密级:（涉密论文填写密级，公开论文不填写）

中国科学院大学
University of Chinese Academy of Sciences

博士学位论文

多标记学习算法研究及在生物医学数据挖掘中的应用

作者姓名：王 普

指导教师：周丰丰 研究员 吉林大学

蔡云鹏 副研究员 深圳先进技术研究院

学位类别：工学博士

学科专业：模式识别与智能系统

研究所：中国科学院深圳先进技术研究院

2017 年 4 月

The Study on Multi-label Learning and its Applications for Biomedical Data Mining

By

Pu Wang

A Dissertation/Thesis Submitted to

The University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the degree of

Doctor of Engineering

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

April, 2017

致　谢

春华秋实，四年多的辛勤耕耘终于迎来了要毕业的时刻。回首四年前来先进院参加博士生入学考试，看到院门口“中国科学院”的牌子时，向往之心油然而生，当时的情景犹如就发生在昨天。很幸运我能成为一名 SIATer。来到先进院，仿佛置身于一片知识的海洋，身边有着近 2000 名优秀的科研工作者和学生为伍，大家有着共同的目标和追求——创建一流的工业研究院。在这里，几乎每天都有来自国内外的专家学者举办各种各样的学术活动和学术报告，从中我的知识面得到极大的丰富，眼界得到极大的开阔。感谢先进院提供这么好的科研平台和学术环境。

感谢周丰丰、蔡云鹏和李烨老师的悉心指导。周老师科研基础扎实，除了自己不断地学习新知识、新理论，还鼓励学生努力学习、勇于创新；周老师学风严谨，对学生严格要求，每次讨论研究进展总要求我们用数据说话，拿结果来讨论；周老师关心学生的生活和工作，对我的生活和职业规划提供了许多宝贵的建议。周老师的治学态度和科研热情令人敬佩，对学生的关怀和理解令人感激，很荣幸能有周老师这样的良师益友。蔡老师知识丰富、思路开阔而又谦逊低调，无论何时何地，无论请教任何问题，蔡老师总是能不厌其烦地耐心解答。李老师公务繁忙但依然保留着敏锐的学术观察力和对科研的热情，工作兢兢业业，对中心的发展和学生培养都倾注了大量心血。从三位老师身上学到的和得到的实在是太多太多，对你们致以最崇高的敬意和由衷的感谢！

感谢身边的每一位同学和朋友，有你们在，再苦再累的博士生涯也充满了乐趣。感谢师兄罗幼喜、李开士、葛瑞泉、李洪刚、何晨光、易称福，从你们那里，我得到了许多的指导和启发；感谢中心里的同事或同学刘记奎、樊小毛、杨玉洁、孟庆汉、周曼丽、彭超、刘利明、洪溪、林剑华等人，我们朝夕相处，一起学习，一起讨论，一起打球，一起爬山，一起徒步，与你们在一起的日子，我将终身难忘。

最后，带着深深的歉意感谢我的家人。为支持我完成学业，爱人不辞辛苦，任劳任怨，担负起照顾孩子和养家糊口的重担。感谢这几年来父母的理解和支持，在你们该颐养天年的时候，我又选择了远走他乡。这些年来，对家人的陪伴和关爱实在是太少了，再次感谢你们无怨无悔的付出和支持。

梦想成就未来，应用创造价值。先进院的创新文化已经深深地融入到我的血液当中，我将带着梦想从这里起航，并衷心祝愿我的 SIAT 越来越好！

声　　明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：　　日期：

论文版权使用授权书

本人授权中国科学院深圳先进技术研究院可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

（保密论文在解密后适用本授权书。）

作者签名：　　导师签名：　　日期：

摘 要

作为机器学习领域的一个分支，多标记学习系统具有单输入样本多输出语义的特点。研究对象具有多语义性在现实世界中是非常普遍的，如一副图片可能同时具有多个主题，一个蛋白质分子在进入细胞后会有针对性地定位于几个特定的亚细胞器中，一个基因往往同时具有多种功能，一个病人可能同时具有多种并发症。为解决这样的特殊问题，研究人员们陆陆续续地提出了许多的多标记学习算法。集成学习和深度学习是近几年最流行的机器学习方法，本文力图将这两种强大方法与多标记学习结合起来，设计出新的多标记学习算法，同时将他们应用于若干生物医学数据挖掘问题当中。

本文首先对当前的多标记学习理论、方法和应用进行了广泛而深入的调研，在此基础之上开展了一系列的算法创新和应用创新工作，具体研究内容如下：

1. 提出一种混合多标记学习算法。一个好的集成学习器要求所构成的基学习器“好而不同”，基于这种思想，本文设计出了一种新的混合多标记学习器，它采用了两种完全不同的异质学习器，一个是特征驱动的方法，另一个是近邻驱动的方法；一个用到的是所有样本的全局信息，一个用到的是近邻的局部信息；一个属于线性学习器，另一个属于非线性学习器，同时还能将标记之间的相关性考虑进模型当中。这两种方法相辅相成，共同构成了一个有机整体。通过在多个多标记数据集上的实验发现，混合多标记学习器比所构成的任一个基学习器都要强，且与已有的几个多标记学习算法相比也具有显著的优势。
2. 提出了一种基于 ReLU 激活函数的新型深度多标记学习算法，为满足多标记输出的需求，该算法将单标记学习中常用的损失函数（如交叉熵或铰链损失）替换多标记损失函数。文中推导了多标记深度网络的迭代优化步骤，并讨论了多标记交叉熵损失函数与对数损失函数之间的关系。通过在多标记数据集上进行实验，我们详细讨论了所提出的模型中各个超参数对模型性能的影响，为合理使用深度多标记学习算法提供了一些经验参考。最后，通过在多个生物医学领域的多标记数据集上的实验比较发现，我们所提出的深度多标记学习算法要优于传统的多标记学习算法，包括一些已知最好的集成多标记学习方法。
3. 将多标记学习算法应用于抗菌肽活性预测问题当中。抗菌肽是一种具有广谱抗菌活性的生物小分子，具有巨大的潜在药物价值。本文试图通过机器学习的方法来预测抗菌肽活性（功能），而抗菌肽的活性预测是一个典型的多标记学习问题，因为已知自然存在的每个抗菌肽分子往往同时具有多种活性。本文根据最新的抗菌肽数据库构建了一个抗菌肽分子活性数据集，比较了多种抗菌肽分子特征提取方法和多标记学习方法，通过详细的实验比较发现，采用氨基酸成分与二联体成分的组合特征提取方法并结合我们提出的多标记学习方法，能取得

最高的预测精度。

4. 将多标记学习算法应用于慢性病预测问题当中。慢性病人如糖尿病患者除了自身的主要病症以外往往会有多种并发症，对于这样的问题，可以使用多标记学习算法来进行建模。通过在 MIMIC-II 数据库中慢性病数据集（包括 19733 个病人，涵盖 10 种慢性疾病）上进行的实验发现，本文提出的深度多标记学习算法要显著优于 14 种传统的多标记学习算法，这可能是因为该数据集比较大，更易于发挥深度学习的优势。同时在算法运行效率上，本文提出的方法也很有竞争力，这主要得力于多标记损失函数和 ReLU 激活函数的高效性。

关键词：多标记学习；集成学习；深度学习；抗菌肽活性；慢性病预测；

The Study on Multi-label Learning and its Applications for Biomedical Data Mining

Wang Pu (Pattern Recognition and Intelligent System)
Directed By Zhou Fengfeng, Cai Yunpeng

As a branch in the field of machine learning, multi-label learning system has the characteristic of single input sample while multiple output semantic topics. The research object has multiple attributes is very common in the real world, such as one picture may have multiple topics simultaneously; after getting into the cell, a protein molecule always locates in several specific subcellular organelles; a gene tends to have a variety of functions at the same time; a patient may also have a variety of complications. To solve such special problems, the researchers have proposed many multi-label learning algorithms. Ensemble learning and deep learning are both the most popular machine learning methods in recent years, this thesis attempts to combine them with multi-label learning to design novel multi-label learning algorithms, and use them for mining biomedical data.

Firstly, this thesis makes a broad and deep investigation in the theory, method and application of multi-label learning, and then a series of innovation and application work is done. The main points of the study are as follows:

1. A hybrid multi-label learning algorithm is proposed. Effective ensemble learning requires the base learner “good but different”, based on this idea, this thesis designed a new hybrid multi-label learner, which is composed of two heterogeneous base learner. One is feature-driven method, while the other is neighbor-driven; one utilizes the global information of all data, while the other only use the local information of the neighbor; one belong to linear algorithm, while the other belong to nonlinear algorithm, in which the label correlation is also considered. The two methods complement each other and work well as a whole. Through the experiments on several multi-label datasets we can find that the hybrid model is better than any base learner, and has significant advantage when comparing with existing methods.
2. A novel deep multi-label learning method is proposed, which is based on ReLU activation function. The loss functions used in single label learning, such as cross-entropy or hingle loss, are changed to be multi-label loss functions in order to cope with the need of multiple output. The iterative optimization for the proposed method is deduced, and the relationship of multi-label loss and log loss is also studied in this paper. Through experiments on several biomedical multi-label datasets, we can find that the proposed method is superior to the traditional methods, including the best known ensemble ones.
3. Multi-label learning for the prediction of antimicrobial peptide activities.

Antimicrobial peptides are kind of biological small molecules with broad-spectrum antimicrobial activities, which demonstrate potential as novel therapeutic agents. This thesis attempts to predict antimicrobial peptide activities by machine learning method, which is a typical multi-label learning problem because any natural antimicrobial peptide may have multiple activities. In this thesis we build a new data set about antimicrobial peptide activity based on the latest antimicrobial peptide database, and compare several feature extraction and multi-label learning methods. Through detailed experimental comparison, it can be found that the best performance is obtained by the proposed multi-label learning algorithm together with the features of Amino acid composition and dipeptide composition.

4. Multi-label learning for the prediction of chronic diseases. Chronic patients tend to have a variety of complications then multi-label learning can be used here for modeling. Though the experiments on a chronic disease data set consisted of 19733 patients and 10 kinds of chronic diseases coming from MIMIC-II database, it can be found that the proposed deep multi-label learning method is significantly better than the fourteen traditional algorithms. Maybe this data set is relatively big and the deep learning can fit it well. What's more, the proposed method is also very competitive in running time, which is mainly due to the efficiency of multi-label loss function and ReLU activation function.

Keywords: multi-label learning, ensemble learning, deep learning, antimicrobial peptide activities, chronic disease prediction,

目 录

图目录

表目录

第一章 绪论

随着高通量测序和医疗数字化技术的发展，各种各样的生物医学数据如基因组学、蛋白质组学、代谢通路、微生物组学、电子病历、心电、脑电、医学影像等正在爆发式增长，生物医学大数据时代已经来临[1]。目前，如何处理和解读这些海量的、异构的生物医学数据对我们还是一个巨大的挑战。近年来，以机器学习为核心的人工智能（Artificial Intelligence，AI）技术迅速崛起，并影响到几乎各行各业。将人工智能技术引入到生物医学领域，将会帮助我们挖掘出生物分子数据库中的隐藏信息、提高药物研发效率并降低成本、提高病理诊断的准确性和医疗服务质量、为个体设计个性化的健康管理计划等[2]。

1.1 课题研究背景和意义

人工智能距离我们还遥远吗？用户在百度搜索里输入的内容千奇百怪，但是由于百度通过机器学习和自然语言处理技术来解析用户的输入，使得搜索引擎“更懂你”，从而能为用户提供所需的内容，减少用户的筛选时间，提升了用户体验；当人们正在驾车的途中，无法分神去打开音乐播放器并播放一首想听的歌曲时，科大讯飞的汽车语音助手就派上用场了，而且你会发现它的语音识别精准度非常高，这主要得力于深度学习技术在语音识别中的应用；现在，人们正越来越多地使用智能手机来浏览新闻或者购物，如果把不同人的手机同时放在一起，你会发现同样的浏览器（如 UC）或购物 APP（如淘宝）其显示内容却因人而异，这是因为这些应用程序能够根据你的用户画像做个性化的推荐。上述几个例子只是人工智能应用的冰山一角，其他方面还有机器人、无人驾驶、计算机视觉、经济政治决策、生物信息学、计算机辅助医学诊断等等[3-5]。

除了人工智能之外，我们还会经常听到与之相关的两个概念：机器学习和深度学习，实际上它们并不是相互独立的方法学或术语，他们之间的关系可以用下图来描述。

图 1.1 人工智能、机器学习、深度学习及多标记学习之间的关系

人工智能是其中最广泛的一个概念，它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门技术科学（百度百科）。通俗地讲，人工智能就是研究如何让机器获取知识并运用知识，做只有人类才能从事的智能工作。

机器学习是人工智能的核心，从人工智能的发展现状来看，它也是实现人工智能的最有前途的手段。2016 年，谷歌首席执行官 Sundar Pichai 就表示：“机器学习是核心，它改变了我们重新思考我们如何做任何事情的方式。我们正把机器学习应用到公司所有产品中，无论是搜索、广告、YouTube 或是应用商城 Play”。机器学习是一种数据驱动的方法，它借助于学习算法从已有的数据中发现规律，并能够对新的数据做出判断和预测（如图 1.2 所示）。使用机器学习时，人们不需要通过编写一套指令集来一步一步告诉机器该怎么做，而是使用算法和数据来训练机器，让机器学到知识并自主决定如何执行任务。机器学习的主要研究内容包括特征工程、回归、分类、聚类和强化学习等，所使用的方法主要有线性回归、贝叶斯理论、近邻法、人工神经网络、决策树、概率图模型、支持向量机和集成学习等[6]。

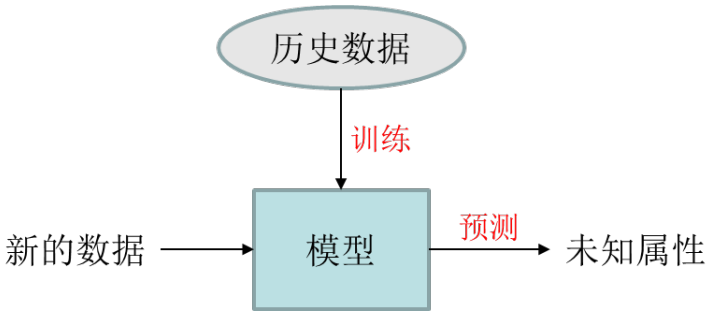


图 1.2 机器学习是一种数据驱动的方法

深度学习自诞生以来迅速在全球掀起一股热潮。从理论和技术上来看，深度学习本身并没有太大的创新，其实它就是一种多层次的人工神经网络。多层神经网络在人工神经网络诞生不久就已经有人提出了，只是受限于当时的计算能力，多层神经网络无法实现和应用。深度学习相比于传统的机器学习方法具有很大的优势，首先是它具有强大的非线性映射能力，这在面对复杂的学习任务时非常有用；其次，深度学习是一种端对端的方法，将特征工程和机器学习算法有机结合起来，这点在工程应用时非常方便[7-11]。

本文所研究的多标记学习也属于机器学习的范畴，它针对的是现实世界中物体的多语义或多属性问题[12-14]。例如图 1.3 所示的一幅图片，它就同时具有海水、沙滩和蓝天的语义。多标记学习就是要用具有多语义的样本来训练出一个模型，使得输入新的样本时，模型能同时输出所有可能与它相关的一个语义集。多标记学习的应用场景包括文本主题、图片语义、音乐情感、基因或蛋白质功能、医疗诊断等属性预测问题。

已有的机器学习方法几乎都可以直接或间接地应用于多标记学习问题，本文主要研究将集成学习和深度学习等最新的机器学习理论和方法与多标记学习问题结合起来，设计出新的多标记学习算法，并应用于生物医学领域的若干多标记学习问题。

图 1.3 一副图片具有多个语义：海水、沙滩和蓝天

1.2 国内外研究现状

1.2.1 AI 发展现状

2016 年是全球人工智能发展的元年，这一年由 Google 基于人工智能技术开发的一款围棋程序 AlphaGo 在与围棋世界冠军、职业九段选手李世石进行的“人机大战”中以 4:1 的绝对优势获胜（图 1.4）。此后的 2016 年末至 2017 年初，AlphaGo 在围棋网站上以“Master”为注册名与中日韩众多围棋高手连续对决 60 局竟没有出现一次败绩（59 胜 1 和），令围棋等级分排名人类第一的柯洁也在微博上感叹道：“感谢 Alphago 最新版给我们棋界带来的震撼，作为一开始就知道真身是谁的我来讲，是多么希望网上的快棋人类能赢一盘”。

图 1.4 AlphaGo（由人类代为执子）对战李世石

回想 20 年前的那次“人机大战”，IBM“深蓝”对战国际象棋大师加里 • 卡斯帕罗

夫，虽然当时机器以微弱的优势获胜，但实际上与 AlphaGo 采用的人工智能技术不同，“深蓝”主要依靠强大的计算能力穷举所有路数来选择最佳策略。围棋由于搜索空间过大，机器无法依靠穷举的策略来取胜。AlphaGo 的成功主要是因为采用了最新的机器学习方法和技术，如神经网络（neural network）、深度学习（deep learning）、强化学习（reinforcement learning）等，使得机器能够像人类一样学习和推断。

“人工智能”这一术语于 1956 年的达特茅斯夏季会议上提出，当时一群卓越的年轻科学家在此聚会，讨论机器模拟人类智能的一系列问题。这次会议标志着人工智能学科的正式诞生。此后的几十年间，人工智能的发展磕磕碰碰，人们对它的态度不断变化，时而认为它是人类文明的未来，时而认为它只是技术垃圾、轻率的概念而已。直到 2010 年前后，随着新的机器学习方法的出现、“大数据”时代的到来和高性能计算（分布式计算、并行计算等）的突破发展，人工智能迎来了新的发展机遇。

人工智能分为“强人工智能”和“弱人工智能”[15]。具有强人工智能的机器拥有人类所有的感知和推理能力，甚至拥有知觉和自我意识，能够像人类一样思考和行动，如我们在电影中看到的机器人 C-3PO、终结者。而具有弱人工智能的机器只能在面对一些特殊任务时才能做得像人类一样或者比人类做得更好。目前，弱人工智能的发展如火如荼，在图像分类、人脸识别、语音识别和机器翻译等领域发挥越来越重要的作用。但是强人工智能的发展尚处于瓶颈期，人类还未发现或创造出通用的人工智能方法。尽管人工智能已经在一些特定的应用场景中表现出了令人吃惊的能力，很多社会学家、科技精英和媒体人士都表达了对人工智能可能危及人类安全的担忧[16]，但实际上当前的人工智能所能做的还非常有限，距离通用人工智能还很遥远。

过去几年里，人工智能得到了飞速发展，并在学术界、政界、产业界和资本界刮起了一阵旋风[17; 18]。

在学术界，越来越多的研究者致力于人工智能和深度学习的研究，发表的有关这方面的论文数也快速增加（图 1.5~1.6）。美国是人工智能研究领域的先行者，起初麻省理工学院、斯坦福大学和加利福尼亚大学等美国大学在 AI 基础研究上领先世界，随后谷歌、Facebook 和微软等 IT 企业开始大力推进。中国的大学和企业在这一领域的表现也非常抢眼。日本文部科学省下属的科学技术和学术政策研究所的分析显示，从主要国际学会的发表成果来看，中美占据压倒性优势，日本落后。而美国白宫在 2016 年 10 月发布的报告中称，中国的人工智能研究已经走在了世界前列，从 2013 年开始发表的提及“深度学习”或“深度神经网络”的期刊论文数量超越了美国，居世界第一。值得一提的是，中国的相关论文不仅数量上远超其他国家，质量上的表现也毫不逊色，被引超过一次的论文数量也远超美国。

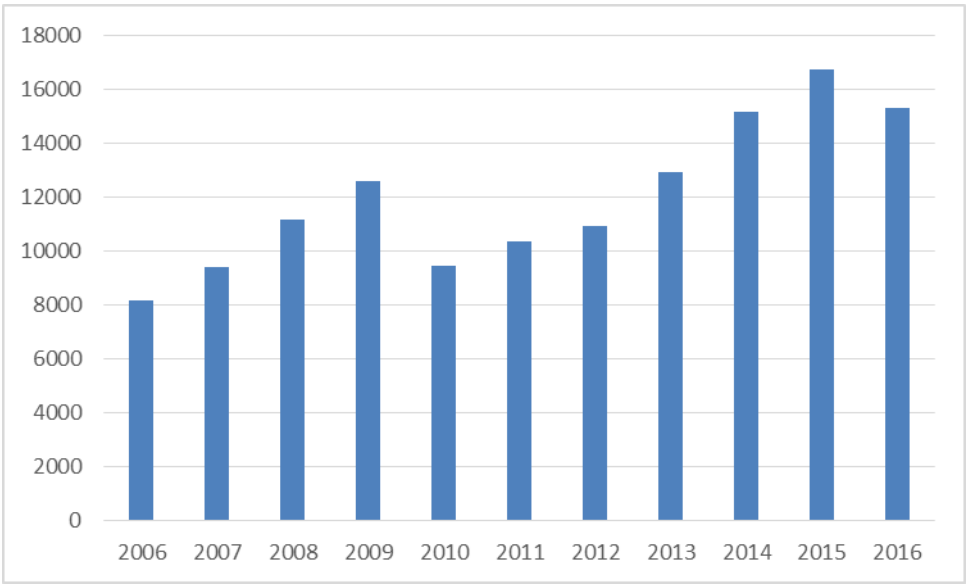


图 1.5 Web of Science 中以“Artificial intelligence”或“deep learning”为主题的文章数

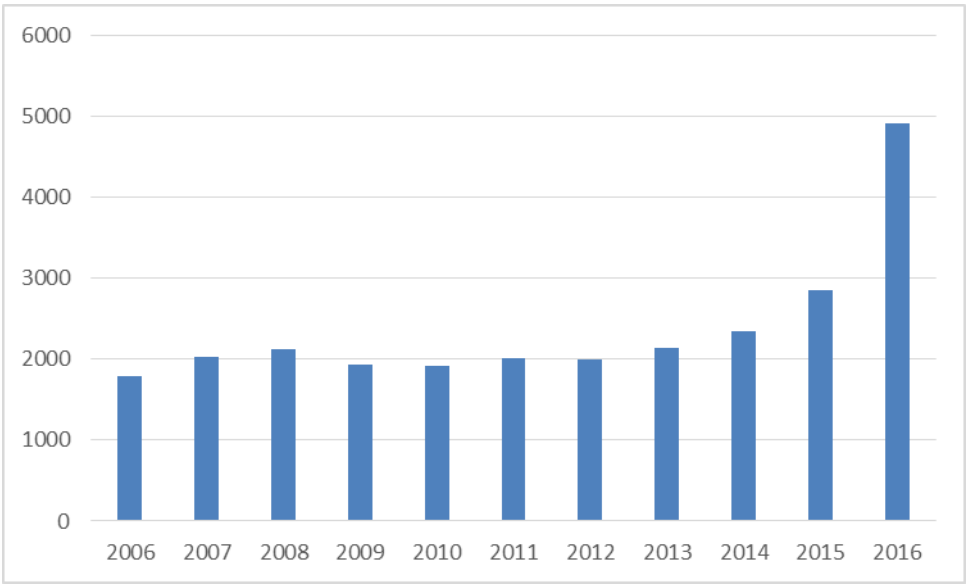


图 1.6 中国知网中以“人工智能”或“深度学习”为主题的文章数

除了基于人工神经网络的深度学习方法，令人眼前一亮的是，2017 年初，我国机器学习、集成学习和多标记学习领域的领军人物、南京大学周志华教授和他的学生又提出了另外一种实现深度学习的思路——深度森林（Deep Forest）[19]，它的基本思想是将随机森林级联起来实现深层次的表征学习（representation learning），如图 1.7 所示。从一些数据集上的实验结果来看，深度森林也能达到或超过深度神经网络模型的效果。

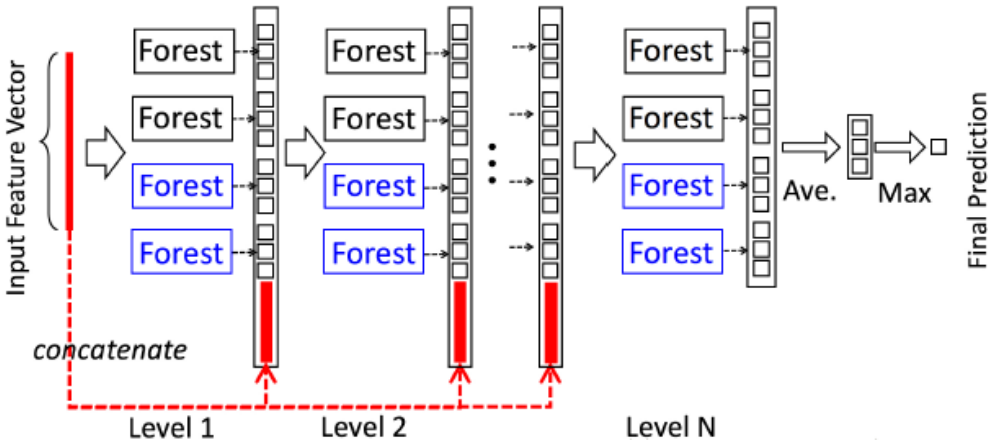


图 1.7 级联森林结构图

2016 年 10 月 13 日，美国白宫发布了《为人工智能的未来做好准备》(Preparing for the Future of Artificial Intelligence)和《国家人工智能研究与发展战略计划》(National Artificial Intelligence Research and Development Strategic Plan)两份重要报告[20]。前者探讨了人工智能的发展现状、应用领域以及潜在的公共政策问题；后者提出了美国优先发展的人工智能七大战略方向及两方面建议，对我国人工智能产业发展具有重要的借鉴意义。今年 3 月 5 日，国务院总理李克强发表 2017 政府工作报告，指出要加快培育壮大包括人工智能在内的新兴产业，“人工智能”也首次被写入了全国政府工作报告。人工智能将在中国的政治、经济、学术领域都成为重中之重，中国人工智能迎来真正的新纪元。

值得关注的是，AI 这一次的兴起并不仅仅停留在学校或实验室当中，而是火速受到产业界和资本界的追捧，包括整个 AI 产业链中的芯片、算法、生物识别、机器视觉、语音识别和各种应用[17; 21]。Venture Scanner 统计数据显示，2015 年全球人工智能公司共获得近 12 亿美元的投资。而据 BBC 预测，到 2020 年，全球人工智能市场规模有望超千亿美元。从全球来看，处于人工智研究和应用巅峰的当属科技巨头 Google、Facebook、微软和 IBM，而国内的领跑者是 BATH（百度、阿里、腾讯、华为）。不容小觑的是，国内还涌现出了很多 AI 新锐势力，如科大讯飞、寒武纪科技、地平线、思必驰、云知声、云从科技、商汤科技等等。AI 在计算机视觉、语音识别、机器翻译、无人驾驶等领域的应用能够改善人们的生产生活，而 AI 与生物医药的结合则会很大程度上影响到人类的生命和健康，例如 AI 可以帮助我们进行蛋白质功能预测、潜在的致病基因发现、计算机辅助诊疗、医学影像分析、药物挖掘和健康管理等[4]。IBM Watson 可以在 17 秒内阅读 3469 本医学专著，24000 篇论文 61540 次试验数据，106000 份临床报告。通过海量汲取医学知识，Watson 在短时间内迅速成为肿瘤专家。2012 年 Watson 通过了美国职业医师资格考试，并部署在美国多家医院提供辅助诊疗的服务。2015 年成立于加拿大的 Deep Genomics 公司就是人工智能和基因组学联姻的产物，该公司的研究者们将深度学习的能量引入到基因组学领域，希望发现 DNA 的某些角落暗藏的疾病线索。同样是这一年，原华大基因 CEO 王俊创立了碳云智能，公司旨在构建一个健康大数据平台，涵

盖基因数据、微生物数据、蛋白及代谢数据等，并通过人工智能技术来处理这些数据，帮助人类做健康管理。

1.2.2 多标记学习的发展现状

近年来，越来越多的研究人员开始关注多标记学习（图 1.8~1.9），并提出了很多的多标记学习算法[22-45]，而对这些算法进行总结和归类有助于加深对已有算法的理解及下一步的创新。

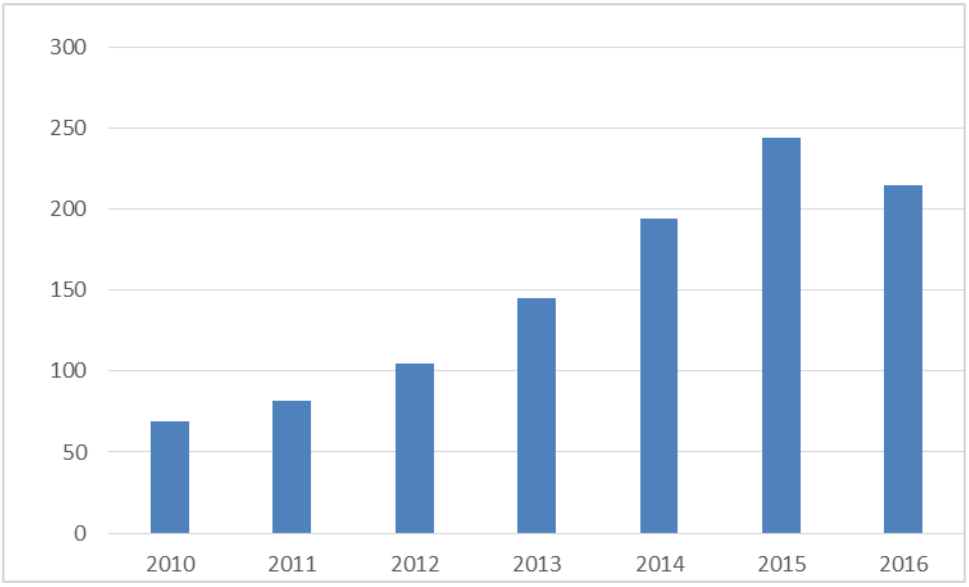


图 1.8 Web of Science 中标题含有“multi-label”的论文数量

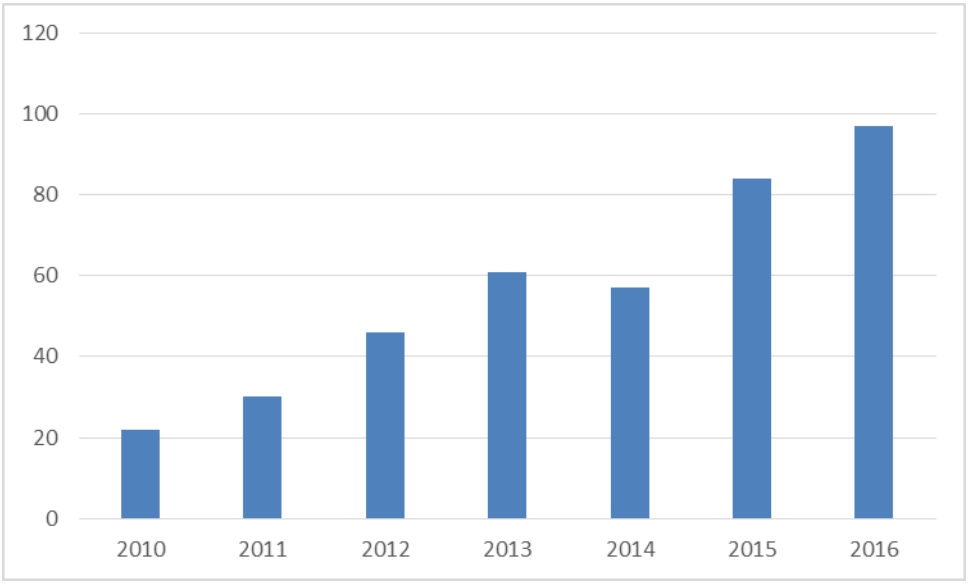


图 1.9中国知网中标题含有“多标记”或“多标签”的论文数量

按照文献[46; 47]的做法，可以将这些方法归为两大类：问题转化方法（Problem transformation methods）和算法适应方法（Algorithm adaptation methods）。问题转化方法将多标记学习任务转化成单标记学习任务，这时大量已有的单标记学习算法可以用在这里，从而间接完成对多标记数据的处理。通俗来讲，这类方法是用数据来适应算法，不

需要改变已有的分类算法，而是直接调用或者组合。问题转化方法的典型代表有 Binary Relevance（BR）[48]、Classifier Chains（CC）[49]、Ranking by Pairwise Comparison (RPC)[50]、Calibrated Label Ranking（CLR）[51]、Label Powerset（LP）[52]、Random k-labelsets（RAkEL）[53; 54]和 label-specific features（LIFT）[55]。而算法适应方法则是对已有的学习算法进行扩展后直接用来处理多标记数据，通俗地讲就是用算法来适应数据，通常需要将已有的经典分类算法进行适度调整或者干脆发明出新的方法出来。典型代表有 Multi-Label k-Nearest Neighbor（ML-kNN）[56]、Multi-Label Decision Tree（ML-DT）[57]、Ranking Support Vector Machine（Rank-SVM）[58]和 Backpropagation for Multi-Label Learning（BP-MLL）[59]。

另外，根据标记之间的相互作用情况可以将多标记学习算法分成三种：一阶策略（First-order strategy）、二阶策略（Second-order strategy）和高阶策略（High-order strategy）。采用一阶策略时，将不考虑标记之间的相关性，每个标记都独立学习，如 BR、ML-kNN、LIFT 等。采用二阶策略时，所有标记以成对的形式学习，一个标记会受到另外一个标记的影响，如 RPC、CLR 等。而采用高阶策略时，对任何一个标记的学习都要同时考虑两个以上的其他标记的影响，如 LP、RAkEL、ECC 等。文献[46]认为如何挖掘标记之间的关联性对多标记学习至关重要，但是目前还没有挖掘这种信息的有效手段和经验法则，完全理解标记之间的相关性将成为多标记学习的圣杯。然而文献[60]认为如何学习有效特征比如何挖掘标记之间的关联性更重要，且如果特征足够好，标记之间将是相互独立的。这一观点也非常符合人类的认知习惯，例如我们在识别沙滩时并不会依赖于它是城市的概率是多少。作者通过一些理论和实验分析后指出，未来多标记学习方法的改进将会受益于更好的特征建模，而不是过分地建模标记之间的关联性。

集成学习和人工神经网络都是非常有效的机器学习方法，近年来正越来越受到人们的关注。集成学习在弱学习器提升为强学习器和强学习器的进一步增强中发挥着重要作用。一些好的多标记学习方法就运用到了集成策略[61]，如 RAkEL、ECC（Ensemble Classifier Chains）[49]和 RF-PCT[62]。人工神经网络（深度学习）是近年来最为火爆的人工智能技术，它同样可以应用在多标记学习场景中。BP-MLL 就是 BP 神经网络在多标记学习场景中的一种具体实现。文献[63]使用基于受限玻尔兹曼机(Restricted Boltzmann Machines，RBM)的深度学习方法研究了特征建模在多标记学习中的重要性。文献[60]也讨论了基于 RBM 的深度学习方法在多标记学习中的应用。

除了在图像、文本中的应用[64-73]，多标记学习在生物医学领域的应用也非常广泛，如下表列出了近几年发表的相关文献及其针对的具体问题。

表 1.1 多标记学习在生物医学领域的应用

问题	参考文献
蛋白质亚细胞定位	[31；74-77]
生物酶功能	[78；79]
膜蛋白功能	[80-83]

ATC 分类	[84]
医学文本中的 Disorder mention 识别	[85]
人类蛋白质交互网络中的活化/抑制关系预测	[86]
人类磷酸化蛋白质分类	[87]
蛋白质功能预测	[88-90]
抗药性预测	[91]
药物副作用预测	[92]
中医诊断	[93-97]
慢性病预测	[98; 99]

1.3 主要研究内容及创新点

1.3.1 本文主要研究内容

多标记学习属于机器学习的一个研究分支，在图像、文本、音乐、基因、蛋白质、医学文本数据挖掘中有着广泛的应用，近年来正受到研究人员越来越多的关注和兴趣。集成学习和深度学习是当前最流行的机器学习方法，本文将他们与多标记学习结合起来，并围绕多标记学习方面的算法创新和应用创新开展了如下工作：

（1）在集成学习框架下提出一种混合多标记学习算法 hMuLab，所构成的基学习器一个是特征驱动方法，另一个是近邻驱动方法，他们“好而不同”，相互补充并形成了一个有机整体，整体学习能力比任何基学习器都要强。在与多种已有的多标记学习算法比较中发现，hMuLab 具有显著的优势。

（2）提出了一种深度多标记学习算法，它采用多层次的前馈网络结构，隐含层使用 ReLU 激活函数。为满足多标记输出的需求，本文设计了两种多标记损失函数——多标记回归损失函数和多标记交叉熵损失函数，并推导了相应的迭代优化步骤。此外，文中还证明了多标记交叉熵与对数损失函数之间的关系。

（3）针对抗菌肽活性预测问题基于多标记学习方法进行了建模。文中创建了一个新的抗菌肽活性数据集，试验了多种抗菌肽分子特征提取方法，并比较了不同的多标记学习算法，发现使用氨基酸成分和二联体成分的组合特征，同时采用本文提出的多标记学习算法时效果最佳。

（4）基于多标记学习方法对慢性病预测问题进行了建模。文中讨论了电子病历样本的特征提取、缺失值处理及标准化问题。在一个包含 19733 个病人的多标记慢性病数据集上比较了 15 种多标记学习算法，最终结果显示，在这个较大的数据集上本文提出的深度多标记学习算法在准确性方面优势较大，同时其运行效率也较高。

1.3.2 创新点

本文的主要贡献包括：

（1）对多标记学习算法进行了较为详细的总结和深入的讨论，并对一些典型方法进行了图表化展示，使读者更容易理解，弥补了原文的不足；

（2）基于集成学习框架提出了一种混合多标记学习算法，并通过在多个数据集上与已有算法的比较验证了有效性；

（3）提出了深度多标记学习算法，其输出层采用了适宜于多标记输出的多标记损失函数，而隐含层则采用当前最流行的 ReLU 激活函数，通过在多个数据集上的实验比较验证了模型的有效性；

（4）将多标记学习算法应用于抗菌肽活性预测问题当中，比较了多种抗菌肽分子特征提取方法和多标记学习算法的效果，相关成果可以开发成一款抗菌肽药物分子筛选工具；

（5）将多标记学习算法应用于慢性病预测问题当中，在一个包含 19733 个病人的多标记慢性病数据集上比较了 15 种多标记学习算法，最终结果显示，在这个较大的数据集上本文提出的深度多标记学习算法优势较大。基于该方法，可以开发一款计算机辅助慢病诊断系统。

1.4 课题来源及论文结构

本文的工作相继得到深圳市海外高层次人才创新创业专项资金“个体化基因组差异检测的软硬件混合优化系统研究”、中国科学院战略性先导科技专项“专项门户网站及非模式生物系统标注”和国家 863 计划“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”的支持。在参与这些项目过程中，碰到了一些具有多标记特点的生物医学数据，于是在导师的指导下，开始开展多标记学习算法及其在生物医学数据挖掘中的应用研究。本课题属于人工智能与生物医学之间的交叉研究，数据和问题来源于生物医学领域，而采用的工具和方法是人工智能和机器学习。

论文第 1 章介绍了人工智能、机器学习、深度学习和多标记学习的概念和相互关系；简要概述了对他们的研究及应用现状，尤其是这些专业知识和技术在推动生物医学领域发展中的作用；最后介绍了本文的主要研究内容和创新点。

论文第 2 章着重介绍多标记学习理论，涉及它的形式化描述、常用评价指标、工作原理和已有算法的分类，并对已有的几种典型多标记学习算法用图表的方式直观地展现出来。

论文第 3 章在集成学习框架下提出了一种混合多标记学习算法。本章首先介绍了集成学习的原理和作用，然后详细介绍了本文提出方法的算法原理，接着通过实验来讨论模型超参数的影响，最后在多个多标记数据集上对多种算法进行了比较，验证本方法的优越性。

论文第 4 章将深度学习与多标记学习结合起来，提出一种深度多标记学习算法。本章首先介绍了多标记深度网络的基本架构和特点，接着推导了采用多标记损失函数时的

迭代优化步骤，随后对模型中的超参调节进行了较为详细的分析，最后在多个数据集上验证了本算法的有效性。

论文第 5 章对抗菌肽的活性预测问题进行了多标记学习建模。本章首先介绍抗菌肽的生物学知识和活性预测的意义，随后介绍了抗菌肽活性数据集的构建和抗菌肽分子的特征提取方法，最后在该数据集上比较了多种多标记学习算法。

论文第 6 章对慢性病预测问题进行了多标记学习建模。本章首先介绍慢性病的医学常识，接着介绍了一个慢性病数据集的构建和病人样本的特征提取方法，最后在该数据集上比较了多种多标记学习算法，包括预测精度和运行效率。

最后，对整篇论文进行了总结，并展望了下一步的研究计划。

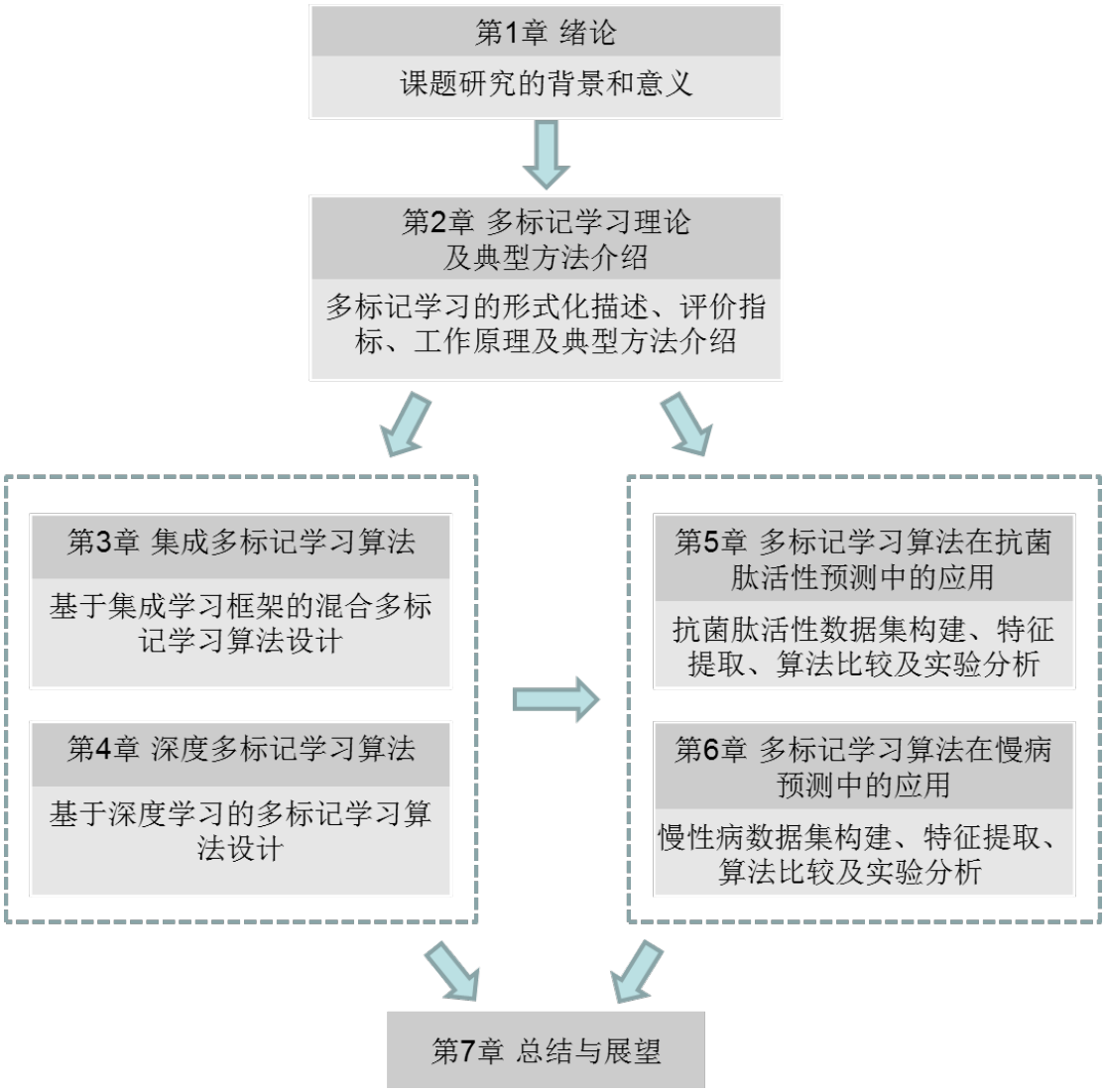


图 1.10 论文内容的逻辑组织图

第二章 多标记学习理论及典型方法介绍

2.1 引言

分类是最常见的机器学习方法，属于有监督学习的一种（下图）。一个分类系统的设计通常包含两个步骤，一个是训练，另一个是预测。在训练阶段，往往需要首先将训练样本表示成方便机器处理的特征向量，然后将带有标记的特征向量输入分类模型，并通过某种机器学习算法来优化分类模型，以求模型尽可能地拟合训练数据，同时有一定的泛化能力。在预测阶段，同样先对待测样本提取特征向量，然后输入到训练好的分类模型以获取它的预测标记。

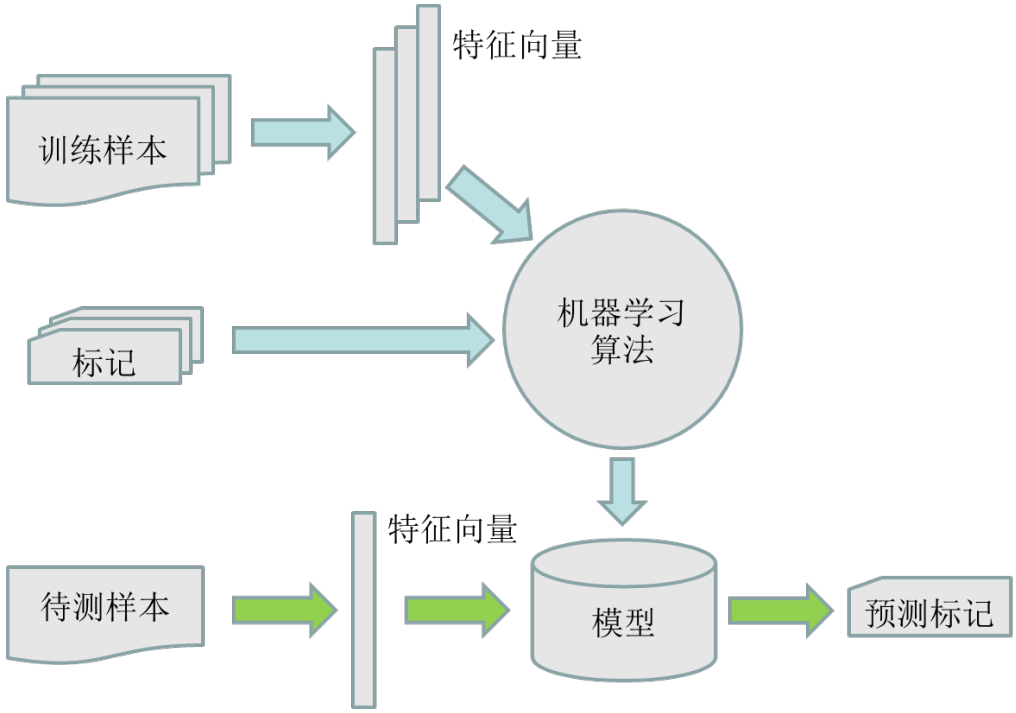


图 2.1 分类的一般流程

本文所说的多标记学习实际上指的是多标记分类，主要研究对于一个输入样本，如何获取所有可能跟它相关的标记，即将待测样本输入到分类系统中，输出的预测标记是一个标记集合，可能只有一个标记，也可能有多个标记。同时需要说明的是，本文研究的标记之间不存在层次关系，所有标记都是平等的。

2.2 多标记学习的形式化描述

在单标记学习场景中，每个样本都只具有一个标记，这样的样本称为单标记样本（single-label instance）。而在多标记学习场景中，每个样本可能具有不止一个标记，我们称这样的标记为多标记样本（multi-label instance）。一个多标记样本所具有的标记称为

该样本的相关标记（relevant label）或正标记（positive label），而不具有的标记称为样本的不相关标记（irrelevant label）或负标记（negative label）。给定一个输入样本时，多标记学习器要能够同时输出所有可能与之相关的标记。接下来分别给出单标记学习（single-label learning）和多标记学习（multi-label learning）的符号化定义：

单标记学习定义 假设$\Xi = \mathbf{R}^{d}$表示 d 维样本空间；$\Lambda = \{l_1, l_2, \ldots, l_c\}$为有限标记集合，共有 c（≥2）种可能的标记；给定由单标记样本构成的训练集 $D=\{(x_i, y_i)\|i=1, 2, ..., m\}$，其中 $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})^{T}\in\Xi$是样本空间中的一个特征向量，$y_i\in\Lambda$是与之关联的一个标记，单标记学习的任务就是基于训练集 D 学习到函数 $h: \Xi\rightarrow\Lambda$。对任意一个未知样本 $x\in\Xi$，单标记分类器 $h(\bullet)$能够预测出 x 可能具有的标记 $h(x)\in\Lambda$。当 c=2 就是单标记二分类（binary classification）问题；当 c>2 就是单标记多分类（multi-class classification）问题。
多标记学习定义 假设$\Xi = \mathbf{R}^{d}$表示 d 维样本空间；$\Lambda = \{l_1, l_2, \ldots, l_c\}$为有限标记集合，共有 c（≥2）种可能的标记；给定由多标记样本构成的训练集 $D=\{(x_i, y_i)\|i=1, 2, ..., m\}$，其中 $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})^{T}\in\Xi$是样本空间中的一个特征向量，$y_i\subseteq\Lambda$是与之关联的标记集合，多标记学习的任务就是基于训练集 D 学习到函数 $h: \Xi\rightarrow 2^{\Lambda}$（$\Lambda$的幂集）。对任意一个未知样本 $x\in\Xi$，多标记分类器 $h(\bullet)$能够预测出 x 可能具有的标记集合 $h(x)\subseteq\Lambda$。通常为了计算方便，将标记集合 y_i 表示成为一个二值向量$[y_{i,1}, y_{i,2}, \ldots, y_{i,c}]$，其中 $y_{i,j}$ 等于 1 时表示 l_j 是样本 x_i 的相关标记，等于 0（或 -1）时表示不是该样本的相关标记。

图 2.2 单标记学习和多标记学习的符号化定义

在大多数情况下，多标记学习算法会得到一个实值函数 $f: \Xi\times\Lambda\rightarrow\mathbf{R}$，其中 $f(x,l_j)$可以看作是样本 x 具有标记 $l_j\in\Lambda$的置信度。特别地，对于一个多标记样本(x, y)，$f(\bullet,\bullet)$对相关标记的输出值应该大于对不相关标记的输出值，即如果 $l'\in y$ 且 $l''\notin y$ 时，$f(x, l') > f(x, l'')$。这个时候多标记分类器 $h(\bullet)$可以通过阈值化实值函数 $f(\bullet,\bullet)$得到：

$$h(x) = \{l \mid f(x, l) > t(x), l\in\Lambda\} \tag{2.1}$$

其中 $t: \Xi\rightarrow\mathbf{R}$ 表示阈值函数，用于根据标记输出值将标记空间划分成为相关标记和不相关标记。

2.3 多标记学习的评价指标

在单标记学习中常用的评价指标有准确率（accuracy）、敏感度（sensitivity，或称召回率 recall）、特异性（specificity）、精度（precision）、ROC 曲线（receiver operating characteristic curve）和 AUC（Area under the Curve of ROC）等。但是在多标记学习中，由于对应每个样本的是一个标记集合，使得评价指标要复杂得多【】。总体来说，多标记学习中使用的评价指标可以分成两大类，分别是基于样本的评价指标（example-based metrics）、基于标记的评价指标（label-based metrics）和基于排序的评价指标（ranking-based

metrics），如下图所示。

图 2.3 多标记学习的评价指标

下面给出多标记评价指标的符号化定义。假定由 m’个多标记样本构成的测试集 S ={(x_i, y_i)|i=1, 2, ..., m’}，其中 x_i∈Ξ是样本空间中的一个特征向量，y_i⊆Λ是与之关联的标记集合，则从训练集 D 上得到的多标记学习器在该测试集上的表现用如下的评价指标来衡量：

（1）基于样本的评价指标分别考察学习器在每个样本上的表现，再取所有样本的平均结果。

- Hamming Loss:

$$HammingLoss = \frac{1}{m'}\sum_{i=1}^{m'}\frac{|h(\boldsymbol{x}_i)\Delta \boldsymbol{y}_i|}{c} \tag{2.2}$$

其中 Δ 表示取两个集合的对称差，|•|用于求集合的势（元素个数）。Hamming Loss 表示预测错误的标记（将相关标记预测为不相关标记，或相反）占所有标记的百分比。

- Subset Accuracy:

$$SubsetAccuracy = \frac{1}{m'}\sum_{i=1}^{m'}\mathrm{I}(h(\boldsymbol{x}_i) = \boldsymbol{y}_i) \tag{2.3}$$

其中 I(•)是指示函数，且 I(true)=1，I(false)=0。Subset Accuracy 表示在所有测试样本中，预测标记集合与真实标记集合相等的测试样本所占的百分比。

- Accuracy，Precision，Recall，F1-score：

$$Accuracy = \frac{1}{m'}\sum_{i=1}^{m'}\frac{|h(\boldsymbol{x}_i)\cap \boldsymbol{y}_i|}{|h(\boldsymbol{x}_i)\cup \boldsymbol{y}_i|};\quad \mathrm{Precision} = \frac{1}{m'}\sum_{i=1}^{m'}\frac{|h(\boldsymbol{x}_i)\cap \boldsymbol{y}_i|}{|h(\boldsymbol{x}_i)|}$$

$$\mathrm{Recall} = \frac{1}{m'}\sum_{i=1}^{m'}\frac{|h(\boldsymbol{x}_i)\cap \boldsymbol{y}_i|}{|\boldsymbol{y}_i|};\quad \mathrm{F1\text{-}score} = \frac{1}{m'}\sum_{i=1}^{m'}\frac{2|h(\boldsymbol{x}_i)\cap \boldsymbol{y}_i|}{|h(\boldsymbol{x}_i)|+|\boldsymbol{y}_i|} \tag{2.4}$$

其中∩表示求两个集合的交集，∪求两个集合的并集。

在上述指标中，Hamming Loss 值越小越好，最优值为 0；而其他指标值越大越好，最优值为 1。

（2）基于标记的评价指标分别考察学习器在每个标记上的表现，再取所有标记的平均结果。对于第 j 个标记，可以用 TP_j、FP_j、TN_j 和 FN_j 来表示学习器在该标记上的二分类结果

$$TP_j = |\{x_i \mid l_j \in y_i \wedge l_j \in h(x_i),\ 1 \le i \le m'\}|$$
$$FP_j = |\{x_i \mid l_j \notin y_i \wedge l_j \in h(x_i),\ 1 \le i \le m'\}|$$
$$TN_j = |\{x_i \mid l_j \notin y_i \wedge l_j \notin h(x_i),\ 1 \le i \le m'\}|$$
$$FN_j = |\{x_i \mid l_j \in y_i \wedge l_j \notin h(x_i),\ 1 \le i \le m'\}|$$

基于上述四个量，很多二分类评价指标都可以用在这里。假设 $B(TP_j,FP_j,TN_j,FN_j)$表示将某个二分类的评价指标 $B \in \{Accuracy, Precision, Recall, F\text{-}score\}$用于第 j 个标记，则基于标记的多标记评价指标有如下两种模式：

- Macro-averaging:

$$B_{macro} = \frac{1}{c}\sum_{j=1}^{c} B(TP_j, FP_j, TN_j, FN_j) \tag{2.5}$$

- Micro-averaging:

$$B_{macro} = B\left(\sum_{j=1}^{c} TP_j, \sum_{j=1}^{c} FP_j, \sum_{j=1}^{c} TN_j, \sum_{j=1}^{c} FN_j\right) \tag{2.6}$$

上述指标值越大越好，最优值为 1。

（3）基于标记排序的指标是根据标记输出值的排序关系来定义的。如果多标记学习器具有实值输出函数 $f(\bullet,\bullet)$，则可以将所有标记按照其输出值从大到小的顺序进行排序，并用 $r(x,l)$表示标记 l 的排序值。

- One-error:

$$OneError = \frac{1}{m'}\sum_{i=1}^{m'} I([\arg\max_{l \in L} f(\boldsymbol{x}_i, l)] \notin \boldsymbol{y}_i) \tag{2.7}$$

One-error 表示所有样本中，其最大输出值所对应的标记不属于相关标记的样本所占的比例。

- Coverage:

$$Coverage = \frac{1}{m'}\sum_{i=1}^{m'} \max_{l \in y_i} r(\boldsymbol{x}_i, l) - 1 \tag{2.8}$$

如果用深度来表示一个样本的所有相关标记的最大排序值，则 Coverage 就是所有样本的平均深度。

- Ranking Loss：

$$RankingLoss = \frac{1}{m'}\sum_{i=1}^{m'} \frac{1}{|\boldsymbol{y}_i||\bar{\boldsymbol{y}}_i|}\left|\{(l,l') \mid f(\boldsymbol{x}_i, l) \le f(\boldsymbol{x}_i, l'), (l,l') \in \boldsymbol{y}_i \times \bar{\boldsymbol{y}}_i\}\right| \tag{2.9}$$

上式中 $\overline{\boldsymbol{y}}_i$ 是 $\boldsymbol{y}_i$ 的补集，即样本 x_i 的不相关标记集合。对于单个样本而言，其排序损失值就是它的所有相关标记与不相关标记对中，发生排序错误（相关标记的输出值小于不相关标记的输出值）的百分比。因此 Ranking Loss 就是所有样本的平均排序损失值。

- Average precision：

$$AveragePrecision = \frac{1}{m'}\sum_{i=1}^{m'}\frac{1}{\left|\boldsymbol{y}_i\right|}\sum_{l\in\boldsymbol{y}_i}\frac{\left|\left\{l' \mid r(\boldsymbol{x}_i,l') \le r(\boldsymbol{x}_i,l), l' \in \boldsymbol{y}_i\right\}\right|}{r(\boldsymbol{x}_i,l)} \tag{2.10}$$

Average precision 用来统计事件——排在相关标记前面的也是相关标记的概率。

- macro-averaged AUC

$$AUC_{macro} = \frac{1}{c}\sum_{j=1}^{c}AUC_j = \frac{1}{c}\sum_{j=1}^{c}\frac{\left|\left\{(x',x'') \mid f(x',l_j) \ge f(x'',l_j), (x',x'') \in Z_j \times \overline{Z}_j\right\}\right|}{\left|Z_j\right|\left|\overline{Z}_j\right|} \tag{2.11}$$

上式中 $Z_j = \left\{x_i \mid l_j \in y_i, 1 \le i \le m'\right\}$，$\overline{Z}_j = \left\{x_i \mid l_j \notin y_i, 1 \le i \le m'\right\}$，分别表示具有标记 l_j 的样本集合和不具有标记 l_j 的样本集合。

- micro-averaged AUC

$$AUC_{micro} = \frac{\left|\left\{(x',x'',l',l'') \mid f(x',l') \ge f(x'',l''), (x',l') \in S', (x'',l'') \in S''\right\}\right|}{\left|S'\right|\left|S''\right|} \tag{2.12}$$

上式中 $S' = \left\{\left(x_i,l\right) \mid l \in y_i, 1 \le i \le m'\right\}$，$S'' = \left\{\left(x_i,l\right) \mid l \notin y_i, 1 \le i \le m'\right\}$，分别表示样本与相关标记对构成的集合和样本与不相关标记对构成的集合。

在上述指标中，One-error、Ranking Loss 和 Coverage 的值越小越好，其中前两者的最优值为 0，Coverage 的最优值为 $\frac{1}{m'}\sum_{i=1}^{m'}\left|\boldsymbol{y}_i\right| - 1$；而其他指标值越大越好，最优值为 1。

上述多标记学习评价指标从不同的角度来刻画多标记学习器的性能，而大多数的多标记学习算法都是通过直接或者间接地优化其中一个指标而来。最近的一些理论研究显示，当以 Subset Accuracy 作为优化目标的多标记学习器的性能较差，而以 Hamming Loss 作为优化目标时的性能较佳。因此，好的学习器需要选择合适的指标来作为优化目标，同时，要公平地比较多标记学习算法，也应该评估尽可能多的指标。

在单标记学习中，一般不会直接以 0-1 损失函数作为优化目标，而是以交叉熵（cross-entropy loss）、合页损失函数（hingle loss）、指数损失函数（exponential loss）等连续可导的凸函数作为 0-1 损失函数的代理损失函数（surrogate loss function）。同样，在多标记学习中，由于上述的多标记评价指标非凸且不连续，也需要使用代理损失函数。

2.4 多标记学习的工作原理及典型方法

目前，人们已经提出了很多算法来学习多标记数据，按照文献[]的做法，可以将这

些方法归为两大类：问题转化方法（Problem transformation methods）和算法适应方法（Algorithm adaptation methods）。问题转化方法的典型代表有 Binary Relevance（BR）、Classifier Chains（CC）、Ranking by Pairwise Comparison (RPC)、Calibrated Label Ranking（CLR）、Label Powerset（LP）、Random k-labelsets（RAkEL）和 label-specific features（LIFT）。算法适应方法的典型代表有 Multi-Label k-Nearest Neighbor（ML-kNN）、Multi-Label Decision Tree（ML-DT）、Ranking Support Vector Machine（Rank-SVM）和 Backpropagation for Multi-Label Learning（BPMLL）。

图 2.4 多标记学习算法的分类

2.4.1 问题转化方法

问题转化方法将多标记学习任务转化成单标记学习任务，这时大量已有的单标记学习算法可以用在这里，从而间接完成对多标记数据的处理。通俗来讲，这类方法是用数据来适应算法，不需要改变已有的分类算法，而是直接调用或者组合。

为了更好地说明问题转化方法是如何工作的，下面给出一个简单的多标记数据集，包含 4 个样本 3 种标记，如下表所示。

样本	标记集合
x_1	$\{l_1, l_2, l_3\}$
x_2	$\{l_1\}$
x_3	$\{l_2, l_3\}$
x_4	$\{l_3\}$

图 2.5 一个简单的多标记数据集

- 简单转化

最简单的转化方法是在维持原有标记的基础上把多标记数据集转化成单标记数据集，这时候面临的问题是该如何处理相关标记多于一个的样本，通常有 3 种机制可以用来解决这个问题：复制（copy）、忽略（ignore）和选择（select）[47; 48; 100]。采用复制机制时，每个多标记样本(x,y)都被替换成$|y|$个单标记样本$\{(x,l) \mid l \in y\}$。采用忽略机制时，那些相关标记多于一个的样本将直接从数据集中剔除，只保留那些相关标记仅有一个的样本。采用选择机制时，如果一个样本的相关标记多于一个，则从它的相关标记集中只选择一个作为该样本的唯一标记。选择方法可以是：最大选择（保留数据集中出现最多

的标记)、最小选择（保留数据集中出现最少的标记）和随机选择（随机挑选一个标记），处理后的结果数据集如图(c)~(e)所示。

样本	标记
x_{1a}	l_1
x_{1b}	l_2
x_{1c}	l_3
x_2	l_1
x_{3a}	l_2
x_{3b}	l_3
x_4	l_3

(a)

样本	标记
x_2	l_1
x_4	l_3

(b)

样本	标记
x_1	l_3
x_2	l_1
x_3	l_3
x_4	l_3

(c)

样本	标记
x_1	l_1
x_2	l_1
x_3	l_2
x_4	l_3

(d)

样本	标记
x_1	l_2
x_2	l_1
x_3	l_3
x_4	l_3

(e)

图 2.6 将多标记数据转化成单标记样本的简单方法

(a) 复制，(b) 忽略，(c) 最大选择，(d) 最小选择，(e)随机选择

这种方法简单粗暴，当数据集中的多标记样本非常少时，也许无关紧要，但是当多标记样本的比重很大时，采用这种方法会丢失很多信息。

- Binary Relevance（BR）

BR 方法是一种非常流行的转化方法，它将多标记数据集分解成 c 个二分类数据集 $D_j = \{(x_i, I(l_j \in y_i)) | 1 \le i \le m\}$，$1 \le j \le c$，并根据数据集 D_j 学习一个二分类器来判断输入样本是否具有标记 l_j。最后将 c 个二分类的预测结果联合起来形成输入样本的预测标记集合。

样本	标记
x_1	l_1
x_2	l_1
x_3	$\sim l_1$
x_4	$\sim l_1$

样本	标记
x_1	l_2
x_2	$\sim l_2$
x_3	l_2
x_4	l_3

样本	标记
x_1	l_3
x_2	$\sim l_3$
x_3	l_3
x_4	l_3

图 2.7 BR 方法产生的多个二分类数据集

~表示非操作

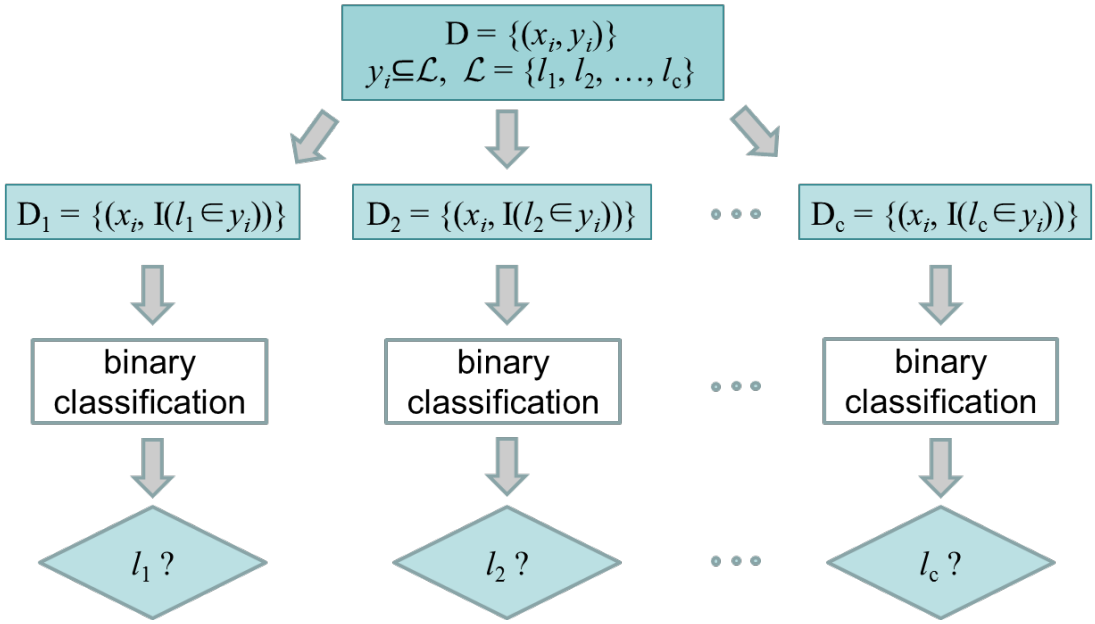


图 2.8 BR 方法工作流程

- Ranking by Pairwise Comparison (RPC)

RPC 方法将多标记数据集转化成 c(c-1)/2 个二分类数据集[50]，每个数据集包含一对标记(l_j, l_k): $D_{j,k} = \{(x_i, I(l_j \in y_i)) | l_j \in y_i \vee l_k \in y_i \wedge \{l_j, l_k\} \subseteq y_i, 1 \le i \le m\}, 1 \le j < k \le c$。根据 $D_{j,k}$ 学习的二分类器可判断输入样本是具有标记 l_j 还是 l_k。对于任意输入样本，所有 c(c-1)/2 个二分类器都将被激活，并输出 c(c-1)/2 个预测标记，根据每个标记的投票情况将所有标记排序，投票最多的几个标记作为输入样本的预测标记集合。如多标记成对感知机算法（multi-label pairwise perceptron, MLPP）就是 RPC 策略的具体应用[101]。另外 Calibrated label ranking (CLR) [51]对 RPC 进行了扩展，又通过 BR 方法引入了一个虚拟标记，将排序好的标记划分成相关标记和不相关标记。

样本	标记
x_2	l_1
x_3	l_2

(a)

样本	标记
x_2	l_1
x_3	l_3
x_4	l_3

(b)

样本	标记
x_4	l_3

(c)

图 2.9 使用 RPC 方法将原始数据集转化成多个二分类数据集

(a) 数据集 $D_{1,2}$，(b) 数据集 $D_{1,3}$，(c) 数据集 $D_{2,3}$

这种方法简单直观，很多已有的性能优异的二分类算法可以直接用在这里，无需进行进一步的算法设计。但是这种方法的缺点是对标记之间的关系做了独立性假设，无法挖掘出标记之间的相关性；此外，最终转化的二分类数据集可能会面临极度不平衡的状况，对二分类算法造成很大的困扰。

- Label Powerset (LP)

LP 方法将多标记数据集中每个特有的标记集合当作一个标记看待，从而将多标记数

据集转化成为一个单标记数据集，这时再运用单标记多分类算法来完成学习任务。对任意一个输入样本，LP 方法可以直接预测出它的标记组合

样本	标记
x_1	$l_{1,2,3}$
x_2	l_1
x_3	$l_{2,3}$
x_4	l_3

图 2.10 LP 方法转化的数据集

这种方法很自然地将标记之间的相关性引入到转化的数据集中，但是它有一个致命缺陷，就是只有训练集中已有的标记组合才能被预测出来；且通常比较大的标记集合所对应的样本很少，这将造成最终的转化数据集也会出现非常不均衡的情况。

Random k-Labelsets（RAkEL）是对 LP 方法的深度改进，它的基本思想是将多标记学习问题转化成集成单标记学习问题[53]。每个基学习器只处理一个大小为 k（1<k<c）的标记子集，基于该标记子集使用 LP 方法将原始数据集转化成单标记数据集，并训练出一个多分类器。每个标记子集都是从原始标记集合中随机选择得到的，如果共进行 T 次随机选择，就可以生成 T 个学习器。对于一个输入样本，T 个学习器会得到 T 个预测标记组合，从中统计出每个标记的得票数，并使用下式得到它的最终预测标记集合：

$$h(x) = \{l_j \mid \alpha(x, l_j) / \beta(x, l_j) > 0.5,\ 1 \le j \le c\} \tag{2.13}$$

其中 $\alpha(x, l_j)$表示标记 l_j 的实际得票数，$\beta(x, l_j)$表示标记 l_j 从所有分类器中所能得到的总得票数。因此上式表示当一个标记的得票数超过它所能获得的投票数目的一半时，就认为该标记是相关标记。

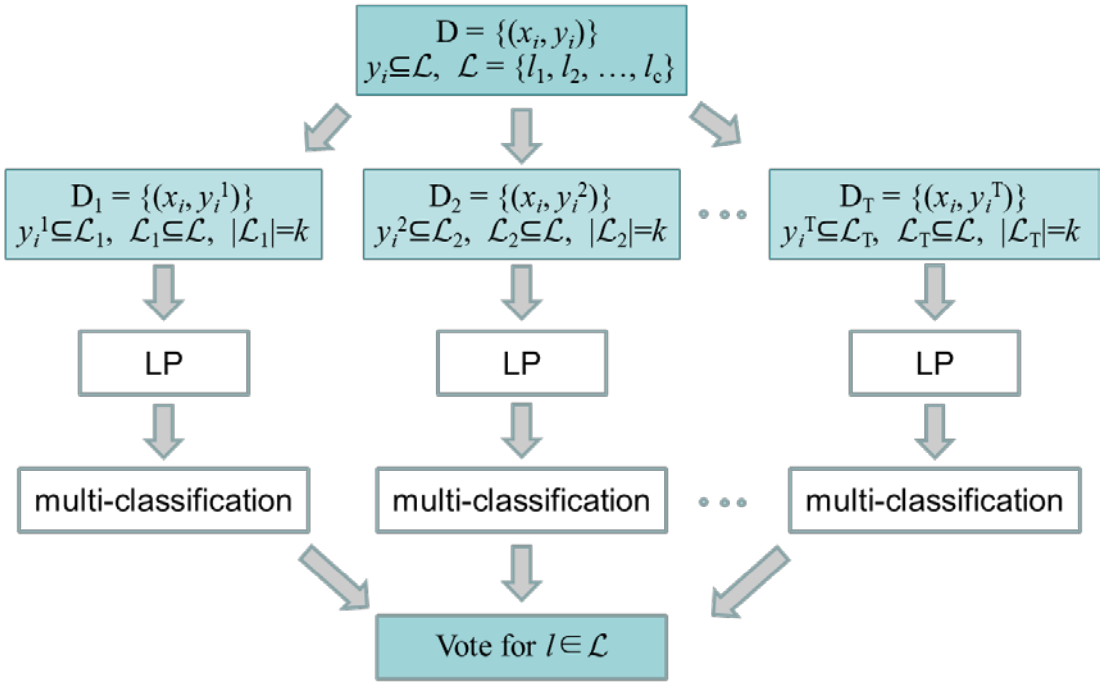


图 2.11 LP 方法原理图

- Classifier Chains（CC）

Classifier Chains 的基本思想是将多标记学习问题转化成一个二分类器链，每个二分类器只负责一个标记的学习和预测，但是区别于 BR 方法的是，链中的每个二分类器会受到排在它前面的二分类器的影响。具体来说，首先将原始的所有标记随机排序：l_1', l_2', …, l_c'，针对标记 l_1'建立数据集 $D_1 = \{(x_i, I(l_1' \in y_i)) \mid 1 \le i \le m\}$，基于该数据集生成二分类器 $h_1(x)$，接着将第一个二分类器的预测结果与原始特征级联起来，并针对第二个标记 l_2'建立数据集 $D_2 = \{([x_i, p_i^1], I(l_2' \in y_i)) \mid 1 \le i \le m\}$，然后基于 D_2 生成二分类器 $h_2(x)$；依次类推，直到建立最后一个数据集 $D_c = \{([x_i, p_i^1, \ldots, p_i^{c-1}], I(l_c' \in y_i)) \mid 1 \le i \le m\}$，并生成最后一个二分类器 $h_c(x)$，其中 $p_i^1 = I(h_1(x_i)=l_1')$, $p_i^{c-1} = I(h_{c-1}(x_i)=l_{c-1}')$。

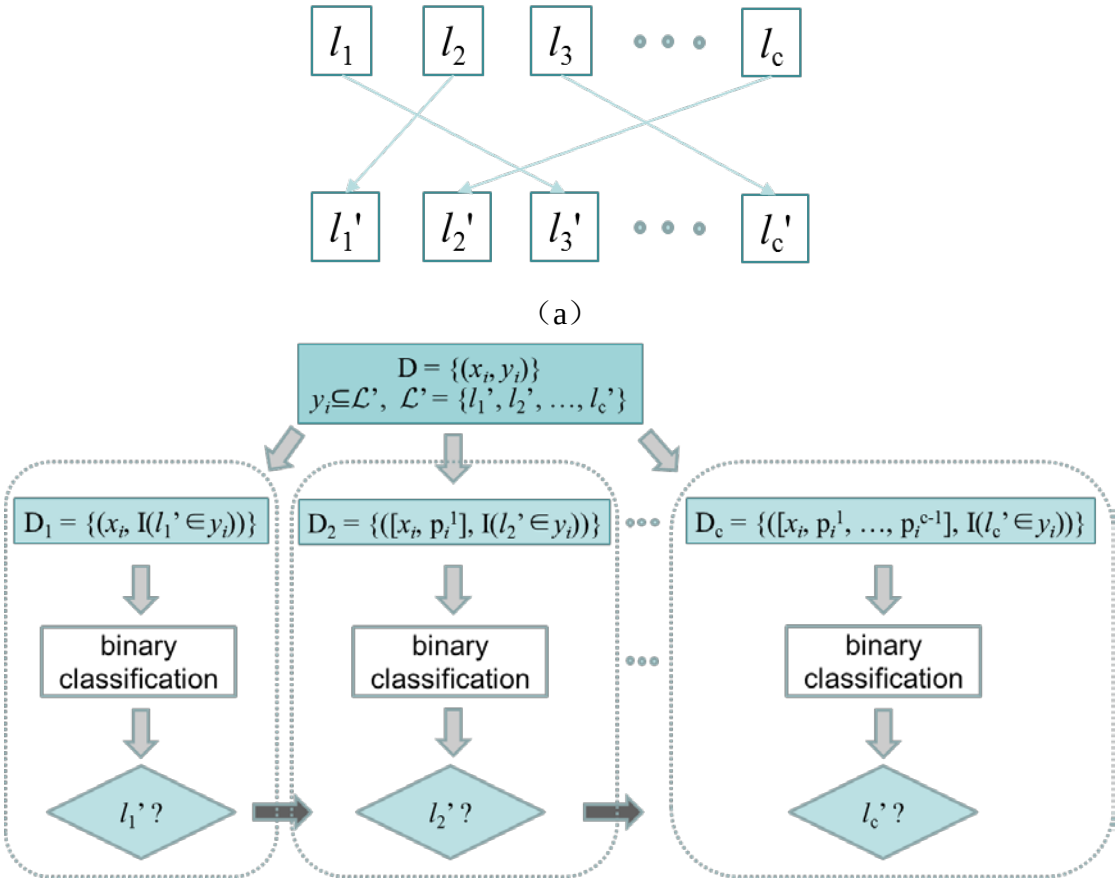


图 2.12 Classifier Chains 工作流程

(a)将原始标记随机排序，(b)依次建立二分类器

从上面的过程来看，Classifier Chains 的性能将会受到标记的排序影响，为减少随机性，Ensemble of Classifier Chains（ECC）对标记做 T 次的随机排序，并建立 T 条分类器链，然后根据 T 条分类器链的集成结果来确定最终的标记集合。

相比于 BR 方法，Classifier Chains 考虑了标记之间的相互影响。但是由于链具有顺序性，使得这种方法无法像 BR 方法那样并行实现。

- Label-Specific Features（LIFT）

一般情况下，已有的多标记学习算法在对不同的标记进行学习时都是基于同样的特

征空间进行的。然而 LIFT 方法认为上述策略可能只是次优的，因为不同的标记具有自身的特性，因此需要特异性的特征。例如在图像自动注释中，基于颜色的特征对区分天空图片和非天空图片很有用，而基于纹理的特征对区分沙漠图片和非沙漠图片更有效。基于上述思想，作者指出应使用标记特异性特征进行不同的标记学习，并提出了一种非常简单的方案。具体来说，首先使用 BR 方法将多标记数据集分解成 c 个二分类数据集 $D_j = \{(x_i, I(l_j \in y_i)) | 1 \le i \le m\}$，$1 \le j \le c$。对每个数据集 D_j，将其进一步分成正负样本集：$P_j = \{(x_i \mid l_j \in y_i), 1 \le i \le m\}$, $N_j = \{(x_i \mid l_j \notin y_i), 1 \le i \le m\}$。分别对 P_j 和 N_j 进行 K 均值聚类，各形成 K_j 个聚类中心：$\{p_1^j, p_2^j, \ldots, p_{Kj}^j\}$, $\{n_1^j, n_2^j, \ldots, n_{Kj}^j\}$。再根据样本 x 到这 $2K_j$ 个中心的距离将 x 映射为 $\Phi_j(x) = [d(x, p_1^j), \ldots, d(x, p_{Kj}^j), d(x, n_1^j), \ldots, d(x, n_{Kj}^j)]$，其中 $d(\bullet,\bullet)$表示两点之间的欧氏距离。最后，在映射空间 $\Phi_j(x)$，$1 \le j \le c$ 中使用二分类算法进行标记 l_j 的学习即可。

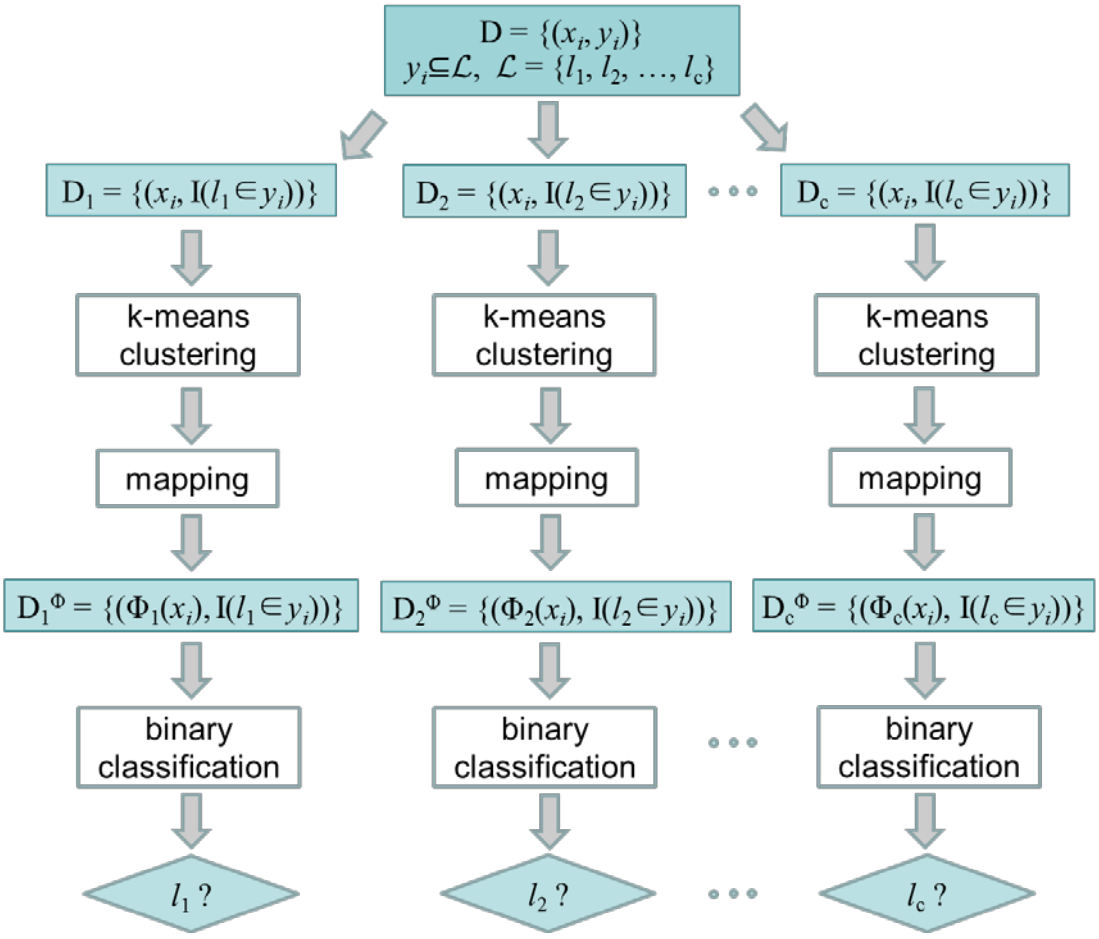


图 2.13 LIFT 原理图

2.4.2 算法适应方法

算法适应方法则是对已有的学习算法进行扩展后直接用来处理多标记数据，通俗地讲就是用算法来适应数据，通常需要将已有的经典分类算法进行适度调整或者干脆发明出新的方法出来，如：

- Multi-Label k-Nearest Neighbor (ML-kNN)

ML-kNN 的基本思想是采用近邻法来预测待测样本的标记。但是不同于经典的 kNN 方法中直接使用近邻所具有的标记进行投票，ML-kNN 首先得到近邻的标记信息，再结合最大后验概率规则（maximum a posteriori rule，MAP）进行标记推断。具体来说，对于一个输入样本 x，算法的第一步需要在训练集中找出它的 k 个近邻：$\{(x_i^*, y_i^* \mid 1 \le i \le k)\}$，并统计出具有标记 l_j 的近邻数：

$$\alpha_j = \sum_{i=1}^{k} \mathrm{I}(l_j \in \mathrm{y}_i^*)\text{，} \quad 1 \leqslant \mathrm{j} \leqslant \mathrm{c} \tag{2.14}$$

用 $P(l_j|\alpha_j)$表示 x 在 α_j 个近邻具有标记 l_j 的条件下具有标记 l_j 的后验概率，$P(\sim l_j|\alpha_j)$表示 x 在 α_j 个近邻具有标记 l_j 的条件下不具有标记 l_j 的后验概率，则根据 MAP 规则，x 的预测标记集合通过下式确定：

$$h(x) = \{l_j \mid P(l_j|\alpha_j)/P(\sim l_j|\alpha_j) > 1, 1 \le j \le c\} \tag{2.15}$$

根据 Bayes 理论，有如下关系：

$$\frac{P\left(l_j \mid \alpha_j\right)}{P\left(\sim l_j \mid \alpha_j\right)} = \frac{P\left(l_j\right) \bullet P\left(\alpha_j \mid l_j\right)}{P\left(\sim l_j\right) \bullet P\left(\alpha_j \mid \sim l_j\right)} \quad 1 \leqslant j \leqslant \mathrm{c} \tag{2.16}$$

上式中 $P(l_j)$和 $P(\sim l_j)$分别表示样本具有标记 l_j 或这不具有标记 l_j 的先验概率，$P(\alpha_j|l_j)$表示样本具有标记 l_j 时恰好有 α_j 个近邻也具有标记 l_j 的似然率，$P(\alpha_j|\sim l_j)$表示样本不具有标记 l_j 时却有 α_j 个近邻具有标记 l_j 的似然率。这四个量都是可以通过对训练集统计得到。因此，不同于传统的 kNN 算法不需要显式的训练过程，ML-kNN 是有一个训练步骤的，这就决定了这种算法比普通的近邻法更慢，对数据的可扩展性较差。

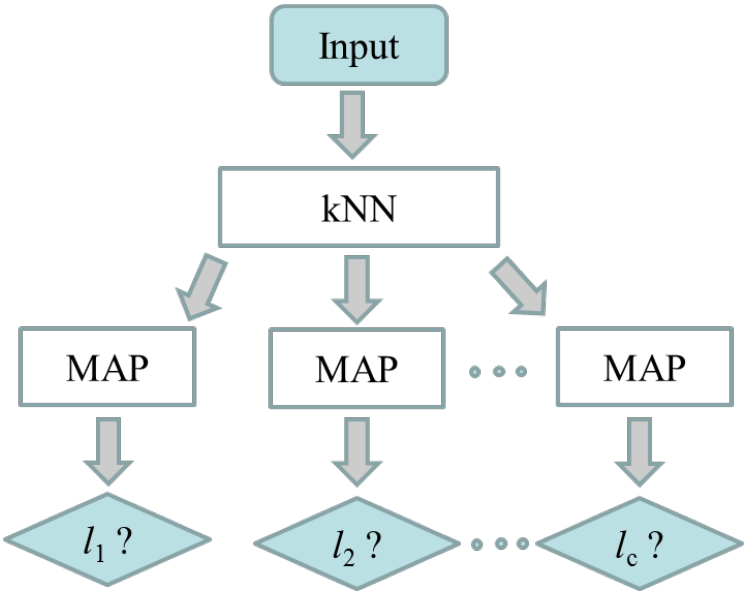


图 2.14 MLkNN 原理图

- Multi-Label Decision Tree (ML-DT)

ML-DT 的基本思想是采用决策树方法处理多标记数据。不同于传统的决策树方法，它在计算信息增益的时候使用的是多标记熵（multi-label entropy，MLE）。

具体来说，给定一个多标记数据集 D={(x_i, y_i)|i=1, 2, ..., m}，在第 a 个特征上的 s 点处将数据集 D 切分成两部分时的信息增益为

$$\mathrm{IG}(D,a,s)=\mathrm{MLE}(D)-\sum_{i=1}^{2}\frac{|D_i|}{|D|}\bullet\mathrm{MLE}(D_i) \tag{2.17}$$

其中 D_1 = {(x_i, y_i) | $x_{i,a}$≤s，1≤i≤m}，D_2 = {(x_i, y_i) | $x_{i,a}$>s，1≤i≤m}，即 D_1（D_2）数据集包含了在第 a 维特征值小于（大于）s 的样本。

从根结点开始，ML-DT 选择最大化信息增益的特征和剪切点，并将原始数据集切分成两个子集 D_1 和 D_2。接着分别以这两个子集作为新的根结点，做进一步的数据集划分。上述过程将被递归地执行直到满足树生长的停止规则，如信息增益过小或者子结点上的样本过少。

对于一个未知样本 x，根据其特征值在生成的决策树上从根结点开始遍历直到叶结点，每个叶结点上都存储着一些多标记样本，根据这些样本的标记对 x 作出如下预测

$$h(x)=\{l_j\,|\,p_j>0.5,\ 1\le j\le c\} \tag{2.18}$$

换句话说，如果样本 x 被分配到某个叶结点后，而落在这个叶结点上的大多数训练样本都具有标记 l_j，则认为 l_j 也是 x 的相关标记。

- Ranking Support Vector Machine (Rank-SVM)

Rank-SVM 的基本思想是采用支持向量机来处理多标记数据，但是它不是简单地对每个标记分别采用 SVM 进行学习（BR-SVM），而是同时考虑所有标记，并试图最小化相关标记与不相关标记的排序损失（ranking loss）。

具体来说，假设学习系统中有 c 个线性分类器 W={(w_j,b_j) | 1≤j≤c }，其中 w_j∈$\mathbf{R}^d$，b_j∈$\mathbf{R}$ 表示标记 l_j 的权重向量和偏置。Rank-SVM 在优化系统参数时考虑了对训练样本的相关标记与不相关标记的排序能力。

$$\min_{(x_i,y_i)\in D}\ \min_{(l_j,l_k)\in y_i\times\overline{y}_i}\frac{\left\langle w_j-w_k,x_i\right\rangle+b_j-b_k}{\left\|w_j-w_k\right\|} \tag{2.19}$$

通过一定简化，上述问题可以用下面的优化问题来近似

$$\min_{W}\sum_{j=1}^{c}\left\|w_j\right\|^2$$

$$\text{Subject to: }\left\langle w_j-w_k,x_i\right\rangle+b_j-b_k\ge 1,\ 1\le i\le m,\ \left(l_j,l_k\right)\in y_i\times\overline{y}_i \tag{2.20}$$

类似于经典的 SVM 方法，对线性不可分的情况可以在优化问题中引入松弛变量（slack variables），而为了提高非线性的学习能力，可以引入核技巧（kernel trick）。另外，Rank-SVM 依然是一个标准的凸二次规划问题（quadratic programming，QP），并可以采用二次规划方法来求解。

- Backpropagation for Multi-Label Learning（BP-MLL）

BP-MLL 是基本思想是运用 BP 神经网络来处理多标记数据，其创新性在于采用了

新的损失函数，主要考虑对相关标记的输出应大于对不相关标记的输出。

$$\min \quad J(\theta)=\sum_{i=1}^{m}\frac{1}{|y_i||\overline{y}_i|}\sum_{(l_p,l_n)\in y_i\times\overline{y}_i}e^{-(o_p^i-o_n^i)} \tag{2.21}$$

其中 o_p^i 和 o_n^i 分别表示网络对训练样本 x_i 的相关标记 l_p 和不相关标记 l_n 的实际输出，取决于样本 x_i 和当前网络参数值 θ。

针对上述损失函数，使用误差反向传播算法即可求得优化的网络参数值 θ。

2.5 本章小结

本章主要研究多标记学习的理论和方法。首先用数学语言对多标记学习进行了形式化的描述；接着介绍了常见的多标记学习评价指标，并对它们做了通俗化释义；最后说明了多标记学习的工作原理，并对几种典型的多标记学习算法进行了概述。本章的最主要亮点是对一些多标记学习算法的工作机制进行了图表化的阐述，使复杂的算法显得清晰直观，弥补了原文的不足。

第三章 一种集成多标记学习算法

3.1 引言

俗话说“三个臭皮匠顶一个诸葛亮”，这一思想在机器学习领域也有体现，如集成学习（Ensemble learning）方法就是同时运用多个个体学习器进行综合决策以期取得比任何单一学习器都更好的预测性能，这种方法有时也被称作基于委员会的学习（committee-based learning）等[102; 103]。

目前，集成学习已广泛应用于几乎所有的机器学习任务当中，包括有监督学习、无监督学习[104; 105]和半监督学习[106]，数据挖掘领域的国际顶级竞赛 KDD-CUP 历年的冠军几乎都用到了集成学习。下面以有监督学习为例对集成学习进行简要的介绍。有监督集成学习的一般架构如图 3.1 所示，首先我们要获得多个个体学习器，然后对于任意输入样本，将其输入每个个体学习器并得到个体输出，最后通过某种集成策略将所有个体输出结合起来以产生最终的输出。如果执行的是回归任务，则个体学习器可以是各种回归学习器，如普通的线性回归、决策树回归或神经网络回归，如果是分类任务，则个体学习器可以是各种有监督的分类器，如逻辑斯蒂回归、决策树、近邻分类器、支持向量机、神经网络等等。用来组成集成学习器的个体学习器通常是同一类型的学习器，如决策树集成中采用的都是决策树，这样的集成称作“同质集成”，集成用的个体学习器称为“基学习器”。用来组成集成学习器的个体学习器也可以是不同类型的学习器，如有的个体学习器采用决策树，而有的个体学习器采用神经网络，这样的集成称为“异质集成”，集成用的个体学习器称为“组件学习器”[6]。

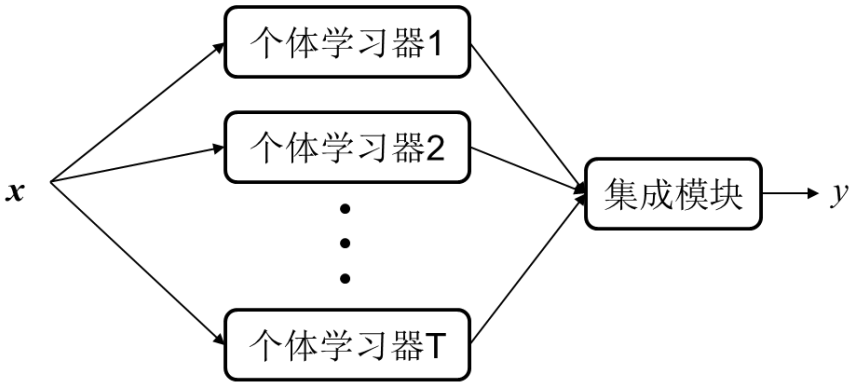


图 3.1 集成学习的一般架构图

有监督学习算法通常被描述成在一个假设空间中搜索到一个最合适的假设，这个假设能针对特定的问题做出最好的预测。尽管假设空间中可能存在非常适合的假设，但往往找到它是非常困难的。集成方法是通过联合多个假设以期形成一个更好的假设，这个新的假设甚至有可能并不在我们的假设空间当中。正因为如此，集成方法在函数表达方

面具有更大的灵活性[107-109]。

集成学习的目的是将多个性能较弱的个体学习器结合起来以产生一个比任一个个体学习器性能都要好的强学习器，那么是不是将任意的个体学习器进行随意的结合都能达到这样的目的呢？我们来看图 3.2 所示的三个简单例子。（a）图中，三个学习器的性能都比较好（66.6%的准确率），且三个学习器的预测结果差异比较大。这样的三个学习器进行简单投票集成后的最终输出完全正确，达到了 100%的预测准确率；（b）图中，三个学习器的性能也都比较好，但三个学习器的预测结果完全一致，使得最终的集成性能没有任何的提升；（c）图中，虽然三个学习器的预测结果差异比较大，但是由于三个学习器的性能都太差（33.3%的准确率），以至于经过集成后的最终预测结果完全错误，这样的性能比单个学习器的还要降低了。通过上述的简单示例可知，并不是任意的个体学习器经过随意的结合都会提升性能，好的个体学习器应该是“好而不同”，即每个个体自身应有较好的性能，同时个体之间又有较大的差异性，能够相互补充。

	测试样本 1	测试样本 2	测试样本 3
分类器 1	√	√	×
分类器 2	×	√	√
分类器 3	√	×	√
集成	√	√	√

(a)

	测试样本 1	测试样本 2	测试样本 3
分类器 1	√	√	×
分类器 2	√	√	×
分类器 3	√	√	×
集成	√	√	×

(b)

	测试样本 1	测试样本 2	测试样本 3
分类器 1	√	×	×
分类器 2	×	√	×
分类器 3	×	×	√
集成	×	×	×

(c)

图 3.2 集成用个体学习器应“好而不同”

(a)集成提升性能，(b)集成不起作用，(c)集成降低性能

在前面对集成学习的架构进行介绍时，并没有说明集成用的个体学习器是如何获得的。按照个体学习器的生成方式，可以将集成学习分为两大类：串行集成学习和并行集成学习。串行集成中个体学习器依次串行生成，相互依赖，典型代表是 Boosting[110; 111]，

这一类型的集成方法的工作原理是：首先在原始数据集上生成一个基分类器，然后根据基分类器的表现对训练样本的概率分布（或权值分布）做出调整，并在调整后的训练集上生成新的基分类器，重复执行这个过程，直到生成足够多的基分类器，最后将这些基分类器加权融合以形成最终的集成分类器。Boosting 方法的最著名代表当属 Adaboost（Adaptive Boosting）算法[112]。并行集成中个体学习器之间不存在依赖关系，所有个体学习器都是独立生成的，典型代表如 Bagging（bootstrap aggregating）[113]，它是在原始训练集上进行多次自助采样以生成多个采样集，并在每个采样集上生成一个基学习器，最后将这些基学习器结合起来形成集成学习器，结合的方法可以是投票（针对分类任务）或平均（针对回归任务）。

集成学习在多标记学习任务中也已经有了很多的成功应用，如 RAkEL，ECC。本研究将提出一种新的集成多标记学习算法，由于集成用的个体分类器是由两种完全不同的学习算法得到的，因此我们称之为混合多标记分类器（hMuLab, Hybrid MUlti-LABel classifier）

3.2 一种混合多标记分类方法

本文所提出的新型多标记分类算法是一种集成特征驱动方法和近邻驱动方法的混合方法，两种方法以并行的方式结合，算法流程如图 3.3 所示。对一个待测样本，特征驱动方法直接根据它的特征信息来评估该样本对所有类标记的隶属度，而近邻驱动方法首先找出该样本的近邻样本，并基于近邻的标记信息来评估它对所有标记的隶属度，而最终的结果输出则是这两种隶属度的加权平均值。无论是特征驱动方法还是近邻驱动方法，他们本身就是一个独立的多标记学习算法，且这两种方法在测试中的表现都还不错。从工作原理上来看，这两种方法又显著不同，一个采用的是全局信息，构造的决策面为线性，另一个采用的是局部信息，构造出的决策面是非线性的。两者“好而不同”，相互补充，我们相信将他们结合起来会形成一个更好的多标记分类器。

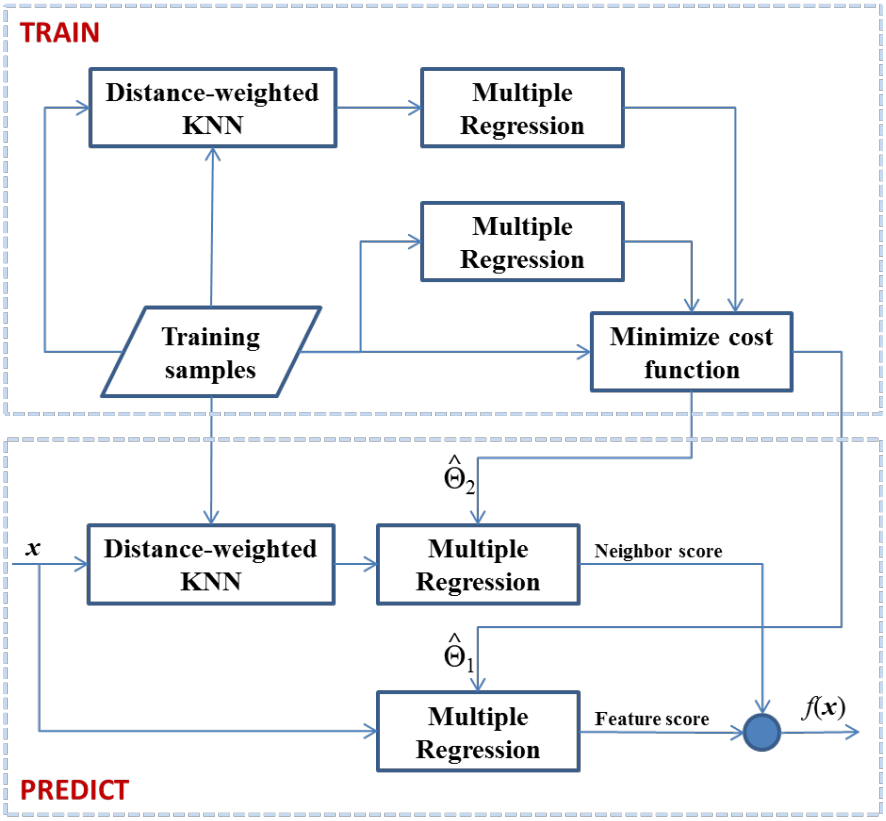


图 3.3 hMuLab 算法流程图

3.2.1 特征驱动方法

特征驱动方法直接用多重回归模型来计算待测样本 x 对所有类标记的隶属度。假设 X_1 是样本矩阵，其中第 i 行表示训练样本 x_i 的增广特征向量，表示为$[1, x_{i1}, x_{i2}, ..., x_{id}]^T$。Y 表示类标记矩阵，其中第 i 行是训练样本 x_i 的类标记向量 y_i，则多输出回归模型定义为

$$\underset{n\times c}{\mathrm{Y}} = \underset{n\times(d+1)}{\mathrm{X}_1} \times \underset{(d+1)\times c}{\Theta_1} + \underset{n\times c}{\mathrm{U}} \tag{3.1}$$

上式中 Θ_1 是回归系数矩阵，U 是残差矩阵，矩阵符号的下标代表矩阵的行列数。回归系数通过最小化残差平方和来计算

$$\begin{aligned} \min J(\Theta_1) &= \frac{1}{2}\left\|\mathrm{Y}-\mathrm{X}_1\Theta_1\right\|_F^2 + \frac{\lambda}{2}\left\|\Theta_1\right\|_F^2 \\ &= tr\left\{\frac{1}{2}\left(\mathrm{Y}-\mathrm{X}_1\Theta_1\right)^{\mathrm{T}}\left(\mathrm{Y}-\mathrm{X}_1\Theta_1\right)+\frac{\lambda}{2}\Theta_1{}^{\mathrm{T}}\Theta_1\right\} \end{aligned} \tag{3.2}$$

上式中 T 是转置操作，$\|\bullet\|_F$ 代表 Frobenius 范数，tr 用来计算矩阵的迹。公式的右边第一项是误差平方和，第二项是正则项，用于减小参数值并降低过拟合风险。非负参数 λ 是这两项的折衷，其值较大时，后一项将在优化目标中起到更大的作用，其值较小时，前一项将更重要。回归系数矩阵 Θ_1 是通过求导来获得，即令 $\nabla_{\Theta_1} J(\Theta_1) = 0$ 来得到如下最

优化参数值

$$\hat{\Theta}_1 = (\mathrm{X}_1^{\mathrm{T}}\mathrm{X}_1 + \lambda\mathrm{I})^{-1}\mathrm{X}_1^{\mathrm{T}}\mathrm{Y} \tag{3.3}$$

其中 I 表示单位阵。由于上述最优化问题属于凸优化，根据最优化理论，驻点即为全局最优点。需要说明的是，有些情况下 $\mathrm{X}_1^{\mathrm{T}}\mathrm{X}_1$ 是不可逆的，但是当 λ 足够大时，$(\mathrm{X}_1^{\mathrm{T}}\mathrm{X}_1 + \lambda\mathrm{I})$ 总是可逆的。有了上面的最优化参数矩阵，再有未知样本的增广特征向量，我们很容易利用下式来计算该未知样本对所有类标记的输出

$$f_1(\boldsymbol{x}, l_j) = \boldsymbol{x}^{\mathrm{T}}\hat{\Theta}_1^j,\ 1 \le j \le c \tag{3.4}$$

其中 $\hat{\Theta}_1^j$ 表示参数矩阵 $\hat{\Theta}_1$ 的第 j 列。在最优化残差平方和的框架下，如果一个样本具有标记 l_j，则它对该标记的输出将趋于 1。相反，如果它不具有该标记，则它对于该标记的输出趋于-1。

3.2.2 近邻驱动方法

近邻驱动方法首先在训练集中找出待测样本的近邻，并根据近邻样本的标记信息计算出待测样本对所有标记的得分，再利用回归模型计算出待测样本对所有标记的最终输出值。K 近邻算法由于简单性、直观性和有效性而成为应用最为广泛的有监督学习算法之一[114]。权重 K 近邻方法是近邻法的一种改进方法[115; 116]。不同于经典的 K 近邻算法中所有近邻投票的权重都是一样的，它为待测样本 x 的近邻赋予不同的投票权重，权重的大小根据待测样本与近邻之间的距离来确定。从训练集 D 中找出的 K 个近邻及其类标记记为(x_k^*, y_k^*)，其中 $1 \le k \le K$，这 K 个近邻与 x 之间的距离为 d_k，且根据值的大小排序为 $d_1 \le d_2 \le ... \le d_K$，则 x 的第 k 个近邻的权重定义为

$$w_k = \begin{cases} \dfrac{d_K - d_k}{d_K - d_1}, & d_K \ne d_1 \\ 1, & d_K = d_1 \end{cases} \tag{3.5}$$

这些权重具有值域[0,1]，且越近的近邻具有越大的权重。计算样本之间距离的方法有很多种，下面的实验中我们采用一阶 Minkowski 距离（https://en.wikipedia.org/wiki/Minkowski_distance）。待测样本 x 对所有类标记的得分如下

$$s_j = \frac{\sum_{l_j \in y_k^*} w_k}{\sum_{1 \le k \le K} w_k},\ 1 \le j \le c \tag{3.6}$$

从上述定义可以看出，越多的近邻具有标记 l_j 则 x 对该标记的得分越大。如果所有的近邻都具有该标记，则得分为 1，如果没有近邻具有该标记，则对该标记的得分为 0.

得到上面的类标记得分后，一个很简单直接的方法是直接根据得分大小来确定待测样本是否具有某一个标记，但是这种策略相当于独立考虑每一个标记而忽略他们之间的

相关性。我们假定最终的标记分配取决于所有的标记得分是一种更好的策略，这样的话每个标记的最终输出不光与该标记的得分有关，还与其他标记的得分有关，这种标记的相关性可以通过下面的回归模型来定量。

假设 X_2 是样本矩阵，其中第 i 行对应训练样本 x_i 的增广标记得分向量 $\boldsymbol{x}_i = [1 \ s_{i1} \ s_{i2} \ \cdots \ s_{ic}]^{\mathrm{T}}$，其中的标记得分由权重 K 近邻算法和训练样本集（除 x_i 之外）确定。假设 Θ_2 是系数矩阵，我们有类似于特征驱动方法的最优化模型

$$\min J(\Theta_2) = \frac{1}{2}\left\|\mathrm{Y} - \mathrm{X}_2\Theta_2\right\|_F^2 + \frac{\lambda}{2}\left\|\Theta_2\right\|_F^2 \tag{3.7}$$

最有的参数矩阵通过求导获得

$$\hat{\Theta}_2 = (\mathrm{X}_2{}^{\mathrm{T}}\mathrm{X}_2 + \lambda\mathrm{I})^{-1}\mathrm{X}_2{}^{\mathrm{T}}\mathrm{Y} \tag{3.8}$$

对任一个待测样本 x，我们首先求出其增广标记得分向量，再利用下式来求出它对所有标记的输出

$$f_2(\boldsymbol{x}, l_j) = \boldsymbol{x}^{\mathrm{T}}\hat{\Theta}_2^j, 1 \le j \le c \tag{3.9}$$

其中 $\hat{\Theta}_2^j$ 是系数矩阵 $\hat{\Theta}_2$ 的第 j 列。

3.2.3 混合方法

综合上面的两种模型，我们定义下面的加权和输出作为待测样本 x 对各标记的隶属度

$$f(x, l_j) = \alpha f_1(x, l_j) + (1-\alpha) f_2(x, l_j) \tag{3.10}$$

其中 $1 \le j \le c$, $0 \le \alpha \le 1$。待测样本预测标记通过阈值法求出

$$h(x) = \{l_j \mid f(x, l_j) \ge 0, 1 \le j \le c\} \tag{3.11}$$

混合多标记分类器 hMuLab 的伪代码如下图

[*h*, *f*] = hMuLab (*D*, λ, *K*, α, *x*)
Inputs:
D: Training dataset (*X*, *Y*)
λ: regularization parameter λ in (3.2)
K: number of neighbors in (3.5)
α: Hybrid parameter in (3.10)
x: query sample
Outputs:
h: predicted label set
f: confidence of each class label
Procedure:
(a) Train:

1. Use (3.3) to calculate coefficient matrix Θ_1.
2. Create the score vector of training samples according to (3.5~3.6), then get coefficient matrix Θ_2 according to (3.8).

(b) Predict:

3. Calculate Feature Score according to (3.4).
4. Create the score vector of the query sample according to (3.5~3.6), and get Neighbor Score according to (3.9).
5. Obtain the final confidence of each class label according to (3.10), and the predicted label set according to (3.11).

图 3.4 hMuLab 伪代码

3.3 实验设计与结果分析

3.3.1 数据集

多标记学习存在于多个应用场景，例如图像、文本、音乐和生物医学等方面的数据挖掘。幸运的是，已经存在多个研究社区收集了许多有关这些方面的多标记数据集，例如 Mulan（http://mulan.sourceforge.net/datasets.html）[117] 和 Keel（http://sci2s.ugr.es/keel/multilabel.php）。本研究主要选取一些生物医学方面的数据集来测试本章所提出的方法的性能。在上述两个网站上可以找到 3 个生物医学方面的多标记数据集，它们分别是 Yeast[58]，Genbase[118]和 Medical[119]，这三个数据集的统计特性如下表所示，其中 LC 是标记势，表示平均每个样本所具有的标记数。

表 3.1 生物医学多标记数据集及其统计特性。

数据集	样本数	特征数	标记数	LC
Yeast	2417	103*n*	14	4.2371
Genbase	662	1185*b*	27	1.2523
Medical	978	1449*b*	45	1.2444

上面表中的三个数据集都来自生物医学领域。Yeast 包含 2417 个 Yeast 基因，每个基因用 103 个特征来表示，这些特征包含基因表达水平和 phylogenetic profiling values，每个基因都具有一种或者多种功能类型。Genbase 数据集对 662 个蛋白质进行了功能注释。每个蛋白质都用 1185 维的二值特征向量来描述，每个二值特征表示该蛋白是否具有某种 motif。每个蛋白质都属于 27 种功能类型中的一种或者多种。第三个数据集 Medical 来源于新西兰 waikato 大学计算医学中心于 2007 年举办的医学自然语言处理挑战赛。该数据集包含 978 条医学文本记录，每条记录都被表示成 1449 维的二值特征向量，每个二值特征用于表示该记录是否存在某个关键字。每条医学记录都与 45 种候选疾病代码中的一种或者多种相关联。

3.3.2 参数讨论

本章所提出的混合多标记学习方法只有三个超参数，分别是正则参数 λ，近邻数目 K，和模型权重参数 α，我们将分开讨论每个超参数对模型的影响。首先来看参数 λ，从我们的实验结果来看，相比于其他参数该参数对模型的影响非常小，且 λ=1 时模型总能取得比较好的性能，因此针对所有的数据集，我们都取默认值 λ=1。下面来看参数 K，令 a=0，这时候混合多标记模型将退化成近邻驱动方法。正如经典的 K 近邻方法，该方法中的近邻数目 K 是唯一的影响因素。以 Yeast 数据集为例，我们对参数 K 进行网格搜索，搜索范围为[1, 5, 10, …, 40]，针对每个参数分别在该数据集上进行 20 次的 10 折交叉验证，图 3.5 给出了每种参数下的平均指标值。从图中可以看出，一开始的时候，随着 K 值的增大，所有的 6 个评价指标都迅速提升，但是当 K 值增大到 15 的时候，指标 *Subset Accuracy* 开始变差，而另外 5 个指标则逐渐进入平台期。我们在 Medical 数据集上进行测试时也能观察到类似的现象，如图 3.6 所示。在 Genbase 数据集上，K 值影响则表现出一定的随机性（图 3.7），但是整体而言 K=5 时的表现最好。综合考虑多个多标记数据集和多个评价指标，我们取 K=15 作为模型中近邻驱动方法的默认参数。

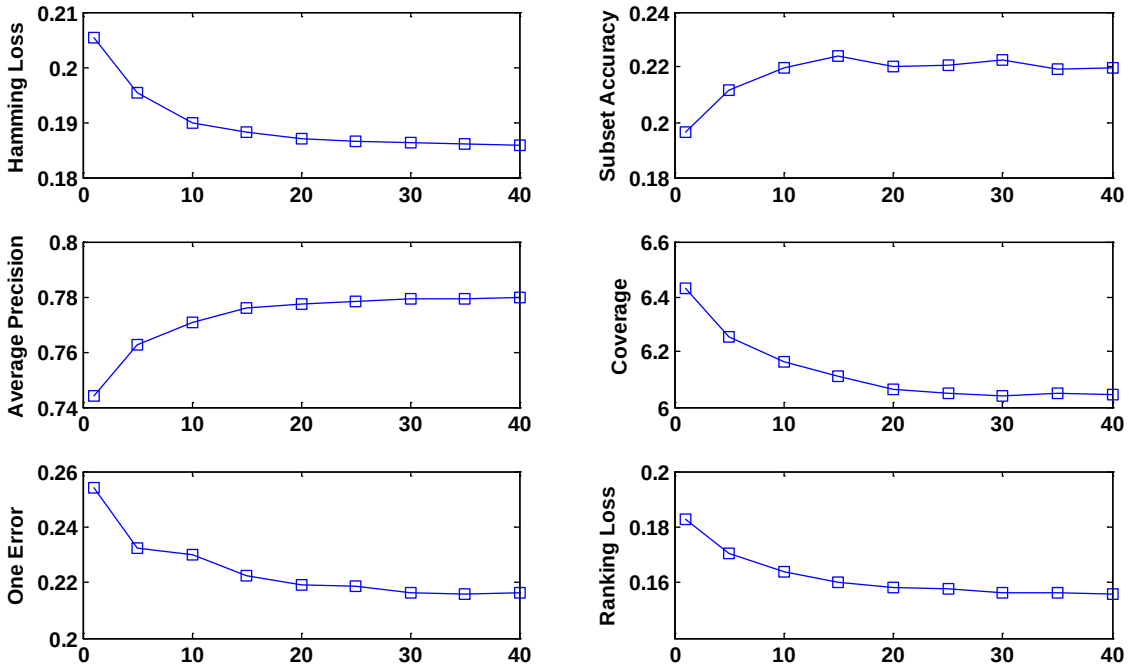


图 3.5 Yeast 数据集上 a=0 时超参数 K 对模型的影响

其中横坐标表示超参数K的取值，纵坐标表示评价指标值。

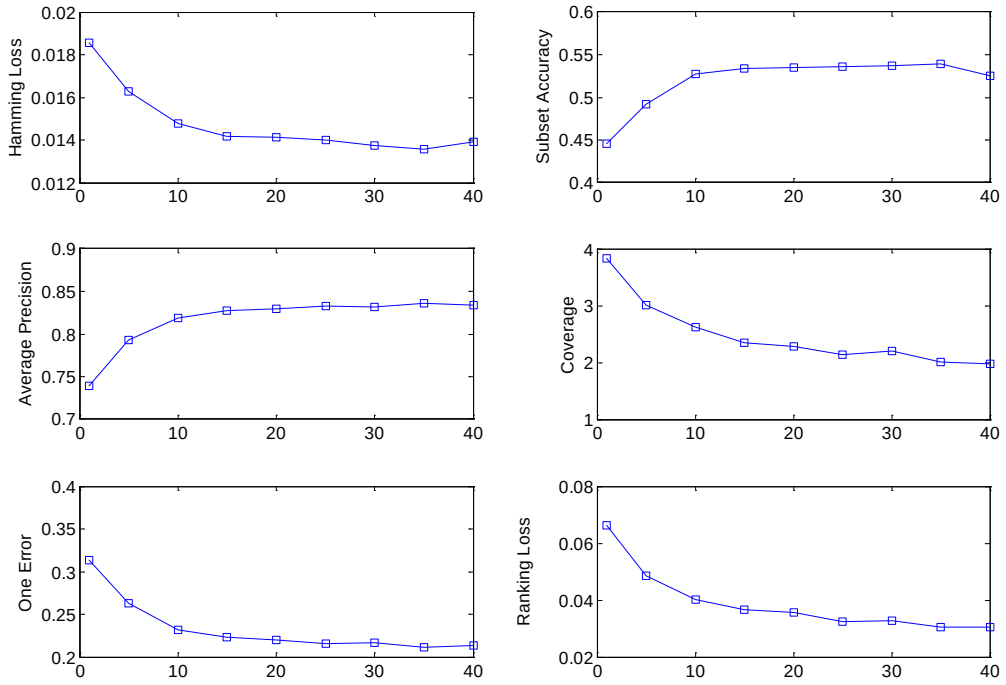


图 3.6 Medical 数据集上 a=0 时超参数 K 对模型的影响

其中横坐标表示超参数K的取值，纵坐标表示评价指标值。

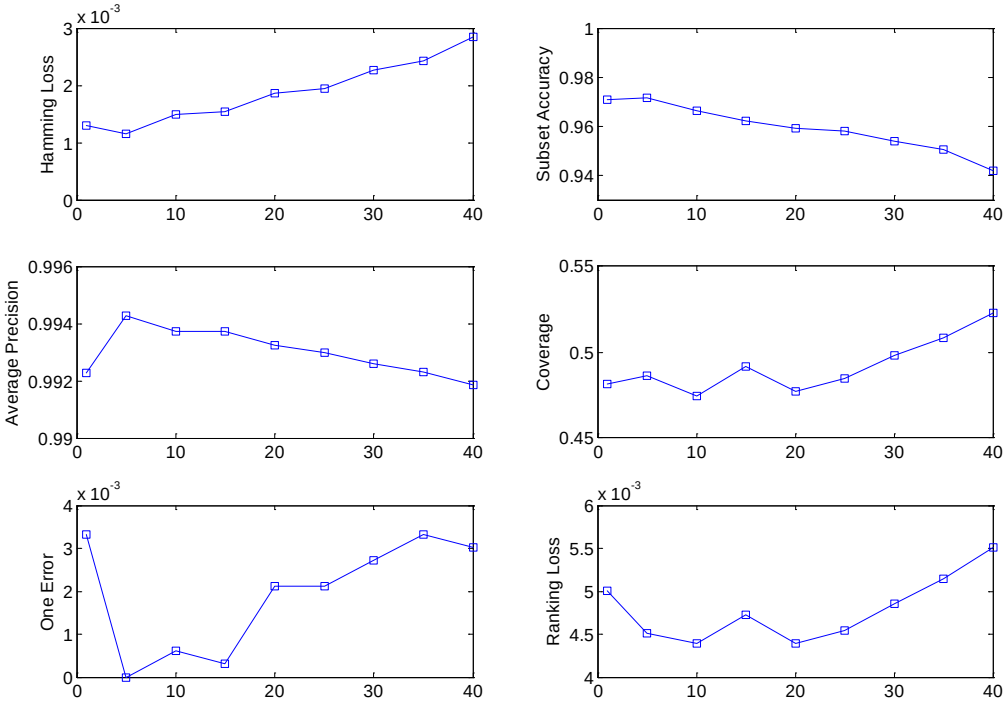


图 3.7 Genbase 数据集上 a=0 时超参数 K 对模型的影响

其中横坐标表示超参数K的取值，纵坐标表示评价指标值。

混合多标记学习方法 hMuLab 利用参数 *a* 来调节特征驱动方法和近邻驱动方法的权重。如果 *a*=1，则近邻驱动方法不起作用，混合模型退化成特征驱动方法。如果 *a*=0，则特征驱动方法不起作用，模型将只根据近邻驱动方法的结果来为待测样本分配预测标记。接下来我们将通过实验来观察参数 *a* 对整个模型性能的影响。分别令 *a* 取值 0, 0.25, 0.5, 0.75, 1，并分别在 3 个数据集上进行 20 次的 10 折交叉验证，测试结果如图 3.8~3.10 所示。从图中可以看出，几乎在所有评价指标上，当 *a* 取值为 0 到 1 之间的某个小数时的模型性能总是优于当 *a* 取值为 0 或者 1 时的性能，即混合模型要优于任何一个孤立模型。综合考虑所有数据集上的所有指标，我们取 *a*=0.5 作为混合模型的默认参数。

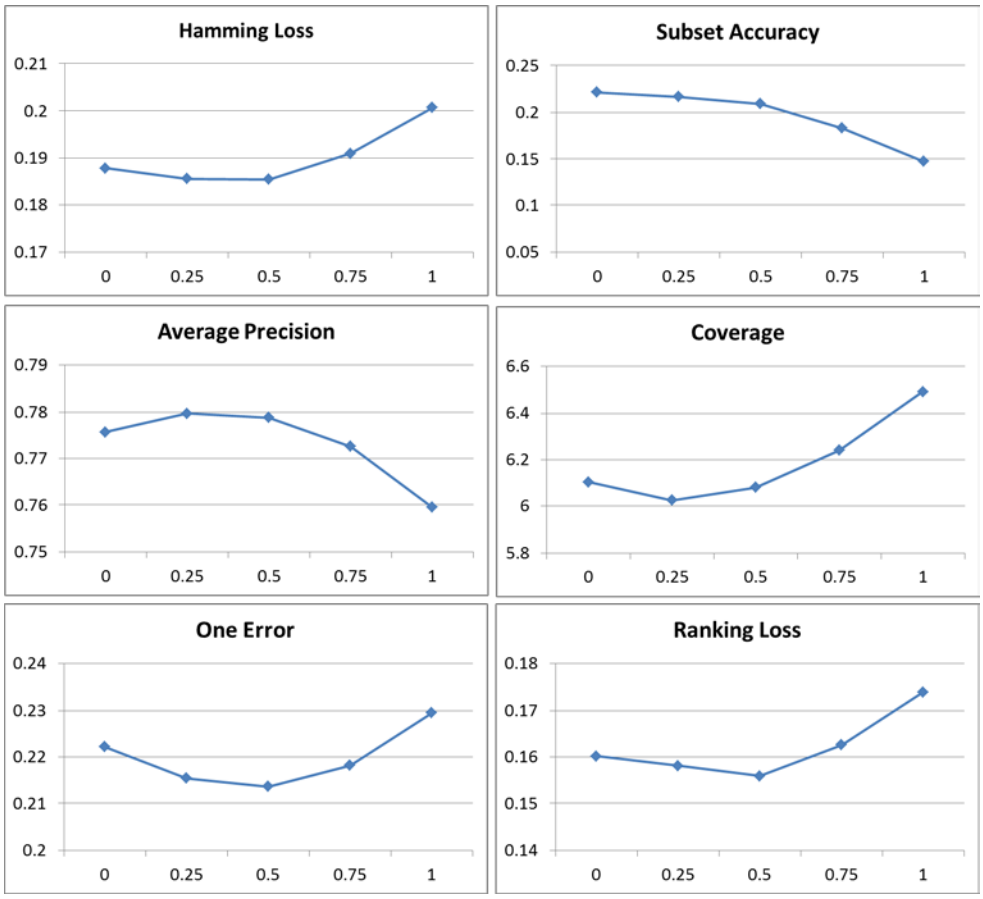


图 3.8 Yeast 数据集上超参数 *a* 对模型性能的影响

其中横坐标表示 *a* 的取值，纵坐标表示评价指标值。

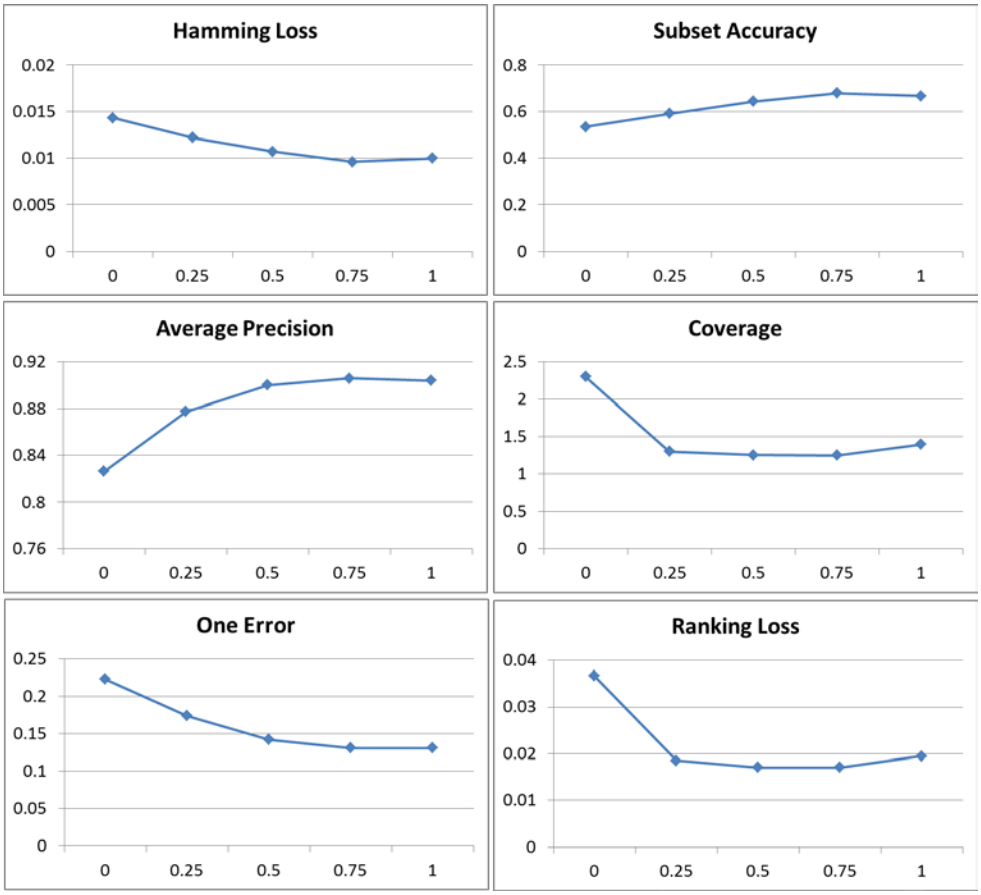


图 3.9 Medical 数据集上超参数 a 对模型性能的影响

其中横坐标表示 a 的取值，纵坐标表示评价指标值。

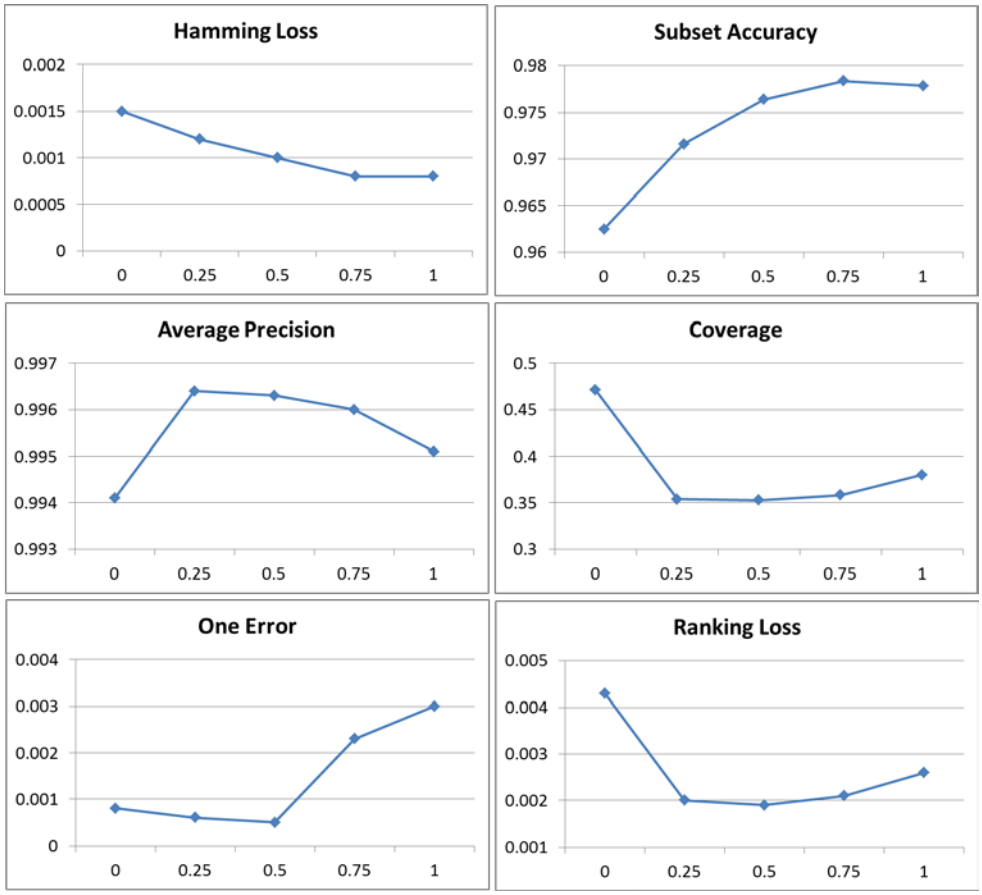


图 3.10 Genbase 数据集上超参数 *a* 对模型性能的影响

其中横坐标表示 *a* 的取值，纵坐标表示评价指标值。

关于超参数优化，需要指出的是，针对不同的数据集要找出最优的参数组合，需要将所有参数联合起来进行网格搜索，但这在训练过程中将是一个很大的计算开销，尤其是我们的测试方法是20次的10折交叉验证，几乎不可能在可接受的时间范围内找出最优的参数组合，因此在以下的实验结果比较中，无论是本章提出的方法，还是其他方法，都将采用默认的参数。

3.3.3 与其他方法的结果比较

本小结中我们将比较 hMuLab 与其他四种流行的多标记学习方法，分别是 MLKNN, RAkEL, ClassifierChain 和 IBLR[120]，所有这四种方法都已在多标记学习包 Mulan 中实现。其中 RAkEL 和 ClassifierChain 需要选择一个基分类器，在此我们用的是 J48（一种 C4.5 算法的具体实现）。所有方法的超参数都将设置为默认参数。分别在三个数据集上进行 20 次的 10 折交叉验证，并将平均指标值列于表 3.2~3.7 中。此外为了使得比较更直观，我们将根据平均指标值对所有方法进行排序，并将排序结果也列在表中。从这些表中我们可以看出，在所有的 3 个数据集 6 个评价指标下，只有在 Medical 数据集上，hMuLab 被 RAkEL 和 ClassifierChain 在指标 Hamming Loss 和 Subset Accuracy 上超越。对 Hamming Loss 而言，其值越小，说明取得该值的方法越好。在该指标上 RAkEL 和 ClassifierChain

分别只比 hMuLab 提升了 4.72%和 3.77%，但是 hMuLab 比排在下一位的 MLKNN 提升了 31.61%。对 Subset Accuracy 而言，其值越大，说明取得该值的方法越好。在该指标上 hMuLab 排名第三，RAkEL 和 ClassifierChain 分别超出 hMuLab 3.03% 和 4.72%，但是 hMuLab 超出排名第四的 MLKNN 23.61%。因此在这两个指标上，RAkEL、ClassifierChain 和 hMuLab 处于一个梯队，且比其他方法要高出很多。而在三个数据集上的其他指标下，hMuLab 都是排名第一。综合来看，本章所提出的混合多标记学习方法要明显优于其他四种方法。

表 3.2 生物医学数据集上 Hamming Loss 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.1854(1)	0.1928(2)	0.2280(4)	0.2688(5)	0.1933(3)
Genbase	0.0009(1)	0.0046(4)	0.0011(2)	0.0011(2)	0.0020(3)
Medical	0.0106(3)	0.0155(4)	0.0101(1)	0.0102(2)	0.0196(5)
平均排序	1.6667	3.3333	2.3333	3	3.6667

表 3.3 生物医学数据集上 Subset Accuracy 比较

值越大越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.2094(1)	0.1865(3)	0.1108(5)	0.1439(4)	0.2024(2)
Genbase	0.9782(1)	0.9121(4)	0.9717(2)	0.9717(2)	0.9579(3)
Medical	0.6439(3)	0.4919(4)	0.6634(2)	0.6743(1)	0.4694(5)
平均排序	1.6667	3.6667	3	2.3333	3.3333

表 3.4 生物医学数据集上 Average Precision 比较

值越大越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.7788(1)	0.7669(3)	0.7151(4)	0.6241(5)	0.7687(2)
Genbase	0.9964(1)	0.9880(5)	0.9905(3)	0.9926(2)	0.9903(4)
Medical	0.8996(1)	0.8081(4)	0.8359(3)	0.8375(2)	0.7561(5)
平均排序	1	4	3.3333	3	3.6667

表 3.5 生物医学数据集上 Coverage 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	6.0805(1)	6.2223(3)	7.5208(4)	8.9455(5)	6.2035(2)
Genbase	0.3562(1)	0.5610(5)	0.3671(2)	0.3687(3)	0.4142(4)
Medical	1.2356(1)	2.6658(2)	4.2104(4)	4.6132(5)	3.8626(3)
平均排序	1	3.3333	3.3333	4.3333	3

表 3.6 生物医学数据集上 One Error 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.2136(1)	0.2283(3)	0.2865(4)	0.3603(5)	0.2258(2)
Genbase	0.0030(1)	0.0107(5)	0.0091(4)	0.0034(2)	0.0072(3)
Medical	0.1426(1)	0.2508(4)	0.1764(3)	0.1758(2)	0.3151(5)
平均排序	1	4	3.6667	3	3.3333

表 3.7 生物医学数据集上 Ranking Loss 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Yeast	0.1559(1)	0.1648(3)	0.2167(4)	0.3298(5)	0.1642(2)
Genbase	0.0020(1)	0.0062(5)	0.0031(3)	0.0030(2)	0.0036(4)
Medical	0.0170(1)	0.0405(2)	0.0740(4)	0.0782(5)	0.0655(3)
平均排序	1	3.3333	3.6667	4	3

为了使得算法之间的比较更具有统计意义，我们采用成对 *t* 检验来比较本章提出的方法与其他方法之间的优劣，其中 *Pvalue* 取值 0.05。对某一个数据集，首先我们分别对每种算法进行 20 次的 10 折交叉验证，这样的话对应每种方法都将有 20×6（6 种评价指标）个实验结果。然后针对每一个评价指标，在 hMuLab 和另外一种方法之间进行成对 *t* 检验，检验结果有三种情况：1）hMuLab 显著优于另一种方法，用 better 表示，2）hMuLab 与另一种方法之间的区别不明显，用 tie 表示，3）hMuLab 显著劣于另一种方法，用 worse 表示。当在所有数据集上比较 hMuLab 与其他算法时，可以用比较三元表(better/tie/worse)来统计 better、tie 和 worse 发生的次数。表 3.8 给出了 hMuLab 与另外四种方法之间的比较三元表。从表中可以看出，本章提出的混合多标记学习方法在所有三个数据集上所有六个评价指标上都要显著优于 MLKNN 和 IBLR。相比于 RAkEL 和 ClassifierChain，worse 发生了 2 次，tie 发生了 1 次，但是 better 的次数多达 15。总体而言，hMuLab 还是要显

著优于其他比较的四种方法。

表 3.8 hMuLab 与其他四种方法之间的比较三元表（better/tie/worse）

Metric	hMuLab versus			
	MLKNN	RAkEL	ClassifierChain	IBLR
Hamming Loss	3/0/0	2/0/1	2/0/1	3/0/0
Subset Accuracy	3/0/0	2/0/1	2/0/1	3/0/0
Average Precision	3/0/0	3/0/0	3/0/0	3/0/0
Coverage	3/0/0	2/1/0	2/1/0	3/0/0
One Error	3/0/0	3/0/0	3/0/0	3/0/0
Ranking Loss	3/0/0	3/0/0	3/0/0	3/0/0
In total	18/0/0	15/1/2	15/1/2	18/0/0

在上述实验中，我们只在生物医学数据集上进行了测试，但是从 hMuLab 的工作原理来看，该混合方法是一种通用的多标记学习方法，它也可以在其他领域的数据集上取得好的结果，例如这里我们又选取了两个多媒体领域的数据集进行了 20 次的 10 折交叉验证，并将平均结果及其排序结果列于表 3.9~3.14 中。从表中的结果来看，hMuLab 几乎在所有的评价指标上都表现的最好。因此，hMuLab 可以作为一个通用且优异的多标记学习方法来使用。

表 3.9 非生物医学数据集上 Hamming Loss 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.0767(1)	0.0860(3)	0.1018(4)	0.1386(5)	0.0841(2)
Emotions	0.1742(1)	0.1954(3)	0.2161(4)	0.2583(5)	0.1864(2)

表 3.10 非生物医学数据集上 Subset Accuracy 比较

值越大越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.6391(2)	0.6299(3)	0.5632(4)	0.5534(5)	0.6495(1)
Emotions	0.3351(1)	0.2870(3)	0.2619(4)	0.2121(5)	0.3180(2)

表 3.11 非生物医学数据集上 Average Precision 比较

值越大越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.8895(1)	0.8657(3)	0.8357(4)	0.7270(5)	0.8669(2)
Emotions	0.8264(1)	0.7994(3)	0.7783(4)	0.6891(5)	0.8163(2)

表 3.12 非生物医学数据集上 Coverage 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.3929(1)	0.4723(3)	0.6022(4)	1.2884(5)	0.4649(2)
Emotions	1.6654(1)	1.7802(3)	1.9552(4)	2.6147(5)	1.6931(2)

表 3.13 非生物医学数据集上 One Error 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.1885(1)	0.2251(3)	0.2682(4)	0.3807(5)	0.2241(2)
Emotions	0.2320(1)	0.2754(3)	0.2995(4)	0.4076(5)	0.2518(2)

表 3.14 非生物医学数据集上 Ranking Loss 比较

值越小越好，括号中数字表示在某个数据集上各种算法的平均结果排序

Dataset	hMuLab	MLKNN	RAkEL	ClassifierChain	IBLR
Scene	0.0613(1)	0.0770(3)	0.1026(4)	0.2369(5)	0.0760(2)
Emotions	0.1382(1)	0.1605(3)	0.1908(4)	0.3101(5)	0.1474(2)

3.4 本章小结

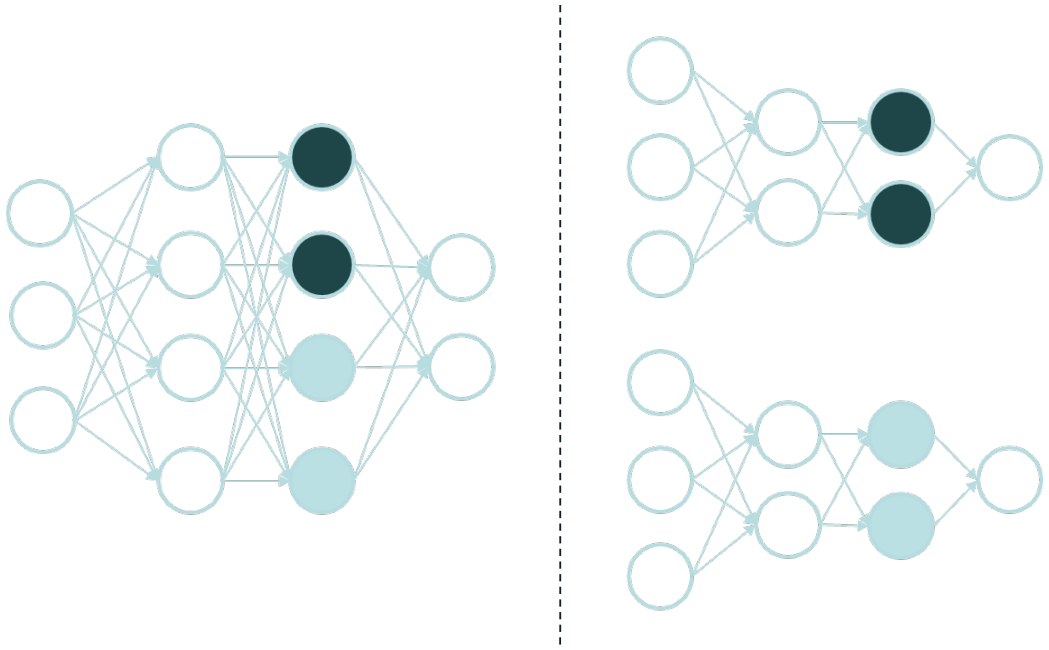
本文提出了一种新型的混合多标记分类算法 hMuLab，从类型上来说它是一种并行异质集成学习方法，融合了两种工作原理截然不同的多标记学习方法，一种是特征驱动的方法，另一种是近邻驱动的方法。这两种方法“好而不同”，相辅相成，共同作用形成了一个更强的集成多标记分类器。通过在多个生物医学方数据集上的测试发现，所提出的混合多标记分类器的预测性能要好于任何一个独立的个体学习器，且与多种已有的多

标记学习算法进行比较时也表现出了绝对的优势。需要说明的是，实验所采用的生物医学数据集在多标记数据集中是非常具有代表性的，如 Yeast 数据集中的样本具有浮点类型的特征值，标记势较大，而 Genbase 和 Medical 数据集中的样本具有二值类型的特征值，标记势较小。此外，后两个数据集的特征具有稀疏和高维的特性。hMuLab 能在这些数据集上取得非常好的预测结果，我们相信，它同样能够胜任其他领域的多标记学习任务。

第四章 一种深度多标记学习算法

4.1 引言

神经网络（深度网络）已成为当前最流行的机器学习工具，它在层次化特征描述和复杂函数映射方面具有突出的优势。利用神经网络架构来解决多标记学习问题具有天然的优势，因为神经网络本身就是一个多输入多输出的系统。虽然我们也可以为每一个标记都建立一个神经网络模型（BR 方法），但是这将在模型训练的时候面临诸多挑战，比如在 ImageNet 大规模场景识别中，有多达 1000 个类别，一千多万的样本，对于这么复杂的一个学习问题，训练一个网络模型就已经需要耗费很大的计算资源和很长的训练时间，如果要训练出 1000 个模型出来几乎是不可能的。而且这种做法也不符合人类学习的自然规律，因为在面对任何一个对象（如一副图片）时，我们的神经系统在接收到它的信息后，经过一定的处理，会很自然地为它同时打上多种标记（图 4.1(a)），而无需将神经系统划分为一个一个子系统，让每个子系统仅负责一个标记的学习任务（图 4.1(b)）。



(a) (b)

图 4.1 BR 神经网络与多标记神经网络在特征学习上的区别

（a）同时为多个标记建立一个神经网络，（b）为每个标记建立一个神经网络

从特征学习的角度来说，同时为多个标记建立一个神经网络是完全可行的，只要经过逐层的学习，最终的特征向量中包含所有标记所需要的信息，那么最终的决策也会变得很容易。隐含层与输出层之间的权重将决定一个标记所需要的特征的重要性。

从标记关联性来说，同时为多个标记建立一个神经网络有利于将标记之间的关联性

融合进来，因为标记之间的相关性说到底是他们相关的特征之间的关联性。在识别某个标记时，除了它自身所需的特异性特征之外，与它关联的标记所相关的特征对它的决策也是有帮助的。例如对于下面一副图片，它具有很多标记，如草地，蒙古包，河流，山，白云等等。我们知道，在一定场景中草地与蒙古包之间有很强的关联性，如果要识别图片是否包含草地，这个时候除了草地自身的特性之外，蒙古包的特征也将是很有用的。

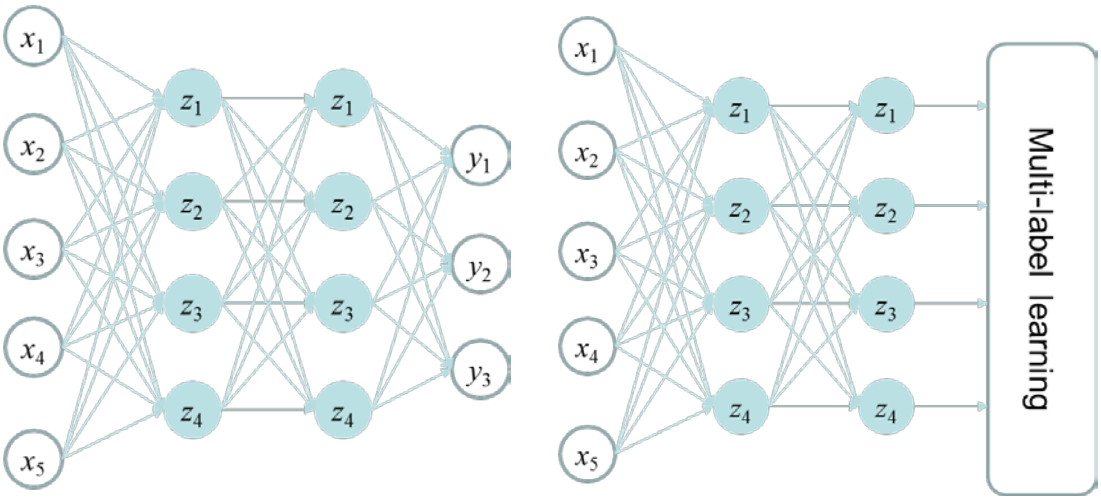
图 4.2 一副具有多标记的图片

将深度网络与多标记学习结合起来还有一个巨大的优势，那就是很容易地将无监督学习集成到模型当中。我们知道，在当前的大数据时代，人们可以获取的数据量往往非常大，其中大部分是无标记的，但是这样的数据并非都是无用的。目前的研究已表明，无监督学习方法（如受限玻尔兹曼机[121; 122]、稀疏自编码[11; 123]等）可以从这些无标记数据中挖掘出很多有用的模式信息，再结合有监督学习，往往能显著提升学习的效果。

多标记神经网络有这么多的好处，其实它的实现是很容易的。最直接的方式就是修改输出层模型使它适应多标记输出的场景，同时建立多标记损失函数，使得网络在优化过程中能自动学习到所有标记所需的特征信息。

4.2 深度多标记学习

本节将提出并验证两种深度多标记学习方案，第一种在已有的深度神经网络架构的基础之上，通过设计多标记损失函数的方式修改输出层，使得网络能够满足多标记输出的要求，其网络架构如图 4.3(a)所示；第二种是将深度网络当作特征学习器，并将最后一个隐含层输出的高度抽象特征作为其他多标记分类器的输入，其网络架构如图 4.3(b)所示。

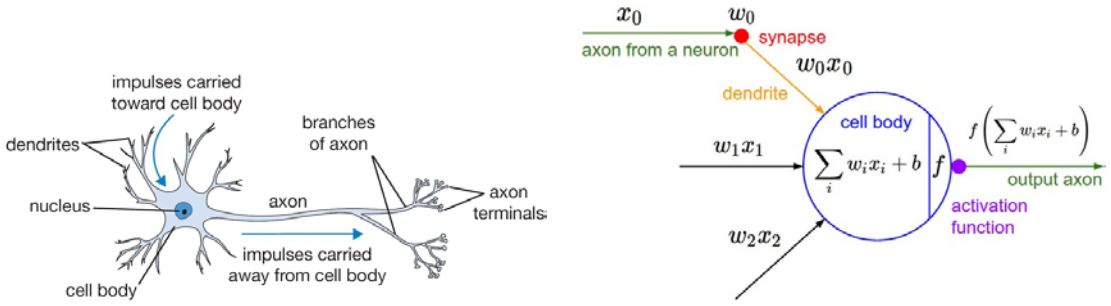


(a) (b)

图 4.3 多标记深度网络架构

4.2.1 人工神经元模型

我们知道，人脑的基本计算单元是神经元。通常一个人的神经系统包含大约 860 亿个神经元，这些神经元通过大约 10^{14}~10^{15} 个突触（synapse）相连接。下图是一个生物神经元以及常用的数学模型示意图。每个神经元通过它的树突（dendrite）接收输入信号，并通过轴突（axon）产生输出信号。轴突发出分支并与其他神经元的突触相连接。在神经元的计算模型中，如果相连接的后一个神经元的突触强度为 w_0，则前一个神经元通过轴突传导的信号 x_0 会与后一个神经元的树突发生乘法交互，并产生信号 w_0x_0。在该计算模型中，突触强度 w 是可学习的，并控制着一个神经元对另一个神经元的影响强度（激发或抑制）。在生物神经元中，细胞体将所有树突携带的信号相加，如果最终的和值超过一定的阈值，神经元就会激活并沿着轴突发出尖峰信号。在计算模型中，我们用激活函数 f 来建模这样一个生物过程。具体来说，每个神经元将它的突触强度与输入信号之间执行点积运算并加上一个偏置值，再输入激活函数进行一个变换后产生输出信号。



（a） （b）

图 4.4 生物神经元与人工神经元模型

（a）生物神经元，（b）人工神经元计算模型

需要说明的是，上述人工神经元只是生物神经元的一个非常粗略的简化模型。实际的神经元信号处理机制是非常复杂的。

激活函数用来接收单个数字并对其进行某种固定的数学变换后产生一个激活值。常见的激活函数有：

Identity 数学形式为 $f(x) = x$，导数为 $f'(x) = 1$。该线性激活函数总是返回与输入相等的值。由于是线性函数，不满足万能逼近定理（universal approximation theorem），对数据的拟合能力差，通常不用于神经网络的隐含层，而只用于输入层和输出层。

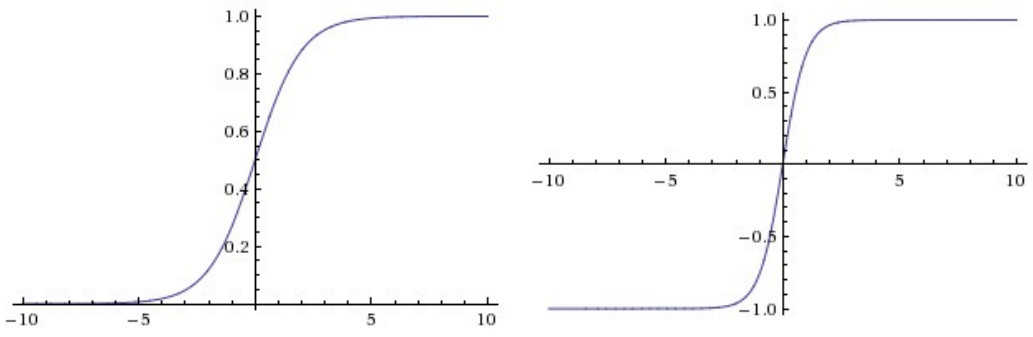
Sigmoid 如图 4.5(a)所示，其数学形式为 $f(x) = \sigma(x) = 1/(1+e^{-x})$，导数为 $f'(x) = f(x)*(1-f(x))$。该非线性函数能接受一个实数并将其压缩到 0 到 1 之间。当输入是非常大的正数时，输出趋于 1，而当输入是非常小的负数时，输出趋于 0。Sigmoid 函数具有良好的数学运算性（无限可导）和可解释性（神经元从完全抑制逐渐过渡到饱和激活），并在以前的神经网络模型中运用非常广泛。但是近几年人们逐渐对它失去兴趣并很少再使用它，主要是因为它容易饱和并杀死梯度。

Tanh 如图 4.5(b)所示，其数学形式为 $f(x) = \tanh(x) = 2/(1+e^{-2x}) - 1$，导数为 $f'(x) = 1 - f(x)^2$。该非线性函数将输入的实数值压缩到-1 到 1 之间。像 Sigmoid 函数一样，该函数在输入比较大或者比较小时会出现饱和现象。

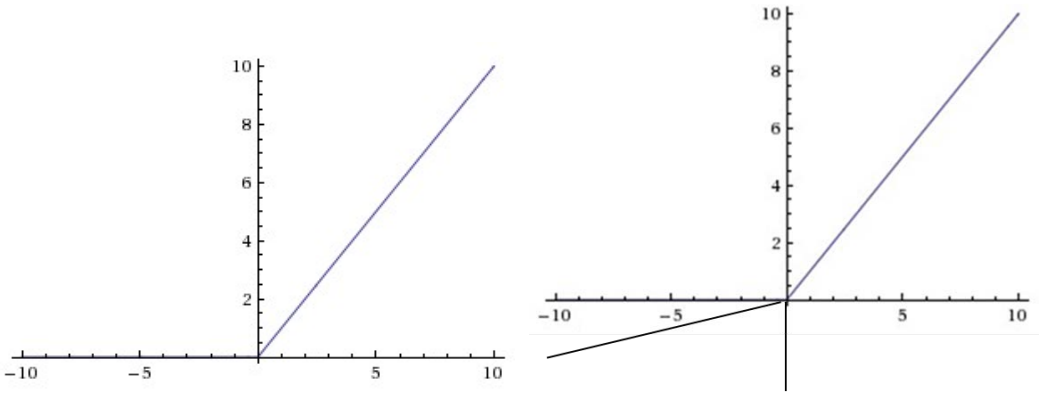
ReLU 如图 4.5(c)所示，其数学形式为 $f(x) = \text{relu}(x) = \max(0, x)$，导数为 $f'(x) = 1(x>=0)$。换句话说，该激活函数仅仅将输入在 0 处进行阈值化处理。近年来，ReLU（Rectified Linear Unit）正变得越来越流行[7; 124]，它具有下面几个显著特点：

- 相对于 Sigmoid 和 Tanh，它能极大地加速随机梯度下降算法的收敛速度
- 相对于 Sigmoid 和 Tanh 要使用的指数运算，它仅仅进行阈值化计算，速度非常快
- 但是，ReLU 单元非常脆弱，在训练过程中容易死掉。在梯度下降算法中设置一个合适的学习率会减轻这个问题[7]。

Leaky ReLU 如图 4.5(d)所示，其数学形式为 $f(x) = \text{lrelu}(x) = 1(x<0)(\alpha x) + 1(x>=0)(x)$，导数为 $f'(x) = 1(x<0)\alpha + 1(x>=0)$，其中 α 是一个小的非零常数。该激活函数允许神经元在未激活时能有一个小的梯度，试图解决 ReLU 单元容易死掉的问题[125]。有些研究者声称在实验中发现它要比 ReLU 性能优越，但是这样的结果并不总是与实际情况一致。



(a) (b)



(c) (d)

图 4.5 几种激活函数

4.2.2 人工神经网络模型

人工神经网络是是很多人工神经元连接而成的非循环图，其中一些神经元的输出可以作为其他神经元的输入。神经网络模型通常以多层神经元的形式组织。在常规神经网络中，最常见的层连接类型为全连接层，其中邻接的两层中的神经元两两互联，而同层中神经元没有连接。

下图即为常见的神经网络拓扑结构，整个网络由结点和连接边构成，带箭头的连接边指示了信号传递的方向。每个能接收输入信号的结点表示一个神经元结点，而不能接收输入信号的结点（标有“+1”的结点）为偏置结点，它总是输出固定值+1。每条连接边上都有一个参数，其中神经元结点与神经元结点之间的连接边上的参数称为权重（weight），偏置结点与神经元结点之间的连接边上的参数称为偏置（bias），神经网络所能学到的东西就蕴含在这些权重和偏置中。网络中最左边的一层称为输入层，用来将接收外来信号并输入到网络当中；最右边的一层称为输出层，用来输出我们想要的目标值；而输入与输出之间的层称为隐含层，用来对输入信号进行逐层的加工处理。通常只有一个隐含层的神经网络被称为浅层网络，而包含两个以上隐含层的网络被称为深度网络。

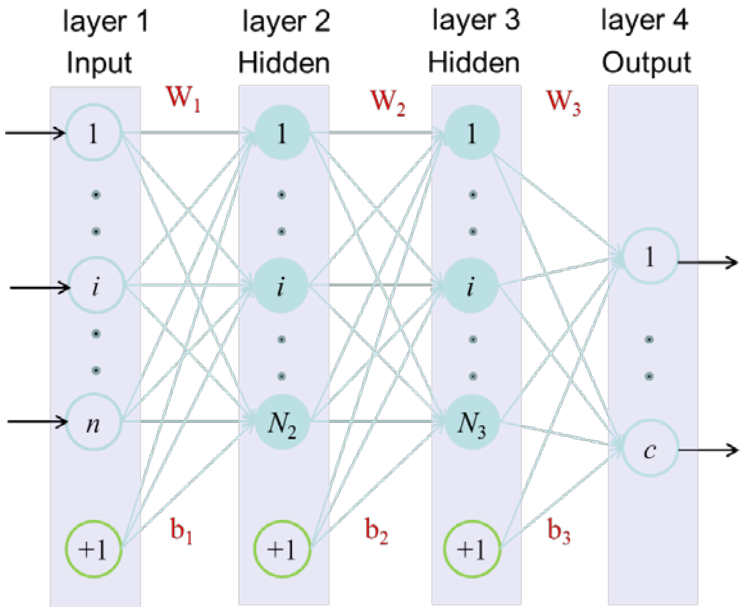


图 4.6 神经网络拓扑结构

这样的网络结构可以用来解决很多机器学习问题。输出层采用回归算法时，就是神经网络回归，包括单响应回归（一个输出结点时）和多响应回归（多个输出结点时）。输出层采用分类算法时，就是神经网络分类，包括二分类（一个输出结点）、多分类（多个输出结点）和多标记分类（多个输出结点）。

4.2.3 多标记损失函数

假设已有 m 个训练样本构成的训练集$\{(x^{(1)},y^{(1)}), \ldots, (x^{(m)},y^{(m)})\}$，我们要用神经网络模型来拟合该数据集，即要学到合适的网络参数（包括所有的权重 W 和偏置 b）使得网络的实际输出与期望输出之间的误差尽可能小。为达到这个目的，通常我们会首先定义一个损失函数（cost function），再使用某种学习方法来获得使损失函数最小化的最优参数。

损失函数的形式往往与具体的学习任务密切相关。如对于回归问题，可采用误差平方和作为损失函数；对于二分类问题，通常采用 logistic 回归损失函数（又称交叉熵损失——Cross Entropy Loss）或 SVM 损失函数（又称铰链损失——Hinge Loss）；对于多分类问题，通常采用 Softmax 回归损失函数（又称 log Loss）或多类 SVM 损失函数；对于多标记分类问题，可以采用排序损失函数（Ranking Loss）[58; 59; 126]。

对于多标记学习而言，最直观的优化目标是最小化相关标记与不相关标记之间的错误排序对的数目，称为排序损失：

$$J(\theta;x,y)=w(y)\sum_{y_i<y_j}\mathbf{I}\left(h_i(x)>h_j(x)\right)+\frac{1}{2}\mathbf{I}\left(h_i(x)=h_j(x)\right) \tag{4.1}$$

其中 $w(y)$是一个正则因子，$\mathbf{I}(\bullet)$是指示函数，$h_i(\bullet)$是对标记 y_i 的预测得分。该损失函数可理解为，对于任意一对相关标记和不相关标记，如果对相关标记的预测得分小于对不相关标记的预测值，则记损失分数为 1，如果两者相等，则记为 1/2。不幸的是，由于

上式是一个非连续可导的损失函数，很难直接优化，因而研究者们提出了各种替代损失函数（surrogate loss）来近似上面的排序损失函数[127; 128]。

多标记神经网络算法 BPMLL 中引入了一个称为“成对误差函数”（pairwise error function）的替代损失函数[59]，定义如下：

$$J_{PWE}(\theta; x, y) = \frac{1}{|y||\bar{y}|} \sum_{(y_p, y_n) \in y \times \bar{y}} e^{-(o_p - o_n)} \tag{4.2}$$

这里的 y 表示相关标记（正标记）集合，$\bar{y}$ 表示不相关标记（负标记）集合其中 y_p 和 y_n 分别表示相关标记和不相关标记，o_p 和 o_n 分别表示神经网络对 y_p 和 y_n 的输出值，|•| 表示集合的势。因此该公示可理解为，如果相关标记的输出值越大于不相关标记的输出值，则损失越小，相反，如果不相关标记的输出值越大于相关标记的输出值，则损失越大。

但是，一些实验表明，该损失函数不但计算量大，且效果不好[63]。因此在本文提出的多标记神经网络中，将不再使用这种类型的损失函数，而是直接把训练样本的实际标记向量作为网络的期望输出向量，把实际标记与期望标记之间的误差作为损失函数。接下来我们将提出两种方案的多标记损失函数，并在随后的实验中用多个数据集进行测试验证。

第一种方案中，我们在神经网络的输出层采用 Identity 神经元，此时网络的实际输出为 $p = z_L$，网络的期望输出向量就是实际标记向量 y。我们称这样的损失函数为多标记回归（Multi-label Regression，MLR）损失函数。

$$J(\theta; \boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{y}\|_2^2 = \frac{1}{2}\|\boldsymbol{z}_L - \boldsymbol{y}\|_2^2 \tag{4.3}$$

该输出层模型实际上就是超限学习机（Extreme learning machine）[129; 130]所采用的模型。

第二种方案中，我们在神经网络的输出层采用 Sigmoid 激活函数，假设此时网络的真实输出为 p，则以实际标记向量作为网络的期望输出向量时的损失函数就是多标记版本的交叉熵，称为多标记交叉熵（Multi-Label Cross Entropy，MLCE）

$$J(\theta; x, y) = -\sum_{j=1}^{c}\left[y_j \log p_j + (1 - y_j)\log(1 - p_j) \right] \tag{4.4}$$

其中 $y_j \in \{0,1\}$ 表示第 j 个标记的真实值（0 或 1），$p_j = p(l_j|x;\theta)$ 表示将样本 x 预测为属于标记 l_j 的概率。

与标准的神经网络相比，BPMLL 的损失函数考虑了标记之间的排序关系，本应在多标记学习中有更好的表现，但是从我们的实验情况以及其他文献中的结果[63]来看，BPMLL 的表现却并不太好。最近，也有些研究者发现，目前所提出的各种凸损失函数包括 BPMLL 所使用的并不与排序损失函数相一致，而实际上如对数损失（log loss）这样的替代损失函数与排序损失是相当一致的[131]。

$$J_{\log}(\theta;x,y)=w(y)\sum_{j}\log\left(1+e^{-y_j z_j}\right) \tag{4.5}$$

其中 y_j 是第 j 个标记的真实值，注意这里的真实值 $y\in\{-1,1\}$；z_j 是第 j 个标记的线性预测值；$w(y)$是一个关于标记 y 的权重函数，如在 BPMLL 中取 $w(y)=\frac{1}{|y||\bar{y}|}$；如果取 $w(y)=1$ 的话，对数损失函数变成

$$J_{\log}(\theta;x,y)=\sum_{j}\log\left(1+e^{-y_j z_j}\right) \tag{4.6}$$

对于第 j 个标记存在如下关系

$$\begin{aligned}&\log\left(1+e^{-y_j z_j}\right)\\&=-\log\frac{1}{1+e^{-y_j z_j}}\\&=\begin{cases}-\log p_j, if\ y_j=1\\-\log(1-p_j), if\ y_j=-1\end{cases}\end{aligned} \tag{4.7}$$

显然，它与采用 0-1 标记法的交叉熵完全一样，即这时的对数损失与我们所使用的多标记交叉熵损失函数是等价的。

4.2.4 梯度下降与误差反向传播算法

对于一个多层神经网络，在给定训练集的情况下最小化损失函数，实际上就是一个非凸的无约束最优化问题，可以运用一些经典的迭代优化方法来优化网络参数，如梯度下降算法（gradient descent）、牛顿法（Newton method）和拟牛顿法（quasi Newton method）[132]。

梯度下降算法（又称最速下降法，steepest descent）属于一阶导数优化方法，它的原理简单，实现起来也很方便，是现代神经网络最常用的优化方法。利用梯度下降算法来优化神经网络（网络参数用 θ 表示）的一般过程如下图所示

输入：训练数据，损失函数 $J(\theta)$，学习率 lr，学习精度 ε

输出：$J(\theta)$的极小点 θ*

（1） 取 θ 的初始值 $\theta^{(0)}$，置迭代次数 s=0

（2） 计算损失函数值 $J(\theta^{(s)})$

（3） 计算当前参数的梯度 $g^{(s)}=\left.\frac{\partial J(\theta)}{\partial\theta}\right|_{\theta^{(s)}}$，如果 $\left\|g^{(s)}\right\|<\varepsilon$，停止迭代，令 $\theta^*=\theta^{(s)}$，否则

（4） 更新参数，置 $\theta^{(s+1)}=\theta^{(s)}-lr*g^{(s)}$，计算 $J(\theta^{(s+1)})$，如果 $\left\|J(\theta^{(s+1)})-J(\theta^{(s)})\right\|<\varepsilon$，停止迭代，令 $\theta^*=\theta^{(s+1)}$，否则

（5） 置 s = s + 1，转（3）

图 4.7 梯度下降算法的一般流程

在上述梯度下降算法当中，根据输入的训练数据的规模，可以将其分为三种类型：

- 随机梯度下降（stochastic gradient descent）
- 小批量梯度下降（mini-batch gradient descent）
- 批量梯度下降（batch gradient descent）

在随机梯度下降中，每次迭代时仅输入一个样本并计算相应的损失函数值和梯度值，并立即更新参数值，因此也被称为在线梯度下降法（on-line gradient descent）[133; 134]。这种方法的迭代次数多，训练时间长，因此在实际应用中比较少见。与随机梯度下降法相对应的是批量梯度下降法，即在每次迭代时将整个训练集同时输入，并计算出所有样本的损失函数值和平均梯度（相当于分别求每个样本的损失函数值和梯度，再取均值）后，再来更新参数值。但是在大规模应用中，如目前 ImageNet 大规模视觉竞赛中使用的图像数据集包含 1400 万张图片，每次更新参数都要同时计算所有训练样本的损失函数将会非常的浪费计算资源和时间。这个时候更常用的训练方法是小批量梯度下降法，即每次迭代时仅随机选择一小部分数据，如 256 个样本，并根据这些样本来计算梯度和更新参数。直观上看，这种训练策略也是合理的，因为通常来说，训练集中的样本并不相互独立，而是具有一定的关联性，从小批量数据所得到的梯度往往就是全局目标下的梯度的一个很好的近似。实践证明，在面临大规模学习任务时，小批量梯度下降法能更快地收敛。

牛顿法属于二阶导数优化方法，由于在每次迭代时它都会沿着更好的方向（通过 Hesse 矩阵的逆矩阵来修正梯度）进行搜索，因此往往只需要很少的迭代次数就能达到收敛。但是牛顿法需要计算所有参数的 Hesse 矩阵的逆，这在参数比较多的时候不适合，尤其像有些深度网络参数在百万甚至千万级别，在现有的计算条件下要计算所有参数的 Hesse 矩阵的逆根本不现实。拟牛顿法是对牛顿法的一系列近似方法的统称，这些方法不需要直接计算参数的 Hesse 矩阵的逆，而是用间接的方法来逼近 Hesse 矩阵的逆。拟牛顿法需要保存之前的梯度信息，且只能用批训练模式，因此在使用深度网络解决大规模学习任务时也已经很少使用了。

从这些经典的优化方法可以看出，要有效地更新网络参数，一个非常关键的信息就是目标函数相对于参数的梯度，这个时候就需要用到非常经典的误差反向传播算法了。误差反向传播算法是利用梯度的链式法则来从后向前地逐层计算出每一层的参数梯度的。为了描述方便，我们先作出如下矩阵-向量形式的符号定义

符号	含义
$\{(x^{(1)},y^{(1)}), \ldots, (x^{(m)},y^{(m)})\}$	包含 m 个训练样本的训练集
l=1,2,…,L	网络层号
N_l	第 l 层的结点数
W_l	第 l 层与第 l+1 层的连接权重矩阵
b_l	第 l+1 层的偏置向量

z_l	第 l 层的输入向量
a_l	第 l 层的输出向量（激活）
$f(z)$	激活函数
δ_l	第 l 层残差向量
$J(W,b)$	整体损失函数
$J(W,b;x,y)$	单样本损失函数

图 4.8 神经网络相关变量的符号化定义

首先来看只输入一个训练样本（$\boldsymbol{x},\boldsymbol{y}$）时的网络训练过程，整个训练过程可以用下面三个步骤来描述：

（1）根据神经元计算模型使用前向传播方法计算样本在各层上的激活值。

输入层的激活值：

$$a_1 = z_1 = \boldsymbol{x} \tag{4.8}$$

隐含层的激活值：

$$a_l = f(z_l) = f(W_{l-1} * a_{l-1} + b_{l-1}),\ \text{for } l = 2,\ldots,L\text{-}1 \tag{4.9}$$

输出层的激活值 a_L 取决于输出层模型。如果输出层采用 Identity 激活函数

$$p = a_L = z_L = W_{L-1}a_{L-1} + b_{L-1} \tag{4.10}$$

如果输出层采用 Sigmoid 激活函数

$$\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_c \end{bmatrix} = \begin{bmatrix} p(y=1 \mid x;\theta) \\ p(y=2 \mid x;\theta) \\ \vdots \\ p(y=c \mid x;\theta) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+e^{-z_{L,1}}} \\ \dfrac{1}{1+e^{-z_{L,2}}} \\ \vdots \\ \dfrac{1}{1+e^{-z_{L,c}}} \end{bmatrix} \tag{4.11}$$

其中 $z_L = W_{L-1}a_{L-1} + b_{L-1}$。

（2）反向传播误差 δ，这里的误差定义为损失函数对结点的输入的偏导数。

输出层误差 δ_L 的计算取决于输出层模型。如果输出层采用 Identity 激活函数，此时的多标记代价函数为 $J(\theta;\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{p}-\boldsymbol{y}\|_2^2 = \frac{1}{2}\|\boldsymbol{z}_L-\boldsymbol{y}\|_2^2$，则有

$$\delta_L = \nabla z_L J = p - y \tag{4.12}$$

如果输出层采用 Sigmoid 激活函数，此时的多标记代价函数为

$J(\theta;x,y) = -\sum_{j=1}^{c}\left[y_j \log p_j + (1-y_j)\log(1-p_j)\right]$，则有

$$\delta_L = p - y \tag{4.13}$$

证明如下：

$$\begin{aligned}\delta_{L,i} &= \frac{\partial J}{\partial z_{L,i}} \\ &= \frac{\partial J}{\partial p_i}\frac{\partial p_i}{\partial z_{L,i}} \\ &= -\left(y_i\frac{1}{p_i}-(1-y_i)\frac{1}{1-p_i}\right)p_i(1-p_i) \\ &= p_i - y_i\end{aligned} \tag{4.14}$$

第三步的获得是因为

$$\begin{aligned}\frac{\partial p_i}{\partial z_i} &= \frac{\partial}{\partial z_i}\frac{1}{1+e^{-z_i}} \\ &= \frac{e^{-z_i}}{\left(1+e^{-z_i}\right)^2} \\ &= p_i(1-p_i)\end{aligned} \tag{4.15}$$

隐含层的误差为

$$\delta_l = \left(\mathrm{W}_l^{\mathrm{T}}\delta_{l+1}\right)\bullet f'(\mathrm{z}_l),\ \text{for } l = \text{L-1},\ldots,2. \tag{4.16}$$

这里的•表示 Hadamard 乘积，用于将两个相同大小的矩阵的相同位置的元素分别相乘而得到一个新的矩阵；$f'(z)$表示激活函数的导数。

（3）根据每层的激活值和误差值来计算梯度值

$$\nabla_{\mathrm{W}_l} J(\theta; x, y) = \delta_{l+1}{a_l}^{\mathrm{T}},\ \text{for } l = 1,\ldots,\text{L-1} \tag{4.17}$$

$$\nabla_{\mathrm{b}_l} J(\theta; x, y) = \delta_{l+1},\ \text{for } l = 1,\ldots,\text{L-1} \tag{4.18}$$

如果是批训练模式，由于 $J(W,b) = \frac{1}{m}\sum_{i=1}^{m} J(W,b;x^{(i)},y^{(i)})$，它实际上就是多个样本损失函数值的平均结果。相应地，此时的总梯度也就是所有样本梯度的平均值。

通常为了提高模型的泛化能力，还要在原有的损失函数上加上一个参数衰减项，如

$$J(W,b) = \frac{1}{m}\sum_{i=1}^{m} J(W,b;x^{(i)},y^{(i)}) + \frac{\lambda}{2}\sum_{l=1}^{L-1}\left\|\mathrm{W}_l\right\|_F^2 \tag{4.19}$$

其中公式右边的前半部分表示的是数据损失项，后半部分表示的正则损失项，λ 是两者之间的权重化因子。相应地，在求各层的参数梯度时，也需要加上参数衰减项，

$$\frac{\partial}{\partial \mathrm{W}_l}\left(\frac{\lambda}{2}\sum_{l=1}^{L}\left\|\mathrm{W}_l\right\|_F^2\right) = \frac{\lambda}{2}\frac{\partial}{\partial \mathrm{W}_l}\left\|\mathrm{W}_l\right\|_F^2 = \frac{\lambda}{2}\frac{\partial}{\partial \mathrm{W}_l}\mathrm{Tr}\left(\mathrm{W}_l\mathrm{W}_l^{\mathrm{T}}\right) = \lambda \mathrm{W}_l \tag{4.20}$$

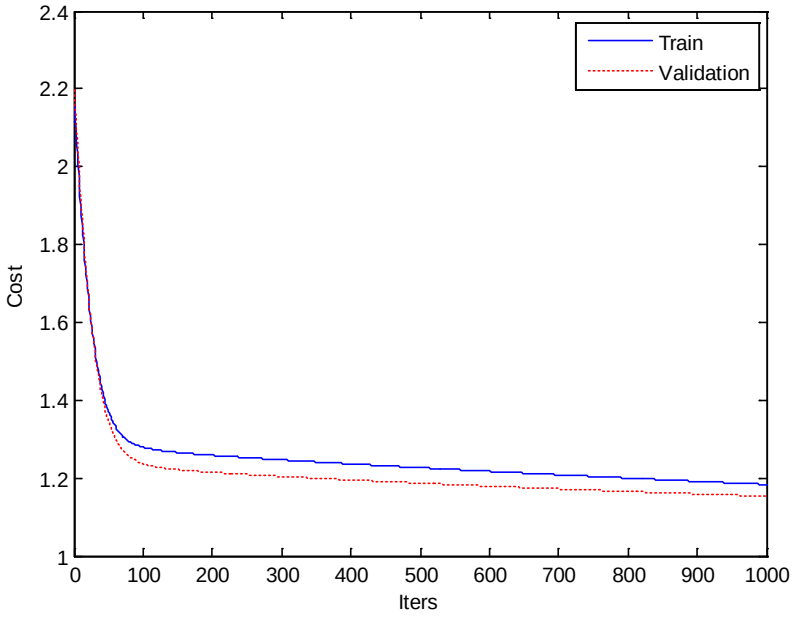
4.3 实验设计与结果分析

4.3.1 超参调节

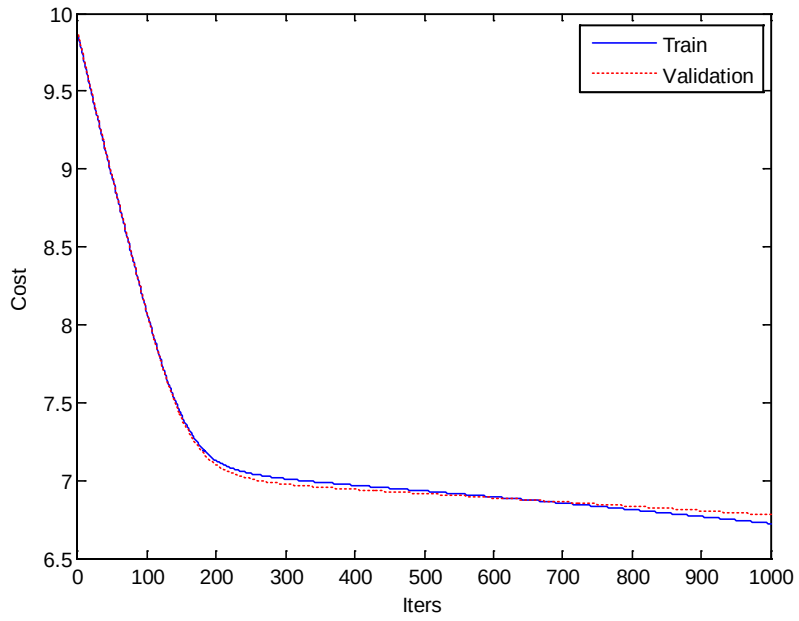
类似于其他神经网络模型，超参数对我们所提出的多标记神经网络的性能有比较大影响，例如损失函数、学习率、正则化权重、隐含层数及结点数等等，接下来我们将通过实验的方式对这些参数对模型的影响一一进行说明。实验时隐含层激活单元均采用当前最流行的 ReLU。测试用数据集用的是 Yeast，该多标记数据集来源于 Mulan（http://mulan.sourceforge.net/datasets-mlc.html），且已被预分成了训练集和测试集。按照标准的机器学习算法参数调节步骤，训练集用于模型学习，验证集用来做参数调节，而测试集用来进行不同模型之间的结果比较。因而这里我们需要把原始训练集根据 4:1 的比例随机划分成了训练集和验证集，并根据这两个子数据集进行参数选择。

学习率的影响 学习率决定了梯度下降时参数变化的步长，它对模型优化时的收敛性和迭代次数都有非常大的影响。假定我们的网络包含两个隐含层，每层结点 100 个，参数正则化权重设置为 0.001，最大迭代次数 1000，下面来看采用不同的学习率时的模型学习曲线和训练集及验证集上的准确率。

首先来看学习率为 0.01 时的情况。



（a）



（b）

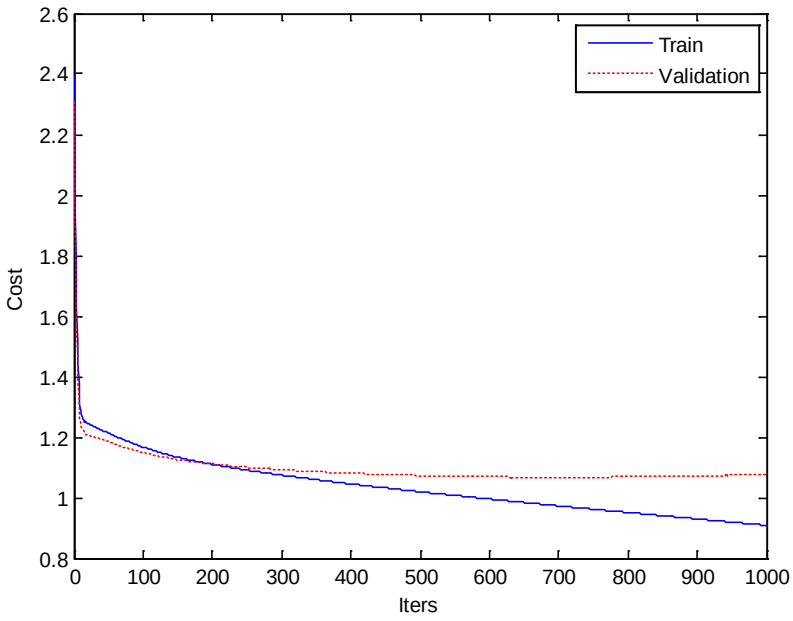
图 4.9 学习率为 0.01 时分别采用两种多标记损失函数的学习曲线

（a）采用 MLR 损失函数的学习曲线，（b）采用 MLCE 损失函数的学习曲线

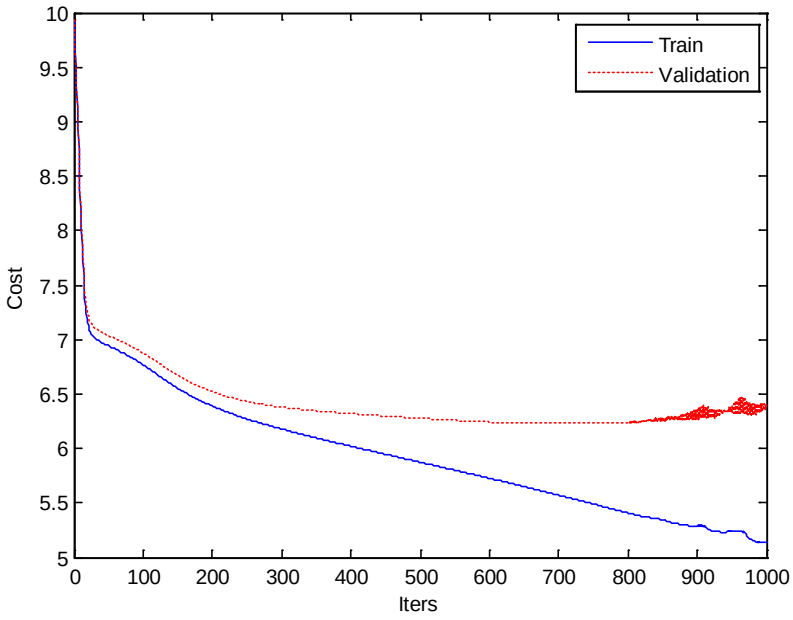
表 4.1 学习率为 0.01 时两种多标记损失函数对应的性能指标值

指标	MLR		MLCE	
	训练集	验证集	训练集	验证集
HammingLoss	0.2166	0.2114	0.2223	0.2221
SubsetAccuracy	0.0608	0.0567	0.0333	0.0267
microP	0.7360	0.7555	0.7515	0.7185
microR	0.4376	0.4552	0.4032	0.3950
microF1	0.5489	0.5681	0.5248	0.5097
macroP	0.3797	0.3579	0.2785	0.2683
macroR	0.2124	0.2132	0.1834	0.1746
macroF1	0.2252	0.2272	0.1855	0.1689
Accuracy	0.4033	0.4143	0.3772	0.3642
AUC	0.8134	0.8287	0.8135	0.8021
macroAUC	0.6501	0.6241	0.6583	0.6118
RankingLoss	0.1902	0.1716	0.1893	0.1960
OneError	0.2592	0.2233	0.2458	0.2700
Coverage	6.6033	6.5000	6.4983	6.4167
AveragePrecision	0.7293	0.7489	0.7327	0.7109

其次来看学习率为 0.1 时的情况。



（a）



（b）

图 4.10 学习率为 0.1 时分别采用两种多标记损失函数的学习曲线

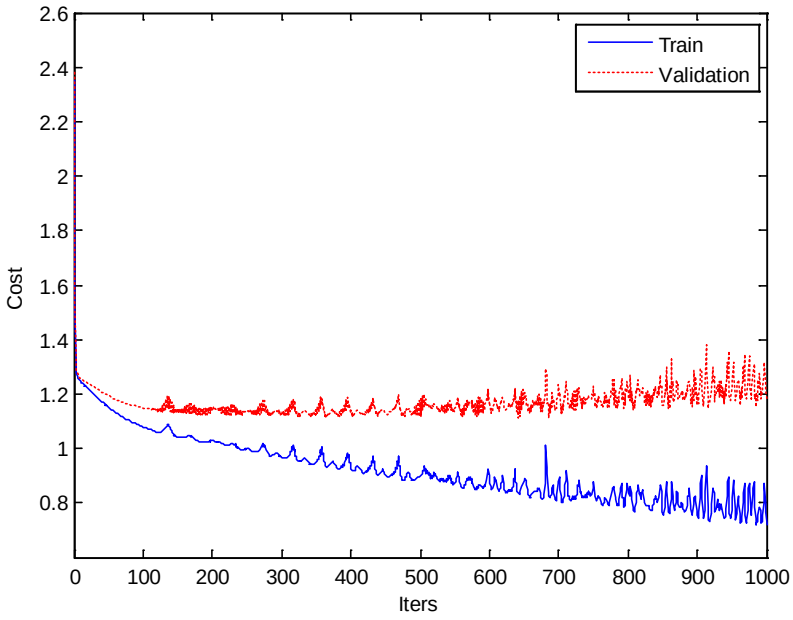
（a）采用 MLR 损失函数的学习曲线，（b）采用 MLCE 损失函数的学习曲线

表 4.2 学习率为 0.1 时两种多标记损失函数对应的性能指标值

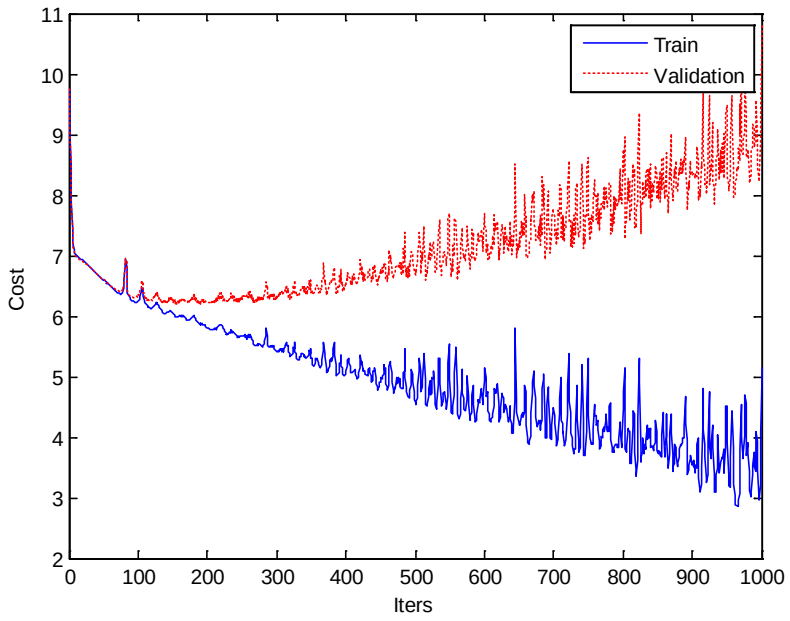
指标	MLR		MLCE	
	训练集	验证集	训练集	验证集
HammingLoss	0.1593	0.1926	0.1550	0.2055

SubsetAccuracy	0.2375	0.1867	0.2508	0.1567
microP	0.7818	0.6962	0.7945	0.6786
microR	0.6610	0.6073	0.6614	0.5702
microF1	0.7163	0.6487	0.7219	0.6197
macroP	0.5741	0.4544	0.6957	0.4781
macroR	0.4236	0.3619	0.4277	0.3610
macroF1	0.4523	0.3702	0.4637	0.3743
Accuracy	0.5888	0.5174	0.6023	0.4844
AUC	0.9020	0.8441	0.9050	0.8375
macroAUC	0.8501	0.6913	0.8534	0.7074
RankingLoss	0.1132	0.1662	0.1068	0.1679
OneError	0.1558	0.2067	0.1458	0.2600
Coverage	5.4575	6.2500	5.3075	6.0767
AveragePrecision	0.8281	0.7689	0.8419	0.7535

最后来看学习率为 0.3 时的情况。



（a）



（b）

图 4.11 学习率为 0.3 时分别采用两种多标记损失函数的学习曲线

（a）采用 MLR 损失函数的学习曲线，（b）采用 MLCE 损失函数的学习曲线

表 4.3 学习率为 0.3 时两种多标记损失函数对应的性能指标值

指标	MLR		MLCE	
	训练集	验证集	训练集	验证集
HammingLoss	0.1022	0.2002	0.1312	0.2733
SubsetAccuracy	0.3875	0.1600	0.3158	0.0967
microP	0.8991	0.7236	0.8354	0.5578
microR	0.7439	0.5599	0.7044	0.4566
microF1	0.8142	0.6313	0.7643	0.5022
macroP	0.7750	0.5698	0.7894	0.4468
macroR	0.5474	0.3634	0.6403	0.3503
macroF1	0.6219	0.4178	0.6327	0.3534
Accuracy	0.7048	0.4863	0.6365	0.3615
AUC	0.9567	0.8277	0.9268	0.7477
macroAUC	0.9437	0.6956	0.9581	0.6866
RankingLoss	0.0511	0.1790	0.0876	0.2673
OneError	0.0617	0.2367	0.1267	0.4367
Coverage	4.3258	6.6667	5.0892	7.5933
AveragePrecision	0.9151	0.7595	0.8652	0.6469

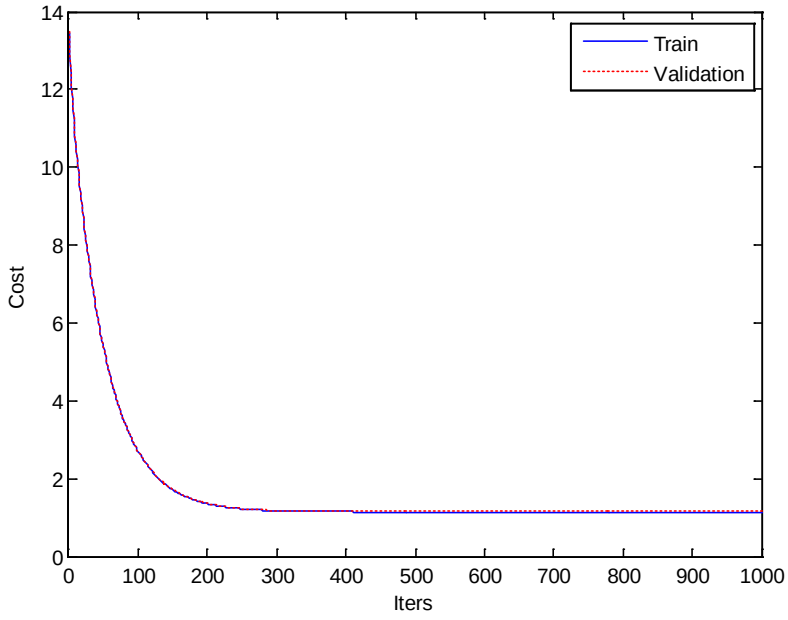
当学习率采用比较小的值（如 0.01）时，从学习曲线图来看，尽管已经迭代了 1000 次，但是无论训练集还是验证集的损失函数值都仍处于下降趋势，说明模型参数还未优

化到一个较好的状态。从测试结果来看，训练集和验证集的各项性能指标都很差，这是模型欠学习的结果。当学习率采用比较大的值（如 0.3）时，从学习曲线图来看，在迭代到 100 次左右时，模型开始进入发散状态。这是由于当参数优化到靠近局部最优点（或极值点）时，由于前进的步骤过大，使的更新后的参数值错过了局部极值点，进入到下一个非极值点状态，再次优化时将继续向着极值点前进，但是由于步长过大，还将由于过冲而错过。由于网络处于发散状态，最终的模型性能也充满了不确定性。只有当学习率设置为一个合适值（这里是 0.1）时才能使参数持续优化并逐步靠近极值点，训练后的模型也会处于一个良好的状态，在训练集也验证集上都能取得比较好的结果。

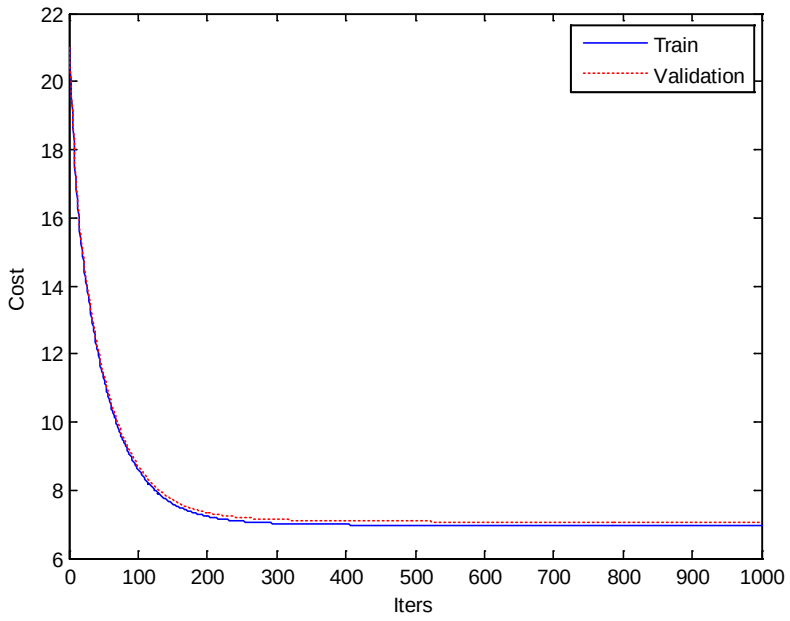
学习率对两种多标记损失函数的影响有一定的差异性。相比 MLCE 而言，MLR 看起来受学习率的影响要小一点；当学习率较大时，采用 MLR 的网络进入发散状态要慢，且发散的震荡幅度也小。

正则化权重的影响 正则化权重决定了数据损失和网络参数损失在目标函数中的相对重要性，通常对模型的泛化性能有较大的影响。假定我们的网络包含两个隐含层，每层结点 100 个，梯度下降的学习率为 0.1，最大迭代次数 1000。下面来看采用不同的正则化参数时的模型学习曲线和训练集及验证集上的实验结果。

首先来看正则化权重为 0.1 时的情况。



（a）



（b）

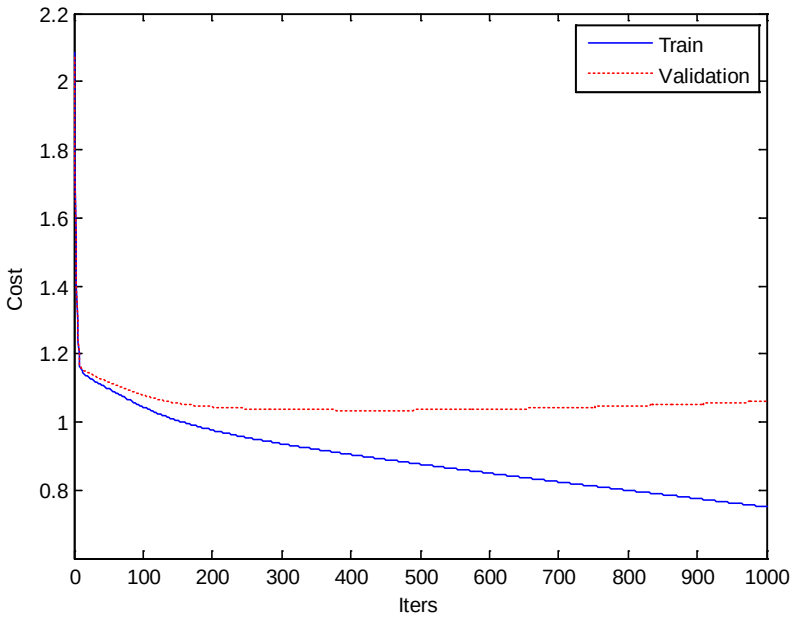
图 4.12 正则化权重为 0.1 时分别采用两种多标记损失函数的学习曲线

（a）采用 MLR 损失函数的学习曲线，（b）采用 MLCE 损失函数的学习曲线

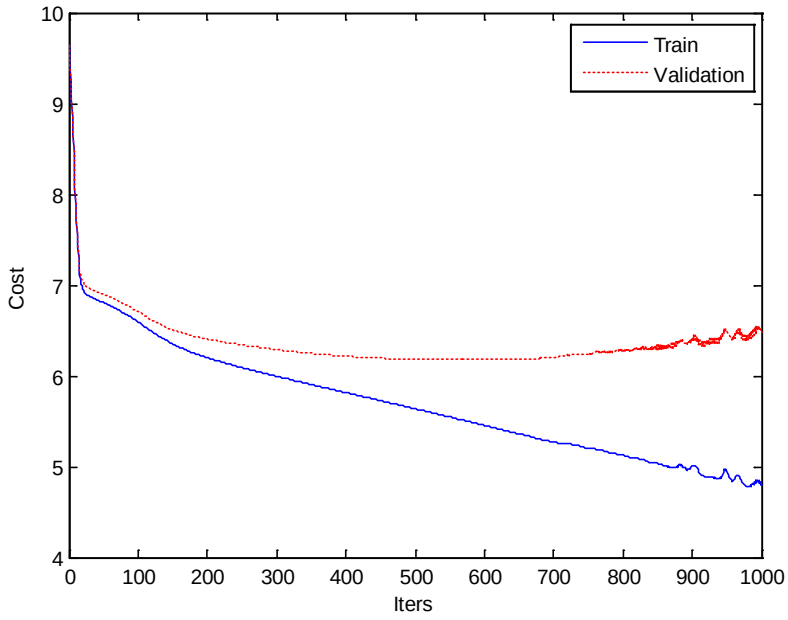
表 4.4 正则化权重为 0.1 时两种多标记损失函数对应的性能指标值

指标	MLR		MLCE	
	训练集	验证集	训练集	验证集
HammingLoss	0.2306	0.2333	0.2305	0.2338
SubsetAccuracy	0.0142	0.0067	0.0125	0.0133
microP	0.7446	0.7617	0.7488	0.7450
microR	0.3540	0.3532	0.3547	0.3503
microF1	0.4799	0.4826	0.4814	0.4765
macroP	0.1064	0.1088	0.1070	0.1064
macroR	0.1429	0.1429	0.1429	0.1429
macroF1	0.1219	0.1235	0.1223	0.1220
Accuracy	0.3350	0.3322	0.3352	0.3314
AUC	0.7795	0.7825	0.7821	0.7723
macroAUC	0.5311	0.5242	0.5572	0.5373
RankingLoss	0.2132	0.2076	0.2099	0.2207
OneError	0.2517	0.2333	0.2475	0.2500
Coverage	6.7775	6.7700	6.7458	6.8967
AveragePrecision	0.6998	0.7094	0.7031	0.6960

其次来看正则化权重为 0 时的情况。



（a）



（b）

图 4.13 正则化权重为 0 时分别采用两种多标记损失函数的学习曲线

（a）采用 MLR 损失函数的学习曲线，（b）采用 MLCE 损失函数的学习曲线

表 4.5 正则化权重为 0 时两种多标记损失函数对应的性能指标值

指标	MLR		MLCE	
	训练集	验证集	训练集	验证集
HammingLoss	0.1474	0.2100	0.1525	0.2002

SubsetAccuracy	0.2700	0.1500	0.2467	0.1600
microP	0.8061	0.6812	0.7761	0.7040
microR	0.6745	0.5680	0.6922	0.6032
microF1	0.7344	0.6195	0.7318	0.6497
macroP	0.6627	0.4877	0.6773	0.4751
macroR	0.4311	0.3556	0.4697	0.3889
macroF1	0.4631	0.3774	0.4994	0.4089
Accuracy	0.6118	0.4825	0.6054	0.5176
AUC	0.9137	0.8238	0.9097	0.8395
macroAUC	0.8738	0.6682	0.8713	0.7169
RankingLoss	0.0972	0.1820	0.1062	0.1612
OneError	0.1242	0.2467	0.1575	0.2267
Coverage	5.1708	6.4500	5.2983	6.3533
AveragePrecision	0.8503	0.7516	0.8307	0.7708

当正则化权重较大（如 0.1）时，由于正则化项在目标函数中的占比较大，模型在优化时一定程度上将忽略数据损失的影响，而驱使网络参数都趋近于 0。这就造成了虽然从学习曲线来看模型收敛情况良好，但模型却处于一个不健康的状态（网络大多参数都趋于 0，有效参数太少），在训练集和验证集上的测试结果当然也不会好了。如果正则化权重太小时（如极端情况下直接设置为 0），这时候损失函数中正则化项分量太轻甚至没有，训练出的模型往往在训练集上表现良好，但是在验证集上的结果却很差。因此，正则化项的权重也需要设置为一个合适的值，如 0.001。

隐含层层数和结点数的影响 隐含层的层数和结点数决定了网络的大小和模型的拟合能力。将正则化权重设为 0.001，梯度下降的学习率设为 0.1，最大迭代次数设为 1000，来看隐含层的层数以及结点数对模型性能的影响。

如下表所示，当隐含层从一层变成两层（每层的结点数目都设置为 100）时，无论是 MLR 还是 MLCE 损失函数，模型性能都有了较大的提升。但是隐含层数并不是越多越好，随着层数的增多，模型将会有较大的过拟合风险，且模型训练时的时间也将更长。所以通常隐含层数的多少跟学习任务的复杂度以及训练数据的规模有很大关系，像这里实验用的数据集比较小，设置两个隐含层就足够了。

表 4.6 使用不同隐含层数时两种多标记损失函数对应的性能指标

[100]表示单隐含层，结点数 100；[100 100]表示双隐含层，结点数都是 100

指标	MLR				MLCE			
	[100]		[100 100]		[100]		[100 100]	
	训练集	验证集	训练集	验证集	训练集	验证集	训练集	验证集
HammingLoss	0.1739	0.2033	0.1593	0.1926	0.1829	0.2138	0.1550	0.2055
SubsetAccuracy	0.1850	0.1533	0.2375	0.1867	0.1733	0.1333	0.2508	0.1567
microP	0.7627	0.6873	0.7818	0.6962	0.7436	0.6711	0.7945	0.6786
microR	0.6194	0.5782	0.6610	0.6073	0.6043	0.5578	0.6614	0.5702
microF1	0.6836	0.6280	0.7163	0.6487	0.6667	0.6092	0.7219	0.6197

macroP	0.5822	0.4362	0.5741	0.4544	0.5506	0.4040	0.6957	0.4781
macroR	0.3750	0.3354	0.4236	0.3619	0.3616	0.3224	0.4277	0.3610
macroF1	0.4002	0.3420	0.4523	0.3702	0.3830	0.3305	0.4637	0.3743
Accuracy	0.5510	0.4948	0.5888	0.5174	0.5357	0.4737	0.6023	0.4844
AUC	0.8848	0.8313	0.9020	0.8441	0.8716	0.8217	0.9050	0.8375
macroAUC	0.8244	0.6484	0.8501	0.6913	0.7792	0.6805	0.8534	0.7074
RankingLoss	0.1263	0.1730	0.1132	0.1662	0.1369	0.1854	0.1068	0.1679
OneError	0.1783	0.2233	0.1558	0.2067	0.1858	0.2467	0.1458	0.2600
Coverage	5.6608	6.3933	5.4575	6.2500	5.8217	6.5967	5.3075	6.0767
AveragePrecision	0.8075	0.7572	0.8281	0.7689	0.7960	0.7447	0.8419	0.7535

假定网络中有两个隐含层，当使用不同的隐含层结点数时的实验结果如下表所示，可见当隐含层结点数从 100 增加到 300 的时候，模型在验证集上的测试结果达到最好，而继续增加到 500 时，模型在训练集上结果进一步提升，但是在验证集上却下降了，说明训练出的模型有过拟合的情况。

表 4.7 使用不同隐含层结点数时 MLCE 损失函数对应的性能指标

指标	100		200		300		500	
	训练集	验证集	训练集	验证集	训练集	验证集	训练集	验证集
HammingLoss	0.1593	0.1926	0.1439	0.1952	0.1437	0.1838	0.1302	0.2010
SubsetAccuracy	0.2375	0.1867	0.2742	0.2000	0.2825	0.1733	0.3300	0.1733
microP	0.7818	0.6962	0.8144	0.7410	0.8051	0.7512	0.8394	0.6788
microR	0.6610	0.6073	0.6722	0.5787	0.6880	0.6052	0.7067	0.6042
microF1	0.7163	0.6487	0.7365	0.6499	0.7420	0.6704	0.7674	0.6393
macroP	0.5741	0.4544	0.7392	0.5140	0.6759	0.5462	0.7698	0.5267
macroR	0.4236	0.3619	0.4269	0.3593	0.4494	0.3617	0.4555	0.3706
macroF1	0.4523	0.3702	0.4622	0.3868	0.4762	0.3855	0.4877	0.3658
Accuracy	0.5888	0.5174	0.6152	0.5134	0.6221	0.5411	0.6565	0.5106
AUC	0.9020	0.8441	0.9178	0.8480	0.9209	0.8541	0.9349	0.8293
macroAUC	0.8501	0.6913	0.8912	0.6973	0.9074	0.7068	0.9277	0.7174
RankingLoss	0.1132	0.1662	0.0934	0.1643	0.0942	0.1467	0.0777	0.1769
OneError	0.1558	0.2067	0.1242	0.2100	0.1333	0.1600	0.0958	0.2433
Coverage	5.4575	6.2500	5.0258	6.4467	5.0467	6.2267	4.7767	6.2400
AveragePrecision	0.8281	0.7689	0.8544	0.7780	0.8464	0.8056	0.8787	0.7541

4.3.2 与其他多标记学习方法的比较

目前，已经有一些研究者将神经网络（深度网络）与多标记学习任务结合起来，并通过实验来验证模型的有效性。接下来将首先用我们提出的模型与这些方法进行比较。最早提出的多标记神经网络模型是 BPMLL，该网络属于浅层结构（只有一个隐含层），采用的激活函数还是当时最流行的 Sigmoid。最近的一个工作[63]是将深度置信网（DBN，Deep Belief Networks）用于多标记学习。这种网络的训练比较耗时，采用的是逐层预训练与微调结合的方式，具体来说就是每一层网络都要预先经过 RBM 训练一段时间，再

将各层预训练过的网络堆栈起来进行微调。

这里进行实验的数据集有三个，分别是 Yeast、Genbase 和 Medical，这三个数据集都来自生物医学领域，且都能从 Mulan 主页上获得。下载到的三个数据集均已预划分为训练集和测试集，下面进行模型比较时的结果都是在测试集上得到的。

这里要进行比较的多标记学习算法共有五种，分别是 1）我们提出的多标记深度网络 MLDL（损失函数为 MLR），2）MLDL^a：用 MLDL 模型进行特征学习，并在学习到的特征空间使用 hMuLab 多标记学习算法，3）多标记神经网络 BPMLL，4）DBN_{bp}：多标记深度置信网 DBN，5）DBN_{ECC}：用 DBN 学到的特征作为多标记学习算法 ECC 的输入。后三种方法得到的实验结果来自文献[63]，由于作者只使用了一个多标记性能指标 Accuracy，因此我们将用这个指标来评价各种方法的优劣。各种方法在三个数据集上的实验结果如下表所示。需要说明的是，所有模型都是在训练集上训练得到的，且超参也已通过在训练集上进行交叉验证进行了优化。

表 4.8 不同深度多标记学习方法在三个数据集上的 Accuracy 比较

Dataset	MLDL	MLDL^a	BPMLL	DBN_{bp}	DBN_{ECC}
Yeast	0.519	**0.533**	0.491	0.529	0.531
Genbase	0.987	**0.993**	0.049	0.984	0.985
Medical	0.769	**0.789**	0.053	0.746	0.742

我们所提出的多标记深度网络算法 MLDL 在损失函数和隐含层采用的激活函数上区别于其他基于神经网络的方法。从实验结果来看，MLDL 在 Yeast 数据集上的表现要比基于 DBN 的方法差，但是在其他两个数据集上则优于所有的其他方法。而将 MLDL 学到的特征作为 hMuLab 的输入时，性能得到进一步提升。

下面来比较一些非基于神经网络的多标记学习算法，主要有 MLKNN，RAkEL，Ensemble Classifier Chain（ECC），HOMER 和 RF-PCT。这些算法都可以在多标记学习 Java 包 Mulan 中直接调用。用来进行测试的生物医学领域的多标记数据集有两个，分别是 Yeast 和 Medical，因为文献[135]中已运用上述几个多标记学习算法在这两个数据集上进行了实验，且所有的结果都是将算法中使用的参数优化之后得到的，因此下面所有的结果比较都是客观公正的。正如前面所说，这两个数据集均已预分为了训练集和测试集，下面两个表中的结果都是在测试集上进行独立测试得到的。

表 4.9 多种多标记学习方法在 Yeast 数据集上的独立测试结果

↓ 表示越小越好，↑ 表示越大越好，每个指标的最好结果加粗

Metrics	MLDL	MLDL^a	ML-kNN	RAkEL	ECC	HOMER	RF-PCT
HammingLoss	0.192	**0.190**	0.198	0.192	0.207	0.207	0.197
SubsetAccuracy	0.190	**0.225**	0.159	0.201	0.215	0.213	0.152
Accuracy	0.519	0.530	0.492	0.531	0.546	**0.559**	0.478

microF1	0.650	0.658	0.625	0.656	0.658	**0.673**	0.617
macroF1	0.370	0.406	0.336	0.359	0.350	**0.447**	0.322
RankingLoss	0.171	**0.163**	0.172	0.259	0.224	0.205	0.167
OneError	0.229	**0.223**	0.234	0.254	0.249	0.248	0.248
Coverage	6.449	6.249	6.414	7.983	7.153	7.285	**6.179**
AveragePrecision	0.764	**0.773**	0.758	0.715	0.734	0.740	0.757

表 4.10 多种多标记学习方法在 Medical 数据集上的独立测试结果

↓ 表示越小越好，↑ 表示越大越好，每个指标的最好结果加粗

Metrics	MLDL	MLDL^a	ML-kNN	RAkEL	ECC	HOMER	RF-PCT
HammingLoss↓	**0.010**	**0.010**	0.017	0.012	0.014	0.012	0.014
SubsetAccuracy↑	0.678	**0.715**	0.462	0.607	0.526	0.610	0.538
Accuracy↑	0.769	**0.789**	0.528	0.673	0.611	0.713	0.591
microF1↑	0.817	**0.820**	0.634	0.714	0.714	0.773	0.693
macroF1↑	**0.338**	0.334	0.192	0.210	0.203	0.282	0.207
RankingLoss↓	**0.024**	**0.024**	0.045	0.159	0.152	0.090	**0.024**
OneError↓	0.141	**0.126**	0.279	0.312	0.315	0.216	0.174
Coverage↓	1.793	1.910	2.844	8.520	7.994	5.324	**1.619**
AveragePrecision↑	0.889	**0.897**	0.784	0.676	0.684	0.786	0.868

需要说明的是，ECC、HOMER 和 RF-PCT 都是基于集成学习的多标记学习算法，文献[135]通过细致的参数优化和大量的实验分析得出结论，这几种方法是当前性能最好的多标记学习算法。通过上面两个表中的结果来看，本章提出的 MLDL 模型与这些主流方法相比还是相当具有竞争力的。而如果将 MLDL 学习到的特征再输入到其他多标记学习算法中（如 hMuLab），则效果会更佳。

4.4 本章小结

本章主要研究通过修改深度网络的损失函数和输出层模型来构造多标记深度网络，并通过详细的实验来讨论超参数对所构造模型性能的影响，为有效使用该模型提供一些经验性的参考。为了验证模型的有效性，选择了当前主流的几种多标记学习算法（主要是基于集成学习方法），并在生物医学方面的几个多标记数据集上进行了测试比较。实验结果表明，本章所提出的多标记学习深度网络模型性能优异，可以作为解决多标记学习任务的一个有力工具。随着学习任务越来越复杂，数据量越来越大，基于深度学习的多标记学习方法将发挥越来越重要的作用。

第五章 多标记学习算法在抗菌肽活性预测中的应用

5.1 引言

抗菌肽（AMPs，Antimicrobial peptides），又称主动防御肽（HDPs，host defense peptides），存在于几乎所有类型生命体的天然免疫系统里面。抗菌肽是一种广谱抗生素，已经被证明能够杀死细菌、病毒、真菌甚至癌细胞[136]。这种类型的肽分子不仅能快速消除侵入性病原体(快于细菌复制的速度),而且能启动辅助的免疫响应以进一步清理系统。目前，人们迫切需要开发出新的抗菌剂用于治疗抗药性病原体，这使得抗菌肽领域的研究得以迅速发展[137]。

研究人员已经建立了多个抗菌肽数据库，如 CAMP（Collection of sequences and structures of antimicrobial peptides，http://www.camp.bicnirrh.res.in/index.php）[138]，APD（Antimicrobial Peptide Database，http://aps.unmc.edu/AP/）[139]，ADAM（A Database of Anti-Microbial peptides，http://bioinformatics.cs.ntou.edu.tw/adam/）[140]，DRAMP（Data Repository of Anti-Microbial Peptides，http://dramp.cpu-bioinfor.org/）[141]。这些抗菌肽数据库提供了多种多样的工具用于抗菌肽的查询、分析和预测。APD 是关于抗菌肽的一个综合性数据系统，致力于抗菌肽的命名、术语建立、分类、信息搜索、预测、设计和统计分析等。十多年来，所有 APD 收录的抗菌肽分子都是通过手工从文献（包括 PubMed, PDB, Google 和 Swiss-Prot）中得到。为了得到一个干净的数据集，APD 为数据收录提前建立了准则，只有满足下列条件的样本才会被收录：

（1）自然存在的

（2）被证明具有抗菌活性

（3）成熟肽的氨基酸序列已被阐明

（4）氨基酸序列长度在 100 以下（从 2012 年 10 月开始，也收录了一些长度大于 100 但是很重要的抗菌蛋白）

近年来，已经有很多的机器学习方法被用于抗菌肽的分类问题当中。但是大多数的研究工作聚焦于抗菌肽的识别(单标记二分类问题)或者预测抗菌肽具有哪一种功能(单标记多分类问题)。事实上，抗菌肽的活性预测是一个多标记学习问题，因为任一个抗菌肽可能同时具有多种活性。2013 年 Xiao 等人推出了一个两层的抗菌肽预测器[142]，第一层分类器用于识别抗菌肽，如果判断是，则第二层分类器会进行抗菌肽活性的多标记预测。但是该研究所使用的抗菌肽数据集只包含 878 个样本，覆盖 5 种活性，且在实验部分只用了一种多标记学习方法进行测试，缺乏特征提取方法和多标记学习算法的比较分析。本研究将根据最新版本的 APD 构建一个更大的数据集，覆盖多达 12 种活性，且进行多种特征提取方法和多标记学习方法的比较分析，包括我们自己新提出的一种多标记

学习算法。

5.2 数据集

本研究所采用的抗菌肽样本来自一个专业的抗菌肽数据库 APD，该数据库致力于收集自然存在的抗菌肽分子，包括序列、结构和活性信息。截止 2016 年 5 月，该数据库共收集了 2501 条抗菌肽分子，每个分子都有一个 APD ID 作为它的标识符。APD ID 以 AP 开头，后面是个五位数。众所周知，抗菌肽的活性并局限于抗菌性，APD 所收集的抗菌肽就覆盖了 12 种活性，且每个分子也不局限于仅仅拥有一种活性，而是可能同时具有多种活性。在机器学习领域，每种活性对应一个标记，因此抗菌肽活性预测可以看作一个多标记学习问题。

APD 中的抗菌肽生物活性术语及每种活性所具有的样本数如表 5.1 所示。从表中可以看出，最流行的活性是 Antibacterial，覆盖了大约 90%的抗菌肽分子，而最稀有的活性是 Anti-protist，仅仅 4 个抗菌肽分子具有这种活性。显而易见，APD 所收集的所有抗菌肽分子是一个多标记数据集，且 $LC = \frac{1}{2501}\sum_{i=1}^{2501}|y_i| = 1.54$，LD = LC/12 = 0.13，其中 $|y_i|$ 表示第 i 个样本所具有的活性种类数。我们也可以统计出具有不同种活性数的样本数，如表 5.2 所示，仅有一种活性的抗菌肽分子有 1449 个，占数据库中所有分子的约 58%。抗菌肽分子可以同时具有 9 种活性，但 APD 数据库中仅存在一个这样的样本。

表 5.1 不同活性的抗菌肽分子数

No.	Activity	Count
1	Antibacterial Peptides (Antibiofilms)	2255
2	Antiviral Peptides (Anti-HIV)	177
3	Antifungal Peptides	988
4	Antiparasitic Peptides (Antimalaria)	84
5	Anticancer Peptides	195
6	Anti-protist Peptides	4
7	Insecticidal Peptides	28
8	Spermicidal Peptides	12
9	Chemotactic peptides	56
10	wound healing	15
11	Antioxidant peptides	19
12	Protease inhibitors	22

表 5.2 具有不同活性数的抗菌肽分子数目及其百分比

Number of activities.	Number of AMPs	Percentage (%)
1	1449	57.94
2	829	33.15
3	172	6.88
4	34	1.36
5	12	0.48
6	2	0.08

7	1	0.04
8	1	0.04
9	1	0.04
10	0	0
11	0	0
12	0	0
In total	2501	100

APD 中的抗菌肽分子序列长度分布如图 5.1 所示，从图中可以看出，大多数的抗菌肽分子长度都在 5~60。该数据库中最短的序列是 AP02381，它仅有两个氨基酸组成，而最长的分子是 AP02157，它由 174 个氨基酸残基组成。

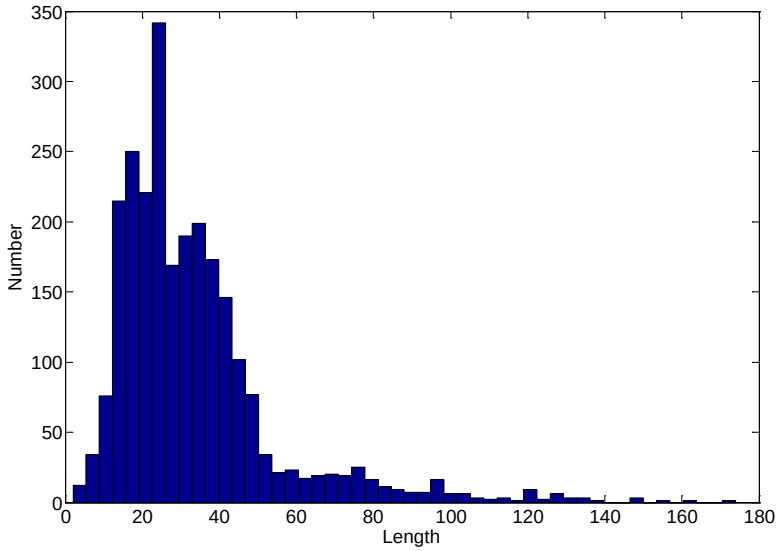


图 5.1 抗菌肽序列长度分布

从表 5.1 可以看出从 APD 中提取的原始数据集非常的不均衡，某些活性所具有的样本数非常少，因此我们又提供了一个过滤数据集，将样本数少于 50 的活性剔除。另外，由于很多小的肽分子可以通过化学修饰的方式展现出抗菌活性，而且一些大的蛋白质分子能够水解成多个肽分子从而拥有抗菌活性，因此我们又删除了序列长度小于 10 或者序列长度大于 60 的抗菌肽分子，这样的话仅保留了 2222 个抗菌肽分子，它们总共覆盖了 5 种活性，如表 5.3 所示。对于上述过滤数据集而言，LC= 1.49 and LD=0.30.

表 5.3 过滤后的数据集中不同活性的抗菌肽分子数

No.	Activity	Count
1	Antibacterial Peptides (Antibiofilms)	2006
2	Antiviral Peptides (Anti-HIV)	155
3	Antifungal Peptides	903
4	Antiparasitic Peptides (Antimalaria)	70
5	Anticancer Peptides	178

5.3 特征提取

序列特征提取是通过机器学习的方法来解决序列分类问题的基础，对于多标记学习也是如此。抗菌肽分子序列可以看作是一个个长度不一的非结构化的文本，我们需要将这些序列量化并提取出特征项来表示文本信息，从而将非结构化的序列数据转化成具有固定长度的结构化的特征向量，这样计算机就可以对序列信息进行识别处理了。对抗菌肽分子序列进行抽象和量化的模型有很多种，常见的有氨基酸成分、二联体成分和关联因子等，下面我们会一一进行详细的阐述，首先让我们看看抗菌肽分子序列的组成是什么样子的。

5.3.1 抗菌肽分子序列

抗菌肽是由氨基酸通过肽键连接而成的短肽链。若用符号来描述的话，可以将一个长度为 L 的抗菌肽序列用下式表示

$$\mathbf{P} = A_1A_2A_3A_4A_5A_6\ldots A_L \tag{5.1}$$

其中 A_1 是序列的第一个氨基酸残基，A_2 是第二个，以此类推。

自然界共存在 22 种氨基酸可以用来合成多肽或者蛋白质，这些氨基酸称为自然氨基酸，其中有 20 种是直接通过通用遗传密码子编码得到的，称为标准氨基酸，他们的名称和符号如表 5.4 所示。这 20 种氨基酸以外的氨基酸统称为非标准氨基酸，在已知的蛋白质或多肽中非常少见，因此通常我们只考虑标准氨基酸[143]。

表 5.4 20 种标准氨基酸的名称及符号

名称	三字符号	单字符号
甘氨酸	Gly	G
丙氨酸	Ala	A
缬氨酸	Val	V
亮氨酸	Leu	L
异亮氨酸	Ile	I
苯丙氨酸	Phe	F
色氨酸	Trp	W
酪氨酸	Tyr	Y
天冬氨酸	Asp	D
天冬酰胺	Asn	N
谷氨酸	Glu	E
赖氨酸	Lys	K
谷氨酰胺	Gln	Q
甲硫氨酸	Met	M

丝氨酸	Ser	S
苏氨酸	Thr	T
半胱氨酸	Cys	C
脯氨酸	Pro	P
组氨酸	His	H
精氨酸	Arg	R

5.3.2 氨基酸成分

通过文献可知，氨基酸组成是肽分子分类与设计的最重要的影响因素，因而我们可以提取氨基酸成分来作为抗菌肽分子的特征。氨基酸成分的提取方法非常简单，它实际上就是经典的向量空间模型（VSM，Vector Space Model）[144; 145]在生物信息学中的应用。VSM 是在 20 世纪 70 年代由 Salton 等人提出，并成功应用于自然语言处理领域。它的概念非常简单，就是将非结构化的语言文本表达为向量空间里的向量（点），从而将对文本内容的处理简化为对向量空间里的向量的操作。向量空间里的每一维对应的是某个词条的出现频率（或者频率的函数）。就我们要提取的氨基酸成分而言，可以将每个抗菌肽分子序列看作是一个文本，且文本中词的长度只能为 1，由于我们只考虑 20 种标准氨基酸，因此这里的词向量空间是 20 维，每一维对应某种标准氨基酸在该序列中的出现频率。每条序列都被表达为 20 维的氨基酸成分向量$[f_1, f_2, \cdots, f_{20}]^T$，其中 $f_i(i=1,2,\cdots,20)$是第 i 种氨基酸在该序列中的出现频率，T 是转置操作符。氨基酸成分特征提取方法已经成功用在了很多蛋白质或肽分类问题当中[146; 147]。

为了展现不同抗菌肽活性之间的异同性，我们将每种活性类型所包含的样本的平均氨基酸成分向量展示在图 5.2 中，从图中可以看出有些活性的氨基酸成分模式还是非常相似的，这不利于区分出不同活性之间的差异性。

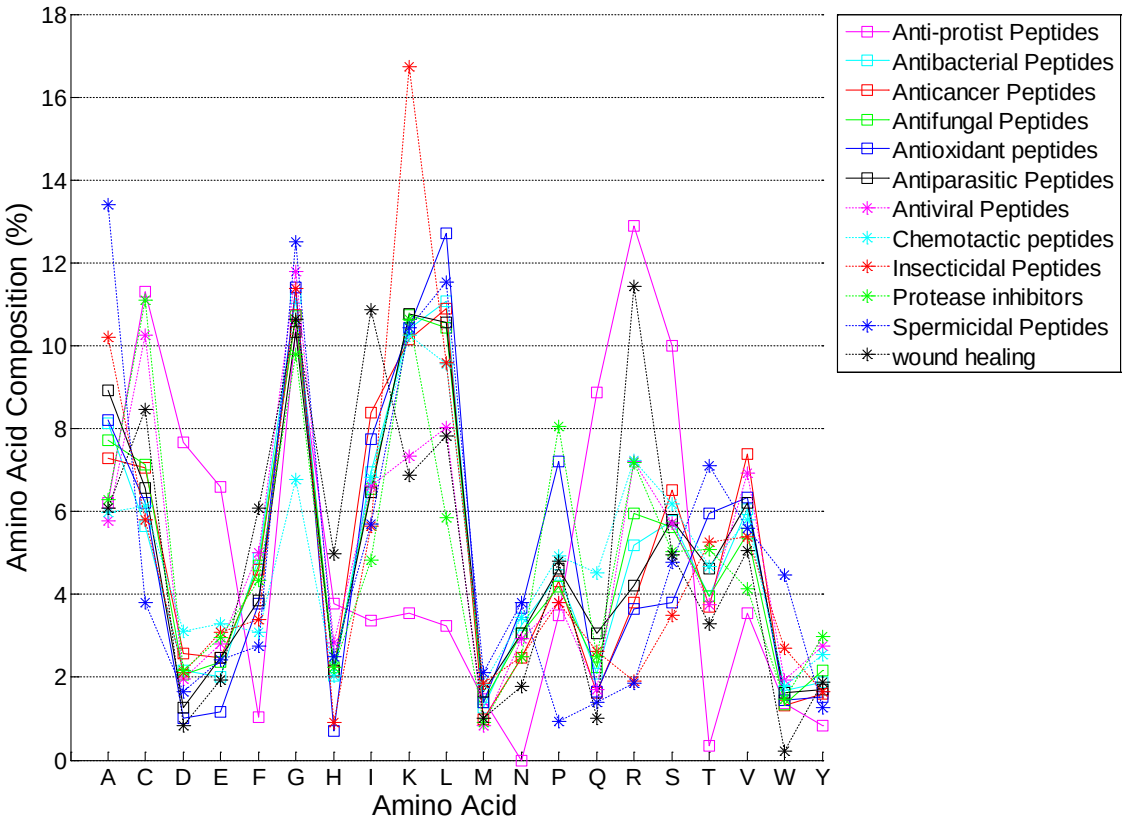


图 5.2 不同活性的抗菌肽分子序列的平均氨基酸成分

5.3.3 二联体成分

仅仅利用氨基酸成分来描述抗菌肽分子有一个致命缺陷，那就是完全丢失了序列的顺序信息，例如抗真菌肽分子 AP02381 由“EL”组成，而序列“LE”与“EL”的氨基酸成分完全相同，但是我们并不确定它也是抗真菌肽，我们甚至不能确定它也属于抗菌肽。因此我们又引入了二联体成分来描述抗菌肽分子序列[148]。二联体指的是相邻两个氨基酸组成的二肽，由于我们只考虑 20 种标准氨基酸，因而这样的二肽共有 400 种。如果将每个抗菌肽分子序列看作是一个文本的话，二联体成分的提取方法同样可以使用 VSM，词向量空间是 400 维，每一维对应某种二联体在该序列中的出现频率。需要说明的是，在提取二联体成分时，并不需要设计复杂的分词规则，这里的词条长度固定为 2，且分词时的步进值为单个字符（每个字符对应一个氨基酸）。根据上述方法可以将任一抗菌肽分子序列表达为一个 400 维的特征向量，可以用符号标记为$[d_1, d_2, \ldots, d_{400}]^{\mathrm{T}}$，其中 d_i（i=1,2,…,400）是第 i 种二联体在该序列中的出现频率。

通常在应用时，人们会将氨基酸成分和二联体成分结合起来，构成 420 维的特征空间，在本研究种就是将抗菌肽分子序列描述为如下的特征向量

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_{20}, x_{21}, x_{22}, \ldots, x_{420}]^{\mathrm{T}} \tag{5.2}$$

其中前 20 个特征是氨基酸成分，后 400 个特征是二联体成分。

5.3.4 伪氨基酸成分

伪氨基酸成分（PseAAC，pseudo amino acid composition）也是非常常用的蛋白质或多肽序列的特征提取方法[149-153]，它实际上是在氨基酸成分的基础上又引入了关联因子特征。在这种方法种，一个氨基酸序列被转化为如下特征向量

$$\mathbf{P}=[\,p_1\ \ \cdots\ \ p_{20}\ \ p_{20+1}\ \ \cdots\ \ p_{20+\lambda}\,]^{\mathrm{T}} \tag{5.3}$$

其中

$$p_u=\begin{cases}\dfrac{f_u}{\sum_{i=1}^{20}f_i+w\sum_{j=1}^{\lambda}\theta_j}, & (1\le u\le 20)\\[2ex] \dfrac{w\theta_{u-20}}{\sum_{i=1}^{20}f_i+w\sum_{j=1}^{\lambda}\theta_j}, & (20+1\le u\le 20+\lambda;\ \ \lambda<L)\end{cases} \tag{5.4}$$

$$\begin{cases}\theta_1=\dfrac{1}{L-1}\sum_{i=1}^{L-1}\Theta(\mathrm{R}_i,\mathrm{R}_{i+1})\\ \theta_2=\dfrac{1}{L-2}\sum_{i=1}^{L-2}\Theta(\mathrm{R}_i,\mathrm{R}_{i+2}) \qquad (\lambda<L)\\ \vdots\\ \theta_\lambda=\dfrac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}\Theta(\mathrm{R}_i,\mathrm{R}_{i+\lambda})\end{cases} \tag{5.5}$$

上式中的 θ_1 是一阶关联因子，θ_2 是二阶关联因子，依次类推；w 是关联因子的权重，通常设置为 0.1；关联函数的定义如下：

$$\Theta(\mathrm{R}_i,\mathrm{R}_j)=H(\mathrm{R}_i)\cdot H(\mathrm{R}_j) \tag{5.6}$$

其中 H(Ri)是氨基酸 R_i 的物理化学属性值。在下面的实验中，将会应用到 5 种物理化学属性，分别是(1) hydrophobicity，(2) pK1 (C^{α}-COOH)，(3) pK2 (NH3) ，(4) PI (25ºC)，和(5) molecular weight。所有 20 种标准氨基酸对应这 5 种物理化学属性的指标值如下表所示，

表 5.5 20 种标准氨基酸的物理化学属性值

	hydrophobicity	pK1	pK2	PI	molecular weight
A	1.80	2.35	9.87	6.01	89.09
C	2.50	1.92	10.70	5.05	121.15
D	-3.50	1.99	9.90	2.85	133.10
E	-3.50	2.10	9.47	3.15	147.13
F	2.80	2.20	9.31	5.49	165.19
G	-0.40	2.35	9.78	6.06	75.07
H	-3.20	1.80	9.33	7.60	155.16
I	4.50	2.32	9.76	6.05	131.17

K	-3.90	2.16	9.06	9.60	146.19
L	3.80	2.33	9.74	6.01	131.17
M	1.90	2.13	9.28	5.74	149.21
N	-3.50	2.14	8.72	5.41	132.12
P	-1.60	1.95	10.64	6.30	115.13
Q	-3.50	2.17	9.13	5.65	146.15
R	-4.50	1.82	8.99	10.76	174.20
S	-0.80	2.19	9.21	5.68	105.09
T	-0.70	2.09	9.10	5.60	119.12
V	4.20	2.39	9.74	6.00	117.15
W	-0.90	2.46	9.41	5.89	204.23
Y	-1.30	2.20	9.21	5.64	181.19

由于不同的物理化学属性值所在数值范围差异较大，在带入公式之前需要将每种属性值做一个标准化

$$y_i = \frac{x_i - \mathrm{mean}(x)}{\mathrm{std}(x)} \tag{5.7}$$

其中 x_i（i=1,2,…,20）是第 i 种氨基酸的原始属性值，y_i 是对应的标准化之后的值，mean(x)是 20 种氨基酸的平均属性值，std(x)是 20 种氨基酸的属性值之标准差。

5.4 多标记学习算法

本研究采用的多标记学习算法由两个串行模块组成，如图 5.3 所示。第一个模块是权重 K 近邻算法，用来将输入样本的特征向量转化成一个 c 维的标记得分向量，这是一个非线性转化过程。第二个模块是多输出线性回归（MLR），用来得到最终多标记学习模型输出，该模块实际上与极限学习机的输出层相同。本章提出的多标记学习模型也可以看成是一个三层的神经网络，其与极限学习机的不同之处在于隐含层，前者是一个有监督的非线性映射，而后者是一个无监督的随机映射。在训练过程，所有训练样本的标记得分向量通过 Leave-one-out 的方式获得，意思是对任一个训练样本，其余的训练样本用作近邻搜索，并用加权 K 近邻算法计算出该训练样本的标记得分向量。这种方式有利于防止训练出的模型具有偏见，否则的话训练样本的标记得分向量与测试样本的标记得分向量将会显著差异（测试样本输入时，它自身总是不在近邻搜索范围）。当所有训练样本的标记得分向量通过第二个模块时，MLR 的参数通过优化一个代价函数而获得。经过训练之后，待测样本的模型输出和预测标记将会很容易获得。

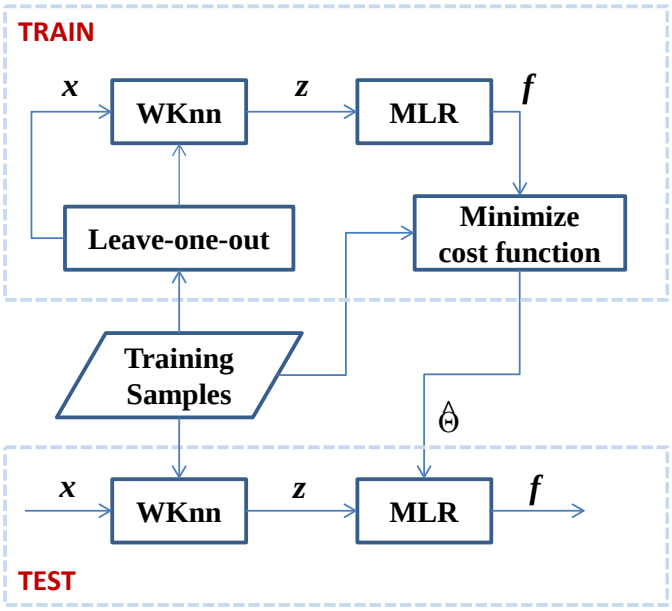


图 5.3 本文提出的多标记学习算法流程

本研究采用的加权 K 近邻算法是一种相似度加权方法，即与待测样本相似度越高的训练样本具有越高的投票权重。样本 x_A 和 x_B 之间的相似度通过最大-最小法获得，

$$Sim(\boldsymbol{x}_A, \boldsymbol{x}_B) = \frac{\sum_{i=1}^{d} x_{A,i} \wedge x_{B,i}}{\sum_{i=1}^{d} x_{A,i} \vee x_{B,i}} \tag{5.8}$$

其中∧表示取小运算，∨表示取大运算。对于一个待测样本 x，它在训练集 D 中的 K 个最近邻及其标记为 $(\boldsymbol{x}_k^*, \boldsymbol{y}_k^*), 1 \le k \le K$。假设 x 与这些近邻之间的相似度为 $s_k\,(1 \le k \le K)$，且相似度已根据大小排序为 $s_1 \le s_2 \le \cdots \le s_K$，则 K 个最近邻的权重可以定义如下

$$w_k = \begin{cases} \dfrac{s_k - s_1}{s_K - s_1}, & s_K \ne s_1 \\ 1, & s_K = s_1 \end{cases} \tag{5.9}$$

得到近邻的权重之后，我们就可以计算出待测样本 x 的标记得分

$$z_j = \sum_{\gamma_j \in y_k^*} w_k \Big/ \sum_{k=1}^{K} w_k, \quad 1 \le j \le c \tag{5.10}$$

显而易见，越多的近邻具有标记 l_j，则对该标记的得分越大，且当所有近邻都具有该标记时得分为最大值 1，当所有近邻都不具有该标记时得分为最小值 0.

当任一样本都通过上述的相似度加权 K 近邻算法转化为标记得分向量后，它的最终输出将通过 MLR 模块得到。假定 $Z \in \mathbf{R}^{m \times (c+1)}$ 为标记得分矩阵，其中第 i 行对应训练样本

x_i 的标记得分向量 z_i，且 z_i 已增广为 $\boldsymbol{z}_i = [1 \ z_{i,1} \ z_{i,2} \ \cdots \ z_{i,c}]^{\mathrm{T}}$。又设 $\mathrm{Y} \in \mathbf{R}^{m \times c}$ 为标记矩阵，其第 i 行对应训练样本 x_i 的二值标记向量。我们有如下优化目标

$$\min J(\Theta) = \frac{1}{2}\left\|\mathrm{Y} - \mathrm{Z}\Theta\right\|_F^2 + \frac{\lambda}{2}\left\|\Theta\right\|_F^2 \tag{5.11}$$

其中 $\Theta \in \mathbf{R}^{(c+1)\times c}$ 是系数矩阵，$\left\|\bullet\right\|_F$ 是 Frobenius 范数。上式的右边第一项是误差平方和项，第二项是正则项，用来降低参数幅值并减小过拟合风险。非负数 λ 是两项之间的权重项。系数矩阵 Θ 可以通过求导的方法获得，即令 $\nabla_{\Theta} J(\Theta) = 0$，得到如下最优的参数估计

$$\hat{\Theta} = (\mathrm{Z}^{\mathrm{T}}\mathrm{Z} + \lambda \mathrm{I})^{-1}\mathrm{Z}^{\mathrm{T}}\mathrm{Y} \tag{5.12}$$

对任一待测样本 x，它的增广标记得分向量为 z，则其最终的输出为

$$f(\boldsymbol{x}, \mathrm{L}) = z^{\mathrm{T}}\hat{\Theta} \tag{5.13}$$

上述模型有利于将标记之间的相关性考虑进来。例如待测样本 x 的最终输出不仅与它自身的标记得分有关，而且还与其他的标记得分有关。在最小化误差平方和的框架下，如果一个样本与某个标记相关，则它对该标记的最终输出将趋于 1，相反，如果它不与该标记相关，则其最终输出将趋于-1。我们将中间值 0 设置为阈值以区分出相关标记和不相关标记。

5.5 实验设计与结果分析

5.5.1 参数讨论

本研究提出的模型值包含两个超参数，分别是近邻数 K 和正则参数 λ。我们将通过 Hold-out 测试来评估这两个参数对模型性能的影响。在该测试中，我们随机挑出了 1/3 的样本用于测试，余下的 2/3 样本用于模型训练。将两个参数联合起来进行二维网格搜索时，多标记学习评价指标值如图 5.4 所示，从图中可以看出，相比于近邻数 K，正则参数 λ 的影响非常小。随着 K 值的增大，一开始的时候模型性能提升非常快，但是当 K=10 之后，模型性能进入了一个平台期。考虑所有的指标，我们选择 K=15 和 λ=1 作为模型的默认参数值。

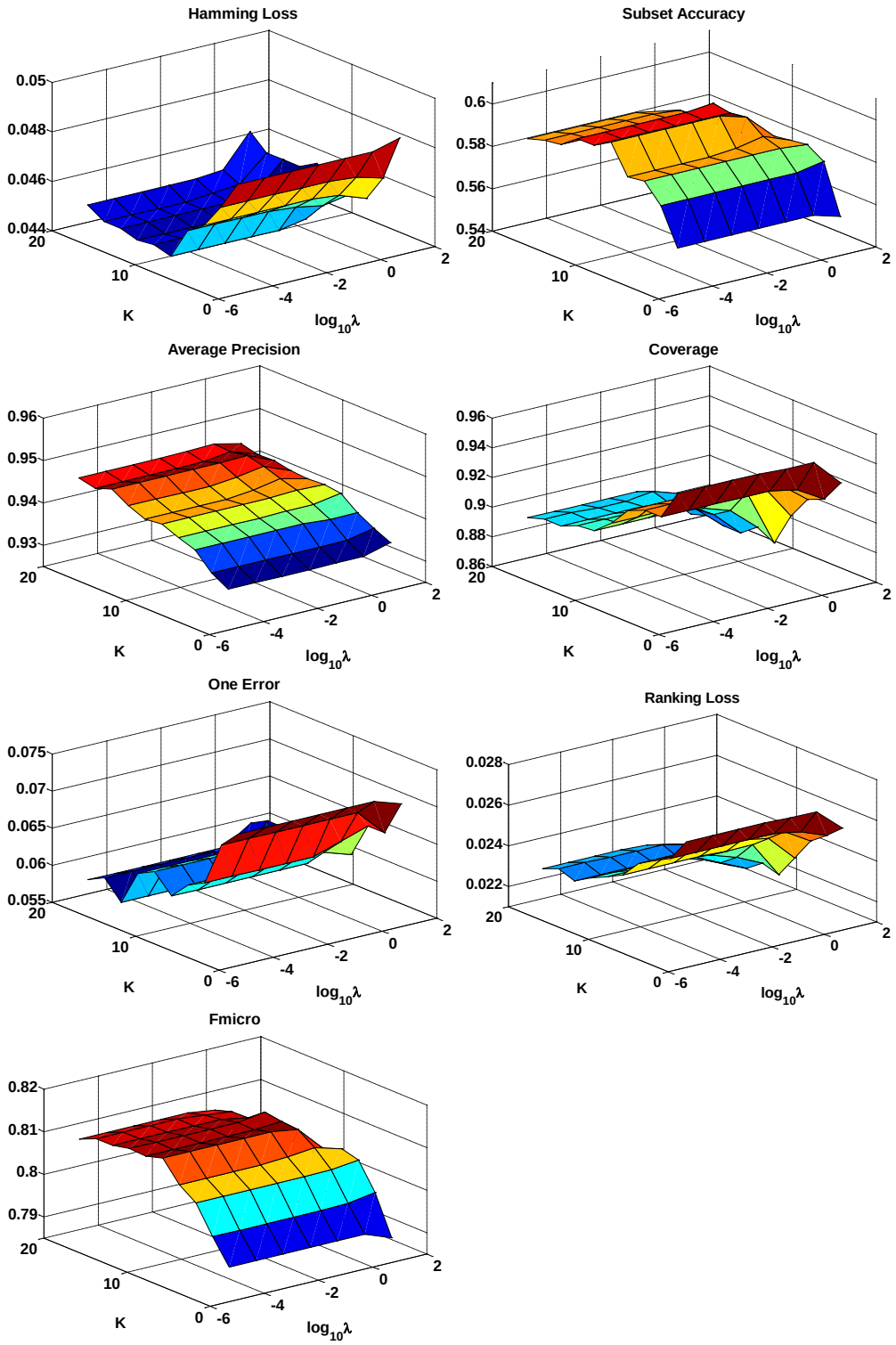


图 5.4 不同超参数下的性能指标

5.5.2 模型比较

在本小结我们将在上面建立的抗菌肽数据集上测试本文提出的多标记学习方法以及其他多种常用的多标记学习算法，包括 MLkNN，BPMLL，IBLR，RAkEL，CC 和 ECC。所有用到的其他方法都可以在多标记学习工具包 Mulan 中实现，且这些方法用到的超参

数通过交叉验证确定，并选择最优参数下的结果用于比较。

在 Mulan 工具包中，MLkNN 和 IBLR 算法都是基于近邻规则，且都只包含一个超参数，那就是近邻数目，在实验中，我们令其从 2 逐渐增大到 20，步进值为 2。BPMLL 算法将使用默认参数。对于需要基分类器的多标记学习算法，我们使用 J48 算法（C4.5 算法的一种具体实现）。RAkEL 算法有两个超参数需要调节，一是 NumberOfModel(number of models)，我们令其分别为 1*c，2*c 和 3*c（c 是所有可能的标记数），另一个参数是 SizeOfSubset(size of subset)，我们考虑在 3,6,9 之间选择最优。分类链方法 CC 不需要额外的超参数，但是其集成方法 ECC 有 3 个参数，分别是 NumOfModels, doUseConfidences (Whether the output is computed based on the average votes or on the average confidences), and doUseSamplingWithReplacement (Whether to use sampling with replacement to create the data of the models of the ensemble)，其中第一个参数设置为 30，其余两个参数设置为“true”。

上述所有的多标记学习方法都在 AMP 数据集上进行 10 次的 5 折交叉验证，7 个评价指标的平均结果及其标准差如表 5.6 所示。从表中易见，本研究所采用的多标记学习方法取得的平均结果要好于所有其他的方法，且指标值的标准差也总是相对较小。实验结果表明，本研究提出的多标记学习方法不仅比其他方法更有效，而且性能很稳定。所有方法中，BPMLL 看起来效果最差。BPMLL 是一种 BP 神经网络的多标记变种，它使用一种称为排序损失的代价函数来实现多标记学习的目的。但是文献中的实验结果表明，将排序损失改为其他另一种损失函数时能显著提高神经网络的学习能力。因此在本研究所采用的方法中，我们也没有采用排序损失函数，而采用如公式所示的损失函数。MLkNN 和 IBLR 都是基于近邻规则的多标记学习方法，且他们在 AMP 数据集上取得的实验结果也非常接近。非常有趣的是，尽管 CC 方法在 AMP 数据集上的表现不佳，但是它的集成版本 ECC 却在性能上提高得非常明显。

表 5.6 在原始数据集上不同算法的五折交叉验证结果

Metric \ Method	Proposed	MLkNN	BPMLL	IBLR	RAkEL	CC	ECC
Hamming Loss ↓	0.0454 ±0.0004	0.0528 ±0.0006	0.2977 ±0.0227	0.0523 ±0.0007	0.0540 ±0.0007	0.0600 ±0.0015	0.0502 ±0.0008
Subset Accuracy ↑	0.5988 ±0.0049	0.5450 ±0.0040	0.0014 ±0.0010	0.5494 ±0.0048	0.5258 ±0.0069	0.4992 ±0.0082	0.5662 ±0.0082
Average Precision ↑	0.9439 ±0.0011	0.9326 ±0.0016	0.6691 ±0.0680	0.9326 ±0.0013	0.8853 ±0.0023	0.8474 ±0.0044	0.9210 ±0.0020
Coverage ↓	0.9337 ±0.0104	0.9859 ±0.0105	2.0595 ±0.1971	0.9980 ±0.0084	1.8996 ±0.0332	2.0774 ±0.0698	1.3728 ±0.0207
One Error ↓	0.0607 ±0.0018	0.0768 ±0.0026	0.4820 ±0.1523	0.0752 ±0.0025	0.1028 ±0.0029	0.1711 ±0.0070	0.0756 ±0.0030
Ranking Loss ↓	0.0234 ±0.0005	0.0269 ±0.0006	0.1120 ±0.0197	0.0275 ±0.0005	0.0809 ±0.0026	0.0947 ±0.0045	0.0473 ±0.0014
Fmicro ↑	0.8082	0.7679	0.4437	0.7758	0.7828	0.7574	0.7896

±0.0015	±0.0022	±0.0190	±0.0025	±0.0030	±0.0048	±0.0035

为了使得各种多标记学习方法之间的比较更具有统计意义，我们又在 10 次的交叉验证结果上进行成对 t 检验（Pvalue<0.05），并使用比较三元表 CT(A,B)=(better/tie/worse)来统计三种比较结果，一是算法 A 要显著优于算法 B，二是两种算法之间没有显著区别，三是算法 A 要显著劣于算法 B。表 5.7 汇总了所有算法之间的比较三元表，表中的每个三元表是行对应的算法（算法 A）与列对应算法（算法 B）之间的比较结果。每个三元表的数字之和为 7，对应的是用到的多标记学习算法评价指标的数目。表中最后一列的三元表是对应行中所有三元表之和。很令人惊讶，本研究所采用的多标记学习算法在所有评价指标上都要显著优于所有其他的方法。表现第二好的是集成多标记学习方法 ECC，除了我们提出的方法，它几乎在所有指标上都要显著优于其他方法（还有两个比较结果为 tie）。IBLR 和 MLkNN 都是基于近邻法的多标记学习方法，前者的表现要略好于后者，但他们都要显著落后于本文提出的方法。根据表 5.7 的最后一列——总比较三元表，可将所有多标记学习算法进行排序为 Proposed > ECC > IBLR > MLkNN > RAkEL > CC > BPMLL，其中符号>表示“好于”。需要指出的是，所有的比较结果都只是在 AMP 数据集上取得的，在其他数据集上的结果还有待验证。

表 5.7 在原始数据集上不同算法之间的比较三元表(better/tie/worse)

A \ B	Proposed	MLkNN	BPMLL	IBLR	RAkEL	CC	ECC	In total
Proposed	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	42/0/ 0
MLkNN	0/0/7	0/7/0	7/0/0	2/4/1	3/1/3	7/0/0	0/1/6	19/6/17
BPMLL	0/0/7	0/0/7	0/7/0	0/0/7	0/0/7	0/1/6	0/0/7	0/1/41
IBLR	0/0/7	1/4/2	7/0/0	0/7/0	3/1/3	7/0/0	0/0/7	18/5/19
RAkEL	0/0/7	3/1/3	7/0/0	3/1/3	0/7/0	7/0/0	0/1/6	20/3/19
CC	0/0/7	0/0/7	6/1/0	0/0/7	0/0/7	0/7/0	0/0/7	6/1/35
ECC	0/0/7	6/1/0	7/0/0	7/0/0	6/1/0	7/0/0	0/7/0	33/2/7

接下来我们又在经过过滤的 AMP 数据集上进行了实验。当使用本文提出的方法进行 Hold-out 测试时，各个评价指标与两个超参数之间关系图与图 5.4 非常相似，因此在这里还可以选择 K=15 和 λ=1 作为默认参数。其他算法用到的超参数寻优过程则类似于本节开始时介绍的步骤，并将最优参数对应的结果用于比较。在过滤数据集上进行 10 次的 5 折交叉验证，所有评价指标的平均结果及其标准差如表 5.8 所示，而所有算法之间的比较三元表如表 5.9 所示，从这两个表来看，在过滤数据集上，本文提出的方法依然表现得最好。

表 5.8 在过滤数据集上不同算法的五折交叉验证结果

Metric \ Method	Proposed	MLkNN	BPMLL	IBLR	RAkEL	CC	ECC
Hamming Loss ↓	0.0992	0.1083	0.6366	0.1073	0.1139	0.1258	0.1055
	±0.0014	±0.0009	±0.0214	±0.0007	±0.0023	±0.0025	±0.0007
Subset Accuracy ↑	0.6141	0.5874	0.0022	0.5901	0.5594	0.5280	0.5928

	±0.0056	±0.0033	±0.0008	±0.0040	±0.0065	±0.0108	±0.0035
Average Precision ↑	0.9553	0.9501	0.4018	0.9506	0.9289	0.8821	0.9505
	±0.0010	±0.0008	±0.0416	±0.0011	±0.0022	±0.0049	±0.0009
Coverage ↓	0.6899	0.7050	2.7546	0.7006	0.8208	1.0545	0.7032
	±0.0054	±0.0038	±0.2624	±0.0034	±0.0108	±0.0253	±0.0051
One Error ↓	0.0565	0.0669	0.9224	0.0670	0.0888	0.1517	0.0661
	±0.0021	±0.0012	±0.0562	±0.0020	±0.0037	±0.0085	±0.0019
Ranking Loss ↓	0.0444	0.0481	0.6203	0.0471	0.0714	0.1223	0.0473
	±0.0010	±0.0006	±0.0829	±0.0008	±0.0025	±0.0053	±0.0010
Fmicro ↑	0.8226	0.8011	0.4509	0.8050	0.8064	0.7834	0.8131
	±0.0026	±0.0020	±0.0135	±0.0014	±0.0037	±0.0038	±0.0011

表 5.9 在过滤数据集上不同算法之间的比较三元表

A \ B	Proposed	MLkNN	BPMLL	IBLR	RAkEL	CC	ECC	In total
Proposed	0/7/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	42/0/ 0
MLkNN	0/0/7	0/7/0	7/0/0	0/3/4	6/0/1	7/0/0	0/4/3	20/7/15
BPMLL	0/0/7	0/0/7	0/7/0	0/0/7	0/0/7	0/0/7	0/0/7	0/0/42
IBLR	0/0/7	4/3/0	7/0/0	0/7/0	6/1/0	7/0/0	0/5/2	24/9/9
RAkEL	0/0/7	1/0/6	7/0/0	0/1/6	0/7/0	7/0/0	0/0/7	15/1/26
CC	0/0/7	0/0/7	7/0/0	0/0/7	0/0/7	0/7/0	0/0/7	7/0/35
ECC	0/0/7	3/4/0	7/0/0	2/5/0	7/0/0	7/0/0	0/7/0	26/9/7

据我们所知，目前绝大部分的抗菌肽预测器都是用来判断肽分子是否属于抗菌肽，只有 2013 年提出的一个工具 iAMP-2L 能够实现抗菌肽活性的多标记预测。事实上 iAMP-2L 是个两层的预测器，第一层用来判断肽分子是否属于抗菌肽，如果是，则第二层预测器将预测出抗菌肽分子有哪些活性。由于本研究致力于抗菌肽活性的多标记学习问题，因此将只与 iAMP-2L 的第二层预测器进行比较。由于 iAMP-2L 中用到的多标记学习方法和特征提取方法都与我们不同，公平起见，在本文构建的抗菌肽数据集上进行试验时，我们将分别比较两种多标记学习方法和两种特征提取方法的优劣，因此将会有 4 组实验结果。进行 10 次 5 折交叉验证的实验结果如表 5.10 所示。显然，本文用到的多标记学习方法和特征提取方法是最佳组合。采用同样的特征提取方法时，本文提出的多标记学习方法要优于 iAMP-2L 中使用的方法。而当采用同样的多标记学习方法时，本文采用的特征提取方法要优于 iAMP-2L 中使用的 PseAAC (pseudo amino acid composition)。可能是由于新构建的 AMP 数据集中的序列长度最短为 2，因而只能够使用一阶的关联因子，这使得 PseAAC 的性能被抑制。

接下来我们又在过滤数据集上进行了实验。由于过滤数据集中的序列长度最短为 10，因此可以提取出更高阶的关联因子。正如 iAMP-2L 所做的，PseAAC 中使用 5 种物理化学属性，同时关联因子的阶数也将从 2 逐渐增大到 8，步进值 2，并选择出最佳的阶数为 4，因而每个抗菌肽分子将被转化为 40 维的特征向量（20 维的氨基酸成分和 20 维的关联因子）。使用本文提出的多标记学习方法，并分别采用两种特征提取方法，在过滤

数据集上进行 10 次的 5 折交叉验证，结果如表 5.11 所示。看起来本文所采用的特征提取方法要略好于 PseAAC，但是 PseAAC 的维度低得多。如果能找到与抗菌肽的活性关联性更强的物理化学属性，有希望构造出更好的 PseAAC。

表 5.10 本文所用方法和 iAMP-2L 在原始数据集上的五折交叉验证结果

a 表示氨基酸成分加二联体成分，b 表示伪氨基酸成分

Metric \ Method	Proposed[a]	Proposed[b]	iAMP-2L[a]	iAMP-2L[b]
Hamming Loss ↓	**0.0454±0.0004**	0.0483±0.0005	0.0580±0.0003	0.0581±0.0007
Subset Accuracy ↑	**0.5988±0.0049**	0.5733±0.0040	0.4880±0.0041	0.4848±0.0043
Average Precision ↑	**0.9439±0.0011**	0.9383±0.0012	0.9361±0.0010	0.9353±0.0015
Coverage ↓	**0.9337±0.0104**	0.9816±0.0096	1.1006±0.0116	1.1121±0.0161
OneError ↓	**0.0607±0.0018**	0.0689±0.0018	0.0658±0.0013	0.0682±0.0025
Ranking Loss ↓	**0.0234±0.0005**	0.0259±0.0005	0.0385±0.0006	0.0400±0.0010
Fmicro ↑	**0.8082±0.0015**	0.7955±0.0021	0.7852±0.0010	0.7851±0.0026

表 5.11 本文所用方法和 iAMP-2L 在过滤数据集上的五折交叉验证结果

a 表示氨基酸成分加二联体成分，b 表示伪氨基酸成分

Metric \ Method	Proposed[a]	Proposed[b]	iAMP-2L[a]	iAMP-2L[b]
Hamming Loss ↓	**0.0992±0.0014**	0.1018±0.0012	0.1221 0.0020	0.1212±0.0023
Subset Accuracy ↑	**0.6141±0.0056**	0.6033±0.0041	0.5149 0.0063	0.5228±0.0078
Average Precision ↑	**0.9553±0.0010**	0.9534±0.0010	0.9526 0.0014	0.9527±0.0016
Coverage ↓	**0.6899±0.0054**	0.6946±0.0034	0.6911 0.0055	0.6953±0.0064
OneError ↓	**0.0565±0.0021**	0.0615±0.0019	0.0652 0.0022	0.0638±0.0028
Ranking Loss ↓	**0.0444±0.0010**	0.0457±0.0007	0.0494 0.0014	0.0498±0.0015
Fmicro ↑	**0.8226±0.0026**	0.8176±0.0023	0.8083 0.0028	0.8091±0.0033

5.6 本章小结

目前已经存在很多的生物信息学工具用于抗菌肽识别，他们中的一些可以获得超过90%的测试准确率。当我们通过这些工具识别出具有高抗菌性的肽分子时，进一步想知道它可能具有的活性类型。由于一个抗菌肽分子可能同时具有多种活性，因此抗菌肽活性预测属于机器学习领域的多标记学习问题，目前关于这方面的研究还非常少。在本章的研究中，我们建立了一个覆盖 12 种活性的新的抗菌肽数据集，同时还提供了一个只覆盖 5 种活性的过滤数据集。在特征提取阶段，我们首先分析了抗菌肽分子的序列信息和活性信息，并确定用氨基酸成分和二联体成分来将长度不一的非结构化的肽序列转化为长度一致的结构化的词频特征向量，这样就非常方便利用机器学习方法来分析和处理。我们在新构建的数据集上测试了 8 个多标记学习算法，据我们所知，这是首次在进行抗菌肽活性预测时使用如此多的学习方法。在所评估的 8 种多标记学习方法中，我们所提出的算法能有效将标记之间的关联性考虑进模型，并取得了最好的预测性能。最后，我

们又比较了氨基酸成分加二联体成分的特征提取方法与伪氨基酸成分的性能，实验表明，前者具有较好的性能，但是后者具有低维度的优势。我们相信如果能找到更合适的物理化学属性将会构造出低维却性能优异的伪氨基酸成分出来。

第六章 多标记学习算法在慢病预测中的应用

6.1 引言

慢性病（chronic diseases）也称为非传染性疾病（noncommunicable diseases，NCDs），它的显著特点是病情持续时间长、发展缓慢。最为流行的慢性病包括心血管疾病（如高血压、冠心病和脑卒中）、慢性呼吸道疾病（如慢性阻塞性肺部疾病）以及糖尿病。慢性病的持续时间长、医护成本高昂，给社会和家庭带来了巨大的经济负担[154]。2015 年国家卫计委发布的《中国疾病预防控制工作进展报告》称，虽然我国已逐步加大了慢性病防控工作，但是慢性病的防治形势依然严峻，慢性病导致的疾病负担占总负担的约 70%，而导致的死亡人数已占到全国总死亡的 86.6%[155; 156]。

尽管医疗技术的进步（如可穿戴医疗[157]、移动健康[158]等）使得可以用更连续的方式来监控病人的状况，但是由于慢性病的成因复杂且通常会发展成多种并发症，采取一个合适的治疗方案对医生来说并不是一件很容易的事。对病人的连续监控会产生大量的数据，且通常这些数据都是异构的，如实验室检验值、体检值或者心电图。医生要做出最优的决策就需要将这些异构数据进行汇总分析，而医生往往又要同时负责很多的病人，这就需要一些数据挖掘或者机器学习工具来辅助医生进行处理和决策[159-161]。开发这样的工具面临着诸多挑战。第一个挑战是选择什么类型的建模方法。我们知道，慢性病人如糖尿病患者除了自身的主要病症以外往往会有多种并发症，对于这样的问题，可以使用机器学习领域的多标记学习算法来进行建模。第二个主要挑战是如何对医学信息进行特征描述。目前已经有一些较为成功的特征提取方法如 k-means clustering、Bag-of-Words(BoW)或者是直接提取统计量[98; 99]。

6.2 慢性病数据集

贝斯以色列女执事医疗中心（Beth Israel Deaconess Medical Center， BIDMC）是国际知名的医疗中心，也是哈佛医学院主要的教学医院之一。本研究所采用的疾病数据来源于该中心发布的 MIMIC-II（https://mimic.physionet.org/）[162]。MIMIC 数据库是一个公开数据库，经过注册之后即可免费获取。这些数据从 2001 年开始收集，前后持续了 7 年时间。数据库中包含了大约 33000 个病人，去掉新生儿和儿童后的成人群体（≥16 岁）包含大约 24000 个病人，再去掉没有慢性病的个体，最终剩下 19733 个病人，他们的平均年龄是 67 岁，男性和女性的比例分别为 56%和 44%。

临床数据包含实验室的检查项和病历上的记录项。每个病人的病历一般都记录有多相检查的结果，如体液检查、生理测量和一些重要功能的评分。有些重要信息如年龄和

性别也都记录在病历上。一个病人可能要做多次的实验室检验或者各种各样的检查，因此，病人的临床数据属于时间序列。为了避免缺失值过多，在实验室检查项和病历记录项中只选择那些至少有 80%的病人都包含的条目，这样的条目只有 76 个，其中 39 个属于定量型属性（如表 6.1 所示），37 个属于类别型属性（如表 6.2 所示）。

从数据库的文档可以发现，数据库中的数据缺失并不是随机的，在这里应用数据插值或填补技术可能不太合理。目前已有一些方法可以用来处理医学数据的缺失值问题[163]。一个最简单的方法是直接用平均值来替代缺失值，但是这种方式在大多数情况下并不为人所接受[164]。一个更好的方式是根据医学知识来从一个合理范围中选择一个值[164]，如本章所采用的慢性病数据集中的缺失值问题就是这么解决的[98]。

表 6.1 慢性病数据集中的定量型属性

Age at the admission (years)
Height at the admission (in)
Weight at the admission (kg)
Body surface area at the admission (m^2)
Heart rate (BPM)
Blood pressure systolic (mmHg)
Blood pressure diastolic (mmHg)
Respiratory rate (BPM)
Saturation of peripheral oxygen (%)
Temperature (deg C)
Hematocrit [volume fraction] of blood (%)
Platelets [#/volume] in blood (K/μL)
Leukocytes [#/volume] in blood (K/μL)
Hemoglobin [mass/volume] in blood (g/dL)
Erythrocyte mean corpuscular volume [entitic volume] (fL)
Erythrocytes [#/volume] in blood (m/μL)
Erythrocyte mean corpuscular hemoglobin concentration [mass/volume] (%)
Erythrocyte mean corpuscular hemoglobin [entitic mass] (pg)
Erythrocyte distribution width [ratio] (%)
Urea nitrogen [mass/volume] in serum or plasma (mg/dL)
Creatinine [mass/volume] in serum or plasma (mg/dL)
Potassium [moles/volume] in serum or plasma (mEq/L)
Sodium [moles/volume] in serum or plasma (mEq/L)
Chloride [moles/volume] in blood (mEq/L)
Bicarbonate [moles/volume] in serum (mEq/L)
Anion gap in blood (mEq/L)
Glucose [mass/volume] in serum or plasma (mg/dL)
Magnesium [mass/volume] in serum or plasma (mg/dL)
INR in blood by coagulation assay
Prothromb in time (PT) in blood by coagulation assay (s)
Activated partial thrombplastin time (aPTT) in blood by coagulation assay (s)
Phosphate [mass/volume] in serum or plasma (mg/dL)
Calcium [mass/volume] in serum or plasma (mg/dL)

pH of urine (units)
Urobilinogen [mass/volume] in urine (mg/dL)
Ketones [mass/volume] in urine (mg/dL)
Specific gravity of urine by test strip
Protein [mass/volume] in urine by test strip (mg/dL)
Glucose [mass/volume] in urine (mg/dL)

表 6.2 慢性病数据集中的类别型属性

Gender	Male, Female
Marital status	Single/Divorced/Widowed, Married
Heart rhythm	Normal sinus, Abnormal sinus (such as arrhythmia)
Ectopic heartbeat	No, Yes (PAC, PNC, PVC)
Level of conscious	Alert, Arouse to pain/stimuli/voice, Unresponsive
Eye opening (Glasgow coma scale)	Spontaneously, To speech/pain, No response
Verbal response (Glasgow coma scale)	Oriented, Confused/Inappropriate, No response
Motor response (Glasgow coma scale)	Obeys commands, Localize spain/Flex-withdraws, No response
Respiratory pattern	Regular, Irregular
Cardiovascular SOFA score (low is better)	0, 1, 2, 3 or 4
Hematologic SOFA score (low is better)	0, 1, 2, 3 or 4
Neurological SOFA score (low is better)	0, 1, 2, 3 or 4
Renal SOFA score (low is better)	0, 1, 2, 3 or 4
LUL lung sounds	Clear, Not clear
LLL lung sounds	Clear, Not clear
RUL lung sounds	Clear, Not clear
RLL lung sounds	Clear, Not clear
Skin integrity	Intact, Impaired
Bowel sounds	Present, Absent
Activity (Braden scale)	Walks frequently/occasionally, Chairfast/Bedfast
Moisture (Braden scale)	Rarely moist, Occasionally moist, Very moist
Mobility (Braden scale)	No limitation, Slightly limited, Very limited
Sensory perception (Braden scale)	No impairment, Slightly limited, Very limited
Nutrition (Braden scale)	Excellent/Adequate, Probably inadequate, Very poor
Friction and shear (Braden scale)	No apparent problem, Potential problem, Problem
Assistance device	Independent, Supervised/Assisted
Urine color	(Light) yellow, Not yellow
Urine appear	Clear, Sediment/Cloudy
Intravenous site appear	Within normal range, Outside normal range
Pain present	No, Yes
Contact precautions	No, Yes
Sedation-agitation (Riker scale)	Calme/Cooperative, Agitated, Sedated
Restraint location	No, Yes
Nitrite [presence] in urine by test strip	Negative, Positive
Bilirubin [presence] in urine	Negative, Positive
Hemoglobin [presence] in urine by test strip	Negative, Positive
Leukocytes [presence] in urine	Negative, Positive

这里共考虑了 10 种大的慢性病类型，这些疾病的名称、ICD-9 编码以及 19733 个病人在这些疾病中的分布情况如表 6.3 所示[98]。其实在这里根据 ICD-9 编码提供一个明确的诊断是不现实的。ICD-9 编码系统主要用于发病率统计和医疗保险，用它来为某个诊断定义一个精确的疾病标记还是不够完善的。文献【】考虑了医学的相关性、数据集的特性以及 ICD-9 编码系统的层次结构后定义了这里的 10 种大的慢性病家族类型。10 种标记共有 1023 种可能的标记组合，其中在该慢性病数据集中实际出现了 522 种。该数据集的标记势（label cardinality）为 2.37，标记密度（label density）为 0.237。

表 6.3 慢性病数据集中具有不同慢性病的病人数目、百分比及其 ICD-9 编码

疾病中文名	疾病英文名	病人数目	百分比	ICD-9 编码
高血压	Hypertensive disease	12309	62.3	[401–405]
体液疾病	Fluid electrolyte disease	6177	31.2	276
糖尿病	Diabetes mellitus	6056	30.6	[249–250]
类脂代谢疾病	Lipoid metabolism disease	5965	30.2	272
肾病	Kidney disease	5828	29.5	[580–589]
慢性阻塞性肺病	COPD	4253	21.5	[490–496]
甲状腺疾病	Thyroid disease	2246	11.4	[240–246]
低血压	Hypotension	1962	9.9	458
肝病	Liver disease	1088	5.5	571
血栓症	Thrombosis	931	4.7	[451–453]

6.3 特征提取与标准化

病人的实验室检验项和病历记录项将被整合成一个特征向量。由于医学数据具有异质性，对不同类型的数据按照如下的方法来提取特征[98]：

对于数值型变量（测量值如血压、肌酸酐或者温度等），如果他们只出现一次（如身高），则直接放入特征向量中；如果出现了多次，则提取出如下几个特征：平均值、中位数、标准差和范围（最大-最小）。

对于类别型变量（观察值如心血管功能评估得分或尿液颜色等），如果一个病人的某项检查进行了多次且结果可以被划分成为互相排斥的几类，则用直方图来描述这个信息，并将直方图中每个类型的相对频率用作特征；如果对于一个病人，某项检查只会出现一个结果，例如性别，则将该属性用一个二进制变量进行编码。

根据上述规则从 76 个检查项中一共提取出 310 个特征。由于各特征之间的尺度差异非常大，这对有些基于距离的算法如 MLkNN 的影响非常大，因此在输入到多标记学习模型中之前还需要做一个标准化处理，这里通过计算 z-score 的方法进行标准化。

6.4 多标记学习算法比较

由于我们这里采用的慢性病数据集比较大（共包含 19773 个样本），这就允许我们将其随机划分成三个大小相同的子集（每个子集包含 6591 个样本），一个用于模型训练，一个用于参数调节和模型选择，最后一个用于结果展示和模型比较（如下图所示）。这 3 个子集的用途可以排列成 6 种情况，并取 6 次结果的平均值作为模型的最终结果[98]。

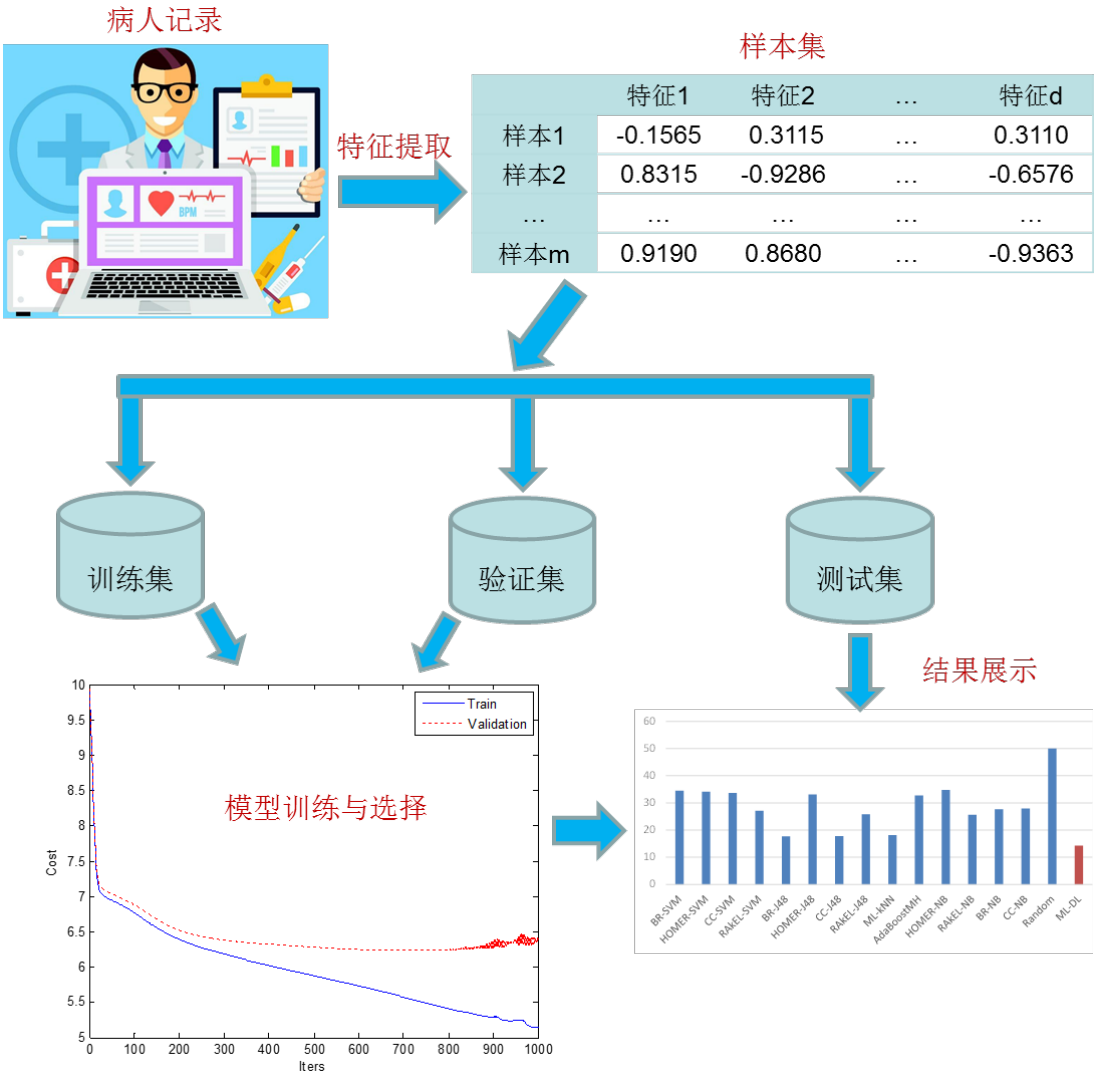


图 6.1 慢性病预测模型测试流程

由于该数据集比较大，适合于采用深度网络建模。且我们提出的多标记深度网络模型如果不进行逐层预训练的话非常高效，在一个普通的个人电脑上即可快速完成模型的训练和预测任务。由于这里要预测的是疾病类型，而通常我们想知道的就是患某种疾病的概率，因此这里适合采用 Sigmoid 输出层（对一个病人样本，输出每种疾病的概率值），相应的损失函数为多标记交叉熵。多标记深度网络中的超参数，如隐含层数、隐含层结点数、学习率、正则化参数等都将通过模型在验证集的表现来确定（具体如第 4 章所述）。

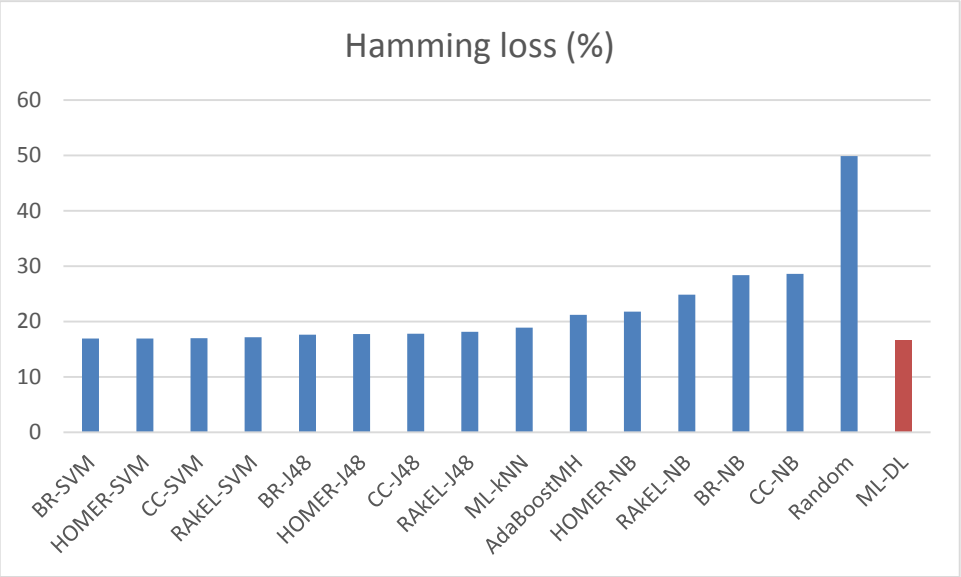
要比较的其他多标记学习算法有 BR、HOMER、CC、RAkEL、ML-kNN 和 AdaBoostMH，而 BR、HOMER、CC、RAkEL 都属于问题转换型多标记学习算法，需要调用单标记学习算法，这里采用 3 种非常经典的方法如 SVM（支持向量机）、J48（一种决策树算法的实现）和 NB（朴素贝叶斯）。因此，实际上这里要比较的多标记学习算法共有 14 种。这些算法的参数都通过网格搜索的方式寻优。其中 AdaBoostMH、BR 和 CC 都没有额外的参数；ML-kNN 有两个参数，分别是近邻数目和平滑因子，对这两个参数的搜索范围分别为[1:50]和[1:10]；RAkEL 有两个参数，一个是集成模型的数目（NumberOfModels），另一个是模型中标记子集的大小（SizeOfSubset），这里将采用作者推荐的默认参数值，即 NumberOfModels = min(2*|L|, 100)，SizeOfSubset = |L|/2；HOMER 要用到 K 均值聚类算法，且只有一个超参数——聚类的数目，对其搜素范围为[2:9]。要调用的基分类器的参数也是需要调优的，其中 NB 没有参数；带有 RBF 核的 SVM 有两个参数，分别是 RBF 核的系数和误差项的惩罚权重，对他们的搜索范围为[0.001:10]和[1:50]；J48 也有两个参数，分别是剪枝的置信度阈值和每个叶子结点的最少样本数，对他们的搜索范围是[0.0001:0.1]和[1:50]。

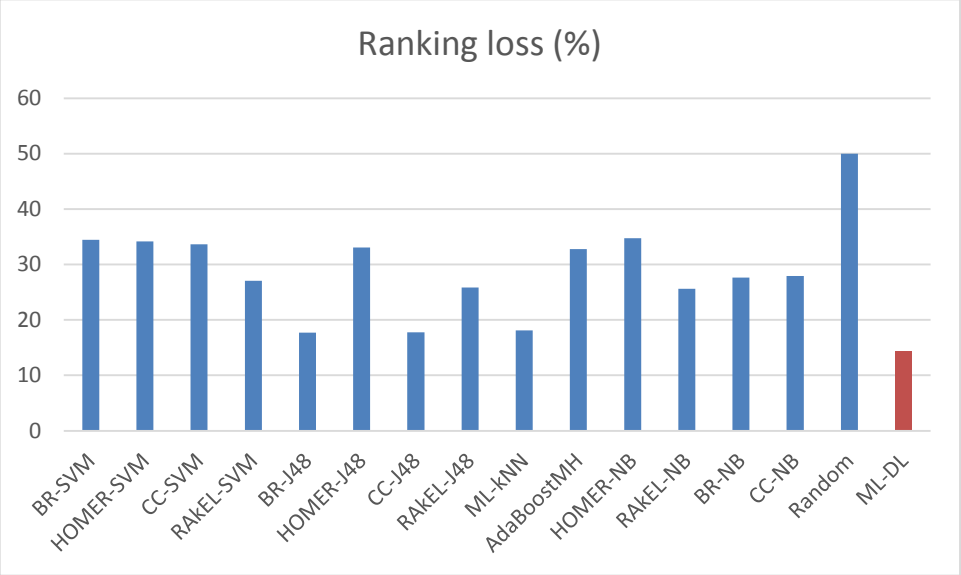
各种多标记学习算法在参数寻优之后的最佳模型对测试集进行实验的结果如表 6.4 和图 6.2 所示。其他方法的实验结果均来自文献[98]。此外，表或图中的 Random 表示不使用任何算法而只对测试样本进行随机猜测的结果，它表示模型性能指标的下限或者上限，任何比随机猜测的结果还要差的算法不具有任何价值[98]。从表和图中的结果来看，我们提出的多标记深度网络模型在各项指标上都要优于传统的多标记学习算法。就平均结果而言，虽然 Hamming loss 的结果相比排在第二位的 BR-SVM 优势不是很大（降低约 2%），但是 Ranking loss 的结果比排在第二位的 BR-J48 降低了约 19%，One-error 结果比排在第二位的 RAkEL-SVM 降低了约 12%，Coverage 的结果比排在第二位的 BR-J48 降低了约 11%，Average precision 的结果比排在第二位的 CC-J48 提高了约 6%。

表 6.4 各种多标记学习算法的独立测试结果

Metric / Method	Hamming loss (%)↓	Ranking loss (%)↓	Average precision (%)↑	One-error (%)↓	Coverage↓
BR-SVM	16.94±0.12	34.47±0.64	61.85±0.56	35.17±0.67	5.35±0.05
HOMER-SVM	16.97±0.11	34.18±0.61	62.01±0.56	35.34±0.74	5.33±0.05
CC-SVM	17.01±0.14	33.67±0.70	62.58±0.59	35.11±0.80	5.28±0.06
RAkEL-SVM	17.18±0.07	27.08±0.45	68.32±0.54	30.68±0.81	4.72±0.04
BR-J48	17.63±0.13	17.70±0.31	72.23±0.55	31.15±0.70	3.50±0.01
HOMER-J48	17.75±0.17	33.06±1.05	62.45±0.89	34.74±0.87	5.31±0.10
CC-J48	17.83±0.15	17.76±0.40	72.14±0.79	31.48±1.32	3.50±0.02
RAkEL-J48	18.17±0.09	25.88±1.32	68.53±0.99	32.28±0.66	4.57±0.13
ML-kNN	18.91±0.16	18.10±0.17	71.37±0.22	32.05±0.36	3.52±0.02
AdaBoostMH	21.23±0.09	32.81±0.13	57.65±0.19	37.75±0.29	4.89±0.01
HOMER-NB	21.78±0.30	34.76±0.47	57.61±0.99	44.35±2.89	5.29±0.03
RAkEL-NB	24.88±0.40	25.63±0.39	64.07±0.76	45.93±2.19	4.37±0.03

BR-NB	28.40±0.34	27.62±0.50	60.32±0.70	54.80±1.37	4.42±0.03
CC-NB	28.61±0.36	27.92±0.51	59.96±0.71	55.55±1.40	4.44±0.03
Random	49.89±0.19	49.99±0.43	40.13±0.40	76.03±0.54	6.30±0.03
ML-DL	**16.59±0.09**	**14.30±0.27**	**76.25±0.46**	**26.74±0.73**	**3.12±0.01**





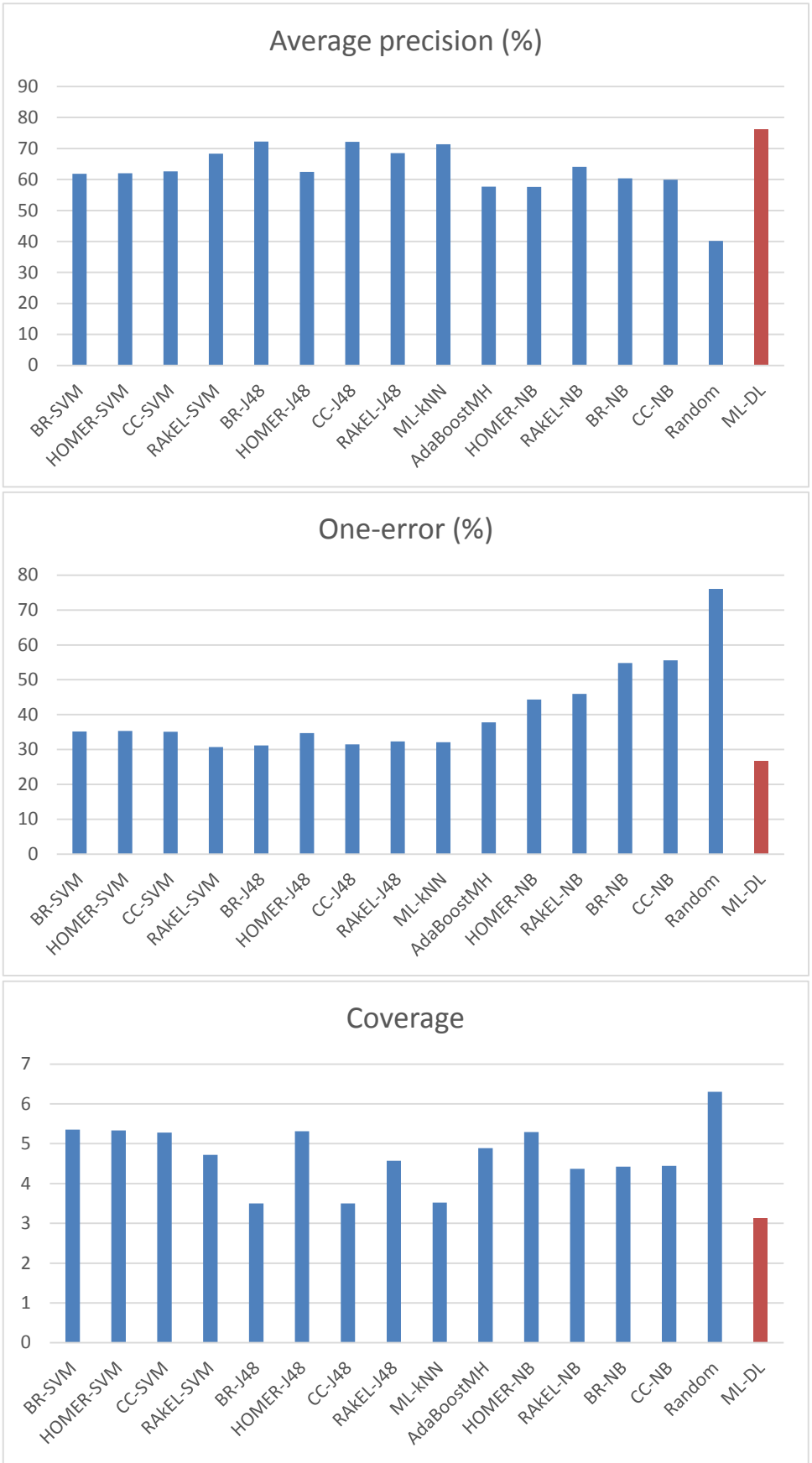


图 6.2 不同多标记学习算法的独立测试结果比较

除了预测准确率之外，我们也比较了各种多标记学习算法的训练时间（精确到分钟），如下表所示。其中 MLDL 是在 Matlab R2013b 环境下测试得到的（使用的是个人电脑，Intel Core i3@3.40GHz, 4GB RAM），而其他方法是由 Damien Zufferey 等人在 Mulan 1.4 环境中测试获得（Intel Core i7@2.93GHz，16GB RAM）[98]。由于这里使用的慢性病数据集比较大，以 SVM 作为基分类器的方法都非常慢，多达几个小时，尤其是与 RAkEL 相结合时，这是因为 RAkEL 采用了集成学习策略。ML-kNN 是一种基于近邻搜索的方法，对数据规模的可扩展性比较差，所以在这里的运行时间也比较长。而其他方法的训练时间都在分钟级别，还是比较快的。MLDL 尽管采用了 4 层的深度网络结构，但是得力于 ReLU 激活函数的高效性，经过 1000 次迭代的训练时间只需要 7 分钟，这为网络结构优化和超参调节带来很大的方便。我们知道，通常 Matlab 语言的运行效率没有 Java 高，Java 版本的 MLDL 将会进一步缩短训练时间。

表 6.5 不同多标记学习算法的训练时间

多标记学习算法	训练时间
BR-NB	1min
CC-NB	1min
HOMER-NB	1min
BR-J48	4min
CC-J48	4min
HOMER-J48	4min
AdaBoostMH	6min
RAkEL-NB	7min
RAkEL-J48	15min
ML-kNN	35min
BR-SVM	3h32min
CC-SVM	3h33min
HOMER-SVM	4h11min
RAkEL-SVM	28h13min
MLDL	7min

6.5 本章小结

本章的主要工作是在多标记学习框架下进行慢性病预测建模。本章所采用的慢性病数据来源于 MIMIC-II，由于数据集包含的样本比较多，对算法的可扩展性提出了挑战，尤其是当模型有较多的超参数需要调节时。通过在该慢性病数据集上对总计十五种多标记学习算法进行的评估发现，我们提出的多标记深度学习算法要优于传统的方法，同时在运行效率和可扩展性方面也具有很强的优势。

第七章 总结与展望

7.1 本文工作总结

研究对象具有多语义或多属性的现象在真实世界中非常普遍，尤其是在生物医学领域，例如一个病人可能同时具有多种并发症，而一个基因或蛋白质也往往拥有不止一个功能。因此，作为机器学习的一个分支，多标记学习方法具有重要的研究意义和研究价值。本文立足于人工智能和机器学习的研究现状，力图将最新的理论和方法与多标记学习结合起来，并围绕多标记学习的算法创新和应用创新开展一系列的研究，具体如下：

（1）对多标记学习的研究现状和应用现状做了广泛的调研，对已有的多标记学习算法进行了概括和总结，并对其中几种典型方法进行了图表化的解构。

（2）在集成学习的理论指导下设计一种混合多标记学习方法，该方法是一种异质的集成学习器，性能比任何构成的基学习器都要强大，同时在与已有的多标记学习算法比较中也表现出了很大的优势。

（3）结合深度学习和多标记学习的特点，设计了一种多标记深度网络架构，它具有多层次的结构，隐含层采用最流行的 ReLU 激活函数，而同时由于采用了多标记损失函数，又能满足多标记输出的要求。本文对提出的多标记深度网络的参数优化步骤进行了详细的推导。此外，还比较了多标记交叉熵与对数损失之间的关系。

（4）基于多标记学习方法对抗菌肽活性预测问题进行了建模，包括数据集的构建、抗菌肽分子的特征提取和多种学习算法的比较。

（5）基于多标记学习方法对慢性病预测问题进行了建模，对数据集的创建、属性缺失值处理、特征提取与标准化和 15 种多标记学习算法的实验结果进行了详细的讨论。

综上所述，本文以多标记学习为中心，顺利开展了一系列的研究工作，初步取得了一些研究成果，完成了开题和中期时制定的研究目标。

7.2 下一步研究方向

集成学习与深度学习博大精深，且一直处于不断的发展当中，如何基于他们设计出更好多标记学习算法值得深入的研究。特征工程是机器学习系统的基础，好的特征将显著提升学习系统的性能，并降低对后续学习算法的依赖性。在这些方面，本文只是做了初步的探索，下一步还有很多的工作可做：

（1）多标记学习算法的有选择集成：本文的混合多标记学习器只采用了两个异质的基学习器进行集成，学习能力有限。下一步考虑增强基学习器的多样性，可以通过数据、属性和参数的扰动来实现。学习器并不是越多越好，如何找出基学习器的最优组合也是

一个很大的挑战，下一步将考虑对基学习器进行有选择的集成。

（2）深度多标记学习算法的提升：将多标记损失函数与排序损失函数结合起来优化网络结构；引入 droupout 机制，增强网络的泛化能力和集成能力。

（3）将字典学习、word2vec 等当前最流行的文本特征提取方法应用到生物医学数据挖掘中，增强对研究对象的描述能力。

参考文献

[1] 宁康, 陈挺. 生物医学大数据的现状与展望[J]. 科学通报, 2015, (Z1): 534-546.
[2] 李春英, 张巍巍. 全球大数据与健康管理的研究热点聚类分析[J]. 中国医院管理, 2016, (10): 63-65.
[3] 人工智能在各领域的应用[J]. 智能机器人, 2016, (12): 28-31.
[4] 韩晔彤. 人工智能技术发展及应用研究综述[J]. 电子制作, 2016, (12): 95.
[5] 胡虎 本: 人工智能刷屏凶猛 各行业“AI+”应用加速落地[N], 2017, 2017-03-13.
[6] 周志华. 机器学习[M]. 清华大学出版社, 2016.
[7] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
[8] Schmidhuber J. Deep learning in neural networks: an overview[J]. Neural Networks, 2014, 61: 85.
[9] Deng L, Yu D. Deep Learning: Methods and Applications[J]. Foundations & Trends® in Signal Processing, 2013, 7(3): 197-387.
[10] Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(8): 1798-828.
[11] Bengio Y. Learning Deep Architectures for AI[J]. Foundations & Trends® in Machine Learning, 2009, 2(1): 1-55.
[12] 冯雪东. 多标签分类问题综述[J]. 信息系统工程, 2016, (03): 137.
[13] 余鹰. 多标记学习研究综述[J]. 计算机工程与应用, 2015, (17): 20-27.
[14] 李志欣, 卓亚琦, 张灿龙, et al. 多标记学习研究综述[J]. 计算机应用研究, 2014, (06): 1601-1605.
[15] Goertzel B, Pennachin C. Artificial general intelligence[M]. 2. Springer, 2007.
[16] 艾伦·达福, 斯图尔特·罗素. 人工智能的真正风险[J]. 中国经济报告, 2017, (02): 118-119.
[17] 朱巍, 陈慧慧, 田思媛, et al. 人工智能:从科学梦到新蓝海——人工智能产业发展分析及对策[J]. 科技进步与对策, 2016, (21): 66-70.
[18] 王沛霖. 2017,人工智能大爆发[J]. 机器人产业, 2017, (01): 2.
[19] Zhou Z-H, Feng J. Deep Forest: Towards An Alternative to Deep Neural Networks[J], 2017.
[20] 尹昊智, 刘铁志. 人工智能各国战略解读:美国人工智能报告解析[J]. 电信网技术, 2017, (02): 52-57.
[21] 田丰, 任海霞, Gerbert P, et al. 人工智能:未来制胜之道[J]. 机器人产业, 2017, (01): 76-87.
[22] 张洛阳, 毛嘉莉, 刘斌, et al. 基于贝叶斯模型的多标签分类算法[J]. 计算机应用, 2016, (01): 52-56+71.
[23] 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, (10): 1-6.
[24] 梁新彦, 钱宇华, 郭倩, et al. 面向多标记学习的局部粗糙集[J]. 南京大学学报(自然科学), 2016, (02): 270-279.
[25] Du B, Wang Z M, Zhang L F, et al. Robust and Discriminative Labeling for Multi-Label

Active Learning Based on Maximum Correntropy Criterion[J]. Ieee Transactions on Image Processing, 2017, 26(4): 1694-1707.
[26] Xia S, Chen P, Zhang J, et al. Utilization of rotation-invariant uniform LBP histogram distribution and statistics of connected regions in automatic image annotation based on multi-label learning[J]. Neurocomputing, 2017, 228: 11-18.
[27] Yu Z L, Hao H, Zhang W P, et al. A Classifier Chain Algorithm with K-means for Multi-label Classification on Clouds[J]. Journal of Signal Processing Systems for Signal Image and Video Technology, 2017, 86(2-3): 337-346.
[28] Jesse R, Martino L, Hollmen J. Multi-label methods for prediction with sequential data[J]. Pattern Recognition, 2017, 63: 45-55.
[29] Jia X, Sun F M, Li H J, et al. Image multi-label annotation based on supervised nonnegative matrix factorization with new matching measurement[J]. Neurocomputing, 2017, 219: 518-525.
[30] Yuan T, Wang J H. Reduced-rank multi-label classification[J]. Statistics and Computing, 2017, 27(1): 181-191.
[31] Wan S B, Mak M W, Kung S Y. Transductive Learning for Multi-Label Protein Subchloroplast Localization Prediction[J]. Ieee-Acm Transactions on Computational Biology and Bioinformatics, 2017, 14(1): 212-224.
[32] Tabatabaei S M, Dick S, Xu W S. Toward Non-Intrusive Load Monitoring via Multi-Label Classification[J]. Ieee Transactions on Smart Grid, 2017, 8(1): 26-40.
[33] Pillai I, Fumera G, Roli F. Designing multi-label classifiers that maximize F measures: State of the art[J]. Pattern Recognition, 2017, 61: 394-404.
[34] Lin Y, Guo F, Cao L J, et al. Person re-identification based on multi-instance multi-label learning[J]. Neurocomputing, 2016, 217: 19-26.
[35] Wu Q H, Liu H M, Yan X S. Multi-label classification algorithm research based on swarm intelligence[J]. Cluster Computing-The Journal of Networks Software Tools and Applications, 2016, 19(4): 2075-2085.
[36] Huang J, Li G R, Huang Q M, et al. Learning Label-Specific Features and Class-Dependent Labels for Multi-Label Classification[J]. Ieee Transactions on Knowledge and Data Engineering, 2016, 28(12): 3309-3323.
[37] Wang M, Luo C Z, Hong R C, et al. Beyond Object Proposals: Random Crop Pooling for Multi-Label Image Recognition[J]. Ieee Transactions on Image Processing, 2016, 26(12): 5678-5688.
[38] Li Y Q, Wu B Y, Ghanem B, et al. Facial action unit recognition under incomplete data based on multi-label learning with missing labels[J]. Pattern Recognition, 2016, 60: 890-900.
[39] Mei S, Zhang K. Multi-label l(2)-regularized logistic regression for predicting activation/inhibition relationships in human protein-protein interaction networks[J]. Scientific Reports, 2016, 6.
[40] Lee J, Kim D W. Efficient Multi-Label Feature Selection Using Entropy-Based Label Selection[J]. Entropy, 2016, 18(11).
[41] Mencia E L, Janssen F. Learning rules for multi-label classification: a stacking and a separate-and-conquer approach[J]. Machine Learning, 2016, 105(1): 77-126.
[42] Wu Q Y, Tan M K, Song H J, et al. ML-FOREST: A Multi-Label Tree Ensemble Method for Multi-Label Classification[J]. Ieee Transactions on Knowledge and Data Engineering,

2016, 28(10): 2665-2680.
[43] Chen W J, Shao Y H, Li C N, et al. MLTSVM: A novel twin support vector machine to multi-label learning[J]. Pattern Recognition, 2016, 52: 61-74.
[44] Xu J H. Multi-label Lagrangian support vector machine with random block coordinate descent method[J]. Information Sciences, 2016, 329: 184-205.
[45] Kanj S, Abdallah F, Denoeux T, et al. Editing training data for multi-label classification with the k-nearest neighbor rule[J]. Pattern Analysis and Applications, 2016, 19(1): 145-161.
[46] Zhang M L, Zhou Z H. A Review on Multi-Label Learning Algorithms[J]. Ieee Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819-1837.
[47] Tsoumakas G, Katakis I, Vlahavas I. Mining Multi-label Data[M]. 2009: 667-685.
[48] Boutell M R, Luo J B, Shen X P, et al. Learning multi-label scene classification[J]. Pattern Recognition, 2004, 37(9): 1757-1771.
[49] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification[J]. Machine Learning, 2011, 85(3): 333-359.
[50] Hullermeier E, Furnkranz J, Cheng W W, et al. Label ranking by learning pairwise preferences[J]. Artificial Intelligence, 2008, 172(16-17): 1897-1916.
[51] Furnkranz J, Hullermeier E, Mencia E L, et al. Multilabel classification via calibrated label ranking[J]. Machine Learning, 2008, 73(2): 133-153.
[52] Read J. A pruned problem transformation method for multi-label classification[C], 2008: 143--150.
[53] Tsoumakas G, Katakis I, Vlahavas I. Random k-Labelsets for Multilabel Classification[J]. Ieee Transactions on Knowledge and Data Engineering, 2011, 23(7): 1079-1089.
[54] Tsoumakas G, Vlahavas I. Random k-labelsets: An ensemble method for multilabel classification[J]. Machine Learning: Ecml 2007, Proceedings, 2007, 4701: 406-+.
[55] Zhang M L, Wu L. LIFT: Multi-Label Learning with Label-Specific Features[J]. Ieee Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(1): 107-120.
[56] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label leaming[J]. Pattern Recognition, 2007, 40(7): 2038-2048.
[57] Clare A, King R D. Knowledge Discovery in Multi-label Phenotype Data[J], 2002, 2168(2168): 42-53.
[58] Elisseeff A, Weston J. A kernel method for multi-labelled classification[J]. Advances in Neural Information Processing Systems 14, Vols 1 and 2, 2002, 14: 681-687.
[59] Zhang M L, Zhou Z H. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization[J]. IEEE Transactions on Knowledge & Data Engineering, 2006, 18(10): 1338-1351.
[60] Read J, Hollmén J. A deep interpretation of classifier chains[C]. International Symposium on Intelligent Data Analysis, 2014: 251-262.
[61] Rokach L, Schclar A, Itach E. Ensemble methods for multi-label classification[J]. Expert Systems with Applications, 2014, 41(16): 7507-7523.
[62] Kocev D, Vens C, Struyf J, et al. Ensembles of multi-objective decision trees[J]. Machine Learning: Ecml 2007, Proceedings, 2007, 4701: 624-+.
[63] Read J, Perezcruz F. Deep Learning for Multi-label Classification[J]. Machine Learning, 2014, 85(3): 333-359.
[64] Li J, Rao Y H, Jin F M, et al. Multi-label maximum entropy model for social emotion

classification over short text[J]. Neurocomputing, 2016, 210: 247-256.
[65] Zou F H, Liu Y, Wang H, et al. Multi-view multi-label learning for image annotation[J]. Multimedia Tools and Applications, 2016, 75(20): 12627-12644.
[66] Elghazel H, Aussem A, Gharroudi O, et al. Ensemble multi-label text categorization based on rotation forest and latent semantic indexing[J]. Expert Systems with Applications, 2016, 57: 1-11.
[67] Wei Y C, Xia W, Lin M, et al. HCP: A Flexible CNN Framework for Multi-Label Image Classification[J]. Ieee Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(9): 1901-1907.
[68] Ding X M, Li B, Xiong W H, et al. Multi-Instance Multi-Label Learning Combining Hierarchical Context and its Application to Image Annotation[J]. Ieee Transactions on Multimedia, 2016, 18(8): 1616-1627.
[69] Zhao K L, Chu W S, De La Torre F, et al. Joint Patch and Multi-label Learning for Facial Action Unit and Holistic Expression Recognition[J]. Ieee Transactions on Image Processing, 2016, 25(8): 3931-3946.
[70] Yan K B, Li Z X, Zhang C L. A New multi-instance multi-label learning approach for image and text classification[J]. Multimedia Tools and Applications, 2016, 75(13): 7875-7890.
[71] Al-Salemi B, Noah S a M, Ab Aziz M J. RFBoost: An improved multi-label boosting algorithm and its application to text categorisation[J]. Knowledge-Based Systems, 2016, 103: 104-117.
[72] Jing X Y, Wu F, Li Z Q, et al. Multi-Label Dictionary Learning for Image Annotation[J]. Ieee Transactions on Image Processing, 2016, 25(6): 2712-2725.
[73] Wang L, Ren F J, Miao D Q. Multi-label emotion recognition of weblog sentence based on Bayesian networks[J]. Ieej Transactions on Electrical and Electronic Engineering, 2016, 11(2): 178-184.
[74] Guo X T, Liu F L, Ju Y, et al. Human Protein Subcellular Localization with Integrated Source and Multi-label Ensemble Classifier[J]. Scientific Reports, 2016, 6.
[75] Wan S B, Mak M W, Kung S Y. Sparse regressions for predicting and interpreting subcellular localization of multi-label proteins[J]. Bmc Bioinformatics, 2016, 17.
[76] Wang X, Zhang W W, Zhang Q W, et al. MultiP-SChlo: multi-label protein subchloroplast localization prediction with Chou's pseudo amino acid composition and a novel multi-label classifier[J]. Bioinformatics, 2015, 31(16): 2639-2645.
[77] Wan S B, Mak M W, Kung S Y. mPLR-Loc: An adaptive decision multi-label classifier based on penalized logistic regression for protein subcellular localization prediction[J]. Analytical Biochemistry, 2015, 473: 14-27.
[78] Che Y X, Ju Y, Xuan P, et al. Identification of Multi-Functional Enzyme with Multi-Label Classifier[J]. Plos One, 2016, 11(4).
[79] De Ferrari L, Mitchell J B O. From sequence to enzyme mechanism using multi-label machine learning[J]. Bmc Bioinformatics, 2014, 15.
[80] Wan S B, Mak M W, Kung S Y. Mem-ADSVM: A two-layer multi-label predictor for identifying multi-functional types of membrane proteins[J]. Journal of Theoretical Biology, 2016, 398: 32-42.
[81] Xiao X, Zou H L, Lin W Z. iMem-Seq: A Multi-label Learning Classifier for Predicting Membrane Proteins Types[J]. Journal of Membrane Biology, 2015, 248(4): 745-752.

[82] Zou H L, Xiao X. A New Multi-label Classifier in Identifying the Functional Types of Human Membrane Proteins[J]. Journal of Membrane Biology, 2015, 248(2): 179-186.
[83] Wang Y L, Jing R Y, Hua Y P, et al. Classification of multi-family enzymes by multi-label machine learning and sequence-based descriptors[J]. Analytical Methods, 2014, 6(17): 6832-6840.
[84] Cheng X, Zhao S G, Xiao X, et al. iATC-mISF: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals[J]. Bioinformatics, 2017, 33(3): 341-346.
[85] Lin W, Ji D, Lu Y. Disorder recognition in clinical texts using multi-label structured SVM[J]. Bmc Bioinformatics, 2017, 18(1): 75.
[86] Mei S, Zhang K. Multi-label l2-regularized logistic regression for predicting activation/inhibition relationships in human protein-protein interaction networks[J]. Sci Rep, 2016, 6: 36453.
[87] Qiu W R, Zheng Q S, Sun B Q, et al. Multi-iPPseEvo: A Multi-label Classifier for Identifying Human Phosphorylated Proteins by Incorporating Evolutionary Information into Chou's General PseAAC via Grey System Theory[J]. Mol Inform, 2017, 36(3).
[88] Xu Y H, Min H Q, Song H J, et al. Multi-instance multi-label distance metric learning for genome-wide protein function prediction[J]. Computational Biology and Chemistry, 2016, 63: 30-40.
[89] Wu F, Liu Q, Hao T Y, et al. Online Multi-Instance Multi-Label Learning for Protein Function Prediction[J]. 2016 Ieee International Conference on Bioinformatics and Biomedicine (Bibm), 2016: 780-785.
[90] Wu J S, Huang S J, Zhou Z H. Genome-Wide Protein Function Prediction through Multi-Instance Multi-Label Learning[J]. Ieee-Acm Transactions on Computational Biology and Bioinformatics, 2014, 11(5): 891-902.
[91] Riemenschneider M, Senge R, Neumann U, et al. Exploiting HIV-1 protease and reverse transcriptase cross-resistance information for improved drug resistance prediction by means of multi-label classification[J]. Biodata Mining, 2016, 9.
[92] Zhang W, Liu F, Luo L Q, et al. Predicting drug side effects by multi-label learning and ensemble learning[J]. Bmc Bioinformatics, 2015, 16.
[93] Xu J, Xu Z X, Lu P, et al. Classifying syndromes in Chinese medicine using multi-label learning algorithm with relevant features for each label[J]. Chinese Journal of Integrative Medicine, 2016, 22(11): 867-871.
[94] Peng Y, Fang M, Wang C J, et al. Entropy Chain Multi-Label Classifiers for Traditional Medicine Diagnosing Parkinson's Disease[J]. Proceedings 2015 Ieee International Conference on Bioinformatics and Biomedicine, 2015: 856-862.
[95] Peng Y, Fang M, Wang C J, et al. Entropy Chain Multi-Label Classifiers for Traditional Medicine Diagnosing Parkinson's Disease[J]. Proceedings 2015 Ieee International Conference on Bioinformatics and Biomedicine, 2015: 1722-1724.
[96] Li G Z, He Z H, Shao F F, et al. Patient classification of hypertension in Traditional Chinese Medicine using multi-label learning techniques[J]. Bmc Medical Genomics, 2015, 8.
[97] Wang H Z, Liu X, Lv B, et al. Reliable Multi-Label Learning via Conformal Predictor and Random Forest for Syndrome Differentiation of Chronic Fatigue in Traditional Chinese Medicine[J]. Plos One, 2014, 9(6).
[98] Zufferey D, Hofer T, Hennebert J, et al. Performance comparison of multi-label learning

algorithms on clinical data for chronic diseases[J]. Computers in Biology and Medicine, 2015, 65: 34-43.
[99] Bromuri S, Zufferey D, Hennebert J, et al. Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms[J]. Journal of Biomedical Informatics, 2014, 51: 165-175.
[100] Chen W Z, Yan J, Zhang B Y, et al. Document transformation for multi-label feature selection in text categorization[J]. Icdm 2007: Proceedings of the Seventh Ieee International Conference on Data Mining, 2007: 451-+.
[101] Mencia E L, Furnkranz J. Pairwise Learning of Multilabel Classifications with Perceptrons[J]. 2008 Ieee International Joint Conference on Neural Networks, Vols 1-8, 2008: 2899-2906.
[102] Kazienko P, Lughofer E, Trawinski B. Hybrid and Ensemble Methods in Machine Learning[J]. Journal of Universal Computer Science, 2013, 19(4): 457-461.
[103] Dietterich T G. Ensemble methods in machine learning[J]. Multiple Classifier Systems, 2000, 1857: 1-15.
[104] Ghahramani Z. Unsupervised learning[J]. Advanced Lectures on Machine Learning, 2004, 3176: 72-112.
[105] Chechile R A. Unsupervised learning: Foundations of neural computation.[J]. Journal of Mathematical Psychology, 2000, 44(1): 235-236.
[106] Grira N, Crucianu M, Boujemaa N. Unsupervised and Semi-supervised Clustering: a Brief Survey[C]. 7th ACM SIGMM international workshop on Multimedia information retrieval, 2004: 125-135.
[107] Touretzky E D S, Sollich P, Krogh A. Learning with ensembles: How over-fitting can be useful[J]. Advances in Neural Information Processing Systems, 1997, 8: 190-196.
[108] Maclin R, Opitz D. Popular Ensemble Methods: An Empirical Study[J]. Journal of Artificial Intelligence Research, 2011, 11: 169-198.
[109] Schapire R E. The Strength of Weak Learnability[J]. Machine Learning, 1990, 5(2): 197-227.
[110] Breiman L. Bias, Variance , And Arcing Classifiers[J]. Additives for Polymers, 1996, 2002(6): 10.
[111] Zhou Z H. Ensemble Methods: Foundations and Algorithms[M]. Taylor & Francis, 2012: 77-79.
[112] Rojas R. AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting[J], 2009.
[113] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.
[114] Cover T M, Hart P E. Nearest Neighbor Pattern Classification[J]. Ieee Transactions on Information Theory, 1967, 13(1): 21-+.
[115] Macleod J E S, Luk A, Titterington D M. A Reexamination of the Distance-Weighted K-Nearest Neighbor Classification Rule[J]. Ieee Transactions on Systems Man and Cybernetics, 1987, 17(4): 689-696.
[116] Dudani S A. The Distance-Weighted k-Nearest-Neighbor Rule[J]. IEEE Transactions on Systems Man & Cybernetics, 1976, SMC-6(4): 325-327.
[117] Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, et al. MULAN: A Java Library for Multi-Label Learning[J]. Journal of Machine Learning Research, 2011, 12: 2411-2414.

[118] Diplaris S, Tsoumakas G, Mitkas P A, et al. Protein classification with multiple algorithms[C]. Panhellenic Conference on Advances in Informatics, 2005: 448-456.
[119] Pestian J P, Brew C, Matykiewicz P, et al. A shared task involving multi-label classification of clinical free text[C]. The Workshop on Bionlp 2007: Biological, Translational, and Clinical Language Processing, 2007: 97-104.
[120] Cheng W W, Hullermeier E. Combining instance-based learning and logistic regression for multilabel classification[J]. Machine Learning, 2009, 76(2-3): 211-225.
[121] Hinton G E. A Practical Guide to Training Restricted Boltzmann Machines[J]. Momentum, 2012, 9(1): 599-619.
[122] Le Roux N, Bengio Y. Representational power of restricted Boltzmann machines and deep belief networks[J]. Neural Computation, 2008, 20(6): 1631-1649.
[123] Vincent P, Larochelle H, Lajoie I, et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion[J]. Journal of Machine Learning Research, 2010, 11: 3371-3408.
[124] Lecun Y, Bottou L, Orr G B, et al. Efficient BackProp[J]. Neural Networks Tricks of the Trade, 1998, 1524(1): 9-50.
[125] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]. Proc. ICML, 2013.
[126] Schapire R E, Singer Y. BoosTexter: A boosting-based system for text categorization[J]. Machine Learning, 2000, 39(2-3): 135-168.
[127] Calauzènes C, Usunier N, Gallinari P. On the (Non-)existence of Convex, Calibrated Surrogate Losses for Ranking[J]. Advances in Neural Information Processing Systems, 2012, 1: 197-205.
[128] Gao W, Zhou Z H. On the consistency of multi-label learning[J]. Artificial Intelligence, 2013, 199: 22-44.
[129] Ding S F, Zhao H, Zhang Y N, et al. Extreme learning machine: algorithm, theory and applications[J]. Artificial Intelligence Review, 2015, 44(1): 103-115.
[130] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1-3): 489-501.
[131] Dembczynski K, Kotlowski W, Huellermeier E. Consistent Multilabel Ranking through Univariate Losses[J]. Computer Science, 2012: 1319-1326.
[132] 李航. 统计学习方法[M]. 清华大学出版社, 2012.
[133] Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent[M]. Physica-Verlag HD, 2010: 177-186.
[134] Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12: 2121-2159.
[135] Madjarov G, Kocev D, Gjorgjevikj D, et al. An extensive experimental comparison of methods for multi-label learning[J]. Pattern Recognition, 2012, 45(9): 3084-3104.
[136] Reddy K V R, Yedery R D, Aranha C. Antimicrobial peptides: premises and promises[J]. International Journal of Antimicrobial Agents, 2004, 24(6): 536-547.
[137] Hancock R E W, Sahl H G. Antimicrobial and host-defense peptides as new anti-infective therapeutic strategies[J]. Nature Biotechnology, 2006, 24(12): 1551-1557.
[138] Waghu F H, Gopi L, Barai R S, et al. CAMP: Collection of sequences and structures of antimicrobial peptides[J]. Nucleic Acids Research, 2014, 42(D1): D1154-D1158.

[139] Wang G S, Li X, Wang Z. APD3: the antimicrobial peptide database as a tool for research and education[J]. Nucleic Acids Research, 2016, 44(D1): D1087-D1093.
[140] Lee H T, Lee C C, Yang J R, et al. A Large-Scale Structural Classification of Antimicrobial Peptides[J]. Biomed Research International, 2015.
[141] Fan L L, Sun J, Zhou M F, et al. DRAMP: a comprehensive data repository of antimicrobial peptides[J]. Scientific Reports, 2016, 6.
[142] Xiao X, Wang P, Lin W Z, et al. iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types[J]. Analytical Biochemistry, 2013, 436(2): 168-177.
[143] 查锡良, 药立波. 生物化学与分子生物学.第 8 版[M]. 人民卫生出版社, 2013.
[144] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the Acm, 1975, 18(11): 613--620.
[145] Mehrbod A, Zutshi A, Grilo A, et al. Matching heterogeneous e-catalogues in B2B marketplaces using vector space model[J]. International Journal of Computer Integrated Manufacturing, 2017, 30(1): 134-146.
[146] Zhou G P, Doctor K. Subcellular location prediction of apoptosis proteins[J]. Proteins-Structure Function and Genetics, 2003, 50(1): 44-48.
[147] Cedano J, Aloy P, Perezpons J A, et al. Relation between amino acid composition and cellular location of proteins[J]. Journal of Molecular Biology, 1997, 266(3): 594-600.
[148] Ahmad K, Waris M, Hayat M. Prediction of Protein Submitochondrial Locations by Incorporating Dipeptide Composition into Chou's General Pseudo Amino Acid Composition[J]. Journal of Membrane Biology, 2016, 249(3): 293-304.
[149] Behbahani M, Mohabatkar H, Nosrati M. Analysis and comparison of lignin peroxidases between fungi and bacteria using three different modes of Chou's general pseudo amino acid composition[J]. Journal of Theoretical Biology, 2016, 411: 1-5.
[150] Li C, Li X Q, Lin Y X. Numerical Characterization of Protein Sequences Based on the Generalized Chou's Pseudo Amino Acid Composition[J]. Applied Sciences-Basel, 2016, 6(12).
[151] Goktepe Y E, Ilhan I, Kahramanli S. Predicting protein-protein interactions by weighted pseudo amino acid composition[J]. International Journal of Data Mining and Bioinformatics, 2016, 15(3): 272-290.
[152] Wu Y, Tang H, Chen W, et al. Predicting Human Enzyme Family Classes by Using Pseudo Amino Acid Composition[J]. Current Proteomics, 2016, 13(2): 99-104.
[153] Jia J H, Liu Z, Xiao X, et al. Identification of protein-protein binding sites by incorporating the physicochemical properties and stationary wavelet transforms into pseudo amino acid composition[J]. Journal of Biomolecular Structure & Dynamics, 2016, 34(9): 1946-1961.
[154] Mendis S, Davis S, Norrving B. Organizational Update The World Health Organization Global Status Report on Noncommunicable Diseases 2014; One More Landmark Step in the Combat Against Stroke and Vascular Disease[J]. Stroke, 2015, 46(5): E121-E122.
[155] 刘晓娜, 张华, 赵根明, et al. 我国慢性病预防与控制发展历程[J]. 公共卫生与预防医学, 2015, 26(2): 79-83.
[156] 朱银潮, 王永, 李辉, et al. 常见慢性病对患者生活质量的影响[J]. 浙江预防医学, 2016, (1): 24-27.
[157] 严妮妮, 张辉, 邓咏梅. 可穿戴医疗监护服装研究现状与发展趋势[J]. 纺织学报,

2015, 36(6): 162-168.
[158] 朱珍民. 移动健康感知与健康普适服务技术探讨[J]. 医学信息学杂志, 2012, 33(11): 10-15.
[159] 张堉灵. 数据挖掘技术在临床疾病诊疗中的应用探究[J]. 通讯世界, 2016, (18): 71-72.
[160] 吴超, 张晓祥. 数据挖掘在疾病诊断相关组项目中的应用[J]. 中国数字医学, 2010, 05(5): 70-72.
[161] 刘妍. 数据挖掘技术及其在医学信息领域的应用[J]. 科技传播, 2016, (19).
[162] Saeed M, Villarroel M, Reisner A T, et al. Multiparameter Intelligent Monitoring in Intensive Care II: A public-access intensive care unit database[J]. Critical Care Medicine, 2011, 39(5): 952-960.
[163] Little R J, D'agostino R, Cohen M L, et al. The Prevention and Treatment of Missing Data in Clinical Trials[J]. New England Journal of Medicine, 2012, 367(14): 1355-1360.
[164] Dan J, White I R, Leese M. How much can we learn about missing data?: an exploration of a clinical trial in psychiatry[J]. Journal of the Royal Statistical Society: Series A (Statistics in Society), 2010, 173(3): 593-612.

作者简介

姓名：王普　　性别：男　　出生日期：1982.4.29　　籍贯：河南南阳

2013.9 – 2017.7	中科院深圳先进技术研究院模式识别与智能系统专业博士生
2009.9 – 至今	景德镇陶瓷大学自动化专业讲师
2006.9 -- 2009.7	景德镇陶瓷大学智能信息处理方向硕士生
2002.9 -- 2006.7	景德镇陶瓷大学电子科学与技术专业本科生

【攻读博士学位期间发表的论文】

[1] **Wang P**, Ge R, Xiao X, et al. hMuLab: a biomedical hybrid MUlti-LABel classifier based on multiple linear regression[J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016. (SCI 收录)

[2] **Wang P**, Ge R, Xiao X, et al. Rectified-Linear-Unit-Based Deep Learning for Biomedical Multi-label Data[J]. *Interdisciplinary Sciences: Computational Life Sciences*, 2016: 1-4. (SCI 收录)

[3] **Wang P**, Ge R, Xiao X, et al. Multi-label Learning for Predicting the Activities of Antimicrobial Peptides[J]. *Scientific Reports*. (SCI 收录)

[4] **Wang P**, Xiao X. NRPred-FS: a feature selection based two-level predictor for nuclear receptors[J]. *Journal of Proteomics & Bioinformatics*, 2015, 8(4): 1.

[5] **Wang P**, Xiao X. Multi-label learning for Scene Category based on Distance-weighted KNN Algorithm[C]. *2015 International Conference on Automation, Mechanical and Electrical Engineering*, JUL 26-27, 2015, Phuket, THAILAND. (CPCI-S 收录)

[6] Ge R, Mai G, **Wang P**, et al. CRISPRdigger: detecting CRISPRs with better direct repeat annotations[J]. *Scientific Reports*, 2016, 6. (SCI 收录)

[7] Liu J, Jiang H, Gao M, He C, Wang Y, **Wang P**, Ma H, Li Y. An Assisted Diagnosis System for Detection of Early Pulmonary Nodule in Computed Tomography Images[J]. *Journal of Medical Systems*, 2017, 41(2):30. (SCI 收录)

[8] 彭超, **王普**, 葛瑞泉, 周丰丰. 宏基因组中可移动序列的精确检测问题研究[J]. *集成技术*, 2016, 5(2):85-96.

【攻读博士学位期间撰写的专利】

[1] 发明专利名称： 一种成簇的规律间隔的短回文重复序列识别方法及装置，

发明人：周丰丰，葛瑞泉，麦国琴，**王普**，刘记奎,赵苗苗，申请号：201410614178.5

[2] 发明专利名称：基于关键点检测的 NBI 胃镜图像处理方法,
发明人：周丰丰，刘记奎，赵苗苗，葛瑞泉，**王普**，申请号: 201410714561.8

[3] 发明专利名称：基于多标记学习的抗菌肽活性预测方法,
发明人：周丰丰，**王普**，肖绚，葛瑞泉，刘记奎，申请号: 201410712399.6

[4] 发明专利名称：对染色体序列和质粒序列进行分类的方法及装置，
发明人：周丰丰，彭超，**王普**，葛瑞泉，申请号：201510956205.1

【攻读博士学位期间参加的科研项目】

[1] “个体化基因组差异检测的软硬件混合优化系统研究”，深圳市海外高层次人才创新创业专项资金（KQCX20130628112914301），2013 年 12 月~2015 年 12 月。

[2] “专项门户网站及非模式生物系统标注”，中国科学院战略性先导科技专项(XDB13040400)，2014 年 1 月~2015 年 12 月。

[3] “脑卒中基因表达谱的系统生物学研究及应用”，深圳市海外高层次人才创新创业专项资金（KQCX20130628112914291），2013 年 12 月~2015 年 12 月。

[4] “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”，863 计划（SS2015AA020109），2015 年 1 月~2017 年 12 月。