1 讲座 - "欢迎和介绍

欢迎参加 CS-E4740 联合学习课程。虽然该课程可以完全远程完成，但我们强烈建议您参加现场活动，以方便学习。任何现场活动都将被录制并通过 YouTube 频道提供给学生。

本课程的基本课程（5 学分）包括讲座（课程表在这里）和相应的编码作业（课程表在这里）。我们通过测验（在 "我的课程 "页面上实施）来测试您完成编码作业的情况。您可以通过完成学生项目（参见第 1.8 节）将课程升级为扩展版（10 学分）。

1.1 学习目标

本讲座提供

- 介绍课程主题和在更广泛课程中的定位、
- 讨论学习目标、作业和学生项目、
- 课程表概览。

1.2 导言

我们身边的各种设备，如智能手机或可穿戴设备，都会产生分散的本地数据集[6-10]。这些本地数据集通常具有内在的网络结构，这种结构源于功能限制（设备之间的连通性）或统计相似性。

例如，大流行病的高精度管理利用联系网络将患者生成的本地数据集联系起来。网络医学通过共病网络关联疾病数据[11]。社会科学利用熟人概念来关联从好友那里收集的数据[12]。网络结构的另一个例子是芬兰气象研究所（FMI）气象站收集的观测数据。芬兰气象研究所的每个气象站都会生成一个本地数据集，这些数据集往往与附近气象站的统计属性相似。

FL 是分布式优化技术的总称，用于从本地数据集的分散集合中训练机器学习（ML）模型 [13-17]。其理念是在数据生成地点（如智能手机或心率传感器）执行 ML 模型训练过程中产生的计算，如梯度步骤（见第 4 讲）。这种设计理念与基本的 ML 工作流程不同，后者首先在单一地点（计算机）收集所有本地数据集，然后在这些汇集的数据上训练单一的 ML 模型。

出于多种原因，在实际数据生成地点训练不同的 ML 模型是有益的[18]：

- **隐私性**FL 方法不需要交换原始数据，只需要（更新）模型参数，因此对涉及敏感数据的应用（如医疗保健）很有吸引力[15, 16]。

由于只交换模型参数，FL 方法不会泄露（太多）本地数据集中包含的敏感信息，因此被认为是隐私友好型方法（见第 9 讲）。

- **稳健性。**依靠分散的数据和计算，FL

这些方法（在一定程度上）可抵御硬件故障（如 "散兵游勇"）和网络攻击（如第 10 讲讨论的数据中毒）。

- **并行计算。**许多 ML 系统都是由配备智能手机的人类构成的。我们可以将移动网络理解为一台并行计算机，它由可以通过无线电链路进行通信的智能手机组成。这种并行计算机可以加快计算任务的速度，例如训练 ML 模型所需的梯度计算（见第 4 讲）。

- **计算与通信的交易。**在 FL 应用中，本地数据集是由远程位置的低复杂度设备（如野生动物相机）生成的，而这些设备不易访问。将原始本地数据集传输到某个中央单元（然后由中央单元训练单一的全局 ML 模型）的成本可能远远高于使用低复杂度设备（部分）训练 ML 模型所产生的计算成本[19]。

- **个性化。**FL 可用于训练本地数据集集合的个性化 ML 模型，这些数据集可能由智能手机（及其用户）生成 [20]。确保个性化的一个关键挑战是本地数据集的异质性[21, 22]。事实上，不同本地数据集的统计属性可能会有很大差异，因此无法将它们很好地模拟

为独立且相同的分布（i.i.d.）。每个局部数据集都会引起一个单独的学习任务，其中包括为局部模型学习有用的参数值。本课程将讨论 FL

通过结合分散和异构数据中的信息来训练个性化模型的方法（见第 6 讲）。

1.3 先决条件

用于研究和设计 FL 算法的主要数学结构是欧几里得空间 $\mathbb{R}^d$。因此，我们希望对 $\mathbb{R}^d$ 的代数和几何结构有一定的了解。所谓代数结构，是指从 $\mathbb{R}^d$ 中的元素（"向量"）得到的（实）向量空间，以及 $\mathbb{R}$ 中向量加法和标量乘法的通常定义 [23, 24]。我们将大量使用线性代数的概念来表示和处理数据及 ML 模型。

$\mathbb{R}^d$ 的度量结构是 FL 算法（收敛）行为的绝佳分析工具。特别是，我们将研究作为 $\mathbb{R}^d$ 上某些非线性算子的定点迭代而获得的 FL 算法。这些算子由 FL 系统中使用的数据（分布）和 ML 模型定义。这种非线性算子的一个典型例子是基于梯度的方法的梯度步骤（见第 4 讲）。这些 FL 算法的计算特性（如收敛速度）可以通过底层算子的收缩特性来表征 [25]。

设计 FL 算法的主要工具是梯度下降（GD）的变体。这些基于梯度

的方法的共同理念是用线性函数局部逼近函数 $f(\mathbf{x})$。这个局部线性近似值由梯度 $\nabla f(\mathbf{x})$ 决定。因此，我们需要对多元微积分有一定的了解[5]。

1.4 相关课程

下面我们将简要介绍 CS-E4740 与阿尔托大学和赫尔辛基大学部分课程的关系。

- **CS-EJ3211 - 使用 Python 进行机器学习。**使用 Python 软件包（库）scikit-learn [26]，教授基本 ML 方法的应用。CS-E4740 利用正则化技术将基本 ML 方法网络耦合起来，从而获得针对本地数据集的定制（个性化）ML 模型。这种耦合需要自适应地将本地数据集汇集到足够大的个性化 ML 模型训练集中。

- **使用 Python 进行数据分析。**可替代 CS-EJ3211。

- **CS-E4510 - 分布式算法。**教授研究和设计通过分布式系统（计算机）实施的分布式算法的基本数学工具[27]。FL 通过分布式算法从分散数据中训练 ML 模型（见第 5 讲）。

- **CS-C3240 - 机器学习（2022 年春季版）。**讲授 ML 模型和方法的基本理论 [4]。CS-E4740 将数据表示和模型等基本 ML 方法的组

成部分与网络模型相结合。特别是，我们将研究由局部数据集和局部模型组成的网络，而不是单一数据集和单一模型（如决策树）。

- **ABL-E2606 - 数据保护。**讨论使用数据以及设计可信 FL 方法的重要法律限制（"法律"），包括欧洲一般数据保护条例（GDPR）。

- **MS-C2105 - 优化导论。**讲授基本优化理论以及如何将应用建模为（线性、整数和非线性）优化问题。CS-E4740使用优化理论和方法来提出FL问题（见第3讲）和设计FL方法（见第5讲）。

- **ELEC-E5424 - 凸优化。**讲授重要的凸优化问题类的高级优化理论[28]。凸优化理论和方法可用于研究和设计 FL 算法。

- **ELEC-E7120 - 无线系统。**讲授蜂窝和无线系统中使用的无线电通信基础知识。这些系统为 FL 算法的实施提供了重要的计算基础设施（见第 5 讲）。

1.5 课程的主要目标

课程的总体目标是演示如何应用图论和数学优化的概念来分析和设计 FL 算法。学生将学会把给定的 FL 应用表述为一个无向经验图 $G = (V,$

E) 上的优化问题，该图的

节点 i∈V 代表单个本地数据集。我们将此图称为本地数据集集合的经验图（见第 3 讲）。

本课程只使用无向经验图，其数量为有限的 n 的节点，我们将其与前 n 个正整数对应：

$$\mathcal{V} := \{1, \ldots, n\}.$$

经验图 G 中的边 $\{i, i'\} \in \mathcal{E}$ 连接两个不同的本地数据集，如果它们具有相似的统计属性。我们用正边权重 $A_{i,i'} > 0$ 来量化相似性的大小。

我们可以将 FL 应用正式表述为一个与经验图相关的优化问题、

$$\min_{\mathbf{w}^{(i)}} \sum_{i \in \mathcal{V}} L_i\big(\mathbf{w}^{(i)}\big) + \alpha \sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'}\, d^{(\mathbf{w}^{(i)}, \mathbf{w}^{(i')})}. \tag{1.1}$$

We refer to this problem as GTV minimization (GTVMin) and devote much of the course to its computational and statistical properties. The optimization variables $\mathbf{w}^{(i)}$ in (1.1) are local model parameters at the nodes $i \in \mathcal{V}$ of an empirical graph. The objective function in (1.1) consists of two components: The first component is a sum over all nodes of the loss values $L_i\big(\mathbf{w}^{(i)}\big)$ incurred by local model parameters at each node i. The second component is the sum of local model parameters discrepancies $d^{(\mathbf{w}^{(i)}, \mathbf{w}^{(i')})}$ across the edges $\{i, i'\}$ of the empirical graph.

1.6 课程大纲

我们的课程大致分为三个部分：

- **第一部分：ML 复习。**第 2 讲介绍了作为 ML 三个主要组成部分的数据、模型和损失函数。本讲座还解释了这些组成部分如何在经验风险最小化（ERM）中相结合。我们还将讨论如何通过操作 ERM 的三个主要组成部分来实现 ERM 的正则化。然后，我们将在讲座中解释何时以及如何通过简单的 GD 方法求解正则化 ERM 。

  4.总之，这一部分有两个主要目的：(i) 在一个简单的集中环境中简要回顾一下 ML 的基本概念；(ii) 重点介绍与 FL 方法的设计和分析特别相关的 ML 技术（如正则化）。

- **第二部分：FL 理论与方法。**第 3 讲介绍了经验图，它是我们表示本地数据集和相应定制模型集合的主要数学结构。经验图的无向边和加权边代表本地数据集之间的统计相似性。第 3 讲还将 FL 表述为正则化经验风险最小化（RERM）的一个实例，我们称之为 GTVMin。GTVMin 使用经验图中各条边上个性化模型参数的变化作为正则化。我们将看到，GTVMin 将定制（或 "个性化"）ML

模型的训练结合起来，这样，经验图中连接良好的节点（集群）将获得类似的训练模型。第 4 讲讨论了梯度下降算法的变种，它是我们解决 GTVMin 的主要算法工具箱。第 5 讲展示了如何通过将优化方法（如基于梯度的方法）应用于 GTVMin，以原则性的方式获得 FL 算法。我们将获得可以实现的 FL 算法

作为分布式训练定制（"个性化"）模型的迭代信息传递方法。第 6 讲推导了作为 GTVMin 特例的 FL 的一些主要类型。GTVMin 的实用性在很大程度上取决于经验图中加权边的选择。第 7 讲讨论了通过局部数据集之间不同的统计相似性概念来确定有用的经验图的图学习方法。

- **第三部分：值得信赖的人工智能。**第 8 讲列举了欧盟对可信人工智能（AI）提出的七项关键要求。这些关键要求包括保护隐私以及防止（故意）破坏数据或计算的鲁棒性。我们将在第 9 讲中讨论 FL 算法如何确保隐私保护。第 10 讲将讨论如何评估和确保 FL 方法对本地数据集故意扰乱（中毒）的鲁棒性。

1.7 作业和评分

课程包括编码作业，要求您在 Python 笔记本中实现讲座中的概念。我们将使用 MyCourses 小测验来测试您对讲座内容和编码作业解决方案的理解。每次测验可获得约 10 分。我们将汇总您在小测验中获得的分数（对每个小测验没有最低要求），并根据以下标准确定您的成绩：**50-59**

分为 1 级；60-69 分为 2 级；70-79 分为 3 级；80-89 分为 4 级；最高级别为

5 至少 90 分。学生可以在课程结束时查看作业评分。

1.8 学生项目

您可以通过完成学生项目和同行评审，将基本变式（5 个学分）扩展为 10 个学分。该项目要求您使用本课程中的概念，将您选择的应用程序表述为一个 FL 问题。然后，您必须使用本课程讲授的 FL 算法解决这个 FL 问题。主要成果将是一份项目报告，该报告必须遵循模板中所示的结构。然后，您将通过回答一份详细的问卷，对同学的报告进行同行评审。

1.9 时间安排

课程讲座于 2024 年 2 月 28 日至 2024 年 4 月 30 日期间的每月一和每周三下午 16:15 举行。您可以通过此链接查看详细的课程表和讲堂。由于课程可以完全远程完成，我们将录制每次讲座，并将录音添加到 YouTube 播放列表中。

每次讲座后，我们都会在本网站发布相应的作业。在我们在课程的

"我的课程"（MyCourses）页面上开放相应的测验之前，您有几天的时间来完成作业（点击我）。

作业和测验的节奏都比较快，以鼓励学生积极完成课程。我们也会严格遵守测验的截止日期。**不过，学生可以在课程结束后的复习会上弥补测验失分。**

此外，积极参与，如在讨论区发言或对讲义提供反馈意见，也会被考虑在内。

1.10 基本规则

请注意，作为选修本课程的学生，您必须遵守阿尔托大学的行为准则（请参阅此处）。本课程的两条主要基本规则是

- **诚实。**本课程包括许多需要独立完成的任务，包括编码作业、学生项目工作和学生项目的同行评审。您不得不恰当地使用他人的作品。例如，不允许复制他人的编码作业解决方案。我们将随机抽取学生解释他们的解决方案（以及测验问题的相应答案）。

- **尊重他人。**我个人的愿望是，本课程能提供一个安全的空间，让学生获得愉快的学习体验。任何形式的不尊重行为，包括任何与课程相关的交流平台，都将受到严厉制裁（包括向学校当局报告）。

1.11 鸣谢

本教材的编写得益于在阿尔托大学开设的 CS-E4740 联合学习课程中收到的学生反馈。

2023 年期间将在美国加州大学伯克利分校学习。感谢 Olga Kuznetsova

、Diana Pfau 和 Shamsiiat Abdurakhmanova 对初稿的反馈意见。

2 讲座 - "ML基础知识

本讲座涵盖了对 FL 至关重要的基本 ML 技术。与下面的讲座相比，本讲座的内容要广泛得多。不过，本讲座应该比下面的讲座更容易理解，因为它主要是复习前提知识。

2.1 学习目标

听完讲座后，您应该

- 熟悉数据点（特征和标签）、模型和损失函数的概念、
- 熟悉作为 ML 系统设计原则的机构风险管理、
- 了解验证的原因和方式、
- 能够通过比较训练误差和验证误差来诊断 ML 方法、
- 能够通过修改数据、模型和损失来规范机构风险管理。

2.2 三个组成部分和一个设计原则

机器学习（ML）围绕着从假设空间 H 中学习一个假设图 h 而展开，该

假设图可以完全根据数据点的特征准确预测数据点的标签。将 ML 方法应用于给定应用领域的最关键步骤之一是定义或选择数据点的确切含义。选择或定义一个好的数据点

点并不简单，因为它以多种不同的方式影响着 ML 方法的整体性能。

本课程将主要关注数据点的一个特定选择。特别是，我们将考虑代表 FMI 气象站周围每日天气状况的数据点。我们用

z.其特点如下

- FMI 气象站的名称，例如 "TurkuRajakari"（图尔库拉贾卡里
- 气象站的经度和纬度，例如 lat := 60.37788、

  lon := 22.0964、
- 测量的时间戳，格式为 YYYY-MM-DD HH:MM:SS，例如 2023-12-

  31 18:00:00

这样的数据点的标签 y∈R 是以摄氏度为单位的白天最高温度，例如 -20。

我们通过假设（映射）$h(\cdot)$ 的函数值 $h(\mathbf{x})$ 来预测具有 $\mathbf{x}$ 特征的数据点的标签。ML 方法使用损失函数 $L(\mathbf{z}, h)$ 来衡量预测误差。损失函数的选择对统计

and computational properties of the resulting ML method. In what follows, unless stated otherwise, we use the squared error loss $L(\mathbf{z}, h) := \left(y - h(\mathbf{x})\right)^2$

来测量预测误差。

在一组给定的数据点 D := 上选择（或学习）一个平均损失（或经验风险）最小的假设似乎很自然。

$\mathbf{x}^{(1)}, y^{(1)}, \ldots, x^{m}, y^{(m)}\}$。这就是所谓的 ERM、

$$\hat{h} \in \operatorname*{argmin}_{h \in H} (1/m) \sum_{r=1}^{m} y^{(r)} - h x^{(r)} \quad (2.1)$$

正如 (2.1) 中的符号所示（用符号"∈"代替":="），优化问题 (2.1) 可能有多个不同的解。除非另有说明，$\hat{h}$ 可以用来表示 H 中的任何假设，即对 D 的平均损耗最小。

几种重要的机器学习（ML）方法都使用参数化模型 H：每个假设 $h \in H$ 都由参数 $w \in R^d$ 定义，通常用符号 $h^{(\mathbf{w})}$ 表示。这种参数化模型的一个突出例子是线性模型 [4，第 3.1 节]、

$$H^{(d)} := h(\mathbf{x}) := \mathbf{w}^T \mathbf{x} \}。 \quad (2.2)$$

例如，线性回归方法通过最小化平均平方误差损失来学习线性模型的参数。对于线性回归，ERM 成为对参数空间 R^d 的优化、

$$\hat{w}^{(LR)} \in \operatorname*{argmin}_{w \in R^d} \underbrace{(1/m) \sum_{r=1}^{m} y^{(r)} - \mathbf{w}^T \mathbf{x}^{(r)}}_{:=f(\mathbf{w})}{}^2 \quad (2.3)$$

请注意，(2.3) 相当于找到一个平滑凸函数的最小值

$$f(\mathbf{w}) = (1/m) \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} - 2\mathbf{y}^T \mathbf{X}\mathbf{w} + \mathbf{y}^T \mathbf{y} \quad (2.4)$$

$$\text{with the feature matrix } \mathbf{X} := \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}{}^T \quad (2.5)$$

$$\text{and the label vector } \mathbf{y} := y^{(1)}, \ldots, y^{(m)}{}^T \text{ of the training set D.} \quad (2.6)$$

将 (2.4) 插入 (2.3) 可以将线性回归表述为

$$\hat{\mathbf{w}}^{(\mathrm{LR})} \in \operatorname*{argmin}_{w \in \mathbb{R}^d} \mathbf{w}^T \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q} \tag{2.7}$$

$$\mathbf{Q} := (1/m)\mathbf{X}^T \mathbf{X}, \quad \mathbf{q} := -(2/m)\mathbf{X}^T \mathbf{y}.$$

矩阵 $\mathbf{Q} \in \mathbb{R}^{d \times d}$ 具有相应的特征值分解（EVD）、

$$\mathbf{Q} = \sum_{j=1}^{d} \lambda_j \mathbf{u}^{(j)} \mathbf{u}^{(j)T}. \tag{2.8}$$

EVD (2.8) 涉及非负特征值

$$0 \leq \lambda_1 \leq \ldots \leq \lambda_d . \tag{2.9}$$

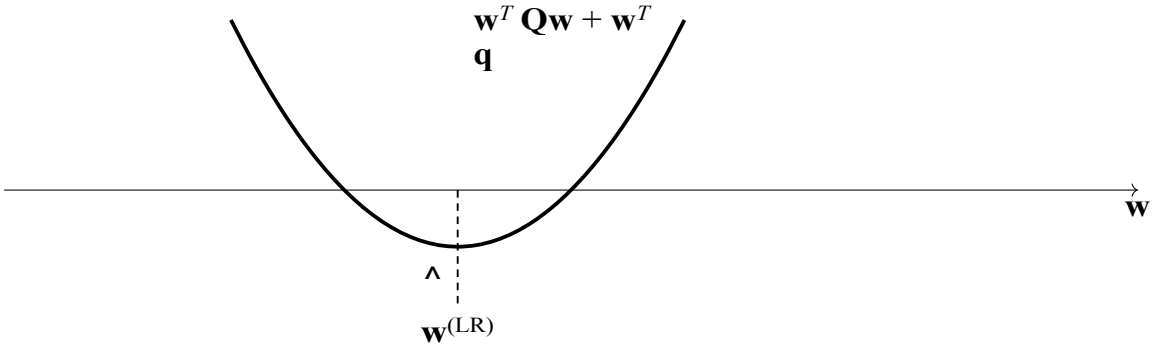


图 2.1：用于线性回归的 ERM (2.1) 将凸二次函数 $\mathbf{w}^T \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q}$ 最小化。

训练 ML 模型 H 意味着求解 ERM (2.1)（或线性回归 (2.3)）；因此数据集 D 被称为训练集。训练模型的结果就是学习假设 $\hat{h}$。我们通过应用优化算法来求解 (2.1)，从而获得实用的 ML 方法。ML 研究的两

个关键问题是

- **计算方面** 求解 (2.1) 需要多少计算量?

- **统计方面** 一般来说，（2.1）的解 $\hat{h}$ 有多大用处、

  即对**任意**标签 y 的预测 $\hat{h}(\mathbf{x})$有多准确。

  数据点的特征为 $\mathbf{x}$?

2.3 机构风险管理的计算方面

ML 方法使用优化算法求解 (2.1)，以学习假设 $\hat{h}$ 。在本课程中，我们使用的优化算法是迭代法：从初始选择 $h^{(0)}$ 开始，它们构建了一个序列

$$h^{(0)}, h^{(1)}, h^{(2)}, \ldots,$$

希望能越来越精确地逼近 (2.1) 的解 $\hat{h}$。这种 ML 方法的计算复杂度可以用保证一定近似程度所需的迭代次数来衡量。

对于参数化模型和平滑损失函数，我们可以用基于梯度的方法求解 (2.3)：从初始参数 $\mathbf{w}^{(0)}$ 开始，我们迭代梯度步骤：

$$\begin{aligned}\mathbf{w}^{(k)} &:= \mathbf{w}^{(k-1)} - \eta \nabla f(\mathbf{w}^{(k-1)}) \\ &= \mathbf{w}^{(k-1)} + (2\eta/m) \sum_{r=1}^{m} y^{(r)} \mathbf{x}^{(r)} - \mathbf{w}^{(k-1)T} \mathbf{x}^{(r)} \quad (2.10)\end{aligned}$$

(2.10) 一次迭代需要多少计算量？我们需要多少次迭代？我们将在本讲

座中尝试回答后一个问题。

4.对于典型的计算基础设施（如 "在商用笔记本电脑上运行的 Python"）来说，第一个问题更容易回答。事实上，对 (2.10) 进行一次简单的计算大约需要 m 次算术运算（加法、乘法）。

对于这种极端情况，ERM（2.3）有一个简单的闭式解：

$$\hat{w} = (1/m) \sum_{r=1} x^{(r)} \quad (2.11)$$

因此，对于线性模型的这种特殊情况，求解 (2.11) 相当于*求 m 个数的和* $x^{(1)}, \ldots, x^{(m)}$。似乎可以合理地假设，计算 (2.11) 所需的计算量与 m 成正比。

2.4 机构风险管理的统计方面

我们可以在给定的训练集上训练一个线性模型，如 ERM (2.3)。但是，(2.3) 的解 **w** 对于预测训练集以外数据点的标签有多大作用呢？考虑将学习到的假设 $h^{(\mathbf{w})}$ 应用于训练集中不包含的任意数据点。一般来说，我们能说得出的预测误差 $y - h^{(\mathbf{w})}(\mathbf{x})$有多大？换句话说，$h^{(\mathbf{w})}$ 在训练集之外的泛化效果如何?

研究 ML 方法广义化最广泛使用的方法可能是概率模型。在这里，我们将每个数据点解释为具有概率分布 $p(\mathbf{x}, y)$ 的 i.i.d. RV 的实现。在

这种情况下

的假设，我们可以评估假设的整体性能

通过预期损失（或风险）计算 h∈ H

$$E\{L((\mathbf{x}, y), h)\}. \tag{2.12}$$

概率分布 $p(\mathbf{x}, y)$ 的一个例子是将标签 y 与数据点的特征 $\mathbf{x}$ 联系起来，如下所示

$$y = \mathbf{w}^T \mathbf{x} + \varepsilon，\ \mathbf{x} \sim \mathsf{N}(\mathbf{0}, \mathbf{I})，\ \varepsilon \sim \mathsf{N}(0, \sigma^2)，\ \mathsf{E}\{\varepsilon \mathbf{x}\} = \mathbf{0}。 \tag{2.13}$$

通过简单计算，可以得出给定线性假设 $h(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$ 的预期平方误差损失为[1]

$$\mathsf{E}\{(y - h(\mathbf{x}))^2\} = \|\mathbf{w} - \hat{\mathbf{w}}\|^2 + \sigma^2. \tag{2.14}$$

分量 σ^2 可以解释为标签 y 的内在噪音水平。我们不希望找到一个预期损失小于这个水平的假设。

(2.14) 中 RHS 的第一个分量是估计误差 $\|\mathbf{w} - \hat{\mathbf{w}}\|^2$，估计值 $\hat{\mathbf{w}}$ 的 ML 方法，该方法读取训练集，并给出 $\hat{\mathbf{w}}$（例如，通过 (2.3)）获得线性假设的参数。

接下来，我们将研究线性回归方法提供的具体估计值 $\hat{\mathbf{w}} = \mathbf{w}^{(LR)}$ (2.7) 所产生的估计误差 $\mathbf{w} - \hat{\mathbf{w}}$。为此，我们首先使用概率模型 (2.13) 将 (2.6) 中的标签向量 $\mathbf{y}$ 分解为

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{n} \text{ , with } \mathbf{n} := \left(\varepsilon^{(1)}, \ldots, \varepsilon^{(m)}\right)^T. \tag{2.15}$$

[1]严格来说，关系式 (2.14) 只适用于不依赖于 RV 的常数（确定性）参数 **w**，而 RV 的实现是观测数据点（见，例如，(2.13)）。然而，参数 **w** 可能是应用于由 i.i.d. RV 变现组成的数据集 D 的 ML 方法（如 (2.3)）的输出结果。在这种情况下，我们需要用条件期望 $\mathrm{E}\{y - h(\mathbf{x})\}$ 代替 (2.14) LHS 上的期望。[2]D 。

将 (2.15) 插入 (2.7) 即可得出

$$\hat{\mathbf{w}}^{(\mathrm{LR})} \in \underset{w \in Rd}{\operatorname{argmin}} \mathbf{w}^T \quad \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q}' + \mathbf{w}^T \mathbf{e} \tag{2.16}$$

$$\mathbf{Q} := (1/m)\mathbf{X}^T \mathbf{X}，\ \mathbf{q}' := -(2/m)\mathbf{X}^T \mathbf{X}\bar{\mathbf{w}}，\ \mathbf{e} := -(2/m)\mathbf{X}^T \mathbf{n}。 \tag{2.17}$$

图 2.2 描述了 (2.16) 的目标函数。它是凸二次函数 $\mathbf{w}^T \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q}'$ 的扰动，在 $\mathbf{w} = \bar{\mathbf{w}}$ 时最小化。一般来说，由于（2.16）中的扰动项 $\mathbf{w}^T \mathbf{e}$，线性回归得到的最小化 $\mathbf{w}^{(\mathrm{LR})}$ 与 $\bar{\mathbf{w}}$ 不同。

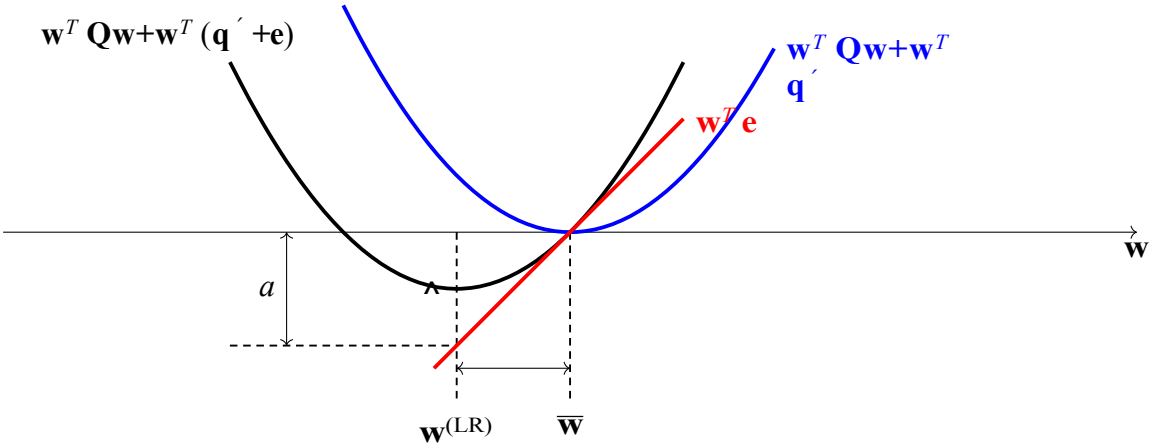


图 2.2：线性回归的估计误差取决于扰动项 $\mathbf{w}^T \mathbf{e}$ 对凸二次函数 $\mathbf{w}^T \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q}'$ 最小化的影响。

在矩阵 $\mathbf{Q} = (1/m)\mathbf{X}^T \mathbf{X}$ 可逆的假设条件下，下面的结果确定了

$\mathbf{w}^{(\mathrm{LR})}$ 与 $\mathbf{w}$ 之间的偏差。[2]

[2]你能想到训练集特征矩阵的充分条件，确保 $\mathbf{Q} = (1/m)\mathbf{X}^T\mathbf{X}$ 是可逆的吗？

命题 2.1. *考虑应用于数据集* (2.15) *的 ERM 实例* (2.16) *的解* $\mathbf{w}^{(\mathrm{LR})}$ *。*

如果矩阵 $\mathbf{Q} = (1/m)\mathbf{X}^T\mathbf{X}$ *是可逆的，且最小特征值* $\lambda_1(\mathbf{Q}) > 0$*、*

$$\|\mathbf{w}^{(\mathrm{LR})} - \mathbf{w}\|_2^2 \leq \|\mathbf{e}\|_2^2/\lambda_1^2 \overset{(2.17)}{=} (4/m^2)\|\mathbf{X}^T\mathbf{n}\|_2^2/\lambda_1^2. \tag{2.18}$$

证明。 让我们把（2.16）重写为

$$\mathbf{w}^{(\mathrm{LR})} \in \operatorname*{argmin}_{w \in \mathbb{R}^d} f(\mathbf{w}) \text{ with } f(\mathbf{w}) := (\mathbf{w}-\bar{\mathbf{w}})^T\mathbf{Q}(\mathbf{w}-\bar{\mathbf{w}}) + \mathbf{e}^T(\mathbf{w}-\bar{\mathbf{w}}). \tag{2.19}$$

显然，$f(\bar{\mathbf{w}}) = 0$，反过来，$f(\mathbf{w}) \geq \min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) \leq 0$*、*

$$\begin{aligned} f(\mathbf{w}) &\overset{(2.19)}{=} (\mathbf{w}-\bar{\mathbf{w}})^T\mathbf{Q}(\mathbf{w}-\bar{\mathbf{w}}) + \mathbf{e}^T(\mathbf{w}-\bar{\mathbf{w}}) \\ &\overset{(a)}{\geq} (\mathbf{w}-\bar{\mathbf{w}})^T\mathbf{Q}(\mathbf{w}-\bar{\mathbf{w}}) - \|\mathbf{e}\|_2\|\mathbf{w}-\bar{\mathbf{w}}\|_2 \\ &\overset{(b)}{\geq} \lambda_1\|\mathbf{w}-\mathbf{w}_2\|_2^2 - \|\mathbf{e}\|_2\|\mathbf{w}-\bar{\mathbf{w}}\|_2. \end{aligned} \tag{2.20}$$

步骤 (*a*) 使用了考奇-施瓦茨不等式，步骤 (*b*) 使用了 $\mathbf{Q}$ 的 EVD (2.8)。对 $\mathbf{w} = \mathbf{w}$ 求值 (2.20) 并与 $f(\hat{\mathbf{w}}) \leq 0$ 结合，得到 (2.18)。 □

界值 (2.18) 表明，如果 $\lambda_1(\mathbf{Q})$ 较大，则估计误差 $\mathbf{w}^{(\mathrm{LR})} - \bar{\mathbf{w}}$ 较小。矩阵 $\mathbf{Q} = (1/m)\mathbf{X}^T\mathbf{X}$ 的最小特征值可以通过适当选择（或变换）数据点的特征 $\mathbf{x}$ 来控制。简单来说，如果我们将每个特征放大 10 倍，就可以将 $\lambda_1(\mathbf{Q})$ 放大 10^2 倍。然而，这种方法也会将 (2.18) 中的误差项 $\|\mathbf{X}^T\mathbf{n}\|_2$ 。在某些应用中，我们可以发现特征转化（"增白"），从而增加 $\lambda_1(\mathbf{Q})$，但不会增加

¨**X** ṅ¨ $\frac{2}{2}$ 最后我们注意到，如果出现以下情况，$\frac{2}{2}$（2.18）中的误差项 ¨**X** ṅ¨ 将消失

噪声矢量 **n** 与特征矩阵 **X** 的列正交。

在这种情况下，概率模型 (2.15) 将简化为 "噪声中的信号 "模型

、

$$y^{(r)} = x^{(r)}\,\overline{w} + \varepsilon^{(r)} \text{ ，其中 } x^{(i)} = 1\text{、} \tag{2.21}$$

噪声项 $\varepsilon^{(r)}$, 对于 $r = 1, \ldots$ 是概率分布为 $\mathcal{N}(0, \sigma^2)$ 的 i.i.d. RV 的实现。

特征矩阵变为 $\mathbf{X} = \mathbf{1}$，反过来，$\mathbf{Q} = 1$，$\lambda_1(\mathbf{Q}) = 1$ 。

将这些值插入 (2.18)，得出约束条件

$$\left(\hat{w}^{(LR)} - \overline{w}\right)^2 \leq 4\|h\|^2 / m \cdot \sigma^2 \tag{2.22}$$

请注意，对于 (2.21) 中的标签和特征，(2.16) 的解由以下公式给出

$$\hat{w}^{(LR)} = (1/m)\sum_{r=1}^{m} y^{(r)} \overset{(2.21)}{=} \overline{w} + (1/m)\sum_{r=1}^{m} \varepsilon^{(r)} \tag{2.23}$$

2.5 ML 的验证和诊断

上述对广义误差的分析是从假设一个生成数据点的概率模型开始的。然而，这种概率模型可能是错误的，因此约束 (2.18) 并不适用。因此，我们可能希望使用更多的数据驱动方法来评估学习假设 $\hat{h}$ 的有用性，例如通过求解 ERM (2.1) 得到的假设 $\hat{h}$。

从广义上讲，验证方法试图找出 $\hat{h}$ 在训练集内外的表现是否相似。从最基本的形式来看，验证相当于在训练集之外的一些数据点上计算所学假设 $\hat{h}$ 的平均损失。我们将这些数据点称为验证集。

算法 2.1 总结了 ML 模型训练和验证的原型工作流程。该工作流程从选择数据集 D、模型 H 和损失函数 L (-, -) 开始。我们通常会多次执行算法 2.1，每次都会选择不同的数据集、模型和损失函数。这些设计选择，包括何时停止重新运行算法 2.1 的决定，可以基于一些基本诊断[4，第 6 章]。

算法 2.1 ML 训练和验证的一次迭代

输入： 数据集 D、模型 H、损失函数 L (-, -)

1: 将 D 分成训练集 $D^{(train)}$ 和验证集 $D^{(val)}$

2: 通过解决机构风险管理问题来学习假设

$$\hat{h} \in \text{最小值}_{h \in H} \sum_{(\mathbf{x},y) \in D^{(火车)}} L((\mathbf{x}, y), h) \tag{2.24}$$

3: 计算得出的训练误差

$$E_t := (1/|D^{(train)}|) \sum_{(\mathbf{x},y) \in D^{(火车)}} L(\mathbf{x}, y), \hat{h}$$

4: 计算验证误差

$$e_v := (1/|d^{(val)}|) \sum_{(\mathbf{x},y) \in D^{(val)}} L(\mathbf{x}, y), \hat{h}$$

输出： 学习假设（或训练模型）h、训练误差 E_t 和估值误差 E_v

我们可以通过比较训练误差和验证误差来诊断基于 ERM 的 ML 方法，如算法 2.1。如果我们知道基线 $E^{(\mathrm{ref})}$，就能进一步进行诊断。

$E^{(\mathrm{ref})}$ 是数据点的概率模型（见第 2.4 节）。

给定一个概率模型 $p(\mathbf{x},y)$，我们就可以计算出最小可实现风险 (2.12) 。事实上，最小可实现风险正是在给 定 数据点特征 $\mathbf{x}$ 的情况下，标签 y 的贝叶斯估计值 $h(\mathbf{x})$ 的预期损失。贝叶斯估计值 $h(\mathbf{x})$ 完全由概率分布 $p(\mathbf{x}, y)$ 决定[29，第 4 章]。

基线 $E^{(\mathrm{ref})}$ 的另一个潜在来源是现有的、但由于某种原因不适合的 ML 方法。这种现有的 ML 方法在计算上可能过于昂贵，无法用于当前的 ML 应用。不过，我们仍可将其统计特性作为基准。

我们还可以将人类专家的表现作为基准。如果我们想开发一种能从皮肤图像中检测出某种类型皮肤癌的 ML 方法，那么经验丰富的皮肤科医生[30]所达到的分类准确率可能就是一个基准。

我们可以通过比较训练误差 E_t 和验证误差 E_v 以及（如有）基准误差 $E^{(\mathrm{ref})}$ 来诊断 ML 方法。

- $E_t \approx E_v \approx E^{(\mathrm{ref})}$：训练误差与验证误差和基线处于同一水平。由于验证误差已经接近基线，因此可能没有太大的改进余地。此外，训练误差并不比验证误差小多少，这表明没有过度拟合。

- $E_v \gg E_t$：验证误差明显大于训练误差。这是过度拟合的指标。

  我们可以通过降低假设空间的有效维度或增加训练集的大小来解决过度拟合问题。我们可以降低假设空间的有效维度

通过使用较少的特征（在线性模型中）、较小的决策树最大深度或在 ANN 中使用较少的层数来减少假设空间。除了这种粗粒度的离散模型选择，我们还可以通过正则化（参见 [4, Ch. 7]）以更平滑的方式降低假设空间的有效维度。

- $E_t \approx E_v \gg E^{(\text{ref})}$：训练误差与验证误差处于同一水平，都明显大于基线误差。因此，学习到的假设似乎没有过度拟合训练集。然而，学习假设的训练误差却明显大于基线误差。出现这种情况可能有几个原因。首先，可能是假设空间太小，即没有一个假设能很好地近似数据点的特征和标签之间的关系。解决这种情况的办法之一是使用更大的假设空间，例如，在线性模型中包含更多特征，在多项式回归中使用更高的多项式度，使用更深的决策树或 ANN（深度 ANN（深度网））。其次，除了模型太小之外，训练误差过大的另一个原因可能是用于求解 ERM (2.24) 的优化算法没有正常工作（见第 4 讲）。

- $E_t \gg E_v$：训练误差明显大于验证误差。ERM（2.24）的思想

是近似地计算一个

假设在训练集 $D = \{(\mathbf{x}^{(r)}, y^{(r)})\}_{r=1}^{m}$ 上的平均损失。

这种近似的数学基础是大数定律，它是 i.i.d. 的平均值（实现值）的特征。

RVs.这种近似方法的准确性取决于两个条件的有效性：首先，用于计算平均损失的数据点 "应表现 "为具有共同概率分布的 i.i.d. RV 的实现。其次，用于计算平均损失的数据点数量必须足够多。

每当训练集或验证集与 i.i.d. RV 的实现有显著差异时，对所学假设的训练误差和验证误差的解释（和比较）就会变得更加困难。在极端情况下，验证集可能由数据点组成，其中每个假设都会产生较小的平均损失（见图 2.3）。在这种情况下，我们可以通过收集更多标注数据点或使用数据扩增（见第 2.6 节）来增加验证集的大小。如果训练集和验证集的规模都很大，但我们仍然得到 $E_t \gg E_v$，那么我们就应该验证这些集中的数据点是否符合 i.i.d. 假设。对于给定数据集，有原则性的统计检验 i.i.d. 假设的有效性（见 [31] 及其中的参考文献）。

2.6 规范化

考虑使用假设空间 H 和数据集 D（我们假设所有数据点都用于训练）的基于 ERM 的 ML 方法。这种 ML 方法的一个关键参数是模型大小

d_{eff} (H) 与数据点数量 $|D|$ 之间的比率 d_{eff} (H) /$|D|$。随着比率 d_{eff} (H) /|D| 的增大，ML 方法的过拟合趋势也会增大。

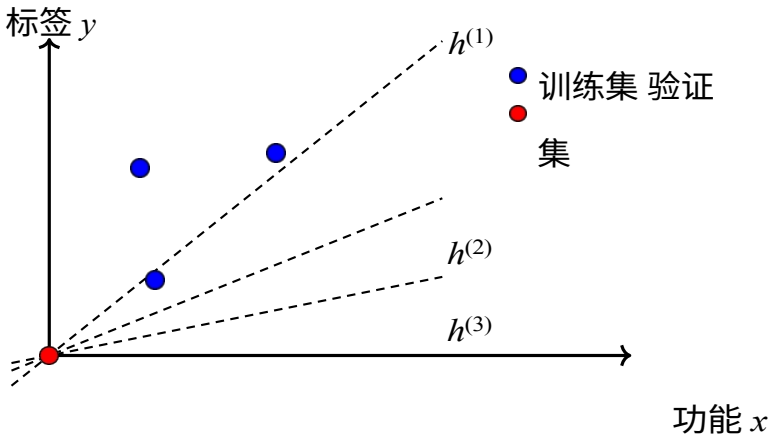


图 2.3：将模型 $\mathcal{H} := \{h^{(1)}, h^{(2)}, h^{(3)}\}$ 的训练集和验证集进行不走运分割的示例。

正则化技术通过三种（本质上等同的）方法降低比率 $d_{\text{eff}}(\mathcal{H})/|D|$：

- 收集更多的数据点，可能通过数据扩增（见图 2.4）、
- 在 ERM (2.1) 的平均损失中加入惩罚项 $\alpha R(h)$（见图 2.4）、
- 缩小假设空间，例如对模型参数添加限制条件，如 $\|\mathbf{w}\|_2 \leq 10$。

可以证明，正则化的这三个角度（对应于数据、模型和损失三个部分）是密切相关的[4，第 7 章]。例如，在 ERM (2.1) 中添加惩罚项 $\alpha R(h)$ 相当于 ERM (2.1) 中的剪枝假设空间 $\mathcal{H}^{(\alpha)} \subseteq \mathcal{H}$。使用较大的 α 通常会导

致较小的 H 。$^{(\alpha)}$

在平均损失中加入惩罚项进行正则化的一个重要例子是脊回归。特别是，脊回归使用

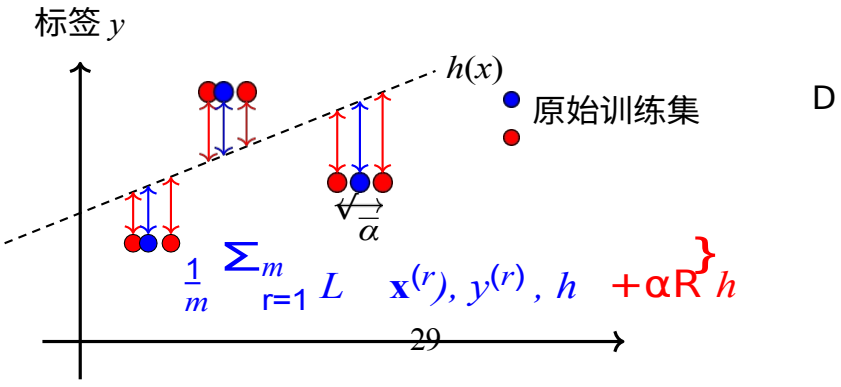
正则 R h } 对于线性假设 $h(\mathbf{x}) := \mathbf{w}^T \mathbf{x}$，R $h := \| \mathbf{w} \|^2$。因此，脊回归通过求解线性假设的参数来学习线性假设。

$$\mathbf{w}^{(\alpha)} \in \underset{w \in Rd}{\text{最小值}} \quad (1/m) \sum_{r=1} y^{(r)} - \mathbf{w}\, \mathbf{x}^{T\ (r)\ 2} + \alpha\, w\, .\|_2^2 \qquad (2.25)$$

(2.25) 中的目标函数可以解释为线性回归的目标函数，应用于对训练集 D 的修改：我们将每个数据点 $(\mathbf{x}, y) \in$ D 替换为足够多的 i.i.d.

$$(\mathbf{x} + \mathbf{n}, y) \text{ ， } \mathbf{n} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\alpha}\mathbf{I}) \text{ 。} \qquad (2.26)$$

因此，脊回归 (2.25) 等同于应用于 D 的增强变体 D´ 的线性回归。增强变体 D´ 是用足够多的噪声副本替换每个数据点 $(\mathbf{x}, y) \in$ D 而得到的。$(\mathbf{x}, y)$ 的每个副本都是通过在特征 $\mathbf{x}$ 上添加协方差矩阵为 αI 的零均值高斯噪声的 i.i.d. 实现 $\mathbf{n}$ 而得到的（见 (2.26)）。$(\mathbf{x}, y)$ 的每个副本的标签都等于 y，即标签未受扰动。



功能 x

图 2.4：数据扩充与损失惩罚之间的等价关系。

为了研究脊回归的计算问题，我们将 (2.25) 重写为

$$\hat{\mathbf{w}}^{(\alpha)} \in \underset{w \in Rd}{\operatorname{argmin}} \mathbf{w}^T \mathbf{Q}\mathbf{w} + \mathbf{w}^T \mathbf{q}$$

$$\mathbf{Q} := (1/m)\mathbf{X}^T \mathbf{X} + \alpha \mathbf{I},\ \mathbf{q} := (-2/m)\mathbf{X}^T \mathbf{y}。 \quad (2.27)$$

因此，与线性回归 (2.7) 一样，脊回归也是最小化凸二次函数。线性回归 (2.7) 与脊回归（$\alpha > 0$ 时）的主要区别在于，对于任何训练集 D，(2.27) 中的矩阵 $\mathbf{Q}$ 都保证是可逆的。[3]

(2.27) 求解（即通过脊回归求得的参数）的统计意义主要取决于 α 的值。这一选择可以通过使用数据概率模型进行误差分析来指导（见命题 2.1）。除了使用概率模型，我们还可以比较假设 $h(\mathbf{x}) = \hat{\mathbf{w}}^{(\alpha)T} \mathbf{x}$ 的训练误差和验证误差通过不同的 α 值进行脊回归学习。

2.7 编码作业概述

Python 笔记本。 MLBasics_CodingAssignment.ipynb

数据文件。 作业_MLBasicsData.csv

说明。 编码任务围绕芬兰气象局收集并存储在 csv 文件中的气象数据展

开。该文件包含芬兰不同地点的温度测量数据。

[3]考虑一种极端情况，即训练集中每个数据点的所有特征 D 为零。

每个温度测量值都是一个数据点，其特征为 $d = 7$

特征 $\mathbf{x} = x_1, \ldots, x_7^T$，标签 y 是温度测量值

本身。特征是 FMI 站点的经纬度（标准化）值，以及测量的年、月、

日、时、分（标准化）。您的任务包括

- 生成 numpy 数组 **X**、**y**，其中第 r 行分别保存 csv 文件中*第 r 个*数据点的特征 $\mathbf{x}^{(r)}$ 和标签 $y^{(r)}$。

- 将数据集分成训练集和验证集。训练集的大小应为 100。

- 使用 scikit-learn 软件包中的 LinearRegression 类在训练集上训练线性模型，并确定由此产生的训练误差

  和验证错误

- 通过多项式组合使用特征增强（见 PolynomialFeatures 类）来训练和验证线性模型。尝试选择这些多项式组合的最大度数。

- 在特征增强步骤中使用固定的多项式度值，通过脊类使用脊回归 (2.25) 训练和验证线性模型。

- **注意！** Ridge 类的输入参数 alpha 与 (2.25) 中的 α 意义不同，尤

其是 (2.25) 中的参数 α 与 alpha 的值 $m\alpha$ 相对应。