现代生物技术前沿

孙向东　刘拥军
黄保续　谢仲伦　编著

蛋白质结构预测

——支持向量机的应用

科学出版社
www.sciencep.com

现代生物技术前沿

孙向东　刘拥军
黄保续　谢仲伦　编著

蛋白质结构预测

——支持向量机的应用

科学出版社
北京

内 容 简 介

统计学习理论是20世纪90年代逐渐成熟的机器学习理论，以这种理论为基础的支持向量机与以往的学习机器相比具有支持小样本、不会陷入局部势井、鲁棒性好以及运算成本低等优势. 实现这种理论的支持向量机算法已经成为机器学习和知识挖掘的标准工具.

自从 2001 年支持向量机被首次用于蛋白质二级结构的预测以来，这种算法发展到蛋白质的结构类型、亚细胞结构和膜蛋白的结构等领域的预测中. 本书详细介绍了依据统计学习理论构建支持向量机的方法、各种相关软件原理和使用方法，并以二级结构和结构域为例介绍了以支持向量机为工具预测蛋白质结构的方法. 书中使用了大量的原创性实验结果，理论联系实际，详细阐述了以支持向量机为工具预测蛋白质结构的全过程.

本书适合从事蛋白质结构基础研究的学生和科技工作者阅读.

图书在版编目(CIP)数据

蛋白质结构预测：支持向量机的应用/孙向东等编著. —北京：科学出版社, 2008

(现代生物技术前沿)

ISBN 978-7-03-022387-6

Ⅰ.蛋… Ⅱ.孙… Ⅲ.向量计算机-算法理论-应用-蛋白质-生物结构-预测 Ⅳ.Q510.1

中国版本图书馆 CIP 数据核字(2008) 第 092886 号

责任编辑：李 晓 李 悦 / 责任校对：曾 茹
责任印制：张克忠 / 封面设计：陈 敬

科学出版社出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

骏杰印刷厂印刷

科学出版社发行 各地新华书店经销
*
2008 年 9 月第 一 版 开本：B5(720×1000)
2008 年 9 月第一次印刷 印张：13 插页：2
印数：1—1500 字数：251 000

定价：50.00 元

(如有印装质量问题，我社负责调换〈环伟〉)

前　言

蛋白质由氨基酸残基线性序列构成, 折叠成特定的空间构象后, 蛋白质就具有相应生物学活性和功能. 了解氨基酸残基序列与其空间结构的关系, 是全面认识蛋白质结构和其生物学功能的关系的重要前提. 近些年来, 蛋白质序列数据库的数据积累速度非常快, 与之相比, 蛋白质结构数据库的数据积累速度远不及序列数据库的数据积累速度. 尽管蛋白质结构测定技术有了较为显著的进展, 但是通过实验方法确定蛋白质结构的过程仍然非常复杂, 实验周期很长.

另外, 随着 DNA 测序技术的发展, 人类基因组及很多模式生物基因组已经或将要完全测序, DNA 序列数量将会急增. 由于 DNA 序列分析技术和基因识别方法的进步, 人们可以从 DNA 序列直接推导出大量的蛋白质序列, 这将导致蛋白质序列数据数量急剧增加. 了解了这些序列的结构, 可以使它们直接为人类服务.

氨基酸残基序列的结构分析是对生物学家的极大挑战. 20 世纪 60 年代后期, Anfinsen 首先发现去折叠蛋白或者说变性蛋白质在允许重新折叠的实验条件下可以重新折叠到原来的结构, 这种天然结构对于蛋白质行使生物功能具有重要作用, 蛋白质只有在折叠成其天然结构的时候才能具有完全的生物活性. 因此 Anfinsen 提出了蛋白质折叠信息隐含在蛋白质的一级结构中的观点. 以这种观点为基础, 通过对蛋白质一级结构的研究, 发现其折叠密码后, 仅通过一级结构信息就能预测蛋白质空间结构.

蛋白质结构预测主要有两大类方法. 一类是蛋白质分子特性理论分析方法或从头算方法, 通过理论计算 (如分子力学、分子动力学计算) 进行结构预测. 该类方法假设折叠后的蛋白质取能量最低的构象. 从原则上来说, 人们可以根据物理、化学原理, 通过计算来进行结构预测. 另一类蛋白质结构预测的方法是统计学方法. 该类方法对已知结构的蛋白质进行统计分析、建立序列到结构的映射模型、进而根据映射模型对未知结构的蛋白质直接从氨基酸序列预测结构. 这是进行蛋白质结构预测较为成功的一类方法. 这类方法包括经验性方法、结构规律提取方法、同源模型化方法等. 统计学方法本身就是不确定性方法, 目前虽然还不能完全替代第一类方法而成为预测蛋白质结构的主要方法, 但是发展前景很广阔. 其中以统计学习理论为基础的支持向量机预测蛋白质结构的方法发展非常迅速.

统计学习理论是在 20 世纪 90 年代逐渐成熟的机器学习理论, 以这种理论为基础的支持向量机与以往的学习机器相比具有支持小样本、不会陷入局部势井、具有很好的鲁棒性以及运算成本低等优势. 实现这种理论的支持向量机算法已经成

为机器学习和知识挖掘的重要工具. 从 2001 年支持向量机首次被运用进行蛋白质二级结构的预测以来, 这种算法已经被用于对于蛋白质的结构类型、亚细胞结构和膜蛋白的结构等领域的预测中.

本书一共包含 8 章, 阐述三部分内容, 包括生物信息学基本知识、蛋白质结构预测基本知识、蛋白质二级结构和结构域预测技术、支持向量机算法以及相应软件的使用方法和实验步骤, 由浅及深, 步步深入, 系统阐述了运用支持向量机预测蛋白质二级结构和结构域的基本原理和过程. 有兴趣的读者可以按照本文描述的实验步骤和相应参数完全重复整个实验过程. 第一部分包括第 1 章和第 2 章, 主要对蛋白质二级结构和结构域预测以及知识背景进行简要介绍. 第二部分包含第 3 章到第 6 章, 系统阐述了统计学习理论、以这种理论为基础的学习算法 —— 支持向量机、支持向量机构造方法以及实现支持向量机算法的程序 Libsvm. 最后一部分包括第 7 章和第 8 章, 这一部分详细论述了运用支持向量机方法进行蛋白质二级结构预测和结构类型预测的实验过程和最终结果.

由于作者水平有限、成文仓促, 文中难免出现这样那样的疏漏和错误. 书中欠妥之处敬请读者批评指正!

孙向东　刘拥军　黄保续　谢仲伦

2008 年于北京

目　录

第 1 章　蛋白质结构预测概述

1.1　蛋白质预测基本方法简介

生物信息学是近年来最有活力的生物学研究领域之一, 人们从生物信息的研究中获得了对生命本质更丰富的知识和更深刻的理解. 核酸序列中蕴含着生命的基本信息, 这些信息是自然界留给人类的、解读生命的 “天书”. 理解这本天书是最终了解自然、了解生命、了解人类自身的重要途径, 是人类从必然王国到自由王国飞跃的基本前提之一.

由基因决定的蛋白质执行着生物体内各种重要的功能, 如生物化学反应的催化、营养物质的输运、生长和分化控制、生物信号的识别和传递等. 基因确定了组成蛋白质的氨基酸序列. 虽然蛋白质由氨基酸的线性序列组成, 但是它们只有折叠成特定的空间构象才能具有相应的活性和相应的生物学功能. 了解蛋白质的空间结构不仅有利于认识氨基酸残基序列与空间结构的关系, 也有利于认识蛋白质的结构与其生物学功能的关系.

根据近些年来的经验, 蛋白质序列数据库数据积累速度非常快, 而且还有加快的趋势. 尽管蛋白质结构测定技术有了较为显著的进展, 但是通过实验方法确定蛋白质结构的过程仍然非常复杂, 实验周期很长. 另外, 随着 DNA 测序技术的发展, 人类基因组及很多的模式生物基因组已经或将要被完全测序, DNA 序列数量将会剧增, 由于 DNA 序列分析技术和基因识别方法的进步, 人们可以从 DNA 序列直接推导出大量的蛋白质序列. 这意味着已知序列的蛋白质数量和已测定结构的蛋白质数量 (如蛋白质结构数据库 PDB 中的数据) 的差距将会越来越大. 面对这种蛋白质结构信息与 DNA 序列信息发展速度的不平衡, 人们希望找到一些预测方法, 通过这些方法加快蛋白质结构产生速度, 缩小二者之间的差距.

为了缩小这种差距, 要么改进现有的蛋白质测序技术和结构预测方法, 要么发展新的理论分析方法, 这是对生物学家的极大挑战. 20 世纪 60 年代后期, Anfinsen 首先发现去折叠蛋白质或者说变性蛋白质在允许重新折叠的实验条件下可以重新折叠到原来的结构, 这种天然结构对于蛋白质行使生物功能具有重要作用, 蛋白质只有在折叠成其天然结构的时候才能具有完全的生物活性. 因此 Anfinsen 提出了蛋白质折叠的信息隐含在蛋白质的一级结构中的观点. 基于这种观点, 人们相信通过对蛋白质一级结构的研究, 发现其折叠密码后能够仅通过一级结构信息就能预测蛋白质空间结构.

到目前为止, 科学家对于蛋白质结构预测进行了大量的研究, 已经尝试了一些预测蛋白质结构的方法. 蛋白质结构预测主要有两大类方法. 一类是蛋白质分子特性的理论分析方法或从头算方法, 通过理论计算 (如分子力学、分子动力学计算) 进行结构预测. 该类方法假设折叠后的蛋白质取能量最低构象. 从原则上来说, 人们可以根据物理、化学原理, 通过计算来进行结构预测. 但是这种方法可操作性很差, 主要有几个原因:

(1) 自然的蛋白质结构和未折叠的蛋白质结构之间的能量差非常小 (1kcal/mol 数量级);

(2) 蛋白质可能的构象空间庞大, 针对蛋白质折叠的计算量非常大;

(3) 计算模型中蛋白质及溶剂系统的力场参数的不准确性、无法从数学上解决局部势井问题, 因此无法证明某蛋白质分子的构象是全局自由能最小的构象.

另一类蛋白质结构预测的方法是统计学方法. 该类方法对已知结构的蛋白质进行统计分析、建立序列到结构的映射模型、进而根据映射模型对未知结构蛋白质直接从氨基酸序列预测结构. 这是进行蛋白质结构预测较为成功的一类方法. 这一类方法包括经验性方法、结构规律提取方法、同源模型化方法等. 但是这类方法不可能是完全独立的, 它们不能脱离对蛋白质分子的物理、化学和生物性质的研究. 统计学方法本身就是不确定性方法, 目前还不可能替代第一类方法而成为预测蛋白质结构的最终方法, 而只能是一种辅助方法.

1.2　蛋白质二级结构和结构域预测方法简介

蛋白质结构预测已经有了几十年的历史. 通过对已知空间结构蛋白质分子的研究和分析, 人们发现, 尽管一条多肽链采取构象的数目是相当大的, 但在蛋白质分子中由三级结构组装而形成的一定空间结构的方式却是有限的. 蛋白质二级结构是这种组装的基本单位, 蛋白质二级结构预测和由二级结构构成的结构域预测就成了解决由蛋白质的一级结构序列预测其空间结构这一问题的关键步骤.

蛋白质二级结构的预测开始于 20 世纪 60 年代中期, 到目前为止人们已经提出几十种预测蛋白质二级结构的方法. 这些方法大体分为三代, 第一代是基于单个氨基酸残基统计分析, 从有限的数据集中提取各种残基形成特定二级结构的倾向, 以此作为二级结构预测的依据, 这种方法的代表是 Chou-Fasman 方法. 第二代预测方法是基于氨基酸片段的统计分析, 使用大量的数据作为统计基础, 统计的对象不再是单个氨基酸残基, 而是氨基酸片段, 片段的长度通常为 11~21 个氨基酸. 片段体现了中心残基所处的环境. 在预测中心残基的二级结构时, 以残基在特定环境中形成特定二级结构的倾向作为预测依据. 这种方法的代表是 GOR 方法. 二级结构预测的第三代方法运用蛋白质序列的长程信息和蛋白质序列的进化信息, 使二级

结构预测的准确程度有了比较大的提高, 特别是对 β 折叠的预测准确率有较大的提高, 预测结果与实验观察趋于一致. 这种方法的代表是人工神经网络方法.

Chou-Fasman 方法是一种基于单个氨基酸残基统计的经验参数方法, 由 Chou 和 Fasman 在 20 世纪 70 年代提出. 通过统计分析, 获得每个残基出现于特定二级结构构象的倾向性因子, 进而利用这些倾向性因子预测蛋白质的二级结构. Chou-Fasman 方法构象参数的物理意义明确, 方法中二级结构的成核、延伸和中止规则可能真实地反映了真实蛋白质中二级结构形成的过程, 并且可以较简单地用手工完成一个蛋白质分子的二级结构预测, 预测准确率约为 50%.

GOR 方法是一种基于信息论和贝叶斯统计学的方法, 方法的名称以三个发明人姓名的第一个字母组合而成 (Garnier, Osguthorpe, Robson). GOR 方法也是建立在对已知的氨基酸构象分析统计基础上的, 计算被预测结构的位置特异的概率. GOR 方法给出了 20 种氨基酸残基出现在不同位置时的直接信息表. 假定相邻阶段所含的信息可以近似表示为若干个直接信息的简单加和, 根据这一公式和相应的直接信息表, 就可以对一条肽链中任一位置残基的构象进行预测. 这种方法的预测准确率约为 63%.

人工神经网络是一种复杂的信息处理的机器学习模型. 这种模型最早在 20 世纪 80 年代末用于蛋白质二级结构的预测、蛋白质结构的分类、折叠方式的预测以及基因序列的分析等. 将神经网络用于二级结构预测最早是由 Qian 和 Sejnowskit 提出的, 他们受到神经网络在文字语言处理方面应用的启发, 将蛋白质序列看作是由各种氨基酸字符组成的字符序列, 将氨基酸残基片段作为输入的一串语言字符, 二级结构即为对应的输出结果. 神经网络可以有效地学习蛋白质二级结构形成的复杂规律或模式, 提取更多的信息, 并利用所掌握的信息进行预测. 利用神经网络方法可以提高二级结构预测准确率. 神经网络方法利用多序列比对的信息, 能够得到超过 70%的二级结构预测准确率. 最近 Petersen 等以位置特异性得分矩阵作为输入, 使二级结构预测的准确率达到更高的水平.

支持向量机方法是最近刚刚发展起来的蛋白质结构预测技术. 2001 年, 支持向量机首次应用于蛋白质二级结构预测, 马上就显示出这种方法的优势. 通过支持向量机方法得到的蛋白质二级结构预测准确率达到了令人惊奇的 73.5%. 之后几年, 科学家又向前走了一步, 预测准确率达到了 75.2%.

蛋白质结构域要比二级结构复杂, 预测结构域也比预测二级结构的不确定性大些. 目前蛋白质结构域的预测主要在于其折叠类型的预测. 对于蛋白质折叠类型没有一个统一的标准, 因此定义也较为混乱. 总的来说蛋白质的结构类型可以分为 α 螺旋、β 折叠、$\alpha+\beta$ 结构和 α/β 结构.

在自然状态下, 蛋白质的折叠类型不超过 1000 种, 蛋白质相互作用的数量也是有限的. 由于不同蛋白质之间的相互作用和蛋白质与相应配体之间的相互作用

都由它们的三维结构决定, 所以收集、探索和挖掘蛋白质结构数据库中的这类信息对于生命本质研究至关重要. 然而, 对于生物体基因序列的研究、这些基因可以表达的生物分子的结构的研究以及这些结构可以表现出来的功能的研究之间存在不平衡. 一方面, 沉淀在序列数据库中的数据越来越多, 通常这些序列是功能不很清楚的原始数据; 另一方面, 在蛋白质数据库 (protein data bank) 中的结构信息积累相对缓慢, 计算方法就成为预测蛋白质结构的实验方法以外的重要补充.

在蛋白质结构域的折叠类型预测方法中, 氨基酸组分方法和双组件效果的氨基酸组分方法的研究最充分. 仅依赖序列中氨基酸成分, 即仅依赖氨基酸残基在序列中的百分比而不考虑其他因素的影响, 预测准确率就可以达到 80%. 在这种方法上发展起来了双组件效果的氨基酸组分方法和双组件算法. 近十年来, 使用双组件算法用于预测蛋白质结构类可以达到很高的准确率.

然而这个准确率仍然不能满足人们的需要, 相对于 X 射线衍射方法和核磁共振方法得到的准确率仍然有一定的差距. 蛋白质二级结构预测识别率不高的原因复杂. 全面提高蛋白质二级结构预测的准确率是一个系统涉及多领域、多学科的系统工程.

第 2 章　相关知识背景

2.1　生物信息学

2.1.1　生物信息学的定义、目的、内容和发展趋势

生物信息学是一门边缘学科, 它的知识体系中包含了生物学 (生物化学、遗传学、结构生物学等)、计算机科学 (计算理论、人工智能、机器学习以及动态规划等)、物理化学 (热力学、分子建模等) 及数学 (算法、建模技术、概率论与数理统计等) 等方面的知识. 自从生物信息学这个研究领域被开辟以来, 它就以极快的速度发展并快速延伸其学科范围, 并逐渐建立了与多个学科之间的联系, 因此很难明确地界定生物信息学中各个学科之间的界限. 生物信息学主要的研究领域涉及基因组学、蛋白质组学、生物化学、数据挖掘、分子进化、分子建模以及算法等 [1].

简单、直观地从字面意思上来看, 生物信息学由 “生物” 和 “信息” 两部分组成. “生物” 部分一般指的是分子生物学, 包括进化论和遗传学; “信息” 部分指的是计算机科学. 这样把这两部分链接在一起指的就是用计算机科学的方法解决分子生物学的问题 [2]. Luscombe 在 2001 年给出一个明确的定义: “生物信息学是根据分子 (从物理化学的角度) 和信息技术 (源自应用数学、计算机科学和统计学的原则) 的应用来理解和组织与这些分子相关的大规模的信息, 即生物信息学是分子生物学的信息管理系统和诸多实践上的应用”[3]. 生物信息学可以简明扼要地定义为利用计算机方法理解、组织和解析分子生物学研究中的信息. 其实 “生物” 和 “信息” 两种学科之间结合的本质原因是有机体的生理学和行为大体上由它的基因导向, 有机体本身的生长和发育受各种信息的指挥和调节, 而生物学本身就可以认为是一种信息技术 [3].

生物信息学的研究是由大量数据驱动的, 各种各样的数据处理方法和工具被应用于分子生物学的研究中. 数据处理方法和工具的革新会推动生物信息学的发展, 计算技术和实验技术的革新使得数据快速沉淀到公共数据库中, 高容量的存储器和高技术处理器的高速处理能力提高了传统实验室数据处理效能. 在生物信息学领域另一个起到决定作用的是互联网的产生和快速发展, 通过互联网人们可以很容易地访问和交流海量的生物信息数据. 这些是生物数据急剧增长的主要原因. 由于生物信息学数据库中的数据急剧增加, 很多生物学问题实际成了计算问题. 计算机是一种理想的工具, 它不但可以大量处理数据, 而且利用恰当的软件包还能寻找这些数

据的复杂的动力学规律 [3].

生物信息学技术的发展使生物分析向两个方向发展：深度和宽度. 深度方面, 主要目标在于药物理性设计. 它的目标是取得单独的蛋白质并对其进行彻底地分析以透彻地理解这个蛋白质在生物体中的功能. 为了高效地实现这一目标, 人们设计了一整套方法. 首先对基因组进行测序, 并从中找到可读框. 然后运用恰当的方法、依据可读框翻译的蛋白质一级序列预测该蛋白质的高级结构. 利用几何计算可以确定蛋白质表面的形状, 模拟计算可以确定周围受力区域. 最后使用分子对接算法鉴别和设计可能与蛋白质结合的配体, 为药物设计铺平了道路. 然而这一整套方法中所应用的技术有些目前还不成熟, 其中有些技术还处于探索阶段, 利用这些技术一般难以得到精确的预测结果. 因此虽然使用计算工具理解生物分子的结构和功能比实验更加方便, 但是确定分子结构和功能的最可靠途径还是通过直接的实验. 从广度方面来分析, 首先是把基因同其他的基因进行比较, 以确定基因在生物进化中的位置和在有机体中可能发挥的功能. 其次, 对于蛋白质结构进行预测、研究蛋白质结构与功能的关系也是生物信息学发展的重要方向.

生物信息学的目的主要在于三个方面 [4]：

(1) 组织信息. 生物信息学组织数据的目标之一是使得查询者可以得到存在的信息并提交他们获得的新数据. 数据储存仅是生物信息学的一项基本任务, 这些存储的数据在分析之前还不能发挥作用.

(2) 数据分析. 寻找新的工具和信息源来分析数据. 例如, 把一个蛋白质序列与已知的特征序列比较, 这就不仅仅需要直接的数据查询. 生物信息学的分析工具还必须能分析有机体的基因组和蛋白质组之间有意义的共同之处, 做到这一点就需要广泛地汇聚计算理论方面的知识以及分析者对生物的生理生化规律的透彻理解.

(3) 信息释义. 使用合适的工具分析并且解释所得数据的生物学含义, 从而发现新的知识. 传统上, 生物学详细考察单个系统, 并且比较与之相关的少数几个系统. 然而对有机体的生物信息学分析则必须从当前可以得到的数据中对该有机体以及与其相关的生物系统进行全面的比较, 以便揭示涉及多个系统的一般规律和这些系统的重要特征.

从目前生物信息学的研究情况来看, 国际上公认的生物信息学的研究内容, 大致包括以下几个方面 [5]：

(1) 生物信息的收集、存储、管理与提供. 包括建立国际基本生物信息库和生物信息传输的国际互联网系统、建立生物信息数据质量的评估与检测系统、生物信息的在线服务以及生物信息可视化和专家系统.

(2) 基因组序列信息的提取和分析. 包括基因的发现与鉴定, 基因组中非编码区的信息结构分析, 提出理论模型, 阐明该区域的重要生物学功能. 进行模式生物完整基因组的信息结构分析和比较研究. 利用生物信息研究遗传密码起源、基因组

结构的演化、基因组空间结构与 DNA 折叠的关系以及基因组信息与生物进化关系等生物学的重大问题.

(3) 功能基因组相关信息分析. 包括与大规模基因表达谱分析相关的算法、软件研究, 基因表达调控网络的研究. 与基因组信息相关的核酸、蛋白质空间结构的预测和模拟以及蛋白质功能预测的研究.

(4) 生物大分子结构模拟和药物设计. 包括 RNA(核糖核酸) 的结构模拟和反义 RNA 的分子设计, 蛋白质空间结构模拟和分子设计, 具有不同功能域的复合蛋白质以及连接肽的设计, 生物活性分子的电子结构计算和设计, 纳米生物材料的模拟与设计, 基于酶和功能蛋白质结构、细胞表面受体结构的药物设计, 基于 DNA 结构的药物设计等.

(5) 生物信息分析的技术与方法研究. 包括发展能支持大尺度作图与测序需要的软件、数据库以及若干数据库工具, 如电子网络等远程通信工具. 改进现有的理论分析方法, 如统计方法、模式识别方法、隐马尔可夫过程方法、分维方法、神经网络方法、复杂性分析方法、密码学方法、多序列比较方法、统计学习理论方法等. 创建一切适用于基因组信息分析的新方法、新技术, 包括引入复杂系统分析技术、信息系统分析技术等. 建立严格的多序列比较方法. 发展与应用密码学方法以及其他算法和分析技术, 用于解释基因组的信息, 探索 DNA 序列及其空间结构信息的新表征, 发展研究基因组完整信息结构和信息网络的研究方法等, 发展生物大分子空间结构模拟、电子结构模拟和药物设计的新方法与新技术.

(6) 应用与发展研究. 汇集与疾病相关的人类基因信息, 发展患者样品序列信息检测技术和基于序列信息选择表达载体、引物的技术, 建立与动植物良种繁育相关的数据库以及与大分子设计和药物设计相关的数据库.

生物信息学发展的未来趋势主要在以下几个方面 [6]: ① 计算基因组学, 包括高通量基因组测序、模型化和注释; ② 计算结构生物学, 包括模型比较和蛋白质折叠解析; ③ 计算大分子化学, 包括解析低分辨率的折叠拓扑和高分辨率的结构; ④ 分子识别的计算分析, 包括分子对接和分子结构仿真; ⑤ 计算细胞生物学 [7].

2.1.2 基因组学

从生物信息学的数据处理的性质来看, 生物信息学包括基因组学和蛋白质组学两个方面 [8]. 20 世纪 90 年代初, 人类基因组组织很多国家的科学家和分子生物学研究机构着手展开人类基因组计划 [9], 这个计划开启了基因组时代的曙光. 人类基因组计划的目的是要测出人类每一条染色体的完整 DNA 序列, 它的主要研究工作集中于大规模的基因组测序. 第一个微生物*H. influenza* 的完整基因组测序工作完成于 1995 年. 第二年, 测序工作进程有所加快, 三个基因组*S. cerevisiae* [9], *M. jannaschii* [10] 和*M.genitalium* [11] 的测序工作相继完成. 测序技术的完善和互联网

技术的发展是基因组测序的进程加快的主要原因. 人类基因组草图于 2000 年中期完成, 于 2001 年公开发表 [12]. 在这个草图中包含了绝大部分的功能基因组和未表达的蛋白质组信息. 虽然它仅仅是草图, 仍然可以从中发现很多有用的信息.

人类基因组中大约包含 30 亿个碱基对, 人们预测包含 3 万~4 万个蛋白质编码基因, 其中包含和关于人类的发展、生理、医药和进化方面的重要、有用的信息. 因此人们需要有效的工具从这些数据中发现信息并快速处理积累的信息 [13]. 目前已经测序的 DNA 序列数据都在互联网上公布, 任何人都可以免费下载这些实验数据.

2.1.3　蛋白质组学

蛋白质组指的是对有机体的整个生命过程起作用的一切蛋白质的总称. 随着人类基因组草图的绘制完成, 生物信息学的研究进入后基因组时代, 并打开了蛋白质组学研究的序幕. 人类基因组计划完成后, 基因的功能和作用并未阐明, 而绘制决定生命体多样性、复杂性及其功能的蛋白质组图谱, 将使人类基因组中绝大部分基因的功能得到揭示和阐述. 人类蛋白质组研究对揭示生命活动规律和本质、探索人类重大疾病发生、发展机制具有深远的意义, 由此必将广泛推动生命科学、生物技术以及信息、分析、材料等科技领域的发展.

蛋白质组是生命活动的执行体, 是基础研究与应用研究、生命科学与医药产业及生物经济的纽带和桥梁, 是极为重要而又有限的生物战略资源. 蛋白质组研究不仅可以实现与基因组的对接与确认、直接揭示生命活动的规律和本质特点以及人类重大疾患发生与发展的病理机制, 而且可广泛推动和促进生命科学基础学科以及分析科学、信息科学、材料科学等应用学科的发展. 随着人类基因组计划的完成, 蛋白质组的研究已经成为 21 世纪生命科学发展的先导, 成为生命科学乃至自然科学最活跃的学科领域.

2003 年底, 国际人类蛋白质组计划正式启动, “人类肝脏蛋白质组计划” 和 “人类血浆蛋白质组计划”、“人类脑蛋白质组计划”、“大规模抗体计划” 和 “蛋白质组标准计划” 五大项目首先开始执行. 其中“人类肝脏蛋白质组计划”由中国科学家领导执行 [14]. 2004 年 10 月 25 日, 以 “蛋白质组学 —— 基因组的诠释” 为主题的第三届国际人类蛋白质组大会在北京隆重开幕, 2000 余位科学家齐聚一堂共同探讨人类蛋白质组研究. 会上, 安捷伦科技蛋白质组市场开发经理 Rudy Grimm 博士说: “人类基因工程成功的关键在于发展自动化程度高、易于操作, 且能够快速、大批量进行基因排序的科学技术. 然而, 目前蛋白质组学研究的开展远没有达到大批量和大产出的程度, 同时还面临着自动化程度较低, 缺乏高水平专业技术人员的局面 ……”[15].

蛋白质组学的研究比基因组学的研究更加困难. 基因的功能由碱基序列完全确

定, 而蛋白质的功能则是通过一级序列确定的、不同空间结构来实现的. 结构完全不同的蛋白质可能具有类似的氨基酸序列, 同时结构相同的蛋白质其序列差别可能很大 [16]. 生物体通常通过复制具有某种基因的多拷贝, 并且不同种类的生物当它们在进化过程中分化时通过遗传使它们具有等价的或相似的蛋白质. 在结构水平上, Chothia 预测蛋白质三维结构的数量是有限的, 这个数目在 1000~10 000[17]. 虽然 PDB 数据库中的蛋白质结构呈指数增长, 但是发现新折叠类型的速率却在下降 [18]. 因为蛋白质的折叠种类大大小于基因的种类, 蛋白质折叠的分类对于基因组的内容提供了一个坚实的简化 [19,20]. 这个基本的发现就是人们通过计算机从蛋白质一级序列预测蛋白质高级结构的依据. 管理这一层面的信息在于发展评估不同生物分子相似性的方法以及鉴别它们的相似性 [18].

2.1.4 数据库

Kanehisa 认为 "发现受数据驱动" 是后基因组时代的特征 [21]. 因此发现、递交、整理和分析数据是生物信息学的重要任务. 人们已经建立了数目庞大、种类众多的各种生物信息数据库. 这些数据库主要包括了基因序列数据库和蛋白质序列数据库, 另外还有一些数据库既收集基因序列也收集蛋白质序列. 近些年来, 人们投入了很大的人力、物力对生物信息数据进行收集和整理, 因为大量数据的存入, 使得当前的生物信息数据以指数速率膨胀. 图 2-1 和图 2-2 直观描述了 PDB 数据库和 GenBank 数据库的增长情况. 从两个图中可以看出两个数据库中的数据量都显示了呈指数增长的趋势. 造成这种现象的原因在于更新、效率更高的分析基因组

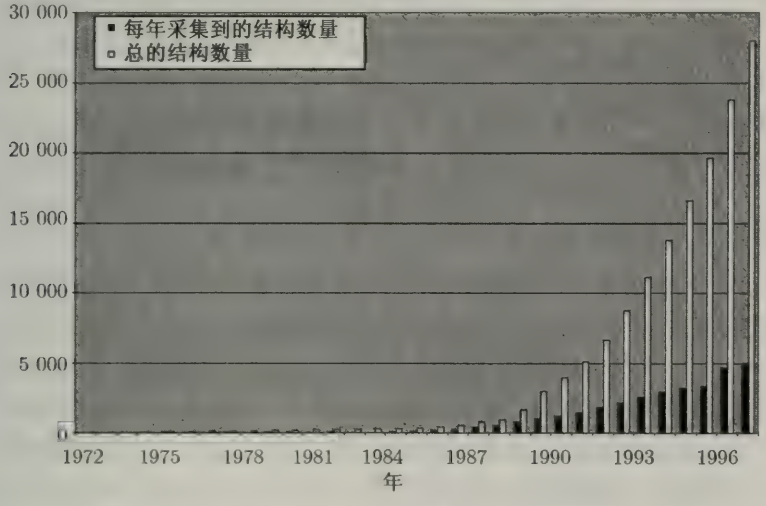


图 2-1 protein data bank 中的数据每年呈指数增长示意图
(图中数据来自 http://www.rcsb.org/pdb/holdings.html)

和蛋白质组的技术的使用. 根据目前数据量的增长趋势, 公共数据库中的 DNA 和蛋白质序列数据 15 个月就会翻倍 [22]. 处理和分析储存在数据库中的信息已经成为生物信息工作者的主要任务.

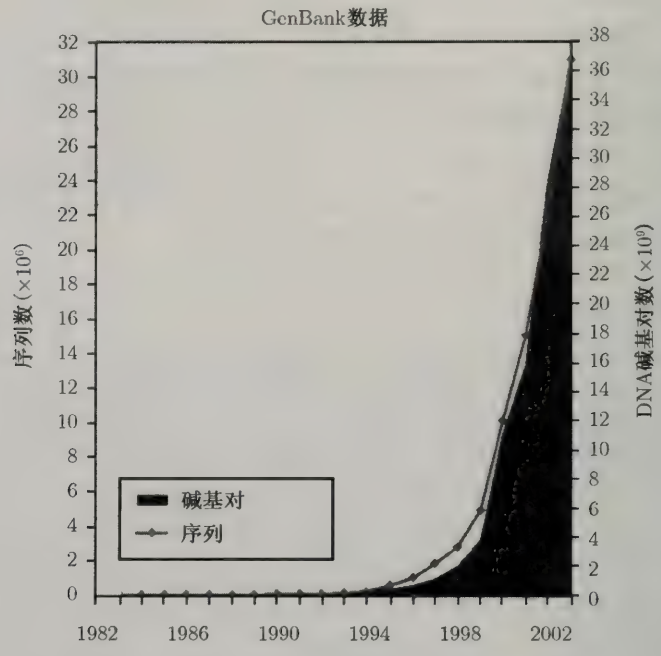


图 2-2　GenBank Data 中的数据每年呈指数增长示意图
(源自 http://www3.ncbi.nlm.nih.gov/Genbank/genbankstats.html)

蛋白质数据库可以分为一级数据库、复合数据库和二级数据库. 一级数据库作为原始数据的仓库包含了约 30 万个蛋白质序列和功能 [55]. 例如, SWISS-PROT[23]、PIR-international[24] 和蛋白质数据库 protein data bank(PDB)[25,26], 这些数据库提供了已经解析的所有类型大分子的三维结构, 包括蛋白质、RNA、DNA 和各种复合体. 其中大多数的结构是运用 X 射线衍射和核磁共振技术得到的, 也有一些是理论模型. 因为 PDB 中的条目信息很难摘录, 所以 PDBsum[27] 为每个结构提供了一个网页来展示该结构的结构分析、示意图和不同分子之间的相互作用数据. 这些数据库同时还对数据库中的序列进行注释、描述蛋白质的功能、结构域以及进行后翻译修正. 复合数据库, 如 OWL[28] 和 NRDB[29], 这些数据库通过从不同一级数据库中编辑和过滤序列数据来产生组合的非冗余集合. 这种数据库比保存单一种类分子的数据库更完善、更丰富, 其中还包含了从 DNA 序列数据库的密码子区翻译得到的蛋白质序列. 二级数据库包含了从蛋白质序列中获得的知

识, 并帮助使用者确定是否一个新的序列属于已知的家族. 最著名的二级数据库为 PROSITE[30]. 下面三个主要的蛋白质结构数据库 CATH[31]、SCOP[32] 和 FSSP 数据库 [33] 对于 PDB 数据库提供的结构对蛋白质进行了分类, 以便提供结构和进化关系方面的信息. 这三个数据库都包含了等级结构分类, 其中每种蛋白质在分类树的等级越低它们的相似性就越接近. 生物信息学研究人员的下一个挑战是从这些数据库中学习、发现和预测有用的信息.

2.2 蛋白质序列、结构与功能的关系

生物信息学在后基因组时代的最终目的是确定每条新发现序列的生物学功能及其在生物体中的角色 [34]. 解析有机体中蛋白质的结构和分析其功能是蛋白质组时期生物信息学工作者的主要任务. 随着人类和其他动物基因组测序工作的完成, 生物学研究面临的最重要的挑战之一, 就是如何依据基因序列预测它所翻译的蛋白质的高级结构、进而预测该蛋白质的功能. 如果能够做到这一点, 将在所有生物技术与药物设计领域产生决定性的影响. 蛋白质是生物体中含量最高、功能最重要的生物大分子, 蛋白质存在于所有生物细胞中, 约占细胞干质量的 50%以上. 作为生命的物质基础之一, 蛋白质在催化有机体内各种反应进行、调节代谢、抵御外来物质入侵及控制遗传信息等方面都起着至关重要的作用. 有机体中几乎所有的生命活动都是靠蛋白质完成的, 蛋白质的功能与它的结构密切相关.

构成天然蛋白质的氨基酸残基共有 20 种, 蛋白质是由氨基酸脱水缩合形成的多肽链折叠成的紧凑三维结构. DNA 中的基因控制合成蛋白质, 碱基序列决定了蛋白质的氨基酸种类及其排列顺序. 蛋白质的结构可以分为 6 个级别: 一级结构、二级结构、超二级结构、三级结构、四级结构以及分子缔合体. 氨基酸残基的线性序列是蛋白质的一级结构. 蛋白质的二级结构为蛋白质分子中多肽主链的规则排布, 主要包含螺旋、折叠和无规则卷曲等三种形式. 超二级结构为二级结构单元间的组合方式. 三级结构指的是蛋白质的三维空间结构, 四级结构是蛋白质亚基之间的相互作用 [35]. 许多有用的蛋白质结构信息可以从蛋白质的结构单位中获得, 比如蛋白质的功能和活性位点、交互作用机制和进化理论等.

通过对于完全变性的核糖核酸酶的复性研究, 建立了关于蛋白质序列与结构关系的一般性结论, 称作热力学假说 [36]. 热力学假说认为 “一个蛋白质在正常的生理环境 (溶液、pH、离子强度、其他离子或非肮基团有无以及温度等) 中固有的三维结构是这样一种结构, 整个系统中的 Gibbs(吉布斯: 吸收单位, 1Gibbs= 10^{-10} mol/cm 的表面浓度) 自由能最小, 即固有构象完全由分子间相互作用决定. 也就是说, 在一定的环境中由氨基酸序列完全决定”. 从自然选择方面来说, 在进化过程中蛋白质分子只有存在于与选择它的环境相似的环境中时才可能出现稳定的

结构, 这种状态称作生理状态. 生物功能与蛋白质分子的几何形状的关系比与氨基酸成分之间的关系更密切些 [37].

蛋白质结构预测是蛋白质结构与功能研究工作的重要组成部分 [38]. 蛋白质在生物体中的角色决定于它的功能, 而蛋白质的功能大体由它们的结构决定. 因此了解蛋白质的三维结构对人们了解其功能提供了很大帮助. 虽然通过实验解析蛋白质结构的速度越来越快, 但是, 由于生物种类繁多, 每种生物的蛋白质组又不完全相同, 所以所有的蛋白质结构都通过实验来进行解析是不可能的. 当越来越多的蛋白质结构被人们了解以后, 人们就可以从中找到蛋白质结构由一级结构折叠成高级结构的规律, 这时人们就可以借助计算机来预测蛋白质的结构. 通过几十年的努力, 人们预测蛋白质结构的技术取得了巨大的进步. 蛋白质结构预测方法大体分为三种: ① 同源建模; ② 穿针引线方法或折叠识别; ③ 从头预测. 一般来说预测的类别反映了可以从数据库中得到哪类信息.

人们之所以热衷于研究蛋白质序列与结构之间的关系, 是与蛋白质的功能由蛋白质的结构所确定的这一论断分不开的 [39]. 通过蛋白质的结构可以识别暴露在蛋白质表面的并能溶解到溶剂中的氨基酸残基和深埋在蛋白质结构内部的氨基酸残基、蛋白质分子的表面形状和分子组成成分以及每个基团的毗邻关系. 同时也可以揭示蛋白质晶体所处的生理环境或者高浓度溶解下的四级结构. 蛋白质与配体的结合方式也是人们想要了解的最有用的功能信息, 因为这些信息揭示了配体与蛋白质结合的本质. 如果蛋白质是酶, 人们还可以通过活性位点的氨基酸排列来推测其催化机制. 根据传统的方法, 这种复合体可以通过设计配体来确定, 如在结晶化方法中加适当的配体. 而当配体未知时, 人们也可以通过结构基因组学方法来确定配体. 这种确定配体的方法对于了解蛋白质的功能来说很重要. 蛋白质结构数据通常仅携带其生化功能信息, 它们在细胞或有机体中的生物学角色更加复杂, 需要额外的实验信息来阐明它. 然而, 在确定生物功能的研究过程中, 有些蛋白质生化功能的信息会指导选择恰当的实验来对其基于结构的功能进行预测.

蛋白质的功能可以在从生物化学通过细胞到生理功能的不同层次来定义. 蛋白质分子固有的结构对于其功能是绝对必要的, 功能相同的蛋白质有类似的结构. 从结构预测功能非常困难, 即便两个蛋白质结构被发现具有同源性, 结构与功能相似性的关系也不是一目了然的, 并且很可能会被其他很多因素影响 [40]. 由于蛋白质的结构极为复杂、蛋白质的生物体中的功能以及发挥功能的条件也极为复杂, 所以虽然经过了 40 年的研究, 蛋白质结构预测问题以及结构和功能关系问题仍是分子生物学领域的热点问题.

蛋白质在几乎所有的生命过程中都起着至关重要的作用. 它们所负担的生理功能包括 [41~43]:

(1) 酶促催化反应. 几乎所有的生物反应都是酶促催化反应. 由于酶的参与使

生物体内的生物化学反应速度加快了10^6 倍.

(2) 运输与储存. 在生理环境中, 小分子通常由蛋白质携带. 例如, 很多药物分子都是与血浆中的血浆血清蛋白结合.

(3) 调整运动. 肌肉几乎都是蛋白质, 肌肉的收缩由两种蛋白质 (肌动蛋白和肌浆球蛋白) 之间的滑动来调节.

(4) 机械支撑. 皮肤和骨骼都由胶原质强化.

(5) 免疫保护. 抗体是特异地反抗机体中外来物质的蛋白质结构.

(6) 神经冲动的产生和传导. 某些氨基酸是神经传递素, 它可以把点信号从一个细胞传导到另一个细胞.

(7) 生长和分化的控制. 蛋白质可以调节生长控制、细胞分化和 DNA 的表达. 蛋白质阻抑物可以与特异的 DNA 片段结合, 保护表达从而使 DNA 片段产生一定的产物. 另外, 很多控制细胞功能的激素和生长因子都是蛋白质, 如胰岛素和甲状腺素.

人们对于蛋白质空间结构与功能的关系问题已经探索了很长的时间. 到目前为止, 这个问题仍然是生物学领域中的热点问题. 每一种蛋白质都有着特有的生物学功能, 这是由它们特定的空间构象决定的. 因为它们的特定的结构允许它们结合特定的配体分子, 蛋白质多种多样的功能与各种蛋白质特定的空间构象密切相关. 其构象发生改变, 功能活性也随之改变.

根据系统科学的观点, 任何执行特定功能的系统都具有内部有序的结构. 系统的总体结构和功能决定于系统各个部分的结构以及这些子结构的排列顺序. 系统的结构决定了系统的功能. 目前, 人们已经了解了一些蛋白质结构和功能的具体的、零散的联系, 从这些具体的联系中人们可以看出蛋白质的空间结构和它们的功能密不可分: 不同的空间结构对应不同的功能, 反之功能不同的蛋白质其结构一定不同. 然而, 人们的知识也只限于此. 因为人们仍然不能对蛋白质结构和功能的关系进行定量分析. 也就是说人们不能找到一种普遍适用的规律, 依据这种规律可以直接从蛋白质的结构来推断蛋白质的功能, 而不需要借助以往已经确立的已知蛋白质的结构与功能的关系.

2.3 机器学习

2.3.1 机器学习的定义和特点

机器学习技术是运用计算机预测蛋白质结构的重要方法, 包括了隐马尔可夫模型、贝叶斯网络、人工神经网络、遗传算法和支持向量机等方法. 20 世纪 80 年代初, 计算机开始应用于生物学中的大规模数据计算 [44]. 从那时起实验生物学家开

始用计算方法对复杂的生物学问题进行建模并逐渐开始与其他领域的科学家 (如计算机学家、物理学家、数学家和晶体学家进行合作). 当时人们已经意识到计算机技术模拟和分析生物学数据的重要性和潜在价值. 第一代生物信息学家运用传统的计算机科学算法开发了计算机程序来分析数据. 然而, 一方面这些利用传统算法开发的计算机程序并不能很切合实际地解决实验中遇到的问题, 其原因主要在于生物系统的复杂性以及当时缺乏分子水平上的基础理论作为指导; 另一方面, 传统的数据处理方法不能有效处理大量的、快速膨胀的数据. 机器学习方法被应用于生物信息学领域以后, 上述困难逐渐开始被克服. 基于机器学习方法编制的程序可以从已知数据中自动地学习并且产生有用的假设, 因此机器学习方法理所当然地成了目前生物信息学领域中比较常用的方法.

牛津英语字典 (The Oxford English Dictionary, OED) 对机器学习的定义为: 计算机从经验中学习的能力, 即通过新获得的知识修改机器自身处理问题的程序. Mitchell 把机器学习定义为引起系统随经验改善的过程, 依据一定的算法利用数据进行学习是机器学习方法的本质. 机器学习可以定义为 "能够通过计算机从有关分类任务的经验进行学习的计算机程序或算法"[45].

学习是智能的本质. 如果一个系统能够学习并通过经验获取知识、自动地改善性能, 那么它就是一个能处理复杂问题的系统, 如生物系统、先进的工具等. 通过从特殊的训练样本产生一般的函数是机器学习的核心思想 [46,47]. 机器学习过程中的学习媒介是事先确定了的训练数据和检测样本, 这些训练数据和检测样本是学习和定义学习机的前提. 学习机必须通过假设空间中的向量来训练, 并且由指定的检测集鉴别挑选出最优假设.

机器学习中所涉及的学习种类主要包括三种:

(1) 监视学习, 学习机的各个运行阶段的输入和输出都可以观察;

(2) 加强学习, 在学习的过程中可以对学习机的行为进行评估, 但是不能指出正确的行为;

(3) 非监视学习, 对学习机的行为既不能观察也不能评估.

在数据库中杂乱无章的数据中寻找存在的规律和发现数据之间必然联系的机器学习方法称作知识获取. 应用这种方法的前提条件是杂乱无章的数据中必须包含有用的信息和知识. 以往对于数据进行总结和归纳、剔除掺杂在数据中的噪声、提取其中有用的知识, 依靠的是具有某个领域内专门技能和知识的专家. 目前利用这种传统的方法对生物数据库中数据的分析却遇到了极大的困难, 因为对这么大量数据的分析所耗费的人力和物力是人们承担不起的 [48~50].

构造机器学习的出发点在于设计能够类似人类一样通过以往的经验学习并能从已知数据发现新知识的机器. 机器学习的这个特点很适应于生物信息学. 首先因为生物信息学研究的主要内容寓于高度复杂的生物系统, 另外更重要的是分子生物

学研究的理论强烈依赖于实验数据. 实际上机器学习最早应用的领域就是分子生物学 [51].

机器学习技术在生物信息学中另一个受欢迎的原因就是它们以问题导向为原则. 学习机能够根据实际情况修改自身结构、适应当前数据环境来更好地解决实际问题. 人们使用机器学习方法寻找数据所蕴涵规律的前提是人们首先能够理解这些技术产生的理论背景和运行规律. 很多生物系统内的规律都必须通过实例或经验数据才能定义, 如蛋白质折叠机制. 人们可以人为地指定这类问题的输入和输出, 但是并不清楚决定这种输入和输出的内在规律. 机器学习的优越性在于能够通过学习机的学习自动调整它们自身的内在结构来得出近似结果.

机器学习的第三个优势在于它们能够很轻易地适应新环境. 这种优势对于分子生物学研究尤其重要. 分子生物学的研究每天都要产生大量新数据, 这些新数据会更新前面研究总结出的概念和结论. 那么用来分析分子生物学数据的工具必须能够随时修改自身的结构以适应产生的新数据, 并可以通过这些新数据获得新知识、产生新假设.

机器学习技术大体包括两类. 一类是数据生成方法, 如隐马尔可夫模型和贝叶斯网络; 另一类是数据判断的方法, 如人工神经网络、遗传算法和支持向量机. 生物信息学中不同类别机器学习方法的选择依赖于学习目标和执行的任务. 在生物信息学中使用正确的方法会改善假说的不确定性并且使发现的知识更可靠.

数据冗余和污染在生物数据中是司空见惯的事情. 生物信息学数据库中的数据一般都是由科研人员通过互联网递交的, 在数据的积累和通过互联网递交到公共数据库的过程中, 检测数据的错误和度量数据的质量都很困难. 多数使用这些数据的生物信息学研究人员没有考虑到数据源的来源和质量. 这些受到污染的生物学数据会影响计算机程序运算结果的精确性. 生物学数据的污染可能由以下原因引起:

(1) 实验错误;

(2) 生物学家的错误解释;

(3) 注释过程中的人为错误;

(4) 实验中使用的非标准技术.

生物学研究是高度数据驱动的, 绝大多数的生物学假设都必须有实验数据作为依据. 虽然机器学习技术具有鲁棒性, 可以在受到污染的数据中进行学习并得到有价值的结论, 但是高质量的数据可以更好地发挥机器学习的效能、得到更为客观的结论. 因此在运用机器学习方法处理生物学问题时应该跟实验生物学家很好沟通以便获取高质量的数据. 数据库中的数据也需要常常校对以保持数据具有较高的质量.

2.3.2 基本的机器学习模型

一般来说, 学习模型包含 4 个主要组件 [44]. 这些组件是:

(1) 学习要素, 改善学习机性能部件;

(2) 执行要素, 选择学习机行为的部件;

(3) 评价要素, 学习机的监视部件;

(4) 问题发生器, 产生新知识的部件.

2.3.3　机器学习方法分类

机器学习方法在生物信息学研究中主要从事在已知的各种生物数据库中发现知识并且把所发现的知识以人们能够理解的方式表达出来. 即,

(1) 分类：预测数据的类别.

(2) 描述：描述数据的类别.

(3) 聚类：规类数据.

(4) 联系分析：寻找关系和联系.

(5) 预测：预测参数的值.

(6) 检测偏差：发现数据的变化.

(7) 可视化：以方便人们观察的方式表现数据.

这些方法大体可以分为两类：一类是数据生成的方法, 包括分类、描述、聚类和可视化; 另一类是数据检测的方法, 包括联系分析、预测和检测偏差.

2.3.4　应用于生物信息学领域的机器学习方法

2.3.4.1　人工神经网络

人工神经网络的灵感来自人脑的生物神经网络的发现. 神经元是大脑执行其功能的独立单位, 它们可以把信息传导给复杂神经网络中的其他神经元. 计算机学家根据大脑工作的原理设计出了执行运算任务的平台和网络, 这种执行运算任务的网络中的各个节点类似于大脑中的神经元 [52]. 在人们可以理解和模拟大脑处理信息的过程后, 科学家着手研究人工神经网络. 经过几年的发展, 人工神经网络技术逐步走向成熟并开始应用于解释生物信息学中遇到的实际问题.

神经网络是由很多节点构成的网状结构, 网状结构中每一个节点都可以被赋予数值. 模式之间的转换依赖于所有链接在一起的节点和简单的信息通过算法. 每一个节点都可以看成是一个统计处理器, 节点的决策依赖于已知数据的概率假设. 人工神经网络利用控制节点的数值和权重来执行对问题的学习和分类.

人工神经网络由相互链接在一起的多层节点构成. 网络中一般包含三个层次：输入层、输出层以及它们之间的隐含层. 由于人工神经网络中内部节点的组织形式不同, 人工神经网络可以分成不同的结构类型, 比如前馈结构、循环结构和层次结构.

人工神经网络是目前生物信息学中应用最广泛的机器学习方法, 它也是生物分析领域应用最早的机器学习技术 [51]. 虽然人工神经网络具有复杂的统计学模型, 但是这种模型非常灵活多变, 善于处理离散值和向量值样本. 另外, 人工神经网络具有鲁棒性, 即对噪音的不敏感性. 这种特征在分析受噪音污染的数据时很有用. 然而, 统计学模型的复杂性同时也对人工神经网络处理问题带来一定的负面影响. 人工神经网络的另外一个缺陷是它缺乏解释能力. 很难运用解释网络中每一个节点的决策和方法来判断网络是否可行. 人工神经网络在蛋白质结构和功能预测及蛋白质分类方面用处很广泛 [53,54].

2.3.4.2 决策树

决策树也被称作分类树和回归树 [55], 这种机器学习方法由 Quinlan 最早开发. 决策树是利用近似离散值函数分类和评估方法的一种感应学习系统. 它具有结构简单、操作方便的特点, 是一种得到广泛应用的机器学习方法.

Divide-and-conquer 策略是决策树方法解决问题的基本方法 [56]. 根据这种策略, 构建决策树需要预先选择样本集, 样本集首先被赋予一系列属性, 决策树通过返回 "是" 或 "否" 的决策来响应样本的检测. 同时样本集中每一类的样本发生的概率也可以从决策树的相应节点上得到. 决策树是一个具有根、茎和叶的树状结构, 与自然界中的树不同的是决策树是倒置的. 枝干是决策树的节点, 每一个节点用来测试样本集中样本的一种属性. 从节点引出的枝干标记测试节点的可能输出项. 如果所有样本都属于同一类, 那么决策树就是一片叶子. 否则, 决策树就会延伸出更多的枝干来测试样本. 决策树的每一片叶子都是一个输入样本的布尔分类器, 节点是这些样本的监测器.

决策树的优势在于它结构简单、操作方便、具有抗噪的鲁棒性以及能清晰地表达学习的结果. 但是决策树方法对于数据的过适应没有好的预防方法, 同时也不能很好地解决各种类别之间的重叠问题, 另外决策树还具有很难优化的缺点.

2.3.4.3 贝叶斯网络

概率是事件的置信度, 贝叶斯网络或贝叶斯信心网络 (BBN) 是一系列相关变量之间概率关系的图形模型 [57]. 贝叶斯网络是由一系列相关变量的独立条件声明编码的网络结构和一系列独立变量的局部概率分布组成, 整合这两个系列产生了相关变量的联合概率分布. 统计学模型结合进贝叶斯网络后, 图形模型能够根据由相关变量之间相互关系得到的概率做出最好的决策. 网络中使用权重概率可以帮助支持假设. 贝叶斯网络提供了一个处理数据分析的通用的方法. 贝叶斯网络的优势在于它能够操作不完整的数据集, 并能够学习和预测缺少的数据. 另外当联系背景知识和数据时, 贝叶斯网络是一个理想的数据表示法. 与其他的机器学习方法相比, 它们可以提供一个标准的优化方法. 对于计算的复杂性来说, 这些网络规避了数据

的过适应. 贝叶斯理论表明：主观的信念应当遵循概率原则, 正确的归纳应以独立的方式推理并通过贝叶斯网络进行传播 [45]. 贝叶斯网络曾经用于 DNA 序列结合位点的建模 [58] 和蛋白质二级结构的预测 [59].

2.3.4.4　遗传算法

遗传算法是 Holland 受到生物进化理论的启发而研究成功的机器学习技术. 遗传算法的主要观点是描述维持问题候选答案的数据结构群落, 利用控制变量来改善学习系统性能的竞争来进行进化, 数据群落通过再结合和突变过程来适应新的环境. 备选答案的最终目的是变成环境中最优化的解决方法 [60] .

遗传算法从群落遗传学的观点来说是宽松的. 利用遗传算法解决问题时, 首先考虑到要解决问题的相关环境. 从环境中随机选择一些样本对其进行二进制编码作为备选数据. 由于这些样本自身性质的不同, 其中包含一些更适应环境的样本, 即更好的结果. 每个循环后样本都会进化. 依据各个个体的生存能力不同, 选择新样本的标准也不尽相同. 在保留了比较成功的个体、删除了不太成功的个体后新的候选样本就产生了. 在处理的过程中依赖突变 (随机的二进制数码变化)、交叉 (相应的子字段的交换) 和在进化循环中的其他字段变化, 进化循环持续进行直到产生了理想的结果 (最高的适应值). 最终个体集合保留了从上一代遗留的最好特征并呈现了最适应的解. 遗传算法在用来发现和解决高维空间中的复杂问题时简单易用, 并且对于不同环境具有鲁棒性. 其不足之处在于在进化过程中不是动态的 [61].

2.3.4.5　隐马尔可夫模型

隐马尔可夫模型是强有力的识别算法, 它是由马尔可夫链演变而来的、用于描述随机过程统计特征的概率模型. 隐马尔可夫模型之所以加上 “隐” 字, 是由于人们不能直接观察到马尔可夫模型处于哪种状态, 只能观察到由那种状态产生的观察矢量. 隐马尔可夫模型及其扩展模型在 “多序列隐藏模式” 的发现方面取得了很好的效果. 典型的隐马尔可夫模型是一种具有匹配、插入和删除节点的节点链, 每一个节点间的状态转换、插入或匹配节点的特征都被赋予一定的概率值. 通过隐马尔可夫链的最佳路径与从开始到结束所遍历的插入或匹配的节点路径是相应的. 隐马尔可夫模型识别系统之所以优于样本匹配系统在于隐马尔可夫模型中保留了更多训练数据的统计信息.

在 20 世纪 90 年代初, 这种模型开始应用于生物信息学的研究中. 从那时起隐马尔可夫模型常用于系列模型化、多重比对和蛋白质结构预测 [62].

基因序列工程所面临的重要任务是确定新蛋白质的功能和结构特征. 未知蛋白质的结构和功能特征可以通过与已知的同源蛋白质的结构和功能进行比对推断出来. 隐马尔可夫模型在同源蛋白质结构比对中得到了广泛的应用.

2.3.4.6 聚类

"聚类" 的基本思想是对结构化数据进行归类, 即把相似度高的样本归为一类. 由于不同类的样本其性质有着明显的差异, 人们可以根据一定的方法根据这些不同点把不同的样本划分成不同的类别. 使用聚类方法所依据的唯一数据就是各个样本点的坐标, 除了各个样本点的数值坐标之外, 不需要任何其他的先验知识. 聚类的方法多种多样, 但是使用聚类方法所需要解决的核心问题有两个: 第一, 样本的相似性度量问题; 第二, 聚类准则问题. 样本相似性度量就是对两个样本间的相似性达到什么程度给出一个量化的指标. 常用的相似性度量有距离、相关系数和夹角方向余弦三种. 聚类方法大致可分为两类: 一类是启发式方法, 根据经验直观地确定一些准则; 另一类是最优化技术, 根据聚类问题的实际背景确定聚类的目标函数. 这样一来, 聚类问题就转化成了优化问题, 从而可用成熟的、经典的优化方法处理聚类问题. 聚类算法可分为两类: 第一, 基于概率密度函数估计的直接方法; 第二, 基于样本间相似度量的间接聚类方法.

聚类是一种探索的方法, 这种方法可以用来组织、鉴别数据. 改良的聚类算法可以用于预测和解释复杂的数据. 等级聚类和 k 聚类是聚类算法的两个主要类型. 等级聚类方法中, 输入数据被聚集成不同层次的丛; k 聚类方法中, 每一个输入目标都根据数据集的性质归入某一类.

有两种聚类观测数据的方法: 第一种基于物理化学理论; 第二种基于计算方法和数据的统计分析. 给定一个数据集, 聚类媒介根据数据的性质把数据归类到较小的类中. 这样, 聚类就是一种描述和表达的方法, 输出的结果可以很容易地被别人理解 [63].

2.3.4.7 支持向量机

支持向量机是由 Vapnik 和他的同事共同开发的基于统计学习理论和 VC 维理论的结构风险最小化原则的机器学习方法. 由于它的卓越功能使得它已经成为机器学习和数据挖掘的标准工具之一 [64]. 支持向量机实现的是如下的思想: 通过某种事先选择的非线性映射将输入向量 x 映射到一个高维空间中构造最优分类的超平面. 支持向量机在处理训练样本时像人工神经网络一样是一个 "黑箱" 算法 [65]. 支持向量机的主要思想是使用超平面来分类不同性质的数据. 对于线性不可分的数据, 首先通过核函数把它们映射到高维特征空间. 在高维特征空间中, 这些数据被线性分类, 那么这些数据就在原输入空间中非线性分类 [47,66]. 支持向量机的劣势在于训练和检测过程中花费的计算成本过高, 并且缺乏表达能力 [67].

另外除了以上阐述的单一方法以外, 解决生物信息学问题的工具中还包括了多重机器学习技术. 也许这是因为把不同的方法组合在一起可以弥补单一方法的弱点, 从而得到更好的学习结果. 虽然多重方法的执行结果好于单一方法的执行结果,

但是怎样组合所要使用的方法并不是一件简单的事情. 这是因为各种机器学习方法的学习过程和所输出的结果不尽相同. 一个成功组合多种机器学习方法的实例是隐神经网络 [68]. 隐神经网络是隐马尔可夫模型和神经网络的组合. 不同的机器学习方法组合在一起可能极大地拓展了机器学习方法在生物信息学中的应用范围.

第 3 章　统计学习理论

3.1　学习问题的表示方法

3.1.1　概述

统计学习理论是 Vapnik 等在 20 世纪 70 年代末提出, 并于 90 年代逐渐完善的一种针对小样本的机器学习理论. 在此理论上构造的支持向量机已经成为构造预测规则的通用方法 [46,47]. 该理论认为根据不同科学领域所描述规律的复杂性不同, 用少数几个变量可以描述的科学领域称为简单世界, 而必须用多个变量才能描述的科学领域称为复杂世界. 建立在两个世界中的推理方法是不一样的, 其中简单世界的推理方法为 [46]:

(1) 演绎, 即由一般到特殊的过程;

(2) 归纳, 即由特殊到一般的过程.

这两种推理过程的数学表示就是概率论和数理统计. 概率论是演绎的数学理论体系, 而数理统计是归纳的数学理论体系. 在复杂世界中这种推理方法是不适用的, 因为复杂世界中的问题很多是不适定的, 即在现实中就会出现这样一种情况: 当人们反演问题的因果关系时, 由于结果的轻微变化会导致对原因的反演很可能与客观现实相去甚远. 这种问题称为不适定问题, 这种问题可以通过正则化技术解决. 面对这种情况, Vapnik 等认为解决复杂世界中的问题的推理方法应该为 "归纳" 和 "转导 —— 从特殊到特殊" 的推理. 而在解决具体问题时, 要避免把解决一个更为一般的问题作为其中间步骤. 它的核心问题是寻找一种归纳原则以实现最小化风险泛函, 从而实现最佳的推广能力. 该理论研究从一些观测数据出发得出目前尚不能从原理分析或实验得到的规律, 并利用这些规律去分析客观对象以实现对未来数据和无法观测的数据进行预测.

在统计学习理论中, 把学习问题看作是利用有限数量的观测来寻找待求的依赖关系的问题. 从实例中学习就是运用了 "转导" 这种推理原则. 从实例学习的方法非常类似老师教小学生认字, 老师并没有描述每个字的精确结构, 而是给他们一些具体文字的例子. 通过仔细观察了解这些文字的一些特征以后, 学生不但能识别印刷体的文字, 还能识别手写体的文字 [47].

以往机器学习理论的核心是经验风险最小化归纳原则. 依据这种原则, 如果能找到一个相当逼近这些样本的函数并以大量的样本进行训练, 那么就可能对工作样

本做出较准确的预测. 然而, 如果学习机器能力过强, 能够无误差地适应任意的训练样本, 就会导致它所采用的函数集过于复杂, 产生过学习的现象.

Vapnik 提出了 VC 维的概念 [46], 它是统计学习理论的核心概念. VC 维是描述函数集或学习机器的复杂性或者说是学习能力的一个重要指标.

与经验风险最小化原则不同, 统计学习理论依据结构风险最小化原则进行推理. 结构风险最小化原则定义了给定数据的逼近精度和逼近函数的复杂性之间的一种折中. 该原则首先定义了一种函数集的嵌套结构:

$$S_1 \subset S_2 \subset \cdots \subset S_n \subset \cdots$$

这些函数集的 VC 维从小到大排列, 这样函数集的 VC 维就成为了可控参数. 对一个给定的训练集, 结构风险最小化归纳原则上在使风险上界最小的子集 S_k 中选择使经验风险最小的函数. 结构风险最小化归纳的一般原则可以用不同的方法实现.

学习理论的最终研究目标就是希望找到从样本学习的公式化方法, 遵循这种方法研究者通过一段时间对数据的学习和训练之后, 能够得到一个分类器使它能够对新样本正确分类.

支持向量机 (support vector machine, SVM) 方法建立在统计学习理论的 VC 维理论和结构风险最小原理基础上, 根据有限的样本信息在模型的复杂性 (即对特定训练样本的学习精度) 和学习能力 (即无错误地识别任意样本的能力) 之间寻求最佳折中, 以期获得最好的推广能力. 支持向量机方法的几个主要优点有:

(1) 它是专门针对有限样本情况的, 其目标是得到现有信息下的最优解而不仅仅是样本数趋于无穷大时的最优值;

(2) 算法最终将转化成为一个对偶寻优问题, 从理论上说, 得到的将是全局最优点, 解决了在神经网络方法中无法避免的局部极值问题.

支持向量机方法在解决线性不可分问题的分类时, 首先将实际问题通过非线性变换转换到高维特征空间, 在高维空间中构造线性判别函数来实现原空间中的非线性判别函数, 这种特殊性质能保证机器有较好的推广能力, 同时它巧妙地解决了运算成本随着维数增加而大幅提高, 即维数灾难问题: 其算法复杂度与样本维数无关.

3.1.2　学习问题的一般表示

学习问题是利用有限数量的观测来寻找待求的依赖关系的问题. 描述样本学习的一般模型包括:

(1) 产生器. 产生器产生随机向量 $x \in R_n$, 这些样本是从固定但未知的概率风函数 $F(x)$ 中独立抽取的. 这些样本就构成了学习机的数据. 发生器是源头, 它确定了训练器和学习机器的工作环境.

(2) 监视器. 监视器对于每个输入向量 x 返回一个输出值 y, 产生输出的根据值是同样固定但未知的条件分布函数 $F(y|x)$. 这些向量输入到目标算子 (训练器), 目标算子返回输出值 y. 监视器由样本集和学习机确定, 并反映样本的类别.

(3) 学习机. 学习机能够实现一定的函数集 $f(x,\alpha),\alpha\in\Lambda$, 其中 Λ 是参数集合. 学习机器的目标是构造适当的逼近, 它依据一定的原则对样本进行分类.

学习的问题就是从给定的函数集 $f(x,\alpha),\alpha\in\Lambda$ 中选择出能够最好地逼近训练器的相应函数. 这种选择是基于训练集的, 训练集由根据联合分布 $F(x,y)=F(x)F(y|x)$ 抽取出的 l 个独立同分布 (i.i.d) 观测数据 $(x_1,y_1),\cdots,(x_l,y_l)$ 构成.

构建学习模型的目的在于通过训练找到学习机, 并用它对未知样本点进行分类, 监视器反映分类结果. 为了计算方便, 在允许的精度下数据集中的样本 (向量) $x\in R^n$ 认为是以未知概率 $P(x)$ 独立同分布地产生的. 监视器根据条件分布函数 $F(y|x)$ 确定样本的实际类别, 对于每一个样本 x 都联系着一个监视器的值 y.

在学习理论中, 这种以 "样本–监视器" 对的形式给出的样本学习方式称作监督学习, 监视器反映样本的类别. 那些具有 "样本–监视器" 功能性, 即监视器已知并利用监视器值调整学习机的样本称作训练数据. "样本–监视器" 数据对反映了输入域与输出域的函数映射关系, 这个从样本集到监视器输出集的函数称为目标函数. 通过学习得到的目标函数的估计值, 即算法的输出值称为学习问题的解. 在分类问题中这个函数有时称作决策函数.

学习的目的就是输出一个对训练数据集进行正确分类的假设和一个用来正确适应数据的学习算法. 数据集一般需要分成两个集合, 其中一个称作训练集, 用来构建学习模型. 另一个称作检测集, 用来检测学习模型的工作效能.

以 l 个观测样本的训练集

$$\{X=(x_1,y_i),\cdots,(x_l,y_l),\quad x_i\in R^n,\quad y_i\in\{+1,-1\}\}$$

为基础, 选择参数 α 的过程称作向量机训练. 训练集 X 由 $P(x)$ 确定, 能够得到的信息都包含在训练集 X 中. 另外一些用来检测向量机对新数据分类能力的数据集称作检验集. 分类错误用于检测机器整体错误率.

3.1.3 学习问题的模型

学习问题的形式多种多样, 包含了很多特殊问题. 这些特殊问题可以一般地表示如下: 设有定义在空间 Z 上的概率测度 $F(z)$. 考虑函数的集合 $Q(z,\alpha),\alpha\in\Lambda$. 学习目标是最小化风险泛函

$$R(\alpha)=\int Q(z,\alpha)\,\mathrm{d}F(z),\quad \alpha\in\Lambda \tag{3.1}$$

其中概率测度 $F(z)$ 未知, 但给定了一定的独立同分布样本

$$z_1, \cdots, z_l \tag{3.2}$$

这种一般问题就是在经验数据 (3.2) 式基础上最小化风险泛函 (3.1) 式, 其中 z 代表了数据对 (x, y). 风险泛函 (3.1) 式是定义在分布域上关于损失函数的不定积分, 简称风险. 其中 $Q(z, \alpha) = y - f(x, \alpha)$ 称作损失, 在二元分类中它的值只可能取 0 或 1.

评价一个学习模型性能优劣的定性标准之一就是检验这个学习机错误分类的概率, 即监视器的值 y 与学习机的预测值 $f(x,\alpha)$ 的差异. 构建风险函数的目的是要找到参数 α^* 以确定整个函数集 $f(x,\alpha)$ 的最小风险 $f(x,\alpha^*)$, 其中 $\alpha \in \Lambda$. 但问题在于 $P(x)$ 的值未知, 不能直接由积分计算得到学习机分类错误值, 所以只有通过最小风险 $f(x,\alpha^*)$ 对学习机分类错误水平进行估计.

3.1.4　经验风险最小化原则

经验风险最小化归纳原则:

(1) 把风险泛函 $R(\alpha)$ 替换为经验风险泛函

$$R_{\text{emp}}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} Q(z_i, \alpha) \tag{3.3}$$

是在训练集 (3.2) 上得到的.

(2) 使用经验风险泛函 (3.3) 式最小的函数 $Q(z, \alpha_l)$ 逼近使风险 (3.1) 式最小的函数 $Q(z, \alpha_0)$.

理论风险函数是积分的形式, 而因为以往的经验是有限的, 所以经验风险的形式是求和的形式. 通过对 (3.3) 式的观察可以看出, 风险函数 $R_{\text{emp}}(\alpha)$ 与损失的平均值有关. 由于公式并不涉及概率分布, 对于特定的 α 和训练集 (3.2), $R_{\text{emp}}(\alpha)$ 可以唯一确定. 因此在训练集 (3.2) 的基础上, 如果用经验风险函数 $R_{\text{emp}}(\alpha)$ 代替理论风险函数 $R(\alpha)$, 那么理论风险函数的最小值 $f(x,\alpha^*)$ 可以作为经验风险函数 $f(x,\alpha^x)$ 的近似估计值.

经验风险最小化原则应用非常普遍. 改变损失函数, 经验风险最小化原则可以用在最小二回归估计和稠密性估计的最大似然估计中. 如果当 $l \to \infty$ 时, $R(\alpha^x)$ 和 $R_{\text{emp}}(\alpha^x)$ 都收敛于风险函数值的下确界 $\inf_{\alpha\in\Lambda} R(\alpha)$ 那么这个学习过程就称作同一化.

3.1.5　复杂性和推广能力

根据人们对机器学习研究的经验, 最小的训练误差不一定产生最好的预测效果. 学习机器对未来输出进行正确预测的能力称为推广性. 以往的学习机器常遇到

过学习的情况, 之所以出现这种情况是因为: ① 学习样本太少; ② 学习机器设计不合理. 如果用一个过于复杂的模型进行拟合有限样本时, 常导致过学习使模型丧失推广能力. 这是有限样本下学习机器的复杂性与推广能力之间的矛盾 [33]. 那么在样本有限情况下:

(1) 经验风险最小并不一定意味着期望风险最小;

(2) 学习机器的复杂性不但与所研究的系统有关, 而且要和有限的学习样本相适应.

3.1.6 模式识别问题

如果存在两类数据, 它们的分布函数服从两个不同的统计规律 $p_1(x,\alpha^*)$ 和 $p_2(x,\alpha^*)$. 若第一类数据出现的概率是 q_1, 第二类出现的概率是 $1-q_1$. 那么模式识别问题就是寻找一个决策规则使错误的概率最小, 即若向量 x 属于第一类的概率不小于它属于第二类的概率, 决策规则就认为这个向量属于第一类. 用不等式表示就是

$$q_1 p_1(x,\alpha^*) \geqslant (1-q_1)\, p_2(x,\beta^*)$$

这个决策规则可以表示成下面的等价形式:

$$f(x) = \operatorname{sign}\left\{\ln p_1(x,\alpha^*) - \ln p_2(x,\beta^*) + \ln\frac{q_1}{1-q_1}\right\}$$

称作判别函数, 使用这个判别函数的前提是必须估计概率密度$p_1(x,\alpha^*)$和$p_2(x,\beta^*)$.

3.2 统计学习理论的四个部分

统计学习理论的四个部分包括: ① 学习过程的一致性理论; ② 学习过程收敛速度的非渐进理论; ③ 控制学习过程的推广能力的理论; ④ 构造学习算法的理论. 前三个理论说明了为什么支持向量机是合理的, 在此基础上后面一个理论说明了怎么构建支持向量机.

3.2.1 学习过程的一致性

3.2.1.1 学习问题的关键定理

阐述这一部分理论的目的是找出学习过程一致性的充分必要条件. 经验风险最小化学习过程一致性的充分必要条件回答了使经验风险最小的学习过程在什么时候能够取得实际风险最小的问题. 学习一致性条件的结论是统计学习理论的基础, 也是该理论与传统渐进统计学的基本联系所在. 学习过程一致性是指当训练样本数目趋于无穷大时, 经验风险的最优值能够收敛到真实风险的最优值. 只有满足一

致性条件, 才能保证在经验风险最小化原则下得到的最优方法, 当样本无穷大时趋近于使期望风险最小的最优结果 [69].

设函数集 $Q(z,\alpha),\alpha\in\Lambda$ 满足条件

$$A\leqslant\int Q(z,\alpha)\,\mathrm{d}F(z)\leqslant B,\quad A\leqslant R(\alpha)\leqslant B \tag{3.4}$$

那么, ERM 原则一致性的充分必要条件是: 经验风险 $R_{\mathrm{emp}}(\alpha)$ 在函数集 $Q(z,\alpha),\alpha\in\Lambda$ 上在如下意义下一致收敛于实际风险 $R(\alpha)$:

$$\lim_{l\to\infty}P\left\{\sup_{\alpha\in\Lambda}\left(R(\alpha)-R_{\mathrm{emp}}(\alpha)\right)>\varepsilon\right\}=0,\quad\forall\varepsilon>0 \tag{3.5}$$

这种一致收敛称为一致单边收敛. 这个定理称为关键性定理.

根据学习问题的关键定理, ERM 原则的一致性等价于 (3.5) 式的一致单边收敛成立.

3.2.1.2　指示函数集的熵与生长函数

指示函数集的熵和生长函数是衡量函数集学习性能的重要指标. 设 $Q(z,\alpha),\alpha\in\Lambda$ 是一个指示函数集, 考虑样本 $z_1,\cdots,z_l$, 定义了一个量 $N^{\Lambda}(z_1,\cdots,z_l)$, 它代表了指示函数集中的函数能把给定的样本分成多少种不同的分类. 这个量表征函数集 $Q(z,\alpha),\alpha\in\Lambda$ 在给定的数据集上的多样性.

若

$$H^{\Lambda}(z_1,\cdots,z_l)=\ln N^{\Lambda}(z_1,\cdots,z_l)$$

那么 $H^{\Lambda}(z_1,\cdots,z_l)$ 称作随机熵. 随机熵在联合分布函数 $F(z_1,\cdots,z_l)$ 上的期望:

$$H^{\Lambda}(l)=E\ln N^{\Lambda}(z_1,\cdots,z_l)$$

称作指示函数集 $Q(z,\alpha),\alpha\in\Lambda$ 在数量为 l 的样本上的熵, 它依赖于函数集 $Q(z,\alpha)$, $\alpha\in\Lambda$, 概率测度一级观测数目 l 反映了给定指示函数集在数目为 l 的样本上的期望的多样性.

3.2.1.3　实函数集的熵

设 $A\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 是一个有界损失函数的集合, 用这个函数集和训练集 $z_1,\cdots,z_l$, 可以构造下面的 l 维向量集合:

$$q(\alpha)=\left(Q(z_1,\alpha),\cdots,Q(z_l,\alpha)\right),\quad\alpha\in\Lambda$$

这个向量集合处在 l 维立方体中, 并且在 C 度量下有一个有限的最小 ε- 网格. 令 $N=N^{\Lambda}(\varepsilon;z_1,\cdots,z_l)$ 是向量集 $q(\alpha),\alpha\in\Lambda$ 的最小 ε- 网格的元素数目. 随机对

数 $H^{\Lambda}(\varepsilon; z_1, \cdots, z_l) = \ln N^{\Lambda}(\varepsilon; z_1, \cdots, z_l)$ 称作函数集 $A \leqslant Q(z, \alpha) \leqslant B, \alpha \in \Lambda$ 在样本 $z_1, \cdots, z_l$ 上的随机 VC 熵. 随机 VC 熵的期望 $H^{\Lambda}(\varepsilon; l) = EH^{\Lambda}(\varepsilon; z_1, \cdots, z_l)$ 称作函数集 $A \leqslant Q(z, \alpha) \leqslant B, \alpha \in \Lambda$ 在数量为 l 的样本上的 VC 熵, 这里的期望是乘积测度 $F(z_1, \cdots, z_l)$ 进行的.

对于指示函数集, $\varepsilon < 1$ 的最小 ε- 网格不依赖于 ε, 且是单位立方体的定点的一个子集. 因此, 对于 ε <1, 有

$$
\begin{aligned}
&N^{\Lambda}(\varepsilon, z_1, \cdots, z_l) = N^{\Lambda}(z_1, \cdots, z_l) \\
&H^{\Lambda}(\varepsilon, z_1, \cdots, z_l) = H^{\Lambda}(z_1, \cdots, z_l) \\
&H^{\Lambda}(\varepsilon, l) = H^{\Lambda}(l)
\end{aligned}
$$

3.2.1.4 一致单边收敛的充分必要条件

考虑有界实函数集 $Q(z, \alpha), \alpha \in \Lambda$ 和一个新的函数集 $Q^*(z, \alpha^*), \alpha^* \in \Lambda^*$, 这个新函数集满足一定的可测性条件: 对于 $Q(z, \alpha), \alpha \in \Lambda$ 中的任意函数, 在 $Q^*(z, \alpha^*), \alpha^* \in \Lambda^*$ 中存在一个函数, 使得

$$
\begin{aligned}
&Q(z, \alpha) - Q^*(z, \alpha^*) \geqslant 0, \quad \forall z \\
&\int (Q(z, \alpha) - Q^*(z, \alpha^*)) \mathrm{d}F(z) \leqslant \delta
\end{aligned} \tag{3.6}
$$

对完全有界函数集 $Q(z, \alpha), \alpha \in \Lambda$, 经验均值一致单边收敛于其期望的充分必要条件是: 对任意正 δ, η 和 ε, 存在一个满足 (3.6) 式的函数集 $Q^*(z, \alpha^*), \alpha^* \in \Lambda^*$ 在 l 个样本上的 ε 熵满足下面的不等式:

$$
\lim \frac{H^{\Lambda^*}(\varepsilon, l)}{l} < \eta \tag{3.7}
$$

有界函数集 $Q(z, \alpha), \alpha \in \Lambda$ 一致单边收敛的充分必要条件是存在与 $Q(z, \alpha), \alpha \in \Lambda$ 非常接近的另一和函数集 $Q^*(z, \alpha^*), \alpha^* \in \Lambda^*$, 对这个新函数集, 条件 (3.7) 式成立.

考虑在 $N^{\Lambda}(z_1, \cdots, z_l)$ 值的基础上构造的两个概念

(1) 退火的 VC 熵

$$
H_{\mathrm{ann}}^{\Lambda}(l) = \ln EN^{\Lambda}(z_1, \cdots, z_l)
$$

(2) 生长函数

$$
G^{\Lambda}(l) = \ln \sup_{z_1, \cdots, z_l} N^{\Lambda}(z_1, \cdots, z_l)
$$

这些概念的定义方法使得对于任何 l, 都有不等式

$$
H^{\Lambda}(l) \leqslant H_{\mathrm{ann}}^{\Lambda}(l) \leqslant G^{\Lambda}(l)
$$

那么等式

$$\lim_{l\to\infty}\frac{H^{\Lambda}(l)}{l}=0$$

描述了 ERM 原则一致性的充分条件.

对于任何 $l>l_0$, 都有下面的指数界成立:

$$P\{R(\alpha_l)-R(\alpha_0)>\varepsilon\}<\mathrm{e}^{-c\varepsilon^2 l}$$

其中 $c>0$ 是某个常数. 那么等式

$$\lim_{l\to\infty}\frac{H^{\Lambda}_{\mathrm{ann}}(l)}{l}=0$$

描述了收敛速度快的充分条件.

等式

$$\lim_{l\to\infty}\frac{G^{\Lambda}(l)}{l}=0$$

描述了履行 ERM 原则的学习机器有快的收敛速度的充分必要条件.

3.2.2　学习过程收敛速度的界

学习过程收敛速度的界的一系列理论是经验风险和实际风险之间关系的重要结论, 这一部分学习理论讨论学习机器推广性的非渐进界, 并讨论如何找到构造性的界与分布无关的界的方法 [44].

3.2.2.1　生长函数与函数集的 VC 维

任何生长函数或者满足等式

$$G^{\Lambda}(l)=l\ln 2$$

或者受下面的不等式约束:

$$G^{\Lambda}(l)\leqslant h\left(\ln\frac{l}{h}+1\right)$$

其中 h 是一个整数, 使得当 $l=h$ 时, 有

$$G^{\Lambda}(h)=h\ln 2$$
$$G^{\Lambda}(h+1)<(h+1)\ln 2$$

即生长函数要么是线性的, 要么以一个对数函数为上界.

指示函数集 $Q(z,\alpha),\alpha\in\Lambda$ 的 VC 维, 是能够被集合中的函数以所有可能的 2^h 种可能方式分成两类大向量 $z_1,\cdots,z_h$ 的最大数目 h(也就是能够被这个函数集打

散的向量的最大数目). 如果对任意的 n, 总存在一个 n 个向量的集合可以被函数集 $Q(z,\alpha),\alpha\in\Lambda$ 打散, 那么函数集的 VC 维就是无穷大.

设 $A\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 是一个以常数 A 和 B 为界的实函数集合 (A 可以是 $-\infty$, B 可以是 ∞). 与实函数集合 $A\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 一起考虑其指示函数集

$$I(z,\alpha,\beta)=\theta\{Q(z,\alpha)-\beta\},\quad \alpha\in\Lambda,\quad \beta\in(A,B) \tag{3.8}$$

其中, $\theta(z)$ 是阶跃函数

$$\theta(z)=\begin{cases}0, & z<0\\ 1, & z\geqslant 0\end{cases}$$

实函数集合 $A\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 的 VC 维定义为相应的指示函数集 (3.8) 式的 VC 维, 其中的参数 $\alpha\in\Lambda,\beta\in(A,B)$.

3.2.2.2 构造性的与分布无关的界

构造性表达式

$$E=4\frac{h\left(\ln\frac{2l}{h}+1\right)-\ln\left(\frac{\eta}{4}\right)}{l} \tag{3.9}$$

与表达式

$$E=2\frac{\ln N-\ln\eta}{l} \tag{3.10}$$

有下面的构造性界成立, 其中在 VC 维有限的情况下使用 (3.9) 式的 E, 而在集合中函数数目有限的情况下使用 (3.10) 式的 E.

1) 完全有界函数集

设 $A\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 是完全有界函数的集合, 那么:

(A) 下面的不等式以至少 $1-\eta$ 的概率同时对 $Q(z,\alpha),\alpha\in\Lambda$ 所有函数 (包括是经验风险最小的函数) 成立:

$$R(\alpha)\leqslant R_{\mathrm{emp}}(\alpha)+\frac{(B-A)}{2}\sqrt{E}$$
$$R(\alpha)\geqslant R_{\mathrm{emp}}(\alpha)-\frac{(B-A)}{2}\sqrt{E}$$

(B) 下面的不等式以至少 $1-2\eta$ 的概率使经验风险最小的函数 $Q(z,\alpha_l)$ 成立

$$R(\alpha_l)-\inf_{\alpha\in\Lambda}R(\alpha)\leqslant(B-A)\sqrt{\frac{-\ln\eta}{2l}}+\frac{(B-A)}{2}\sqrt{E}$$

2) 完全有界非负函数集

设 $0\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda$ 是完全有界非负函数的集合, 那么:

(A) 下面的不等式以至少 $1-\eta$ 的概率同时对 $Q(z,\alpha) \leqslant B, \alpha \in \Lambda$ 的所有函数 (包括是经验风险最小的函数) 成立

$$R(\alpha) \leqslant R_{\text{emp}}(\alpha) + \frac{BE}{2}\left(1+\sqrt{1+\frac{4R_{\text{emp}}(\alpha)}{BE}}\right) \tag{3.11}$$

(B) 下面的不等式以至少 $1-2\eta$ 的概率使经验风险最小的函数 $Q(z,\alpha_l)$ 成立

$$R(\alpha_l) - \ln \underset{\alpha\in\Lambda}{f} R(\alpha) \leqslant B\sqrt{\frac{-\ln\eta}{2l}} + \frac{BE}{2}\left(1+\sqrt{1+\frac{4}{E}}\right)$$

3) 完全非负函数集

设 $Q(z,\alpha) \geqslant 0, \alpha \in \Lambda$ 是完全有界非负函数的集合, 那么:

(A) 下面的不等式以至少 $1-\eta$ 的概率同时对满足:

$$\sup_{\alpha\in\Lambda} \frac{\left(\int Q^p(z,\alpha)\,\mathrm{d}F(z)\right)^{1/p}}{\int Q(z,\alpha)\,\mathrm{d}F(z)} \leqslant \tau < \infty$$

的所有函数成立

$$R(\alpha) \leqslant \frac{R_{\text{emp}}(\alpha)}{\left(1-a(p)\tau\sqrt{E}\right)_+} \tag{3.12}$$

其中

$$(u)_+ = \max(u,0)$$
$$a(p) = \sqrt[p]{\frac{1}{2}\left(\frac{p-1}{p-2}\right)^{p-1}}$$

(B) 下面的不等式以至少 $1-2\eta$ 的概率使经验风险最小的函数 $Q(z,\alpha_l)$ 成立

$$\frac{R(\alpha_l) - \ln \underset{\alpha\in\Lambda}{f} R(\alpha)}{\ln \underset{\alpha\in\Lambda}{f} R(\alpha)} \leqslant \frac{\tau a(p)\sqrt{E}}{\left(1-\tau(p)\sqrt{E}\right)_+} + O\left(\frac{1}{l}\right)$$

这些关于学习机器一致收敛和收敛速度的一系列条件有重要的理论意义, 但在实践中一般无法直接应用.

3.2.3　控制学习过程推广能力的理论

所谓推广能力是指超平面对不属于训练集的新数据正确分类的能力. 从数学角度来说推广能力能通过风险函数来表达. 风险函数就是学习机的输出值与实际值的期望差异. 因为风险不能直接计算出, 但它可以有风险的上界估计. 训练集的经验

风险可以直接计算出. 容量是描述向量机对于任何训练集无错误的学习能力的量. 一个向量机的容量过大会导致过度适应, 过小又会发生分类错误 [43,44].

推广能力理论依据原理不同有很多名字, 统计学中称之为一致收敛速率研究, 计算机科学中称之为正确近似概率. 模型中的一个重要假设就是训练和检测中的数据都是依据未知但固定的分布独立同分布产生. 假设这是一种输入输出对 $(x,y)\in X\in(-1,+1)$ 的分布, 那么输出值 y 决定于定义在输入域的目标函数 t, 即 $y=t(x)$. 模型的适应过程中考虑到了样本的动态性、非完全独立分布, 但模型忽视了学习机可能影响样本的选择. 为了简便起见把样本都视作独立同分布的情况.

经验风险最小化原则是从处理大样本数问题出发的, 这一原则的合理性可以通过考虑不等式 (3.11) 式和 (3.12) 式来证明. 当 l/h 较大时, E 就较小, 因此不等式 (3.11) 右边的第二项就变得较小, 于是实际风险就接近经验风险的取值. 在这种情况下, 较小的经验风险值就能保证期望风险的值也较小. 如果 l/h 较小, 那么一个小的 $R_{\mathrm{emp}}(\alpha_1)$ 并不能保证小的实际风险值. 在这种情况下, 要最小化实际风险 $R(\alpha)$, 必须对不等式 (3.11) 右边两项同时最小化, 但是需要注意, 不等式 (3.11) 右边的第一项取决于函数集中的一个特定的函数, 而第二项则取决于整个函数集的 VC 维. 因此要对风险的界 (3.11) 式右边两项同时最小化, 必须使 VC 维成为一个可以控制的变量. 结构风险最小化归纳原则旨在针对经验风险和置信范围这两项最小化风险泛函.

对于 $R_{\mathrm{emp}}(\alpha)$=0 的训练机 $f(x,\alpha^*)$ 考虑分类器

$$f'(x,\alpha^*)=\begin{cases} f(x,\alpha^*), & x\in X \\ -f(x,\alpha^*), & x\notin X \end{cases}$$

在分类数据集 X 中的样本时 f' 与 f 的作用相同, 但在分类不属于数据集 X 中的新样本时 f' 与 f 的作用相反. 因此学习机中函数的选择似乎受到了限制.

同样的问题称作过适应, 如果一个学习机是一个高度灵活函数的富集, 称作有高容量. 如果机器容量很高, 能无错误的学习任何训练数据集, 这样它就有可能在面对新的输入数据时不能很好的工作. 这些机器过于适合训练数据, 以至于不能识别训练数据普遍的特性. 因此最后的推广运算是在描述训练集的精确性和机器的容量中寻找一种平衡.

通过限制风险可以把这个想法公式化: 如果 α 的概率是 $1-\eta$, 那么

$$R(\alpha)\leqslant R_{\mathrm{emp}}(\alpha)+\sqrt{\frac{1}{2}\left(h\left(1+\ln\frac{2l}{h}\right)+\ln\frac{4}{\eta}\right)}+\frac{1}{l} \tag{3.13}$$

这里 h 是 VC 维, 它是容量的度量单位. 不等式的左边一般无法计算, 但是如果VC

维已知, 风险的上界可以很容易地由不等式右边计算得到. $\ln\dfrac{2l}{h}$ 是单调增加的, 它被称作 VC 置信.

以上公式表明风险不仅依赖函数集 (学习机) 的选择, 还依赖学习算法对函数的选择. 如果使用低 VC 维的学习机对数据集作较好的分类, 那么不等式右边会较小. 这是结构风险最小化的基本思想.

从以上定义可以看出结构风险最小化归纳原理是通过选择合适的子集 S_k 来限制实际风险上界以最小化经验风险. 为了最小化经验风险, 可以针对每一个子集训练一个学习机, 由这个学习机序列可以得到经验风险的和 VC 置信最小的一个学习机.

通过选择 h_k 可以保持置信间隔不变来最小化的经验风险, 这种方法首先用于神经网络中来选择合适的结构和消除分类错误. 结构决定了神经网络的适应性也同样决定了 VC 维. 因此结构体系一旦确定, 那么 $\sqrt{\dfrac{1}{2}\left(h\left(1+\ln\dfrac{2l}{h}\right)+\ln\dfrac{4}{\eta}\right)}+\dfrac{1}{l}$ 也就固定了. 另一种方法是保持经验风险固定 (即等于 0) 最小化置信间隔.

设函数 $Q(z,\alpha),\alpha\in\Lambda$ 的集合 S 有一定的结构, 这一结构是有一系列嵌套的函数子集 $S_k=\{Q(z,\alpha),\alpha\in\Lambda\}$ 组成的, 它们满足

$$S_1\subset S_2\subset\cdots\subset S_n\cdots$$

其中, 结构的元素满足下面的两个性质:

(1) 每个函数集 S_k 的 VC 维 h_k 是有限的, 因此

$$h_1\leqslant h_2\leqslant\cdots\leqslant h_n\cdots$$

(2) 结构的任何元素 S_k 或者包含一个完全有界的函数集 $0\leqslant Q(z,\alpha)\leqslant B,\alpha\in\Lambda_k$, 或者包含对一定的 (p,τ_k) 对满足下列不等式的函数集:

$$\sup_{\alpha\in\Lambda}\frac{\left(\int Q^p(z,\alpha)\,\mathrm{d}F(z)\right)^{1/p}}{\int Q(z,\alpha)\,\mathrm{d}F(z)}\leqslant\tau,\quad p>2$$

这个结构成为允许结构. 对于某个训练集, 结构风险最小化归纳原则上选择使风险函数

$$R(\alpha)=\int\frac{1}{2}|y-f(x,\alpha)|\,\mathrm{d}P(x,y)$$

达到最小的子集 S_k.

在结构风险最小化原则下, 一个分类器的设计过程包括以下两个方面的任务:

(1) 选择一个适当的函数子集 (使之对问题来说有最优的分类能力);

(2) 从这个子集中选择一个判别函数 (使之经验风险最小).

第一步相当于模型选择, 第二步相当于在确定了函数形式后的参数估计.

对于一个给定的观测集 $z_1, \cdots, z_h$, 结构风险最小化原则在使保证风险最小化的子集 S_k 中选择使经验风险最小化的函数 $Q\left(z, \alpha_1^k\right)$.

结构风险最小化归纳原则定义了对给定数据逼近的精度和逼近函数复杂性之间的一种折中. 随着子集序号 n 的增加, 经验风险的最小值减小, 但决定置信范围的项却增加. 结构风险最小化归纳原则通过选择子集 S_k 将这两者都考虑在内, 子集 S_k 的选择是使得在这个子集中, 最小化经验风险会得到实际风险的最好的界.

支持向量机通过以下步骤实现结构风险最小化归纳原则 [46]:

(1) 用非线性变换把输入向量映射到一个高维特征空间中;

(2) 在这个空间中, 在线性决策规则集合上按照正规超平面权值的模构造了一个结构;

(3) 选择结构中最好的元素以及这个元素中最好的函数, 以达到最小化错误率的界的目标.

第 4 章　构造支持向量机

这一章阐明了怎么构建支持向量机, 以及构造支持向量机的过程中需要用到的相关二次规划优化方法也略加介绍. 构造学习算法的理论是统计学习理论的重要组成部分, 是蛋白质结构预测的基础.

4.1 优 化 理 论

支持向量机理论中, 由于只涉及损失函数是凸二次函数而约束条件是线性函数的情况, 所以为了节约篇幅这里只对这种情况进行讨论. 解决目标函数是凸二次函数而约束条件是线性函数问题的方法称作凸二次规划, 通常求解二次规划问题使用拉格朗日理论. 此外对偶理论在构建支持向量机的过程中也扮演重要角色. 拉格朗日理论最初只适用于目标函数是等式的理论, 后来由 Kuhn 和 Tucker 发展到了可以适用于目标函数是不等式的理论. 约束的推广理论实际上就是 Karush-Kuhn-Tucker (KKT) 理论 [70].

4.1.1 问题公式化

对于给定的函数 f, 有 $g_i, i=1,\cdots,k$, 以及 $h_i, j=1,\cdots,m$ 在 $\Omega \subseteq R^n$ 上有定义

$$\begin{array}{lll} \text{minimize} & f(w), & w \in \Omega \\ \text{subject to} & g_i(w) \leqslant 0, & i=1,\cdots,k \\ & h_j(w)=0, & j=1,\cdots,m \end{array}$$

其中 $f(w)$ 称作目标函数, g_i 称作不等式约束, h_i 称作等式约束. 为了简便起见, 这里用 $g(w) \leqslant 0$ 以及 $h(w)=0$ 表示它们对所有的 i 成立. f 的优化值称作优化问题的值.

$F=\{w \in \Omega : g(w) \leqslant 0, h(w)=0\}$ 称作可行域, f 在这个区域中有定义, 并且约束条件得到满足. 如果没有 $w \in F$ 使 $f(w) < f(w^*)$ 存在, 那么优化问题的一个解 $w^* \in F$ 称作全域最小化解. 如果存在 $\varepsilon > 0$, 当 $\|w-w^*\| < \varepsilon$ 时, 对于所有 $w^* \in F$ 有 $f(w) \geqslant f(w^*)$ 成立, 那么点 w^* 称作优化问题的一个局域最小化解.

如果目标函数和约束条件都是线性的, 那么这个优化问题称作线性规划. 如果目标函数是二次的而约束条件是线性的, 那么这个优化问题称作二次规划.

如果 $g_i(w)=0$, 那么约束条件 g_i 称作在点 w 激活. 因此如果点 w 在可行域的边界上, 任何约束条件都在点 w 激活. 从这种角度来看等式约束总是激活的.

凸集定义: 如果 $\forall w,u\in\Omega$, 并且对于任一 $\theta\in(0,1)$, 点 $(\theta w+(1-\theta)u)\in\Omega$, 那么 $\Omega\subseteq R^n$ 称作凸集. 如果 $\forall w,u\in\Omega$, 并且对于任一 $\theta\in(0,1)$, 有 $f(\theta w+(1-\theta)u)\leqslant\theta f(w)+(1-\theta)f(u)$ 成立, 那么实值函数 $f(w)$ 称作凸集. 如果小于号成立, 该函数成为严格凸集.

定义在凸集上的凸函数的最小化问题称作凸规划问题. 它的很好的一个特性是, 如果 $f(w)$ 是严格凸集, 那么对于一个凸问题任何一个局域解都是全域解, 并且这个解是唯一的.

4.1.2 拉格朗日理论

拉格朗日函数定义为目标函数加上等式约束的线性组合 $h_i(w)=0, i=1,\cdots,m$

$$L(w,\alpha)=f(w)+\sum_{i=1}^{m}\alpha_i h_i(w)$$

其中 α_i 称作拉格朗日乘子.

拉格朗日定理是二次规划寻优的重要依据 [70,71]. 对于一个带有目标函数 $f(w)$ 和等式约束 $h_i(w)=0, i=1,\cdots,m$ 的优化问题, f 和 $h_i\in C$, 那么 $w^*\in\Omega$ 是该问题的解的必要条件是

$$\frac{\partial}{\partial w}L(w^*,\alpha^*)=0$$

和

$$\frac{\partial}{\partial\alpha}L(w^*,\alpha^*)=0$$

如果 $L(w^*,\alpha^*)$ 是 w 的凸函数, 这些条件也是充分条件.

这个条件给出了一个 $n+m$ 个方程的线性方程组, 其中最后 m 个是等式约束. 求出这个方程组的解就得到原问题的解. 因为约束条件的最优点的值都是零, 所以拉格朗日函数的值等于目标函数的值

$$L(w^*,\alpha^*)=f(w^*)$$

现在把不等式条件也包含在了里面得到拉格朗日推广函数.

对于一般优化问题

$$\begin{array}{lll}\text{minimize} & f(w), & w\in\Omega\\ \text{subject to} & g_i(w)\leqslant 0, & i=1,\cdots,k\\ & h_j(w)=0, & j=1,\cdots,m\end{array}\tag{4.1}$$

拉格朗日推广函数为

$$\begin{aligned} L(w,\alpha,\beta) &= f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_i h_j(w) \\ &= f(w) + \alpha^t g(w) + \beta^t h(w). \end{aligned}$$

拉格朗日对偶问题描述如下：已知初始问题 (4.1) 定义对偶问题：

$$\begin{aligned} &\text{minimize} && \theta(\alpha,\beta) = \inf_{w} L(w,\alpha,\beta) \\ &\text{subject to} && \alpha > 0 \end{aligned}$$

目标函数的最大值称作优化问题的值.

弱对偶定理：对于初始优化问题的可行解 w 和对偶优化问题的可行解 (α,β), 不等式

$$f(w) \geqslant \theta(\alpha,\beta)$$

成立.

从上面的定理知道, 初始问题的值是对偶问题值的上界和. 初始问题的值和对偶问题值之间的差距称作对偶沟. 如果 $f(w^*) = \theta(\alpha^*,\beta^*)$, 同时约束条件成立, 那么 w^* 和 (α^*,β^*) 分别是初始优化问题的解和对偶优化问题的解. 因为值相等, 上面定理的证明中的不等式转化成等式. 其中对于任意 i 有 $\alpha_i g_i(w) = 0$. 所以比较初始问题的值与对偶问题的值可以检验最优性. 如果对偶沟为零, 就得到了最优解.

对于 $w^* \in \Omega$ 和 $\alpha^* \geqslant 0$, 满足

$$L(w^*,\alpha,\beta) \leqslant L(w^*,\alpha^*,\beta^*) \leqslant L(w,\alpha^*,\beta^*)$$

点 w^*,α^*,β^* 为鞍点. 对于 $w^* \in \Omega$ 和 $\alpha^* \geqslant 0$, 鞍点相对于 w 是最小值, 相对于 (α,β) 是最大值. 如果点 w^*,α^*,β^* 为初始拉格朗日函数的鞍点, 那么它是初始问题和对偶问题的最优解, 并且没有对偶沟：$f(w^*) = \theta(\alpha^*,\beta^*)$.

强对偶定理：已知初始最优化问题 (4.1), 如果定义域是凸集并且约束条件 g_i 和 h_j 是仿射函数, 那么对偶沟为零.

4.1.3　KKT 理论

KKT 定理：已知初始最优化问题 (4.1), 并且 $f_i \in C$ 是凸集, 约束条件 g_i 和 h_i 是仿射函数, 那么 w^* 是最优值的必要充分条件是存在 α^*、β^* 有

$$\frac{\partial}{\partial w} L(w^*,\alpha^*,\beta^*) = 0$$

$$\frac{\partial}{\partial \beta} L(w^*,\alpha^*,\beta^*) = 0$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, \cdots, k$$

$$g_i(w^*) \leqslant 0, \quad i = 1, \cdots, k$$

$$\alpha_i^* \geqslant 0, \quad i = 1, \cdots, k$$

解对偶问题一般要比解初始问题容易. 对偶函数不依赖初始变量. 对偶变量是问题的基本未知数, 所以对偶方法为构建优化任务提供了新的视角.

由初始问题转化为对偶问题一般分两步: ① 把初始拉格朗日函数关于初始变量的偏微分设为零; ② 运算出结果后把它们代回初始拉格朗日函数. 这种替换依赖初始变量, 即 $\theta(\alpha, \beta) = \inf\limits_{w} L(w, \alpha, \beta)$.

这种方法是一种标准的支持向量机技术, 它提供了最优化理论的算法. 对偶描述可以工作在高维空间, 而不会发生维数灾难. 利用 KKT 条件可以有效地减少计算中的数据量. 只有激活的约束条件才有非零的对偶变量, 所以变量数会大大小于初始训练集的变量数 [69].

4.2 支持向量机

4.2.1 支持向量机基本原理简介

支持向量机是一种利用高维特征空间线性函数的假设空间的学习系统. 支持向量机通过优化理论得到的学习算法训练, 优化的目标是从统计学习理论获得的学习偏差. 简单地说, 支持向量机就是在输入空间求值的高维最大分类间隔超平面. 如果把一个有代表性的样本集 (线性不可分, 包含多种样本点) 进行分类, 要做的第一件事是要找到一个合适的高维特征空间, 通过核函数把原样本集映射到这个高维空间中, 使得原样本集线性可分 (能够被超平面无错误的分开). 然后通过最优化技术计算得出最优分类超平面来最大分离这些样本点. 那么样本点就可以在原低维空间中非线性地分类. 因此, 支持向量分类的最终目的就是设计一种高维空间中计算快捷的方法来得到最优分类超平面, 即支持向量机 [46].

支持向量机是从线性可分情况下的最优分类超平面发展而来的, 它由离它最近的少数样本点决定, 而与其他样本无关. 这些与最优分类面最近的少数样本点就是支持向量. 这样在运算时就可以把那些无关的样本点剔除, 只保留那些支持向量, 以最大限度地减小运算成本 [71]. 支持向量机基本思想可用图 4-1 的两维情况说明. 图 4-1 中, 圆点和方点代表两类样本点, 最优分类面为最优分类线. 边界 H_1、H_2 是平行于最优分类线的直线, 它们分别由各类样本点中离最优分类线最近的样本点决定. 离最优分类面最近的样本点与最优分类面之间的距离称作分类间隔.

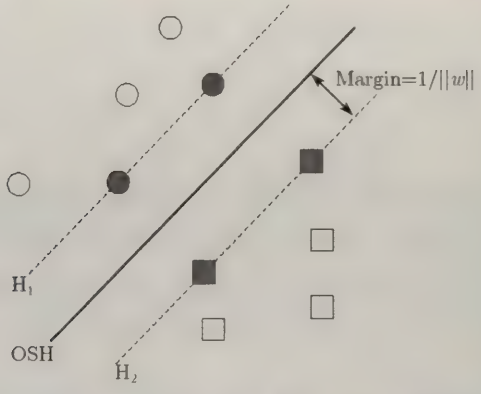


图 4-1　支持向量机的基本思想 (源自：Florian Markowetz, 2002)

如果向量集合被没有错误地分开，并且离超平面最近的向量与超平面的距离 (Margin) 是最大的，则说这个向量集合被这个最优超平面分开

一个分类超平面是最优分类超平面要达到两点要求：① 能将两类正确分开 (训练错误率为 0)；② 使分类间隔最大 [72].

4.2.2　线性分类

假定训练数据

$$(x_1, y_1), \cdots, (x_l, y_l), x \in R^n, \quad y \in \{+1, -1\}$$

可以被一个超平面

$$\langle w, x \rangle + b = 0$$

分开. 如果这个向量集合被超平面没有错误地分开，并且离超平面最近的向量与超平面之间的距离是最大的，则这个向量集合被这个最优超平面分开 [46,72].

为了使用简单的线性分类超平面构造支持向量机，首先考虑超平面族

$$\langle w, x \rangle + b = 0, \quad w \in R^n, \quad b \in R$$

相应的决策函数是

$$f(x) = \text{sign}(\langle w, x \rangle + b)$$

决策函数是符号函数

$$f(x) = 1, \quad \langle w, x \rangle + b \geqslant 0$$
$$f(x) = -1, \quad \langle w, x \rangle + b < 0$$

它的值决定样本点属于哪一个样本点类. 从样本点学习是通过考察属于某一类的样本点来了解这个类的属性.

既然有很多种方法可以对两类样本进行分类, 那么就需要建立一个标准来评价各种分类的优劣. 一般来说符合下面条件的最优的分离方法最优:

$$\max_{w,b} \quad \min\{\|x - x_i\| : x \in R^n,\ \langle w, b\rangle + b = 0,\ i = 1, \cdots, l\}$$

如果存在一个单位向量 $w\,(\|w\| = 1)$ 和一个常量 b, 有方程

$$\langle w, x_i\rangle + b > 0, \quad 若 \quad y_i = +1 \tag{4.2}$$

$$\langle w, x_i\rangle + b < 0, \quad 若 \quad y_i = -1 \tag{4.3}$$

成立, 那么样本集 $X = \{(x_1, y_1), \cdots, (x_l, y_l), x_i \in R^n, y_i\{+1, -1\}\}$ 被超平面 $\langle w, x\rangle + b = 0, x_i \in R^n, b \in R$ 可分. 这个由 w 和 b 定义的超平面称作分类超平面 [73].

(4.2) 式和 (4.3) 式可以采用一种紧凑的形式:

$$y_i[\langle w, x_i\rangle + b] > 0, \quad i = 1, \cdots, l \tag{4.4}$$

对于由 $\langle w, x\rangle + b = 0$ 定义的分类超平面 H.

(1) H 到 x_i 的距离为样本点 x_i 的分类间隔

$$\gamma_i\,(w, b) = y_i\,(\langle w, x_i\rangle + b)$$

(2) 向量集 $S = \{x_1, \cdots, x_n\}$ 的分类间隔 $\gamma_s\,(w, b)$ 为 H 到 S 中每一个向量的距离的最小值

$$\gamma_s(w, b) = \min \gamma_i(w, b), \quad x_i \in S$$

如果单位向量 w^* 和常量 b^* 确定了样本集 $S = \{x_1, \cdots, x_n\}$ 到分类超平面 H 的最小距离, 并且满足 (4.2) 式和 (4.3) 式, 那么由 (w^*, b^*) 确定的超平面称作最优分类超平面, 即样本集 S 的最优分类超平面定义为

$$(w^*, b^*) = \arg\max_{w,b} \gamma_x\,(w, b)$$

这个超平面具有最优性, 即不但能将两类样本集正确地分开, 而且能使距离它最近的向量到它的距离最大. 最优超平面就是满足 (4.4) 式并且使得 $\gamma_s\,(w, b)$ 最小的超平面. 这样的分类超平面会获得最佳的推广能力 [74]. 所谓推广能力是指超平面对不属于训练集的新数据正确分类的能力. 最优分类超平面是唯一的.

4.2.2.1 构造最优分类超平面

构造最优分类超平面就是找到能把属于两个不同类 $y \in \{-1, +1\}$ 的样本集

$$(y_1, x_1), \cdots, (y_l, x_l)$$

分开, 并且系数的模最小的超平面.

最优分类超平面实际就是最优化问题

$$\begin{aligned} &\text{maximize} && \gamma_s(w,b) \\ &\text{subject to} && \gamma_s(w,b)>0 \\ & && \|w\|^2=1 \end{aligned} \tag{4.5}$$

的解. 它不是规范的最优化问题表达形式, 因为约束条件不是线性的, 从代数的角度来看它难于求解 [75]. 下面把它转化成规范的形式. 因为 $w^*=\dfrac{w_0}{\|w_0\|}$ 且

$$\begin{aligned} &\frac{\langle w_0,x_i\rangle+b}{\|w_0\|}\geqslant\frac{1}{\|w_0\|}, \quad \forall i\in I_+ \\ &-\frac{\langle w_0,x_j\rangle+b}{\|w_0\|}\geqslant\frac{1}{\|w_0\|}, \quad \forall i\in I_- \end{aligned}$$

得到

$$\left\langle\frac{w_0}{\|w_0\|},x_i\right\rangle-\left\langle\frac{w_0}{\|w_0\|},x_j\right\rangle\geqslant\frac{2}{\|w_0\|}$$

又因为 $\gamma_s(w^*)=\dfrac{1}{\|w_0\|}$, 所以分类间隔的 $\gamma_s(w^*)$ 最大值是 $\|w_0\|$ 的最小值. 因此原最优化问题就可以转化成规范形式

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 \tag{4.6}$$

$$\text{subject to} \quad \langle w,x_i\rangle+b\geqslant 1, \quad \text{若} \quad y_i=+1 \tag{4.7}$$

$$\langle w,x_i\rangle+b\leqslant -1, \quad \text{若} \quad y_i=-1 \tag{4.8}$$

因此只要求出 (4.6) 式的解, 就可以得到最优分类超平面. 首先为了计算方便把 (4.7) 式和 (4.8) 式合并为等价形式

$$y_i\left(\langle w,x_i\rangle+b\right)-1\geqslant 0, \quad i=1,\cdots,l \tag{4.9}$$

该优化问题是一个带线性限制条件的二次优化问题, 拉格朗日方法可以用来对以上优化问题求解, 即找到拉格朗日函数

$$L_p(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1\right] \tag{4.10}$$

的鞍点, L_p 是初始条件下的拉格朗日函数. 即, 需要用拉格朗日函数求关于 w,b 的最小值和关于 $\alpha_i>0$ 的最大值, 其中 $\alpha_i\geqslant 0, i=1,\cdots,l$ 是拉格朗日乘数 [76]. 这里的目标函数使用 $\dfrac{1}{2}\|w\|^2$ 而不是 $\|w\|$ 主要是为了避免开方运算, $\dfrac{1}{2}\|w\|^2$ 中 $\|w\|$ 指

数 2 称作范数. 既然鞍点是一个稳定点, 那么它应该满足 KKT 条件, 即在这个点上的 $L_p(w,b,\alpha)$ 对于 w 和 b 的偏微分是 0, 即有

$$\frac{\partial L_p(w,b,\alpha)}{\partial w}=0 \text{ 和 } \frac{\partial L_p(w,b,\alpha)}{\partial b}=0$$

通过运算得到

$$w=\sum_{i=1}^{l}\alpha_i y_i x_i \text{ 和 } \sum_{i=1}^{l}\alpha_i y_i=0 \tag{4.11}$$

把 (4.11) 式代回到 (4.10) 式中得到 Wolfe 对偶条件优化问题 [72]

$$L_D(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1\right]=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle$$

对偶问题可以用公式

$$\begin{aligned}&\text{maximize} && L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle \\ &\text{subject to} && \alpha_i\geqslant 0,\quad i=1,\cdots,l \\ & && \sum_{i=1}^{l}\alpha_i y_i=0\end{aligned} \tag{4.12}$$

表示. 那么可以计算得

$$w^*=\sum_{i=1}^{l}\alpha_i^* y_i x_i$$

这里把原问题转化为它的对偶问题是因为: ① 对偶描述可以把最优分类面在非线性分类中推广; ② 简化参数.

向量 w^* 定义最优分类面, 那么函数 $L_D(\alpha)$ 的最大值

$$L_D(\alpha^*)=\frac{1}{2}\|w^*\|^2=\frac{1}{2}\sum_{i=1}^{l}\alpha_i^*$$

因为 $\gamma(w^*)^2=\left(\sum_{i=1}^{l}\alpha_i^*\right)^{-1}$, 所以最优分类面的分类间隔 $\gamma(w^*)$ 能够由系数 α_i^* 计算出.

4.2.2.2　支持向量

对于以上的初始问题, KKT 条件可以写成

$$\begin{aligned}
&\frac{\partial}{\partial w_v} L_p = w_v - \sum_i \alpha_i y_i x_{iv} = 0, \quad v = 1, \cdots, n \\
&\frac{\partial}{\partial b} L_p = -\sum_i \alpha_i y_i = 0 \\
&y_i \left(\langle w, x_i \rangle + b\right) - 1 \geqslant 0, \quad i = 1, \cdots, l \\
&\alpha_i \geqslant 0, \quad \forall i \\
&\alpha_i \left[y_i \left(\langle w, x_i \rangle + b\right) - 1\right] = 0, \quad \forall i
\end{aligned} \tag{4.13}$$

支持向量机问题是凸集问题. 对于凸集问题 KKT 条件是充要条件, 因此解支持向量机问题等价于求 KKT 条件的解. (4.13) 式称作 KKT 补充条件. 如果一个训练点 x_i 满足该条件, 那么相应的拉格朗日乘子等于零, 并且 x_i 位于 H_1 或 H_2 上

$$\begin{aligned}
H_1 &: \langle w, x_i \rangle + b = +1 \\
H_2 &: \langle w, x_i \rangle + b = -1
\end{aligned}$$

这两个超平面称作边界超平面. 离最优分类面最近的训练点就在这两个超平面上. 它们确定了分类间隔的边界. 这些满足 $\alpha_i \geqslant 0$ 并且位于 H_1 或 H_2 上的向量称作支持向量 (SV). 一个样本点如果相应的拉格朗日乘数 α_i 满足 $\alpha_i > 0$

$$w^* = \sum_{i=1}^{l} \alpha_i y_i x_i = \sum_{SV} \alpha_i y_i x_i \tag{4.14}$$

那么这个样本点称作支持向量 (support vector). 注意 (4.14) 式仅说明了支持向量都在 H_1 或 H_2 上, 它并没有说位于 H_1 或 H_2 上的点都是支持向量. 如果 α_i 和 $y_i \left(\langle w, x_i \rangle + b\right) - 1$ 同时为零, 那么这样的点虽然在 H_1 或 H_2 上, 它们并不是支持向量.

超平面把 R_n 分成 $\langle w, x_i \rangle + b > 0$ 和 $\langle w, x_i \rangle + b < 0$ 两个区域, 为了使用最大边界分类器, 首先要确定测试模式在哪边, 并且指定相应的标签. 因此测试点 x 的预测集

$$f(x) = \operatorname{sign}\left(\langle w^*, x \rangle + b^*\right) = \operatorname{sign}\left(\sum_i^{SV} \alpha_i y_i \langle x_i^{SV}, x \rangle + b^*\right)$$

常数 b 的值为

$$b = \frac{1}{2}\left[\langle w^*, x^*(1) \rangle + \langle w^*, x^*(2) \rangle\right]$$

其中 $x^*(1)$ 表示属于第一类的点, $x^*(2)$ 表示属于第二类的点.

4.2.2.3 线性非分离数据集中的最优分类面

有很多的实际问题不能用以上的算法解决, 一般说来噪声数据会破坏数据的可分性. 因为目标函数 (也可以说是对偶拉格朗日函数) 过大, 分类间隔最大化问题就会找不到可行解.

最优分类面最大的弊端就是不允许出现分类错误. 为了推广最大分类间隔超平面, 给 (4.7) 式和 (4.8) 式增加松弛条件, 使它在有分类错误时也可以应用, 但是每一次出现分类错误都要进行一次错误分类罚分 (即增加一个原始的目标函数).

这里引入一个正定松弛变量 $\xi_i, i=1,\cdots,l$, 那么限制条件变为

$$\begin{aligned}
&\langle w, x_i\rangle + b \geqslant +1-\xi_i, \quad \text{当} \quad y_i = +1\\
&\langle w, x_i\rangle + b < -1+\xi_i, \quad \text{当} \quad y_i = -1\\
&\xi_i \geqslant 0, \quad \forall i
\end{aligned}$$

上面的优化问题的限制条件就转化为

$$y_i(\langle w, x_i\rangle + b) - 1 + \xi_i \geqslant 0, \quad i=1,\cdots,l \tag{4.15}$$

满足 (4.15) 式的最小松弛变量为

$$\xi_i = \max(0, 1 - y_i(\langle w, x_i\rangle + b))$$

它表示一个点偏离边界的程度 (图 4-2):

$$\begin{aligned}
&\xi_i \geqslant 1 \leftrightarrow y_i(\langle w, x_i\rangle + b) \leqslant 0, \text{即 } x_i \text{ 错误的分类}\\
&0 \leqslant \xi_i \leqslant 1 \leftrightarrow x_i \text{ 分类正确, 但是位于边界以内}\\
&\xi_i = 0 \leftrightarrow x_i \text{ 分类正确, 而且位于边界以外或在边界上}
\end{aligned}$$

为了在最大化分类间隔的同时最小化分类错误, 可以把目标函数 $\frac{1}{2}\|w\|^2$ 转化为 $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_l^k$ 函数, 此时最初的优化问题转化为带松弛变量的优化问题

$$\begin{aligned}
&\text{minimize} \quad && \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_l^k \qquad (4.16)\\
&\text{subject to} \quad && y_i\left(\langle w, x_i\rangle + b\right) - 1 + \xi_i \geqslant 0, \quad i=1,\cdots,l\\
& && \xi_i \geqslant 0, \quad i=1,\cdots,l
\end{aligned}$$

错误权重是 C 由设计者自行定义的参数, 它表示错误罚分的多少. 对于任何正整数 k 优化问题是一个凸集, $k=1$ 和 $k=2$ 时都二次规划问题, 这里的 k 是范

数. 这种方法称作最优分类面的软边界推广, 相对应的最初的没有错误的推广称作硬边界推广 [72].

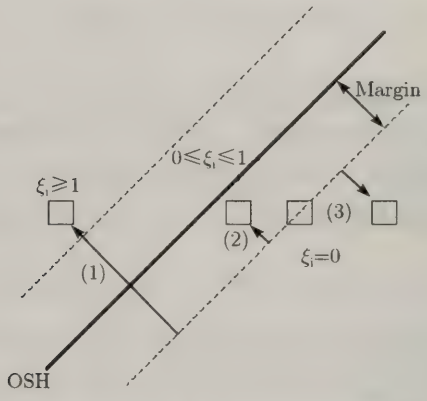


图 4-2　松弛变量的值 (源自: Florian Markowetz, 2002)

(1) 错误分类; (2) 正确分类, 但样本点在边界以内; (3) 正确分类, 样本点在边界上或边界外

对于 $k=1$ 初始问题的拉格朗日函数为

$$L_p(w,b,\xi,\alpha,\beta)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1+\xi_i\right]-\sum_{i=1}^{l}\beta_i\xi_i$$

其中 α_i, β_i 是拉格朗日乘子, 且 $\alpha_i\geqslant 0$, $\beta_i\geqslant 0$. 那么拉格朗日函数关于 w, ξ 和 β 的偏微分为

$$\frac{\partial L_p}{\partial w}=w-\sum_{i=1}^{l}\alpha_i y_i x_i=0$$

$$\frac{\partial L_p}{\partial \xi_i}=C-\alpha_i-\beta_i=0$$

$$\frac{\partial L_p}{\partial b}=\sum_{i=1}^{l}\alpha_i y_i=0$$

把计算结果代入初始问题得

$$\begin{aligned}&\text{maximize}\qquad L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle\\&\text{subject to}\qquad 0\leqslant\alpha_i\leqslant C,\quad i=1,\cdots,l\\&\qquad\qquad\sum_{i=1}^{l}\alpha_i y_i=0\end{aligned}\tag{4.17}$$

把 (4.17) 式与 (4.12) 式比较, 发现它们几乎相同. 唯一的不同之处就是拉格朗

日乘子 α_i 受到限制, 所以对于软边界问题 C 是 α_i 的上界. 因为 $\beta_i=(\alpha_i-C)$, 从初始问题得到的 KKT 补充条件为

$$\begin{aligned}&\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1+\xi_i\right]=0,\quad \forall i\\&\beta_i\left(\alpha_i-C\right)=0,\quad \forall i\end{aligned}\tag{4.18}$$

从 KKT 补充条件可以看出只有 $\alpha_i=C$ 才有可能出现非零的松弛变量. 相应的样本点 x_i 与超平面的距离小于 $\dfrac{1}{\|w\|}$. 相对应 $0\leqslant\alpha_i\leqslant C$ 的样本点位于超平面上. 同时也可以看到, 如果使 C 为无穷大, 那么相应的软边界问题转化为硬边界问题.

在正交坐标系中, 因为向量 α 位于边界为 C 的盒内部, 所以拉格朗日乘数的上界成为盒约束.

当 k=2 时的情况, (4.15) 式的第一个约束条件在 ξ_i <0 时也成立, 同时目标函数的值减小. 因此可以删去正约束条件 ξ_i 优化问题仍然有最优解. 初始函数的拉格朗日函数为

$$L_p\left(w,b,\xi,\alpha,\beta\right)=\frac{1}{2}\|w\|^2+\frac{C}{2}\sum_{i=1}^{l}\xi_i^2-\sum_{i=1}^{l}\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1+\xi_i\right]$$

在上式中, 分别对 w,ξ 和 β 求偏导数

$$\frac{\partial L_p}{\partial w}=w-\sum_{i=1}^{l}\alpha_i y_i x_i=0\tag{4.19}$$

$$\frac{\partial L_p}{\partial \xi_i}=C\xi-\alpha=0\tag{4.20}$$

$$\frac{\partial L_p}{\partial b}=\sum_{i=1}^{l}\alpha_i y_i=0\tag{4.21}$$

由 (4.19) 式和 (4.20) 式分别可以得到

$$w=\sum_{i=1}^{l}\alpha_i y_i x_i\tag{4.22}$$

$$\xi=\frac{1}{C}\alpha\tag{4.23}$$

把 (4.21) 式、(4.22) 式和 (4.23) 式分别代入 (4.17) 式, 计算得

$$L_D\left(\alpha\right)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle-\frac{1}{2C}\langle\alpha,\alpha\rangle$$

使用方程

$$\langle \alpha, \alpha \rangle = \sum_{i=1}^{l} \alpha_i^2 = \sum_{i=1}^{l} \alpha_i \alpha_j \delta_{ij} = \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \delta_{ij}$$

其中 δ_{ij} 称作克罗内克符号, 当 $i = j$ 时, $\delta_{ij} = 1$; 当 $i \neq j$ 时, δ_{ij}=0. 因此解决范数为 2 的软边界优化问题等价于求问题

$$\begin{aligned} \text{maximize} \quad & L_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \frac{1}{C} \delta_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0 \\ & \alpha_i \geqslant 0 \end{aligned} \tag{4.24}$$

的解. 从初始问题得到的 KKT 补充条件为

$$\alpha_i \ [y_i (\langle w, x_i \rangle + b) - 1 + \xi_i \geqslant 0], \quad \forall i \tag{4.25}$$

4.2.2.4　非平衡数据集的非线性分类

前面一段的推理建立在对每个数据集的罚分都一样的基础上, 对于非平衡数据集则可以对每个数据集的罚分有所不同: 较小的数量级的罚分高些, 较大的数据集罚分低些. 这样做可以保护小数据集中的点, 以免它们被当作大数据集分类错误的点. 基本方法是对于正负两个数据集采取不一样的权重 C^+ 和 C^-, 这样重要的数据集会有一个大的乘数. 这样就使得较小数据集的决策边界比较大的数据集的决策边界大. 那么初始拉格朗日函数就有两种类型错误的损失函数: 使 $I_+ = \{i \,|y_i = +1\}$ 和 $I_- = \{i \,|y_i = -1\}$, 那么初始最优化问题就是

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|w\|^2 + C^+ \sum_{i \in I^+} \xi_i^k + C^- \sum_{i \in I^-} \xi_i^k \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) - 1 + \xi_i \geqslant 0, \quad i = 1, \cdots, l \\ & \xi_i \geqslant 0, \quad i = 1, \cdots, l \end{aligned} \tag{4.26}$$

对于 $k = 1$ 初始拉格朗日函数变为

$$\begin{aligned} L_p(w, b, \xi, \alpha, \beta) = & \frac{1}{2} \|w\|^2 + C^+ \sum_{i \in I^+}^{l} \xi_i - C^- \sum_{i \in I^-}^{l} \xi_i \\ & - \sum_{i=1}^{l} \alpha_i \left[y_i (\langle w, x_i \rangle + b) - 1 + \xi_i \right] - \sum_{i=1}^{l} \beta_i \xi_i \end{aligned}$$

对偶拉格朗日函数除了约束条件

$$0 \leqslant \alpha_i \leqslant C^+, \quad i=+1$$
$$0 \leqslant \alpha_i \leqslant C-, \quad i=-1$$

对于 $k=2$ 初始拉格朗日函数变为

$$L_p(w,b,\xi,\alpha,\beta)=\frac{1}{2}\|w\|^2+C^+\sum_{i\in I^+}^{l}\xi_i^2-C^-\sum_{i\in I^-}^{l}\xi_i^2-\sum_{i=1}^{l}\alpha_i\left[y_i\left(\langle w,x_i\rangle+b\right)-1+\xi_i\right]$$

对偶公式为

$$L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle-I_{[i\in I^+]}\frac{1}{C^+}\delta_{ij}+I_{[i\in I^-]}\frac{1}{C^-}\delta_{ij}$$

其中 $I_{[\bullet]}$ 是指示函数.

4.2.2.5 线性机器的对偶

线性学习机对偶描述是支持向量机中一个很重要的概念. 一个向量可以写成训练点的线性组合的形式 $w=\sum_{i=1}^{l}\alpha_i y_i x_i$, 其中 $X=\{(x_i,y_i):i=1,\cdots,l\}$ 是已经分类的训练集. α_i 是在最大分类间隔问题中引入的拉格朗日函数的解, 称作对偶变量. 它们是基本的未知数. 展开 w 得到拉格朗日函数

$$L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j}^{l}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle$$

由此可以得到决策函数

$$f(x)=\operatorname{sign}\left(\langle w,x\rangle+b\right)=\operatorname{sign}\left(\sum_{i=1}^{l}\alpha_i y_i\left\langle x_i,x\right\rangle+b\right)$$

从这个公式中可以看到训练和测试点没有表现出独特的属性. 在训练阶段只需要把训练点输入 Gram 矩阵, 测试阶段新点伴随训练数据以内积的形式出现. 这个性质可以使最优分类超平面的概念在非线性分类器中泛化.

4.2.3 非线性分类

因为样本点有严格的限制, 所以线性分类问题可以通过规范的最优化技术求解. 非线性分类问题的求解需要技巧性更强些. 非线性分类问题求解的一般思路是把输入数据映射到高维空间, 然后再进行线性分类, 这样就会使输入空间数据非线性分类. 把输入数据映射到高维空间要通过一种方法称作核技术. 创建非线性机器需要两个步骤: 首先是利用非线性映射把数据传递到特征空间, 然后是用线性机器在特征空间中对这些数据分类.

4.2.3.1　特征映射

把训练样本映射到希尔伯特空间 H(可能是无限维的) 可以把训练点进行转换. 一般来说, 希尔伯特空间 H 比较样本所在的空间 L 具有很高的维数. 在训练之前把每一个训练点作映射 $\Phi: L \to H$, 那么优化分类超平面在 H 中构建. 把 $\Phi(x)$ 的第 i 个元素称为在映射 Φ 下的第 i 个特征, H 称为特征空间. 选择数据最适合描述的行为称为特征选择.

最优化问题的解是一个向量 $w \in H$, 它能够写成训练点的线性组合. 从学习任务的对偶公式得到算法仅仅依赖于训练样本与测试点的内积. 所以格兰姆矩阵 $G = \langle \Phi(x_i), \Phi(x_j) \rangle$ 与向量 $(\Phi(x_i), \Phi(x))$ 承担学习的任务, 其中 x 是新的测试点.

在特征空间的决策函数为

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{l} \alpha_i y_i \langle \Phi(x_i, \Phi(x)) \rangle + b \right)$$

假设在输入空间有一个非线性分离的数据集, 一般可以看到使用一个到高维空间的映射能使它特征线性分离. 所以使用 Φ 可以使分类问题更加适合最大分类间隔方法.

4.2.3.2　核函数

核这个名字来自整算子理论. 因为在一个方程中没有出现特征向量, 用来估计内积函数的内积计算的运算数量就是不必要的. 可以利用核来把数据映射到特征空间, 并且在这个空间训练线性机器. 能够利用的训练样本的唯一信息是它们在特征空间的 Gram 矩阵. 这个矩阵也称为核矩阵, 用 K 来表示. 这种方法的关键是找到一个能够进行有效估计的核函数. 线性学习机的一个重要特性是它能用对偶的形式来表示. 这就意味着假设可以表示成训练点的线性组合, 所以决策函数可以仅仅利用检测点的内积和训练点来估计

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \langle \Phi(x_i) \cdot \Phi(x_i) \rangle + b$$

如果有办法像计算原输入点的一个函数一样在特征空间直接计算内积 $\langle \Phi(x) \cdot \Phi(z) \rangle$, 那么就有可能利用两个步骤来创建一个非线性学习机. 这样的直接计算方法称为核技术.

有一个从输入空间 L 到内积空间 H 的映射 $\Phi: L \to H$. 如果对于所有的 $x, z \in L$ 都有

$$k(x, z) = \langle \Phi(x), \Phi(z) \rangle_H$$

k 就称 $k: L \times L \to R$ 为核函数. 核函数的行为好像是在高维特征空间中进行内积运算, 而在输入空间中估计结果.

由于学习和训练都依赖特征空间内积的值, 那么它们都可以从核函数的角度加以阐明. 一旦选择了核函数, 决策函数可以写成

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{l} \alpha_i y_i k\langle x_i, x\rangle + b\right).$$

那么不必知道是否正在进行的特征映射能够在特征空间完成学习任务. 可以成为核函数的条件 [46,75]:

(1) 对称函数

$$k(x,z) = \langle \Phi(x), \Phi(z)\rangle = \langle \Phi(z), \Phi(x)\rangle = k(z,x)$$

(2) 满足柯西–施瓦兹不等式

$$\begin{aligned} k(x,z)^2 &= \langle \Phi(x), \Phi(z)\rangle^2 = \|\Phi(x)\|^2 \|\Phi(z)\|^2 \\ &= \langle \Phi(x), \Phi(z)\rangle \langle \Phi(z), \Phi(x)\rangle = k(x,x)\, k(z,z) \end{aligned}$$

(3) (Mercer 定理) 若 L 是 R_n 的密集子集, 假设 k 是连续对称函数使整数算子

$$T_k : L_2(L) \to L_2(L)$$

如果

$$T_k f(\cdot) = \int_L k(\cdot, x) f(x)\, \mathrm{d}x$$

为正, 即对于所有的 $f \in L_2(L)$ 有

$$\int_{L\times L} k(x,z) f(x) f(z)\, \mathrm{d}x\mathrm{d}z \geqslant 0$$

那么根据 T_k 的特征函数 $\Phi_j \in L_2(L)$ 把 $k(x, z)$ 扩张到一个一致收敛序列 (在 $L \times L$ 上), 并且正交化这个特征向量 $\|\Phi_j\|_{L2} = 1$, 那么联合特征值 $\lambda_j \geqslant 0$

$$k(x,z) = \sum_{j=1}^{\infty} \lambda_j \Phi_j(x) \Phi_j(z)$$

4.2.3.3 基于核的分类

有很多方法可以选择特征空间. 一般来说应该选择能够包含原属性基本信息的最小特征集. 这种方法称为特征减少, 它对运算和推广很有利

$$x = (x_1, \cdots, x_n) \to \Phi(x) = (\Phi(x_1, \cdots, \Phi(x_n))), \quad d < n$$

为了使线性机器学习非线性相关, 需要选择一个非线性特征集并且用新的描述方法重新写入数据. 这个过程实际上就是利用一个固定的数据非线性映射到一个特征空间. 在这个新特征空间中可以使用线性机器. 因此, 假设集将会是这种类型

$$f(x)=\sum_{i=1}^{N} w_i \Phi_i(x)+b$$

其中 $\Phi: x \to F$ 是一个从输入空间到某个特征空间的非线性映射. 这意味着创建非线性机器需要两个步骤: 首先是利用非线性映射把数据传递到特征空间, 其次是用线性机器在特征空间中分类这些数据.

根据上面一节所给的三个条件, 优化分离平面和低维空间到高维空间的映射联系在一起. 一般来说, 一个支持向量机就是一个在输入空间求值的高维最大分离间隔超平面. 对于非线性分离的数据集要用特征空间的内积代替输入空间的结果重新阐述先前取得的结论.

硬边界最优化问题 (没有交叉的样本点)

$$\begin{aligned} \text{maximize} \quad & L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \\ \text{subject to} \quad & \alpha_i \geqslant 0, \quad i=1,\cdots,l \\ & \sum_{i=1}^{l}\alpha_i y_i=0 \end{aligned}$$

对于范数为 1 的软边界最优化问题写成

$$\begin{aligned} \text{maximize} \quad & L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \\ \text{subject to} \quad & 0 \leqslant \alpha_i \leqslant C, \quad i=1,\cdots,l \\ & \sum_{i=1}^{l}\alpha_i y_i=0 \end{aligned}$$

(有交叉的样本点) 与硬边界最优化问题的不同点就在于对拉格朗日乘数的额外限制: $0 \leqslant \alpha_i \leqslant c$, 这个推广条件称作盒限制.

基于核的范数为 2 的软边界向量机为 [74]

$$\begin{aligned} \text{maximize} \quad & L_D(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j\left(k(x_i,x_j)+\frac{1}{C}\delta_{ij}\right) \\ \text{subject to} \quad & \alpha_i \geqslant 0, \quad i=1,\cdots,l \\ & \sum_{i=1}^{l}\alpha_i y_i=0 \end{aligned}$$

硬边界分类问题与软边界分类问题的唯一不同之处在于软边界问题在训练点内积矩阵的对角线上多了一个 $\frac{1}{C}$. 这使得核矩阵 K 的特征值多了一个 $\frac{1}{C}$, 因为

$$Kv = \lambda v \Rightarrow \left(K + \frac{1}{C}I\right)v = Kv + \frac{1}{C}v = \left(\lambda + \frac{1}{C}\right)v$$

这个新问题可以理解为一个简单的变化

$$k'(x,z) = k(x,z) + \frac{1}{C}\delta_{xz}$$

上式为支持向量机的适应性提供了一个新的视野来看待前面提到的非平衡数据集, 因为必须从核的观点重新阐述它.

$$k'(x_i,z) = \begin{cases} k(x_i,z) + \dfrac{1}{C^+}\delta_{x_i z}, & y_i = +1 \\ k(x_i,z) + \dfrac{1}{C^-}\delta_{x_i z}, & y_i = -1 \end{cases}$$

相应的线性决策函数如下:

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{SV} \alpha_i y_i k\langle x, x_i^{SV}\rangle + b^*\right)$$

通过以上分析支持向量分类图示如图 4-3 所示.

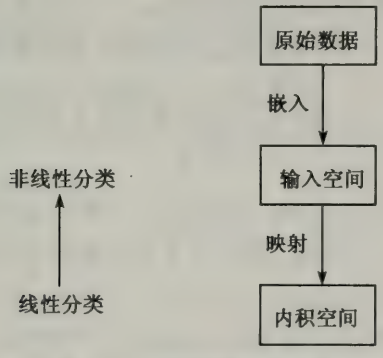


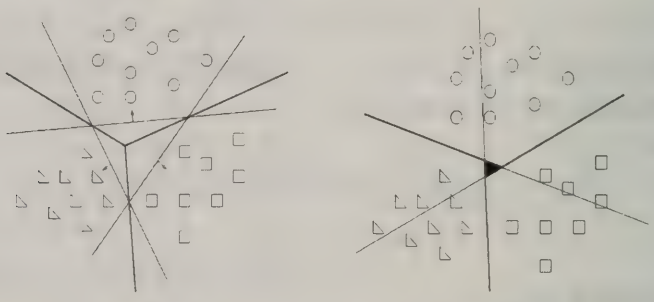
图 4-3 支持向量分类过程

这个过程的主要思想是首先把原始数据嵌入一个低维的输入空间, 再运用构造的核技术把它们映射到高维空间并进行线性分类, 并且最终使得数据在低维空间非线性分类

核函数的运用是计算上的捷径. 使用这种方法, 首先创建一个复杂的特征空间, 其次计算出这个空间中的内积, 再次找到直接计算初始输入空间值的方法. 然而在非线性分类中采用的方法是直接定义一个核函数, 因此隐含定义了一个特征空间. 从这个角度来说, 规避了定义特征空间就是规避了内积的运算.

4.2.4　多重分类

前面的讨论都以区分两类点为基础, 然而在现实世界中常常处理多于两类点的分类. 训练集由数对 (x_i, y_i) 组成, 其中 $x_i \in R^n$, 且 $y_i \in \{1, \cdots, n\}$, $i = 1, \cdots, l$. 下面阐述二元分类扩展到多元分类的方法. 多元分类是以二元分类为基础的, 由二元分类扩展到多元分类一般包括两种方法 [74]: ① 一对多的分类; ② 一对一的分类. 这两种多重分类方法的比较见图 4-4.



(a) 一对多分类　　(b) 一对一分类

图 4-4　两种多重分类方法的比较 (源自: Florian Markowetz, 2002)

4.2.4.1　一对多的分类

使用二元分类器构造多重分类的过程如下: 构造多个二元分类函数 $f_k(x) = \text{sign}(\langle w_k, x\rangle + b_k), k = 1, \cdots, n$, 那么训练点的第 k 类分类样本可以表示为

$$f_k(x) = \begin{cases} +1, & x \text{ 是属于第 } k \text{ 类的点} \\ -1, & \text{其他} \end{cases}$$

在二元分类中使用这样的分类方法, 但在多元分类中这样进行分类就会出现混乱. 因为要把平面分成好几个区域就必须有多个分类器处于激活状态.

考虑实值向量 $f(x) = (f'_1(x), \cdots, f'_n(x))$, 其中 $f'_k(x) = \langle w_k, x\rangle + b_k$. 从几何的角度来看, 这个向量的每一个组分是 x 到第 n 个超平面的距离. 选择 x 使相应的 $f'_k(x), k = 1, \cdots, n$ 的值到超平面的距离最大

$$k^* = \arg\max_k f'_k(x)$$

这种方法称 "winner-takes-all". 这种启发式方法主要的缺点是所得边界上的三个点与 n 个支持向量机计算得到的初始决定函数必须一致, 所以没有发挥出最大分类超平面的优势. 这种方法改善了硬边界二元决策函数, 但没有得到最优决策边界, 这个边界只能由同时优化所有三个边界得到.

4.2.4.2 一对一的分类

定义了一个决策函数 $f_{kl}: R^n \to \{+1, -1\}$ 对于每一个数对 (k,l), 有

$$f_{kl}(x) = \begin{cases} +1, & \text{如果 } x \text{ 属于 } k \text{ 类} \\ -1, & \text{如果 } x \text{ 属于 } l \text{ 类} \end{cases}$$

因为数对是对称的, 有 $f_{kl} = -f_{lk}$. 另外定义 $f_{kk} = 0$, 可以得到 $\begin{pmatrix} n \\ 2 \end{pmatrix} = \dfrac{n(n-1)}{2}$ 对不同的决定函数. 为了得到一类决策函数, 求和

$$f_k = \sum_{l=1}^{n} f_{kl}(x)$$

4.2.4.3 直接的分类方法

非线性分类数据的最大分类间隔超平面优化问题可以推广到下面的公式. 有训练集 $X = \{(x_i, y_i) : i = 1, \cdots, l\}$, 其中 $x_i \in \{1, \cdots, n\}$. k 类发生 l_k 次, $\sum\limits_k l_k = l$. 那么求解

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\sum_{k=1}^{n} \|w_k\|^2 + C\sum_{k=1}^{n}\sum_{i=1}^{lk} \xi i^k \\ &\text{subject to} && \langle w_k, \dot{x}_i\rangle + b_k - \langle w_m, x_i\rangle - b_m \geqslant 1 - \xi_i^k, \quad y_i = k \end{aligned} \tag{4.27}$$

其中 $\xi_i \geqslant 0, k = 1, \cdots, n, m \neq k, i = 1, \cdots, k$, 这样就可以给出决策函数

$$f(x) = \arg\max_k f_k(x) = \arg\max_k(\langle w_k, x_i\rangle + b_k), \quad k = 1, \cdots, n$$

这个函数与二元分类的不同之处在于该函数是一个 n 类的和. 引入拉格朗日乘数可以把这个函数表达成对偶变量. 求拉格朗日函数

$$\begin{aligned} L_p(w, b, \xi, \alpha, \beta) = &\frac{1}{2}\sum_{k=1}^{n} \|w\|^2 + C\sum_{k=1}^{n}\sum_{i=1}^{lk} \xi_i^k - \sum_{k=1}^{n}\sum_{i=1}^{lk} \beta_i^k \xi_i^k \\ &- \sum_{k=1}^{n}\sum_{m\neq k}^{l}\sum_{i=1}^{lk} \alpha_i^{k,m}[\langle (w_k - w_m), x_i^k\rangle + b_k - b_m - 1 + \xi_i^k] \end{aligned}$$

对于鞍点上的 w, b 和 ξ 的偏微分都为零, 所以有方程

$$\frac{\partial L_p}{\partial w_k} = w_k - \sum_{m\neq k}\sum_{i=1}^{lk} \alpha_i^{k,m} x_i^k + \sum_{m\neq k}\sum_{j=1}^{lm} \alpha_j^{m,k} x_j^m = 0 \tag{4.28}$$

$$\frac{\partial L_p}{\partial b_k} = \sum_{m\neq k}\sum_{i=1}^{lk} \alpha_i^{k,m} + \sum_{m\neq k}\sum_{j=1}^{lm} \alpha_j^{m,k} = 0 \tag{4.29}$$

$$\frac{\partial L_p}{\partial \xi_i^k} = C - \sum_{m\neq k} \alpha_i^{k,m} - \beta_i^k = 0 \tag{4.30}$$

由 (4.28) 式得到 n 个超平面的法向量 w_k 的展开式

$$w_k = \sum_{m\neq k}\sum_{i=1}^{l_k}\alpha_i^{k,m}x_i^k + \sum_{m\neq k}\sum_{j=1}^{lm}\alpha_j^{m,k}x_j^m, \quad k=1,\cdots,n \tag{4.31}$$

由 (4.29) 式得到拉格朗日乘数的约束

$$\sum_{m\neq k}\sum_{i=1}^{lk}\alpha_i^{k,m} = \sum_{m\neq k}\sum_{j=1}^{lm}\alpha_j^{m,k} \tag{4.32}$$

把 (4.31) 式和 (4.32) 式代入 (4.27) 式中得到 L_p 的对偶拉格朗日函数

maximize

$$L_D(\alpha) = \sum_{k=1}^{n}\sum_{m\neq k}\left[\sum_{i=1}^{lk}\alpha_i^{k,m} - \frac{1}{2}\sum_{m^*\neq k}\left(\sum_{i,j=1}^{lk}\alpha_i^{k,m^*}\alpha_j^{k,m}\langle x_i^k x_j^k\rangle + \sum\sum\alpha_i^{m,k}\alpha_j^{m^*,k}\langle x_i^m, x_j^{m^*}\rangle - 2\sum\sum\alpha_i^{k,m^*}\alpha_j^{m,k}\langle x_i^k, x_j^m\rangle\right)\right]$$

subject to　$0 \leqslant \sum_{m\neq k}\alpha_i^{m,k} \leqslant C$

$$\sum_{m\neq k}\sum_{i=1}^{lk}\alpha_i^{k,m} = \sum_{m\neq k}\sum_{j=1}^{lm}\alpha_j^{m,k}, \quad k=1,\cdots,n$$

所以可以得到函数 $f(x)$ 为展开的支持向量

$$f(x) = \sum_{m\neq k}\sum_{i=1}^{lk}\alpha_i^{k,m}\langle x_i^k, x\rangle + \sum_{m\neq k}\sum_{j=1}^{lm}\alpha_j^{m,k}\langle x_j^k, x\rangle + b_k$$

对于 $n=2$ 这个结果与二元分类的情况相符; 对于 $n>2$ 时, 同时估计 $l(n-1)$ 个参数 $\alpha_i^{k,m}$.

4.2.4.4　多重分类法的比较

直接的方法是把支持向量的概念直接推广到两个以上的类. 一对一的方法保存了大多数的最大边界超平面, 而一对多的方法则产生出一个超平面结构和一个点. 一对多的方法比一对一的好处就在于只须对每一类点构造一个决定函数, 而不必构造 $\begin{pmatrix} n \\ 2 \end{pmatrix}$ 个决策函数, 这样提高了计算效率.

第5章　应用于支持向量机的主要算法

5.1　支持向量机算法中目前的研究状况

由于具有较好的理论基础和在一些领域的应用中表现出来的优秀的推广性能, 支持向量机方法近年颇受关注. 支持向量机算法经过几年的探索已经有了很大的改进, 实际应用也越来越广. 尽管支持向量机算法的性能在许多实际问题的应用中得到了验证, 但是该算法在计算上仍然存在着一些问题, 这些问题包括训练算法速度慢、算法复杂而难以实现以及检测阶段运算量大等 [76,77].

传统的利用标准二次型优化技术解决对偶问题的方法可能是训练算法慢的主要原因. 首先, 支持向量机方法需要计算和存储核函数矩阵, 当样本点数目较大时, 需要很大的内存; 其次, 支持向量机在二次型寻优过程中要进行大量的矩阵运算, 多数情况下, 寻优算法是占用算法时间的主要部分 [77].

支持向量机方法的训练运算速度是限制它应用的主要方面, 近年来人们针对方法本身的特点提出了许多算法来解决对偶寻优问题. 大多数算法的一个共同的思想就是循环迭代: 将原问题分解成为若干子问题, 按照某种迭代策略, 通过反复求解子问题, 最终使结果收敛到原问题的最优解. 根据子问题的划分和迭代策略的不同, 又可以大致分为两类.

第一类是 "块算法"[46]. "块算法" 基于这样一个事实, 即去掉拉格朗日乘子等于零的训练样本不会影响原问题的解. 对于给定的训练样本集, 如果其中的支持向量是已知的, 寻优算法就可以排除非支持向量, 只需对支持向量计算权值 (即拉格朗日乘子) 即可. 实际上支持向量是未知的, 因此 "块算法" 的目标就是通过某种迭代方式逐步排除非支持向量. 具体的做法是: 选择一部分样本构成工作样本集进行训练, 剔除其中的非支持向量, 并用训练结果对剩余样本进行检验, 将不符合训练结果 (一般是指违反 Kohn-Tucker 条件) 的样本 (或其中的一部分) 与本次结果的支持向量合并成为一个新的工作样本集, 然后重新训练. 如此重复下去直到获得最优结果.

当支持向量的数目远远小于训练样本数目时, "块算法" 显然能够大大提高运算速度. 然而, 如果支持向量的数目本身就比较多, 随着算法迭代次数的增多, 工作样本集也会越来越大, 算法依旧会变得十分复杂. 因此第二类方法把问题分解成为固定样本数的子问题: 工作样本集的大小固定在算法速度可以容忍的限度内, 迭代过程中只是将剩余样本中部分 "情况最糟的样本" 与工作样本集中的样本进行等

量交换, 即使支持向量的个数超过工作样本集的大小, 也不改变工作样本集的规模, 而只对支持向量中的一部分进行优化.

固定工作样本集的方法和块算法的主要区别在于: 块算法的目标函数中仅包含当前工作样本集中的样本, 而固定工作样本集方法虽然优化变量仅包含工作样本, 其目标函数却包含整个训练样本集, 即工作样本集之外的样本的拉格朗日乘子固定为前一次迭代的结果, 而不是像块算法中那样设为 0. 而且固定工作样本集方法还涉及一个确定换出样本的问题 (因为换出的样本可能是支持向量). 这样, 这一类算法的关键就在于找到一种合适的迭代策略使得算法最终能收敛并且较快地收敛到最优结果.

5.2　分 解 算 法

当支持向量的数目远远小于训练样本数目时, 块算法显然能够大大提高运算速度; 然而, 如果支持向量的数目本身就比较多, 随着算法迭代次数的增多, 工作样本集也会越来越大, 算法依旧会变得十分复杂. 因此, 如果把问题分解成为固定样本数的子问题: 工作样本集的大小固定在算法速度可以容忍的限度内, 迭代过程中只是将剩余样本中部分 "情况最糟的样本" 与工作样本集中的样本进行等量交换, 即使支持向量的个数超过工作样本集的大小也不改变工作样本集的规模, 而只对支持向量中的一部分进行优化, 这就是分解算法的基本思想 [88].

分解算法的基本思想是把原训练集索引 $\{1,\cdots,l\}$ 分解为两个子集 B 和 N, 其中 B 是工作集, $N=\{1,\cdots,l\}\setminus B$[77]. 如果向量集 α_B 和 α_N 分别表示对应的元素, 那么优化的目标值等于

$$\frac{1}{2}\alpha_B^{\mathrm{T}}Q_{BB}\alpha_B-(e_B-Q_{BN}\alpha_N)^{\mathrm{T}}\alpha_B+\frac{1}{2}\alpha_N^{\mathrm{T}}Q_{NN}\alpha_N-e_N^{\mathrm{T}}\alpha_N$$

在每一个循环中, α_N 是固定的, 仅解决子问题

$$\begin{aligned}\min\ & \frac{1}{2}\alpha_B^{\mathrm{T}}Q_{BB}\alpha_B-(e_B-Q_{BN}\alpha_N)^{\mathrm{T}}\alpha_B\\ & 0\leqslant(\alpha_B)_i\leqslant C,\quad i=1,\cdots,q\\ & y_B^{\mathrm{T}}\alpha_B=-y_N^{\mathrm{T}}\alpha_N\end{aligned}$$

其中 $\begin{bmatrix}Q_{BB} & Q_{BN}\\ Q_{NB} & Q_{NN}\end{bmatrix}$ 是矩阵 Q 的一个排列, q 是 B 的大小. 目标函数保持严格减小, 在一些情况下这种方法收敛于优化解.

分解算法如下 [76,88]:

(1) 人为选择样本集合 B, 构造子问题.

(2) 求子问题最优解 $\alpha_i, i \in B$ 及 b, 并置 $\alpha_j, j \in N$.

(3) 对于 $j \in N$, 有

$$\begin{cases} \lambda_j = 0, & f(x_j)y_j < 1 \\ \lambda_j = C, & f(x_j)y_j > 1 \\ 0 < \lambda_j < C, & f(x_j)y_j \neq 1 \end{cases}$$

用 λ_j 替换 $\lambda_i, i \in B$ 来解上面的子问题.

为了减小篇幅, 后面一章对分解算法进行更详细的介绍.

5.3 顺序最小优化算法

在分解算法的基础上, 微软研究院的研究人员提出并且改进了顺序最小优化算法 [89]. 这种算法将工作样本集的规模减到了最小 —— 两个样本. 之所以需要两个样本是因为等式线性约束的存在使得同时至少有两个拉格朗日乘数发生变化. 由于子问题的优化只涉及两个变量, 而且应用等式约束可以将其中一个变量用另一个变量线性表示出来, 所以迭代过程中每一步的子问题的最优解可以直接用解析的方法求出来, 无须使用数值分析中的二次规划软件包, 提高了子问题的运算速度. 他们还设计了一个两层嵌套循环分别选择进入工作样本集的两个样本, 外层循环选择第一个样本, 内层循环选择第二个样本. 外层循环首先在整个样本空间循环一遍, 决定哪些样本违反了 Kohn-Tucker 条件 $0 < \alpha_i < C$. 如果找到了不满足 Kohn-Tucker 条件的样本, 它即被选作进入工作集的第一个样本. 然后根据第二个启发式规则选择第二个样本. 最后用解析的方法快速对选定的样本进行优化. 为了加快算法的运行速度, 外层循环不总是每次检查所有训练样本. 每次在所有样本上循环一遍以后, 外层循环只在拉格朗日乘数大于零和小于 C 的样本上进行循环, 直到所有拉格朗日乘数大于零和小于 C 的样本都满足了最优化所应该满足的 Kohn-Tucker 条件, 然后再在整个样本空间循环一遍. 这样, 外层循环是交替地在整个样本空间和拉格朗日乘数大于零且小于 C 的样本上循环. 内层循环选择第二个进入工作集的样本, 选择的原则是使目标函数靠近最优点的速度达到最快. 这种启发式的样本选择策略大大加快了算法的收敛速度. 顺序最小优化算法表现在速度方面的良好性能, 它可以看作是分解算法的一个特例, 它将子问题的规模减少到了最小. 子问题的规模和迭代的次数是一对矛盾, 顺序最小优化算法将工作样本集的规模减少到两个样本, 一个直接的后果就是迭代次数的增加. 所以顺序最小优化算法实际上是将求解子问题的耗费转嫁到迭代上, 然后在迭代上寻求快速算法 [76].

5.3.1 顺序最小优化算法的原理

对于标准的支持向量机二次规划问题, 最小可能优化问题涉及两个拉格朗日乘

子, 因为拉格朗日乘子必须满足线性方程的限制. 在优化的每一步, 序列最小优化选择两个拉格朗日乘子进行优化, 来寻找乘子的优化值. 序列最小优化所谓的最大好处就是可以用解析的方法求解每一个最小规模的优化问题, 从而完全避免了迭代算法.

当然, 这样一次优化不可能保证其结果就是所优化的拉格朗日乘子的最终结果, 但会使目标函数向极小值迈进一步. 再对其他拉格朗日乘子做最小优化, 直到所有乘子都符合 Kohn-Tucker 条件时, 目标函数达到最小, 算法结束.

序列最小优化算法要解决两个问题: ① 用解析方法优化两个拉格朗日乘子; ② 使用启发式方法选择需要进行优化的拉格朗日乘子.

5.3.2　两个拉格朗日乘子的优化问题

序列最小优化首先计算乘子的上下限限制. 不妨设正在优化的两个拉格朗日乘子对应的样本正是第一个和第二个, 对两个拉格朗日乘子 α_1 和 α_2, 在其他乘子不改变的情况下, 它们的约束条件应表达为正方形内的一条线段 [113].

α_2 的上下限应为

$$\text{下限：} L=\max\left(0,\quad \alpha_2+y_1y_2\alpha_1-\frac{1}{2}\left(y_1y_2+1\right)C\right)$$

$$\text{上限：} H=\min\left(C,\quad \alpha_2+y_1y_2\alpha_1-\frac{1}{2}\left(y_1y_2-1\right)C\right)$$

而 α_1 和 α_2 在本次优化中所服从的等式约束为

$$\alpha_1+y_1y_2\alpha_2=\alpha_1^0+y_1y_2\alpha_2^0=d$$

目标函数二阶导数

$$\eta=K(\boldsymbol{x}_1,\boldsymbol{x}_1)+K(\boldsymbol{x}_2,\boldsymbol{x}_2)-2K(\boldsymbol{x}_1,\boldsymbol{x}_2)>0$$

若 $E_i=u_i-y_i$ 为第 i 个训练样本的误差, 那么无条件的极值点就为

$$\alpha_2=\alpha_2^0+\frac{y_2\left(E_1-E_2\right)}{\eta}$$

那么最终的 α_2 为

$$\alpha_2*=\begin{cases}H, & \alpha_2\geqslant H\\ \alpha_2, & L<\alpha_2<H\\ L, & \alpha_2\leqslant L\end{cases}$$

最后, 由等式约束确定 α_1 为

$$\alpha_1*=\alpha_1+y_1y_2\left(\alpha_2-\alpha_2*\right)$$

5.3.3 选择待优化拉格朗日乘子的启发式方法

为了使算法收敛得更快, 系列最小优化方法使用了启发式方法寻找拉格朗日乘子进行优化. 启发式方法先选择最有可能需要优化的 α_2, 再针对这样的 α_2 选择最有可能取得较大修正步长的 α_1. 这个过程使用两个层次的循环, 外层循环遍历非边界样本或所有样本: 优先选择遍历非边界样本, 因为非边界样本更有可能需要调整, 而边界样本常常不能得到进一步调整而留在边界上. 大部分样本都不是支持向量, 它们的拉格朗日乘子不为零和 C, 而拉格朗日乘子取得零值就无须再调整. 循环遍历非边界样本并选出它们当中违反 Kohn-Tucker 条件的样本进行调整, 直到非边界样本全部满足 Kohn-Tucker 条件为止. 当某一次遍历发现没有非边界样本得到调整时, 就遍历所有样本, 以检验是否整个集合也都满足 Kohn-Tucker 条件. 如果在整个集合的检验中又有样本被进一步优化, 就有必要再遍历非边界样本. 这样直到整个训练集都满足 Kohn-Tucker 条件为止.

内层循环针对违反 Kohn-Tucker 条件的样本选择另一个样本与它配对优化, 选择的依据是尽量使这样一对样本能取得最大优化步长. 对其中一个拉格朗日乘子 α_2 来说, 优化步长为 $\left|\dfrac{(E_1-E_2)}{\eta}\right|$. 对于核函数估算耗时较大, 用 $|E_1-E_2|$ 来大致估计有可能取得的步长大小, 即选出使得 $|E_1-E_2|$ 最大的样本作为第二个样本.

5.3.4 每次最小优化后的重置工作

每做完一次最小优化, 必须更新每个样本的误差, 以便用修正过的分类面对其他样本再做 Kohn-Tucker 检验, 以及选择第二个配对优化样本时估计步长之用.

更新样本的误差需要首先重置阈值 b, 以使得两个样本都满足 Kohn-Tucker 条件. 直接利用刚刚被优化的两个样本的信息在原阈值 b_0 基础上作简单修正, 而不需要调用所有支持向量重新计算 b. 最小优化后的 α_1* 如果不在边界上, b 的计算公式为

$$b_1 = E_1 + y_1\left(\alpha_1 * -\alpha_1^0\right) K\left(x_1, x_1\right) + y_2\left(\alpha_2 * -\alpha_2^0\right) K\left(x_1, x_2\right) + b$$

最小优化后的 α_2* 如果不在边界上, b 的计算公式为

$$b_2 = E_2 + y_1\left(\alpha_1 * -\alpha_1^0\right) K\left(x_1, x_2\right) + y_2\left(\alpha_2 * -\alpha_2^0\right) K\left(x_2, x_2\right) + b$$

α_1*, α_2* 都不在边界上时, b_1 和 b_2 是相等的. 两个拉格朗日乘子都在边界上时, b_1 和 b_2 以及它们之间的数都可作为符合 Kohn-Tucker 条件的阈值. 这时顺序最小优化算法选择 b_1 , b_2 之中点作为阈值.

非线性的情况, 误差的计算要用到所有已找到的支持向量及它们的拉格朗日乘

子

$$u_j = \sum_{\text{支持向量}} y_i \alpha_i K(x_i, x_j) - b$$
$$E_j = u_j - y_j$$

线性的情况则是先重置分类超平面的法向量 w, 再根据 $u_j = \langle w, x_j \rangle - b$ 计算输出 u_j 和误差 $E_j = u_j - y_j$. 同阈值的重置一样, 法向量的重置也不需要调用所有的支持向量, 只需在原来的法向量基础上作改动

$$w* = w + y_1 (\alpha_1 * - \alpha_1) x_1 + y_2 (\alpha_2 * - \alpha_2) x_2$$

5.3.5　顺序最小优化算法的特点和优势

顺序最小优化算法和以往流行的支持向量机优化算法 (如块算法、固定工作样本集法) 相比, 既有共同点, 又有自己的独特之处. 共同点在于它们都是把一个大的优化问题分解为很多小问题来处理. 块算法在每一步中将新加入样本中违反 Kohn-Tucker 条件的样本与原有的支持向量一起组成小问题的样本集进行优化, 优化完毕后只保留其中的支持向量, 再加进来新的样本进入下一步. 分解算法是每一步只收集新加入样本中“最坏”的样本, 并将原来保留的支持向量集中较好的替换出去, 以保持样本集大小不变. 顺序最小优化算法法则是把每一步的优化问题缩减到了最小, 它可以看作是固定工作样本集法的一种特殊情况: 把工作样本集的大小固定为 2, 并且每一步用两个新的拉格朗日乘子替换原有的全部乘子 [74,81].

顺序最小优化算法的最大特色在于它可以采用解析的方法而完全避免了二次规划数值解法的复杂迭代过程. 这不但大大节省了计算时间, 而且不会牵涉迭代法造成的误差积累. 理论上顺序最小优化算法的每一步最小优化都不会造成任何误差积累, 而如果用双精度数计算, 舍入误差几乎可以忽略, 于是所有的误差只在于最后一遍检验时以多大的公差要求所有拉格朗日乘子满足 Kohn-Tucker 条件. 可以说顺序最小优化算法在速度和精度两方面都得到了保证.

由于顺序最小优化算法不涉及二次规划数值解法, 就不必将核函数矩阵整个存在内存里, 而数值解法每步迭代都要拿这个矩阵作运算. 于是顺序最小优化算法使用的内存是与样本集大小呈线性增长的, 而不像以往的算法那样呈平方增长.

顺序最小优化算法对线性支持向量机最为有效, 对非线性则不能发挥出全部优势, 这是因为线性情况下每次最小优化后的重置工作都是很简单的运算, 而非线性时有一步加权求和, 占用了主要的时间. 其他算法对线性和非线性区别不大, 因为凡是涉及二次规划数值解的算法都把大量时间花在求数值解的运算中了.

第 6 章　Libsvm 简介

6.1 公　式

首先介绍 Libsvm 中使用的与支持向量分类相关的主要公式 [82].

6.1.1 C 支持向量分类 (二元)

如果训练向量使 $x_i \in R^n,\quad i=1,\cdots,l$ 分为两类, 向量 $y \in R^l$ 有 $y_i \in \{1,-1\}$, C-SVC [44] 解决以下优化问题

$$\begin{aligned} \min\ & \frac{1}{2}w^{\mathrm{T}}w + C\sum_{i=1}^{20}\xi_i \\ \text{subject to}\ & y_i\left(w^{\mathrm{T}}\Phi\left(x_i\right)+b\right) \geqslant 1-\xi_i \\ & \xi_i \geqslant 0,\quad i=1,\cdots,20 \end{aligned} \tag{6.1}$$

它的对偶问题为

$$\begin{aligned} \min_{\alpha}\ & \frac{1}{2}\alpha^{\mathrm{T}}Q\alpha - p^{\mathrm{T}}\alpha \\ & 0 \leqslant \alpha_i \leqslant C,\quad i=1,\cdots,l \\ & y^{\mathrm{T}}\alpha = 0 \end{aligned} \tag{6.2}$$

其中 p 代表所有向量之一, $C>0$ 是拉格朗日乘数的上界, Q 是 $l\times l$ 阶的半正定矩阵, 并且 $Q_{ij} \equiv y_i y_j K\left(x_i,x_j\right)$, $K\left(x_i,x_j\right) \equiv \Phi\left(x_i\right)^{\mathrm{T}}\Phi\left(x_j\right)$ 是核函数. 训练向量 x_i 被函数 Φ 映射到高维空间. 决策函数为

$$\operatorname{sign}\left(\sum_{i=1}^{l} y_i\alpha_i K\left(x_i,x\right)+b\right)$$

6.1.2 v 支持向量分类 (二元)

v 支持向量分类 [83] 使用参数 v, 这个参数可以同时控制支持向量的参数和分类错误. 参数 $v\in(0,1]$ 是部分训练错误点的上界和部分支持向量的下界 [84].

如果训练向量 $x_i \in R^n,\quad i=1,\cdots,l$ 分为两类, 向量 $y\in R^l$ 有 $y_i \in \{1,-1\}$, 那么 v 支持向量分类的原始优化问题为

$$\min_{w,b,\xi,\rho}\quad \frac{1}{2}w^{\mathrm{T}}w - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \tag{6.3}$$

$$y_i\left(w^{\mathrm{T}}\Phi\left(x_i\right)+b\right)\geqslant\rho-\xi_i$$

$$\xi_i\geqslant 0,\quad i=1,\cdots,l,\quad \rho\geqslant 0$$

它的对偶问题为

$$\begin{aligned}\min_{\alpha}\ & \frac{1}{2}\alpha^{\mathrm{T}}Q\alpha\\ & 0\leqslant\alpha_i\leqslant 1,\quad i=1,\cdots,l\\ & e^{\mathrm{T}}\alpha\geqslant\nu l\\ & y^{\mathrm{T}}\alpha=0\end{aligned}$$

其中 $Q_{ij}\equiv y_iy_jK\left(x_i,x_j\right)$. 决策函数为

$$\operatorname{sign}\left(\sum_{i=1}^{l}y_i\left(\frac{\alpha_i}{\rho}\right)K\left(x_i,x\right)+b\right)$$

那么两个边界与 C 支持向量分类器的相同, 为

$$y_i\left(w^{\mathrm{T}}\Phi\left(x_i\right)+b\right)=\pm 1$$

6.2　二次规划问题的解决

6.2.1　C-SVC 的分解算法

考虑 C-SVC 的一般形式

$$\begin{aligned}\min_{\alpha}\ & \frac{1}{2}\alpha^{\mathrm{T}}Q\alpha+p^{\mathrm{T}}\alpha\\ & y^{\mathrm{T}}\alpha=\Delta\\ & 0\leqslant\alpha_t\leqslant C,\quad t=1,\cdots,l\end{aligned}\tag{6.4}$$

其中 $y_t=\pm 1,\quad t=1,\cdots,l$. 解决 (6.4) 式的困难在于 Q 的密度, 因为 Q_{ij} 一般不是零.

在 Libsvm 中使用了分解算法. 分解方法如下:

(1) 给定数值 $q\leqslant l$ 作为工作集的容量, 找到 α^1 作为初始解, 令 $k=1$.

(2) 如果 α^k 是 (6.2) 式的优化解, 停止. 否则, 找到一个容量是 q 工作集 $B\subset\{1,\cdots,l\}$. 定义 $N\equiv\{1,\cdots,l\}\backslash B$ 和 α_B^k 以及 α_N^k 分别作为 α^k 相对于 B 和 N 的子向量.

(3) 解以下带变量 α_B 的子问题

$$\begin{aligned}\min_{\alpha_B}\ & \frac{1}{2}\alpha_B^{\mathrm{T}}Q_{BB}\alpha_B+\left(p_B+Q_{BN}\alpha_N^k\right)^{\mathrm{T}}\alpha_B \\ & 0\leqslant(\alpha_B)_t\leqslant C,\quad t=1,\cdots,q \\ & y_B^{\mathrm{T}}\alpha_B=\Delta-y_N^{\mathrm{T}}\alpha_N^k\end{aligned}\tag{6.5}$$

其中 $\begin{bmatrix}Q_{BB} & Q_{BN}\\ Q_{NB} & Q_{NN}\end{bmatrix}$ 是矩阵 Q 的转置.

(4) 设 α_B^{k+1} 是 (6.5) 式的优化解, 并且 $\alpha_N^{k+1}\equiv\alpha_N^k$. 使 $k\leftarrow k+1$ 并且返回第二步.

分解方法的基本思想是在每一个循环, 把训练集的 $\{1,\cdots,l\}$ 的元素分解为两个集 B 和 N, 其中 B 是工作集, 并且 $N\equiv\{1,\cdots,l\}\backslash B$. 由于 α_N 是固定的, 所以目标值为

$$\frac{1}{2}\alpha_B^{\mathrm{T}}Q_{BB}\alpha_B-\left(p_B-Q_{BN}\alpha_N\right)^{\mathrm{T}}\alpha_B+\frac{1}{2}\alpha_N^{\mathrm{T}}Q_{NN}\alpha_N-p_N^{\mathrm{T}}\alpha_N$$

然后解这个含有变量 α_B 一个子问题, 即解得 (6.5) 式. 最后更新工作集 B, 并进入下一个循环.

6.2.2 工作集的选择和停止循环的标准

分解方法的一个重要的方面是如何选择工作集 B. (6.4) 式的 KKT 条件显示有一个标量 b 和两个非负向量 λ 和 μ, 优化问题的原始形式和对偶形式中 λ 和 μ 在 KKT 条件下相同, 那么有

$$Q\alpha+p+by\begin{cases}\geqslant 0, & \alpha=0\\ =0, & 0<\alpha<C\\ \leqslant 0, & \alpha=C\end{cases}$$

如果 $y_i=\pm1,\ \ i=1,\cdots,l$, 假设 $C>0$, 上面的 KKT 条件可以表示为

$$\left.\begin{aligned}&y_t=1,\alpha_t<C\ \Rightarrow\ (Q_\alpha+p)_t+b\geqslant 0\ \Rightarrow\ b\geqslant-(Q_\alpha+p)_t=-\nabla f(\alpha)_t\\ &y_t=-1,\alpha_t>0\ \Rightarrow\ (Q_\alpha+p)_t-b\leqslant 0\ \Rightarrow\ b\geqslant(Q_\alpha+p)_t=\nabla f(\alpha)_t\\ &y_t=-1,\alpha_t<C\ \Rightarrow\ (Q_\alpha+p)_t-b\geqslant 0\ \Rightarrow\ b\leqslant(Q_\alpha+p)_t=\nabla f(\alpha)_t\\ &y_t=1,\alpha_t>0\ \Rightarrow\ (Q_\alpha+p)_t+b\leqslant 0\ \Rightarrow\ b\leqslant-(Q_\alpha+p)_t=-\nabla f(\alpha)_t\end{aligned}\right\}\tag{6.6}$$

其中 $f(\alpha)\equiv\frac{1}{2}\alpha^{\mathrm{T}}Q\alpha+p^{\mathrm{T}}\alpha$ 和 $\nabla f(\alpha)$ 是 $f(\alpha)$ 在 α 的梯度. 考虑

$$i\equiv\arg\max(\ \{-\nabla f(\alpha)_t\,|y_t=1,\alpha_t<C\},\ \ \{\nabla f(\alpha)_t\,|y_t=-1,\alpha_t>0\}\ )\tag{6.7}$$

$$j\equiv\arg\min(\ \{\nabla f(\alpha)_t\,|y_t=-1,\alpha_t<C\},\ \ \{-\nabla f(\alpha)_t\,|y_t=1,\alpha_t>0\}\ )\tag{6.8}$$

$B=\{i,j\}$ 作为分解方法子问题 (6.5) 式的工作集. 这里 i 和 j 是最达不到 KKT 条件的两个元素.

定义

$$g_i \equiv \begin{cases} -\nabla f(\alpha)_i, & \text{如果} y_i = 1, \alpha_i < C \\ \nabla f(\alpha)_i, & \text{如果} y_i = -1, \alpha_i > 0 \end{cases}$$

以及

$$g_j \equiv \begin{cases} -\nabla f(\alpha)_j, & \text{如果} y_i = -1, \alpha_i < C \\ \nabla f(\alpha)_j, & \text{如果} y_i = 1, \alpha_i > 0 \end{cases}$$

从 (6.6) 式得到

$$g_i \leqslant -g_j$$

这就意味着 α 是优化问题 (6.2) 式的一个解.

循环停止的标准为

$$g_i \leqslant -g_j + \varepsilon$$

其中 ε 是一个小正数.

6.2.3　v 支持向量分类的分解方法

考虑 v 支持向量分类的一般形式 [74]

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2}\alpha^{\mathrm{T}} Q\alpha + p^{\mathrm{T}}\alpha \\ & y^{\mathrm{T}}\alpha = \Delta_1 \\ & e^{\mathrm{T}}\alpha = \Delta_2 \\ & 0 \leqslant \alpha_t \leqslant C, \quad t = 1, \cdots, l \end{aligned} \tag{6.9}$$

其中　$y_t = \pm 1, \quad t = 1, \cdots, l.$

在 Libsvm 中使用的 v 支持向量分类的分解算法 C 支持向量分类的分解算法相同, 但是其子问题的表示方法有所不同

$$\begin{aligned} \min_{\alpha_B} \ & \frac{1}{2}\alpha_B^{\mathrm{T}} Q_{BB}\alpha_B + \left(p_B + Q_{BN}\alpha_N^k\right)^{\mathrm{T}} \alpha_B \\ & 0 \leqslant (\alpha_B)_t \leqslant C, \quad t = 1, \cdots, q \\ & y_B^{\mathrm{T}}\alpha_B = \Delta_1 - y_N^{\mathrm{T}}\alpha_N^k \\ & e_B^{\mathrm{T}}\alpha_B = \Delta_2 - e_N^{\mathrm{T}}\alpha_N^k \end{aligned} \tag{6.10}$$

如果仅选择了两个元素 i 和 j, 且 $y_i \neq y_j$, 那么 $y_B^{\mathrm{T}}\alpha_B = \Delta_1 - y_N^{\mathrm{T}}\alpha_N^k$ 和 $e_B^{\mathrm{T}}\alpha_B = \Delta_2 - e_N^{\mathrm{T}}\alpha_N^k$ 表示有两个方程带有两个变量, 所以 (6.10) 式仅有一个可行点, 因此其解就是 α_k. 另外, 如果 $y_i = y_j$, 那么 $y_B^{\mathrm{T}}\alpha_B = \Delta_1 - y_N^{\mathrm{T}}\alpha_N^k$ 和 $e_B^{\mathrm{T}}\alpha_B = \Delta_2 - e_N^{\mathrm{T}}\alpha_N^k$ 相同, 那么 (6.10) 式就有多个可行解, 因此在选择工作集时保持 $y_i = y_j$.

由 (6.9) 式得

$$\nabla f(\alpha)_i - \rho + by_i \begin{cases} = 0, & 0 < \alpha_i < C \\ \geqslant 0, & \alpha_i = 0 \\ \leqslant 0, & \alpha_i = C \end{cases}$$

若

$$r_1 \equiv \rho - b, \quad r_2 \equiv \rho + b$$

如果 $y_i = 1$, 那么 KKT 条件变为

$$\nabla f(\alpha)_i - r_1 \begin{cases} \geqslant 0, & \alpha_i < C \\ \leqslant 0, & \alpha_i > 0 \end{cases}$$

如果 $y_i = -1$, 那么 KKT 条件变为

$$\nabla f(\alpha)_i - r_2 \begin{cases} \geqslant 0, & \alpha_i < C \\ \leqslant 0, & \alpha_i > 0 \end{cases}$$

因此, 工作集元素 i 和 j 从

$$\begin{aligned} i &\equiv \arg\min_t \left(\{\nabla f(\alpha)_t \,|y_t = 1, \quad \alpha_t < C\}\right) \\ j &\equiv \arg\max_t \left(\{\nabla f(\alpha)_t \,|y_t = 1, \quad \alpha_t > 0\}\right) \end{aligned} \tag{6.11}$$

中选择还是从

$$\begin{aligned} i &\equiv \arg\min_t \left(\{\nabla f(\alpha)_t \,|y_t = -1, \quad \alpha_t < C\}\right) \\ j &\equiv \arg\max_t \left(\{\nabla f(\alpha)_t \,|y_t = -1, \quad \alpha_t > 0\}\right) \end{aligned} \tag{6.12}$$

中选择依赖于那个工作集能给出较小的 $\nabla f(\alpha)_i - \nabla f(\alpha)_j$.

6.2.4 解析解法

(6.5) 式可以表示为只有两个变量的简单问题 [82]

$$\begin{aligned} &\min_{\alpha_i, \alpha_j} \frac{1}{2} \begin{bmatrix} \alpha_i & \alpha_j \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (Q_{i,N}\alpha_N - 1)\alpha_i + (Q_{j,N}\alpha_N - 1)\alpha_j \\ &y_i\alpha_i + y_j\alpha_j = \Delta' \equiv \Delta - y_N^{\mathrm{T}}\alpha_N^k \\ &0 \leqslant \alpha_i, \quad \alpha_j \leqslant C \end{aligned} \tag{6.13}$$

在 (6.5) 式的目标函数中替换 $\alpha_i = y_i\left(\Delta - y_N^{\mathrm{T}}\alpha_N - y_j\alpha_j\right)$, 并且在 α_j 上解一个非限制性的最小化值. 可以得到下面的解:

$$\alpha_j^{\mathrm{new}} = \begin{cases} \alpha_j + \dfrac{-G_i - G_j}{Q_{ii} + Q_{jj} + 2Q_{ij}}, & y_i \neq y_j \\ \alpha_j + \dfrac{G_i - G_j}{Q_{ii} + Q_{jj} - 2Q_{ij}}, & y_i = y_j \end{cases} \tag{6.14}$$

其中

$$G_i \equiv \nabla f(\alpha)_i \text{并且} G_j \equiv \nabla f(\alpha)\, j$$

如果这个值落在 α_j 的可行域之外, 那么 (6.11) 式的值也会超出可行域. 那么 α_j 就要被赋予一个新值. 例如, 如果 $y_i = y_j$ 并且 $C \leqslant \alpha_i + \alpha_j \leqslant 2C$, α_j^{new} 必须满足下面式子:

$$L \equiv \alpha_i + \alpha_j - C \leqslant \alpha_j^{\mathrm{new}} \leqslant C \equiv H$$

α_i^{new} 和 α_j^{new} 最大取值可以是 C. 那么如果

$$\alpha_j + \frac{G_i - G_j}{Q_{ii} + Q_{jj} + 2Q_{ij}} \leqslant L$$

这时令

$$\alpha_j^{\mathrm{new}} = L$$

那么

$$\alpha_i^{\mathrm{new}} = \alpha_i + \alpha_j - \alpha_j^{\mathrm{new}} = C \tag{6.15}$$

这相当于使用线段来优化二次函数. 这个线段为线性限制条件

$$y_i\alpha_i + y_j\alpha_j = \Delta'$$

和边界限制条件

$$0 \leqslant \alpha_i, \quad \alpha_j \leqslant C$$

中间的一部分.

从数值上来说, 最后的方程 (6.15) 可能不成立, 即会发生浮点运算

$$\begin{aligned} &\alpha_i + \alpha_j - \alpha_j^{\mathrm{new}} \\ =&\alpha_i + \alpha_j - (\alpha_i + \alpha_j - C) \\ \neq& C \end{aligned}$$

因此在多数的 SVM 软件中需要指定一个小的偏差 ε_α, 并且认为所有的 $\alpha_i \geqslant C - \varepsilon_\alpha$ 都是上界、所有 $\alpha_i \leqslant \varepsilon_\alpha$ 都是零. 在有些数据被误认为是支持向量时这种指定是必要的. 另外, 计算偏差项也是出于正确确定自由拉格朗日乘子 $\alpha_i\,(0 \leqslant \alpha_i \leqslant C)$ 的需要.

6.2.5 b 和 ρ 的计算

由于 b 和 ρ 包含于决策函数中, 所以取得了对偶优化问题的解 α 后必须计算这两个参数. 当 $y_i = 1$ 时, 如果 α_i 满足 $(0 \leqslant \alpha_i \leqslant C)$, 那么 $r_1 = \nabla f(\alpha)_i$. 实际上为了规避数量错误, 必须求平均值

$$r_i = \frac{\sum_{0<\alpha_i<C, y_i=1} \nabla f(\alpha)_i}{\sum_{0<\alpha_i<C, y_i=1} 1}$$

另外, 如果 α_i 不满足 $(0 \leqslant \alpha_i \leqslant C)$, 那么 r_i 必须满足

$$\max_{\alpha_i=C, y_i=1} \nabla f(\alpha)_i \leqslant r_1 \leqslant \min_{\alpha_i=0, y_i=1} \nabla f(\alpha)_i$$

这时取 r_1 作为取值范围的中点 [74].

对于 $y_i = -1$, 依据同样方法可以计算出 r_2. 计算得到了 r_1 和 r_2 后, 可以求得 b 和 ρ, 有

$$\rho = \frac{r_1 + r_2}{2} 和 - b = \frac{r_1 - r_2}{2}$$

现在 KKT 条件可以写成

$$\max_{\alpha_i>0, y_i=1} \nabla f(\alpha)_i \leqslant \min_{\alpha_i<C, y_i=1} \nabla f(\alpha)_i$$

与

$$\max_{\alpha_i>0, y_i=-1} \nabla f(\alpha)_i \leqslant \min_{\alpha_i<C, y_i=-1} \nabla f(\alpha)_i$$

在这种情况下可以使用下面条件停止循环: 如果循环 α 满足下面的条件:

$$\max \begin{pmatrix} -\min\limits_{\alpha_i<C, y_i=1} \nabla f(\alpha)_i + \max\limits_{\alpha_i>0, y_i=1} \nabla f(\alpha)_i \\ -\min\limits_{\alpha_i<C, y_i=-1} \nabla f(\alpha)_i + \max\limits_{\alpha_i>0, y_i=-1} \nabla f(\alpha)_i \end{pmatrix} < \varepsilon \tag{6.16}$$

分解方法停止. 其中 $\varepsilon > 0$ 是预先选择的停止偏差.

6.3 压缩和缓存

6.3.1 压缩

很多问题中自由支持向量 (即 $0 < \alpha_i < C$) 的数目很小, 压缩技术减小了没有考虑到边界变量的预解决问题的工作集大小. 循环过程即将结束时, 分解方法能够确定可行集合 A. 集合 A 中包含几乎所有的自由拉格朗日乘子 α_i[107]. 下面的理

论表明在选择工作集和循环停在标准时, 在假定的分解方法循环的结尾处仅仅对应于小集合的变量仍然可以变化.

如果 $\{\alpha^k\}$ 是由分解算法产生的序列, 那么任何收敛的子序列的极限是 (6.4) 式的优化解. 因此当 $\lim\limits_{k\to\infty}\alpha^k=\bar{\alpha}$ 时, $\bar{\alpha}$ 是一个优化解. 另外, k 足够大时仅仅在

$$\{t|-y_t\nabla f(\bar{\alpha})_t\}=\begin{cases}\max\left(\max\limits_{\bar{\alpha}_t<C,y_t=1}-\nabla f(\bar{\alpha})_t,\quad\max\limits_{\bar{\alpha}_t>0,y_t=-1}\nabla f(\bar{\alpha})_t\right)\\ \min\left(\min\limits_{\bar{\alpha}_t<C,y_t=-1}-\nabla f(\bar{\alpha})_t,\quad\min\limits_{\bar{\alpha}_t>0,y_t=1}\nabla f(\bar{\alpha})_t\right)\end{cases}$$

中的元素可以被替换.

因此在几个循环后当变量 α_i 等于 C 时, 在最终解中这个变量仍然为上界. 那么 (6.2) 式可以通过解小一些的问题来代替

$$\begin{aligned}\min\ &\frac{1}{2}\alpha_A^{\mathrm{T}}Q_{AA}\alpha_A-\left(p_A-Q_{AN}\alpha_N^k\right)^{\mathrm{T}}\alpha_A\\ &0\leqslant(\alpha_A)_t\leqslant C,t=1,\cdots,l\\ &y_A^{\mathrm{T}}\alpha_A=\Delta-y_N^{\mathrm{T}}\alpha_N^k\end{aligned}\tag{6.17}$$

其中 $N=\{1,\cdots,l\}\setminus A$.

Libsvm 从开始就进行压缩过程. 其过程如下:

(1) 每一个循环 $\min(l,1000)$ 后, 试图压缩一些变量. 注意在循环过程

$$\begin{aligned}&\min\Big(\ \{\nabla f(\alpha_k)_t|y_t=-1,\alpha_t<C\},\quad\{\nabla f(\alpha_k)_t|y_t=1,\alpha_t>0\}\ \Big)=g_j\\ <\ &g_i=\max\Big(\ \{-\nabla f(\alpha_k)_t|y_t=1,\alpha_t<C\},\quad\{\nabla f(\alpha_k)_t|y_t=-1,\alpha_t>0\}\ \Big)\end{aligned}$$

中 $g_i\leqslant -g_j$ 这个条件并没有满足.

如果

$$g_t=\begin{cases}-\nabla f(\alpha)_t, & y_t=1, \quad \alpha_t<C\\ \nabla f(\alpha)_t, & y_t=-1, \quad \alpha_t>0\end{cases}\tag{6.18}$$

且

$$g_t\leqslant -g_j\tag{6.19}$$

若 α_t 残基位于边界上, 那么 α_t 的值不变. 固定这个变量. 同样, 对于那些

$$g_t=\begin{cases}-\nabla f(\alpha)_t, & y_t=-1, \quad \alpha_t<C\\ \nabla f(\alpha)_t, & y_t=1, \quad \alpha_t>0\end{cases}\tag{6.20}$$

如果

$$g_t\geqslant -g_j$$

且 α_t 残基位于边界上, 这个变量也被固定. 因此集合 A 在每一个循环 $\min(l, 1000)$ 中动态地减少.

(2) 上面的压缩方法很苛刻. 因为分解方法收敛速度很慢, 并且大部分循环对于完成最终需要的准确率是无效的. 因为 (6.17) 式的错误压缩会浪费运算时间. 因此当分解方法首先完成容限

$$g_t \leqslant -g_j + 10\varepsilon$$

其中 ε 是指定的循环停止偏差. 在重新构建整个斜率后基于正确的信息使 (6.18) 式和 (6.20) 式的循环固定一些变量, 分解算法得以继续.

在 Libsvm 中, (6.17) 式中集合 A 的大小是动态减小的. 为了减小重新构建斜率 $\nabla f(\alpha)$ 的计算成本, 在循环期间, 总是保持

$$\bar{G}_i = C \sum_{\alpha_j = C} Q_{ij}, \quad i = 1, \cdots, l$$

然后为了得到斜率 $\nabla f(\alpha)$, $i \notin A$, 有

$$\nabla f(\alpha)_i = \sum_{j=1}^{l} Q_{ij}\alpha_j = \bar{G}_i + \sum_{0<\alpha_j<C} Q_{ij}\alpha_j$$

6.3.2 缓存

另一项减小计算成本的技术是缓存. 因为 Q 是高密度的, 而且没有保存在计算机的内存中, 在需要的时候要计算元素 Q_{ij}. 那么当前的 Q_{ij} 可以储存在缓存中.

6.4 多元分类

Libsvm 中使用 "一对一" 的方法进行多元分类. 应用这种方法需要构建 $k(k-1)/2$ 个二元分类器, 每个分类器对两个不同的训练数据集进行分类. 为了训练从第 i 类和第 j 个类数据集中训练数据, 需要解下面的二元分类问题:

$$\begin{aligned}
\min_{w^{ij}, b^{ij}, \xi^{ij}} \quad & \frac{1}{2}\left(w^{ij}\right)^{\mathrm{T}} w^{ij} + C\left(\sum_t \left(\xi^{ij}\right)_t\right) \\
& \left(\left(w^{ij}\right)^{\mathrm{T}} \Phi(x_t)\right) + b^{ij} \geqslant 1 - \xi_t^{ij}, \quad \text{如果}x_t\text{在第}i\text{类} \\
& \left(\left(w^{ij}\right)^{\mathrm{T}} \Phi(x_t)\right) + b^{ij} \leqslant -1 + \xi_t^{ij}, \quad \text{如果}x_t\text{在第}j\text{类} \\
& \xi_t^{ij} \geqslant 0
\end{aligned}$$

分类过程中, Libsvm 使用投票策略: 每次二元分类被看成一次投票过程, 每个数据点看成是一个选票, 最后该数据点被归类为得票最多的类. 如果两类具有相同的票数, 由于没有找到更好的方法 Libsvm 认为它属于索引序号较小的类.

另一种多元分类方法为 “一对多”, 其中构建了 k 个 SVM 模型, 其中第 i 个 SVM 模型被第 i 类中所有的样本训练, 在第 i 类中一部分样本的标签为正, 另一部分为负. Libsvm 没有采用这种策略.

虽然程序训练了 $k(k-1)/2$ 个子分类器, 因为每个子问题都很小 (数据仅来自两个类), 总的训练时间并不多于 “一对多” 方法.

6.5　非平衡数据集

对于一些分类问题, 每个类别中数据的数量是不平衡的. 那么一些研究者设想在 SVM 公式中使用不同的罚分参数 [72]: 例如, C-SVM 变成

$$\begin{aligned}\min_{w,b,\xi}\quad & \frac{1}{2}w^{\mathrm{T}}w+C_{+}\sum_{y_i=1}\xi_i+C_{-}\sum_{y_i=-1}\xi_i\\ & y_i\left(w^{\mathrm{T}}\Phi\left(x_i\right)+b\right)\geqslant 1-\xi_i\\ & \xi_i\geqslant 0,\quad i=1,\cdots,l\end{aligned}$$

它的对偶形式为

$$\begin{aligned}\min_{\alpha}\quad & \frac{1}{2}\alpha^{\mathrm{T}}Q\alpha-e^{\mathrm{T}}\alpha\\ & 0\leqslant\alpha_i\leqslant C_{+},\quad \text{如果}y_i=1\\ & 0\leqslant\alpha_i\leqslant C_{-},\quad \text{如果}y_i=-1\\ & y^{\mathrm{T}}\alpha=0\end{aligned}$$

对于使用不同的 C_i, $i=1,\cdots,l$, 替换 C, 前面进行的解析分析的很大部分仍然正确. 现在使用 C_+ 和 C_- 只是特例. 因此运行结果几乎是一样的. 但是前面的子问题 (6.13) 式的解成为

$$\begin{aligned}\min_{\alpha_i,\alpha_j}\ & \frac{1}{2}\left[\alpha_i\quad\alpha_j\right]\left[\begin{array}{cc}Q_{ii} & Q_{ij}\\ Q_{ji} & Q_{jj}\end{array}\right]\left[\begin{array}{c}\alpha_i\\ \alpha_j\end{array}\right]+\left(Q_{i,N}\alpha_N-1\right)\alpha_i+\left(Q_{j,N}\alpha_N-1\right)\alpha_j\\ & y_i\alpha_i+y_j\alpha_j=\Delta-y_N^{\mathrm{T}}\alpha_N^k\\ & 0\leqslant\alpha_i\leqslant C_i,\ 0\leqslant\alpha_j\leqslant C_j\end{aligned}$$

其中 C_i 和 C_j 是否可以由 C_+ 和 C_- 代替取决于 y_i 和 y_j.

6.6　模型的选择

Libsvm 提供了一个使用 RBF 核函数的模型选择工具: 平行搜索的交叉验证试验. 虽然这个工具目前仅仅支持 C-SVC 中两个参数 C 和 γ, 然而这个工具可以轻而易举地进行修改而适用于其他核函数, 如线性核和多项式核.

在使用这个工具的过程中, 使用者首先必须提供一个可能的 C(或 γ) 的步长. 然后计算每一组 (C,γ) 格点的值, 看看哪个格点能给出最高的交叉验证准确率. 然后使用者就可以使用最好的参数训练整个训练集并得到最终的模型. 在进行多元分类时, 所有的二元分类模型都使用相同的 (C,γ).

6.7 预测蛋白质结构中运用 Libsvm 的基本操作方法

Libsvm 软件包操作简单、容易使用, 实验中, 选择了 Linux 操作系统安装 Libsvm. 因为实验只涉及模式识别的问题, 所以对于回归函数估计问题这里不作讨论.

使用 Libsvm 的第一步是向量化残基序列, 即把字母形式的残基序列转化成为向量形式. 这种转化称作嵌入, 后面一章详细介绍嵌入的方法. 得到的向量要转化成为 Libsvm 可用的形式 [84,85]. Libsvm 要求向量为文本书件, 其格式为 "类别向量", 即文件的第一个位置为欲分类向量的类别. 这个类别一般用一个整数表示, 操作者可以自己定义, 比如 -1、0 或 1 等. 把所有要分类的样本按照上面的格式都写入一个文件中. 然后使用 grid.py 程序对其进行优化. 通过这个优化可以得到相关的优化参数, 并且 grid.py 还会以图形形式给出优化结果. 使用这个优化结果可以得到样本的最优分辨率. 利用上面优化得到的参数、使用 train 程序就可以进行训练支持向量机的工作了.

下面说明train程序的参数. train的参数说明如下(http://www.csie.ntu.edu.tw/~cjlin/libsvm/):

(1) 参数的格式例子: -s 0 -c 1000 -t 1 -g 1 -r 1 -d 3 表示: 多项式核函数二元分类 $(u'v+1)^3$, 其中 $C=1000$.

(2) -s: 支持向量机的类型 (默认是 0). 其中 0 表示 C- 支持向量分类; 1 表示 nu- 支持向量分类; 2 表示一类支持向量机; 3 表示 epsilon- 支持向量回归; 4 表示 nu- 支持向量回归.

(3) -t: 核函数的类型 (默认是 2). 其中 0 表示线性内核: $u'\times v$; 1 表示多项式内核 $(\text{gamma}\times u'\times v+\text{coef0})^{\text{degree}}$; 2 表示径向基内核: $\exp\left(-\text{gamma}\times|u-v|^2\right)$; 3 表示 S 函数: $\tanh(\text{gamma}\times u'\times v+\text{coef0})$.

(4) -d, degree: 设置核函数的乘方数 (默认为 3).

(5) -g, gamma: 设置核函数中 gamma 的值 (默认为 $\frac{1}{k}$).

(6) -r, cost: 设置 C- 支持向量分类中参数 C 的值 (默认为 0).

(7) -c, cost: 设置 C- 支持向量分类、epsilon- 支持向量回归和 nu- 支持向量回归中参数 C 的值 (默认为 1).

(8) -n, nu: 设置 nu- 支持向量分类、一类支持向量机和 nu- 支持向量回归中参

数 nu 的值 (默认为 0.5).

(9) -p, epsilon：设置 epsilon- 支持向量回归中损失函数 epsilon 的值 (默认为 0.1).

(10) -m, 缓存的大小：设置缓存 (单位 MB, 默认为 40).

(11) -e, epsilon：设置中止条件的容限 (默认为 0.001).

(12) -h, 缩减：是否使用启发式缩减, 0 或 1(默认为 1).

(13) -wi, 权重：设置 i 类参数 C 为权重乘以 C, 对于 C- 支持向量分类 (默认为 1).

(14) -v: 计算交叉验证时把训练集分割的数量.

第 7 章　蛋白质二级结构预测

7.1　蛋白质结构

蛋白质序列的基本单位是氨基酸, 在天然状态下可以构成蛋白质的氨基酸共有 20 种, 它们都是 L- 型氨基酸. 不同氨基酸之所以可以形成不同的三维结构, 主要区别在于它们侧链的大小、形状、反应性和形成氢键的能力不同. 蛋白质的分子结构可分为一级、二级、超二级结构、三级、四级结构以及分子缔合体六个层次, 后三者统称为高级结构或空间构象. 蛋白质的空间构象涵盖了蛋白质分子中每一个原子在三维空间的相对位置. 并非所有蛋白质都有四级结构, 由二条或二条以上多肽链形成的蛋白质才有四级结构 [35]. 蛋白质的折叠是有序的、由疏水作用力推动的协同过程. 伴侣分子在蛋白质的折叠中起着辅助性的作用. 蛋白质多肽链在生理条件下折叠成特定的构象符合热力学原理的作用过程, 即可以保持分子处于结构上最稳定的状态. 折叠的天然蛋白质在变性因素影响下可以失去活性. 在某些条件下, 变性的蛋白质可能会恢复活性. X 射线晶体衍射和核磁共振是测定蛋白质以及其他生物大分子结构的有效方法.

7.1.1　蛋白质的一级结构

蛋白质的一级结构是指蛋白质分子中氨基酸的排列顺序. 主要化学键是肽键和二硫键. 参与肽键的 6 个原子 $C_{\alpha 1}$、C、O、N、H、$C_{\alpha 2}$ 位于同一平面, 且 $C_{\alpha 1}$、$C_{\alpha 2}$ 在平面上所处的位置为反式构型, 此 6 个原子即构成了肽单元, 其基本结构见图 7-1.

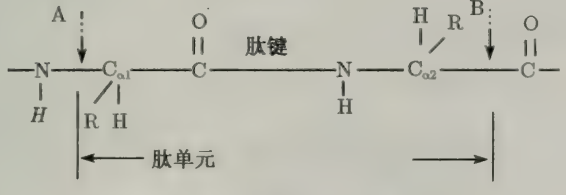


图 7-1　肽单元示意图

图 7-1 中的 A、B 键是单键, 可在一定程度上自由旋转, 也正由于这两个单键的自由旋转角度, 决定了相邻肽单元之间的相对空间位置. 其中的肽键有一定程度双键性质, 不能自由旋转. 两个氨基酸残基之间通过肽键相互链接. 肽基或肽单元

是有极性的, 也是一种具刚性的平面. N—C_α 和 C_α—C 单键旋转的角度分别用 Φ 和 ψ 描述. 这两个角旋转的角度决定两个相邻肽基的空间位置. 如果这两个旋转角分别相等, 则多肽链主链是有规律的构象.

一级结构是蛋白质空间结构和特异生物学功能的基础. 氨基酸排列顺序的差别意味着从多肽链骨架伸出的侧链 R 基团的性质和顺序对于每一种蛋白质是特异的 —— 因为 R 基团大小不同, 所带电荷数目不同, 对水的亲和力不相同, 所以蛋白质的空间构象也不同.

一级结构中有些氨基酸的作用却是非常重要的, 若蛋白质分子中起关键作用的氨基酸残基缺失或被替代, 都会严重影响其空间构象或生理功能, 产生某种疾病, 这种由蛋白质分子发生变异所导致的疾病, 称为 "分子病". 蛋白质一级结构与功能的关系如下:

(1) 一级结构是空间构象和功能的基础, 空间构象遭破坏的多肽链只要其肽键未断, 一级结构未被破坏, 就能恢复到原来的三级结构, 功能依然存在;

(2) 即使是不同物种之间的多肽和蛋白质, 只要其一级结构相似, 其空间构象及功能也越相似;

(3) 物种越接近, 其同类蛋白质一级结构越相似, 功能也相似.

7.1.2　蛋白质的二级结构特征

蛋白质的二级结构指蛋白质分子中某一段肽链的局部空间结构, 也就是该段肽链主链骨架原子的相对空间位置, 并不涉及氨基酸残基侧链的构象. 维系二级结构的化学键主要是氢键. 二级结构的主要形式包括: α 螺旋结构、β 折叠和无规则卷曲, 在 α 螺旋和 β 折叠中, 这两个旋转角都是分别相等的. 因此, α 螺旋和 β 折叠是有规律的构象, 其中螺旋是各种二级结构中最具刚性、最致密、最稳定的构象. 它们是构成蛋白质高级结构的基本要素. 由于蛋白质结构中氨基酸残基之间的空间位置没有一种特定的模式, 并且残基之间, 以及残基的分子之间在生理环境中不是僵化不变的, 所以目前对于蛋白质二级结构的确切定义还没有统一定论, 各种不同版本的定义都是根据不同的具体需要定义的.

7.1.2.1　α 螺旋

从蛋白质晶体的 X 射线衍射图中看到有 0.5 ～ 0.55nm 的重复单位, 这种重复性结构一般为 α 螺旋. α 螺旋的结构特点如下 [41]:

(1) 多个肽键平面通过 α 碳原子旋转, 相互之间紧密盘曲成稳固的右手螺旋.

(2) 主链呈螺旋上升, 每 3.6 个氨基酸残基为一个循环, 每个氨基酸残基向上平移 0.15nm, 螺距 0.54nm. 这与 X 射线衍射图相吻合.

(3) 相邻两个螺旋之间借肽键中的 C═O 双键和 N—H 单键之间形成许多链内

氢键, 即每一个氨基酸残基中的 N—H 和前面相隔三个残基的 C=O 之间形成氢键, 氢键的方向与螺旋长轴基本平行, 这是使 α 螺旋形成具有稳定结构的主要化学键.

(4) 肽链中氨基酸侧链 R, 分布在螺旋外侧, 其形状、大小及电荷影响 α 螺旋的形成. 酸性或碱性氨基酸集中的区域, 由于同电荷相斥, 不利于 α 螺旋形成. 较大的 R(如苯丙氨酸、色氨酸、异亮氨酸) 集中的区域, 也妨碍 α 螺旋形成. 脯氨酸因其 α 碳原子位于五元环上, 不易扭转, 加之它是亚氨基酸, 不易形成氢键, 故不易形成上述 α 螺旋. 甘氨酸的 R 基为 H, 空间占位很小, 也会影响该处螺旋的稳定.

7.1.2.2 β 折叠

从蛋白质晶体的 X 射线衍射图中看到有 0.7nm 的重复单位. 两段以上的这种折叠成锯齿状肽链, 通过氢键相连而平行成片层状的结构称为 β 折叠. β 片层结构特点是 [41]:

(1) β 折叠是肽链相当伸展的结构, 肽链平面之间折叠成锯齿状, 相邻肽键平面间呈 110° 角. 氨基酸残基的 R 侧链伸出在锯齿的上方或下方.

(2) 依靠两条肽链或一条肽链内的两段肽链间的 C=O 双键与 N—H 单键之间形成氢键, 此氢键方向与折叠的长轴垂直. 这是使 β 折叠形成具有稳定结构的主要化学键.

(3) 两段肽链可以是平行的, 也可以是反平行的. 平行的 β 折叠从 "N 端" 到 "C 端" 是同方向的, 反平行的 β 折叠从 "N 端" 到 "C 端" 是反方向的. β 折叠的形式十分多样, 正、反平行能相互交替.

(4) 平行的 β 折叠结构中, 两个残基的间距为 0.65nm; 反平行的 β 片层结构, 则间距为 0.7nm.

7.1.2.3 β 转角

蛋白质分子中, 肽链经常会出现 180° 的回折, 通常由 4 个氨基酸残基组成, 在这种回折角处的构象就是 β 转角. β 转角中, 第一个氨基酸残基的 C=O 双键与第四个残基的 N—H 单键之间形成氢键, 从而使结构保持相对稳定. β 转角第二个残基常为脯氨酸, 因为其 N 原子位于环中, 形成肽键 N 原子上已没有 H 原子, 不能再形成氢键, 因而走向发生转折. β 转角常发生在蛋白质分子的表面, 这与蛋白质的生物学功能有关.

7.1.2.4 无规卷曲

没有确定规律性的部分肽链构象称为无规则卷曲. 肽链中肽键平面不规则排列, 属于松散的无规卷曲.

7.1.3　蛋白质结构域、三级结构与四级结构

在大多数球状蛋白质中, 往往可以观察到可明显区分的二级结构组合. 这种组合称为超二级结构或基元. 基元也许具有结构和功能上的作用. 分子较大的多肽常折叠成两个或多个球状簇, 这种球状簇叫做结构域或域结构. 大多数域结构由100～200 个氨基酸残基构成, 平均直径约 2.5nm. 一条多肽链在一个域范围内来回折叠, 但相邻的域常被一个或两个多肽片段连接. 因而域在结构上是独立的、具有小分子球状蛋白质的特性的单位. 域结构往往有特殊的功能, 例如结合小分子.

三级结构主要针对球状蛋白质而言, 是指主链和侧链在空间中的走向. 在球状蛋白质中, 侧链基团的定位是根据它们的极性安排的. 蛋白质特定的空间构象由氢键、离子键、偶极与偶极间的相互作用、范德华力以及疏水作用等作用力维持, 疏水作用是主要的作用力. 有些蛋白质还涉及二硫键. 疏水键是蛋白质分子中疏水基团之间的结合力, 酸性和碱性氨基酸的 R 基团可以带电荷, 正负电荷互相吸引形成离子键, 与氢原子共用电子对形成的键为氢键.

蛋白质的四级结构是由有生物活性的两条或多条肽链组成, 肽链与肽链之间不通过共价键相连, 而由非共价键维系. 每条多肽链都有其完整的三级结构, 称为蛋白质的亚基, 这种蛋白质分子中各个亚基的空间排布及亚基接触部位的布局和相互作用, 称为蛋白质的四级结构. 在四级结构中, 各亚基之间的结合力主要是疏水作用, 氢键和离子键也参与维持四级结构. 含有四级结构的蛋白质, 单独的亚基一般没有生物学功能, 只有完整的四级结构才有生物学功能.

7.2　蛋白质二级结构定义

蛋白质二级结构因子的鉴别是确定蛋白质结构的主要步骤. 这种鉴别是以后的可视化、结构比较、分类、同源建模、Threading 和序列比对的基础. 在溶液中的蛋白质的结构不是固定不变的, 结构片段之间柔性程度变化很大, 这种变化对于蛋白质实现其功能来说必不可少 [86].

对于蛋白质二级结构目前仍没有普遍认可的、适用于各个方面应用的定义. 虽然很多人都根据蛋白质空间结构的物理特征进行了定义, 但是这些定义都是针对某种用途的. 每种定义都是根据定义者以往的经验以及对数量众多的结构进行观察总结出来的. 我认为这种定义方法论可以简述为: 看着像、并且这么定义有用. 通过精确的理论推理和计算的结构无疑会对人们理解蛋白质具有更广泛的用途. 下面介绍几种主要的定义, 这些定义分别根据氨基酸序列的氢键、氢键键能和主链扭曲角度即肽键平面的 Φ 和 ψ 角度以及 C_α 原子之间的相对距离来定义.

7.2.1 DSSP 数据库中的蛋白质二级结构特征识别

7.2.1.1 二级结构因子及氢键的定义

1951 年 Linus Pauling 和 Robert Corey 根据氢键和协调性标准对于 α 螺旋和 β 折叠进行了预测. 后来人们使用 X 射线衍射技术看到了这些结构的详细的原子结构. 但是这只是模糊的直觉概念, 只有这些模糊的概念不能满足人们在实验中的要求, 必须依据清晰的算法确认这些结构.

对蛋白质二级结构清晰、客观和准确的定义是正确分析氨基酸序列与蛋白质二级结构的关系的前提条件. DSSP 数据库制订了一套由 X 射线衍射坐标确定的氢键和氨基酸序列的几何特征来识别蛋白质二级结构的方法. 然而到目前为止还没有一个氢键的通用定义, 任何氢键都是根据某种特殊目的、依据经验定义的 [87].

这种方法认为构成二级结构的基本要素为重复的氢键模式 "转弯" 和 "桥". 重复的转弯是 "螺旋"; 重复的桥是 "梯子", 联结在一起的梯子是 "折叠". 几何结构由具有不同几何特征的扭曲和转弯定义. 局部的手性指的是四个连续的碳原子扭曲的方向. 右手螺旋的手性为正, 折叠的手性是负. 卷曲的片段定义为 "弯曲". 溶解的 "暴露" 指的是可能接触到一个残基的水分子的数量.

Pauling 等认为结构模式识别的过程就是从原子坐标提取蛋白质结构特征的过程. 为了区分不同的基本二级结构模式, 必须明确所涉及的参数. 这里定义二级结构主要使用决定氢键有无的参数 —— 键能 [88]. DSSP 数据库中二级结构识别算法主要建立在氢键模式的基础上

(1) "n 转弯". 残基 i 的 C=O 基团与残基 $i+n$ 的 N—H 基团之间的氢键, 其中 $n=3,4,5$.

(2) "桥". 不相邻残基之间的氢键.

这两种模式基本耗尽了所有骨架中的所有氢键. 重复的 "4 转弯" 形成了 α 螺旋, 重复的桥构成了 β 折叠. 基本模式以外的模式还包括 3_{10} 螺旋、π 螺旋、孤立的转弯以及孤立的 β 桥.

蛋白质二级结构等级也可以根据氢键特征定义 [87]:

(1) 基本定义为氢键;

(2) 以氢键定义为基础的转弯和桥;

(3) 在此基础之上定义了 α 螺旋和 β 梯子, 其中包括一般的不完整的二级结构, 比如螺旋纽结和 β 桥;

(4) 几何特征定义为弯曲、手性、二硫键和溶解暴露.

每种结构特征都是独立定义的. 氢键模式定义可以用方程

$$\text{H键}\,(i,j) =: [-0.5\text{kcal/mol}]$$

表示, 其意义为: 如果 E 小于 -0.5kcal/mol, 就存在一个氢键.

7.2.1.2　基本二级结构因子定义

利用 C, O $(+q_1, -q_1)$ 和 N, H $(-q_2, +q_2)$ 原子团之间的部分电量计算的氢键之间的电子能量为

$$E = q_1 q_2 \left(1/r\left(\mathrm{ON}\right) + 1/r\left(\mathrm{CH}\right) - 1/r\left(\mathrm{OH}\right) - 1/r\left(\mathrm{CN}\right)\right) * f$$

其中 $q_1 = 0.42e, q_2 = 0.20e$, 这里 e 为单位电子电量, $r(AB)$ 为 A 和 B 之间的距离, 单位是埃. 空间因子 f=332Å, E 的单位是 kcal/mol. 一个稳定的氢键应该有 -3kcal/mol 的键能. DSSP 指定了一个判断氢键的界限: 如果 E 小于这个界限, 即

$$\text{H键}\left(i, j\right) =: \left[-0.5\text{kcal/mol}\right]$$

就认为存在一个残基 i 的 C=O 键和残基 j 的 N—H 键之间的氢键. DSSP 方法的定义仅适合定义蛋白质二级结构, 不会引起二级结构的错误识别 [87].

转弯模式就是一个 $(i, i+n)$ 类型的氢键. 从 C=O(i) 到 N—H$(i+n)$ 的氢键为残基的 n 转弯, 即

$$n\text{转弯}\left(i\right) =: \text{H键}\left(i, i+n\right), \quad n = 3, 4, 5$$

两个不重叠的三残基片段 $(i-1, i, i+1)$ 和 $(j-1, j, j+1)$ 可以根据两种基本的匹配模式形成平行的或反平行的桥. 如果两个氢键以

$$\text{平行桥}\left(i, j\right) =: \text{H键}\left(i-1, j\right)\text{和H键}\left(j, i+1\right), \text{或}$$

$$\text{H键}\left(j-1, i\right)\text{和H键}\left(i, j+1\right)$$

$$\text{反平行桥}\left(i, j\right) =: \text{H键}\left(i, j\right)\text{和H键}\left(j, i\right), \text{或}$$

$$\text{H键}\left(i-1, j+1\right)\text{和H键}\left(j-1, i+1\right)$$

为特征, 那么残基 i 和残基 j 之间就存在一个桥.

7.2.1.3　二级结构定义

两个连续的 n 转弯确定了最小的螺旋. 例如, 从残基 i 到残基 $i+3$ 的一个 4 螺旋最小长度为 4, 这个螺旋需要位于残基 $i-1$ 和残基 i 的两个 4 转弯

$$4\text{螺旋}\left(i, i+3\right) =: \left[4-\text{转弯}\left(i-1\right)\text{和}4-\text{转弯}\left(i\right)\right]$$

即氢键 $(i-1,i+3)$ 和氢键 $(i,i+4)$. 这里残基 $i+1$ 和残基 $i+2$ 不需要氢键. 同样, 一个从残基 i 到残基 $i+2$ 的 3 螺旋最小长度为 3, 需要两个连续的转弯. 一个从残基 i 到残基 $i+4$ 的 5 螺旋最小长度为 5

$$3\text{螺旋}\,(i,i+2) =: \left[3\text{转弯}\,(i-1)\,\text{和}3\text{转弯}\,(i)\right]$$

$$5\text{螺旋}\,(i,i+4) =: \left[5\text{转弯}\,(i-1)\,\text{和}5\text{转弯}\,(i)\right]$$

较长的螺旋为最小螺旋单位的重复. 传统上这些结构称作 α 螺旋、3_{10} 螺旋和 π 螺旋.

梯子和折叠定义为

梯子: 一个或多个连续相同类型的桥结构.

折叠: 一个或多个由共同残基联结的梯子构成的结构.

长的螺旋中可能存在氢键缺失现象. 例如两个重叠的最小螺旋被两个或三个残基链接在一起形成一个螺旋. 跟规则的 7 残基螺旋或 8 残基螺旋相比, 它失去了第三个或第四个氢键. 这种不完美的结构是螺旋中的绞结.

β 结构是由一个 β 折叠股上带有至多一个额外的残基, 另一股上带有至多 4 个残基链接的同一类型的梯子或桥组成的凸联结. 这个定义与拉式图 (Fichardson's)[122] 的观察结果一致, 除了通常存在的 β 桥中点格错误, 还有更多的突起. 在名义上的梯子中, 凸链接的梯子也当作梯子 (线性的桥) 来处理. 总之包括外来残基在内的所有的凸链接梯子上的残基都标记为 "E".

7.2.1.4 几何结构

(1) 弯曲. 弯曲指的是在蛋白质的二级结构中曲度高的区域 [120]. 曲度是指以 5 个残基中位于中心的残基 i 为基点, 前三个残基的主链方向和后三个残基的主链方向夹角的大小. 以 i 为基点的弯曲曲度大于 70°, 即

$$\text{弯曲}\,(i) =: \left[\text{角度}\left\{\left(\mathrm{C}^{\alpha}\,(i)-\mathrm{C}^{\alpha}\,(i-2)\right),\left(\mathrm{C}^{\alpha}\,(i+2)-\mathrm{C}^{\alpha}\,(i)\right)\right\} > 70^{\circ}\right]$$

那么这个二级结构为弯曲, 标记为 "S".

(2) 手性. 残基的手性 (除了链的两个末端) 为

$$\alpha\,(i) =: \text{两面角}\left(\left(\mathrm{C}^{\alpha}\,(i-1),\mathrm{C}^{\alpha}\,(i)\right),\left(\mathrm{C}^{\alpha}\,(i+1),\mathrm{C}^{\alpha}\,(i+2)\right)\right)$$

多数螺旋手性是正的, 多数扭曲的 β 梯子的手性是负的.

(3) SS 键. SS 键, 即两个半胱氨酸的硫原子之间的共价连接, 直接从 PDB 数据库中取得, 它们是氨基酸序列中的一部分. 根据坐标数据, 两个 S 之间的距离小于 3Å.

(4) 链的断裂. 如果肽键的长度超过 2.5Å, 就认为是键的断裂. 标记为 "! ", 作为一个断裂残基. 键的断裂反映了化学键的缺失、衍射图的密度丢失或坐标错误.

7.2.2　蛋白质二级结构鉴别方法

STRIDE 是根据氢键键能和主链扭曲角度进行蛋白质结构自动比对的方法 [90]. 由于二级结构模式的性质、α 螺旋的四残基转弯、β 折叠桥都由相关的氢键键能和氨基酸残基中的 Φ 和 φ 值的统计学性质共同确定, 所以两种因素在决定二级结构时的权重要预先确定. 每种氢键模式都由这些量的阈值精确协调来共同定义, 如果螺旋两端的残基有可以合适的 Φ-ψ 角度的话, STRIDE 就会把螺旋末端的一个或两个末端残基都包含在螺旋内 [91]. 同样, 如果 Φ-ψ 角度不合适, 在 DSSP 中定义的短的 α 螺旋可能被认为是别的二级结构. 也就是说如果 Φ-ψ 角度不合适, 由氢键模式定义的螺旋结构可能被否定 [90].

7.2.2.1　氢键能

STRIDE 中氢键的键能定义方法与 DSSP 中有所不同. 在 STRIDE 中, 氢键能 E_{hb} 由通过大量实验数据分析得到的经验能量函数计算, 这种函数是由分析大量多肽、肽键、氨基酸和小有机化合物的晶体几何结构的氢键经验数据得到的

$$E_{hb} = E_r \times E_t \times E_p$$

其中 E_r 是氢键的长度, E_t 和 E_p 是方向的参数. 距离项是函数

$$E_r = \frac{C}{r^8} + \frac{D}{r^6}$$

其中 $C = -3E_m r_m^8 \text{kcal}\text{Å}^8/\text{mol}$, $D = -4E_m r_m^6$ $\text{kcal}\text{Å}^8/\text{mol}$. r 是提供电子和接受电子的原子之间的距离. E_m 和 r_m 分别是优化氢键能和键长. 对于 “主链–主链” 氢键 N—H—O, $E_m = -2.8\text{kcal/mol}$, $r_m = 3.0\text{Å}$. 方向的参数项 E_t 和 E_p 分别为

$$E_p = \cos^2 p$$

和

$$E_t = \begin{cases} (0.9 + 0.1\sin 2t_i)\cos t_0, & 0 < t_i < 90° \\ K_1\left(K_2 - \cos^2 t_i\right)^3 \cos t_0, & 90° < t_i < 110° \\ 0, & t_i > 110° \end{cases}$$

其中 $K_1 = 0.9/\cos^6 110°$, $K_2 = \cos^2 110°$, t_0 分别是氢原子与氧原子的连线与这条连线到它们在肽平面上射影的角度, t_i 是这个射影与碳氧双键的夹角 (图 7-2). 为了修正数据的噪声, 另外一个能量函数的限制为

$$E_r = E_m, \quad r < r_m$$

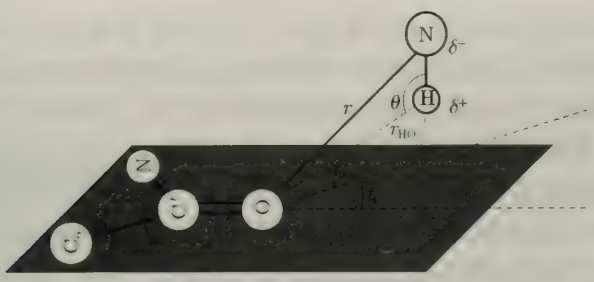


图 7-2 肽平面中键的夹角 (源自: Dmitrij Frishman 和 Patrick Argos, 1995)

7.2.2.2 α 螺旋和 β 折叠扭曲角度概率

以拉式图中的 $20° \times 20°$ 为单位考虑蛋白质结构主链两面角, 在第 i 个区域的 α 螺旋和 β 折叠残基的扭曲角度的概率为

$$P_i^{\alpha} = \begin{cases} \dfrac{N_i^{\alpha}}{N_t^{\text{total}}}, & \text{如果} -180° < \varphi < 10°, \quad -120° < \Phi < 45° \\ 0, & \text{其他} \end{cases}$$

以及

$$P_i^{\beta} = \begin{cases} \dfrac{N_i^{\beta}}{N_t^{\text{total}}}, & \text{如果} -180° < \varphi < 0°, \quad -180° < \Phi < -120°, \quad 45° < \Phi < 180° \\ 0, & \text{其他} \end{cases}$$

其中 N_i^{α} 和 N_i^{β} 分别是在给定的 φ 和 ψ 区域定义为 α 螺旋和 β 折叠的残基数量, N_i^{total} 是指区域 i 内发生了角度扭曲的残基总数量. 在广泛接受的 α 螺旋和 β 折叠区域, P_i^{α} 以及 P_i^{β} 为零.

7.2.2.3 二级结构的识别

STRIDE 算法定义最小的 α 螺旋应该在残基 k 和 $k+4$ 之间包括至少两个连续的氢键, 有

$$E_{hb}^{k,k+4}\left(1 + W_1^{\alpha} + W_2^{\alpha} \cdot \frac{P_k^{\alpha} + P_{k+4}^{\alpha}}{2}\right) < T_1^{\alpha}$$

如果在残基对 $(k, k+4)$ 和 $(k+1, k+5)$ 之间的两个连续氢键具备了这个条件, 那么中间的四个残基确定为 α 螺旋 H. 如果边缘的残基 k 和 $k+5$ 分别满足下面附属条件:

$$P_k^{\alpha} < T_2^{\alpha} \text{和} P_{k+5}^{\alpha} < T_3^{\alpha}$$

那么这两个也包括在 α 螺旋中. 上面的公式中, P_k^{α}、P_{k+1}^{α}、P_{k+2}^{α}、P_{k+3}^{α}、P_{k+4}^{α} 和 P_{k+5}^{α} 分别是残基 k、$k+1$、$k+2$、$k+3$、$k+4$ 和 $k+5$ 的扭曲角度概率. W_1^{α} 和 W_2^{α} 以及 T_1^{α}、T_2^{α} 和 T_3^{α} 分别是经验权重和优化阈值.

STRIDE 算法定义最小的 β 折叠为两个连续的氢键桥. 氢键桥的稳定性决定于 β 折叠中内部残基的内在平均统计学倾向以及由这种倾向决定的两个键之间氢键的强度. 内部残基指的是那些通过主链羰基和肽平面上的氢键参与形成两个氢键或侧面参与任一氢键的两个残基. 因为 β 折叠的边缘上主链方向的变化常常很大, 所以后面的构型就没有考虑进去. 这会使得至少在 N 端折叠边缘的角度 φ 以及在 C 端折叠边缘的角度 Φ 落在拉式图的 β 折叠区域之外. 相应地, β 桥涉及的两个氢键必须满足下面条件 [90]:

$$\begin{cases} E_{hb1}\left(1+W_1^{\beta}+W_2^{\beta}\cdot \mathrm{CONF}_{\mathrm{Antiparallel}}\right) < T^{\beta}_{\mathrm{Antiparpllel}} \\ E_{hb2}\left(1+W_1^{\beta}+W_2^{\beta}\cdot \mathrm{CONF}_{\mathrm{Antiparallel}}\right) < T^{\beta}_{\mathrm{Antiparpllel}} \end{cases}$$

对于平行的 β 桥有

$$\begin{cases} E_{hb1}\left(1+W_1^{\beta}+W_2^{\beta}\cdot \mathrm{CONF}_{\mathrm{parallel}}\right) < T^{\beta}_{\mathrm{parpllel}} \\ E_{hb2}\left(1+W_1^{\beta}+W_2^{\beta}\cdot \mathrm{CONF}_{\mathrm{parallel}}\right) < T^{\beta}_{\mathrm{parpllel}} \end{cases}$$

其中 E_{hb1} 和 E_{hb2} 分别是第一和第二氢键, 并且

$$\mathrm{CONF}=\frac{P^{\beta}_{\mathrm{Int1}}+P^{\beta}_{\mathrm{Int2}}}{2}$$

如果内部残基出现在 β 桥的两端或 $\mathrm{CONF}=P^{\beta}_{\mathrm{Int}}$, 如果只有一个残基在给定 β 桥的内部. W_1^{β} 和 W_2^{β} 是需要优化的经验权重.

如果符合上述标准的相邻桥结合形成反平行或平行的 β 折叠, 那么在一个折叠股的两个桥之间不超过 4 个插入的残基, 在另一个折叠股不超过一个插入残基. 如果在毗邻的桥之间所有的残基以及在它们之间可能带有的突起, 那么都可以认为是 β 折叠的扩展状态 “E”. 对于那些没有与其他桥结合在一起的孤独 β 桥称作 “B”.

Dmitrij Frishman 和 Patrick Argos 对于其他二级结构没有特别定义, 而是使用了其他已经定义了的二级结构 [90].

与 DSSP 数据库定义二级结构的方法一样, STRIDE 方法认为一个基本的 α 螺旋单位至少包含两个连续的残基 i 到残基 $i+4$ 的氢键. 与 DSSP 不同的是, 如果螺旋两端的残基有可以合适的 Φ-ψ 角度的话, STRIDE 就会把螺旋末端的一个或两个末端残基都包含在螺旋内. 同样, 如果 Φ-ψ 角度不合适, 在 DSSP 中定义的短的 α 螺旋可以被否定. 也就是说如果 Φ-ψ 角度不合适, 由氢键模式定义的螺旋结构可能被否定. 那么可以认为 STRIDE 的螺旋由氢键和 Φ-ψ 角度共同定义. 在 STRIDE 的定义中, β 折叠的类别不区分平行与反平行. 最小折叠可以由两个残基构成, 这两个残基都保持 5 个可能的氢键构型之一. 同样, β 折叠也是由氢键和 Φ-ψ

角度共同定义的. 突起的定义与 DSSP 的定义相同. 3_{10} 螺旋和 π 螺旋与 DSSP 的定义一样, 不同的是使用了经验的氢键标准. 转弯根据残基 $i+1$ 和 $i+2$ 的 Φ-ψ 角度定义 [92].

7.2.3 DEFINE 算法对于蛋白质二级结构的定义

算法 DEFINE 由 Richards 和 Kundrot 设计, 该算法依据与理想二级结构的线性距离矩阵中 C_α 原子之间距离的坐标匹配来确定待求蛋白质的二级结构 [93]. DEFINE 方法在确定一个氨基酸序列的二级结构过程中, 首先得到这个氨基酸序列主链分子之间的相对距离矩阵, 然后找到这个矩阵中与标准分子之间距离矩阵严格的匹配部分, 最后拓展这个匹配的部分, 把不太严格的匹配加入其中. 这种算法可以确定 α 螺旋、β 折叠、转弯和 Ω 回旋.

DEFINE 算法使用了由 α 碳原子坐标计算的原子间距离预测蛋白质二级结构, 得到的结果同视觉直观判断的结构非常吻合 [93]. 原子中心之间的距离矩阵提供了描述结构的笛卡儿坐标. 在这种公式中, 一个由 N 个原子组成的结构可以产生 $N \times N$ 矩阵, 这个矩阵的元素 (i, j), 指的是原子 i 和原子 j 之间的距离.

DEFINE 算法中使用的距离矩阵只涉及 C_α 原子之间的距离 (图 7-3), 在主链的 N 端的 C_α 原子在矩阵的左上角. 以这个 C_α 原子为原点, 计算其他 C_α 原子与这个 C_α 原子之间的距离形成了距离矩阵. 两个毗邻残基侧链之间的距离构成另外一个矩形子矩阵, 这些子矩阵称为 “盒子”. $C_{\alpha i}$ 和 $C_{\alpha j}$ 之间的距离简化为 (i, j). 理想结构中的两个原子之间的距离形成的子矩阵的元素称为 “面具”.

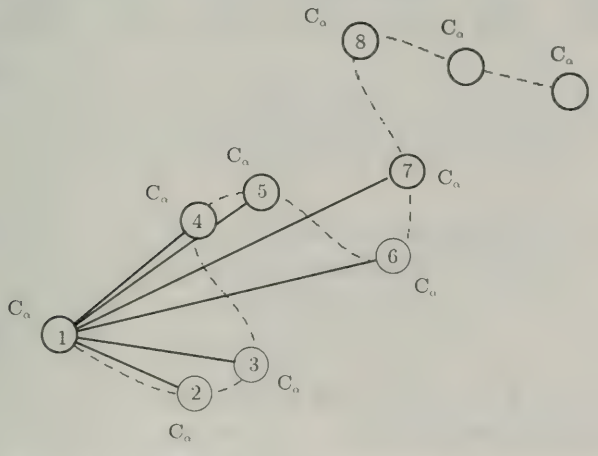


图 7-3 C_α 原子之间相对距离 (源自: Claus Andersen 和 Burkhard Rost, 2002)

DEFINE 算法使用蛋白质结构主链的 C_α 原子之间距离矩阵的方法定义二级

结构因子. 该算法认为两个连续的 C_α 原子 $(i, i+1)$ 之间的距离依赖于两个相互干涉的主链构象角度. 然而由于肽键平面构象的限制, 实际上 $(i, i+1)$ 之间的距离取值范围很窄. 因为 $(i-1, i, i+1)$ 的角度基本固定, $(i, i+2)$ 之间的距离的取值范围也很窄. 为了把握主要因素, 实际计算两个原子之间的距离时忽略一些实验中的错误. $(i, i+1)$ 之间的距离为 (3.75±0.02)Å, 对于转移肽键来说 $(i, i+2)$ 之间的距离为 (5.9±0.6)Å. 这样主链构象的二级结构特征出现在第三个位置 $(i, i+3)$ 上, 其中 α 螺旋的 $(i, i+3)$ 距离是 5.0 Å, β 折叠股的 $(i, i+3)$ 约为 9.9 Å.

蛋白质中的顺势肽键很少, 但是这种构象确实存在, 通常涉及脯氨酸氨基化合物团. 在顺势肽键中 $(i, i+1)$ 距离大约为 2.9 Å, 因此距离 (2.9±0.2)Å标记为顺势肽键.

在所有蛋白质二级结构中, 螺旋最具刚性、在空间上最容易定义. 表 7-1 给出了带有 L 个残基的理想距离子矩阵. 该矩阵由螺旋中的 C_α 原子之间的距离构成, 其中第一行表示的距离为从 (i, i) 到 $(i, i+L-1)$. 下一行的元素表示的距离为 $(i+1, i+1)$ 到 $(i+1, i+L-2)$. 原则上, 螺旋的长度没有限制. 但该算法限制螺旋的长度为 50.

表 7-1　二级结构距离参考值 (源自: Frederic M. Richards Craig E. Kundrot, 1988)

```
DATA ALPHA/
1    0.00,   3.75,   5.36,   5.02,   6.11,   8.53,   9.75,  10.43,  12.18,  14.09,
2   15.15,  16.32,  18.19,  19.77,  20.85,  22.33,  24.13,  25.49,  26.70,  28.36,
3   30.01,  31.27,  32.65,  34.35,  35.84,  37.11,  38.64,  40.30,  41.67,  43.06,
4   44.64,  46.20,  47.52,  48.97,  50.61,  52.07,  53.41,  54.95,  56.55,  57.93,
5   59.33,  60.93,  62.45,  63.81,  65.29,  66.89,  68.33,  69.72,  71.27,  72.82/

DATA BETA/    0.00,   3.75,   6.47,   9.89,  12.94,
1            16.28,  19.40,  22.72,  25.87,  29.17,
2            32.34,  35.62,  38.81,  42.09,  45.28,
3            48.55,  51.74,  55.01,  58.21,  61.48/

DATA TURN/    0.00,   0.00,   0.00,   3.70,   0.00,
1             0.00,   5.60,   3.70,   0.00,   5.00,
2             5.10,   3.70,   5.40,   6.90,   6.10/

DATA BB/      4.95,   6.20,   8.45,  11.45,  14.55,
1            17.65,  20.90,  24.15,  27.45,  30.65,
2            34.00,  37.25,  40.55/
DATA A310/
1  −0.042,  −1.388,  1.834,   1.472,  −1.616,  −1.673,   3.013,   1.872,  −1.334,
2   4.464,   1.004,  2.097,   5.939,  −2.228,   0.679,   7.427,  −0.325,  −2.287,
3   8.917,   2.295,  0.069,  10.402,  −0.459,   2.268,  11.884,  −2.174,  −0.812,
```

续表

```
DATA A310/
4 13.360,   1.146, −2.011, 14.844,   1.792,   1.456, 16.334, −1.730,   1.525,
5 17.815, −1.221, −1.965, 19.289,   2.143, −0.899, 20.777,   0.548,   2.244,
6 22.266, −2.289,   0.155, 23.743,   0.238, −2.309, 25.222,   2.260,   0.581,
7 26.712, −0.916,   2.104, 28.197, −1.898, −1.281, 29.673,   1.605, −1.704,
8 31.158,   1.456,   1.818, 32.647, −1.992,   1.096, 34.126, −0.717, −2.193,
9 35.607,   2.325, −0.409, 37.096,   0.079,   2.304, 38.572, −2.247, −0.349,
1 40.053,   0.737, −2.226, 41.558,   2.124,   1.009, 43.051, −1.299,   1.839,
2 44.986, −0.755, −1.696, 46.921,   1.843,   0.220, 48.857, −1.132,   1.471,
3 50.792, −0.684, −1.726, 52.728,   1.833,   0.295, 54.663, −1.192,   1.424,
4 56.598, −0.613, −1.752, 58.534,   1.819,   0.370, 60.469, −1.249,   1.373,
5 62.404, −0.541, −1.776, 64.340,   1.803,   0.444, 66.275, −1.304,   1.321,
6 68.210, −0.467, −1.797, 70.146,   1.783,   0.518, 72.081, −1.357,   1.266,
7 74.017, −0.393, −1.814, 75.952,   1.760,   0.591, 77.887, −1.408,   1.210,
8 79.823, −0.318, −1.829, 81.758,   1.734,   0.662,/
```

在形成矩阵的过程中, 首先检测 α 螺旋的位置. 每个转弯处的 C_α 都认为是一个可能的 N 端原子. 一个螺旋从一个给定的原子 $C_{\alpha i}$ 增长. 沿着垂直线 i 的方向在距离矩阵中每次移动一步. $C_{\alpha j}$ 原子是 C 端的原子. 潜在螺旋中所有的距离都在三角 (i,i)、(i,j) 和 (j,j) 中给出. 沿着右手垂直线方向 (i,j) 和 (j,j) 描述 $C_{\alpha j}$ 到前面的所有 α 螺旋中原子之间的距离. 这些距离与理想 α 螺旋面具中相应位置的距离比较, 如果阈值发生超越, 螺旋中止.

长于 4 个残基的片段在允许的积累误差限制范围内 (e = 1 Å), 那么它就是 α 螺旋. 然后需要检测 α 螺旋的开始与结尾处是否是 3_{10} 螺旋, 3_{10} 螺旋和 π 螺旋不用进行这种检测.

转弯的 C_α 距离矩阵中的元素比螺旋和折叠的 C_α 距离矩阵中的元素的阈值分布范围小得多. 表中列出了组成距离矩阵的 5 个 C_α 原子 $(i,i+1,i+2,i+3,i+4)$ 之间的距离的 "面具"[93].

为了把 β 折叠设计为单独的一类, 作者使用了理想的折叠线性距离矩阵. 由于从定义中删除了非刚性折叠, 折叠主链的柔性和较大折叠的曲率问题得到了解决. 最小长度的折叠至少包括 4 个残基. 根据 Pauling 的定义, 在 β 折叠中每个股一定要和另一个股配对来形成一个折叠 [88].

β 折叠构象的变化比 α 螺旋要大, 即便在一个折叠股中 Φ-ψ 的角度分布也很宽. 反平行的 β 折叠中的片段比平行的 β 折叠中氨基酸残基更加舒展一些. 平行的 β 折叠 C_α 原子距离矩阵显示在表中.

Ω 回旋比紧密环长度短些, 这种二级结构比其他二级结构更加不规则、更加富于变化. Ω 回旋的长度一般为 6~16 个残基. 任何长度的 Ω 回旋两段的距离小于

10Å, 这个距离小于转弯两段距离的 2/3.

7.2.4 P-Cruve 方法

P-Curve 运用微分几何学对蛋白质主链曲率进行了数学结构分析 [94]. 他们在固定了一系列肽平面的固定轴系统的基础上计算了螺旋轴. 通过 Motif 匹配可以确认二级结构, 其中 Motif 中的参数是使用微分几何方法得到的两个肽平面之间的倾斜、转动和扭曲的螺旋系统的半径等一系列参数. 这个参数分析主要使用了 C_α 坐标来完成的. 因为使用了不同的参数, P-Curve 确认二级结构的方法与 Φ-ψ 角度或氢键的分析方法很不一样, 匹配 P-Curve 的 Motif 自由度与匹配 DEEINE 的线性距离掩码的自由度也大不一样. 例如, 当 DEFINE 的线性距离 "面具" 无法确认孤立的 β 股, 而使用局部 P-Curve 参数就能很好地确认 [94].

定义螺旋结构的第一步是选择结构重复结构. 然后确定每一个重复结构用于定义该结构空间位置相对固定的坐标系统. 这个坐标系统的中心就是肽键平面的中点, 坐标轴是人为指定的垂直向量单位 $(\boldsymbol{J}, \boldsymbol{K}, \boldsymbol{L})$(图 7-4). 第一个向量是在方向 N—C 上的肽键向量 $\boldsymbol{J} \cdot \boldsymbol{L}$ 位于肽键平面上, 方向指向羰基. $\boldsymbol{K}$ 是垂直于肽平面的向量, 它由向量的乘积 $\boldsymbol{J} \times \boldsymbol{L}$ 定义.

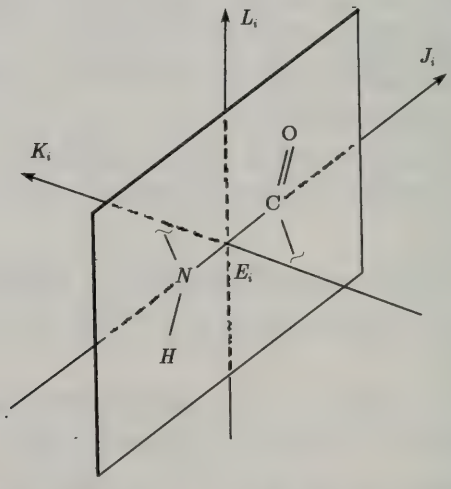


图 7-4　肽平面的坐标系统 $(\boldsymbol{J}, \boldsymbol{K}, \boldsymbol{L})$(源自: Heinz Sklenar et al., 1989)

每一个重复因子的位置可以由这个局部坐标系统确定. 确定坐标系统需要四个变量, 其中包括两个平移变量, 两个旋转变量 (图 7-5). 定义坐标系统的中心为点 P, 三个螺旋轴为 $\boldsymbol{U}$、$\boldsymbol{V}$ 和 $\boldsymbol{W}$. 把坐标系统和肽平面的固定轴联系起来的两个向量 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 称作 X 变换和 Y 变换. 肽系统的转动位置由一个右手转动项通过 E 点、平行于 $\boldsymbol{V}$ 的向量 "倾度" 和另一个右手转动项围绕肽系统的 "倾斜" 共同取得.

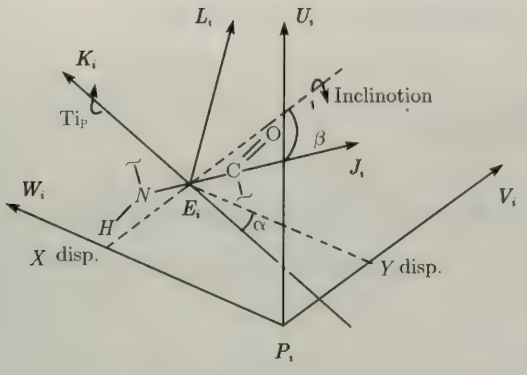


图 7-5 螺旋参数的定义 (源自: Heinz Sklenar et al., 1989)

为了描述一个单独的螺旋构型 (α 螺旋或 β 折叠) 需要增加一个平移变量和一个旋转变量. 这两个额外增加的变量对应于沿着螺旋轴 $\boldsymbol{U}$ 的连续重复结构 (称作 "增加") 以及它们绕着这个轴的相对的右手旋转 (称作 "扭曲").

上面定义的系统由坐标轴系统 $\boldsymbol{JKL}$ 和一个参照点 E 以及具备螺旋坐标轴系统 $\boldsymbol{UVW}$ 和一个参照点 P 构成. 这两个系统由螺旋变量 X 变换和 Y 变换以及倾度和倾斜联系在一起. 实际上, 蛋白质的原子坐标可以知道, 所以 $\boldsymbol{JKL}$ 轴系统是在空间中是固定的, 那么找到局部螺旋轴系统 $\boldsymbol{UVW}$ 的优化位置和方向是关键问题.

这个目标可以由一个公式化函数完成, 首先定量化两个连续肽平面螺旋参数的不规则性, 其次定量化两个连续局部螺旋轴的中断位置. 第一个目标可以由描述相对于具备螺旋坐标系统的肽平面连续位置变化的项的和表示. 这些项涉及计算具备坐标轴系统 $\boldsymbol{U}$ 和向量 $\overrightarrow{P-E}$ 的投影的差异. 它们的射影分别定义如下:

$$D_i = \sum_{X \epsilon J,K,L} \left(\boldsymbol{U}_i^{\mathrm{T}} X_i - \boldsymbol{U}_{i-1}^{\mathrm{T}} X_{-1i} \right)^2$$

以及

$$C_i = \sum_{X \epsilon J,K,L} \left[(P_i - E_i)^{\mathrm{T}} X_i - (P_{i-1} - E_{i-1})^{\mathrm{T}} X_{i-1} \right]^2$$

令

$$A_1 = \sum_{i=2N} D_i$$

$$A_2 = \sum_{i=2N} C_i$$

为了处理两个连续局部螺旋坐标的变型, 需要用来比较它们向量方向的项

$$B_1 = \sum_{i=2N} (\boldsymbol{U}_i - \boldsymbol{U}_{i-1})^2$$

如果定义两个连续螺旋坐标的平均单位向量为

$$\langle \boldsymbol{U}_i \rangle = (\boldsymbol{U}_i + \boldsymbol{U}_{i-1}) / |\boldsymbol{U}_i + \boldsymbol{U}_{i-1}|$$

两个连续 P 点之间的向量为

$$\boldsymbol{S} = \boldsymbol{P}_i - \boldsymbol{P}_{i-1}$$

那么可以计算侧面的错位点、并且垂直于中间平面轴的向量

$$\boldsymbol{Q}_i = \boldsymbol{S}_i - \langle \boldsymbol{U}_i \rangle \left(\langle \boldsymbol{U}_i \rangle^{\mathrm{T}} \boldsymbol{S}_i \right)$$

上面函数的最后一项为

$$B_2 = \sum_{i=2N} \boldsymbol{Q}_i^2$$

为了取得函数转动项 (A_1, B_1) 和平动项 (A_2, B_2) 的权重平衡, 必须把包含在前面项中的转动角度乘以聚合体连续单位平均分离距离. 也就是说 A_1 和 B_1 应该乘以这个距离的平方. 权重取 6, 相应的平均分离距离大体上为 2.5Å.

最小化的函数完整的表达式为

$$F(h) = 6(A_1 + B_1) + A_2 + B_2$$

由字母 h 表示函数的变量在每一个股中的肽键中仅仅包含 4 个螺旋变量 (X 置换、Y 置换、倾度和倾斜). $F(h)$ 的每一项应该选择以便能使沿着 N 端到 C 端或者相反的方向得到恒等的和. 为了使函数 $F(h)$ 快速的收敛, 首先计算每个肽键螺旋变量的微分.

最后, 要考虑在一般情况下的每个肽内部的函数的定义. 在这种情况下, 前面给出的上升和扭曲的简单定义不再使用, 必须考虑的是两个螺旋轴的相对空间位置. 定义两个螺旋轴的相对空间位置使用平均的轴系统 (以点 q 为中心的 $\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{f}$). 这个系统由下面的等式定义:

$$q = \frac{(P_{i-1} + P_i)}{2}$$

$$\boldsymbol{n} = \frac{(\boldsymbol{U}_i + \boldsymbol{U}_{i-1})}{|\boldsymbol{U}_i + \boldsymbol{U}_{i-1}|}$$

$$\boldsymbol{g} = \frac{(\boldsymbol{V}_{i-1} + \boldsymbol{V}_i)}{|\boldsymbol{V}_{i-1} + \boldsymbol{V}_i|}$$

$$\boldsymbol{d} = \frac{[\boldsymbol{g} - \boldsymbol{n}(\boldsymbol{n}^{\mathrm{T}}\boldsymbol{g})]}{|\boldsymbol{g} - \boldsymbol{n}(\boldsymbol{n}^{\mathrm{T}}\boldsymbol{g})|}$$

$$\boldsymbol{f} = \boldsymbol{n} \times \boldsymbol{d}$$

向量 $\boldsymbol{U}$ 与平均平面的交集 (垂直于 $\boldsymbol{n}$) 为

$$\boldsymbol{P}_{i-1} = \boldsymbol{P}_{i-1} + \frac{\boldsymbol{U}_{i-1}[\boldsymbol{n}^{\mathrm{T}}(\boldsymbol{q} - \boldsymbol{P}_{i-1})]}{\boldsymbol{n}^{\mathrm{T}}\boldsymbol{U}_{i-1}}$$

以及

$$\boldsymbol{P}_i = \boldsymbol{P}_i - \frac{\boldsymbol{U}_i\left[\boldsymbol{n}^{\mathrm{T}}(\boldsymbol{P}-\boldsymbol{q})\right]}{\boldsymbol{n}^{\mathrm{T}}\boldsymbol{U}_i}$$

现在可以得到描述交叉点螺旋轴系统相对位置的参数表达式. 两个沿着 $\boldsymbol{d}$ 和 $\boldsymbol{f}$ 轴的平移变量为

$$\text{轴 } X \text{ 置换} = \boldsymbol{d}^{\mathrm{T}}\left(\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\right)$$
$$\text{轴 } Y \text{ 置换} = \boldsymbol{f}^{\mathrm{T}}\left(\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\right)$$

简单来说, 两个类似于倾度和倾斜的转动变量定义为

$$\text{轴倾度} = 2\arccos\left(\boldsymbol{f}^{\mathrm{T}}\boldsymbol{t}\right), \quad \text{如果}\boldsymbol{d}^{\mathrm{T}}(\boldsymbol{f}\times\boldsymbol{t})>0, \quad \text{那么}\Phi>0$$

$$\text{轴倾斜} = 2\arccos\left(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{U}_i\right), \quad \text{如果}\boldsymbol{t}^{\mathrm{T}}(\boldsymbol{r}\times\boldsymbol{U}_i)>0, \quad \text{那么}\varphi>0$$

其中 $\boldsymbol{t}=\dfrac{(\boldsymbol{U}_i\times\boldsymbol{d})}{|\boldsymbol{U}_i\times d_i|}$, $\boldsymbol{r}=\dfrac{(\boldsymbol{d}\times\boldsymbol{t})}{|\boldsymbol{d}\times\boldsymbol{t}|}$.

下面介绍三个辅助参数. 这些参数衡量形成于两个连续螺旋轴向量间的网格角度 (轴的弯曲, $Ad=\arccos(\boldsymbol{U}_{i-1}^{\mathrm{T}}\boldsymbol{U}_i)$), 网格的两个连续 $\boldsymbol{P}$ 点间的侧面断层 (轴断层, $Ad=\sqrt{(A\alpha^2+Ay^2)}$), 两个连续 $\boldsymbol{P}$ 点间的距离 (路径长度, 路径 $=|\boldsymbol{P}_i-\boldsymbol{P}_{i-1}|$).

下面定义一些解释肽键交会点的通用参数, 三个平移参数

$$\text{平移} = dx(i) + Ax - dx(i-1)$$

$$\text{滑动} = dy(i) + Ay - dy(i-1)$$

$$\text{上升} = \left|\boldsymbol{p}_{i-1} - \boldsymbol{P}_{i-1} + |\boldsymbol{P}_i - \boldsymbol{p}_i|\right|$$

和三个转动参数

$$\text{倾斜} = \eta(i) + \eta_A - \eta(i-1)$$

$$\text{滚动} = \theta(i) + \theta_A - \theta(i-1)$$

$$\text{扭曲} = \pm\arccos\left(\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{f}^{+}\right) \pm \arccos\left(\boldsymbol{W}_{i-1}^{\mathrm{T}}\boldsymbol{f}^{-}\right)$$

其中 $\boldsymbol{f}^{+}$ 和 $\boldsymbol{f}^{-}$ 向量分别由向量 $\boldsymbol{f}$ 绕 $\boldsymbol{d}$ 点转动 $\dfrac{\text{轴倾斜}}{2}$ 和 $-\dfrac{\text{轴倾斜}}{2}$ 得到的. 如果 $\boldsymbol{U}_i^{\mathrm{T}}\left(\boldsymbol{f}^{+}\times\boldsymbol{W}_i\right)>0$, 那么第一项的扭曲为正. 如果 $\boldsymbol{U}_{i-1}^{\mathrm{T}}\left(\boldsymbol{f}^{-}\times\boldsymbol{W}_{i-1}\right)<0$, 那么第二项为正.

这些参数可以分成三类: ① 肽轴参数; ② 肽键内部参数; ③ 轴倾斜参数.

7.3 蛋白质二级结构预测

7.3.1 概述

蛋白质二级结构预测是世界性难题, 人们已经尝试使用了很多方法进行蛋白质

二级结构的预测研究. 然而这些预测方法所达到的预测准确率一般不超过 77%. 所有预测方法可以分成理论分析方法和统计知识方法两类, 从时间上来说大体可以分为三代. 理论分析方法其实就是从头预测方法, 主要是从假设或理论计算 (如分子力学、分子动力学计算) 出发来预测蛋白质的结构. 该类方法假设折叠后的蛋白质取能量最低的构象 [95]. 统计知识方法主要是从观察和总结已知蛋白质结构的统计规律出发来预测未知蛋白质的结构. 该类方法对已知结构的蛋白质进行统计分析, 建立序列到结构的、定性的或定量的映射模型, 进而对未知结构的蛋白质根据映射模型直接从氨基酸序列预测结构. 这一类方法包括经验性方法、结构规律提取方法、同源模型方法等 [96]. 经验性方法对已知结构的蛋白质 (如蛋白质结构数据库 PDB、蛋白质二级结构数据库 DSSP 中的蛋白质) 进行统计分析, 发现各种氨基酸形成不同二级结构的倾向, 形成一系列关于二级结构预测的规则. 结构规律提取方法从蛋白质结构数据库中提取关于蛋白质结构形成的一般性规则, 指导建立未知结构的蛋白质的模型. 同源建模方法通过同源序列分析或模式匹配预测蛋白质的空间结构. 其原理基于下述假设: 每一个自然蛋白质具有一个特定的结构, 但许多不同的序列会采用同一个基本的折叠, 即具有相似序列的蛋白质倾向于折叠成相似的空间结构. 如果未知结构的蛋白质与已知结构的蛋白质具有足够的序列相似性, 那么根据相似性原理可以给未知结构的蛋白质构造近似的三维模型. 它是现在蛋白质结构预测中最可靠的方法 [35].

蛋白质二级结构预测大体可以分为三代. 第一代预测方法是基于单个氨基酸残基统计分析, 该法以有限数据集中各种残基形成特定二级结构的倾向作为预测的依据. 第二代方法是基于氨基酸片段的统计分析 (片段长度通常为 11 ～ 21), 片段体现了中心残基所处的环境. 在预测中心残基的二级结构时, 以残基在特定环境形成特定二级结构的倾向作为预测依据.

第一代方法的代表是 Chou-Fasman 方法与 GOR 方法. Chou-Fasman 方法是单序列预测方法中的一种, 它是使用氨基酸物理化学数据中派生出来的规律来预测二级结构. 首先统计出 20 种氨基酸出现在 α 螺旋、β 折叠和无规则卷曲中出现频率的大小, 然后计算出每一种氨基酸在这几种构象中的构象参数 Px. 构象参数值的大小反映了该种残基出现在某种构象中的倾向性的大小. 按照构象参数值的大小可以把氨基酸分为六个组: Ha(强螺旋形成者)、ha(螺旋形成者)、Ia(弱螺旋形成者)、ia(螺旋形成不敏感者)、ba(螺旋中断者)、Ba(强螺旋中断者). Chou 和 Fasman 根据残基的倾向性因子提出二级结构预测的经验规则, 要点是沿蛋白质序列寻找二级结构的成核位点和终止位点. 这种方法可能能够正确反映蛋白质二级结构的形成过程, 但预测成功率并不高, 仅有 50%左右.

GOR 算法也是单序列预测方法中的一种, 因其作者 Garnier, Osguthorpe 和 Robson 而得名. 这种方法是以信息论为基础的, 也属于统计学方法的一种. GOR

方法不仅考虑被预测位置本身氨基酸残基种类对该位置构象的影响, 也考虑到相邻残基种类对该位置构象的影响. 这样使预测的成功率提高到 65% 左右. GOR 方法的优点是物理意义清楚明确, 数学表达严格, 而且很容易写出相应的计算机程序, 但缺点是表达式复杂.

第二代预测方法可以归为如下几类:

(1) 基于统计信息的方法;

(2) 基于物理化学性质的方法;

(3) 基于序列模式的方法;

(4) 基于多层神经网络的方法;

(5) 基于图论的方法;

(6) 基于多元统计的方法;

(7) 基于机器学习的专家规则的方法.

最邻近算法.

第一代和第二代方法都以蛋白质序列的局部信息作为预测的基础, 预测的准确率不高. 第二代方法的代表主要包括多序列列线预测方法和神经网络方法.

多序列列线预测首先对序列进行多序列比对, 并利用多序列比对的信息进行结构的预测. 调查者可找到和未知序列相似的序列家族, 然后假设序列家族中的同源区有同样的二级结构, 预测不是基于一个序列而是一组序列中的所有序列的一致序列.

反馈式神经网络算法是目前二级结构预测应用最广的神经网络算法, 它通常是由三层相同的神经元构成的层状网络, 使用反馈式学习规则, 底层为输入层, 中间为隐含层, 顶层是输出层. 信号在相邻各层间逐层传递, 不相邻的各层间无联系, 在学习过程中根据输入的一级结构和二级结构的关系的信息不断调整各单元之间的权重, 最终目标是找到一种好的输入与输出的映像, 并对未知二级结构的蛋白质进行预测. 神经网络方法的优点是应用方便, 获得结果较快较好, 主要缺点是没有反映蛋白质的物理和化学特性, 而且利用大量的可调参数, 使结果不易理解. 许多预测程序如 PHD、PSIPRED 等均结合利用了神经网络的计算方法.

第三代预测方法运用蛋白质序列的长程信息和蛋白质序列的进化信息, 使二级结构预测的准确程度有了比较大的提高, 预测结果与实验观察趋于一致. 这些方法的代表方法有基于知识的方法和混合方法.

基于知识的预测方法根据氨基酸残基的物理化学性质, 包括疏水性、亲水性、带电性以及体积大小等, 并考虑残基之间的相互作用而制订出一套预测规则. 它们的基本原理大体为: 疏水性残基决定了二级结构的相对位置, 螺旋亚单元或扩展单元是结构域的核心, α 螺旋和 β 折叠组成了结构域. 混合方法将以上几种方法选择性的混合使用, 并调整他们之间使用的权重可以提高预测的准确率.

蛋白质二级结构预测是三维结构预测的重要组成部分. 二级结构预测的结果既可以应用于进一步预测蛋白质的高级结构, 又能应用于推测蛋白质的功能. 科学家已经尝试了很多方法来预测蛋白质二级结构, 这些方法主要包括APSSP2、JPred、JUFO、PHD、PHDpsi、PROF_king、Prospect、PSIpred、SAM-T99sec、SCRATCH (SSpro3)、SSpro1、SSpro2以及SSpro4等方法. 关于这些方法的详细信息可以在服务器http://cubic.bioc.columbia.edu/eva/sec/res_sec.html上找到.

随着计算机技术的快速发展和具有很强计算能力的计算机的开发, 使得强烈依赖计算机计算能力的机器学习技术在蛋白质结构预测领域得到应用. 2001 年, 支持向量机首次被应用于蛋白质二级结构预测. 通过对非同源蛋白质数据集 CB513[97] 进行训练和检测, 使得总的以残基为单位三态预测准确率 (Q_3) 达到 73.5%[98]. 在 2003 年, YANG 利用 SVM 方法得到蛋白质二级结构预测准确率 Q_3=75.2%[99]. KIM 利用 SVMpsi 方法使预测准确率 Q_3=76.6%[100]. GUO 在 2004 年把 PSI-BLAST Profiles 引进到 SVM 中, 并且发展了一个新的方法称为双层 SVM[134], 利用这种方法也取得了 Q_3=75.2%的成绩.

7.3.2　样本集的选择

随着 PDB 等主要蛋白质结构数据库中的蛋白质结构资源越来越多, 人们可以选择的蛋白质二级结构预测的样本的回旋余地也越来越大. 虽然折叠过程可能需要辅助分子的参与, 但是蛋白质三维结构仅仅由它们的氨基酸序列决定的假说仍然成立 [102]. 利用同源蛋白质来预测蛋白质二级结构的准确率要比利用非同源蛋白质的准确率要高. 因此, 在蛋白质结构预测中要使用非同源蛋白质. 对于蛋白质二级结构的预测来说, 选择合适的训练数据集至关重要. 目前人们使用的训练数据集包括很多个, 我们只使用了 RS126 和 CB513 两个训练和检测样本集. 附表 1 收集了 RS126 数据集的所有蛋白质序列代码, 附表 2 收集了 CB513 数据集的所有蛋白质序列代码.

7.3.2.1　RS126 数据集

RS126 非同源蛋白质数据集是 Burkhare Rost 和 Chris Sander 在 1993 年给出的 [102]. 他们根据 HSSP[103] 定义的非同源蛋白质的条件, 在当时已有的数据库中选择了 126 条非同源蛋白质序列. HSSP 给出了一个依赖长度的非同源相似性规则, 即对于长度超过 80 的氨基酸序列两两相对同源性要小于 25%. 这种方法同时依赖比对的长度和序列的组分. 通过对 1992 年 PDB 数据库中的 700 个蛋白质的相互比对, 从中选择了 150 个非同源序列. 其中分辨率小于 2.5Å的一共有 130 个, 其中包含了 126 个球状蛋白、4 个膜蛋白. 也就是说这 126 个非冗余蛋白质中长度超过 80 个残基的任何两个都不包含超过 25%的相同残基. 这 130 个蛋白中一共包

含了 24 395 个氨基酸残基, α 螺旋占 32%、β 折叠占 21%, C 卷曲占 47%.

7.3.2.2 CB513 数据集

然而, 经过几年的时间, 人们发现 Burkhare Rost 和 Chris Sander 在 1993 年给出的非同源蛋白质序列数据集需要更新. 因为首先非同源蛋白质的定义有缺陷, 另外随着时间的推移人们已经解析了更多的蛋白质的空间结构. 因此 James A. Cuff 和 Geoffrey J. Barton 在 1999 年构造了 CB513 非同源蛋白质数据集. 到目前为止的几年时间里, 人们还没有发现更好的定义非同源蛋白质的新准则, 因此 CB513 数据集还应用于蛋白质二级结构预测的实践中.

对于 RS126 数据集的非同源蛋白质的定义规则中, 通过百分比的方法不能很好地确定序列相似性, 特别是相似性低于 30%的序列 [130]. 这个非冗余数据集中的两两样本相似性通过以下方法得到: 首先通过标准动态规划算法比对序列 A 和 B 之间的相似性, 得到一个 V 的分值. 其次随机改变每一条蛋白质序列中的氨基酸的顺序, 通过标准动态规划算法进行再次比对. 这个过程重复至少 100 次, 计算每次得分的平均值 $\bar{x}$ 和标准差 σ. 再次计算 SD 分值: $(V-\bar{x})/\sigma$. CB513 样本集通过 SD 得分确定样本. 所有样本来自 3Dee 结构域数据库. 通过 SD 的分值首先取得了 1233 个结构域. 然后剔除多序列结构域, 使样本集的容量从 1233 减少到 988. 最后选取通过 X 射线衍射实验得到的解析度大于 2.5Å的结构域 554 个, 称作 CB554. 为了证明 CB554 结构域跟 RS126 数据集没有序列同源性, 合并两个数据集, 并通过 AMPS 的 blosum62 矩阵的 AMPS 两两比较数据集中的序列, 缺口罚分为 10. 通过比对, SD 分值大于等于 5 认为是序列相似的. 通过这种比对方法, RS126 中 11 条序列、CB554 和 RS126 之间的 119 条序列以及 CB554 中的 21 条序列是相似的. 这样, CB554 中有 140 条序列要么与 CB554 中的序列相似, 要么与 RS126 中的序列相似. 由于其中三条序列匹配多条序列, 那么其中 137 条序列与其他的序列是相互匹配的. 剔除了这 137 条序列以后 CB554 中剩下了 417 条序列, 这 417 条序列之间不相互匹配, 也不与 RS126 中的序列匹配. 这 417 条序列中, 21 条没有完整的 DSSP 定义, 被剔除, 最后剩下了 396 个蛋白质 (CB396). RS126 中 11 条序列与其他的序列之间的 SD 分值大于 5, 其中两条匹配多条序列, 所以提出 9 个. 剩下的与 CB513 合并得到 CB513 非冗余数据集.

7.3.3 二级结构规类方法

由于 DSSP 数据库是 Wolfgang Kabsch 与 Christian Sander 根据区别蛋白质二级结构的需要而构造的, 所以我们使用了该数据库的定义方法来定义我们实验中的二级结构. 根据 DSSP 的定义, 实验中依据 DSSP 定义的方法对蛋白质二级结构规类方法进行规类. 根据 DSSP 的定义, 所有蛋白质二级结构包括 8 种: H(α

螺旋)、G(3_{10} 螺旋)、I(π 螺旋)、E(β 折叠)、B(β 桥)、T(转弯)、S(弯曲) 以及 -(其他结构). 这 8 种二级结构依据以下规则合并为 3 类二级结构：H、G 和 I→H(螺旋), E→E(折叠) 和 B、T、S 和 − →C(卷曲), 并且进行以下调整：EC→EE, 而 ECE→CCC.

实验中提取了具有相同二级结构属性的连续残基序列作为样本集的元素, 并且按如下方法确定每个元素的二级结构类型：连续的螺旋序列作为螺旋 (H); 连续的折叠序列作为折叠 (E); 连续的无规则卷曲序列作为卷曲 (C).

例如, CB513 数据集中的1htrp-1-AS的一级结构和二级结构分别为

一级结构：AVVKVPLKKFKSIRETMKEKGLLGEFLRTHKYDPAWKYRFGDL

二级结构: CCEEEEEEECCCHHHHHHHCCCHHHHHCCCCCCHHHHHCCCCC

其中 IRETMKE、LGEFL 和 PAWKY 为螺旋, VKVPLKK 为折叠, AV、FKS、KGL、RTHKYD 和 RFGDL 为卷曲. 依据上述规则在 CB513 和 RS126 中提取的每种二级结构类型元素的数量和平均长度显示在表 7-2 中.

表 7-2　三种二级结构的样本数量和平均长度

二级结构类别	CB513		RS126	
	样本数量	平均长度	样本数量	平均长度
H	3083	9.4	813	9.1
E	3326	5.1	959	5.0
C	5111	6.2	1418	6.4

7.3.4　运用支持向量机进行蛋白质结构预测的样本提取方法与编码规则

客观地比较不同方法的结果需要使用相同的数据集 (包括相同的比对 profiles)、相同的二级结构定义 (包括相同的压缩方法) 和相同的准确率评估方法, 否则比较就是不客观的 [98]. 蛋白质结构预测的样本提取方法与编码规则是相互紧密相连的, 因为 SVM 要求所有的输入向量必须在一个输入空间内, 而向量在一个输入空间内的向量必须维数相同. 以往蛋白质结构预测中使用的蛋白质结构预测的样本提取方法与编码规则大体分为两类：①氨基酸序列组分方法; 氨基酸组分方法以滑窗方法为基础衍生出来. ②n 肽频数方法. 氨基酸组分方法多用于蛋白质二级结构预测样本长度较短的预测中. 这类方法的特点是一定要求氨基酸片段的长度相同. 而 n 肽频数方法则没有这个限制, n 肽频数方法可以对任意长度的氨基酸序列进行编码得到等长的输入向量.

7.3.4.1　滑窗方法

滑窗方法给出了一个 n 个相邻氨基酸的序列. 该方法的目的是正确预测中间氨基酸的正确二级结构 [54,98]. 输入向量由一个氨基酸残基的输入窗口决定, 这个

窗口一次向前滑动一个残基. 相对应的编码方法为: 对于一个单独的序列每一个残基被一个正交向量编码, 由于自然状态下可以形成蛋白质的氨基酸共有 20 个, 那么这种正交向量共有 20 个. 例如, (1, 0, ···, 0), 这个向量是 21 维的, 向量的前 20 维中, 每一维表示一种氨基酸残基. 为了能使滑窗从 N 端滑到 C 端, 也能从 C 端滑到 N 端, 第 21 维被加到每个残基中. 如果窗口的长度是 l, 那么特征向量的维数是 $21\times l$ 维. 在编码的时候, 首先取序列的第一个残基, 确定了相应的向量. 再取第二个残基, 把取得的向量放在第一个向量的后面. 这样当一个长度为 l 的氨基酸序列编码以后就可以形成一个 $21\times l$ 维的向量. 当加入了进化信息以后, 对单独序列进行多重序列比对. 通过比对计算出在某个位置的氨基酸残基发生频率.

7.3.4.2 N 格片段提取方法

根据 N 格模型把从训练集的每一个蛋白质序列中取出一系列长度为 N 的蛋白质片段 [99]. N 是片段的长度, N 由不同结构片段的平均长度决定.

首先进行向量化. 根据氨基酸的化学特征把所有氨基酸归为 5 类, 每种类别以一个向量为代表 (表 7-3).

表 7-3 根据氨基酸的化学特性划分的 5 种类别

氨基酸类别	5 维坐标向量	氨基酸残基
非极性, 脂肪族	(1, 0, 0, 0, 0)	A, V, L, I, M
芳香族	(0, 1, 0, 0, 0)	F, Y, W
极性, 不带电荷	(0, 0, 1, 0, 0)	G, S, P, T, C, N, Q
带正电	(0, 0, 0, 1, 0)	K, H, R
带负电	(0, 0, 0, 0, 1)	D, E

其次, 把每一种氨基酸的化学特性向量插入到 20 种氨基酸形成的向量中, 得到了一个 100 维的氨基酸化学特性向量. 例如氨基酸残基序列 “LWQ” 向量化后得到

$$V=\underbrace{\underbrace{0,\cdots,0}_{10},\underbrace{0.33}_{P_{HL}},\underbrace{0,\cdots,0}_{89}}_{L}\underbrace{\underbrace{0,\cdots,0}_{35},\underbrace{0.08}_{P_{HW}},\underbrace{0,\cdots,0}_{62}}_{W}\underbrace{\underbrace{0,\cdots,0}_{70},\underbrace{0.33}_{P_{HQ}},\underbrace{0,\cdots,0}_{29}}_{Q}$$

再次, 根据每种氨基酸在训练集中样本的氨基酸链以及每个氨基酸在样本集中出现的概率构造 N×100 维的输入向量. 然后预测之间氨基酸的正确二级结构. 在数据集 CB513 和 RS126 上进行检验得到预测准确率分别为 Q_3=75.2%和 Q_3=73.8%.

7.3.4.3 双层 SVM

双层 SVM 编码方法首先为每一个样本集中的 n 个残基的序列定义一个 n 行 20 列的 PSI-BLAST 矩阵 [101]. 在预测系统的第一层, 每个残基编码为一个 21 维

的向量. 这个向量中的前 20 个元素对应 PSI-BLAST 矩阵中的元素. 至于第二层, 对应于一个残基的向量有 4 个元素, 其中前三个表示了三种二级结构 (H, E 和 C), 最后一个表示序列的方向. 如果窗口的长度是 l, 那么特征向量的第一层的维数为 $21\times l$ 维, 第二层的维数为 $4\times l$ 维.

用于预测系统的双层 SVM 结构的第一层是一个 SVM 分类器, 它把每个预测序列的每个残基都归入 H, E 和 C 三类之一. 第二层 SVM 分类器过滤第一层的输出值. 第二层的目标输出值与第一层的输出值一样.

这种方法根据 PSI-BLAST 程序产生样本集中的每一个蛋白质序列的多重序列比对图构造输入向量.

7.3.4.4　n 肽频数方法

由于自然情况下的蛋白质序列所包含的氨基酸残基数量是不同的, 所以可以把不同长度的氨基酸序列转化成等长向量的 n 肽频数方法有了用武之地. 这种方法是统计连续 n 个氨基酸残基, 然后计算他们在样本中出现的频率, 作为向量对应位置的坐标. 设一个长度为 q 的序列 P 对应的向量为 x, 那么 $R^{20\times n}$ 中的 x 的每一维是对应的 n 个不同的连续的氨基酸在序列 P 中出现的频率. 最简单的一肽频数方法是把序列 P 中的各个氨基酸残基数量直接作统计, 求它们的相对频数. 那么

$$x_i = \#\{s_k = A_i : k = 1, \cdots, q\}, \quad i = 1, \cdots, 20$$

然而, 仅仅计算得到的频率没有相邻氨基酸的信息, 不能很好地反映序列信息. 为了所得向量不但能反映序列的氨基酸成分还能反映氨基酸之间的相邻信息, 就需要计算连续两个或多个氨基酸残基片段的相对频率. 二肽频数方法是计算任意的二肽组合在一个氨基酸序列中的相对频数, 称作二肽频数. 二肽频数编码方法的计算结果是得到一个 20×20 的二肽频数矩阵

$$x_{ij} = \#\{(s_k, s_{k+1}) = (A_i, A_j) : k = 1, \cdots, q-1\}, \quad i, j = 1, \cdots, 20$$

这个向量处于 R^{400} 空间中. 为了使任意长度的序列嵌入后具有独立性, 要对这些向量进行归一化. 三肽频数方法是计算任意的三肽组合在一个氨基酸序列中的相对频数, 称作三肽频数. 三肽频数编码方法的计算结果是得到一个 $20\times 20 \times 20$ 的三肽频数立方阵

$$x_{ij} = \#\{(s_k, s_{k+1}, s_{k+2}) = (A_i, A_j, A_m) : k = 1, \cdots, q-1\}, \quad i, j, m = 1, \cdots, 20$$

这个向量处于 R^{8000} 空间中. 为了使任意长度的序列嵌入后具有独立性, 要对这些向量进行归一化. 对于一肽频数有

$$\sum_{i=1}^{20} x_i = 1$$

对于二肽频数有

$$\sum_{i,j=1}^{20} x_{ij} = 1$$

对于三肽频数有

$$\sum_{i,j,m=1}^{20} x_{ijm} = 1$$

得到的频数称作相对频数. 同理对 n 肽频数也是一样. n 肽频数矩阵

$$\begin{aligned} x_{(i+1)\cdots(i+n)} &= \#\left\{(s_{k+1},\cdots,s_{k+n})\right. \\ &\left.= (A_i,\cdots,A_j) : k = 1,\cdots,q-n\right\}, \quad i = 1,\cdots,20, n \leqslant 20 \end{aligned}$$

归一化

$$\sum_{i=1}^{20} x_{(i+1)\cdots(1+n)} = 1$$

这样得到的向量处于 20^n 维空间中. 从以往的工作来看 n 一般取 1~3. 这种方法多用于蛋白质的结构类预测、亚细胞结构预测等样本序列比较长的预测中. 例如 ASMWERVKSIIKSSLA 为一段螺旋结构, 图 7-6 以该二级结构段为例说明二肽频数编码方法, 图 7-7 说明三肽频数编码方法. 在的实验中使用了二肽频数编码方法.

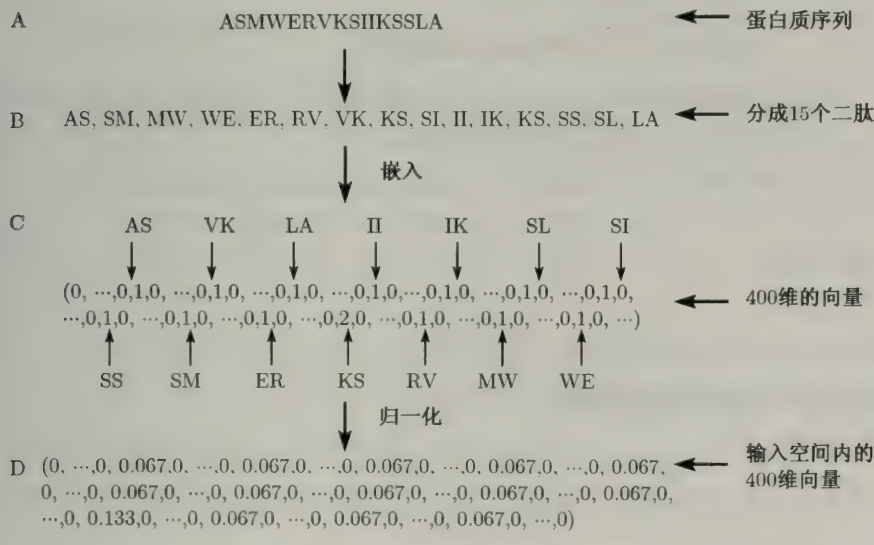


图 7-6 二肽频数编码方法工作流程

ASMWERVKSIIKSSLA 一个结构域的一级序列, 从这段序列中可以取得 15 个二肽. 通过计算每个二肽出现的频率, 得到一个 400 维的向量. 这个向量归一化后成为所需的向量

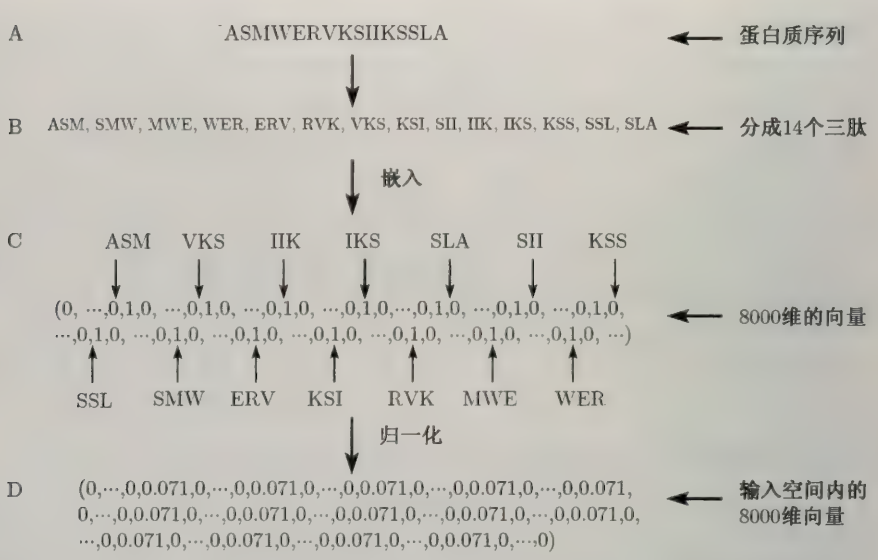


图 7-7　三肽频数编码方法工作流程

ASMWERVKSIIKSSLA 是一个结构域的一级序列, 从这段序列中可以取得 14 个三肽. 通过计算每个三肽出现的频率, 得到一个 8000 维的向量. 这个向量归一化后成为所需的向量

7.3.5　二级结构预测准确率评估方法

7.3.5.1　二级结构片段预测准确率评估方法

二级结构段预测准确率的评估是以二级结构段作为训练和检测的基本单位进行的. 定义了每种状态的三态总的预测准确率: $Q_H^{\text{prd}}, Q_E^{\text{prd}}, Q_C^{\text{prd}}, Q_H^{\text{obs}}, Q_E^{\text{obs}}$ 和 Q_C^{obs} 和 Q_3 来评估预测结果 [102]. 这些评估方法定义如下:

如果 i、$j \in \{H, E, C\}$, 二级结构 i 预测成为二级结构 j 的数量为 N_{ij}, i 类二级结构的总数量

$$\text{obs}_i = \sum_{j=1}^{3} N_{ij}$$

预测为 i 类二级结构片段总数量

$$\text{prd}_i = \sum_{j=1}^{3} N_{ji}$$

样本总量

$$N_{\text{seg}} = \sum_i \text{obs}_i = \sum_i \text{prd}_i = \sum_{i,j} N_{ij}$$

那么, 基于片段的三态预测准确率

$$S_3 = 100 \cdot \frac{1}{N_{\text{seg}}} \cdot \sum_{i=1}^{3} N_{ii}$$

每种二级结构三态预测百分比为

$$S_i^{\text{obs}} = 100 \cdot \frac{N_{ii}}{\text{obs}_i} \text{和} S_i^{\text{prd}} = 100 \cdot \frac{N_{ii}}{\text{prd}_i}, \quad i \in \{H, E, C\}$$

7.3.5.2 每个残基的预测准确率评估方法

(1) 预测矩阵 [102].

M_{ij}= 状态 i 预测为状态 j 的氨基酸残基数量, 其中 i、$j \in \{H, E, C\}$. 其中状态 i 预测残基的总数量为

$$\text{obs}_i = \sum_{i=1}^{3} M_{ij}, \quad j \in \{H, E, C\}$$

预测成为 j 的残基总数量为

$$\text{prd}_i = \sum_{j=1}^{3} M_{ji}$$

三种状态的残基总数量为

$$N_{\text{res}} = \sum_i \text{obs}_i = \sum_i \text{prd}_i = \sum_{i,j} M_{ij}$$

(2) 三种状态总的预测准确率.

$$Q_3 = 100 \cdot \frac{1}{N_{\text{res}}} \cdot \sum_{i=1}^{3} M_{ii}$$

(3) 每种状态正确率百分比. 两个变量可以定义每种状态的正确率百分比, 它们分别回答了以下问题: ① 有多少观测为螺旋 (折叠或卷曲)的残基被正确预测? 观察为某种状态的残基正确预测的百分比为 $Q^{\%\text{obs}} = 100 \cdot \frac{M_{ii}}{\text{obs}_i}$. ② 有多少预测的残基为螺旋 (折叠或卷曲) 的残基是正确的? 预测为某种状态的残基是正确的百分比为 $Q^{\%\text{prd}} = 100 \cdot \frac{M_{ii}}{\text{prd}_i}$.

(4) 信息索引. 信息索引是一种跟熵相关的信息, 它把不同的百分率混合到了一个量中, 在这个量中准确率矩阵中的元素被平等地处理. 信息索引为 info =

$\ln\left\{\dfrac{P_{\text{prd}}}{P_{\text{obs}}}\right\}$, 其中 P_{obs} 描述了一个观测为 i 的样本状态为点概率, P_{prd} 为预测矩阵$\{M_{ij}\}$的实现. 信息索引可以改写成

$$\text{info} = \frac{\text{info}^{\%\text{obs}} + \text{inf}\,o^{\%\text{prd}}}{2}$$

其中

$$\text{info}^{\%\text{obs}} = 1 - \frac{\sum_{i=1}^{3} \text{prd}_i \cdot \ln \text{prd}_i - \sum_{i=1}^{3} M_{ij} \ln M_{ij}}{N_{\text{res}} \cdot \ln N_{\text{res}} - \sum_{i=1}^{3} \text{obs}_i \cdot \ln \text{obs}_i}$$

$$\text{info}^{\%\text{prd}} = 1 - \frac{\sum_{i=1}^{3} \text{obs}_i \cdot \ln \text{obs}_i - \sum_{i=1}^{3} M_{ij} \ln M_{ij}}{N_{\text{res}} \cdot \ln N_{\text{res}} - \sum_{i=1}^{3} \text{prd}_i \cdot \ln \text{prd}_i}$$

(5) Matthew 相关系数

$$C_i = \frac{p_i \cdot n_i - u_i \cdot o_i}{\sqrt{(p_i + u_i) \cdot (p_i + o_i) \cdot (n_i + u_i) \cdot (n_i + o_i)}}$$

其中 $p_i = M_{ii}$, $n_i \sum_{j\neq i}^{3} \sum_{k\neq i}^{3} M_{jk}$, $o_i = \sum_{j\neq i}^{3} M_{ji}$, $u_i = \sum_{j\neq i}^{3} M_{ij}$.

(6) 片段交叠准确率评估方法. 每种状态的片段交叠

$$\text{SOV}_i = \frac{1}{N_i} \sum_{Si} \frac{\text{MINOV}\,(S_1; S_2) + \text{DELTA}\,(S_1; S_2)}{\text{MAXOV}\,(S_1; S_2)}$$

其中, S_1 和 S_2 是状态 i(包括螺旋 H、折叠 E 和卷曲 C) 观测的和预测的二级结构片段. LEN(S_1) 为片段 S_1 中残基的数量. MINOV(S_1; S_2) 为 S_1 和 S_2 实际交叠的长度. MAXOV(S_1; S_2) 为 S_1 和 S_2 中包含的所有状态 i 的残基总长度. DELTA(S_1; S_2) 为下面值的整数部分:

$$\text{DELTA}\,(S_1; S_2) = \min \left\langle \begin{array}{c} \text{MAXOV}\,(S_1; S_2) - \text{MINOV}\,(S_1; S_2) \\ \text{MINOV}\,(S_1; S_2) \\ \text{INT}\,(0.5 \cdot \text{LEN}\,(S_1)) \\ \text{INT}\,(0.5 \cdot \text{LEN}\,(S_2)) \end{array} \right\rangle$$

$N(i)$ 状态 i 残基的数量为

$$N_i = \sum_{S(i)} \text{LEN}\,(S_1) + \sum_{S'(i)} \text{LEN}\,(S_1)$$

其中 $S(i)$ 是片段$\{S_1; S_2\}$中所有残基对的数量, S_1 和 S_2 中至少有一个残基是相同的. $S(i)$ 是 S_1 中没有配对的残基数量.

三态片段交叠量为

$$\mathrm{SOV} = \mathrm{SOV}_3 = \frac{1}{N} \cdot \sum_{S(i)} \frac{\mathrm{MINOV}\left(S_1; S_2\right) + \mathrm{DELTA}\left(S_1; S_2\right)}{\mathrm{MAXOV}\left(S_1; S_2\right)} \cdot \mathrm{LEN}\left(S_1\right)$$

其中

$$N = \sum_i N_i$$

7.3.6 蛋白质二级结构预测结果

为了客观地评估预测结果, 实验中采用了多重交叉验证试验来减小由一次随机选择训练集和检测集带来的预测偏差. n 重交叉验证试验需要首先把样本集平均分成 n 等份, 然后其中一个子集作为检测集, 其余 $n-1$ 个子集合并成为训练集. 对训练集进行训练得到的分类器用于对检测集进行的检测. 然后以另外一个子集作为检测集进行下一轮试验. 这个过程重复 n 次. 这样, 每一个样本都可以被预测一次, 所以交叉验证试验的准确率就是被正确分类的样本的百分比. 本书报告的所有结果都由 7 重交叉验证试验取得.

在实验中, 我们使用 SVM 分类与鉴别蛋白质序列片段. 运用 SVM 预测蛋白质二级结构一般包括以下四个步骤: ① 采集蛋白质序列片段构建样本集. ② 编码样本以嵌入到输入空间. 在这个过程中, 氨基酸残基序列依据它的二级结构特征转化为输入空间中的向量. ③ 利用核函数把输入空间的向量映射到高维特征空间. ④ 为了对这些向量进行分类, 在高维特征空间中寻找优化分类超平面. 最优分类超平面这样优化得到: 最大化两类数据集中与这个超平面最近的训练样本的距离. 因为高维特征空间中的向量根据它的类别被最优分类超平面线性分开, 所以样本就在输入空间中被非线性地分开 [11].

实验中, 我们首先采用了径向基核函数 (RBF)

$$k_{\mathrm{rbf}}(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0 \tag{7.1}$$

来把向量映射到高维特征空间. 其中 γ 是参数, 它的值由使用者确定. 然后我们选择了模型 C-SVC

$$\min \ \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{l} \xi_i \tag{7.2}$$

$$\text{subject to} \ \ y_i \left(w^{\mathrm{T}} \Phi\left(x_i\right) + b\right) \geqslant 1 - \xi_i \tag{7.3}$$

$$\xi_i \geqslant 0, \quad i = 1, \cdots, l \tag{7.4}$$

来进行二元分类. (7.2) 式中的误差权重 C 也是由使用者来确定的参数, 它衡量误差罚分的大小.

根据 DSSP 的原则, 8 种蛋白质二级结构归类为三种, 那么蛋白质二级结构的分类是三元分类. 由于 SVM 的基本算法只能进行二元分类, 所以运用在 SVM 算法编程得到的软件系统要进行多次二元分类才能实现多元分类. 软件系统 Libsvm 运用投票法进行三元分类, 所谓投票分类法指的是每一次二元分类被看作是一次投票过程, 对每一个样本的分类就是一次投票. 最后, 样本划归得票最多的二级结构类. Libsvm 运用指数序列搜索方法 (如 $C = 2^{-5}, 2^{-4}, \cdots, 2^{15}; \gamma = 2^{-15}, 2^{-14}, \cdots, 2^{5}$) 逐格搜索 (7.2) 式中的参数 C 和 (6.1) 式中的参数 γ 以确定最优取值. 通过对 C 和 γ 逐格搜索, 数据集 RS126 在 C=2 和 γ=2 时预测准确率最优, 数据集 CB513 在 C=512 和 γ= 0.125 时预测准确率最优. 图 7-8 显示了对于 RS126 进行参数优化的结果. 图 7-9 显示了对于 CB513 进行参数优化的结果.

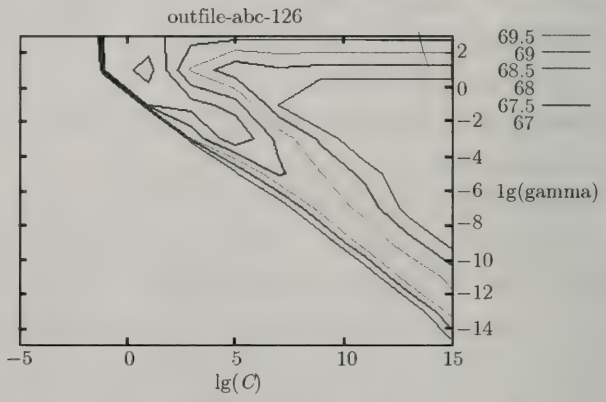


图 7-8　非冗余数据集 RS126 的参数 C 和 γ 的优化结果

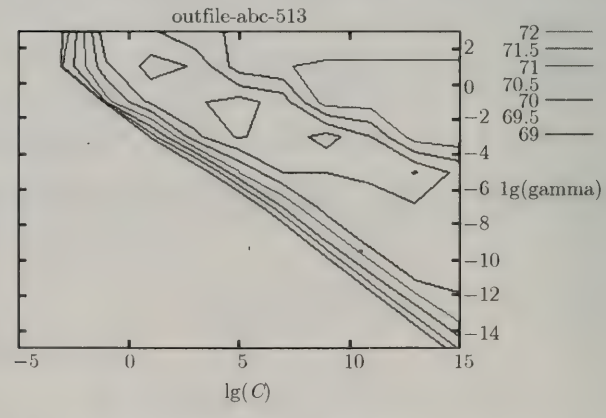


图 7-9　非冗余数据集 CB513 的参数 C 和 γ 的优化结果

实验中还利用 6 个二元分类器对样本进行分类, 每一个二元分类器训练两类不同的样本. 在实验中根据蛋白质序列片段的二级结构特征取得样本集中的样本. 样本的长度从 2 个残基到 99 个残基不等. 6 个二元分类器为: H/~H, 表示螺旋对折叠和无规则卷曲; E/~E, 表示折叠对螺旋和无规则卷曲; C/~C, 表示无规则卷曲对折叠和螺旋; H/E, 表示螺旋对折叠; E/C, 表示折叠对无规则卷曲; C/H, 表示无规则卷曲对螺旋. 6 个二元分类器利用最优参数对 RS126 和 CB513 进行分类. 表 7-4 展示了 6 个二元分类器以及它们对 RS126 和 CB513 的分类准确率.

表 7-4 二级结构段的二元预测准确率 (%)

二元分类器	RS126 样本集	CB513 样本集
H/~H	81.2	83.2
E/~E	80.7	82.0
C/~C	77.9	80.7
H/E	77.4	80.1
E/C	81.6	81.9
C/H	79.4	81.7

注: 该表显示的结果都来自于 7 重交叉验证试验. 6 个二元分类器为: H/~H, 表示螺旋对折叠和无规则卷曲; E/~E, 表示折叠对螺旋和无规则卷曲; C/~C, 表示无规则卷曲对折叠和螺旋; H/E 表示, 螺旋对折叠; E/C, 表示折叠对无规则卷曲; C/H, 表示无规则卷曲对螺旋.

根据二元分类的结果, 表 7-5 总结了三元预测的结果. CB513 样本集中以二级结构片段为单位的三态总的预测准确率 (S_3) 为 72.2%, RS126 样本集中以二级结构片段为单位的三态总的预测准确率 (S_3) 为 69.7%.

表 7-5 二级结构段的三元预测准确率 (%)

评估方法	RS126 样本集	CB513 样本集
S_3	69.7	72.2
S_H^{obs}	57.1	63.1
S_E^{obs}	66.9	67.2
S_C^{obs}	78.9	80.8
H^{prd}	65.9	71.2
E^{prd}	69.1	69.7
C^{prd}	72.8	74.1

注: 该表显示的结果都来自于 7 重交叉验证试验.

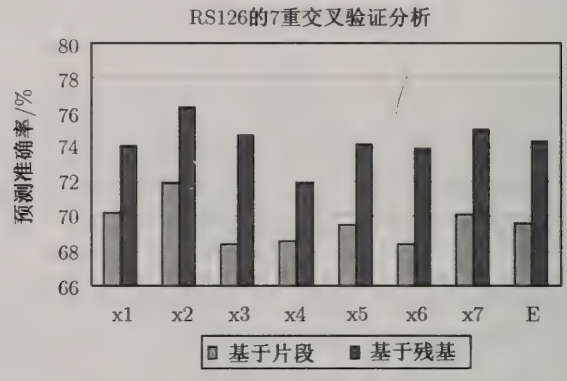
在得到了基于片段的准确率基础上, 得到基于残基的准确率是一件很容易的事. 表 7-6 展示基于残基的预测准确率.

由于数据量很大, 采用完全的 Jack-knife 试验的计算量是无法忍受的, 因此我们的实验采取了国际上通用的交叉验证方法. 图 7-10 展示了每一次交叉验证试验

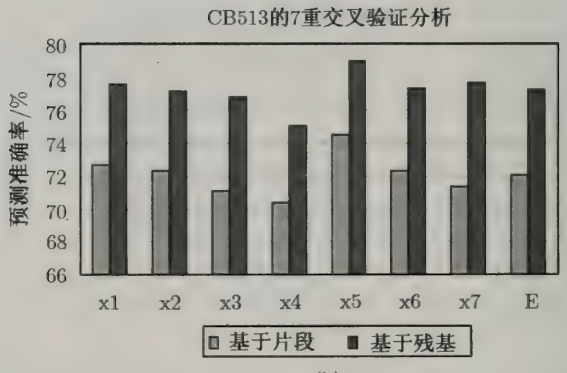
表 7-6　二元分类每个残基预测准确率

分类器	RS126 集	CB513 集
H/∼H	81.1	82.7
E/∼E	85.1	85.5
C/∼C	82.3	82.9
H/E	80.7	83.9
E/C	85.7	85.2
C/H	83.3	85.6

注：二元分类器意为：H/∼H: 螺旋对折叠和卷曲; E/∼E: 折叠对螺旋和卷曲; C/∼C: 卷曲对螺旋和折叠; H/E: 螺旋对折叠; E/C: 折叠对卷曲; C/H: 卷曲对螺旋.



(a)



(b)

图 7-10　RS126 数据集中基于片段的准确率和基于残基的准确率比较 (a) 和 CB513 数据集中基于片段的准确率和基于残基的准确率比较 (b). 准确率使用了 7 重交叉验证分析由 7 个检测集给出. 最后一栏给出了打分的期望

的结果. 其中图 (a) 表示的是基于片段的预测准确率, 图 (b) 表示的是基于残基的预测准确率. 每种准确率的实际数值由表 7-7 给出.

表 7-7 基于残基的和基于片段的 7 重交叉验证预测准确率

类别	基于残基的预测准确率/%		基于片段的预测准确率/%	
	RS126	CB513	RS126	CB513
总准确率	74.3	77.3	69.6	72.1
H^{obs}	73.4	79.1	57.1	63.1
E^{obs}	63.3	64.6	66.9	67.2
C^{obs}	80.7	82.3	78.9	80.8
H^{prd}	73.6	78.5	65.9	71.2
E^{prd}	69.8	70.9	69.1	69.7
C^{prd}	77.4	79.1	72.8	74.1

表 7-8 展示了三元预测的结果. 总的三态残基准确率 (Q_3) 对于 CB513 的预测是 77.3%, 标准差为 0.98%, 对于 RS126 的 Q_3 预测准确率是 74.3%, 标准差 1.2%. 总的三态准确率为 77.3%是一个很好的预测结果. 通过这个实验还可以看到仅仅改善编码方法, 而保持算法不变就可以提高蛋白质二级结构预测的准确率.

表 7-8 方法 SVM^{sw}, SVM^{ssp}, PMSVM 和 SVM^{df} 的预测准确率比较

方法	Q_3/%	Q_H^{obs}/%	Q_E^{obs}/%	Q_C^{obs}/%	Q_H^{prd}/%	Q_E^{prd}/%	Q_C^{prd}/%	MCC		
								C_H	C_E	C_C
SVM^{sw}	71.2	73	58	75	77	66	69	0.61	0.51	0.52
SVM^{ssp}	73.8	75	61	81	81	69	70	-	-	-
PMSVM	74.0	79.3	69.3	72	79.4	66.4	73.6	0.7	0.6	0.59
SVM^{df}	74.3	73	63	81	74	70	77	0.59	0.57	0.63
SVM^{sw}*	73.5	75	60	79	79	67	70	0.65	0.53	0.54
SVM^{ssp}*	75.2	78	65	80	82	71	71	-	-	-
PMSVM*	75.2	80.4	71.5	72.8	79.4	66.4	76.4	0.71	0.61	0.61
SVM^{df}*	77.3	79	65	82	79	71	79	0.66	0.59	0.67

注: SVM^{sw}、SVM^{ssp} 和 SVM^{df}: 结果取自 RS126 数据集;

PMSVM: 结果取自 CB396 数据集;

SVM^{sw}*、SVM^{ssp}*、PMSVM 和 SVM^{df}*: 结果取自 CB513 数据集;

SVM^{df} 与 SVM^{df}* 的结果源自二肽频数编码方法;

-: 结果不能从原文献中找到.

三种蛋白质二级结构的统计特征是不一样的. 为了对各种二级结构进行进一步的研究, 我们计算了每种二级结构预测成为其他二级结构的预测准确率. 表 7-9 中展示了计算结果, 其中行中显示的是观测到的比率, 列中显示的是预测的比率. 以 Helix 行为例说明表 7-9 的意义, 该行中数值的意义为观测为螺旋的二级结构中有

63.1%预测为螺旋, 13.2%预测为折叠, 23.7%预测为卷曲. 其他数值的意义以此类推.

表 7-9　每种二级结构预测为其他二级结构的百分率

		预测比率		
		螺旋	折叠	卷曲
观测比率	螺旋	63.1	13.2	23.7
	折叠	11.3	67.2	21.5
	卷曲	8.1	11.1	80.1

注: 表中所有结果来自 CB513 数据集.

支持向量机是由支持向量确定的分类机, 同时运算的复杂度由训练样本集中的支持向量和 VC 维确定. 支持向量越多该训练样本集越复杂. 表 7-10 展示了几种支持向量机方法中出现的支持向量与样本集中训练样本数量的比率.

表 7-10　每种方法的支持向量与总向量的比率 (%)

分类器	二肽频数方法 SVM^{df}	滑窗方法 SVM^{sw*}	N 格模型方法 SVM^{ssp*}
H/∼H	49.2	50.46	40.9
E/∼E	47.1	43.92	36.5
C/∼C	54.9	59.02	55.0
H/E	53.8	50.27	36.0
E/C	50.3	53.16	48.5
C/H	51.1	52.62	46.1

注: 表中的所有分类器都是二元分类器. 结果基于 CB513 数据集的 7 重交叉验证试验. SVM^{sw*} 的结果来自 Hua & Sun (1993), SVM^{ssp*} 的结果来自 Yang (2003), SVM^{df} 是我们的运算结果.

SVM 方法于 2001 年由 Hua 首次应用于蛋白质二级结构预测. 从那时起, 这种方法逐渐被人们认识并应用于蛋白质折叠类预测、蛋白质亚细胞定位预测 [137] 以及蛋白质相对溶解性特征评估 [105] 等方面. 同时, SVM 的编码方法和预测准确率评估方法也得到改进. SVM 的编码方法大体包括两类, 一类为滑窗方法, 这种方法多用于比较短的氨基酸序列的向量化; 另一种为氨基酸片段的统计信息方法, 这种方法多用于比较长的氨基酸序列的向量化. 实验中介绍了以二级结构段为基本单位的蛋白质二级结构预测准确率评估方法. 基于蛋白质二级结构段的预测准确率评估方法从一个新的角度评估蛋白质二级结构预测准确率, 这种方法要比基于氨基酸残基的预测准确率评估更能反应三级结构的本质 [106]. 同时, 对于二级结构段的预测准确率评估以二肽频数编码方法为基础. 根据二肽频数编码方法, 具有相同二级结构特征的相邻氨基酸残基都会处于样本集的同一元素中. 如果一个样本被准确预测, 那么它所包含的所有氨基酸残基都被正确预测. 反之, 如果向量被错误预测, 它所包含的氨基酸残基都会被错误预测.

利用二肽频数编码方法, 如果氨基酸序列中相邻的氨基酸残基属于同一个二级结构区域, 那么不管这个二级结构序列片段的长度如何, 它们都会被分配到一个样本中. 否则, 如它们属于不同的二级结构区域, 则它们会被分配到不同的样本中. 如果样本预测正确, 样本中所有的残基都会被正确预测. 反之, 如果样本预测错误, 那么样本中的所有残基都会被错误预测. 因此预测的残基准确率与片段准确率是一致的. 通过该实验, 在 CB513 和 RS126 预测分别得到的 Q_3 准确率超过 77%和 74%. 而在 CB513 上预测得到的结果比当前使用相同数据集预测的结果要高出两个百分点左右.

利用以前的编码方法, 为了使向量保持在同一个输入空间, 样本长度必须固定. 由于构成同一蛋白质二级结构的氨基酸残基链的长度是不同的, 所以为了保持样本长度一定, 属于不同二级结构的氨基酸残基就必须被分配到同一个样本中. 残基预测准确率与片段预测准确率不是一致的. 同一个片段中的残基, 有的可能正确预测, 而另外一些则错误预测. 同时, 使用这些编码方法, 必须小心地选择样本长度以确保预测的准确率较高. 因为样本长度过短, 残基序列片段会丢失重要的分类信息, 而太长的残基序列则会带来噪音.

采取二肽频数编码方法的主要优势在于: ① 不同长度的蛋白质序列片段可以编码进入同一个输入空间, 所以实验者可以仅仅依据氨基酸序列的二级结构属性来采集训练和检测样本; ② 从蛋白质序列片段中插入或删除个别残基不会引起二肽频数的很大变化; ③ 二肽频数编码方法可以保持输入空间有较小的维数.

然而这种方法也有不足之处, 即对于给定的氨基酸序列目前还没有有效的方法确定二级结构的分段情况.

第 8 章　蛋白质折叠类型的预测

8.1　简　　介

大约半个世纪以前人们已经初步证明了蛋白质一级序列可以确定三维结构, 蛋白质通过三维结构实现自身的功能 [107]. 在自然状态下, 蛋白质的折叠类型不超过 1000 种 [108]. 然而生物体正是依靠这神奇的、区区 1000 种的折叠类型及其组合实现了整个生命历程中的全部生物功能. 此外, 对蛋白质相互作用数据的分析表明蛋白质相互作用的数量也是有限的, 它们的数量约为 10 000[109]. 研究还表明生物界中控制林林总总的生命体遗传和生长发育的基因的数量是有限的, 大体上这个数量为 30 000[110]. 由于不同蛋白质之间的相互作用和蛋白质与相应配体之间的相互作用都由它们的三维结构决定, 所以收集、探索和挖掘蛋白质结构数据库中的这类信息对于生命本质研究至关重要.

由于实验方法、实验设备、工作人员知识结构和思想以及人类掌握的相关知识等方面的问题, 对于生物体基因序列的研究、这些基因可以表达的生物分子的结构的研究以及这些结构可以表现出来的功能的研究之间存在不平衡. 一方面, 沉淀在序列数据库中的数据越来越多, 通常这些序列是功能不很清楚的原始数据; 另一方面, 在蛋白质数据库 (protein data bank) 中的结构信息积累相对缓慢 [111]. 因此同源建模方法和计算机仿真方法等计算方法就成为预测蛋白质结构的实验方法以外的重要补充.

蛋白质结构预测的计算方法的核心问题是明晰残基序列与蛋白质三维结构之间有机的、必然的联系和挖掘蕴涵在残基序列中的结构信息. 可以想象, 由于构成合理长度蛋白质一级序列的 20 种氨基酸之间的组合是一个极大的数字, 而蛋白质的折叠类型仅为 1000 种的话, 那么序列构成这些折叠类型的方式一定遵循某种规律. 到目前为止人们已经尝试了几种方法来探索这些规律. 这些方法包括氨基酸组分方法 (ACC)[112~120], 神经网络方法 (ANN)[121], 隐马尔可夫模型 (HMM) 方法 [122] 以及支持向量机方法 (SVM)[123].

这些方法中, 氨基酸组分方法和双组件效果的氨基酸组分方法的研究最充分 [124,125]. 仅依赖序列中氨基酸成分, 即仅依赖氨基酸残基在序列中的百分比而不考虑其他因素的影响, 预测准确率就可以达到 80%[126]. 在这种方法上发展起来了双组件效果的氨基酸组分方法和双组件算法 [124,127,128]. 近十年来, 使用双组件算法用于预测蛋白质结构类可以达到很高的准确率 [129~133].

支持向量机方法是很好的机器学习方法, 它利用最大边界对两类样本进行分类, 这种分类方法可以得到很好的泛化效果. 其他的机器学习方法比如 ANN 与 HMM 方法利用最小错误方法对两类数据进行分类, 这种方法的缺陷在于泛化能力有限 [134]. 支持向量机方法曾用于蛋白质结构类型的预测中 [135,136].

根据以前的定义, 蛋白质根据二级结构组成的不同可以分为四种结构类: α、β、$\alpha+\beta$ 和 α/β[137,138]. 探索蛋白质二级结构是构建 3D 结构的第一步 [139]. 由于如果知道了蛋白质的结构类型就可以有针对性地专注于某些类型的研究, 因此可以简化探索的过程 [123].

结构域是能够独立折叠为稳定的三级结构的多肽链, 由不同的二级结构和超二级结构组合形成. 一条多肽链在一个域范围内来回折叠, 相邻的域常被一个或两个多肽片段连接. 一个蛋白质可以只包含一个结构域也可以由几个结构域组成. 结构域通常由 20~700 个氨基酸残基组成, 其特点是在三维空间可以明显区分和相对独立, 并且具有一定的生物功能如结合小分子. 模体 (motif, 又称基序) 是结构域的亚单位, 通常由 2~3 二级结构单位组成, 一般为 α 螺旋、β 折叠和环.

结构域在蛋白质中这种组合的数目是有限的, 一些结合方式似乎是蛋白质结构所偏爱的, 并且相似的结构域结构在具有不同功能不同残基序列的蛋白质中经常重复出现. 通常多结构域蛋白质中不同的结构域是与不同的功能相关联的. 某个种属的多个独立的多肽链完成的几种生物学功能可以由另一个种属的一个蛋白质中的不同结构域来完成 [140]. 对那些较小的球状蛋白质分子或亚基来说, 结构域和三级结构是一个意思, 也就是说这些蛋白质或亚基是单结构域的. 较大的蛋白质分子或亚基其三级结构一般含有两个以上的结构域, 即多结构域的, 其间以柔性的铰链相连, 以便相对运动.

一般认为结构域是蛋白质功能的基本结构单位, 结构域有时也指功能域. 功能域是蛋白质分子中能独立存在的功能单位, 它可以是一个结构域, 也可以是由两个或两个以上结构域组成. 另外, 很多的折叠过程有基本的结构域编码, 靠理解这些相对较小的折叠过程的细节, 可以获得关于蛋白质怎么进行折叠的比较一般的观点. 因此蛋白质结构域类型的预测对于蛋白质折叠过程的研究和蛋白质结构和功能关系的研究具有重要意义.

PDB 数据库中已知结构的蛋白质在过去 30 年中呈指数增长. 由于从结构基因组工程获得的动力和技术使得允许高通量的结构解析, 这种蛋白质结构增长的趋势可以持续下去. 随着已知三级结构数据库的快速扩容, 蛋白质结构比较的重要性与序列比对的重要性具有同等地位. 虽然到目前为止结构域还没有统一的定义, 然而人们已经根据不同的需要建立了很多蛋白质结构比较的方法, 这些方法使用了不同的蛋白质结构、评估相似性的方法以及优化算法.

下面介绍几个比较常用的结构域数据库.

8.2　蛋白质结构域数据

与蛋白质二级结构的定义方法相比, 蛋白质结构域的定义方法显得更加主观一些. 其中除了 FSSP 数据库是根据 DALI 算法 [33] 自动定义的以外, CATH 数据库 [141~145] 和 SCOP 数据库 [146] 结构确认的过程中都包含了主观的成分. 蛋白质折叠分类问题可以归为两步. 第一步, 一个蛋白质序列或复合体被分解为结构域子序列; 第二步, 这些结构域子序列按照一定的规则划归一个结构域类别. 结构域分类方法中既有完全自动的方法, 也有依赖于专家的理解和知识的手工分类方法. 这里主要介绍三个常用的蛋白质结构域数据库和分类方法, 这三种方法使用了不同的分类策略, 它们通过建立蛋白质相似性等级来比较蛋白质结构域的类别.

8.2.1　DALI 算法和 FSSP 数据库 —— 距离矩阵比对的蛋白质结构比较

DALI 算法是一种在蛋白质结构域确认中常用的算法, 这种算法用来进行蛋白质结构的两两优化比对. DALI 算法使用了二维矩阵来进行蛋白质结构比对, 通过该算法进行比对所得的结构是一系列从结构上等价的氨基酸残基对, 这些残基对构成的等价的片段可以自由地改变其在序列中的位置 [147].

距离矩阵常常用来对蛋白质的构造进行描述和比较. 最常用的距离矩阵是两个氨基酸残基中 C_α 原子之间的距离矩阵, 这种距离矩阵可以很好地描述蛋白质的三维结构. 矩阵中的元素对于结构单位坐标保持了相对的独立性, 这种矩阵不但可以描述蛋白质的结构, 而且其中还包含了利用几何方法重新构建三维结构的信息. 然而一般情况下蛋白质的手性信息不会包含在其中.

相似的三维结构具有相似的残基之间的距离, 这是这种 C_α 原子之间的距离矩阵工作的基础. 当比较两个蛋白质的结构时, 把其中一个代表某种结构域的理想的蛋白质 C_α 原子之间的距离矩阵与另一个待求结构的蛋白质的 C_α 原子之间距离矩阵的最顶端部分相比较, 然后垂直或平行移动第一个矩阵. 通过这种矩阵的相对位置移动, 如果找到了与其相匹配的子矩阵, 对齐第一个矩阵与第二个矩阵的子矩阵的主对角线. 那么与这个子矩阵对应蛋白质结构就是相似的二级结构. 与第一个矩阵不匹配的矩阵对应的部分对应的结构就是二级结构之间的链接.

DALI 算法中首先把需要比对的两个蛋白质的三维坐标输入一个用于计算残基序列中两个 C_α 原子之间的距离矩阵. 距离矩阵首先分解一定规模的子矩阵, 比如 "六肽 – 六肽" 子矩阵; 然后对两个矩阵的子矩阵进行匹配, 并由此组合成较大协调匹配子矩阵集; 最终运用蒙特卡罗优化方法对这个分子间距离矩阵进行相似性打分. 由于几个比对运算可以同时进行, 那么就可以同时产生最好的和稍逊的检测结果. 这种方法允许进行比对的序列中存在任意大小的序列缺失、反转以及插入序

列. 这种优化比对方法完全自动进行, 并且可以在结构出现了扭曲变化时准确地鉴别蛋白质三维结构的相似性和三维结构的核心.

8.2.1.1 定义矩阵

如果两个相互比较的蛋白质为 A 和 B, 与这两个蛋白质对应的矩阵的相似性打分 S 为

$$S=\sum_{i=1}^{L}\sum_{j=1}^{L}\Phi(i,j) \tag{8.1}$$

其中 i 和 j 分别是匹配结构残基对的序号, L 是匹配残基的数量, Φ 是比较方法, 在距离矩阵中就是 C_α 原子之间的距离 d_{ij}^A, d_{ij}^B. 不匹配的残基对不会提高总的打分值. 当所比对结构完全相似时, 由 $\Phi(i,j)$ 确定的 S 值达到最大.

结构相似性的比较可以分为两类 [147]:

(1) 在结构数据库中寻找一个预定义的结构模式;

(2) 在两个蛋白质中查找最大的匹配结构.

前者定义目标函数很容易, 就是最大化相似性打分. 而后者是更一般的情况, 需要定义一个相似性的衡量方法协调两个矛盾的需要: 最大化等价残基数量和最小化结构偏差. 这种匹配结构的刚性相似性打分方法为

$$\Phi^R(i,j)=\theta^R-\left|d_{ij}^A-d_{ij}^B\right|$$

其中上标 R 代表刚性打分, d_{ij}^A, d_{ij}^B 为在蛋白质 A 和 B 的距离矩阵中的等价子结构, $\theta^R=1.5$Å 是指相似性为零.

与上面的刚性相似性打分对应的还有一种柔性打分方法. 这种打分方法使用等价残基之间距离的相对性代替了等价残基之间距离之间的绝对偏差. 柔性打分中增加了一个弹性变量 E. 这个弹性打分 $\Phi^E(i,j)$ 对于逐渐积累渐进几何扭曲具有容忍性

$$\Phi^R(i,j)=\begin{cases}\left(\theta^E-\dfrac{\left|d_{ij}^A-d_{ij}^B\right|}{d_{ij}^*}\right)w\left(d_{ij}^*\right), & i\neq j\\ \theta^E, & i=j\end{cases}$$

其中, d_{ij}^* 是 d_{ij}^A, d_{ij}^B 的平均值, θ^E 是相似性阈值, w 是封装函数. θ^E=0.2 的意思是20%的偏差. 在结构比对中 20%的偏差意为 β 折叠中毗邻的折叠股之间的匹配偏差小于 1Å, 而正常的 β 折叠股之间的距离为 4~5 Å. 封装函数 $w(r)=\mathrm{e}^{-\frac{r^2}{\alpha^2}}$, 其中 α=20 Å.

8.2.1.2 贪婪算法

比对算法包括两步. 第一步是两个距离矩阵中的所有相关模式子结构的两两比对. 相似相关子结构储存在匹配列表中, 这个列表是结构比对的行. 第二步的目的

是把相关子结构的行合并成为较大匹配相容集, 使得 (8.1) 式的相似性打分达到最大. 处理相似子结构比对过程中使用了蒙特卡罗方法处理组合复杂性. 这种算法包括了两个步骤: 构建距离矩阵和矩阵的综合比对.

1. 分解距离矩阵

矩阵的容量的增加与序列长度 N 的平方成正比, 两个相似子矩阵之间可能比较的数量与两个矩阵容量的乘积 $N_A^2 \cdot N_B^2$ 成正比.

(1) 缩小矩阵. 相邻的链接模式可能重叠 11~12 个残基. 为了防止反复的重叠, 需要把蛋白质链分割成为较小的片段. 这些片段的长度大体上与二级结构片段的平均长度相似. 连续的六肽片段如果具有相似的模式, 那么这些片段可以链接成为较长的片段, 例如在一个 α 螺旋中.

(2) 匹配的相似子结构列表. 每个距离矩阵中的相似子结构根据平均内模式距离进行分类, 那么根据这个分类可以构建从较短的子结构直到长程交互作用的匹配结构列表. 减小后的距离矩阵 A 首先与对原来的距离矩阵 B 进行比较. 然后减小后的矩阵 B 再与原来的矩阵 A 进行比较. 进行了两次比较以后, 矩阵 A 和 B 中的冗余子结构就可以被删除.

2. 综合比对

第一, 蒙特卡罗优化. 蒙特卡罗的核心思想是靠随机游动来进行循环改善, 这种游动可能常常进入非优化领域. 随机移动的概率是

$$p = \mathrm{e}^{\beta * (S' - S)}$$

其中 S' 是移动后的打分, S 是移动前的打分, β 是参数.

基本的随机移动相当于增加或删除残基, 这种移动的效果是相似性打分的增加或减少. 通过进行蒙特卡罗搜索可以产生一个轨迹, 这个轨迹由匹配的子结构构成. 蒙特卡罗优化从一个匹配的子结构开始, 这种优化有两种基本的运行模式: 膨胀模式与削减模式. 从任何四肽片段的比对中删除 (但不需要它们与邻居交叠) 这样对总的相似打分就给了一个净减少的贡献.

第二, 选择优化方案. 在优化过程中, 几个轨道优化同时进行可以覆盖较宽的优化范围. 为了使优化更简便, 在此过程中可以选择冗余的或低打分值的比对. 在下面介绍的选择方案的三个阶段中, 最高打分的矩阵为数不多, 他们之间相互比对范围很窄. 这个阶段需要重复一次或多次扩张和削减循环. 首先进行 5 次膨胀模式, 然后进行一次削减模式, 如此循环. 每次循环后比对会打分有所改善.

(1) 播种阶段. 扫描相似子结构序列得到所有非覆盖六肽三连子. 例如子结构对 $(a,b)-(a',b')$, $(a,c)-(a',c')$, $(b,c)-(b',c')$ 可以形成三连子 $(a,b,c)-(a',b',c')$. 这些种子产生于所有比对结构子结构, 例如包含在三连子中的 "a-a". 种子的数量一般来说是 100. 强相似结构对产生了少和长的种子比对. 每个种子用于初始化轨

道, 这个轨道每次膨胀、削减循环增加一个的长度. 如果两个轨道的等价确认小于50%的同一性, 低打分的一个被消除. 所有的比对根据打分排序, 最高打分的比对保留下来.

(2) 分歧轨道优化. 在同时进行的比对中优化连续进行, 直到所有比对都是最优值, 即直到打分在 20 个膨胀、削减中不再改善.

(3) 最优比对的改进. 上一个阶段的比对基本完成后, 结果中可能有次优化的比对. 这是因为有些片段很难全部完成相关四肽片段的转移、有限次数的步骤以及高的相似性打分. 为了找到近优化比对的局部环境, 最好的比对用于初始化 10 个平行的轨道, 其中 30%的比对为随机游动. 它们在第二步时得到优化. 轨道在每 20 个膨胀或削减循环后重新初始化, 直到得到最优结果.

DALI 算法使用了二维矩阵, 可以用于测定了 C_α 碳原子坐标的蛋白质结构比对中. 这种方法完全是自动进行、概念上简单、鲁棒性强的方法. 虽然蒙特卡罗优化方法不能保证得到全局优化解, 这种算法还会得到精度很高的比对. 由这种算法比对得到的数据库 FSSP 已经成为常用的蛋白质结构域数据库之一.

8.2.1.3 FSSP 数据库

FSSP 是结构相似蛋白质家族 (families of structurally similar proteins) 的缩写, 其蛋白质来源于蛋白质数据库. 目前 FSSP 数据库中约有 330 种具代表性的蛋白结构家族. 收录蛋白标准为: 彼此结构同源性范围为 30%~70%. 小于 30%被认为同源性较小, 高于 70%, 则被认为结构差别不大 [92].

FSSP 的功能大体可以包括下面几个方面 [92]:

(1) 可用于研究蛋白质折叠进化中保守性与多样性;

(2) 研究结构相似蛋白质间关系;

(3) 确定蛋白质结构的核心部分, 以便进行模建及蛋白质改造;

(4) 检测同源性分析结果的可靠性;

(5) 蛋白质结构统计分析.

8.2.2 CATH 蛋白质结构域数据库

8.2.2.1 简介

CATH 是蛋白质结构等级结构域分类数据库. 其中的结构类晶体分辨率大于 3Å. CATH 中的结构域来自 PDB, 其中 53%的结构域是自动定义的, 其余的是手工定义的 [148]. 如图 8-1 所示.

蛋白质的结构类由蛋白质进化产生, 其中同一个结构类中的序列相似性很低. 因此, 通过于已知结构的比较来确定未知结构的功能有可能进行. CATH 采用半自动的方法来获得一种新的蛋白质结构域的分类等级. 其中 4 种主要的层次为蛋白

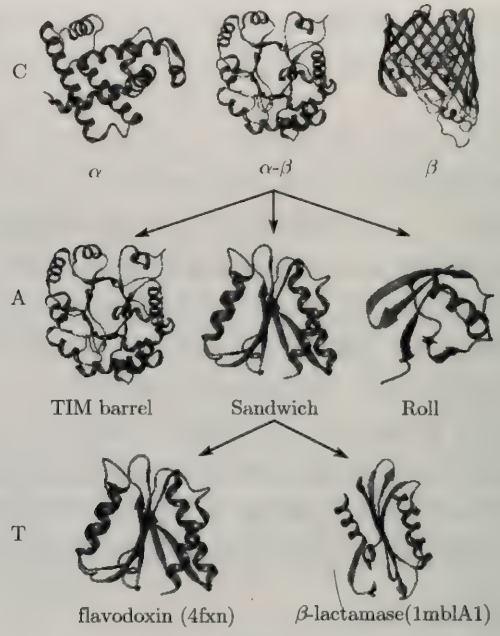


图 8-1　CATH 数据库中结构层次示意图

(源自 http://www.biochem.ucl.ac.uk/bsm/cath/cath_info.html)

质类 (protein class, C)、结构体系 (architecture, A)、拓扑 (topology, T) 和同源超族 (homologous superfamily, H). 类是最简单的层次, 它描述了每个结构域的二级结构成分. 结构体系说明了有二级结构单位构象倾向. 由于考虑了连续的连接, 在拓扑层同一个结构体系中的成员可能拓扑类型完全不同. 如果同一拓扑中的结构具有很高的相似性并且具有类似的功能, 这些结构被认为是进化上相似的, 那么它们就归为同一个同源超族 (S).

下面简要介绍 CATH 的 4 种主要的层次 [149].

(1) 类 (C 层): 二级结构成分和链接方式. 类是根据蛋白质二级结构成分和链接方式确定的. 使用 Michie 等 (1996) 方法 [150] 类中 90%的结构自动归类, 其余的使用手工方法归类. C 层是结构域分类的最顶层, 该层包含了三大类: α 类、β 类和 α-β 类. α-β 类虽然可以分开构成 α/β 和 $\alpha+\beta$ 类, 在 CATH 数据库中由于考虑到二级结构之间的连接, 还是把它们放在一起. 另外, CATH 还把包含很少数量二级结构的结构域归为单独的一类.

(2) 结构体系 (A 层): 二级结构和独立连接的总排列. 该层在同一类中区分结构体系, 但不区分不同的拓扑. 结构体系描述了蛋白质折叠形态的一般特征, 所以它包含的成分有时很杂. 结构体系包含不同连接的结构, 这些结构可以在拓扑中区

分开. CATH 中的结构体系层根据结构的相似性由手工分组, 而不管其中的二级结构的大小和数量.

(3) 拓扑 (T 层): 折叠族. 该层的结构域运用了 SSAP 结构比较算法进行分组, 其拓扑类别根据总的二级结构的形状和连接确定. 其中的结构域具有 70 以上的 SSAP 得分和至少 60%的大蛋白与小蛋白的匹配. T 层的结构具有同样的总折叠, 也就是说它们具有同样数量的二级结构和排列方式, 而且连接这些二级结构的链接也是一样的. 同一个拓扑中的结构域是相似的, 但是功能可能不同.

(4) 同源超族 (H 层): 高度相似的结构和功能的相似性. 在 H 层的结构域具有高度的结构上的相似性和功能的相似性, 也就是说它们可能来自同一个祖先, 在祖先蛋白中它们可能是核心包或活性位点. 该层的结构域同样根据 SSAP 结构域比较算法进行分组. 满足下面条件的结构域属于同一个同源超族:

① 序列同源性大于 35%, 较大蛋白匹配小蛋白的比率大于 60%;

② SSAP 的比较得分大于 80, 序列同源性大于 20%;

③ SSAP 的比较得分大于 80, 较大蛋白匹配小蛋白的比率大于 60%.

序列家族 (S 层): 极高的序列相似性和高度的结构和功能相似性. 该层中的结构域具有大于 35%的序列相似性, 因此可以推测具有极高的结构和功能相似性.

关于几种结构类别在不同层次上所包含的结构域种类见表 8-1.

表 8-1 CATH 的 4 种主要的层次包含结构域数量

类别	2.5.1 版				2.6.0 版			
	A	T	H	S	A	T	H	S
α	5	227	428	948	5	251	465	1402
β	19	139	292	951	19	160	311	1443
α-β	12	368	648	2010	14	414	706	3014
few secondary structures	1	86	91	114	1	82	90	144

8.2.2.2 蛋白质结构比对算法 ——SSAP

在比较蛋白质序列的过程中, 这种算法使用定义氨基酸类型的矩阵来确定蛋白质序列之间的关系. 这个矩阵提供了被比较的序列之间氨基酸两两相似性打分衡量方法. 由两个序列所有的氨基酸对确定的矩阵 (打分矩阵) 使用动态规划方法来发现两个序列的最佳比对 [151].

由于进行结构比较的距离矩阵独立于残基的坐标框架, 并且矩阵中包含了笛卡儿坐标中的所有信息, 这种矩阵可以提供接近实际结构比较问题的理想描述方法 (不包括手性). 为了能够比较在不同蛋白质高级结构中的残基, 从而使结构矩阵具有通用性, 距离矩阵中的元素使用了结构中残基的相对距离, 即固定一个残基的位置, 计算其他残基与这个残基的距离. 这样, 仅仅依赖蛋白质高级结构中的残基就

可以为算法定义一个操作环境.

1. 动态规划算法

动态规划算法是用于寻找两个线性序列优化比对的用途非常广泛的方法, 这种算法能够计算出两个序列组成成分之间的关系. 序列之间的关系的定义可以是种间的, 其中每一个给定类型的成分与其他类型有固定的关系, 或者每一个位置上的残基对有唯一的关系 [152].

对于两个段序列之间的比对 (图 8-2), 假设比对开始于序列的 C 端 (矩阵的左下端), 分值向序列的开始的方向积累 (左上). 这样可以找到最高的分数 (S), 通过追寻路径可以找到分值积累的途径. 计算方法方法由以下循环方程定义:

$$S_{ij} = D_{ij} + \left\{ \begin{array}{c} S_{i+1,j+1} \\ \max\limits_{l=j+2\to N_B} S_{i+1,l} - g \\ \max\limits_{k=i+2\to N_A} S_{k,j+1} - g \end{array} \right\} \tag{8.2}$$

其中 S 是矩阵中的任意元素, D 是序列的两个成员之间关系的衡量方法, g 是罚分常数. 序列 A、B 的长度 N_A 和 N_B. 最大行和列的计算效果值保留下来, 并不计算每一步的值. 最后效果表示为: S=max$\{\cdots\}$.

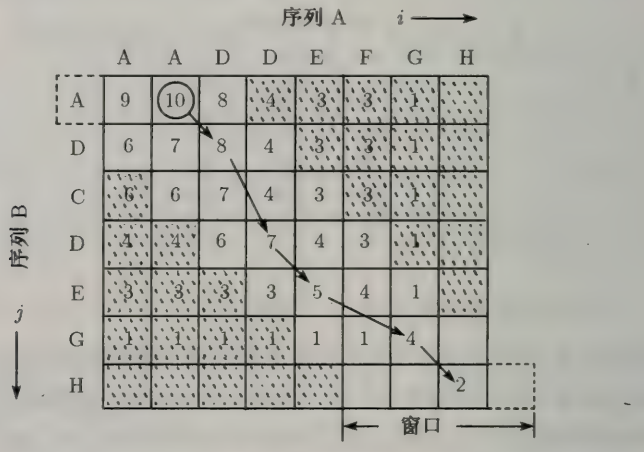


图 8-2　基本动态规划算法 (源自: Taylor W R, Orengo, C A. 1989)

2. 距离比较方法

在最简单的公式中, 这种方法只考虑两种结构的 C_α 原子以及它们之间距离的比较. 两对残基之间距离比对打分公式如下:

$$s = a/\left(\left|{}^{A}d_{ij} - {}^{B}d_{kl}\right| + b\right)$$

其中 s 是距离比较打分, d 是蛋白质 A 中的原子 i 和 j 以及原子 B 中的原子 k 和 l 之间的距离. 分母加上常数 b 的主要原因是防止分母为零. 常数 a 用来限制可能的最高分. 两个相互比较的序列在比对打分过程中, 所有的距离都用上面的方法比较. 最后总的得分为各个比较得分的和

$$S_{ik} = \sum_{m=-n}^{+n} a/\left(\left|{}^{A}d_{i,i+m} - {}^{B}d_{k,k+m}\right| + b\right) \tag{8.3}$$

如果两个位置的总的比较得分为 S_{ik}, 那么距离比较的次数为 $2n$, 结构 A 和结构 B 中的原子 i 和 k 分别为每次比较的中心. 定义了两个位置的得分后, 由动态规划算法来计算所有位置的最优比对.

另一种距离比较的方法使用了 C_β 原子之间的距离 (其中甘氨酸使用了虚拟的 C_β 原子). 这种方法增加两个残基之间差异的程度, 特别是 β 折叠两边的残基, 从残基的视角来看, 增加了分离的差异.

除了蛋白质高级结构中残基位置完全匹配的情况外, 距离矩阵匹配方法都会遇到在序列中插入和删除残基的问题. 基本的比对方法虽然对于局部比对运算 (即方程 (7.3) 中的 n 很小) 已经足够用了, 但是对于比较的范围 ($-n$ 到 $+n$) 跨越了插入或删除的不连续区域时, 这种方法就会误差很大.

动态规划算法可以解决这个问题, 并产生两个被比较的结构之间的最佳比对. 方程 (8.3) 转换为

$$S_{ik} = \max\left\{a/\left(\left|{}^{\mathrm{A}}d_{ij} - {}^{\mathrm{B}}d_{kl}\right|\right) + b\right\} \tag{8.4}$$

其中 S_{ik} 是蛋白质 A 与蛋白质 B 之间比较的最大得分. S_{ik} 是高层 (残基) 打分矩阵的一个因素, 这个矩阵包含了 S 中蛋白质 (A 和 B) 序列中所有匹配残基之间距离比对的值. 常数 a 的取值一般是 50, 常数 b 的取值不大于 5.

如果仅有最高得分 S_{ik} 进入高层 (残基) 水平矩阵, 包含在底层 (距离) 比对中的信息就会丢失. 这种底层的信息可以提高高层比对的准确性. 为了减小比对的偏差, 在高层水平矩阵相应元素的沿着低级水平矩阵追溯路径上的所有值都会被累加.

3. 向量比较方法

距离比较方法仅依赖于原子间的距离, 距离接近得分就高, 没有考虑原子的相对方向, 所以具有一定的局限性. 为了不丢失原子间方向的信息, 向量比较方法使用原子间的向量进行比较, 而不仅仅使用原子间的距离. 由于这种方法在不同坐标框架下考虑了结构之间不同方向上的位置, 原子间的向量必须在每一个残基的局部参照框架下定义. 这样每一个残基的坐标框架很容易地从 α 碳原子的几何结构中得到. 局部的 X 轴有 N—C 向量定义, 假设的 Y 轴有 C_β– H 向量定义. Z 轴定义为与它们相互垂直的方向, 并且 Y 轴由 X 和 Z 轴的垂直方向再定义, 以确保正交.

比较向量的与方程 (2) 的等价形式为

$$s = a \Big/ \left(\left(\left| {}^{\mathrm{A}}V_{ij} - {}^{\mathrm{B}}V_{kl} \right| \right)^2 + b \right)$$

其中 V 是原子之间的向量. 常数 a 为 50, 向量 b 为 2.

向量比较方法利用了三个距离标示, 一个标示占用一维, 三个标示产生了一个合并的距离方法. 最有用的度量方法是氨基酸残基本身的属性. 这种信息是矩阵中最重要的信息

$$S_{ik} = \max \left\{ (wD_{\mathrm{RiRk}} + a) \Big/ \left(\left(\left| {}^{\mathrm{A}}V_{ij} - {}^{\mathrm{B}}V_{kl} \right| \right)^2 + b \right) \right\}$$

其中 D_{XY} 是由 Dayhoff 矩阵为了交换氨基酸类型 X 和 Y 定义的值, w 是确定序列和结果相对贡献的权重.

8.2.3　SCOP 数据库

SCOP 数据库定义了已知蛋白质详细的、容易理解的结构和进化关系 [153,154]. 几乎所有的蛋白质都与其他蛋白质具有一定程度的结构相似性. 如果两个蛋白质具有非常类似的结构, 它们很可能具有共同的祖先. 这种关系的知识对于分子生物学和其他领域的科学具有重要的贡献. 它是理解蛋白质的进化和结构的核心. SCOP 数据库依据蛋白质的进化关系和三维结构特征进行蛋白质结构分类. 较大蛋白质的结构域根据进化和结构关系等级确认它们在数据库中的位置.

(1) 家族. 满足下面两个标准之一的蛋白质可以收录进一个家族: 第一, 所有蛋白质残基的同源性都要大于 30%; 第二, 低于 30%同源性的蛋白质序列功能和结构必须很相似.

(2) 超家族. 虽然序列同源性比较低, 但是如果结构和功能特征可能具有相同的进化祖先的蛋白质放在同一个超家族.

(3) 共同折叠. 如果家族和超家族中的蛋白质的主要二级结构具有相同组织形式的拓扑链接, 那么它们定义为共同折叠. 相同折叠里的不同蛋白质具有不同大小和形状的二级结构的外围因子和转弯区域. 放在同一个折叠类中的蛋白质的结构相似性.

(4) 类. 类包含了共同折叠. 大多数的折叠都根据它们的二级结构成分属于 5 个类中的一个:

① 螺旋. 这种结构基本由 α 螺旋构成.

② 折叠. 这种结构基本由 β 折叠构成.

③ 螺旋和折叠. 这种结构主要由 α 螺旋和 β 折叠交织在一起.

④ 螺旋加折叠. 这种结构主要由 α 螺旋和 β 折叠相互分离的.

⑤ 多重结构域. 这种结构中不同折叠的结构域目前还没有找到同源结构域.

其他的不常出现的蛋白质、理论模型、核酸和糖类放到其他类里.

8.2.4 SCOP、CATH 和 FSSP 的关系

三种分类方法 SCOP、CATH 和 DALI 对于折叠的归属有很大不同. 很多结构可能在一个系统中定义为一种折叠, 在另外一个系统中定义为一个完全不同的类别. 对于结构域和折叠的不同定义是两个数据库的主要分歧 [153].

DALI 结构域词典全部自动定义和分类结构域 [154], 该方法依据 PUU 算法定义结构域 [147,154]. DALI 算法用于结构域相似性的比较过程中首先使用一个快速算法把二级结构子序列表示成向量, 然后使用一个慢速算法比较残基的中心点位置. FSSP 数据库对于蛋白质结构域的分类是根据 DALI 打分得到的.

SCOP 是最早把蛋白质结构域进行分类的数据库 [151]. SCOP 中的几乎全部分类过程都是依靠有声望的科学家通过人工方法进行的. SCOP 的目的是通过蛋白质序列结构关系建立一种研究蛋白质进化的工具. 在确定结构域、折叠和同源关系时, SCOP 的分类原则侧重于进化关系 [92]. 这个数据库被认为是蛋白质结构分类的标准.

CATH 数据库使用了人工方法和自动方法相互结合来定义和分类蛋白质结构域. CATH 依赖三个自动分类方法的一致性来把蛋白质分解成为结构域. 这种方法有效地定义了蛋白质序列中的大约 53%结构域, 其中没有一致性的序列的结构域通过人工方法定义. 虽然蛋白质序列的同源性主要由一级序列确定, 但是结构相似性很高的蛋白质的距离矩阵之间的匹配性也可以定义蛋白质序列的同源性. 同源性低的折叠类的结构域基于 CORA[11,65] 或 SSAP 算法进行拓扑层的分类.

8.3 蛋白质结构域的支持向量机预测方法

8.3.1 蛋白质结构域预测中的样本集选择

因为样本集构成了学习机和分类器的工作环境, 所以样本集的选择很重要. 根据统计学习理论, 样本集中的元素应该独立同分布, 即样本集中的样本是信息非冗余的, 通过训练非冗余样本的特征向量构造出的 SVM 才具有最好的泛化性. 然而从蛋白质结构域折叠类型预测这个具体问题来看, 目前还没有构造非冗余样本集的统一标准. 考虑到样本集中的元素既要有较大的非同源性又要有足够的数据量, 实验中我们使用 CATH 数据库的拓扑结构层 (T 层)、同源超族 (H 层) 以及序列家族 (S 层) 的结构域作为数据集中的元素.

CATH 中的蛋白质主要分为四个层次: 折叠类型 (class level, C)、结构体系 (architecture level, A)、拓扑结构 (topology level, T) 和同源超族 (homologous superfamily level, H). 在此基础上, 比同源超族更低的层是序列家族 (sequence family, S). 折叠类型 (C) 描述了蛋白质二级结构的成分. 在这个水平上, 结构域被分为四

大类型：以 α 螺旋为主的类 (mainly α)、以 β 折叠为主的类 (mainly β)、α 螺旋和 β 折叠类 (alpha beta, α-β)(其中包括 α/β 和 $\alpha+\beta$ 结构) 以及含少量的二级结构的小型二级结构类 (few secondary structures, fss). 根据 Taylor W. R. 和 Orengo, C A 的定义在 α 螺旋为主的类中的二级结构主要是 α 螺旋, 而 β 折叠在 β 折叠为主的类的二级结构中占有数量优势 [152]. α/β 类结构域中的 α 螺旋和 β 折叠以 β-α-β 为单位存在, 其中的 β 折叠相互平行. 在 α+β 单位中, α 螺旋和 β 折叠在空间上相互分离, 分别处在蛋白质的不同部位. 结构体系 (A) 描述了蛋白质二级结构和独立连接的大致形态. 拓扑结构层 (T) 根据蛋白质二级结构的数量和链接方式来遴选结构域. 在这个水平上结构域中的二级结构具有大致相同的折叠方式, 也就是说它们包含的二级结构数量相同、二级结构的排列方式相同、这些二级结构的链接结构也相同. 拓扑结构虽然具有相似的结构但其功能不同. 结构和功能都相似的结构域共同构成了一个同源超族 (H). 序列家族中的结构域的序列同源性大于 35%. 因为在进化过程中结构的保守性比序列的保守性更强, 所以大于 35%的序列同源性表示了很高的结构同源性和功能的相似性. CATH 数据库可以在线得到 (地址：http://www.biochem.ucl.ac.uk/bsm/cath/index.html).

需要补充的是结构域由超过一个的序列片段构成的情形很普遍. 实验中当出现这种情况时, 我们只在每个片段的内部取样本. 也就是说, 我们把具有多个片段的结构域序列中的每个片段分别进行向量化, 再把这些向量加在一起, 然后进行归一化.

我们分别取 CATH 数据库 2.5.1 版的拓扑层 (topology level, T) 的 820 个样本、2.6.0 版同源超族层 (homology superfamily level, H) 的 1572 个样本以及 2.6.0 版序列家族 (sequence family level, S) 的 α、β 和 α-β 类的 5859 个样本加上 few secondary structures 中的结构域层 (domain level, D) 的 1098 个样本合并成的混合样本集. 最后一个样本集之所以从两个层次来取样本的主要目的在于防止产生非平衡数据集. 从两样本集样本的统计数据来看, 每个样本集中的 α、β 和 α-β 类中元素的长度范围与平均长度差别不大, 而它们与 fss 中元素的长度范围与平均长度则差别较大 (见表 8-2).

8.3.2　编码方法

结构域的氨基酸残基序列转化成为输入空间的向量的过程称作编码. 在本实验中使用同一样本集分别利用一肽频数编码方法、二肽频数编码方法和三肽频数编码方法编码产生三组向量样本, 即一肽频数向量样本集、二肽频数向量样本集和三肽频数向量样本集. 这几种方法都属于多肽频数编码方法. 一肽频数实际上就是每个残基在蛋白质序列中出现的频率, 由一肽频数编码方法产生的向量为 20 维. 这种编码方法仅仅考察序列的氨基酸残基成分. 二肽是氨基酸残基对, 三肽是连

表 8-2 2.5.1 版拓扑层 (T-level)、2.6.0 版同源超族层 (H-level) 和序列家族层 (S-level) 层的各个折叠类中结构域氨基酸残基序列的长度范围和平均长度(单位: 残基)

类别		结构域数量	长度范围	平均长度
拓扑层 *	α 类	227	38~740	155
	β 类	139	33~574	145
	α-β 类	368	36~534	170
	fss	86	16~119	57
同源超族层 **	α 类	465	21~740	123
	β 类	311	22~571	143
	α-β 类	706	30~759	156
	fss	90	16~105	57
序列家族层 ***	α 类	1402	11~872	129
	β 类	1143	19~582	131
	α-β 类	3014	28~759	171
	fss	1098	16~119	66

注: 数据来自互联网, 地址: http://www.biochem.ucl.ac.uk/bsm/cath/releases.html

*: 源自 2.5.1 版本; **: 源自 2.6.0 版本; ***: 样本来自结构域层.

续三个氨基酸残基的组合. 因为二十个氨基酸残基可以构成 400 个可能的二肽组合、8000 个可能的三肽组合, 所以所有可能的二肽可以构成一个 400 维的输入空间、所有可能的三肽可以构成一个 8000 的输入空间. 这两种编码方法不仅可以考察序列的氨基酸成分, 还考察了序列中相邻残基的关系. 二肽频数和三肽频数编码方法在第 7 章中已经进行了详细的介绍.

8.3.3 拓扑预测准确率的评估方法

为了客观地评估预测准确率, 实验结果使用交叉验证试验进行结果准确率的评估. 交叉验证的功能在于缩小由一次随机选择训练和检测样本集引起的预测结果波动. 交叉验证主要分为三种类别: 单一检验集分析、子样本检验和 Jackknife 检测[95]. 由于进行 Jackknife 检测的数据集轮流作为训练集和检测集, 所以每一个样本都可以分别作为检测样本和训练样本. 交叉验证试验的基本方法是首先把包含 n 个样本的样本集平均分成 k 个子集. 然后每个子集分别作为检测集, 而其余 $k-1$ 个子集合并成为训练集. 经过 k 次训练和检测, 得到结果的平均值作为最终预测的准确率. 这样做的目的就是要尽量减小由于随机选择检测集和训练集预测结果的偏差.

如果计算能力允许可以使用 Jackknife 试验进行交叉验证. Jackknife 试验又称为全面交叉验证试验, 这种试验把包含有 n 个样本的样本集分成 n 个子集, 这样每个子集中就仅包含有一个样本. 然后, 每个子集都被由其余 $n-1$ 个子集构成的训练集训练而成的分类器分类. 这样整个样本集中的每一个样本都被预测了一次, 所

以全面的 Jackknife 试验预测准确率是正确分类样本的百分比. Jackknife 检测的缺点在于当样本很小的时候会出现信息丢失和不准确的检测结果. 当出现这种情况时可以使用蒙特卡罗抽样产生模拟的样本.

实验中对于取自 CATH 数据库 2.5.1 版本中拓扑层的 820 个结构域和 2.6.0 版本中同源超族的 1572 个结构域和序列家族中的 6957 个结构域分别构成样本集. 为了能够搜索到最优参数并且使用最短的时间, 在优化参数的过程中使用了 7 重交叉验证试验来获得最优参数, 之后进行 Jackknife 试验检测该参数所能得到的实际预测准确率. 这么做的原因在于使用 7 重交叉验证试验进行参数优化使用的时间仅为使用 Jackknife 试验进行参数优化时间的百分之一到千分之一, 同时使用 Jackknife 试验进行准确率检测又可以得到真实的预测准确率. 使用 7 重交叉验证试验与使用 Jackknife 试验进行参数优化相比, 缺点在于可能得不到最优参数和最高预测准确率.

对于样本进行预测的准确率是由训练和检测样本本身的性质决定的, 跟样本中各类数据提供的信息中是否掺有噪音点直接相关. 从技术上来说, 在既定样本的前提下要想通过支持向量机方法获得最高预测准确率就要对进行分类的超平面进行最优化. 选择最优的参数才能得到最优的分类超平面. 使用 Jackknife 试验进行参数的优化无疑是最理想的, 但是付出的计算成本极为高昂. 下面试验以拓扑层 820 个样本的一肽频数向量为实验材料, 评估与比较在最优参数搜索过程中多重交叉验证试验与全面 Jackknife 试验的优缺点.

实验中选定径向基内核, 红帽子 Linux9.03 系统以及 Libsvm2.4 软件. 硬件系统核心组件为一个 Inter 奔腾 4 主频 2.4G 的 CPU, 512M 的 DDR 内存. 实验包括三个步骤:

(1) 运用 7 重交叉验证试验方法寻找最优参数;

(2) 运用 28 重交叉验证试验方法寻找最优参数;

(3) 运用 Jackknife 试验方法寻找最优参数.

一肽频数实际上就是结构域一级序列中各个残基出现的频率, 在三步实验中使用 7 重交叉试验的预测结果和运算时间作为基准. 为了对多重交叉验证的优化结果有一个客观的评价, 实验中使用了 Jackknife 试验进行最优点的最终多元预测准确率比较. 实验是在 400 对 (C, γ) 参数中寻找最优参数.

首先运用 7 重交叉验证试验方法寻找最优参数. 7 重交叉验证试验的各个参数所确定分类超平面的分类准确率形成的轮廓见图 8-7. 优化所花费的计算时间为 12 分钟. 图 8-8 是计算由图 8-7 给出的最优点的 Jackknife 试验预测准确率的屏幕截图, 结果为 54.7561%.

然后, 运用 28 重交叉验证试验方法寻找最优参数. 使用命令

$python grid.py –log2c 0,2,0.1 –log2g 5,7,0.1 –v 28 文件名

这个命令与前面命令的区别仅在于由 “-v 28” 代替了 “-v 7”, 得到的优化轮廓图见图 8-3.

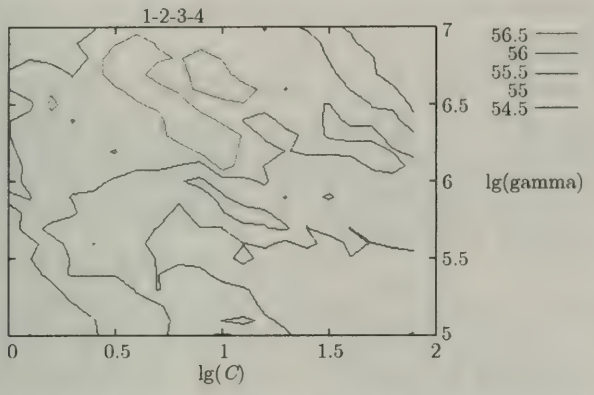


图 8-3 拓扑层的 820 个样本的一肽频数输入样本集的 28 重交叉验证试验优化结果

进行本次优化所花费的时间为 47 分钟, 约为上次优化所花费时间的 4 倍, 这个比值约等于交叉验证重数比的 28/7. 对计算得到的最优点进行 Jackknife 试验得到的预测准确率为 55.3659%. Jackknife 试验结果的屏幕截图见图 8-4.

图 8-4 拓扑层的 820 个样本的一肽频数输入样本集在 C=$2^{1.0}$, γ=$2^{6.6}$ 时, 通过 Jackknife 试验得到的预测准确率

最后, 运用 Jackknife 试验方法寻找最优参数. 使用命令

$python grid.py –log2c 0,2,0.1 –log2g 5,7,0.1 –v 820 文件名

进行 Jackknife 试验优化所花费的时间为 26 小时零 8 分钟, 约为 7 重交叉验证试验的优化所花费时间的 131 倍, 这个值与 820/7 的值大体相当. Jackknife 试验得到的预测准确率为 56.4634%.

比较图 8-7、图 8-3 和图 8-5, 可以发现三张图的预测准确率轮廓线明显不同, 最优区域 (绿色线) 也不一样, 通过最优点的 Jackknife 试验证实图 8-7 给出的最优点的实际分类准确率最低, 图 8-3 给出的最优点的实际分类准确率介于中间, 而图 8-5 所给出最优点的分类准确率最高. 从 7 重交叉验证优化试验到 Jackknife 优化试验, 预测准确率提高了 1.7073%.

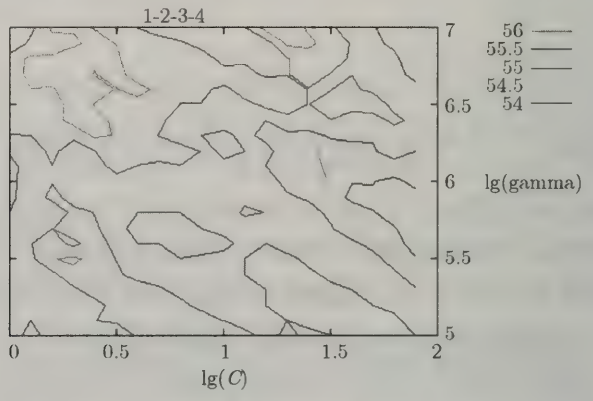


图 8-5　拓扑层的 820 个样本的一肽频数输入样本集的 Jackknife 优化结果

实验表明, 在其他条件不变的情况下, 寻找最优参数所花费的计算时间与所用的交叉验证的重数成正比.

另外, 为了比较输入空间中的样本个数与维数与运算成本的关系, 我们在实验中还记录了其他部分实验所花费的计算时间. 在其他实验条件不变的情况下, 即硬件系统和软件系统不变, 使用 7 重交叉运算, 搜索了 400 个参数 (C, γ). 其中计算 1572 个样本的一肽频数向量的优化参数的时间为 49 分钟, 计算 820 个样本的二肽频数向量的优化参数时间为 112 分钟. 根据以上实验数据以及以往实验的经验可以推断当样本数量为原来的 n 倍时, 花费的计算时间约为原来的 n^2; 当输入空间向量的维数为原来的 20 倍时, 所花费的计算时间约为原来的 10 倍.

从上面的分析可以得出, 当样本数量和输入空间中向量维数增加时, 所花费的计算成本急剧上升. 虽然全面 Jackknife 试验能够得到最优的准确率, 但是当输入向量维数较高、数量较大时计算成本将难以负担, 因此使用 n 重交叉验证试验还是十分必要的, 其中 n 一般小于 10.

8.3.4 分类器设计与软件使用方法

蛋白质折叠类型预测的直接目的在于预测 CATH 数据库中的结构域属于 C 层的何种类别. 由于 CATH 数据库在 C 层结构域分为 α 螺旋、β 折叠、α-β 以及 fss 四种折叠类型, 所以对结构域折叠类型的预测是四元预测. 第 3 章和第 5 章已经介绍过, 支持向量机通过多次二元分类实现多元分类. 实现多元分类的二元分类器包括 "一对一" 分类器和 "一对多" 分类器. Libsvm 通过$\frac{n(n-1)}{2}$ 个 "一对一" 二元分类器利用投票法多元分类. 投票法是在对一个检测样本的预测时, 把每一次二元分类看成是一次投票过程, 检测样本被归为得票最多的类. Libsvm 软件包自动实现多元分类, 这个过程中不用人为干预, 最后总的预测结果是根据六个 "一对一" 的二元分类器的预测结果通过投票法得到的.

然而, 为了详细研究折叠类相互之间的关系, 我们实验中还构建了六个 "一对一" 的二元分类器和四个 "一对多" 的二元分类器. 这里所谓 "一对一" 是指四种折叠类型中两两分别编组分类, "一对多" 是指四种折叠类型中的每一种类型分别跟其他三种的组合进行编组分类. 六个 "一对一" 的分类器包括: α vs. β、α vs. α-β、α vs. fss、β vs. α-β、β vs. fss 和 α-β vs. fss. 四个 "一对多" 的二元分类器包括: α vs. 其他 (包括: β、α-β 和 fss)、β vs. 其他 (包括: α、α-β 和 fss)、α-β vs. 其他 (α、β 和 fss) 和 fss vs. 其他 (包括 α、β 和 α-β).

实验中选择了 RBF 核函数来把输入空间中的向量映射到高维特征空间中

$$k_{\mathrm{rbf}}(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \quad \gamma > 0 \tag{8.5}$$

式中的 γ 为影响高维空间中的特征向量的坐标.

同时选择了 C-SVM 样本分类器

$$\min \ \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{l} \xi_i \tag{8.6}$$

$$\text{s.t.} \ \ y_i \left(w^{\mathrm{T}} \Phi(x_i) + b\right) \geqslant 1 - \xi_i \tag{8.7}$$

$$\xi_i \geqslant 0, \quad i = 1, \cdots, l \tag{8.8}$$

(8.3) 式中的 C 影响由支持向量确定的边界的大小.

在上面选择的条件下, 可以调节的参数为 (8.5) 式中的 γ 和 (8.6) 中的 C.

Libsvm 软件包可以在 linux 操作系统中使用. 其中对 (8.5) 式中的 γ 和 (8.6) 中的 C 进行优化的子程序为 "grid.py". 使用的命令为

$python grid.py 文件名

其中 python 是图形处理的软件名, "grid.py" 是 Libsvm 中进行参数优化的软件名. 这个命令可以通过逐格搜索的方法确定最优的 C 和 γ 的取值. 由于 Libsvm 使

用了特有的对数取值方法, 所以参数 C 和 γ 的实际取值步长的增长序列为：2^{-5}, 2^{-6}, $\cdots$, 2^{5} $\cdots$, 即每次 C 和 γ 实际增加值为原来值的 2 倍. 默认的 C 的取值范围为 2^{C}：2^{-5} 到 2^{15}, $\log_2 C$ 的步长为 1; 默认的 γ 的取值范围为 2^{γ}：2^{-15} 到 2^{5}, $\log_2 \gamma$ 的步长为 1, 交叉验证为 5 重.

Libsvm 中另外一个使用到的子程序为 “svm-train”. 这个子程序用来计算取到最优参数时的预测准确率.

8.3.5　结果与分析

8.3.5.1　拓扑层 820 个结构域样本的分类结果

1. 一肽频数向量四元分类结果

为了比较不同编码方法对预测准确率的影响, 每个结构域层中的结构域一级序列都由一肽频数编码方法、二肽频数编码方法和三肽频数编码方法三种编码方法分别进行编码. 实验中首先进行的是拓扑层样本的折叠类型预测准确率评估.

由于一肽频数编码方法编码得到的向量维数为 20, 拓扑层共有 820 个样本, 那么样本与维数的比为 41. 为了找到最优的分类结果, 就要对参数 C 和 γ 进行优化. 这个优化过程不是一蹴而就的, 要进行耐心地寻找和分析. 首先使用 “grid.py” 程序提供的默认参数对 C 和 γ 进行优化, 得到结果如图 8-6 所示. 默认参数为 “2^{C}=2^{-5}, $\cdots$, 2^{15}; 2^{g}=2^{-15}, $\cdots$, 2^{5}; -v=5”. 其中 “-v” 表示交叉验证的重数. 通过这次优化过程, 确定了一肽频数编码的向量集在这个向量集上目标函数 (8.6) 中的参数为 C=2^{1}, 核函数 (8.3) 中的参数都为 γ=2^{3}(图 8-6). 这时一肽频数输入样本集总的四元预测准确率为 55.7317%.

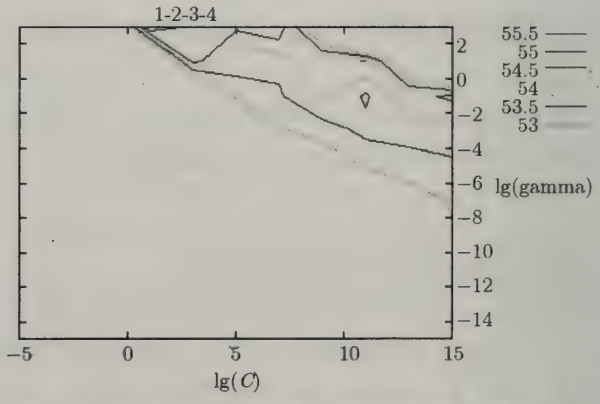


图 8-6　拓扑层的 820 个样本的一肽频数输入样本集的默认参数 C 和 γ 的优化结果

然而, 从图 8-6 中可以看出 (C, γ) 的最优取值区域并不完全包含在参数的默认取值范围之内. 根据常识, 只有可能达到最优准确率的区域全部包含在参数优化

图中的时候才可以断定所得到的预测准确率是最优的, 即准确率最高的区域在参数优化图中已经形成了岛形. 由图 8-6 可以确定: 可能有更好的点存在.

调整命令中参数 C 和 γ 的取值范围到可以覆盖最优区域, 使用命令

$python grid.py -log2c 0,2,0.1 -log2g 5,7,0.1 -v 7 文件名

这个命令中首先对参数 C 的取值范围和步长进行了调整, 其中 "-log2c 0,2,0.1" 表示参数 C 的取值范围从 2^0 到 2^2, 步长为 $2^{0.1}$; 然后对 γ 的取值范围和步长进行了调整, "-log2g 5,7,0.1" 表示参数 γ 的取值范围从 2^5 到 2^7, 步长为 $2^{0.1}$; "-v 7" 表示 7 重交叉验证试验. 图 8-7 为使用 7 重交叉验证试验进行参数调整后, 规定参数范围内各个点的预测准确率轮廓图.

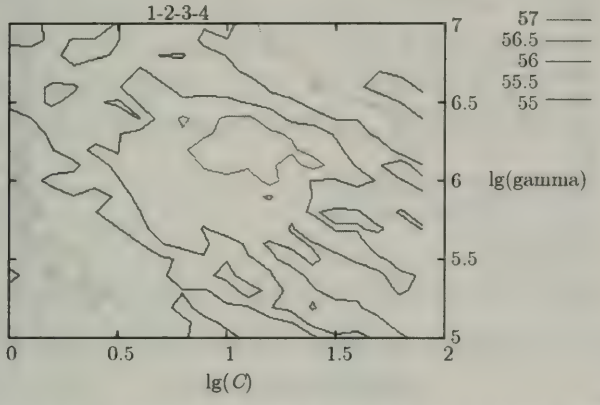


图 8-7 拓扑层的 820 个样本的一肽频数输入样本集的 7 重交叉验证试验优化结果

调整参数后, 通过 7 重交叉验证试验得到的预测准确率为 57.1951%, 参数 C=$2^{1.0}$, γ=$2^{6.2}$. 使用这组参数对样本进行 Jackknife 试验使用命令

svm-train –c 2 –g 73.5166947198 –v 820 文件名

得到结果见图 8-8.

图 8-8 是一张屏幕截图. 截取了 Jackknife 试验的最后一个循环和总的计算结果. 图中显示了六个 "一对一" 的二元分类的结果. Libsvm 通过投票法从这六个二元分类中得到了总的四元分类准确率. 从图 8-8 中可以看出一肽频数向量集在 C=$2^{1.0}$, γ=$2^{6.2}$ 时 Jackknife 预测准确率为 54.7561%, 这个准确率是真实的准确率. 图中显示总的支持向量数为 742, 那么支持向量与样本的比值为 0.904.

2. 一肽频数向量二元分类结果

Libsvm 允许直接进行多元分类, 但是为了深入研究各个折叠类之间的关系, 我们对属于总的预测准确率优化中间步骤的各个折叠类之间的二元分类结果单独进

行了计算. 这些二元分类分别表示了四个折叠类型样本序列之间两两关系的情况. 前面已经介绍过, 二元分类器包括 “一对一” 分类器和 “一对多” 分类器两类. 六个 “一对一” 分类器和四个 “一对多” 分类器参数优化结果以及预测准确率详情见表 8-3.

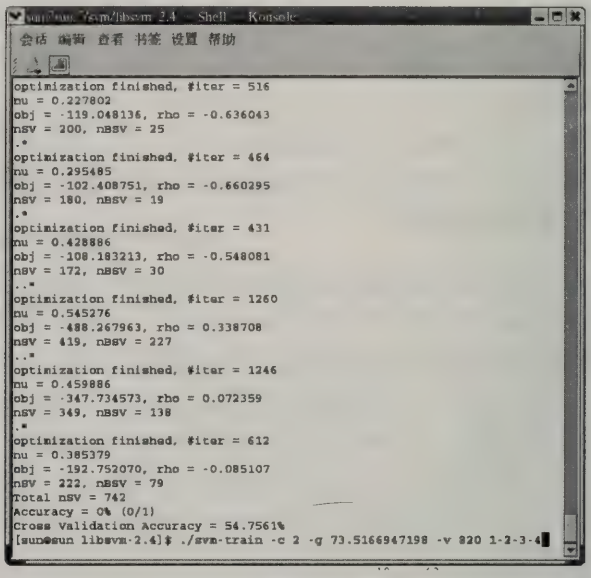


图 8-8　拓扑层的 820 个样本的一肽频数输入样本集在 C=$2^{1.0}$, γ=$2^{6.2}$ 时, 通过 Jackknife 试验得到的预测准确率

表 8-3　拓扑层 820 个一肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	84.3836	82.1918
α vs. α-β	72.9412	72.1008
α vs. fss	84.345	82.4281
β vs. α-β	75.9369	75.1479
β vs. fss	75.5556	71.5556
α-β vs. fss	87.6652	86.7841
α vs. 其他	78.6585	77.6829
β vs. 其他	83.1707	82.8049
α-β vs. 其他	65.4878	64.1463
fss vs. 其他	90.2439	89.878
平均准确率	79.838 84	78.472 04

从表 8-3 中可以观察到通过 7 重交叉验证试验得到的准确率无一例外地高于

通过 Jackknife 试验得到的预测准确率. 后面的实验中也出现了类似的结果, 即 7 重交叉验证试验的预测准确率一般不低于 Jackknife 试验的预测准确率. 出现这种现象的原因在于, 通过 Jackknife 试验得到的预测准确率是参数取某个值时真实的准确率, 而通过 7 重交叉验证试验得到的准确率由于在选择训练集和检测集时具有一定的偶然性, 预测准确率不免偏离真实的准确率, 有时偏高些, 有时偏低些. 参数优化和选择的过程是一个循环计算过程, 计算机程序反复比较取得各种参数时预测准确率并把最高的预测准确率记录下来, 所以通过 7 重交叉验证试验得到的准确率总会高于通过 Jackknife 试验得到的预测准确率.

表 8-4 报告了二元分类器取得最高预测准确率时模型参数的最优值、支持向量数和支持向量数与样本数的比率. 由于使用的是具有一定偏差的 7 重交叉验证优化参数, 同时参数取值又不是连续的, 所以在对某个二元分类器的优化过程中往往得到多个最优参数. 为了计算方便我们取得计算机记录下来的参数值, 即第一个值. 这样做可能会带来一些问题, 比如不能得到真正的最优值. 之所以这么做首先因为得到的预测准确率与真实最优值相差不大, 另外也考虑了计算成本.

表 8-4 拓扑层 820 个一肽频数样本二元分类最优参数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量数与样本数的比率
	C	γ		
α vs. β	$2^{7.3}$	$2^{0.5}$	143	0.391
α vs. α-β	$2^{1.0}$	$2^{5.2}$	400	0.672
α vs. fss	$2^{8.5}$	$2^{-0.5}$	124	0.396
β vs. α-β	$2^{2.0}$	$2^{5.9}$	345	0.683
β vs. fss	$2^{5.5}$	$2^{-1.5}$	156	0.693
α-β vs. fss	$2^{5.9}$	$2^{1.0}$	133	0.293
α vs. 其他	$2^{7.3}$	$2^{-1.3}$	424	0.517
β vs. 其他	$2^{5.2}$	$2^{1.8}$	328	0.4
α-β vs. 其他	$2^{7.0}$	$2^{-1.0}$	649	0.791
fss vs. 其他	$2^{5.3}$	$2^{1.0}$	191	0.233

另外需要说明的是 Libsvm 使用 "一对一" 的投票法以二元分类为基础进行多元分类. 在 8.3.5.1 节的第一部分中给出的四元分类与 8.3.5.1 节的第二部分中的二元分类是单独进行的. 8.3.5.1 节的第一部分中构成四元分类的各个二元分类器把四种折叠类型的样本放在一起作为一个整体进行优化, 共用一个参数. 而 8.3.5.1 节的第二部分中的二元分类器的参数是单独优化的, 它们根据各个二元样本集的性质单独进行优化, 分别使用各自的参数.

3. 二肽频数向量四元分类结果

二肽频数编码方法编码得到的向量维数为 400, 拓扑层共有 820 个样本, 样本与维数的比为 2.05. 这个比值大大小于一肽频数编码方法编码的向量的样本与维

数的比值.

经过第一轮优化之后, 确定了在这个向量集上目标函数 (8.6) 式中的参数为 C=2^3, 核函数 (8.5) 式中的参数都为 γ=2^3. 这时二肽频数输入样本集总的四元预测准确率为 52.2589%. 调整命令中参数 C 和 γ 的取值范围到可以覆盖最优区域, 使用命令

$python grid.py -log2c 1,3,0.1 -log2g 3,5,0.1 -v 7 文件名

参数 C 的取值范围从 2^1 到 2^3, 步长为 $2^{0.1}$, γ 的取值范围从 2^3 到 2^5, 步长为 $2^{0.1}$, 使用了 7 重交叉验证试验. 图 8-9 为使用 7 重交叉验证试验进行参数调整后, 规定参数范围内各个点的预测准确率轮廓图.

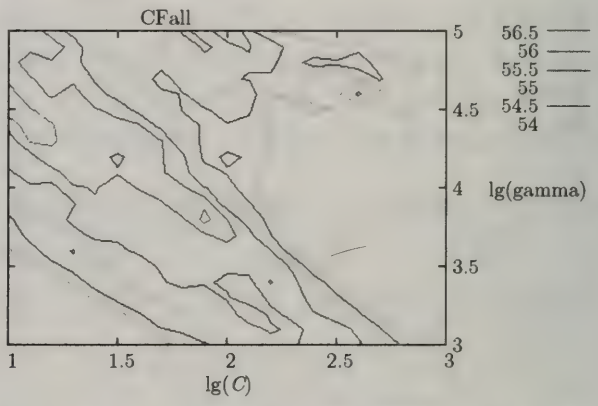


图 8-9　拓扑层的 820 个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

通过 7 重交叉验证试验进行参数优化得到 ($C = 2^{2.0}$, $\gamma = 2^{4.5}$) 和 ($C = 2^{1.1}$, γ= $2^{4.4}$) 两个点可以得到的最优预测准确率都为 56.7766%. Jackknife 试验证明当 C=$2^{2.0}$, γ=$2^{4.5}$ 时, 四元预测准确率达到最高 54.7009%. 图 8-10 显示了最后一个循环的预测结果. 图中最后一行显示了根据二元分类的投票结果得到最后一个样本的预测结果. 所进行的分类中一共得到了约 767 个支持向量, 支持向量占总样本数的 93.5%.

4. 二肽频数向量二元分类结果

二肽频数向量二元分类器参数优化结果以及预测准确率详情见表 8-5. 一肽频数向量的四元预测准确率与二肽频数向量的四元预测准确率之间的差别不大, 相差约为 0.5%, 一肽频数向量的预测准确率略高. 10 个二元分类器的平均 Jackknife 试验预测准确率也相差不大, 二肽频数向量的预测准确率略高些, 约为 1%. 一肽频数向量预测结果中的支持向量与样本数量的比率为 0.5069, 二肽频数向量预测结果中

的支持向量与样本数量的比率为 0.6577. 显然一肽频数向量预测结果中的支持向量与样本数量的比率比二肽频数向量预测结果中的支持向量与样本数量的比率高得多, 高了约为 15%. 表 8-6 报告了二元分类器取得最高预测准确率时模型参数的最优值、支持向量数和支持向量数与样本数的比率.

图 8-10 拓扑层的 820 个样本的二肽频数输入样本集在 C=$2^{2.0}$, γ=$2^{4.5}$ 时, 通过 Jackknife 试验得到的预测准确率

表 8-5 拓扑层 820 个二肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
αvs. β	84.9727	83.3333
αvs. α-β	71.8855	71.2121
αvs. fss	86.2179	85.5769
βvs. α-β	73.3728	71.7949
βvs. fss	78.7517	77.2321
α-βvs. fss	89.6476	88.7665
αvs. 其他	77.7778	77.4115
βvs. 其他	83.1502	83.1502
α-βvs. 其他	66.2169	64.878
fss vs. 其他	90.4672	90.1099
平均准确率	80.246 03	79.346 54

表 8-6　拓扑层 820 个二肽频数样本二元分类最优参数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量数与样本数的比率
	C	γ		
αvs. β	$2^{20.2}$	$2^{-15.4}$	247	0.675
αvs. α-β	$2^{0.2}$	2^{5}	475	0.798
αvs. fss	$2^{0.2}$	$2^{5.5}$	192	0.613
βvs. α-β	2^{5}	$2^{0.1}$	325	0.641
βvs. fss	$2^{0.5}$	$2^{6.3}$	179	0.796
α-βvs. fss	$2^{1.3}$	$2^{5.6}$	203	0.447
αvs. 其他	$2^{0.6}$	$2^{6.6}$	669	0.816
βvs. 其他	2^{5}	2^{-7}	362	0.441
α-βvs. 其他	2^{0}	$2^{7.0}$	801	0.977
fss vs. 其他	$2^{0.3}$	$2^{6.1}$	306	0.373

5. 三肽频数向量四元分类结果

通过三肽频数编码方法得到的向量的维数为 20^3=8000 维. 这样, 每个样本的大小就是通过二肽频数编码方法得到的向量的 20 倍, 计算成本随着样本大小的增加而增加. 这时向量数量与向量维数的比值约为 0.1, 这个比值说明样本数量很小. 经过首次优化确定了参数 C=2^5, γ=2^3 时得到的预测结果最优, 这时三肽频数输入样本集总的四元预测准确率为 52.0147%. 修改参数进行再次优化, 使用命令为

$python grid.py -log2c 0.5,4,0.1 -log2g 5.5,6.5,0.1 -v 7 文件名

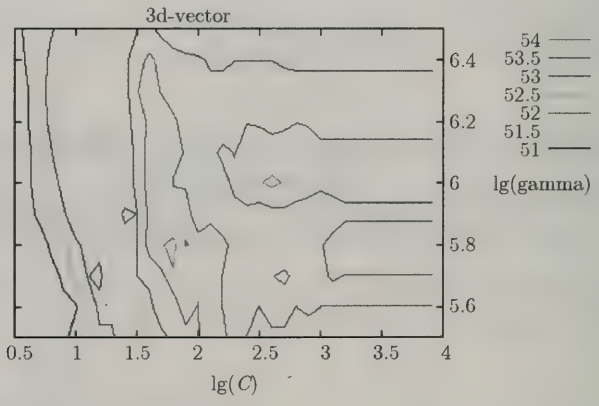


图 8-11　拓扑层的 820 个样本的三肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

得到的参数优化图见图 8-11. 经过参数调整后得到的预测准确率有所提高, 达到了 54.0904%. 能够达到这个预测准确率的参数为 C=$2^{2.6}$, g=$2^{6.0}$ 和 C=$2^{1.8}$, g=$2^{5.8}$. 当 C=$2^{2.6}$, g=$2^{6.0}$ 时 Jackknife 试验的预测准确率为 53.3578%, 详见图

8-12. 支持向量约 817 个, 占总向量数的 99.6%. 支持向量数与总样本数的比值极高, 几乎每个样本都成了支持向量. 显而易见这么高的支持向量与总样本数是不合理的, 再考虑到样本数量与向量维数的比值, 增加样本数量会提高预测准确率.

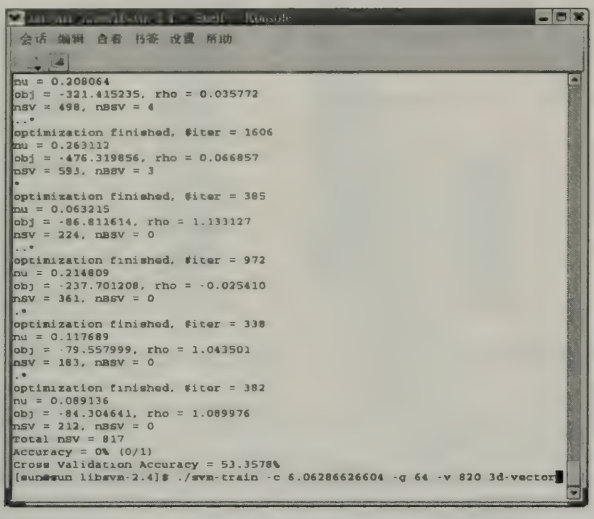


图 8-12 拓扑层的 820 个样本的三肽频数输入样本集的通过 Jackknife 试验得到的预测准确率

这个预测结果与二肽频数编码方法得到的样本的预测结果相比较, 预测准确率有所降低. 通过 Jackknife 试验得到的预测准确率降低了 54.7009%−53.3578%= 1.3431%, 而支持向量数和样本数的比率有所升高, 升高了 99.6%−93.5%=6.1%.

6. 三肽频数向量二元分类结果

拓扑层 820 个三肽频数编码方法编码的样本的二元分类器的 7 重交叉验证预测准确率和 Jackknife 试验预测准确率显示在表 8-7 中. 最优参数、支持向量数以及支持向量与样本数的比率的详细信息显示在表 8-8 中.

观察表 8-4 到表 8-8 四个表, 可以看出支持向量数与样本数的比率和 Jackknife 试验预测准确率之间存在反向变化趋势. 支持向量数与样本数比率高的二元分类准确率相应较低, 反之亦然. 在一肽频数向量折叠类分类中, 支持向量数与样本数的比率最高的三个二元分类器为 α-β vs. 其他 (0.791)、β vs. fss(0.693) 以及 β vs. α-β(0.683). 与它们相对应的 Jackknife 试验分类准确率为 64.1463%、71.5556%和 75.1479%, 这三个预测准确率在几个二元分类器中的准确率分别在倒数第一、第二和第四位. 支持向量数与样本数的比率最低的三个二元分类器为 fss vs. 其他 (0.233)、α-β vs. fss(0.293) 和 α vs. fss(0.396). 与之相对应的 Jackknife 试验分类准确率为 89.878%、86.7841%和 82.4281%, 这三个预测准确率在几个二元分类器中的准确率分别在第一、第二和第四位. 在二肽频数向量分类和三肽频数向量分类中也

出现了类似的情况.

表 8-7　拓扑层 820 个三肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	80.5479	80.274
α vs. α-β	68.6869	68.1818
α vs. fss	84.6154	83.9744
β vs. α-β	72.9249	72.332
β vs. fss	80.3571	79.9107
α-β vs. fss	89.8455	89.404
α vs. 其他	74.4811	74.359
β vs. 其他	83.0281	83.0281
α-β vs. 其他	63.7363	63.7363
fss vs. 其他	90.5983	90.3541
平均准确率	78.882 15	78.555 44

表 8-8　拓扑层 820 个三肽频数样本的最优参数、支持向量数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量数与样本数的比率
	C	γ		
α vs. β	$2^{10.5}$	$2^{2.5}$	338	0.923
α vs. α-β	$2^{6.5}$	$2^{0.2}$	576	0.968
α vs. fss	2^{5}	2^{5}	215	0.709
β vs. α-β	$2^{3.5}$	$2^{4.4}$	481	0.949
β vs. fss	$2^{-0.5}$	$2^{7.0}$	180	0.8
α-β vs. fss	$2^{5.0}$	$2^{6.0}$	225	0.495
α vs. 其他	2^{3}	2^{5}	774	0.944
β vs. 其他	2^{5}	2^{-4}	588	0.717
α-β vs. 其他	2^{-1}	$2^{7.0}$	786	0.959
fss vs. 其他	2^{3}	2^{3}	375	0.457

7. 拓扑层不同维数向量预测结果的比较

相同样本集的不同维数向量的预测结果比较的目的在于考察不同编码方法对知识的挖掘能力. 从上面的预测的结果来看, 一肽频数向量的四元预测准确率和二元预测准确率与二肽频数向量的预测结果大体相当, 而二肽频数向量的无论是四元预测准确率还是二元预测准确率要高于三肽频数向量相对应的预测准确率.

图 8-13 显示了四元分类的 Jackknife 试验预测准确率, 可以看出输入向量的维数越高预测准确率越低. 出现这种情况的原因在于样本数与输入向量的维数的比是影响到预测准确率的一个重要原因. 因此得出结论: 当样本数量较小时, 尽量使用维数低的向量.

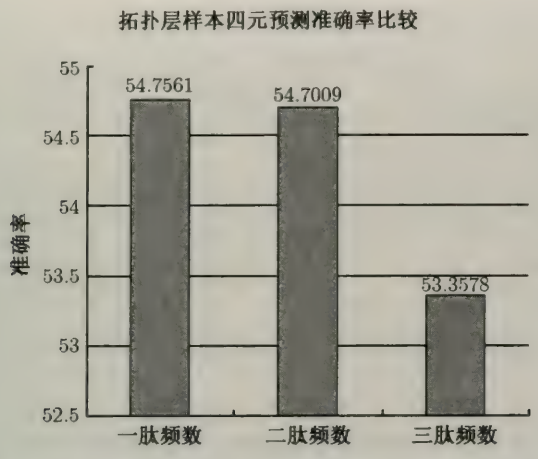


图 8-13 拓扑层的代表个样本的一肽、二肽、三肽频数向量预测准确率比较图

下一阶段的实验中取了 CATH 数据库中的同源超族层 (homology superfamily level) 的结构域作为实验样本, 其总数为 1572 个.

8.3.5.2 同源超族层 1572 个结构域的预测结果

1. 一肽频数向量四元分类结果

在 CATH 数据库的 2.6.0 版本中的同源超族层有 1572 个样本, 这些样本是该层的所有代表样本. 2.6.0 与 2.5.1 的相同层的样本不尽相同. 同源超族层的结构域比拓扑层的结构域的同源性要高些 (本章 8.2.2 节). 对于同源超族的 1572 个样本的处理在程序上跟对于拓扑层 820 个样本的处理一样. 首先把同源超族的样本通过多肽频数编码方法嵌入到输入空间, 然后再使用 Libsvm 软件包进行结构类的预测.

首先进行四元预测. 使用命令

$$\text{\$python grid.py –log2c 6,8,0.1 –log2g 0,2,0.1 –v 7 文件名}$$

得到的参数 C 与 γ 的优化图 (图 8-14). 使用命令

$$\text{\$svm-train –c 128 –g 1.74110112659 –v 7 文件名}$$

得到 C=2^7, γ=$2^{0.8}$ 时的 Jackknife 试验预测准确率的屏幕截图 (图 8-15).

同源超族层的代表样本比拓扑层的代表样本的数量增加了接近一倍, 同源性也有所增加. 从四元预测准确率来看一肽频数向量的预测准确率大幅增加, 增加了 61.9593%−54.7561%=7.2032%. 总的支持向量数为 1241, 占样本数的 78.9%.

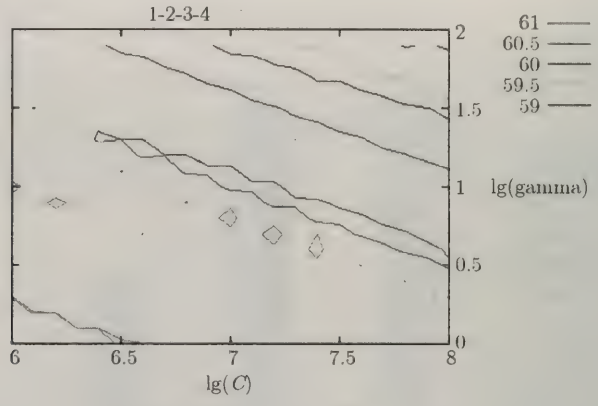


图 8-14　同源超族层的 1572 个样本的一肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

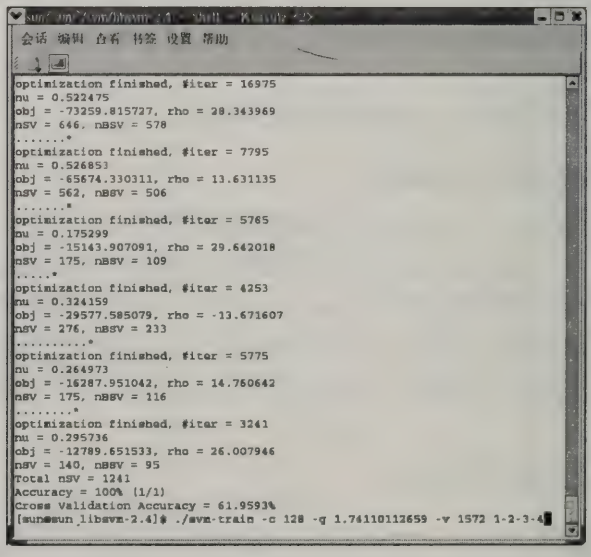


图 8-15　同源超族层的 1572 个样本的一肽频数输入样本集在 C=$2^{7.0}$, γ=$2^{0.8}$ 时, 通过 Jackknife 试验得到的预测准确率

2. 一肽频数向量二元分类结果

同源超族的 1572 个一肽频数编码方法编码的样本的二元分类器的 7 重交叉验证预测准确率和 Jackknife 试验预测准确率显示在表 8-9 中. 最优参数、支持向量数以及支持向量与样本数的比率的详细信息显示在表 8-10 中.

表 8-9 同源超族层 1572 个一肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	86.4691	84.7938
α vs. α-β	77.199	77.3698
α vs. fss	87.3874	86.1261
β vs. α-β	75.9095	75.4149
β vs. fss	86.5337	86.0349
α-β vs. fss	92.0854	92.0854
α vs. 其他	80.1527	80.2799
β vs. 其他	83.7786	83.3969
α-β vs. 其他	68.9567	67.6209
fss vs. 其他	94.6565	94.402
平均准确率	83.312 86	82.752 46

表 8-10 同源超族层 1572 个一肽频数样本二元分类最优参数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量数与样本数的比率
	C	γ		
α vs. β	$2^{7.6}$	$2^{0.8}$	267	0.344
α vs. α-β	$2^{2.5}$	$2^{3.2}$	400	0.568
α vs. fss	$2^{2.1}$	$2^{3.8}$	124	0.326
β vs. α-β	$2^{0.3}$	$2^{3.8}$	345	0.570
β vs. fss	$2^{6.6}$	$2^{-0.9}$	157	0.392
α-β vs. fss	$2^{3.0}$	$2^{3.0}$	177	0.222
α vs. 其他	$2^{3.1}$	$2^{2.4}$	424	0.480
β vs. 其他	$2^{2.0}$	$2^{4.7}$	328	0.426
α-β vs. 其他	$2^{2.9}$	$2^{2.3}$	649	0.764
fss vs. 其他	$2^{4.0}$	$2^{3.0}$	255	0.162

3. 二肽频数向量四元分类结果

二肽频数编码方法编码得到的向量维数为 400, 同源超族层共有 1572 个样本, 那么样本与维数的比为 3.93. 这个比值仍然大大小于一肽频数编码方法编码的向量的样本与维数的比值 (表 8-11).

经过第几轮优化之后, 确定了在这个向量集上目标函数 (8.6) 式中的参数为 C=$2^{-0.1}$, 核函数 (8.5) 式中的参数都为 γ=$2^{5.0}$. 这时二肽频数输入样本集总的四元预测准确率为 59.9237%. 调整命令中参数 C 和 γ 的取值范围到可以覆盖最优区域, 使用命令

$python grid.py –log2c -0.3,0.2,0.05 –log2g -4,6,0.05 –v 7 文件名

表 8-11　同源超族 1572 个二肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	85.9536	85.8247
α vs. α-β	74.5517	74.7225
α vs. fss	86.3063	85.9459
β vs. α-β	76.0079	75.5162
β vs. fss	85.7855	85.2868
α-β vs. fss	92.8392	91.9598
α vs. 其他	78.3079	77.6718
β vs. 其他	84.0967	83.5242
α-β vs. 其他	65.9033	65.7761
fss vs. 其他	94.5929	94.3384
平均准确率	82.4345	82.056 64

参数 C 的取值范围从 2^1 到 2^3, 步长为 $2^{0.1}$, γ 的取值范围从 2^3 到 2^5, 步长为 $2^{0.1}$, 使用了 7 重交叉验证试验. 图 8-16 为使用 7 重交叉验证试验进行参数调整后, 规定参数范围内各个点的预测准确率轮廓图.

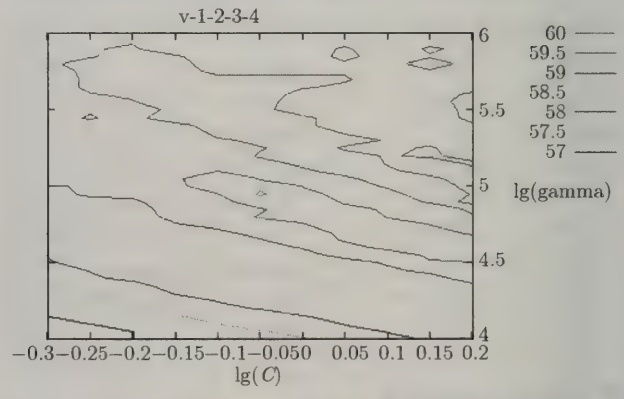


图 8-16　同源超族层的 1572 个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

通过 7 重交叉验证试验进行参数优化得到在点 (C=$2^{-0.05}$, γ=$2^{4.95}$) 的最优预测准确率为 60.0509%. 为了确定最优准确率分别对上面三个点进行了 Jackknife 试验, 使用命令

svm-train –c 0.965936328925 –g 30.9099625256 –v 1572 文件名

实验证明当 C=$2^{-0.05}$, γ=$2^{4.95}$ 时四元预测准确率达到最高, 为 59.7328%. 屏幕截图见图 8-17. 图 8-17 显示了最后一个循环的预测结果. 所进行的分类中一共

得到了约 1421 个支持向量, 支持向量占总样本数的 90.3%. 与拓扑层的 820 个样本由二肽频数编码方法得到的向量预测结果相比较, 支持向量占总样本数的比率下降了 94.0%−90.3%=3.7%, 预测准确率上升了 59.7328%−54.7009%=5.0318%.

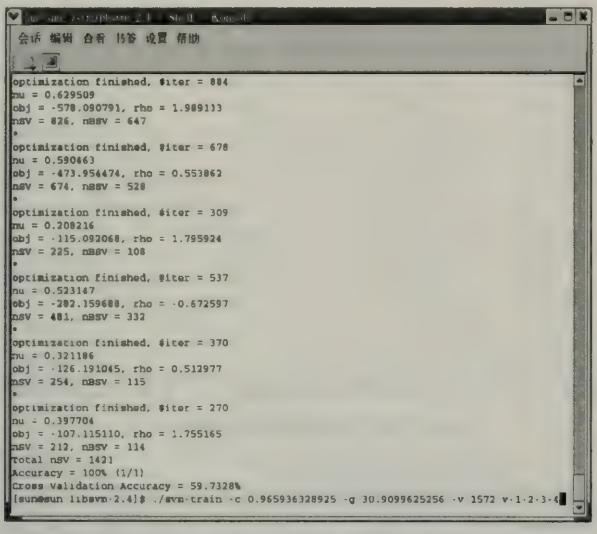


图 8-17 同源超族层的 1572 个样本的二肽频数输入样本集在 $C=2^{-0.05}$, $\gamma=2^{4.95}$ 时, 通过 Jackknife 试验得到的预测准确率

4. 二肽频数样本二元分类结果

同源超族 1572 个样本由二肽频数编码方法编码的向量集的二元分类器的 7 重交叉验证试验分类准确率与 Jackknife 试验分类准确率显示在表 8-12 中. 二元分类器的优化相应参数、支持向量数以及支持向量数与样本数的比率的详细信息显示在表 8-13 中.

CATH 数据库中同源超族层的 1572 个代表结构域所构成的样本比拓扑层 820 个代表所构成的样本数量扩大了将近一倍. 从预测结果来看预测准确率有所提高, 支持向量与样本的比率有所下降.

5. 三肽频数样本四元分类结果

同源超族的 1572 个样本嵌入到输入空间后得到的向量文件非常大, 所有文件容量的和超过了 2.3G. 大数据量的运算十分困难, 仅仅对于三肽频数输入样本集的四元分类的参数优化在使用了 4 个 CPU 的 SGI 图形工作站上就花去了一个多月的时间.

经过几次优化确定了参数 $C=2^3$, $\gamma=2^3$ 时得到的预测结果最优, 这时三肽频数输入样本集总的四元预测准确率为 56.6158%. 修改参数进行再次优化, 使用命令为

表 8-12　同源超族 1572 个二肽频数样本的最优参数、支持向量数以及支持向量数和样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量与样本的比率
	C	γ		
α vs. β	$2^{-0.1}$	$2^{6.0}$	528	0.680
α vs. α-β	$2^{-0.1}$	$2^{5.4}$	829	0.708
α vs. fss	$2^{3.7}$	$2^{1.3}$	223	0.402
β vs. α-β	$2^{1.3}$	$2^{6.0}$	725	0.713
β vs. fss	$2^{0.5}$	$2^{6.3}$	247	0.616
α-β vs. fss	$2^{0.1}$	$2^{5.8}$	276	0.359
α vs. 其他	$2^{-15.4}$	$2^{10.2}$	840	0.534
β vs. 其他	$2^{1.2}$	$2^{6.8}$	1096	0.697
α-β vs. 其他	$2^{0.2}$	$2^{7.3}$	1512	0.962
fss vs. 其他	$2^{1.0}$	$2^{5.0}$	369	0.235

表 8-13　同源超族 1572 个三肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	85.4194	86.1935
α vs. α-β	73.1045	72.7583
α vs. fss	85.5597	85.1986
β vs. α-β	74.7296	73.2547
β vs. fss	85.5362	84.5387
α-β vs. fss	92.5879	92.5879
α vs. 其他	76.145	75.1908
β vs. 其他	82.2519	81.8702
α-β vs. 其他	64.9491	64.5647
fss vs. 其他	94.5293	94.4656
平均准确率	81.481 26	81.0623

```
$python grid.py -log2c 1,3,0.1 -log2g 4,6,0.1 -v 7 文件名
```

经过参数调整后得到的预测准确率有所提高, 达到了 57.1883%. 见图 8-18. 能够达到这个预测准确率的参数为 C=$2^{2.0}$, g=$2^{5.3}$, 此时时 Jackknife 试验的预测准确率为 56.4885%, 屏幕截图见图 8-19. 支持向量数约 1489 个, 占总向量数的 94.7%. 支持向量数与总样本数的比值仍然很高. 这个预测结果与二肽频数编码方法编码的样本的预测结果相比通过 Jackknife 试验得到的预测准确率降低了 59.7328%−57.1884%=2.5444%.

6. 三肽频数样本二元分类结果

同源超族 1572 个样本的三肽频数编码方法编码的向量的二元分类器的 7 重交

叉验证试验分类准确率与 Jackknife 试验分类准确率比较的详细信息显示在表 8-13 中. 最优参数、支持向量数以及支持向量数与样本数的比率显示在表 8-14 中.

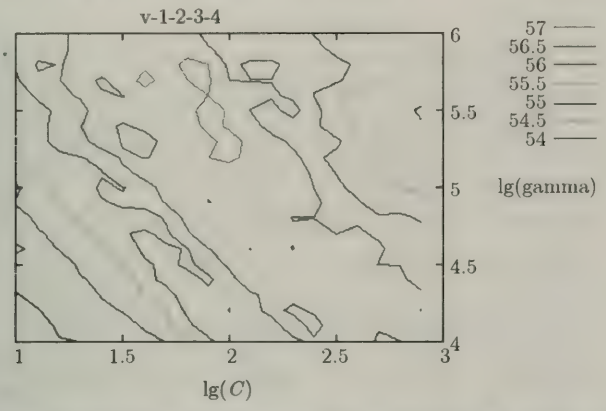


图 8-18 同源超族层的 1572 个样本的三肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

图 8-19 同源超族层的 1572 个样本的三肽频数输入样本集在 C=$2^{2.0}$, γ=$2^{5.3}$ 时, 通过 Jackknife 试验得到的预测准确率

7. 同源超族层不同维数向量预测结果的比较

上面几节给出了同源超族层 1572 个样本依次依据一肽频数编码方法、二肽频数编码方法和三肽频数编码方法把氨基酸残基序列映射到不同向量空间后的预测

准确率评估, 见图 8-20.

表 8-14　同源超族 1572 个三肽频数样本的优化参数、支持向量数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量与样本的比率
	C	γ		
α vs. β	$2^{4.1}$	$2^{2.1}$	659	0.849
α vs. α-β	$2^{1.5}$	$2^{5.5}$	1105	0.944
α vs. fss	$2^{5.2}$	$2^{0.6}$	413	0.744
β vs. α-β	$2^{2.7}$	$2^{4.8}$	910	0.850
β vs. fss	$2^{1.5}$	$2^{4.8}$	277	0.691
α-β vs. fss	$2^{2.0}$	$2^{6.4}$	275	0.345
α vs. 其他	$2^{4.4}$	$2^{3.4}$	1232	0.784
β vs. 其他	$2^{7.0}$	$2^{5.0}$	1089	0.693
α-β vs. 其他	2^{0}	$2^{7.0}$	1457	0.927
fss vs. 其他	$2^{2.9}$	$2^{2.7}$	547	0.348

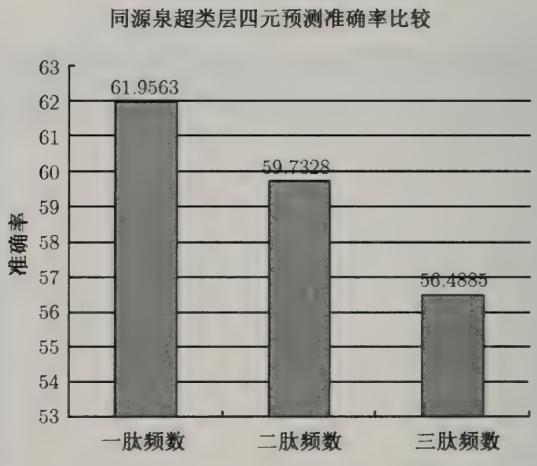


图 8-20　同源超族层的代表个样本的一肽、二肽、三肽频数向量预测准确率比较图

图 8-20 显示出的准确率的趋势与图 8-13 所显示出的趋势是一样的, 输入向量的维数越高预测准确率越低. 与图 8-13 不同的是, 一肽频数样本的预测准确率与二肽和三肽频数向量的预测准确率的差距拉大了, 它们的差距达到了 1.8635%和 5.4678%.

下一阶段的实验中取了 CATH 数据库中的序列家族层 (sequence family level) 层的结构域作为实验样本, 其总数为 6957 个.

8.3.5.3 序列家族层 6957 个结构域的预测结果

1. 一肽频数向量四元分类结果

在 CATH 数据库的 2.6.0 版本中的序列家族层共有 6003 个样本, 其中 α 类包含了 1402 个样本, β 类包含了 1443 个样本, α-β 类包含 3014 个样本, few secondary structures 类包含了 144 个样本. 可以看出 few secondary structures 类别的样本数量过少, 不到 α-β 类样本数量的 1/20. 对于这种不平衡数据集直接进行分类是不合理的, 因为在 α-β 类对 few secondary structures 类的二元分类中即便把 few secondary structures 类的所有样本全都错划归 α-β 类分类准确率仍然超过 95%, 这种结论没有意义. 在样本数既定的情况下, 处理这种不平衡数据集的方法一般有两类: ① 在大样本集中随机抽取一定的样本, 数量约与小样本集的样本数量相同, 组成新的数据集; ② 重复使用小样本集中的样本. 我们在折叠类型预测中处理非平衡数据集的方法与前面提到的方法有所不同. 由于 CATH 数据库的结构域数量大大多于我们使用的结构域数量, 因此在选择样本时还有余地. 因此实验中选择的全部的 few secondary structure 类的结构域共 1098 个作为 few secondary structure 类的样本. 那么该类与 α 类、β 类和 α-β 类的所用样本构成了一个 6957 个样本的样本集, 这样不平衡样本集的问题就解决了. 这 6947 个样本不在是纯粹的序列家族的代表结构域, 但是为了方便起见还是称作序列家族的样本.

对于序列家族的一肽频数样本进行了几轮优化以后得到大致的最优参数取值范围, 进行最终的参数优化, 使用命令

$python grid.py –log2c 6,8,0.1 –log2g 0,2,0.1 –v 7 文件名

得到的参数 C 与 γ 的优化图 (图 8-21).

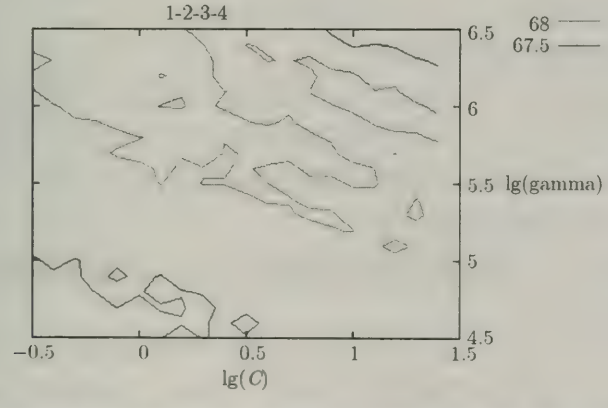


图 8-21 序列家族层的 6957 个样本的一肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

使用最优参数进行 Jackknife 试验计算真实最优预测准确率, 使用命令

$svm-train –c 0.933032991537 –g 78.7932424541 –v 7 文件名

得到 Jackknife 试验预测准确率的屏幕截图 (图 8-22).

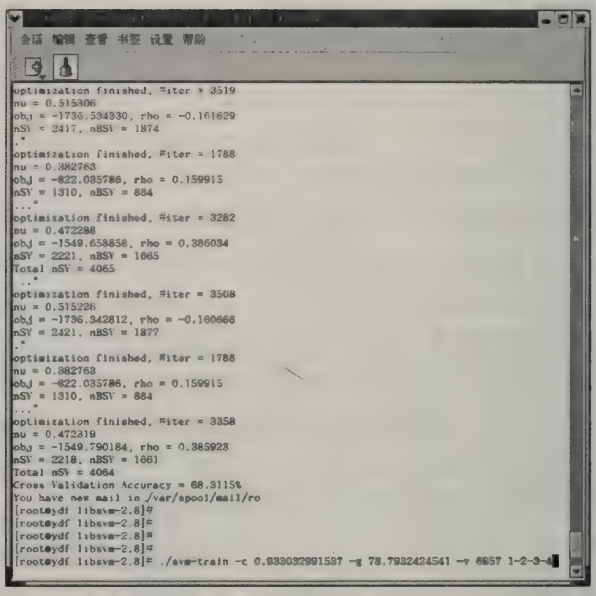


图 8-22　序列家族层的 6957 个样本的一肽频数输入样本集在 C=$2^{-0.1}$, γ=$2^{6.3}$ 时, 通过 Jackknife 试验得到的预测准确率

总的支持向量数为 4064, 占总样本数的 58.4%.

2. 一肽频数向量二元分类结果

序列家族层的 6957 个一肽频数编码方法编码样本的二元分类器的 7 重交叉验证预测准确率和 Jackknife 试验预测准确率显示在表 8-15 中. 最优参数、支持向量数以及支持向量与样本数的比率的详细信息显示在表 8-16 中.

3. 二肽频数向量四元分类结果

二肽频数编码方法编码得到的向量维数为 400, 同源超族层共有 6957 个样本, 那么样本与维数的比为 17.39.

经过第几轮优化之后, 确定了在这个向量集上目标函数 (8.6) 式中的参数为 C=$2^{1.0}$, 核函数 (8.5) 式中的参数都为 γ=$2^{7.0}$. 这时二肽频数输入样本集总的四元预测准确率为 74.2849%. 调整命令中参数 C 和 γ 的取值范围到可以覆盖最优区域, 使用命令

表 8-15 序列家族层 6957 个一肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	86.4991	84.6749
α vs. α-β	80.865	80.865
α vs. fss	97.72	97.72
β vs. α-β	77.4736	77.6043
β vs. fss	97.442	96.3007
α-β vs. fss	98.2977	98.4922
α vs. 其他	86.0716	86.5747
β vs. 其他	84.3754	84.6917
α-β vs. 其他	76.1248	75.9667
fss vs. 其他	98.9363	99.31
平均准确率	88.380 55	88.220 02

表 8-16 序列家族层 6957 个一肽频数样本二元分类最优参数以及支持向量数与样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量数与样本数的比率
	C	γ		
α vs. β	$2^{2.0}$	$2^{5.0}$	1106	0.389
α vs. α-β	$2^{5.0}$	$2^{4.0}$	2084	0.472
α vs. fss	$2^{5.0}$	$2^{7.0}$	837	0.335
β vs. α-β	$2^{5.0}$	2^{0}	2368	0.531
β vs. fss	$2^{6.0}$	$2^{7.0}$	465	0.183
α-β vs. fss	$2^{5.0}$	$2^{7.0}$	895	0.218
α vs. 其他	$2^{2.0}$	$2^{6.0}$	2711	0.390
β vs. 其他	$2^{5.0}$	$2^{4.0}$	2575	0.370
α-β vs. 其他	$2^{2.0}$	$2^{6.0}$	3994	0.574
fss vs. 其他	$2^{5.0}$	$2^{7.0}$	1481	0.213

$python grid.py –log2c -1,3,1 –log2g 6,9,1 –v 7 文件名

图 8-23 为使用 7 重交叉验证试验进行参数调整后, 规定参数范围内各个点的预测准确率轮廓图

为了确定最优准确率分别对上面三个点进行了 Jackknife 试验. 使用命令

svm-train –c 0.965936328925 –g 30.9099625256 –v 1572 文件名

实验证明, 当 C=$2^{1.0}$, γ=$2^{7.0}$ 时, 四元预测准确率达到最高, 为 74.5436%. 屏幕截图见图 8-24. 最后一轮循环一共得到了约 5538 个支持向量, 支持向量占总样本数的 79.6%.

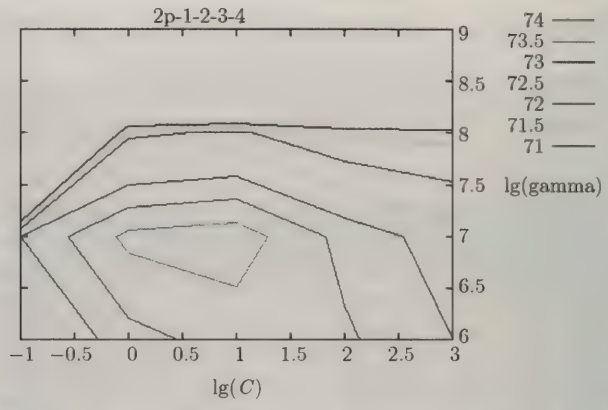


图 8-23　序列家族层的 6957 个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

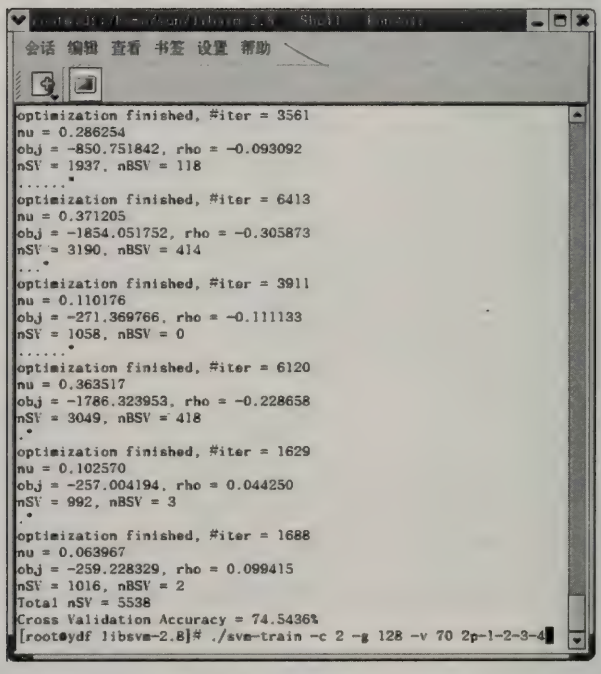


图 8-24　序列家族层的 6957 个样本的二肽频数输入样本集在 C=$2^{1.0}$, γ=$2^{7.0}$ 时, 通过 Jackknife 试验得到的预测准确率

4. 二肽频数样本二元分类结果

序列家族的 6957 个样本由二肽频数编码方法编码的向量集的二元分类器的 7 重交叉验证试验分类准确率与 Jackknife 试验分类准确率显示在表 8-17 中. 二元

分类器的优化相应参数、支持向量数以及支持向量数与样本数的比率的详细信息显示在表 8-18 中.

表 8-17 序列家族 6957 个二肽频数样本的折叠类二元分类 7 重交叉验证预测准确率与 Jackknife 试验预测准确率的比较

分类器	7 重交叉验证预测准确率/%	Jackknife 试验预测准确率/%
α vs. β	88.0098	88.225
α vs. α-β	80.5933	80.6159
α vs. fss	96.68	97.8
β vs. α-β	79.9417	80.1885
β vs. fss	97.6781	97.9142
α-β vs. fss	98.2977	98.4436
α vs. 其他	86.0141	86.2297
β vs. 其他	84.9073	85.5254
α-β vs. 其他	76.5129	76.2973
fss vs. 其他	99.1951	99.2957
平均准确率	88.783	89.053 53

表 8-18 序列家族 6957 个二肽频数样本的最优参数、支持向量数以及支持向量数和样本数的比率

分类器	最优参数		支持向量数 (约)	支持向量与样本的比率
	C	γ		
α vs. β	$2^{0.1}$	$2^{6.8}$	1780	0.626
α vs. α-β	$2^{0.0}$	$2^{7.0}$	2955	0.669
α vs. fss	$2^{7.0}$	$2^{8.0}$	1014	0.406
β vs. α-β	$2^{2.0}$	$2^{7.0}$	2588	0.581
β vs. fss	$2^{5.0}$	$2^{7.0}$	954	0.375
α-β vs. fss	$2^{7.0}$	$2^{7.0}$	1008	0.245
α vs. 其他	$2^{7.0}$	$2^{3.0}$	2906	0.418
β vs. 其他	$2^{9.0}$	$2^{1.0}$	2540	0.365
α-β vs. 其他	$2^{0.2}$	$2^{7.3}$	4486	0.645
fss vs. 其他	$2^{7.0}$	$2^{6.0}$	1475	0.212

在样本数量增加后二元和四元预测准确率明显提高. 由于考虑到了样本集不平衡的问题大幅增加了 (few secondary structures) 类的样本数量, 使预测该类对其他类别的分类准确率快速上升. 例如二元分类器 “fss vs. 其他” 的 Jackknife 试验预测

准确率超过了 99%, 这么高的预测准确率可以使我们断定 few secondary structures 类结构域的一级序列与其他三类结构域的一级序列有绝对的差别.

5. 序列家族层不同维数向量预测结果的比较

上面几节给出了序列家族层 6957 个样本依次依据一肽频数编码方法、二肽频数编码方法和三肽频数编码方法把氨基酸残基序列映射到不同向量空间后的预测准确率评估, 见图 8-25.

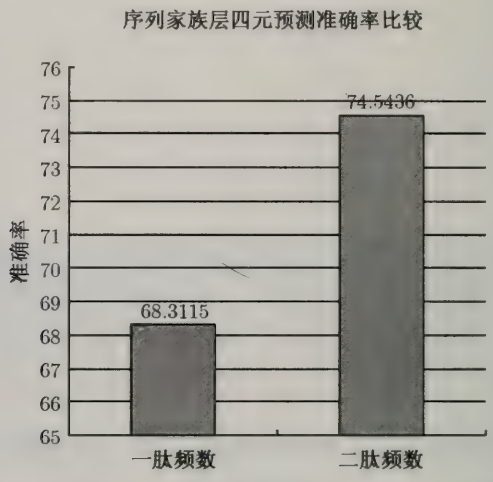


图 8-25　序列家族层的代表个样本的一肽、二肽、三肽频数向量预测准确率比较图

图 8-25 显示出的准确率变化与图 8-13 和图 8-20 所显示出的准确率变化趋势大不一样: 图 8-13 和图 8-20 所显示出的准确率变化趋势是前高后低, 图 8-25 而所显示出的准确率变化趋势是前低后高. 此时样本的数量为 6957 个, 而二肽频数编码方法编码的向量维数为 400. 这是样本数量与向量维数的比值已经超过了 17, 这时二肽频数编码方法编码的向量已经能够显示出比一肽频数编码方法编码的向量在获取蛋白质一级序列信息的优势了. 因此预测准确率大大高于一肽频数向量的预测准确率.

8.3.5.4　不同数量样本的预测结果比较

为了明晰由于样本数量的增加致使预测准确率的变化情况, 我们作了下面的比较. 首先比较一肽频数编码方法编码的向量预测准确率.

根据图 8-26, 一肽频数编码方法编码的向量为 20 维的向量, 当样本数量从 820 增加到 6957 时预测准确率增加了约 14%.

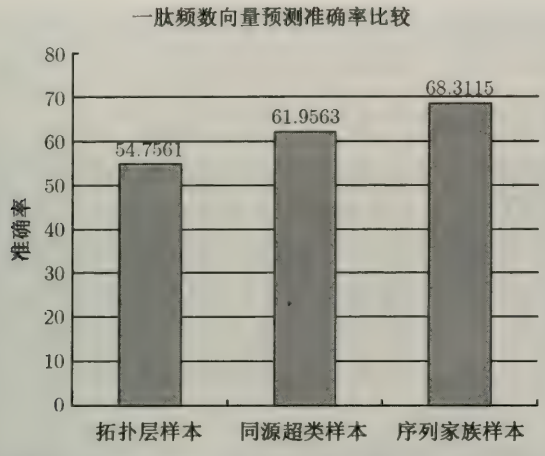


图 8-26 一肽频数编码方法编码的向量在样本数量分别为 820、1572 和 6957 时的预测准确率比较

图 8-27 显示出, 当样本数量从 820 增加到 6957 时二肽频数编码方法编码的向量的预测准确率增加了约 20%. 这个增加值比图 8-22 显示的增加值大些, 说明二肽频数编码方法当样本数量足够大时能更多反映残基序列的结构本质. 因此当样本数量少时, 尽量用产生低维数的编码方法; 当样本数量多时, 可以考虑使用维数高的编码方法.

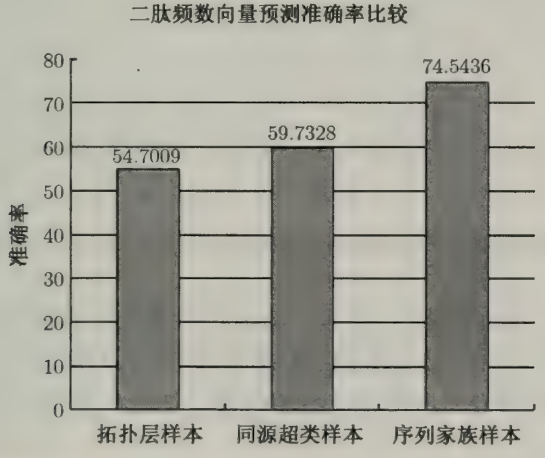


图 8-27 二肽频数编码方法编码的向量在样本数量分别为 820、1572 和 6957 时的预测准确率比较

对于三肽频数编码方法编码的向量当样本数量增加时的预测准确率暂时不作比较.

8.3.5.5　样本同源性提高对预测准确率的影响

在考察样本数量增加时蛋白质折叠类型预测准确率的变化情况, 我们在 CATH 数据库的不同层取得样本. CATH 数据库不同层样本的同源性是不一样的, 级别越高同源性越高. 拓扑层的样本之间的同源性不如同源超族层样本之间的同源性高, 同源超族层的同源性不如序列家族层样本之间的同源性高. 特别是在做序列家族层的折叠类型预测时, 为了防止产生样本集不同的问题我们取了 CATH 数据库中全部的 few secondary structures 类型的全部结构域来作为 fss 类的样本. 这些都会对预测准确率产生一定的影响.

为了考察序列同源性对预测准确率的影响, 设计了下面的试验. 首先, 从同源超族样本中取出 820 个样本. 在每个类型中连续取样本数量如下: α 类: 227; β 类: 139; α-β 类: 368; fss 类: 86. 为了方便比较, 每个类别所取的样本数与拓扑层每个类别样本数量一样. 运用一肽频数编码方法编码结构域一级序列, 进行优化的结果见图 8-28.

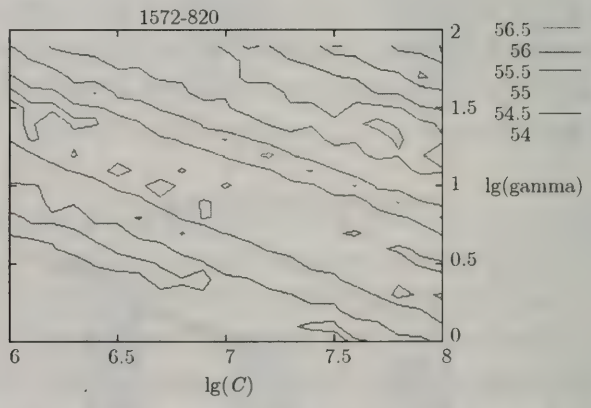


图 8-28　同源超族层的 820 个样本的一肽频数输入样本集的 7 重交叉验证试验优化结果

经过优化可知, 样本集在 C=$2^{7.2}$, γ=$2^{1.2}$ 时取得最高预测准确率. 计算 C=$2^{7.2}$, γ=$2^{1.2}$ 时样本集的 Jackknife 试验预测准确率. 结果见屏幕截图 8-29.

然后, 从序列家族样本中取出 820 个样本. 在每个类型中连续取样本数量如下: α 类: 227; β 类: 139; α-β 类: 368; fss 类: 86. 运用一肽频数编码方法编码结构域一级序列, 进行优化的结果见图 8-30.

经过优化可知, 样本集在 C=$2^{6.6}$, γ=$2^{0.1}$ 时取得最高预测准确率. 计算 C=$2^{6.6}$,

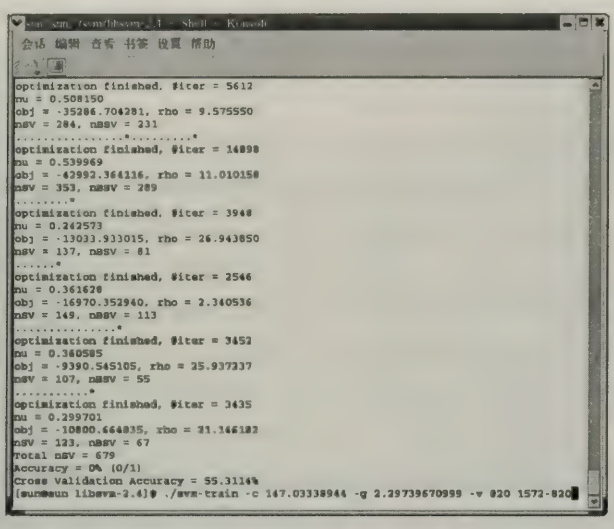


图 8-29 同源超族层的 820 个样本的一肽频数输入样本集在 C=$2^{7.2}$, γ=$2^{1.2}$ 时, 通过 Jackknife 试验得到的预测准确率

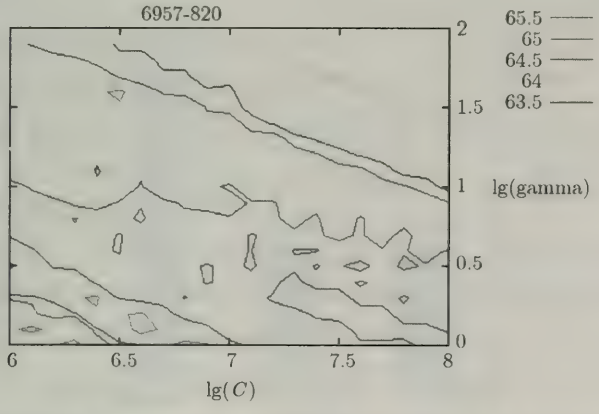


图 8-30 序列家族层的 820 个样本的一肽频数输入样本集的 7 重交叉验证试验优化结果

γ=$2^{0.1}$ 时样本集的 Jackknife 试验预测准确率. 结果见屏幕截图 8-31.

把图 8-29 和图 8-31 的结果与图 8-4 的结果比较, 见图 8-32. 可以看出当其他条件不变的情况下, 同源性增加时预测准确率提高.

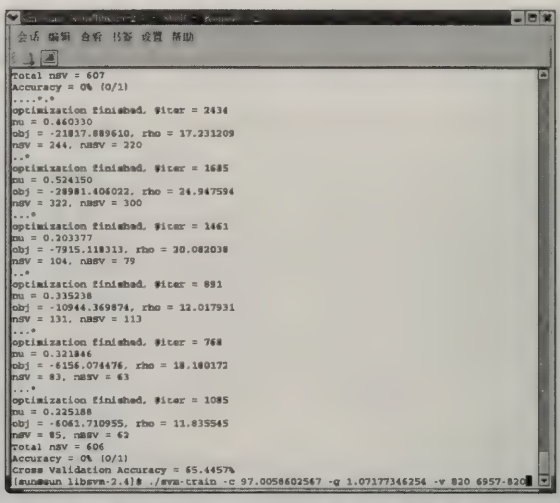


图 8-31　序列家族层的 820 个样本的一肽频数输入样本集在 $C=2^{7.2}$, $\gamma=2^{1.2}$ 时, 通过 Jackknife 试验得到的预测准确率

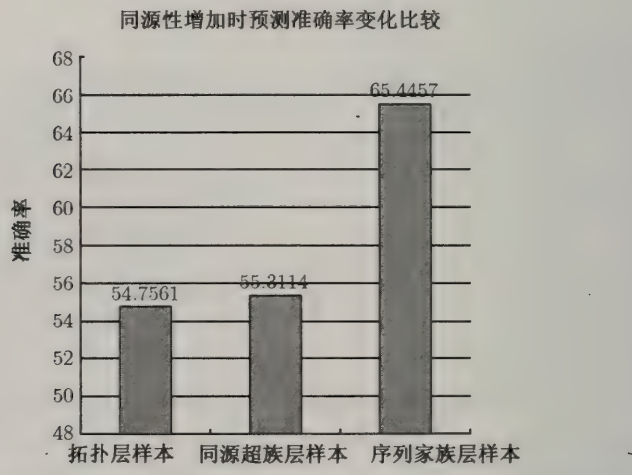


图 8-32　一肽频数编码方法编码的向量在样本数量为 820 时取自不同层次样本的预测准确率比较

8.4　小　　结

8.4.1　结论

从 8.3.5 节的结果中可以总结出如下规律:

(1) 四种不同的蛋白质折叠类型之间的氨基酸成分有明显区别. 但是仅利用氨基酸残基在序列种出现的频率而不考虑其他情况, 在样本数量充分时不能得到好的预测效果.

(2) 结构域的一级序列中相邻残基的关系对折叠类型有重大的影响作用. 在折叠类型预测时, 考虑到了紧邻残基后预测准确率有所增加.

(3) 结构差别大的结构域之间的二元分类准确率高, 结构差别小的结构域之间的二元分类准确率低. 比如二元分类器 "α vs. β"、"α vs. fss"、"β vs. fss"、"α-β vs. fss"、"β vs. 其他" 以及 "fss vs. 其他" 等二元分类器的预测准确率都较高; 而 "α vs. α-β"、"β vs. α-β"、"α vs. 其他" 以及 "α-β vs. 其他" 的预测准确率较低.

(4) 预测准确率高的二元分类器的支持向量数与样本数的比率较低, 反之亦然.

(5) 增加样本数量是治疗预测准确率低这个顽症的灵丹妙药.

(6) 样本数量的增加同样也可以导致支持向量与样本数量的比率的下降.

(7) 预测准确率不但与样本数量有密切联系还与输入空间维数有密切关系.

(8) 在其他条件不变的情况下, 样本同源性增加预测准确率提高.

(9) 最优参数 C 与 γ 的变化没有明显规律.

另外, 在运算过程中还可以总结出:

(1) 最优参数不是通过一次优化就能得到的, 必须经过反复几次调整参数的取值阈值才能最终确定最优参数的范围;

(2) 通过对参数的微调可以提高预测准确率;

(3) 使用 7 重交叉验证试验优化参数能得到最优准确率的参数有时是一个点, 有时是几个离散点, 有时则是一个或几个连续的区域;

(4) 样本集容量的增加使计算成本急剧上升;

(5) 虽然对于约 7000 个样本折叠类型预测的运算几乎已经达到了实验室计算能力的极限, 但是考虑到三肽频数编码的向量的预测准确率在样本数量增量的情况下仍然没有大幅改观, 说明样本数量显然仍然偏小.

8.4.2 讨论

进行蛋白质结构域折叠类型预测准确率依赖于两个因素: 预测方法和样本集. 恰当的、有更好泛化能力的预测方法能够更快、更准地进行结构域的预测. 作为预测蛋白质折叠类型工具的 SVM 包含了三个基本要素: 样本集和编码规则、学习机和决策函数. 实验中, n 肽频数编码规则用来把结构域的一级序列嵌入到输入空间中, 得到的向量决定了监视器和学习机的工作环境. 能被 SVM 分类的向量所具备的基本条件就是它们都必须在同一个向量空间中, 向量处于同一个空间中的必要条件是它们的维数必须相同. 然而, 不同的结构域往往包含不同数量的氨基酸残基. 因此把包含不同数量残基的结构域转化为同一个空间中的向量非常重要. 在以往的

机器学习方法中, 把氨基酸序列嵌入输入空间的编码方法分为两类: ① 滑窗方法; ② n 肽频数方法. 这两种方法的目的是相同的, 就是把用字母表示的不等长的氨基酸序列转换成为用数字表示的具有相等维数的向量, 并且通过这种转换尽可能地从氨基酸序列中挖掘更多的信息. 而这两种方法所使用的策略则有所不同. 滑窗方法使用长度 l 固定且为单数的滑窗从氨基酸序列中采集样本, 采集到的整个样本的类别由样本中间一个氨基酸的类别来确定. 嵌入到输入空间后, 相对应的向量长度为 $21\times l$. 这种编码方法适用于样本长度比较短的样本集, 所以多用于蛋白质的二级结构预测中. 运用 n 肽频数编码方法首先要把氨基酸序列包含所有的 n 肽频数计算出来, 归一化后成为输入空间中的向量, 向量的长度为 $20n$. 实验中一般 n 取 1、2 或 3. n 为 1 时实际得到的样本就是每个氨基酸在序列中的百分率, 这种样本反映不出相邻氨基酸之间的相互关系. n 大于 3 通过这种方法得到的输入向量维数过高, 超出了一般计算机的计算能力, 所以不常用. 由于样本空间向量的长度与氨基酸序列的长度无关, 所以这种方法既适用于样本长度较短的样本集, 也适用于样本长度较长的样本集. 这种方法多用于蛋白质结构域预测和亚细胞结构预测中.

实验中使用的样本集中的元素取自 CATH 结构域数据库中的拓扑层的结构域和同源超族层的结构域. 由于样本集中的样本数量和相互之间的同源性影响预测的泛化能力, 所以样本集尽可能选用同源性较小的样本. 又因为样本数量越多预测准确率就越高 [123], 为了客观地比较预测方法的优劣选择了 CATH 中的拓扑作为样本集中的元素. CATH 数据库和常常使用的 SCOP 数据库有所不同, CATH 数据库中把 α/β 和 $\alpha+\beta$ 结构同归于 α-β 类, 而且引入了小型二级结构 (fss) 这一类别, 通过这些样本训练得到的 SVM 具有更好的泛化性能.

使用支持向量机方法对蛋白质结构折叠类型预测不能达到百分之百准确的原因在于:

(1) 人们对于残基的分子力学和分子动力学性质了解不够透彻, 因此不能从氨基酸残基的理化性质来对其形成的高级结构进行精确的理论分析.

(2) 一级结构到高级结构的折叠受到多种因素的影响, 这些因素包括物理因素、化学因素以及生物因素.

(3) 一级结构到高级结构的折叠属于不适定问题. 即高级结构的轻微变化, 在反演原因时可能会导致完全不同的一级结构. 例如类似的高级结构可能是由完全不同的一级结构确定的.

(4) 解析高级结构时的误差, 即数据噪音.

(5) 二级结构和结构域定义的不统一. 蛋白质的二级结构和结构域到目前为止没有一个统一的定义, 每种定义都是根据一定的需要定义的.

(6) 在使用机器学习理论进行预测蛋白质的结构时, 没有理想的向量化方法. 目前使用的向量化方法都存在缺陷.

在支持向量机的应用过程中, 二元分类是多元分类的基础. 然而处理的问题往往是多种类型数据的分类问题 [155]. 因此对于生物领域的科学工作者来说处理各种多元分类的问题尤为重要. 在未来生物信息学的研究中, 支持向量机的主要应用领域一方面在于对于已经存在的数据库或新建立的数据库中的数据进行多元分类以便积累更多的知识; 另一方面在于开发新的多元分类算法 [156].

参 考 文 献

1. Trifonov E N. Earliest pages of bioinformatics. Bioinformatics, 2000, 16(1): 5–9.
2. Backofen R, Gilbert D. Bioinformatics and constraints. Constraints, 2001, 6: 141–156.
3. 钟杨, 张亮, 赵琼. 简明生物信息学. 北京: 高等教育出版社, 2001.
4. Luscombe N, Greenbaum D, Gerstein M. What is bioinformatics? An introduction and overview. IMIA. 2001.
5. http://www.bioon.com/biology/Class45/bioinfo/200409/78825.html.
6. Wooley J C. Trends in computational biology: a summary based on a RECOMB plenary lecture. J Comp Biol, 1999, 6: 459–474.
7. Bertone P, Gerstein M. Integrative data mining: the new direction in bioinformatics. 2001 IEEE Eng Med Biol Mag, 2001, 20: 33–40.
8. Tan A C, Gilbert D. Machine Learning and its Application to Bioinformatics: An Overview. 2001, http://www.brc.dcs.gla.ac.uk/~actan/papers/MachineLearning-Bioinformatics.pdf.
9. Goffeau A, Barrell B G, Bussey H et al. Life with 6000 genes. Science, 1996, 274: 563–567.
10. Bult C J, White O, Olsen G J et al. Complete genome sequence of the methanogenic archaeon, *Methanococcus jannaschii*. Science, 1996, 273: 1058–1073.
11. Fraser, C M., Gocayne, J D, White O et al. The minimal gene complement of Mycoplasma genitalium. Science, 1995, 270: 397–403.
12. International Human Genome Sequence Consortium. Initial sequencing and analysis of the human genome. Nature, 2001, 409: 860–921.
13. Economist T. Drowning in data. The Economist, 1999, 26 June.
14. http://cn.tech.yahoo.com/031216/84/1xos8.html.
15. http://www.henaninvest.gov.cn/gxjs/Article_Show.asp?ArticleID=9272.
16. Lesk A M, Chothia C. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. J Mol Biol, 1980, 136(3): 225–270.
17. Chothia C. One thousand families for the molecular biologist. Nature, 1992, 357(6379): 543–544.
18. Nicholas L et al. What is bioinformatics? A Proposed Definition and Overview of the Field, 40 Methods in Informatics & Med, 2001, 346.
19. Tatusov R L, Koonin E V, Lipman D J. A genomic perspective on protein families. Science, 1997, 278(5338): 631–637.
20. Gerstein M, Hegyi H. Comparing genomes in terms of protein structure: surveys of a finite parts list. FEMS Microbiol Rev, 1998, 22(4): 277–304.

21. Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res, 2000, 28(1): 27–30.
22. Benson D A, Karsch-Mizrachi I, Lipman D J et al. GenBank. Nucleic Acids Re, 2000, 28: 15–18.
23. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res, 2000, 28(1): 45–48.
24. McGarvey P B, Huang H, Barker W C et al. PIR: a new resource for bioinformatics. Bioinformatics, 2000, 16(3): 290–291.
25. Bernstein F C, Koetzle T F, Williams G J et al. The Protein Data Bank. A computer-based archival file for macromolecular structures. Eur J Biochem, 1977, 80(2): 319–324.
26. Berman H M, Westbrook J, Feng Z et al. The Protein Data Bank. Nucleic Acids Res, 2000, 28(1): 235–242.
27. Laskowski R A, Hutchinson E G, Michie A D et al. PDBsum: a web-based database of summaries and analyses of all PDB structures. TIBS, 1997, 22(12): 488–490.
28. Bleasby A J, Akrigg D, Attwood T K. OWL–a non-redundant composite protein sequence database. Nucleic Acids Res, 1994, 22(17): 3574–3577.
29. Bleasby A J, Wootton J C. Construction of validated, non-redundant composite protein sequence databases. Protein Eng, 1990, 3(3): 153–159.
30. Hofmann K, Bucher P, Falquet L, Bairoch A. The PROSITE database, its status in 1999. Nucleic Acids Res, 1999, 27(1): 215–219.
31. Pearl FM, Lee D, Bray J E et al. Assigning genomic sequences to CATH. Nucleic Acids Res., 2000, 28(1): 277–282.
32. Lo Conte L, Ailey B, Hubbard T J et al. SCOP: a structural classification of proteins database. Nucleic Acids Res, 2000, 28(1): 257–259.
33. Holm L, Sander C. Touring protein fold space with Dali/FSSP. Nucleic Acids Res, 1998, 26(1): 316–319.
34. Orengo C A, Pearl F M G, Bray J E et al. The CATH database provides insights into protein structure/function relationships. Nucleic Acids Res, 1999, 27: 275–279.
35. 来鲁华. 蛋白质的结构预测与分子设计. 北京: 北京大学出版社, 1993.
36. Anfinsen C B. Principles that govern the folding of protein chains. Science, 1973, 181: 223–230.
37. Burkhard Rost. Review: protein secondary structure prediction continues to rise. Journal of Structural Biology, 2001, 134: 204–218.
38. Zhiyong Zhang. An overview of protein structure prediction: from Homology to Ab Initio, 2003.
39. Thornton J M, Todd A E, Milburn D et al. From structure to function: approaches and limitations. Nature Sturctural Biology, 2000, 11: 1991–1994.

40. Orengo C A, Frances M, Pearl G et al. The CATH database provides insights into protein structure/function relationships. Nucleic Acids Res, 1999, 27(1): 275–279.
41. 沈同, 王镜严. 生物化学. 第二版. 北京: 高等教育出版社, 1990.
42. 蛋白质结构预测. http://www.lmbe.seu.edu.cn/chenyuan/xsun/bioinfomatics/Web/ChapterSeven/7.1.htm.
43. 蛋白质化学. http://www.hutc.zj.cn/smkxxy/biochemistry-web/text/4.htm.
44. Aik Choon Tan. David Gilbert. Machine learning and its application to bioinformatics: an overview, 2001, August 31.
45. Mitchell T M. Machine Learning. McGraw-Hill International, Singapore, 1997.
46. Vapnik V. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.
47. Vapnik V. Statistical Learning Theory. New York: John Wiley and Sons, 1998.
48. Frawley W J, Piatetsky-Shapiro G, Matheus C J. Knowledge discovery in databases: an overview. AI Magazine. 1992, Fall: 57–70.
49. Fayyad U M, Piatetsky-Shapiro G, Smyth P et al. Advances in Knowldee Discovery and Data Mining. AAAI Press, 1996.
50. Fayyad U M, Piatetsky-Shapiro G, Smyth P. The kdd process for extracting useful knowledge from volumes of data. Communication of the ACM. 1996, 39: 27–34.
51. Stormo G, Schneider T, Gold L, Ehrenfeucht A. Use of the perceptron algorithm to distinguish translational initiation in E.coli. Nuclei Acids Res, 1982, 10: 2997–3011.
52. Kröse B, van der Smagt, P. An introduction to neural networks. The Unversity of Amesterdam, 1996.
53. Hirst J D, Sternberg M J E. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. Biochemistry, 1992, 31: 7211–7218.
54. Qian N, Sejnowski T J. Predicting the secondary structure of globular proteins using neural network models. J Mol Biol, 1988, 202: 865–884.
55. Breiman L, Friedan J, Olshen R et al. Classification and regression trees. Wadsworth, Belmont, 1984.
56. Quinlan J R. C4.5: Programs for machine learning. Morgan Kaufmann, 1993.
57. Howard R, Matheson J. Influence diagrams. In: Readings on the Principles and Applications of Decision Analysis, volume II. CA: Strategic Decisions Group, Menlo Park, 1981, 721–762.
58. Cai D, Delcher A, Kao B et al. Modeling splice sites with Bayes networks. Bioinformatics, 2000, 16: 152–158.
59. Schmidler S C, Liu J S, Brutlag D L. Bayesian segmentation of protein secondary structure. J Comp Biol, 2000, 7: 233–248.
60. Davis L. Handbook of genetic algorithms. New York: Van Nostrand Reinhold, 1991.

61. Zhang C, Wong A K. A genetic algorithm for multiple molecular sequence alignment. Comput Appl Biosci, 1997, 13: 565–581.
62. Mamitsuka H. A learning method of hidden Markov models for sequence discrimination. J Comput Biol, 1996, 3: 361–373.
63. Glasgow J, Steeg E, Fortier S. Motif discovery in protein structure databases. In: Pattern Discovery in Biomolecular Data, Tools, Techniques, and Application. New York: Oxford University Press, 1999, 77–96.
64. Theodoros E, Pontil M. Machine learning and its applications. Advanced Lectures Lecture Notes In Computer Science, 2001, 2049: 249–257.
65. Burges C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 1998, 2: 121–167.
66. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20: 273–297.
67. Burges C, Scholkopf B. Advances in Kernel Methods: Support Vector Learning. Cambridge. MA: MIT Press, 1999, 185–208.
68. Baldi P, Brunak S. Bioinformatics: The Machine Learning Approach. MIT Press, 1998.
69. 边肇祺, 张学工. 模式识别. 北京: 清华大学出版社, 2000.
70. 魏权龄, 王日爽, 数学规划引论. 北京: 北京航空航天大学出版社, 1991.
71. 夏道行, 吴卓人. 实变函数论与泛函分析. 北京: 高等教育出版社, 1988.
72. Markowetz F. Support Vector Machines in Bioinformatics. Heidelberg, January 23, 2002.
73. Gunn S R. Support Vector Machines for Classification and Regression. Faculty of Engineering and Applied Science Department of Electronics and Computer Science, 1998.
74. Chang C C, Lin C J. Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
75. Cristianini N, Shawe-Taylor J. Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000.
76. 姬水旺, 姬旺田. 支持向量机训练算法综述. 微机发展, 2004, 14(1).
77. Boser B E, Guyon I M, Vapnik V. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth An2 nual Workshop on Computational Learning Theory. Pitts burgh, PA, USA: 1992, 144–152.
78. Osuna E, Freund R,Girosi F. An improved training algorithm for support vector machines. Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing. New York: IEEE Press, 1997, 276–285.
79. Platt J C. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. MA: MIT Press Cambridge, 1999, 185–208.
80. Keerthi S S, Shevade S K, Bhattachar yya C et al. Improvements to Platt's SMO algorithmfor SVM classifier design. Neural Computation, 2001, 13 (3): 637-649. In: Platt J C. Using analytic QP and sparseness to speed training support vector machines.

Kearns M S, Solla S A, Cohn D A. Advances in Neural Information Processing Systems. [s. l.]: MIT press, 1999, 557–563.

81. Schölkopf, Platt B J C, Shawe-Taylor J, Smola A J, Williamson R C. Estimating the support of a high-dimensional distribution. Neural Computation, 2001, 13 (7), 1443–1471.
82. Hsu C W, Lin C J. A simple decomposition method for support vector machines. Machine Learning, 2002, 46(1): 291–314.
83. Schölkopf, Smola B A, Williamson R C, Bartlett P L. New support vector algorithms. Neural Computation, 2000, 12: 1207–1245.
84. Chang, C C, Lin C J. Training v-support vector classifiers: theory and algorithms. Neural Computation, 2001, 13(9): 2119–2147.
85. Hsu C W, Chang C C, Lin C J. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf.
86. Andersen C A, Palmer A G, Brunak S et al. Continuum secondary structure captures protein flexibility. Structure, 2002, 10(2): 175–184.
87. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers, 1983, 22: 2577–2637.
88. Pauling L, Corey R B, Branson H R. The structure of proteins: two hydrogenbonded helical conformations of the polypeptide chain. Proc Natl Acad Sci USA, 1951, 37: 205–211.
89. Richardson J S. The anatomy and taxonomy of protein structure. Adv Protein Chem, 1981, 34: 167
90. Frishman D, Argos P. Knowledge-based protein secondary structure assignment. Proteins, 1995, 23, 566–579.
91. Heinig M, Frishman D. STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. Nucleic Acids Research, 2004, 32(Web Server issue): 500–502.
92. Andersen C A F, Rost B. Secondary structure assignment. Methods Biochem Anal, 2003, 44, 341–363.
93. Richards F M, Kundrot L E. Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins, 1988, 3: 71–84.
94. Sklenar H, Etchebest C, Lavery R. Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. Proteins, 1989, 6: 46–60.
95. 王大成. 蛋白质工程. 北京：化学工业出版社, 2003.
96. 张春霆. 生物信息学的现状与展望. 世界科技研究与发展, 2000.

97. Cuff J A, Barton G J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins, 1999, 34: 508–519.

98. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol, 2001, 308: 397–407.

99. Yang X, Wang B, Ng Y K. A protein secondary structure prediction framework based on the support vector machine. Proceedings of the Fourth International Conference on Web-Age Information Management (WAIM'03). LNCS 2762, Springer, 2003, 266–277.

100. Kim H, Park H. Protein secondary structure prediction based on an improved support vector machines approach. Protein Engineering, 2003, 16(8): 553–560.

101. Guo J, Hu C, Sun Z et al. A novel method for protein secondary structure prediction using dual-layer SVM and profiles. Proteins: Structure, Function, and Bioinformatics, 2004, 54: 738–743.

102. Rost B, Sander C. Prediction of secondary structure at better than 70% accuracy. J Mol Biol, 1993, 232: 584–599.

103. Sander C, Schneider R. Database of homology derived structures and the structural meaning of sequence alignment. Proteins: Sturct Funct, and Genet, 1991, 9(1): 56–68.

104. Hua S, Sun Z. Support vector machine approach for protein subcellular localization prediction. Bioinformatics, 2001, 17(8): 721–728.

105. Kim H, Park H. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor. Proteins: Structure, Function, and Bioinformatics, 2004, 54(3): 557–562.

106. Zemla A, Venclovas C, Fidelis K et al. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. Proteins: Struct, Funct, and Genet, 1999, 34: 220–223.

107. Sela M, White F H Jr, Anfinsen C B. Reductive cleavage of disulfide bridges in ribonuclease. Science, 1957, 125: 691–692.

108. Chothia C. One thousand families for the molecular biologist. Nature, 1992, 357: 543–544.

109. Aloy P, Russell R. Ten thousand interactions for the molecular biologist. Nature Biotechnol, 2004, 22: 1317–1321.

110. Saxonon S, Gilbert W. The universal of exons revisited. Genetica, 2003, 118: 267–278.

111. Berman H M, Battistuz T, Bhat TN et al. The protein data bank. Acta Cryst D, 2002, 58: 899–907.

112. Chou P Y. Amino acid composition of four classes of proteins. In: Abstracts of Papers, Part I, Second Chemical Congress of the North American Continent, 1980, Las Vegas, Nevada.

113. Chou P Y. Prediction of protein structural classes from amino acid composition,. In: Prediction of Protein Structure and the Principles of Protein Conformation. New York: Plenum Press, 1989, 549–586.

114. Nakashima H, Nishikawa K, Ooi T. The folding type of a protein is relevant to the amino acid composition. J Biochem, 1986, 99: 152–162.

115. Klein P, Delisi C. Prediction of protein structural class from amino acid sequence. Biopolymers, 1986, 25: 1659–1672.

116. Zhang C T, Chou K C. An optimization approach to predicting protein structural class from amino acid composition. Protein Sci, 1992, 1: 401–408.

117. Dubchak I, Holbrook S R, Kim S H. Predicting protein secondary structure content: a tandem neural network approach. Proteins, 1993, 16: 79–91.

118. Metfessel B A, Saurugger P N, Connelly D P et al. Cross-validation of protein structural class prediction using statistical clustering and neural networks. Protein Sci, 1993, 2: 1171–1182.

119. Rost B, Sander C. Combining evolutionary information and neural networks to predict protein secondary structure. Proteins, 1994, 19: 55–72.

120. Chandonia J M, Karplus M. Neural networks for secondary structure and structural class prediction. Protein Sci, 1995, 4: 275–285.

121. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol, 1993, 232: 584–599.

122. Hubbard T J P, Park J. Fold recognition and ab initio structure predictions using Hidden Markov models and b-strand pair potentials. Proteins, 1995, 23: 398–402.

123. Isik Z, Yanikoglu B, Sezerman U. Protein structural class determination using support vector machines. In: Lecture notes in computer science. Computer and information sciences. New York: Springer-Verlag, 2004, 3280: 82–89.

124. Chou K C, Zhang C T. Review: prediction of protein structural classes. Crit Rev Biochem Mol Biol, 1995, 30: 275–349.

125. Chou K C. Review: prediction of protein structural classes and subcellular locations. Curr Protein and Pept Sci, 2000, 1: 171–208.

126. Chou J J, Zhang C T. A joint prediction of the folding types of 1490 human proteins from their genetic codons. J Theor Biol, 1993, 161: 251–262.

127. Chou K C, Zhang C T. Predicting protein folding types by distance functions that make allowances for amino acid interactions. J Biol Chem, 1994, 269: 22014–22020.

128. Chou K C. A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space. Proteins, 1995, 21: 319–344.

129. Chou K C, Liu W, Maggiora G M et al. Prediction and classification of domain structural classes. Proteins, 1998, 31: 97–103.

130. Chou K C, Elrod D W. Using discriminant function for prediction of subcellular location of prokaryotic proteins. Biochem Biophys Res Commun, 1998, 252: 63–68.

131. Zhou G P. An intriguing controversy over protein structural class prediction. J Protein Chem, 1998, 17: 729–738.

132. Chou K C, Elrod D W. Protein subcellular location prediction. Protein Eng, 1999, 12: 107–118.

133. Chou K C, Elrod D W. Prediction of membrane protein types and subcellular locations. Proteins, 1999, 34: 137–153.

134. Yang Z R, Chou K C. Bio-support vector machines for computational proteomic. Bioinformatics, 2004, 20: 735–741.

135. Cai Y D, Liu X J, Xu X B et al. Support vector machines for predicting the specificity of GalNAc-transferase. Peptides, 2002, 23: 205–208.

136. Cai Y D, Liu X J, Li Y X et al. Prediction of β-turns with learning machines. Peptides, 2003, 24: 665–659.

137. Levitt M, Chothia C. Structural patterns in globular proteins. Nature, 1976, 261: 552–558.

138. Rechardson J S, Rechardson D C. Principles and patterns of protein conformation. In: Fasman G D. Prediction of protein structure and the principles of protein conformation. New York: Plenum Press, 1989, 1–98.

139. Creighton T. Proteins, structures and molecular properties. 2nd ed. New York: Freeman and Company, 1993.

140. Holm L, Sander C. Dictionary of Recurrent Domains in Protein Structures. Proteins, 1998, 88–96.

141. Orengo C A, Michie A D, Jones S et al. CATH- a hierarchic classification of protein domain structures. Structure, 1997, 5, 1093–1108.

142. Pearl F M G, Lee D, Bray J E et al. Assigning genomic sequences to CATH. Nucleic Acids Res, 2000, 28(1): 277–282.

143. Pearl F M, Martin N, Bray J E et al. A rapid classification protocol for the CATH domain database to support structural genomics. Nucleic Acids Res, 2001, 29(1), 223–227.

144. Orengo C A, Brown N P, Taylor W T. Fast structure alignment for protein database searching. Proteins, 1992, 14: 139–167.

145. Orengo C A, Taylor W R. SSAP: sequential structure alignment program for protein structure comparison. Methods in Enzymol, 1996, 266: 617–643.

146. Tan A C, Gilbert D, DevilleY. Integrative machine learning approach for multi-class SCOP protien topologyification. GCB03: German Conference on Bioinformatics, 2003, 153–159.

147. Holm L, Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol, 1993, 233: 123–138.
148. Orengo C A, Michie A D, Jones S et al. CATH- a hierarchic classification of protein domain structures. Structure, 1997, 5: 1093–1108.
149. CATH Protein Structure Classification. http://www.biochem.ucl.ac.uk/bsm/cath/class.html.
150. Michie A D, Orengo C A, Thornton J M. Analysis of domain structural class using an automated class assignment protocol. J Mol Biol, 1996, 262: 168–185.
151. Murzin A G, Brenner S E, Hubbard T et al. SCOP–a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol, 1995, 247: 536–540.
152. Taylor W R, Orengo C A. Protein structure alignment. J Mol Biol, 1989, 208: 1–22.
153. Ryan D, David A C, Beck R S et al. A consensus view of fold space: combining SCOP, CATH, and the dali domain dictionary protein. Science, 2003, 12: 2150–2160.
154. Dietmann S, Holm L. Identification of homology in protein structure classification. Nat Struct Biol, 2001, 8(11): 953–957.
155. Orengo CA. CORA–topological fingerprints for protein structural families. Protein Sci, 1999, 8: 699–715.
156. Anguita D, Ridella S, Sterpi D. A new method for multiclass support vector machines. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference. 2004, 412–417.

附表 1　RS126 数据集

1fc2c	1wsya	6cpp	1gd1o	2rspa
1mrt	1rbp	9apia	5ldh	6acn
1ovoa	3rnt	1fkf	2glsa	2stv
1ppt	5cytr	1gdj	1crn	1gp1a
1ubq	2aat	7cata	3pgm	9pap
2mhu	1fnd	2pcy	4pfk	1paz
2or1l	1l58	5hvpa	2gbp	3ait
2tgpi	2ak3a	1lap	1bbpa	1pyp
3b5c	3blm	1sh1	1azu	3hmgb
9wgaa	1fxia	1wsyb	4sdha	2cyp
1eca	2ccya	2ltnb	7icd	1bmv1
2cab	6dfr	2sns	8abp	4bp2
5lyz	9insb	1s01	2ltna	4ts1a
2hmza	1lmb3	4rhv1	6tmne	1hip
1acx	2gn5	1r092	2fox	1cdta
2alp	2gcr	2fxb	3tima	9apib
1cc5	1tgsi	4cms	2i1b	2mev4
3icb	5er2e	6cpa	1bmv2	4rxn
1fdlh	1fdx	1tnfa	3hmga	3cla
3cln	4cpv	2paba	4rhv3	4rhv4
1il8a	3gapa	2tsca	2wrpr	1etu
2utga	4sgbi	1mcpl	6cts	256ba
1csei	8adh	4xiaa	4gr1	1bds
3ebx	6hir	3cd4	2phh	2lhb
7rsa	1rhd	2tmvp	2sodb	4cpai
1cbh				

源自：http://www.compbio.dundee.ac.uk/~www-jpred/data/pred_res/126_set.html

附表 2　CB513 数据集

1*aozb*-1-*AS*	1*bncb*-3-*AS*	1*gog*-2-*AS*.1	2*tmdb*-3-*AS*
1atpi-1-DOMAK	1gdj	1comc-1-DOMAK	1vokb-1-AS
1ayab-1-GJB	2hft-1-AS	1vjs-3-GJB	1bmv2
1bsdb-1-DOMAK	1gky-2-AS	2reb-2-DOMAK	3hmga
1coi-1-AS	1krca-1-AUTO.1	1ecl-4-AS	4rhv3
1cthb-1-DOMAK	1smpi-1-AS	1lmb3	1mmoh-1-AS
1ctm-2-DOMAK	7cata	1rhgc-1-DOMAK	1nlkl-1-DOMAK
1ctn-1-AS.1	1ncg-1-AUTO.2	1ubdc-2-AS	1mof-1-AS
1edmc-1-AUTO.1	1gln-4-AS	2gn5	1ndh-1-AS
1fc2c	1hmy-2-AS	2gcr	1tsp-1-AS
1gln-3-AS	1dnpb-1-AUTO.1	1oacb-3-AS.1	1dar-3-AS
1gp2a-1-AUTO.1	1lap	1amg-2-AS	1sfe-2-AS
1grj-2-AS	1sh1	1bncb-1-AS	2wrpr
1hcgb-1-AS	1wsyb	2asr-1-DOMAK	1taq-2-AS
1htrp-1-AS	1clc-2-AS.1	2hhmb-1-DOMAK	1brse-1-DOMAK
1hup-1-AS.1	2ltnb	1fbab-1-DOMAK	1krcb-1-AS
1ilk-2-AS	2sns	5er2e	2hft-2-AS
1isub-1-DOMAK	3pmgb-1-AS	1ctu-1-AUTO.1	6cts
1lpe-1-DOMAK	1cpcl-1-DOMAK	1lbu-1-AS	4gr1
1mcti-1-AUTO.1	1bcx-1-DOMAK	1pii-2-DOMAK	1delb-2-AUTO.1
1mdta-1-AS	1s01	1cbg-1-AS	1hslb-2-DOMAK
1mrt	1powb-1-DOMAK	1powb-3-DOMAK	2bopa-1-DOMAK
1ndh-2-AS	4rhv1	1fdx	2phh
1ovoa	1vcab-1-AUTO.1	1horb-1-AUTO.1	2sodb
1pga-1-DOMAK	1mdta-2-AS	2spt-1-DOMAK	1qbb-4-AUTO.1
1powb-4-DOMAK	1han-1-AUTO.1	3ecab-1-AS	1alkb-1-AS
1ppt	1kuh-1-AS	1aorb-3-AS	1aozb-3-AS
1reqc-1-AS	1aazb-1-DOMAK	1cxsa-4-AUTO.1	2cmd-2-GJB
1rpo-1-AUTO.1	1pda-3-AS	3pgk-2-AS	2afnc-2-AUTO.1
1svb-2-AS	1dkza-1-JAC	1lbu-2-AS	1nbac-1-AS
1tabi-1-DOMAK	1pdo-1-GJB	1hcra-1-DOMAK	2rspa
1ubdc-1-AS	1svb-1-AS	1sfe-1-AS	1oacb-1-AS.1
1ubq	1trb-2-AS	1umub-1-AS	1vmob-1-AS
1wapv-1-AUTO.1	1cei-1-GJB	3gapa	1pmi-2-GJB
1wfbb-1-AUTO.1	1r092	1rvvz-1-AUTO.1	3ecab-2-AS
2aaib-2-DOMAK	1vid-1-JAC	1znbb-1-AS	1amp-1-AS
2erl-1-AUTO.1	1rie-1-GJB	1pda-2-AS	2yhx-3-DOMAK

续表

1*aozb*-1-*AS*	1*bncb*-3-*AS*	1*gog*-2-*AS*.1	2*tmdb*-3-*AS*
2mhu	2sil-1-AS	4sgbi	6acn
2mltb-1-GJB	1masb-1-AUTO.1	1oxy-3-AS	1mdam-1-DOMAK
2or1l	1powb-2-DOMAK	1hnf-1-AS	3chy-1-DOMAK
2tgpi	1cgu-3-GJB	1ese-1-AUTO.1	1hplb-2-AS
3b5c	1isab-1-GJB	1otgc-1-AS	3pmgb-4-AS
3pmgb-2-AS	1vpt-1-JAC	1ptr-1-AUTO.1	1bfg-1-DOMAK
6rlxd-1-DOMAK	1epbb-1-DOMAK	8adh	1lki-1-AS
9wgaa	2npx-3-AS.1	1qrdb-1-AUTO.1	1vcc-1-AS
1nga-2-AS.1	2polb-1-AS	1oacb-2-AS.1	2stv
1gpmd-5-AS	2fxb	1gep-3-AS	1gp1a
1asw-1-AUTO.1	1scud-1-AS	1grj-1-AS	2end-1-DOMAK
1eca	1chd-1-AS	1gym-1-AUTO.1	9pap
1fuqb-1-AUTO.1	1hjrd-1-AUTO.1	1dlc-3-AS.1	1dik-1-AS.1
1zymb-2-AUTO.1	1srja-1-DOMAK	6hir	1dfnb-1-DOMAK
2cab	1hvq-1-AUTO.1	1jud-1-GJB	1fdt-1-AS
5lyz	3pmgb-3-AS	1find-2-AUTO.1	1nozb-2-AUTO.1
1cnsb-1-AUTO.1	1din-1-AS	1pbwb-1-AS	1paz
1mspb-1-AS	1gln-2-AS	1rhd	3ait
1mai-1-JAC	1ghsb-1-GJB	1lba-1-DOMAK	1dik-4-AS.1
1dlc-1-AS.1	1gog-1-AS.1	1seib-1-AUTO.1	1pyp
1dynb-1-AUTO.1	1ktq-1-AUTO.1	1hplb-1-AS	1lis-1-DOMAK
2hmza	2rsla-1-GJB	1qbb-3-AUTO.1	1tndb-2-DOMAK
3mddb-2-AS	6cpa	1nar-1-DOMAK	1daab-2-AS
1vcab-2-AUTO.1	1lehb-3-AS	1reqc-2-AS	1vhh-1-AS
1acx	1pnmb-2-AS	1smnb-1-AUTO.1	1rlds-1-DOMAK
1cewi-1-DOMAK	1tnfa	1dik-3-AS.1	1fjmb-2-AS
1ilk-1-AS	2paba	1gmpb-1-DOMAK	1rec-1-DOMAK
1sesa-2-AS	2tsca	2olba-3-AS	1cqa-1-AUTO.1
1irk-2-AS	1hyp-1-DOMAK	1edd-1-DOMAK	1eft-3-DOMAK
1cfb-1-AS	2afnc-1-AUTO.1	1gd1o	1thx-1-AUTO.1
2alp	2tgi-1-DOMAK	1daab-1-AS	3hmgb
1stfi-1-DOMAK	154l-1-AUTO.1	5ldh	1bet-1-DOMAK
1thtb-1-AUTO.1	1dih-2-AS	1tie-1-DOMAK	2cyp
1nal4-1-AUTO.1	2dln-3-AS	1spbp-1-AS	1ceo-2-AUTO.1
1ris-1-DOMAK	1cem-1-GJB	1pyta-1-AS	1bmv1
1tml-1-AS	1nol-1-AUTO.2	2glsa	1cksc-1-AUTO.1
2ebn-1-AS	4xiaa	1ppi-2-AS	4bp2
1gep-2-AS	5sici-1-DOMAK	1gal-2-AS	1tul-1-JAC
1dpgb-1-AUTO.1	3cd4	1trh-1-AS	1dfji-1-AUTO.1
1tig-1-AUTO.1	1wsya	1crn	1yrna-2-AS
1celb-1-AUTO.1	1aorb-1-AS	1gflb-1-AS	1bam-1-AS

续表

1*aozb*-1-*AS*	1*bncb*-3-*AS*	1*gog*-2-*AS*.1	2*tmdb*-3-*AS*
2hpr-1-DOMAK	1kptb-1-AUTO.1	1gtqb-1-AUTO.1	1trkb-3-AS
1cc5	1mla-2-AS.1	1ignb-2-GJB	4ts1a
1fuqb-2-AUTO.1	1rbp	1mjc-1-DOMAK	1gtmc-2-AUTO.1
1pht-1-AUTO.1	1cpn-1-DOMAK	3pgm	1tssb-2-DOMAK
2spt-2-DOMAK	1ecl-1-AS	1udh-1-AUTO.1	1hip
1mdta-3-AS	3rnt	4pfk	1mrrb-1-DOMAK
1onrb-1-AUTO.1	1bovb-1-DOMAK	1gcb-2-AS	1aozb-2-AS
1mns-2-AS	5cytr	1inp-1-AS.1	2admb-2-AUTO.1
1nfp-1-AS	1clc-1-AS.1	1eceb-1-AUTO.1	1cdta
3icb	1find-1-AUTO.1	1efud-2-AUTO.1	1tiic-1-GJB
1latb-1-AUTO.1	1pkyc-2-AUTO.1	2gbp	9apib
4fisb-1-DOMAK	1ecpf-1-AUTO.1	1qbb-1-AUTO.1	2mev4
1fdlh	1vhrb-2-AUTO.1	2dkb-2-AS	1gpmd-4-AS
3cln	1xvab-1-GJB	2reb-1-DOMAK	1han-2-AUTO.1
1il8a	1euu-2-JAC	1inp-2-AS.1	1pkyc-3-AUTO.1
1oacb-4-AS.1	1oyc-1-AS	1tfr-1-GJB	4rxn
2utga	2cpo-1-AUTO.1	1bbpa	3cla
1ctf-1-DOMAK	1gcmc-1-AUTO.1	1scue-3-AS	1edn-1-AS
1rsy-1-AS	2aat	1lpba-1-DOMAK	2dln-1-AS
1fuqb-3-AUTO.1	2trt-1-AUTO.1	1azu	1cgu-4-GJB
1dik-2-AS.1	1fnd	1kte-1-AS	1chmb-1-DOMAK
1dsbb-2-AUTO.1	1rlr-2-JAC	2mtac-1-AS	1poc-1-DOMAK
2pgd-2-AUTO.1	1l58	3cox-1-AS.1	2hipb-1-DOMAK
1csei	1lib-1-DOMAK	2phy-1-GJB	1adeb-2-AUTO.1
7rsa	1ctu-2-AUTO.1	4sdha	4rhv4
2nadb-2-AS.1	1tupc-1-AUTO.1	7icd	1add-1-AS
1qbb-2-AUTO.1	1gnd-2-JAC	3cox-2-AS.1	1etu
3inkd-1-DOMAK	1tplb-3-AS	1fua-1-AUTO.1	1pbp-2-DOMAK
2pgd-1-AUTO.1	2ak3a	1rec-2-DOMAK	2scpb-1-DOMAK
1dnpb-2-AUTO.1	3blm	1scue-2-AS	256ba
1esl-1-GJB	1cgu-2-GJB	1stme-1-AUTO.1	1pdnc-2-AS
1gp2g-2-AS	1fxia	1mdaj-1-GJB	1colb-1-DOMAK
1bncb-4-AS	1ptx-1-AS	2ltna	1fbl-1-AS
6cpp	1vnc-1-JAC	1bdo-1-AS	1bds
1sftb-2-AS	2ccya	1nox-1-GJB	2abk-2-AS
1seib-2-AUTO.1	1chkb-2-AUTO.1	1ovb-1-GJB	1ahb-2-GJB
9apia	1cyx-1-AUTO.1	1irk-1-AS	1avhb-4-AS
2bat-1-GJB	1cfr-1-GJB	6tmne	2bltb-2-AUTO.1
2gsq-2-AS	1dts-1-AUTO.1	2fox	1avhb-3-AS
821p-1-DOMAK	3bcl-1-DOMAK	2admb-1-AUTO.1	1clc-3-AS.1
1isab-2-GJB	1gpc-1-AS	1gog-3-AS.1	4cpai

续表

1*aozb*-1-*AS*	1*bncb*-3-*AS*	1*gog*-2-*AS*.1	2*tmdb*-3-*AS*
1fkf	1gal-3-AS	1hnf-2-AS	1yptb-1-AUTO.1
1tcba-1-AS	1knb-1-AS	2dnja-1-AS	1bpha-1-DOMAK
1hxn-1-AS	6dfr	1dupa-1-AS	2hhmb-2-DOMAK
1pnt-1-AS	1tcra-2-GJB	2olba-2-AS	1rlr-1-JAC
1chbe-1-DOMAK	1sra-1-AS	1csmb-1-AUTO.1	1whi-1-AS
1hiws-1-AS	1regy-1-AUTO.1	3tima	1cdlg-1-DOMAK
1dpgb-2-AUTO.1	3mddb-1-AS	2i1b	2tmvp
1kinb-1-AUTO.1	9insb	1hmpb-1-AUTO.1	1ctn-3-AS.1
3mddb-3-AS	1trkb-1-AS	1tif-1-AS	1cbh
6rlxc-1-DOMAK			

源自：http://www.compbio.dundee.ac.uk/~www-jpred/data/pred_res/513_set.html

附表 3　蛋白质结构域拓扑层预测样本集

1. 以 α 螺旋为主							
1cuk03	1kgqA1	1bgw03	1lxl00	1bgf00	1mhlC0	1bg8A0	1j09A2
1lea00	1agrE2	1anv01	1e3aA1	1ddf00	1frvB0	1ihp02	1qbhA0
1ey3A2	2trcP2	1zymA2	2end00	1bucA1	1b4uA0	1a6q02	1f6vA0
1mpgA3	1bvp13	1c3cA1	1pmi02	1pbwA0	1ft5A0	1a81A2	1eijA0
1bfmA0	1vom04	1bgvA3	2occD0	1derA1	1a8vA1	1eyvA0	1dnyA0
1hryA0	1jvr00	1a8rA1	1aihA0	1lbd00	1ile01	1qqvA0	1mylA0
1rlr01	1kblA2	1fjgO0	1rss00	1aorA2	1b7eA2	1pbv02	1bmtA1
1ahuA4	1vpu00	1msk02	1ecl02	2pgd02	1lm8V1	1d0cA3	1jhgA0
2tdx02	1agrE1	1ecl04	1vom05	1qmgA2	1l0lD2	2dpmA2	1bt3A0
6insE0	1utg00	1adeA2	1prcC2	1ak000	2hdhA2	1a9xA4	16vpA0
1lbu01	1ala01	2abk01	1vsgA2	1dnpA3	1fgjA1	1bg602	1f0jA0
1hyp00	1nkl00	2tct02	1vin01	1csh01	1ag200	1c05A1	1aroP1
1bip00	1csc02	1gdz00	1pjr04	1csmA0	1pah00	1qlaB2	1daqA0
1a36A4	1iku01	1npc02	1cpo00	1aorA3	1hp800	1qmmA5	1e3aA2
1crkA1	1beo00	1rlr02	1hlm00	1uby00	1a1vA3	2hgsA3	1qhdA2
1cmbA0	2ts102	1cipA2	1wer02	1sig00	1nvvS2	1be3F0	1hhsA4
1bpb01	1ycqA0	1d5tA1	1wer01	1vnc02	1ej5A0	1tbaA0	1i1iP3
1af701	1e7aA1	1aoa01	1apmE1	5eat02	1jweA0	1du2A0	1dmtA2
1vom03	1bvp11	1mkrA2	1aru01	1afrA0	1iieA0	1e39A2	1hm5A3
1jud02	1dulA0	1cf9A2	153l00	1phb00	1sknP0	1h8eD3	1e3aB3
1fa0A3	1ewqA3	1lrv00	1qsaA1	1a1700	1pprM1	1cem00	5eas01
1vdfA0	1clqA4	1qlaC0	1cb8A1	1fgjA2	1dqsA2	1lre00	1bqv00
1ehs00	1dlc01	1om2A0	1m1nB4	1abz00	1qb4A1	1lbeA1	1be3C0
1aty00	1fupA2	1brwA3	1poc00	1bkdS1	1eulA2	1qovM1	1hbnB2
1knyA2	2occA0	1ja1A3	1aa7A1	1l6eA0	1f31A2	1bgxT4	1vnc01
2erl00	1e68A0	1dg3A1	2ccyA0	1by1A0	1jb0A0	1brrA0	2lisA0
1mtyG1	1yge05	1f7uA2	1bucA3	1n45A0	1g7dA0	1fx8A0	1dvkA0
1ecmA0	1cii01	1f81A0	1chkA1	1b91A0	1g1eB0	1efyA1	1eo0A0
1hb6A0	1a5t03	1glqA2					
2. 以 β 折叠为主							
1pdc00	1ahl00	1ytfD2	1havA1	1npoA0	3bcl00	1kapP1	
1skz01	1pft00	1dkgA2	1jsg00	2occF0	1p35A0	1lxa01	
1tle00	1rkd01	1auuA0	1pk400	1aqt01	1oen02	1air00	
1extA2	1lkoA2	1dorA2	1eft03	1amm01	1dhx01	1dlc03	
1lml03	1dar03	1fgp00	1bli02	1mdaL0	1ois02	1f8dA0	
1cdq00	1ospO2	1aqcA0	1pkm03	1dkxA1	1lktA0	1rie00	
1bi6H0	1mknA0	1ihvA0	1flmA0	1mwaA2	1preA3	2bbkH0	

续表

1tpm00	1ospO1	1pcfA0	1qu4A1	1hplA2	1vdeA1	1m6pA0	
2viuA2	1mkcA0	1qldA0	1bw300	1knb00	2cas00	1eulA1	
1lpbA0	1qs1A1	1aonO0	1jic00	1lla02	1gc1G0	1g6q12	
1hcnB0	1ci0A0	1lml04	1fivA0	1svb01	1bdfA2	1qd6C0	
1fbr00	1pnkB2	1seiA2	1clh00	1thv00	1by5A1	1dfuP0	
1umuA0	1dhx04	1yagA2	1bucA2	1cauB0	1dqcA0		
1ctl00	4htcI0	1eh6A1	1smpI0	3pcgA0	1fwqA0		
1ejxB0	1ecl03	1lci03	1whi00	1gff10	1faeA2		
1kmxA0	1dupA0	1hp7A1	1ema00	1vpsA0	1d8cA2		
1be3I0	1rhoA0	1a5mC1	1prn00	1ygs00	1fjrA2		
1igrA2	1gpr00	1pmi01	1af6A0	1d00A0	1g8lA3		
1exkA0	1jz8A5	1pdr00	1ggeA1	1cb8A3	3aahA0		
1ep3B3	1celA0	1l5bA0	1otcB0	1cq3A0	1hcd00		
1jhnA2	1a2vA3	1hxn00	2sli03	1jpc00	1tl2A0		
3. α-β 结构							
1rthA1	1p32A0	1bp101	1aa8A2	2ifeA0	1msc00	1mil00	1bo1A1
1igd00	1dm9A0	1bp102	1hymA0	1stu00	1chkA2	1gnd02	1bo1A2
1fmtA2	1bnkA0	3daaA2	2baa02	1cksA0	1svb03	3pmgA4	1l0lA1
1af500	1bwzA1	1fuiA3	1d5tA3	1rblM0	1mdl01	1vom02	1nos03
1bcpB1	1hdmB1	1cmvA0	1ica00	1phk01	1b4vA2	2hhmA1	1qkfA0
1edqA3	1cz4A2	1ami04	1chc00	1bpb03	1preA2	1gd1O2	1byrA0
2msbA0	1qlmA1	1bd0A1	1uxy02	1pkp02	1geo01	3cla00	1qk9A0
1dpe03	1e01A0	1b8bA0	1kvdB0	1copD0	1rthA5	1gal03	1dujA0
1aak00	1fd4A0	1bowA0	1xvaA1	2reb02	4rhn00	1ah600	1ejkA0
1fcbA1	1e53A0	1c8zA0	1gatA0	1kp8A2	1fim00	1tys00	1dlxA0
1mkaA0	1e44B0	1d8hA0	1ecrA2	1ejxA0	1kptA0	1crkA2	1bia02
1onc00	1h5pA0	1qndA0	1b7yB1	1gpmA3	2phy00	1ay200	1cbf02
2polA3	1g8fA2	1fvgA0	9wgaA1	1cii02	1oacA1	1t1dA0	1dt9A1
1ezm01	1eg7A3	1ev0A0	2nef00	1ytbA2	1bpb02	191400	1ei1A2
1kw3B1	1fx3A0	1ff9A2	1ex7A1	1noyA1	1uxy03	1gccA0	1qf6A2
1msk01	1ejdA1	1bx4A2	1yua01	2sicI0	3daaA1	1f3cA0	1qd1B1
2cba00	1plq00	1fsu02	1vcc00	1dih02	1b66A0	1auz00	1e7uA4
1by200	2bnh00	1fbxA2	1svb02	1hlrA4	2antI2	1ap800	1prxA2
1mb100	1igrA1	1fjgC2	1otfA0	1brsD0	1c96A3	2if100	1qlmA2
1uox00	1dctA2	4jdwA0	1cbn00	1c7sA2	1jufA1	1jbwA2	1dtvA0
1div01	1rhs01	1b94A0	1dioB0	2hgf00	1aorA1	1d0cA2	1qqcA4
1adn00	1fuiA2	1ag8A1	1eq6A0	1ab8A0	2dnjA0	1g3p02	1bolA0
1svq00	1ag8A2	2ctc00	1cbf01	1jon00	1znbA0	1dj7A0	1ile02
1aba00	1chmA1	1tplA2	1tdj03	1rgs02	1gdoA0	1bxeA0	1ebfA2
1cfe00	1mla02	1c8kA2	1brwA2	1efuB2	4kbpA2	1geo02	1fcdA3
1ble00	1qhaA2	1vpt00	1ekjA0	1ecrA1	1a6q01	1qgwA0	1qaxA2

续表

1pxtA1	1hfc00	2bltA0	1ami02	1ibvB0	3pvaA0	1mhdA0	1ush02
1m1nB2	1avpA0	1iso00	1poiA1	1dik03	1b65A0	1d4uA0	1g59A3
1lba00	1br6A1	1alkA0	1cjyA2	1b7yB3	1qr0A0	7ceiB0	1f3mA0
1cfr00	1ra900	2cevA0	1hynP0	3ladA2	1jenA0	1qg8A0	1gh9A0
1nox00	1adeA1	1poiB0	1eulA3	1qjfA0	1hzgA0	1mwpA0	1dn1A2
3pmgA3	1ayl01	1b4uB0	1jj2L0	1eyqA2	1emsA1	1d0qA0	1pfo01
1ctt02	3pmgA1	1bg200	1k30A2	1qklA0	1i7sA0	1gsoA4	1qcnA2
1anf02	1e8gA3	1dhs00	1qhlA0	1b78A0	1a4400	1efdN2	1ey2A1
1lehA1	1udg00	1kekA3	1auk01	1dbuA0	1apj00	1yagA4	1hruA0
1cby00	2ts101	1a3aA0	1b7bA0	1ccwB2	1aol00	1cliA2	1k2fA2
1k0zB0	1aua01	1hfeL3	1ewqA1	1dd9A1	1avqA0	1b37A2	1hhsA1
1lam01	1uch00	1vsrA0	1f75A0	1qorA1	1ahjA0	1qlaA4	1fs7A2
1fua00	1g8tB0	1qhkA0	1uag02	1acc01	1d0cA1	1qmmA1	1g71A1
1cxsA2	1cl8A0	1di6A0	1dxrH2	1rlr04	1b7eA1	1kssA1	1ospO3
2cy300	1lml02	1rlr03	1eu1A3	1ytn00	1bob01	1fgs01	1admA2
1a31A3	1anv02	1masA0	1hqi00	1gpc00	1ueaB1	1uxy01	1dubA1
1htmB0	1luvA0	1dhx03	1ckmA1	1bgw05	1o7nA1	1mut00	1efyA2
1udb01	1vsgA1	1f13A2	1theA0	1glaG1	1hbnA1	1yppA0	1chmA2
1dhx02	1adeA3	1lgr01	1a73A0	1ltsA0	1uok02	1ordA4	1fokA1
1def00	1ddt01	1gpb01	1dpe01	1fib01	1soxA2	2cmd02	1co4A0
4. Few secondary structures							
1bg503	2pspA1	2pgd03	1b35D0	1jpwD0	1edxA0	1fib02	1bct00
1ba305	1etrL0	1ldl00	1cwxA0	1jjuC0	1ceeB0	1prcH1	1fc2C0
4mt200	3aahB0	1d0dA0	1devB0	1jj2A3	1qojA0	1cl1A1	1ebdC0
1tiv00	1olgA0	1kvdA0	1cfi00	1jj2B3	1faeA3	2prgC0	2ilk02
1iva00	2bbvD0	1hnr00	1hykA0	1jeyA3	1rdr01	1ekcC0	1pnbB0
1fre00	1tvs00	1cf3A2	1b8xA3	1fqjC0	1qa4A0	1aoo00	1jsuC0
2occL0	1kzuA0	1inp01	1kekA4	2occM0	1g3jB0	1d6gA0	1f8nA3
2occK0	1aml00	1br6A2	1kekA7	1ckmA3	1fjgM2	1cf4B0	1yge02
1aaf00	1aw600	2frvA2	1icfI0	1qqp40	2occG0	1h7dA0	1f02T0
2ech00	1gp2G0	1isuA0	1gp8A0	1occJ0	1gzi00	2occI0	1fjgN0
1fleI0	1wdcA0	1pyaA0	1k5mD0	1mdyA0	1hueA0		

说明：附表 3 中使用的是 CATH 数据库六字母代码

源自：http://www.biochem.ucl.ac.uk/bsm/cath/class.html

附表 4　蛋白质结构域同源超族层预测样本集

1. 以 α 螺旋为主

1iw0A0	1oaiA0	1lvk04	1df4A0	1b4uA0	1rykA0	1nkzB0
1e6iA0	1aua02	1jvr00	1nkd00	1ft5A0	1iw7E0	1ezvI0
1eo0A0	1efuB1	1kblA2	1joyA0	1e7lA2	1iyrA0	1pp9E1
1dvkA0	1gab00	1vpu00	1mswD2	1pfvA3	1mkiA2	1f8vD0
1qlaC0	1mu5A2	1dk8A1	1b0nB0	1musA3	1ng6A1	1fjkA0
1om2A0	1c1kA1	1utg00	1mswD4	1lm8V1	1o0wA1	1svfA0
1khdA1	1j09A4	1axn01	1eejA2	1c75A0	1nxuA1	1g2cB0
1jr3D2	1g8pA2	1l9lA0	1g6uA0	1fgjA1	1mi1A2	1avyA0
1jb0K0	1h72C3	1iomA2	1jj2U0	1i4mA0	1pujA2	1ic2A0
2cblA1	1qamA2	2pvbA0	1ezjA1	1ltzA0	1jeyA4	1j1dB0
1de4C3	1jb0F0	1lriA0	1ezjA2	1hp800	1l5oA1	1m7lA0
1rdr03	1gvnA0	1n3lA2	1dowA1	1heiA3	1jj210	1ik9A2
1ja1A3	1eijA0	1rv1A0	1fxkA0	1nvvS2	1jjsA0	1jy2N0
1f5nA1	1i27A0	1n5uA1	1gaxA5	1b79A0	1jj2O1	1jocA1
1f81A0	1ois01	1bvp11	1k3eA1	1iieA0	1r8eA2	1no4A0
1k3yA2	1gvdA0	1hq1A0	1k92A3	1sknP0	1kg2A1	1a2xB0
1nk4A2	1v54H0	5reqB3	1cpy02	1dj8A0	1etoA0	1g1jA0
1m0kA0	1pjr02	1r8sE1	1go3F2	1ihp02	1ceuA0	1jekA0
1ldfA0	1aa7A2	1qc7A0	1fehA2	1m61A2	1dvoA1	1e5wA4
1o2dA2	1erd00	1fkmA1	1g8mA1	1eyvA0	1e5rA2	1dd9A3
1jqnA1	1lfpA1	1c1kA2	1pq1A0	1jeqA5	1eg3A2	1ez3A0
1wpgA2	1mmsA2	1oohA0	1gk9A1	1h9fA0	1h3lA0	2e2aA0
1epwA2	1j09A5	1dekA2	2end00	1eyqA1	1h8eI0	1h7cA0
1jb0A0	1ji8A2	1iioA0	1pmi02	1qqvA0	1lx8A0	1chuA3
1g7dA0	1jj2G0	1dc1A2	1v54D0	1r8sE2	1j1vA0	1fjgT0
1pd7A0	1bob03	1kdxA0	1ae9A0	2dpmA2	1mz9A0	1hx1B0
1qoyA0	1e3pA2	1h5wA1	1rss00	1a9xA4	1ehs00	2a3dA0
1lxa02	1ng6A2	1dw9A1	1mw9X2	1qmgA2	1a9100	1hf8A2
1nogA0	1a6q02	1bjt03	1lvk05	1c05A1	2erl00	1nafA2
1p7tA4	1uzcA0	1anv01	1dxrC1	1gteA2	2spcA0	1m62A0
1jadA0	1dp3A0	1zymA2	2vsgA2	1e7uA5	1ecmA0	1kblA5
1jb0L0	1dciA2	1gkmA1	1guxA0	1m0wA3	1mixA1	1c5a00
1bg1A1	1a3qA2	1bgvA3	1pjr04	1pp9F0	1lre00	1gs9A0
1fioA0	1jj2V2	1a8rA1	1cpo00	1tbaA0	1r12A1	1vls00
2ilk01	1f1eA0	1ail00	1a6m00	1du2A0	1rzhL1	2hmqA0
1lb3A0	1qrvA0	1msk02	1wer02	1m1qA0	1m1nB4	1ei7A0
1dkxA2	1rlr01	1mw9X4	1wer01	1h8eD3	1mc2A0	1m56C2

续表

1mtyG1	1c3cA3	1adeA2	1rdqE1	1j09A2	1aa7A1	1ls1A1
1gnlA1	1i19A5	1kg2A2	1gwuA1	1tw6A0	1e85A0	3fapB0
1oaoA3	1on2A2	2tct02	3lzt00	1f6vA0	1is2A3	2a0b00
1fs7A1	1imxA0	1m9xC0	1ugpB1	1l0iA0	1qgiA2	1avoB0
1fpoA2	1lbu01	1u4gA2	1jboA0	1irqA0	1efyA1	1c17M0
1mtyG2	1fk5A0	1rlr02	1colA0	1bmtA1	1vns01	1dowA2
1jw2A0	1bea00	1cipA2	1f0lA2	1fr2A0	2lisA0	1f1mA0
1fs1A0	1a31A4	1d5tA1	1abv00	1a8o00	1i5pA2	1he1A0
1ib2A0	1is1A1	1bkrA0	1lw9A0	1iuqA1	1gw300	1h6gA2
1qsaA1	1m15A1	1gwuA2	1bgf00	1jj2O2	1aq5A0	1nzeA0
1hz4A0	1cmbA0	1p80A2	1cy5A0	1qsaA2	1gdtA2	1jr8A0
1n0qA0	1nk4A4	1tfe02	1jqiA1	1jhgA0	1fzcA0	1l3pA0
1bpoA2	1ci4A0	1el6A1	1tx4A0	1js8A1	1afoA0	1o3uA0
1v54E0	1ji7A0	1ezvH0	1kp8A1	16vpA0	1hgvA0	1gkzA2
1epuA3	1c20A0	1dtoA1	1fcyA0	1tbbA0	1vpc00	1h8eA3
1hs6A3	1go3F1	1sesA1	1b25A2	1mswD1	1pp9D1	1qjaA0
1qnf02	1ed1A0	1g4yB0	1ah700	1daqA0	1n7sA0	1gkmA2
1elkA0	1bpyA1	1h8eG1	1qnf03	1e3aA2	1avoA0	1v54A0
1ld8A0	1n62A2	1m56B1	1iomA1	1qhdA2	1cw5A0	1o82A0
1ho8A2	1f44A2	1nh2B0	5csmA0	1uvjA4	1dp5B0	1f8nA5
1l5jA1	1iipA2	1fpoA1	1b25A3	1uzeA2	1byyA0	1cii01
1oxjA2	1m6yA2	1vib00	1uby00	1r1hA2	1m56D0	1a5t03
1h2vC1	1af701	1ixmA1	1sig00	1u0eA3	1czqA0	1pp9C0
1k8kG0	1lvk03	1hwxA1	1vns02	1gk9B3	1jcdA0	1hbnA3
1kpsB0	1qq5A2	1gk9A2	1n1bA2	1q79A3	1tiiC0	1fgjA2
1pprM1	1kgqA1	1cxzB0	1mxrA0	1e3mA3	1ezvG0	1abz00
1kwfA0	1dk8A2	1bha00	1jfbA0	1gx5A3	1be3K0	1nvvS1
1n1bA1	2trcP2	1grj01	1d2vC0	1i5jA0	1pfiA0	1l6eA0
1gxmA0	1bvp13	1nh2D1	1ubkL0	1g8qA0	1ltsC0	1dbhA1
1dl2A0	1qazA0	1n7oA2				
2. 以 β 折叠为主						
1h8pA1	1amuA3	1jiwI0	1k5nA2	1lm8V2	1sdwA1	1h6lA0
1bx700	1mtpA2	1whi00	1f8nA1	1ejfA0	1ig0A2	1pjxA0
1g1tA2	4ubpC1	1oxdA0	1qhvA0	1acc04	1odmA0	1k32A1
1extA1	1nteA0	2por00	1svb01	1h6fA0	1nlqA0	1ofzA0
1lml03	3ezmA0	1a0sP0	1rqwA0	1e2wA1	1oh4A0	1fwxA1
1f94A0	1ospO1	1gweA0	1nxmA0	1g4mA1	1gwmA0	1a12A0
1c2aA1	1mkcA0	1jb7B0	1dmhA0	1f00I1	1jopA0	1k3iA2
1elvA3	1d09B2	2sli03	1gff10	1kyfA1	1nqjA0	1utcA0
1jsdA2	1bkb01	1befA1	1vpsA0	1p5vA2	1f35A0	1h2wA2
1lpbA0	1g2bA0	1h8eD1	1khxA0	1f00I2	1g6gA0	1g72A0
1mkkA0	1dj7B0	1pqhA0	1lb6A0	1n12A0	1c3gA1	1qksA2

续表

1jhfA1	1vie00	1iw7D4	1n7oA3	1d5rA2	1k3wA1	1k7iA1
1a7i00	1lvk01	1n10A1	1cq3A0	1h8lA2	1ni5A4	1qreA0
4ubpB0	1fx7A3	1qb5D0	1p35A0	1dceA2	1nc7A0	1l0sA0
1kmxA0	1kq1A0	1afp00	1wkt00	1d2oA1	1nwbA0	1k5cA0
1be3I0	1m1fA0	1c4qA0	1bhu00	1e42A1	1js8A2	1qq1A0
1igrA2	1bco02	2sns00	1ok0A0	1i31A1	1gwyA0	1dbgA0
1exkA0	1mhnA0	1oxxK3	2hft01	1hx0A2	1p2zA4	1ezgA0
1ep3B3	1igqA0	3chbD0	1gtfA0	1i82A0	4htcI0	1ofeB4
1jhnA2	1khcA1	1jb3A0	1o6sB0	1im3D0	1mw9X3	1hf2A2
1goiA2	1m9sA3	1c9oA0	1uowA0	1d7bA0	1euwA0	1ayl02
1mvfD0	1ib8A2	1iw7C6	1f86A0	1lyqA0	1kmtA0	1p2zA1
1ahl00	1lplA0	1x8pA0	1mfmA0	1gyvA0	1gpr00	1ois02
1n2fA1	1jj2S0	1luqA0	1noa00	1lmiA0	1jz7A5	1lktA0
1rkd01	1qs1A1	1ei5A2	1c7sA1	1l6pA0	1gpiA0	1preA3
1rb900	1uscA0	1pbyA1	1jz7A2	1ifrA0	1gp0A0	1gppA0
1ospO2	1gk9B2	1qjpA0	1a3qA3	1o75A3	1wpgA1	4dpvZ0
1mknA0	1iq8A4	1k3bA0	1svb04	1n67A1	1f3lA2	1g9mG0
1i5hW0	1inlA2	1kmoA2	1p5vA1	1n67A2	1uw6A0	1bdfA2
1qf8A2	1iw7C4	1h8eA1	4kbpA1	1o75A2	1njhA0	1kmoA1
1jj2Y0	1fjrA1	1qd6C0	1sfdA0	1p5vB0	1iw7D3	1dqcA0
1pfvA1	1jjdA0	1dfuP0	1sluA0	1kzqA1	1lnzA1	1hxrA0
1nh2C0	1jj2T0	1h9dB0	1bg1A2	1mkfA1	1pu5A0	1g9gA2
1dkgA2	1pq7A1	1qwzA0	1g4mA2	1mkfA2	1jmaA0	1p7tA2
1auuA0	1a1x00	1iw7C1	1soxA3	1qqp10	1tul00	1fjrA2
1jubA2	1i71A0	1p6vA0	1b4rA0	1dg6A0	1siiA3	1g8lA3
1g3p01	1krhA2	1jeyA2	1aohA0	1qhdA1	1rg8A0	1dtoA2
1unqA0	1e43A2	1c5eA0	1ayoA0	1h4gA0	1b2pA0	1ik9A1
1c0mA2	1e0tA3	4bcl00	1f0lA3	1nls00	1ciy02	1a8h02
1pcfA0	1bd0A2	1jk4A0	1g87A2	1a34A0	1vmoA0	1b12A2
1qldA0	2eng00	1v54F0	1eaqA0	1pgs01	1ouwA0	1jbiA0
1g31A0	1pfbA0	1h8eH0	1dqiA0	1acc02	1jm1A0	
1lml04	1oewA1	1h4aX1	1amx00	1od3A0	1hxn00	
1qz5A2	1awqA0	2bbkL0	1who00	2arcA0	1tl2A0	
1qntA1	1is2A2	1dkxA1	1nepA0	1nziA1	3sil00	
3. α-β 结构						
1acc01	1mml01	1vfyA0	1mpgA1	1nrwA2	1bfd02	1hfeL3
1rlr04	1igd00	1n62C1	1kyfA2	1nynA0	1amuA1	1vsrA0
1lyvA0	1fmtA2	1kvdB0	1m3qA1	1lfpA2	1o08A1	1qhkA0
1gpc00	1g9zA0	1xvaA1	1p5dX4	1gtdA0	1fiuA0	1di6A0
1bjt05	1bcpB1	1dszA0	1qmhA2	1mopA2	1g7sA3	1eq6A0
1jsdA1	1itxA2	1ecrA2	1qgiA1	1j8bA0	1crzA1	1cbf01
1s5dA0	1qddA0	1b7yB1	1svb03	1fjgP0	1jr2A1	1tdj03

续表

1jc9A1	1jetA3	1q9bA0	1kkoA1	1cliA1	1o4wA0	1khdA2
1g38A2	1jatA0	1efnB0	1mxtA2	1opd00	1c8bA0	1iu8A0
1dciA1	1mj4A0	1kgdA2	1preA2	1ejgA0	1em8A0	1i12A0
1efyA2	1vh5A0	1yua01	1aop01	1ofuA2	1j23A0	1c3pA0
1qxyA0	1dy5A0	1vcc00	1cxqA0	1pxwA0	1fyxA0	1ekjA0
2fokA1	1ok7A1	1svb02	1kpf00	1ufyA0	1eexB0	1c96A2
1rlr03	1u4gA1	1iqzA0	1gd0A0	1iv3A0	1kjnA0	1ooyA1
1hozA0	1nkiA0	1tig00	1kptA0	1bdfA1	1mgpA1	1cjyA2
1ayl03	1msk01	1qmeA3	1nwzA0	1dtjA0	1o0uA2	1hynP0
4uagA3	1lugA0	1ba102	1oacA1	1vhh00	1t15A1	1wpgA3
1p2zA3	1by200	1poiA2	1q79A2	1ctf00	1jeyA1	1jj2L0
1vjjA2	1bm800	1qf6A3	1n62C3	1f46A0	1l5oA2	1iuqA2
1fjjA0	1r4uA0	1b7yB4	1i2kA1	1jg5A0	1fuiA1	1qhlA0
1apj00	1n62A1	1ptq00	1b66A0	1ji8A1	1qopB1	1e19A0
1aol00	1htoA2	1r0rI0	1mtpA1	1jj2B2	1k7cA0	1e3mA1
1avqA0	1tif00	1i3jA2	1c96A1	1iq4A0	1o7jA1	1uehA0
1ugpA0	1ogwA0	1cqmA0	1k5nA1	1jtgB1	1d4oA0	4uagA2
1d0cA1	1acc03	1scjB0	1lkkA0	1jyaA0	1e58A0	1j9jA0
1musA2	3seb02	1cc8A0	3grs03	1jb0D0	16pk01	1ihnA0
1bob01	2sak00	1nh8A3	1n62C2	1k0rA4	16pk02	1l1sA0
1br901	1bmlC3	1d09B1	1qz5A1	1seiA2	1vl1A0	1o6dA0
1o7nA1	1l4dB0	1npk00	1dt9A2	1dk0A0	1duvG1	1r9wA0
1hbnA1	1ip9A0	1gk8A1	1fjgK0	1usmA0	1g8mA4	1dzfA1
1uok02	1mg4A0	1n0uA6	1e4fT3	1l0wA3	2rslA0	1t0fA1
1soxA2	1oeyJ0	1mla01	1jj2M0	1b4bA0	1dozA1	1a79A2
1co4A0	1n6zA0	1lk5A2	1e3mA2	1bjt04	1ofuA1	1bx4A1
1d0cA2	3eipA0	1gx5A2	1p90A0	1iba00	1cfzA0	1l2mA0
1g3p02	1grj02	1l3kA1	1l9vA2	1f7uA3	1qtnA0	1dd9A2
1dj7A0	1bkf00	1aye01	3nul00	1j98A0	2pth00	1dmgA0
1i4jA0	1lml01	1cg2A2	1f5mA0	1f0xA4	1li4A1	1h8eG2
1hbnC0	1p32A0	1nk4A3	1ifqA0	1seiA1	1b8oA0	1e8cA1
1p2zA3	1dm9A0	1b7yB6	1nrjA0	1lvk06	1ex1A2	1o1xA0
1vjjA2	1ewnA0	1b3tA0	1h3qA0	1mszA0	1m0wA4	1e0tA1
1fjjA0	1gqzA1	1eayC0	1l3lA1	1bxyA0	1l7dA2	1ko7A1
1apj00	1uvqA1	1dqaA3	1g6oA1	1k8kD1	1hfeL1	1bhtA1
1aol00	1qcsA2	1kp6A0	1otgA0	1kjqA2	1f8yA0	1fx2A0
1avqA0	1qlmA1	1hbnA2	1i19A1	1bjt01	1g66A0	1kid00
1ugpA0	1e0gA0	1ftrA1	1kjqA3	1gyfA0	1jflA1	1ecrA1
1d0cA1	1fd3A0	1f9yA0	1fviA3	1m0wA5	1mv8A3	1ibvB0
1musA2	1e53A0	1dj0A2	1tfe01	1fviA1	1epuA4	1kblA3
1bob01	1e44B0	1q79A1	1d5tA2	2hgsA1	1vimA0	1b7yB3
1br901	1oqjA0	1fjgJ0	1icxA0	1i50A4	1mvlA0	1mxtA1

续表

1o7nA1	1jhdA2	1ii9A2	1lvk02	1jx4A4	1o2dA1	1eyqA2
1hbnA1	1eg7A3	1ekrA0	1ka1A1	1e4fT1	1m1nA1	1jkeA0
1uok02	1qynA0	1regX0	1nm8A2	1div01	1fjgB1	1a9xB1
1soxA2	1div02	1dj0A1	1gpeA3	1adn00	1jg7A1	1de4C2
1co4A0	1jj2W0	1qd1A2	1uylA0	1m4jA0	1fsgA0	1vi4A0
1d0cA2	1molA0	1f3vA0	1qqqA0	1m2dA0	1gnlA3	1j5uA1
1g3p02	1ugiA0	1gmuA1	1m15A2	1qnxA0	1epuA1	1dl5A2
1dj7A0	1lniA0	1h72C2	1qveA0	1nrzA0	1jbeA0	1b25A1
1i4jA0	1siiA1	1kr4A0	1t1dA0	1ox0A1	1dz3A0	1ako00
1f3mA0	1oh0A0	1kn6A0	191400	1kjqA1	1mx3A1	1mqoA0
1gh9A0	1f8nA4	1in0A1	1gccA0	1ohtA0	1qczA0	1gk9B1
1epuA2	1eejA1	1n0uA4	1cmiA0	1dc1A1	1h05A0	1uteA0
1pfo01	1jj220	1kkhA2	1h4xA0	1nox00	1eiwA0	1a6q01
1hyoA2	1iq8A3	1lq9A0	1l8bA0	1p5dX1	1usgA1	2pvaA0
1k4iA0	1o54A1	1i1gA2	1d1rA0	1p6oA0	1dfmA0	1b65A0
1k2fA2	1ewfA1	1lxjA0	1bo1A1	1ixh01	1m0dA0	1jl0A0
1uvjA1	1ewfA2	1m1gA1	1bo1A2	1c1dA1	1g8mA2	1hq0A0
1fs7A2	1i2kA2	1ivzA0	1pp9A1	1cby00	1ka1A2	1uf5A0
1g71A1	1fuiA3	1nxiA0	1ixmA2	3pviA0	1ni9A2	1i1qA0
1ospO3	1iedA0	1lfpA3	1fc6A1	1lam01	1gmxA0	1hp1A1
1i50F0	1c96A4	1udvA0	1k32A4	1ojrA0	1fuiA2	1g6sA1
1v7rA0	1bd0A1	1hc7A3	1fjgS0	1eu1A2	1o04A2	1qmhA1
1dbuA0	1h7bA0	1di2A0	1v0wA1	1pn0A2	1chmA1	1iz5A0
1ccwB2	1jyhA0	1puc00	1d9nA0	1m3kA2	1mla02	1g61A0
1dd9A1	1c8zA0	1gk8I0	1go4A0	1o7jA2	1czaN2	1io0A0
1o22A0	1d8hA0	1rdqE2	1ghhA0	1krhA3	1c7kA0	1igrA1
1mzgA0	1eyeA0	1bpyA4	1ew4A0	1mw9X1	1nlnA0	1j0pA0
1iq8A2	1lucA0	1nz0A0	1evlA1	1fp2A1	1uq5A1	1a31A3
1gd8A0	1oc7A0	1orc00	1cbf02	1jkxA0	1kmvA0	1jsdB0
1rl6A1	1s2wA0	1u94A2	1dt9A1	1chd00	1adeA1	1i24A2
1dzfA2	1n55A0	1kp8A2	1qf6A2	1gci00	1ayl01	1p2zA2
1ni5A3	1i1wA0	4ubpA0	1qd1A1	1c4kA1	1e8gA3	1lm4A0
1iw7D2	1us0A0	1gpmA3	1e7uA4	1oi7A2	4eugA0	1rzhH2
1l6rA2	1r5yA0	1cii02	1prxA2	1jf8A0	1aua01	1eu1A3
1iw7C2	1gk8A2	1e42A2	1qlmA2	1ccwA0	1uch00	1ckv00
1iw7C3	1kkoA2	1noyA1	1dtdB0	1byi00	1g8tA0	1ckmA1
1iw7D1	1hzyA0	2sicI0	1pz4A0	1oboA0	1d02A0	1me3A0
1jh6A0	1muwA0	1lc0A2	1nwaA0	1mf7A0	1sx5A0	1a73A0
1lc5A1	1o1zA0	1n62B2	1ev0A0	1pfkA1	1o04A1	1jetA2
1el6A3	1r3sA0	1ay7B0	1bx4A2	1pfkA2	1dmuA0	1uxy01
1jj2H0	1v93A0	1jakA2	1hdhA2	1pdo00	1rtqA0	1hztA0
1j3aA0	1reqA1	1msc00	1pfo02	1kqpA0	1lc5A2	1i40A0

续表

1nd4A0	1ccwB1	1gtkA3	1dhn00	1bjt02	1m40A0	1c4kA4
1ucsA0	1ex1A1	1llmC1	1fjgC2	1ubkS1	1hqsA0	1t2dA2
1e8pA0	1lt8A0	1mgtA1	1tolA2	1lqtA1	1hdhA1	1g55A1
1d0cA3	1eexA0	1bbg00	1dw9A2	1eu1A1	1d3vA0	1lml02
1qpoA1	1h16A0	1dq3A2	1gaxA4	1i2aA2	1poiB0	1dkgA1
1brwA1	1c0pA2	1i2aA1	1mgpA3	1evlA2	1b4uB0	1anv02
1fm0E0	1cseI0	1josA0	1nr3A0	1nf9A0	1f9vA0	2vsgA1
1n62B1	1dxjA2	1amuA4	1jrmA0	1p5fA0	1rlzA0	1adeA3
1jj2K2	1d5tA3	3proC1	1nijA2	1v1rB1	1kekA3	1ikpA3
	1aho00	1ib8A1	1g2rA0	1hqkA0	1a3aA0	1n8kA1
4. Few secondary structures						
1bg503	1ckmA3	2bbvD0	1f8nA2	1ubkS2	1cwxA0	1g9gA3
1ba305	1qqp40	1tvs00	1j8eA0	1iuaA0	1devB0	1rdr01
4mt200	1v54J0	1nkzA0	1g6xA0	1ibvA0	1j34C0	1qa4A0
1jfwA0	1h7dA0	1aml00	1kvdA0	1b8zA0	1hykA0	1g3jB0
1agg00	1v54I0	1aw600	1i4oC0	1jc9A2	1b8xA3	1fjgM2
1fre00	1v54G0	1gotG0	1a92A0	1rzhH1	1kekA4	1jpwD0
1v54L0	2pspA1	1wdcA0	1hfeS0	1cl1A1	1kekA7	1pbyC0
1v54K0	1h59B0	1nkpA0	1ioj00	2prgC0	1icfI0	1jj2A3
1a1tA0	1hy9A0	1bct00	1oe9A6	1g1xC0	1gp8A0	1jj2B3
1j2lA0	1jo6A0	1ebdC0	1hnr00	1aoo00	1f02T0	1jeyA3
1fleI0	1jouA0	2ilk02	1gpeA2	1d6gA0	1fjgN0	1fqjC0
1aym40	1g72B0	1jsuC0	1inp01	1cf4B0	1edxA0	1jj2K1
1v54M0	1olgA0	1f8nA3	1uq5A2	1b35D0	1qojA0	

说明：附表 4 中使用的是 CATH 数据库六字母代码

源自：http://www.biochem.ucl.ac.uk/bsm/cath/class.html

附表 5　蛋白质结构域序列家族层样本集

1. 以 α 螺旋为主							
1oaiA0	1ji8A2	1g4yR2	2tct02	1ie9A0	1jj2O2	1quuA2	1bg1A1
1cuk03	1jj2G0	1m45A1	1jt6A2	1t7rA0	1irqA0	1g8xA6	1fioA0
1a5t02	1bob03	1m45A2	1m9xC0	1nq7A0	2cpgA0	1g8xA7	1dn1B0
1g41A2	1e3pA2	1k94A0	1em9A0	1g2nA0	1bazA0	1hciA1	1nk4A2
1e94E2	1ng6A2	1k9uA0	1p7nA0	1oshA0	1mntA0	1hciA4	1x9mA2
1im2A2	1a6q02	1top02	1eia01	1qkmA0	1bmtA1	1ez3A0	1kfd02
1bvsA3	1uzcA0	1ahr00	1g03A0	1fm6D0	1qsaA2	1g73A0	1m0kA0
1g4aE2	1dp3A0	1alvA0	1u4gA2	1pq9A0	1jhgA0	1hs7A0	1c8rA0
1kyiS2	1dciA2	1g8iA2	1keiA2	1pzlA0	1js8A1	2e2aA0	1e12A0
1ixsA0	1ef8A1	1mr8A0	1trlA0	1s9pA0	1bt3A0	1h7cA0	1h2sA0
1g3iW2	1mj3A2	1e8aA0	1rlr02	1n46A0	16vpA0	1lrzA3	1l9hA0
1dv0A0	1a3qA2	1sra00	1cipA2	1ovlA0	1tbbA0	1qsdA0	1ldfA0
1ifyA0	1nfkA2	1wdcB1	1azsC2	1pk5A0	1mswD1	1chuA3	1j4nA0
1otrA0	1jj2V2	1wdcB2	1d5tA1	1r1kD0	1daqA0	1qlaA3	1rc2A0
1aua02	1f1eA0	1wdcC1	1f8rA3	1b25A2	1e3aA2	1nekA3	1o2dA2
1efuB1	1b67A0	2scpA0	1vg0A3	1aorA2	1qhdA2	1kf6A3	1jq5A2
1gab00	1n1jA0	1oe9B1	1bkrA0	1ah700	1uvjA4	1fjgT0	1sg6A2
1mu5A2	1n1jB0	1auiB0	1pa7A0	1ak000	1uzeA2	1hx1B0	1ujnA2
1fjgM1	1tzyA0	1uhnA0	1bhdA0	1ca101	1i1iP3	2a3dA0	1jqnA1
1c1kA1	1tzyB0	2cblA2	1sh5A1	1qnf03	1j36A2	1hf8A2	1wpgA2
1in4A3	1tzyC0	1dtlA2	1aoa01	1iomA1	1r1hA2	1nafA2	1epwA2
1nvmA2	1tzyD0	1a4pA0	1aoa02	1csh01	1u0eA3	1m62A0	1jb0A0
1d2nA2	1kx5A0	1bg1A3	1h67A0	1k3pA2	1gk9B3	1ecmA0	1g7dA0
1fnnA1	1kx5D0	1djxA1	1v5kA0	1o7xA1	1fm2B2	1mixA1	1e91A0
1njgB1	1tafA0	1djxB1	1gwuA2	5csmA0	1q79A3	1h4rA2	1qoyA0
1r6bX3	1tafB0	1hqvA0	1oafA2	1b25A3	1e3mA3	1hb6A0	1lxa02
1r6bX5	1h3oB0	2sas00	1jdrA2	1aorA3	1gx5A3	1hbkA0	1nogA0
1g41A3	1m19B0	1k90E1	1lycA2	1uby00	1i5jA0	1gg3A3	1nigA0
1jr3D3	1eqzD0	1ij5A3	1itkA2	1sig00	1g8qA0	1kblA5	1p7tA4
1iqpA2	1bh9A0	1s26D1	1p80A2	1vns02	1rykA0	1lre00	1jadA0
1e32A4	1bh9B0	1c07A0	1pq1A0	1qi9A0	1iw7E0	1r12A1	1jb0L0
1jqjD3	1jfiA0	1dguA0	1ohuA0	1up8A0	1iyrA0	1rzhL1	2ilk01
1j09A4	1jfiB0	1eh200	1bxlA0	1n1bA2	1koyA0	1rzhL2	1bgc00
1g8pA2	1qrvA0	1fi5A0	1ddbA0	1ezfA0	1mkiA2	1rzhM1	1alu00
1e9rA2	1cktA0	1fi6A0	1f16A0	5eau02	1ng6A1	1rzhM2	1eerA0
1h72C3	1gt0D0	1h8bA0	1k3kA0	1kiyA0	1o0wA1	1l9bM1	1m48A0
1qamA2	1cg7A0	1j7qA0	1lxl00	1di1A0	1jfzA0	1m1nB4	1n1fA0

续表

1i4wA2	1k99A0	1jbaA1	1q59A0	1ps1A0	1nxuA1	1mioB3	1d9cA0
1jb0F0	1l8yA0	1jfjA1	1gk9A1	1mxrA0	1mi1A2	1mc2A0	1huw00
1gvnA0	2lefA0	1jfjA2	1fm2A0	1mtyB0	1pujA2	1g4iA0	1lki00
1eijA0	1rlr01	1jfkA2	1kehA2	1mtyD0	1jeyA4	1le6A0	1m4rA0
1i27A0	1c3cA3	1nyaA0	2end00	1kgnA0	1jeyB4	1poc00	1hziA0
1fp2A2	1furA3	1qjtA0	1pmi02	1h0oA0	1l5oA1	1aa7A1	1b5l00
1dp7P0	1k7wA3	1oohA0	1v54D0	1afrA0	1j33A1	1c5a00	1au1A0
1hw1A1	1dofA3	1ow4A0	1ae9A0	1jfbA0	1jj210	1e85A0	1eteA0
1kgsA2	1q5nA3	1r5rA0	1f44A1	1n40A0	1h8eI0	256bA0	1evsA0
1gvjA0	1j3uA3	1dqeA0	1a0p02	1io7A0	1jjsA0	2ccyA0	1scfA0
1b6a02	1jswA3	1c3yA0	1aihA0	1qmqA0	1jj2O1	1cpq00	1f6fA0
1in4A2	1i19A5	1dekA2	1floA2	1po5A0	1r8eA2	1mqvA0	1ax800
1on2A1	1e8gA4	1iioA0	1rss00	1bu7A0	1r8dA0	1bbhA0	1cnt10
1l3lA2	1on2A2	1dc1A2	1iqvA0	1lfkA0	1lx8A0	1s05A0	1hulA0
1r1tA0	1fx7A2	1lriA0	1mw9X2	1n97A0	1pm6A0	1gs9A0	1i1rB0
1b9mA1	1ddnA2	1n3lA2	1i7dA2	1odoA0	1kg2A1	1aep00	2gmfA0
1xgsA2	1imxA0	1i6mA2	1gkuB6	1q5dA0	1ornA2	1vls00	1lqsL1
3htsB0	1zeiA0	1h3fA2	1lvk05	1u13A0	1mpgA3	2hmqA0	1f45B0
1jhfA2	3lriA0	1jilA2	1lkxA4	1izoA0	1m3qA2	1ei7A0	1ga3A0
1ku3A0	1lbu01	1rv1A0	1dxrC1	1cpt00	1keaA1	1cgmE0	1jli00
1mgtA2	1l6jA1	1ycqA0	1dxrC2	1d2vC0	1etoA0	1m56C2	1lb3A0
1hw5A2	1fk5A0	1n5uA1	2vsgA2	1q4gA2	1fipA0	1fftC0	1o9rA0
1j75A0	1hyp00	1n5uA2	1vsgA2	1ubkL0	1ntcA0	1ls1A1	1jigA0
1t0fA2	1l6hA0	1n5uA3	1guxA0	1cc1L0	1ceuA0	1j8mF1	1lkoA1
1bc8C0	1bea00	1n5uA4	1guxB0	1b4uA0	1dvoA1	1fts01	1nfvA0
1opc00	1hssA0	1n5uA5	1h1rB1	1ft5A0	1e5rA2	3fapB0	1eumA0
1bm9A0	1a31A4	1n5uA6	1h1rB2	1e7lA2	1eg3A2	2a0b00	1ji4A0
1d3yA1	1is1A1	1kxpD1	1aisB1	1a6201	1h3lA0	1c02A0	1jgcA0
1fc3A0	1dd5A1	1kxpD2	1aisB2	1jeqA5	1j1vA0	1i5nA0	1dkxA2
1fnnA3	1m15A1	1kxpD3	1f5qB1	1h9fA0	1l8qA3	1avoB0	1gnlA1
1fx7A1	1qh4A1	1kxpD4	1f5qB2	1pfvA3	1mz9A0	1c17M0	1gnlA2
1fy7A1	1cmbA0	1kxpD5	1jkw01	1a8h03	1debA0	1dowA2	1oaoA3
1gxqA0	1nk4A4	1kdxA0	1jkw02	1h3nA4	1gw300	1qkrA0	1fs7A1
1lddA0	1jx4A3	1h5wA1	1c9bA1	1f7uA2	1aq5A0	1h6gA1	1oahA2
1fseA0	1cuk02	1bvp11	1c9bA2	1ffyA3	1gdtA2	1f1mA0	1mtyG1
1q06A0	1a7602	1hq1A0	1h4lD0	1iq0A3	1fzcA0	1he1A0	1fpoA2
1pueE0	1tfr02	1dulA0	1g3nC1	1ile01	1fzcB3	1hy5A0	1mtyG2
1qbjA0	1mswD5	1qb2A0	1g3nC2	1gaxA2	1fzcC3	1h6gA2	1jw2A0
1sfe02	1x9mA4	1dw9A1	1bu2A2	1musA3	1fzgD0	1nzeA0	1fs1A0
1je8A0	1bpyA2	1lmb30	1ugpB1	1lm8V1	1m1jA0	1jr8A0	1nexB1
1lvaA1	1bgxT2	1b0nA0	2ahjB1	1c75A0	1m1jB3	1oqcA0	1ldkE0
1lvaA2	1jmsA2	1e3oC1	1pjr04	1ctj00	1m1jC3	1l3pA0	1ib2A0

续表

1lvaA3	1exnA1	1r6900	1qhhC0	1i8oA0	1lt9E3	1nlxA0	1jdhA0
1lvaA4	1hjp02	1jftA1	1uaaA4	1gu2A0	1lt9F3	1o3uA0	1ee4A0
1bjaA0	1iw7D5	1adr00	1cpo00	1e29A0	1lwuA0	1k04A2	1oyzA0
1a04A2	1dgsA5	1neq00	1a6m00	1ycc00	1lwuB3	1jogA0	1b3uA0
1ixcA1	1dgsA6	1uxc00	1h97A0	1m70A1	1lwuC3	1knyA2	1ibrB0
1kyzA1	1bvsA2	1bjt03	1irdA0	1m70A2	1afoA0	1is2A3	1qgrA0
1mkmA1	1lb2B0	1bgw03	1irdB0	1c5200	1hgvA0	1is2A4	1lrv00
2irfG0	1b22A0	1ab403	1eca00	1qksA1	1ifi00	1jqiA3	1ho8A1
1k78A1	1doqA0	1anv01	1jf3A0	1ql3A0	1vpc00	1ivhA3	1qbkB0
1k78A2	1kftA0	1zymA2	3sdhA0	1h32A2	1pp9D1	1gkzA2	1qsaA1
1qo0D2	1ci4A0	1gkmA1	1b0b00	1h32B0	1n7sA0	1jm6A2	1hz4A0
1bia01	1ji7A0	1c3cA1	1kr7A0	1cxc00	1n7sB0	1qgiA2	1na3A0
1hsjA3	1kw4A0	1furA1	1q1fA0	351c00	1n7sC0	1chkA1	1elwA0
1jgsA1	1oxjA1	1k7wA1	1dlwA0	1iqcA1	1n7sD0	1efyA1	1hxiA0
2fokA2	1b4fA0	1dofA2	1cg5A0	1iqcA2	1gl2A0	1gs0A1	1na0A0
1b4aA1	1dxsA0	1q5nA1	1cg5B0	1dw0A0	1gl2C0	1vns01	1hh8A0
1o7fA2	1bqv00	1bgvA3	1it2A0	1pp9D2	1gl2D0	1d2tA0	1ihgA2
6paxA2	1uqvA0	1a8rA1	1mba00	1h1oA1	1jthB0	2lisA0	1elrA0
1cf7A0	1c20A0	1is8A1	1s69A0	1h1oA2	1nhlA0	1gakA0	1klxA0
1cf7B0	1ig6A0	1tfe02	1gdj00	1nirA1	1sfcD0	1h8eA3	1nznA0
1ft9A2	1kkxA0	1el6A1	1cqxA1	1mg2D0	1l4aA0	1i5pA2	1fchA0
1hstA0	1go3F1	1ail00	1ew6A0	1eb7A1	1l4aB0	1ciy01	1a1700
1iw7F3	1d8bA0	1a3200	1ngkA0	1cc500	1avoA0	1qjaA0	1b89A0
1repC1	1ed1A0	1d2dA0	1ash00	1diiC0	1cw5A0	1gkmA2	1kt0A3
1repC2	1hekA0	1ezvH0	1hlb00	1fcdC1	1dp5B0	1c3cA2	1qqeA0
1hkqA0	1a6s00	1l0lH0	1ithA0	1fcdC2	1byyA0	1furA2	1n0qA0
1ku9A1	1bpyA1	1dtoA1	1jboA0	1dvh00	1m56D0	1k7wA2	1n0rA0
1omiA3	1jmsA1	1sesA1	1jboB0	1gks00	1czqA0	1q5nA2	1bd800
1i1gA1	1nzpA0	1serA1	1b8dA0	1fgjA1	1llmC2	1v54A0	1k1aA0
1l0oC0	1n62A2	1g4yB0	1kn1A0	1i4mA0	1j2jB0	1ehkA0	1ot8A0
1q1hA0	1vlbA2	1r3jC0	1colA0	1il7A0	1gu4A0	1o82A0	1uohA0
1ka8A0	1f44A2	1m56C1	1cii03	2prp00	1piqA0	1f8nA5	1mj0A0
1aoy00	1a0p01	2occC1	1f0lA2	1ltzA0	1gd2E0	1lox01	1sw6A0
1bby00	1iipA2	1h8eG1	1wer02	1j8uA0	1jnmA0	1cii01	1ycsB1
1d5vA0	1m6yA2	1fs0G2	1nf1A1	1phzA0	1gk4A0	1a5t03	1iknD0
1d8jA0	1af701	1ohhG0	1wer01	1hp800	1ci6A0	1jr3A2	1ixvA0
1dpuA0	1lvk03	1m56B1	1nf1A2	1heiA3	1ci6B0	1jr3D2	1bi7B0
1fshA0	1oe9A3	2occB1	1rdqE1	1nvvS2	1a02F0	1jqjC2	1bpoA2
1g4dA0	1kk8A3	1fftB1	1gz8A2	1b79A0	1nwqA0	1pp9C0	1v54E0
1hks00	1lkxA2	1nh2B0	1fmk04	1iieA0	1dh3A0	1hbnA3	1epuA3
1iuyA0	1qq5A2	1nvpB0	1jksA2	1sknP0	1hjbA0	1hbnB2	1mqsA4
1j9iA0	1swvA2	1fpoA1	1p4oA2	1dj8A0	1hbwA0	1fgjA2	1hs6A3

续表

1p4wA0	1kgqA1	1nz6A0	1mp8A2	1eyqA1	1junA0	1abz00	1e7uA3
1p6rA0	1dk8A2	1bq000	1t46A2	1ihp02	1jcdA0	1jb0K0	1he8A3
1uhmA0	1fqiA2	1fafA0	1om1A2	1m61A2	1tiiC0	1nvvS1	1qnf02
1uhwA0	1htjF1	1vib00	1ia8A2	1a81E2	1ezvG0	1l6eA0	1np7A2
1ois01	1omwA2	1ixmA1	1uu3A1	1eyvA0	1l0lG0	1dbhA1	1dnpA2
1gvdA0	1agrE2	1hwxA1	1j1bA2	1q8cA0	1be3K0	1ki1B1	1elkA0
1gv2A2	2trcP2	1gk9A2	1mq4A1	1ey1A0	1pfiA0	1kz7A1	1eyhA0
1hcrA0	1b9xC2	1ajqA2	1csn02	1qqvA0	2ifo00	1foeA1	1ujkA0
1e3oC2	1bvp13	1cxzB0	1pme02	1ujsA0	1ltsC0	1by1A0	1hf8A1
1pufA0	1lvk04	1urfA0	1tkiA2	1r8sE2	1nkzB0	1f5xA0	1ld8A0
1pufB0	1oe9A4	1bha00	1vjyA2	2dpmA2	1lghB0	1iw0A0	1dceA1
2hddA0	1kk8A4	1grj01	1q8yA2	1a9xA4	1ezvI0	1j77A0	1ho8A2
1jggA0	1br2A4	1nh2D1	1o6yA2	1qmgA2	1pp9E1	1n45A0	1l5jA1
1k61A0	1jvr00	1df4A0	1f3mC2	1evyA2	1ezvE1	1j02A0	1oxjA2
1mh3A3	1kblA2	1qbzA0	1kobA2	1bg602	1f8vD0	1e6iA0	1h2vC1
2tct01	1vpu00	1mof00	1nxkA2	1dljA3	1f8vE0	1eqfA1	1h2vC2
1ignA1	1dk8A1	2eboA0	1gwuA1	1f0yA2	1fjkA0	1jspB0	1h2vC3
1ignA2	1fqiA1	1mg1A3	1oafA1	1i36A2	1svfA0	1eo0A0	1h6kA1
1mnmC0	1agrH1	1eboA0	1jdrA1	2pgd02	1g2cA0	2cblA1	1hu3A0
1au7A2	5reqB3	1favA0	1lycA1	1pgjA2	1g2cB0	1de4C3	1k8kG0
1bl0A1	1utg00	1nkd00	1itkA1	1c05A1	1avyA0	1dvkA0	1kpsB0
1bl0A2	1axn01	1joyA0	1itkA3	1gteA2	1avyB0	1qlaC0	1pprM1
1le8A0	1axn02	1mswD2	1abv00	1h7wD1	1ox3A0	1om2A0	1kwfA0
1b72A0	1axn03	1aroP2	3lzt00	1qlaB2	2pgd03	1rdr03	1h12A0
1tc3C0	1axn04	1b0nB0	1b9oA0	1kf6B2	1aa000	1khvA4	1ks8A0
1hlvA1	1a8a02	1mswD4	153l00	1e7uA5	1pgjA3	1khdA1	1nc5A0
1hlvA2	1n00A1	1eejA2	1qsaA3	1cjaA2	1ic2A0	1brwA3	1ayx00
1pb6A1	1n00A2	1g6uA0	1ltm02	1m0wA3	1j1dB0	2tpt01	1gai00
1jt6A1	1n00A3	1jj2U0	1dxjA1	2hgsA3	1j1dE0	1ja1A3	1clc02
1ic8A2	1n00A4	1ezjA1	1am7A0	1pp9F0	1m7lA0	1f20A2	1g9gA1
1i50J0	1m9iA7	1ezjA2	1iizA0	1ezvF0	1ik9A2	1ddgA2	1fp3A0
1lfb00	1r8sE1	1dowA1	1lw9A0	1tbaA0	1fu1A2	1f5nA1	1lf6A2
1gdtA3	1ku1A1	1fxkA0	1bgf00	1du2A0	1jy2N0	1f81A0	1h54A2
1ba500	1qc7A0	1gaxA5	1cy5A0	1m1qA0	1jy2O0	1l3eB0	1gxmA0
1bw500	1fkmA1	1k3eA1	1d2zA0	1q9iA2	1jy2P0	1k3yA2	1qqfA0
1ef4A0	1c1kA2	1k92A3	1d2zB0	1h8eD3	1jocA1	1k0mA2	1ld8B0
1fexA0	1l9lA0	1cpy02	3ygsP0	1j09A2	1no4A0	1e6bA2	1dceB0
1ftt00	1n69A0	1go3F2	1a1w00	1qtqA3	1a2xB0	1oe8A2	2sqcA1
1g2hA0	1m12A0	1fehA2	1ddf00	1tw6A0	1lp1A0	1b4pA2	2sqcA2
1irzA0	1nkl00	1g8mA1	1dgnA0	1jd5A0	1deeG0	1iyhA2	1hzfA0
1ityA0	1iomA2	1msk02	1e3yA0	1g73C0	4hb100	1jlvA2	1n4qB0
1iufA2	1csh02	1j6rA2	1ichA0	1f3hA0	1g1jA0	1m0uA2	1dl2A0

续表

1o4xA2	1a5902	1k7yA4	1n3kA0	1i3oE0	1jekA0	1aqwA2	1kktA0
2ezh00	1k3pA3	1mw9X4	1ngr00	1f6vA0	1fs0E2	1dugA2	1n7oA2
2ezk00	2pvbA0	1i7dA4	3crd00	1l0iA0	1e79H2	1n2aA2	1cb8A1
1v54H0	1exrA1	1gkuB8	1jqiA1	1af800	1e5wA4	1oyjA2	1j0mA1
1pjr02	1exrA2	1adeA2	1egdA1	1dnyA0	1ehs00	1eemA2	1qazA0
1uaaA2	1psrA0	1iweA2	1bucA1	1dv5A0	1a9100	1aw902	1n1bA1
1aa7A2	1qv0A0	1kg2A2	1ivhA1	1klpA0	2erl00	1gnwA2	5eau01
1erd00	1k8uA0	1ornA1	1tx4A0	1fr2A0	1dd9A3	1k0dA2	
1lfpA1	1g33A0	1mpgA2	1pbwA0	1a8o00	1eqnA3	2gsq02	
1k6yA1	1ggzA2	1m3qA3	1f7cA0	1eia02	2spcA0	1gwcA2	
1mmsA2	1ig5A0	1keaA2	1kp8A1	1d1dA2	1cunA1	1f2eA2	
1j09A5	1omrA1	1ngnA0	1a6dA1	1qrjB2	1cunA2	1nhyA2	
1irxA5	1omrA2	1lmzA0	1fcyA0	1iuqA1	1quuA1	1g7oA2	
2. 以 β 折叠为主							
1mml01	1kwgA1	1ad201	1pp9B1	1exmA1	1jj2K2	1kqfA3	2pvaA0
1har01	1v3hA0	1rdqE2	1pp9B2	1f5nA2	1jj2N0	2napA3	1b65A0
1igd00	1cnv00	1gz8A1	1ezvA1	1f6bA0	1es9A0	1aa603	1jl0A0
1pgx00	1jfxA0	1obdA1	1ezvA2	1htwA0	1ivnA0	1gmxA0	1i7bA0
1hz6A0	1qhoA1	1fmk03	1ezvB1	1n0wA0	1esc00	1rhs01	1mhmA0
1mhhE0	1ta3A0	1jksA1	1ezvB2	1nrjB0	1o7jA1	1rhs02	1hq0A0
1c9fA0	1h4pA0	1p4oA1	1hr6A1	1qf9A0	1d4oA0	1e0cA1	1uf5A0
1d4bA0	1jakA1	1fbnA1	1hr6A2	1ckeA0	1m2kA1	1e0cA2	1f89A0
1f2rI0	1c7sA3	1mp8A1	1hr6B2	1d2nA1	1q1aA1	1qb0A0	1emsA1
1n62A1	1clxA0	1o6lA2	1fjgS0	1fukA0	1bfd01	1hzmA0	1i1qA0
1czpA0	1eokA0	1t46A1	1v0wA1	1i2mA0	1j8fA1	1fuiA2	1k0eA0
1fm0D0	1fobA0	1om1A1	1v0wA2	1e6cA0	1zpdA2	1o04A2	1qdlA0
1oqqA0	1nar00	1ia8A1	1byrA0	1g16A0	1efvA1	1euhA2	1g6sA1
1krhA1	1hjxA1	1uu3A2	1d9nA0	1g5tA0	1poxA2	1o20A3	1g6sA2
1i7hA0	1ceo00	1opjA1	1qk9A0	1jbkA0	1pvdA2	1ez0A2	1ejdA1
1fehA1	1edt00	1j1bA1	1ub1A0	1kjwA2	1ovmA2	1ad3A2	1ejdA2
1ayfA0	1eswA0	1r0pA1	1go4A0	1kk1A1	1e58A0	1chmA1	1qmhA1
1doi00	1iv8A1	1blxA1	1ghhA0	1l4uA0	1h2eA0	1az901	1iz5A0
2pia03	1uwsA0	1mq4A2	1ew4A0	1q0uA0	1bif02	1mla02	1rwzA0
1l5pA0	7taa01	1csn01	1ekgA0	1d2mA2	1nd6A0	1czaN2	1plq00
1qlaB1	1aq0A0	1ia9A1	1evlA1	1ex7A2	1ihp01	1c7kA0	1axcA0
1fiqA1	1bf202	1pme01	1nnhA0	1mkyA1	16pk01	1eb6A0	1g61A0
1ni3A2	1j11A1	1r3cA1	1b8aA2	1mkyA2	16pk02	1jk3A0	1h70A0
1qf6A1	1ll7A1	1tkiA1	1g5hA1	1np6A1	1vl1A0	1kufA0	1jdw00
1f0zA0	1uok01	1vjyA1	1l0wA2	1q44A0	1fs5A0	1k7iA2	1g62A0
1jq4A0	2ebn00	1q8yA1	1e1oA2	1u94A1	1duvG1	1g12A0	1io0A0
1rwsA0	1gjwA1	1fvrA1	12asA0	1cp2A0	1duvG2	1iab00	1oznA0
1ryjA0	1h09A1	1j7lA1	1bia02	1svmA3	1ml4A1	1r1hA1	1ogqA0

续表

1htoA2	1h3gA2	1o6yA1	1h4vB1	1g8fA3	1ml4A2	1bkcA0	1p9aG0
1tif00	1nowA2	1phk01	1hc7A1	1gvnB0	1othA2	1i1iP2	1d0bA0
1ogwA0	1ua7A1	1pmnA1	1b7yA0	1ko7A2	1g8mA4	1j7nA1	1a4yA0
1euvB0	1egzA0	1ckiA1	1b7yB5	1nlfA0	1a9xA8	1j7nA4	1dceA3
1gnuA0	1m7xA2	1f3mC1	1sesA2	4tmkA0	1b93A0	1ck7A1	1jl5A0
1h4rA1	1pbgA0	1a0601	1atiA1	1j99A0	1c30A8	1nlnA0	1a9nA0
1lm8B0	1c0dA0	1omwA3	1eiyA0	1bif01	2rslA0	1uq5A1	1yrgA0
1c1yB0	1lwjA1	1b6cB1	1cbf02	1dekA1	1dozA1	1llnA1	1fo1A2
1e7uA1	1bhgA3	1m14A1	1dt9A1	1fmjA0	1dozA2	1mrj01	1fs2A0
1oeyA0	1eh9A1	1nxkA1	1qf6A2	1fnnA2	1hrkA1	1qi7A1	1ds9A0
1lfdA0	1us0A0	1bpyA4	1qd1A1	1g3qA0	1hrkA2	1abrA1	1igrA1
1gg3A2	1gveA0	1jmsA4	1e7uA4	1g7sA1	1qgoA1	1dm0A1	1igrA3
1s3sG0	1lqaA0	1jajA2	1cjaA1	1h65A0	1qgoA2	1kmvA0	1m6bA1
1k8rB0	1mi3A0	1nz0A0	1prxA2	1h8eA2	1ofuA1	1ra900	1m6bA5
1a5r00	1hw6A0	1mg7A2	1qlmA2	1h8eD2	1cfzA0	1aoeA0	1j0pA0
1gjzA0	1exbA0	1uekA1	1dtdB0	1ji0A0	1qtnA0	1df7A0	1gyoA0
1h8cA0	1ur3M0	1h72C1	1pfo02	1knqA0	1pyoA0	1juvA0	1up9A0
1i42A0	1r5yA0	1pvgA2	1pz4A0	1ly1A0	1nmsA0	3dfr00	1os6A0
1iyfA0	1iq8A1	1b63A2	1iktA0	1nijA1	1sc3A0	1dyr00	1ofwA1
1j0gA0	1gk8A2	1h7sA2	1nwaA0	1puiA0	1nw9B0	1cz3A0	1ofwA2
1j8cA0	5rubA2	1mu5A3	1fvgA0	1qdeA0	2pth00	1j3kA0	3caoA0
1m94A0	1gehA2	1oj4A1	1ev0A0	1g8pA1	1rybA0	1vdrA0	2cy300
1p1aA0	1kkoA2	1fi4A1	1bx4A2	1heiA1	1li4A1	1adeA1	1czj00
1uh6A0	1r6wA1	1ei1A2	1liiA2	1heiA2	1b8oA0	1ayl01	1gwsA1
1acc03	1oneA2	1dar04	1hdhA2	1n0uA1	1vhwA0	1ii2A1	1gwsA2
3seb02	1mdl02	1kkhA1	1auk02	1ihuA1	1je0A0	1e8gA3	1gwsA3
1dyqA2	1mucA2	1kvkA1	1fsu02	1ihuA2	1cb0A0	4eugA0	1h29D3
1eu3A1	1ec7A2	1k47A1	1p49A3	1a5t01	1rxyA0	1ui0A0	1h29D4
1et9A1	1tkkA2	1a6f00	1dhn00	1c9kA0	1g2oA0	1mugA0	1a31A3
1m4vA1	1jpdX2	1pkp02	1a8rA2	1e3mA5	1jysA0	1oe4A0	1jsdB0
3tss01	1hzyA0	1fjgI0	1b9lA0	1njgB2	1ex1A2	1aua01	1htmB0
1an801	1rk6A2	1orc00	1fjgC2	1r0wA0	1m0wA4	1uch00	1ha0A2
2sak00	1p1mA2	1u94A2	1tolA2	1sq5A0	1l7dA2	1cmxA0	1flcB0
1bmlC3	4ubpC2	1kp8A2	1dw9A2	1odfA0	1pjcA1	1g8tA0	1dkgA1
1l4dB0	1m65A0	1a6dA2	1gaxA4	1r6bX4	1hfeL1	1d02A0	1i24A2
1qqrA0	1bf6A0	4ubpA0	1mgpA3	1cr0A0	1fehA4	1cl8A0	1e6uA2
1ip9A0	1j79A0	1gpmA3	1nr3A0	1g41A1	1f8yA0	1sx5A0	1ek6A1
1mg4A0	1j6oA0	1p7lA1	1jrmA0	1jwyB0	1g66A0	1azo00	1k6xA2
1oeyJ0	1a4mA0	1p7lA2	1n91A0	1nstA0	1mj5A0	1ev7A1	1oc2A2
1pqsA0	1j5sA1	1p7lA3	1nijA2	1dar01	1cex00	1xhvA0	1kewA2
1q1oA0	1muwA0	1qm4A1	1g2rA0	1e9rA1	1qj4A0	1dmuA0	1n7hA1
1n6zA0	1qtwA0	1xrb02	1nrwA2	1egaA1	1qlwA0	1o04A1	1eq2A2

续表

1fmtA2	1d8wA0	1josA0	1nynA0	1jj7A2	1b6g00	1euhA1	1gy8A1
1g9zA0	1i60A0	1k0rA1	1lfpA2	1oe0A0	1ispA0	1o20A1	1p2zA2
1dfaA2	1k77A0	1k0rA2	1mw7A2	1a7j00	1lzlA0	1a4sA1	1lm4A0
1dfaA3	1a0cA0	1qwiA0	1gtdA0	1eg7A1	1h2wA1	1ez0A1	1g2aA0
1dq3A3	1o1zA0	1n2fA2	1mopA2	1g6oA2	1gklA0	1uxtA1	1rl4A0
1dq3A4	2plc00	1hh2P2	1ihoA2	1jqlB0	1llfA0	1ad3A1	1rn5A0
1ef0A2	2ptd00	1egaA2	1j8bA0	1khtA0	1brt00	1rtqA0	1rzhH2
1b24A1	1djxA2	1ml8A2	1pugA0	1knxA2	1dqzA0	1m4lA1	1eu1A3
1b24A2	1r3sA0	1lqlA2	1pugB0	1pjr01	1fj2A0	2ctc00	1kqfA1
1m5xA0	1j93A0	1fjgC1	1fjgP0	1pjr03	1l7aA0	1qq9A0	2napA1
1af500	1v93A0	1amuA4	1cliA1	1qhxA0	1qe3A0	1lam02	1ckv00
1bcpB1	1k87A3	1lci04	1ofuA2	1lnzA2	1tca00	1aye02	1g10A0
1preA1	1reqA1	1mdbA4	1fsz02	1gkuB2	1ju3A1	1lfwA1	1hqi00
1itxA2	1reqB1	3proC1	1pxwA0	1l8qA1	1cvl00	1obr00	2mobA0
1jndA2	1ccwB1	3proC2	1h7mA0	1iqpA1	1iupA0	1cg2A1	1ckmA1
1edqA3	1ex1A1	1ib8A1	1nmuB0	1ni3A1	1ufoA0	1h8lA1	1me3A0
1kfwA2	1lt8A0	1cii02	1ipaA1	1e32A2	2bce00	1de4C1	1fh0A0
1hjxA2	1eexA0	1e42A2	1dt9A3	1lw7A2	1m33A0	1iu8A0	1khqA0
1ll7A2	1h7bA0	1qnaA1	1e7kA0	2pjrA3	1qgeD0	1augA0	1deuA0
3eipA0	1h16A0	1qnaA2	1ufyA0	1hv8A1	1jmkC1	1i12A0	1cs8A0
1grj02	1jyhA0	1mp9A2	1qu9A0	1e69A0	1jjfA0	1m4iA0	2cb5A1
1bkf00	1r8eA1	1mpgA1	1pf5A0	1osnA0	1auoA0	1gheA0	1gmyA0
1jvwA0	1d5yA3	1kyfA2	1iv3A0	1rflA0	1bu8A1	1mk4A0	1qmyA0
1r9hA0	1c8zA0	1m3qA1	1opd00	1oboA0	1din00	1ne9A1	1csbB0
1fd9A1	1d8hA0	1p5dX4	1ptf00	1f4pA0	1ea5A0	1ne9A2	1jqpA2
1kt0A2	1c0pA2	1kfiA4	1ejgA0	5nul00	1mtzA0	1qstA0	1mirA0
1m5yA2	1pn0A3	3pmgA4	1bhp00	1d4aA0	1orvA2	1cjwA0	1pciA0
1m5yA3	1el5A2	1noyA1	1bdfA1	1ja1A1	1qo7A0	1kzfA0	1a73A0
1eq3A0	1k0iA2	2sicI0	1i50C2	2fcr00	1tib00	1n71A0	1jetA2
1hxvA0	1ng4A2	1lc0A2	1i50K0	1bvyF0	1jkmA0	1fy7A2	1dpe01
1ix5A1	1an9A2	1t4bA2	1usmA0	1e5dA1	1j1iA0	1q2yA0	1uxy01
1j6yA0	1cseI0	1nvmB2	1dcoA0	1nni10	1jfrA0	1lrzA1	1hskA3
1jnsA0	1lw6I0	1obfO2	1l0wA3	1mf7A0	1mpxA1	1lrzA2	1hztA0
1l1pA0	1cq4A0	1j5pA2	1b4bA0	1mjnA0	3tgl00	1iicA1	1nqzA0
1qddA0	1egpA0	1gr0A2	1is1A2	1atzA0	1ku0A0	1iicA2	1k2eA0
1pwbA0	1dwmA0	1dpgA2	1xxaA0	1ijbA0	1whtA0	1on0A0	1ktgA0
1e87A0	1hymA0	1ff9A2	1ab402	1pt6A0	1ehyA0	1bo4A0	1g0sA0
1g1tA1	1dxjA2	1b7gO2	1bjt04	1pcxA4	1thtA0	1bob02	1mk1A0
1gz2A0	1d5tA3	1h6dA2	1iba00	1pfkA1	1u4nA0	1iykA1	1f3yA0
1koe00	1aho00	1f06A2	1f7uA3	1rcuA0	1wm1A0	1qsmA0	1iryA0
1j34A0	1npiA0	1dih02	1iq0A2	1kzhA1	1imjA0	1nslA0	1mut00
1j34B0	2sn300	1ebfA2	1j98A0	1pfkA2	1ivyA0	1lc5A2	1q27A0

续表

1sl4A0	1bcg00	1p9lA2	1j6wA0	1kzhA2	1ei9A0	1cs1A1	1i40A0
1rdl10	1ayj00	1qkiA2	1dtjA0	1pdo00	1ac500	1fg7A1	1e9gA0
1h8uA0	1bmr00	1qmhA2	1j4wA1	1kqpA0	1c4xA0	1lk9A2	1c4kA4
1f00I3	1brz00	1n62B2	1j4wA2	1od6A0	1mx1A0	1eluA2	1t2dA2
1hq8A0	1c55A0	1n62B3	1j5kA0	1k92A1	1cpy01	1ajsA2	1bdmA2
1byfA0	1gps00	1n62B4	1k1gA0	1mopA1	1k8qA0	1m7yA2	2cmd02
1esl00	1i2uA0	1n62B5	1vig00	1o97C0	1pjaA0	1bs0A2	1hyeA2
1jwiA0	1ica00	1vlbA4	2fmr00	1jhdA1	1kezA2	1gc0A1	1o6zA1
1tn300	1jkzA0	1vlbA5	1f0xA4	1mjhA0	1jflA1	1s0aA2	1lldA2
1r13A0	1jxcA0	1vlbA6	1seiA1	1pfvA2	1jflA2	1ars02	6ldh02
1jznA0	1mr4A0	1vlbA7	1i6uA1	1i6mA1	1b73A1	1b5pA2	1hyhA2
1cwvA5	1myn00	1ffvB4	1lvk06	1j09A1	1b73A2	1jg8A1	1g55A1
1eggA0	1ne5A0	1fiqC2	1vom06	1qnf01	1mv8A3	1cl1A2	6mhtA2
1qo3C0	1qmeA3	1fiqC3	1kk8A6	1jmvA0	1dljA2	1c7nA2	1dctA2
1b6e00	1qmeA4	1fiqC4	2mysA6	1ej2A0	1epuA4	1kkjA2	1lml02
1kg0C0	1ba102	1fiqC5	1lkxA5	1k4mA0	1mqsA2	1d7uA2	1anv02
1o7bT0	1poiA2	1jroB4	1kk7A6	1np7A1	1vimA0	2oatA2	1aduB2
2afpA0	1vfyA0	1ay7B0	1mszA0	1nupA0	1moq01	1b9hA1	2vsgA1
1jetA3	1rmd01	1jakA2	1vhh00	1jgtA2	1moq02	1eg5A2	1vsgA1
1dpe03	1fbvA2	1c7sA2	1lbu02	1a8h01	1u0eA1	1fc4A1	1adeA3
1jatA0	1ldjB0	1nowA1	1ctf00	1cozA0	1u0eA2	1jf9A2	1ikpA3
1jatB0	1bor00	1msc00	1mbxC0	1ct9A2	1j5xA1	1ax4A2	1f0lA1
1i7kA0	1chc00	1qbeA0	1mg9A0	1o94D0	1j5xA2	1js3A2	1n8kA1
1s1qA0	1e4uA0	1qgiA1	1bxyA0	1sur00	1nriA0	1bjnA1	1kolA1
1u9aA0	1f62A0	1chkA2	1jj2V1	1efvA2	1m3sA0	1ohvA2	1gu7A1
1pzvA0	1fp0A0	1svb03	1f46A0	1efvB0	1b0zA1	2gsaA2	1jvbA1
1c4zD0	1g25A0	1kkoA1	1jg5A0	1f7uA1	1b0zA2	1bw0A2	1jqbA1
1br702	1iymA0	1r6wA2	1ji8A1	1gpmA2	1mvlA0	1d2fA2	1f8fA1
1mj4A0	1jm7A0	1oneA1	1jj2B2	1jilA1	1g63A0	1h0cA2	1qorA1
1cyo00	1jm7B0	1mdl01	1iq4A0	1dnpA1	1p3y10	1c4kA2	1e3jA2
1cxyA0	1n87A0	1mucA1	1jj2D0	1iq0A1	1o2dA1	1qgnA1	1iz0A1
1kbiA1	1n62C1	1ec7A1	1jtgB1	1li5A1	1jq5A1	1m40A0	1acc01
1hkoA0	1i19A3	1tkkA1	1jtgB2	1qtqA5	1sg6A1	1pwgA0	1rlr04
1j03A0	1uxy02	1jpdX1	1jyaA0	1ile03	1ujnA1	1rgzA0	1lyvA0
1vh5A0	1f0xA2	1chrA1	1jyoA0	1irxA1	1m1nA1	1k55A0	1pa1A0
1iq6A0	1e8gA1	3grs03	1k3sA0	1ni5A1	1m1nA2	1es5A0	1i9sA0
1lo7A0	1hskA2	1mo9A3	1k3eA2	1q77A0	1m1nA3	1m6kA0	1jlnA0
1q4tA0	1fiqB1	1fecA3	1k8kD1	1gaxA1	1m1nB1	1o7eA0	1fpzA0
1j1yA0	1jroA3	1nhp03	1k8kD2	1lw7A1	1m1nB2	1bueA0	1larA1
1c8uA1	1kvdB0	1d7yA3	1k8kF0	1qu3A1	1m1nB3	1ghpA0	2shpA3
1c8uA2	1xvaA1	3ladA3	1jb0D0	1bjt02	1toaA1	1ci9A0	1d5rA1
1njkA0	1dszA0	1ojt03	1k0rA4	1bgw02	1toaA2	1nzoA1	1vhrA0

续表

1ixlA0	2nllA0	1n62C2	1hh2P4	1ubkS1	1efdN1	1e25A0	1oheA2
1mkaA0	2nllB0	1fiqB3	1seiA2	1lqtA1	1efdN2	1ei5A1	1mkp00
1q6wA0	1hcqA0	1jroA5	1i6uA2	1t2dA1	1h1lD3	1mwsA4	1r6hA0
1dy5A0	1kb2A0	1mxtA2	1kjqA2	1n8kA2	1n2zA1	1qmeA2	4uagA3
1gqvA0	1gatA0	1kdgA2	1gsoA2	1hdoA0	1n2zA2	1nrfA0	1p3dA3
1n1xA0	1gnf00	1preA2	1a9xA3	1c0pA1	1pszA1	1hqsA0	1e8cA3
1agi00	1lv3A0	1aop01	1a9xA7	1i24A1	1pszA2	1cnzA0	1gg4A1
1onc00	1ecrA2	1aop03	1dv1A3	1lc0A1	1l5hA2	1lwdA0	1j6uA3
1ok7A1	1qf6A3	1cxqA0	1i7nA1	1hxhA0	1mioA1	1hdhA1	1gpc00
1ok7A2	1b7yB1	1jl1A0	1iow03	1oi7A1	1mioB2	1o98A2	1bjt05
1ok7A3	1b7yB4	1exqA0	1kblA1	1c1dA2	1mioB4	1ed8A0	1ab401
1u4gA1	1q9bA0	1nk4A1	1c30A7	1oaa00	1fjgB1	1ew2A0	1jsdA1
1keiA1	1ptq00	1j54A0	1gsa03	1b16A0	1jg7A1	1auk01	2viuA1
1lml01	1faq00	1i39A1	1b6rA3	1uayA0	1jg7A2	1fsu01	1s5dA0
1nkiA0	1kbeA0	2kfnA1	1eucB2	1e6uA1	1c8kA1	1p49A1	1ltsA0
1kw3B1	1tbn00	1x9mA1	1ehiA3	1a4iA1	1c8kA2	1d3vA0	1lt3A0
1kw3B2	1r0rI0	1noyA2	1bjt01	1ek6A2	1iirA1	1gq6A0	1bcpA0
1f1uA1	1n13A0	1rthA5	1gyfA0	1h5qA0	1iirA2	1c3pA0	1jc9A1
1f1uA2	1lr7A0	1bgxT3	1m0wA5	1ja9A0	1f0kA1	1poiB0	1fib01
1f9zA0	1tgsI0	1mu2A5	1fviA1	1k6xA1	1f0kA2	1b4uB0	1fzdA1
1kllA0	1ldtL0	1bco01	1a0i01	1nytA2	1pswA1	1f9vA0	1g38A2
1qtoA0	4sgbI0	1fxxA1	1b04A2	1oc2A1	1pswA2	1ry6A0	1dciA1
1k4nA0	1pjuA2	1hjrA0	1dgsA2	1orrA1	1uqtA1	1bg200	1fc6A3
1ecsA0	1tbrR1	1j9aA0	2hgsA1	1jtvA0	1uqtA2	2kinA0	1j7xA2
1qipA0	1i3jA2	1qz5A1	1i50A4	1geeA0	1f6dA1	1rlzA0	1ef8A2
1jc4A0	1efnB0	1qz5A3	1jx4A4	1iy8A0	1f6dA2	1kekA3	1k32A5
1cjxA1	1avv00	1ba101	1e4fT1	1qmgA1	1fsgA0	1a3aA0	1mj3A1
1cjxA2	1kgdA2	1ba103	1dk0A0	1t4bA1	1tc1A0	1a6jA0	1pixA2
1mpyA1	1yua01	1d4xA1	1div01	1gteA3	1g2qA0	1hfeL3	1pixA3
1mpyA2	1vcc00	1czaN1	1adn00	1jayA0	1qb7A0	1vsrA0	1tyfA0
1msk01	1svb02	1e4fT4	1m4jA0	1kolA2	1a3c00	1qhkA0	1uyrA1
1lugA0	1iqzA0	1jceA1	1kcqA0	1ooeA0	1dqnA0	1di6A0	1uyrA3
1jd0A0	2fdn00	1jceA2	1d4xG0	4uagA1	1nulA0	1uuyA0	1efyA2
1kopA0	7fd1A0	1ig8A3	1svy00	1fjhA0	1l1qA0	1mkzA0	1ayl03
1zncA0	1hfeL2	1g99A1	1f7sA0	1gegA0	1hgxA0	1g8lA1	1qxyA0
1by200	1jnrB1	1g99A2	1p8xA1	1gu7A2	1bd3A0	1eq6A0	1xnzA0
1bm800	1kqfB2	1dt9A2	1p8xA2	1iukA0	1ecfA2	1cbf01	1b6a01
1l3gA0	1h98A0	1fjgK0	1p8xA3	1jw9B0	1lh0A0	1tdj03	1chmA2
1r4uA0	1fxd00	1e4fT3	1hqz10	1ks9A1	1dkuA1	1khdA2	1az902
1p32A0	1fehA3	1jj2M0	1d0nA3	1nvmB1	1dkuA2	1o17A2	2fokA1
1dm9A0	1h0hB1	1ilyA0	1d0nA4	1obfO1	1o57A2	1brwA2	1rlr03
1h3fA3	1h0hB2	1e3mA2	1pcxA5	1p3dA1	1gnlA3	1ekjA0	1hozA0

续表

1fjgD2	1h0hL1	1ewqA2	1ak600	1r12A2	1gnlA4	1i6pA0	2masA0
1jh3A0	1h7wD5	1p90A0	1m2dA0	1e7wA0	1oaoA1	1g5cA0	1p2zA3
1p9kA0	1kekA5	1o13A0	1lu4A0	1evyA1	1oaoA2	1c96A2	1vjjA2
1ewnA0	1xer00	1eo1A0	1kngA0	1npyA2	1oaoC2	1l5jA4	1fjjA0
1gqzA1	1h7xD1	1kpf00	1j9bA1	1qsgA0	1epuA1	1ooyA1	1behA0
1gqzA2	1blu00	1gupA1	1k3yA1	1bdmA1	1mqsA1	1poiA1	1qouA0
1qyaA1	1fxrA0	1gupA2	1o8xA0	1bg601	1jbeA0	1cjyA2	1apj00
1qyaA2	1jb0C0	1fit00	1k0mA1	1cydA0	1mb3A0	1hynP0	1ksqA0
1uvqA1	1cqmA0	1l9vA2	1qgvA0	1dljA1	8abp01	1wpgA3	1aol00
1hdmA1	1f60B0	1gd0A0	1r26A0	1f0yA1	8abp02	1mo7A0	1avqA0
1hdmB1	1gh8A0	1dptA0	1aba00	1fmcA0	1jx6A1	1jj2L0	1hbnC0
1qcsA2	1scjB0	1mwwA0	1hd2A0	1n7hA2	1jx6A2	1iuqA2	1ugpA0
1cr5A2	1itpA0	1kptA0	1jfuA0	1pwxA0	1kgsA1	1qhlA0	1d0cA1
1e32A3	1cc8A0	1nwzA0	1t1vA0	2nacA1	1usgA2	1e19A0	1m7vA1
1cz4A2	1gxuA0	1d06A0	1e6bA1	2nacA2	1dbwA0	1gs5A0	1dd7A1
1o54A1	1qupA1	1n9lA0	1erv00	1l7dA1	2dri01	1e3mA1	1musA2
1i9gA1	2acy00	1mzuA0	1oe8A1	1jvbA2	2dri02	1ewqA1	1bob01
1qlmA1	1afi00	1oj5A0	2trxA0	2cmd01	1gca01	1a79A2	1br901
1e0gA0	1aw000	1bywA0	1b4pA1	1bgvA1	1gca02	1uehA0	1o7nA1
1fd3A0	1cpzA0	1ll8A0	1fvkA0	1d7oA0	1gudA1	4uagA2	1hbnA1
1kj6A0	1fvqA0	1p97A0	1h75A0	1dxy01	1jyeA1	1p3dA2	1uok02
1e53A0	1jwwA0	3nul00	1iyhA1	1dxy02	1jyeA2	1e8cA2	1lwjA2
1e44B0	1mwyA0	1acf00	1qmvA0	1hyeA1	1qkkA0	1gg4A2	1soxA2
1oqjA0	1nh8A3	1fil00	1thx00	1j4aA1	1p2fA1	1j6uA2	1co4A0
1mr1C0	2pii00	1f5mA0	1jlvA1	1j5pA1	1srrA0	1bx4A1	1d0cA2
1h5pA0	1o51A0	1mc0A1	1aqwA1	1lu9B2	1pea02	1ekqA0	1g3p02
1jhdA2	1d09B1	1mc0A2	1faaA0	2ae2A0	1a04A1	1kyhA0	1dj7A0
1g8fA2	1npk00	1ifqA0	1a8l01	1gpjA2	1dbqA1	1liiA1	1i4jA0
1eg7A3	1gk8A1	1h8mA0	1a8l02	1gr0A1	1ewkA2	1rkd02	1jj2Q0
1qynA0	5rubA1	1nrjA0	1eejA3	1mx3A2	1qo0D1	1l2lA1	1aop02
1fx3A0	1n0uA6	1gw5M1	1n2aA1	1o6zA2	1bykA1	1ub0A0	1qgwA0
1div02	1dar05	1gw5S0	1q98A0	1jqbA2	1bykA2	1lhpA0	1ozjA0
1jj2W0	1n0vC6	1h3qA0	1oyjA1	1bdb00	1dz3A0	1j5vA0	1d4uA0
1molA0	1mla01	1l3lA1	1eemA1	1dpgA1	1k66A0	1j9jA0	1fr2B0
1cewI0	1nm2A1	1g6oA1	1gp1A0	1eq2A1	1tmy00	1l5xA0	1o7qA0
1kwiA0	1lk5A2	1oacA1	1hyuA2	1ff9A1	1dp4A2	1ihnA0	1e5kA0
1stfI0	1psdA3	1otgA0	1hyuA4	1i36A1	1jdpA1	1l1sA0	1fo8A1
1eqkA0	1gx5A2	1q79A2	1prxA1	1lldA1	1oxkB0	1jx7A0	1i52A0
1ugiA0	1jx4A2	1bpyA3	1bed00	1o0sA2	1dbqA2	1o6dA0	1jykA0
1lniA0	1mml02	1jmsA3	1n8jA0	1o94A2	1a2oA1	1ns5A0	1qg8A0
1i0vA0	1uvjA2	1knyA1	1aw901	1pjcA2	1dcfA0	1mxiA0	1fxoA0
1a2pA0	1har02	1jajA1	1k0dA1	2pgd01	1m2eA0	1ualA1	1hm9A1

续表

1aqzA0	1rthA2	1n62C3	1kte00	6ldh01	1p6qA0	1vhyA2	1vicA0
1rtu00	1rthA3	1uxy03	2gsq01	1li4A2	1mx3A1	1k3rA1	1jv1A2
1siiA1	1rthA4	1hskA1	1gh2A0	1b7gO1	1psdA1	1vh0A0	1ll2A0
1siiA2	1rthB2	1fiqB2	1a8y01	1h6dA1	1qczA0	1ipaA2	1pztA0
1oacA2	1rdr02	1i19A1	1a8y02	1q7bA0	1h05A0	1gz0E0	1eziA0
1oacA4	1s1tB3	1f0xA3	1a8y03	1f06A1	1j2yA0	1r9wA0	1g9rA0
1ksiA1	1s1tB4	1e8gA2	2trcP1	1isiA2	1eiwA0	1f08A0	1kwsA0
1ksiA2	1khvA3	1i2kA1	1nm3A1	1dih01	1usgA1	1l2mA0	1omzA0
1a2vA1	1s1vB2	1iyeA1	1nm3A2	1eny00	1dp4A1	1tbd00	1qwjA0
1a2vA2	3hvtB3	1kt8A1	1nhyA1	1f8fA2	1jdpA2	1dzfA1	1mwpA0
1oh0A0	1jlcB3	3daaA1	1bjx00	1gtmA2	1pea01	1t0fA1	1d0qA0
1nwwA0	1jlcB4	1kjqA3	1ego00	1gz4D4	1ewkA1	1gefA0	1gsoA4
1o7nB0	1l3kA1	1obdA2	1fo5A0	1hyhA1	1fiuA0	1hh1A0	1qz5A4
1idpA0	1l3kA2	1gsoA3	1g7eA0	1kyqA1	1g7sA3	1a79A1	1ba104
1gy6A0	2bopA0	1a9xA2	1g7oA1	1qorA2	1crzA1	1dd9A2	1e4fT2
1q42A0	1fxlA1	1m0wA2	1iloA0	1e3jA1	1jr2A1	1d3yA2	1k8kA4
1jkgA0	1fxlA2	1dv1A2	1mek00	1ebfA1	1jr2A2	1dmgA0	1jceA3
1jkgB0	1nu4A0	1i7nA3	1on4A0	1iz0A2	1c8bA0	1jj2C0	1cliA2
1q40B0	1oo0B0	1iow02	1pn0A2	1j6uA1	1em8A0	1h8eG2	1b37A2
1ocvA0	1f9fA0	1kblA6	1qnxA0	1lssA0	1j23A0	1fs0G1	1f8rA1
1hkxA0	1h2vZ0	1gsa02	1cfe00	1n5dA0	1fyxA0	1e0tA1	1qlaA4
1of5A0	1n52B0	1b6rA2	1nrzA0	1npdA2	1eexB0	1a49A1	1q9iA1
1of5B0	1jmtA0	1bxrA5	1ox0A1	1p9lA1	1nbwB0	1pklA1	1chuA2
1f8nA4	1a9nB0	1eucB1	1ox0A2	1nhwA0	1kjnA0	1a3wA1	1qlaA2
1lox03	1jj2R0	1ehiA2	1i88A1	1nvtA2	1mgpA1	1e8cA1	1ucdA0
1eejA1	1hl6A0	1e4eA1	1i88A2	1gz6A0	1o0uA2	1gg4A3	1iqqA0
1jj220	1cvjA1	1fviA3	1m3kA1	1gdhA1	1t15A1	1ko7A1	1iooA0
1iq8A3	1cvjA2	1a0i03	1oeqA1	1gdhA2	1t15A2	1knxA1	1bolA0
1ewfA1	1cvjH2	1b04A1	1oeqA2	1i8tA1	1kzyC2	1o1xA0	1jy5A0
1ewfA2	3sxlA2	1b66A0	1afwA1	1id1A0	1cdzA0	1bhtA1	1udzA0
1i2kA2	1fo1A1	1tfe01	1m3kA2	1an9A1	1imoA0	1i8nA0	1h3nA2
1iyeA2	1ft8A1	1efuB3	1kjqA1	1hwxA3	1l7bA0	1fx2A0	1gaxA3
1kt8A2	1d8zA0	1mtpA1	1gsoA1	1o5iA0	1jeyA1	1azsA0	1qu3A2
3daaA2	1fj7A0	1jrrA2	1a9xA1	1tt5A1	1jeyB1	1azsB0	1fcdA3
1fuiA3	1fjcA0	1m93B1	1dv1A1	1tt5D1	1l5oA2	1kid00	1dqaA2
1iedA0	1n88A0	1lj5A1	1i7nA2	1pqwA0	1ohtA0	1srvA0	1r31A2
1fl1A0	1no8A0	1qlpA2	1iow01	1psdA2	1lba00	1gmlA0	1hp1A2
1at3A0	1o0pA0	1sek01	1gsa01	1edzA1	1j3gA0	1ecrA1	1j09A3
1c96A4	1owxA0	1jmoA1	1b6rA1	1qp8A1	1dc1A1	1ibvB0	1qtqA4
1l5jA2	1p1tA0	1f0cA1	1ehiA1	1qp8A2	1cfr00	1kblA3	1f3mA0
1bd0A1	1qm9A1	1tb6I2	1e4eA3	1lsuA0	1knvA0	1zymA1	1ceeB0
7odcA2	1qm9A2	1imvA2	1o7jA2	1qkiA1	1dfmA0	1a9xB1	1e0aB0

续表

1ct5A0	1sxl00	1c96A1	1krhA3	1eu1A1	3bamA0	1de4C2	1ej5A0
1hkvA2	1u2fA0	1c96A3	1ogiA2	1kqfA2	1m0dA0	1vi4A0	1gh9A0
1eyeA0	1ufwA0	1l5jA3	1fdr02	2napA2	2fokA3	1b7yB3	1epuA2
1f6yA0	2u1a00	1l5jA6	1cqxA3	1aa602	1nox00	1mxtA1	1mqsA3
1lucA0	1aye01	1k5nA1	1umkA2	1i2aA2	1f5vA0	1d5tA4	1pfo01
1lucB0	1pca01	1lqvA0	1ja1A4	1ad202	1icrA0	1lqtA2	1hyoA2
1nfp00	1nsa01	1onqA1	1f20A1	1mzpA2	1vfrA0	1kdgA1	1gttA1
1ezwA0	1cg2A2	3fruA1	1a8p02	1evlA2	1p5dX1	3grs01	1nr9A0
1nqkA0	1nk4A3	1je6A1	2pia02	1g5hA3	1p5dX2	3grs02	1k4iA0
1oc7A0	1mswD3	1a6zA1	1ep3B2	1h4vB2	1p5dX3	1q9iA3	1k7jA0
1dysA0	1x9mA3	1kcgC0	1qfjA2	1hc7A2	1kfiA1	1gteA4	1snnA0
1tml00	1b7yB6	1zagA1	1mw9X1	1httA2	1kfiA2	1mo9A1	1hruA0
1s2wA0	1b3tA0	1jfmA0	1i7dA1	1qe0A2	1kfiA3	1mo9A2	1jcuA0
1izcA0	1eayC0	1lkkA0	1gkuB5	1atiA2	3pmgA3	1fecA1	1k2fA2
1o66A0	1dqaA3	1d4tA0	1fp2A1	1nf9A0	1p6oA0	1fecA2	1uvjA1
1dxeA0	1r31A1	1jyrA0	1ej0A0	1im5A0	1uwzA0	1pn0A1	1fs7A2
1e0tA2	1kp6A0	1nrvA0	1inlA1	1yacA0	1ctt01	1el5A1	1g71A1
1f8mA0	1hbnA2	1h9oA0	1jg4A0	1nbaA0	1ctt02	1gpeA1	1ospO3
1mumA0	1hbnB1	1bmbA0	1o9gA1	1p5fA0	1g8mA2	1k0iA1	1rl6A1
1kblA4	1ftrA1	1opkA2	1o54A2	1fy2A0	1g8mA3	1m6iA2	1rl6A2
1jqnA2	1ftrA2	2shpA1	1g60A0	1kwgA2	1ixh01	1b37A1	1jj2E1
1n55A0	1f9yA0	2cblA3	1eg2A0	1l9xA0	1ixh02	1f8rA2	1jj2E2
1vyrA0	1dj0A2	1bg1A4	1ne2A0	1n57A0	1nnfA1	1hyuA1	1i50F0
1p1xA0	1q79A1	1jwoA0	1dl5A1	1p80A3	1nnfA2	1hyuA3	1qklA0
1q6oA0	1fa0A1	1m61A1	1dusA0	1vhqA0	1atg01	1nhp01	1dzfA2
2tpsA0	1fjgJ0	1m61A3	1g55A2	1o1yA0	1atg02	1nhp02	1v7rA0
1ujpA0	1i19A2	1mil00	1mjfA2	1a9xB2	1jetA1	1o94A3	1k7kA0
1p4cA0	1f0xA1	1fu5A0	1v3900	1rw7A0	1ryoA1	1trb01	1ex2A0
1jubA1	1ekrA0	1ju5A0	2dpmA1	1i1qB0	1ryoA2	1trb02	1dbuA0
1qopA0	1regX0	1luiA0	1m6yA1	1q7rA0	1mqiA1	1d7yA1	1ccwB2
1ub3A0	1dj0A1	2pldA0	1jqeA0	1g2iA0	1mqiA2	1d7yA2	1dd9A1
1gvfA0	1qd1A2	1d5tA2	1l3iA0	1jvnA1	1pb7A1	1chuA1	1o22A0
1thfD0	1f3vA0	1vg0A2	1i9gA2	1gpmA1	1pb7A2	1qlaA1	1mzgA0
1dvjA0	1gmuA1	1icxA0	1af702	1ka9H0	1ii5A1	1vg0A1	1ni7A0
1of8A0	1earA2	1kcmA0	1g38A1	1qdlB0	1ii5A2	3ladA1	1iq8A2
1d3gA0	1h72C2	1em2A0	1h1dA0	1v1rB1	1j1nA1	3ladA2	1gd8A0
1f6kA0	1kr4A0	1jssA0	1kpgA0	1l8aA4	1j1nA2	1ng4A1	1ni5A3
1fcqA0	1naqA0	1ln1A0	1f3lA1	1gpuA3	1gtkA1	1jehA2	1iw7D2
1h7nA0	1p1lA0	1lvk02	6mhtA1	1kekA2	1gtkA2	1lvl01	1l6rA2
1qwgA0	1kn6A0	1oe9A2	1im8A0	1ni4B2	1anf01	1fcdA1	1iw7C2
1gteA5	1in0A1	1ka1A1	1kyzA2	1hqkA0	1anf02	1fcdA2	1iw7C3
1gzgA0	1n0uA4	1nuwA1	1nv8A2	1ejbA0	1sbp01	1onfA2	1i50B3

续表

1dosA0	1dar03	1jp4A1	1qamA1	1di0A0	1sbp02	1h6vA2	1iw7D1
1pe1A0	1u2rA4	1lbvA1	1xvaA2	1c41A0	1amf01	1eyqA2	1jh6A0
1gqnA0	1kkhA2	1ni9A1	1hnnA0	1bfd02	1amf02	1jkeA0	1iuhA0
1epxA0	1k47A3	2hhmA1	1jsxA0	1bfd03	1al301	1j5uA1	1lc5A1
1hl2A0	1lq9A0	1g0hA1	1l9kA2	1v1rA0	1al302	1jw3A0	1cs1A2
1o5kA0	1iujA0	1inp02	1khhA0	1v1rB2	1lst01	1dl5A2	1fg7A2
1vc4A0	1q4rA0	1nm8A2	1nw3A2	1gpuA1	1lst02	1b25A1	1lk9A3
1qo2A0	1i1gA2	3cia00	1i4wA1	1gpuA2	1nh8A1	1ako00	1eluA1
1h1yA0	1lxjA0	1sczA0	1booA0	1zpdA1	1nh8A2	1i9zA0	1ajsA1
1onrA0	1lxnA0	1eaf00	1dctA1	1zpdA3	1pot01	1hd7A0	1m7yA1
1ktbA1	1m1gA1	1gpeA3	1nkvA0	1kekA1	1pot02	2dnjA0	1bs0A1
1me8A0	1ivzA0	1uylA0	1pjzA0	1kekA6	1eljA1	1mqoA0	1gc0A2
1ojxA0	1nxiA0	1i58A0	1jkxA0	1ni4A0	1eljA2	1qh5A0	1s0aA1
1p0kA0	1lfpA3	1id0A0	1fmtA1	1ni4B1	1eu8A1	1m2xA0	1ars01
1zfjA0	1mw7A3	1pvgA1	1chd00	1poxA1	1eu8A2	1k07A0	1b5pA1
1l6wA0	1tig00	1b63A1	1gci00	1poxA3	1nkxA1	1smlA0	1jg8A2
1euaA0	1udvA0	1h7sA1	1ic6A0	1pvdA3	1nkxA2	1jjeA0	1cl1A3
1m5wA0	1nfjA0	1mu5A1	1ga6A0	1qs0A0	1wdnA1	1a7tA0	1c7nA1
1n7kA0	1hc7A3	1gkzA1	1t1gA0	1jscA3	1wdnA2	1e5dA2	1kkjA1
1nsj00	1nj1A3	1aj600	1r64A1	1amuA1	1cb6A1	1gk9B1	1d7uA1
1o94A1	1di2A0	1th8A0	1c4kA1	1amuA2	1cb6A3	1xffA0	2oatA1
1oya00	1o0wA2	1jm6A1	1oi7A2	1lci01	1dpe02	1g3kA0	1b9hA2
1pii01	1pkp01	1bxdA0	1eucB3	1lci02	3thiA1	1ryp10	1eg5A1
1pii02	1ekzA0	1ixmA2	1jf8A0	1mdbA1	3thiA2	1ryp20	1fc4A2
1dqwA0	1qu6A1	1qqqA0	1d1qA0	1mdbA2	1a99A2	1rypA0	1jf9A1
1ep3A0	1qu6A2	1b5eA0	1iibA0	1o08A1	1ixcA2	1rypB0	1ax4A1
1kbiA2	1gtkA3	1bkpA0	1ccwA0	1l6rA1	1ixcA3	1rypC0	1p3wA1
1dbtA0	1llmC1	1m15A2	1reqA2	1q92A1	1i6aA1	1rypD0	1m32A1
1eepA0	1a1hA1	1htoA1	1reqB2	1l7mA1	1i6aA2	1rypE0	1bjnA2
1qpoA2	1rmd02	1qveA0	1bmtA2	1qq5A1	1ka1A2	1rypF0	1ohvA1
2btmA0	1meyC1	1dzoA0	1byi00	1nnlA1	1nuwA2	1rypG0	1rv3A1
1ofeB2	1ubdC1	1ay200	1r2qA0	1k1eA0	1jp4A2	1rypH0	2gsaA1
1ofeB3	1ubdC3	1t1dA0	1ls1A2	1n8nA0	1lbvA2	1rypI0	1bw0A1
1i4nA0	2gliA1	1fs1B0	1ctqA0	1nrwA1	2hhmA2	1rypJ0	1d2fA1
1jcnA0	2gliA2	1lm8C0	1kgdA1	1swvA1	1g0hA2	1rypK0	1h0cA1
1jr1B0	2gliA4	1buoA0	1ky3A0	1wpgA4	1inp03	1rypL0	1c4kA3
1hg3A0	2drpA1	3kvt00	1mh100	1ltqA2	1ni9A2	1jgtA1	1el6A3
1i1wA0	2drpA2	1nn7A0	1m7gA0	1o4wA0	1c1dA1	1ct9A1	1jj2H0
7a3hA0	1b69A0	1nexA0	1oxxK1	1a7601	1a4iA2	1ecfA1	1qpoA1
1ug6A0	1bbo02	1hv2A0	1r8sA0	1tfr01	1nytA1	1fm2B1	1o4uA1
1itxA1	1ncs00	191400	1b0uA0	1bgxT1	1npyA1	1ofeB1	1brwA1
1h1nA0	1tf3A1	1gccA0	1c4oA1	1taq01	1bgvA2	1q5qA0	2tpt03

续表

1e4mM0	1tf3A3	1cmiA0	1c4oA3	1exnA2	1p77A1	1q5qH0	1fm0E0
1jndA1	1yuiA0	1h4xA0	1cipA1	1fuiA1	1gtmA1	1ao0A1	1n62B1
1hx0A1	2adr01	1auz00	1nn5A0	1qopB1	1lehA1	1j2qH0	1vlbA3
1odzA0	1mgtA1	1fc6A1	1gtvA0	1qopB2	1npdA1	1uteA0	1fiqC1
1qnrA0	1sfe01	1k32A4	1vhtA0	1o58A1	1edzA2	1jk7A0	1jroB1
1ur1A0	1bbg00	1l8bA0	1aquA0	1o58A2	1cby00	1nnwA0	1j3aA0
1bqcA0	1dq3A2	1ap800	1g6hA0	1f2dA1	3pviA0	1auiA0	1nd4A0
1ht6A1	1puc00	1d1rA0	1in4A1	1f2dA2	1lam01	1g5bA0	1j7lA2
1jz7A3	1cksA0	2if100	1p5zB0	1j0aA1	1ojrA0	1ii7A0	1ucsA0
1edqA2	1gk8I0	1bo1A1	1zin00	1j0aA2	1e4cP0	4kbpA2	1e8pA0
1edg00	1rblM0	1bo1A2	1aky00	1tdj01	1k0wA0	1hp1A1	1d0cA3
1gcyA1	1bwvS0	1pp9A1	1f60A1	1tdj02	1pvtA0	1a6q01	1dd7A3
1ji1A2	1i2aA1	1pp9A2	1e2kA0	1k7cA0	1eu1A2		
3. α-β 结构							
1h8pA1	1pls00	1gj7B1	1go3E1	1ogaD2	1gmiA0	2cavA1	1i7dA3
1l6jA3	1c0mA2	1gj7B2	1khiA2	1ogaE1	1rsy00	1ne6A1	1euwA0
1l6jA5	1ex4A2	1os8A2	1luzA0	1ogaE2	1e7uA2	1ne6A2	1sixA0
1j7mA0	1d09B2	1kliH1	1jb7A1	1nkoA0	1bdyA0	1o5lA0	1f7dA0
1goiA2	1bkb01	1kliH2	1jb7A2	1pkoA0	1djxA3	1o7fA1	1pkhA0
1aiw00	1khiA1	1agjA1	1jb7A3	1smoA0	1rlw00	1o7fA3	1sjnA0
1ed7A0	1m1gA3	1agjA2	1pxfA0	1my7A0	1dsyA0	1ft9A1	1tul00
1bx700	1rl2A2	1elvA1	1b8aA1	1kgcD1	1rh8A0	1rc6A0	1kmtA0
1skz01	1g2bA0	1elvA2	1cuk01	1kgcE1	1f86A0	1fxzA1	1ds6B0
1g1tA2	1jo8A0	1eq9A1	1g2912	1oaqL0	1mfmA0	1fxzA2	1gpr00
1nziA2	1i1jA0	1eq9A2	1je5A0	1xfpA0	1oalA0	1m4oA0	1jz7A5
1kliL1	1ootA0	2hlcA1	1kl9A1	1edqA1	1ej8A0	1qqp10	1n7oA1
1g2lB1	1ckaA0	2hlcA2	1qzgA0	1ji1A1	1noa00	1qqp20	1jovA0
1danL1	1fmk01	1nn6A1	1fviA2	1ngzB2	1akp00	1qqp30	1cb8A2
1q4gA1	1bbzA0	1nn6A2	1gd7A0	1pewA0	1c7sA1	1aym10	1hn0A3
1fakL1	1gcqC0	1eptA0	1l0wA1	1q0xL2	1e5bA0	1aym20	1j0mA2
1klo01	1ng2A1	1eptB0	1mkhA0	1l6xA1	1exg00	1aym30	1siiA3
1klo02	1ng2A2	1lvmA1	1hh2P3	1l6xA2	1jz7A2	1hxs10	1oacA3
1klo03	1i07A0	1cqqA1	1e1oA1	1fltX0	1jz7A4	1c8nA0	1ksiA3
1dx5I1	1kjwA1	1cqqA2	1qvcA0	1mfa02	1c7sA4	1b35A0	1gpiA0
1dx5I2	1pht00	1q2wA1	1eovA1	1nkr01	1bhgA2	1b35B0	1gp0A0
1dx5I3	1bb900	1q2wA2	1k3rA2	1pbyA2	1a3qA3	1b35C0	1m6pA0
1xkaL1	1ycsB2	1g2lA2	1n9wA1	1pbyA3	1iknA2	2stv00	1wpgA1
1eaiC0	1ad5A1	1qa7A1	1rl2A1	1qhoA3	1p7hL1	1ddlA0	1f3lA2
1moxC0	1jqqA0	1qa7A2	1jj2A2	1qhoA4	1svb04	1ng0A0	1g6q12
1nt0A2	1ark00	2hrvA1	1jmcA1	1w8oA2	1okeA4	1f2nA0	1uw6A0
1autL1	1awj00	2hrvA2	1jmcA2	2fcbA1	1p5vA1	1pgl10	1njhA0
1nqlB0	1aww00	1lcyA1	3ullA0	2fcbA2	1m1sA0	1pgl20	1iw7D3

续表

1ccvA0	1azeA0	1lcyA2	1b7yB2	1cczA1	1n0lA1	1tme10	1lnzA1
1couA0	1hsq00	1mbmA1	1ckmA2	1cczA2	1mspA0	1tme20	1pu5A0
1hae00	1j3tA0	1mbmA2	1quqA0	1cs6A1	4kbpA1	1tme30	1jmaA0
1hj7A1	1jegA0	1svpA1	1quqB0	1cs6A2	1sfdA0	2bbvA0	1rg8A0
1hj7A2	1nm7A0	1svpA2	1a0i02	1cs6A3	1oe1A1	4sbvA0	1knlA0
1hx2A0	1udlA0	2pkaA0	1p16A3	1cs6A4	1oe1A2	2tbvA0	1pwaA0
1k36A0	1ue9A0	2pkaB0	1eygA0	1my5A0	1plc00	1bev10	1l2hA0
1l3yA0	1uffA0	1befA2	1i50H0	1ugnA1	1bqk00	1f8vA0	1a8d02
1tpg02	1dj7B0	1dleA1	1l1oC1	1uvqA2	1jzgA0	1gff20	1md6A0
1extA1	2ahjB2	1dleA2	1dgsA4	1vcaA1	1kv7A1	1ny710	1dqgA0
1extA2	1jb0E0	1dy9A2	1fjgL0	1vcaA2	1kv7A2	1e57A0	1q1uA0
1d4vA1	1vie00	1fiwA1	1fjgQ0	1epfA1	1e30A0	1m06G0	1avwB0
1d4vA2	1lvk01	1fiwA2	1ltlA2	1epfA2	1qhqA0	1dg6A0	1wba00
1d4vA3	1kk8A1	1ucyH0	1hr0W0	1qfoA0	1fwxA2	2tnfA0	1m2tB1
1jmaB1	1jj2P0	1azzA1	1d7qA0	1clc01	1jer00	1pk6A0	1m2tB2
1jmaB2	1g8xA4	1ekbB2	1ewiA0	1gsmA1	2cuaA0	1pk6B0	1avaC0
1lml03	2mysA1	1m9uA1	1j6qA0	1gsmA2	1ibyA0	1o91A0	1epwA4
1f94A0	1fx7A3	1orfA2	1jjgA0	1j0hA1	1hfuA1	1aly00	1jlxA1
1m9zA0	1bi103	1bqyA2	1jt8A0	1jmxA2	1hfuA2	1kxgA0	1jlxA2
1nxb00	1bymA0	2hntC0	1ne3A0	1neu00	1hfuA3	1iqaA0	1qqlA0
1bteA0	1kq1A0	2hntE0	1pfsA0	1tvdA0	1gskA1	1rj8A0	1hwmB2
1ff4A0	1mgqA0	1ezxC0	1rip00	1vjjA1	1gskA2	1tnrA0	1dfcA1
1tgxA0	1ljoA0	1autC1	1sro00	1vjjA3	1gskA3	1qhdA1	1dfcA2
1fas00	1d3bA0	1ky9A1	1iw7C6	1vjjA4	1kdj00	1ahsA0	1dfcA3
1hc9A0	1d3bB0	1md8A1	1i50B7	1wwcA0	2cbp00	1bvp12	1dfcA4
1rewC0	1m5q11	1gpzA2	1oewA1	1cqyA0	1aozA1	1h4gA0	1hcd00
1cdq00	1hk9A0	1befA1	1oewA2	2fbjH2	1aozA2	1olrA0	1j0sA0
1drs00	1bia03	1dy9A1	1nh0A0	1a64A0	1aozA3	2nlrA0	1b2pA0
1jgkA0	1b34A0	1jxpA1	1htrB1	1bf201	1ikoP0	1h0bA0	1ciy02
1c2aA1	1b34B0	1a1rA1	1htrB2	1cdy01	1kbvA1	1t6gC0	1ji6A3
1h34A0	1n9rA0	1h8eD1	1b5fA0	1cdy02	1kbvA2	1nls00	1vmoA0
1df9C0	1m1fA0	1a1x00	1w50A1	1dqtA0	1cyx00	1kqrA0	1ouwA0
1bi6H0	3vub00	1jnpA0	1w50A2	1ex0A1	1m56B2	1ikpA1	1ugxA0
1elvA3	1ub4A0	1i71A0	1dpjA2	1fhgA0	2occB2	1is3A0	1c3kA0
1h03P1	1ne8A0	1pmlA0	1fmb00	1gl4B0	1gw0A1	1gv9A0	1jm1A0
1h03P2	1bco02	2hppP0	1hrnA2	1lp9E1	1gw0A2	1o4yA0	1nykA0
1ly2A1	1mhnA0	1jfnA1	1j71A1	1g1cA0	1gw0A3	1dypA0	1o7nA2
1ly2A2	1igqA0	1krhA2	1j71A2	1iam01	1kcw01	1d2sA0	1rie00
1ridA1	1khcA1	1ogiA1	1lf2A1	1iam02	1kcw02	1a8d01	1fqtA0
1ridA3	1h3zA0	1f60A2	1lf2A2	1mh5B2	1sluA0	1gzcA0	1g8kB0
1ridA4	1n27A0	1f60A3	2-Apr-01	1nezG0	1bg1A2	1n1tA2	1rfs00
1g44C4	1m9sA3	1exmA3	2-Apr-02	1uadC0	1g4mA2	1w0pA1	1hxn00

续表

1qubA1	1m9sA4	1fdr01	4fiv00	1uctA1	1soxA3	1w0pA3	1itvA0
1qubA2	1ib8A2	1cqxA2	1pfzA2	1onqA2	1b4rA0	2ayh00	1gen00
1qubA3	1lplA0	1umkA1	1mpp02	1ev2E1	1aohA0	1w6nA0	1qhuA1
1qubA4	1ixdA0	1ja1A2	2rspA0	1ev2E2	1g1kA0	1mveA0	1fbl02
1qubA5	1jj2S0	1kk1A2	1lyaA0	1hkfA0	1ayoA0	1ukgA0	1tl2A0
1e88A3	1pcfA0	1kk1A3	1lyaB0	1qfhA1	1f0lA3	2ltnA0	3sil00
1gknA1	1e8rA0	1f20A3	1nsoA0	1qfhA2	1g87A2	1g86A0	1f8dA0
1gknA2	1g31A0	1g7sA2	1awqA0	1zxq01	1nbcA0	2sli01	1n1tA1
1hfi00	1p3hA0	1g7sA4	1v9tA0	1zxq02	1eaqA0	1dhkB0	1w0pA2
1tpg01	1lml04	2pia01	1is2A2	3fruA2	1uolA0	1bkzA0	1nscA0
1jsdA2	1qz5A2	1ddgA1	1jqiA2	1dn0B2	1dqiA0	1epwA3	1w8oA1
2visC2	1k8kA2	1ep3B1	1jiwI0	1fp5A1	1dfx00	1dykA1	2sli02
1lpbA0	1qntA1	1n0uA3	1smpI0	1fp5A2	1amx00	1dykA2	1e8uA0
1imt00	1amuA3	1qfjA1	1x8pA0	1ji2A1	1who00	1sacA0	1h6lA0
1mkkA0	1lci03	1dar02	1kt6A0	1m7xA1	1nepA0	1v6iA0	1pjxA0
2tgi00	1mdbA3	1jj2B1	1i4uA0	1f2qA2	1ktjA0	1h30A1	1cruA0
1rewA0	1mtpA2	1aipB2	1ifc00	1i1rA1	1lm8V2	1h30A2	1crzA2
1agqA0	1jrrA1	1aipB3	1lf7A0	1n26A2	1ejfA0	1c4rA0	1q7fA0
1aocA0	1m93B2	1d1nA0	1qftA0	1b88A0	1gmeA0	1hlcA0	1npeA0
1b8kA0	1lj5A2	1h8eA1	1kqwA0	1f42A2	1gmeB0	1jhnA1	1k32A1
1m4uA1	1uhgA1	1i8dA1	1hmr00	1f97A1	1shsA0	1a34A0	1ofzA0
1hcnA0	1a7cA2	1i8dA2	1dzkA0	1f97A2	1acc04	1stmA0	1fwxA1
1hcnB0	1qlpA1	1kzlA2	1e5pA0	1fo0A0	1h6fA0	1pgs01	1gxrA0
1jpyA0	1as4A1	1e43A2	1qy1A0	1fo0B0	1e2wA1	1pgs02	1nr0A1
1fl7B0	1hp7A1	1hvxA3	1mdc00	1g0dA1	1g4mA1	1sdwA2	1nr0A2
1pdgA0	1sek02	1e0tA3	1bebA0	1hdmA2	1f00I1	1acc02	1pbyB0
1jhfA1	1jmoA2	1pklA3	1bj700	1hdmB2	1cwvA2	1od3A0	2bbkH0
1f39A0	1qmnA1	1pkyC3	1cbs00	1itbB1	1cwvA3	1k3iA1	1jmxB0
1b12A1	1k9oI2	1bd0A2	1kxoA0	1itbB2	1kyfA1	1d7pM0	1gotB0
1umuA0	1tb6I1	7odcA1	1epaA0	1itbB3	1p5vA2	1jz7A1	1k32A2
1a7i00	1imvA1	1f3tA1	1gm6A0	1je6A2	1l4iA2	1ju3A2	1k8kC0
1b8tA2	4ubpC1	1hkvA1	1ftpA0	1a6zA2	1n0lA2	1jhjA0	1ri6A0
1g47A0	1nteA0	2eng00	1lfo00	1cd800	1f00I2	1gnyA0	2madH0
1j2oA0	1mfgA0	1bw300	1l6mA0	1o0vA1	1cwvA4	1j83A0	1erjA0
1nypA0	1qauA0	1pqhA0	1avgI0	1p7hL2	1n12A0	1w8oA3	1jofA0
4ubpB0	1g9oA0	1eu1A4	1jzuA0	1mcpH2	1pdkB0	1guiA0	1p22A2
1e9yA1	1n7eA0	1kqfA7	1luqA0	1ncnA0	1klfB1	1k12A0	1a12A0
1kmxA0	1obzA1	1pyuB0	1nqnA0	1cid01	1klfB2	1kexA0	1jtdB0
1be3I0	1fc6A2	1qcsA1	1ei5A2	1cid02	1d5rA2	1mpxA3	1k3iA2
1igrA2	1ihjA0	2napA4	1ei5A3	1imhC2	1h8lA2	1dyoA0	1utcA0
1exkA0	1be9A0	1aa604	1pbyA1	1dr9A1	1dceA2	1gqpA0	1h2wA2
1ep3B3	1qavA0	1cr5A1	1jmxA1	1dr9A2	1d2oA1	1i5pA1	1g72A0

续表

1jhnA2	1kwaA0	1e32A1	1qjpA0	1eh9A3	1d2oA2	1r64A2	1fgA0
1hhnA0	1k32A3	1cz4A1	1qj8A0	1ktkE1	1e42A1	1shwB0	1qksA2
1mvfD0	1lcyA3	1iw7D4	1p4tA0	1ac000	1i31A1	1ciy03	1k7iA1
1ub4C0	1nf3C0	1n10A1	1k3bA0	1ehxA0	1i31A2	1gu3A0	1qreA0
1ahl00	1rzxA0	1pfbA0	1whi00	1gxeA0	1hx0A2	1ji6A2	1v3wA0
1b8wA0	1l6oA0	1c8cA0	1oxdA0	1ie5A0	1ht6A2	1bhgA1	1hm9A2
1bds00	1ky9A3	1b3aA0	1ggxA0	1jbjA1	1gcyA2	1p8jA2	1mr7A0
1h5oA0	1ky9B4	1m8aA0	2por00	1jbjA2	1ji1A5	1cx1A0	1kgqA2
1sh100	1d5gA0	1nr4A0	3prn00	1nct00	1kwgA3	1k42A0	1fxjA2
1n2fA1	1i1600	1tvxA0	1e54A0	1tit00	1mxgA3	1xnaA0	1krrA2
1ml8A1	1iu0A0	1dokA0	1hxxA0	1wit00	1e43A3	2arcA0	1lxa01
1i50I1	1m5zA0	1e0bA0	1a0sP0	1ok0A0	1qhoA2	1nziA1	1xat00
1i50I2	1p1eA0	1o7zA0	1af6A0	2hft01	1iv8A5	1sfp00	1l0sA0
1dl6A0	1uepA0	1f2lA0	1kmoA2	2hft02	1j0hA3	1sppB0	1m8nA0
1pft00	1uewA0	1qg7A0	1nqeA2	1fna00	1ktbA2	1nt0A3	1k5cA0
1qyp00	1uezA0	3il800	1fepA2	1ten00	7taa02	1sdwA1	1h80A0
1tfi00	1uf1A0	1cm9A0	1qfgA2	1eerB1	1bf203	1ig0A2	1pe9A0
1yua02	1ufxA0	1knaA0	1gweA0	1eerB2	1uok03	1ig3A1	1czfA0
1qf8A2	1ujvA0	1dz1A0	1p80A1	1bquA1	1gjwA2	1odmA0	1qcxA0
1jj2Y0	3ezmA0	1eigA0	1jb7B0	1bquA2	1h3gA3	1dcs00	1gq8A0
1jj2Z0	1ospO1	1el0A0	2sli03	1cfb01	1m53A3	1gp6A0	1bn8A0
1rkd01	1mkcA0	1g6zA0	1qd6C0	1cfb02	1ua7A2	1e5rA1	1bhe00
1rb900	1qs1A1	1j8iA0	1dfuP0	1danT0	1ji2A3	1nlqA0	1rmg00
1lkoA2	1qs1A2	1rjtA0	1feuA1	1ex0A3	1m7xA3	1k5jA0	1air00
1dx8A0	1ojqA0	1qb5D0	1qtqA1	1ex0A4	1lwjA3	1oh4A0	1ee6A0
1pfvA1	1g24A0	1afp00	1qtqA2	1fnf01	1eh9A2	1gwmA0	1qjvA0
1ospO2	1gxyA0	1c4qA0	1e50B0	1fnf02	1i82A0	1jopA0	1qq1A0
1mknA0	1giqA1	2sns00	1qwzA0	1fnf03	1im3D0	1nqjA0	1dabA0
1i5hW0	1j7nA2	2sob00	1t2wA0	1fyhB1	1d7bA0	1f35A0	1dbgA0
1jmqA0	1uscA0	1oxxK3	1iw7C1	1fyhB2	1lyqA0	1dmhA0	1ezgA0
1o6wA1	1flmA0	1fr3A0	1i50B6	1axiB1	1m9sA2	3pcgA0	1ofeB4
1o6wA2	1i0rA0	1e2wA2	1p6vA0	1axiB2	1gyvA0	3pcgM0	1hf2A2
1nh2C0	1dnlA0	1h9mA1	1jeyA2	1qg3A1	1p4uA0	1gff10	1ayl02
1nh2D2	1ejeA0	1b9mA2	1jeyB3	1qg3A2	1lmiA0	1vpsA0	1p2zA1
1nvpC0	1gk9B2	1b9mA3	1c5eA0	1f6fB1	1l6pA0	1sva10	1ois02
1nvpD2	1fm2B3	1bdo00	4bcl00	1f6fB2	1ifrA0	1khxA0	1lktA0
1dkgA2	1iq8A4	1h9sA2	1jk4A0	1iarB1	1o75A3	1dd1A0	1preA3
1auuA0	1inlA2	1g2913	1v54F0	1iarB2	1n67A1	1g6gA0	1gppA0
1jubA2	1mjfA1	1htp00	1h8eH0	1n26A3	1n67A2	1lgpA0	1at000
1g3p01	1uirA1	1iw7C5	1fs0E1	1f42A1	1o75A2	1gxcA0	1mi8A0
1tolA1	1iy9A2	1q12A3	1h4aX1	1f42A3	1p5vB0	1dmzA0	1dq3A1
1unqA0	1iw7C4	1dczA0	1h4aX2	1g0dA3	1kzqA1	1g3gA0	1am200

续表

1eazA0	1fjrA1	1fyc00	1npsA0	1g0dA4	1kzqA2	1mzkA0	4dpvZ0
1fgyA0	1jjdA0	1ghj00	2bb201	1qr4A1	1mkfA1	1r21A0	1lp3A0
1ntvA0	1jj2T0	1gjxA0	2bb202	1lqsR1	1mkfA2	1uhtA0	1g9mG0
1btkA0	1pq7A1	1k8mA0	1hdfA0	1lqsR2	1f8nA1	1lb6A0	1bdfA2
1ddwA0	1pq7A2	3chbD0	1bd7B2	1cd9B2	1bu8A2	1czyA0	1i50C1
1mixA2	1ssxA1	3seb01	1wkt00	1egjA0	1ca102	1k2fA1	1kmoA1
1evhA0	1ssxA2	1dyqA1	1bhu00	1fnhA1	1lox02	1n7oA3	1nqeA1
1faoA0	1gvkB1	1eu3A2	1c01A0	1fnhA2	1qhvA0	1cb8A3	1fepA1
1h4rA3	1gvkB2	1enfA1	1f53A0	1fnhA3	1h7zA0	1f1sA3	1qfgA1
1ddvA0	1hj9A1	1et9A2	1g6eA0	1kv3A3	1svb01	1j0mA3	1jbiA0
1mai00	1hj9A2	1m4vA2	2bbkL0	1gh7A1	1okeA1	1cq3A0	1dqcA0
1btn00	1a7s01	3tss02	1dkxA1	1gh7A2	1rqwA0	1p35A0	1hxrA0
1dynA0	1a7s02	1bcpB2	1bpr00	1gh7A3	1nxmA0	1c3gA1	1h6qA1
1aqcA0	1arb01	1bcpD0	1k5nA2	1bpv00	1ep0A0	1c3gA2	1g9gA2
1dbhA2	1arb02	1bcpF0	1k5nB0	1j8kA0	1x82A0	1k3wA1	1p7tA2
1qqgA1	1eaxA1	1jb3A0	1mjuH1	1k85A0	1fi2A0	1l1zA2	1fjrA2
1qqgA2	1eaxA2	1br902	1mjuH2	1lwrA0	1vj2A0	1ee8A2	1g8lA3
1kz7A2	1gg6B0	1d2bA0	1mjuL1	1n6uA1	1dgwA0	1k82A2	1dtoA2
1j0wA0	1gg6C0	1uapA0	1mjuL2	1owwA0	1gqgA1	1gwyA0	1qqhA0
1omwA5	1h8dH1	1c9oA0	1mexH1	1uemA0	1pmi01	1ni5A4	1ik9A1
1k5dB0	1h8dH2	1o7iA0	1mexH2	1uenA0	1pmi03	1nc7A0	1a8h02
1foeA2	1sgpE1	1oxxK2	1mexL1	1ujtA0	1j58A1	1nwbA0	1b12A2
1gg3A1	1sgpE2	1fl0A0	1mqkH0	2fnbA0	1j58A2	1js8A2	
1ddmA0	1gvzA1	1a6202	1ncwH1	1gtfA0	1hw5A1	1p2zA4	
1fhoA0	1gvzA2	1gvp00	1eajA0	1o6sB0	1o5uA0	4htcI0	
1mkeA0	1bio01	1k0rA3	1k3iA3	1edhA2	1lr5A0	1hic00	
1n3hA0	1bio02	1bkb02	1ogaD1	1uowA0	1o4tA0	1mw9X3	
4. Few secondary structures							
1kekA7	1bg503	1v54M0	1olgB0	1an4A0	3btmI0	1ubkS2	1fzaE2
1kekB7	1ba305	1v54Z0	1olgC0	1an4B0	1ejmB0	1ublS2	1fzbB2
1b0pA7	4mt200	1v55M0	1olgD0	1uklC0	1ejmD0	1ubrS2	1fzbE2
1b0pB7	1jfwA0	1v55Z0	1olhA0	1uklD0	1ejmF0	1ubtS2	1fzeB2
2pdaA7	1fkuA0	2occM0	1olhB0	1uklE0	3bteI0	1ubhS2	1fzeE2
2pdaB7	1k5kA0	2occZ0	1olhC0	1uklF0	3btdI0	1ubjS2	1m1jB2
1icfI0	1tac00	1ocrM0	1olhD0	1bct00	3btgI0	1uboS2	1m1jE2
1icfJ0	1tbc00	1ocrZ0	1saeA0	1ebdC0	3btqI0	1ubuS2	1jfeB2
1l3hA0	1tiv00	1occM0	1saeB0	2pdd00	3bttI0	1h2rS2	1jfeE2
1gp8A0	1agg00	1occZ0	1saeC0	2pde00	3btwI0	1ubmS2	1m1jC2
2gp8A0	1oma00	1ocoM0	1saeD0	1bal00	1ld5A0	1h2aS2	1m1jF2
1f02T0	1omb00	1ocoZ0	1safA0	1bbl00	1co7I0	1e3dA2	1jfeC2
1n32N0	1iva00	1oczM0	1safB0	2ilk02	1brbI0	1e3dC2	1jfeF2
1fjgN0	1oav00	1oczZ0	1safC0	1ilk02	1fakI0	1cc1S2	1lwuB2

续表

1j5eN0	1oaw00	1jb0J0	1safD0	1lk3A2	1ld6A0	2frvA2	1lwuE2
1iblN0	1qdp00	1jo6A0	1sagA0	1lk3B2	1uubA0	2frvC2	1lwuH2
1i94N0	1vtx00	1ckmA3	1sagB0	1inr02	1kthA0	2frvE2	1lwuK2
1hr0N0	1h59B0	1ckmB3	1sagC0	1j7vL2	2knt00	2frvG2	1n73B2
1hnzN0	1boeA0	1cknA3	1sagD0	1vlk02	1knt00	2frvI2	1n73E2
1ibkN0	1hy9A0	1cknB3	1sahA0	1lqsL2	1kun00	2frvS2	1lwuC2
1ibmN0	1fre00	1cko03	1sahB0	1lqsM2	1tfxC0	1frvA2	1lwuF2
1n33N0	1v54L0	1qqp40	1sahC0	1jsuC0	1tfxD0	1frvC2	1lwuI2
1hnwN0	1v54Y0	1bbt40	1sahD0	1f8nA3	1adz00	1frfS2	1lwuL2
1hnxN0	1v55L0	1fhp40	1saiA0	1yge03	1aapA0	1iuaA0	1n73C2
1n36N0	1v55Y0	1fod40	1saiB0	1fgrA3	1aapB0	1eytA0	1n73F2
1n34N0	2occL0	1fmd40	1saiC0	1fgoA3	1tawB0	1b0yA0	1rzhH1
1fjfN0	2occY0	1tme40	1saiD0	1fgtA3	1ca0D0	1ckuA0	1qovH1
1edxA0	1ocrL0	2mev40	1sajA0	1fgqA3	1ca0I0	1ckuB0	1ry5H1
1qojA0	1ocrY0	1mec40	1sajB0	1fgmA3	1brcI0	1hrq00	1aijH1
1qojB0	1occL0	1v54J0	1sajC0	2sblB3	1irhA0	1hrr00	1aijT1
1e52A0	1occY0	1v54W0	1sajD0	1ik3A3	1shp00	1neh00	1e6dH1
1e52B0	1ocoL0	1v55J0	1sakA0	1rovA3	1d0dA0	1js2A0	1ogvH1
1g9gA3	1ocoY0	1v55W0	1sakB0	1jnqA3	1kigI0	1js2B0	1l9bH1
1g9jA3	1oczL0	2occJ0	1sakC0	1n8qA3	1tap00	1js2C0	1rzzH1
1faeA3	1oczY0	2occW0	1sakD0	1no3A3	1tcp00	1js2D0	1rzzT1
1fbwA3	1v54K0	1ocrJ0	1salA0	1hu9A3	1dtx00	1hip00	1ds8H1
1fce03	1v54X0	1ocrW0	1salB0	1lnh03	1bf000	1noe00	1ds8T1
1f9dA3	1v55K0	1occJ0	1salC0	1byt03	1dem00	3hipA0	1dv3H1
1fboA3	1v55X0	1occW0	1salD0	1f8nA2	1den00	3hipB0	1dv3T1
1f9oA3	2occK0	1ocoJ0	3sakA0	1yge02	1dtk00	3hipC0	1dv6H1
1l1yA3	2occX0	1ocoW0	3sakB0	1fgrA2	1jc6A0	1hlqA0	1dv6T1
1l1yB3	1ocrK0	1oczJ0	3sakC0	1fgoA2	1bunB0	1hlqB0	1kbyH1
1l1yC3	1ocrX0	1oczW0	3sakD0	1fgtA2	1bik00	1hlqC0	1rg5H1
1l1yD3	1occK0	1h7dA0	2bbvD0	1fgqA2	1kvdA0	1isuA0	1m3xH1
1l1yE3	1occX0	1v54I0	2bbvE0	1fgmA2	1kvdC0	1isuB0	1mpsH1
1l1yF3	1ocoK0	1v54V0	2bbvF0	2sblB2	1kveA0	1hpi00	1aigH1
1l2aA3	1ocoX0	1v55I0	1-Nov-00	1ik3A2	1kveC0	2hipA0	1aigP1
1l2aB3	1oczK0	1v55V0	1novF0	1rovA2	1hnr00	2hipB0	1fnpH1
1l2aC3	1oczX0	2occI0	1novD0	1jnqA2	1hns00	1pih00	1fnqH1
1l2aD3	1a1tA0	2occV0	1i4oC0	1n8qA2	1gpeA2	1pij00	1s00H1
1l2aE3	1f6uA0	1ocrI0	1i4oD0	1no3A2	1gpeB2	1ibvA0	1s00T1
1l2aF3	1mfs00	1ocrV0	1kmcC0	1hu9A2	1cf3A2	1ibvC0	1pcrH1
1rdr01	1aaf00	1occI0	1kmcD0	1lnh02	1gal02	1ibvE0	1e14H1
1rajA1	1a6bB0	1occV0	1i51E0	1byt02	1inp01	1ibtA0	1jgyH1
1qa4A0	1bj6A0	1ocoI0	1i51F0	1j8eA0	1uq5A2	1ibtC0	1jgzH1
1qa5A0	1eskA0	1ocoV0	1tvs00	1d2lA0	1j1mA2	1ibtE0	1rqkH1

续表

1g3jB0	1cl4A0	1oczI0	1tvt00	1f5yA1	1ift02	1ibuA0	1rvjH1
1g3jD0	1j2lA0	1oczV0	1nkzA0	1ldl00	1ifs02	1ibuC0	1f6nH1
1n32M2	1fvl00	1v54G0	1nkzC0	1cr8A0	1br6A2	1ibuE0	1jgwH1
1fjgM2	1jypA0	1v54T0	1nkzE0	1d2jA0	1rtc02	1ibwA0	1rgnH1
1j5eM2	1n4yA0	1v55G0	1kzuA0	1f5yA2	1ifu02	1ibwC0	1umxH1
1iblM2	1kst00	1v55T0	1kzuD0	1ldr00	1br5A2	1ibwE0	4rcrH1
1hr0M2	1l3xA0	2occG0	1kzuG0	1jrfA0	2aaiA2	1pyaA0	1pssH1
1hnzM2	1iq2A0	2occT0	1ijdA0	1k7bA0	1il4A2	1pyaC0	1pstH1
1ibmM2	1mpzA0	1ocrG0	1ijdC0	1g6xA0	1il5A2	1pyaE0	1ystH1
1ibkM2	1ro3A0	1ocrT0	1ijdE0	1k6uA0	1fmp02	1hq6A0	1jgxH1
1n33M2	2ech00	1occG0	1lghA0	1qlqA0	1il3A2	1hq6C0	1k6lH1
1hnwM2	1fleI0	1occT0	1lghD0	5pti00	1il5B2	1b8zA0	1k6nH1
1hnxM2	2rel00	1ocoG0	1lghG0	1bpi00	1apgA2	1b8zB0	2rcrH1
1n36M2	1rel00	1ocoT0	1lghJ0	4pti00	1il9A2	1p71A0	1l9jH1
1n34M2	1udkA0	1oczG0	1a92A0	1f7zI0	1uq4A2	1p71B0	1l9jT1
1fjfM2	1aym40	1oczT0	1a92B0	1d0dB0	1obs02	1p78A0	1jh0H1
1jpwD0	1nd2D0	2pspA1	1a92C0	1f5rI0	1obt02	1p78B0	1dxrH1
1jpwE0	1ncrD0	2pspB1	1a92D0	1fy8I0	1llnA2	1p51A0	6prcH1
1jpwF0	1qju40	1pcp01	1aml00	6pti00	1mrj02	1p51B0	1prcH1
1pbyC0	1qjx40	1pspA1	1ba400	3tgkI0	1mrk02	1p51C0	5prcH1
1jjuC0	1qjy40	1pspB1	1iytA0	3tgiI0	1gisA2	1p51D0	3prcH1
1jmxG0	1ayn40	1e9tA0	1ba600	3btkI0	1tcs02	1huuA0	4prcH1
1jmzG0	1c8m40	1pe310	1aw600	1tpaI0	1giuA2	1huuB0	2prcH1
1jj2A3	1nd3D0	1pe320	125d00	2ptcI0	1qd2A2	1huuC0	7prcH1
1m90C3	1fpn40	1hi7A0	1cld00	2tgpI0	1j4gA2	1hueA0	1r2cH1
1qvgA3	1hxs40	1hi7B0	1f4sP0	3tpiI0	1j4gB2	1hueB0	1eysH1
1q81C3	2plv40	1ps200	1f5eP0	1bzxI0	1j4gC2	1mulA0	1cl1A1
1q82C3	1al240	2pspA2	2alcA0	2hexA0	1j4gD2	1owfA0	1cl1B1
1k8aC3	1ar640	2pspB2	3alcA0	2hexB0	1nliA2	1owgA0	1cl2A1
1k9mC3	1ar740	1pspA2	1pyc00	2hexC0	1mrg02	1ouzA0	1cl2B1
1kd1C3	1ar840	1pspB2	1gotG0	2hexD0	1ahc02	1ihfA0	2prgC0
1n8rC3	1ar940	1pcp02	1a0rG0	2hexE0	1mrh02	1owfB0	1g1xC0
1njiC3	1asj40	1jouA0	1tbgE0	3tgjI0	1mom02	1ouzB0	1g1xH0
1q86C3	1po140	1jouC0	1tbgF0	1bthP0	1aha02	1owgB0	1fkaR0
1k73C3	1po240	1jouE0	1tbgG0	1bthQ0	1ahb02	1ihfB0	1ekcC0
1kc8C3	1vbd40	1jmoL0	1tbgH0	2kaiI0	1f8qA2	1exeA0	1ekcH0
1kqsA3	1eah40	1etrL0	2trcG0	1bz5A0	1mri02	1exeB0	1fjgR0
1qvfA3	1pvc40	1ucyJ0	1b9xB0	1bz5B0	1nioA2	1wtuA0	1n32R0
1m1kC3	1vbb40	1ucyL0	1b9yB0	1bz5C0	1bryY2	1wtuB0	1j5eR0
1q7yC3	1vbc40	1ucyM0	1gp2G0	1bz5D0	1bryZ2	1jc9A2	1iblR0
1w2bA3	1vbe40	1bbrJ0	1gg2G0	1bz5E0	1cf5A2	1fib02	1i94R0
1s72A3	1piv40	1bbrL0	1omwG0	1cbwD0	1cf5B2	1fid02	1hr0R0

续表

1jj2B3	1vba40	1bbrM0	1hfeS0	1cbwI0	1d8vA2	2fibA2	1hnzR0
1s72B3	1d4m40	1etsL0	1hfeT0	1bhcA0	1gikA2	3fib02	1ibkR0
1m90D3	1cov40	1ettL0	1ioj00	1bhcB0	1j1qA2	1fzcC2	1ibmR0
1qvgB3	1h8tD0	1avgL0	1wdcA0	1bhcC0	1j1rA2	1fzcF2	1n33R0
1q81D3	1oopD0	1tbrJ0	1scmA0	1bhcD0	1j1sA2	1ficA2	1hnwR0
1q82D3	1mqtD0	1tbrL0	1kk8A7	1bhcE0	1qciA2	1ficB2	1hnxR0
1k8aD3	1ev140	1hrtL0	1b7tA7	1bhcF0	1qciB2	1fzgC2	1n36R0
1k9mD3	1ncqD0	1tbqJ0	1sr6A7	1bhcG0	1d6aA2	1fzgF2	1n34R0
1kd1D3	1k5mD0	1tbqL0	1l2oA7	1bhcH0	1d6aB2	1fzfC2	1fjfR0
1n8rD3	1r0940	1vitL0	1kqmA7	1bhcI0	1qcgA2	1fzfF2	1aoo00
1njiD3	1rud40	1vitM0	1s5gA7	1bhcJ0	1qcgB2	1lt9C2	1aqq00
1q86D3	1rue40	1mkxL0	1kk7A7	1b0cA0	1qcjA2	1lt9F2	1aqr00
1k73D3	1ruf40	1mkwL0	1kwoA7	1b0cB0	1qcjB2	1ltjC2	1fmyA0
1kc8D3	1hri40	1uvtL0	1qviA7	1b0cC0	1pafA2	1ltjF2	1aqs00
1kqsB3	1hrv40	1ycpJ0	1oe9A6	1b0cD0	1pafB2	1fzaC2	1d6gA0
1qvfB3	1r0840	1ycpL0	1nkpA0	1b0cE0	1pagA2	1fzaF2	1cf4B0
1m1kD3	1rmu40	1id5L0	1nkpD0	1mtnD0	1pagB2	1fzbC2	1b35D0
1q7yD3	1rug40	1uvuL0	1nkpB0	1mtnH0	1apa02	1fzbF2	1cwxA0
1w2bB3	1ruh40	1tocA0	1nkpE0	1eawB0	1m2tA2	1fzeC2	1devB0
1jeyA3	1rui40	1tocC0	1nlwB0	1eawD0	1sz6A2	1fzeF2	1devD0
1jeqA3	1ruj40	1tocE0	1nlwE0	1uuaA0	1puuA2	1fzdA2	1j34C0
1fqjC0	1vrh40	1tocG0	1an2A0	1oa550	1pumA2	1fzdB2	1j35C0
1jj2K1	2hwb40	1g72B0	1an2C0	1oa650	1onkA2	1fzdC2	1nl0G0
1s72L1	2hwc40	1g72D0	1hloA0	1pit00	1tfmA2	1fzdD2	1cfi00
1m90M1	2r0440	4aahB0	1hloB0	9pti00	1pc8A2	1fzdE2	1mgx00
1qvgK1	2r0640	4aahD0	1r05A0	8pti00	1oqlA2	1fzdF2	1iodG0
1q81M1	2r0740	1b2nB0	1r05B0	1nag00	1ce7A2	1fzdG2	1whe01
1q82M1	2rm240	1b2nD0	1nlwA0	1fan00	2mllA2	1fzdH2	1whf01
1k8aM1	2rmu40	3aahB0	1nlwD0	1bpt00	1qi7A2	1fzcB2	1p0sL1
1k9mM1	2rr140	3aahD0	1am9A0	2tpiI0	1abrA2	1fzcE2	1cfh00
1kd1M1	2rs140	1h4iB0	1am9B0	1bti00	1ggpA2	1fzgB2	1hykA0
1n8rM1	2rs340	1h4iD0	1am9C0	4tpiI0	1dm0L2	1fzgE2	1qu8A0
1njiM1	2rs540	1h4jB0	1am9D0	1jv8A0	1dm0A2	1fzfB2	1b8xA3
1q86M1	4rhv40	1h4jD0	1a0aA0	1jv9A0	1r4qA2	1fzfE2	1kekA4
1k73M1	1ruc40	1h4jF0	1a0aB0	1aalA0	1r4qL2	1lt9B2	1kekB4
1kc8M1	1na1D0	1h4jH0	1mdyB0	1aalB0	1hwmA2	1lt9E2	1b0pA4
1kqsK1	1rvf40	1lrwB0	1mdyC0	7pti00	1hwnA2	1ltjB2	1b0pB4
1qvfK1	1rhi40	1lrwD0	1mdyD0	3bthI0	1hwoA2	1ltjE2	2pdaA4
1m1kM1	1bev40	1olgA0	1mdyA0	3btfI0	1hwpA2	1fzaB2	2pdaB4
1q7yM1	1w2bK1						

说明：附表 4 中使用的是 CATH 数据库六字母代码

源自：http://www.biochem.ucl.ac.uk/bsm/cath/class.html

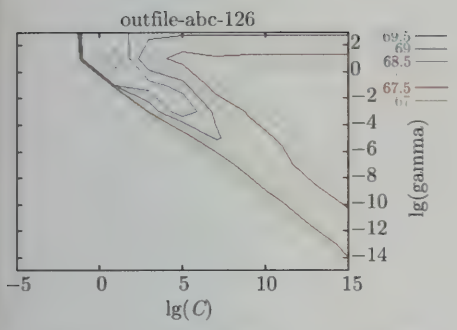


图7-8　非冗余数据集 RS126 的参数 C 和 γ 的优化结果

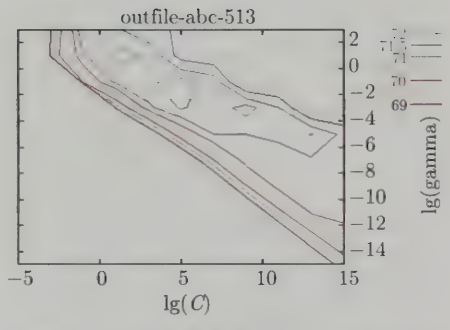


图7-9　非冗余数据集 CB513 的参数 C 和 γ 的优化结果

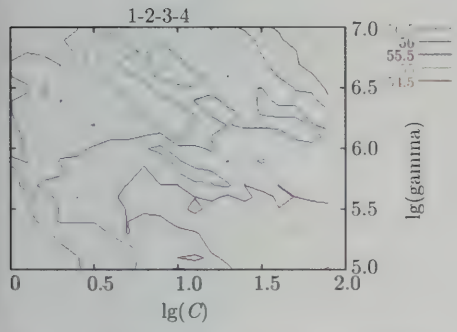


图8-3　拓扑层的820个样本的一肽频数输入样本集的28重交叉验证试验优化结果

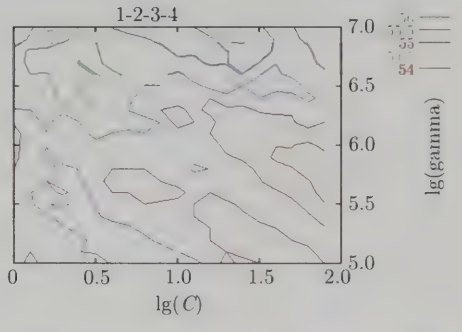


图8-5　拓扑层的820个样本的一肽频数输入样本集的Jackknife优化结果

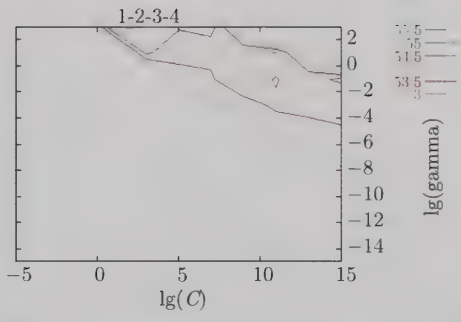


图8-6　拓扑层的820个样本的一肽频数输入样本集的默认参数 C 和 γ 的优化结果

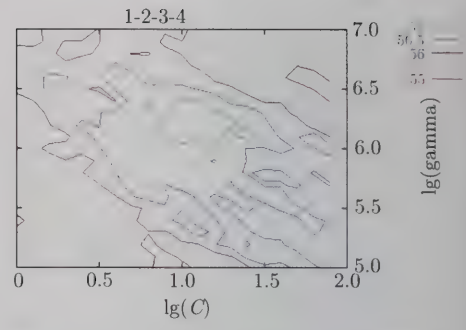


图8-7　拓扑层的820个样本的一肽频数输入样本集的7重交叉验证试验优化结果

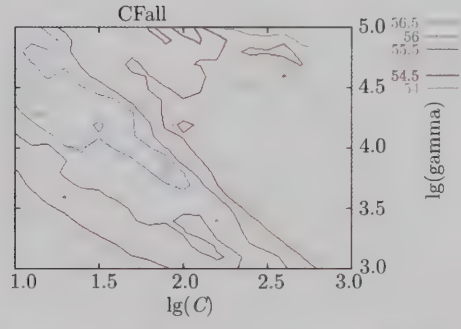


图8-9　拓扑层的820个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

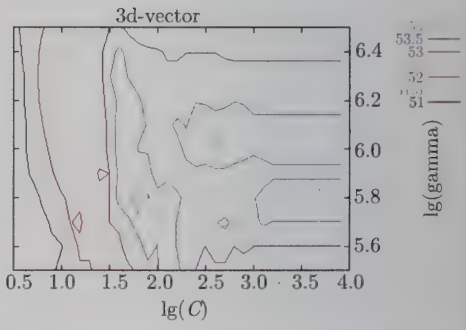


图8-11　拓扑层的820个样本的三肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

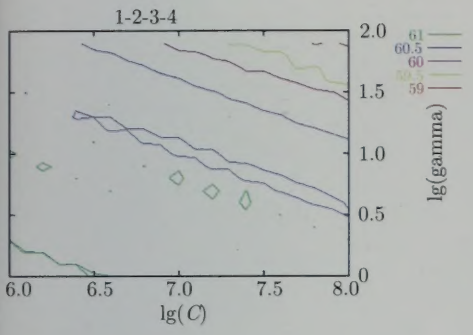


图8-14　同源超族层的1572个样本的一肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

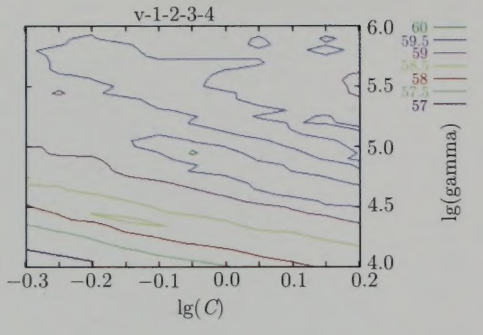


图8-16　同源超族层的1572个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

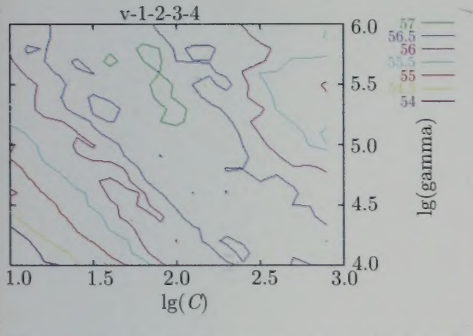


图8-18　同源超族层的1572个样本的三肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

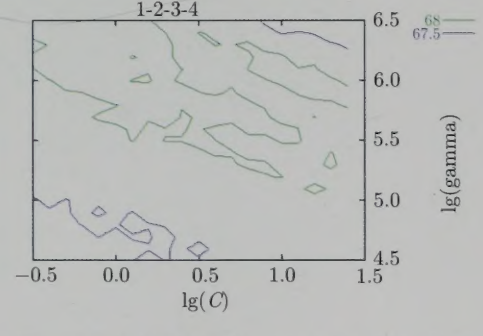


图8-21　序列家族层的6957个样本的一肽频数输入样本集的参数 C 和 γ 经过微调后的 7 重交叉验证试验优化结果

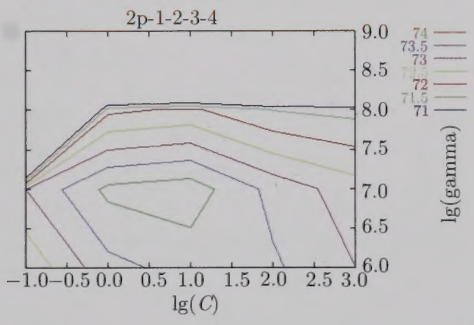


图8-23 序列家族层的6957个样本的二肽频数输入样本集的参数 C 和 γ 经过微调后的7 重交叉验证试验优化结果

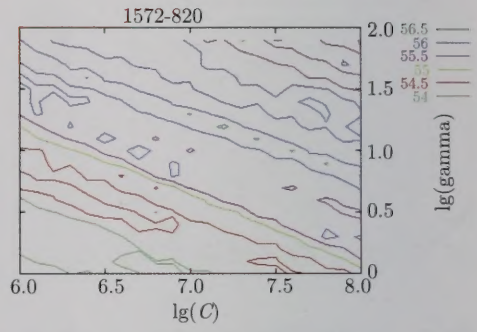


图8-28 同源超族层的820个样本的一肽频数输入样本集的7重交叉验证试验优化结果

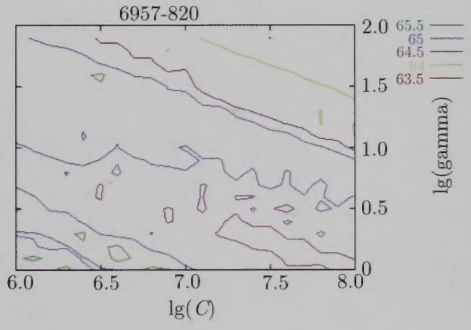
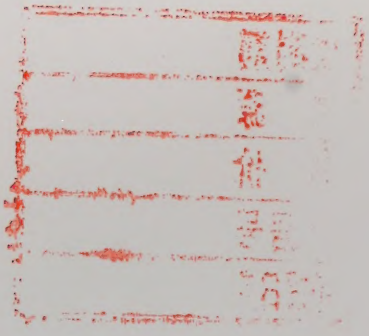


图8-30 序列家族层的820个样本的一肽频数输入样本集的7重交叉验证试验优化结果



(Q-2134.0101)

现代生物技术前沿丛书

书　名	书号　ISBN	定价
基因免疫的原理和方法	7-03-012588-6	38元
DNA芯片和基因表达（影印版）	7-03-012248-8	29元
组织工程（影印版）	7-03-013407-9	55元
生物芯片分析(影印版)	7-03-012247-X	80元
蛋白质芯片(影印版)	7-03-015183-6	65元
肽：化学与生物学(翻译版)	7-03-014874-6	75元
药物基因组学		
——寻求个性化治疗（翻译版）	7-03-015612-9	78元
生命科学中的单分子行为及细胞内实时检测	7-03-015448-7	68元
细胞通讯与疾病	7-03-016524-1	78元
遗传修饰植物	978-7-03-018102-2	65元
RNA干扰技术		
——从基础科学到药物开发	978-7-03-018859-5	85元
比较基因组学	978-7-03-019430-5	38元
表观遗传学（影印版）	978-7-03-020774-6	96元
干细胞移植		
——机理与临床	978-7-03-021575-8	58元
结构生物学与现代药学研究	978-7-03-022253-4	75元
蛋白质结构预测		
——支持向量机的应用	978-7-03-022387-6	50元

生物分社
联系电话：010-64012501
http://www.lifescience.com.cn
e-mail:lifescience@mail.sciencep.com

销售分类建议：生物医学/生物信息学

ISBN 978-7-03-022387-6
9 787030 223876 >

定 价：50.00 元