★CACM 中国版★

计算机协会通讯

CACM.ACM.ORG

2014 年 5 月第 57 卷第 5 期

国家安全局和斯诺登：需要多严密的安全措施才能阻止泄密

机器人来了

弥补软件价值缺口

理解 NP 完全问题的实证难度

神经科学与密码学交汇：

Association for Computing Machinery

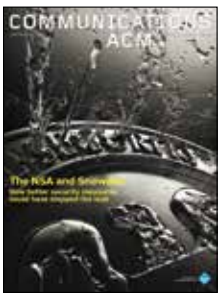
ACM中国理事会办公室
中国北京清华大学
东主楼 11-236 室
邮编：100084
电话：+86-10-62785025
电子邮件：acmchina@acm.org
联系人：辛爽

ACM通讯
(ISSN 0001-0782) 由计算机协会（2 Penn Plaza, Suite 701, New York, NY 10121-0701）按月发行。

Association for Computing Machinery

观点

实践



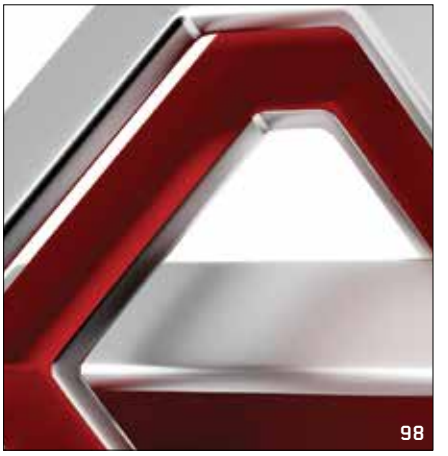
关于封面：

爱德华·斯诺登这位美国国家安全局的分包商因泄露该机构有关直监视活动的绝密资料而名噪全球。相关情况已获大书特书。虽然各媒体仍在争议他这么做的原因，本月的封面故事（第 44 页）描述了他这么做的方法，进而指出美国国安局自身安全实践中刺眼的漏洞。
封面插图照片：Peter Crowther Associates。

拼图：ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES；插图由 MICHELANGELUS、EUGENE KOSTSOV 提供

投稿文章

评论文章



研究亮点

Association for Computing Machinery
Advancing Computing as a Science & Profession

DOI:10.1145/2594457

Ruzena Bajcsy

观点

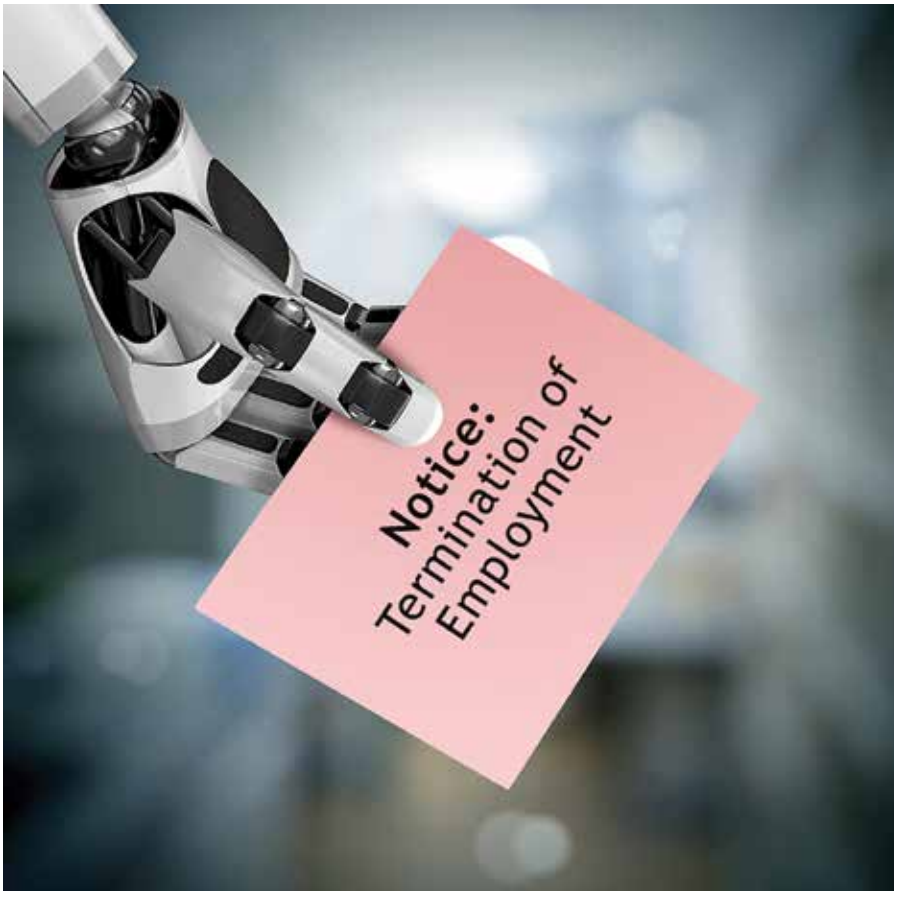
思考机器人技术革命对社会的影响。

最近，关于技术对大众生活的影响，尤其是机器人技术和智能系统是创造了工作机会还是减少了工作机会，学术界与政策制定者一直争论不休。美国奥巴马总统和其他政客表达了对中产阶级“空心化”的担忧，即财富集中在少数人手中，而绝大多数美国公民沦为穷困一族。因此，有责任感的科学家和IT从业人员（特别是机器人技术工作者）应认真研究这些问题，并思考如何才能构建一个公平、公正和民主的社会。机器人不仅可以代替体力劳动，而且还能将体力劳动与脑力劳动相结合，但根本问题是：如何分配机器人带来的利益？

在本期“观点”中，我研究分析了将信息技术和通信技术与机器人技术相结合的第三次革命所带来的社会和经济影响。

首先，让我们了解几个事实：

事实 1：社会进步和技术创新不会停止。实际上，更快更强的计算机、更轻的新型材料、更灵敏的传感器、轻松互联能力以及无线网络等因素使得机器人技术能够以更快的速度向前发展。比尔盖茨在他2007年1月发表的《科学美国人》(*Scientific American*) 文章“让机器人深入千家万户”(A Robot in Every Home) 中写道，机器人技术的发展状态与上世纪70年代计算机产业所处的状态如出一辙。当然，2007年以来，人类取得了重大进步，特别是在汽车和电子行业。预计在2015年，服务行业的发展将非常乐观。



事实 2：机器人和智能系统不仅本身就是一种生产力，而且还有助于生产力的提高，就像在以往的技术革命中，蒸汽机替代了马力，而随之又被电力取代。

事实 3：与以往的工业革命一样，新技术取代旧技术，而使部分工人失业。过去的电气化革命使工人的工资翻番，从而增加了中产阶级的规模，但信息革命并没有达到相同的效果。而实际上，中产阶层已经萎缩！

甚至亨利·福特早在上世纪30年代就明白，为了受益于新技术（装配线），他必须创造一个有能力购

拼图：ALICIA KUBISTA/ANDRIJ BORYS

买他的车的中产阶级。但更重要的是他还认识到，为了保持和平及工人的生产力，他必须通过建设经济适用住房、学校和医院来至少为他的工人和雇员提供社会福利。

所以第三次工业革命、其引导者和社会的社经政治意义是什么？社会如何受益于机器人技术和信息革命所带来的更高生产力？

杰伦·拉尼尔 (Jaron Lanier) 在 2013 年《IEEE 综览》(*IEEE Spectrum*) 杂志对他的专访中建议到，我们都应为我们向 Google、Facebook 等公司提供的信息索要报酬。问题是我们如何将信息变现？汽车等看得见摸得着的实物可以很轻松地用货币来衡量，但个人健康、地理位置等信息却难以用金钱来衡量。但是同理，我们每次访问 Google、Facebook 及类似数据库时都应当为知识买单吗？

再回到一开始提出的问题：如何分配机器人执行的工作所带来的利益？

这是我们这个时代的问题！谁应该得到利益？发明者、制造商、用户、倡导者？各分配多少？例如：房屋清洁机器人代替清洁工人。买了房屋清洁机器人之后，你在省钱的同时，也让之前给你打扫卫生的人丢了工作，从此没有了收入。节省下来的这笔钱应如何分配 / 分享？

2013 年 7 月，马丁·福特在其 *通信* 观点中给经济学家提出了一个挑战性的问题，即构建一个全新的经济学模型来应对今天快速的技术变革，而该模型既不同于严格以市场为基础的经济，同时又不是福利经济。我认为我们需要一个经济模型来支撑中产阶级的增长，从而使生产（人和 / 或机器人）与消费达到平衡。

除了对机器人工作所带来的利益进行财政分配以外，就业还给人们带来了另一个社会效应。那就是：社会告诉他们，他们是社会需要的有用之人。这满足了人类的基本需求。

社会如何受益于机器人技术和信息革命所带来的更高生产力？

Frank Levy 与 Richard Murnane 在他们的文章“与机器人共舞：人类的计算机辅助工作技能”（“Dancing with Robots: Human skills for computerized work”，请参见 http://content.thirdway.org/publications/714/Dancing-With-Robots.pdf）中提到，计算机辅助工作的挑战不是大规模失业，而是培养更多的人从事计算机无法胜任的高级工作。

在此我将话题展开一下，建议我们的教育应该包罗万象，让下一代有能力去拥抱全球化的世界，这个世界不仅包括机器人和技术，还包括不同的文化、语言和历史。

使用互联网来扩大受教育机会这种做法最近取得了可圈可点的发展，但无法解决机器人革命带来的所有问题。除此之外，家庭必须将学习奉为最高目标，从而掀起一场文化革命。 C

Ruzena Bajcsy (bajcsy@eecs.berkeley.edu) 是加利福尼亚大学伯克利分校电气工程和计算机科学专业的教授，同时也是社会发展前瞻科技研发中心（Center for Information Technology Research in the Interest of Society，CITRIS）的荣誉退休主管。

本期“观点”中所表达的观点属于作者，感谢我的同事和学生对本材料先前版本提出的宝贵意见。我还要感谢 Moshe Vardi 给我机会就机器人技术革命的社会经济影响提出重要问题。

译文责任编辑：陈文光

活动日程

6 月 16 日 - 19 日
第 11 届移动系统、应用和服务国际年会
地点：新罕布什尔州布雷顿森林
赞助商：SIGMOBILE
联系人：David Kotz，
电子邮件：kotz@cs.dartmouth.edu

6 月 16 日 - 20 日
ACM SIGMETRICS/ 计算机系统测量和建模国际会议
地点：德克萨斯州奥斯汀市
赞助商：SIGMETRICS，
联系人：Sanjay Shakkottai
电子邮件：shakkott@austin.utexas.edu

6 月 21 日 - 25 日
计算机科学教育创新和技术
地点：瑞典乌普萨拉
赞助商：SIGCSE
联系人：Asa Cajander
电子邮件：asa.cajander@it.uu.se

6 月 22 日 - 27 日
数据管理国际会议
地点：犹他州盐湖城
赞助商：SIGMOD
联系人：Curtis Dyreson
电子邮件：curtis.dyreson@usu.edu

6 月 23 日 - 26 日
ACM Web 科学会议
地点：印第安纳州布卢明顿
赞助商：SIGWEB，
联系人：Filippo Menczer
电子邮件：fil@indiana.edu

6 月 25 日 - 27 日
ACM 电视和在线视频交互体验国际会议
地点：英国泰恩河畔纽卡斯尔
赞助商：SIGCHI，
联系人：Patrick Olivier，
电子邮件：p.l.olivier@ncl.ac.uk

6 月 25 日 - 27 日
第 19 届 ACM 访问控制模型和技术专题讨论会
地点：加拿大安大略省伦敦市
赞助商：SIGSAC，
联系人：Sylvia L. Obsborn
电子邮件：sylvia@csd.uwo.ca

6 月 30 日 - 7 月 2 日
系统和存储国际会议
以色列海法
赞助商：SIGOPS，
联系人：Eliezer Dekel
电子邮件：dekel@il.ibm.com

DOI:10.1145/2594502

文章编写主导者 acmqueue
queue.acm.org

美国国家安全局需要多严密的安全措施才能阻止斯诺登的行为。

作者：BOB TOXEN

国家安全局和斯诺登：保护全视之眼

爱德华·斯诺登曾经供职于美国国家安全局 (NSA) 的项目承包商——博思艾伦咨询公司，在夏威夷工作期间，他将多达 170 万份绝密文档复制到 U 盘上，偷运出他工作的安全机构，并将其中许多文档披露给新闻媒体。[2] 他的这一行为导致美国政府与美国民众及至与其他一些国家的关系发生转变。本文阐述了国家安全局可以在计算机安全方面采取哪些措施防止此类事件发生，这次事件也许是美国历史上破坏最为严重的一次泄密事件。[19] 本文附文则论述了相关的宪法、法律和道德问题。

根据美国第 13526 号总统令，“‘绝密’级别应用于如下类型信息，这些信息一旦被未经授权披露，可以合理预期将对国家安全造成极大损害。”[24] 对于绝密信息有不同的授权级别，例如 SCI（敏感隔离信息）、SAP（特别通行行动）和 CNWDI（重要核武器设计信息）。[9] 在英国，绝密 相当于最高机密。

斯诺登有哪些行为？

斯诺登是一名计算机系统管理员。防范心存恶意的系统管理员，其难度远高于防范普通用户，但也是可以做到的。请注意，美国国家安全局拥有几乎不受限制的预算和资源，因此可以始终遵循良好安全实践。正如白宫计算机安全顾问 Richard Clarke 所说：“如果你在咖啡上的花费都超出 IT 安全资金，你就会遭受黑客攻击，而且也理所当然应该遭受黑客攻击。”[20]

全国公共广播电台去年 12 月 17 日的“All Things Considered”（时事纵观）节目指出，这些失窃文档存储在微软的 SharePoint 文档管理系统上。在可能被复制的 170 万份文档中，斯诺登向记者共享了多达 20 万份文档；国家安全局并未否认这一点。[2,19] 负责评估斯诺登事件损害的国家安全局工作组负责人 Rick Ledgett 表示：“系统管理员掌握了密码，因而能够绕开这些安全措施，斯诺登正是利用这一点乘虚而入。”[19]

身为国家安全局官员的 Rick Ledgett 承认没有认识到防止系统管理员盗窃数据的技术在过去 30 年内取得的发展，这一点令人困惑。[10,15,29] 在本文后面的“桔皮书和双人授权”部分中，我们将对相关内容进行讨论。国家安全局不再使用 SharePoint 进行文档存储，这就引出了一个问题：为什么他们处理机密数据的计算机系统没有遵守桔皮书规范，并且舍弃了其他良好安全实践呢？

在哥伦比亚广播公司 2013 年 12 月 15 日的“60 分钟”节目访谈中，国家安全局局长 Keith B. Alexander 承认，斯诺登的一部分工作是在国家安全局计算机系统之间传输大量机密数据。[19] 因此斯诺登就

插图由 PETER CROWTHER ASSOCIATES 提供


Snowden

将文件复制到一个 U 盘上并私自藏匿，将大量数据从国家安全局盗出。[11,26] 事实上，如果采用运输安全管理局 (TSA) 和其他法院大楼的做法，使用手持式金属探测器在出口处进行简单的一分钟扫描，即可发现任何闪存设备。

安全环

现在我们暂时转开话题，讨论安全环 这个重要概念，我用这个词代表含义不太明确的行业标准术语纵深安全。它的含义是我们具有多个同心的安全环，当攻击者通过最外面的第一个环时，随后很可能遇到第二个、第三个或第四个环的阻拦；没有任何安全措施是 100% 有效的。这些安全环大多采用身份验证方法，但在用户经过身份验证之后，并不限制用户可以进行哪些操作。思考一下安全环如何应用于普通网络；在需要很高安全性的环境下，例如在国家安全局和银行以及处理大量社保或信用卡号码的系统中，这种“普通”安全级别不足以满足需求。

假定我们希望有这样一个网络：系统管理员能够从家里通过 SSH（安全外壳协议）方式访问服务器。在第一个安全环中，防火墙允许从系统管理员家庭网络的少数几个 IP 地址进行 SSH 访问。因此，攻击者无法从互联网上的数十亿个系统中的任意一个系统发起攻击，而只能从数十个系统管理员家庭网络上的某一个系统发起攻击，从而显著减少了安全漏洞。SSH 使用的现代 TCP/IP 协议实现可以很好地抵御 IP 地址欺骗。结合使用端对端加密，几乎可以完全消除中间人攻击。

第二个安全环可以只允许通过这些家庭 Linux 或 Unix 系统上的公钥 / 私钥进行 SSH 身份验证。禁止 SSH 接受密码，可以防止密钥猜测风险，从而防止从未经授权的系统进行访问。第三个安全环将会监控攻击日志文件，并自动阻止这些 IP 地址。第四个安全环可以是该 SSH 私钥上的强密码。第五个安全环可以要求系统管理员的家庭系统（当然也包括办公室的所有系统）在无操作几分钟之后锁定屏幕。

国家安全局原本可以采取众多安全方法阻止斯诺登的行为。其中有些方法已经应用了十年以上，但国家安全局却没有采用。

阻止斯诺登

国家安全局原本可以采取众多安全方法阻止斯诺登的行为。其中有些方法已经应用了十年以上，但国家安全局却没有采用。

安全岛。显而易见，在这个案例中，首先应该防止系统管理员或其他人访问未经授权的系统。安全岛概念可以在有人成功渗透网络的情况下起到防御作用。在具有高安全性的组织中，不同的部门（甚至不同的系统）应视为浩瀚系统海洋中的一个个安全岛，它们彼此互不信任，也不信任网络。这意味着不同的系统应该具有不同的根密码、不同的用户密码和不同的 SSH 密码，系统之间的几乎所有通信都应该加密。系统应该具有加密的文件系统和加密的备份。

物理安全性。每个安全岛都应进行物理安全防御。保护范围当然包括系统和外围设备，以及传输任何未加密机密数据的网络。即使是大型商业共置设施也应受到保护，这些系统的周围应该安装钢笼，并使用视频摄像头监视这些区域。支付卡行业 (PCI) 安全标准要求为大型信用卡处理器提供此类保护。高安全性的机构应该安装视频摄像头，而且长时间保存记录。

一种简单的安全防御方式是在每个钢笼上安装两把高安全性锁，每把锁必须由一个掌握的不同钥匙的人打开。这样，在操作设备时，两个人必须同时到场。同样，网络线缆也可以采取安全保护措施（例如放置在钢质管道内部），或者在 LAN 或 WAN 上发送数据之前对数据进行加密。虽然物理安全性对保护至关重要，但没有任何迹象表明斯诺登利用了这方面的漏洞。

防止未经授权的复制。应该禁用插入 U 盘或插入空白 DVD 进行写操作的功能。此外应该移除大多数 DVD 刻录机和 USB 接口。应该禁止使用相机、记录器、移动电话和其他任何未经授权的存储设备，并对此类设备加以防范。门口应该使用金属探测器来检测窃取者。应该对射频 (RF) 发射进行监控，并且使用法拉第笼（注：静态屏蔽罩）来屏蔽射频发射。所有这些技术的成本都非常低。

双因素身份验证即使斯诺登拥有绝密级授权，也不足以允许他访问所窃取的某些文档。国家安全局承认，斯诺登使用了某些安全局高层官员的用户帐户，这些账户拥有比绝密级更高的授权。他之所以可以做到这一点，是因为这些帐户是由他创建的，或许他利用系统管理员权限修改了帐户，以使用国家安全局的机密内联网 NSAnet 来远程访问更加机密的文档。[13] 斯诺登能够访问具有更高安全授权的帐户，这一点违反了我们长期接受的安全策略，即系统应该禁止任何用户利用高于自身的授权来访问数据。计算机原本可以非常简单地阻止这种行为，要求具有更高级别授权的系统管理员提供服务，以便根据需要调节这些帐户。

这种行为也违反了双因素身份验证的概念。身份验证是计算机（或保安甚至商店柜员）用于判定你的身份是否属实的一项功能。身份验证方法丰富多样，通常包括利用你*知道*的信息（密码或 PIN）、你*拥有*的物件（信用卡、颁发给员工和顾问的 RFID 工牌、USB 软件狗）、你的*身份证据*（签名、指纹或视网膜扫描，或者难于伪造的文档上的照片，包括员工许可证、员工工牌或护照）。它们都称为一个*因素*。所有这些方法的成本都非常低，而且都是行之有效的。虽然指纹可以通过某些手段伪造，但有了市场上出售的现代化高质量指纹阅读器，这种伪造的难度越来越大。

很多组织使用非常流行的双因素身份验证来授予对计算机、设施或资金的访问权限，例如，如果用户不能同时提供密码或 RFID 工牌以及指纹，则将无法获取访问权限。三因素身份验证的效果更好。

如果国家安全局要求进行合理的双因素身份验证，例如将指纹和密码与斯诺登没有管理权限的中央数据库进行比较，就可以防止斯诺登假冒他人身份使用帐户 — 他正是利用这种手段获取了高于自身安全授权的文档。为数据库收集这些因素的工作应该由两组不同人员完成，而且这两组人员都不应该是和斯诺登一样负责管理机密文档的人员。这种权限分离对于确保良好安全性至关重要，因为它需要几个人共同行动才能产生威胁。

即使管理用户密码的人员有所企图，她也无法访问指纹数据库。用户通过单人通行入口，进入一间独立的内室，而密码管理人员不能进入该房间，因此也无法看到用户输入密码。房间里有一个虚拟键盘，用户可在物理强化的触摸屏上进行输入，很难通过按键记录器来窃取密码。由于篇幅所限，我们不再深入讨论各种攻击方式，例如欺骗性指纹、防御按键记录器、TEMPEST（国家安全局自身的一系列针对射频信息泄漏的安全标准）、社会工程学等等。

社会工程学是指攻击者诱骗他人泄漏应该保密的信息。例如，电子邮件谎称自己是银行工作人员，要求你单击某个链接并提供密码，或者提出把偷来的钱和你分赃。斯诺登利用社会工程学手段，获取了国家安全局多名工作人员的密码，这些人员后来也相续辞职；其他论文和书籍对相关内容进行了更详细的介绍。反复进行良好安全教育，制定严格的政策，禁止在任何情况下共享密码、工牌或软件狗，可以防止斯诺登的一部分违规行为。

桔皮书和双人身份验证。如果处于他人监控之下，人们从事不轨行为的可能性就会降低。正因为如此，很多商店要求至少有两名员工一起工作，运钞车也使用至少两名员工。也正是因为这个原因，你可以看到有些支票底部注有“超过 5000 美元的金额必须有两人签名”字样。

美国国家安全局在 30 年前制定了桔皮书规范，其正式名称为“可信计算机系统评估标准”。联邦政府和承包商在使用计算机处理不同安全保密级别的数据时必须使用该规范。本文作者曾经对一个符合桔皮书规范的 Unix 系统进行增强，以提供更多安全功能。举例来说，此类计算机可以防止仅具有机密授权的用户查看绝密文档。用户还可以创建不同的“隔离舱”，在其中保存几组单独的文档。只有被允许访问特定隔离舱的用户，才能够访问该隔离舱中的文档，即便该用户具有足够的安全授权也是如此。

这种高安全性的授权称为“隔离舱式安全”（仅访问自己需要知道的信息）。保护机密的一个重要方面是只允许绝大部分用户访问一小部分信息。使用某一个重要隔离舱的用户应被禁止访问其他重要隔离舱。而那些知道大量机密的人员，例如安全局局长 Alexander，则将受到持续的“机密服务”保护。

一个隔离舱可以是“在无有效搜查令的情况下监听美国人电话记录”。另一个隔离舱可以是“在无有效搜查令的情况下监听美国人的国内电话谈话，读取他们的电子邮件。”[3,12,17,22] 第三个隔离舱可以是“窃听同盟国家领导人的电话”。由于斯诺登从未参与其中任何项目，因此他缺少足够的授权，也无法访问这些计划的文档，甚至根本不知道这些计划的存在。但实际情况是，国家安全局纵容斯诺登一个人在不受任何约束和监控的情况下访问了 170 万份文档。

桔皮书还提出了一个重要概念：不要相信任何一名系统管理员。

而是应该采用双人授权方式，例如，承担角色 1 的系统管理员负责管理系统更改队列，如添加新帐户或更改现有帐户。承担角色 2 的系统管理员无法发出此类请求，但在更改生效之前，必须由他批准这些队列请求。当请求其他程序无法显示的密码时，符合桔皮书规范的操作系统也会显示一个特殊符号，从而防止使用登录模拟器。而斯诺登可能使用了登录模拟器。

这种双人授权方式的成本如何呢？ 2013 年，国家安全局有大约 40000 名雇员，也许还有 40000 承包商员工，其中包括 1000 名系统管理员。[8,25] 再增加 1000 名系统管理员对前一批系统管理员进行监控，其工资成本也就增加微不足道的 1%。

鉴于这一点，国家安全局是否打算采用双人授权方法，执行他们自己制定的桔皮书策略呢？正好相反，国家安全局打算解雇 90% 的系统管理员，以限制人员访问，而将大多数服务器放在他们自己的云中。[1] 云只是可以通过网络远程访问的一系列计算机的别称而已，通常由其他方（通常是供应商，也就是承包商）进行管理。或许他们还会聘用斯诺登的前雇主博思艾伦咨询公司来管理云。

日志事件和监控器。国家安全局应该监控用户以多快的速度访问多少文档，并对此进行检测和限制。令人震惊的是，除了国家安全局数据失窃之外，还发生过类似的大量数据失窃事件，Target 超市在 2013 年底丢失了 4000 万张信用卡的数据（包括本人的信用卡），而且所有人都对此毫无察觉，更没有采取任何措施。如果有严密的实时监控和自动的事件响应，他们应该已经及早检测到这些事件，防止大多数违规现象的发生。

开源的 Logcheck 和 Logwatch 程序可以接近实时地生成异常事件报警，Fail2Ban 程序则可将攻击者拒之门外。所有这些程序都是免费的，而且可以轻松进行自定义，以检测文档下载数量过多的异常情况。市场上有很多类似的商用应用程序，而国家安全局当然有预算开发自己的应用程序。

禁止访问互联网或使用家庭网络。显而易见，这一策略旨在防止机密数据被带出安全大楼。但如果在非工作时间出现问题，系统管理员每次都必须驾车赶到办公室，或者必须始终留在工作现场。美国中央情报局前局长因为将机密数据带回家工作而被辞退，因为他违反了禁止这种行为的严格安全策略。（他并不是窃取数据；他只是想在家工作。）斯诺登将机密材料带回家进行处理，并用风帽遮盖自己和计算机，防止被女朋友看到。[19] 显然，他可能已经拍摄了屏幕照片。

防止可移动介质离开大楼。回忆一下我们前面所说的安全环。其中一个安全环就是防止可移动介质离开大楼。所有加油站运营商都明白这一点，在每把休息室钥匙上都附带一个大型物件。国家安全局可以将所有 U 盘放在一个大型钢盒内，或者更换标准 USB 接口，采用自定义设计接口的计算机很难复制数据。

创造性地使用加密。斯诺登的工作之一是将大量机密数据从一台计算机复制到 U 盘上，再将该 U 盘连接到另一台计算机并下载数据。他很可能在下载所需数据之后，私自藏匿了 U 盘并将它带回家。国家安全局原本可以使用公钥加密方法，轻松地防止这种行为发生。[33] 在公钥加密中，有两个相关密钥：公钥和私钥。如果原始“明文”使用公钥进行加密，则只能使用私钥进行解密，而不再使用加密数据的公钥。

国家安全局应该为需要传输数据的每位系统管理员创建公钥 / 私钥对，并在每台计算机上为每位系统管理员提供单独帐户，用于传输数据。在源计算机上生成此加密数据的人员（例如斯诺登）必须向被允许写入 U 盘的自定义程序提供另一位系统管理员（例如 Julia）的公钥 ID；软件不允许使用他自己的公钥。被允许进行数据传输的这组系统管理员，不能与具有源计算机和目标计算机上的根访问权限的一组系统管理员有任何共同成员。换言之，“数据传输系统管理员”（例如斯诺登）不具有对计算机的根访问权限或物理访问权限，而具有根访问权限或物理访问权限的系统管理员则被禁止在系统之间传输数据。这种责任分离至关重要。只

插图由 PETER CROWTHER ASSOCIATES 提供

有自定义程序（而非系统管理员）才被允许向 U 盘进行写操作。该计算机会使用 Julia 的公钥加密数据，并将该加密数据写入 U 盘。

然后，斯诺登在目标计算机上（安装了具备对 USB 驱动器的唯一访问权限的程序）登录帐户之后，使用该计算机上的自定义程序，将加密数据下载到目标计算机。该程序将向斯诺登提示要将加密数据传输到的帐户（例如 Julia 的帐户），然后将加密文件移动至她的帐户。Julia 将会登录目标计算机，并向自定义程序提供密码，对她的加密私钥进行解锁，或者提供她的指纹或 RFID 工牌，随后自定义会将数据解密到 Julia 的帐户。完成后，她可将数据移动到目标计算机上的最终位置。这个实施过程非常繁琐。

当然，承担数据传输工作的系统管理员不应具备对这些计算机的根管理访问权限，因为该权限会绕过此自定义程序的限制。这些计算机必须运行符合桔皮书规范的最新版本操作系统，在任何情况下都需要两位系统管理员，才能获取访问权限。此外，斯诺登不应该掌握 Julia 的指纹或密码，以及用于身份验证的工牌（如果使用）。开源的 GNU Privacy Guard (GPG) 软件使用加密格式，将私钥存储在磁盘上或其他位置，只能通过提供密码或其他身份验证方式进行解密。[15]

因此，任何一名管理员都无法单独对在加密存储到 U 盘上的数据进行解密。这样可以阻止斯诺登通过 U 盘窃取数据的行为。使用开源 GPG 加密程序，阅读本文的很多读者都可以在一两天内编写这些自定义程序（在源计算机和目标计算机上运行）。因此，即便 U 盘被偷运出国家安全局大楼，也将毫无价值。

同样，文件级加密也可以利用独立的公钥 / 私钥对，为高度机密文件提供另一道安全环。只有具备读取这些文件的权限的用户（而不是承担复制文件的任务的用户）才拥有解密它们所需的私钥。使用目标系统的用户（在斯诺登和 Julia 进行合法复制之后）将能够解密文件。但是，这些系统管理员即便通过读取原始磁盘的方法，也无法查看解密的文档。仅通过这种简单的预防措施，就足以防止斯诺登批量窃取很多文档。还可以结合使用公钥加密来控制系统间数据传输，这样斯诺登必须通过这两道极其挑战性的安全环，才能窃取数据。在处理机密数据的所有计算机上使用加密文件系统或全磁盘加密，将会提供另一道安全环。

如果每个季度或每年执行一次外部安全审核，应该可以发现国家安全局的安全问题，或许能够及时修复以阻止斯诺登的行为。

针对入侵做好准备，以最大程度减小损害。国家安全局官员 Ledgett 承认：“我们还首次了解到，损害评估中考虑到了斯诺登可能在国家安全局系统上留下了 BUG 或病毒，就如同安放定时炸弹那样。”[19] 国家安全局应该针对可能发生的入侵做好准备，最大程度地减小破坏，并且迅速可靠地评估损害。例如，可以做好准备，将系统的当前状态与在入侵之前创建的受信任备份进行比较。这种比较可在不同的受信任系统上运行。[29] 如果使用安全鸟，再加上分散风险，可以最大程度地减少损害。他们本可以持续运行文件系统完整性检查器，以检测文件篡改。

定期安全审核。安全是一个长期持续的过程。如果每个季度或每年执行一次外部安全审核，应该可以发现国家安全局的安全问题，或许能够及时修复以阻止斯诺登的行为。此类审核非常普遍，而且也被视为最佳实践。它类似于美国政府要求大型企业进行的外部财务审计。安全审核报告应该经过最高管理层审查，以避免底层人员直接忽略不易调查的问题。

总结

虽然国家安全局拥有几乎不受限制的资源，能够聘用全国最优秀的计算机安全专家，但在利用极为重要

合宪性

国家安全局对所有美国人进行监听，还涉及到另一个非常重要的方面，就是这种行为的合宪性和道德，斯诺登企图利用这一点吸引广泛关注，而且在很大程度上取得了成功。美国宪法第四修正案这样论述：

“人民保护其人身、住房、文件和财物不受无理搜查扣押的权利不得侵犯；除非有合理的根据认为有罪，以宣誓或郑重声明保证，并详细开列应予搜查的地点、应予扣押的人或物，不得颁发搜查和扣押证。”

为什么宪法制定者关注这些权利？为什么我们应该关注这些权利？简而言之，在正直称职的法官执法时，第四修正案可以防止政府官员对无辜公众的严重侵害，包括对他们私人事务的干涉。在殖民地时期的美国，英国国王乔治授权官员在不发出搜查令的情况下对住宅、人身和财物进行大规模搜查，而无视英国法庭 1603 年的 Saman 案例，该案例承认，在没有基于合理根据的搜查令的情况下，房屋主人有权利保护自己的房屋不受非法侵入，即使是国王的使者也不例外。[6,31] 这正是“每个人的住宅就是他的城堡”的涵义（威廉皮特在 1763 年的一次议会演讲中发表的最有力宣言之一，“最贫穷的人可以在其村舍中与王室的一切军队对抗。村舍可能脆弱 ... 但是英国国王不得进入，他的一切武装力量不敢跨越已经倒塌的村舍的门槛。”）

1705 年，这项权利在*恩蒂克诉卡林顿*案中再次得到承认。英国法官认定，任意性搜查令导致很多住宅（包括恩蒂克的住宅）遭到侵害，国王的手下闯入恩蒂克的家，打开上锁的桌子和箱子，扣押与搜查目标毫不相关的很多文件，这种行为触犯了英国法律。法庭认为针对恩蒂克的搜查令过于宽泛，并非基于合理的根据，而且允许扣押不相关文件，甚至没有记录扣押了哪些物件。请注意，这起法庭案件是由恩蒂克向英格兰王座法院提起诉讼的。[16,31] 在现代社会，个人的计算机和电话不就相当于上锁的桌子吗？电子信息当然也属于私人物品，*牛津英语词典*正是这样定义财物的。个人财物受到第四修正案保护。

2013 年 12 月 28 日，美国法官 William H. Pauley III 裁定美国人不得针对国家安全局的监听行为提出上诉。具体来说，他驳回了美国公民自由联盟 (ACLU) 的上诉，并表示：“若非爱德华·斯诺登未经授权泄密，ACLU 根本不知道《爱国者法案》第 215 条允许收集与电话号码相关的电话元数据。”[7,34]《爱国者法案》第 215 条还要求对美国人的监听永远保密。

Pauley 的裁决认为，美国人不能置疑政府行为的合宪性，因为他们只是通过其他人的非法行为才知道这种政府行为。在作者看来，这种裁决更像是前苏联的论调。而且，它还与 200 年前宪法制定者的精神背道而驰，国父们认为，个人——无论何种身份——都有资格享受宪法赋予的权利，对于侵犯其权利的行为可以随时进行抗争。政府的唯一辩护是没有发生违法行为。

1969 的一次美国法庭裁决认为“第四修正案在很大程度上是对任意性搜查令和无搜查令搜查的回应，这些搜查导致英政府与殖民地人民的关系恶化，帮助加快了独立运动 [也就是美国革命] 的进程。因此，在修正案体系中，‘除依照合理根据，不得发出搜查和扣押状’这条规定具有至关重要的作用。”[4,31] 我们可以很容易找到更多类似的美国法庭裁决。简而言之，在没有搜查令情况下的合理搜查需要合理根据，也就是有充足理由相信某人持有非法物品或掌握了犯罪证据。

美国司法部门认为：“法律认定某种类型的搜查是否合理，是在两个重要因素的权衡之下作出决定的。一方面要考量对个人的第四修正案权利的侵犯。另一方面也要考量政府的合法利益，例如公共安全。”[30] “然而，第四修正案的效力在搜查电子设备方面并未终止。”[18]

据*华盛顿邮报*透露，与奥巴马总统处于同一阵营的独立机构隐私与民权监督委员会 (PCLOB) 表示，国家安全局的电话监听计划是非法的，应该终止。这份 238 页的报告称：“我们还没有发现任何一个涉及美国面临威胁的实例，能够证明电话记录计划在反恐调查中发挥了实质性的重要作用。”

PCLOB 的报告还称，国家安全局的电话数据计划依据《爱国者法案》第 215 条也站不住脚，因为该法案“要求政府搜索的记录（例如电话号码）与经过授权的调查相关。”[28] 随意窃取任何美国人的所有电话记录，显然无法依据宪法做出合理解释。

2013 年 12 月 16 日，美国联邦法官 Richard J. Leon 裁决批量收集美国电话公司的电话元数据的行为触犯了美国宪法。这位法官写道：“我想象不出比这种做法更‘不加区分’和‘随意’的侵犯行为，他们未经事先司法许可，以查询和分析信息为目的，利用系统化高科技手段来收集和保留任意一位公民的个人数据 ... 显而易见，这项计划侵犯了国父们在第四修正案中规定的‘特定程度的隐私权’。”Leon 表示，政府“无法援引任何一个实例，证明国家安全局的批量元数据收集计划真的阻止了即将发生的攻击，或者为政府提供了帮助 ...”[21]

最近，我的朋友 Josh 问我对国家安全局监听美国人有何看法，他补充说：“如果这样有助于抓捕恐怖分子，我不介意他们对我进行监听。”我提醒他说，在国会面前宣誓作证时，国家安全局局长 Keith B. Alexander 承认，他们蓄意对 3 亿美国人进行监听，却从未通过这种方式挽救过任何一个美国人的生命。我问 Josh，如果国家安全局的分析人员监听他和妻子的亲密对话，他会做何感想。他似乎对此感到不太高兴。

Josh 有一个小女儿，我问：“假设几年后，16 岁的女儿打电话说‘爸爸，我在朋友这里。你能来

接我吗？我喝了一点酒，开车不安全。非常对不起。’” 如果国家安全局监听到这段对话，使用电话的 GPS 定位到他女儿的位置，向当地警方提供这份私人电话对话记录，然后警方以未成年人酗酒为名拘捕他的女儿。他会高兴吗？ Josh 此时显得有些愤怒了。你希望对自己的性取向或癖好保密吗？你有什么样的宗教信仰？在总统选举中投了谁一票？你有怎样的股票情报或者专利创意？你和谁通话管政府的事吗？

没错，国家安全局的确在监听国内电话和偷窥电子邮件，获取有关通话方的私人信息。[3,12,17,22] 据路透社在 2013 年 8 月 5 日报道，美国禁毒署 (DEA) 承认他们使用了从国家安全局非法获取的信息并且伪造证据来源。其中包括国家安全局通过情报拦截、窃听、告密者和大型电话记录数据库获取的信息，所有信息的收集既没有相应的搜查令，也没有合理的理由。然后，美国禁毒署将这些信息提供给全国的机构，帮助他们发起对美国人的犯罪调查。[27] 显而易见，这正是第四修正案意图禁止的行为。这些是政府的职责所在吗？

曾经担任新泽西州高级法院最年轻法官的 Andrew P. Napolitano 将奥巴马总统在 2014 年 1 月 17 日公开承诺的国家安全局改革称为总统注射的一针安慰剂。[23,32] 电子前线基金会 (EFF) 对总统改革的评分为 3.5 分（满分 12 分）。[5]（EFF 是一个非盈利性组织，致力于捍卫公民在电子世界中的权利，也许是与国家安全局监控美国人的行为做斗争最积极的组织。）参议员 Rand Paul 认为奥巴马建议的改革只会引发“换汤不换药的同样违宪的计划。”[14] 国家安全局的很多行动都是在 911 事件之后小布什担任总统期间启动的。国家安全局监听美国人的行为是否不符合宪法，或者违反了宪法第四修正案？从本文回顾的 400 年历史来看，作者只能得出一个结论。

而又简单和成本低廉的良好计算机安全实践方面，他们似乎比较懈怠。

本文涵盖的大多数良好安全实践都在作者于 2000 年出版的《*Real World Linux Security*》中进行了讨论。[29] 其中最重要的安全实践还在作者的文章“The Seven Deadly Sins of Linux Security”中进行了讨论，该文章发布在 ACM *Queue* 的 2007 年五月 / 六月刊上。

很荣幸国家安全局总部保存了本人拙作的作者签名本。国家安全局的绝大多数雇员和承包商都是具备出色工作能力且守法敬业的爱国公民。令人遗憾的是也有极少一部分人无视警告，认识不到这些安全问题急需采取补救措施，才能避免严重违规行为。 C

queue.acm.org 上的相关文章

计算机协会隐私和安全面临风险
Whitfield Diffie 和 Susan Landau
http://queue.acm.org/detail.cfm?id=1613130

更多加密并非解决方案
Poul-Henning Kamp
http://queue.acm.org/detail.cfm?id=2508864

四十亿个“小监视者”：隐私、移动电话和无处不在的数据收集
Katie Shilton
http://queue.acm.org/detail.cfm?id=1597790

参考资料

1. Allen, J. NSA to cut system administrators by 90 percent to limit data access.Reuters.Aug. 9, 2013; http://www.reuters.com/article/2013/08/09/us-usa-security-nsa-leaks-idUSBRE97801020130809.
2. Block, M. Snowden’s document leaks shocked the NSA, and more may be on the way.National Public Radio.Dec. 17, 2013; http://www.npr.org/templates/story/story.php?storyId=252006951.
3. Brosnahan, J. and West, T. Brief of Amicus Curiae Mark Klein.May 4, 2006; https://www.eff.org/files/filenode/att/kleinamicus.pdf.
4. *Chimel v.California*, 395 U.S.752, 761 (1969).
5. Cohn, C. and Higgins, P. Rating Obama’s NSA reform plan:EFF scorecard explained.Electronic Frontier Foundation, Jan. 17, 2014; https://www.eff.org/deeplinks/2014/01/rating-obamas-nsa-reform-plan-eff-scorecard-explained.
6. Coke’s Reports 91a, 77 Eng.Rep. 194 (K.B.1604).
7. Davidson, A. Judge Pauley to the N.S.A.:Go Big.*The New Yorker.*Dec. 28, 2013; http://www.newyorker.com/online/blogs/closeread/2013/12/judge-pauley-to-the-nsa-go-big.html.
8. Davidson, J. NSA to cut 90 percent of systems administrators.*Washington Post.*Aug. 13, 2013; http://www.washingtonpost.com/blogs/federal-eye/wp/2013/08/13/nsa-to-cut-90-percent-of-systems-administrators/.
9. Defense Logistics Agency.Critical nuclear weapon design information access certificate; http://www.dla.mil/dss/forms/fillables/DL1710.pdf.
10. Department of Defense Trusted Computer System Evaluation Criteria, a.k.a., Orange Book 1985; http://csrc.nist.gov/publications/history/dod85.pdf.
11. Dilanian, K. Officials:Edward Snowden took NSA secrets on thumb drive.*Los Angeles Times.* June 13, 2013; http://articles.latimes.com/2013/jun/13/news/la-pn-snowden-nsa-secrets-thumb-drive-20130613.
12. Electronic Frontier Foundation (eff.org).NSA spying video, includes comments from many well-known respected people and reminders of past violations; http://www.youtube.com/watch?v=aGmiw_rrNxk.
13. Esposito, R. Snowden impersonated NSA officials, sources say.NBC News.Aug. 28, 2013; http://investigations.nbcnews.com/_news/2013/08/28/20234171-snowden-impersonated-nsa-officials-sources-say?lite.
14. Everett, B. and Min Kim, S. Lawmakers praise, pan President Obama’s NSA plan.*Politico.*Jan. 17, 2014; http://www.politico.com/story/2014/01/rand-paul-response-nsa-speech-102319.html.
15. GNU Privacy Guard; http://www.gnupg.org.
16. Howell’s State Trials 1029, 95 Eng.807 (1705).
17. Klein, M. and Bamford, J. *Wiring Up the Big Brother Machine...and Fighting It.*Booksurge Publishing, 2009.
18. Legal Information Institute, Cornell University Law School.Fourth Amendment: an overview; http://www.law.cornell.edu/wex/fourth_amendment.
19. Miller, J. CBS News “60 Minutes.” Dec. 15, 2013; http://www.cbsnews.com/news/nsa-speaks-out-on-snowden-spying/.
20. Lemos, R. Security guru:Let’s secure the Net.ZDnet, 2002; http://www.zdnet.com/news/security-guru-lets-secure-the-net/120859.
21. Mears, B. and Perez, E. Judge:NSA domestic phone data-mining unconstitutional.CNN.Dec. 17, 2013; http://www.cnn.com/2013/12/16/justice/nsa-surveillance-court-ruling/.
22. Nakashima, E. A story of surveillance.*Washington Post.* Nov 7, 2007; http://www.washingtonpost.com/wp-dyn/content/article/2007/11/07/AR2007110700006.html.
23. Napolitano, A.P. A presidential placebo – Obama’s massive NSA spying program still alive and well. Fox News.Jan. 23, 2014; http://www.foxnews.com/opinion/2014/01/23/presidential-placebo-obama-massive-nsa-spying-program-still-alive-and-well/.
24. Presidential Executive Order 13526 12/29/2009; http://www.whitehouse.gov/the-press-office/executive-order-classified-national-security-information.
25. Rosenbach, M. Prism exposed:Data surveillance with global implications.Spiegel Online International. June 10, 2013:2; http://www.spiegel.de/international/world/prism-leak-inside-the-controversial-us-data-surveillance-program-a-904761.html.
26. Schwartz, M. Thumb drive security:Snowden 1, NSA 0.InformationWeek.June 14, 2013; http://www.informationweek.com/infrastructure/storage/thumb-drive-security-snowden-1-nsa-0/d/d-id/1110380.
27. Shiffman, J., Cooke, K. Exclusive:U.S. directs agents to cover up program used to investigate Americans.Reuters.Aug. 05, 2013; http://www.reuters.com/article/2013/08/05/us-dea-sod-idUSBRE97409R20130805.
28. Smith, C. BGR.Jan. 23, 2014; http://news.yahoo.com/watchdog-says-nsa-phone-spying-program-illegal-end-130014396.html.
29. Toxen, B. *Real-world Linux Security:Intrusion Detection, Prevention, and Recovery.*2nd Edition. Prentice Hall, 2002.
30. U.S. Courts.What does the Fourth Amendment mean?; http://www.uscourts.gov/educational-resources/get-involved/constitution-activities/fourth-amendment/fourth-amendment-mean.aspx.
31. U.S.Government Printing Office.Fourth Amendment; http://beta.congress.gov/content/conan/pdf/GPO-CONAN-2013-10-5.pdf.
32. *Washington Post.*Transcript of President Obama’s Jan. 17 speech on NSA reforms, 2014; http://www.washingtonpost.com/politics/full-text-of-president-obamas-jan-17-speech-on-nsa-reforms/2014/01/17/fa33590a-7f8c-11e3-9556-4a4bf7bcbd84_story.html.
33. Wikipedia.Public-key cryptography; http://en.wikipedia.org/wiki/Public-key_cryptography
34. Wikipedia.Edward Snowden; http://en.wikipedia.org/wiki/Edward_Snowden#NSA_rulings_in_federal_court.

Bob Toxen (bob@VerySecureLinux.com) 是专长于 Linux 和网络安全的 Horizon Network Security 公司的首席技术官，也是 Berkeley Unix 的开发者之一。

译文责任编辑：诸葛建伟

DOI:10.1145/2594413.2594422

如何在不增加成本的情况下应对软件解决方案日益增长的需求?

作者：SHIMEON PASS 和 BOAZ RONEN

弥补软件价值缺口

软件价值缺口指 IT 部门中尚未挖掘的潜力，可用于增加组织的整体价值。在今日动态的商业环境中，很多公司依赖源于软件解决方案的价值创造，IT 部门则负责交付这些方案。[a] 这些解决方案对日常的业务运行、控制和增长以及遵循管理需求至关重要。在很多情况下，合适的软件解决方案成为推出新的商业行动和创新的先决条件。

在大多数公司中，对软件解决方案和功能的需求超出开发和维护方面的 IT 预算（或相关人力资源的产能）的程度高达 500%，特别是考虑了软件解决方案的”隐藏队列”后，情况尤为如此。[9] 出现这种情况的原因是，即便是业务最繁荣的公司也无力为 IT 部门分配无限的资源，因为它会给组织整体的价值带来不利影响。[b]

a 通过软件解决方案得到的价值创造等于因使用解决方案而获取的边际贴现现金流。
b 公司的价值为其贴现现金流。

公司董事会和高管会定期商讨如何确定可承受的 IT 预算，这也说明业务需求尚未得到完全满足。不仅如此，最终的软件交付流太小，太慢，太贵。在主要行业（比如金融服务、电信、保险、航空、保健和互联网零售业）以及政府机构中，软件价值缺口的问题尤为扰人。

常规方法

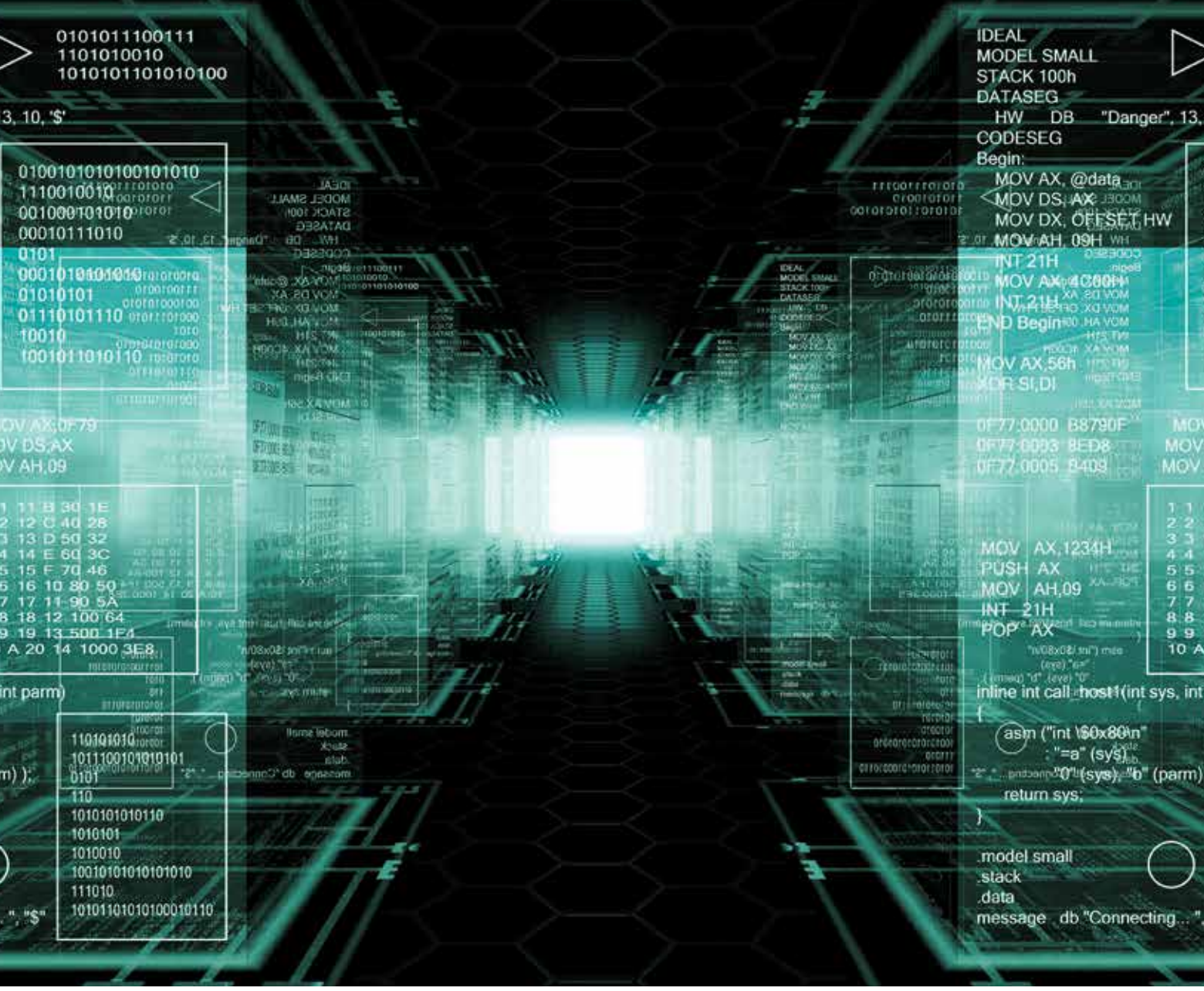
弥补软件价值缺口的方法之一是投入更多的 IT 资源，如雇佣更多的 IT 员工和分包商员工，把项目外包到分包公司（在岸外包和 / 或离岸外包）或者购买现成的软件包。然而，公司能够增加的员工人数或预算并非没有限制。而且，仅仅在 IT 方面投入更多资金并不能解决问题。在外包软件开发和购买软件包的过程中，均需要投入大量的内部 IT 资源来开展需求定义、系统分析、软件解决方案集成、数据迁移、数据库、数据仓库、实施和维护。多数情况下，组织初期定义需要和达成一致的能力，以及随后实施、吸收和接受新系统的能力限制了软件解决方案的部署程度。

另一种方法是实施下列一种或多种手段来提高软件开发的生产率：

- 敏捷、迭代式增量软件开发（Scrum）和极限编程（XP）中的开发方法；[12]

» 重要见解

- **IT 部门交付的软件解决方案往往没能发挥全部的价值创造潜力。**
- **需要采用综合性的方法提升生产率，确保 IT 的价值创造。**
- **提升产量、缩短提前期的目标可通过简便的管理工具达成。**



插图由 MICHELANGELUS 提供

- 关键链法和工具，用于减少软件开发项目的生产周期，提高交付日期的可靠性；[1,7]
- 精益技术，用于建立精打细算的 IT 组织；[2]
- DevOps（开发和运营的组合）技术；[8] 以及
- 需求管理和软件重用。[3]

还有一种方法是通过某些优先级评估准则确定软件开发请求的先后顺序。利用这些常规方法后，可以缩小软件价值缺口，缩小的程度各异，但通常来说都不太大。因此，缩小软件价值缺口还有继续改进的空间。

软件价值缺口的范围

为了理解为何软件开发和维护未能为组织创造足够的价值，我们列出并分析了与常见的 IT 开发和维护环境有关的多个通用问题。通过聚焦后的当前现实树（fCRT），我们分析了这些问题，并指出造成这些软件价值缺口的根本原因。[11] 首先，我们 [6] 列举了与软件解决方案开发和维护有关的各种通用不良效果（UDE）。

- 软件解决方案的开发并未给组织创造足够的价值（主要的通用不良效果）；
- 软件解决方案的生产率不足；
- IT 部门的成本太高；
- 已交付的软件解决方案往往被弃之不用。
- “实现组合”[c] 中的软件解决方案无法实现价值最大化；
- 选择软件组合时，无有效的选择机制；
- IT 开发人员的无效时间过多；
- 未能妥善定义需求；
- 软件解决方案只能部分满足要求；
- 生产周期太长；
- 缺乏软件性能评估，或者性能评估造成误导；
- 无有效的价值聚焦；
- 内部客户的参与度不足；以及

c 实现组合由为开发选择的软件解决方案组成。

图 1 用于软件价值缺口的聚焦后当前现实树

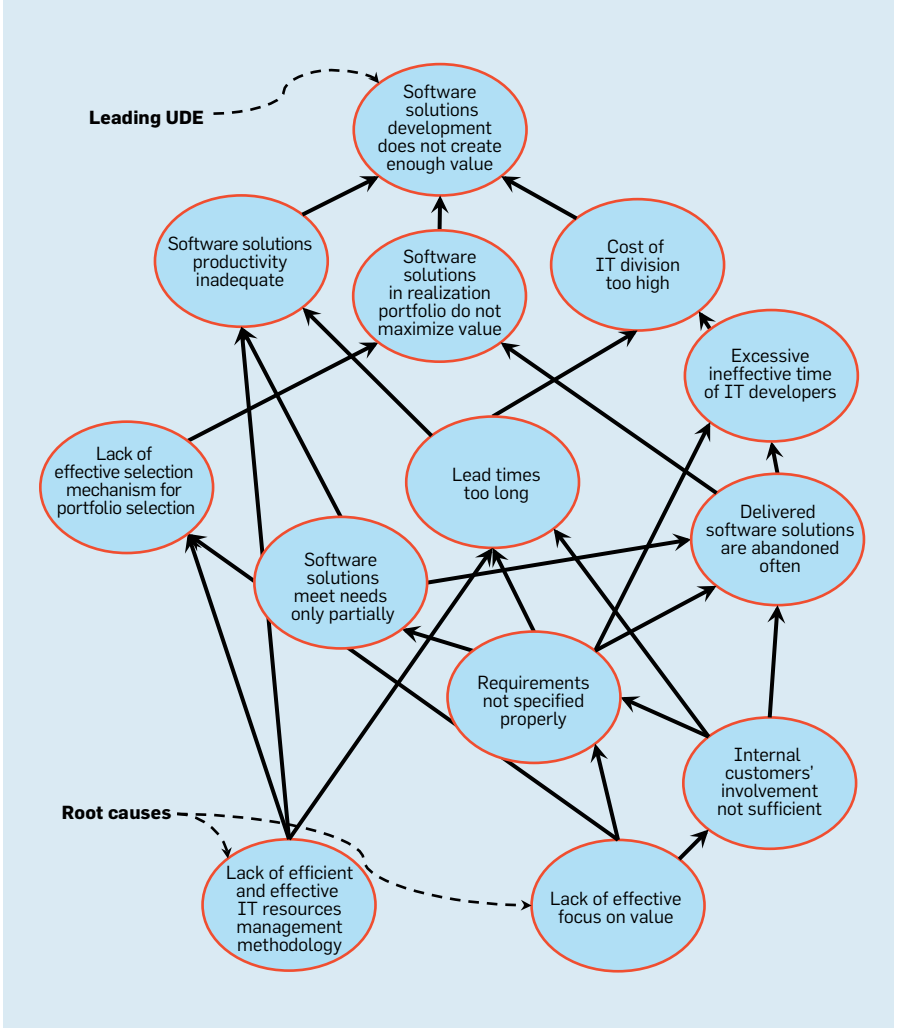
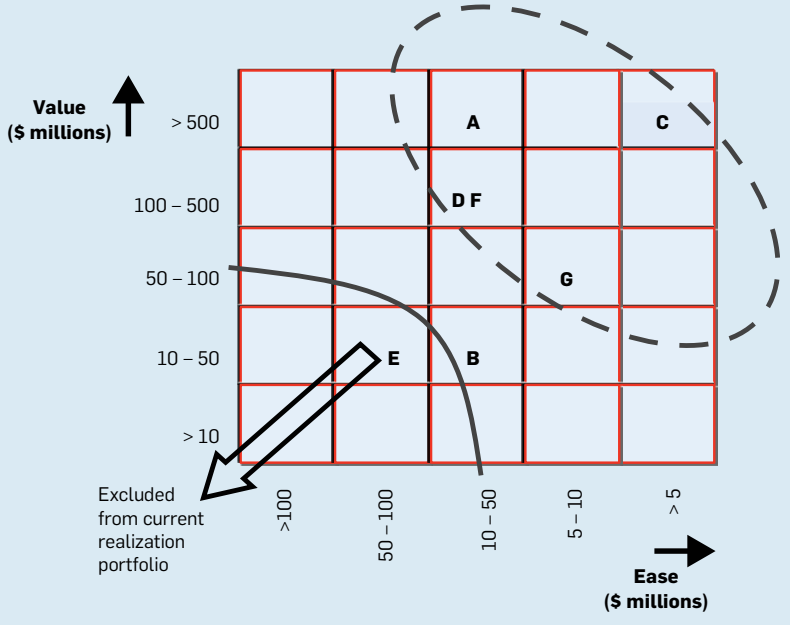


图 2 选择软件解决方案时使用的聚焦矩阵（不按比例）



▸ 缺乏高效、有效地管理 IT 资源的方法。

在此，我们把通用不良效果（UDE）放在了聚焦后的当前现实树（fCRT）中（见图 1）。图 1 中，箭头说明了通用不良效果之间的因果关系，箭头方向从表示原因的通用不良效果指向表示结果的通用不良效果。构建聚焦后的当前现实树时，首先把主要通用不良效果放在树的顶端。然后从列表中选出造成该主要通用不良效果的各种通用不良效果，把这些通用不良效果放在主节点下面，再用因果箭头把它们连接起来。接着继续进行这一步骤，把作为其原因的更多通用不良效果加在树中已经存在的通用不良效果下，随后再画上必要的因果箭头。对于位于聚焦后的当前现实树底端的通用不良效果，由于它们没有连上其他的通用不良效果作为其原因，所以我们认为它们是软件价值缺口的根本原因。

该分析确认了两个主要的根本原因：缺乏有效的价值聚焦、缺乏有效的 IT 资源管理方法。

软件价值缺口的根本原因不仅仅是由于 IT 部门的管理实践出了问题，同时也归结于组织的行为和规范存在不足。弥补缺口意味着通过更好地响应公司和组织的业务需要，为他们创造更多的价值。价值的提升可以通过做好下列两个方面得以实现：选择价值最高的 IT 系统进行开发和维护（效益），以及提升 IT 部门的生产率（效率）。在此，我们勾勒了一个综合方案，通过利用现有资源在弥补软件价值缺口方面取得突破。不仅如此，它还能和之前概括的常规方法互补，相互促进。

聚焦价值

因为软件解决方案的需求远远超出供给，所以从本质上来说，IT 部门会成为其所在组织的永久瓶颈，通

常也无法满足公司指定的所有 IT 需求。[d] 不幸的是，实现组合中是否包含某一项目通常并非依据项目为组织带来的实际经济价值而定，它往往依赖于提出请求的部门在组织中拥有的权力，或是依赖于该请求在代办队列中的滞留时间。

改变这种反生产的现实的途径之一是实施战略关卡流程，以获得实现组织价值最大化的软件开发项目组合。[5] 与众人的直觉相反，在很多情况下，促成价值最大化的组合只包含了少量项目。战略关卡流程由评估和立项组成，依照 IT 请求所需的 IT 资源对价值创造的相对期望贡献程度来确定 IT 请求的优先级。这样还可以确保 IT 请求与公司的增长战略和计划步调一致。

战略关卡流程可每年或（最好）每季度开展一次。提出请求的部门需为每个软件解决方案的请求提供价值创造的估算值，并辅以相应的商业计划作为支撑。同时，IT 专家需要提供方案总体拥有成本（TCO）的粗略估算。对于法规要求的软件解决方案，因为它们是强制的，所以给它们一个无穷大的值。

战略关卡流程中最好不要考虑非常小的软件解决方案和变更请求（CR），而是留出总体 IT 预算的 15% 到 25%，然后把这些预算拨给各个部门。为变更请求（CR）拨出高比例的工作量 / 预算后，往往会提高内部用户的满意度。而在大型和中型软件解决方案中投入较大比例的资源，则有利于提升组织的总体价值。如何达到理想的平衡不亚于一个战略决策。

可以通过下列步骤选择软件解决方案：

- 列出软件解决方案，并标出价值创造和总体拥有成本（TCO）的估算值（见表 1）；[11]
- 按软件解决方案的价值创造和总体拥有成本（TCO）把方案映射到聚焦矩阵中（见图 2）。[11]
- 实现组合中的待选项目主要从该矩阵的右上象限处选取（价值最高，成本最低）；以及
- 在相应阶段可用的总体拥有成本（TCO）预算全部分配完成前，继续进行选择。

在软件解决方案的交付过程中，价值创造也是批准范围变更的合理依据。潜在的范围变更涵盖的内容很多，从功能的小改动到大量的范围变更，甚至是项目置换。如果范围变更的结果创造了正净值，则证明该变更合理。净值等于软件解决方案因范围变更而增加的价值创造减去实现范围变更的成本，减去项目中断的成本，再减去可能发生的延期交付的成本。

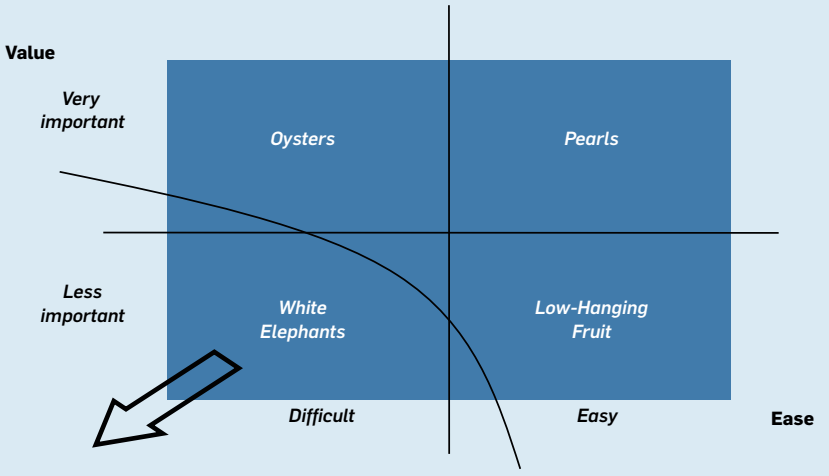
战略关卡流程

为了说明如何开展数值化和图形化的战略关卡流程，让我们来看一个例子。在该例中，公司利用战略关卡机制筛选了多个公司部门提出的七种软件解决方案。经过估算，这些解决方案所需的总体拥有成本（TCO）的预算总额为 1.88 亿美元。由于批准的下一年的总体拥有成本（TCO）预算只有 1.1 亿美元，所以需要进行筛选；表 1 列出了所有的软件解决方案，然后再依其价值创造和实现难度把这些方案映射到聚焦矩阵中，如图 2 所示。价值最高的软件解决方案主要位于聚焦矩阵的右上象限中。在矩阵的帮助下，公司高层选择了接近总体拥有成本（TCO）总额限制（1.1 亿美元）的软件解决方案。在本次战略关卡流程中，公司管理层选择在实现组

表 1 通过聚焦表选择软件解决方案

#	软件解决方案	价值（单位：百万美元）	难度 = 总体拥有成本（TCO）（单位：百万美元）
A	客户关系管理	550	45
B	资产管理	45	29
C	账单透明度	法规	10
D	呼叫中心的知识库	170	15
E	人力资源	25	55
F	升级商业智能	145	25
G	升级营销管理	55	9
	总体拥有成本（TCO）总额		188

图 3 概念性的聚焦矩阵



d 永久瓶颈指一直为瓶颈的资源，因为其需求非常大。

合中纳入软件解决方案 A、C、D、F 和 G。

还可以通过按软件解决方案的具体贡献或软件解决方案的价值与其总体拥有成本（TCO）的比值对其进行排序，再选择在实现组合中纳入哪些软件解决方案。[11] 不过，聚焦矩阵还能让管理层按战略考量调整组合，并考虑业务和市场中无形因素。

在概念性的聚焦矩阵（见图 3）中，右上的象限被称为”珍珠”，因为其中包含了价值最高的待选项。其他三个象限被分别称为“牡蛎”（有价值但难实现），“低挂的果实“（容易实现但价值稍低）和“白象”（需要避免）。这些术语便于管理层之间进行交流。

战略关卡的最终决定仍由高层做出，如果高层觉得牡蛎的战略重要性超过了价值创造的估算值，他们可决定在实现组合中纳入牡蛎。与此类似，如果高层认为取得快速成功较为合理，则他们也可以考虑纳入低挂的果实。

旁观者可能会怀疑，那些急切推动自己的软件解决方案请求的部门领导是否会故意夸大价值创造的估计值。为了消除这种风险，软件解决方案的每个请求必须附有商业计划作为支持（或至少是微型的商业计划）。另外，部门领导必须对预期的价值创造做出承诺，并在其收入目标中包含该方案产生的现金流估计值。与实现这些价值创造目标有关的后续机制则需要与该组织的关键绩效指标和激励制度加以结合。

浪费可以避免

研究发现，由于下列原因，IT 开发人员至少 50% 的时间 [11] 被浪费了：

- 需求定义不完整或定义模糊导致返工（“不完整的工作包”）；
- 需求和范围直到最终交付阶段仍然频繁变更造成返工；大多数的此类变更并非“必须”，而是“更佳”。

要想发挥效果，绩效评估必须与价值创造的目标妥善关联。

- 软件解决方案已经开发和交付，但是最终没有使用（时常发生）；
- 需求超规格，其中包含了很少应用或从未应用的功能和特性；以及
- 为单个开发人员分配了太多的任务，导致他在多个任务间切换，浪费了时间（多任务处理不当）。

浪费无法完全消除，但却可通过七种简单的管理举措或补救措施大大降低。

完整工作包概念。[e] 为了避免返工造成 IT 资源浪费，组织应该从整体出发采取和推行完整工作包概念。[10,11] 如果项目的商业理由、需求、商业计划或工作说明书不完整或模糊不清，则不应该批准该项目的开发工作。与此类似，如果需求 / 规格说明 / 设计不完整或模糊不清，则不应该把任务分配给单个开发人员。提出请求的部门和 IT 员工应携手定义用于需求和规格说明的完整工作包，列出其内容。

由于提出请求的部门必须提供需求和规格说明的完整工作包，他们必须了解自身的需要和所需的软件解决方案。这可以尽可能地减少造成返工的需求变更的程度，也能使已开发和交付但最终放弃的项目数量降到最低。不仅如此，推行完整工作包概念后，往往会使业务部门和 IT 部门之间的交流和协作变得更为紧密。

同时，还需要为项目生命周期内执行的各种任务和活动定义完整工作包；例如，项目经理给程序员的系统分析文件必须包含完整的工作包信息，能让程序员正常开展工作。与此类似，给测试人员的文件也必须包含完整的工作包信息，能让测试员正常完成工作。

消除过度需求，过度设定规格和过度设计的情况。仔细审查软件解决方案中的需求后，往往会发现

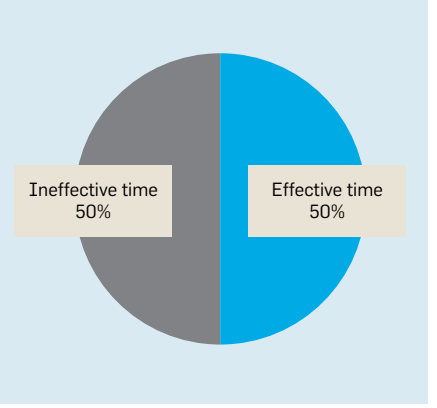
e 任务的完整工作包为在不中断任务的情况下完成任务所需的所有项目的列表。

过度需求的情况出现。软件解决方案中需要的大量功能和特性往往只是锦上添花，而不是必须。在很多情况下，IT 专业人员倾向在项目中引入过度设定的规格和设计，从他们的观点来看这让项目更安全，或者让项目更具挑战性。Coman 和 Ronen[4] 估计，浪费在这些情况中的工时数量超过 30%。观念要改改了。管理人员和程序员，无论其是否身处 IT 领域，都应对其认为是过度要求、过度设定规格或过度设计的挑战性功能和特性负责。在交付周期中，应该多次提起这些问题。例如，在启动会上，应该确定过度的需求，并把它们从拟开发的系统范围中剔除出去。在与风险管理、控制关卡和设计评审有关的团队会议中，也应按此方式确定和剔除过度需求、过度设定的规格和过度设计。

25/25 实践。25/25 法则指出，管理层应尝试中止和停止软件开发管道中约 25% 的项目。对于剩下的项目，应剔除其中不需要或过度要求的特性（约 25%）。高层应通过聚焦矩阵每季度审查一次管道中的所有软件解决方案。业务情况可能已发生变化，价值创造可能严重降低。与此类似，对于某些软件解决方案而言，残存的交付成本最终可能会比期望值高很多。在这种情况下，如果项目已经丧失了其价值创造的潜力，高层应停止项目交付，而不要考虑已经在项目中投入的“沉没成本”。在某些组织中，很多项目可按这种方式加以剔除，进而释放高达 25% 的 IT 部门预算或产能。

与此类似，团队必须详细审查仍留在管道中的那些最复杂且成本高昂的软件解决方案，找出其中的过度需求、过度设定的规格和过度设计，然后把它们剔除出去。此实践移除了不必要的功能和特性，使这些解决方案的交付成本最高可降低 25%。

图 4 有效和无效时间的对比图（改进前）



把大型的、有风险的解决方案分解在多次发布中。在组织中，如果软件可用定期发布（比如每季度）的方式上线，我们建议避免在一次发布中开发和实施多个大型的软件解决方案，特别是在交付会涉及大量的商业和技术方面的不确定因素时，更应如此。如果可能的话，把软件解决方案拆开，放在两或三次连续的发布中上线，可确保提出请求的部门更好的理解其需要和其对解决方案的需求；同时，也降低了技术方面的风险。这种实践不仅降低了因需求变更造成的返工风险以及最终忽略该软件解决方案的可能性，还能更好的适应商业需要。

绩效评估。系统性的绩效评估已经证明是一个提高绩效（包括避免浪费）的强大工具。不过，要想发挥效果，绩效评估必须与价值创造的目标妥善关联。

绩效评估不应是“完美的”或“科学的”或涵盖所有的极端情况。确定完美的评估指标通常很难，也是不可能实现。绩效评估的主要目的是促进组织随时间逐步改进。评估指标虽不完美，但仍能发挥作用。而且随着时间的推移按一致的方式进行评估本身就够好了。我们推荐采用下列七种定期使用的绩效评估指标，它们涵盖了 IT 领域的大多数运营因素。如果其他的绩效评估指标符合组织的总体商业目标，也可以纳入。

*IT 部门的产量。*T= 评估期内交付的软件解决方案的价值创造的估计值总额；

*IT 部门的生产率。*Prod = 评估期内开发的 CR 当量的数额；可采用评估期内交付的 { 大型软件解决方案的数量乘以 9+ 中型软件解决方案的数量乘以 3+ 变更请求的数量 } 之和来确定该当量。

*运营开支。*OE= 评估期内 IT 的 TCO 开支；

在研项目。WIP_1= 评估实例中尚未开发完成的软件解决方案的数量；WIP_2 = 评估实例中为 IT 部门每位开发人员分配的活动的平均数；

*生产周期。*LT= 评估期内所有大型软件解决方案从需求引入至交付之间的平均时间跨度；

质量。Q_1 = 所有软件解决方案交付后六个月内发现的致命缺陷的数量；Q_2 = 评估期内交付的所有软件解决方案的[f] 平均范围稳定性；以及

准时交付率。DDP_1 = 评估期内准时交付的软件解决方案的百分比；DDP_2 = 评估期内准时交付的开发活动的百分比。

这些评估指标是可靠的绩效指标，在 IT 部门的领导和管理人员建立有效的激励机制时相当有用。

较短的生产周期。在较长的生产周期内，有时会提出过度需求和不必要的范围变更。对于这种情况，可利用我们随后讨论的实践大量缩短项目生产周期来加以防止。

范围变更请求的净价值创造。为了防止引入不必要的范围变更请求，评审标准须为前面讨论的结果，即变更的净价值创造为正。

上述七种补救措施可助组织避免在 IT 部门内部产生浪费。为了理解上述措施在生产率提升方面的潜力，我们来看下面这个例子。此

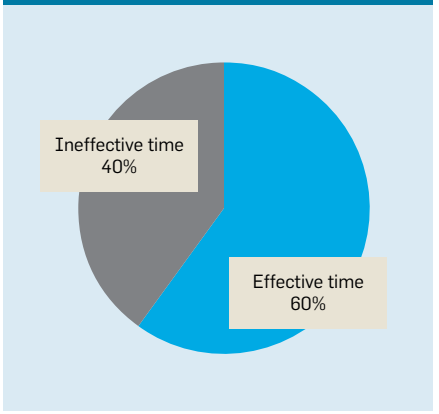
f 范围稳定性反映了在软件解决方案的范围定义中引入的变更的程度。

例中，IT 部门的专业人员因各种原因浪费了约 50% 的工时（见图 4）。假设在主要浪费源头上施行一种或多种补救措施后，浪费虽然未能完全消除，但是保守估算已经降低到了 40%。这意味着，同一资源的生产时间实际已经从 50% 增长至 60%（见图 5）。有效时间从 50% 升至 60% 后，如同增加了 20% 受过培训、经验丰富的资源，而不会在工资、招聘、培训、指导、工作空间、工作站或软件许可方面增加任何成本。

受控发布

提高 IT 效率的可靠实践之一为控制纳入系统的项目，以及控制分配给单个开发人员的任务，这能加快系统中的工作流程，缩短生产周期，并提高生产率。同时，还可以更好地利用 IT 部门的瓶颈资源。

图 5 有效和无效时间的对比图（减少浪费后）



高层次的战术性关卡按照预先制定的优先机制控制纳入开发环节的项目；一次启动太多项目可能会引发混乱，因此最好能错开项目。低层次的战术性关卡机制则按照预先制定的优先机制控制分配给开发人员的任务。只有在下属拥有的已分配任务数少于三或四个时，或工作的所需时长在二到四周之间时，管理人员和团队主管才向下属分配新的任务。这些任务的需求 / 规格说明 / 设计也必须遵循完整工作包策略。我们推荐把大型的任务细分为 5 天到 10 天可完成的各种小活动，而不是把大型任务直接分配给开发人员。除此之外，根据低层次的战术性关卡，IT 管理人员应尽量避免瓶颈处的开发人员参与中断工作的日常活动，比如客户支持和不必要的会议。高层次和低层次的战术性关卡机制同步实施后，会缩短生产周期，加快项目流程，并提高准时交付率和软件质量。

电信业案例分析

B 公司是一家市值几十亿美元的电信公司，其所有业务活动均依赖 IT 部门，包括销售、营销、运营、工程、客户服务、计费和财务。B 公司的高层已经决定在整个组织内实施价值创造方法和工具。他们还参加了为期六天的研讨班。研讨班涵盖了价值创造的管理哲学、方法论和工具，形式包括演示、讨论和实践作业。研讨班结束时，公司高层批准了用于下列六大价值驱动因素以及 IT 部门本身的实施方案：

- 确定和管理 IT 部门的瓶颈；
- 实施战略关卡和 25/25 机制；
- 在部门和其分包商的主要工作流程中推行完整工作包概念；
- 实施高层次和低层次的战术性关卡，剔除过度需求、大型活动和不良的多任务处理；
- 避免无效时间；以及
- 确定 IT 部门的绩效评估。

IT 部门的所有管理人员（直至团队主管）参加了后续为期三天的研讨会，其中涵盖了提升价值创造的举措。另外，组建了几个工作组来处理价值驱动因素。

高层报告了三个主要的结果（三年内）：

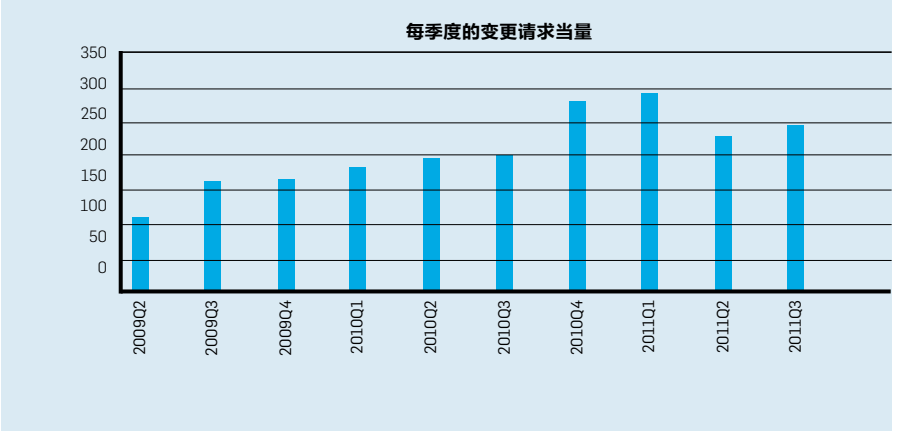
生产率。每季度的 CR 当量从 109 增加至 241，提升了 120%（见图 6）；

运营开支。每年的 TCO 预算基本保持不变；以及

准时交付率（*DDP1*）（DDP_1）。准时交付率从 69% 提高至 76%。

我们研究的其他七家公司中也发现了类似的结果。

图 6 随时间变化而呈现的生产率提升



价值杠杆

预测战略关卡机制的效果并不容易，因为它因具体情况不同而不同，需反复探讨为特定情况提出的所有潜在软件解决方案的实际价值及其难度，以及缺少战略关卡机制的 IT 部门理应选择的实现组合。不过，可以清楚地看出在推行 IT 生产率的改进措施之后公司价值的变化情况。

价值杠杆的示例。让我们来看下这样一家公司，其约 30% 的收入依赖于或者说源于新的软件解决方案。上一年的收入总额为 12.9 亿美元。它的实际变动成本（RVC）为 2.1 亿美元（收入的 16.3%）。实现的

产量（所有收入减去 RVC）为 10.8 亿美元。该公司的固定资产总额为 9.1 亿美元，因此它上一年度的税息折旧及摊销前利润（EBITDA）为 1.7 亿美元（见表 2）。

现在假设公司可以稳定获得的年收入为 12.9 亿美元。如果我们采用保守的税息折旧及摊销前利润（EBITDA）乘数 10，则可得出公司的市值为 17 亿美元。

假设 IT 部门成功实施了本文描述的此类改进实践，进而使得软件解决方案的交付数量提升了 20%。再假设，与当前实现组合中的项目均值相比，20% 的新增软件解决方案的价值创造潜力平均只有 25%。因为仅有 30% 的收入源自新软件解决方案的引入，所以下一年度的新增收入将为 12.9 亿美元乘以 30%，乘以 20%，再乘以 25%=1900 万美元。RVC 可能会成比例的增加至 2.13 亿美元，即仍为约 16% 的收入。

如果 IT 部门利用相同的资源提升了软件解决方案的生产率，而且提升的过程中也未给公司增加人力资源或资产，那么固定成本没有发生变化。在我们的计算中，可以看出公司的税息折旧及摊销前利润（EBITDA）将增至 1.86 亿美元，较上一年度增加 9.4%（见表 3）。使用相同的税息折旧及摊销前利润（EBITDA）乘数 10 之后，可以得出公司的市值达到了 18.6 亿美元，但公司的资源或其他投资并未显著增加。我们的经验表明，实施战略关卡机制后，公司总体价值的增加值甚至会更高。

这些改进步骤需要改变组织的总体文化，因此首席执行官（CEO）、首席信息官（CIO）和高管团队的领导方式也需随之改进。战略关卡机制和 IT 部门的内部改进活动相辅相成，必须同时实施。如果业务部门对 IT 部门推行所需改进的决心抱有疑虑，他们就不情愿参与战略关卡选择机制，也不乐意按完整工作包的形式提交项目请求。与此类似，如果 IT 管理层定义请求时未遵守不含过度需求的完整工作包概念，他们也就不大可能有动力改进自己的流程。在其他改进步骤到位后，可立即在第二个阶段推行把大型软件解决方案细分为多个“阶段”或多次“发布”的文化。

表 2 损益汇总表

	去年（单位：百万美元）	
收入	1290	
实际变动成本	210	[16.3%]
产量	1080	
固定成本	910	[70.5%]
息税折旧摊销前利润	170	[13.2%]

表 3 从损益汇总表中得出的价值创造

	去年（单位：百万美元）		明年（单位：百万美元）	
收入	1290		1309	
实际变动成本	210	[16.3%]	213	[16.3%]
产量	1080		1096	
固定成本	910	[70.5%]	910	[69.5%]
税息折旧及摊销前利润（EBITDA）	170	[13.2%]	186	[14.2%]
Δ（税息折旧及摊销前利润 EBITDA）[~Δ 价值]			+9.4%	

结论

通过高度协作的方式，弥补软件价值缺口的方法可作为本文概述的常规方法的补充。此方法已经过实践检验，无需增加任何投资，但可实现价值提升。而且，几个月内便可以实现改进。如果首席执行官（CEO）或董事会提出此类改善项目的动议，并由首席执行官担任项目的“主管”，则较容易按这些步骤取得成功。当组织中的其他部门（比如营销、销售、研发、工程、项目管理，以及运营部门）寻求其他的价值创造活动时，用于弥补软件价值缺口，进而提升价值创造的措施还能作为此类活动的补充。

参考资料

1. Anderson, D.J. *Agile Management for Software Engineering*. Prentice-Hall, Upper Saddle River, NJ, 2003.
2. Bell, S.C. and Orzen, M.A. *Lean IT*. Productivity Press, New York, 2011.
3. Boehm, B.W. *Software Engineering Economics*. Prentice-Hall, Upper Saddle River, NJ, 1981.
4. Coman, A. and Ronen, B. Icarus' predicament: Managing the pathologies of overspecification and overdesign. *International Journal of Project Management 28*, 3 (Apr. 2010), 237–244.
5. Cox, J.F.III, Boyd L.H., Sullivan T.T., Reid R.A., and Cartier, B. *The TOCICO Dictionary, Second Edition*. McGraw-Hill, Inc., New York, 2012.
6. Goldratt, E.M. *It's Not Luck*. North River Press, Croton-on-Hudson, NY, 1994.
7. Goldratt, E.M. *Critical Chain*. North River Press, Croton-on-Hudson, NY, 1997.
8. Loukides, M. *What Is DevOps?* O'Reilly Media, Inc., Sebastopol, CA, 2012.
9. Martin, J. *Applications Development Without Programmers*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
10. Ronen, B. The complete kit concept. *International Journal of Production Research 30*, 10 (1992), 2457–2466.
11. Ronen, B. and Pass, S. *Focused Operations Management*. John Wiley & Sons, Inc., Hoboken, NJ, 2008.
12. Schwaber, K. *Agile Project Management with Scrum*. Microsoft Press, Redmond, WA, 2004.

Shimeon Pass（shimeon@passmgmt.com）是以色列特拉维夫市聚焦管理有限公司（Focused Management Ltd.）的高级咨询师和实施协调员。

Boaz Ronen（boazr@post.tau.ac.il）是创新性价值创造的 Professor Simon I. Domberger Chair（Professor Simon I. Domberger Chair for Innovative Value Creation），担任以色列特拉维夫大学 Recanati 商学院技术管理和价值创造方面的教授，兼任以色列特拉维夫市聚焦管理有限公司（Focused Management Ltd.）高级咨询师。

译文责任编辑：崔斌

DOI:10.1145/2594413.2594424

利用机器学习预测算法运行时间

作者：KEVIN LEYTON-BROWN，HOLGER H. HOOS，FRANK HUTTER，LIN XU

理解*NP*完全问题的实证难度

如果“问题能够解决，但是解决方案不够快，不能用”，此类问题就是难以驾驭的问题。[13]人们通常说，*NP*完全问题难以驾驭；然而，现实情况更为复杂。对于求解*NP*完全问题的所有已知算法，在最坏情况下均需要指数时间；然而，这些算法对很多具有重要现实意义的问题，速度反而快的惊人，所以在范围广泛的应用中，这些算法备受依赖。命题逻辑可满足性问题（SAT）就是这方面的佳例。用于进行硬件和软件形式化验证的方法有多种，其中使用程度最高的方法之一便依赖于通用SAT求解器和SAT编码，其中变量往往多达数十万个。此类实例通常可以在几秒内解决，不过，相同的求解器可能会被手动创建的实例弄瘫，而那些实例仅包含数百个变量。

很明显，与仅仅关注渐进意义的最坏情况的分析相比，在对算法行为进行更深入细致的探索后，我们可以从中获益。在研究中，我们提出了与终端用户的关系最为密切的问题：*“倘若使用可获取的最佳方法，解决给定类型的问题实例有多难？”*如果需要对该问题进行形式化的复杂度理论分析，任务似乎无法完成：可获取的最佳算法极为复杂（而且有时候仅以编译后的形式存在）。在实际应用的各种实例分布中，具有代表性的分布是异质的，且结构丰富。出于上述原因，我们放弃了组合分析，转向了统计分析。

本文的主要观点是，可用严谨的统计方法刻画算法运行时间的特征，且置信度高。具体来说，本文调研了十年来关于如何构建*实证难解性模型*（*EHMs*）的研究工作，阐释了构建该模型的方法：给定一个新的问题实例后，估算用低阶多项式时间表示的算法运行时间。[14,16,18,21,26–29,32,33,39,40]我们的工作表明，可以构建相当精确的模型，对不同的*NP*完全问题求解（我们研究了SAT、组合拍卖竞胜标确定问题、混合整数规划以及旅行商问题），研究问题实例的分布（我们探讨了几十个）和求解器（也是几

» 重要见解

- **严谨的统计方法可用于刻画算法运行时间的特征，且置信度高。**
- **可以分析由此得出的模型，进而洞悉造成问题难度高的原因，它甚至能在理论分析无法取得进展的场景中取得洞见。**
- **准确预测运行时间有很多的实际应用，比如从算法组合中选择最佳算法，让基准分布更难，以及找出优化算法性能的参数设置。**

十个）。我们已经发现了相当可靠的结果：即使非常简洁的EHM也能达到高准确度，也就是说，它们描绘了实例特性与算法运行时间之间的简单关系。[a] 这使得我们的方法甚至对那些偏向理论、觉得证明胜过实验结果的计算机科学家也有意义。EHM能揭示实例特性与运行时间之间存在的新的简单关系，从而推动新的理论工作。

本文重点分析了EHM如何帮助我们增进对*NP*完全问题的*理解*；然而，在各种类型的实际应用中，它们也能发挥作用。最直接的，它们有助于在集群中分配问题实例，或预测完成运行所需的时间。更有趣的是，还可以用它们把一组差异巨大的算法聚在一起形成性能超过各单独算法的“算法组合”；还能利用它们自动构建更具挑战性的基准分布；不仅如此，它们还能辅助专业人员配置（或“调优”）高度参数化的算法，以便在给定实例分布上取得良好的性能。在本文的侧边栏中，我们详细阐述了这些应用。

均匀随机3-SAT中的相变

我们从最广为人知的存在于固定规模的随机SAT实例的特性与求解器运行时间之间的关系开始。（后面我们会探讨更接近现实情况的SAT实例和其他的*NP*难问题。）用$p(c, v)$表示通过均匀随机采样c个子句生成一个可满足3-SAT公式[b]的概率，其中每一个子句从v个变量的集合中（均匀）选取三个变量，并对每个变量以0.5的概率取反在二十世纪九十年代初，研究人员发现，如果让v保持恒定，则在c/v越过4.26附近的临界值后，$p(c, v)$会出现“相变”。[8,31]从直觉上来说，包含极少子句的实例受到的约束过少，因此几乎总是可满足的，而包含很多子句的实例则约束过多，几乎总是不可满足的。一个有趣的事实是，对于到目前为止我们测试的所有定值v，$p(c, v)$正好等于0.5时的相变点，看起来正好与SAT求解器的运行时间峰值一致，甚至对于在此类实例上性能最佳的SAT求解器而言，情况也不例外。因此该发现把实例中独立于算法的特征(c/v)与依赖于具体算法的运行时间关联了起来，且已经证明，这种联系在不同的求解器中是鲁棒的。

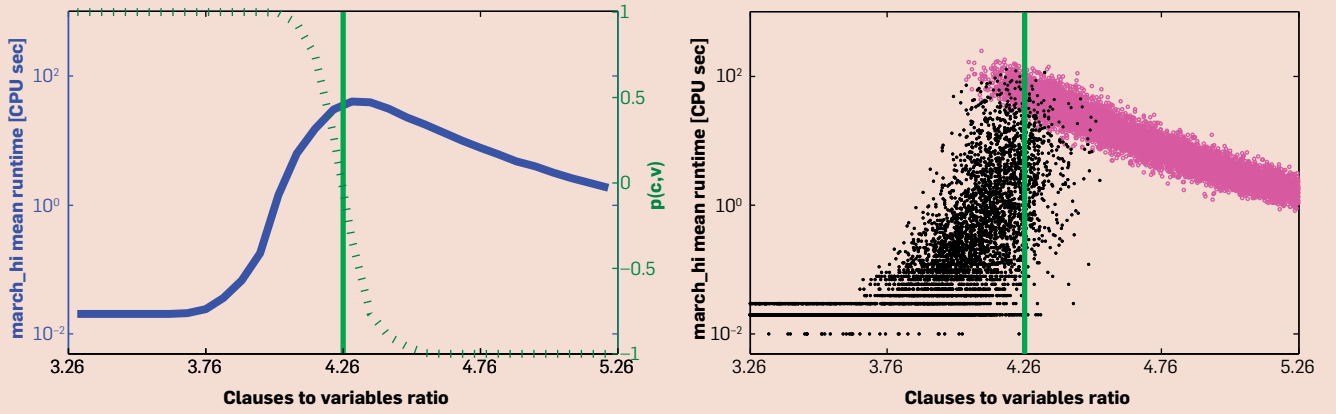
图1（左侧）使用实际数据描述了这种关系。虚线指用于均匀随机3-SAT实例（其中$v = 400$）的$p(c, v)$，而实线则指用于求解同一实例的`march_hi`的平均运行时间[11]。`march_hi`是用于均匀随机3-SAT实例的最佳SAT求解器。我们确实观察到了相变，还发现了相变点处出现的难度峰值。不过，情况要复杂得多。图1（右）绘制了`march_hi`的运行时间的原始数据（在对数标尺上），其中每个点对应一个单一的（随机）3-SAT公式。现在我们可以看出，c/v比不足以完全解释`march_hi`在这些实例上的实证行为：沿x轴的各点仍然存在显著的差别——在“难”相变点处，变化高于两个数量级。运行时间的模式还取决于可满足性的状态：难实例较罕见，而且与不可满足的实例相比，在可满足的实例之间，运行

a 本文中，我们未撰写算法性能预测的文献综述，而把重点放在了自己的工作上。对于相关工作的详细探讨，参加Hutter等人[21]和Leyton-Brown等人的著作。[29]

b 求解一个SAT公式F是指判定是否存在一组变量的赋值，使得F的值为真。重要性尤为突出的子类为3-SAT。3-SAT实例为*子句*的合取，其中每个子句为三个变量或其取反的析取。例如，$(v_1 \vee \neg v_2 \vee v_4) \wedge (\neg v_1 \vee \neg v_3 \vee v_4)$是一个简单的公式，其中$v = 4$个变量以及$c = 2$个子句。该公式有多个可满足的赋值（比如，$[v_1, v_2, v_3, v_4] = [\text{true, true, false, false}]$）。

插图由EUGENE KOSTSOV提供

图 1 march_hi 在均匀随机3-SAT实例上的运行时间，其中v =400，c/v 比可变。左：平均运行时间以及$p(c, v)$；右：每个实例的运行时间，其中用颜色标出了可满足性的状态。使用0.01s的准确度对运行时间进行了度量，得出了接近图形底部的、明显可见的离散化效果。每个点代表一个SAT实例。



时间的方差更大。上述现象的原因之一是，对于可满足的实例，求解器可在发现可满足的赋值后立即停止，而对于不可满足的实例，求解器必须证明在搜索树中不存在任何可满足的赋值。

均匀随机3-SAT的案例分析

现在我们来探讨，如果考虑的实例特征不仅限于c/v，我们是否能够更好的理解实例结构和求解器运行时间的关系。然后，我们再利用机器学习技术来推断这些特征与运行时间之间的关系。言归正传，首先我们选择实例集合I，对于每个 $i \in I$，获取特征向量x_i，然后对于每个$i \in$ I，通过在i上运行给定的算法获取运行时间的观测值y_i。我们的目标是确定一个映射$f : x \to y$，在给定 x_i时，可以尽可能准确地预测y_i。我们把这样一个映射称为EHM。[c]注意，我们刚刚描述了一个有监督学习问题，更确切的来说，一个回归问题。求解该问题时可以使用很多不同的回归算法，而且我们这些年也确实探讨了约十多种备选算法。在下文中，我们提倡采用相对复杂的学习范式（回归树的随机森林），但是开始的时候，我们会讨论一个非常简单的方法：二次岭回归。[5] 这种方法基于给定的特征及其成对乘积进行线性回归，并对特征系数（“岭”）的增加进行惩罚。我们从两个方面细化了这个方法。首先，我们把响应变量转换成其（以10为底的）对数；这样便于利用线性模型对其运行时间进行描述，因为运行时间的变化程度达到了几个数量级。然后，我们通过执行前向选择缩小特征集：从空集开始，然后迭代加入对改进预测贡献最大（从狭窄的范围选取）的特征。结果我们得到了更简单、更鲁棒的模型。该模型也更不容易出现数值问题。总体来说，我们发现，即使使用像上面这样的简单学习算法，人们往往也能构建出强大的EHM；而且确定一个好的实例特征集更重要。

实例特征。实例难度与c/v之间存在强关联，但是确定与其关联程度一样强的其他特征可能困难重重。因此，我们提倡先纳入带有一点可预测性迹象的所有特征，然后依赖机器学习算法确认最有用的特征。我们唯一的要求是，这些特征必须能够在低阶多项式时间内计算完成；在一些应用中，我们还把自己的选择限制在那些计算时间以平方时间递增的或更快的特性。在SAT领域中，我们定义了138个特征，总结如下：

- 问题规模的度量指标c 和 v，加上我们觉得重要的非线性组合，例如c/v和$c/v - 4.26$；
- 实例的句法特征（接近霍恩子句的程度，正负文字的平衡等）；
- 约束图的统计特征。我们考虑了三种图：节点表示变量，边代表共享的约束（子句）；节点表示子句，边代表极性相反的共享变量；节点表示子句和变量，边则代表变量在给定子句中出现了。对于每种图，我们根据节点度、路径长度和集聚以及其他因素计算出了各种统计数据。
- 在对给定SAT实例采用线性规划松弛后，求出最优解的整数性指标-具体来说，就是该解与最近（可行或不可行）整数点的距离；
- 搜索树规模的高德纳估算值；[25] 以及
- 通过运行局部搜索和树搜索算法并标识长度受限的轨迹而计算出探查性特征，然后从这些探查器中抽取统计特性（例如，在局部搜索中，到达一个局部最小值前执行的步骤数量，或者在树搜索中单文字传播的数量）。

c 构建EHM预测各种运行时间的概率分布，而不是仅预测单一的运行时间有时候也有用；参见Hutter 等[21]。为简化起见，本文仅讨论了对平均运行时间的预测。

模型的性能

我们可以利用上述技术为均匀随机3-SAT构建多种模型。现在让我们来探讨下这些模型。我们考虑了两个实例集：c/v的比值在相变点附近变化的实例集，和c/v的比值固定为$c/v = 4.26$的实例集。第一个实例集的难度较小，因为我们已经知道c/v的比值足以解释很多运行时间的变化情况。不过，这个实例集在进行合理性检查时仍然有用，它可确保我们的方法发现了c/v特征的重要性，也可用于研究在增加的特征中最后哪些有用。第二个实例集包含了位于$c/v = 4.26$的难区域内的实例，其规模是固定的；因为我们不能依据c/v比区分实例，所以我们在此发现的任何模式均能诱发兴趣。在这两种情况下（如同我们已经阐释的所有其他实证结果一样），我们随机地把数据划分成用于构建EHM的“训练集”和仅用于评估EHM性能的不相交的“测试集”，从而确保可利用在构建模型时未使用的数据评估模型预测的准确度。

图2描述了我们的调查结果，如果预测的`march_hi`运行时间正确，则为真。图中每个点对应不同的测试集问题实例。总体来说，在两张图中，点聚集在对角线附近，也就是说预测的准确度尚可。（这些结果实际上比看起来要好，因为更多的点落在了较靠近对角线的区域。）注意，与可满足的实例相比，对不可满足的实例的预测准确度要高很多。均方根误差（RMSE）是衡量模型准确度的数值指标；两个模型的均方根误差分别达到了0.31 和0.56。如果某模型预测的运行时间与实际值之间一直差10倍，则该模型的均方根误差将达到1.0。

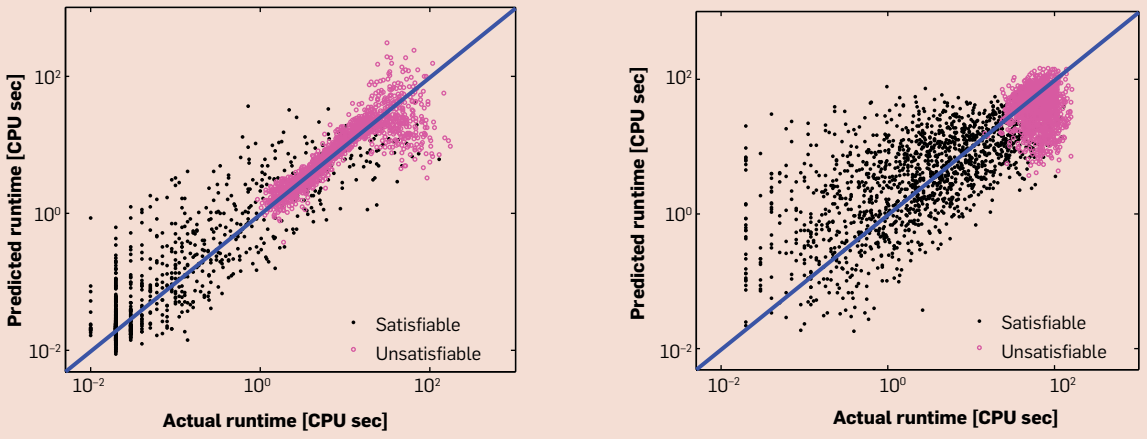
我们使用可变比值的实例集来验证模型是否识别出了c/v的重要性，与此类似，我们还想确定其他的有用特征。仅仅通过观察散点图，或是研究模型的系数本身并不能完成上述任务：因为我们的特征中，有很多是强相关的，重要的特征拥有的系数可能不大，不重要的特征拥有的系数反而可能大。与此相反，我们采用前文描述的前向选择方法确定了一些仅使用少量不相关特征的新模型，但是我们终止的时间要早得多，因为在添加新特征获得的好处开始逐渐减少时我们便终止运行。然后（按照Friedman[10]的方法），我们通过度量忽略各特征后均方根误差的损失来量化各特征的重要性，尔后我们按比例给这些值评分，其中最重要的特征评分为100。表1说明，该模型确实能确定c/v（以及其变形$|c/v - 4.26|$）为重要特征；这儿再提下，我们的二次回归可以利用各特征的所有成对乘积。其他的特征也很重要，特别是`SAPS_BestSolution_Mean`（通过局部搜索程序SAPS的短时间运行而获得的可满足子句的平均数量）。[20]。这点很耐人寻味，因为人们可能想像不到局部搜索算法的性能会揭示树搜索算法的性能。

对于固定比值的数据，c/v为常量，因此我们可以看到一个包罗重要特征的不同画面（表2）。与上文类似，局部搜索的探查性特征的数值非常突出（`GSAT_BestSolution_Mean`给出了通过GSAT的短时间运行而获得的可满足子句的平均数量[36]）。另一个重要的特征是`CG_entropy`。其给出了“约束图”中节点度的熵。在该图中，节点对应子句，边表示子句对共享了一个或多个符号相反的变量。对于其他的SAT求解器（比如`kcnfs` 和`satz`），我们重复了这种分析，并

表1 特征的重要性，$c/v \in [3.26,5.26]$

特征	分数
$\|c/v - 4.26\|$ × SAPS_BestSolution_Mean	100
c/v × SAPS_BestSolution_Mean	19
GSAT_BestSolution_CoeffVar × SAPS_BestStep_CoeffVar	19
SAPS_BestStep_CoeffVar × CG_entropy	18

图 2 `march_hi`在均匀随机3-SAT上的运行时间的实际值和预测值的对比每个点代表未在训练模型时使用的测试实例；完美的预测值会落在对角线附近。左：$c/v \in [3.26, 5.26]$；右：$c/v = 4.26$。



取得了相同的定性结果：运行时间可以预测，且准确度高；小模型也足以得到高的性能；局部搜索和约束图的特征相当重要。[33]

同时，我们已经观察到，在算法运行时间的分布方面，可满足的实例和不可满足的实例呈现出巨大的差异。因此，我们考虑了构建仅用于可满足的实例的EHM时，或构建仅用于不可满足的实例的EHM时出现的问题。（我们称之为“条件模型”，因为他们取决于我们已经知道了给定实例的可满足性。）我们发现，条件EHM比无条件EHM的准确度高；更有趣的是，最后发现*单特征* 条件模型足以以极高的正确率预测运行时间。对于可满足的实例，该特征为GSAT _ Best-Solution _ Mean，而对于不可满足的实例，该特征为Knuth _ Mean。在下文中，我们解释了引发上述发现的原因：因为局部搜索算法采用启发式的方法尽快求解，所以依据此类算法实现迅速求解的可靠度，可以推断出完整的算法成功求解时的速度。树搜索算法必须剔除树中每一个节点来证明不可满足性；因此，对于树大小的估计值便成为在不可满足的情况下最重要的特征。

表2特征的重要性，其中$c/v = 4.26$

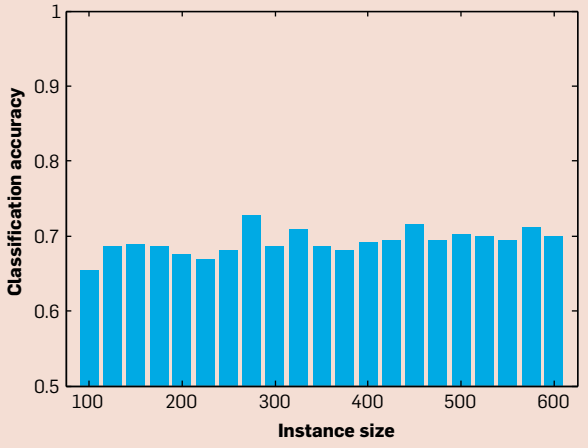
特征	分数
GSAT_BestSolution_Mean[2]	100
GSAT_BestSolution_Mean	88
SAPS_BestSolution_CoeffVar × SAPS_AvgImprove_Mean	33
SAPS_BestStep_CoeffVar × CG_entropy	22

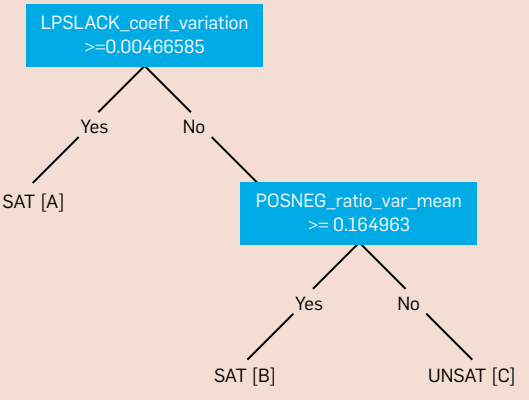
预测可满足性的状态

这些观察使我们产生了一个新的想法：构建一个直接预测可满足性状态的分类器，然后根据预测结果反过来影响条件EHM。不过，因为下面的两个原因，我们对这种方法抱有怀疑。首先，如果实例的“可满足性”状态出现判断错误，条件EHM可能会做出极不准确的预测，因此尚不清楚我们能否获得更佳的整体准确度。其次，也是更为根本的原因，就是我们准确预测可满足性状态的能力似乎值得怀疑—这相当于在实际求解前猜测一个实例是否可解！尽管有这种保留意见，为了减少预测失误带来的潜在成本（参见 Xu 等[38]），我们应用了复杂的统计技术，并成功构建了用于均匀随机3-SAT实例的*分层难度模型*。这些模型在固定比值和可变比值的实例集上均实现了（较为一般的）预测准确度的改进。很明显，只有我们的分类器能够准确预测可满足性状态时，分层难度模型的性能才能超过通常的EHM。在可变比值的情况下，一个自然的基准是，分类器对$c/v < 4.26$的实例预测可满足，对其它实例预测不可满足此分类器达到了96%的准确度，突出了c/v 特征的预测能力。我们的分类器把准确度进一步提升到了98%（把错误率降低了一半）。与基线不同，我们的分类器还适用于固定比值的情况。在固定比值的情况中，我们的分类器达到了86%的准确度。

我们发现这个结果相当意外，足以推动我们更深入地对其进行研究。[39]我们分析了规模不同的各种实例，其中的变量数量从100个（求解时间在毫秒数量级）到600（求解时间在一天之内；约为我们实际可解的最大实例）不等。我们把重点放在相变点附近生成的实例上，因为其构成了最难预测的问题：在该点处生成可满足的实例的概率为50%；实际上，在可满足的实例和不可满足的实例之间，我们的数据集确实分布得非常均匀。我们的目标是调查预测准确度是否会受到对较大问题的随机猜测的影响；如果不会受到影响的话，则构建一个容易理解的模型作为理论分析的出发点。为此，我们用三种方式限制了我们的模型，其中每种方式均降低了准确度。不过，这能让

图 3左：我们使用的简单决策树在相变点处对均匀随机3-SAT实例的分类准确度，其中变量的数量各不相同。该树仅用了包含100个变量的数据进行训练。右：决策树。对落在三个区域内的实例的预测准确度为：60%到70%之间（A区）；约50%（B区）；70%到80%之间（C区）。





我们更好地回答这些问题。首先，我们允许通过只包含100个变量的实例训练我们的分类器。其次，我们把探讨范围限制在最多只有两个决策节点的决策树上[7]。（我们通过标准的决策树学习程序来获得这些模型：采用贪心法选择一个在把训练数据划分成可满足的实例与不可满足的实例方面表现最佳的特征，然后再采用递归的方法多次划分由此得来的分区。）最后，我们忽略了所有的探查性特征。虽然探查性（例如局部搜索）特征在预测时非常有用，但是他们在小实例上的效果却不成比例，因此它们可能会让我们对尺度行为的研究复杂化。不仅如此，因为我们的目标是获得容易理解的模型，我们也倾向于避免使用那些基于复杂启发式算法的特征。

考虑了所有上述限制后，我们惊讶地观察到预测的准确度一直保持在65%以上，而且明显没有受到问题规模的影响（见图3左；在统计检验中，无证据表明准确度会因问题规模而下降）。实际上，我们的前两个限制对总体的影响似乎较小，准确度仅降低了约5%。（此外，解除这些限制后，我们仍然发现，没有证据表明准确度会受问题规模的影响。）在此，读者可能会有兴趣详细了解我们的双特征模型；我们希望该模型能成为我们新的理论分析的出发点，让我们深入研究位于相变处的有限规模SAT实例。图3（右）描绘了该模型。LPSLACK _ coeff _ variation的数值基于对SAT实例的整数规划表示的线性规划松弛的求解。对每一个变量i，其在LP中对应的解为$S_i \in [0, 1]$，LPSLACK$_i$定义为$\min\{1 - S_i, S_i\}$：S_i与整数性的偏差。LPSLACK _ coeff _ variation是向量LPSLACK的偏差系数（标准差除以均值）。向量POSNEG _ ratio _ var _ mean的第i个分量指各变量正负值出现次数的平均比值。对每一个变量i，设正值出现P_i次，负值出现N_i次，POSNEG _ ratio _ var$_i$定义为$|0.5 - P_i / (P_i + N_i)|$。POSNEG _ ratio _ var _ mean为向量POSNEG _ ratio _ var中元素的平均值。最后，再次提下，我们的模型是通过恒定规模的实例进行训练的；我们对LPSLACK _ coeff _ variation和POSNEG _ ratio _ var _ mean特征进行了归一化处理，使其在该训练集上的均值为0，标准差为1。为了评估模型在不同规模的给定实例上的效果，我们随机抽样了与该实例规模相同的很多新实例，通过这种方式计算新的归一化因子，然后再把这些因子应用到给定实例上。

超越均匀随机3-SAT

我们最初的动机中涉及对从业者面临的实际问题的研究，这些问题拥有均匀随机结构的可能性很小。因此，论证EHM能可靠地用于大范围的、更接近现实的实例分布是相当重要的一点。另外，还要论证它们不仅限于SAT。简而言之，就是它们不仅能行，而且能做更多。迄今为止，我们已经为四类*NP*完全问题构建了EHM。SAT[14,16,21,33,38,40]组合拍卖中确定竞标获胜者的问题（WDP）、[28,29]混合整数规划（MIP，对于拥有离散和连续变量的问题来说，这是标准的编码方式）、[14,19,21]以及旅行商问题（TSP）。[21]可以观察到，我们已经考虑了优化和决策这两类问题，这两类问题中涉及了离散变量、连续变量以及两种变量都存在的情况。对于每种问题，我们都衍生了一个新的实例特征集。此事虽然是非平凡的，但也不是难于登天；在所有的情况下，我们都使用了问题规模的度量、句法特征和探查性特征。现在，把我们的知识拓展到一个新的领域可能只需要几天的工作。在我们发表的文章和其他技术工作（例如，提交给SAT竞赛的作品）中，我们已经考虑了30多种实例分布。其中包括了复杂的随机生成器（例如，从图的着色和因子分解入

应用1算法选择（SATzilla）

现在不存在“最好的”SAT求解器；不同的求解器在不同性质的实例上表现各异，但是它们之间的性能差异往往非常大。基于EHM的有效性，本文指出了一个求解*算法选择问题*的简单方案：[35]给定一个新的问题实例后，预测几个SAT求解器的运行时间，然后运行预测结果中最快的求解器。这种方法[27]成为了SATzilla[32,33,40]的核心。SATzilla一种基于组合的SAT算法选择器。

SATzilla于2003年首次参加了SAT竞赛（http://www.satcompetition.org），在多个类别中取得了第二和第三的成绩。自那以后，我们已经大大改进了该方法，允许使用随机化的和局部搜索的算法作为组件求解器；引入了预解器（presolver）的概念，在运行选中的求解器前，短暂地运行预解器一段固定的时间；增加了为复杂评分函数进行优化的新功能；以及自动为选择器构建给定数据（比如，预解器选择；组件求解器选择）。我们对SATzilla所利用的组件求解器进行了持续改进。在利用上述改进，并发挥持续改进所带来的优势后，SATzilla引领了2007年和2009年的SAT竞赛，每次均获得了五枚奖牌。

最近，我们演进了SATzilla的设计。原来它依据对运行时间的预测（EHM）进行选择。经过演进后，它采用了代价敏感的分类方法，可直接选出性能最佳的求解器，而不需要预测运行时间。[41]在2012年SAT挑战赛中（http://baldur.iti.kit.edu/SAT-Challenge-2012），SATzilla获准进入四个类别；在其中三个类别中获得第一名，在第四个类别中获得第二名。总的来说，SATzilla的成功说明，在组合现有求解器（包括那些因平均性能不佳而“无竞争力”的求解器）方面，自动统计方法卓有成效。如果不考虑我们模型所使用的实例特征，我们的方法是完全通用的。对于那些运行时间的方差相当大的其他问题而言，本方法的效果或许也相当好。我们所有的软件均公开提供；请访问http://www.cs.ubc.ca/labs/beta/Projects/SATzilla。

手进行SAT简化；基于经济模型的组合拍卖基准）；来自公共基准和竞赛的实例集（例如MIPLIB；SAT竞赛）；以及从实际应用中衍生的实例集（例如，从软件验证和有界模型检测中获取的SAT编码实例；从机器工作分配到野生生物资源保护规划等各领域中的工业MIP实例；用于描绘电路板的钻孔路线以及实际城市之间的旅行路线的TSP实例）。我们还研究了50多个最先进的求解器，其中包括开源项目和业内开发的专有工具。我们的求解器既有确定性的，也有随机的；既包括完全的（也就是说，如果存在解，保证能求出解），也包括非完全的。在很多情况下，我们只能获得求解器的可执行文件，而且我们也从未利用过我们了解到的求解器的内部工作机制。正如前文所述，我们的研究已经超出了二次基函数回归的范围，分析了十多种其他的统计建模方法，包括Lasso回归、多元自适应回归样条、支持向量机回归、神经网络、高斯过程、回归树和随机森林（参见Leyton-Brown等[29]和Hutter等[21]）。此处我们不再描述细节，而是陈述结论：我们倾向于回归树的随机森林方法，[6]特别是在实例分布呈现异质化时，情况更为明显。为了让读者有个完整的了解，我们简要地描述下这个模型类别，但是建议读者阅读相关文献获取详细信息。[6,7]。回归树很像决策树（本文中我们使用决策树预测可满足性的状态）。然而，作为一个分类方法，决策树把类别的标签与每片树叶关联（例如，“可满足的”，“不可满足的”），而回归树把实数预测值与每片树叶关联。随机森林反映了利用各回归树集合做出的预测值的均值；我们随机化了训练过程，使得构建出的这些树均有所差异。

图4描述了我们拓展EHM应用领域后取得的结果，其中突出了三个不同的求解器，每个求解器来自不同的领域。针对每一种情况，我们均绘制了二次岭回归和随机森林的图，用于说明学习算法的效果。首先（第1列），我们考虑了著名的SAT求解器Minisat 2.09。该求解器在国际SAT竞赛提供的极为异质化的实例混合体上表现卓著。虽然在竞赛中实例被归为几类（“工业/应用”、“人工创建/精心设置”以及“随机”），但是我们把所有这些实例都放在一起了。可能是因为所获实例集的异质性较大，此次二次回归的效果相对较差。随机森林提供了更为可靠的估计值；值得注意的事，它们可以把特征控件分成性质不同的部分，而且它们预测的运行时间从未大于或小于在训练数据中观察到的极值。然而，请注意，即使是准确度稍差的二次回归模型，在把此领域内算法运行分为快和慢两部分方面，它们往往

应用 2 生成难基准

逼真的难基准分布相当重要，因为它们是衡量算法开发是否成功的一个客观指标。然而，在有些情况下，发现新的难基准的困难程度不亚于找出一种新策略来求解之前的难基准。为了填补这一空缺，我们使用了EHM来自动调整现有的实例生成器，使其能够生成用于给定算法集的更难的实例。[26,29]

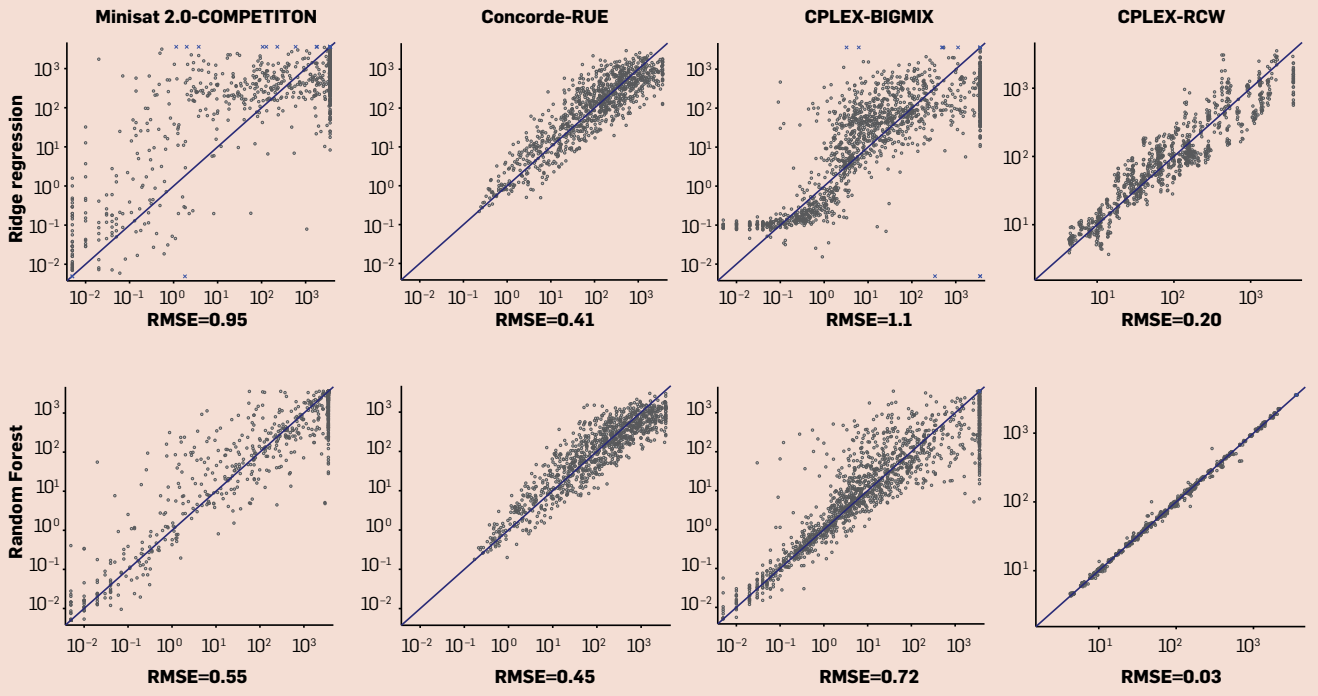
我们从拥有参数p的实例生成器开始。使用此类生成器时，往往简单地使用了默认的参数设置；然而，为了搜索更难的实例，我们用了与此相反的办法，从一个固定范围的区间中均匀抽取了各参数的值。让我们把由此得出的实例分布称为$\mathcal{D}$。我们的目标是对相同实例的新分布$\mathcal{D}'$进行采样，然后依据给定算法A在各实例上的难度评出实例的权重。（设想一下，对于给定集合中的所有算法，A是其中平均运行时间最短的算法，或者A是此类算法中符合SATzilla风格的选择器。我们可以通过重要性抽样的形式实现上述功能。我们可以利用我们的标准实例特征f来为$\mathcal{D}$上的A构建EHM；对于某个实例x，我们把该模型的预测称之为$H_f(x)$。我们希望从D中生成相当大的实例集，依照$H_f(x)$按比例定出各实例x的权重，然后按照权重的比例从该集合中抽出一个单一的实例。这种方法有用，但是，如果$\mathcal{D}$中的难实例非常少，那就需要非常大的样本数量。为了提高性能，我们学习出一个二次EHMH_p，其仅使用生成器参数p作为特征。然后，我们便能依照$\mathcal{D}(x)\cdot H_p(x)$按比例抽取实例$x$的样本，而不是从$\mathcal{D}$中采样（通过直接对多项式函数$H_p$进行采样，尔后使用由此得出的参数运行实例生成器），然后再通过$H_f(x)/H_p(x)$求出各样本实例x的权重。因此，H_p可以把我们的搜索引向难度更大的实例，而不会使权重发生偏移。在利用组合拍卖测试集[30]开展的实验中，此方法让所生成实例的平均难度提高了多达100倍[26,29]，而且与之前使用生成器默认参数而观察到的难度相比，新创造的实例的难度往往要高得多。

应用3算法配置

设想一下，启发式算法的设计师每次面对给定设计元素，并做出选择时，她只需把该元素作为单一求解器的一个自由参数。最后，在为给定问题域设计好算法时，她所面临的问题可简化为一个寻找配置的随机优化问题，找出一种可达到高性能的配置。[12,14,24]

我们已经应用了自动化的方法来求解该问题，以确定新的算法配置，其在广泛的领域（包括基于SAT的形式验证、[15] MIP求解[17]以及自动规划）内已经实现了多个数量级的加速。[37]我们用于求解该问题的最先进的方法基于EHM：使用*基于模型的序列算法配置*(SMAC)[19]在下列步骤中进行循环迭代：使用EHM选出有希望的配置进行进一步研究，尔后利用这些配置执行算法，再利用由此得来的信息更新模型。SMAC是免费提供的；请访问http://aclib.net/SMAC。EHM也可用于根据每个实例的具体情况选择合适的配置。[16]

图 4 用图形化的方法比较各模型在未见实例上所预测的运行时间在每张图中，*x*轴指实际运行时间，*y*轴指各模型预测的运行时间。其中用蓝色的*x*标出了大于3,000 或小于0.001的预测值。



也足够准确；见边栏内对SATzilla的阐述。其次（第2列），一组使用广泛、同质化程度高且随机生成的TSP实例，我们研究了业界领先的完全TSP求解器Concorde的性能。[2,23]我们再次看到了高性能，而且这次二次回归和随机森林均表现优异。第三（第3列和第4列），为了说明只改变实例分布后的效果，我们考虑了一个求解器在两种不同分布下的表现。IBM ILOG CPLEX[22]是应用最为广泛的商用MIP求解器。BIGMIX 是公开提供的，用于混合整数规划问题的混合集，该混合集的异质化程度高。正如在我们第一个基准中对SAT实例混合集的预测情况一样，碰到某些类型的实例时，线性回归进行预测的难度不小，且有时候会出现非常严重的预测错误。同样，随机森林的性能要可靠地多。RCW 对红顶啄木鸟的扩散和栖息地的建立进行了建模，使用在拟保护的地块方面的决策作为条件。[1]可以预测出CPLEX在该领域的运行时间，这有点出人意料；随机森林给出了我们观察到的最佳的EHM性能。

超越单一的算法

不同于平均复杂度的结果是对计算问题的固有复杂度进行刻画，EHM总是描述给定算法的性能。从某种程度上来说，这是一种固有的局限性：对于尚未发明的算法，统计方法无法概述其性能。不过，仍有一种有用的方法可让我们放松对单一算法的限制：我们可以构建一个模型来描述现有算法的*空间*。

具体来讲，大多数用于难组合问题的最先进的算法提供了一系列的算法参数，以便让用户自定义或调优算法的行为。我们对“参数”的定义非常宽泛，其包括传给求解器，然后改变求解器行为（及其运行时间）的任何参量，但是不包括求解器返回的解的本质。因此，参数可以是连续值，分类值，序数或布尔值，甚至可能以其他参数的取值作为条件。更重要的是，分类参数和布尔参数可用于表示非常抽象的决策- 有效地从不相关的代码块中进行选择 - 从而打开巨大的算法设计空间。例如，IBM ILOG CPLEX 提供了76个参数（45个分类参数、6个布尔参数、18个整型参数以及7个实值参数）；[17]对这些参数进行相当粗略的离散化处理后，产生了超过10^{47}种不同的算法初始化方式，每种初始化的性能情况截然不同。我们把利用特定值初始化给定算法的所有参数这种行为称为*配置*。本文中，参数化求解器的第二个例子是SAT求解器SPEAR[4]，其提供了26个参数（七个分类参数、三个布尔参数、四个整型参数以及12个实值参数），造成了超过10^{17}种不同的算法初始化方式。

现在我们来探讨如何归纳EHM方面的知识，用其描述参数化的算法。原则上来说，变化不大：我们探讨了把配置、实例组成联合空间映射到运行时间预测值的模型。问题是这种方法的效果有多好。在我们给出回答前，我们需要确定方法的评价标准。我们可以采用与训练EHM时相同的配置进行检测。不过，检测是在新的问题实例上，或者在采用新的配置但在之前未见的实例上，或者采用之前未见的配置和实例的组合上进行。

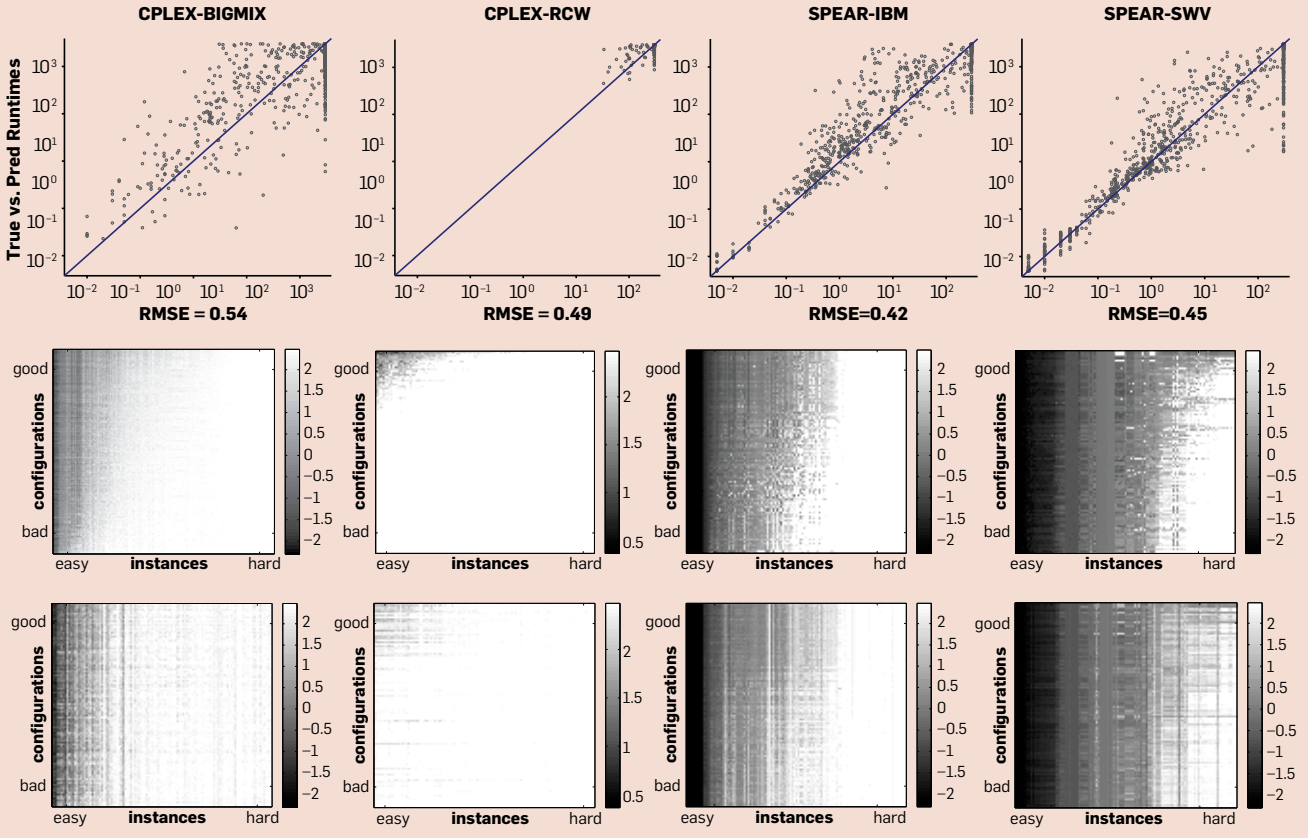
第三中情况最难；这也是在本文中描述结果的唯一的一种情况。图5说明了该设置中的一些代表性结果，其重点放在了随机森林模型上。第一行说明了与前文图形相似的散点图，其中每个点代表采用随机选择的、之前未见的配置在之前未见的实例上进行的一次运行。我们还提供了一种不同的方式来观察参数配置和实例特征向量的联合空间。第二行说明了利用每对（配置、实例）数据后获得的真实运行时间，其中按其平均性能对配置进行了排序，按其平均难度对实例进行了排序。因此，此图概略描述了采用各种实例和配置时的运行时间变化情况，也让我们有可能衡量仅因配置不同而造成的变化程度，并将其与仅因实例不同而造成的变化程度进行比较。最后，第三行说明了通过EHM获取的预测结果，其格式与第二行相同。这样的话，我们有办法用可视化的方法将模型的性能与真实数据（ground truth）进行比较；理想情况下，第二行和第三行看起来应该一模一样。（当两组数据彼此非常相似时，EHM的确可以作为原有算法的替代，也就是说，在对算法的性能进行实证分析时，可以用EHM代替算法；参见Hutter等人的著作[18]）。

我们的实验的主要结论为，即使对于训练中未曾检查过的算法配置，我们的模型也能够达到高准确度（均方根误差在0.5附近；第二行和第三行之间定性相似）。图5的第一列涉及在本文异质化MIP实例混合集上CPLEX运行时间的预测值。本基准中的实例难度差异巨大，比因CPLEX的配置不同而造成的性能差异要大得多（见第2行）。因此，实例的难度成为影响运行时间的主要因素，模型的重点放在（更多重要的）特征空间上，付出的代价是无法捕获因配置不同而造成的某些性能差异。其次，在RCW中，大多数（随机抽样的）CPLEX配置只能解数量极少的实例；我们把这些失败记录为非常长的运行时间。虽然如此，模型却能够确定哪些配置好，哪些实例

图 5 用图形化的方法比较各模型在之前未见测试配置和实例的组合上所预测的运行时间

第1行：在每张图中，*x*轴指实际运行时间，*y*轴指各模型预测的运行时间。每个点代表一种实例与参数配置的组合，其中实例和参数配置均未在之前训练模型时见过。

第2行和第3行：各应用领域内运行时间的实际值和预测值在热度图中，每个点代表在一个实例上运行一种参数配置；灰度值代表以$_{10}$为底的运行时间的对数值（较黑的点说明速度更快）。



容易。最后，我们探讨了如何预测SPEAR在两组形式验证实例集上的运行时间。IBM是一组有界模型检测实例集[42]，而SWV则是一组利用Calysto静态检查程序生成的软件验证实例集。[3]对于这两种实例的分布，不同配置下的SPEAR的运行时间均被准确预测，且准确度高。我们的随机森林模型准确地预测了各实例的实证难度（empirical hardness）和配置的实证性能，甚至捕获了两者之间相互的影响。

结论

统计方法可以刻画利用可获得的最佳算法对给定分布中的实例进行求解的难度 — 即使这些算法极为复杂，使用传统的理论分析行不通。在实践中，此类EHM的效果惊人，它可以用于不同的难组合问题、现实世界中的实例分布以及最先进的求解器。这些模型的分析结果可超越最坏情况，作为对复杂度进行新的理论研究的出发点，因为可通过它们确定哪些问题特征可以预测难度，或是足以直接预测目标函数（例如，可满足性的状态）。高度参数化的算法涵盖了相当大的可选算法设计空间。在这种背景下，我们发现EHM甚至可以预测之前未检测的算法设计在之前未见实例上的运行时间。本文已经证明EHM能在各种实际应用中发挥作用，包括算法组合的自动设计、难基准分布的自动合成以及在大算法设计空间中进行自动搜索，找出优化性能的设计。我们编写了开源软件来完成构建和分析EHM、构造算法组合、自动配置参数化的算法和其他功能：参见http://www.cs.ubc.ca/labs/beta/Projects/EPMs/。

鸣谢

文中描述的部分工作由更多的合作者完成的，其中包括：Eugene Nudelman 和Yoav Shoham 为本文做出了持久卓越的贡献，Galen Andrew、Alex Devkar以及Jim McFadden的工作也颇有亮点。

参考资料

1. Ahmadizadeh, K., Dilkina, B., Gomes, C.P. and Sabharwal, A. An empirical study of optimization for maximizing diffusion in networks.In *Proceedings for Principles and Practice of Constraint Programming* (2010), 514–521.
2. Applegate, D.L.Bixby, R.E., Chvátal, V. and Cook, W.J.*The Traveling Salesman Problem:A Computational Study.*Princeton University Press, 2006.
3. Babic´, D. and Hu, A.J.Structural abstraction of software verification conditions.In *Proceedings for Computer Aided Verification* (2007), 366–378.
4. Babic´, D. and Hutter, F. Spear theorem prover.Solver description.SAT 2007 Competition.
5. Bishop, C.M.*Pattern Recognition and Machine Learning.* Springer, 2006.
6. Breiman, L. Random forests.*Machine Learning 45*, 1 (2001), 5–32.
7. Breiman, L., Friedman, J.H., Olshen, R. and Stone, C.J.*Classification and Regression Trees.*Wadsworth, Belmont, CA, 1984.
8. Cheeseman, P., Kanefsky, B. and Taylor, W.M.Where the really hard problems are.In *Proceedings for International Joint Conference on Artificial Intelligence* (1991), 331–337.
9. Eén, N. and Sörensson, N. An extensible SAT-solver. *Theory and Applications of Satisfiability Testing* (2004), 502–518.
10. Friedman, J. Multivariate adaptive regression splines. *Annals of Statistics 19*, 1 (1991), 1–141.
11. Heule, M. and Maaren, M.v. march_hi.Solver description, SAT 2009 competition.
12. Hoos, H.H.Programming by optimization.*Commun. ACM 55*, 2 (Feb. 2012), 70–80.
13. Hopcroft, J.E., Motwani, R. and Ullman, J.D. *Introduction to Automata Theory, Languages, and Computation.*Pearson Education, 2007.
14. Hutter, F. Automated Configuration of Algorithms for Solving Hard Computational Problems. Ph.D. thesis, University Of British Columbia, Department of Computer Science, Vancouver, Canada (Oct. 2009).
15. Hutter, F., Babic´, D., Hoos, H.H. and Hu, A.J.Boosting verification by automatic tuning of decision procedures. In *Proceedings for Conference on Formal Methods in Computer-Aided Design* (2007), 27–34.
16. Hutter, F., Hamadi, Y., Hoos, H.H., and Leyton-Brown, K. Performance prediction and automated tuning of randomized and parametric algorithms.In *Proceedings for Principles and Practice of Constraint Programming* (2006), 213–228.
17. Hutter, F, Hoos, H.H. and Leyton-Brown, K. Automated configuration of mixed integer programming solvers.In *Proceedings for Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* (2010), 186–202.
18. Hutter, F, Hoos, H.H. and Leyton-Brown, K. Trade-offs in the empirical evaluation of competing algorithm designs.*Annals of Mathematics and Artificial Intelligence 60*, (2010), 65–89.
19. Hutter, F, Hoos, H.H. and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration.In *Proceedings for Learning and Intelligent Optimization Conference* (2011), 507–523.
20. Hutter, F, Tompkins, D.A.D. and Hoos, H.H.Scaling and probabilistic smoothing:Efficient dynamic local search for SAT.In *Proceedings for Principles and Practice of Constraint Programming* (2002), 233–248.
21. Hutter, F., Xu, L., Hoos, H.H. and Leyton-Brown, K. Algorithm runtime prediction:Methods and evaluation. *Artificial Intelligence J.206* (Jan. 2014), 77–111).
22. IBM.CPLEX Optimizer, 2014. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.
23. Johnson, D.S.Random TSP generators for the DIMACS TSP Challenge, 2011; http://dimacs.rutgers.edu/Challenges/TSP/.
24. KhudaBukhsh, A., Xu, L., Hoos, H.H. and Leyton-Brown, K. SATenstein:Automatically building local search SAT sol vers from components.In *Proceedings for International Joint Conference on Artificial Intelligence* (2009), 517–524.
25. Knuth, D. Estimating the efficiency of backtrack programs.*Mathematics of Computation 29*, 129 (1975), 121–136.
26. Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J. and Shoham.Boosting as a metaphor for algorithm design.In *Proceedings for Principles and Practice of Constraint Programming* (2003), 899–903.
27. Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J. and Shoham, Y. A portfolio approach to algorithm selection.In *Proceedings for International Joint Conference on Artificial Intelligence* (2003), 1542–1543.
28. Leyton-Brown, K., Nudelman, E. and Shoham, Y. Learning the empirical hardness of optimization problems:The case of combinatorial auctions.In *Proceedings for Principles and Practice of Constraint Programming* (2002), 556–572.
29. Leyton-Brown, K., Nudelman, E. and Shoham, Y. Empirical hardness models:Methodology and a case study on combinatorial auctions.*Journal of the ACM 56*, 4 (2009), 1–52.
30. Leyton-Brown, K., Pearson, M. and Shoham, Y. Towards a universal test suite for combinatorial auction algorithms.In *Proceedings for ACM Conference on Electronic Commerce* (2000), 66–76.
31. Mitchell, D., Selman, B. and Levesque, H. Hard and easy distributions of SAT problems.In *Proceedings for Conference on Artificial Intelligence* (1992), 459–465.
32. Nudelman, E., Leyton-Brown, K., Andrew, G., Gomes, C., McFadden, J., Selman, B. and Shoham, Y. Satzilla 0.9.Solver description.SAT Competition, 2003.
33. Nudelman, E. Leyton-Brown, K., Hoos, H.H., Devkar, A. and Shoham, Y. Understanding random SAT:Beyond the clauses-to-variables ratio.In *Proceedings for Principles and Practice of Constraint Programming* (2004), 438–452.
34. Prasad, M.R., Biere, A. and Gupta, A. A survey of recent advances in SAT-based formal verification. *International Journal on Software Tools for Technology Transfer 7*, 2 (2005), 156–173.
35. Rice, J.R.The algorithm selection problem.*Advances in Computers 15* (1976), 65–118.
36. Selman, B., Levesque, H.J. and Mitchell, D. A new method for solving hard satisfiability problems.In *Proceedings for Conference on Artificial Intelligence* (1992), 440–446.
37. Vallati, M., Fawcett, C., Gerevini, A.E., Hoos, H.H. and Saetti, A. Generating fast domain-optimized planners by automatically configuring a generic parameterised planner.In *Proceedings for Automated Planning and Scheduling Workshop on Planning and Learning* (2011), 21–27.
38. Xu, L., Hoos, H.H. and Leyton-Brown, K. Hierarchical hardness models for SAT.In *Proceedings for Principles and Practice of Constraint Programming* (2007), 696–711.
39. Xu, L., Hoos, H.H. and Leyton-Brown, K. Predicting satisfiability at the phase transition.In *Proceedings for Conference on Artificial Intelligence* (2012), 584–590.
40. Xu, L., Hutter, F., Hoos, H.H. and Leyton-Brown, K. SATzilla:Portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research 32* (June 2008), 565–606.
41. Xu, L., Hutter, F., Hoos, H.H. and Leyton-Brown, K. Evaluating component solver contributions to portfolio-based algorithm selectors.In *Proceedings for Theory and Applications of Satisfiability Testing* (2012, 228–241.
42. Zarpas, E. Benchmarking SAT solvers for bounded model checking.In *Proceedings for Theory and Applications of Satisfiability Testing* (2005), 340–354.

Kevin Leyton-Brown（kevinlb@cs.ubc.ca）是加拿大英属哥伦比亚大学计算机科学系副教授。

Holger H. Hoos (hoos@cs.ubc.ca)是加拿大英属哥伦比亚大学计算机科学系教授。

Frank Hutter (fh@cs.uni-freiburg.de)是德国弗赖堡大学计算机科学系艾米-努特（Emmy Noether）研究小组的负责人。

Lin Xu（xulin730@cs.ubc.ca）是加拿大英属哥伦比亚大学计算机科学系的在读博士生。

译文责任编辑：孙晓明

DOI:10.1145/2594446

技术视角
神经科学与密码学的交汇

作者：Ari Juels 和 Bonnie Wong

若要查阅相关论文，
请访问 doi.acm.org/10.1145/2594445 rh

有一个广袤深邃、规模未知的未开发资源，仍然隐匿在神秘面纱之后。科学家们正在探索、试图勾勒其面貌，但对于它的范围、它如何用于满足重要的人类需求，科学家只有一个初步的认识。以上文字似乎是在描述天然气或地热能源储备。但这段文字同样也适合描述人类大脑，尤其是涉及记忆和计算机安全领域时。

根据对人类大脑的最佳估算（Paul Reber，下列论文的合著者之一），其记忆容量大约为 2.5 PB（*拍字节*），即 2.5×10^{12} 字节，相当于数千块普通硬盘的容量之和。然而对于大多数人而言，要方便地记住并准确地想起包含 20 *比特*以上随机值的密码颇有难度，也就是说，这些密码的猜测难度大于 20 比特随机字符串。

相比之下，“7UquO91”这样的随机字母数字密码包含了 40 多比特的随机值（因为密码强度呈指数增长，其强度是包含 20 比特随机值的密码的约 100 万倍）。不可思议的是，当今计算机安全领域中的一大未解决难题，居然是如何在带宽有限、但容量足以容纳美国国会图书馆中所有书籍的存储设备中高效地读取和写入“7UquO91”这样微小的机密。

影响是巨大的。弱密码容易被破解，正如最近涉及数百万密码的著名失窃案例所示。遗忘密码时，网站会使用私人问题进行验证，如“您上的高中是什么？”，而这比密码本身更容易攻破。（只要问问前州长 Sarah Palin 就知道。）普通密码的另一问题是它们还能通过不当方式泄密。人们可能会受到人身胁迫或威胁而透露其密码，或者主动提供给不应使用它们的其他人员。

将密码存储在人类大脑中的理想方案是，在人不知不觉的前提下在其大脑中输入和输出超过 20 比特随机值的密码，因此他就无法向别人提供密码或在受到胁迫时透露。

下面这篇论文介绍了一种完全可以实现该目的的方法。它涉及一种颇有趣味且意想不到的机制：让用户玩一个视频游戏。玩游戏的人通过*内隐学习*获得比较强的密码，借助这一方式，信息可以通过练习存到长期记忆中，但不能被有意识地访问。正如此处所述 — 我们的文章中也有类似表述 — 这种方式目前还不实用，不能在登录电子邮件帐户等常见的身份验证任务中加以利用。玩游戏需要的时间太长（约 10 分钟）。但这不是这篇论文的意义或主要贡献所在。它提供的重要成果强调了神经科学和密码学之间尚未充分开发的丰富资源，从更广义的层面来说，则是神经科学和计算机安全之间的交汇贯通。

神经科学中一个令人振奋的前沿是利用接口直接读取和刺激神经活动。例如，脑电图学 (EEG) 能够以非侵入方式检测神经活动的模式。低成本 EEG 头戴式装置正在为消费级人机接口 (BCI) 的发展扫清障碍。有一部分甚至已向游戏玩家提供。此类接口或许能免去用户在响应刺激时键入操作的必要，也能加快以内隐记忆为基础的用户身份验证。未来某一天，更加先进的技术可能会提供精细的大脑实时功能图，可直接通过神经方式执行问答式身份验证协议，而无需用户的有意识行为。事实上，有证据表明，以刺激神经可塑性（即大脑适应性）为目标的技术可以增强许多形式的学习和记忆，其中或许也包含密码。经颅直流电刺激 (tDCS) 就是这样的一种技术，现在已应用于让游戏玩家感知“刺激”的低成本头戴式装置中。奥巴马政府近期宣布了雄心勃勃的“通过推动创新型神经技术开展大脑研究 (BRAIN)”计划，该计划有望能促进此类工具的发明。

神经科学中一个令人振奋的前沿是利用接口直接读取和刺激神经活动。

有关神经科学与计算机安全的相互作用还有许多未解答的问题。可否利用大脑的自然计算能力实现与智能卡或硬件身份验证令牌相当的效用？现有的内隐记忆可否通过精心制作的刺激而诱出？或许通过人机接口来实现？最后，可否直接从大脑读取用户的意图来检测和预防恶意活动？人机接口对于隐私意味着什么？

现在，请阅读一篇可激发此类问题的精彩论文，文中当然也提供了对一些问题的回答。 C

Ari Juels (ajuels@gmail.com) 是位于美国马萨诸塞州波士顿的一名独立研究员，主攻计算机安全研究。

Bonnie Wong (bonniewong38@gmail.com) 是位于美国马萨诸塞州波士顿的贝斯以色列女执事医疗中心 (Beth Israel Deaconess Medical Center) 的一名临床神经心理学家。

译文责任编辑：孙晓明

DOI:10.1145/2594445

神经科学与密码学交汇：通过密码原语抵御软磨硬泡式攻击

作者：Hristo Bojinov、Daniel Sanchez、Paul Reber、Dan Boneh 和 Patrick Lincoln

摘要

密码系统常常依赖于提供给用户的密钥的机密性。但在攻击者强制用户提供密钥时，许多方案都无法抵御胁迫式攻击。此类攻击称为*软磨硬泡式破译*，通常是攻破密码的最简单方式。本文介绍胁迫攻击的一种抵御方式，它利用来自认知心理学的*内隐学习*概念。内隐学习指的是一种在不知不觉中学习的方式。我们使用一款精心制作的计算机游戏，让用户通过内隐方式学习密码，而用户却对训练的密码没有任何明显或有意识的认知。虽然训练的密码可用于身份验证，但参与者无法被强迫提供此密码，因为他们根本不知道密码。我们利用 Amazon 的 Mechanical Turk 进行了一系列用户研究，确认了参与者在一段时间后仍可再次通过身份验证，但无法重新构建或明确识别训练的密码。

1. 引言

想象这样一个场景：某一安保措施严密的设施中运用了复杂的身份验证系统，只有知道密钥、持有硬件令牌并具备授权生物识别信息的人才能进入其中。门卫确保只有成功通过身份验证的人才可进入该设施。假设一个聪明的攻击者俘获到一名通过身份验证的用户。攻击者能够盗取该用户的硬件令牌，伪造其生物识别信息，借助橡胶管等武器迫使受害者泄露其密钥。此时，攻击者可以假冒受害者，攻破该设施中部署的造价昂贵的身份验证系统。

所谓的软磨硬泡式攻击一直是安全系统的天敌，也常常是攻破密码的最简单方式。[12] 原因在于，用户必须持有身份验证凭据才能通过身份验证，而这些凭据可以通过武力 [10] 或其他方式获取。

本文介绍一种可预防某类软磨硬泡式攻击的新方法，它利用来自认知心理学的*内隐学习* [2, 7] 概念。一般认为，内隐学习涉及大脑中称为*基底神经节*的部分，它支持通过重复执行任务来学习骑自行车或打高尔夫球等技能。以内隐学习为主要课题的实验表明，通过这种方式学习的知识无法被受训练的人有意识地访问。[7] 这种现象在日常生活中的一个例子是骑自行车：我们知道如何骑车，但无法解释我们是怎么做到的。第 2.1 节中提供了相关神经科学的更多背景。

内隐学习为设计出抗胁迫式的安全系统提供了一个令人向往的工具。在本文中，我们重点关注用户身份验证：利用内隐学习，训练人类大脑掌握可在身份验证期间检测、但无法被用户明确描述的密码。此系统避免了人们因受他人劝说而泄露密码的问题。要使用此系统，首先需要训练参与者执行一项称为串行拦截序列学习（Serial Interception Sequence Learning，SISL）的特定任务，如下一节中所述。训练时将使用一款主要依赖内隐学习的计算机游戏，训练结果是掌握一组充当身份验证密码的特定按键序列。在我们的实验中，训练活动为时大约 30 到 45 分钟，参与者将掌握一个拥有约 38 比特熵的随机密码。我们开展的实验表明，参与者在训练之后无法重新构建其受训的序列。

今后进行身份验证时，参与者将再次执行包含多个嵌入序列的 SISL 任务，其中包括之前已训练序列的元素。若在训练元素上的表现稳定优于未训练元素，参与者可在 5 到 6 分钟内验证其身份。不知道训练序列的攻击者无法呈现用户的表现特征（在训练结束时衡量）。请注意，身份验证过程是一个互动式游戏，服务器知道参与者的密码训练序列，并使用它来验证参与者身份。读者如果想要试玩该系统，可以在 brainauth.com/testdrive 上查看训练任务。

本文的最初版本刊登在 2012 年《第 21 届 USENIX 安全性专题研讨会论文集》* 中。

虽然本文关注的是抗胁迫用户身份验证系统，但身份验证仅仅是冰山一角。我们希望可以利用内隐学习设计许多其他的抗胁迫安全原语。

威胁模型。我们所提出的系统，被设计成一个本地密码机制，该机制需要用户本人在场。也就是说，我们考虑的是安全位置入口处的身份验证，门卫可以确保真人在不借助电子仪器的前提下参与测试。

为了欺骗身份验证测试，对手可以拦截一名或多名受过训练的用户，使他们（或许通过胁迫方式）尽可能泄露信息。然后，对手亲自参与到实时身份验证测试中，其目标是通过该测试。

我们要强调的是，该系统的设计和标准密码身份验证一样，不能抵御窃取式攻击，如在身份验证过程中偷窥密码输入。尽管问答式协议是标准的窃取预防手段，但要基于内隐学习来设计问答式协议，目前依然是尚待解决的问题。我们将在本文结尾部分重新讨论这一问题。

胜于生物识别身份验证的优点。训练的机密序列可以视为一种能够验证受训练参与者身份的生物识别密钥。不过，与生物识别密钥不同的是，这种身份验证信息不能被秘密复制，参与者即使愿意也无法泄露所训练的机密。此外，如果训练序列被攻破，可以训练新的身份识别序列作为替代，从而导致更换密码。

在相关的文章中，Denning 等人[1] 提出使用图像来训练用户以内隐方式记忆密码。这种方式可能无法抵御软磨硬泡式攻击，因为用户将记住哪些图像他们看到过，哪些则没有。此外，基于图像的方式还需要准备大量的图像，并且每个用户仅使用一次，这使得系统的部署难度更高。我们这种基于组合学的方法，可以让我们对被学习的密码的熵有一个下界，设置简单，也可经过设计，不留下任何训练序列的有意识踪迹。

用户研究。为了验证我们的方案，我们使用 Amazon 的 Mechanical Turk 进行了一系列用户研究。我们询问了以下核心问题，探索通过内隐学习进行身份验证的可行性：

- 个人身份识别是否可靠？即，被训练的用户能否重新进行身份验证，可否在过段时间后依然能够验证？
- 攻击者能否通过从受训练参与者那里轻松获取的表现数据来对该序列进行反向工程？

通过三个实验，我们展示了颇有前景的初步结果，可以为设计的实际实施提供支持。首先，我们展示了通过相对较短的训练和简单测试来进行身份识别是可以实现的。其次，用户习得的信息可以保持一到两周的时间：虽然有些人一周就已忘记，两周后又有一些人忘记，但这表明了比较长（指数形）的遗忘曲线。最后，在第三个实验中，我们审视了基于下述内容的攻击：请参与者完成包含所有最短长度片段的序列，以此尝试重建身份识别序列。我们的结果显示，参与者无法在此情况下表达对序列的可靠认知，这表明底层序列信息能够抵御攻击，直到攻击者正确猜测出更长的子序列为止。

2. 人类记忆系统概述

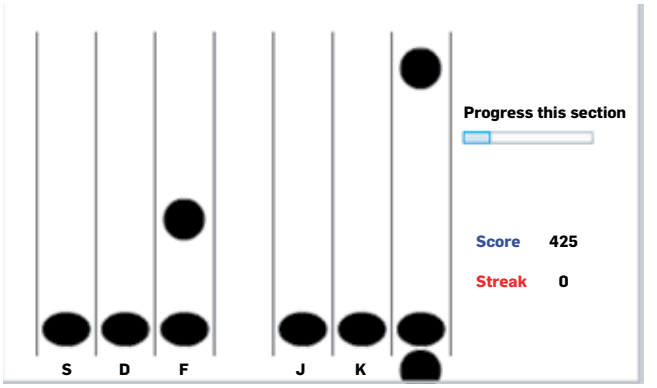
任何掌握了某种技能专长的人应该都熟悉，了解如何使用某项学好的技能和解释如何使用，这两者是有区别的。这种分离性反映了人类大脑中存在多个记忆系统。[5] 对于可口头表达的事实和事件的记忆依赖于内侧颞叶记忆系统（包括海马体）。然而，即便是内侧颞叶受损的患者（如因老年痴呆症而导致）也具备以内隐方式掌握新信息的完好能力，包括正常学习某些运动任务。[3] 通过数十载的实验性认知心理学研究，科学家们已经开发出可选择性地依赖这种内隐式、无意识学习系统的任务。

2.1.SISL 任务

在串行拦截系列学习 (SISL) 任务[7] 中，人类参与者在不知不觉中学习一个字母序列。该任务要求参与者拦截以预先确定的序列送出的移动物体（圆圈），具体的操作与热门游戏“吉他英雄”非常相似（图 1）。

在我们的修改版 SISL 中，每个圆圈出现在一列的顶部（共有六列），以恒定的速度垂直落下，直到其到达底部的“槽口”为止，此时它就会消失。玩家的目标是在物体接近槽口时将它拦截。拦截的方式是在物体处于正确的垂直位置时，按下与物体所在列对应的按键。按键错误或未按任何键将导致该物体的结果

图 1.进行中 SISL 任务的屏幕截图。



为不正确。在典型的 30–60 分钟训练活动中，参与者完成数千次尝试，在其中 80% 的尝试中，提示的顺序将遵循一个秘密嵌入的重复序列。该任务经过设计，可通过逐渐改变圆圈掉落的速度，使击中率达到大约 70%，从而使每个用户达到（但不超过）其能力的极限。通过将提示遵循所训练序列期间的表现水平（准确率）与提示遵循非训练序列期间的水平进行对比，就可以评估用户对嵌入的重复序列的掌握情况。

向用户呈现的所有序列都经过设计，防止出现显而易见、容易记忆的模式。具体而言，训练序列和随机序列都经过设计，其包含的每个有序字符对在一行中仅出现一次，无一字符出现两次（因此序列长度必须为 $6 \times 5 = 30$）。其结果为，虽然训练序列的表现要好于非训练序列，但参与者通常不能有意识地识别出训练序列。为了在实验中确认这一点，在 SISL 之后，参与者通常会被要求完成外显识别的测试，指出他们对各个不同序列的熟悉程度。

与使用 4 个键的原版 SISL 任务相比，我们的版本将可能序列的范围从仅 256 提高到超过 2400 亿。此外，我们在视觉显示的中部加入了一个间隙，使它更容易地将每一列与负责的左、右手产生正确的关联。

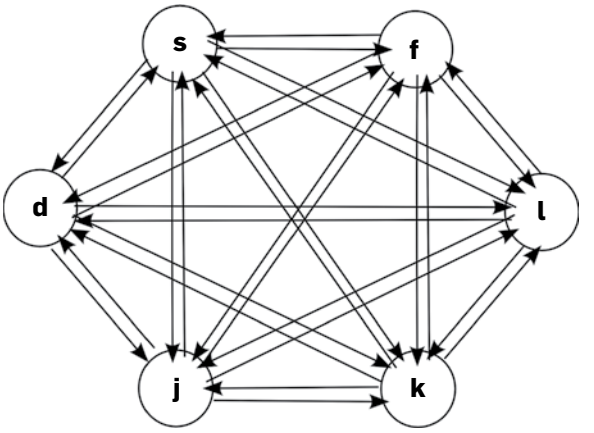
SISL 任务作为一个 Flash 应用程序通过网页浏览器提供给用户。参与者前往我们的网站 (www.brainauth.com)，然后会看到一份同意书。他们同意参加后，网页小程序即会下载一个随机训练序列，参与者就可以开始执行任务。在完成训练和测试尝试后，我们进行外显识别测试，然后将结果上传到服务器。讨论完身份验证系统后，我们会回过头来介绍 SISL 小程序在我们有多名用户参与的大型实验计划中发挥怎样的作用。

3. 利用内隐学习的基本身份验证系统

借助 SISL 任务，可以在人类大脑中存储可在身份验证期间检测、但无法被用户明确描述的机密密钥。这样的系统避免了人们因受他人劝说而泄露密码的问题，可以形成抗胁迫身份验证协议的基础。如果信息被泄露，可以训练新的身份识别序列作为替代，从而导致更换密码。

身份识别系统的运作分两个步骤：训练，然后身份验证。在训练阶段，用户学习的机密密钥如扩展 SISL 任务中所示，即一个由 30 个字符组成的序列，集合 $S = \{s, d, f, j, k, l\}$。我们仅使用 30 字符的序列，其与图 2 中的图上的一个 Euler 圈 (即每一条边恰出现一次的圈) 对应。这些序列有一个特性，S 集合中的每个非重复双字组（如 sd、dj 和 fk）都仅出现一次。为了预测下一项（例如，用于展示表现成绩提高），需要学习由三个或更多个项组成的小组之间的关联性。这将消除学习字母频率或常见字母对的可能，也就减少了对嵌入重复序列的有意识识别。[2]

图 2.我们生成的机密密钥是来自这一有向图中所有Euler 圈的随机 30 字符序列。生成的序列所包含所有双字组仅出现一次，并且不含重复字符。



我们用 Σ 表示所有可能机密密钥的集合，即与图 2 中 Euler 圈对应的 30 字符序列的集合。此图中的 Euler 圈的数量可通过 BEST 定理计算 [11]（其中 K_6 图的生成树数量为 6^4（根据 Cayley 公式），每个顶点的入度为 5，因此 $\Pi_v (deg(v)-1)!$ 是 24^6），得出下列等式

$$\#\ 密钥 = |\Sigma| = 6^4 \cdot 24^6 \approx 2^{37.8}。$$

因此，学习的随机机密为大约 38 比特的熵，比标准可记忆密码的熵大得多。

训练。用户在可信环境中玩 SISL 任务游戏，学习一个随机的 30 项的密钥 $k \in \Sigma$。在训练用户时，我们实验了下列程序：

- 在执行 SISL 任务期间，向受训人员提供包含 30 项的密钥序列（重复三次）以及从其他随机序列中选择的 18 个项（但有一个限制，即不能出现同一提示的连续重复），总计为 108 个项。
- 这一序列重复五次，所以一共向受训人员显示 540 个项。
- 在这一序列末尾，SISL 任务出现一个简短的暂停，然后包含 540个项的完整序列（包括其末尾的暂停）再重复六次。

在整个训练活动中，一共向受训人员显示 7 × 540 = 3780 个项，需要大概 30–45 分钟时间完成。在训练阶段完成后，受训人员进行身份验证测试（如下所述），确保训练成功。系统记录用户达到的最终游戏速度。

SISL 身份验证。 过段时间后再次身份验证时，向训练的用户呈现 SISL 任务，其提示结构中包含了来自训练的身份验证序列的元素，以及用于比较的非训练元素。若在训练元素上的表现成绩稳定优于未训练元素，参与者即可验证其身份。具体而言，我们对下列身份验证过程进行了实验：

- 我们假设 k_0 是训练的包含 30 个项的序列，k_1 和 k_2 是另外两个从 Σ 中随机选取的包含 30 个项的序列。所有身份验证过程中都使用相同的序列 (k_0, k_1, k_2)，这样就不会显露出有关 k_0 的附加信息。
- 系统选择 (0, 1, 2, 0, 1, 2) 的随机排列 π（如 π = (2, 1, 0, 0, 2, 1)），向用户呈现包含下列序列（540 = 18 × 30 项）的 SISL 任务：

  $$k_{\pi 1}, k_{\pi 1}, k_{\pi 1}, \ldots, k_{\pi 6}, k_{\pi 6}, k_{\pi 6}。$$

  即，k_0、k_1、k_2 中的每个都向用户刚好显示六次（两组三个重复），但次序是随机的。任务开始的速度与用户在训练结束时的速度相同。

- 对于 $i = 0, 1, 2$，使 p_i 等于用户在输入序列 k_i 的所有轮数中密钥输入正确的比率。满足以下条件时，系统宣告身份验证成功：

  $$p_0 > \text{average}\,(p_1, p_2) + \sigma \qquad (3.1)$$

  其中 $\sigma > 0$ 足够大，可以将偶然出现此差距的可能性降到最低，又不会导致身份验证失败。

在上述初步构想中，身份验证流程存在易受以下攻击的可能：非训练用户在两个区块之间降低其表现水平，以呈现出有利于训练序列的人为表现差异（获得通过身份验证的 1/3 概率）。我们将在第 5 章节中讨论抵御这种情形的可靠方式。但现在，对于这个简单的评估过程，我们可以通过两个简单的预防措施提供一些保护。首先，验证身份验证者是真人，确保难以持续改变陪衬区块 k_1 和 k_2 之间的表现。其次，在掌握序列过程中得到的最终训练速度对身份验证服务器是已知的，攻击者无法匹配训练区块和陪衬区块之间的表现成绩差异。表现成绩差别如果与训练后获得的存在显著不同，这就表明存在攻击。

分析。 以下两个章节讨论此系统的两个重要方面：

- 有用性：受训练的用户能否在以后可靠地完成身份验证任务？
- 安全性：如果拦截到受训练用户并胁迫其提供足够多的信息，攻击者可否通过身份验证？

4. 有用性实验

我们在初步实验中的报告表明了 SISL 身份验证系统的可行性和前景。我们分为三个阶段开展这些实验。首先，我们确认，使用 Mechanical Turk 可以通过 SISL 任务的全新扩展版观察到可靠的学习成果。其次，我们确认用户在 1 周和 2 周后仍保持对训练序列的认知。最后，我们对可用于重新构建原始系列的最小片段进行采样，并据此调查了对参与者的序列知识进行攻击的有效性。

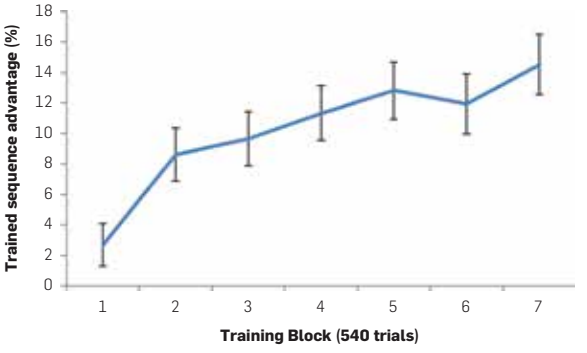
这些实验在 Amazon 的 Mechanical Turk 平台中在线开展。Mechanical Turk 的优势在于可以招募到基本不受限制的参与者，成本也将对较低。在线实验有一个缺点，即对于那些之后回来重复进行评估的用户，我们相对缺乏控制力。

4.1. 实验 1：内隐学习和外显学习

我们第一个实验确认了内隐学习可以被明确地检测到，而且对序列的外显有意识认知达到最小。分析中包含了来自 35 名参与者的实验数据。

该实验使用了前一章节中所述的训练过程，其中训练阶段包含总共 3780 次尝试，大约耗时 30-45 分钟。注意，该训练由七个各包含 540 次尝试的训练块组成。完成训练活动后，参与者进行了 SISL 身份验证测试，

图 3.在训练期间，参与者逐渐开始表现出对重复序列的认知，即在训练序列上的表现成绩优于随机散布的干扰段上的表现。请注意，任务的总体表现成绩始终保持在大约 70%，原因在于该任务的自适应性，即随着参与者在 SISL 上的总体表现的改善，其速度也会加快。



该测试将训练序列上的表现与两个随机测试序列上的表现作对比。

如图 3 所示，训练序列的学习成果取决于训练序列相对于随机出现的干扰段的表现优势（正确回答百分比的提升）。在训练之后的测试块中，参与者完成 SISL 的平均正确率为：训练序列 79.2%，非训练序列 70.6%。正确率相差 8.6% (SE[a] 2.4%) 表明，训练序列的表现存在可信的优势（成对样本 *t*-test 与零比较，$t(34) = 3.55, p < 0.01$）。

群体层面上的表现差别常常体现在内隐学习的测试中，但对于身份验证方法而言，必须能够进行可靠的个体评估。就个体参与者而言，在 540 测试尝试中，35 个案例中有 25 个可以辨别出训练序列与非训练序列的表现差别（卡方分析，$p < 0.05$）。出于身份验证的目的，需要通过更长的训练来确立内隐习得的序列，以此进一步加强评估的个体可靠性。然而，SISL 任务的特点就是能够通过相对较短的训练在大量个体中识别学习成果，而大多数内隐学习测试都不具备此特点。[7] 传统而言，内隐学习的测量依赖于评估个人群体的表现，不能在个体层面上识别学习成果。[9]

外显识别测试。完成训练和测试模块后，向参与者显示五个不同的动画序列，并询问对每一个序列的熟悉程度（从 0 到 10 打分）。在这五个序列中，其中一个是训练序列，另外四个是随机选择的陪衬序列。此测试评估了训练序列的外显识别记忆。

在识别测试中，参与者将对序列的熟悉程度按照 0 到 10 分进行打分，其平均分为：训练序列 6.5 分 (SE 0.4)，新的非训练序列 5.15 分 (SE 0.3)。从群体层面看，训练序列的识别率稍高是确凿的（$t(34) = 3.69$，$p < 0.01$），但并不与 SISL 表现关联 ($r = 0.13$)，表明这对内隐表现的好坏没有什么作用。内隐学习实验中经常会看到训练序列识别率稍高的现象，因为健康的参与者会在练习之后发现训练序列的某些部分比较熟悉。值得一提的是，内隐记忆并不转变为外显知识，即使经过重复使用也是如此；而且，训练的结构和长度以及测试序列都经过了专门设计，以降低随时间推移而积累外显知识的可能性。

识别率的总体差别较小（5.15 对 6.5）表明，参与者无法回忆出包含 30 个项的序列。这意味着，他们无法有意识地生成训练信息（例如，用于破坏身份验证方式的安全性）。我们将在第三个实验中进一步讨论重建问题。

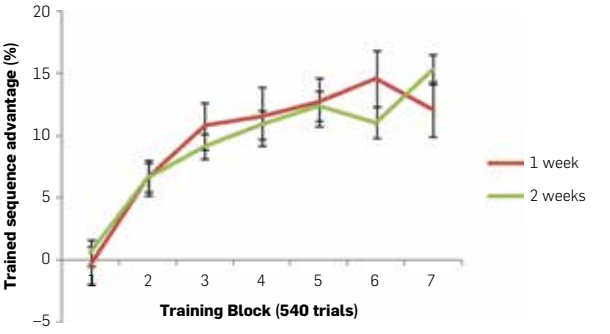
4.2. 实验 2：长期保持

只有在密码被记忆了一段时间以后身份验证依然可以准确执行，这样的身份验证机制才有用。我们在实验 2 中确认，用户获得的与序列相关的知识可以保持比较长的时间。尽管学得的技能通常会维持一段时间，但此前从未进行过具有较长延期和大量参与者的基于 SISL 的测试。

在实验 2 中，参与者同意分两个阶段完成 SISL 任务。参与者在第一阶段中完成与实验 1 中结构相同的训练序列。训练之后立即进行相同的 SISL 测试，以评估延期前的序列知识。一个由 32 名参与者组成的小组在 1 周后，返回在线小程序以进行训练序列的保持力测试和识别评估。另一个由 80 名参与者组成的小组在 2 周后进行保持力测试和识别测试。对于 1 周后返回的小组，测试由一个包含 540 次尝试的内隐序列学习评估组成。对于 2 周后返回的小组，测试的时长加倍，以进一步评估更长的测试能否提高对个体序列知识的敏感度。对于两个小组，延期测试的初始测试速度都设为与参与者在训练活动末尾执行该任务时的速度相匹配。通过一个由 180 次尝试组成的热身块来调整这一初始速度，以便参与者能够在保持力测试开始时，以大约 70% 的目标正确率执行任务。

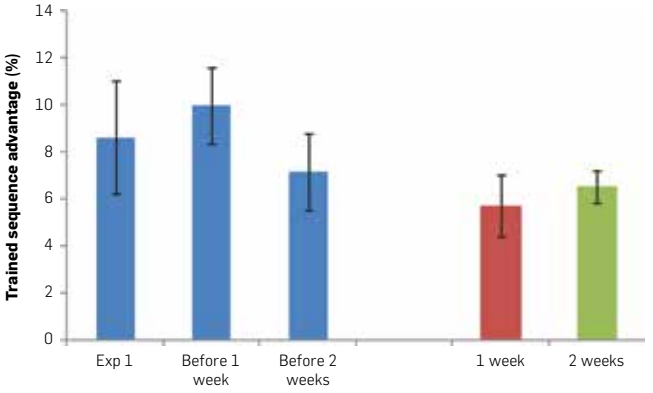
图 4 显示，和实验 1 一样，两组人都在第一阶段中逐渐地学习训练序列。图 5 显示了立即测试和延期测试的内隐序列知识。在所有五个评估中，参与者在

图 4.在训练期间，参与者逐渐开始表现出对重复序列的认知，即在训练序列上的表现成绩优于随机散布的干扰段上的表现。和预计的一样，两个组的学习表现相似，并与实验 1 相似。



[a] SE 是 *Standard Error*（标准误差）的缩写。换言之，如果训练序列和非训练序列的正确率测量遵循相同的正态分布，由 $N = 35$ 样本（所以有 $N - 1 = 34$ 个自由度）计算出来的 *t* 值应当接近于零，小于此处获得的 *t* 值（3.55），这表示有 99% 的概率这一差距是显著的。*t*-test 是一种标准的统计方法，用于确认被控变量（此处为序列类型）对测得的变量（正确率）的影响。

图 5.参与者在两个即时评估（实验 1 所示，以及实验 2 的两种状况）上展现了可靠的序列知识，即在测试时，训练序列上的表现成绩优于未训练的新序列上的表现。1 周和 2 周延期测试都体现了序列知识的可保持性。尽管在两个延期后都体现出知识的表达有一些缩减，但 1 周到 2 周之间没有显著的进一步衰减，这表明信息有可能会在 2 周后还能保持比较长的时间（许多类型的记忆都观察到指数或幂律衰减曲线）。



整体上展现了可靠的序列学习成果，$ts > 4.3$，$ps < 0.01$。在 1 周延期测试中，32 名参与者中有 15 名在个体上展现了可靠的序列知识。不过，对于 2 周延期小组，80 名参与者中有 49 名在个体上展现了可靠的序列知识，反映出使用较长的评估测试可以提高敏感度。未来的研究不仅将考察延长的训练时间，也将考察对个体知识敏感度更高的评估测试，以根据 SISL 表现提供可靠而准确的识别方式。

即使是延期 1 周和 2 周后，参与者仍表现出同样的趋势（虽然这个趋势并不明显），即对训练序列的识别度更高，$ts > 2.8$，$ps < 0.05$。需要重申的是，识别表现和序列知识表达没有关联（$rs < 0.16$），也没有人能够回忆整个包含 30 个项的训练序列。

5. 安全性分析

在这一章节中，我们将分析第 3 章节中的基本身份验证协议的安全性，并提出可提高安全性的一系列拓展。我们还对一次特定的攻击进行了实验，该攻击试图一次一个片段地从用户处获取机密序列。我们的 Mechanical Turk 实验显示，这种攻击对人类效果不佳。

5.1. 内隐学习作为密码原语

我们先看看通过内隐学习实现的新功能的抽象模型。在传统的建模中，密码协议中的参与者被建模为拥有对手不知的机密的实体。这些假设在面对胁迫时遭到瓦解，因为此时可以从参与者身上提取到所有的机密。

内隐学习提供了以下新的抽象功能：训练阶段将断言

$$p : \Sigma \to \{0, 1\}$$

嵌入到用户大脑中，对某个很大的集合 Σ。任何人都可以要求用户评估其断言 p 在某个点上 ($k \in \Sigma$) 的值。用户习得了 k 时，断言评估为 1，否则为 0。p 评估为 1 的输入数量相对较小。在绝大多数情形下，p 仅在一个点上评估为 1，也就是说，用户仅针对一个机密序列接受训练。

内隐学习的主要特征是，即便在受到胁迫的情况下，也无法从用户提取 $p(k) = 1$ 的点 $k \in \Sigma$。这一抽象属性抓住了这一事实，即机密序列 k 是用户通过内隐方式习得的，无法被有意识地访问。在这篇论文中，我们使用内隐学习原语来构建身份验证系统，但我们可以设想它在安全系统中得到更广泛的使用。

第 3 章节中所述的身份验证过程提供了 Σ 中某序列 k_0 的断言 $p(\cdot)$ 的实现。如果该过程宣告成功，我们可以认为 $p(k_0) = 1$，否则 $p(k_0) = 0$。断言 p 在训练活动中嵌入到用户的大脑中。

基础胁迫威胁模型。第 3 章节中的 SISL 身份验证系统经过设计，可抵御企图欺骗身份验证测试的对手。我们假设测试要求真人在场并从活性检查开始，以确保真人在没有任何仪器的协助下参加测试。为了欺骗身份验证系统，允许对手进行以下步骤：

- 提取步骤：拦截一名或多名受训练的用户，使他们（或许通过胁迫方式）尽可能泄露信息。
- 测试步骤：对手亲自提交信息到身份验证测试，其目标是通过该测试。在现实中，这可能意味着对手出现在安全设施的入口处，试图通过那里的身份验证测试。如果失败，他可能被扣留下来质询。

这种基础威胁模型给予攻击者一次机会挑战身份验证测试。本章节稍后部分中，我们将考虑另一种模型，即攻击者可能会重复提取和测试步骤，并在提取和测试之间交替进行。

我们也应注意，基础威胁模型假设，在训练阶段（即用户被授予凭据时），用户会遵循相关指示，而不会有意尝试误导训练过程。实际上，对手仅被允许在训练过程完成之后胁迫用户。

显而易见，第 3 章节中的系统在这一基础威胁模型下是安全的（假定训练过程将内隐习得的断言 p 嵌入在用户的大脑中）。事实上，如果攻击者拦截 u 名

受训练用户，使每一人遭受 q 次查询，其能够找到有效序列的概率最多为 $qu/|\Sigma|$。由于每一测试用时约五分钟，我们可以假设每个被俘用户的最多尝试次数为 $q = 10^5$ 次（此数量大概是每名用户不间断测试大约一年，而这可能会干扰用户习得的密码，导致用户对攻击者无用；或者向安全管理员发出用户不在场警报，导致凭据被撤销）。因此，即使在俘获了 $u = 100$ 名用户后，攻击者成功的可能性也仅为

$$100 \times 10^5/|\Sigma| \approx 2^{-16}.$$

让攻击者更麻烦的是，使一个人通过 SISL 查询许多个随机序列可能会导致其忘掉已习得的序列，或者导致其学会不正确的序列，从而使提取变得不可能。

我们应注意，设计为抵御胁迫攻击的身份验证系统需要真人在场。如果系统支持远程身份验证，那么攻击者可以胁迫受训练用户通过远程服务器进行身份验证，并劫持认证会话。

安全性增强。上述安全模型给予攻击者一次身份验证的机会，攻击者必须有较高的成功概率。如果攻击者被允许进行多次身份验证尝试，即重复执行提取和测试步骤并在两者之间交替进行，那么该协议可能会变得不安全。其原因是，攻击者在身份验证尝试期间能够看到三个序列 k_0、k_1 和 k_2，有可能会记住其中之一（30 个符号）。然后他可以对该序列进行离线训练，这样在下一次身份验证尝试时他可以拥有 1/3 的成功概率。如果攻击者能够记住所有三个序列（90 个符号），然后再重建 SISL 任务，他就能够迫使受训练的用户离线接触所有三个记住的序列，通过用户的表现可靠地判断哪个是正确的身份验证序列。之后攻击者可以针对该特定序列训练自己。这样，他在下一次身份验证尝试时就能*确保*获得成功。我们要补充的是，这一攻击者很难实现其目标，因为对于人类攻击者而言，要以执行任务的速度记住整个序列非常困难。

第 3 章中提到了另一种可能的攻击，攻击者碰巧是一名高手玩家，但有意在所提供的其中序列上降低其表现成绩。他有 1/3 的概率可以在正确的序列上展示出表现差别，从而通过身份验证测试。我们在第 3 章节中介绍了几种抵御方式。这里我们介绍一种更鲁棒的方式。

以上两种攻击都可通过组合学击破。我们训练用户时不针对单个序列，而是针对若干序列，例如四个。实验 [8] 表明，人类大脑可以学习多个序列，而且这些习得的序列不会互相干扰。另外，我们进行的新实验也表明，用户可以在 24 到 48 小时的间隔时间内训练多个包含 30 字符的序列，序列之间并没有出现可测量到的干扰。同样，我们也能够针对较长的序列训练用户，并使用其片段来进行身份验证。尽管上述数据表明，最短的片段（3 个项）无法用于评估对较长序列的认知，但最近我们发现，针对较长的片段（如 5 - 7 个项），该序列知识却能够可靠地表达出来。[6] 因此，通过在初始训练中投入更多时间对更多信息进行编码，我们可以使用基于片段的测试来提高对上述窃取式攻击的抵御能力。

在身份验证期间，我们不使用一个正确序列和两个陪衬序列，而是使用四个正确的序列（或片段）并随机搭配 8 个陪衬序列。如果攻击者在 12 个预设序列中的 4 个正确序列上显示出可测量到的表现差别，那么就可通过身份验证。而在随机序列上速度减慢的攻击者现在最多有 $1/\binom{12}{4} \approx 1/500$ 的几率通过测试。可以通过调节训练序列数量 (4) 和陪衬序列数量 (8)，在安全性和有用性之间达成可接受的平衡。

与之类似，少量的身份验证尝试将不能帮助直接攻击者通过测试。不过，记住身份验证测试（360 个符号），稍后再呈现给受胁迫的用户，可以给对手带来优势。为进一步防御这种记忆式攻击，我们在身份验证过程中加入了一个额外步骤：一旦身份验证服务器发现用户无法在部分训练序列上展现可测量的表现差别，所有剩余的训练序列就被替换为随机陪衬序列。这样一来，若是攻击者在没有先前知识的情况下尝试身份验证，就不能看到全部的训练序列，因而就无法从受胁迫的用户那里提取到所有训练序列。因此，就无法对受胁迫用户发起*“一击致命”的*攻击。然而，通过重复这一过程，即进行身份验证测试、记忆观察到的序列，再在受胁迫的训练用户身上测试它们，攻击者或许最终能学会所有训练序列，并成功欺骗身份验证测试。但在这一过程中，攻击者必须参与身份验证测试，在该测试中，他能够证明自己对严格的一小组训练序列的认知，但无法证明对所有序列的认知。这给予系统一个受到攻击的明确信号，此时参与身份验证的人员可能会被扣留进行质询，而合法的用户将被阻止通过该系统进行身份验证，直到其重新训练了一组新的序列。

窃取安全性。传统的密码身份验证容易遭受窃取（通过客户端恶意软件或肩窥方式），此处所示的身份验证系统亦是如此。窃取者如果获取了受训练用户的多个有效身份验证脚本，就可重新构建学习的序列。设计一套抗胁迫系统并在服务器的问答式协议中使用

内隐习得的机密，是一个很有前景的未来研究方向。我们将在本文结尾部分重新讨论这一问题。

5.2. 实验：提取序列片段

我们的系统可能会受到这样的攻击：怀有恶意的一方分析出合法用户的知识特征并使用该信息对训练序列进行反向工程，进而通过身份验证测试。虽然可能的训练序列数量过多，无法对任何一个序列进行穷举测试，但每一序列的构建都有已知的局限，掌握序列片段或许可让攻击者能够重建原始序列或者其足够的部分，以至通过身份验证测试。

训练序列被限制为以均等的频率使用所有 6 个回答按键，因此对个别回答概率的分析无法提供有关训练序列的信息。与此类似，所有 30 个回答按键对（6 × 5 = 30，因为按键都不重复）在训练期间以均等的频率出现，这意味着双字组频率也不能提供有关训练序列的信息。不过，每个包含 30 个项的序列拥有 30 个唯一三字组（共 150 个可能）。如果特定的训练三字组片段可以被识别，那么其中含有的训练序列就有可能被重建。

基于这一信息的攻击将必须让受训练用户执行 SISL 测试，该测试包含频率均等的所有 150 个三字组。如果用户在 30 个训练三字组上的表现优于在 120 个非训练三字组上的表现，那就可以重建该序列。此攻击能够削弱该方法对外部压力的相对抵抗力，而致泄露身份验证信息。

然而，虽然可以在三字组层面上确定序列信息，但目前尚不清楚参与者在如此短的片段上能否可靠地展现序列知识。在实验 3 中，我们评估了在这类三字组测试上的表现，以此衡量重建序列信息的可能性。

我们再一次通过 Mechanical Turk 招募参与者，并完成了与实验 1 和 2 中相同的训练活动。在测试时，参与者所执行的序列被构建为提供 150 个三字组，每个都刚好提供 10 次，即构建 10 组不同的包含 150 次尝试的单元，每一组以不同的顺序包含所有可能的三字组。将每个三字组上的表现成绩作为当前回答与前两个回答的函数来衡量（表示为正确率）。

为评估此数据是否能用于重新构建序列，我们逐一计算各个三字组的正确率，并逐一创建所有三字组的排名。如果训练三字组的表现优于其他三字组，那么训练三字组的排名往往较低（例如，成绩表现会导致序列三字组成为前 30 个最佳回答）。然而，在平均排名和平均正确率上，训练三字组和非训练三字组之间没有明显的差别。参与者并未在这一类型的测试上展现出他们的训练序列知识，这表明无法借助基于三字组的方式攻击其序列知识。更具体而言，我们针对每一用户比较了 30 个训练序列三字组和 120 个剩余三字组的平均正确率测量结果。34 名参与者的三字组平均正确率为：训练序列 73.9% (SE 1.2%)，其余 73.2% (SE 1.1%)。这一差别不具足够的说服力。

虽然三字组测试并不能表现出序列知识，但也存在通过一些更长的片段评估出参与者序列知识的可能。实际上，我们的进一步实验证实了这一理论：当片段长度为 4 时，我们发现片段中的第三个字符确实会显现较好的表现成绩，与训练一致；当片段长度为 5 时，第三和第四个字符显现表现成绩的提高。有趣的是，向用户呈现的训练片段中的最后一个字符并未显现出优于非训练序列平均表现的成绩。其原因可能是，下一片段中的第一个（意料之外的）字符“重置”了用户的表现。

使用片段分析的攻击难度很高，因为即便是中等长度的片段，其数量也过多，而且分析每一片段的用时也过长。例如，对于长度为 4 的片段（四字组），存在 750 种可能性并且需要运行多次。在训练序列中添加可变时序可以进一步减少对协议的这一类攻击，这种方式可以快速扩展组合空间。通过实验我们发现，时序变化可以“抹掉”在以不同时序模式训练的序列上的表现。[4]

6. 总结和未来研究

我们介绍了胁迫攻击的一种新型抵御方式，它利用来自认知心理学的*内隐学习*概念。我们介绍了一个概念验证协议，以及通过 Mechanical Turk 开展的初步实验，这些实验让我们相信，构建抗软磨硬泡式攻击的身份验证系统是可行的。

尚有许多工作要去完成。我们希望进一步分析内隐习得的密码的遗忘速度，以及巩固训练的所需频率。此外，我们也希望找到方法来检测或预测个体用户可靠学习的时间（收集更多用户的人口统计数据，以及开展多阶段长期实验，或许是这一方向的良好开端）。我们还希望探索这一方法的其中一些限制。例如，通过确定习得序列中对攻击者与合法身份验证者存在差别的部分的最短长度，以及加强测试过程和分析，来提高在更大比例用户之中的可靠性，或缩短必要的测试时间，减少误报和漏报等。在提示之间使用可变时序，并将用户表现成绩作为任务速度的函数来测量，可进一步提高测试协议的可靠性。多个凭据的内隐学习同样可从更多实验中获益。这些实验所依据的以往研究工作已发现，用户在学习不同的包含 12 个项的序列时

没有出现互相干扰的迹象，同时用户还能以内隐方式学习最长为 80 个项的序列。

这一研究工作的另一未来方向是测试能否以内隐方式学习更为复杂的结构，如 Markov 模型。我们希望利用此类学习来构建可抵御窃取和胁迫的问答式身份验证体系。最后，除了身份验证外，我们也希望探究各种以内隐学习为基础的密码原语的构建。

致谢

在此感谢所有有偿志愿者的参与，他们为我们的用户研究做出了重要的贡献。本研究获得了 NSF 和 MURI 项目的资助。

参考资料

1. Denning, T., Bowers, K.D., van Dijk, M., Juels, A. Exploring implicit memory for painless password recovery.In *CHI*. D. S. Tan, S. Amershi, B. Begole, W. A. Kellogg, and M. Tungare, eds. ACM, 2011, 2615–2618.
2. Destrebecqz, A., Cleeremans, A. Can sequence learning be implicit?New evidence with the process dissociation procedure.*Psychonomic Bull.Rev. 8* (2001), 343–350.
3. Gobel, E., Blomeke, K., Zadikoff, C., Simuni, T., Weintraub, S., Reber, P. Implicit perceptual-motor skill learning in mild cognitive impairment and Parkinson' s disease.*Neuropsychology*, *27*, 3 (2013), 314–321.
4. Gobel, E., Sanchez, D., Reber, P. Integration of temporal and ordinal information during serial interception sequence learning.*J. Exp.Psychol.Learn. Mem.Cognit.37*, 4 (2011), 994–1000.
5. Reber, P. Cognitive neuroscience of declarative and non-declarative memory.*Parallels in Learning and Memory*.M. Guadagnoli, M.S. deBelle, B. Etnyre, T. Polk, and A. Benjamin, eds.North-Holland, 2008, 113–123.
6. Sanchez, D., Bojinov, H., Lincoln, P., Boneh, D., Reber, P. Statistical learning in perceptual-motor sequences and planning effects in performance.In *Poster at the Meeting of the Society for Neuroscience* (2012).
7. Sanchez, D., Gobel, E., Reber, P. Performing the unexplainable: implicit task performance reveals individually reliable sequence learning without explicit knowledge.*Psychonomic Bull. Rev. 17* (2010), 790–796.
8. Sanchez, D., Reber, P. Operating characteristics of the implicit learning system during serial interception sequence learning.*J. Exp.Psychol.Hum. Percept.Perform.38*, 2 (2012), 439–452.
9. Schwarb,H., Schumacher,E.H.Generalized lessons about sequence learning from the study of the serial reaction time task.*Adv. Cognit.Psychol.8*, 2 (2012), 165–178.
10. Soghoian, C. Turkish police may have beaten encryption key out of TJ Maxx suspect, 2008. news.cnet.com/8301-13739_3-10069776-46.html.
11. van Aardenne-Ehrenfest, T., de Bruijn, N.G.Circuits and trees in oriented linear graphs.*Simon Stevin 28* (1951), 203–217.
12. Wikipedia.Rubber-hose cryptanalysis, 2011.

Hristo Bojinov 和 Dan Boneh 来自美国加利福尼亚州斯坦福的斯坦福大学

Daniel Sanchez 和 Paul Reber 来自美国伊利诺伊州埃文斯顿的西北大学

Patrick Lincoln 来自美国加利福尼亚州门洛帕克的斯坦福国际研究所

译文责任编辑：孙晓明