WILEY

多智能体机器学习：强化学习方法

Multi-Agent Machine Learning: A Reinforcement Approach

[加拿大] 霍华德 M.施瓦兹(Howard M.Schwartz) 著

连晓峰 谭励 等译

本书提供了一种多智能体不同学习方法的框架。同时还提供了多智能体微分博弈中的最新进展以及在博弈理论和移动机器人中应用的全面概述。本书向读者介绍了多智能体机器学习的不同方法。主要包括单智能体强化学习、随机博弈和马尔科夫博弈、自适应模糊控制和推理、时间差分学习和Q学习。本书具有如下特点：

- 全面涵盖了多人博弈、微分博弈和博弈理论；
- 基于梯度算法的简单策略学习方法；
- 多人矩阵博弈和随机博弈的详细算法和示例；
- 群机器人和性格特征进化中的学习示例。

机械工业出版社
CHINA MACHINE PRESS

多智能体机器学习：强化学习方法

[加拿大] 霍华德 M. 施瓦兹（Howard M. Schwartz） 著
连晓峰 谭励 等译

机 械 工 业 出 版 社

图书在版编目（CIP）数据

多智能体机器学习：强化学习方法 /（加）霍华德 M. 施瓦兹（Howard M. Schwartz）著；连晓峰等译 . —北京：机械工业出版社，2017. 6

书名原文：Multi – Agent Machine Learning: A Reinforcement Approach

ISBN 978-7-111-56960-2

Ⅰ. ①多… Ⅱ. ①霍… ②连… Ⅲ. ①机器学习 Ⅳ. ①TP181

中国版本图书馆 CIP 数据核字（2017）第 119153 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：顾 谦 责任编辑：顾 谦

责任校对：刘秀芝 封面设计：马精明

责任印制：李 昂

三河市国英印务有限公司印刷

2017 年 7 月第 1 版第 1 次印刷

169mm × 239mm · 12. 25 印张 · 228 千字

0 001— 4 000 册

标准书号：ISBN 978-7-111-56960-2

定价：69.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：010 – 88361066

读者购书热线：010 – 68326294

010 – 88379203

网络服务

机 工 官 网：www. cmpbook. com

机 工 官 博：weibo. com/cmp1952

金 书 网：www. golden – book. com

教育服务网：www. cmpedu. com

封面无防伪标均为盗版

本书主要介绍了多智能体机器人强化学习的相关内容。全书共6章，首先介绍了几种常用的监督式学习方法，在此基础上，介绍了单智能体强化学习中的学习结构、值函数、马尔科夫决策过程、策略迭代、时间差分学习、Q学习和资格迹等概念和方法。然后，介绍了双人矩阵博弈问题、多人随机博弈学习问题，并通过3种博弈游戏详细介绍了纳什均衡、学习算法、学习自动机、滞后锚算法等内容，并提出L_{R-I}滞后锚算法和指数移动平均Q学习算法等，并进行了分析比较。接下来，介绍了模糊系统和模糊学习，并通过仿真示例详细分析算法。最后，介绍了群智能学习进化以及性格特征概念和应用。全书内容丰富，重点突出。

本书可作为从事机器学习、多智能体协同控制等领域的工程技术人员的参考书，也可作为高等院校相关专业本科生、研究生以及教师的参考用书。

译 者 序

“多智能体”——一般专指多智能体系统（Multi - Agent System，MAS）或多智能体技术（Multi - Agent Technology，MAT）。多智能体系统是分布式人工智能的一个重要分支，是20世纪末~21世纪初国际上人工智能的前沿学科。多智能体学习相关的研究领域已成为人工智能发展的热点。

本书主要介绍了多智能体学习的相关内容，目的在于解决大型、复杂的现实问题，而解决这类问题已超出了单个智能体的能力。研究者主要研究智能体之间的交互通信、协调合作、冲突消解等方面，强调多个智能体之间的紧密群体合作，而非个体能力的自治和发挥，关于Lyapunov技术的非线性自适应控制方面的理论材料被减少，取而代之的是有关强化学习的思想。强化学习的目标是取得最大化的奖励（回报）。强化学习和非监督学习最有趣的部分就是奖励的选择，这是一个全新的发展迅速的应用领域。机器人团队必须要学会共同工作和相互竞争。本书是一本专门介绍多智能体强化学习的著作。

本书中重点研究了双人阶段博弈和矩阵博弈问题。其中主要通过3个不同的博弈游戏：猜硬币、石头－剪刀－布和囚徒困境来进行阐述。这些都被称为矩阵博弈（matrix games）或阶段博弈（stage games）的游戏，因为在游戏过程中没有发生状态转移。本书没有过于深入研究博弈论本身，而是专注于与这些游戏相关的学习算法。另外，作者还结合自己的教学实践，探讨了多机器人智能体的微分博弈问题，并通过“逃跑者－追捕者”博弈和“疆土防御”博弈进行了深入讨论。

需要指出的是，书中矩阵、矢量为保持与原书一致，并未使用黑斜体，请读者注意。

本书第1~3章由谭励翻译，第4~6章由连晓峰翻译，全书由连晓峰审校统稿，彭森、于嘉骥、李世明、李伟男、蔡有林、侯宝奇、窦超、张鹏、侯秀林、张欣、邵妍洁、张吉东、张丹瑶、赵辰等人也参与了部分内容的翻译。

由于译者的水平有限，书中不当或错误之处恳请各位业内专家学者和广大读者不吝赐教。

译者

原书前言

十年来，本人一直在教授自适应控制课程。这门课程主要是讲授系统辨识的常用经典方法，并使用经典的教材，例如 Ljung[1,2]。该课程着重介绍了参考模型自适应控制的常用方法以及基于 Lyapunov 技术的非线性自适应控制方法。然而，这些理论已不再适用于当前的工程实践。因此，在本人的研究工作以及研究生课程的重点内容中进行了相应调整，增加了自适应信号处理的内容，并融合了基于最小方均（LMS）算法的自适应信道均衡和回声消除的内容。同时，课程名称也相应地从“自适应控制”变为“自适应与学习系统”。本人的研究工作仍主要集中于系统辨识和非线性自适应控制在机器人方面的应用。然而，直到 21 世纪初，才开始与机器人团队开展合作。目前，已能够利用常用的机器人套件和低成本的微控制器来构建可协同工作的若干个机器人。这使得“自适应与学习系统”的研究生课程内容再次发生变化：减少了基于 Lyapunov 技术的非线性自适应控制方面的理论知识，取而代之的是有关强化学习的思想。这是一个全新的应用领域，机器人团队必须要学会相互协作和竞争。

目前，研究生课程主要是集中于采用基于递归最小二乘（RLS）算法的系统辨识、基于参考模型的自适应控制（仍然采用 Lyapunov 技术）、基于 LMS 算法的自适应信号处理以及基于 Q 学习算法的强化学习。本书的前两章简要介绍了上述思想，但也足以说明这些学习算法之间的联系，以及它们之间的相同之处和不同之处。与这些内容相关的其他材料可详见文献［2－4］。

由此，进一步的研究工作开始着重于机器人团队如何学习以实现相互合作。这些研究工作用于验证机器人在合作搜索和救援以确保重要设施和边界区域安全方面的应用。同时，也逐步开始关注强化学习和多智能体强化学习的研究。这些机器人就是具有学习能力的智能体。孩子们是如何学习玩捉人游戏的？人们是如何练习踢足球的？以及在追捕罪犯的过程中警察是如何协作的？应该采用什么样的策略？如何制定这些策略？当和一群新朋友玩足球时，如何能够快速评估每个人的能力，并在比赛中采用特殊策略呢？

随着研究团队开始致力于深入研究多智能体机器学习和博弈理论，逐渐发现尽管已有很多相关论文发表，但并不集中也不够全面。虽然已有一些综述性文章[5]，但均未能充分说明这些不同方法的具体细节。本书旨在向读者介绍一种特殊形式的机器学习。全书主要是关于多智能体机器学习，同时也包括一般学习算法的核心内容。学习算法的形式各不相同，然而往往都具有相似方法。在此，

将着重比较这些方法的相同和不同之处。

本书的主要内容是基于本人的研究工作，以及过去 10 年里所指导下的博士生、硕士生的研究工作。在此，特别感谢 Sidney Givigi 教授。Givigi 教授为本书第 6 章中所介绍的主要思路和算法提供了坚实基础。另外，本书中还包含了 Xiaosong（Eric）Lu 博士的研究成果。其中，关于疆土守卫部分的内容主要来源于其博士论文。同时，还有一些研究生也为本书做出了贡献，他们是 Badr Al Faiya、Mostafa Awheda、Pascal De Beck - Courcelle 和 Sameh Desouky。如果没有研究小组中学生们的辛勤工作，本书是不可能完成的。

Howard M. Schwartz

于加拿大渥太华

2013 年 9 月

参 考 文 献

[1] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1999.

[2] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, Massachusetts: The MIT Press, 1983.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 1998.

[4] Astrom, K. J. and Wittenmark, B., *Adaptive Control*. Boston, Massachusetts: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1994, ISBN = 0201558661.

[5] L. Buşoniu and R. Babuška, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst. Man Cybern. Part C*, Vol. **38**, no. 2, pp. 156–172, 2008.

目　录

第1章　监督式学习概述

在系统辨识、自适应控制、自适应信号处理和机器学习中往往会用到一些算法，这些算法都有一定的相似性和差异性。然而，上述算法都需要处理某种类型的实验数据。如何采集和处理数据决定了采用哪种最合适的算法。在自适应控制中，具有一个称为自校正调节器的装置。在此情况下，算法主要是用于测量状态以作为输出，估计模型参数，并输出控制信号。在强化学习中，算法的作用是处理奖励、估计值函数并输出相应操作。尽管递归最小二乘（RLS）算法在自校正调节器中称为监督式学习算法，而在强化学习中看作一种非监督式学习算法，但实际上两者十分相似。在本章中，将介绍一些常用的监督式学习算法。

1.1　LS 算法

最小二乘（LS）算法是一种将实验数据拟合为模型的著名的鲁棒算法。首先是为用户定义一个适合于拟合数据的数学结构或模型；其次是要设计一个在适用条件下采集数据的实验，“适用条件”通常是指在系统典型运行的操作条件；接下来是运行可能具有多种形式的估计算法；最后验证所辨识的或“学习”的模型。LS 算法通常用于拟合数据。在此，以大多非常熟悉的经典二维线性回归拟合为例：

$$y(n) = ax(n) + b \tag{1.1}$$

在上述简单线性回归模型中，输入为采样信号 $x(n)$，输出为 $y(n)$。所定义的模型结构是一条直线。因此，假设所采集的数据将会拟合成一条直线。由此可表示为

$$y(n) = \phi^{\mathrm{T}}\theta \tag{1.2}$$

式中，$\phi^{\mathrm{T}} = [x(n) \quad 1]$；$\theta^{\mathrm{T}} = [a \quad b]$。

如何选择 ϕ 决定了模型结构，这也反映了认为数据所应表现的形式。这就是机器学习的本质，而且也是几乎所有的大学生在某种程度上学习线性回归的基本情况。线性回归算法的计算可以表示为标量成本函数，由下式给出：

$$V = \sum_{n=1}^{N} (y(n) - \phi^{\mathrm{T}}(n)\hat{\theta})^2 \tag{1.3}$$

式中，$\hat{\theta}$是 LS 算法中参数 θ 的估计值，目的是在估计值 $\hat{\theta}$ 下使得成本函数 V 最小。

为得到参数估计 $\hat{\theta}$ 的“最优”值，应计算成本函数 V 对于 $\hat{\theta}$ 的偏导并设其为零。由此可得

$$\frac{\partial V}{\partial \hat{\theta}} = \sum_{n=1}^{N} (y(n) - \phi^{\mathrm{T}}(n)\hat{\theta})\phi(n)$$

$$= \sum_{n=1}^{N} \phi(n)y(n) - \sum_{n=1}^{N} \phi(n)\phi^{\mathrm{T}}(n)\hat{\theta} \tag{1.4}$$

令 $\frac{\partial V}{\partial \hat{\theta}} = 0$，可得

$$\sum_{n=1}^{N} \phi(n)\,\phi^{\mathrm{T}}(n)\,\hat{\theta} = \sum_{n=1}^{N} \phi(n)y(n) \tag{1.5}$$

求解 $\hat{\theta}$，可得 LS 的解：

$$\hat{\theta} = [\sum_{n=1}^{N} \phi(n)\,\phi^{\mathrm{T}}(n)]^{-1}[\sum_{n=1}^{N} \phi(n)y(n)] \tag{1.6}$$

其中，逆矩阵 $[\sum_{n=1}^{N} \phi(n)\,\phi^{\mathrm{T}}(n)]^{-1}$ 存在。如果逆矩阵不存在，则该系统是不可辨识的。例如，如果在直线情况下只有一个单点，则逆矩阵不会扩展到二维空间，因此不可能存在。所以，至少需要两个相互独立的点才能绘制一条直线。或者如果具有一个不断重复的同一点，逆矩阵也不可能存在。矩阵 $[\sum_{n=1}^{N} \phi(n)\,\phi^{\mathrm{T}}(n)]$ 称为信息矩阵，这关系到参数估计的程度。信息矩阵的逆矩阵是协方差矩阵，与参数估计的方差成正比。上述两个矩阵都是正定对称矩阵。这些特性广泛用于算法性能的分析。在一些文献中，通常将协方差矩阵表示为 $P = [\sum_{n=1}^{N} \phi(n)\,\phi^{\mathrm{T}}(n)]^{-1}$。在此，将式（1.4）中右边第二项表示为

$$\frac{\partial V}{\partial \hat{\theta}} = 0 = \sum_{n=1}^{N} (y(n) - \phi^{\mathrm{T}}(n)\,\hat{\theta})\phi(n) \tag{1.7}$$

由此可定义预测误差为

$$\epsilon(n) = (y(n) - \phi^{\mathrm{T}}(n)\hat{\theta}) \tag{1.8}$$

式(1.7)中括号内的项称为预测误差，或也可称为“新项”。$\epsilon(n)$表示系统输出的预测误差。在此情况下，输出项 $y(n)$ 为所需估计的正确值。由于已知正确值，因此称为监督式学习。值得注意的是，预测误差与数据矢量的乘积等于零。因此，可认为预测误差与数据正交，或者说，数据不在预测误差的空间中。简单来说，这意味着如果已经选择了一个良好的模型结构 $\phi(n)$，则预测误差应表现为白噪声。通常通过绘制预测误差可快速检查所设计的预测器性能。如果误差是相关的（即

不是白噪声），那么就应该继续优化模型以获得更好的预测结果。

一般而言，并不常用式（1.2）中的线性回归形式，而通常会增加白噪声项，由此，线性回归可表示为

$$y(n)=\phi^{\mathrm{T}}(n)\hat{\theta}+v(n) \tag{1.9}$$

式中，$v(n)$为白噪声项。

式（1.9）可表示无限个可能的模型结构。例如，假设要学习一个二阶线性系统的动态性能或一个二阶无限冲激响应（IIR）滤波器的参数。可以选择下式给出的二阶模型结构：

$$y(n)=-a_1y(n-1)-a_2y(n-2)+b_1u(n-1)+b_2u(n-2)+v(n) \tag{1.10}$$

那么，模型结构可由 $\phi(n)$定义为

$$\phi^{\mathrm{T}}(n)=[y(n-1)\quad y(n-2)\quad u(n-1)\quad u(n-2)] \tag{1.11}$$

一般情况下，可将任意一个 k 阶自回归外生（ARX）模型结构表示为

$$\begin{aligned}y(n)=&-a_1y(n-1)-a_2y(n-2)-\cdots-a_my(n-k)\\&+b_1u(n-1)+b_2u(n-2)+\cdots+b_{n-k}u(n-k)+v(n)\end{aligned} \tag{1.12}$$

则 $\phi(n)$表示为

$$\phi^{\mathrm{T}}(n)=[y(n-1)\cdots y(n-m)\quad u(n-1)\cdots u(n-m)] \tag{1.13}$$

然后，需选择一个适当的实验进行数据采集（这并不容易!），并根据式（1.6）计算参数。矢量 $\phi(n)$可具有多种不同形式，实际上还可包含数据的非线性函数，如对数项或二次方项，以及具有不同的延迟项。在很大程度上，可根据专业经验来确定 $\phi(n)$的形式。通常数据以矩阵形式表示，此时，矩阵可定义为

$$\Phi=[\phi(1)\ \phi(2)\cdots\phi(N)] \tag{1.14}$$

而输出矩阵为

$$Y=[y(1)\ y(2)\cdots y(N)] \tag{1.15}$$

由此，LS 估计可写为

$$\hat{\Theta}=(\Phi\Phi^{\mathrm{T}})^{-1}\Phi Y \tag{1.16}$$

此外，还可将预测误差表示为

$$E=Y-\Phi^{\mathrm{T}}\hat{\Theta} \tag{1.17}$$

同时，正交条件也可表示为 $\Phi E=0$。

用于参数辨识或机器学习的 LS 方法已非常成熟，并且与此技术相关的还有许多特性。实际上，统计推理的许多研究成果都来自于本节中所介绍的几个公式。这也是包括社会科学工作在内的许多科学调查研究的根源。

1.2　RLS 算法

LS 算法现已扩展到 RLS 算法。在此，参数估计发展为利用机器来实时采集

数据。在 1.1 节中，都是首先采集所有数据，然后根据式（1.6）计算参数估计值。RLS 算法是在假设已知 LS 算法的一个解并增加单个数据点的基础上推导而得的。具体推导过程详见文献［1］。在 RLS 算法的实现过程中，成本函数稍有不同。此时的成本函数为

$$V = \sum_{n=1}^{N} \lambda^{(N-t)} (y(n) - \phi^{\mathrm{T}}(n)\hat{\theta})^2 \tag{1.18}$$

式中，$\lambda \leqslant 1$，λ 项称为遗忘因子。

数据点越早，则遗忘因子权重越小。这样，所得到的 RLS 算法就能够跟踪参数的变化。同样，取 V 相对于 $\hat{\theta}$ 的偏导并设为零，可得

$$\hat{\theta} = [\sum_{n=1}^{N} \lambda^{(N-t)} \phi(n)\phi^{\mathrm{T}}(n)]^{-1} [\sum_{n=1}^{N} \lambda^{(N-t)} \phi(n) y(n)] \tag{1.19}$$

其中，遗忘因子应设为 $0.95 \leqslant \lambda \leqslant 1.0$。若遗忘因子设为 0.95 左右，则之前的数据会很快遗忘。经验法则表明，参数 $\hat{\theta}$ 的估计主要是根据 $1/(1-\lambda)$ 个数据点。

RLS 算法如下：

$$\hat{\theta}(n+1) = \hat{\theta}(n) + L(n+1)(y(n+1) - \phi^{\mathrm{T}}(n+1)\hat{\theta}(n))$$

$$L(n+1) = \frac{P(n)\phi(n+1)}{\lambda + \phi^{\mathrm{T}}(n+1)P(n)\phi(n+1)}$$

$$P(n+1) = \frac{1}{\lambda}\left(P(n) - \frac{P(n)\phi(n+1)\phi^{\mathrm{T}}(n+1)P(n)}{\lambda + \phi^{\mathrm{T}}(n+1)P(n)\phi(n+1)}\right) \tag{1.20}$$

通过将参数估计矢量 $\hat{\theta}$ 初始化为用户所需参数的最佳估计来实现式（1.20），为简单起见，通常设为零。协方差矩阵 P 通常初始化为一个相对较大的对角矩阵来表征参数估计过程中的不确定性。

尽管可以根据式（1.20）来实现 RLS 算法，但应注意到协方差矩阵 P 总是正定对称的。如果由于重复计算 RLS 而产生的数值误差，导致 P 矩阵不再正定对称，则算法将发散。现已有一些改进算法能够确保 P 矩阵保持正定。通常是采用 P 矩阵可进行 Cholesky 分解或 UDU 分解的二次方根法。这些方法可详见文献［1］。

观察式（1.20）可知，是通过将之前的估计值与矩阵 $L(n)$ 和当前预测误差的乘积相加来实现参数估计的更新。在机器学习的几乎所有算法中将会发现全部采用这种结构。在此情况下，已具有一个实际的正确值，即量测值 $y(n)$，因此该算法称为监督式学习。

1.3 LMS 算法

在信号处理领域中，有一些常用技术来建模或表征通信信道的动态特性，并

补偿信道效应对信号的影响，这些技术称为信道均衡和回声消除。目前关于自适应信号处理和自适应滤波已有许多相关著作[2]。上述技术大多采用最小均方（LMS）算法来辨识信道模型的系数。同样，正如 LS 和 RLS 算法所述，必须选择一个合适的模型结构来定义通信信道的动态特性。在信号处理领域中，通常采用有限冲激响应（FIR）滤波器作为描述系统的基本模型结构。为了与之前保持一致，在此将信道动态性表示为

$$y(n) = b_0 u(n) + b_1 u(n-1) + \cdots + b_k u(n-k) + v(n) \tag{1.21}$$

式中，$y(n)$为时间步长 n 时滤波器或通信信道的输出；b_i 为欲估计或“学习”的滤波器系数；$u(n)$为输入信号。

通常，信号 $u(n)$是需要从输出信号 $y(n)$中恢复的通信信号。在此，定义误差信号为

$$\epsilon(n) = y(n) - \hat{y}(n) \tag{1.22}$$

式中，$\hat{y}(n) = \phi^{\mathrm{T}}(n)\hat{\theta}$，这是与式（1.18）中预测误差相同的信号。

在 LMS 算法中，定义成本函数为预测误差的预期值：

$$J(n) = E[\epsilon^2(n)] \tag{1.23}$$

方均误差项可表示为

$$\begin{aligned}\epsilon^2(n) &= (y(n) - \phi^{\mathrm{T}}(n)\hat{\theta})^2 \\ &= y^2(n) - 2y(n)\phi^{\mathrm{T}}\hat{\theta} + \hat{\theta}^{\mathrm{T}}\phi(n)\phi^{\mathrm{T}}(n)\hat{\theta}\end{aligned} \tag{1.24}$$

由此可得期望为

$$E[\epsilon^2(n)] = E[y^2(n)] - 2\hat{\theta}^{\mathrm{T}}E[y(n)\phi(n)] + \hat{\theta}^{\mathrm{T}}E[\phi(n)\phi^{\mathrm{T}}(n)]\hat{\theta} \tag{1.25}$$

接下来，定义方差 $\sigma_y = E[y^2]$为均方幂，互相关矢量定义为 $p = E[y(n)\phi(n)]$。然后，定义信息矩阵为 $R = E[\phi(n)\phi^{\mathrm{T}}(n)]$，这与 1.1 节中的矩阵几乎相同。若系统为静态统计，即统计数据不变，则 σ_y、p 和 R 项为常数，而作为变化的 $\hat{\theta}$ 的函数的成本函数将呈现碗状。成本函数 $J(n)$可写为

$$J(n) = \sigma_y^2 - 2\hat{\theta}^{\mathrm{T}}p + \hat{\theta}^{\mathrm{T}}R\,\hat{\theta} \tag{1.26}$$

同样，正如在式（1.4）中所述，为得到最佳的参数估计 $\hat{\theta}$ 以使得成本函数最小，需取成本函数 $J(n)$相对于 $\hat{\theta}$ 的偏导数，并确定使偏导数为零的 $\hat{\theta}$ 值。$J(n)$的偏导数可表示为

$$\frac{\partial J(n)}{\partial \hat{\theta}} = \frac{\partial \sigma_y^2}{\partial \hat{\theta}} - 2\frac{\partial \hat{\theta}^{\mathrm{T}}p}{\partial \hat{\theta}} + \frac{\partial \hat{\theta}^{\mathrm{T}}R\,\hat{\theta}}{\partial \hat{\theta}} \tag{1.27}$$

然后计算式（1.27）中右侧每一项的偏导数。分别计算每项，可得

$$\frac{\partial \sigma_y^2}{\partial \hat{\theta}} = 0$$

$$2\frac{\partial \hat{\theta}^{\mathrm{T}} p}{\partial \hat{\theta}} = 2p$$

$$\frac{\partial \hat{\theta}^{\mathrm{T}} R\,\hat{\theta}}{\partial \hat{\theta}} = 2R\,\hat{\theta} \tag{1.28}$$

代入式（1.27），可得

$$\frac{\partial J(n)}{\partial \hat{\theta}} = -2p + 2R\,\hat{\theta} = 0 \tag{1.29}$$

求解 $\hat{\theta}$，即可得到参数估计的最优解为

$$\theta^{*} = R^{-1}p \tag{1.30}$$

式（1.30）即著名的维纳（Wiener）解。但式（1.30）中的维纳解需要计算大矩阵 R 的逆矩阵。值得注意的是，维纳解与式（1.6）中的 LS 解非常相似。若要估计式（1.25）中的期望值，则可通过计算下式得到平均值：

$$R_{\mathrm{avg}} = \left[\frac{1}{N}\sum_{n=1}^{N}\phi(n)\phi^{\mathrm{T}}(n)\right]$$

$$p_{\mathrm{avg}} = \left[\frac{1}{N}\sum_{n=1}^{N}\phi(n)y(n)\right] \tag{1.31}$$

将上述值代入式（1.30），即可得到式（1.6）所给出的 LS 解。本质上，LMS 算法的维纳解和 LS 解完全相同。

在信号处理领域，尤其是自适应信号处理中，处理速度非常重要。此外，自适应信号处理中，特别是通信应用中的模型结构具有许多参数。在矢量 $\phi(n)$ 中具有 200 项的参数是很常见的，也就是说，在式（1.21）中的 $k=200$。在此情况下，矩阵 R 的大小将会是 200×200，若要求解式（1.30）中的逆矩阵，计算量非常庞大。为此，通常采用梯度最速下降法。这是一种在工程领域中普遍应用的技术，与用于求解各种函数的零点和根的著名 Newton – Raphson 方法（牛顿 – 拉夫逊迭代法）非常类似。梯度最速下降法是一种迭代方法。其基本思想是，首先从一个参数的初始假设值开始：为简单起见，通常选择为零。在信号处理的术语中，该参数称为抽头权重。然后，不断迭代调整参数使得成本函数沿梯度下降。设参数矢量的当前估计值为 $\phi(\mathrm{now})$，接下来，计算参数矢量的下一值：

$$\hat{\theta}(\mathrm{next}) = \hat{\theta}(\mathrm{now}) - \mu g \tag{1.32}$$

式中，g 是由成本函数对式（1.29）中定义的参数估计矢量 $\hat{\theta}$ 求导所得的梯度值，将其代入式（1.32），可得

$$\hat{\theta}(\mathrm{next}) = \hat{\theta}(\mathrm{now}) - \mu 2p - \mu 2R\hat{\theta}(\mathrm{now}) \tag{1.33}$$

以递归形式可表示为

$$\hat{\theta}(n+1) = \hat{\theta}(n) - \mu 2p - \mu 2R\,\hat{\theta}(n) \tag{1.34}$$

也可将式（1.34）写为

$$\hat{\theta}(n+1)=(I-\alpha R)\hat{\theta}(n)-\alpha p \tag{1.35}$$

式中，$\alpha=2\mu$。

由系统理论可知，如果（$I-\alpha R$）的特征值小于 1，则式（1.35）将会递归收敛。这是对最速下降法中步长的一种限制。在 1.4 节中介绍随机逼近法时将会继续讨论这一点。步长是机器学习算法的一个重要参数。

在式（1.34）中递归计算的难度在于计算统计项 R 和 p，其中，R 是信息矩阵或自相关矩阵，p 是互相关矩阵。这些矩阵的统计数据往往是未知的，必须利用式（1.31）进行估计。然而，这些估计值计算强度大，且需要直到 N 个数据点均采集完才能计算。而在 LMS 算法中提出了一种基于每次取样时刻的单个数据点来估计这些矩阵的方法：

$$\begin{aligned}\hat{R}(n)&=\phi(n)\phi^{\mathrm{T}}(n)\\ \hat{p}(n)&=\phi(n)y(n)\end{aligned} \tag{1.36}$$

该方法有时也称为脏梯度法或随机梯度法。具体思想是，只需沿梯度的大致方向下降，而无需完全沿着梯度方向。想象下山的场景，可以直接下山，如果非常陡峭，也可像滑雪者一样选择迂回穿越的方式。无论采用何种方式，最终会到达山脚下。在此，将式（1.36）给出的 $\hat{R}$ 和 $\hat{p}$ 估计值代入式（1.34）的递归方程中，可得

$$\hat{\theta}(n+1)=\hat{\theta}(n)+2\mu\phi(n)y(n)-2\mu\phi(n)\phi^{\mathrm{T}}(n)\hat{\theta}(n) \tag{1.37}$$

此时，可因式分解出 $2\mu\phi(n)$，并得到标准的 LMS 递归算法：

$$\hat{\theta}(n+1)=\hat{\theta}(n)+2\mu\phi(n)(y(n)-\phi^{\mathrm{T}}(n)\hat{\theta}(n)) \tag{1.38}$$

切记上述等式中右边括号项中的 $y(n)-\phi^{\mathrm{T}}(n)\hat{\theta}(n)$ 为预测误差或新值。$\phi(n)\hat{\theta}(n)$ 项为输出 $y(n)$ 的当前预测值。比较式（1.20）和式（1.38）中的 RLS 算法，可知更新形式也类似。参数的更新是通过将前一估计值与矩阵矢量和预测误差的乘积相加而实现的。事实上，这表明在固定点或协方差矩阵更新时的值取式（1.20）中 $P(n+1)=P(n)$ 时，LMS 算法等效于特定参数集时的 RLS 算法。

目前已有关于 LMS 算法的各种实现和收敛结果分析的大量文献，但本书的重点是通过机器基于已有实验数据和由 $y(n)$ 提供的正确值知识来学习系统预设模型的参数。新的参数是由原有参数加上数据和预测输出中已知误差的乘积而形成的矢量获得的。

1.4　随机逼近法

随机逼近法是一种用于系统辨识的老方法。事实上，这是一种与 RLS 和