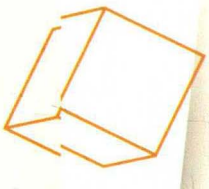
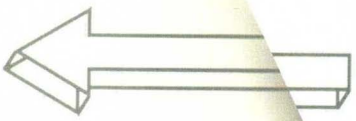
张雨萌◎编著

机器学习线性代数基础

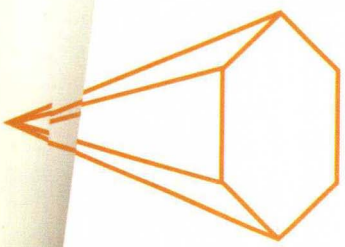
Python语言描述





60段代码 + 53幅图表 + 2个项目

帮助你理解线性代数与机器学习紧密结合的最核心内容

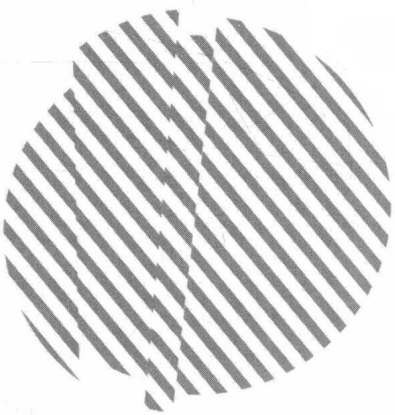


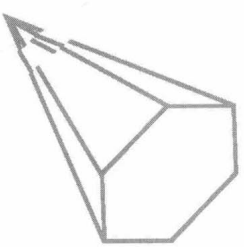
北京大学出版社
PEKING UNIVERSITY PRESS

张雨萌◎编著

机器学习线性代数基础

Python语言描述





北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

数学是机器学习绕不开的基础知识，传统教材的风格偏重理论定义和运算技巧，想以此高效地打下机器学习的数学基础，针对性和可读性并不佳。本书以机器学习涉及的线性代数核心知识为重点，进行新的尝试和突破：从坐标与变换、空间与映射、近似与拟合、相似与特征、降维与压缩这5个维度，环环相扣地展开线性代数与机器学习算法紧密结合的最核心内容，并分析推荐系统和图像压缩两个实践案例，在介绍完核心概念后，还将线性代数的应用领域向函数空间和复数域中进行拓展与延伸；同时极力避免数学的晦涩枯燥，充分挖掘线性代数的几何内涵，并以Python语言为工具进行数学思想和解决方案的有效实践。

本书适合实践于数据分析、信号处理等工程领域的读者，也适合在人工智能、机器学习领域进行理论学习和实践，希望筑牢数学基础的读者，以及正在进行线性代数课程学习的读者阅读。

图书在版编目(CIP)数据

机器学习线性代数基础 : Python语言描述 / 张雨萌编著. -- 北京 : 北京大学出版社, 2019.9

ISBN 978-7-301-30601-7

Ⅰ. ①机… Ⅱ. ①张… Ⅲ. ①机器学习 Ⅳ. ①TP181

中国版本图书馆CIP数据核字（2019）第148115号

书　　名　机器学习线性代数基础：Python语言描述

JIQI XUEXI XIANXING DAISHU JICHU : PYTHON YUYAN MIAOSHU

著作责任者　张雨萌 编著

责任编辑　吴晓月 王继伟

标准书号　ISBN 978-7-301-30601-7

出版发行　北京大学出版社

地　　址　北京市海淀区成府路205 号　100871

网　　址　http：//www. pup. cn　　新浪微博：@ 北京大学出版社

电子信箱　pup7@ pup. cn

电　　话　邮购部 010-62752015　发行部 010-62750672　编辑部 010-62570390

印 刷 者　大厂回族自治县彩虹印刷有限公司

经 销 者　新华书店

787毫米×1092毫米　16开本　10.75印张　254千字

2019年9月第1版　2019年9月第1次印刷

印　　数　1—4000册

定　　价　49.00 元

前言

INTRODUCTION

为什么要写这本书?

当下，机器学习、人工智能的火爆程度无须多言。薪酬之高、职业发展道路之宽阔，吸引着大量优秀的学子投身于这个领域进行学习和探索。

然而与较为基础的程序语言、编程框架的学习不同，机器学习是一个更为综合的学科领域，一个零基础的学生想要涉足该领域是有一定难度的，因为机器学习需要有大量的前序知识作为铺垫，其中最核心的基础知识就是以线性代数、概率统计等为代表的数学知识和思想方法。

线性代数作为利用空间来投射和表征数据的基本工具，可以方便地对数据进行各种变换，从而让研究人员更加直观、清晰地探查到数据的主要特征和不同维度的所需信息，因此线性代数的核心基础地位不言而喻。只有熟练地运用好这个工具，才能为自己搭建起攀登机器学习高峰的牢固阶梯。

初学者可能会问，在机器学习、数据分析中，有哪些地方需要用到线性代数呢？我们举例如下：

（1）如何定量地描述日常生活中的事物，如个体的不同属性、自然语言中的词语和句子等，来支撑我们所要进行的算法分析？

（2）如何将待处理的数据在不同维度的空间中进行变换处理，以寻找到最佳的观测角度，使数据处理达到最好的效果？

（3）如何从采样的海量数据中提取主要特征成分，梳理出数据的主要脉络，从而协助我们对一个文本进行主题建模，并利用协同过滤技术，成功地给用户推荐他们感兴趣的东西？

（4）如何用数字表示图像，并且在不太影响观察效果的前提下，利用很小的存储空间就能近似地达到原有图像的视觉效果？

（5）如何对采集到的观测数据进行拟合，以帮助我们找到其中暗含的规律，对未知数据进行

预测？

（6）在实际的数据采样分析过程中，如何在无法找到精确解的情况下，探索出最接近真相的近似解？

……

这些实用而有趣的问题，我们在数据分析和机器学习中，几乎时时刻刻都会遇到。想要解决好这些问题，线性代数的核心概念和思想方法都必须牢固掌握，而这也正是写这本书的目的之所在。

本书有何特色？

既然已经明确了线性代数的核心基础地位，那么很多读者一定摩拳擦掌，准备大干一场。但是，翻开许多线性代数教材，读者就会感觉有些迷茫，因为从教材里似乎很难找到解决上述实际问题的有效方法。

有些读者在学习时会有这样一种感觉：一门课学完了、考试过了，却不知道学了有什么用，尤其是数学类的课程。因为传统教材大多数是按照“定义—例题—计算”的步骤来大篇幅罗列数学概念，偏重理论定义和运算技巧，不注重梳理学科内在的逻辑脉络，更没能深刻挖掘出本学科与当下前沿技术的交汇点。传统教材往往应付考试有余，但想以此高效地打下机器学习的数学基础，效果并不理想。

明确了不足，本书就将在传统教材的薄弱环节做出突破，设计一条有针对性的学习路径。

一方面，紧紧围绕**空间变换**这个线性代数的主要脉络，从**坐标与变换、空间与映射、近似与拟合、相似与特征、降维与压缩**这 5 个维度，环环相扣地展开线性代数与机器学习紧密结合的核心内容，深刻阐述如何**用空间表示数据、用空间处理数据、用空间优化数据**，用一条线索贯穿整个学科的主干内容。

另一方面，深度结合机器学习中的典型实战案例，面向应用，帮助读者将线性代数这一数学工具用会、用熟、用好，同时以 Python 语言为工具，进行数学思想和解决方案的有效实践，无缝对接工程应用。

本书在内容组织和知识展现方面具有以下 3 个特色。

第一，避免纸上谈兵。全书以 Python 语言作为工具进行概念和方法的有效实践，无缝对接机器学习工程应用，可操作性强。

第二，避免生硬枯燥。全书务求结合线性代数的几何意义，对重点概念进行剖析和演绎，避

免传统教材的既视感，强化逻辑性和可读性。

第三，避免大水漫灌。全书以机器学习所亟须的线性代数内容为立足点，讲解相关知识，从而使读者提高学习效率。

本书内容及知识体系

全书内容安排如下。

第 1 章　坐标与变换：高楼平地起。从空间坐标表示与线性变换入手，快速建立线性代数直观感受，理解向量和矩阵运算的几何本质。

第 2 章　空间与映射：矩阵的灵魂。围绕线性代数的概念基石——空间，详细阐述空间中映射和变换的本质，深入剖析矩阵在其中的灵魂作用。

第 3 章　近似与拟合：真相最近处。展现线性代数在近似与拟合中的理论基础，并阐述最小二乘法的实际应用。

第 4 章　相似与特征：最佳观察角。重点分析矩阵的相似性及特征的提取方法，打好数据降维的理论基础。

第 5 章　降维与压缩：抓住主成分。作为全书知识脉络的交汇，讲解如何对数据进行降维和特征分析，深入剖析矩阵分析的核心内容：特征值分解和奇异值分解。

第 6 章　实践与应用：线代用起来。展现线性代数在推荐系统、图像压缩分析中的实际应用。

第 7 章　函数与复数域：概念的延伸。帮助读者将线性代数的核心概念向函数空间和复数域中进行延伸和拓展，在概念的比较过程中，实现对线性代数领域更为深刻和广阔的认知。

适合阅读本书的读者

- **实践于数据分析、信号处理等工程领域的读者。**本书中所着重强调的思维逻辑和处理方法将会给你们提供一种新的视角和启发。
- **在人工智能、机器学习领域进行理论学习和实践，希望筑牢数学基础的读者。**无论你是在校园学习还是已经走上工作岗位，都将获得很大的收获和共鸣。
- **正在进行线性代数课程学习的读者。**阅读本书有利于你们对线性代数产生更浓厚的兴趣、多角度的认识和更深层的思考，会收获同学习传统教材不一样的思维体验。

阅读本书的建议

- 建议结合线性代数的几何意义进行数学概念的理解。
- 建议利用 Python 语言进行数学知识的学习以提高实践能力。
- 对于案例章节，建议先思考一下解决的方法，再与书中的代码内容进行对照学习。
- 建议仔细体会全书的知识脉络，以建立更好的数学思维和感觉。

本书所涉及的源代码已上传到百度网盘，供读者下载。请读者关注封底“博雅读书社”微信公众号，找到“资源下载”栏目，根据提示获取。

第1章

坐标与变换：高楼平地起

第 1 章

坐标与变换：高楼平地起

作为全书内容的开篇部分，本章将从空间的角度出发，详细介绍向量和矩阵的基本概念，并在空间思维的框架下描述矩阵和向量运算的基本法则，揭示其几何意义，以求迅速帮助读者搭建起关于空间的宏观知识框架，奠定学习全书内容的思想方法。

在本章知识内容的演绎、推进过程中，会逐步引出基底的选取、空间的张成、基底的转化与坐标的变换这些和空间紧密相关的概念，并使用 Python 语言，对相关概念和运算过程进行描述。

本章主要涉及的知识点

- 介绍向量的概念和基本运算
- 介绍基底的用途和构成条件
- 介绍坐标与基底之间的关系
- 介绍矩阵的概念和基本运算
- 介绍矩阵对向量空间位置的改变
- 介绍在矩阵的作用下，向量的基变换原理及过程

1.1 描述空间的工具：向量

空间是贯穿整个线性代数的主干脉络和核心概念。那么在全书开篇的第一节，我们将重点学习如何利用向量这个重要工具对空间进行描述，从而使读者完成对“空间”从感性认识到定量描述的重要转变。

首先，我们将在向量知识基础上，开始学习行向量及列向量的基本概念，并且运用 Python 语言对向量进行代码表示，这也是本书的一个重要特色；然后，我们会利用 Python 语言熟悉和掌握如何对多个向量进行加法和乘法运算；最后，综合以上的这些知识和运算法则，引出向量线性组合的重要概念，使读者了解线性组合的构成方法和基本形式。

1.1.1 重温向量

对于向量而言，我们一定不会感到陌生。向量的概念其实很简单，直观地说，把一组数字排列成一行或一列，就称为向量。它可以作为对空间进行描述的有力工具。

例如，对于一个简单的二维向量$\begin{bmatrix}4\\5\end{bmatrix}$，这个向量有两个成分：第一个成分是数字 4，第二个成分是数字 5。

向量$\begin{bmatrix}4\\5\end{bmatrix}$可以理解为二维平面中 x 坐标为 4、y 坐标为 5 的一个点，也可以将其理解为以平面中的原点 (0, 0) 为起点，以 (4, 5) 为终点的一条有向线段，如图 1.1 所示。

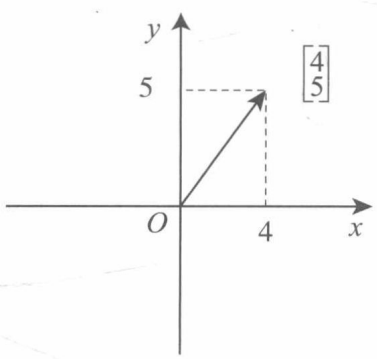


图 1.1 二维向量的空间表示

由此可见，一个向量中成分的个数就是该向量的维数。因此，如果进一步推广下去，还会有三维向量，如$\begin{bmatrix}3\\2\\4\end{bmatrix}$。同理，这个三维向量可以用来表示三维空间中的一个指定点，或者用来表示在三

维空间中以原点 (0, 0, 0) 为起点，以 (3, 2, 4) 为终点的一条有向线段，如图 1.2 所示。

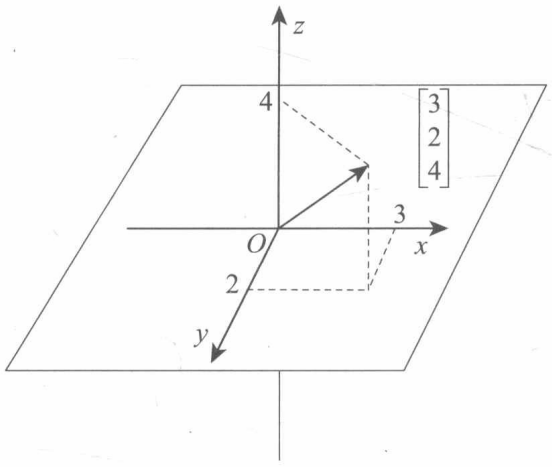


图 1.2　三维向量的空间表示

当然，以此类推，自然还存在更高维的向量，只不过不太好利用图形化的方式进行描述，这里就不继续展开举例了。

不过向量也不仅局限于用来直接描述空间中的点坐标和有向线段，也可以凭借基础的数据表示功能，成为一种描述事物属性的便捷工具。例如，在一次考试中，你的考试成绩为：语文 85 分，数学 92 分，外语 89 分。由于这 3 门课具有不同科目属性，因此，可以使用一个三维向量来对其进行表示，即$\text{score} = \begin{bmatrix} 85 \\ 92 \\ 89 \end{bmatrix}$。其实，这样看来，此时不仅仅可以把向量 score 看作是一个盛放数据的容器，似乎也可以利用它将科目考试成绩和空间建立起某种关联。

又如，在自然语言处理的过程中，也少不了向量这个重要的工具。程序进行文本阅读时，首先会对文本材料进行分词处理，然后使用向量对词汇进行表示。这是因为向量很适合将对象的属性和特征对应到高维空间中进行定量表达，同时在此基础上进行进一步的后续处理，如判断词汇之间的相似性等。

在本书的后续章节中，将会陆续接触到一些数据处理的基本方法：如投影、降维等，这些方法都是在向量描述的基础上实现的。

1.1.2　通常使用列向量

根据上面所讲述的向量的定义“把一组数字排列成一行或一列，就称为向量”，向量对应地就拥有两种表达方式：如果元素是纵向排列的，就将其称为列向量，如$\begin{bmatrix} 4 \\ 5 \end{bmatrix}$，$\begin{bmatrix} -4 \\ 15 \\ 6.7 \end{bmatrix}$；如果元素是横向

排列的，就将其称为行向量，如 [4 5 7]。

在实际使用向量工具进行描述和计算时，应该具体使用哪一种方式呢？在没有特殊说明的情况下，一般都默认为列向量。

从直觉上来看，似乎行向量显得更为直观，但是，这里为什么会如此偏爱列向量呢？这么做主要是为了方便后续的向量坐标变换、空间之间的映射等计算过程的处理。

在这里先不详细展开讨论，读者对此有一个直观的印象就可以了。将一个矩阵 $\boldsymbol{A}$ 所表示的映射作用于某个向量 $\boldsymbol{x}$ 上时，习惯上将其写成矩阵乘以向量的表达形式，即 $\boldsymbol{Ax}$。而这种写法的数据表示基础便是：向量 $\boldsymbol{x}$ 必须是一个列向量。

目前出现好几个概念，如转置、矩阵、映射等，这里先不做介绍，后面会一一详细描述。需要记住的是：一般都用列的形式来表示向量。

1.1.3 使用 Python 语言表示向量

了解了基本概念后，开始使用工具。对应地，应如何使用 Python 语言表示行向量和列向量呢？这里，需要使用 Python 语言中的一个常用工具库：numpy。先看如何用代码描述行向量 $\boldsymbol{a}$ = [1 2 3 4]。

代码如下：

```
import numpy as np
a = np.array([1, 2, 3, 4])
print(a)
```

运行结果：

```
[1 2 3 4]
```

在 Python 语言中，一般使用工具库 numpy 来生成一个向量，但其默认生成的是行向量。但正如前面内容中所介绍的，一般情况下，通常使用列向量的形式，因此还需要对其做一些处理工作。

也许有些读者会想，用转置这个概念（后面会详细讲解）是不是就可以了，也就是把向量的行索引和列索引交换位置。但是 numpy 中的转置方法对于一维数组是无效的，代码如下。

```
import numpy as np
a = np.array([1, 2, 3, 4])
print(a.transpose())
```

运行结果：

```
[1 2 3 4]
```

从程序的运行结果来看，这段代码确实没有出现预期的效果。那应该如何表示一个列向量 $\begin{bmatrix}1\\2\\3\\4\end{bmatrix}$ 呢？具体的做法我们来演示一下。