) 8. , 9. End if 10. 根据式3-28与3-29更新，根据式3-30与3-31更新 11. End For 12. Return , |

## 本章小结

本章为了解决经典不平衡数据集分类算法中一些不合理的设计过程，设计了两种在训练过程中使训练结果与目标输出间F1评价不断升高的算法。首先介绍了以F1值为训练目标对于解决不平衡数据集训练问题的合理性，以及其与经典不平衡分类算法的异同；然后介绍了一种基于最大化F1值分类过程的神经网络算法，通过将传统神经网络训练过程中的传统分类函数替换为最大化F1值分类过程，然后通过替换后的输出计算损失来使损失更为贴近不平衡数据集的损失，最终达到训练集上F1值不断提升的效果；最后由于该算法中存在时间复杂度过高等难以解决的问题，所以设计了一种基于最小化损失学习思想的神经网络，直接求解分类器训练中的F1值期望，并找到近似下界，然后通过优化下界来优化F1值的，最终使训练过程可以求解出使训练数据集获取极高的F1值的假设，以解决不平衡数据集训练的问题。

1. 实验结果与分析

## 实验环境及数据集介绍

### 实验环境介绍

本章实验在PC机上完成，操作系统为Win10 x64企业版，处理器为Intel i7 4710MQ。主算法编译环境采用jdk 1.8.0，运行环境为jre 1.8.0，编译器为IntelliJ Idea 2016.1.3。比试验以及实验预处理时用到了怀卡托智能分析环境(Waikato Environment for Knowledge Analysis,WEKA)，实验数据格式转换是，代码编译环境为GCC4.6，编译器为Code::Blocks 13.12。

### 实验数据预处理

由于本课题主要目的是针对算法的研究，为了单纯的通过原始数据集来验证算法的有效性，所以不对数据集做去除噪声以及特征提取等相对针对数据集的预处理，本节中的数据预处理主要包含类标转换、缺失数据以及进行非数字化特征删除处理。

* 1. 类标转换：本章中的实验数据主要采用的是UCI机器学习公认验证数据集，由于这些数据集通常都是真实数据，所以多数都是包含了多类别数据的，而本算法主要是针对二分类不平衡数据集，所以我们将数据集中的多类问题人工划分为两类，并且保证实际问题中相似的类别或者相对分布可能靠近的类别被分为同一类，并且要保证数据划分之后的相对不平衡性，最终把样本数目较多的一类作为多数类，样本数目较少的一类作为少数类。
  2. 缺失数据处理：这里由于UCI数据集都是真实数据，对于有一些数据集，存在部分样本的部分属性缺失的情况，我们选取的数据集一定不是缺失十分严重的那一类数据集，在这里对多数类缺失与少数类缺失采取不同的处理方式，对于多数类由于样本相对较多，所以如果存在属性缺失的情况那么直接删除该样本，而对于少数类来说，由于样本比较稀有，所以对于属性缺失的情况则遍历其余少数类样本，通过欧氏距离来寻找最近的样本点，并用这一最近的样本点的属性来代替缺失部分属性。
  3. 非数字化特征删除：对于本文中采用的神经网络模型，虽然可以先将离散的特征数字化，然后在进行训练，但是可能数字化的规则对于实验结果产生扰动，所以对于属性相对较多并且非数字化特征较少的那些数据集来说，直接将数据集中的那些特征进行删除，而对于非数字化特征较多的那些数据集，在本章试验中，我们将放弃使用它们。

### 实验数据集介绍

本章中的实验数据集均来自于UCI机器学习数据集，对于数据集的选取过程，主要是选取那些在其他不平衡数据集分类算法研究中出现过的数据集，共以下n个数据集，经过4.1.2节中的处理之后，其最终参数如下表4-1所示。

表4-1 数据集参数表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据集 | 样本个数 | 少数类比例 | 属性个数 |
| 1 | YEAST | 1484 | 12.60% | 8 |
| 2 | Abalone | 4177 | 8.02% | 8 |
| 3 | Glass | 214 | 23.83% | 10 |
| 4 | Breast Canser | 699 | 34.50% | 9 |
| 5 | Vehicle Silhouettes | 946 | 23.43% | 18 |
| 6 | Haberman | 305 | 26.47% | 3 |
| 7 | Ecoli | 335 | 2.69% | 7 |
| 8 | Credit | 30000 | 22.12% | 24 |

## 算法基础参数设置

对于本文中介绍的两种算法，它们的基础均为传统机器学习方法中的神经网络模型，所以在本章试验中对于两种算法的参数设置类似。首先上文中提到的目标F1值，在本章中由于数据集不同，所以均不作特殊设置，试验中默认为1，对于上限迭代次数，两种算法均设置为一万次，不过这样一定会使在有些数据集上的迭代次数过多，早已达到了最优解还在继续迭代，所以这里设置收敛规则，对于基于最大化F1值分类过程的神经网络，如果训练过程中F1值下降，或涨幅低于0.001超过3次，对于最小化损失学习神经网络，如果超过10次，那么就直接停止训练过程，获取模型结果。

## 算法有效性验证试验数据

本节主要通过实验来证明第三章中阐述的两种算法在不平衡分类问题上的有效性以及性能方面的展示。由于本文第三章中的算法均是以F1值为最终训练目标，所以主要用来进行评价的评价标准为F1值，A-Mean，G-Measure以及通用的全局分类准确率，既式2-9，式2-8与式2-1。为了与第三章中的算法形成对比以证明算法的有效性，本章中选取传统神经网络模型(ANN)，传统支持向量机模型(SVM)，经典不平衡分类方法中的过采样方法SMOTE算法，经典提升算法AdaBoost算法，敏感代价学习算法以及结构化支持向量机六种算法作为进行对比的算法。

### 算法分类结果对比试验

4.3.1.1 与传统机器学习方法的拟合能力对比

首先进行的是第三章中两种算法与传统机器学习方法的对比，第一次试验采取的是训练数据集以及测试数据集采用同一个数据集的策略，主要是为了验证算法对数据集空间分布的拟合能力，其结果如下表4-2。

表4-2 与传统机器学习方法对比实验结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 评价标准 | | | | | | |
| 数据集 | 算法 | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 1 | SVM | 0.373 | 0.958 | 0.920 | 0.537 | 0.938 | 0.938 |
| ANN | 0.589 | 0.769 | 0.931 | 0.667 | 0.858 | 0.853 |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 2 | SVM | 0.000 | 0 | 0.920 | 0 | 0.460 | 0 |
| ANN | 0.245 | 0.626 | 0.928 | 0.352 | 0.782 | 0.766 |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 3 | SVM | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| ANN | 0.922 | 0.979 | 0.977 | 0.949 | 0.978 | 0.978 |
| MVC-ANN | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 4 | SVM | 0.996 | 0.992 | 0.996 | 0.994 | 0.995 | 0.995 |
| ANN | 0.996 | 0.972 | 0.989 | 0.984 | 0.985 | 0.985 |
| MVC-ANN | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| ML-ANN | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| 5 | SVM | 0.960 | 0.964 | 0.982 | 0.962 | 0.976 | 0.976 |
| ANN | 0.914 | 0.978 | 0.975 | 0.945 | 0.976 | 0.976 |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 6 | SVM | 0.432 | 0.921 | 0.840 | 0.588 | 0.875 | 0.873 |
| ANN | 0.556 | 0.938 | 0.873 | 0.698 | 0.899 | 0.898 |
| MVC-ANN | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 7 | SVM | 0.667 | 1.000 | 0.991 | 0.800 | 0.995 | 0.995 |
| ANN | 0.889 | 1.000 | 0.997 | 0.941 | 0.998 | 0.998 |
| MVC-ANN | 0.889 | 0.889 | 0.994 | 0.889 | 0.943 | 0.941 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 8 | SVM | 0.760 | 0.862 | 0.985 | 0.807 | 1 | 0.64 |
| ANN | 0.431 | 0.618 | 0.815 | 0.508 | 0.735 | 0.726 |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |

上表4-2中为四种算法在8个数据集上六种评价标准的数据，其中召回率，准确率以及F1值都是针对正例的也就是少数类的。其中算法MVC-ANN代表最大化F1值分类过程的神经网络，也就是3.3节中的算法，而ML-ANN代表最小化损失的神经网络(Minimize Loss-Artificial neural networks)也就是3.4节中介绍的算法，可以看到表中关于MVC-ANN算法有一部分数据集是没有结果的，这是由于该算法的时间复杂度过高，对于一些样本较多的数据集来说，每次迭代都将耗费巨量的时间，尤其是训练刚刚开始的时候，由于所有概率都趋于中间，所以算法耗费时间接近其上限，第1,3,8个数据集都包含了数千甚至数万个样本，所以无法通过得到结果，这也是上文中提到的该算法对于样本过多的数据集存在时间复杂度过高的问题。

上表中支持向量机算法的结果均是对于每一个数据集，选取了不同核函数训练之后选取了最优的核函数的结果，而神经网络的训练结果为WEKA默认的参数所得到的结果，由于表中的训练结果均采用了测试数据集与训练数据集同为同一集合的策略，所以体现的是算法对测试空间的拟合能力。可以看到对于传统分类算法效果已经较好的3,4,5这一类数据集，由于其本身结果已经很好，而算法都是存在波动的，所以本文中的两种算法在这些数据集上都可以得到相当的效果，而对于那些不平衡分布较为严重，传统分类算法受其影响较大的数据集，如1,2,6这一类数据集，可以看到本文中的两种算法均超越了传统分类方法的表现。而对于数据集8，其相对特殊，通过查看其实际意义发现这一数据集中的特征通常是离散的，而本文中的算法是默认将数值化的离散特征认为是连续的，并且该数据集中的特征无用特征较多并且特征值相差悬殊，本章最初已经提过，本算法没有对数据集进行特征选择以及归一化等特殊处理，所以其最终结果不是十分理想，由于该数据集数据特殊，所以进行了预处理之后再使用本算法，其最终的实验结果如表4-3所示，可以看到是有所提升的，所以可以认为本文中的算法对不平衡数据集的拟合是有效果的，在传统分类方法表现不好的数据集上通常是有提升的。

表4-3 数据集8经过预处理后的实验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 8 | 0.475 | 0.568 | 0.804 | 0.517 | 0.713 | 0.698 |

4.3.1.2 与不平衡分类方法的拟合能力对比

本小节将延续上小节中的多数实验设计，进行本文中的两种分类算法与几种经典的不平衡分类处理方法以及结构化支持向量机进行对比试验以验证本文中算法的有效性，其最终结果如下表4-4所示。

表4-4 与不平衡分类处理方法的对比试验结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 评价标准 | | | | | | |
| 数据集 | 算法 | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 1 | SMOTE | 0.373 | 0.958 | 0.920 | 0.537 | 0.938 | 0.938 |
| AD | 0.589 | 0.769 | 0.931 | 0.667 | 0.858 | 0.853 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 2 | SMOTE | 0.000 | 0 | 0.920 | 0 | 0.460 | 0 |
| AD | 0.245 | 0.626 | 0.928 | 0.352 | 0.782 | 0.766 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 3 | SMOTE | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| AD | 0.922 | 0.979 | 0.977 | 0.949 | 0.978 | 0.978 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 4 | SMOTE | 0.996 | 0.992 | 0.996 | 0.994 | 0.995 | 0.995 |
| AD | 0.996 | 0.972 | 0.989 | 0.984 | 0.985 | 0.985 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| ML-ANN | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| 5 | SMOTE | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| AD | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 6 | SMOTE | 0.000 | 0 | 0.759 | 0 | 0.379 | 0 |
| AD | 0.803 | 0.884 | 0.925 | 0.841 | 0.910 | 0.910 |
| SSVM |  |  |  |  |  |  |
| SCL |  |  |  |  |  |  |
| MVC-ANN | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |

续表4-4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 评价标准 | | | | | | |
| 数据集 | 算法 | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 7 | SMOTE | 0.667 | 1.000 | 0.991 | 0.800 | 0.995 | 0.995 | |
| AD | 0.889 | 1.000 | 0.997 | 0.941 | 0.998 | 0.998 | |
| SSVM |  |  |  |  |  |  | |
| SCL |  |  |  |  |  |  | |
| MVC-ANN | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 | |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 | |
| 8 | SMOTE | 0.760 | 0.862 | 0.985 | 0.807 | 1 | 0.64 | |
| AD | 0.8 | 0.769 | 0.979 | 0.784 | 1 | 0.64 | |
| SSVM |  |  |  |  |  |  | |
| SCL |  |  |  |  |  |  | |
| MVC-ANN | -- | -- | -- | -- | -- | -- | |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 | |

上表中的分别记录了SMOTE算法，AdaBoost.M1算法(AD)，结构化支持向量机算法(Structured support vector machine，SSVM)，敏感代价学习方法(Sensitive cost learning，SCL)，以及本文中的两种算法的实验效果。对于上表中的三种经典的不平衡数据集处理方法，他们都需要依赖传统机器学习算法，在本章试验中，对于这三种经典不平衡处理方法，均采用朴素贝叶斯分类器以及逻辑回归两种基分类器，表中记录数据是对于每个算法两次试验中效果相对较好的结果，选取这两种分类器的主要理由是SMOTE算法以及AdaBoost.M1算法都会提高原始算法的时间复杂度，所以为了使建模过程的耗费的时间可以令人接受，所以选取了两种结构相对简单，并且是机器学习中常用的效果较好的传统分类算法。

从上表中可以看出本文中的两种算法在多数数据集上都可以达到与经典的不平衡处理方法相当的效果，从本文的两种算法主要的优化目标F1值来看，在8个数据集上的分类效果整体上是略好于经典不平衡处理方法的。

对于经典不平衡处理方法，上文中已经讲述过，他是通过改变了原始样本空间的分布，来解决不平衡问题的，所以这一类不平衡分类方法对于数据集拟合来说，是上限很高的算法，有可能某一次的采样结果恰好使分布较为理想或者使权值改变的较为理想就会获得一个很高的评价，所以本文中的算法理论上来说是不可能全面好于经典不平衡分类算法的。

4.3.1.3 算法泛化能力对比

上文已经证明了本文中阐述的算法的对训练空间的拟合能力，也就同时证明了本文中的算法作为机器学习算法来解决不平衡数据集训练的可能性。然而单单满足对训练集的拟合可能还不能完美的用于工程问题，良好的机器学习方法需要具有一定的泛化能力，并且不能过拟合，所以本节便对比本算法与前面试验中用来对比的算法的泛化能力。

对于算法的泛化能力，通常可以采用交叉验证的方法，既将数据集分割成k份，然后进行k次试验，每次分别取其中一份作为测试集，其他的合集作为训练集进行算法验证。本文中由于是要验证不平衡数据集分类算法的泛化能力，由于少数类样本可能十分的少，如上文中有的训练集中少数类样本只有9个，所以k不宜取特别大，本节实验中k的值选取3。对于前两小节中的八种算法，本小节选取其中四种效果较好的算法进行泛化能力对比试验，他们分别是传统神经网络，AdaBoost算法，结构化支持向量机以及本文中的最小化损失学习神经网络，下表4-5便是交叉验证试验结果。

表4-5 交叉验证试验结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 评价标准 | | | | | | |
| 数据集 | 算法 | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 1 | AD | 0.373 | 0.958 | 0.920 | 0.537 | 0.938 | 0.938 |
| ANN | 0.589 | 0.769 | 0.931 | 0.667 | 0.858 | 0.853 |
| SSVM | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 2 | AD | 0.000 | 0 | 0.920 | 0 | 0.460 | 0 |
| ANN | 0.245 | 0.626 | 0.928 | 0.352 | 0.782 | 0.766 |
| SSVM | -- | -- | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 3 | AD | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| ANN | 0.922 | 0.979 | 0.977 | 0.949 | 0.978 | 0.978 |
| SSVM | 0.941 | 0.923 | 0.967 | 0.932 | 0.952 | 0.952 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 4 | AD | 0.996 | 0.992 | 0.996 | 0.994 | 0.995 | 0.995 |
| ANN | 0.996 | 0.972 | 0.989 | 0.984 | 0.985 | 0.985 |
| SSVM | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| ML-ANN | 0.996 | 0.996 | 0.997 | 0.996 | 0.997 | 0.997 |
| 5 | AD | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| ANN | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| SSVM | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |
| 6 | AD | 0.000 | 0 | 0.759 | 0 | 0.379 | 0 |
| ANN | 0.803 | 0.884 | 0.925 | 0.841 | 0.910 | 0.910 |
| SSVM | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 |

续表4-5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 评价标准 | | | | | | |
| 数据集 | 算法 | 召回率 | 准确率 | %Acc | F1 | A-mean | G-measure |
| 7 | AD | 0.667 | 1.000 | 0.991 | 0.800 | 0.995 | 0.995 | |
| ANN | 0.889 | 1.000 | 0.997 | 0.941 | 0.998 | 0.998 | |
| SSVM | 0.96 | 0.615 | 0.969 | 0.749 | 1 | 0.926 | |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 | |
| 8 | AD | 0.760 | 0.862 | 0.985 | 0.807 | 1 | 0.64 | |
| ANN | 0.8 | 0.769 | 0.979 | 0.784 | 1 | 0.64 | |
| SSVM | -- | -- | -- | -- | -- | -- | |
| ML-ANN | 0.8 | 0.87 | 0.984 | 0.833 | 0.96 | 0.63 | |

由上表4-5可以看出无论是本文中的算法，还是现有的公认的算法，对于本章实验中的八个数据集，其交叉验证的结果与前两小节中的结果并不会相差非常大，所以在这里可以认为本文的算法也具有可靠的泛化能力，前两小节中的拟合结果并不是因为过拟合等原因而得到的良好效果，算法的对训练空间的拟合能力是真实可靠的。

### 算法性能对比试验

本小节主要为了验证本文中算法的性能问题，具体验证的是算法中相对较为敏感的时间复杂度，也就是耗时情况。由于本文中的算法主要是通过进行训练空间F1值优化来解决不平衡数据集的训练问题，这与经典不平衡数据集处理方法的思路与流程基本没有联系，所以经典不平衡数据集处理算法与本文中的算法在性能上没有可比性。这里我们为本文中的算法选取的对比算法是结构化支持向量机，由于他们的算法原理都是对训练空间进行拟合使得其拟合结果的F1值达到最大，并且结构化支持向量机与本文3.4节中提到的算法都是从最小化特定损失学习角度出发的算法，所以选取其用来对比是合理的。

而结构化支持向量机的原型是普通的支持向量机，所以他也包含核函数的功能，对于普通的多项式核函数与最常用的径向基核函数(RBF核)来说，他们的时间复杂度又有天壤之别，而神经网络模型的VC维或者说分类超平面其实是与径向基核函数的效果类似的，所以这里不单对比结构化支持向量机中性能较好的普通线性核函数，也对比了使用径向基核函数的结构化支持向量机的性能。下表4-6便是性能对比试验结果。

表4-6 算法性能对比试验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 算法 | 数据集 | 评价标准 | | 数据集 | 评价标准 | |
| F1值 | 时间 | F1值 | 时间 |
| SSVM-Linear | 1 | 0.373 | 0.958 | 5 | 0.537 | 0.938 |
| SSVM-RBF | 0.589 | 0.769 | 0.667 | 0.858 |
| MVC-ANN | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.833 | 0.96 |
| SSVM-Linear | 2 | 0.000 | 0 | 6 | 0 | 0.460 |
| SSVM-RBF | 0.245 | 0.626 | 0.352 | 0.782 |
| MVC-ANN | -- | -- | -- | -- |
| ML-ANN | 0.8 | 0.87 | 0.833 | 0.96 |
| SSVM-Linear | 3 | 0.941 | 0.923 | 7 | 0.932 | 0.952 |
| SSVM-RBF | 0.922 | 0.979 | 0.949 | 0.978 |
| MVC-ANN | 0.941 | 0.923 | 0.932 | 0.952 |
| ML-ANN | 0.8 | 0.87 | 0.833 | 0.96 |
| SSVM-Linear | 4 | 0.996 | 0.992 | 8 | 0.994 | 0.995 |
| SSVM-RBF | 0.996 | 0.972 | 0.984 | 0.985 |
| MVC-ANN | 0.996 | 0.996 | 0.996 | 0.997 |
| ML-ANN | 0.996 | 0.996 | 0.996 | 0.997 |

表中的F1值为采用训练数据集测试的结果，通过观察上表4-6与表4-1可以发现，表中的算法的运行时间都主要受到训练集样本个数的影响，而数据集的训练难以程度既最终算法获取的F1值的大小对于本文中的算法影响相对较小，其主要是因为本文中的算法都采用神经网络模型，其每轮迭代过程的时间通常是稳定的，而对于支持向量机的话其训练时间受数据集可分程度影响相对于本文中的算法略大一些，这也是因为支持向量机训练的特性。

由上表4-6的时间结果可以看出，结构化支持向量机的线性核算法运行时间通常较短，有些数据集是与最小化损失学习的神经网络类似的，主要如2.3节中介绍，该算法的作者进行了数年的研究对该算法的训练过程进行了较大的优化，不过其优化只能针对线性核。

对于线性核结构化支持向量机，其最终结果在很多数据集上的F1值都是略差与最小化损失学习神经网络的，这也主要是由于线性核的分类面为超平面而神经网络的分类面可以为曲面的缘故。而对于与神经网络类似的径向基核函数，可以看到在样本数目较多或相对较为难分的数据集上，是并不能完成一次训练的，无法得到其最终结果。

综上，本文中的最小化损失学习神经网络通常是可以在可以接受的时间内得到一个与其他算法相当的结果的，所以从性能的角度来分析，本文中的算法也是可行的。

## 本章小结

本章通过四个主要的对比试验，从算法的分类结果以及算法的时间性能两个角度与传统分类算法，经典不平衡数据集处理方法以及结构化支持向量机三类方法进行了对比。

由本章中的数据可以发现，对于算法分类结果，无论是拟合能力还是泛化能力都达到了经典不平衡数据集处理方法的同一水平，而算法时间性能，通过与两类结构化支持向量机对比，也可以证明无论是在F1值优化还是运行时间上，都达到了令人满意的结果，综上可以认为本文中的算法是一种可靠并有效的不平衡数据集训练方法。

# 结 论

随着机器学习研究的不断工业化与工程化，出现了越来越多的不同分类需求，不平衡数据集便是其中一类问题，由于其分布的不平衡性，导致了传统分类方法分类效果不佳的情况，这一类问题也不断吸引着研究者进行研究。对于绝大多数的研究，通常都是采用某种方案来使原始数据集达到一种相对的“平衡态”，便可以使用传统分类方法来解决问题。虽然这样的算法可以在实际问题中取得一定的效果，但是却存在改变空间分布违背传统分类方法训练理论的缺陷，所以本文先后提出两种算法，从另外一个角度切入来解决不平衡数据集分类难的问题。

本文算法主要从不平衡数据集分类评价标准入手，并不再以传统分类方法的全局准确率为训练目标，改为以训练数据分类F1值为训练目标，直接从不平衡数据集分类效果差的根本入手，切中肯綮。本文中的两种算法均已传统人工神经网络为基础模型，要以F1值为训练目标，势必要改变传统分类方法的训练过程，以下便为两种算法的处理方法：

(1)基于最大化F1值分类过程的神经网络：该算法基于最大化F1值分类过程，针对不平衡数据集改进了神经网络模型训练过程中的分类步骤，使其分类过程可以达到当前模型下的最大期望，并通过此分类结果与目标输出的误差来更新当前模型，已达到不平衡数据集的训练目的。

(2)最小化损失学习神经网络：该算法直接从分类器的损失函数入手，不再使用传统损失函数，而是直接构建了与F1值相关的损失函数，通过当前分类器输出求解F1值的近似期望来作为损失，通过证明该近似期望为F1值期望的下界，来确认对其优化的可行性，并且由于其不再具有原F1函数的离散性，而是直接与分类器输出建立了联系，所以可以通过反向传播算法进行迭代，以此完整的解决不平衡数据集训练问题。

本文中的两种算法各有优劣，第一中算法从算法角度来讲，更为精确的获取了每一状态下的F1值，并可以根据精确的F1值进行损失求解，不过这一过程耗费巨量时间，可能在真实情况下会难以进行，第二种算法通过近似的求解期望来解决了这一问题，不过也因为带来了相应的精度损失。

对于本文中算法，还有以下问题可以进行研究或优化：

(1)从严格意义上来讲，本文两种算法均没有将F1值的精确期望与神经网络模型的输出或者模型参数建立直接的数学联系，如果建立起直接联系可能将会使算法分类准确度有进一步的提升。

(2)本文中的算法无法并行运算，并且无法使用传统机器学习方法中的batch求解过程，如果可以解决此问题，也许可以在算法时间性能上有质的飞跃。

# 参考文献

# 攻读硕士学位期间发表的论文及其它成果

1. 发表的学术论文

[1] Zhang C, Jiang J, Shi F. Research on approach for classification of Within imbalanced data sets[C]//Proceedings of the International Conference on Bioinformatics & Computational Biology (BIOCOMP). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016: 85.

[2] Zhang C, Jiang J, Hu Z. The Research of Human Activity State Recognition Base on Acceletometers[C]// Proceedings of the International Conference on Health Informatics and Medical System(HIMS). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016: 85.

# 哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《针对类内不平衡样本分类方法的研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名： 日期： 年 月 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名： 日期： 年 月 日

导师签名： 日期： 年 月 日

# 致 谢

逝者如斯，已逝两载星霜，朱颜辞岁，却是仰取俯拾。在此我要向身边陪伴我一路走来的人表示由衷的感谢。

首先要感谢的一定是我的导师张春慨副教授。张老师他生活中为人谦和，风趣幽默，课余时间一直像长辈一样对我们悉心照料，关爱有加。工作中认真负责，治学严谨。教学中条例清晰，循循善诱，尤其是在与我讨论论文的时候，经常一两句话就令我豁然开朗，受益匪浅。

其次我要感谢我的家人，长我育我，顾我复我，对我百般呵护，无微不至，是你们给了我前进的力量。

最后，要感谢的是身边的同学和朋友，无论是点头之交或是莫逆之交，无论已成为匆匆过客还是将风雨同舟，你们都已给我带来了可以永藏的美好回忆。

1. Menon A K, Narasimhan H, Agarwal S, et al. On the statistical consistency of algorithms for binary classification under class imbalance[C]// International Conference on Machine Learning. 2013:603-611. [↑](#endnote-ref-2)
2. He H, Garcia E A. Learning from imbalanced data[J]. IEEE Transactions on knowledge and data engineering, 2009, 21(9): 1263-1284. [↑](#endnote-ref-3)
3. Tomanek K, Hahn U. Reducing class imbalance during active learning for named entity annotation[C]//Proceedings of the fifth international conference on Knowledge capture. ACM, 2009: 105-112. [↑](#endnote-ref-4)
4. Marqués A I, García V, Sánchez J S. On the suitability of resampling techniques for the class imbalance problem in credit scoring[J]. Journal of the Operational Research Society, 2013, 64(7): 1060-1070. [↑](#endnote-ref-5)
5. Fattahi S, Othman Z, Othman Z A. NEW APPROACH FOR IMBALANCED BIOLOGICAL DATASET CLASSIFICATION[J]. Journal of Theoretical & Applied Information Technology, 2015, 72(1). [↑](#endnote-ref-6)
6. Chawla N V, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets[J]. ACM Sigkdd Explorations Newsletter, 2004, 6(1): 1-6. [↑](#endnote-ref-7)
7. Yoon K, Kwek S. A data reduction approach for resolving the imbalanced data issue in functional genomics[J]. Neural Computing and Applications, 2007, 16(3): 295-306. [↑](#endnote-ref-8)
8. Huang K, Yang H, King I, et al. Learning classifiers from imbalanced data based on biased minimax probability machine[C]//Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, 2004, 2: II-558-II-563 Vol. 2. [↑](#endnote-ref-9)
9. Japkowicz N. Learning from imbalanced data sets: a comparison of various strategies[C]//AAAI workshop on learning from imbalanced data sets. 2000, 68: 10-15. [↑](#endnote-ref-10)
10. Batista G E, Prati R C, Monard M C. A study of the behavior of several methods for balancing machine learning training data[J]. ACM Sigkdd Explorations Newsletter, 2004, 6(1): 20-29. [↑](#endnote-ref-11)
11. Chaudhuri S, Motwani R, Narasayya V. On random sampling over joins[C]//ACM SIGMOD Record. ACM, 1999, 28(2): 263-274. [↑](#endnote-ref-12)
12. Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357. [↑](#endnote-ref-13)
13. Wang S, Li Z, Chao W, et al. Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning[C]//The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, 2012: 1-8. [↑](#endnote-ref-14)
14. Gao M, Hong X, Chen S, et al. Probability density function estimation based over-sampling for imbalanced two-class problems[C]//The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, 2012: 1-8. [↑](#endnote-ref-15)
15. Drummond C, Holte R C. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling[C]// ICML Workshop on learning from imbalanced datasets II. 2003, 11. [↑](#endnote-ref-16)
16. Zhang Y P, Zhang L N, Wang Y C. Cluster-based majority under-sampling approaches for class imbalance learning[C]//Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on. IEEE, 2010: 400-404. [↑](#endnote-ref-17)
17. Kubat M, Matwin S. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection[C]//In Proceedings of the Fourteenth International Conference on Machine Learning. 1997. [↑](#endnote-ref-18)
18. Provost F, Fawcett T. Robust classification for imprecise environments[J]. Machine learning, 2001, 42(3): 203-231. [↑](#endnote-ref-19)
19. Japkowicz N, Stephen S. The class imbalance problem: A systematic study[J]. Intelligent data analysis, 2002, 6(5): 429-449. [↑](#endnote-ref-20)
20. Zhang C, Jiang J, Shi F. Research on approach for classification of Within imbalanced data sets[C]//Proceedings of the International Conference on Bioinformatics & Computational Biology (BIOCOMP). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016: 85. [↑](#endnote-ref-21)
21. Wang C, Hu L, Guo M, et al. imDC: an ensemble learning method for imbalanced classification with miRNA data[J]. Genetics and Molecular Research, 2015, 14(1): 123-133. [↑](#endnote-ref-22)
22. [↑](#endnote-ref-23)
23. Freund Y, Schapire R E. A desicion-theoretic generalization of on-line learning and an application to boosting[C]//European conference on computational learning theory. Springer Berlin Heidelberg, 1995: 23-37. [↑](#endnote-ref-24)
24. Cheng K, Chen Q, Yang X, et al. Classification of imbalanced bioinformatics data by using boundary movement-based ELM[J]. Bio-Medical Materials and Engineering, 2015, 26(s1): S1855-S1862. [↑](#endnote-ref-25)
25. Jansche M. Maximum expected F-measure training of logistic regression models[C]//Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2005: 692-699. [↑](#endnote-ref-26)
26. Dembczynski K J, Waegeman W, Cheng W, et al. An exact algorithm for F-measure maximization[C]//Advances in neural information processing systems. 2011: 1404-1412. [↑](#endnote-ref-27)
27. Mitchell T M. The need for biases in learning generalizations[M]. New Jersey: Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ., 1980. [↑](#endnote-ref-28)
28. Daskalaki S, Kopanas I, Avouris N. Evaluation of classifiers for an uneven class distribution problem[J]. Applied artificial intelligence, 2006, 20(5): 381-417. [↑](#endnote-ref-29)
29. Gu Q, Zhu L, Cai Z. Evaluation measures of the classification performance of imbalanced data sets[C]//International Symposium on Intelligence Computation and Applications. Springer Berlin Heidelberg, 2009: 461-471. [↑](#endnote-ref-30)
30. Van Rijsbergen C J. Foundation of evaluation[J]. Journal of Documentation, 1974, 30(4): 365-373. [↑](#endnote-ref-31)
31. Provost F, Fawcett T. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions[C]//In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. 1997. [↑](#endnote-ref-32)
32. Jansche M. A maximum expected utility framework for binary sequence labeling[C]//Annual Meeting-Association For Computational Linguistics. 2007, 45(1): 736. [↑](#endnote-ref-33)
33. Joachims T. A support vector method for multivariate performance measures[C]//Proceedings of the 22nd international conference on Machine learning. ACM, 2005: 377-384. [↑](#endnote-ref-34)
34. Tsochantaridis I, Hofmann T, Joachims T, et al. Support vector machine learning for interdependent and structured output spaces[C]//Proceedings of the twenty-first international conference on Machine learning. ACM, 2004: 104. [↑](#endnote-ref-35)
35. Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines[J]. 1998. [↑](#endnote-ref-36)
36. Finley T, Joachims T. Training structural SVMs when exact inference is intractable[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 304-311. [↑](#endnote-ref-37)
37. Joachims T. Training linear SVMs in linear time[C]//Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006: 217-226. [↑](#endnote-ref-38)
38. Joachims T, Finley T, Yu C N J. Cutting-plane training of structural SVMs[J]. Machine Learning, 2009, 77(1): 27-59. [↑](#endnote-ref-39)
39. Gordon D F, Desjardins M. Evaluation and selection of biases in machine learning[J]. Machine Learning, 1995, 20(1-2): 5-22. [↑](#endnote-ref-40)
40. Blumer A, Ehrenfeucht A, Haussler D, et al. Learnability and the Vapnik-Chervonenkis dimension[J]. Journal of the ACM (JACM), 1989, 36(4): 929-965. [↑](#endnote-ref-41)
41. Bartlett P L. Lower bounds on the Vapnik-Chervonenkis dimension of multi-layer threshold networks[C]//Proceedings of the sixth annual conference on Computational learning theory. ACM, 1993: 144-150. [↑](#endnote-ref-42)
42. 顾乃杰, 赵增, 吕亚飞, 等. 基于多 GPU 的深度神经网络训练算法[J]. 小型微型计算机系统, 2015, 36(5): 1042-1046. [↑](#endnote-ref-43)