MLAPP-C1

炽聚AI

1. 引言

1.1 机器学习：是什么和为什么？

1.1.1 机器学习的分类

1.2 监督学习

1.2.1 分类

1.2.2 回归

1.3 非监督学习

1.3.1 发现聚类

1.3.2 发现潜在因子

1.4 机器学习中的一些基本概念

1.4.1 参数模型和非参数模型

1.4.2 一个简单的非参数分类器：K近邻（K-nearest neighbors）

1.4.3 维度灾难

1.4.4 分类和回归中的参数模型

1.4.5 线性回归

1.4.6 逻辑回归

1.4.7 过拟合

1.4.8 模型选择

1.4.9 没有免费的午餐理论

1. 引言

1.1 机器学习：是什么和为什么？

We are drowning in information and starving for knowledge. ——John Naisbitt.

我们正在进入大数据时代。举例来说，在网络上有近1万亿张网页；每分钟有近100小时的视频上传至Youtube；10世纪人的基因组，每一个都有长度达3.8×109的基本基因对，其测序工作已经被不同的实验室完成；沃尔玛每小时完成1M的交易，并且拥有包含超过2.5×1015字节信息的数据库。

**数据的泛滥要求有对数据进行自动分析的方法**，**这便是机器学习所能解决的问题。**我们定义**机器学习（**Machine Learning**）为一系列可以自动挖掘数据中潜在模式的方法**，并且利用发现的模式去预测未来的数据，或者在不确定的情况下执行其他的决策（比如计划如何获取更多的数据！）

全书基于如下观点:解决这类问题（探索数据中的模式）的最好方法是使用概率论中的工具。概率论可以应用在任何涉及**不确定度**的问题上。在机器学习中，不确定度以很多形式出现：在给定历史数据的前提下未来最有可能出现的数据是什么？能够解释某些数据的最好的模型是什么？在机器学习中所使用的概率论相关的方法与统计学领域有密切的关联，但是两者的侧重点和一些术语上存在些许不同。

本书将介绍多种概率模型，它们分别适用于不同类型的数据和任务。我们也会描述各种机器学习算法并且使用这些算法解决一些问题。我们的目的并不是简单的提供一本关于一项特殊技术的工具书，反之，我们希望透过概率模型和概率推理的视角来展示关于这个领域的统一的观点。尽管我们会关注算法的计算效率，但是将这些算法更好地应用在真正大规模数据集上的细节部分最好还是参考其他的书籍。

然而需要注意的是，即使某人拥有一个表面上十分庞大的数据集，但对于某些特定感兴趣的情况，其中有效的数据量可能也是相当小的。事实上，在很多领域中，数据都呈现出一种**长尾**（long tail）效应，意味着经常发生的那些事件往往是很少的，大部分事件是很少发生的。比如说，我们经常遇到的那些单词往往也就是那么一小部分（比如“the”和“and”），但是大部分单词（比如“pareidolia”）却很少使用。类似的，有些电影和书很有名，但大部分并非如此。章节2.4.7将进一步介绍长尾效应。长尾效应所导致的一个结果是：**我们只需要较少的数据就可以预测或者理解大部分行为**。本书主要讨论处理这种数据集的技术。

1.1.1 机器学习的分类

机器学习一般分为2大类。在**预测**（predictive）或者**监督**（supervised）学习方法中，模型的目的是在给定标记数据集（标记数据集是指针对每一个样本**x**，都赋予一个标签值y）的情况下，通过学习得到一个从输入**x**到输出y的映射。其中D被称为**训练集**（training set），N为训练集中样本的数目。

从最简单的情况出发，假设训练集中的每一个输入**x**i是一个D维向量，向量中的每一个分量分别代表一个人的身高和体重。这些分量被称为**特征**（features）、**属性**（attributes）或者**协变量**（covariates）。然而在通常情况下，输入向量**x**i可以是一个复杂的结构化对象，比如一张图片，一个句子，一条邮件，一个时间序列，一个分子形状，一个图表等等。

类似的，模型的输出或者**响应**（response）变量的形式原则上也可以是多样的，但是在大部分方法中，都假设yi是一个**类别**（categorical）或者**名义**（nominal）变量，变量取值于一个有限的集合yi∈{1,…,C}（比如男性或者女性类别），或者yi是一个属于实数域的标量（比如一个人的收入）。当yi是一个类别变量，相应的问题被称为**分类**（classification）或者**模式识别**（pattern recognition），当yi是一个属于实数域的标量，相应的问题被称为**回归**（regression）。回归问题的一个变种是**有序回归**（ordinal regression），这种情况出现在需要输出多个标签且标签之间具备一些自然的顺序，比如等级A-F。

机器学习中的另一类方法被称为**描述性**（descriptive）或者**无监督**（unsupervised）学习方法。在这种情况下，模型给定的输入为，模型的目标是在数据中找到“感兴趣的模式”。有时也称这个方法为**知识发掘**（knowledge discovery）。无监督学习是一个不够明确的问题，因为我们并没有被告知发掘什么类型的模式，所以也没有明确的评价标准可以使用（不像监督学习，我们可以将模型的预测值与观察到的真实值作对比）。

在机器学习中还有一类被称为**强化学习**（reinforcement learning），在某种程度上很少被使用。在随机的奖励或者惩罚信号的情况下，模型可以更好地学习采取什么的行为。（比如说，考虑一个儿童如何学习走路。）不幸的是，强化学习的内容已经超出本书范畴，尽管我们在5.7节讨论了决策论的内容（强化学习的基础）。

1.2 监督学习

我们从监督学习开始对机器学习展开探索，它在实际过程中被广泛使用。

1.2.1 分类

本节我们将讨论分类。分类的目的是通过算法学习得到一个从输入**x**到输出y的映射，其中y∈{1,…,C}，C为类的数量。如果C=2，则被称为**二分类**（binary classification）（在这种情况下，我们通常假设y∈{0,1}）；如果C>2，则被称为**多分类**（multiclass classification）。如果类的标签彼此之间互不排斥（比如某人可以被归类为高的和强壮的），我们称它为**多标签分类**（multi-label classification），当然更好的方式是将它理解为预测多个相关二元类标签的问题（即所谓的多变量输出模型）。当我们使用术语“分类”时，我们是指输出单个变量的多分类问题，除非我们特别做出说明。

一种关于这个问题的形式化表达是**函数近似**（function approximation）。假设存在一个未知函数f，使得，我们的目的是在给定一个含标签的训练集的基础上估计函数f的形式，然后使用得到的函数f进行预测。（我们使用**^**符号表示估计量）。模型最终的目的是为了在新的输入下，即对我们还未观察到的输入进行预测（**泛化 generalization**），因为基于训练集进行输入的预测是一件十分容易的事（我们可以仅仅通过查表的方式找到答案）。

* + - 1. 举例

举一个分类问题的小例子，考虑如图1.1（a）所示的问题。我们有两类对象，分别对应标签0（no）和1（yes）。输入为彩色的形状，每个输入通过D个属性或特征进行描述，所有输入的属性可以被存储在一个大小为N×D的设计矩阵X中，如图1.1（b）所示。输入特征x的取值可以是离散的，连续的，或者是两者的组合。除了输入，在训练集中，包含一个标签向量**y。**

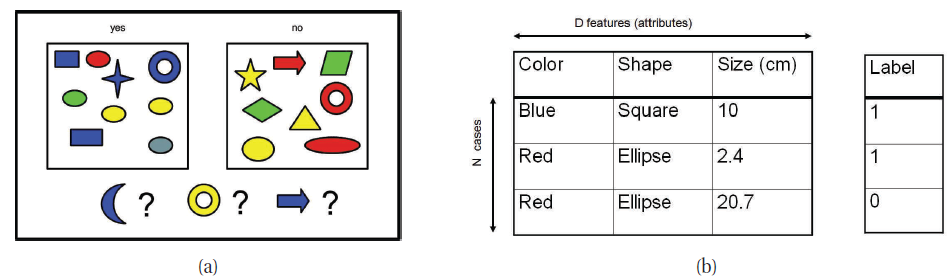


图1.1 左：一些含标签的训练样本和3个未标记的测试案例；右：表示训练样本的设计矩阵。每一行代表一个特征向量，最后一列代表标签。

在图1.1中，测试样本为一个蓝色的新月，一个黄色的圆环和一个蓝色的箭头。这些形状在训练集中都没有出现过，所以我们需要模型具备泛化能力，从而在新的数据上也能表现良好。一个合理的猜测是蓝色的新月应该属于y=1，因为在训练集中所有的蓝色形状都被标记为1。黄色的圆环很难被分类，因为在训练集中有些黄色被标记为1，有些则被标记为0，有些圆被标记为1，有些圆被标记为0。因此很难说黄色的圆环应该属于哪一类。类似的情况在蓝色箭头上一样适用。

* + - 1. 概率性预测的必要性

为了处理如上文所提及的预测结果比较模糊的情况（比如黄色的圆环属于哪一类），可以利用概率论中的相关知识。我们假设读者已经具备一定的概率论的相关知识。当然，我们也将会在第二章进行相关知识的回顾。

在给定训练集D的情况下，输入向量x所属的类y服从的分布由p(y|**x**,D)表示。通常情况下，这个概率分布的取值有C个（即y的取值有C种，在二分类问题中，y的取值只有两种，此时我们只需要关心p(y=1|**x**,D)的取值即可，因为p(y=1|**x**,D)+p(y=0|**x**,D)=1）。在我们的符号表达中，我们明确概率分布以测试输入（新的输入）**x**和训练集D作为条件，在形式上表现为**x**，D被放置在符号“|”后面。当然，在上述的概率分布中我们隐藏了概率分布所基于的模型种类，当我们的预测是基于多种模型时，我们会显式地表达出概率分布对模型种类的依赖p(y|**x**,D,M)，其中M表示模型的种类。然而，如果模型的种类在上下文中已经很明确，我们在公式中将不再书写。

如果已知上文中所提及的概率分布p(y|**x**,D)，那么我们总可以使用下式作出最好的预测，并将其作为真实的标签输出：

 （1.1）

式中估计量是输入**x**最有可能的类的标签，同时也被称为分布p(y|**x**,D)的**众数**（mode）（在一个概率分布中发生频率最高的情况）；上式又被称为**最大后验估计**（MAP estimate）。使用最有可能的输出作为最终的选择从直觉上是正确的，我们将在5.7节给出更加正式的证明。

现在回到之前的例子中，假设我们对黄色圆环的标签存在一个估计值，但其概率值远离1。在这种情况下，我们对这个选择并不是很确定，所以相较于给出这个不信任的答案，直接指出“我不知道”可能会更好。这在医疗或者金融等需要规避风险的领域尤其重要。

* + - 1. 现实世界中的应用

分类可能是在机器学习中最广泛应用的形式，已经被用来解决很多有趣的但在现实生活中很难解决的问题。接下来，我们将给出更多的例子。

**文本分类和垃圾邮箱过滤**

在**文本分类**（**document classification**）问题中，我们的目标是将一个文本分类成C类中的一种，这里的文本可以是一个网页或者一个邮件信息，从概率的角度分析，即需要计算，其中**x**为某个文本的向量化表示。在文本分类问题中，有一类问题被称为**垃圾邮件过滤**（email spam filtering），在这类问题中，y的取值可以是1（垃圾邮件）或者0（非垃圾邮件）。

大部分分类器都假设输入向量**x**的大小是固定的。一种将变长文本转化为固定长度的特征向量的方法是**词袋法**（bag of words）。我们将在3.4.4.1节解释更多的细节，其基本的思想是如果单词j在文本i中出现，则。如果我们使用这种表达方式，我们将得到一个大小为“文本数量×单词数量”的二元矩阵（这里的二元是指矩阵中的每一个元素只能取0或者1），图1.2为我们提供了案例。

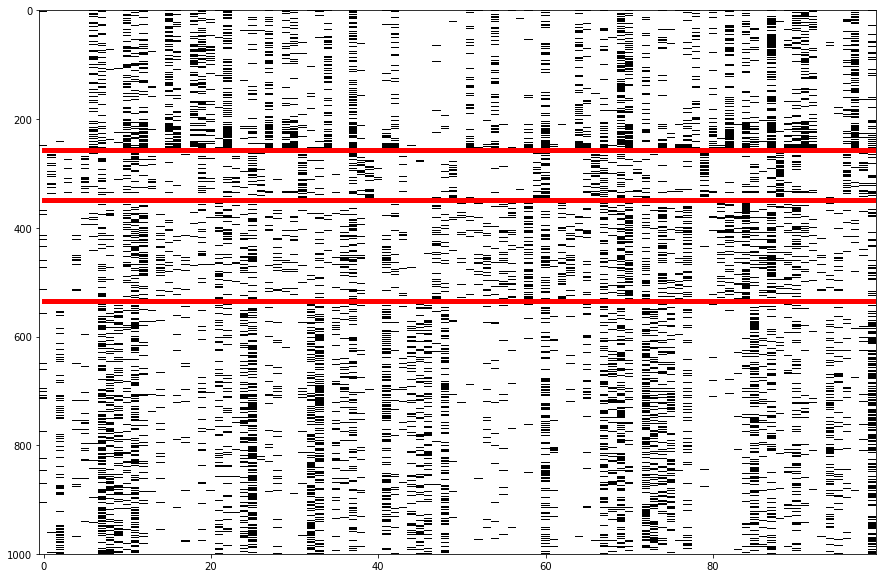


图1.2 图中每一行代表一个文本（1000个），每一列代表一个单词（100个），图中黑色表示某个单词在某个文本中出现，白色代表不出现。红色水平线将不同类别的文本分割开。不难发现，同一类别的文本的图形十分相似。图中数据可以从http://cs.nyu.edu/~roweis/data.html得到。图形由程序**newsgroupsVisualize**生成。

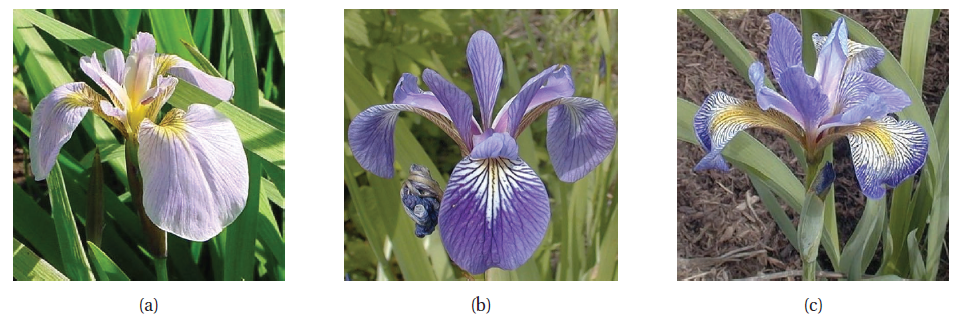


图1.3 不同种类的鸢尾花

**花朵分类**

图1.3给出了分类问题的另一个例子，其目的是区分三种不同的鸢尾花，分别为setosa，versicolor和virginica。幸运的是，我们不需要直接对图像进行分类，一个植物学家已经提取了对分类有用的4种特征：萼片长度和宽度，花瓣长度和宽度。类似于这种选择合适特征的工作被称为**特征提取**（feature extraction），这是一项特别重要而且困难的任务。大部分机器学习方法使用人工选择的特征。后面我们将讨论一些可以从数据中习得好的特征的方法。如果我们如图1.4那样绘制鸢尾花数据的散点图，不难发现，我们仅仅通过花瓣的长度或者宽度就可以将setosas（紫色圆）与其他两种区分开。然而，要想将versiclor与virginica区分开来却有一点难度；任何决定都基于至少2个特征。（在应用机器学习方法之间，对数据进行探索性的分析是十分有用的，比如绘制数据图。）

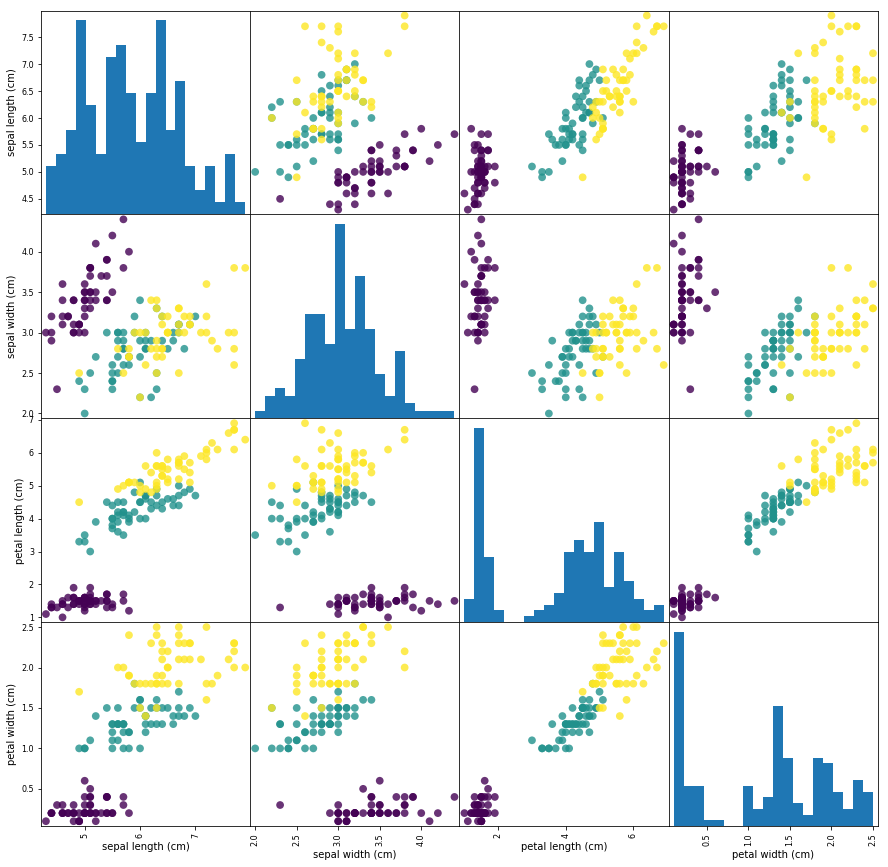


图1.4 不同鸢尾花数据的统计结果，图片由程序fisheririsDemo生成。

**图像分类和手写识别**

在花朵分类问题中，我们并未对花朵的图像直接处理，而是采用人为提取的特征进行分类。下面考虑一个更困难的问题，对图片直接进行分类，这里并不涉及对数据的预处理。我们可能需要对一张图片进行整体上的分类，比如：这张图片是室内还是室外景？它是水平还是垂直的？图中是否包含一条狗？这类任务被称为**图像分类**（image classification）。

有一种情况比较特殊，即图片中仅包含单独的手写数字和字母，比如一封信的邮政编码，我们可以通过分类实现**手写识别**（handwriting recognition）。在这个领域中所使用的标准数据集是MNIST，全称为：“Modified National Institute of Standards”（单词“Modified”是指为了使图片中的数字居中而进行了预处理。）这个数据集包含60000张训练图片和10000张测试图片，每张图片上为不同人书写的数字0-9。图片尺寸为28×28，灰度值范围为0:255。图1.5（a）展示了相关案例。

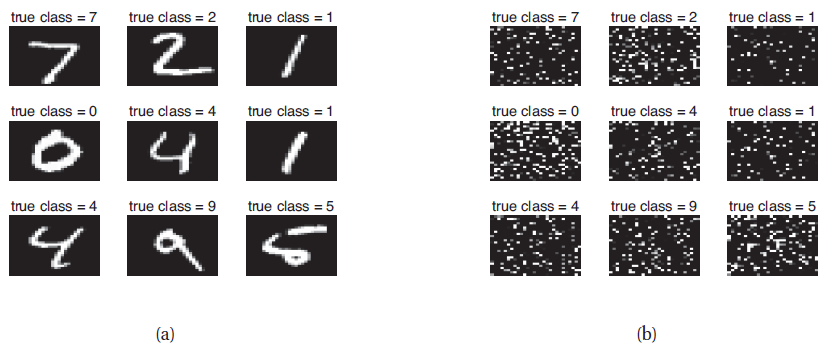
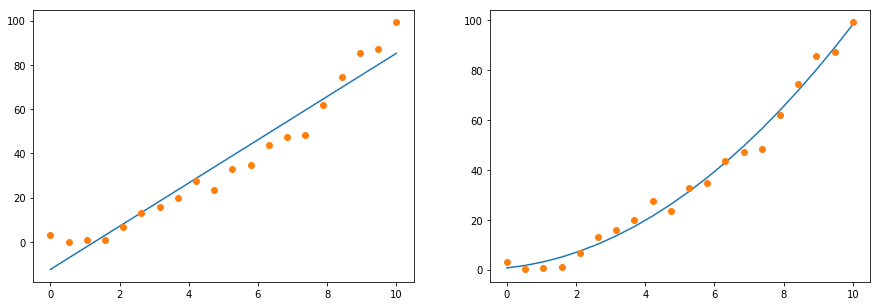


图1.5 (a) MNIST中前9张手写数字灰度图。(b)与(a)相同，但其图中像素被随机变换。在两个版本的数据集上分类的性能是一样的（训练集按照相同的方法进行像素的变换）



（a） （b）

图1.6 针对离散数据进行拟合：(a)一阶曲线拟合；(b)二阶曲线拟合。图形由程序linregPolyVsDegree生成。

1.2.2 回归

回归与分类很像，不同之处在于，回归所输出的变量是连续的。图1.6展示了一个简单的例子：我们拥有一个实值输入xi∈，和一个相应的输出yi∈。我们考虑使用两种模式去匹配这些数据：一条直线和一条二次函数曲线（我们将在后文介绍如何训练这些模型）。基于这个基本问题的展开还有很多，比如高维的输入，异常值，非光滑相应等等。我们将在后面介绍处理这些问题的方法。

以下是在现实世界中的一些回归问题。

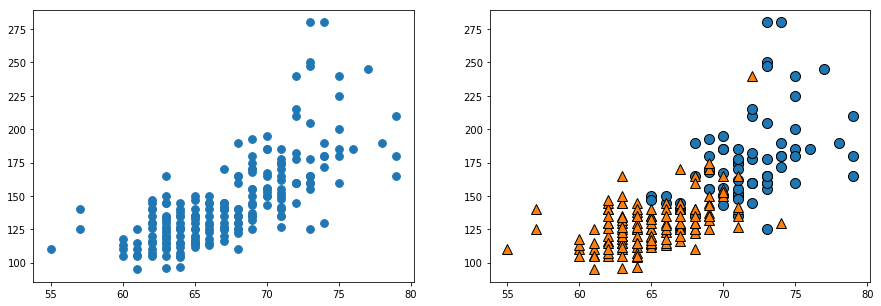
* 基于当前市场的情况和一些其他的辅助信息预测明天的股票价格；
* 预测一个观看YouTube上特定视频的观众的年龄；
* 使用天气数据，时间等信息预测某建筑内任意位置的温度。
  1. 非监督学习

我们现在考虑非监督学习，其目标是发现数据中“有趣的结构”；这通常被称为知识挖掘（knowledge discovery）。与监督学习不同，我们并没有被告知每个输入所对应的预期输出。相反，我们将无监督学习任务作为密度估计中的一种，即我们希望建立一个模型。这个形式与监督学习中的概率表达存在两个方面的不同。首先，我们用代替；即监督学习是一个条件密度估计，然而非监督学习是一个无条件概率估计。第二，**x**i是一个特征向量，所以我们需要建立一个多变量概率模型。相反，在监督学习中，yi通常只是一个单独的变量。这就意味着，在大部分监督学习中，我们可以使用单变量概率模型，从而大大简化了问题本身。

非监督学习可以说是人类和动物学习的典型方法，相较于监督学习，其使用更加广泛，因为它不需要人为的标记数据。含标签的数据不仅很难获得，同时它所具备的信息也相对较少，以至于难以用于估计复杂模型的参数。Geoff Hinton是在ML（机器学习）领域著名的教授，他说：

当我们学习观察事物时，没有人告诉我们正确的答案是什么——我们只是在观察。通常情况下，你的母亲会说“那是一条狗”，但其所具备的信息是很少的。如果你获得一些信息，哪怕一秒钟只有一比特，那也是很幸运的。人类大脑的视觉系统拥有1014个神经连接。一个人的寿命为109秒。所以每秒钟学习一个比特的信息并没有用，你需要额外的105的学习速率。拥有如此多的信息的地方只有一个，那就是输入本身。——Geoffrey Hinton,1996.

接下来，我们将描述一些在非监督学习中经典的例子。



（a） （b）

图1.7 （a）一些人的身高体重的数据；（b）当K=2时可能存在的簇。图片由程序kmeansHeightWeight绘制。

* + 1. 发现聚类

作为非监督学习中的典型案例，考虑将数据聚类为**簇**（clusters）。举例来说，图1.7（a）绘制了一些2维数据，代表210个人的身高和体重。从图中可以发现，数据中好像存在不同的簇或者子群，尽管我们并不知道这些簇到底有多少个。假设K为簇的数目。我们首先需要估计簇的数目的分布p(K|D)。为简单起见，我们通常利用p(K|D)的众数作为估计量，即。在监督学习中，我们会被告知预测值有多少个类别，但在非监督学习中，我们可以自由选择簇（类）的数目。

我们的第二个目标是估计每个数据点属于哪一个簇。令zi∈{1,…,K}代表数据点i所归属的簇。（zi又被称为隐变量或者潜变量，因为在训练集中从未见过）。我们可以通过计算来推测每个数据点所归属的簇。如图1.7（b）所示，我们使用不同的颜色代表不同的簇，其中我们假设K=2。

在本书中，我们主要关注基于模型方法的聚类，这就意味着我们利用概率模型去适应数据，而不是其他的一些算法。基于模型的方法的优势在于，我们可以客观的比较不同的模型，并且在一些大规模系统中，我们可以将不同的模型组合起来。

* 在电子商务中，我们经常会根据客户的购买行为或者网站搜索行为将它们聚类到不同簇中，然后针对不同的簇发送定制的广告。

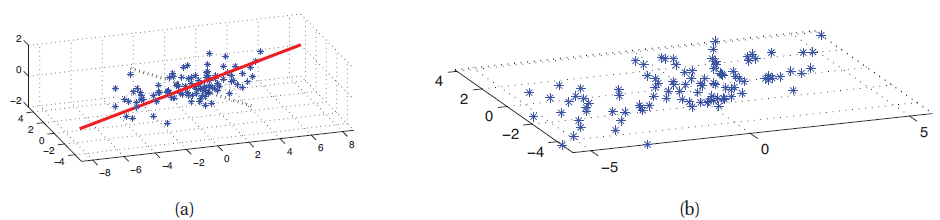


图1.8 (a)将3维空间中的数据映射到二维线性空间中，其中红色实线是第一组分方向，黑色虚线为第二组分方向。（b）数据映射在二维空间上的情况。

* + 1. 发现潜在因子

在处理维度特别高的数据时，通常需要通过将数据投影到低维空间中实现数据的降维，从而获得数据的“本质”。一个简单的例子如图1.8所示，其中我们将一些3d数据投影到一个2d平面。这个2d的近似效果很好，因为大部分点都靠近这个子空间。将数据点映射到直线上可以将数据降到1维空间，如图1.8（a）所示；这种近似的效果并不好。

降维背后的动力在于尽管实际的数据可能是高维的，但其可能只有少量的主导因素，这些潜在的主导因素被称为**潜在因子**（latent factors）。举例来说，当我们对人脸图像进行建模时，可能只有少数潜在的因子主导着不同图像的变化，比如光线，位置，身份（但是一个图像的维度却很高）等等。

将低维的向量表示作为统计模型的输入往往会取得更好的预测精度，因为输入仅仅关注了数据的“本质”，排除了那些不必要特征。同时，低维表示可以加快最近邻目标的搜索，数据在二维空间的投影可以使数据的可视化更加直观。

最常用的降维方法是主元分析（principal components analysis,PCA）。这可以被认为是线性回归的非监督版本，在PCA中，我们观察到的是最终的高维响应数据y，而不是低维的“起因”**z**，所以模型具备形式**z**→**y**，我们需要“反转这个箭头”，从观察到的高维数据**y**推断出潜在的低维数据**z**。

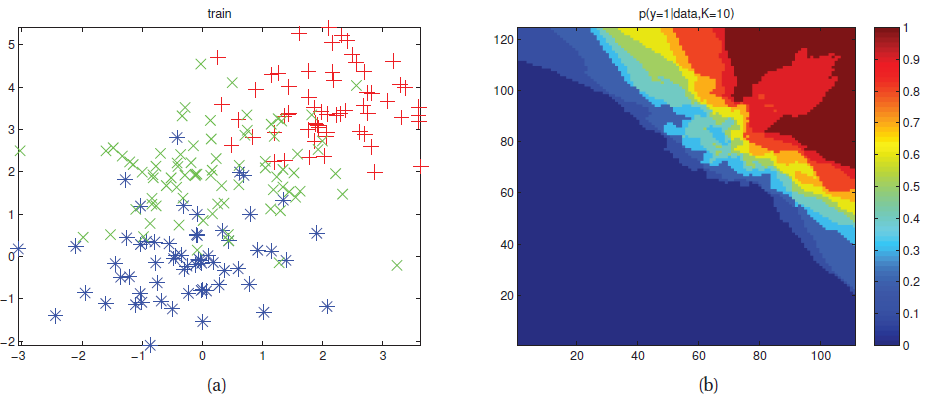
降维，尤其是PCA，已经在很多领域中得到应用。

* 1. 机器学习中的一些基本概念

本章，我们将介绍机器学习中一些核心概念。我们将在后面的内容中扩展这些概念。

1.4.1 参数模型和非参数模型

在本书中，我们将关注概率模型p(y|x)或p(x)，采用哪种模型取决于研究的问题是监督学习还是非监督学习。有很多方法来确定概率模型，不同方法中最重要的区别在于：模型是否具备固定的参数数量，或者说参数数量是否随训练样本的增加而增加？前者被称为**参数模型**（parametric model），后者被称为**非参数模型**（non-parametric model）。参数模型的优势在于使用起来更加快捷，其缺点在于它需要对数据分布作很强的假设。非参数模型则更加灵活，但是对于大的数据集而言，其计算难度通常很大。我们在后面的章节中给出两种模型的例子。为了方便，我们主要针对监督学习进行讨论，尽管我们很多的讨论同样适用于非监督学习。



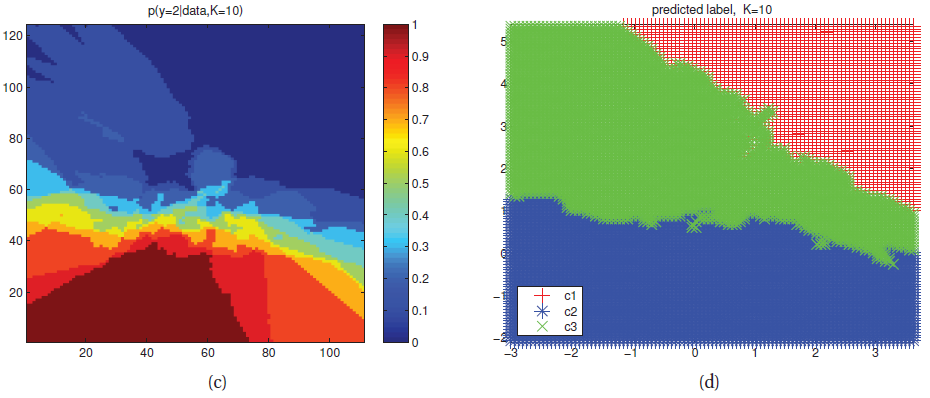


图1.9 (a)二维空间中的训练集（共3个类别）；(b)K=10时，数据属于类1的概率分布；(c)数据属于类2的概率分布；(d)类标签的最大后延概率。

1.4.2 一个简单的非参数分类器：K近邻（K-nearest neighbors）

非参数分类器的简单例子是K近邻（KNN）分类器。在这个模型中，针对测试输入**x**的分类仅仅参考距离该点最近的K个训练集中的样本，统计在这个集合（K个邻居）中，每个类的数量，返回在这个集合中每个类别的占比作为估计量，形式上表达为：

 （1.2）

其中为在数据集D中距离**x**最近的K个数据点，为指示函数，定义为：

 （1.3）

这个方法被称为**基于存储的学习**（memory-based learning）或者**基于样例的学习**（instance-based learning）。在这个方法中，最常用的距离测度为欧拉距离（该距离将这个技术限制在了实数领域），尽管其他的测度也可以使用。

图1.9 给出了这个方法在实际使用过程中的一个例子，其中输入为2维向量，所有数据共有3类，此时我们选择K=10（我们会在后面的内容中讨论K的影响）。图（a）绘制了训练数据，图（b）绘制了p(y=1|**x**,D)，图（c）绘制了p(y=2|**x**,D)，我们不需要绘制p(y=3|**x**,D)，因为每个点属于哪个类的概率满足归一化定理。图（d）绘制了最大后验概率分布。

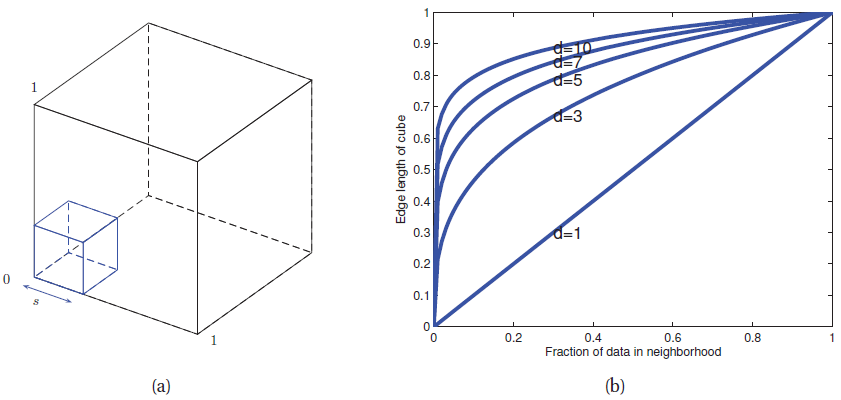


图1.10 维度灾难的示意图。(a)将一个边长为s的小立方体嵌入到一个更大的单元立方体。(b)不同维度下小立方体的边长与数据包含占比的关系。

1.4.3 维度灾难

KNN分类器十分简单且效果很好，其前提是需要一个好的距离测度以及足够多的标记数据。然而，KNN的劣势在于它不适用于高维输入。造成这个问题的原因是**维度灾难**（curse of dimensionality）。

为了解释维度灾难，我们给出一个案例。考虑一个在D维单元空间中非均匀分布的数据集，基于该数据集进行KNN的分类。为了估计测试输入**x**的类标签分布密度，我们在**x**附近建立一个超立方体，为了保证**x**的邻居数目达到一定的比例f，我们不断放大这个超立方体。考虑到整个数据集分布在一个单元空间（体积为1）中，所以当比例f满足时，小的超立方体的体积为f，那么超立方体的边长为。如果D=10（即输入特征维度为10），我们希望基于整个数据集10%的数据做密度估计，则，也就是说我们需要在每个维度扩展超立方体80%。哪怕我们只需要1%的数据量，：可以结合图1.10加深理解。因为数据在每个维度的整个范围只是1，所以当使用KNN时，这种方法不再具备局域性。换句话说，KNN需要考察最近的邻居，而当数据维度特别高时，最近的邻居实际上也很远，那么它对测试输入**x**的估计的参考价值就很低了。

1.4.4 分类和回归中的参数模型

为了解决维度灾难的问题，一种主要的方式是对数据的分布（p(y|x)或者p(x)）作出一些假设。这些假设通常以参数模型的形式展示出来（比如我们假设数据分布服从高斯分布，那么相应的模型就由参数期望和方差决定）。接下来，我们将简要的描述两种广泛使用的参数模型；我们会在后面的内容中重新接触这些模型。

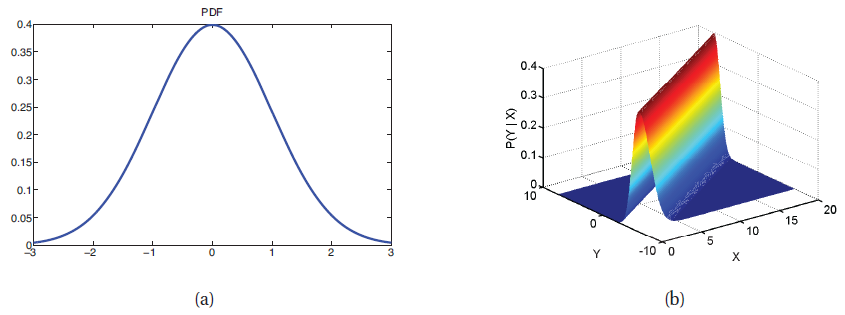


图1.11.(a)高斯分布的概率密度函数；(b)条件概率密度函数的可视化。

1.4.5 线性回归

在回归模型中最广泛使用的线性回归，在这个模型中，输出是输入的线性函数。

 （1.4）

其中代表输入向量**x**和模型权重向量**w**的内积。表示线性预测值与真实值之间的残差。

通常情况下，我们假设服从高斯或者正态分布，符号上表示为，其中表示期望，表示方差。高斯分布的图形如图1.11（a）所示。

为了建立线性回归与高斯分布之间的联系，我们重新书写模型的形式为：

 （1.5）

上式表明该模型是一个条件概率密度。在最简单的情况下，我们假设为关于输入**x**的线性函数，即，方差取固定值。在这种情况下，为模型的参数。

举例来说，假设输入是1维变量，我们可以表示期望输出为：

 （1.6）

其中w0为偏置项，w1为斜率，向量**x**=（1,x）（通过在特征中引入常数项1，可以将偏置项与其他项组合起来，这是一个常用的技巧）。如果w1>0，意味着我们期望模型的输出随着输入的增加而增加。

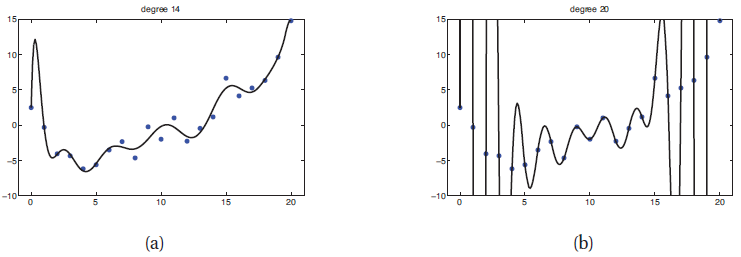


图1.12 针对21个数据点，利用阶数为14和20的多项式拟合。

线性回归也可以表达关于**x**的非线性关系，只需要将输入**x**代替为一些非线性函数，即：

 （1.7）

这被称为基函数扩展（basis function expansion）。举例来说，图1.12说明了这种情况，其中，图中分别对应d=14和d=20两种情况；这被称为多项式拟合。我们会在本书的后面部分介绍其他的基函数。事实上，许多著名的机器学习算法——比如支持向量机，神经网络，分类和决策树等等——都可以看作是从数据当中估计基函数的不同方法，我们将在14和16章讨论。

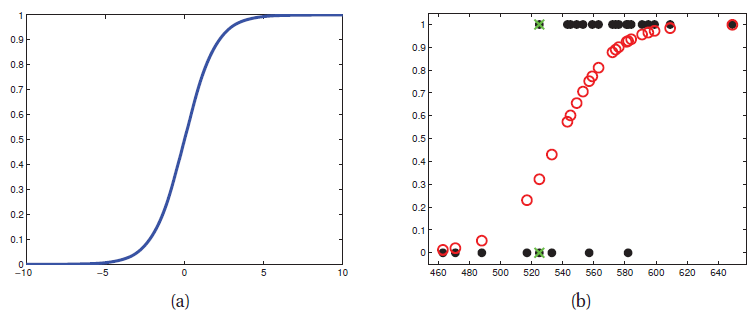


图1.13 （a）sigmoid函数；(b)针对SAT分数的逻辑回归。实心黑点为原始数据。空心红点为预测的概率值。绿色的×表示分类错误的情况。

1.4.6 逻辑回归

我们只需要作出两个调整，就可以将线性回归泛化到（二元）分类问题。首先，我们将关于y的分布从高斯分布替换为伯努利分布。即：

 （1.8）

其中。其次，跟之前一样，我们计算关于输入的线性组合，但是为了保证，需要将传入一个非线性函数：

 （1.9）

其中为sigmoid函数，又被称为logistic或者logit函数。定义为：

 （1.10）

术语“sigmoid”表示S-形：如图1.13（a）所示。该函数又被称为截断函数，因为它将整个实数轴映射到[0,1]区间，这对于需要输出具有概率解释（需要保证在区间[0,1]内）的变量来说很重要。

综合考虑前文内容，得到：

 （1.11）

上式被称为**逻辑回归（**logistic regression**）**，我们称其为回归，是因为它与线性回归的形式十分相似（尽管它实质上是分类而非回归问题）。

一个简单的关于逻辑回归的例子如图1.13（b）所示，图中我们绘制了

 （1.12）

其中xi为学生i的SAT10的成绩，yi表示学生是否通过了课程。黑点展示了训练数据，红圈绘制了曲线，其中为根据训练样本得到的关于参数的估计值（我们将在后面讨论如何计算这个估计值）。

如果我们为输出概率设置阈值为0.5，我们可以决定一个**决策规则**（decision rule）为：

 （1.13）

通过观察图1.13（b），我们发现时对应的。我们可以想象在图中绘制了一条垂直曲线，这被称为决策边界。边界左边的所有情况都归为0（没通过），右边归类为1（通过）。

我们发现基于这条决策规则的分类结果存在误分类。因为数据不是**线性可分**（linearly separable）的，也就是说没有一条直线可以将0类和1类分开。我们可以利用基函数建立模型从而输出非线性决策边界。我们在后面的章节中将会看到更多的例子。

1.4.7 过拟合

当我们训练非常灵活的模型时，我们需要关注模型是否过分适应数据（**过拟合**），也就是说我们应该避免对输入中的任何微小的变化进行建模，因为那些微小的变化很有可能是噪音而非真实的信号。这个在图1.12（b）中可以体现，我们发现当使用一个阶数特别高的多项式拟合数据时，曲线会非常“扭曲”。真实的函数基本上不会是这样的变化趋势。因此使用这样的模型很难保证未来预测的精度。

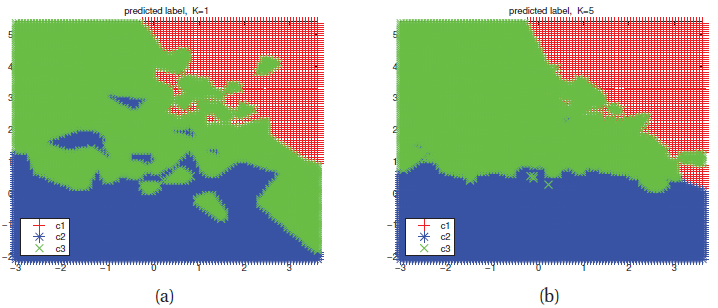


图1.14 （a）当K=1时的决策边界；（b）当K=5时的决策边界。

另一个例子，考虑KNN分类器。K值对模型的性能有重要的影响。当K=1时，模型在训练样本上没有错误，但最终的预测表面却非常“扭曲”（如图1.14（a））所示。所以这种方法在预测未来的数据时未必奏效。在图1.14（b）中，我们发现当使用K=5时，结果的预测表面将会更加光滑，因为我们在一个更大的邻居域内求平均。当K不断增加时，决策边界将会更加光滑，当K=N时，我们将所有预测输入的类别都判定为数据集中的占主要部分的类别。接下来，我们将讨论如何选择合适的K值。

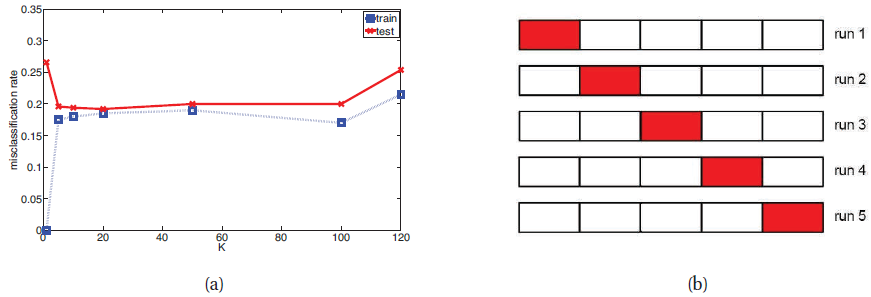


图1.15 （a）KNN分类误分类率与K值的关系。在图形左侧，当K非常小时，模型因为过于复杂因此出现过拟合，在图形右侧，当K很大时，模型简单出现欠拟合。蓝色点画线为训练集误分类率。红色实线为测试集上的误分类率。（b）5-flod交叉验证示意图。

1.4.8 模型选择

当我们拥有一系列不同复杂度的模型时（比如，包含不同阶数的线性或逻辑回归模型，或者具有不同K值的KNN模型），我们该如何选择最合适的那一个呢？一个自然的方法是计算每个方法在训练集上的误分类率（**misclassification rate**）。定义为：

 （1.14）

其中f(x)为分类器。在图1.15（a）中，我们绘制了在KNN分类器中误分类率与K的关系（点蓝线）。我们发现当K的数量增加时，训练集上的错误率将会增加。正如前文所述，如果我们选择K=1，那么模型在训练集上的错误率为0。

然而，我们真正关心的是**泛化误差**（generalization rate）。这个泛化误差可以近似为在一个大的独立的**测试集**（test set）上模型的错误率，这个测试集在模型训练过程中不被使用。我们在图1.15（a）上绘制了测试误差与K的关系。现在我们发现一条U-型曲线：对于复杂的模型（K比较小），模型过拟合，对于简单的模型（K值很大），模型**欠拟合**（underfits）。所以，一个明显的方法是挑选在测试集上错误率最小的K值（比如说，10到100之间的取值应该都很好）。

不幸的是，当我们训练模型时，我们接触不到测试集，所以我们不能使用测试集来辅助我们选择正确的模型复杂度。然而，我们可以通过将训练集分成两部分的方法创建一个测试集：一部分用于训练模型，另一部分称为**验证集**（validation set），用于选择模型的复杂度。我们基于训练集优化所有的模型，然后评估他们在验证集上的性能，并且选择性能最好的模型。一旦我们选择了最好的模型作为最终的模型，我们可以基于全部的数据重新优化模型。如果我们还有一个独立的测试集，我们可以基于这个测试集进行性能评估，用来估计我们模型的精度。

通常情况下，我们使用80%的数据作为训练集，20%作为验证集。但是如果可以获取的训练集样本数量有限，这个技术将会面临挑战，因为模型将没有足够的数据用于训练，也不会有足够的数据用于对未来性能的合理估计。

一个简单但是著名的解决方法是使用**交叉验证**（cross validation）。这个思想很简单：我们将训练集分成K个**包**（folds）；然后对于每个包k∈{1,…,K}，我们基于除了k以外的所有包进行模型训练，然后在包k上进行测试，这个过程循环进行K次，如图1.15（b）所示。然后计算模型在所有包上的错误率的平均值，作为对测试误差的代理。通常情况下，我们使用K=5，这被称为5-fold CV。如果设置K=N，相应的方法被称为**留一法交叉验证**（leave-one out cross validation）。

选择KNN分类器中的K值是一个更一般的问题——**模型选择**（model selection）的一个特例，在模型选择问题中，我们需要在具有不同复杂度的模型中作出选择。交叉验证被广泛用于解决这类问题，尽管我们会在书中介绍其他方法。

1.4.9 没有免费的午餐理论

All models are wrong, but some models are useful.—George Box

很多机器学习都涉及到设计不同的模型，以及使用不同的算法去训练这些模型。我们可以针对特定的问题使用诸如交叉验证的方法去选择最好的模型。然而，没有放之四海而皆好的模型——这被称为**没有免费的午餐理论**（no free lunch theorem）。这是因为很多假设在某个领域可能会十分奏效，在另一个领域则不尽然。

因为没有免费的午餐理论，我们需要开发不同类型的模型，去涵盖现实世界中不同领域的数据。对于每个模型，可能有很多不同算法去训练它们，且不同算法都在速度—精度—复杂度之间权衡。数据、模型和算法三者是后面章节需要学习的内容。