



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习基础

金曜松

北京聚睿智能科技有限公司

2017 年 5 月 4 日



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

① Outline

② 学习算法

③ 容量、过拟合和欠拟合

④ 监督学习算法

⑤ 无监督学习算法

⑥ 超参数和验证集

⑦ 参考书目



学习算法

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习算法是一种能够从数据中学习的算法。

定义 (学习, Mitchell, 1997)

对于某类任务 T 和性能度量 P , 一个计算机程序被认为可以从经验 E 中学习是指, 通过经验 E 改进后, 它在任务 T 上由性能度量 P 衡量的性能有所提升。



任务 T

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 从“任务”的相对正式的定义上说，学习过程本身不能算是任务。学习是我们所谓的获取完成任务的能力。
- 通常机器学习任务定义为机器学习系统应该如何处理样本。
- 样本是指我们从某些希望机器学习系统处理的对象或事件中收集到的已经量化的**特征** (feature) 的集合。
- 我们通常会将样本表示成一个向量 $x \in \mathbb{R}^n$ ，其中向量的每一个元素 x_i 是一个特征。



任务 T

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 从“任务”的相对正式的定义上说，学习过程本身不能算是任务。学习是我们所谓的获取完成任务的能力。
- 通常机器学习任务定义为机器学习系统应该如何处理**样本**。
- 样本是指我们从某些希望机器学习系统处理的对象或事件中收集到的已经量化的**特征** (feature) 的集合。
- 我们通常会将样本表示成一个向量 $x \in \mathbb{R}^n$ ，其中向量的每一个元素 x_i 是一个特征。



任务 T

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 从“任务”的相对正式的定义上说，学习过程本身不能算是任务。学习是我们所谓的获取完成任务的能力。
- 通常机器学习任务定义为机器学习系统应该如何处理**样本**。
- 样本是指我们从某些希望机器学习系统处理的对象或事件中收集到的已经量化的**特征** (feature) 的集合。
- 我们通常会将样本表示成一个向量 $x \in \mathbb{R}^n$ ，其中向量的每一个元素 x_i 是一个特征。



任务 T

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 从“任务”的相对正式的定义上说，学习过程本身不能算是任务。学习是我们所谓的获取完成任务的能力。
- 通常机器学习任务定义为机器学习系统应该如何处理样本。
- 样本是指我们从某些希望机器学习系统处理的对象或事件中收集到的已经量化的**特征** (feature) 的集合。
- 我们通常会将样本表示成一个向量 $x \in \mathbb{R}^n$ ，其中向量的每一个元素 x_i 是一个特征。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

分类 在这类任务中，计算机程序需要指定某些输入属于 k 类中的哪一类。为了完成这个任务，学习算法通常会返回一个函数 $f: \mathbb{R}^n \rightarrow \{1, \dots, k\}$ 。当 $y = f(x)$ 时，模型将向量 x 所代表的输入分类到数字码 y 所代表的类别。

输入缺失分类 当输入向量的每个度量不被保证的时候，分类问题将会变得更有挑战性。为了解决分类任务，学习算法只需要定义一个从输入向量映射到输出类别的函数。当一些输入可能丢失时，学习算法必须学习一组函数，而不是单个分类函数。每个函数对应着分类具有不同缺失输入子集的 x 。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

分类 在这类任务中，计算机程序需要指定某些输入属于 k 类中的哪一类。为了完成这个任务，学习算法通常会返回一个函数 $f: \mathbb{R}^n \rightarrow \{1, \dots, k\}$ 。当 $y = f(x)$ 时，模型将向量 x 所代表的输入分类到数字码 y 所代表的类别。

输入缺失分类 当输入向量的每个度量不被保证的时候，分类问题将会变得更有挑战性。为了解决分类任务，学习算法只需要定义一个从输入向量映射到输出类别的函数。当一些输入可能丢失时，学习算法必须学习一组函数，而不是单个分类函数。每个函数对应着分类具有不同缺失输入子集的 x 。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

回归 在这类任务中，计算机程序需要对给定输入预
测数值。为了解决这个任务，学习算法需要输出
函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 。除了返回结果的形式不一样
外，这类问题和分类问题是很像的。

转录 这类任务中，机器学习系统观测一些相对非结
构化表示的数据，并转录信息为离散的文本形
式。例如：语音识别，google 街景处理街道编号

机器翻译 在机器翻译任务中，输入是一种语言的符号序
列，计算机程序必须将其转化成另一种语言的
符号序列。这通常适用于自然语言，如将英语译
成汉语。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

回归 在这类任务中，计算机程序需要对给定输入预测数值。为了解决这个任务，学习算法需要输出函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 。除了返回结果的形式不一样外，这类问题和分类问题是很像的。

转录 这类任务中，机器学习系统观测一些相对非结构化表示的数据，并转录信息为离散的文本形式。例如：语音识别，google 街景处理街道编号

机器翻译 在机器翻译任务中，输入是一种语言的符号序列，计算机程序必须将其转化成另一种语言的符号序列。这通常适用于自然语言，如将英语译成汉语。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

回归 在这类任务中，计算机程序需要对给定输入预测数值。为了解决这个任务，学习算法需要输出函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 。除了返回结果的形式不一样外，这类问题和分类问题是很像的。

转录 这类任务中，机器学习系统观测一些相对非结构化表示的数据，并转录信息为离散的文本形式。例如：语音识别，google 街景处理街道编号

机器翻译 在机器翻译任务中，输入是一种语言的符号序列，计算机程序必须将其转化成另一种语言的符号序列。这通常适用于自然语言，如将英语译成汉语。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

结构化输出 结构化输出任务的输出是向量或者其他包含多个值的数据结构，并且构成输出的这些不同元素间具有重要关系。这是一个很大的范畴，包括上述转录任务和翻译任务在内的很多其他任务。

异常检测 在这类任务中，计算机程序在一组事件或对象中筛选，并标记不正常或非典型的个体。异常检测任务的一个示例是网络攻击。

合成和采样 在这类任务中，机器学习程序生成一些和训练数据相似的新样本。通过机器学习，合成和采样可能在媒体应用中非常有用，还可以用来写小说。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

结构化输出 结构化输出任务的输出是向量或者其他包含多个值的数据结构，并且构成输出的这些不同元素间具有重要关系。这是一个很大的范畴，包括上述转录任务和翻译任务在内的很多其他任务。

异常检测 在这类任务中，计算机程序在一组事件或对象中筛选，并标记不正常或非典型的个体。异常检测任务的一个示例是网络攻击。

合成和采样 在这类任务中，机器学习程序生成一些和训练数据相似的新样本。通过机器学习，合成和采样可能在媒体应用中非常有用，还可以用来写小说。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

结构化输出 结构化输出任务的输出是向量或者其他包含多个值的数据结构，并且构成输出的这些不同元素间具有重要关系。这是一个很大的范畴，包括上述转录任务和翻译任务在内的很多其他任务。

异常检测 在这类任务中，计算机程序在一组事件或对象中筛选，并标记不正常或非典型的个体。异常检测任务的一个示例是网络攻击。

合成和采样 在这类任务中，机器学习程序生成一些和训练数据相似的新样本。通过机器学习，合成和采样可能在媒体应用中非常有用，还可以用来写小说。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

缺失值填补 在这类任务中，机器学习算法给定一个新样本 $x \in \mathbb{R}^n$ ， x 中某些元素 x_i 缺失。算法必须填补这些缺失值。

去噪 在这类任务中，机器学习算法的输入是，干净样本 $x \in \mathbb{R}^n$ 经过未知损坏过程后得到的损坏样本 $\tilde{x} \in \mathbb{R}^n$ 。算法根据损坏后的样本 \tilde{x} 预测干净的样本 x ，或者更一般地预测条件概率分布 $Pr(x|\tilde{x})$ 。



常见的机器学习任务

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

缺失值填补 在这类任务中，机器学习算法给定一个新样本 $x \in \mathbb{R}^n$ ， x 中某些元素 x_i 缺失。算法必须填补这些缺失值。

去噪 在这类任务中，机器学习算法的输入是，干净样本 $x \in \mathbb{R}^n$ 经过未知损坏过程后得到的损坏样本 $\tilde{x} \in \mathbb{R}^n$ 。算法根据损坏后的样本 \tilde{x} 预测干净的样本 x ，或者更一般地预测条件概率分布 $Pr(x|\tilde{x})$ 。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

根据学习过程中的不同经验，机器学习算法可以大致分类为**无监督算法**和**监督算法**。

无监督算法 训练含有很多特征的数据集，然后学习出这个数据集上有用的结构性质。

监督算法 训练含有很多特征的数据集，不过数据集中的样本都有一个标签。

- 大致说来，无监督算法涉及到观察随机向量 x 的好几个样本，试图显式或隐式地学习出概率分布 $p(x)$ ，或者是该分布一些有意思性质；
- 监督算法包含观察随机向量 x 及其相关联的值或向量 y ，然后从 x 预测 y ，通常是估计 $p(y|x)$ 。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

根据学习过程中的不同经验，机器学习算法可以大致分类为**无监督算法**和**监督算法**。

无监督算法 训练含有很多特征的数据集，然后学习出这个数据集上有用的结构性质。

监督算法 训练含有很多特征的数据集，不过数据集中的样本都有一个标签。

- 大致说来，无监督算法涉及到观察随机向量 x 的好几个样本，试图显式或隐式地学习出概率分布 $p(x)$ ，或者是该分布一些有意思性质；
- 监督算法包含观察随机向量 x 及其相关联的值或向量 y ，然后从 x 预测 y ，通常是估计 $p(y|x)$ 。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

根据学习过程中的不同经验，机器学习算法可以大致分类为**无监督算法**和**监督算法**。

无监督算法 训练含有很多特征的数据集，然后学习出这个数据集上有用的结构性质。

监督算法 训练含有很多特征的数据集，不过数据集中的样本都有一个标签。

- 大致说来，无监督算法涉及到观察随机向量 x 的好几个样本，试图显式或隐式地学习出概率分布 $p(x)$ ，或者是该分布一些有意思性质；
- 监督算法包含观察随机向量 x 及其相关联的值或向量 y ，然后从 x 预测 y ，通常是估计 $p(y|x)$ 。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

根据学习过程中的不同经验，机器学习算法可以大致分类为**无监督算法**和**监督算法**。

无监督算法 训练含有很多特征的数据集，然后学习出这个数据集上有用的结构性质。

监督算法 训练含有很多特征的数据集，不过数据集中的样本都有一个标签。

- 大致说来，无监督算法涉及到观察随机向量 x 的好几个样本，试图显式或隐式地学习出概率分布 $p(x)$ ，或者是该分布一些有意思性质；
- 监督算法包含观察随机向量 x 及其相关联的值或向量 y ，然后从 x 预测 y ，通常是估计 $p(y|x)$ 。



泛化

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望泛化误差 (也被称为测试误差) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - 降低训练误差。
 - 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
 - 泛化误差被定义为新输入的误差期望。
 - 训练集中分出测试集，来评估机器学习模型的泛化误差。
 - 以下是决定机器学习算法效果是否好的因素：
 - 降低训练误差。
 - 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - 降低训练误差。
 - 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - 降低训练误差。
 - 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - ① 降低训练误差。
 - ② 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - ① 降低训练误差。
 - ② 缩小训练误差和测试误差的差距。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

机器学习的主要挑战是我们的算法必须能够在先前未观测的新输入上表现良好，而不只是在训练集上表现良好。在先前未观测到的输入上表现良好的能力被称为**泛化** (generalization)。

- 使用某个训练集，目标是降低训练误差。
- 我们也希望**泛化误差** (也被称为**测试误差**) 很低。
- 泛化误差被定义为新输入的误差期望。
- 训练集中分出测试集，来评估机器学习模型的泛化误差。
- 以下是决定机器学习算法效果是否好的因素：
 - ① 降低训练误差。
 - ② 缩小训练误差和测试误差的差距。



主要挑战

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- ① 欠拟合是指模型不能在训练集上获得足够低的误差。
- ② 过拟合是指训练误差和测试误差之间的差距太大。

通过调整模型的容量，我们可以控制模型是否偏向于过拟合或者欠拟合。

- 模型的容量是指其拟合各种函数的能力。
- 容量低的模型可能很难拟合训练集。
- 容量高的模型可能会过拟合。



主要挑战

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- ① **欠拟合**是指模型不能在训练集上获得足够低的误差。
- ② **过拟合**是指训练误差和测试误差之间的差距太大。

通过调整模型的容量，我们可以控制模型是否偏向于过拟合或者欠拟合。

- 模型的容量是指其拟合各种函数的能力。
- 容量低的模型可能很难拟合训练集。
- 容量高的模型可能会过拟合。



主要挑战

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- ① **欠拟合**是指模型不能在训练集上获得足够低的误差。
- ② **过拟合**是指训练误差和测试误差之间的差距太大。

通过调整模型的容量，我们可以控制模型是否偏向于过拟合或者欠拟合。

- 模型的容量是指其拟合各种函数的能力。
- 容量低的模型可能很难拟合训练集。
- 容量高的模型可能会过拟合。



主要挑战

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- ① 欠拟合是指模型不能在训练集上获得足够低的误差。
- ② 过拟合是指训练误差和测试误差之间的差距太大。

通过调整模型的容量，我们可以控制模型是否偏向于过拟合或者欠拟合。

- 模型的容量是指其拟合各种函数的能力。
- 容量低的模型可能很难拟合训练集。
- 容量高的模型可能会过拟合。



主要挑战

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- ① 欠拟合是指模型不能在训练集上获得足够低的误差。
- ② 过拟合是指训练误差和测试误差之间的差距太大。

通过调整模型的容量，我们可以控制模型是否偏向于过拟合或者欠拟合。

- 模型的容量是指其拟合各种函数的能力。
- 容量低的模型可能很难拟合训练集。
- 容量高的模型可能会过拟合。



假设空间

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

一种控制训练算法容量的方法是选择**假设空间**，即学习算法可以选择为解决方案的函数集。

- 线性回归函数将关于其输入的所有线性函数作为假设空间。
- 广义线性回归的假设空间包括多项式函数，而非仅有线性函数。

这样做就增加了模型的容量。



假设空间

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

一种控制训练算法容量的方法是选择**假设空间**，即学习算法可以选择为解决方案的函数集。

- 线性回归函数将关于其输入的所有线性函数作为假设空间。
- 广义线性回归的假设空间包括多项式函数，而非仅有线性函数。

这样做就增加了模型的容量。



一个例子

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

① 简单回归：一次多项式

$$\hat{y} = b + wx. \quad (1)$$

② 关于 x 的二次函数模型：

$$\hat{y} = b + w_1x + w_2x^2. \quad (2)$$

尽管该模型是输入的二次函数，但输出仍是参数的线性
函数。因此我们仍然可以用规范方程得到模型的闭解。

③ 继续增加高次项：例如 9 次项

$$\hat{y} = b + \sum_{i=1}^9 w_i x^i. \quad (3)$$



一个例子

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

① 简单回归：一次多项式

$$\hat{y} = b + wx. \quad (1)$$

② 关于 x 的二次函数模型：

$$\hat{y} = b + w_1x + w_2x^2. \quad (2)$$

尽管该模型是输入的二次函数，但输出仍是参数的线性
函数。因此我们仍然可以用规范方程得到模型的闭解。

③ 继续增加高次项：例如 9 次项

$$\hat{y} = b + \sum_{i=1}^9 w_i x^i. \quad (3)$$



一个例子

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

① 简单回归：一次多项式

$$\hat{y} = b + wx. \quad (1)$$

② 关于 x 的二次函数模型：

$$\hat{y} = b + w_1x + w_2x^2. \quad (2)$$

尽管该模型是输入的二次函数，但输出仍是参数的线性
函数。因此我们仍然可以用规范方程得到模型的闭解。

③ 继续增加高次项：例如 9 次项

$$\hat{y} = b + \sum_{i=1}^9 w_i x^i. \quad (3)$$



例子 (续)

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 当机器学习算法的容量适合于所执行任务的复杂度和所提供训练数据的数量时，算法效果通常会最佳。
- 容量不足的模型不能解决复杂任务。
- 容量高的模型能够解决复杂的任务，但是当其容量高于任务所需时，有可能会过拟合。



例子 (续)

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 当机器学习算法的容量适合于所执行任务的复杂度和所提供训练数据的数量时，算法效果通常会最佳。
- 容量不足的模型不能解决复杂任务。
- 容量高的模型能够解决复杂的任务，但是当其容量高于任务所需时，有可能会过拟合。



例子 (续)

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- 当机器学习算法的容量适合于所执行任务的复杂度和所提供训练数据的数量时，算法效果通常会最佳。
- 容量不足的模型不能解决复杂任务。
- 容量高的模型能够解决复杂的任务，但是当其容量高于任务所需时，有可能会过拟合。



例子 (续)

ML

金曜松

Outline

学习算法

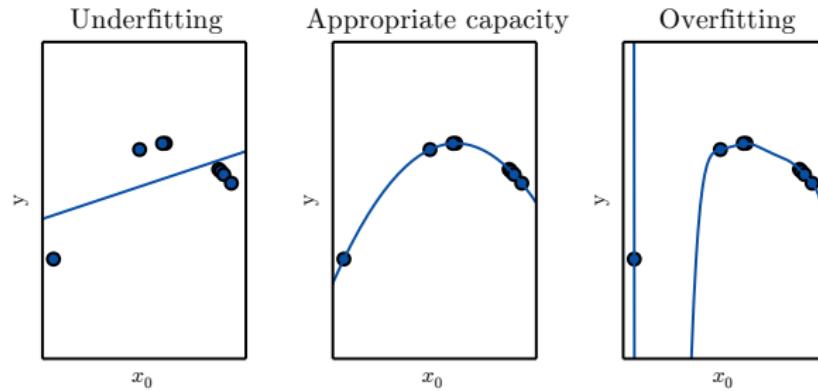
容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目



- (左) 用一个线性函数拟合数据会导致欠拟合——曲率信息缺失。
- (中) 用二次函数拟合数据在未观察到的点上泛化得很好。
- (右) 一个 9 阶的多项式拟合数据会导致过拟合。



例子 (续)

ML

金曜松

Outline

学习算法

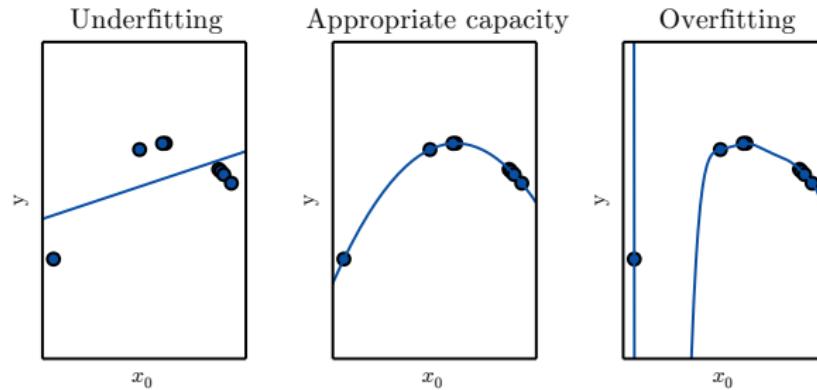
容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目



- (左) 用一个线性函数拟合数据会导致欠拟合——曲率信息缺失。
- (中) 用二次函数拟合数据在未观察到的点上泛化得很好。
- (右) 一个 9 阶的多项式拟合数据会导致过拟合。



例子 (续)

ML

金曜松

Outline

学习算法

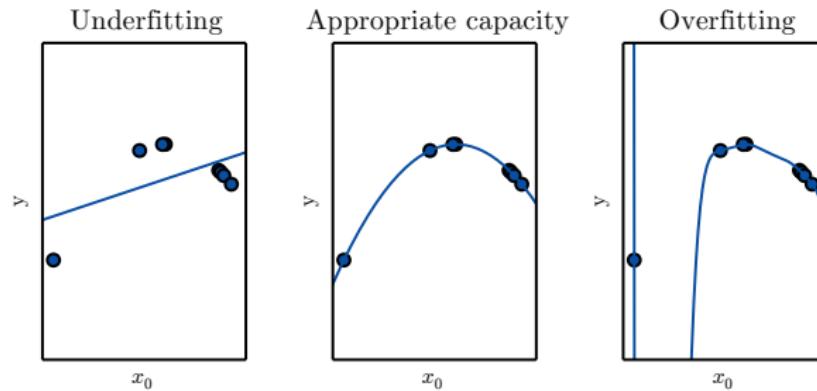
容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目



- (左) 用一个线性函数拟合数据会导致欠拟合——曲率信息缺失。
- (中) 用二次函数拟合数据在未观察到的点上泛化得很好。
- (右) 一个 9 阶的多项式拟合数据会导致过拟合。



没有免费午餐定理

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

在某种意义上，没有一个机器学习算法总是比其他的要好。

- 我们能够设想的最先进的算法和简单地将所有点归为同一类的简单算法有着相同的平均性能（在所有可能的任务上）。
- 这些结论仅在我们考虑所有可能的数据生成分布时才成立。
- 在真实世界应用中，如果我们对遇到的概率分布进行假设的话，那么我们可以设计在这些分布上效果良好的学习算法。
- 机器学习研究的目标不是找一个通用学习算法或是绝对最好的学习算法。



没有免费午餐定理

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

在某种意义上，没有一个机器学习算法总是比其他的要好。

- 我们能够设想的最先进的算法和简单地将所有点归为同一类的简单算法有着相同的平均性能（在所有可能的任务上）。
- 这些结论仅在我们考虑所有可能的数据生成分布时才成立。
- 在真实世界应用中，如果我们对遇到的概率分布进行假设的话，那么我们可以设计在这些分布上效果良好的学习算法。
- 机器学习研究的目标不是找一个通用学习算法或是绝对最好的学习算法。



没有免费午餐定理

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

在某种意义上，没有一个机器学习算法总是比其他的要好。

- 我们能够设想的最先进的算法和简单地将所有点归为同一类的简单算法有着相同的平均性能（在所有可能的任务上）。
- 这些结论仅在我们考虑所有可能的数据生成分布时才成立。
- 在真实世界应用中，如果我们对遇到的概率分布进行假设的话，那么我们可以设计在这些分布上效果良好的学习算法。
- 机器学习研究的目标不是找一个通用学习算法或是绝对最好的学习算法。



没有免费午餐定理

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

在某种意义上，没有一个机器学习算法总是比其他的要好。

- 我们能够设想的最先进的算法和简单地将所有点归为同一类的简单算法有着相同的平均性能（在所有可能的任务上）。
- 这些结论仅在我们考虑所有可能的数据生成分布时才成立。
- 在真实世界应用中，如果我们对遇到的概率分布进行假设的话，那么我们可以设计在这些分布上效果良好的学习算法。
- 机器学习研究的目标不是找一个通用学习算法或是绝对最好的学习算法。



监督学习

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 监督学习算法是给定一组输入 x 和输出 y 的训练集，学
习如何关联输入和输出。
- 在许多情况下，输出 y 很难自动收集，必须由人来提供
“监督”。



监督学习

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 监督学习算法是给定一组输入 x 和输出 y 的训练集，学
习如何关联输入和输出。
- 在许多情况下，输出 y 很难自动收集，必须由人来提供
“监督”。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 基于线性函数 $w^T x + b$ 。
- 不同于逻辑回归的是，支持向量机不输出概率，只输出类别。
- 当 $w^T x + b$ 为正时，SVM 预测属于正类。
- 当 $w^T x + b$ 为负时，SVM 预测属于负类。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 基于线性函数 $w^T x + b$ 。
- 不同于逻辑回归的是，支持向量机不输出概率，只输出类别。
 - 当 $w^T x + b$ 为正时，SVM 预测属于正类。
 - 当 $w^T x + b$ 为负时，SVM 预测属于负类。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 基于线性函数 $w^\top x + b$ 。
- 不同于逻辑回归的是，支持向量机不输出概率，只输出类别。
- 当 $w^\top x + b$ 为正时，SVM 预测属于正类。
- 当 $w^\top x + b$ 为负时，SVM 预测属于负类。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 基于线性函数 $w^\top x + b$ 。
- 不同于逻辑回归的是，支持向量机不输出概率，只输出类别。
- 当 $w^\top x + b$ 为正时，SVM 预测属于正类。
- 当 $w^\top x + b$ 为负时，SVM 预测属于负类。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

核技巧 (kernel tricks) 观察到许多机器学习算法都可以写成样本间点乘的形式。

- SVM 中的线性函数可以重写为

$$w^\top x + b = b + \sum_{i=1}^m \alpha_i x^{(i)} \top x^{(i)}, \quad (4)$$

其中， $x^{(i)}$ 是训练样本， α 是系数向量。

- 我们将 x 替换为特征函数 $\phi(x)$ 的输出，点乘替换为被称为核函数 (kernel function) 的函数 $k(x, x^{(i)}) = \phi(x) \cdot \phi(x^{(i)})$ 。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

核技巧 (kernel tricks) 观察到许多机器学习算法都可以写成样本间点乘的形式。

- SVM 中的线性函数可以重写为

$$w^\top x + b = b + \sum_{i=1}^m \alpha_i x^\top x^{(i)}, \quad (4)$$

其中， $x^{(i)}$ 是训练样本， α 是系数向量。

- 我们将 x 替换为特征函数 $\phi(x)$ 的输出，点乘替换为被称为核函数 (kernel function) 的函数 $k(x, x^{(i)}) = \phi(x) \cdot \phi(x^{(i)})$ 。

- 使用核估计替换点乘之后，我们可以使用如下函数进行预测

$$f(x) = b + \sum_i \alpha_i k(x, x^{(i)}). \quad (5)$$

- 这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。
- 最常用的核函数是高斯核 (Gaussian kernel)

$$k(u, v) = \mathcal{N}(u - v; 0, \sigma^2 I), \quad (6)$$

其中 $\mathcal{N}(x; \mu, \Sigma)$ 是标准正态密度。这个核也被称为径向基函数 (radial basis function) 核。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 使用核估计替换点乘之后，我们可以使用如下函数进行预测

$$f(x) = b + \sum_i \alpha_i k(x, x^{(i)}). \quad (5)$$

- 这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。
- 最常用的核函数是高斯核 (Gaussian kernel)

$$k(u, v) = \mathcal{N}(u - v; 0, \sigma^2 I), \quad (6)$$

其中 $\mathcal{N}(x; \mu, \Sigma)$ 是标准正态密度。这个核也被称为径向基函数 (radial basis function) 核。



- 使用核估计替换点乘之后，我们可以使用如下函数进行预测

$$f(x) = b + \sum_i \alpha_i k(x, x^{(i)}). \quad (5)$$

- 这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。这个函数关于 x 是非线性的，关于 $\phi(x)$ 是线性的。 α 和 $f(x)$ 之间的关系也是线性的。
- 最常用的核函数是高斯核 (Gaussian kernel)

$$k(u, v) = \mathcal{N}(u - v; 0, \sigma^2 I), \quad (6)$$

其中 $\mathcal{N}(x; \mu, \Sigma)$ 是标准正态密度。这个核也被称为径向基函数 (radial basis function) 核。



无监督学习

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 无监督算法只处理“特征”，不操作监督信号。
- 监督和无监督算法之间的区别没有规范严格的定义，因
为没有客观的判断来区分监督者提供的值是特征还是目
标。



无监督学习

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 无监督算法只处理“特征”，不操作监督信号。
- 监督和无监督算法之间的区别没有规范严格的定义，因
为没有客观的判断来区分监督者提供的值是特征还是目
标。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$, h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - 每一个训练样本分配到最近的中心点 $\mu^{(j)}$ 所代表的类别。
 - 每一个中心点 $\mu^{(j)}$ 更新为聚类 j 中所有训练样本 $x^{(j)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$, h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - 每一个训练样本分配到最近的中心点 $\mu^{(j)}$ 所代表的类别。
 - 每一个中心点 $\mu^{(j)}$ 更新为聚类 j 中所有训练样本 $x^{(i)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$ ， h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - 每一个训练样本分配到最近的中心点 $\mu^{(j)}$ 所代表的聚类 j 。
 - 每一个中心点 $\mu^{(j)}$ 更新为聚类 j 中所有训练样本 $x^{(i)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$ ， h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - ① 每个训练样本分配到最近的中心点 $\mu^{(i)}$ 所代表的聚类 i 。
 - ② 每一个中心点 $\mu^{(i)}$ 更新为聚类 i 中所有训练样本 $x^{(j)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$ ， h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - ① 每个训练样本分配到最近的中心点 $\mu^{(i)}$ 所代表的聚类 i 。
 - ② 每一个中心点 $\mu^{(i)}$ 更新为聚类 i 中所有训练样本 $x^{(j)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$ ， h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - ① 每个训练样本分配到最近的中心点 $\mu^{(i)}$ 所代表的聚类 i 。
 - ② 每一个中心点 $\mu^{(i)}$ 更新为聚类 i 中所有训练样本 $x^{(j)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



- k-均值聚类算法将训练集分成 k 个靠近彼此的不同聚类。
- 因此我们可以认为该算法提供了 k -维的 one-hot 编码向量 h 以表示输入 x 。
- 当 x 属于聚类 i 时，有 $h_i = 1$ ， h 的其他项为零。
- k-means 聚类初始化 k 个不同的中心点 $\{\mu^{(1)}, \dots, \mu^{(k)}\}$ ，然后迭代交换两个不同的步骤直到收敛。
 - ① 每个训练样本分配到最近的中心点 $\mu^{(i)}$ 所代表的聚类 i 。
 - ② 每一个中心点 $\mu^{(i)}$ 更新为聚类 i 中所有训练样本 $x^{(j)}$ 的均值。
- 聚类问题本身是病态的。没有单一的标准去度量聚类的数据在真实世界中效果如何。



超参数

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



超参数

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



超参数

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



超参数

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- 大多数机器学习算法都有超参数，可以设置来控制算法行为。
- 超参数的值不是通过学习算法本身学习出来的（尽管我们可以设计一个嵌套的学习过程，一个学习算法为另一个学习算法学出最优超参数）。
- 在前面多项式回归示例中，多项式次数即为超参数。
- 有时一个选项被设为学习算法不用学习的超参数，是因为它太难优化了。
- 更多的情况是，该选项必须是超参数，因为它不适合在训练集上学习。这适用于控制模型容量的所有超参数。
- 如果在训练集上学习超参数，这些超参数总是趋向于最大可能的模型容量，导致过拟合。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



验证集

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 为了解决这个问题，我们需要一个训练算法观测不到的验证集 (validation set)。
- 注意测试集 (test set) 和验证集的区别。
- 我们总是从训练数据中构建验证集。
 - 我们将训练数据分成两个不相交的子集。
 - 其中一个用于学习参数。
 - 另一个作为验证集，用于估计训练中或训练后的泛化误差，更新超参数。
 - 用于挑选超参数的数据子集被称为验证集。
 - 通常，80% 的训练数据用于训练，20% 用于验证。
- 由于验证集是用来“训练”超参数的，尽管验证集的误差通常会比训练集误差小，验证集会低估泛化误差。
- 所有超参数优化完成之后，泛化误差可能会通过测试集来估计。



交叉验证

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 一个小规模的测试集意味着平均测试误差估计的统计不确定，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。
- 当数据集太小时，有替代方法允许我们使用所有的样本估计平均测试误差，代价是增加了计算量。
- 这些过程是基于在原始数据上随机采样或分离出的不同数据集上重复训练和测试的想法。
- 最常见的是 k -折交叉验证过程，将数据集分成 k 个不重合的子集。
- 测试误差可以估计为 k 次计算后的平均测试误差。



交叉验证

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 一个小规模的测试集意味着平均测试误差估计的统计不确定性，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。
- 当数据集太小时，有替代方法允许我们使用所有的样本估计平均测试误差，代价是增加了计算量。
- 这些过程是基于在原始数据上随机采样或分离出的不同数据集上重复训练和测试的想法。
- 最常见的是 k -折交叉验证过程，将数据集分成 k 个不重合的子集。
- 测试误差可以估计为 k 次计算后的平均测试误差。



交叉验证

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- 一个小规模的测试集意味着平均测试误差估计的统计不确定性，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。
- 当数据集太小时，有替代方法允许我们使用所有的样本估计平均测试误差，代价是增加了计算量。
- 这些过程是基于在原始数据上随机采样或分离出的不同数据集上重复训练和测试的想法。
- 最常见的是 k -折交叉验证过程，将数据集分成 k 个不重合的子集。
- 测试误差可以估计为 k 次计算后的平均测试误差。



交叉验证

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

- 一个小规模的测试集意味着平均测试误差估计的统计不确定性，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。
- 当数据集太小时，有替代方法允许我们使用所有的样本估计平均测试误差，代价是增加了计算量。
- 这些过程是基于在原始数据上随机采样或分离出的不同数据集上重复训练和测试的想法。
- 最常见的是 k -折交叉验证过程，将数据集分成 k 个不重合的子集。
- 测试误差可以估计为 k 次计算后的平均测试误差。



交叉验证

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- 一个小规模的测试集意味着平均测试误差估计的统计不确定性，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。
- 当数据集太小时，有替代方法允许我们使用所有的样本估计平均测试误差，代价是增加了计算量。
- 这些过程是基于在原始数据上随机采样或分离出的不同数据集上重复训练和测试的想法。
- 最常见的是 k -折交叉验证过程，将数据集分成 k 个不重合的子集。
- 测试误差可以估计为 k 次计算后的平均测试误差。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况下代表（输入、目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 在无监督学习的情况下仅用于输入 $x^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
 - ① 单个样本上的误差可用于计算平均值周围的置信区间。
 - ② 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证 k-fold cross-validation

ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



ML

金曜松

Outline

学习算法

容量、过拟合和欠拟合

监督学习算法

无监督学习算法

超参数和验证集

参考书目

- ① 当给定数据集 \mathbb{D} 对于简单的训练/测试或训练/验证分割而言太小难以产生泛化误差的准确估计时，k-折交叉验证算法可以用于估计学习算法 A 的泛化误差。
- ② 数据集 \mathbb{D} 包含的元素是抽象的样本 $z^{(i)}$ （对于第 i 个样本），
 - 在监督学习的情况代表（输入，目标）对 $z^{(i)} = (x^{(i)}, y^{(i)})$ ；
 - 无监督学习的情况下仅用于输入 $z^{(i)} = x^{(i)}$ 。
- ③ 该算法返回 \mathbb{D} 中每个示例的误差向量 e ，其均值是估计的泛化误差。
- ④ 单个样本上的误差可用于计算平均值周围的置信区间。
- ⑤ 当算法 A 误差的置信区间低于并且不与算法 B 的置信区间相交时，算法 A 比算法 B 更好。



k-折交叉验证算法

ML

金曜松

Outline

学习算法

容量、过拟合
和欠拟合

监督学习算法

无监督学习算
法

超参数和验证
集

参考书目

Define KFoldXV (\mathbb{D} , A, L, k):

Require \mathbb{D} 为给定数据集，其中元素为 $z^{(i)}$

Require A 为学习算法，可视为一个函数（使用数据集
作为输入，输出一个学好的函数）

Require L 为损失函数，可视为来自学好的函数 f，将样
本 $z^{(i)} \in \mathbb{D}$ 映射到 \mathbb{R} 中标量的函数。

Require k 为折数

过程 将 D 分为 k 个互斥子集 \mathbb{D}_i ，它们的并集为 \mathbb{D} :

for i from 1 to k do

$f_i = A(\mathbb{D} \setminus \mathbb{D}_i)$

 for $z^{(j)}$ in \mathbb{D}_i

$e_j = L(f_i, z^{(j)})$

 end for

end for

Return e



参考书目

ML

金曜松

Outline

学习算法

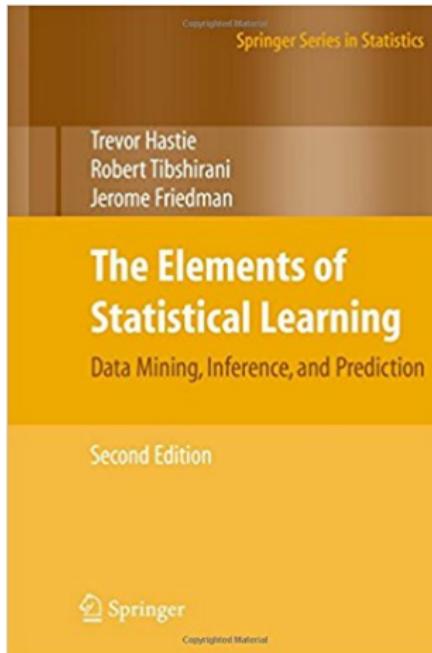
容量、过拟合和欠拟合

监督学习算法

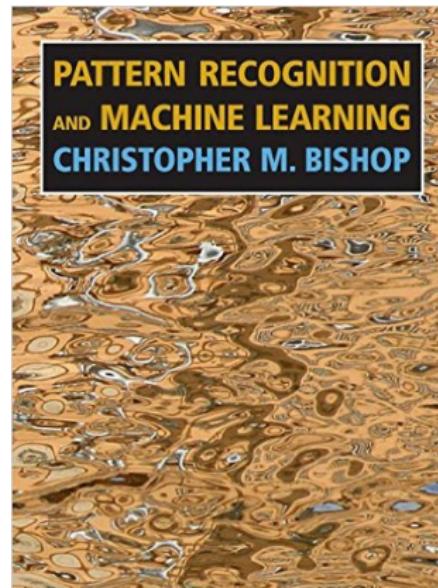
无监督学习算法

超参数和验证集

参考书目



(a)



(b)



参考书目

ML

金曜松

Outline

学习算法

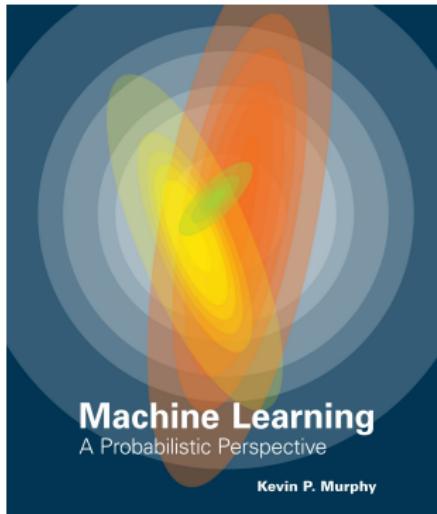
容量、过拟合和欠拟合

监督学习算法

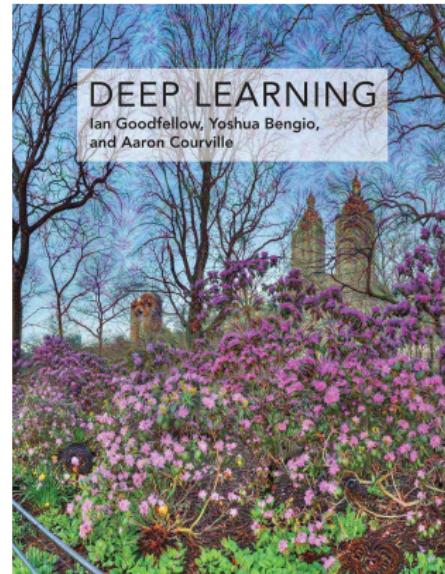
无监督学习算法

超参数和验证集

参考书目



(c)



(d)