

第一章 绪论

第 1 讲 数据科学与工程的数学基础课程介绍

黄定江

DaSE @ ECNU

djhuang@dase.ecnu.edu.cn

1 1.1 课程介绍

2 1.2 从图像感知到自然语言处理

- 猫、分类和神经网络
- 影评、文本表示和逻辑回归

3 1.3 从数据分析到数学基础

- 数据分析与机器学习概览
- 数据
- 模型
- 学习
- 所需数学基础

1 1.1 课程介绍

2 1.2 从图像感知到自然语言处理

- 猫、分类和神经网络
- 影评、文本表示和逻辑回归

3 1.3 从数据分析到数学基础

- 数据分析与机器学习概览
- 数据
- 模型
- 学习
- 所需数学基础

这是一门什么样的课程？

- 数据科学与工程、机器学习和人工智能等学科深深扎根于“大数据”这一广袤的土地上
- 大数据的海量、高维、多模态（多样性）和高速率（到达）以及无处不在的噪声和缺失值等特征，决定了对大数据的处理和分析有别于传统计算机科学或统计学的数据处理方式
- 为了处理这些问题，需要培养新的数学基础，如对数据表示的高维空间的直观认识，熟悉用概率的方式思考问题，并且能够同步优化建模和设计算法，最终落地于计算机上的实现以及实际应用解决方案

这是一门什么样的课程？

- 这就需要对传统计算机科学以离散数学为重点的数学体系做重大改进，转移到以矩阵计算、概率论和数值优化为基础的符合数据科学与工程特色的新的数学体系
- 这一体系非常庞大，不宜一门一门的教，因此设计一门新的《数据科学与工程数学基础》课程来满足这一专业的需求
- 这门课程在数据科学、机器学习和人工智能领域的定位类似于《离散数学》在计算机科学中的定位

这是一门什么样的课程？

- 数学科学与工程的数学基础是数据科学、机器学习和人工智能的基础数学理论课
- 关于矩阵计算、概率建模和优化求解的基础知识和逻辑思维是数据科学相关专业的基本功
- 数据科学与工程的数学基础涉及的基本概念是理科学生进行数据科学、机器学习和人工智能类课程学习的重要基础

谁需要/可以学这门课程

- 这是一门面向数据科学与大数据技术专业、人工智能相关专业学科本科生的数学理论基础课
- 如果你的专业涉及到不少数据科学与人工智能类课程
- 如果你是计算机、IT、数据科学、机器学习与人工智能从业人士，希望巩固和深入一下数据科学的数学理论基础
- 如果你具有良好的数学背景，希望了解数学怎么应用于数据科学
- 那么本课程适合你！

课程的内容：数学线

- 矩阵计算：是利用计算机进行科学与工程计算的核心，是对数据进行确定性表示和函数建模的必备要素，是设计快速可靠数据分析算法必备技术。
- 概率论：数据中的不确定性怎么量化？如何描述带有噪声的数据？概率准则和概率模型将带给你理性的哲学思考。
- 优化理论：如何找到数据模型中的参数，使得基于此的预测量能对未知的数据进行更准确的预测？基于函数微分和梯度的数值优化算法将帮助你一步一步实现目标。

课程的内容：数据线

- 数据分析与机器学习的一般处理流程：可以清晰的告诉你所处的数据任务处理阶段和所需的对应数学
- 信息检索、自然语言处理和图像感知的例子贯穿全书，告诉你对于这些任务，你怎么使用数学来实现你的任务
- 数据科学、统计机器学习、深度学习、强化学习中各种优化问题类型会渗透全书

左手：数据线；右手：数学线。

明线：数学线；暗线：数据线。

课程的难度和深度怎么样

- 本课程的目标重点在于基本概念的理解、基础计算方法的掌握和应用；
- 不涉及过多的数学证明；
- 会为后续更深度的数学基础内容预留接口；
- 培养采用数据思维、计算思维来分析问题、解决问题的能力
- 数据思维和计算思维贯穿始终：数据 → 信息 → 知识
- 两大类应用案例贯穿全书：信息检索（自然语言处理）、图像识别

对接的后续课程

本课程的内容足够支撑你学习后续课程所需的数学，这些课程包括：

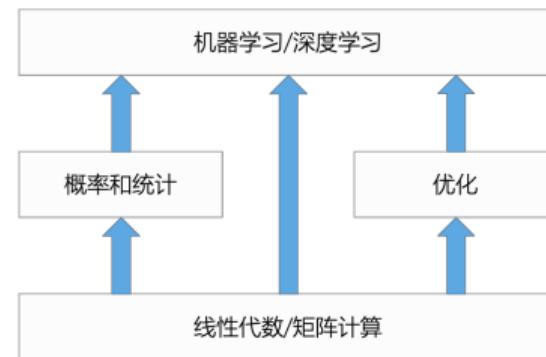
- 《数据科学与工程算法基础》
- 《机器学习》
- 《概率图模型》
- 《人工智能》
- 《计算机视觉》
- 《自然语言处理》
- ...

该怎么学习本课程

- 如果你具备工科微积分、线性代数、概率论和数理统计的基础知识，你可以相对轻松的学完本课程；
- 本课程第 2 章和第 7 章会从数据表示的角度帮你重新回顾线性代数、概率论和数理统计的基础知识，让你非常光滑平稳的深入本课程更高级的内容学习；
- 如果你不具备工科线性代数、概率论和数理统计的基础知识，也不用害怕，你只需要更用心学习本课程第 2 章和第 7 章的内容即可；
- 本课程第 2 章和第 7 章会为你提供一个足够本课程使用的简明的线性代数和概率论与数理统计基础知识，并配备足量的习题供你练习巩固；
- 本课程每一讲内容会配备适量的习题或动手实践练习供你巩固所学内容；
- 本课程的参考教材是我们的新书《数据科学与工程数学基础》，即将 2020 暑期出版！

欢迎选修《数据科学与工程数学基础》

欢迎选修《数据科学与工程数学基础》！



1 1.1 课程介绍

2 1.2 从图像感知到自然语言处理

- 猫、分类和神经网络
- 影评、文本表示和逻辑回归

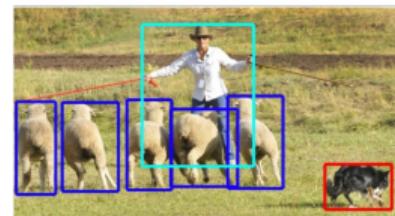
3 1.3 从数据分析到数学基础

- 数据分析与机器学习概览
- 数据
- 模型
- 学习
- 所需数学基础

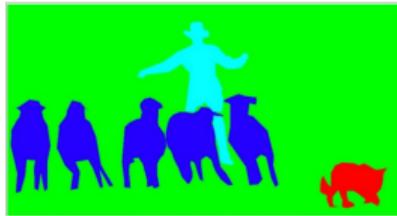
1.2.1 猫、分类和神经网络: 计算机视觉任务



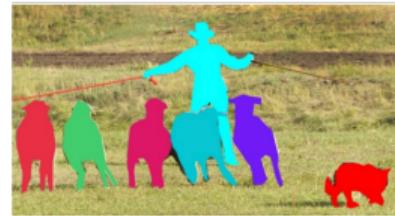
(a) 图像分类



(b) 目标检测



(c) 语义分割



(d) 实例分割

图 1: 计算机四大视觉任务

计算机视觉任务



(a) 语义分割



(b) 分类 + 定位



(c) 多物体目标检测



(d) 多物体语义分割

图 2: 计算机视觉任务 (以猫为例)

图像分类问题

- **图像识别问题**: 计算机如何识别一张猫的图像? 该问题可以转化为设计一个分类器, 让它去预测图像属于哪个分类标签。
- **图像分类问题**: 已有固定的分类标签集合, 然后对于输入的图像, 从分类标签集合中找出一个分类标签, 最后把分类标签分配给该输入图像。

图像分类问题

例 1

已知一个类别标签集合 $\{airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck\}$,
计算机如何判断下面左图属于哪个类别?



图 3: 输入图像

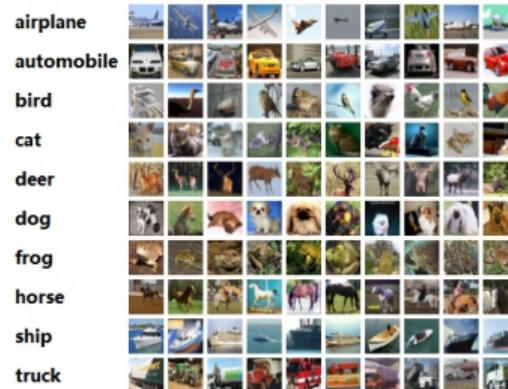


图 4: 十个类别

图像的数据表示

- 计算机可以使用 RGB 位图表示彩色图像，RGB 位图用三元数组表示，数组元素的取值范围为 $[0,255]$ 的整数，数组的大小是宽度 \times 高度 \times 通道数，RGB 图像的通道数即红、绿、蓝三个通道。

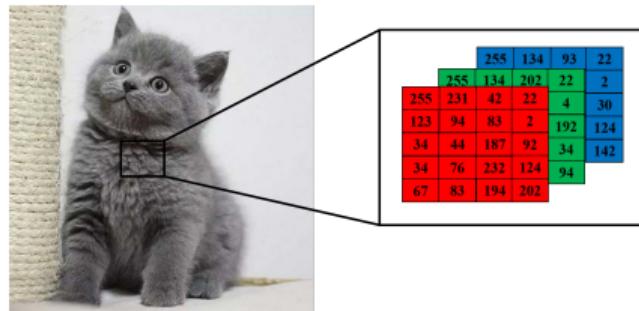


图 5: 图像的数据表示。该图像可以用大小为 $32 \times 32 \times 3$ 的三元数组表示。

图像分类流程

在机器视觉中，经常采用基于数据驱动的方法对图像分类，即给计算机很多图像数据和标签，然后实现学习算法，让计算机学习到每个类别的特征。该方法流程如下：

- 输入：输入是 N 个图像的集合 (即训练集)，每个图像的标签是所有分类标签中的一种；
- 学习：使用训练集来学习每个类的特征，这一步也称训练分类器或学习一个模型；
- 评价：让分类器预测未曾见过的图像 (测试图像) 的标签，将预测标签与真实标签进行对比，来评价分类器的质量。通常使用测试集准确率评价分类器，准确率 = 预测正确的数量/测试集样本数量。

数据集划分和预处理

例 2

图像分类数据集：*CIFAR-10* 是一个常用图像分类的数据集，包含 60000 张 $32 \times 32 \times 3$ 的图像，10 个类别，每个类别有 6000 张图像和对应标签。

- 数据集划分：每个类别的训练集 5000 张，测试集 1000 张，每张图像都有对应标签。
- 在训练模型之前，一般需要对数据进行预处理，例如重塑、归一化和降维。
- 重塑：可以将 $32 \times 32 \times 3$ 的 3 元数组重塑为一个 3072×1 的 1 元数组，1 元数组等价于向量，这个向量的维数是 3072 维 (向量的维数是向量分量的个数)。
- 归一化：对数据进行归一化，可以保证计算距离时各个维度的量纲保持一致。

数据集划分和预处理

- 降维：如果数据是高维数据 (向量的维数或者向量空间的维数)，可以考虑使用降维方法对数据降维，比如 PCA 降维，下图表示从 2 维向量降到 1 维向量。

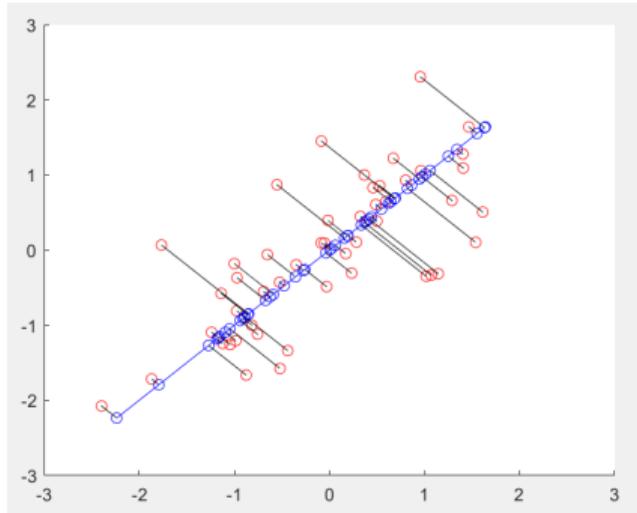


图 6: PCA 降维

分类器举例

三种类型的分类器：

- KNN
- 线性分类
- 卷积神经网络

KNN 分类器 - 概述

- KNN 分类器在训练集图像中找出测试图像的 K 个近邻图像 (即最相似的 K 个图像), 将 K 个近邻图像出现频率最高的类别作为测试图像的类别。
- 为了找出 K 个近邻图像, 需要计算测试图像与每个训练图像之间的距离。

KNN 分类器 - 距离

- 在 *CIFAR-10* 中，首先将测试图像和训练图像先转化为两个 3072 维向量 I_1 和 I_2 ，然后计算它们之间的 L_1 距离

$$d_1(I_1, I_2) = \sum_{p=1}^{3072} |I_1^p - I_2^p|$$

- 注意：还可以选择 L_2 距离

$$d_2(I_1, I_2) = \sqrt{\sum_{p=1}^{3072} (I_1^p - I_2^p)^2}$$

KNN 分类器 - 距离

$$\begin{array}{c} \begin{array}{c} \text{test image} \\ \begin{array}{|c|c|c|c|} \hline 56 & 32 & 10 & 18 \\ \hline 90 & 23 & 128 & 133 \\ \hline 24 & 26 & 178 & 200 \\ \hline 2 & 0 & 255 & 220 \\ \hline \end{array} \end{array} - \begin{array}{c} \begin{array}{c} \text{training image} \\ \begin{array}{|c|c|c|c|} \hline 10 & 20 & 24 & 17 \\ \hline 8 & 10 & 89 & 100 \\ \hline 12 & 16 & 178 & 170 \\ \hline 4 & 32 & 233 & 112 \\ \hline \end{array} \end{array} = \begin{array}{c} \begin{array}{c} \text{pixel-wise absolute value differences} \\ \begin{array}{|c|c|c|c|} \hline 46 & 12 & 14 & 1 \\ \hline 82 & 13 & 39 & 33 \\ \hline 12 & 10 & 0 & 30 \\ \hline 2 & 32 & 22 & 108 \\ \hline \end{array} \end{array} \xrightarrow{\text{add}} 456 \end{array} \end{array}$$

图 7: 以图片中的一个颜色通道为例进行说明。两张图像通过 L_1 距离进行比较，逐个像素求差值，再将所有差值求和。如果两张图像完全一样，则 L_1 距离为 0；如果两张图片差异极大，则 L_1 值将会非常大。

KNN 分类器 -超参数和评价

KNN 的 K 值该如何选取？向量之间相似性选择 $L1$ 距离还是 $L2$ 距离进行度量？

- 这些选择被称为超参数。在数据驱动的机器学习算法设计中，超参数很常见，但如何选取往往需要通过验证集进行参数调优。

评价：

- 使用 KNN 能够在 $CIFAR-10$ 上得到将近 40% 的准确率，该算法简单易实现，但需要存储所有训练数据，并且测试时过于耗费算力。

线性分类器 - 概述

线性分类器主要有两部分组成：

- **评分函数**：原始图像数据到类别标签得分的映射；
- **损失函数**：用来量化预测标签的得分与真实标签之间的一致性。

该方法可以转化为一个最优化问题，在最优化过程中，将通过更新评分函数的参数来最小化损失函数值。

线性分类器 - 评分函数

线性分类器的评分函数可以表示为

$$f(\mathbf{W}, b; x_i) = \mathbf{W}x_i + b$$

其中，

- x_i 表示第 i 张图像转换的大小为 $D \times 1$ 的列向量。
- 评分函数的参数包括 \mathbf{W} 和 b ， \mathbf{W} 是 $K \times D$ 的矩阵， b 是 $K \times 1$ 的列向量， K 表示类别的数量。

在 $CIFAR-10$ 中， x_i 大小为 3072×1 ， \mathbf{W} 大小为 10×3072 ， b 大小为 10×1 。因此，评分函数输入是 3072 个原始像素数值，函数输出 10 个不同类别的得分。

线性分类器 - 评分函数

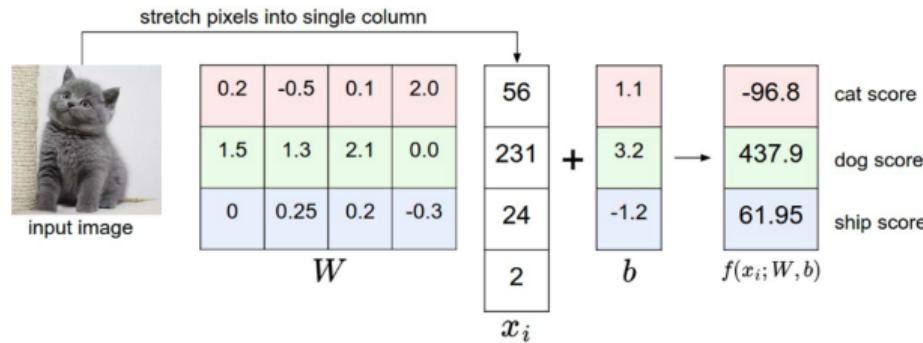


图 8: 评分函数可视化。假设图像只有 4 个像素 (也不考虑 RGB 通道), 有 3 个分类 (cat、dog、ship)。首先将图像像素拉伸为一个列向量, 与 W 进行矩阵乘, 然后得到各个分类的分值, 分值最大对应的类别的是函数判断的类别, 上图根据 4 个像素将图像误判为 1 条狗。

线性分类器 - 损失函数

- 为了方便损失函数计算，使用 Softmax 函数将一组 $(-\infty, +\infty)$ 的得分 f 转换为一组 $(0, 1)$ 概率，并且这组概率的和为 1。每张训练图像属于类别 i 的概率得分用公式表示为

$$p_i = \frac{e^{f_i}}{\sum_{j=1}^K e^{f_j}}.$$

线性分类器 - 损失函数

- 根据预测类别的概率得分，使用交叉熵函数作为损失函数，计算真实标签与预测标签之间的损失。
- 将标签 y 转换为 $one-hot$ 向量 \mathbf{y} ，例如真实标签为 4，则 $\mathbf{y} = [0, 0, 0, 0, 1]$ 。
- 每张训练图像对应的交叉熵损失用公式表示为

$$l = - \sum_{c=1}^K \mathbf{y}_c \log(p_c) = - \sum_{c=1}^K \mathbf{y}_c \log(\mathbf{p}_c)$$

- 其中, K 是类别的数量； \mathbf{y}_c 表示当前图像的指示变量，如果预测标签与真实标签相同就是 1，否则为 0； \mathbf{p}_c 是当前图像属于类别 c 的概率得分。

线性分类器 - 损失函数

显然每张图像只需要计算一个类别的概率得分和真实标签的交叉熵。因此第 i 张图像的交叉损失函数又可以表示为

$$l_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_{j=1}^K e^{f_j}}\right) = -f_{y_i} + \log\left(\sum_{j=1}^K e^{f_j}\right)$$

- 其中, K 是类别的数量; y_i 表示第 i 张图像的类别标签 ID ; p_c 是当前图像属于类别 c 的概率得分。

线性分类器 - 最优化过程

最优化的目标即对所有训练集的图像的损失和最小：

$$\min Loss = \min \sum_{i=1}^N l_i = \min - \sum_{i=1}^N \log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

寻找能使得损失函数值最小化的参数 \mathbf{W} 的过程。可考虑尝试多个策略：

- 策略 1：随机搜索。从随机权重开始，然后迭代取优，从而获得更低的损失值。
- 策略 2：随机本地搜索。从随机权重开始，然后生成一个随机的 $\delta \mathbf{W}$ ，只有当 $\mathbf{W} + \delta \mathbf{W}$ 的损失值变低，才可以更新。
- 策略 3：跟随梯度。从数学上计算最陡峭的方向，然后向着最陡峭的方向下降。

线性分类器 - 最优化过程

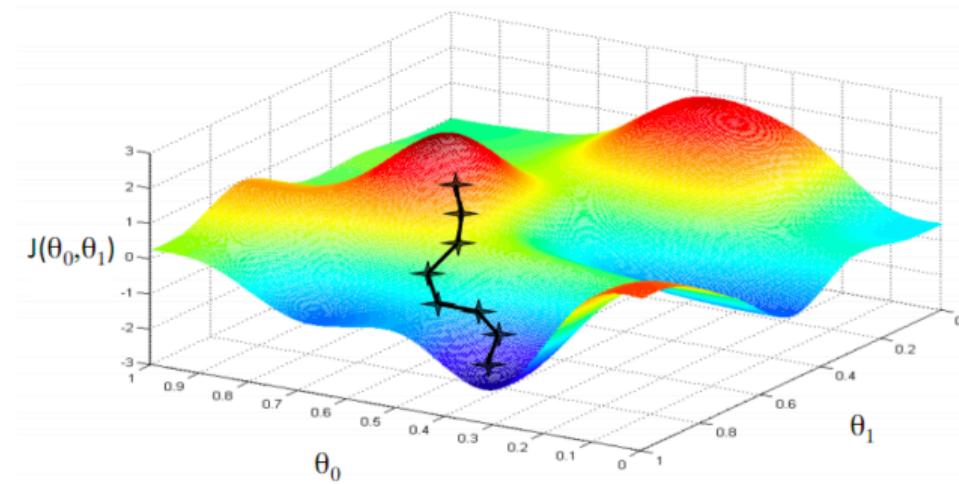


图 9: 梯度下降算法

卷积神经网络

CNN 概述：输入一张图像，经过一系列卷积层、非线性层、池化（下采样）层和完全连接层，最终得到输出。如下图所示。输出可以是描述了图像内容的一个单独分类或一组分类的概率。

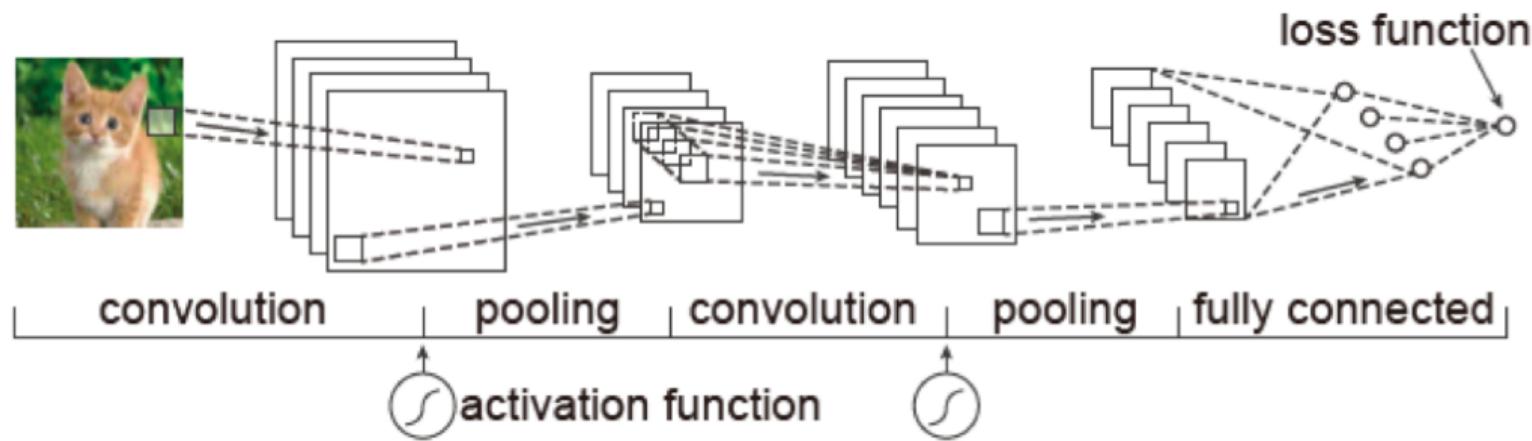


图 10: 一个典型的 CNN 架构图

卷积神经网络 - 网络结构

在本例中用于 CIFAR-10 图像数据分类的 CNN 结构可以是 [输入层 - 卷积层 - ReLU 层 - 池化层 - 全连接层]。

- 输入层：输入 $[32 \times 32 \times 3]$ 存有图像的原始像素值。
- 卷积层：卷积层检测边缘和曲线一类的低级特征。核（有时也被称作滤波或神经元）与输入层中的一个局部区域相连，每个核都计算自己与输入层相连的小区域与自己权重的内积。卷积层会计算所有核的输出，若使用 12 个卷积核，则输出为 $[32 \times 32 \times 12]$ 。下面我们将对输入为 3×4 ，核为 2×2 的卷积操作做简单的说明。

卷积神经网络 - 网络结构

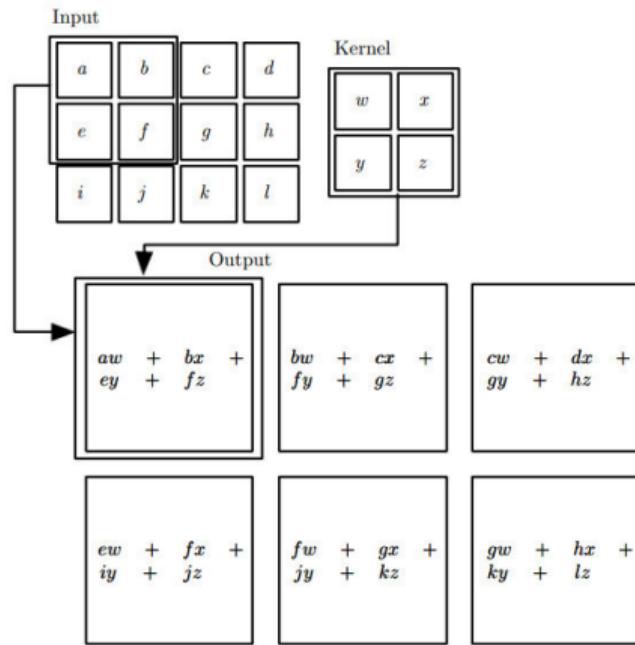


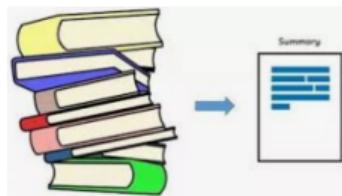
图 11: 卷积操作

卷积神经网络 - 网络结构

- ReLU 层：对卷积操作后的每个元素进行激活函数操作，比如使用 $\max(0, x)$ 作为激活函数。该层对数据尺寸没有改变，仍是 $[32 \times 32 \times 12]$ 。在该层中使用的激活函数都是非线性函数，主要是用来加入非线性因素的，因为线性模型的表达能力不够。
- 池化层：在空间维度上进行降采样，输出尺寸变为 $[16 \times 16 \times 12]$ 。主要是对 ReLU 层的输入进行压缩，使得特征图变小，提取主要特征。
- 全连接层：计算分类评分，数据尺寸变为 $[1 \times 1 \times 10]$ ，对应 10 个类别的分类评分值。

在本例中，得到全连接层对应每个类别的分类评分值后，再通过 Softmax 归一化得到每个类别的概率。

1.2.2 影评、文本表示和逻辑回归: 自然语言处理四类常见的任务



- **文本分类:** 舆情监测, 新闻分类.
- **序列标注:** 分词, 词性标注, 命名实体识别.
- **文本匹配:** 搜索引擎, 自动问答.
- **文本生成:** 机器翻译, 文本摘要.
- ...

文本分类问题

- **文本分类问题：** 文本分类或者称为自动文本分类，是指给定文档 p （可能含有标题 t ），将文档分类为 n 个类别中的一个或多个。
- **文本分类方法：** 传统机器学习方法（逻辑回归模型，`svm` 等），深度学习方法（`fastText`，`TextCNN` 等）
- **文本分类应用：** 常见的有垃圾邮件识别、情感分析、电影评论分析，...

文本分类流程

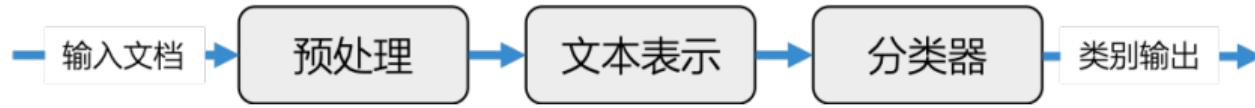


图 12: 文本分类流程

电影影评分类

例 3

以文本分类中的影评分类为例，介绍自然语言处理的建模流程。影评分类数据如下所示：

电影影评	类别
the plot of this movie is funny, excellent!!!	1
this movie is awful indeed.	0

图 13: 2 条电影影评数据

要对电影分类问题进行建模，第一个问题是如何在计算机中表示电影评论数据（为简化处理，忽略影评数据中的标点符号），第二个问题是基于影评数字化表示，对其进行分类建模。

文本表示

主流的文本表示方法有

- 独热编码
- 词袋模型
- TF-IDF
- 共现矩阵
- 词嵌入

独热编码 (one-hot)

- *one-hot* 向量是最简单的词向量，用一个 $\mathbb{R}^{|V| \times 1}$ 向量来表示每个单词，将所有的词排序，每个词对应下标由 0 和 1 组成， $|V| = 10$ 是词汇表的大小。

$$w^{the} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{plot} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{of} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, w^{indeed} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- 每个单词表示为一个完全独立的实体，但是任意两个词向量没有体现相似性的概念：

$$(w^{the})^T w^{plot} = (w^{the})^T w^{of} = 0$$

词袋模型 (bag-of-word)

词袋表示，也称为计数向量 (Count Vectors) 表示。将文本看做词袋，忽略文本的词序、语法和句法，仅仅将文本看做一些列词的组。

- 基于影评分类文本示例，构建一个词表：

```
V={1:"the",2:"plot",3:"of",4:"this",5:"moive",6:"is",7:"funny",8:"excellent",  
9:"awful", 10:"indeed" }
```

- 这个词表一共包含 10 个不同的单词，利用词表的索引号，上面两个文档可以用两个 10 维向量表示：

文本 1 可以表示为： [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]

文本 2 可以表示为： [0, 0, 0, 1, 1, 1, 0, 0, 1, 1]

TF-IDF

词袋模型是基于计数得到的，而 TF-IDF 则是基于频率统计得到的。TF-IDF 的分数代表了词语在当前文档和整个语料库中的相对重要性。TF-IDF 分数由两部分组成：第一部分是词语频率（Term Frequency），第二部分是逆文档频率（Inverse Document Frequency）

$$TF(\text{单词}) = \frac{\text{该词在当前文档出现次数}}{\text{当前文档中的词语总数}}$$

$$IDF(\text{单词}) = \ln \frac{\text{文档总数}}{\text{出现该词语的文档总数}}$$

TF-IDF

对于文本 1 中 “plot” 计算 tf-idf 值：

$$tf_{plot, \text{文本 1}} = \frac{1}{8}$$

$$idf_{plot, \text{文本 1}} = \ln \frac{2}{1} = \ln 2$$

$$\begin{aligned} tf\text{-}idf_{plot, \text{文本 1}} &= tf_{plot, \text{文本 1}} \cdot idf_{plot, \text{文本 1}} \\ &= \frac{1}{8} \cdot \ln 2 \approx 0.125 \end{aligned}$$

类似的，可以对语料中每个文本中的每个词计算 tf-idf 值，并将 tf-idf 值放入到词袋中得到每个文本的向量表示。

共现矩阵

记录每个单词在目标单词的特定大小的窗口（取窗口大小为 1）中出现的次数，得到的关联矩阵 \mathbf{X} ，称为共现矩阵：

	<i>the plot of this moive is funny excellent awful indeed</i>									
<i>the</i>	0	1	0	0	0	0	0	0	0	0
<i>plot</i>	1	0	1	0	0	0	0	0	0	0
<i>of</i>	0	1	0	1	0	0	0	0	0	0
<i>this</i>	0	0	1	0	2	0	0	0	0	0
$\mathbf{X} =$	<i>moive</i>	0	0	0	2	0	2	0	0	0
<i>is</i>	0	0	0	0	2	0	1	0	1	0
<i>funny</i>	0	0	0	0	0	1	0	1	0	0
<i>excellent</i>	0	0	0	0	0	0	1	0	0	0
<i>awful</i>	0	0	0	0	0	1	0	0	0	1
<i>indeed</i>	0	0	0	0	0	0	0	0	1	0

词嵌入 (word embedding)

- 词嵌入可以理解为一种映射，将文本空间中的某个单词，通过一定的方式，映射或嵌入到另外一个数值向量空间。
- 常见词嵌入模型
 - Word2Vec: 包含 CBOW 和 Skip Gram
 - Glove (Global Vectors for Word Representation)
 - Fasttext
 - BERT
 - ...

连续词袋模型 (CBOW)

- CBOW 模型给定上下文 $w_{t-i}, \dots, w_{t-1}, w_{t+1}, w_{t+i}$ 来预测目标词 w_t , 模型结构如下图:

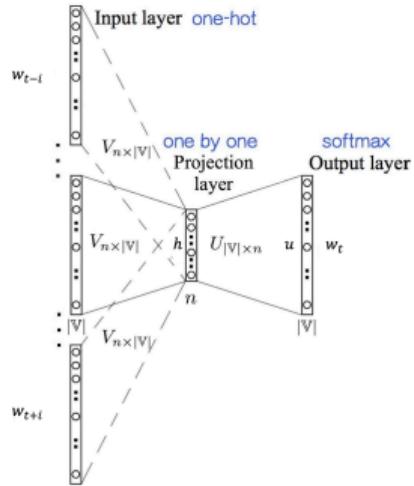


图 14: CBOW 模型。 w_t 为目标词, 其余词

$w_i, i \neq t$ 为上下文 $w_{context}$

- 第一步: 计算隐层 h

$$h = \frac{1}{C} V^T \cdot \left(\sum_{i=1}^C w_i \right)$$

- 第二步: 计算输出层输出

$$u_j = U^T \cdot h$$

$$y_j = p(w_t | w_{context}) = \frac{\exp(u_j)}{\sum_{j'=1}^C \exp(u_{j'})}$$

连续词袋模型（CBOW）

- CBOW 的损失函数

$$E = -\log p(w_t | w_{context})$$

- 该损失函数可以理解为一种特殊的衡量两个概率分布的交叉熵形式
- 通过最小化该损失函数，利用梯度下降法可得到最终词向量表示矩阵 V 和 U

文本分类建模

文本分类建模

主要考虑使用两类方法对文本分类问题进行数学建模：

- 传统方法：在表示上，使用 TF-IDF 对文档进行表示，使用逻辑回归 (Logistics Regression, LR) 模型对文本分类进行数学建模。
- 神经网络方法：在表示上，使用词向量 word2vec 对单词进行表示，然后使用循环神经网络 (Recurrent Neural Network, RNN) 对词向量特征进行进一步表达并用 softmax 映射输出进行非线性分类建模。

TF-IDF 与逻辑回归

逻辑回归模型

逻辑回归是一种分类模型，它假设数据标签服从伯努利分布，使用条件概率 $P(y = 1|x)$ 进行建模，其中 x 就是影评评论的 TF-IDF 表示，参数模型如下：

$$P(y = 1|x; w) = \frac{\exp^{(w^T x + b)}}{1 + \exp^{(w^T x + b)}}$$

其中 w 是权重参数向量，它的维数与 x 的维数相同， b 是偏执项。

TF-IDF 与逻辑回归

参数估计

对于概率模型，使用“极大似然法”来构建对数损失：

$$L = -\frac{1}{m} \sum_{i=1}^m \ln(P(y_i|x; w))$$

其中：

$$P(y_i|x; w) = P(y=1|x; w)^{y_i} (1 - P(y=1|x; w)^{y_i})^{1-y_i}$$

最后使用优化算法（如梯度下降法）对参数进行估计。

神经网络建模方法: word2vec 的缺陷

Word2Vec 的缺陷

- Word2Vec 是在大量无监督语料上使用浅层神经网络训练出来的词嵌入模型, 它将单词映射成低维稠密向量, 仅仅是缓解词了词语相似度的表达
- Word2Vec 未能彻底解决语言学中的一词多义问题

RNN: 序列数据建模

对于序列数据建模 (文本, 语音, 股票等), RNN 引入了隐状态 h (hidden state) 的概念。经过 RNN 编码后, h 可以提取序列数据的特征。RNN 架构图如下所示:

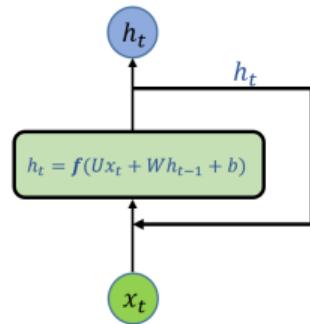


图 15: RNN 结构图

第 t 时刻的输入以及第 $t-1$ 时刻的隐藏状态 h_{t-1} 经非线性变换 f 得到 h_t 。

RNN: 序列数据建模

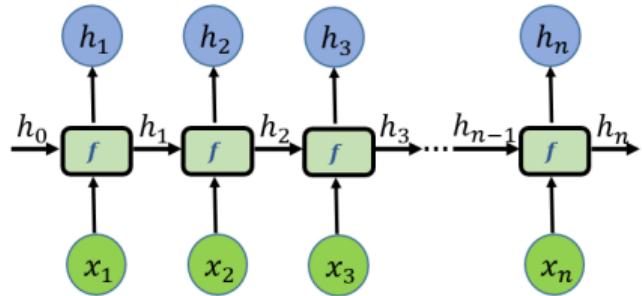


图 16: RNN 按时间步展开

- 在处理文本数据时, 图 16 中的 x_1 可以看做是第一个单词的词向量, x_2 可以看做是第二个单词的词向量, 依次类推
- 在处理语音数据时, 此时 $x_1 x_2 x_3, \dots$ 是每帧的声音信号
- 隐藏状态 h_i 编码了第 i 以及之前时刻的数据特征

基于 RNN 的文本分类建模

在文本分类问题中，对于一个包含 n 的单词的文本 $W = (w_1, w_2, \dots, w_n)$ ，我们使用 RNN 对文本进行序列建模编码，取第 n 时刻的隐藏状态 h_n 来表示文本并使用其进行文本分类。

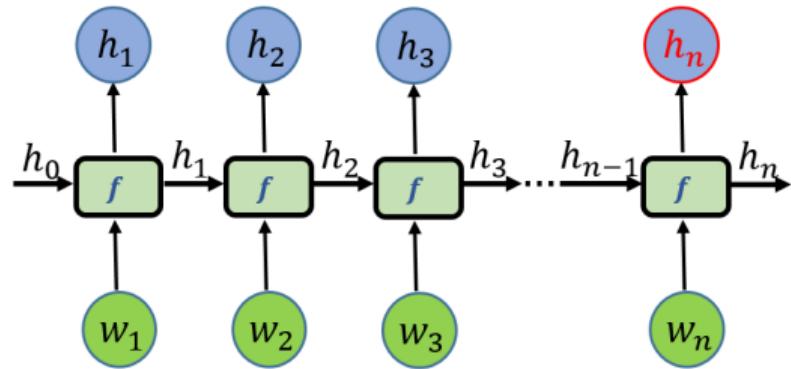


图 17: RNN 对文本进行编码

基于 RNN 的文本分类建模

得到文本表示后，先使用线性变换对获得的特征进行加权组合，然后用 softmax 进行映射输出：

$$\begin{pmatrix} \text{logit}^{(0)} \\ \text{logit}^{(1)} \end{pmatrix} = Gh_n + t = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1d} \\ g_{21} & g_{22} & \cdots & g_{2d} \end{bmatrix} h_n + \begin{bmatrix} t_{11} \\ t_{21} \end{bmatrix}$$

$$P(\text{负类}|x; w) = P(y = 0|x; w) = \frac{\exp^{\text{logit}^{(0)}}}{\exp^{\text{logit}^{(0)}} + \exp^{\text{logit}^{(1)}}}$$

$$P(\text{正类}|x; w) = P(y = 1|x; w) = \frac{\exp^{\text{logit}^{(1)}}}{\exp^{\text{logit}^{(0)}} + \exp^{\text{logit}^{(1)}}}$$

这里 G 是 $2 \times d$ 的参数矩阵、 t 是 2×1 的列向量， w 是模型参数，由 RNN 中的 U, W, b 以及 softmax 分类层中的 G, t 组成 $w = (U, W, b, G, t)$ 。

基于 RNN 的文本分类建模

参数估计

得到各个类别的概率后，使用“极大似然法”来构建对数损失：

$$L = -\frac{1}{m} \sum_{i=1}^m \ln(P(y_i|x; w))$$

其中：

$$P(y_i|d) = P(y_i|x; w)^{y_i} (1 - P(y_i|x; w))^{1-y_i};$$

最后使用优化算法（如梯度下降法）对参数 $w = (U, W, b, G, t)$ 进行估计。

小结

模式分析的主要任务：

计算机视觉

- 图像识别
- 目标定位或目标检测
- 语义分割或实例分割
- ...

自然语言处理

- 信息查询（信息检索）
- 语音识别
- 机器翻译
- ...

从数据驱动方法的角度看，模式分析的主要任务最后都归结为数据分析中各种基本运算，如**分类**、**回归**、**标注**、**聚类**和**降维**等等，实现这些运算的方法理论支撑是机器学习。

1 1.1 课程介绍

2 1.2 从图像感知到自然语言处理

- 猫、分类和神经网络
- 影评、文本表示和逻辑回归

3 1.3 从数据分析到数学基础

- 数据分析与机器学习概览
- 数据
- 模型
- 学习
- 所需数学基础

数据智能与机器学习

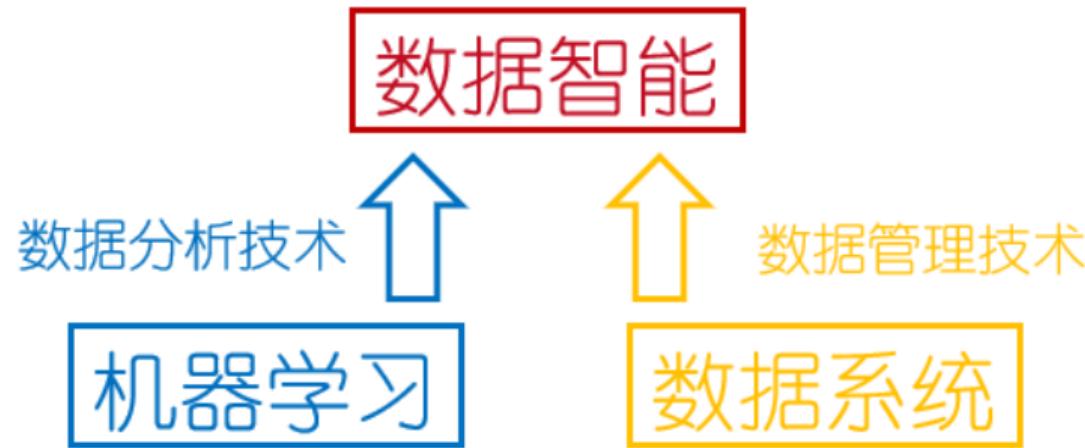


图 18: 机器学习与数据智能的关系

1.3.1 机器学习的定义

机器学习的定义

机器学习就是关于如何用计算机基于数据构建概率统计模型并运用模型对数据进行预测与分析的一门学科。

深度学习的定义

粗略的说，深度学习是主要使用深度神经网络为工具的机器学习算法，也即通过多层非线性变换对高复杂度数据建模的算法的合集。

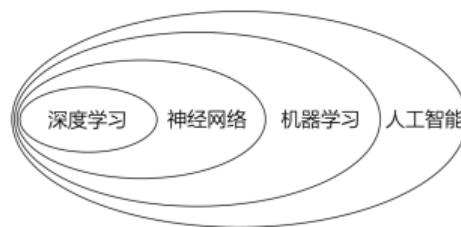


图 19: AI 中四个概念的包含关系

机器学习是数据全生命周期中的核心环节

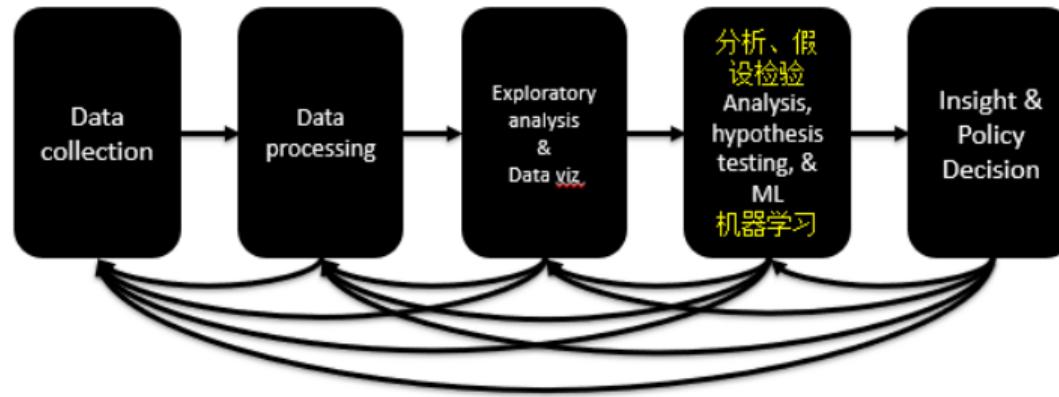


图 20: 数据分析与机器学习在数据全生命周期所处的阶段

典型的机器学习过程

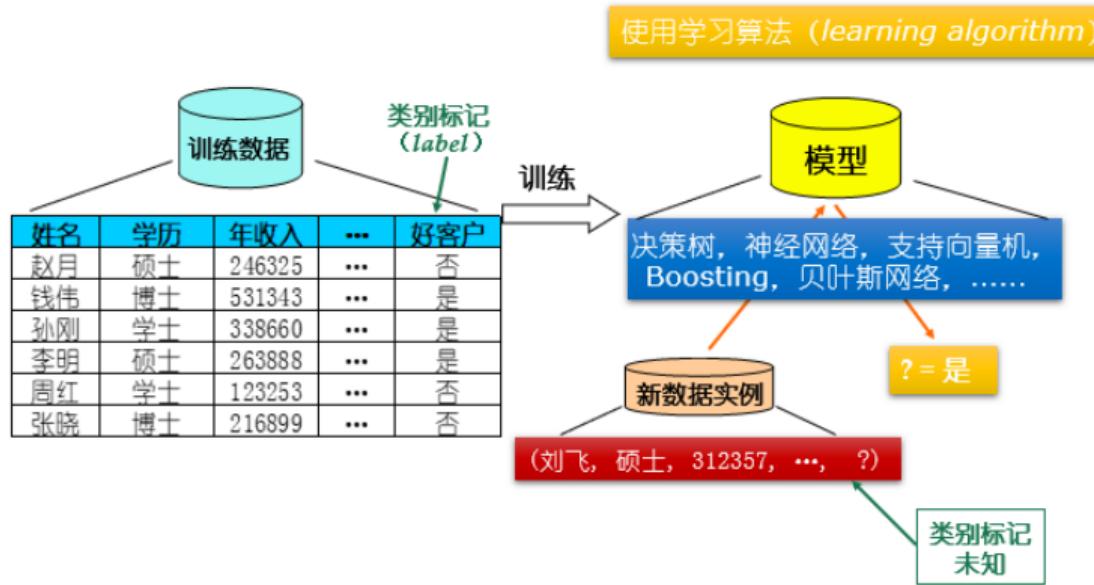


图 21: 典型的机器学习过程

基本术语：机器学习系统构成

机器学习系统组成

- 数据：训练数据、测试数据
- 模型：确定性模型和不确定性模型
- 学习：策略和算法

基本术语：机器学习方法的三要素

机器学习方法的三要素

- 模型：模型的假设空间
- 策略：模型选择的准则
- 算法：模型学习的算法

机器学习方法概括

- 从给定的、有限的、用于学习的训练数据集合出发，假设数据是独立同分布产生的；
- 并且假设要学习的模型属于某个函数的集合，称为假设空间；
- 应用某个评价准则，从假设空间中选取一个最优的模型，使它对已知的训练数据及未知的测试数据在给定的评价准则下有最优的预测；
- 最优的模型选取由算法实现



实现机器学习方法的步骤

步骤

- 得到一个有限的训练数据集合；
- 确定包含所有可能的模型的假设空间，即学习模型的集合；
- 确定模型选择的准则，即学习的策略；
- 实现求解最优模型的算法，即学习的算法；
- 通过学习方法选择最优模型；
- 利用学习的最优模型对新数据进行预测和分析。

基本术语：机器学习的主要任务

大数据的各种计算任务：预测目标

- 分类：离散值
- 回归：连续值
- 标注：离散值
- 聚类：无标注信息
- 降维：无标记信息
- 概率密度估计：无标记信息
- ...

基本术语：机器学习的学习模式

实现任务的方式：有无标记信息

- 监督学习：分类、回归、标注
- 无监督学习：聚类、降维、概率密度估计
- 半监督学习：两者结合
- ...

基本术语：机器学习的目标和泛化能力

基本术语：泛化能力

机器学习的目标是找到好模型，使得学到的模型能很好的适用于“未知的测试数据”，而不仅仅是训练数据，我们称模型适用于未知数据的能力为泛化 (generalization) 能力。

一般而言训练数据越多越有可能通过学习获得强泛化能力的模型。

1.3.2 数据：类型和特性

数据类型

- 图像（非结构化）
- 视频（非结构化）
- 文本（非结构化）
- 语音（非结构化）
- 网页（半结构化）
- 图数据（半结构化）
- 时间序列（半结构化）
- ...

数据特性

- 高维
- 海量
- 多模
- 高速
- 噪声
- 稀疏
- 非平衡、缺失
- ...

如何对数据进行表示？

例 4

考虑如下人力资源数据。假设数据按表格存放，表的每一行表示某个人，每一列表示人的某个特征，如何把表格转换成可以由计算机读取并以数字表示的数据？

姓名	性别	学位	邮编	年龄	年薪
赵月	女	硕士	710001	34	246325
钱伟	男	博士	518051	44	531343
孙刚	男	学士	410013	52	338660
李明	男	硕士	100010	31	263888
周红	女	学士	150010	25	123253

Table 1: 人力资源数据

数据表示的指导准则

类变量转化为数字

性别列（类变量）可以被转换为表示“男性”的数字 0 和表示“女性”的 1。或可以分别用数字 -1 , $+1$ 表示

运用领域知识

在构建表示时使用特定领域知识通常很重要，例如知道大学学位从学士学位到硕士学位到博士学位，或者知道邮政编码不仅仅是一串字符，而实际上是某一个区域的编码。

合理的单位

可能直接读入机器学习算法的数值数据都应该仔细考虑单位，合理缩放和约束。本例中，年薪在转化后以万为单位。

姓名	性别	学位	邮编	年龄	年薪
赵月	女	硕士	710001	34	246325
钱伟	男	博士	518051	44	531343
孙刚	男	学士	410013	52	338660
李明	男	硕士	100010	31	263888
周红	女	学士	150010	25	123253

性别	学位	纬度	经度	年龄	年薪 (万)
-1	2	34.2304	108.9343	34	24.6325
+1	3	22.5329	113.9303	44	53.1343
+1	1	28.2351	112.9313	25	33.8660
+1	2	39.9316	116.4101	52	26.3888
-1	1	45.7570	126.6425	31	12.3253

Table 2: 转换后的人力资源数据

数据表示：数据作为向量

每个输入 x_n 都是 D 维向量

经过合适转换后的数据，每个输入 x_n 都是由数字组成的一元 D 维数组，后面我们称之为向量，这些向量也被称为特征、属性或协变量。更一般的， x_n 也可以是复杂的结构化对象（例如，图像、文字、电子邮件消息、时间序列、分子形状、图等）。

数据表示：数据作为矩阵

N 个输入 \mathbf{x}_n 形成 $N \times D$ 矩阵

数据集中 N 个输入 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 经过合适转换后的数据，按行排成一个 $N \times D$ 的二元数组，后面我们称之为矩阵，这些矩阵也被称为特征或属性矩阵，记作 $\mathbf{X} \in \mathbb{R}^{N \times D}$ 。

特征矩阵每一行是一个实例，每一列是一个特征

特征矩阵每一行是某个个体 \mathbf{x}_n ，称为机器学习中的实例 (instance) 或数据点。一般，使用 N 来表示数据集中的实例数，并使用小写 $n = 1, \dots, N$ 来索引实例，下标 n 指的是数据集中总共 N 个实例中的第 n 个实例；使用 D 来表示数据集中总的特征数，每列表示关注的特征，用 $d = 1, \dots, D$ 索引特征。

监督学习中的实例标签对

对于监督学习问题，有一个与每个输入实例 x_n 相关联的输出标签 y_n 。

数据集被写为一组实例标签对或输入输出对：

$$\{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\},$$

实例标签对或输入输出对也称为样本或样本点。图22表示一维输入 x 和对应标签 y 的实例。

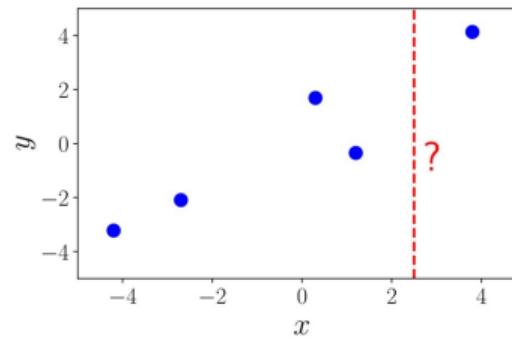


图 22: 线性回归的实例数据 (x_n, y_n) ：
 $\{(-4.200, -3.222), (-2.700, -2.093),$
 $(+0.300, +1.690), (+1.200, -0.348),$
 $(+3.800, +4.134)\}$, 注意 $x = 2.5$ 处的函数值不属于训练数据

基本术语：输入空间与输出空间

基本术语：输入空间与输出空间

- 在监督学习中，将模型输入与输出的所有可能取值的集合，分别称为输入空间与输出空间，并且通常将输入实例 x_n 和输出标签 y_n 分别看作定义在输入空间和输出空间上的随机变量 X 和 Y 的取值
- 可以是有限元素的集合，也可以是欧氏空间，...
- 可以是同一空间，也可以是不同空间

基本术语：特征空间

基本术语：特征空间

- 在监督学习中，每个具体的输入实例 x_n ，如果由特征向量表示，这时所有特征向量存在的空间称为特征空间，特征空间的每一维对应于一个特征
- 有时假设输入空间与特征空间为相同的空间，对它们不予区分
- 有时假设输入空间与特征空间为不同的空间，将实例从输入空间映射到特征空间，模型实际上都是定义在特征空间上的

数据更好的表示和数据比较

数据更好的表示

因为数据用向量表示，所以很多时候我们可以通过处理数据来更好的表示数据，主要有两种处理方式：

- 找到原始特征向量的低维近似向量：降维
- 使用原始特征向量的非线性高维组合：特征映射、核方法、深度神经网络

数据比较

- 相似的实例具有相似的标签
- 计算两个实例之间相似性和距离

数据的基本假设

联合概率分布

- 在统计机器学习中，通常假设输入与输出的随机变量 X 和 Y 遵循联合概率分布 $P(X, Y)$, $P(X, Y)$ 表示分布函数或分布密度函数
- 在学习过程中，假定这些联合概率分布存在，但对学习系统来说，联合概率分布的具体定义是未知的
- 训练数据与测试数据被看作是依联合概率分布 $P(X, Y)$ 独立同分布产生的

与数据表示相关的数学知识

线性代数和概率论

- 线性代数的基本概念：向量、矩阵、对称矩阵、正交矩阵、特征值、向量空间、基本子空间（第 2 章）
- 线性代数的几何：范数、内积、角度、正交性、投影、欧氏空间（第 3 章）
- 矩阵分解：特征值分解和奇异值分解（第 4 章）
- 概率论的基本概念：随机变量、联合概率分布（第 7 章）
- ...

1.3.3 模型

模型类型

- 非概率类模型：由决策函数 $Y = f(X)$ 表示，对具体的输入进行相应的输出预测时，写作 $y = f(x)$
- 概率类模型：由条件概率分布 $P(X, Y)$ 表示，对具体的输入进行相应的输出预测时，写作 $P(y|x)$

模型是函数

当模型是一种函数，给定特定输入实例（特征向量）时，会生成输出。如果考虑将输出视为实值输出，这时可以写作

$$f: \mathbb{R}^D \rightarrow \mathbb{R}, \quad (1)$$

其中输入向量 x 是 D 维（具有 D 个特征），函数 $f(x)$ 返回实数值。

函数类型

- 线性函数
- 非线性函数

线性模型：仿射函数

我们考虑仿射函数

$$f(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x} + \theta_0, \quad (2)$$

当 $\theta_0 = 0$ 时退化为标准的线性函数。

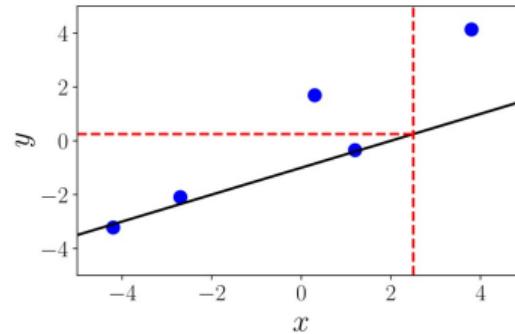


图 23: 预测函数在 $x = 2.5$ 时的预测: $f(2.5) = 0.25$.

非线性模型：非线性函数

考虑深度学习中的函数

$$f(\mathbf{x}) = f_L(f_{L-1}(\dots f_2(f_1(\mathbf{x}))). \quad (3)$$

其中 $f_i(\mathbf{x}) = \text{ReLU}(A_i \mathbf{x} + b_i) = (A_i \mathbf{x} + b_i)_+ = \max(A_i \mathbf{x} + b_i, 0)$ 是非线性函数，是非线性激活函数和仿射变换的复合。

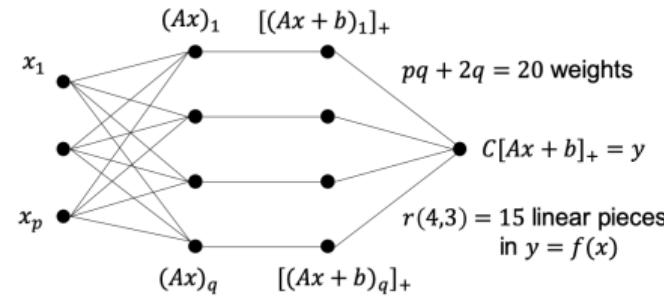


图 24: 数据向量 x 的分段线性函数的神经网络构造

模型是概率分布

模型应用概率的动机

- 对数据噪声建模
- 量化预测的置信度

概率工具

- 有限维参数的特殊分布: 多元概率分布
- 图的描述: 概率图模型

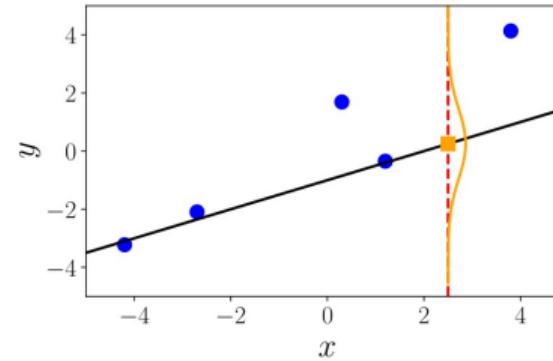


图 25: 预测函数 (黑色实心对角线) 及其在 $x=2.5$ 时的预测不确定性 (绘制为高斯分布)

利用概率进行建模

高斯判别模型

- 现实生活中，我们可以根据一个人的体型特征（身高、体重、三围、头发长短等）数据并利用高斯分布建模来推测一个人的生物学性别，这种利用体型特征推性别实际上在做分类问题。
- 当我们假定每一类数据的分布是含有参数的高斯分布时，这种分类方法就称为高斯判别分析（GDA）。
- 又由于这样构建的分类器的决策边界是二次的，又称为二次判别分析。

$$\hat{Y}(x) = \arg \max_Y p(Y|x; \theta)$$

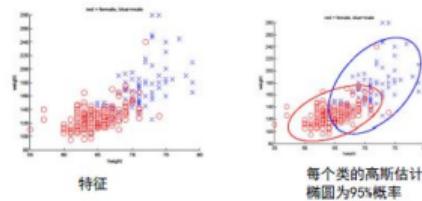


图 26: 男生和女生的身高和体重

假设空间

假设空间 (hypothesis space)

- 模型属于由输入空间到输出空间的映射集合，这个集合称为假设空间
- 假设空间的确定意味着学习范围的确定

函数模型的假设空间

假设空间可以定义为决策函数的集合：

$$\mathcal{F} = \{f \mid Y = f(X)\}, \quad (4)$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的变量。这时 \mathcal{F} 通常是由一个参数向量决定的函数族：

$$\mathcal{F} = \{f \mid Y = f_\theta(X), \theta \in R^n\}, \quad (5)$$

参数向量 θ 取值于 n 维欧氏空间 R^n ，称为参数空间。

概率模型的假设空间

假设空间也可以定义为条件概率的集合：

$$\mathcal{F} = \{P|P(Y|X)\}, \quad (6)$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的随机变量。这时 \mathcal{F} 通常是由一个参数向量决定的条件概率分布族：

$$\mathcal{F} = \{P|P_\theta(Y|X), \theta \in R^n\}, \quad (7)$$

参数向量 θ 取值于 n 维欧氏空间 R^n ，也称为参数空间。

与数据模型相关的数学知识

矩阵计算、概率论和信息论

- 线性代数：线性映射、线性变换、仿射映射、向量和矩阵函数、激活函数、泛函空间，...（第 2,3,6,10 章）
- 概率论和信息论：分布、似然、后验、信息熵、图模型，...（第 7,8,9 章）
- ...

1.3.4 学习：策略

有了模型的假设空间，机器学习需要考虑按照什么样的准则学习或选择最优的模型？

损失函数

监督学习问题是在假设空间 \mathcal{F} 中选取模型作为决策函数，对于给定的输入 X ，由 $f(X)$ 给出相应的输出 Y ，这个输出的预测值 $f(X)$ 与真实值 Y 可能一致也可能不一致，用一个损失函数或代价函数来度量预测错误的程度。损失函数是 $f(X)$ 和 Y 的非负实值函数，记作 $L(Y, f(X))$ 。

常见的损失函数：

- 0-1 损失函数

$$L(Y, f(X)) = \begin{cases} 1 & Y \neq f(X) \\ 0 & Y = f(X) \end{cases},$$

- 平方损失函数

$$L(Y, f(X)) = (Y - f(X))^2,$$

- 绝对损失函数

$$L(Y, f(X)) = |Y - f(X)|,$$

- 对数损失函数或对数似然损失函数

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

期望损失

期望损失 (expected loss)

损失函数值越小，模型就越好。由于模型的输入和输出 (X, Y) 是随机变量，遵循联合分布 $P(X, Y)$ ，所以损失函数的期望是

$$R_{\text{exp}}(f) = E_P[L(Y, f(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x)) P(x, y) dx dy, \quad (8)$$

这是理论上模型 $f(X)$ 关于联合分布 $P(X, Y)$ 的平均意义上的损失，称为风险函数或期望损失。

学习的目标就是选择期望损失最小的模型。由于联合分布 $P(X, Y)$ 是未知的， $R_{\text{exp}}(f)$ 不能直接计算。监督学习是一个病态问题!!!

经验损失

基本思路：用经验风险估计期望风险！理论基础：大数定律（第 7 章）。

经验损失 (empirical loss)

给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

模型 $f(X)$ 关于训练数据集的平均损失称为经验风险或经验损失，记作 R_{emp} ：

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (9)$$

经验风险最小化准则

经验风险最小化 (empirical risk minimization, ERM)

在假设空间、损失函数以及训练数据集确定的情况下，经验风险函数式(9)就可以确定。经验风险最小化的策略认为，经验风险最小的模型是最优的模型。根据这一策略，按照经验风险最小化求最优模型就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (10)$$

其中， \mathcal{F} 是假设空间。

当样本容量足够大时，经验风险最小化能保证有很好的学习效果。

过拟合

过拟合 (over-fitting)

- 过拟合：过拟合是指学习时选择的模型所包含的参数过多，以至出现这一模型对已知数据预测得很好，但对未知数据预测得很差的现象。
- 原因：对于特定模型 f (参数固定)，当来自训练数据经验风险估计低估期望风险时，会发生过度拟合现象。

结构风险

结构风险 (structural risk)

在假设空间、损失函数以及训练数据集确定的情况下，结构风险定义为在经验风险上加上表示模型复杂度的正则化项或罚项：

$$R_{srm}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (11)$$

其中 $J(f)$ 为模型的复杂度，是定义在假设空间 \mathcal{F} 上的泛函。模型 f 越复杂，复杂度 $J(f)$ 就越大；反之，模型 f 越简单，复杂度 $J(f)$ 就越小。也就是说，复杂度表示了对复杂模型的惩罚。 $\lambda \geq 0$ 是系数，用以权衡经验风险和模型复杂度。结构风险小需要经验风险与模型复杂度同时小。结构风险小的模型往往对训练数据以及未知的测试数据都有较好的预测。

结构风险最小化准则和正则化

结构风险最小化准则 (structural risk minimization, SRM)

结构风险最小化策略认为结构风险最小的模型是最优的模型。所以求最优模型，就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (12)$$

上述最优化问题一般也称为正则化 (regularization)，正则化是结构风险最小化策略的实现。

常用的正则化项

以最小二乘问题为例，一般最小二乘问题的损失函数：

$$\min_{\theta} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\theta\|^2$$

通过添加仅涉及 θ 的正则项可以改善病态最小二乘问题的稳定性。

比如添加易于计算导数的 L_2 正则项：

$$\min_{\theta} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\theta\|^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

也可以添加 L_1 正则项得到参数更为稀疏的模型：

$$\min_{\theta} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\theta\|^2 + \lambda \|\theta\|_1$$

模型泛化能力评估

泛化能力评估

- 理论：泛化误差上界，涉及概率不等式
- 实践：交叉验证

算法：优化问题

监督学习

目标函数依赖于损失函数和正则化项。

- 经验风险最小化问题: $\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$
- 结构风险最小化问题: $\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$

无监督学习

优化目标函数依赖于:

- 聚类: 样本与所属类别中心距离的最小化
- 降维: 样本从高维空间转换到低维空间过程中信息损失的最小化
- 概率模型估计: 模型生成数据概率的最大化

例 5

对于一个监督学习问题，设其数据集为

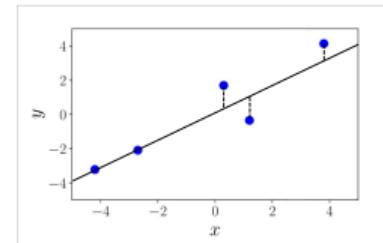
$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ，以一元线性回归作为模型，根据经验风险最小化可得：

$$\min_{(k, b) \in \mathbb{R}^2} \frac{1}{N} \sum_{i=1}^N |kx_i + b - y_i|$$

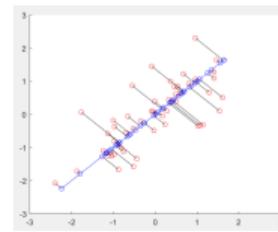
对于一个无监督学习问题，设其数据集为

$\{(x_1^1, x_1^2), (x_2^1, x_2^2), \dots, (x_n^1, x_n^2)\}$ ，使用 PCA 对其降维，将原来位置到新位置的距离看做信息损失（即原来位置到 1 维直线的距离）可得：

$$\min_{(a, b, c) \in \mathbb{R}^3} \frac{1}{N} \sum_{i=1}^N \frac{|ax_i^{(1)} + bx_i^{(2)} + c|}{\sqrt{a^2 + b^2}}$$



(a) 一元线性回归



(b) PCA 降维

图 27: 监督学习与无监督学习

优化算法

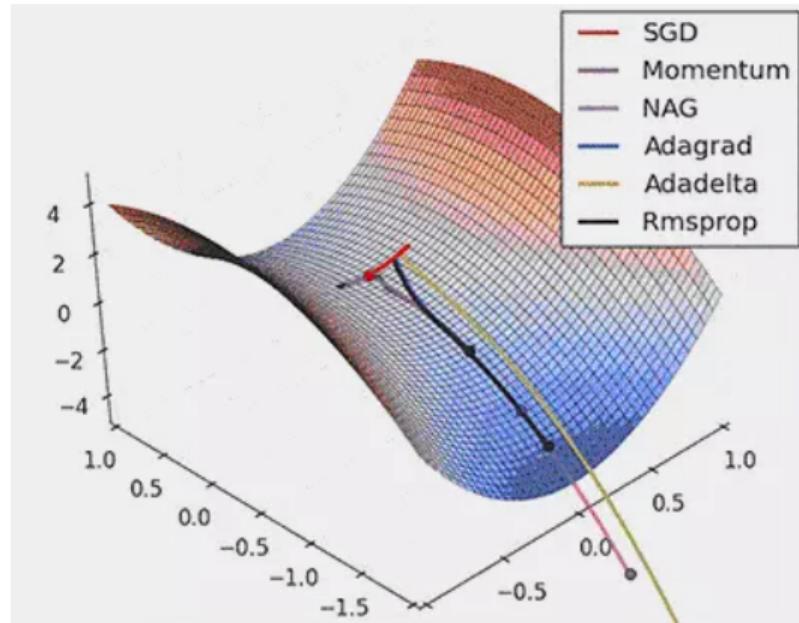
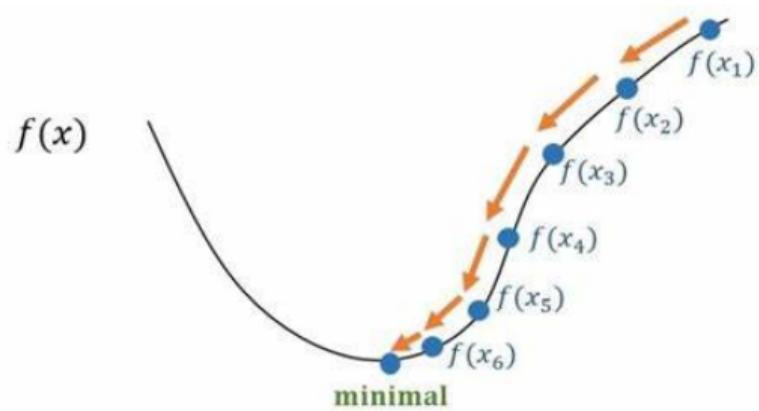
在求解模型时，会使用一些优化算法来找到“适合”数据的良好模型参数。根据最终得到的优化问题具有不同的性质，我们往往需要采用不同的优化算法。

- 梯度下降算法
- 随机梯度下降算法
- 牛顿法
- ...

优化算法

梯度下降法:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \nabla f(\mathbf{x}^{(k)})$$

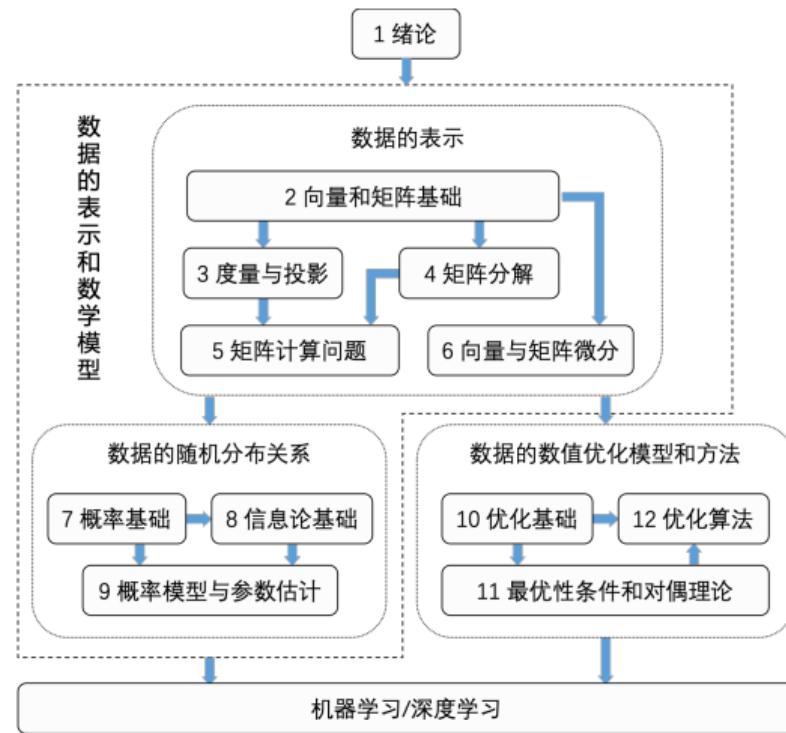


与学习相关的数学知识

矩阵计算、概率论和优化方法

- 矩阵计算：矩阵分解、线性方程组求解、最小二乘问题、特征值问题、向量和矩阵函数微分、梯度，…（第 4,5,6 章）
- 概率论：大数定律、概率平均、最大似然、最大后验、概率不等式，…（第 8,9 章）
- 优化方法：凸集、凸函数、凸优化问题、凸优化方法、对偶、KKT 条件、梯度下降、牛顿法、随机梯度下降，…（第 10,11,12 章）
- ...

1.3.5 所需的数学基础：本课程的内容组织架构图



所需数学基础

数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已。

数学给了这个上限无限可能！