Q/CZCB

浙 江 稠 州 商 业 银 行 企 业 标 准

Q/CZCB 10601—01

信贷智能风控业务规范

Credit intelligent risk control business specification

（报批稿）

（本稿完成日期：2020-05）

2020－05－31 发布　　2020－06－01 实施

浙江稠州商业银行　　发 布

目　　次

企业标准信息公共服务平台
公开 2020年09月30日 10点34分

前　言

本标准按照GB/T 1.1-2009标准化工作导则第1部分：标准的结构和编写给出的规则起草。

请注意本标准的某些内容可能涉及专利。本标准的发布机构不承担识别这些专利的责任。

本标准起草单位：浙江稠州商业银行。

本标准主要起草人：XX

企业标准信息公共服务平台
公开
2020年09月30日 10点34分

引　言

信贷智能风控业务规范性文档是运用系统理论指导信贷智能风控体系标准化工作的一种方法。本规范性文档是开展信贷智能风控体系建设的基础和前提工作，也是修订规划和计划的依据。

随着金融科技的发展，信贷机构风险管控能力面临新的挑战与机遇，融合了大数据技术、人工智能等科技的信贷智能风控体系应运而生。本规范性文档为信贷智能风控体系的建立提供指导支持，主要包括在以下几个方面：

--信贷智能风控概念，包括信贷智能风控定义，信贷智能风控目的和作用；

--信贷智能风控过程，包括信贷智能风控过程概述，反欺诈，风险筛查，跟踪预警，风险应对，效果评价及调优；

--信贷智能风控技术，包括数据获取，数据保护，数据存储，数据校验、加工与整合，模型开发，模型应用；

--信贷智能风控实施，包括团队，基础设施，数据，管理机制。

信贷智能风控业务规范

1 范围

本标准规定了信贷智能风控概念、信贷智能风控过程、信贷智能风控技术及信贷智能风控实施指南。

本标准并未涉及信贷智能风控的所有技术，标准中未予以介绍的技术并不意味着其无效。

本标准适用于指导组织识别信贷智能风控概念、建立信贷智能风控过程、选择信贷智能风控技术及具体落实信贷智能风控实施。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 23694-2013 风险管理术语

GB/T 24353-2009 风险管理原则与实施指南

GB/T 27921-2011 风险管理风险评估技术

有关物联网技术方面的标准参考以下文件：

GB/T 33474-2016 物联网参考体系结构

GB/T 36951-2018 信息安全技术物联网感知终端应用安全技术要求

GB/T 37024-2018 信息安全技术物联网感知层网关安全技术要求

GB/T 37025-2018 信息安全技术物联网数据传输安全技术要求

GB/T 37032-2018 物联网标识体系总则

GB/T 37044-2018 信息安全技术物联网安全参考模型及通用要求

有关大数据方面的标准参考以下文件：

GB/T 4943.23-2012 信息技术设备安全第23部分：大型数据存储设备

GB/T 35274-2017 信息安全技术大数据服务安全能力要求

GB/T 35295-2017 信息技术大数据术语

GB/T 35589-2017 信息技术大数据技术参考模型

3 术语和定义

下列文件中界定的术语和定义适用于本文件。

GB/T 5271.31-2006 信息技术词汇第31部分:人工智能机器学习

GB/T 5271.34-2006信息技术词汇第34部分:人工智能神经网络

GB/T 23694-2013 风险管理术语

GB/T 33745-2017 物联网术语

GB/T 35295-2017 信息技术大数据术语

4 信贷智能风控概念

4.1 信贷智能风控定义

信贷智能风控以大数据为基础，以科技为依托，通过机器学习模型和算法等模拟人类的智慧和能力，运用于信贷反欺诈、风险筛查、跟踪预警、风险应对等风险管控领域，实现数据驱动的精益化风险管理。

4.2 信贷智能风控目的和作用

信贷智能风控旨在：利用大数据及人工智能技术，搭建以反欺诈、客户风险筛查、跟踪预警、风险应对为核心的“一体化”风控体系，全面提高信贷风险控制能力，提升核心竞争力。

信贷智能风控的主要作用包括：

——深入挖掘数据价值，推动风险管理与大数据高度融合；

——全方位防控信用风险、欺诈风险、操作风险和输入性风险；

——建立覆盖风险识别、计量、分析、处置全流程的信贷智能风控体系。

5 信贷智能风控过程

5.1 信贷智能风控过程概述

信贷智能风控综合运用了人工智能、大数据分析、知识图谱等先进技术，覆盖贷前、贷中和贷后三个阶段，以机器自动决策为主，人工审核为辅，有效识别、筛查、预警、应对风险，实现信贷业务全流程的智能化。

信贷智能风控流程和信贷产品设计密不可分，两者有机结合，共同促进信贷业务健康发展。在信贷产品设计阶段，应考虑产品贷前、贷中、贷后各环节风险点，将风险前置，平衡风险与收益，合理设计信贷产品。

5.2 反欺诈

信贷欺诈是发生在信贷领域内的欺诈行为，具体指信贷机构内部人员、客户或第三方，单独或伙同他人，通过虚构事实或隐瞒真相的方法，从信贷机构或客户骗取不正当好处或利益，造成信贷机构资金、声誉或其他损失的行为。

信贷智能风控体系下反欺诈主要是在信贷业务发生前对欺诈行为的识别，分为两个方面，即线上反欺诈模块和线下反欺诈模块。线上反欺诈，主要针对客户线上申请产品时，根据客户信息核验、客户特征分析、客户关联图谱等方式进行反欺诈检验，主要运用Logistic模型、神经网络模型等相对成熟的反欺诈技术进行线上反欺诈识别与检验。线下反欺诈，主要根据专家经验，产品、客户类型的特征和信贷机构实际情况，制定相应的反欺诈规则，从中识别可能的欺诈类客户。

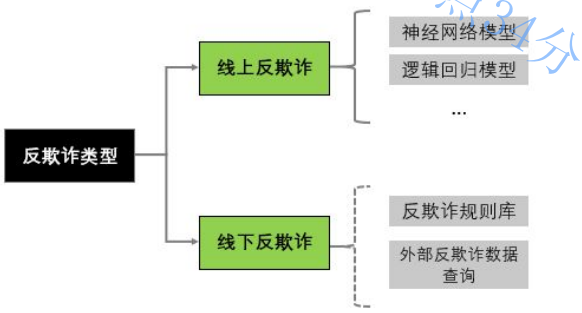


图 1

线上申请客户由于具有申请量大，审批需求多，容易获取客户特征且无法当面核查客户信息等特点，通常采用建立模型的方式识别欺诈类客户。线上反欺诈具体流程通常包括：建立单方面反欺诈模型；机器集成学习，集成单方面模型；定期更新模型，验证模型效果，更新欺诈类客户行为特征等步骤。

对于线下反欺诈策略可采取通过挖掘专家经验、信贷机构内部数据、外部大数据，设计完整的反欺诈规则库，并从中提炼出反欺诈规则。通过反欺诈规则，判断客户的欺诈风险程度，从而给出相关决策建议。专家经验数据挖掘规则流程包括：（1）根据同行业实践和专家经验，提炼出反欺诈规则；（2）针对产品、客户特点和行内实际情况，筛选、修改相关规则；（3）通过对行内历史数据进行验证测算，查验规则效果，最终生成反欺诈。内部数据挖掘规则流程包括：（1）获取行内历史“类欺诈”客户样本及数据，进行欺诈变量分析；（2）对欺诈相关变量进行特征信息的相似度分析，列出各阈值下的欺诈客户占比；（3）根据统计结果及经验，选取合适的阈值，设定反欺诈规则。外部数据规则挖掘流程包括：（1）对行内外部大数据进行分析，选出反欺诈相关指标；（2）直接引入严重类的名单性质指标，设定名单查询类反欺诈规则。

反欺诈策略模块可分为四类：

（一）反欺诈模块：主要针对信贷机构及外部数据名单类的检查；

（二）征信体检报告模块：针对申请规则中与征信相关的强规则，进行预先提醒；

（三）基础信息检查模块：根据录入的业务信息进行反欺诈的基础信息检查；

（四）业务人员异常行为检查模块：后台检查业务人员的欺诈风险，提示给审批人员。

反欺诈模块为申请开始最初的查验，对三要素输入（身份证、姓名、手机号）进行核验，免去后续征信查询及业务信息录入的时间及成本，在第一道风险把控关卡就提示业务人员该笔业务的欺诈严重程度。该模块从内部数据、外部数据、专家经验等维度进行提炼，主要提取名单类严重规则及基础信息的检查。欺诈等级分为两类，拒绝类为提示出严重触犯了反欺诈模块的规则，欺诈程度较高，建议直接拒绝该客户的申请；提示类为轻度触犯了反欺诈规则，欺诈程度一般，需业务人员进一步核查。

征信体检报告模块是将申请策略中的征信类规则进行查验，对全部产品的准入规则均存在的规则（非某产品特色的征信类规则）进行提前展示，在获取客户征信查询后，即可展示于业务人员及审批人员，。业务人员可以根据展示的内容判断是否继续提交或者修改提交信息。征信体检报告属于对征信报告内容的分析解读，里面涵盖了逾期类、征信查询类、特殊事项类等强规则，此规则库分为人工审核、建议拒绝及直接拒绝。

基础信息检查模块基础信息检查均为提示类规则，根据客户录入的客户信息及业务信息，进行逻辑检查、与征信核对及信息检查，发现客户存在欺诈嫌疑的关键点，并对业务人员进行提示。其中，逻辑检查是对申请人资料中存在不合理的地方进行提示，此种不合理有一定欺诈嫌疑；征信核对主要是对比征信系统与客户申请的资料存在差异的地方，找出异常点；信息检查是对命中名单、特殊字段进行抓取，排除欺诈风险。

业务人员异常行为检查模块为排查业务人员可能存在的与客户共同欺诈的嫌疑，需要进行一定规则检查，例如业务人员近亲属任职情况、业务人员家属与客户资金往来、业务人员对外任职问题等。

5.3 风险筛查

信贷智能风控通过系统内嵌科学风险评估计量模型与规则进行风险筛查，在信贷业务进行中，实现系统自动化建议各场景下多维网状风险筛查结果。风险筛查结果可以运用于客户准入审批、额度及利率定价等方面。

模型作为一个综合体反映整体的、主要的、标准的风险规律，是风险判断的核心。规则是辅助手段，捕捉局部的、次要的、特殊的风险规律，如根据经验的极端风险情况、产品政策、地域等当前尚不适合或难以在模型中充分体现出来的风险特征，将以规则的形式进行补充。信贷智能风控体系下模型+规则的风险筛查模式具有批量化、标准化，自动化的特点，可以提高效率、节约成本。

规则梳理和提炼主要基于风险的区分性、数据的可获取性，规则设计的基本原则为“主观因素客观化，客观因素定量化，定量因素标准化”，信贷智能风控体系下系统自动计算规则结果。

信贷智能风控在判定一笔申请准入审批时分别计算模型决策结果和规则决策结果，然后将两部分的决策结果取孰严重自动计算最终准入决策结果。模型决策结果刻画了客户基础风险画像，规则决策结果分为三类：R-人工审核，RD-建议拒绝，D-快速拒绝。结合模型结果与规则结果形成差异化的准入审批。

信贷智能风控额度策略主框架为模型结果与风险缓释的结合，即根据风险评估模型结果与风险缓释品的种类判别客户适当的授信额度。抵押类产品额度可以根据押品决定，偏重对押品的评估和认定，额度策略主框架为模型结果与抵押、担保因素（押品类型、主担保方式）。无抵押类产品主要考虑模型结果、第一还款来源（可支配收入、经营利润等）、净资产情况，并适当考虑担保情况。

信贷智能风控每一笔信贷业务利率的确定在基础利率上进行调整，需要考虑的调整因素包括客户风险等级、抵押品情况、客户对信贷机构的综合贡献度（如存款、理财等）、客户与信贷机构的历史合作（如合作年限）等。信贷机构需要确定不同的产品需要考虑的差异化利率调整因素，并针对每个因素（或因素组合）确定差异化调整映射表。其中，可以根据模型结果确定客户风险等级的调整系数；在合作年限方面，对与信贷机构合作一定年限以上的客户可以给予一定利率的折扣；客户的综合贡献度，可考虑存款、理财等（若有）的日均余额请款，对综合贡献达到一定水平的客户可以给予一定利率的折扣。给予的利率折扣需要信贷机构根据实际情况进行决策。

5.4 跟踪预警

跟踪预警的目标是在信贷业务发生后，对客户风险进行跟踪识别，提早防范风险。信贷智能风控跟踪预警利用海量外部数据的信息整合优势，将人行征信、工商、法院等数据集中处理，及时发出预警信号，为存量客户风险评估提供决策依据同时督促业务人员及时处置并提出解决措施。同时，充分利用行内数据，对客户账户状态、资金交易流水、押品状态等进行监测，保障客户信贷资金用途的合规性，有效防范风险。

预警系统主要包含了两块内容，一个是预警信号的设置，另一个是预警流程的处置。预警信号应按照严重程度划分等级。客户风险等级则是根据触发各级预警信号的数量来定义。预警流程的处置包括预警发起、预警处置、预警解除三个步骤。预警发起过程包括系统发起及业务人员手工发起两种情况；预警处置是根据客户所对应的预警等级，按照设定的处置流程，实行分级审批，对客户风险状况进行认定，以客观、真实、准确地反映所客户的风险状况；预警解除根据解除方式的不同，可分为系统自动解除及人工解除两种。

5.5 风险应对

信贷智能风控体系下，信贷机构基于内部数据及外部海量数据分析结果，形成智能风险应对系统，建立主动预防、全场景、立体化的智能风控体系，针对信贷业务发生后客户风险行为进行管理、识别和催收。

信贷智能风控对于存量客户行为，结合大数据、人工智能技术，识别客户贷后风险程度，自动形成客户使用频率、还款能力及还款意愿的客户画像，精细化制定永久额度调整与额度临增策略。人工智能技术下，可自动发送语音/文本形式额度调整通知。

催收是逾期款项的一种回收方式。智能风控综合运用大数据和人工智能技术，根据不同客户画像，及时优化调整催收策略，提高催收效果，节约信贷机构催收成本。智能催收体现在：

（一）逾期客户画像，精细化刻画逾期客户还款能力与还款意愿维度；

（二）利用人工智能技术，建立催收模型，结合多维度数据预测逾期客户催收难度；

（三）根据催收案件难易级别分配不同等级催收人员，实现催收资源合理分配；

（四）利用大数据和人工智能技术，与失联逾期客户复联，适当催收；

（五）自动化催收技术，由系统自动拨号或发送短信/邮件，节省催收人力资源；

（六）语义识别客户情绪，选择合适有效的催收话术。

5.6 效果评价及调优

信贷智能风控体系需要在实际业务过程中根据信贷业务需求和技术的发展，不断迭代优化。因此信贷智能风控效果评价及调优在整个信贷智能风控体系中具有重要地位，同时也是十分必要的。

信贷智能风控效果评价及调优包括对信贷智能风控模型、规则、策略技术效果的评价及优化，也包含对整体信贷智能风控体系、框架及配套制度的效果评价及优化。应从技术结果的准确性、及时性，风险流程的规范性、合理性，制度体系的完整性、落地性等方面进行考量。

阶段一：基于信贷智能风控结果调整

信贷智能风控积累一定的数据量之后，根据信贷智能风控模型区分度指标判定是否需要对模型调整优化，根据模型结果分布判定是否需要对规则和策略进行调整。

阶段二：基于信贷智能风控风险表现调整

在信贷智能风控数据积累相对充分之后，根据智能风控模型结果分布及坏账表现，判定模型是否需要调整优化或者重新开发，规则和策略是否需要进行调整。同时，鉴于信贷智能风控制度体系运营已经有一定的时间积累，可对流程及制度体系效果进行评价并做出适时调整。

6 信贷智能风控技术

6.1 数据获取

6.1.1 数据获取需求

信贷智能风控体系的数据具有获取渠道种类繁多的特征。从信贷产品设计阶段开始，就应该从产品应用场景的角度出发，考虑数据获取的渠道。信贷智能风控体系内部数据获取渠道为信贷机构业务过程存储，外部数据获取渠道包括但不限于政府权威机关、数据供应商、互联网爬取等。

6.1.2 数据内容维度

信贷智能风控引入外部数据，融合内部数据，数据维度包括但不限于：

（一）基本信息：例如性别、年龄、学历、证书、婚姻状况、子女情况、居住地、户口所在地等；

（二）工作信息：工作单位、所属行业、职位、职务、收入水平、行业/所在地区平均收入水平、收入来源、社保公积金缴纳情况、是否小企业主、企业经营业绩等；

（三）公共信息：法院诉讼、执行、黑名单信息；税务负面信息（针对法定代表人）；

（四）房产信息：是否自有住宅、房产数量及价值、按揭贷款数量及金额、是否容易变现等；

（五）金融资产信息：他行及我行资产（含理财产品、股票、贷款、信用卡等）；

（六）消费信息：电商购物行为、银联消费交易行为、出行行为、上网浏览行为、移动通话缴费行为等；

（七）生活信息：欠费信息：运营商、水电煤、有线电视等；

（八）逾期信息：P2P、信贷机构等；

（九）各行政部门发布的处罚信息：如道路、轨交违章；

（十）车辆信息：驾驶证、自有车辆数量及价值、按揭贷款数量及金额等；

（十一）社交信息：联系人数量、联系人所在行业分布、联系人年龄层次、社交频率、社交方式等。

6.1.3 数据获取方式

内部数据可采用实时获取或批量获取的方式。实时获取适用于数据量较小且对实时性要求较高的场景，实现方式包括webservice、企业服务总线、广播等。批量获取适用于数据量较大且对实时性要求不高的场景，通常是T+1模式，实现方式包括ETL、调度平台等。

外部数据获取方式包括人工采集、第三方批量获取、公开数据获取及物联网获取等。其中，根据GB/T 33745-2017 物联网术语物联网指通过感知设备，按照约定协议，连接物、人、系统和信息资源，实现对物理和虚拟世界的信息进行处理并作出反应的智能服务系统。物联网层级机构可以分为感知层、网络层及应用层，通过部署传感器获取物体的状态信息和外部环境信息，并通过传感网络实现数据信息的初步处理和交互传输。

6.1.4 外部数据评估方法

信贷智能风控外部数据评估方法包括六个方面：

（一）评估该类数据的市场份额，例如案例的个数和代表性；

（二）数据公司的数据来源，例如是否为一手数据源；

（三）数据来源合法合规性，例如是否获得授权；

（四）提供数据的维度，例如数据维度的丰富程度；

（五）提供的数据质量，例如数据的完整性、查的率、准确率及更新频率；

（六）提供数据的方式，例如批量打包，网站查询等。

信贷智能风控外部数据评估方法包括六个步骤：

（一）行业数据抽样；

（二）数据商反馈结果；

（三）分析反馈数据维度；

（四）比对数据的查的率；

（五）对比权威数据源判断准确性；

（六）得出测试结果。

6.2 数据保护

6.2.1 使用授权

根据信贷智能风控数据源特征及使用需求，制定信贷智能风控数据使用权限。使用授权应采取保守原则，选择最小的权限满足数据使用部门需求。数据使用部门将依据使用权限办法获取并使用信贷智能风控数据。

6.2.2 敏感信息加密

信贷智能风控数据涉及到敏感信息，需要对此类信息进行加密。敏感信息加密为通过特定加密算法将原始未处理信息（即明文）转换为处理后的信息（即密文），将密文还原为明文，需要特定密钥进行解密。

根据信贷智能风控数据敏感级别和标准法规要求，如《中华人民共和国商用密码管理条例》、《信息系统安全等级保护基本要求》等，选择适当的加密算法类型、最小密钥长度以及加解密机制的实现方式。

6.2.3 传输加密

信贷智能风控数据传输环境的信息泄露防范，应遵循相应的传输加密管理要求。对于需通过互联网平台传输的信贷智能风控数据，必须采取信道级加密措施对数据机密性和完整性提供保护。对于需通过广域网平台传输的信贷智能风控数据，应采用适当的报文加密措施对数据机密性和完整性提供保护。

6.2.4 数据防篡改

信贷智能风控数据应建立数据防篡改机制，例如根据关键数据生成原始校验码，定期发起校验，校验时，比对由关键数据生成的密文与原始校验码，检测关键数据是否被篡改。

6.3 数据存储

数据存储是指存储器将采集到的信贷智能风控数据存储的过程，通过建立相应的数据库，进行管理和调用。主要解决信贷智能风控数据的可存储、可表示、可处理、可靠性及有效传输等问题。重点解决复杂结构化、半结构化和非结构化数据管理与处理技术。

信贷智能风控数据存储应满足但不限于以下几种基本要求：

（一）大容量：信贷智能风控数据总量呈指数上升，数据存储备份系统必须具有足够的容量以适应不断增长的数据量。存储系统不光要有大量的现实容量，还应该具有很好的可扩展性，能根据数据量的增长提供无缝的、不停机的容量扩充；

（二）高性能：信息是具有时效性的，及时获得所需的信贷智能风控数据非常关键；较高的访问速度是服务质量的重要指标。对于宽带应用，存储系统的带宽要与网络带宽相适应。因此，存储系统的响应速度和吞吐率是数据存储备份的必须要求；

（三）高可用性：数据存储备份存储了大量的信贷智能风控关键数据，因此，必须保证这些数据始终是安全可用的。在任何情况下，例如系统产生错误或遇到意外灾难，数据都不能丢失。系统应具有快速故障恢复能力，保证应用系统永不停机（7×24小时不间断工作），数据始终保持完整性和一致性；

（四）可管理性：对信贷智能风控数据的管理不光体现在应用层的管理，还体现在存储系统的管理。这主要表现在集中的自动化管理，如数据按特定规则的备份、对系统性能和流量等特性的监测、存储设备的负载平衡等。

6.4 数据校验、加工与整合

信贷智能风控数据需进行校验、加工与整合，保证数据的完整性、准确性与一致性。将信贷智能风控数据划分为结构化数据和非结构化数据，结构化数据可采用提取、转换、加载(ETL)工具进行处理，非机构化数据可采用OCR影像识别交叉校验、自然语言处理、知识图谱等方式进行处理。

ETL技术分三部分：数据抽取、数据的清洗转换、数据的加载。数据的抽取是从各个不同的数据源抽取到统一的数据存储软件中。数据的抽取分为全量抽取（适用于系统数据初始化时）和增量抽取（适用于数据库中发生数据新增、修改或者抽取数据量比较大时）。数据清洗目的为过滤不符合要求的数据，包括不完整的数据、错误的数据、重复的数据三大类。数据转换为进行不一致的数据转换、数据颗粒的转换，以及一些业务规则的计算。

OCR（Optical Character Recognition）影像识别技术为将印刷品文字（例如票据报纸等）通过光学扫描，检测影像暗、亮模式确定图像信息，再根据字符识别方式，将形状转化为可以使用的计算机输入技术。衡量OCR系统性能的主要指标为：最终识别率、识别速度、版面理解正确率及版面还原满意度等。

自然语言处理为通过智能、高效的方式，对非结构化数据（例如新闻、政策、社交媒体等）进行系统化分析、理解与信息提取的过程。自然语言处理运用了推测学，概率学及统计学知识，其范畴包括语音识别、文字识别、语义识别、智能问答、信息抽取等。

知识图谱的基础是多维度海量数据库、语言关系认知能力和知识库表示结构。知识图谱整合和关联结构化数据、非结构化数据，将所有数据以实体-关系-事件-属性的形式存储，直观展示数据以及数据背后的关联。知识图谱包括信息抽取、知识表示、知识融合及知识推理。

6.5 模型开发

模型开发是构建信贷智能风控体系的核心。信贷智能风控体系模型包含如下几类：

（一）通过积累大量数据总结规则，辅助信贷智能风险决策，如身份核验、信贷黑白灰名单库、人脸识别等。此类模型简单易操作，但是规则创建依赖专家经验和已发生风险事实，新风险因素无法及时更新；

（二）数据挖掘算法模型，将信贷智能风控数据汇总分析，常用的数据挖掘算法包括聚类分析、逻辑回归、决策树、神经网络等。此类模型通过训练模型，不断优化和调整模型，在模型精度和适用性上有了质的提升；

（三）利用机器学习、人工智能等新技术多维度分析风险因素，例如支持向量机、集成学习、降维与度量学习等。此类模型通过不断自我改进与自我优化，自动识别新的风险模式，提升了信贷智能风控体系的快速反应能力。

模型开发技术详见附录A（资料性附录）信贷智能风控技术。

6.6 模型应用

模型应用采取了客户风险类别区分技术，包括“及格分数线”和“风险等级区间”。“及格分数线”依赖评分进行业务决策的理想情况，理论上低于某一分数以下的直接拒绝，高于该分数以上则批准。“风险等级区间”通过设置评分区间将人群分成不同风险类别，不同类别采取差异化对策。

及格分数线的划分根据业务方向的不同而设置不同的原则，一般主要包括以下三种考虑：（一）保持目前的审批通过率；（二）保持目前的坏账率；（三）同时提高审批通过率和降低坏账率。一般通行做法是综合考虑风险（坏账率）和收益（审批通过率）之间的平衡，即在能接受的坏账率水平和审批通过率之间综合考虑。

实际业务上需要更多、更精细的划分，以对客户进行风险差异化，不同风险等级区间的申请采取差异化信贷措施。风险等级区间考虑lift提升率，整体坏账率等指标，结合经验和业务需求综合确定。评分等级划分遵从“业务出发，数据参考”的原则：人群占比、坏账率差异明显、分数圆整。

7 信贷智能风控实施

基于信贷智能风控落地实施目的，还应配备专业的团队人员、高性能的基础设施、安全合规的信贷智能风控数据、完善的信贷智能风控管理机制。

7.1 团队

信贷智能风控体系是一项复杂的，需要管理、统计、金融、计算机科学等多项技术的工程。建立并发展信贷智能风控体系需要持续投入高素质的专业团队人员。

信贷智能风控团队应至少包括信贷专家小组、模型专家小组、信息技术专家小组。各专家小组职能明确，职责清晰，即分工负责智能风控体系各专业条线，又协同合作在实践中不断优化信贷智能风控体系。

信贷专家的职责为：

（一）负责识别现有及新增信贷业务风险点；

（二）负责与模型专家及信息技术专家对接风险管控流程；

（三）负责制定具体的应用方案，应用策略覆盖贷前、贷中、贷后；

（四）负责设计并维护策略监测指标体系。

模型专家的职责为：

（一）负责制定信贷智能风控模型细分方案；

（二）负责开发并持续优化信贷智能风控模型。

信息技术专家的职责为：

（一）负责评估外部数据需求；

（二）负责保护、存储、校验、加工与整合信贷智能风控数据；

（三）负责开发并维护信贷智能风控系统。

信贷智能风控专业团队可通过定向培养、专业培训、引进核心人才等渠道建立。

7.2 基础设施

信贷智能风控的落地实施依赖于高性能的基础设施，包括软件架构和硬件设备两个方面。智能风控的基础设施应该能够解决海量数据容纳问题、多业务数据源整合问题、多数据格式转换问题，高效融合结构化、非结构化数据的管理，实现灵活、高效的分布式存储和运算。

7.3 数据

信贷智能风控数据应从合规性、稳定性、穿透性、前瞻性、验证性五个维度满足业务运营、监管要求、管理与决策需求。

合规性原则：合法合规是数据使用的核心原则，数据的使用必须遵守法律法规的要求。。

稳定性原则：稳定性主要是指数据来源的稳定性，具体包括两个方面：一是数据供给的可持续性。数据是能够长期提供的，否则可能影响信贷机构业务的连续性；二是数据口径的稳定性。数据口径是业务内涵的体现，数据口径的变动会影响数据的可用性，从而影响业务应用。

穿透性原则:穿透性主要是指数据源的本源性。信贷机构在进行数据交易的时候，应该尽可能的寻找第一手的数据源。

前瞻性原则：前瞻性是指信贷机构在引进外部数据的时候，要有前瞻性的思维，要以数据驱动的思维去引进数据。

验证性原则：验证性是指信贷机构在引进外部数据的时候都必须经过验证，通过与外部数据源联合测试，验证数据质量、数据的可用性以及数据的价值，从而，为数据引入决策提供依据。

7.4 管理机制

信贷智能风控体系应在数据、模型、应用、信息系统等方面建立健全的管理机制。

数据方面的管理机制涉及外部数据采购和使用、数据质量管理、数据安全管理（包括数据合规、数据使用授权、数据加密、数据防篡改等）、数据存储备份、数据校验加工与整合等。

模型方面的管理机制涉及模型开发管理、模型验证管理、模型调优管理等。

应用方面的管理机制涉及贷前审核、身份认证、反欺诈、准入、授信等策略；贷中评分、风险定价、审批、交易反欺诈等策略；贷后监控、预警、催收等策略等。

信息系统方面的管理机制涉及信息系统开发及维护、信息系统安全、信息系统使用等。

附 录 A
（资料性附录）
智能风控技术

A.1 层次分析法

层次分析法是指20世纪70年代中期由美国数学家、匹兹堡大学杰出教授Thomas.L.Saaty正式提出，它具有如下的特点：

- 定性和定量相结合，系统化、层次化的分析方法；
- 先分解后综合，整理和综合人们的主观判断；
- 分析时需要的定量数据不多；
- 对问题所包含的因素及其关系要具体而明确。

具体说来，是将一个具体问题（如本项目中要建立风险评分）分解为若干定性指标，然后通过问卷调研的方式由业务专家对这些具体指标进行重要性排序，通过对排序表进行量化分析，则可以分别确定每一个评分项目的权重和得分。当不同专家的比较矩阵通过一致性检验时，则获得满意的模型。

1．专家调查

专家评分模型开发的一个重要过程为专家调查，将采用问卷方式对各个专家征询意见，综合全部专家的意见，并整理出综合意见。在问卷调查过程中，采取专家座谈会的方法，调查人员直接向每位专家解释问卷内容，在对指标大类之间以及各大类指标下面的细类指标之间两两进行比较的基础上，通过分析、协商、修改、最终汇总成专家基本一致的看法，作为最终评分模型的专家权重。

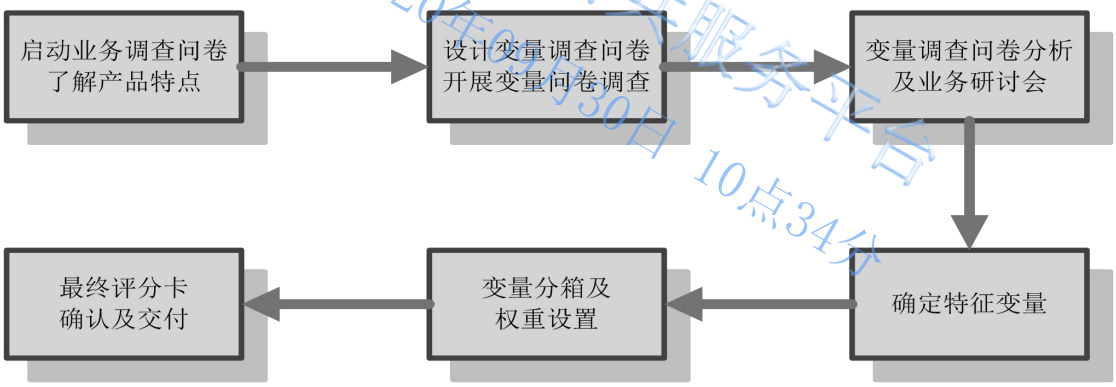
本方法主要包含以下几个步骤：

第一，组成专家小组。

第二，设计调查问卷。根据层次结构模型，设定变量之间两两比较的问卷。

第三，展开问卷调查。与专家小组召开会议，就调查问卷进行详细说明，保证专家能够充分理解，以做出正确的判断。

第四，初步确定变量权重。对专家的意见进行分析处理，综合确定入选变量权重。



2．层次分析步骤

步骤一：建立层次结构模型

层次结构模型是指将有关的各个因素按照不同属性自上而下地分解成若干层次，同一层的诸因素从属于上一层的因素或对上层因素有影响，同时又支配下一层的因素或受到下层因素的作用。最上层为目标层，通常只有1个因素。

步骤二：构造成对比较矩阵/专家问卷

从层次结构模型的第2层开始，对于从属于(或影响)上一层每个因素的同一层诸因素，用成对比较法和1—5比较尺度构造成对比较矩阵。比较第 i 个元素与第 j 个元素相对上一层某个因素的重要性时，使用数量化的相对权重aij来描述。假设共有 n 个元素参与比较，则称为（aij）n×n成对比较矩阵。成对比较矩阵中aij的取值可参考 Satty 的提议，按下述标度进行赋值。aij在 1-5 及其倒数中间取值。

- aij= 1，元素 i 与元素 j具有同等重要性；
- aij= 2，元素 i 比元素 j 稍重要；
- aij= 3，元素 i 比元素 j 明显重要；
- aij= 4，元素 i 比元素 j 强烈重要；
- aij= 5，元素 i 比元素 j 极端重要；
- aij= 1/2，元素 i 比元素 j 稍不重要；
- aij= 1/3，元素 i 比元素 j 明显不重要；
- aij=1/4，元素 i 比元素 j 强烈不重要；
- aij= 1/5，元素 i 比元素 j 的极端不重要；
- aij= 1/aji。

步骤三：计算权向量并做一致性检验

方法：

对于每一个成对比较阵计算最大特征根及对应特征向量，利用一致性指标、随机一致性指标和一致性比率做一致性检验。若检验通过，特征向量(归一化后)即为权向量：若不通过，需重新构造成对比较矩阵。

检验成对比较矩阵A一致性的步骤如下：

- 计算衡量一个成对比较矩阵A（n>1 阶方阵）不一致程度的指标
- $$CI = \frac{\lambda_{max}(A) - n}{n-1}$$
- λmax是矩阵 A 的最大特征值。
- 按下面公式计算成对比较矩阵 A 的随机一致性比率 CR：
- $$CR = \frac{CI}{RI}$$
- RI称为平均随机一致性指标，它只与矩阵阶数有关，可以通过查表获得。

N	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46	1.49
N	11	12	13	14	15	16	17	18	19	20
RI	1.52	1.54	1.56	1.58	1.59	1.59	1.61	1.61	1.62	1.63

表 A.1

- 判断方法如下：当CR<0.1时，判定为成对比较阵A具有满意的一致性，或其不一致程度是可以接受的；否则就调整成对比较矩阵 A，直到达到满意的一致性为止。

具体含义：每个指标大类下面的指标进行两两比较，则会生成判断矩阵，而判断矩阵是否具有满意的一致性是层次分析法的一个重要前提。由于专家在构造判断矩阵的过程中很难保证这一前提的满足，而问卷的往复耗时长、效率低，因此在本次评分卡的开发过程中，采用对专家构造的原始判断矩阵进行人工的修改和调整的方法，来得到一致性较好的判断矩阵。当一个矩阵的CR参数小于0.1时，这个矩阵的一致性被认为是可以接受的。

A.2 聚类分析法

聚类分析本质上是将研究的对象进行分类，以确定个体在群体意见中所占权重的方法。其基本思想是通过定义样本之间的距离，将相近的样本归为一类，直至所有类之间的距离满足某种条件。

聚类分析的算法过程如下：

1．计算n个样本两两之间的距离，形成距离矩阵D，D为对称矩阵，且对角元素为0；

2．首先构造n个类，每一类中只包含一个样本；

3．合并距离最近的两类为新类；

4．计算新类与当前各类的距离，若类的个数已经等于m（m是预先给定的最终分类个数，一般根据样本数量n决定），转(步骤5)，否则转(步骤3)；

5．加权合并样本。

类的距离可以采用以下算法：

最短距离法：

$$D(C_1,C_2)=\min_{x_i\in C_1,x_j\in C_2}\{d(x_i,x_j)\}$$

其中，

$$d(x_i,x_j)=\sqrt{\sum_{k=1}^{m}(x_{ik}-x_{jk})^2}$$

由4-10位专家填写的AHP问卷结果能获得4-10组专家建议的的初始权重。为了科学汇总多个专家的意见，对各个专家建议的打分卡权重进行聚类分析。根据专家的数量，将总体样本最终分成2～3类，类别数取决于专家反馈数量。最终根据每类中样本的个数确定权重，对样本进行加权平均确定最终权重。例如，六份样本{a，b，c，d，e，f}，最终形成三类{A，B，C}，其中a，c属于A类，b，d，e属于B类，f属于C类。那么最终的权重为：

$$w=\frac{2}{14}a+\frac{3}{14}b+\frac{2}{14}c+\frac{3}{14}d+\frac{3}{14}e+\frac{1}{14}f$$

其中2+3+2+3+3+1=14。聚类过程的图示如下：

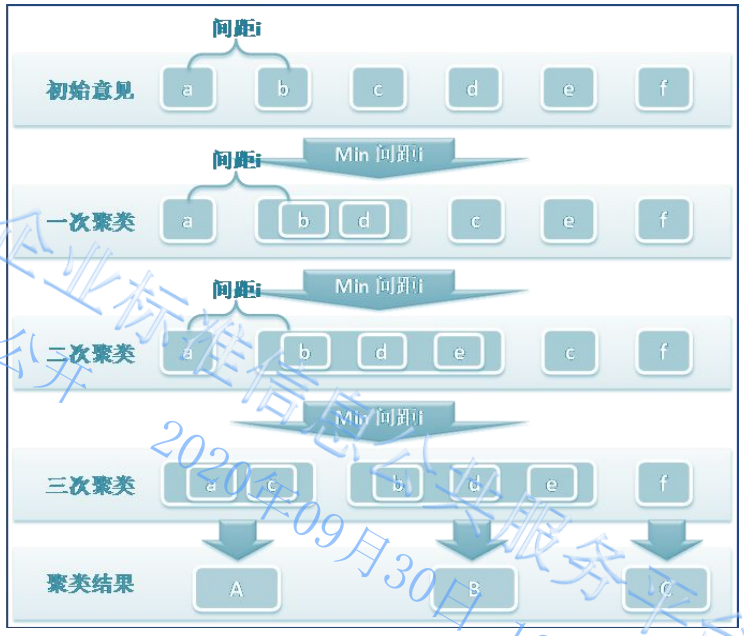


图 A.1

A.3 Logistic模型

如果假设残差服从对数分布，则是Logistic函数：

$$F(\beta^T x_i)=\frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$$

所以事件发生的概率为：

$$\mathrm{P}(y_i=1)=\frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$$

事件不发生的概率为：

$$\mathrm{P}(y_i=0)=\frac{1}{1+e^{\beta^T x_i}}$$

两者相除：

$$\frac{\mathrm{P}(y_i=1)}{\mathrm{P}(y_i=0)}=e^{\beta^T x_i}$$

其中，$\frac{\mathrm{P}(y_i=1)}{\mathrm{P}(y_i=0)}$ 称为事件的发生率。

两侧取对数：

$$\ln\left(\frac{P(y_i=1)}{P(y_i=0)}\right)=\beta^{T}x_i$$

类似的，可以采用极大似然法来估计上式中的参数。

在运用Logistic模型进行多变量建模时，模型需要进一步对目前为止留下的备选变量进行选择，使得这些变量组合起来可以获得一个具有较强区分能力的模型，同时这些变量对应的系数具有统计意义上的显著性且其正负符号符合我们的预期。

常用的变量组合搜索策略包括前向搜索、后向搜索和逐步搜索，简要介绍如下。

前向搜索：事先给定挑选自变量进入模型的统计意义水平。开始方程中没有自变量，然后按自变量对因变量的贡献大小由大到小依此挑选进入模型，直到模型外没有具有统计意义的自变量进入模型为止。

后向搜索：事先给定挑选自变量进入模型的统计意义水平。开始全部变量都在模型之中，然后按自变量对因变量的贡献大小由小到大依此剔除，一旦剔除，则再不能进入模型，直到模型中没有不具有统计意义的自变量可剔除为止。

逐步搜索：逐步搜索是前向搜索的修正，对于已经在模型中的自变量，不必一直在模型中。事先给定挑选自变量进入模型的统计意义水平和从模型剔除自变量的统计意义水平。在引入每一个新变量的同时，对已进入的变量进行逐个检验，将无统计意义的变量剔除。这样保证最后所得的变量子集中所有变量对因变量的影响都是有统计意义的。

A.4　决策树模型

1）基本流程

决策树的构造是一个递归的过程，有三种情形会导致递归返回：(1) 当前结点包含的样本全属于同一类别，这时直接将该节点标记为叶节点，并设为相应的类别；(2) 当前属性集为空，或是所有样本在所有属性上取值相同，无法划分，这时将该节点标记为叶节点，并将其类别设为该节点所含样本最多的类别；(3) 当前结点包含的样本集合为空，不能划分，这时也将该节点标记为叶节点，并将其类别设为该节点中所含样本最多的类别。

2）划分准则

2.1）信息增益

ID3算法使用信息增益为准则来选择划分属性，“信息熵”(information entropy)是度量样本结合纯度的常用指标，假定当前样本集合D中第k类样本所占比例为pk，则样本集合D的信息熵定义为：

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|\mathcal{Y}|}p_k\log_2 p_k\ .$$

值越大表示越混乱，易知只有一个类别时，信息熵为0

假定通过属性划分样本集D，产生了V个分支节点，v表示其中第v个分支节点，易知：分支节点包含的样本数越多，表示该分支节点的影响力越大。故可以计算出划分后相比原始数据集D获得的“信息增益”（information gain）。

$$\mathrm{Gain}(D,a)=\mathrm{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)\ .$$

2.2）基尼指数

CART决策树使用“基尼指数”（Gini index）来选择划分属性，基尼指数反映的是从样本集D中随机抽取两个样本，其类别标记不一致的概率，因此Gini(D)越小越好，基尼指数定义如下：

$$\begin{aligned}\mathrm{Gini}(D) &= \sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k} p_k p_{k'} \\ &= 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 .\end{aligned}$$

任取两个样本类标不一致的概率

越小表示集合越纯

进而，使用属性 α 划分后的基尼指数为：

$$\mathrm{Gini\_index}(D,a) = \sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Gini}(D^v) .$$

故选择基尼指数最小的划分属性

A.5　神经网络模型

人工神经网络（artificial neural network，ANN）是指一系列受生物学和神经学启发的数学模型。这些模型主要是通过对人脑的神经元网络进行抽象，构建人工神经元，并按照一定拓扑结构来建立人工神经元之间的连接，来模拟生物神经网络。在人工智能领域，人工神经网络也常常简称为神经网络（NEURAL NETWORK）或神经模型（neural model）。

从机器学习的角度来看，神经网络一般可以看作是一个非线性模型，其基本组成单位为具有非线性激活函数的神经元，通过大量神经元之间的连接，使得神经网络成为一种高度非线性的模型。神经元之间的连接权重就是需要学习的参数，可以通过梯度下降方法来进行学习。

1)神经元

人工神经元（artificial neuron），简称神经元（neuron），是构成神经网络的基本单元，其主要是模拟生物神经元的结构和特性，接受一组输入信号并产出输出。生物学家在20世纪初就发现了生物神经元的结构。一个生物神经元通常具有多个树突和一条轴突。树突用来接受信息，轴突用来发送信息。当神经元所获得的输入信号的积累超过某个阈值时，它就处于兴奋状态，产生电脉冲。轴突尾端有许多末梢可以给其他多个神经元的树突产生连接（突触），并将电脉冲信号传递给其它神经元。

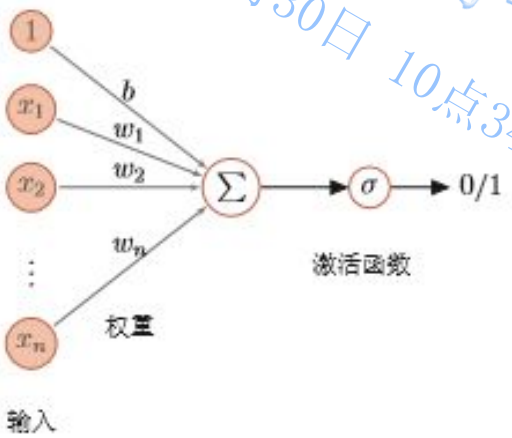


图 A.2

1.1）净输入（线性部分）

假设一个神经元接受n个输入x1,x2,…,xn，我们用向量x=[x1;x2;…;xn]来表示这组输入，并用净输入（net input）z ∈R表示一个神经元所获得的输入信号x的加权和。

其中w=[w1;w2;…;wn]∈Rn 是n维的权重向量，b∈R是偏置。

1.2）激活函数

为了增强网络的表达能力以及学习能力，一般使用连续非线性激活函数（activation function）。因为连续非线性激活函数可导，所以可以用最优化的方法来学习网络参数。下面介绍几个在神经网络中常用的激活函数。

激活函数	函数	导数
Logistic函数	$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$
Tanh函数	$f(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$
ReLU	$f(x)=\max(0,x)$	$f'(x)=I(x>0)$
SoftPlus函数	$f(x)=\ln(1+e^x)$	$f'(x)=\frac{1}{1+e^{-x}}$

表 A.2

Logistic函数是一种sigmoid型函数。其定义为：

$$\sigma(x)=\frac{1}{1+e^{-x}}.$$

Logistic函数可以看成是一个“挤压”函数，把一个实数域的输入“挤压”到(0,1)。当输入值在0附近时，sigmoid型函数近似为线性函数；当输入值靠近两端时，对输入进行抑制。输入越小，越接近于0；输入越大，越接近于1。这样的特点也和生物神经元类似，对一些输入会产生兴奋（输出为1），对另一些输入产生抑制（输出为0）。和感知器使用的阶跃激活函数相比，logistic函数是连续可导的，其数学性质更好。

Tanh函数是也一种sigmoid型函数。其定义为

$$\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$$

Tanh函数可以看作是放大并平移的logistic函数：

$$\tanh(x)=2\sigma(2x)-1.$$

Logistic函数和tanh函数都是sigmoid型函数，具有饱和性，但是计算开销较大。因为这两个函数都是在中间（0附近）近似线性，两端饱和。因此，这两个函数可以通过分段函数来近似。

修正线性单元（rectfied linear unit，ReLU），也叫rectfier函数，是目前深层神经网络中经常使用的激活函数。ReLU实际上是一个斜坡（ramp）函数，定义为

$$\text{rectifier}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

采用ReLU的神经网络只需要进行加、乘和比较的操作，计算上也更加高效。此外，rectifier函数被认为有生物上的解释性。神经科学家发现在部分生物神经元除了具有单侧抑制的特性外，其兴奋程度也可以非常高，即有一个宽兴奋边界。此外，生物神经元只对少数输入信号选择性响应，处于兴奋状态的神经元非常稀疏，大脑中在同一时刻大概只有1%~4%的神经元处于活跃状态。Sigmoid系激活函数会导致一个非稀疏的神经网络，这不符合神经科学的发现。而ReLU却具有很好的稀疏性，大约50%的神经元会处于激活状态。 Rectifier函数为左饱和函数，在x>0时导数为1，在x≤0时导数为0。这样在训练时，如果学习率设置过大，在一次更新参数后，一个采用ReLU的神经元在所有的训练数据上都不能被激活。那么，这个神经元在以后的训练过程中永远不能被激活，这个神经元的梯度就永远都会是0。

Softplus函数可以看作是rectifier函数的平滑版本，其定义为：

$$\textbf{softplus}(x) = \log(1 + e^x)$$

softplus函数其导数刚好是logistic函数。softplus虽然也有具有单侧抑制、宽兴奋边界的特性，却没有稀疏激活性。

2）前馈神经网络结构

给定一组神经元，我们可以以神经元为节点来构建一个网络。不同的神经网络模型有着不同网络连接的拓扑结构。一种比较直接的拓扑结构是前馈网络。前馈神经网络（Feedforward Neural Network，FNN）是最早被发明的简单人工神经网络。

在前馈神经网络中，各神经元分别属于不同的层。每一层的神经元可以接收前一层神经元的信号，并产生信号输出到下一层。第一层叫输入层，最后一层叫输出层，其它中间层叫做隐藏层。整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。前馈神经网络也经常称为多层感知器（multi-layer perceptron，MLP）。但多层感知器的叫法并不是十分合理，因为前馈神经网络其实是由多层的logistic回归模型（连续的非线性函数）组成，而不是由多层的感知器（不连续的非线性函数）组成。下图给出了前馈神经网络的示例。

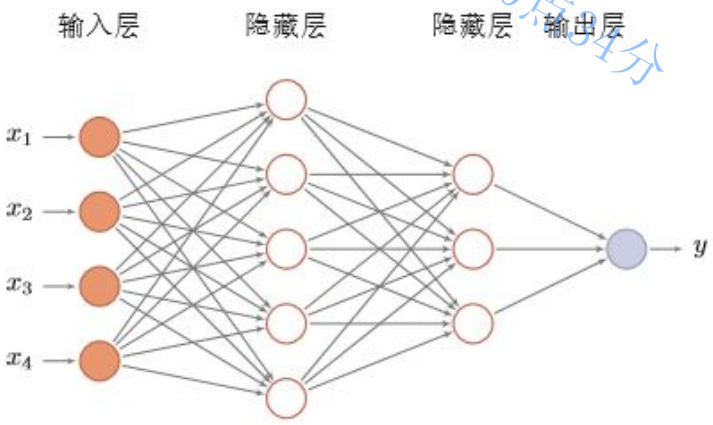


图 A.3

现在用下面的记号来描述一个前馈神经网络：

- L：表示神经网络的层数；
- n^l：表示第l层神经元的个数；
- $f_l(\cdot)$：表示l层神经元的激活函数；
- $W^{(l)} \in \mathbb{R}^{n^l \times n^{l-1}}$：表示$l-1$层到第$l$层的权重矩阵；
- $\mathbf{b}^{(l)} \in \mathbb{R}^{n^l}$：表示$l-1$层到第$l$层的偏置；
- $\mathbf{z}^{(l)} \in \mathbb{R}^{n^l}$：表示$l$层神经元的净输入（净活性值）；
- $\mathbf{a}^{(l)} \in \mathbb{R}^{n^l}$：表示$l$层神经元的输出（活性值）。

前馈神经网络通过下面公式进行信息传播，

$$\mathbf{z}^{(l)} = W^{(l)} \cdot \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$
$$\mathbf{a}^{(l)} = f_l(\mathbf{z}^{(l)})$$

上面两个公式合并写为：

$$\mathbf{z}^{(l)} = W^{(l)} \cdot f_{l-1}(\mathbf{z}^{(l-1)}) + \mathbf{b}^{(l)}$$

这样，前馈神经网络可以通过逐层的信息传递，得到网络最后的输出 a (L)。整个网络可以看作一个复合函数ϕ(x;W,b)，将输入x作为第1层的输入 a (0)，将第L层的输出 a (L) 作为整个函数的输出。

$$\mathbf{x} = \mathbf{a}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{a}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \cdots \rightarrow \mathbf{a}^{(L-1)} \rightarrow \mathbf{z}^{(L)} \rightarrow \mathbf{a}^{(L)} = \varphi(\mathbf{x}; W, \mathbf{b})),$$

其中W,b表示网络中所有层的连接权重和偏置。

3）反向传播算法

假设采用随机梯度下降进行神经网络参数学习，给定一个样本(x,y)，将其输入到神经网络模型中，得到网络输出为ŷ。假设损失函数为L(y, ŷ)，要进行参数学习就需要计算损失函数关于每个参数的导数。

不失一般性，对第1层中的参数W(1)和b(1)计算偏导数。因为 ∂L(y, ŷ) /∂W(1)的计算涉及矩阵微分，十分繁琐，因此我们先计算偏导数 ∂L(y, ŷ) /∂W (1)。根据链式法则，

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial W_{ij}^{(l)}} = \left(\frac{\partial \mathbf{z}^{(l)}}{\partial W_{ij}^{(l)}}\right)^{\mathrm{T}} \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}},$$
$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{(l)}} = \left(\frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{b}^{(l)}}\right)^{\mathrm{T}} \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}}.$$

上述两个公式的第二项是都为目标函数关于第1层的神经元z(1)的偏导数，称为误差项，因此可以共用。我们只需要计算三个偏导数，分别为∂z(1)/ ∂W (1)，∂z(1) /∂b(1)和∂L(y, ŷ)/ ∂z(1)。下面分别来计算这三个偏导数。

（1）计算偏导数 ∂z(1) /∂W (1)

因为z(1)和W(1)的函数关系为z(1) = W(1)a(1−1)+b(1)，因此偏导数

$$\frac{\partial \mathbf{z}^{(l)}}{\partial W_{ij}^{(l)}} = \frac{\partial (W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})}{\partial W_{ij}^{(l)}}$$

$$= \begin{bmatrix} \frac{\partial (W_{1:}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})}{\partial W_{ij}^{(l)}} \\ \vdots \\ \frac{\partial (W_{i:}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})}{\partial W_{ij}^{(l)}} \\ \vdots \\ \frac{\partial (W_{n^l:}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})}{\partial W_{ij}^{(l)}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ a_j^{(l-1)} \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{第} i \text{行} \quad (4.45)$$

（2）计算偏导数∂z(l)/ ∂b(l)

因为z(l)和b(l)的函数关系为z(l) = W(l)a(l−1) +b(l)，因此偏导数

$$\frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{b}^{(l)}} = \mathbf{I}_{n^l},$$

（3）计算误差项 ∂L(y, ŷ) /∂z(l) 我们用 δ (l)来定义第l层神经元的误差项，

$$\delta^{(l)} = \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}} \in \mathbb{R}^{n^l}.$$

误差项 δ (l)来表示第l层的神经元对最终误差的影响，也反映了最终的输出对第 l层的神经元对最终误差的敏感程度。根据z(l+1) = W(l+1)a(l) +b(l+1)，有

$$\frac{\partial \mathbf{z}^{(l+1)}}{\partial \mathbf{a}^{(l)}} = (W^{(l+1)})^T$$

根据a(l) = f(z(l))，其中f(•)为按位计算的函数，因此有

$$\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} = \frac{\partial f_l(\mathbf{z}^{(l)})}{\partial \mathbf{z}^{(l)}}$$
$$= \mathbf{diag}(f_l'(\mathbf{z}^{(l)})).$$

因此，根据链式法则，第l层的误差项为

$$\delta^{(l)} \triangleq \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}}$$

$$
\begin{aligned}
&= \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \cdot \frac{\partial \mathbf{z}^{(l+1)}}{\partial \mathbf{a}^{(l)}} \cdot \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l+1)}} \\
&= \mathbf{diag}(f_l'(\mathbf{z}^{(l)})) \cdot (W^{(l+1)})^{\mathrm{T}} \cdot \delta^{(l+1)} \\
&= f_l'(\mathbf{z}^{(l)}) \odot ((W^{(l+1)})^{\mathrm{T}} \delta^{(l+1)}),
\end{aligned}
$$

其中⊙是向量的点积运算符，表示每个元素相乘。从上式可以看出，第l层的误差项可以通过第l +1层的误差项计算得到，这就是误差的反向传播。反向传播算法的含义是：第l层的一个神经元的误差项（或敏感性）是所有与该神经元相连的第l+1层的神经元的误差项的权重和。然后，再乘上该神经元激活函数的梯度。

在计算出上面三个偏导数之后，可得

$$
\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial W_{ij}^{(l)}} = \mathbf{I}_j (a^{(l-1)})^{\mathrm{T}} \delta^{(l)} = \delta_i^{(l)} a_j^{(l-1)}.
$$

进一步，L(y, ŷ)关于第l层权重W(l)的梯度为

$$
\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial W^{(l)}} = \delta^{(l)} (\mathbf{a}^{(l-1)})^{\mathrm{T}}.
$$

同理可得，L(y, ŷ)关于第l层偏置b(l)的梯度为

$$
\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{(l)}} = \delta^{(l)}.
$$

在计算出每一层的误差项之后，我们就可以得到每一层参数的梯度。因此，基于误差反向传播算法（backpropagation，BP）的前馈神经网络训练过程可以分为以下三步：

1. 前馈计算每一层的净输入z(l)和激活值a(l)，直到最后一层；
2. 反向传播计算每一层的误差项δ(l)；
3. 计算每一层参数的偏导数，并更新参数。

A.6 支持向量机

支持向量机是一种经典的二分类模型，基本模型定义为特征空间中最大间隔的线性分类器，其学习的优化目标便是间隔最大化，因此支持向量机本身可以转化为一个凸二次规划求解的问题。

1)函数间隔与几何间隔

对于二分类学习，假设现在的数据是线性可分的，这时分类学习最基本的思想就是找到一个合适的超平面，该超平面能够将不同类别的样本分开，类似二维平面使用ax+by+c=0来表示，超平面实际上表示的就是高维的平面，如下图所示：

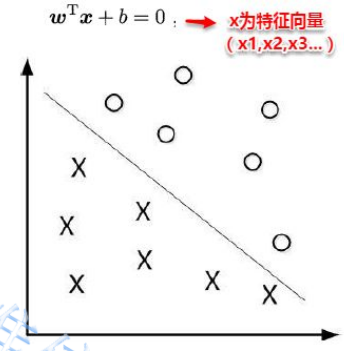


图 A.4

对数据点进行划分时，易知：当超平面距离与它最近的数据点的间隔越大，分类的稳健性越好，即当新的数据点加入时，超平面对这些点的适应性最强，出错的可能性最小。因此需要让所选择的超平面能够最大化这个间隔Gap（如下图所示）。常用的间隔定义有两种，一种为函数间隔，另一种为几何间隔，下面将分别介绍这两种间隔，并对SVM为什么会选用几何间隔做一些阐述。

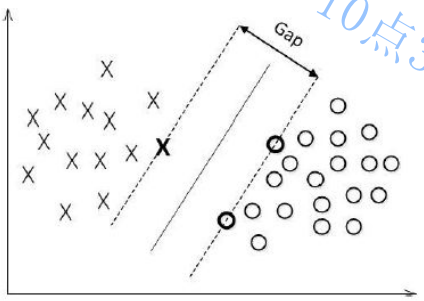


图 A.5

2)函数间隔

在超平面w’x+b=0确定的情况下，|w’x*+b|能够代表点x距离超平面的远近，易知：当w’x+b>0时，表示x在超平面的一侧（正类，类标为1），而当w’x+b<0时，则表示x在超平面的另外一侧（负类，类别为-1），因此（w’x+b）y* 的正负性恰能表示数据点x*是否被分类正确。于是便引出了函数间隔的定义（functional margin）：

$$\hat{\gamma} = y(w^T x + b) = yf(x)$$

而超平面（w,b）关于所有样本点（Xi，Yi）的函数间隔最小值则为超平面在训练数据集T上的函数间隔：

$$\hat{\gamma} = \min \hat{\gamma}_i, \quad (i = 1, ..., n)$$

可以看出：这样定义的函数间隔在处理SVM上会有问题，当超平面的两个参数w和b同比例改变时，函数间隔也会跟着改变，但是实际上超平面还是原来的超平面，并没有变化。例如：w1x1+w2x2+w3x3+b=0 其实等价于2w1x1+2w2x2+2w3x3+2b=0，但计算的函数间隔却翻了一倍。从而引出了能真正度量点到超平面距离的概念--几何间隔（geometrical margin）。

3)几何间隔

几何间隔代表的则是数据点到超平面的真实距离，对于超平面w’x+b=0，w代表的是该超平面的法向量，设x为超平面外一点x在法向量w方向上的投影点，x与超平面的距离为r，则有x=x-r(w/||w||)，又x在超平面上，即w’x+b=0，代入即可得：

$$\gamma = \frac{w^T x + b}{\|w\|} = \frac{f(x)}{\|w\|}$$

为了得到r的绝对值，令r呈上其对应的类别y，即可得到几何间隔的定义：

$$\tilde{\gamma} = y\gamma = \frac{\hat{\gamma}}{\|w\|}$$

从上述函数间隔与几何间隔的定义可以看出：实质上函数间隔就是|w’x+b|，而几何间隔就是点到超平面的距离。

4)最大间隔与支持向量

通过前面的分析可知：函数间隔不适合用来最大化间隔，因此这里我们要找的最大间隔指的是几何间隔，于是最大间隔分类器的目标函数定义为：

$$\max \tilde{\gamma}$$

$$y_i(w^T x_i + b) = \hat{\gamma}_i \geq \hat{\gamma}, \quad i = 1, ..., n$$

一般地，我们令γ^为1（这样做的目的是为了方便推导和目标函数的优化），从而上述目标函数转化为：

$$\max \frac{1}{\|w\|}, \quad s.t. \quad y_i(w^T x_i + b) \geq 1, i = 1, ..., n$$

对于y(w’x+b)=1的数据点，即下图中位于w’x+b=1或w’x+b=-1上的数据点，我们称之为支持向量（support vector），易知：对于所有的支持向量，它们恰好满足y*(w’x*+b)=1，而所有不是支持向量的点，有y*(w’x*+b)>1。

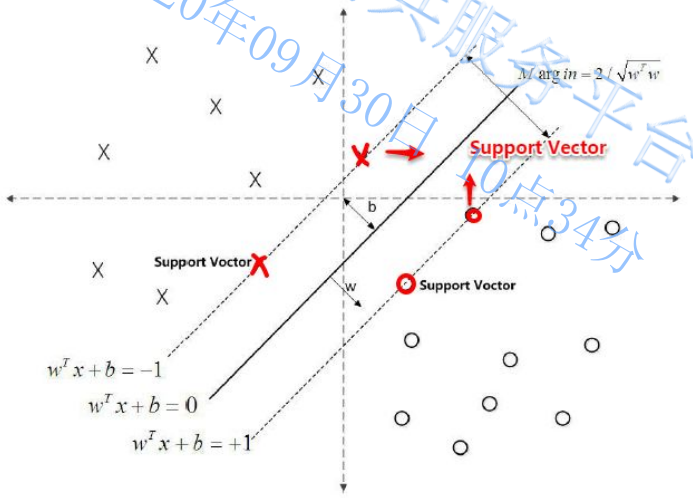


图 A.6

5)从原始优化问题到对偶问题

对于上述得到的目标函数，求1/||w||的最大值相当于求||w||^2的最小值，因此很容易将原来的目标函数转化为：

$$\min \frac{1}{2}\|w\|^2, \quad s.t. \quad y_i(w^T x_i + b) \geq 1, i = 1, ..., n$$

即变为了一个带约束的凸二次规划问题，可以使用现成的优化计算包（QP优化包）求解，但由于SVM的特殊性，一般我们将原问题变换为它的对偶问题，接着再对其对偶问题进行求解。为什么通过对偶问题进行求解，有下面两个原因：

* 一是因为使用对偶问题更容易求解；
* 二是因为通过对偶问题求解出现了向量内积的形式，从而能更加自然地引出核函数。

对偶问题可以理解成优化等价的问题，更一般地，是将一个原始目标函数的最小化转化为它的对偶函数最大化的问题。对于当前的优化问题，首先我们写出它的朗格朗日函数：

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

上式很容易验证：当其中有一个约束条件不满足时，L的最大值为∞（只需令其对应的α为 ∞即可）；当所有约束条件都满足时，L的最大值为1/2||w||^2（此时令所有的α为0），因此实际上原问题等价于：

$$\min_{w,b} \theta(w) = \min_{w,b} \max_{\alpha_i \geq 0} \mathcal{L}(w, b, \alpha) = p^*$$

由于这个的求解问题不好做，所以我们一般交换最小和最大的位置（需满足KKT条件），变成原问题的对偶问题：

$$\max_{\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w, b, \alpha) = d^*$$

这样原问题的求最小变成了对偶问题求最大（用对偶这个词还是很形象），接下来便可以先求L对w和b的极小，再求L对α的极大。

（1）首先求L对w和b的极小，分别求L关于w和b的偏导，可以得出：

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

将上述结果代入L得到：

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$ → 现在只包含α

（2）接着L关于α极大求解α（通过SMO算法求解）。

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

$$s.t. \quad \alpha_i \geq 0, i = 1, ..., n$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

（3）最后便可以根据求解出的α，计算出w和b，从而得到分类超平面函数。

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

$$b^* = -\frac{\max_{i:y_i=-1} w^{*T}x_i + \min_{i:y_i=1} w^{*T}x_i}{2}$$

在对新的点进行预测时，实际上就是将数据点x*代入分类函数f(x)=w'x+b中，若f(x)>0，则为正类，f(x)<0，则为负类，根据前面推导得出的w与b，分类函数如下所示，此时便出现了上面所提到的内积形式。

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

$$b^* = -\frac{\max_{i:y_i=-1} w^{*T}x_i + \min_{i:y_i=1} w^{*T}x_i}{2}$$

这里实际上只需计算新样本与支持向量的内积，因为对于非支持向量的数据点，其对应的拉格朗日乘子一定为0，根据最优化理论（K-T条件），对于不等式约束y(w'x+b)-1≥0，满足：

$$\partial_i(y_i(\omega^T x_i + b) - 1) = 0$$ → 即总有一个为0

6)核函数

由于上述的超平面只能解决线性可分的问题，对于线性不可分的问题（如异或问题），我们需要使用核函数将其进行推广。一般地，解决线性不可分问题时，常常采用映射的方式，将低维原始空间映射到高维特征空间，使得数据集在高维空间中变得线性可分，从而再使用线性学习器分类。如果原始空间为有限维，即属性数有限，那么总是存在一个高维特征空间使得样本线性可分。若∅代表一个映射，则在特征空间中的划分函数变为：

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b$$

按照同样的方法，先写出新目标函数的拉格朗日函数，接着写出其对偶问题，求L关于w和b的极小，最后运用SOM求解α。

核函数可以直接计算隐式映射到高维特征空间后的向量内积，而不需要显式地写出映射后的结果，它虽然完成了将特征从低维到高维的转换，但最终却是在低维空间中完成向量内积计算，与高维特征空间中的计算等效，从而避免了在高维空间无法直接计算的问题。引入核函数后，原来的对偶问题与分类函数则变为：

（1）对偶问题：

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$
$$s.t. \quad \alpha_i \geq 0, i = 1, ..., n$$
$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

（2）分类函数：

$$f(x) = \sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b$$

因此，在线性不可分问题中，核函数的选择成了支持向量机的最大变数，若选择了不合适的核函数，则意味着将样本映射到了一个不合适的特征空间，则极可能导致性能不佳。

由于核函数的构造十分困难，通常我们都是从一些常用的核函数中选择，下面列出了几种常用的核函数：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$	即处理线性可分的情形,这样使得它们在形式统一起来
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^d$	$d \geqslant 1$ 为多项式的次数
高斯核RBF	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j + \theta)$	tanh 为双曲正切函数, $\beta > 0, \theta < 0$

表 A.3

7)软间隔支持向量机

前面的讨论中，我们主要解决了两个问题：当数据线性可分时，直接使用最大间隔的超平面划分；当数据线性不可分时，则通过核函数将数据映射到高维特征空间，使之线性可分。然而在现实问题中，对于某些情形还是很难处理。例如数据中有噪声的情形，噪声数据（outlier）本身就偏离了正常位置，但是在前面的SVM模型中，我们要求所有的样本数据都必须满足约束。如果舍弃这些噪声数据还好，如下图所示，当加入这些outlier后，划分超平面被挤歪了，对支持向量机的泛化性能造成很大的影响。

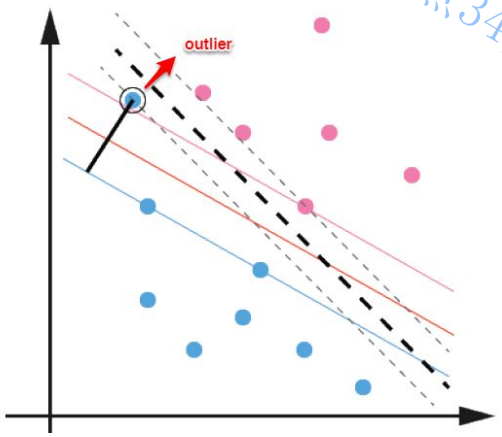


图 A.7

为了解决这一问题，我们需要允许某一些数据点不满足约束，即可以在一定程度上偏移超平面，同时使得不满足约束的数据点尽可能少，这便引出了“软间隔”支持向量机的概念

* 允许某些数据点不满足约束y(w’x+b)≥1；
* 同时又使得不满足约束的样本尽可能少。

这样优化目标变为：

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) - 1\right)$$

损失函数

$$\ell_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0; \\ 0, & \text{otherwise.} \end{cases}$$

如同阶跃函数，0/1损失函数虽然表示效果最好，但是数学性质不佳。因此常用其它函数作为“替代损失函数”。

hinge 损失: $\ell_{hinge}(z) = \max(0, 1-z)$;

指数损失(exponential loss): $\ell_{exp}(z) = \exp(-z)$;

对率损失(logistic loss): $\ell_{log}(z) = \log(1+\exp(-z))$.

支持向量机中的损失函数为hinge损失，引入“松弛变量”，目标函数与约束条件可以写为：

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

称为正则项或惩罚项

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, ..., n$$

$$\xi_i \geq 0, i = 1, ..., n$$

松弛变量

其中C为一个参数，控制着目标函数与新引入正则项之间的权重，这样显然每个样本数据都有一个对应的松弛变量，用以表示该样本不满足约束的程度，将新的目标函数转化为拉格朗日函数得到：

$$\mathcal{L}(w, b, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} r_i\xi_i$$

按照与之前相同的方法，先让L求关于w，b以及松弛变量的极小，再使用SMO求出α：

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - r_i = 0, \quad i = 1, ..., n$$

将w代入L化简，便得到其对偶问题：

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$ 与原对偶问题相同

$$s.t. \quad 0 \le \alpha_i \le C, i = 1, ..., n$$ 多出一个上限

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

将“软间隔”下产生的对偶问题与原对偶问题对比可以发现：新的对偶问题只是使得约束条件中的α多了一个上限C，其它部分完全相同，因此在引入核函数处理线性不可分问题时，能使用与“硬间隔”支持向量机完全相同的方法。

A.7 贝叶斯分类器

贝叶斯分类器是一种概率框架下的统计学习分类器，对分类任务而言，假设在相关概率都已知的情况下，贝叶斯分类器考虑如何基于这些概率为样本判定最优的类标。

定理 设试验 E 的样本空间为 S. A 为 E 的事件，$B_1, B_2, \cdots, B_n$ 为 S 的一个划分，且 $P(A)>0, P(B_i)>0\ (i=1,2,\cdots,n)$，则

$$P(B_i \mid A) = \frac{P(A \mid B_i)P(B_i)}{\sum_{j=1}^{n}P(A \mid B_j)P(B_j)}, \quad i = 1,2,\cdots,n. \qquad (5.7)$$

P(ABi) / P(A)

(5.7) 式称为贝叶斯(Bayes)公式①. 实际上由条件概率与全概率公式推导出

1) 贝叶斯决策论

若将上述定义中样本空间的划分Bi看做为类标，A看做为一个新的样本，则很容易将条件概率理解为样本A是类别Bi的概率。在机器学习训练模型的过程中，往往我们都试图去优化一个风险函数，因此在概率框架下我们也可以为贝叶斯定义“条件风险”（conditional risk）。

$$R(c_i \mid \boldsymbol{x}) = \sum_{j=1}^{N}\lambda_{ij}P(c_j \mid \boldsymbol{x}) .$$ 损失

我们的任务就是寻找一个判定准则来最小化所有样本的条件风险总和，因此就有了贝叶斯判定准则（Bayes decision rule）：为最小化总体风险，只需在每个样本上选择那个使得条件风险最小的类标。

$$h^*(\boldsymbol{x}) = \arg\min_{c\in\mathcal{Y}} R(c \mid \boldsymbol{x}) ,$$

若损失函数λ取0-1损失，则有：

$$R(c \mid \boldsymbol{x}) = 1 - P(c \mid \boldsymbol{x}) ,$$

$$h^*(\boldsymbol{x}) = \arg\max_{c\in\mathcal{Y}} P(c \mid \boldsymbol{x}) ,$$

即对于每个样本x，选择其后验概率P（c|x）最大所对应的类标，能使得总体风险函数最小，从而将原问题转化为估计后验概率P（c|x）。一般这里有两种策略来对后验概率进行估计：

* 判别式模型：直接对 P（c|x）进行建模求解。例如决策树、神经网络、SVM都是属于判别式模型。
* 生成式模型：通过先对联合分布P（x,c）建模，从而进一步求解 P（c|x）。

贝叶斯分类器就属于生成式模型，基于贝叶斯公式对后验概率P（c|x）进行一项变换：

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\, P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})},$$

类先验概率 类条件概率或称似然 证据因子

对于给定的样本x，P（x）与类标无关，P（c）称为类先验概率，p（x|c）称为类条件概率。这时估计后验概率P（c|x）就变成为估计类先验概率和类条件概率的问题。这里介绍先验概率和后验概率的基本概念。

* 先验概率：根据以往经验和分析得到的概率。

* 后验概率：后验概率是基于新的信息，修正原来的先验概率后所获得的更接近实际情况的概率估计。

实际上先验概率就是在还没有得到任何结果的情况下估计的概率，而后验概率则是在有一定依据后的重新估计，直观意义上后验概率就是条件概率。

对于类先验概率P（c），P（c）就是样本空间中各类样本所占的比例，根据大数定理（当样本足够多时，频率趋于稳定等于其概率），当训练样本充足时，p(c)可以使用各类出现的频率来代替。因此只剩下类条件概率p（x|c），它表示在类别c中出现x的概率，涉及到属性的联合概率问题。若只有一个离散属性还好，当属性多时采用频率估计就十分困难，因此这里一般采用极大似然法进行估计。

2）极大似然法

极大似然估计（Maximum Likelihood Estimation，简称MLE），是一种根据数据采样来估计概率分布的经典方法。常用的策略是先假定总体具有某种确定的概率分布，再基于训练样本对概率分布的参数进行估计。运用到类条件概率p（x|c）中，假设p（x|c）服从一个参数为 θ 的分布，问题就变为根据已知的训练样本来估计 θ 。极大似然法的核心思想就是：估计出的参数使得已知样本出现的概率最大，即使得训练数据的似然最大。

令 D_c 表示训练集 D 中第 c 类样本组成的集合，假设这些样本是独立同分布的，则参数 $\boldsymbol{\theta}_c$ 对于数据集 D_c 的似然是

统计学的基础假设

$$P(D_c \mid \boldsymbol{\theta}_c) = \prod_{\boldsymbol{x} \in D_c} P(\boldsymbol{x} \mid \boldsymbol{\theta}_c). \tag{7.9}$$

连乘

连乘操作便得求解变得十分复杂，一般我们使用对数似然（log-likelihood）：

$$\begin{aligned} LL(\boldsymbol{\theta}_c) &= \log P(D_c \mid \boldsymbol{\theta}_c) \\ &= \sum_{\boldsymbol{x} \in D_c} \log P(\boldsymbol{x} \mid \boldsymbol{\theta}_c), \end{aligned}$$

对数似然变乘为加

此时参数 $\boldsymbol{\theta}_c$ 的极大似然估计 $\hat{\boldsymbol{\theta}}_c$ 为

一般采用对每个参数求偏导等于0的方法求解

$$\hat{\boldsymbol{\theta}}_c = \arg\max_{\boldsymbol{\theta}_c} LL(\boldsymbol{\theta}_c).$$

所以，贝叶斯分类器的训练过程就是参数估计过程。最大似然法估计参数的过程，一般总结为以下四个步骤：

* 1. 写出似然函数；

* 2. 对似然函数取对数，并整理；

* 3. 求导数，令偏导数为0，得到似然方程组；

* 4. 解似然方程组，得到所有参数即为所求。

例如：假设样本属性都是连续值，p（x|c）服从一个多维高斯分布，则通过MLE计算出的参数恰好分别为：

$$p(\boldsymbol{x} \mid c) \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$$

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{|D_c|} \sum_{\boldsymbol{x} \in D_c} \boldsymbol{x} \,,$$

样本均值向量

$$\hat{\boldsymbol{\sigma}}_c^2 = \frac{1}{|D_c|} \sum_{\boldsymbol{x} \in D_c} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c)(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}} \,.$$

样本协方差矩阵

上述结果看起来十分合乎实际，但是采用最大似然法估计参数的效果很大程度上依赖于作出的假设是否合理、是否符合潜在的真实数据分布。

3）朴素贝叶斯分类器

原始的贝叶斯分类器最大的问题在于联合概率密度函数的估计，首先需要根据经验假设联合概率分布，其次当属性很多时，训练样本往往覆盖不够，参数的估计会出现很大的偏差。为了避免这个问题，朴素贝叶斯分类器（naive Bayes classifier）采用了“属性条件独立性假设”，即样本数据的所有属性之间相互独立。这样类条件概率p（x|c）可以改写为：

$$P(\boldsymbol{x} \mid c) = \cdot \prod_{i=1}^{d} P(x_i \mid c)$$

这样，为每个样本估计类条件概率变成为每个样本的每个属性估计类条件概率。

- 对于**离散属性**，属性的类条件概率可估计为：

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|} \,.$$

- 对于**连续属性**，若假设属性服从正态分布，则属性的类条件概率可估计为：

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right) \,.$$

相比原始贝叶斯分类器，朴素贝叶斯分类器基于单个的属性计算类条件概率，操作更加容易。需要注意的是，若某个属性值在训练集中和某个类别没有一起出现过，这样会抹掉其它的属性信息，因为该样本的类条件概率被计算为0。因此在估计概率值时，常常用进行平滑（smoothing）处理，拉普拉斯修正（Laplacian correction）就是其中的一种经典方法，具体计算方法如下：

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N} \,,$$

出现的类别数

$$\hat{P}(x_i \mid c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i} \,.$$

属性xi可能的取值数

当训练集越大时，拉普拉斯修正引入的影响越来越小。对于贝叶斯分类器，模型的训练就是参数估计，因此可以事先将所有的概率储存好，当有新样本需要判定时，直接查表计算即可。

A.8　集成学习

1）Bagging

Bagging是一种并行式的集成学习方法，即基学习器的训练之间没有前后顺序且可以同时进行，Bagging使用“有放回”采样的方式选取训练集，对于包含m个样本的训练集，进行m次有放回的随机采样操作，从而得到m个样本的采样集。按照相同的方式重复进行，我们就可以采集到T个包含m个样本的数据集，从而训练出T个基学习器，最终对这T个基学习器的输出进行结合。

Bagging算法的流程如下所示：

输入： 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程：

1: **for** $t=1,2,\ldots,T$ **do**
2: $\quad h_t=\mathfrak{L}(D,\mathcal{D}_{bs})$
3: **end for**

输出： $H(\boldsymbol{x})=\arg\max_{y\in\mathcal{Y}}\sum_{t=1}^{T}\mathbb{I}(h_t(\boldsymbol{x})=y)$

可以看出Bagging主要通过样本的扰动来增加基学习器之间的多样性，因此Bagging的基学习器应为那些对训练集十分敏感的不稳定学习算法，例如神经网络与决策树等。从偏差-方差分解来看，Bagging算法主要关注于降低方差，即通过多次重复训练提高稳定性。不同于AdaBoost的是，Bagging可以十分简单地推广应用到多分类、回归等问题。总的说起来则是：AdaBoost关注于降低偏差，而Bagging关注于降低方差。

2）Boosting

Boosting是一种串行的工作机制，即个体学习器的训练存在依赖关系，必须一步一步序列化进行。其基本思想是：增加前一个基学习器在训练过程中预测错误样本的权重，使得后续基学习器更加关注这些被打标的错误训练样本并尽可能纠正这些错误，一直向下串行直至产生所需要的T个基学习器，Boosting最终对这T个学习器进行加权结合，产生学习器委员会。

Boosting族算法最著名、使用最为广泛的就是AdaBoost，因此下面主要对AdaBoost算法进行介绍。AdaBoost使用的是指数损失函数，因此AdaBoost的权值与样本分布的更新都是围绕着最小化指数损失函数进行的。

定义基学习器的集成为加权结合，则有：

AdaBoost算法的指数损失函数定义为：

$$loss_{\exp}(h)=\mathbb{E}_{x\sim D,y}[e^{-yh(x)}]$$

打标正确为负
打标错误为正

具体说来，整个Adaboost 迭代算法分为3步：

初始化训练数据的权值分布。如果有N个样本，则每一个训练样本最开始时都被赋予相同的权值，即1/N。

训练弱分类器。具体训练过程中，如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权值就被降低；相反，如果某个样本点没有被准确地分类，那么它的权值就得到提高。然后，权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。

将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起较大的决定作用；同时降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起较小的决定作用。

整个AdaBoost的算法流程如下所示：

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm L;
Number of learning rounds T.

Process:

1. $\mathcal{D}_1(i) = 1/m$. % Initialize the weight distribution
2. **for** $t = 1, \cdots, T$:
3. $h_t = L(D, \mathcal{D}_t)$; % Train a learner h_t from D using distribution $\mathcal{D}_t$
4. $\epsilon_t = \Pr_{x \sim \mathcal{D}_t, y}[h_t(\boldsymbol{x}) \neq y]$; % Measure the error of h_t
5. **if** $\epsilon_t > 0.5$ **then break**
6. $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of h_t （1）计算基学习器权重
7. $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$ （2）样本权重分布更新
$= \frac{\mathcal{D}_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$ % Update the distribution, where
% Z_t is a normalization factor which
% enables $\mathcal{D}_{t+1}$ to be distribution
8. **end**

Output: $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$

A.9 降维与度量学习

样本的特征数称为维数（dimensionality），维数非常大时即为现在所说的“维数灾难”。具体表现为：在高维情形下，数据样本将变得十分稀疏，因为此时要满足训练样本为“密采样”的总体样本数目是一个触不可及的天文数字，训练样本的稀疏使得其代表总体分布的能力大大减弱，从而消减了学习器的泛化能力；同时当维数很高时，计算距离也变得十分复杂，甚至连内积计算都不再容易。

缓解维数灾难的一个重要途径就是降维，即通过某种数学变换将原始高维空间转变到一个低维的子空间。在这个子空间中，样本的密度将大幅提高，同时距离计算也变得容易。这时也许会有疑问，这样降维之后不是会丢失原始数据的一部分信息吗？这是因为在很多实际的问题中，虽然训练数据是高维的，但与学习任务相关的也许仅仅是其中的一个低维子空间，也称为一个低维嵌入，例如数据属性中存在噪声属性、相似属性或冗余属性等，对高维数据进行降维能在一定程度上达到提炼低维优质属性或降噪的效果。

1）MDS算法

不管是使用核函数升维还是对数据降维，我们都希望原始空间样本点之间的距离在新空间中基本保持不变，这样才不会使得原始空间样本之间的关系及总体分布发生较大的改变。“多维缩放”（MDS）正是基于这样的思想，MDS要求原始空间样本之间的距离在降维后的低维空间中得以保持。

假定m个样本在原始空间中任意两两样本之间的距离矩阵为D∈R(m*m)，我们的目标便是获得样本在低维空间中的表示Z∈R(d’*m,d’<d)，且任意两个样本在低维空间中的欧式距离等于原始空间中的距离，即||zi-zj||=Dist(ij)。因此接下来我们要做的就是根据已有的距离矩阵D来求解出降维后的坐标矩阵Z。

令 $\mathbf{B} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{m \times m}$，其中 $\mathbf{B}$ 为降维后样本的内积矩阵，$b_{ij} = \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j$

高维距离 = 低维欧氏距离

$$dist_{ij}^2 = \|\boldsymbol{z}_i\|^2 + \|\boldsymbol{z}_j\|^2 - 2\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j \quad (1)$$
$$= b_{ii} + b_{jj} - 2b_{ij} .$$

令降维后的样本坐标矩阵Z被中心化，中心化是指将每个样本向量减去整个样本集的均值向量，故所有样本向量求和得到一个零向量。这样易知：矩阵B的每一列以及每一列求和均为0，因为提取公因子后均有一项为所有样本向量的和向量。

$$B=\begin{bmatrix} z_1 \\ \dots \\ z_m \end{bmatrix} * \begin{bmatrix} z_1 & \dots & z_m \end{bmatrix} = \begin{bmatrix} z_1z_1 & z_1z_2 & \dots & z_1z_m \\ z_2z_1 & z_2z_2 & \dots & z_2z_m \\ \dots & \dots & \dots & \dots \\ z_mz_1 & z_mz_2 & \dots & z_mz_m \end{bmatrix}$$

和为零向量

和为零向量

根据上面矩阵B的特征，我们很容易得到等式（2）、（3）以及（4）：

$$\sum_{i=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + mb_{jj} \quad (2) \quad *1/m$$

$$\sum_{j=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + mb_{ii} \quad (3) \quad *1/m$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = 2m\,\mathrm{tr}(\mathbf{B}) \quad (4) \quad *1/(m\hat{}2)$$

这时根据(1)--(4)式我们便可以计算出bij，即bij=(1)-(2)(1/m)-(3)(1/m)+(4)*(1/(mˆ2))，再逐一地计算每个b(ij)，就得到了降维后低维空间中的内积矩阵B(B=Z'*Z)，只需对B进行特征值分解便可以得到Z。MDS的算法流程如下图所示：

输入： 距离矩阵 $\mathbf{D} \in \mathbb{R}^{m\times m}$，其元素 $dist_{ij}$ 为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离；
低维空间维数 d'.

过程：

1: 根据式(10.7)~(10.9)计算 $dist_{i\cdot}^2$, $dist_{\cdot j}^2$, $dist_{\cdot\cdot}^2$;

2: 根据式(10.10)计算矩阵 $\mathbf{B}$; 低维内积矩阵

3: 对矩阵 $\mathbf{B}$ 做特征值分解; 特征值分解求解

4: 取 $\mathbf{\Lambda}$ 为 d' 个最大特征值所构成的对角矩阵，$\tilde{\mathbf{V}}$ 为相应的特征向量矩阵.

输出： 矩阵 $\tilde{\mathbf{V}}\tilde{\mathbf{\Lambda}}^{1/2} \in \mathbb{R}^{m\times d'}$，每行是一个样本的低维坐标 并没有得到投影向量

2） 主成分分析（PCA）

不同于MDS采用距离保持的方法，主成分分析（PCA）直接通过一个线性变换，将原始空间中的样本投影到新的低维空间中。简单来理解这一过程便是：PCA采用一组新的基来表示样本点，其中每一个基向量都是原来基向量的线性组合，通过使用尽可能少的新基向量来表出样本，从而达到降维的目的。

假设使用d'个新基向量来表示原来样本，实质上是将样本投影到一个由d'个基向量确定的一个超平面上（即舍弃了一些维度）。要用一个超平面对空间中所有高维样本进行恰当的表达，最理想的情形是这些样本点都能在超平面上表出且这些表出在超平面上都能够很好地分散开来。但是一般使用较原空间低一些维度的超平面来做到这两点十分不容易，这要求超平面具有如下两个性质：

最近重构性：样本点到超平面的距离足够近，即尽可能在超平面附近；

最大可分性：样本点在超平面上的投影尽可能地分散开来，即投影后的坐标具有区分性。

最近重构性与最大可分性虽然从不同的出发点来定义优化问题中的目标函数，但这两种特性最终得到了完全相同的优化问题：

$$\min \sum_{i=1}^{m}\left\|\sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j-\boldsymbol{x}_i\right\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i+\text{const}$$

$$\min -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\mathbf{W}\right)$$

基于最大重构性

$$\max_{\mathbf{W}} \operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right)$$
$$\text{s.t. } \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I},$$

基于最大可分性

等价

接着使用拉格朗日乘子法求解上面的优化问题，得到：

即X中心化后的协方差矩阵

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}=\lambda\mathbf{W}$$

因此只需对协方差矩阵进行特征值分解即可求解出W，PCA算法的整个流程如下图所示：

输入： 样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$；
低维空间维数 d'.

过程：

减去均值向量

1: 对所有样本进行中心化：$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i-\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解；

特征值分解再次登场

4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_{d'}$.

输出： 投影矩阵 $\mathbf{W}=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_{d'})$.

得出了投影矩阵，对于新样本只需乘上投影矩阵即可

3）核化线性降维

正如SVM在处理非线性可分时通过引入核函数将样本投影到高维特征空间，接着在高维空间再对样本点使用超平面划分，此处也引入了核函数，先将样本映射到高维空间，再在高维空间中使用线性降维的方法。下面主要介绍核化主成分分析（KPCA）的思想。

若核函数的形式已知，即我们知道如何将低维的坐标变换为高维坐标，这时我们只需先将数据映射到高维特征空间，再在高维空间中运用PCA即可。但是一般情况下，我们并不知道核函数具体的映射规则，例如：Sigmoid、高斯核等，我们只知道如何计算高维空间中的样本内积，这时就引出了KPCA的一个重要创新之处：空间中的任一向量都可以由该空间中的所有样本线性表示。证明过程十分简单：

$$\left(\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right)\mathbf{W}=\lambda\mathbf{W}$$

zi为样本点在高维特征空间中的坐标向量
W为高维特征空间中的一个新基向量

$$\mathbf{W}=\frac{1}{\lambda}\left(\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right)\mathbf{W}=\sum_{i=1}^{m}\boldsymbol{z}_i\frac{\boldsymbol{z}_i^{\mathrm{T}}\mathbf{W}}{\lambda}$$

计算出来是一个常数

$$=\sum_{i=1}^{m}\boldsymbol{z}_i\alpha_i,$$ 证毕

这样我们便可以将高维特征空间中的投影向量wi用所有高维样本点线性表出，接着代入PCA的求解问题，得到：

$$\Phi(X)\Phi(X)^T w_i = \lambda_i w_i$$

空间中的任一向量，都可以由该空间中的所有样本线性表示

$$w_i = \sum_{k=1}^{N} \alpha_i \Phi(x_i) = \Phi(X)\alpha$$

$$\Phi(X)\Phi(X)^T \Phi(X)\alpha = \lambda_i \Phi(X)\alpha$$

核函数矩阵

$$\Phi(X)^T \Phi(X)\Phi(X)^T \Phi(X)\alpha = \lambda_i \Phi(X)^T \Phi(X)\alpha$$

$$K^2\alpha = \lambda_i K\alpha$$

$$K\alpha = \lambda_i \alpha$$

特征值分解无处不在~

化简到最后一步，只需对核矩阵K进行特征分解，得出投影向量wi对应的系数向量α，因此选取最大的d’个特征值所对应的特征向量，所得便是d’个系数向量。这时对于需要降维的样本点，只需按照以下步骤便可以求出其降维后的坐标。可以看出：KPCA在计算降维后的坐标表示时，需要与所有样本点计算核函数值并求和，因此该算法的计算开销十分大。

$$\hat{x}_{new} \square$$

$$= w_i^T x_{new}$$

新样本点在wi维度上的投影坐标
共有d’个投影向量w

$$= (\sum_{i=1}^{N} \Phi(x_i)\alpha_i)^T \Phi(x_{new})$$

$$= (\Phi(X)\alpha)^T \Phi(x_{new})$$

$$= \alpha^T \Phi(X)^T \Phi(x_{new})$$

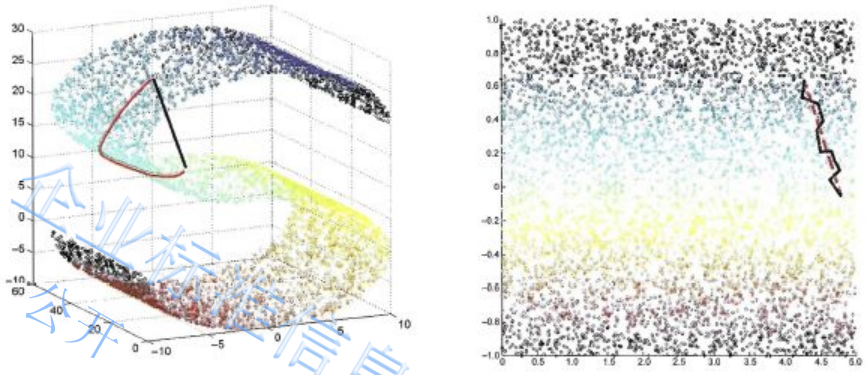
$$= [\alpha_1, \ldots, \alpha_N][k(x_1, x_{new}), \ldots, k(x_N, x_{new})]^T$$

4）流形学习

流形学习（manifold learning）是一种借助拓扑流形概念的降维方法，流形是指在局部与欧式空间同胚的空间（即在局部与欧式空间具有相同的性质）能用欧氏距离计算样本之间的距离。这样即使高维空间的分布十分复杂，但是在局部上依然满足欧式空间的性质，基于流形学习的降维正是遵循这种“邻域保持”的思想。其中等度量映射（Isomap）试图在降维前后保持邻域内样本之间的距离，而局部线性嵌入（LLE）则是保持邻域内样本之间的线性关系，下面将分别对这两种著名的流行学习方法进行介绍。

4. 1）等度量映射（Isomap）

等度量映射的基本出发点是：高维空间中的直线距离具有误导性，因为有时高维空间中的直线距离在低维空间中是不可达的。因此利用流形在局部上与欧式空间同胚的性质，可以使用近邻距离来逼近测地线距离，即对于一个样本点，基于欧式距离找出其近邻点，这样整个样本空间就形成了一张近邻图，高维空间中两个样本之间的距离就转为最短路径问题。可采用著名的Dijkstra算法或Floyd算法计算最短距离，得到高维空间中任意两点之间的距离后便可以使用MDS算法来其计算低维空间中的坐标。



(a) 测地线距离与高维直线距离　　(b) 测地线距离与近邻距离

图 A.8

从MDS算法的描述中我们可以知道：MDS先求出了低维空间的内积矩阵B，接着利用特征值分解计算出样本在低维空间中的坐标，但并没有给出通用的投影向量w，以致于对需要降维的新样本无从下手，由此可利用已知高/低维坐标的样本作为训练集学习出一个“投影器”，继而通过高维坐标预测出低维坐标。Isomap算法流程如下图：

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

整个样本集形成一张可达图

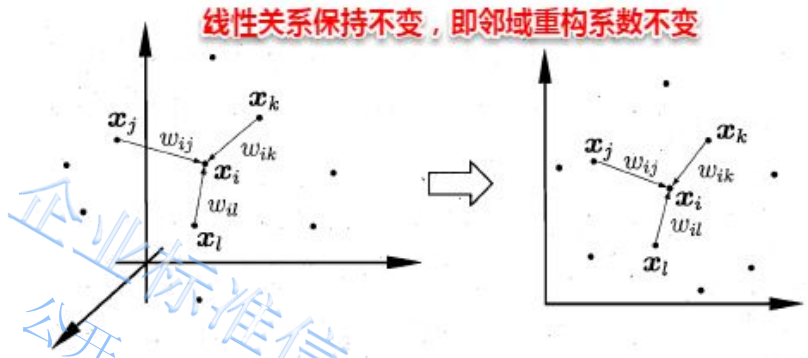
过程：
1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离，与其他点的距离设置为无穷大
4: **end for**
5: 调用最短路径算法计算任意两样本点之间的距离 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
6: 将 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 作为 MDS 算法的输入;
7: **return** MDS 算法的输出
输出： 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

对于近邻图的构建，常用的有两种方法：一种是指定近邻点个数，像kNN一样选取k个最近的“邻居”；另一种是指定邻域半径，距离小于该阈值的被认为是它的近邻点。但两种方法均会出现下面的问题：若邻域范围指定过大，则会造成“短路问题”，即本身距离很远却成了近邻。若邻域范围指定过小，则会造成“断路问题”，即有些样本点无法可达了。

4.2）局部线性嵌入(LLE)

不同于Isomap算法中需保持邻域距离，LLE算法试图去保持邻域内的线性关系，假定样本xi的坐标可以通过它的邻域样本线性表出：

$$\boldsymbol{x}_i = w_{ij}\boldsymbol{x}_j + w_{ik}\boldsymbol{x}_k + w_{il}\boldsymbol{x}_l$$



LLE算法分为两步走，首先第一步根据近邻关系计算出所有样本的邻域重构系数w：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m}\left\|\boldsymbol{x}_i-\sum_{j\in Q_i} w_{ij}\boldsymbol{x}_j\right\|_2^2$$
$$\text{s.t.} \sum_{j\in Q_i} w_{ij}=1,$$

其中 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 均为已知，令 $C_{jk}=(\boldsymbol{x}_i-\boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{x}_k)$，$w_{ij}$ 有闭式解

$$w_{ij}=\frac{\sum\limits_{k\in Q_i} C_{jk}^{-1}}{\sum\limits_{l,s\in Q_i} C_{ls}^{-1}}.$$

接着根据邻域重构系数不变，去求解低维坐标：

$$\min_{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m} \sum_{i=1}^{m}\left\|\boldsymbol{z}_i-\sum_{j\in Q_i} w_{ij}\boldsymbol{z}_j\right\|_2^2$$

令 $\mathbf{Z}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m)\in\mathbb{R}^{d'\times m}$，$(\mathbf{W})_{ij}=w_{ij}$

$$\mathbf{M}=(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W})$$

这样利用矩阵M，优化问题可以重写为：

$$\min_{\mathbf{Z}} \operatorname{tr}(\mathbf{Z}\mathbf{M}\mathbf{Z}^{\mathrm{T}})$$
$$\text{s.t. } \mathbf{Z}\mathbf{Z}^{\mathrm{T}}=\mathbf{I}.$$

特征值分解又来了~

M特征值分解后最小的d’个特征值对应的特征向量组成Z，LLE算法的具体流程如下图所示：

输入: 样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.
过程:
1: **for** $i=1,2,\ldots,m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: 从式(10.27)求得 $w_{ij}, j\in Q_i$;
4: 对于 $j\notin Q_i$, 令 $w_{ij}=0$;
5: **end for**
6: 从式(10.30)得到 M;
7: 对 M 进行特征值分解;
8: **return** M 的最小 d' 个特征值对应的特征向量
输出: 样本集 D 在低维空间的投影 $Z=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m\}$.

局部(邻域)线性关系保持不变

和Isomap一样只得到了低维坐标

5）度量学习

考虑到维数灾难（即在高维空间进行机器学习任务遇到样本稀疏、距离难计算等诸多的问题），上述所述的降维方法都试图将原空间投影到一个合适的低维空间中，接着在低维空间进行学习任务从而产生较好的性能。事实上，不管高维空间还是低维空间都潜在对应着一个距离度量，那是否可以直接学习出一个距离度量以完成等效降维呢？

要学习出距离度量首先须定义一个合适的距离度量形式。对两个样本xi与xj，它们之间的平方欧式距离为：

$$\mathrm{dist}_{\mathrm{ed}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2=dist_{ij,1}^2+dist_{ij,2}^2+\ldots+dist_{ij,d}^2$$

若各个属性重要程度不一样（即各自具有权重），则有加权的平方欧式距离为：

$$\begin{aligned}\mathrm{dist}_{\mathrm{wed}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)&=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2=w_1\cdot dist_{ij,1}^2+w_2\cdot dist_{ij,2}^2+\ldots+w_d\cdot dist_{ij,d}^2\\&=(\boldsymbol{x}_i-\boldsymbol{x}_j)^{\mathrm{T}}\mathbf{W}(\boldsymbol{x}_i-\boldsymbol{x}_j)\end{aligned}$$，其中 $\mathbf{W}=\mathrm{diag}(\boldsymbol{w})$ 是一个对角矩阵, $(\mathbf{W})_{ii}=w_i$.

此时各个属性之间都是相互独立无关的，但现实中往往会存在属性之间有关联的情形（例如身高和体重，一般来讲，身高越高的人体重也会重一些），它们之间存在较大的相关性。这样计算距离时就不能分属性单独计算，由此便引入了经典的马氏距离(Mahalanobis distance)：

$$\mathrm{dist}_{\mathrm{mah}}^2(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i-\boldsymbol{x}_j)^{\mathrm{T}}\mathbf{M}(\boldsymbol{x}_i-\boldsymbol{x}_j)=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_{\mathbf{M}}^2$$ 通用马氏距离

标准的马氏距离中M是协方差矩阵的逆，马氏距离是一种考虑属性之间相关性且尺度无关（即无须去量纲）的距离度量。

$$d(\vec{x},\vec{y})=\sqrt{(\vec{x}-\vec{y})^T\Sigma^{-1}(\vec{x}-\vec{y})}$$ 标准马氏距离

矩阵M也称为“度量矩阵”，为保证距离度量的非负性与对称性，M必须为(半)正定对称矩阵，这样就为度量学习定义好了距离度量的形式，换句话说：度量学习便是对度量矩阵进行学习。

A.10 特征选择与稀疏学习

在机器学习中特征选择是一个重要的“数据预处理”（data preprocessing）过程，即试图从数据集的所有特征中挑选出与当前学习任务相关的特征子集，接着再利用数据子集来训练学习器；稀疏学习则是围绕着稀疏矩阵的优良性质来完成相应的学习任务。

1）子集搜索与评价

一般地，我们可以用很多属性/特征来描述一个示例（例如用性别、身高、体重、年龄、学历、专业等属性来描述一个人）。现在考虑训练出一个学习器来预测人的收入：根据生活经验易知，并不是所有特征都与学习任务相关（如年龄/学历/专业可能在很大程度上影响收入，而身高/体重这些外貌属性影响收入的可能性较小）。因此我们需要的只是那些与学习任务紧密相关的特征，特征选择便是从给定的特征集合中选出相关特征子集的过程。

与降维技术有着异曲同工之处的是，特征选择也可以有效地解决维数灾难的难题。具体而言：降维从一定程度起到了提炼优质低维属性和降噪的效果，特征选择则是直接剔除那些与学习任务无关的属性从而挑选出最佳特征子集。若直接遍历所有特征子集，当维数过多时遭遇指数爆炸显然就行不通了；若采取从候选特征子集中不断迭代生成更优候选子集的方法，则可使得时间复杂度大大减小。这时涉及到两个关键环节：1.如何生成候选子集；2.如何评价候选子集的好坏，这便是早期特征选择的常用方法。

下面介绍基于“贪心算法”的三种策略：

前向搜索：初始将每个特征当做一个候选特征子集，然后从当前所有的候选子集中选择出最佳的特征子集；接着在上一轮选出的特征子集中添加一个新的特征，同样地选出最佳特征子集；直至最后选不出比上一轮更好的特征子集。

后向搜索：初始将所有特征作为一个候选特征子集；接着尝试去掉上一轮特征子集中的一个特征并选出当前最优的特征子集；直到最后选不出比上一轮更好的特征子集。

双向搜索：将前向搜索与后向搜索结合起来，即在每一轮中既有添加操作也有剔除操作。

对于特征子集的评价，假设数据集的属性皆为离散属性，这样给定一个特征子集，便可以通过这个特征子集的取值将数据集合划分为V个子集。例如：A1={男,女}，A2={本科,硕士}就可以将原数据集划分为2*2=4个子集，其中每个子集的取值完全相同。这时我们就可以像决策树选择划分属性那样，通过计算信息增益来评价该属性子集的好坏。

$$\mathrm{Gain}(A)=\mathrm{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$

信息增益越大
越有助于分类

$$\mathrm{Ent}(D)=-\sum_{i=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

此时，信息增益越大表示该属性子集包含着越多有助于分类的特征，使用上述这种子集搜索与子集评价相结合的机制，便可以得到特征选择方法。值得一提的是若将前向搜索策略与信息增益结合在一起，这种操作便与ID3决策树十分地相似。事实上，决策树也可以用于特征选择，树节点划分属性组成的集合便是选择出的特征子集。

2）过滤式选择（Relief）

过滤式方法是一种将特征选择与学习器训练相分离的特征选择技术，即首先将相关特征挑选出来，再使用选择出的数据子集来训练学习器。Relief是其中著名的代表性算法，它使用一个“相关统计量”来度量特征的重要性，该统计量是一个向量，其中每个分量代表着相应特征的重要性，因此我们最终可以根据这个统计量各分量的大小挑选出合适的特征子集。

易知Relief算法的核心在于如何计算相关统计量。对于数据集中的每个样例xi，Relief首先找出与xi同类别的最近邻与不同类别的最近邻，分别称为猜中近邻（near-hit）与猜错近邻（near-miss），接着便可以分别计算出相关统计量中的每个分量。对于j分量：

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2$$

直观上理解：对于猜中近邻，两者j属性的距离越小越好，对于猜错近邻，j属性距离越大越好。更一般地，若xi为离散属性，diff取海明距离，即相同取0，不同取1；若xi为连续属性，则diff为曼哈顿距离，即取差的绝对值。分别计算每个分量，最终取平均便得到了整个相关统计量。

标准的Relief算法只用于二分类问题，后续产生的拓展变体Relief-F则解决了多分类问题。对于j分量，新的计算公式如下：

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \sum_{l \neq k} \left(p_l \times \mathrm{diff}(x_i^j, x_{i,l,\mathrm{nm}}^j)^2 \right)$$

其中pl表示第l类样本在数据集中所占的比例，易知两者的不同之处在于：标准Relief 只有一个猜错近邻，而Relief-F有多个猜错近邻。

3）包裹式选择（LVW）

与过滤式选择不同的是，包裹式选择将后续的学习器也考虑进来作为特征选择的评价准则。因此包裹式选择可以看作是为某种学习器量身定做的特征选择方法，由于在每一轮迭代中，包裹式选择都需要训练学习器，因此在获得较好性能的同时也产生了较大的开销。

4）嵌入式选择与正则化

前面提到了两种特征选择方法，过滤式选择中特征选择与后续学习器完全分离，包裹式选择中使用学习器作为特征选择的评价准则，而嵌入式则是一种将特征选择与学习器训练完全融合的特征选择方法（即将特征选择融入学习器的优化过程中）。经验风险指的是模型与训练数据的契合度，结构风险则是模型的复杂程度，机器学习的核心任务就是：在简化模型的基础上保证模型的契合度。以岭回归为例，作为加上了L2范数的最小二乘法，有效地解决了奇异矩阵、过拟合等诸多问题。下面的嵌入式特征选择则是在损失函数后加上了L1范数。

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1 \ .$$

L1范数（Lasso Regularization）指的是向量中每个元素的绝对值之和，这样在优化目标函数的过程中可使得w尽可能地小，在一定程度上起到了防止过拟合的作用，而与与L2范数（Ridge Regularization）不同的是，L1范数会使得部分w变为0，从而达到了特征选择的效果。

总的来说：L1范数会趋向产生少量的特征，其他特征的权值都是0；L2会选择更多的特征，这些特征的权值都会接近于0。这样L1范数在特征选择上就十分有用，而L2范数则具备较强的控制过拟合能力。可以从下面两个方面来理解：

（1）下降速度：L1范数按照绝对值函数来下降，L2范数按照二次函数来下降。因此在0附近，L1范数的下降速度大于L2范数，L1范数能很快地下降到0，而L2范数在0附近的下降速度非常慢，因此较大可能收敛在0的附近。

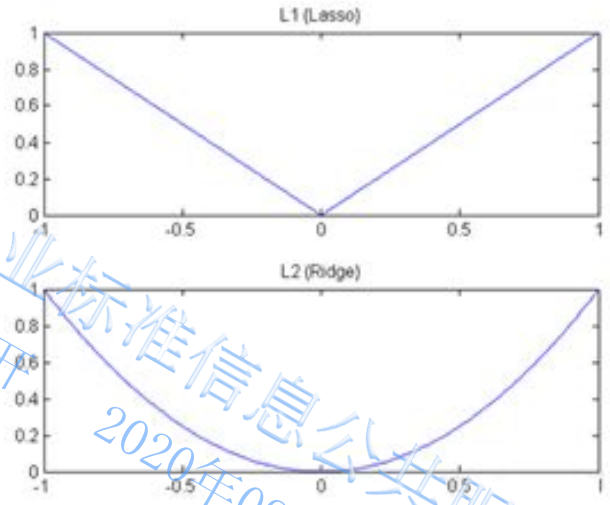


图 A.9

（2）空间限制：L1范数与L2范数都试图在最小化损失函数的同时，让权值W也尽可能地小。我们可以将原优化问题视为下述问题（即让后面的规则则都小于某个阈值）。这样从图中可以看出：L1范数相比L2范数更容易得到稀疏解。

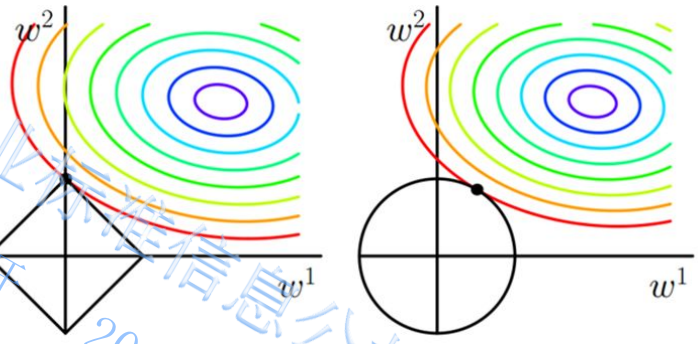


图 A.10

5）稀疏表示与字典学习

当样本数据是一个稀疏矩阵时，对学习任务来说会有不少的好处，例如可使得很多问题变得线性可分，储存更为高效等，这也正是稀疏表示与字典学习的基本出发点。稀疏矩阵表示为矩阵的每一行/列中都包含大量的零元素，且这些零元素没有出现在同一行/列。对于一个给定的稠密矩阵，若我们能通过某种方法找到其合适的稀疏表示，则可以使得学习任务更加简单高效，我们称之为稀疏编码（sparse coding）或字典学习（dictionary learning）。

给定一个数据集，字典学习/稀疏编码指的便是通过一个字典将原数据转化为稀疏表示，因此最终的目标就是求得字典矩阵B及稀疏表示 α 。

$$\min_{\mathbf{B},\boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1$$

xi：d*1
B：d*k字典矩阵
ai：k*1，xi的稀疏表示

使得ai尽可能稀疏

参 考 文 献

[1] GB/T 4943.23-2012 信息技术设备 安全 第23部分：大型数据存储设备

[2] GB/T 15843.6-2018信息技术 安全技术 实体鉴别 第6部分：采用人工数据传递的机制

[3] GB/T 20000.4-2003 标准化工作指南

[4] GB/T 29265.1-2017 信息技术 信息设备资源共享协同服务 第1部分：系统结构与参考模型

[5] GB/T 31186.1-2014 银行客户基本信息描述规范 第1部分：描述模型

[6] GB/T 31916.1-2015 信息技术 云数据存储和管理 第1部分：总则

[7] GB/T 31916.2-2015 信息技术 云数据存储和管理 第2部分：基于对象的云存储应用接口

[8] GB/T 31916.3-2018 信息技术 云数据存储和管理 第3部分：分布式文件存储应用接口

[9] GB/T 31916.5-2015 信息技术 云数据存储和管理 第5部分：基于键值（Key-Value）的云数据管理应用接口

[10] GB/T 34950-2017 非结构化数据管理系统参考模型

[11] GB/T 34960.5-2018 信息技术服务 治理 第5部分：数据治理规范

[12] GB/T 35294-2017 信息技术 科学数据引用

[13] GB/T 36073-2018 数据管理能力成熟度评估模型

[14] GB/T 36344-2018 信息技术 数据质量评价指标

[15] GB/T 36345-2018 信息技术 通用数据导入接口

[16] GB/T 36625.1-2018 智慧城市 数据融合 第1部分：概念模型

[17] GB/T 36625.2-2018 智慧城市 数据融合 第2部分：数据编码规范

[18] IEC 61165:2006 Application of Markov techniques

[19] IEC 61649-2008 Analyse de Weibull

[20] 郭为民.用数据驱动银行智能转型发展[J]. TSINGHUA FINANCIAL REVIEW,2017[08]:25-27.

[21] 韩平、鲁银辉.智能时代下金融业的发展研究[A]. 齐齐哈尔大学学报（哲学社会科学版）.2018[10]:25-28.

[22] 韩志雄,冯学奋,赵权. 智慧金融的产生、发展与前景探析[A].海南金融.2018[08]:19-27.

[23] 罗榕.智能风控为工商银行信用卡业务保驾护航[J].中国信用卡,2017[10]:8-11.

[24] 欧建林. 基于Hadoop 的商业银行大数据平台研究与实现[J]. 金融科技创新,2019:50-53.

[25] 邵康宁.计算机网络信息安全中数据加密技术的研究[A].信息安全与技术,2016:29-32.

[26] 邵理煜. 银行大数据风控能力建设与实践[J]. 中国金融电脑,2018[08]:26-29.

[27] 邢桂伟. 依托大数据技术构建商业银行智能风控体系[J].中国金融电脑,2018[08]:19-22.

[28] 严圣阳. 我国互联网消费金融风控模式及优化路径[J].现代商业,2017:85-86.

[29] 叶志艺. 智能风险管理的银行社会责任风险控制探讨[A].现代经济信息,2017:134.

[30] 亿欧智库.2018年中国智能风控研究报告[R/OL] ,2018.

[31] 张靳. 智能风控：传统金融变革之翼[J].上海信息化.2018:70-72

[32] 中国信息通信研究院云计算与大数据研究所CCSA TC60、大数据技术标准推进委员会. 数据资产管理白皮书 3.0版[R/OL] ,2018.

[33] 周志华.机器学习[M].清华大学出版社.北京,2018.

[34] Bengio, Y.,A. Courville, and P. Vincent. (2013).″Representation learning: A review and new perspecives.″ IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798-1828.

[35] Bishop, C. M. (2006). "Pattern Recognition and Machines Learning." Springer New York, NK.

[36] Frome, A., Y. Singer, and J. Malik. (2007). "Image retrieval and classification using local distance functions." In advances in Neural Information Processing Systems 19 (NIPS) (B. Schölkopf, J. C. Platt, and T. Hoffman, eds.), 417-424, MIT Press, Cambridge, MA.

[37] Halil Ibrahim Erdal&Aykut Ekinci(2012). "A Comparison of Various Artificial Intelligence Methods in the Prediction of Bank Failures." Springer Science+Business Media New York, 199-215.

[38] Lee, Y., Y. Lin, and G. Wahba. (2004). "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data." Journal of the American Statistical Association, 99(465):67-81.

[39] Tibshirani, R., G.Walther, and T. Hastie. (2001). "Estimating the number of clusters in a data set via the gap statistic." Journal of the Royal Statistial Society – Series B, 63(2):411-423.

企业标准信息公共服务平台
公开 2020年09月30日 10点34分