Introductory Lectures on Federated Learning

Dipl.-Ing. Dr.techn. Alexander Jung*

December 27, 2023

Abstract

This course discusses theory and algorithms for federated learning (FL) from decentralized data with an intrinsic network structure, i.e., networked data. Such networked data consists of local datasets that are related by different notions of similarities. FL methods collaboratively train local (" personalized") models for each local dataset. These tailored local models are coupled via a network structure that reflects statistical similarities between local datasets. We use the network structure of data to guide this coupling via regularization techniques. In particular, we use different measures for the variation of local models as a regularization term. The resulting generalized total variation minimization provides a unifying design principle for practical FL algorithms. These algorithms solve generalized total variation via distributed optimization methods that are able to cope with limited computational resources and imperfections.

*AJ is currently Associate Professor for Machine Learning at Aalto University (Finland).
This work has been partially funded by the Academy of Finland (decision numbers 331197, 331197) and the European Union (grant number 952410).

Contents

1 Lecture - "Welcome and Intro"

This lecture offers

- introduction of course topic and positioning in wider curricula
- overview of course schedule
- discussion of exercises, notebooks, quizzes
- discussion of student project
- discussion of student feedback and changes to last year

1.1 Introduction

Many important application domains generate decentralized collections of local datasets that are related via an intrinsic network structure [1]. Two timely application domains that generate such networked data are (i) the high-precision management of pandemics and (ii) the Internet of Things (IoT) [2, 3]. Here, local datasets are generated by smartphones, wearables or other IoT devices [4, 5]. These local datasets are related via physical contact networks, social networks [6], co-morbidity networks [7], or communication networks [8].

FL is an umbrella term for distributed optimization techniques to train machine learning (ML) models from decentralized collections of local datasets [9–13]. These methods carry out computations, such as gradient steps (see Lecture 4), for ML model training at the location of data generation. This design philosophy is different from a naive application of ML techniques which

is to first collect all local datasets at a single location (computer). This pooled data is then fed into a conventional ML method such as linear regression.

The distributed training of ML models, at locations close to the actual data generation, can be beneficial in several aspects [14]:

- **Privacy.** FL methods are appealing for applications involving sensitive data (such as healthcare) as they do not require the exchange of raw data but only model (parameter) updates [11, 12]. By exchanging only model updates, FL methods are considered privacy-friendly in the sense of not leaking (too much) sensitive information that is contained in the local datasets (see Lecture 9).

- **Robustness.** By relying on decentralized data and computation, FL methods offer robustness against hardware failures (such as "stragglers") and data poisonings (see Lecture 10).

- **Parallel Computing.** A main application domain for FL are mobile networks, consisting of humans equipped with smartphones. We can interpret a mobile network as a parallel computer which is constituted by smartphones that can communicate via radio links. This parallel hardware allows to speed up computational tasks such as the computation of gradients required to train ML models (see Lecture 4).

- **Trading Computation against Communication.** Consider a FL application where local datasets are generated by low-complexity devices at remote locations that cannot be easily accessed. The cost of communicating raw local datasets to some central unit (which then trains a single global ML model) might be much higher than the computational

cost incurred by using the low-complexity devices to (partially) train ML models [15].

- **Personalization.** FL can be used to train personalized ML models for collections of local datasets, which might be generated by smartphones (and their users) [16]. A key challenge for ensuring personalization is the heterogeneity of local datasets [17, 18]. Indeed, the statistical properties of different local datasets might vary significantly such they cannot be well modelled as independent and identically distributed (i.i.d.). Each local dataset induces a separate learning task that consists of learning useful parameter values for a local model. This course discusses FL methods to train personalized models via combining the information carried in decentralized and heterogeneous data (see Lecture 6).

1.2 Prerequisites

The main mathematical structure used to study and design FL algorithms is the Euclidean space $\mathbb{R}^d$. We therefore expect some famililartiy with the algebraic and geometric structure of $\mathbb{R}^d$. By algebraic structure of $\mathbb{R}^d$, we mean the (real) vector space obtained from the elements ("vectors") in $\mathbb{R}^d$ along with the usual definitions of vector addition and mulitplication by scalars in $\mathbb{R}$ [19, 20]. We will make heavy use of concepts from linear algebra to represent and maninpulate data and ML models.

The metric structure of $\mathbb{R}^d$ will be used to study the (convergence) behaviour of FL algorithms. In particular, we will study FL algorithms that are obtained as fixed-point iterations of some non-linear operator on $\mathbb{R}^d$

which depends on the data (distribution) and ML models used within a FL system. The computational properties (such as convergence speed) of these FL algorithms can then be characterized via the contraction properties of the underlying operator [21].

A main tool for the design the FL algorithms are varitions of gradient descent (GD). These gradient-based methods are based on approximating a differentiable function $f(\mathbf{x})$ locally by a linear function given by the gradient $\nabla f(\mathbf{x})$. We therefore expect some familiarity with multivariable calculus [22].

1.3 Related Courses

In what follows we briefly explain how this course CS-E4740 relates to selected courses at Aalto University.

- **CS-EJ3211 - Machine Learning with Python.** Teaches the application of basic ML methods using the Python package (library) `scikit-learn` [23]. CS-E4740 couples a network of basic ML methods using regularization techniques to obtain tailored (personalized) ML models for local datasets. This coupling is required to adaptive pool local datasets obtain sufficiently large training sets for the personalized ML models.

- **CS-E4510 - Distributed Algorithms.** Teaches basic mathematical tools for the study and design of distributed algorithms that are implemented via distributed systems [24]. FL is enabled by distributed algorithms to train ML models from decentralized data (see Lecture 5).

- **CS-C3240 - Machine Learning (spring 2022 edition).** Teaches basic theory of ML models and methods [25]. CS-E4740 combines the components of basic ML methods, such as data representation and models, with network models. In particular, instead of a single dataset and a single model (such as a decision tree), we will study networks of local datasets and local models.

- **ABL-E2606 - Data Protection.** This course discusses important legal constraints ("laws"), including the European general data protection regulation (GDPR), for the use of data and, in turn, for the design of trustworthy FL methods.

- **MS-C2105 - Introduction to Optimization.** This course teaches basic optimisation theory and how to model applications as (linear, integer, and non-linear) optimization problems. CS-E4740 uses optimization theory and methods to formulate FL problems (see Lecture 3) and design FL methods (see Lecture 5).

- **ELEC-E5424 - Convex Optimization.** This course teaches advanced optimisation theory for the important class of convex optimization problems [26]. Convex optimization theory and methods can be used for the study and design of FL algorithms.

1.4 Main Goal and Learning Outcomes

The main goal of the course is to teach students a mathematical toolbox for the analysis and design of FL algorithms. This toolbox revolves around the formulation of a given FL application as an optimization problem over an

undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes $i \in \mathcal{V}$ represent individual local datasets. We refer to this refer as the empirical graph of a collection of local datasets (see Lecture 3). This course uses only empirical graphs with a finite number n of nodes, which we we identify with the first n positve integers:

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

An edge $\{i, i'\} \in \mathcal{E}$ in the empirical graph $\mathcal{G}$ connects two different local datasets if they have similar statistical properties. The amount of similarity is quantified by a positive edge weight $A_{i,i'} > 0$.

FL applications can be formalized as instances of the generic optimization problem

$$\min_{\mathbf{w}^{(i)}} \sum_{i \in \mathcal{V}} L_i \left(\mathbf{w}^{(i)} \right) + \lambda \sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} d \left(\mathbf{w}^{(i)}, \mathbf{w}^{(i')} \right). \tag{1}$$

We refer to this problem as GTV minimization (GTVMin) and devote much of the course to the discussion of its computational and statistical properties. The optimization variables $\mathbf{w}^{(i)}$ in (1) are local model parameters at the nodes $i \in \mathcal{V}$ of an empirical graph. The objective function in (1) consists of two components: The first component is a sum over all nodes of the loss values $L_i \left(\mathbf{w}^{(i)} \right)$ incurred by local model parameters at each node i. The second component is the sum of local model parameters varitions across the edges $\{i, i'\}$ of the empirical graph.

1.5 Outline of the Course

Our course is roughly divided into three parts:

- **Part I: ML Refresher.** Lecture 2 introduces data, models and loss functions as three main components of ML. This lecture also explains

how these components are combined within empirical risk minimization (ERM). We also discuss how regularization of ERM can be achieved via manipulating its three main components. We then explain when and how to solve regularized ERM via simple GD methods in Lecture 4. Overall, this part serves two main purposes: (i) to briefly recap basic concepts of ML in a simple centralized setting and (ii) highlight ML techniques (such as regularization) that are particulary relevant for the design and analysis of FL methods.

- **Part II: FL Theory and Methods.** Lecture 3 introduces the empirical graph as our main mathematical structure for representing collections of local datasets and corresponding tailored models. The undirected and weighted edges of the empirical graph represent statistical similarities between local datasets. Lecture 3 also formulates FL as an instance of regularized empirical risk minimization (RERM) which we refer to as GTVMin. GTVMin uses the variation of personalized model parameters across edges in the empirical graph as regularizer. We will see that GTVMin couples the training of tailored (or "personalized") ML models such that well-connected nodes (clusters) in the empirical graph will obtain similar trained models. Lecture 4 dsicusses variations of gradient descent as our main algorithmic toolbox for solving GTVMin. Lecture 5 shows how FL algorithms can be obtained in a principled fashion by applying optimization methods, such as gradient descent, to GTVMin. We will obtain FL algorithms that can be implemented as iterative message passing methods for the distributed training of tailored ("personalized") models. Lecture 6 derives some main flavours

of FL as special cases of GTVMin. The usefulness of GTVMin crucially depends on the choice for the weighted edges in the empirical graph. Lecture 7 discusses graph learning methods that determine a useful empirical graph via different notions of statistical similarity between local datasets.

- **Part III: Trustworthy AI.** Lecture 8 enumerates seven key requirements for trustworthy artificial intelligence (AI) that have been put forward by the European Union. These key requirements include the protection of privacy as well as robustness against (intential) perturbations of data or computationa. We then discuss how FL algorithms can ensure privacy protection in Lecture 9. Lecture 10 discusses how to evaluate and ensure robustnes of FL methods against intentional perturbations (poisoning) of local datasets.

2 Lecture - "ML Basics"

After this lecture, you should

- be familiar with the concept of data points (their features and labels), model and loss function 1
- be familiar with ERM as a design principle for ML systems
- know why and how validation is performed
- know three different ways to regularize a ML method

2.1 Three Components and a Design Principle

Machine Learning (ML) revolves around learning a hypothesis map h out of a hypothesis space $\mathcal{H}$ that allows to accurately predict the label of a data point solely from its features. One of the most crucial steps in applying ML methods to a given application domain is the definition or choice of what precisely a data point is. Coming up with a good choice or definition of data points is not trivial as it influences the overall performance of a ML method in many different ways.

During this course we will focus mainly on one specific choice for the data points. In particular, we will consider data points that represent the daily weather condition around a weather station of the Finnish Meteorological Institute (FMI). We denote a specific data point by $\mathbf{z}$. It is characterized by the following features:

- name of the FMI weather station, e.g., "TurkuRajakari"

- latitude lat and longitude lon of the weather station, e.g., $\mathsf{lat} := 60.37788$, $\mathsf{lon} := 22.0964$,
- date of the day in format DDMMYYYY, e.g., 01022022
- minimum daytime temperature $x \in \mathbb{R}$.

It is convenient to stack the features into a feature vector $\mathbf{x}$. The label $y \in \mathbb{R}$ of such a data point is the maximum daytime temperature.

We predict the label by the function value hypothesis $h(\mathbf{x})$. The prediction will typically be not perfect, i.e., $h(\mathbf{x}) \neq y$. We measure the prediction error by a loss function such as the squared error loss $L\left(\mathbf{z}, h\right) := \big(y - h(\mathbf{x})\big)^2$. It seems natural to choose (or learn) a hypothesis that incurs minimum average loss (or empirical risk) on a given set of data points $\mathcal{D} := \big\{\big(\mathbf{x}^{(1)}, y^{(1)}\big), \ldots, \big(\mathbf{x}^{m}, y^{(m)}\big)\big\}$. This is known as ERM,

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} (1/m) \sum_{r=1}^{m} \big(y^{(r)} - h\big(\mathbf{x}^{(r)}\big)\big)^2 \tag{2}$$

As our notation indicates (using the symbol "$\in$" instead of ":="), there might be several different solutions to the optimization problem (2). Unless specified otherwise, $\hat{h}$ can be used to denote any hypothesis in $\mathcal{H}$ that has minimum average loss over $\mathcal{D}$.

A large class of ML methods use a paramterized model $\mathcal{H}$ with each hypothesis $h \in \mathcal{H}$ specified by a parameter vector $\mathbf{w} \in \mathbb{R}^d$. The prime example for a parametrized model is the linear model: $h(\mathbf{x}) := \mathbf{w}^T \mathbf{x}$ [25, Sec. 3.1]. For the linear model, the ERM is equivalent to an optimization over the

parameter space $\mathbb{R}^d$,

$$\widehat{\mathbf{w}} \in \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} \underbrace{(1/m)\sum_{r=1}^{m}\left(y^{(r)} - \mathbf{w}^T\mathbf{x}^{(r)}\right)^2}_{:=f(\mathbf{w})}. \tag{3}$$

Note that (3) amounts to finding the minimum of a smooth and convex function

$$f(\mathbf{w}) = (1/m)\Big[\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y}\Big]. \tag{4}$$

Here, we used the feature matrix $\mathbf{X} := \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^T$ and the label vector $\mathbf{y} := \left(y^{(1)}, \ldots, y^{(m)}\right)^T$ of the training set $\mathcal{D}$.

To train a ML model $\mathcal{H}$ means to solve ERM (2) (or (3) for parametrized models); the dataset $\mathcal{D}$ is therefore referred to as a training set. The trained model results in the learnt hypothesis $\hat{h}$. We obtain practical ML methods by applying optimiziation algorithms to solve (2). If we use this approach to obtain practical ML methods, we should carefully think about two key questions:

- **computational aspects** How much computation is need to solve (2) ?
- **statistical aspects** How useful is the solution $\hat{h}$ to (2) in practice, i.e., how accurate is the prediction $\hat{h}(\mathbf{x})$ for the label y of an **arbitrary** data point with features $\mathbf{x}$?

2.2 Computational Aspects of ERM

ML methods use optimization algorithms to solve (2) in order to learn a hypothesis $\hat{h}$. Within this course, we use optimization algorithms that are

iterative methods: Starting from an initial choice $h^{(0)}$, they construct a sequence

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

which are hopefully increasingly accurate approximations to a solution $\hat{h}$ of (2). The computational complexity of such a ML method can be measured by the number of iterations required to guarantee some prescribed level of approximation.

When using a parameterized model and a smooth loss function, we can solve (3) by (variants of) gradient step descen: Starting from some initial parameters $\mathbf{w}^{(0)}$, we iterate the gradient step:

$$\begin{aligned} \mathbf{w}^{(k)} &:= \mathbf{w}^{(k-1)} - \alpha \nabla f\big(\mathbf{w}^{(k-1)}\big) \\ &= \mathbf{w}^{(k-1)} + (2\alpha/m) \sum_{r=1}^{m} \mathbf{x}^{(r)} \big(y^{(r)} - \big(\mathbf{w}^{(k-1)}\big)^T \mathbf{x}^{(r)}\big). \end{aligned} \tag{5}$$

How much computation do we need for one iteration of (5)? How many iterations do we need ? We will try to answer the latter question in Lecture 4. The first question can be answered more easily. Indeed, a naive evaluation of (5) requires around m arithmetic operations (addition, multiplication).

It is instructive to consider the special case of a linear model which does not use any feature, i.e., $h(\mathbf{x}) = w$. For this extreme case, the ERM (3) has a simple closed-form solution:

$$\widehat{w} = (1/m) \sum_{r=1}^{m} x^{(r)}. \tag{6}$$

Thus, for this special case of the linear model, solving (6) amounts to summing m numbers $x^{(1)}, \ldots, x^{(m)}$. It seems reasonable to assume that the amount of computation required for computing (6) is proportional to m.

```
for iter_m=range(100)
    x = randn()
    tic
    np.sum(x)
    toc
```

2.3 Statistical Aspects of ERM

We have formulated the training of a linear model on a given training set as ERM (3). But how useful is its solution $\widehat{\mathbf{w}}$ for predicting the labels of data points outside the training set? Consider applying the learnt hypothesis $h^{(\widehat{\mathbf{w}})}$ to an arbitrary data point with label y and features $\mathbf{x}$ that is not contained in the training set. What can we say about the resulting prediction error $y - h^{(\widehat{\mathbf{w}})}(\mathbf{x})$ in general? In other words, how well does $h^{(\widehat{\mathbf{w}})}$ generalize beyond the training set.

Maybe the most widely used approach to study generalization of ML methods is via a probabilistic perspective. Here, we interpret each data point as a realization of an i.i.d. RV with probability distribution $p(\mathbf{x}, y)$. Under this i.i.d. assumption, we can evaluate the overall performance of a hypothesis $h \in \mathcal{H}$ via the expected loss

$$\mathbb{E}\{L\left((\mathbf{x}, y), h\right)\}.$$

One example for a probability distribution $p(\mathbf{x}, y)$ is obtained via relating the label y with the features $\mathbf{x}$ of a data point as

$$y = \overline{\mathbf{w}}^{T}\mathbf{x} + \varepsilon \text{ with } \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \varepsilon \sim \mathcal{N}(0, \sigma^{2}). \tag{7}$$

It is instructive to consider the special case of using no features at all,

$$y = \overline{w} + \varepsilon \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{8}$$

A simple calculation reveals the expected squared error loss of a given hypothesis $h(\mathbf{x}) = \hat{w}$ as

$$\mathbb{E}\{(y - h(\mathbf{x}))^2\} = \|\overline{\mathbf{w}} - \widehat{\mathbf{w}}\|^2 + \sigma^2. \tag{9}$$

The component σ^2 can be interpreted as the level of the noise contained in the label y. We cannot hope to find a hypothesis with expected loss smaller than this level. The first component of the RHS in (9) is the estimation error $\|\overline{\mathbf{w}} - \widehat{\mathbf{w}}\|^2$ of a ML method that reads in the training set and delivers an estimate $\hat{\mathbf{w}}$ (e.g., via (3)) for the parameters of a linear hypothesis.

2.4 Exercise

Carefully read and follow the instructions stated in the comments of the Python program `MLBasics.py`. This program reads in data points that represent hourly temperature measurements at different locations in Finland. Each data point is characterized by

- the coordinates (latitude and longitude) of the location,
- a time stamp that indicates when the temperature has been measured
- the temperature measurement

3 Lecture - "FL Design Principle"

After this lecture, you should

- be familiar with the concept of an empirical graph
- know how connectivity is related to spectrum of Laplacian matrix
- know some measures for the variation of local models l
- be familiar with the concept of GTVMin

4 Lecture - "Gradient Methods"

5 Lecture - "FL Algorithms"

6 Lecture - "FL Main Flavors"

7 Lecture - "Graph Learning"

8 Lecture - "Trustworthy FL"

9 Lecture - "Privacy-Protection in FL"

10 Lecture - "Data Poisoning in FL"

Glossary

k-means The k-means algorithm is a hard clustering method which assigns each data points to precisely one out of k different clusters. The method iteratively updates this assignment in order to minimize the average distance between data points in their nearest cluster mean (centre). 20

activation function Each artificial neuron within an artificial neural network (ANN) consists of an activation function that maps the inputs of the neuron to a single output value. In general, an activation function is a non-linear map of the weighted sum of neuron inputs (this weighted sum is the activation of the neuron). 54

artificial intelligence Artificial intelligence aims to develop systems that behave rational in the sense of maximizing a long-term reward. 11

artificial neural network An artificial neural network is a graphical (signal-flow) representation of a map from features of a data point at its input to a predicted label at its output. 19, 34, 54

Bayes estimator A hypothesis h whose Bayes risk is minimal [27]. 19

Bayes risk We use the term Bayes risk as a synonym for the risk or expected loss of a hypothesis. Some authors reserve the term Bayes risk for the risk of a hypothesis that achieves minimum risk, such a hypothesis being referred to as a Bayes estimator [27]. 19

bias Consider some unknown quantity $\bar{w}$, e.g., the true weight in a linear model $y = \bar{w}x + e$ relating feature and label of a data point. We might

use an ML method (e.g., based on ERM) to compute an estimate $\hat{w}$ for the $\bar{w}$ based on a set of data points that are realizations of RVs. The (squared) bias incurred by the estimate $\hat{w}$ is typically defined as $B^2 := \big(\mathbb{E}\{\hat{w}\} - \bar{w}\big)^2$. We extend this definition to vector-valued quantities using the squared Euclidean norm $B^2 := \big\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\big\|_2^2$. 53

classification Classification is the task of determining a discrete-valued label y of a data point based solely on its features $\mathbf{x}$. The label y belongs to a finite set, such as $y \in \{-1, 1\}$, or $y \in \{1, \ldots, 19\}$ and represents a category to which the corresponding data point belongs to. 52

clustering Clustering methods decompose a given set of data points into few subsets, which are referred to as clusters. Each cluster consists of data points that are more similar to each other than to data points outside the cluster. Different clustering methods use different measures for the similarity between data points and different representation of clusters. The clustering method k-means uses the average feature vector ("cluster means") of a cluster as its representative. A popular soft-clustering method based on Gaussian mixture model (GMM) represents a cluster by a multivariate normal distribution. 19

computational aspects By computational aspects of a ML method, we mainly refer to the computational resources required for its implementation. For example, if a ML method uses iterative optimization technniques to solve ERM, then its computational aspects include (i) how many arithmetic operations are needed to implement a single iteration (gradient step) and (ii) how many iterations are needed to

obtain useful model parameters. One important example for an iterative optimization technique is GD. 14, 31

convex A set $\mathcal{C} \subseteq \mathbb{R}^d$ is convex if it contains the line segment between any two points of that set. We define a function as convex if its epigraph is a convex set [26]. 14

covariance matrix The covariance matrix of a RV $\mathbf{x} \in \mathbb{R}^d$ is defined as $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$. 25, 33, 50

data A (indexed) set of data points. 5, 9, 34, 35

data point A data point is any object that conveys information [28]. Data points might be students, radio signals, trees, forests, images, RVs, real numbers or proteins. We characterize data points using two types of properties. One type of property is referred to as a feature. Features are properties of a data point that can be measured or computed in an automated fashion. Another type of property is referred to as labels. The label of a data point represents some higher-level fact (or quantity of interest). In contrast to features, determining the label of a data point typically requires human experts (domain experts). Roughly speaking, ML aims at predicting the label of a data point based solely on its features. 12–14, 16, 17, 19–25, 27–32, 34, 36–39, 46, 50–53, 55

data poisoning FL methods allow to leverage the information contained in local datasets generated by other parties to improve the training of a tailored model. Depending on how much we trust the other parties, FL can be compromised by data poisoning. Data poisoning refers to the

intentional manipulation (or fabrication) of local datasets to steer the training of a specific local model [29, 30]. 5

dataset With a slight abuse of notation we use the terms "dataset" or "set of data points" to refer to an indexed list of data points $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots$. Thus, there is a first data point $\mathbf{z}^{(1)}$, a second data point $\mathbf{z}^{(2)}$ and so on. Strictly speaking a dataset is a list and not a set [31]. By using indexed lists of data points we avoid some of the challenges arising in concept of an abstract set. 10, 14, 23, 28, 31, 51, 55

decision region Consider a hypothesis map h that reads in a feature vector $\mathbf{x} \in \mathbb{R}^d$ and delivers a value from a finite set $\mathcal{Y}$. The decision boundary induced by h is the set of vectors $\mathbf{x} \in \mathbb{R}^d$ that lie between differentdecision regions. More precisely, a vector $\mathbf{x}$ belongs to the decision boundary if and only if each neighbourhood $\{\mathbf{x}' : \|\mathbf{x} - \mathbf{x}'\| \leq \varepsilon\}$, for any $\varepsilon > 0$, contains at least two vectors with different function values. 28

decision region Consider a hypothesis map h that delivers values from a finite set $\mathcal{Y}$. We refer to the set of features $\mathbf{x} \in \mathcal{X}$ that result in the same output $h(\mathbf{x}) = a$ as a decision region of the hypothesis h. 22, 30, 54

decision tree A decision tree is a flow-chart like representation of a hypothesis map h. More formally, a decision tree is a directed graph which reads in the feature vector $\mathbf{x}$ of a data point at its root node. The root node then forwards the data point to one of its children nodes based on some elementary test on the features $\mathbf{x}$. If the receiving children node

is not a leaf node, i.e., it has itself children nodes, it represents another test. Based on the test result, the data point is further pushed to one of its neighbours. This testing and forwarding of the data point is repeated until the data point ends up in a leaf node (having no children nodes). The leaf nodes represent sets (decision regions) constituted by feature vectors $\mathbf{x}$ that are mapped to the same function value $h(\mathbf{x})$. 8, 54

differentiable A function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable if it has a gradient $\nabla f(\mathbf{x})$ everywhere (for every $\mathbf{x} \in \mathbb{R}^d$) [22]. 26, 37, 54

eigenvalue We refer to a number $\lambda \in \mathbb{R}$ as eigenvalue of a square matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ if there is a non-zero vector $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$. 54

empirical graph Empirical graphs represent collections of local datasets and corresponding local models [32]. An empirical graph is an undirected weighted empirical graph whose nodes carry local datasets and models. FL methods learn a local hypothesis $h^{(i)}$, for each node $i \in \mathcal{V}$, such that it incurs small loss on the local datasets. 9, 10, 18, 31, 33, 55

empirical risk The empirical risk of a given hypothesis on a given set of data points is the average loss of the hypothesis computed over all data points in that set. 13, 23, 33, 38, 53

empirical risk minimization Empirical risk minimization is the optimization problem of finding the hypothesis with minimum average loss (or empirical risk) on a given set of data points (the training set). Many ML methods are special cases of empirical risk. 10, 12–15, 20, 33, 36, 39

Euclidean space The Euclidean space $\mathbb{R}^d$ of dimension d refers to the space of all vectors $\mathbf{x} = \big(x_1, \ldots, x_d\big)$, with real-valued entries $x_1, \ldots, x_d \in \mathbb{R}$, whose geometry is defined by the inner product $\mathbf{x}^T\mathbf{x}' = \sum_{j=1}^{d} x_j x'_j$ between any two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ [22]. 6, 25, 30, 35

feature A feature of a data point is one of its properties that can be measured or computed in an automated fashion. For example, if a data point is a bitmap image, then we could use the red-green-blue intensities of its pixels as features. Some widely used synonyms for the term feature are "covariate","explanatory variable", "independent variable", "input (variable)", "predictor (variable)" or "regressor" [33–35]. However, this book makes consequent use of the term features for low-level properties of data points that can be measured easily. 12–14, 16, 21, 24, 25, 27, 28, 30–32, 34, 36–38, 46, 50–55

feature map A map that transforms the original features of a data point into new features. The so-obtained new features might be preferable over the original features for several reasons. For example, the shape of datasets might become simpler in the new feature space, allowing to use linear models in the new features. Another reason could be that the number of new features is much smaller which is preferable in terms of avoiding overfitting. The special case of feature maps that deliver two numeric features are particulary useful for data visualization. Indeed, we can then depict data points in a scatterplot by using these two features as the coordinates of a data point. 54

feature space The feature space of a given ML application or method is

constituted by all potential values that the feature vector of a data point can take on. Within this book the most frequently used choice for the feature space is the Euclidean space $\mathbb{R}^d$ with dimension d being the number of individual features of a data point. 27, 51, 54

federated learning (FL) Federated learning is an umbrella term for ML methods that train models in a collaborative fashion using decentralized data and computation. 1, 4–11, 21, 23

Finnish Meteorological Institute The Finnish Meteorological Institute is a government agency responsible for gathering and reporting weather data in Finland. 12

Gaussian mixture model Gaussian mixture models (GMM) are a family of probabilistic models for data points characterized by a numeric feature vector $\mathbf{x}$. A GMM interprets $\mathbf{x}$ as being drawn from one out of k different multivariate normal distributions $p^{(c)} = \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)$, indexed by $c = 1, \ldots, k$. The probability that $\mathbf{x}$ is drawn from the c-th multivariate normal distribution is denoted p_c. Thus, a GMM is parametrized by the probability p_c, the mean vector $\boldsymbol{\mu}^{(c)}$ and covariance matrix $\boldsymbol{\Sigma}^{(c)}$ for each $c = 1, \ldots, k$. 20

General Data Protection Regulation The General Data Protection Regulation (GDPR) is a law that has been passed by the European Union (EU) and put into effect on May 25, 2018 `https://gdpr.eu/tag/gdpr/`. The GDPR imposes obligations onto organizations anywhere, so long as they target, collect or in any other way process data related to people (i.e., personal data) in the EU. 8

generalized total variation Generalized total variation measures the changes of vector-valued node attributes of a graph. 27

gradient For a real-valued function $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, a vector $\mathbf{g}$ such that $\lim_{\mathbf{w}\to\mathbf{w}'} \frac{f(\mathbf{w})-\left(f(\mathbf{w}')+\mathbf{g}^T(\mathbf{w}-\mathbf{w}')\right)}{\|\mathbf{w}-\mathbf{w}'\|} = 0$ is referred to as the gradient of f at $\mathbf{w}'$. If such a vector exists it is denoted $\nabla f(\mathbf{w}')$ or $\nabla f(\mathbf{w})\big|_{\mathbf{w}'}$ [22]. 4, 5, 23, 26, 37, 54

gradient descent (GD) Gradient descent is an iterative method for finding the minimum of a differentiable function $f(\mathbf{w})$. 7, 10, 21, 29, 30

gradient step Given a differentiable real-valued function $f(\mathbf{w})$ and a vector $\mathbf{w}'$, the gradient step updates $\mathbf{w}'$ by adding the scaled negative gradient $\nabla f(\mathbf{w}')$, $\mathbf{w}' \mapsto \mathbf{w}' - \alpha\nabla f(\mathbf{w}')$. 15, 20

gradient-based method Gradient-based methods are iterative algorithms for finding the minimum (or maximum) of a differentiable objective function of the model parameters. These algorithms construct a sequence of approximations to an optimal choice for model parameters that results in a minimum objective function value. As their name indicates, gradient-based methods use the gradients of the objective function evaluated during previous iterations to construct new (hopefully) improved model parameters. 7, 52

graph A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair that consists of a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. In general, a graph is specified by a map that assigns to each edge $e \in \mathcal{E}$ a pair of nodes [36]. One important family of graphs (simple undirected graphs) is obtained by identifying each edge $e \in \mathcal{E}$

with two different nodes $\{i, i'\}$. Weighted graphs also specify numeric weights A_e for each edge $e \in \mathcal{E}$. 23

GTV minimization GTV minimization is an instance of RERM using the generalized total variation (GTV) of local model parameters as a regularizer. 9–11, 18

hinge loss Consider a data point that is characterized by a feature vector $\mathbf{x} \in \mathbb{R}^d$ and a binary label $y \in \{-1, 1\}$. The hinge loss incurred by a specific hypothesis h is defined as

$$L\left((\mathbf{x}, y), h\right) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{10}$$

A regularized variant of the hinge loss is used by the support vector machine (SVM) [37] to learn a linear classifier with maximum margin between the two classes (see Figure 1). 28, 38

hypothesis A map (or function) $h : \mathcal{X} \to \mathcal{Y}$ from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$. Given a data point with features $\mathbf{x}$ we use a hypothesis map h to estimate (or approximate) the label y using the predicted label $\hat{y} = h(\mathbf{x})$. ML is about learning (or finding) a hypothesis map h such that $y \approx h(\mathbf{x})$ for any data point. 12–14, 16, 17, 19, 22, 23, 27, 29, 31–34, 36–39, 46, 52–55

hypothesis space Every practical ML method uses a specific hypothesis space (or model) $\mathcal{H}$. The hypothesis space of a ML method is a subset of all possible maps from the feature space to label space. The design choice of the hypothesis space should take into account available

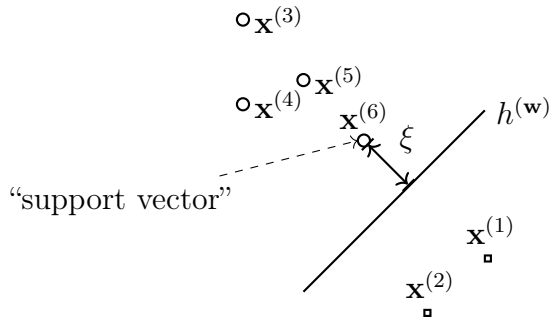

Figure 1: The SVM learns a hypothesis (or classifier) $h^{(\mathbf{w})}$ with minimum average soft-margin hinge loss. Minimizing this loss is equivalent to maximizing the margin ξ between the decision boundary of $h^{(\mathbf{w})}$ and each class of the training set.

computational resources and statistical aspects. If the computational infrastructure allows for efficient matrix operations, and there is a (approximately) linear relation between features and label, a useful choice for the hypothesis space might be the linear model. 12, 30–33, 36, 38, 39, 52

i.i.d. It can be useful to interpret data points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ as realizations of independent and identically distributed RVs with a common probability distribution. If these RVs are continuous, their joint probability density function (pdf) is $p\big(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big) = \prod_{r=1}^{m} p\big(\mathbf{z}^{(r)}\big)$ with $p(\mathbf{z})$ being the common marginal pdf of the underlying RVs. 6, 16, 28, 37

i.i.d. assumption The i.i.d. assumption interprets data points of a dataset as the realizations of i.i.d. RVs. 16, 37

interpretability A ML method is interpretable for a specific user if she can well anticiapte the predictions delivered by the method. The notion of interpretability can be made precise using quantitative measures of the uncertainty about the predictions [38]. 31

label A higher level fact or quantity of interest associated with a data point. If a data point is an image, its label might be the fact that it shows a cat (or not). Some widely used synonyms for the term label are "response variable", "output variable" or "target" [33–35]. 12–14, 17, 21, 27–32, 34, 36, 38, 46, 50, 52, 55

label space Consider a ML application that involves data points characterized by features and labels. The label space is constituted by all potential values that the label of a data point can take on. Regression methods, aiming at predicting numeric labels, often use the label space $\mathcal{Y} = \mathbb{R}$. Binary classification methods use a label space that consists of two different elements, e.g., $\mathcal{Y} = \{-1, 1\}$, $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{$“cat image”,”no cat image”$\}$ 27

learning rate Consider an iterative method for finding or learning a good choice for a hypothesis. Such an iterative method repeats similar computational (update) steps that adjust or modify the current choice for the hypothesis to obtain an improved hypothesis. A prime example for such an iterative learning method is GD and its variants. We refer by learning rate to any parameter of an iterative learning method that controls the extent by which the current hypothesis might be modified or improved in each iteration. A prime example for such a parameter

is the step size used in GD. Some authors use the term learning rate mostly as a synonym for the step size of (a variant of) GD 52

learning task A learning tasks consists of a specific choice for a collection of data points (e.g., all images stored in a particular database), their features and labels. 6, 53

least absolute shrinkage and selection operator (Lasso) The least absolute shrinkage and selection operator (Lasso) is an instance of structural risk minimization (SRM) for learning the weights $\mathbf{w}$ of a linear map $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The Lasso minimizes the sum consisting of an average squared error loss (as in linear regression) and the scaled ℓ_1 norm of the weight vector $\mathbf{w}$. 36

linear classifier A classifier $h(\mathbf{x})$ maps the feature vector $\mathbf{x} \in \mathbb{R}^d$ of a data point to a predicted label $\hat{y} \in \mathcal{Y}$ out of a finite set of label values $\mathcal{Y}$. We can characterize such a classifier equivalently by the decision regions $\mathcal{R}_a$, for every possible label value $a \in \mathcal{Y}$. Linear classifiers are such that the boundaries between the regions $\mathcal{R}_a$ are hyperplanes in the Euclidean space $\mathbb{R}^d$. 27

linear model This book uses the term linear model in a very specific sense. In particular, a linear model is a hypothesis space which consists of all linear maps,

$$\mathcal{H}^{(d)} := \left\{ h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\}. \tag{11}$$

Note that (11) defines an entire family of hypothesis spaces, which is parametrized by the number d of features that are linearly combined

to form the prediction $h(\mathbf{x})$. The design choice of d is guided by computational aspects (smaller d means less computation), statistical aspects (increasing d might reduce prediction error) and interpretability (a linear model using few carefully chosen features might be considered interpretable). 13, 19, 28

linear regression Linear regression aims at learning a linear hypothesis map to predict a numeric label based on numeric features of a data point. The quality of a linear hypothesis map is measured using the average squared error loss incurred on a set of labeled data points (which we refer to as training set). 5, 30

local dataset The concept of a local dataset is in-between the concept of a data point and a dataset. A local dataset consists of several individual data points which are characterized by features and labels. In contrast to a single dataset used in basic ML methods, a local dataset is also related to other local datasets via different notions of similarities. These similarities might arise from probabilistic models or communication infrastructure and are encoded in the edges of an empirical graph. 1, 4–10, 21–23, 31, 33, 55

local model Consider a collections of local datasets that are assigned to the nodes of an empirical graph. A local model $\mathcal{H}^{(i)}$ is a hypothesis space that is assigned to a node $i \in \mathcal{V}$. Different nodes might be assigned different hypothesis spaces, i.e., in general $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ for different nodes $i, i' \in \mathcal{V}$. 1, 6, 8, 18, 22, 23, 55

loss With a slight abuse of language, we use the term loss either for the loss

function itself or for its value for a specific pair of a data point and a hypothesis. 19, 23, 28, 32, 33, 36–39, 53, 55

loss function A loss function is a map

$$L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}_+ : \big(\big(\mathbf{x}, y\big), h\big) \mapsto L\left((\mathbf{x}, y), h\right)$$

which assigns a pair consisting of a data point, with features $\mathbf{x}$ and label y, and a hypothesis $h \in \mathcal{H}$ the non-negative real number $L\left((\mathbf{x}, y), h\right)$. The loss value $L\left((\mathbf{x}, y), h\right)$ quantifies the discrepancy between the true label y and the predicted label $h(\mathbf{x})$. Smaller (closer to zero) values $L\left((\mathbf{x}, y), h\right)$ mean a smaller discrepancy between predicted label and true label of a data point. Figure 2 depicts a loss function for a given data point, with features $\mathbf{x}$ and label y, as a function of the hypothesis $h \in \mathcal{H}$. 9, 12, 13, 15, 31, 32, 37

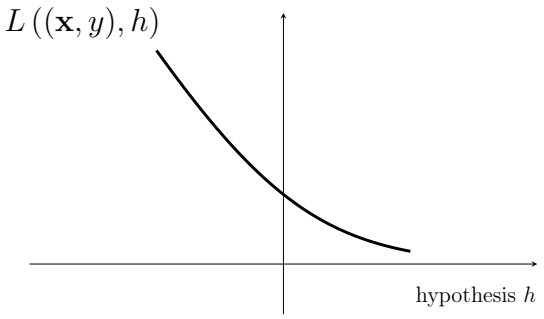

Figure 2: Some loss function $L\left((\mathbf{x}, y), h\right)$ for a fixed data point, with feature vector $\mathbf{x}$ and label y, and varying hypothesis h. ML methods try to find (learn) a hypothesis that incurs minimum loss.

model We use the term model as a synonym for hypothesis space 9, 12, 14, 21, 23, 27, 33, 52

model parameters Model parameters are numbers that select a hypothesis map out of a hypothesis space. 9, 21, 26, 27, 54

multivariate normal distribution The multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ is an important family ofprobability distributions for a continuous RV $\mathbf{x} \in \mathbb{R}^d$ [?, 39, 40]. This family is paramtrized by the mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$ of $\mathbf{x}$. If the covariance matrix is invertible, the probability distribution of $\mathbf{x}$ is

$$p(\mathbf{x}) \propto \exp\Big(-(1/2)\big(\mathbf{x}-\mathbf{m}\big)^T \mathbf{C}^{-1}\big(\mathbf{x}-\mathbf{m}\big)\Big).$$

20, 25, 50

networked data Networked data consists of local datasets that are related by some notion of pair-wise similarity. We represent networked data using an empirical graph whose nodes carry local datasets and an edge indicates a similarity between the connected nodes. 1, 4

overfitting Consider a ML method that uses ERM to learn a hypothesis with minimum empirical risk on a given training set. Such a method is "overfitting" the training set if it learns hypothesis with small empirical risk on the training set but unacceptably large loss outside the training set. 24

parameters The parameters of a ML model are tunable (learnable or adjustable) quantities that allow to choose between different hypothesis maps. For example, the linear model $\mathcal{H} := \{h : h(x) = w_1 x + w_2\}$ consists of all hypothesis maps $h(x) = w_1 x + w_2$ with a particular choice

for the parameters w_1, w_2. Another example of parameters are the weights assigned to the connections of an ANN. 15, 17

positive semi-definite A symmetric matrix $\mathbf{Q} = \mathbf{Q}^T \in \mathbb{R}^{d \times d}$ is referred to as positive semi-definite if $\mathbf{x}^T \mathbf{Q} \mathbf{x} \geq 0$ for every vector $\mathbf{x} \in \mathbb{R}^d$. 49, 54

prediction A prediction is an estimate or approximation for some quantity of interest. ML revolves around learning or finding a hypothesis map h that reads in the features $\mathbf{x}$ of a data point and delivers a prediction $\widehat{y} := h(\mathbf{x})$ for its label y. 16, 31, 37, 46, 53

probabilistic model A probabilistic model interprets data points as realizations of RVs with a joint probability distribution.This joint probability distribution typically involves parameters which have to be manually chosen (=design choice) or learnt via statistical inference methods [27]. 25, 31, 38

probability density function (pdf) The probability density function (pdf) $p(x)$ of a real-valued RV $x \in \mathbb{R}$ is a particular representation of its probability distribution. If the pdf exists, it can be used to compute the probability that x takes on a value from a (measurable) set $\mathcal{B} \subseteq \mathbb{R}$ via $p(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [41, Ch. 3]. The pdf of a vector-valued RV $\mathbf{x} \in \mathbb{R}^d$ (if it exists) allows to compute the probability that $\mathbf{x}$ falls into a (measurable) region $\mathcal{R}$ via $p(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [41, Ch. 3]. 28

probability distribution The data generated in some ML applications can be reasonably well modelled as realizations of a RV. The overall

statistical properties (or intrinsic structure) of such data are then governed by the probability distribution of this RV. We use the term probability distribution in a highly informal manner and mean the collection of probabilities assigned to different values or value ranges of a RV. The probability distribution of a binary RV $y \in \{0,1\}$ is fully specified by the probabilities $p(y=0)$ and $p(y=1)\big(=1–p(y=0)\big)$. The probability distribution of a real-valued RV $x \in \mathbb{R}$ might be specified by a probability density function $p(x)$ such that $p(x \in [a,b]) \approx p(a)|b-a|$. In the most general case, a probability distribution is defined by a probability measure [39, 42]. 16, 28, 33, 34, 37, 38, 50

random variable (RV) A random variable is a mapping from a probability space $\mathcal{P}$ to a value space [42]. The probability space, whose elements are elementary events, is equipped with a probability measure that assigns a probability to subsets of $\mathcal{P}$. A binary random variable maps elementary events to a set containing two different values, e.g., $\{-1,1\}$ or $\{\text{cat}, \text{no cat}\}$. A real-valued random variable maps elementary events to real numbers $\mathbb{R}$. A vector-valued random variable maps elementary events to the Euclidean space $\mathbb{R}^d$. Probability theory uses the concept of measurable spaces to rigorously define and study the properties of (large) collections of random variables [39, 42]. 16, 20, 21, 28, 33–35, 37, 39, 50, 53

realization Consider a RV x which maps each element (outcome, or elementary event) $\omega \in \mathcal{P}$ of a probability space $\mathcal{P}$ to an element a of a measurable space $\mathcal{N}$ [22,42,43]. A realization of x is any element $a' \in \mathcal{N}$

such that there is an element $\omega' \in \mathcal{P}$ with $x(\omega') = a'$. 34, 37, 53

regression Regresion problems revolve around the problem of predicting a numeric label solely from the features of a data point. 52

regularization Regularization techniques modify the ERM principle such that the learnt hypothesis performs well (generalizes) beyond the training set. One specific implementation of regularization is to add a penalty or regularization term to the objective function of ERM (which is the average loss on the training set). This regularization term can be interpreted as an estimate for the increase in the expected loss (risk) compared to the average loss on the training set. 1, 7, 10, 36–38, 54

regularized empirical risk minimization Synoym for SRM. 10, 27

regularizer A regularizer assigns each hypothesis h from a hypothesis space $\mathcal{H}$ a quantitative measure $\mathcal{R}\big\{h\big\}$ for how much its prediction error on a training set might differ from its prediction errors on data points outside the training set. Ridge regression uses the regularizer $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_2^2$ for linear hypothesis maps $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [25, Ch. 3]. The least absolute shrinkage and selection operator (Lasso) uses the regularizer $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_1$ for linear hypothesis maps $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [25, Ch. 3]. 10, 27

ridge regression Ridge regression learns the parameter (or weight) vector $\mathbf{w}$ of a linear hypothesis map $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The quality of a particular choice for the parameter vector $\mathbf{w}$ is measured by the sum of two components. The first components is the average squared error loss

incurred by $h^{(\mathbf{w})}$ on a set of labeled data points (the training set). The second component is the scaled squared Euclidean norm $\lambda\|\mathbf{w}\|_2^2$ with a regularization parameter $\lambda > 0$. It can be shown that the effect of adding to $\lambda\|\mathbf{w}\|_2^2$ to the average squared error loss is equivalent to replacing the original data points by an ensemble of realizations of a RV centered around these data points. 36

risk Consider a hypothesis h that is used to predict the label y of a data point based on its features $\mathbf{x}$. We measure the quality of a particular prediction using a loss function $L\left((\mathbf{x}, y), h\right)$. If we interpret data points as the realizations of i.i.d. RVs, also the $L\left((\mathbf{x}, y), h\right)$ becomes the realization of a RV. Using such an i.i.d. assumption allows to define the risk of a hypothesis as the expected loss $\mathbb{E}\big\{L\left((\mathbf{x}, y), h\right)\big\}$. Note that the risk of h depends on both, the specific choice for the loss function and the probability distribution of the data points. 19, 36

scatterplot A visualization technique that depicts data points by markers in a two-dimensional plane. 24

smooth We refer to a real-valued function as smooth if it is differentiable and its gradient is continuous [44, 45]. In particular, a differentiable function $f(\mathbf{w})$ is refered to as β-smooth if the gradient $\nabla f(\mathbf{w})$ is Lipschitz continuous with Lipschitz constant β, i.e.,

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq \beta\|\mathbf{w} - \mathbf{w}'\|.$$

14

squared error loss The squared error loss measures the prediction error of a hypothesis h when predicting a numeric label $y \in \mathbb{R}$ from the features $\mathbf{x}$ of a data point. It is defined as

$$L\left((\mathbf{x}, y), h\right) := \big(y - \underbrace{h(\mathbf{x})}_{=\hat{y}}\big)^2. \tag{12}$$

13, 17, 31, 36, 37

statistical aspects By statistical aspects of a ML method, we refer to (properties of) the probability distribution of its ouput given a probabilistic model for the data fed into the method. 14, 31

structural risk minimization Structural risk minimization is the problem of finding the hypothesis that optimally balances the average loss (or empirical risk) on a training set with a regularization term. The regularization term penalizes a hypothesis that is not robust against (small) perturbations of the data points in the training set. 30, 36

support vector machine A binary classification method for learning a linear hypothesis map that maximally separates data points from the two different classes in the feature space ("maximum margin"). Maximizing this separation is equivalent to minimizing a regularized variant of the hinge loss (10). 27, 28

training error The average loss of a hypothesis when predicting the labels of data points in a training set. We sometimes refer by training error also the minimum average loss incurred on the training set by any hypothesis out of a hypothesis space. 39, 53

training set A set of data points that is used in ERM to learn a hypothesis $\hat{h}$. The average loss of $\hat{h}$ on the training set is referred to as the training error. The comparison between training error and validation error of $\hat{h}$ allows to diagnose ML methods and informs how to improve them (e.g., using a different hypothesis space or collecting more data points). 14, 16, 17, 23, 28, 31, 33, 36–39, 52, 53

validation Consider a hypothesis $\hat{h}$ that has been learn via ERM on some training set $\mathcal{D}$. Validation refers to the practice of trying out a hypothesis $\hat{h}$ on a validation set that consists of data points that are not contained in the training set $\mathcal{D}$. 12

validation error Consider a hypothesis $\widehat{h}$ which is obtained by ERM on a training set. The average loss of $\widehat{h}$ on a validation set, which is different from the training set, is referred to as the validation error. 39, 53

validation set A set of data points that have not been used as training set in ERM to learn a hypothesis $\widehat{h}$. The average loss of $\widehat{h}$ on the validation set is referred to as the validation error and used to diagnose the ML method (see [25, Sec. 6.6.]). The comparison between training error and validation error can inform directions for improvements of the ML method (such as using a different hypothesis space). 39, 53

variance The variance of a real-valued RV x is defined as the expectation $\mathbb{E}\big\{\big(x - \mathbb{E}\{x\}\big)^2\big\}$ of the squared difference x and its expectation $\mathbb{E}\{x\}$. We extend this definition to vector-valued RVs $\mathbf{x}$ as $\mathbb{E}\big\{\big\|\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big\|_2^2\big\}$. 53

References

[1] S. Cui, A. Hero, Z.-Q. Luo, and J. Moura, Eds., *Big Data over Networks*. Cambridge Univ. Press, 2016.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017. [Online]. Available: https://doi.org/10.1109/MC.2017.9

[4] H. Ates, A. Yetisen, F. Güder, and C. Dincer, "Wearable devices for the detection of covid-19," *Nature Electronics*, vol. 4, no. 1, pp. 13–14, 2021. [Online]. Available: https://doi.org/10.1038/s41928-020-00533-1

[5] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361517307285

[6] M. E. J. Newman, *Networks: An Introduction*. Oxford Univ. Press, 2010.

[7] A. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: a network-based approach to human disease," *Nature Reviews Genetics*, vol. 12, no. 56, 2011.

[8] K. Grantz, H. Meredith, D. Cummings, C. Metcalf, B. Grenfell, J. Giles, S. Mehta, S. Solomon, A. Labrique, N. Kishore, C. Buckee, and A. Wesolowski, "The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology," *Nature Communications*, vol. 11, no. 1, p. 4961, 2020. [Online]. Available: https://doi.org/10.1038/s41467-020-18190-5

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020.

[11] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving ai," *Communications of the ACM*, vol. 63, no. 12, pp. 33–36, Dec. 2020.

[12] N. Agarwal, A. Suresh, F. Yu, S. Kumar, and H. McMahan, "cpSGD: Communication-efficient and differentially-private distributed sgd," in *Proc. Neural Inf. Proc. Syst. (NIPS)*, 2018.

[13] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated Multi-Task Learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6211080fa89981f66b1a0c9d55c61d0f-Paper.pdf

[14] J. You, J. Wu, X. Jin, and M. Chowdhury, "Ship compute or ship data? why not both?" in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, April 2021, pp. 633–651. [Online]. Available: https://www.usenix.org/conference/nsdi21/presentation/you

[15] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[16] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," 2018. [Online]. Available: https://arxiv.org/abs/1812.02903

[17] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.

[18] F. Sattler, K. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[19] G. Strang, *Computational Science and Engineering*. Wellesley-Cambridge Press, MA, 2007.

[20] ——, *Introduction to Linear Algebra*, 5th ed. Wellesley-Cambridge Press, MA, 2016.

[21] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.

[22] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York: McGraw-Hill, 1976.

[23] F. Pedregosa, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[24] J. Hirvonen and J. Suomela. (2023) Distributed algorithms 2020.

[25] A. Jung, *Machine Learning: The Basics*, 1st ed. Springer Singapore, Feb. 2022.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.

[27] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York: Springer, 1998.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New Jersey: Wiley, 2006.

[29] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

[30] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3310–3322, 2021.

[31] P. Halmos, *Naive set theory*. Springer-Verlag, 1974.

[32] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[33] D. Gujarati and D. Porter, *Basic Econometrics*. Mc-Graw Hill, 2009.

[34] Y. Dodge, *The Oxford Dictionary of Statistical Terms*. Oxford University Press, 2003.

[35] B. Everitt, *Cambridge Dictionary of Statistics*. Cambridge University Press, 2002.

[36] R. T. Rockafellar, *Network Flows and Monotropic Optimization*. Athena Scientific, Jul. 1998.

[37] C. Lampert, "Kernel methods in computer vision," *Foundations and Trends in Computer Graphics and Vision*, 2009.

[38] A. Jung and P. Nardelli, "An information-theoretic approach to personalized explainable machine learning," *IEEE Sig. Proc. Lett.*, vol. 27, pp. 825–829, 2020.

[39] R. Gray, *Probability, Random Processes, and Ergodic Properties*, 2nd ed. New York: Springer, 2009.

[40] A. Lapidoth, *A Foundation in Digital Communication.* New York: Cambridge University Press, 2009.

[41] D. Bertsekas and J. Tsitsiklis, *Introduction to Probability*, 2nd ed. Athena Scientific, 2008.

[42] P. Billingsley, *Probability and Measure*, 3rd ed. New York: Wiley, 1995.

[43] P. R. Halmos, *Measure Theory.* New York: Springer, 1974.

[44] Y. Nesterov, *Introductory lectures on convex optimization*, ser. Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004, vol. 87, a basic course. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-8853-9

[45] S. Bubeck, "Convex optimization. algorithms and complexity." in *Foundations and Trends in Machine Learning.* Now Publishers, 2015, vol. 8.

[46] W. Rudin, *Real and Complex Analysis*, 3rd ed. New York: McGraw-Hill, 1987.

[47] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.

Lists of Symbols

Sets and Functions

$a \in \mathcal{A}$	This statement indicates that the object a is an element of the set $\mathcal{A}$.
$a := b$	This statement defines a to be shorthand for b.
$\lvert\mathcal{A}\rvert$	The cardinality (number of elements) of a finite set $\mathcal{A}$.
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$.
$\mathcal{A} \subset \mathcal{B}$	$\mathcal{A}$ is a strict subset of $\mathcal{B}$.
$\mathbb{N}$	The set of natural numbers $1, 2, \ldots$.
$\mathbb{R}$	The set of real numbers x [46].
$\mathbb{R}_+$	The set of non-negative real numbers $x \geq 0$.
$\mathbb{R}_{++}$	The set of positive real numbers $x > 0$.
$h(\cdot): \mathcal{A} \to \mathcal{B} : a \mapsto h(a)$	A function (map) that accepts any element $a \in \mathcal{A}$ from a set $\mathcal{A}$ as input and delivers a well-defined element $h(a) \in \mathcal{B}$ of a set $\mathcal{B}$. The set $\mathcal{A}$ is the domain of the function h and the set $\mathcal{B}$ is the codomain of h. ML aims at finding (or learning) a function h ("hypothesis") that reads in the features $\mathbf{x}$ of a data point and delivers a prediction $h(\mathbf{x})$ for its label y.

$\{0, 1\}$	The binary set that consists of the two real numbers 0 and 1.
$[0, 1]$	The closed interval of real numbers x with $0 \leq x \leq 1$.
$\operatorname{argmin} f(\mathbf{w})$	The set of minimizers for a real-valued function $f(\mathbf{w})$.
$\log a$	The logarithm of the positive number $a \in \mathbb{R}_{++}$.

Matrices and Vectors

$\mathbf{I}_{l\times d}$	A generalized identity matrix with l rows and d columns. The entries of $\mathbf{I}_{l\times d} \in \mathbb{R}^{l\times d}$ are equal to 1 along the main diagonal and equal to 0 otherwise.
$\mathbf{I}$	A square identity matrix whose shape should be clear from the context.
$\mathbb{R}^d$	The set of vectors $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ consisting of d real-valued entries $x_1, \ldots, x_d \in \mathbb{R}$.
$\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$	A vector of length d. The jth entry of the vector is denoted x_j.
$\|\mathbf{x}\|_2$	The Euclidean (or "ℓ_2") norm of the vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ given as $\|\mathbf{x}\|_2 := \sqrt{\sum_{j=1}^d x_j^2}$.
$\|\mathbf{x}\|$	Some norm of the vector $\mathbf{x} \in \mathbb{R}^d$ [47]. Unless specified otherwise, we mean the Euclidean norm $\|\mathbf{x}\|_2$.
$\mathbf{x}^T$	The transpose of a vector $\mathbf{x}$ that is considered a single column matrix. The transpose is a single-row matrix $\big(x_1, \ldots, x_d\big)$.
$\mathbf{X}^T$	The transpose of a matrix $\mathbf{X} \in \mathbb{R}^{m\times d}$. A square real-valued matrix $\mathbf{X} \in \mathbb{R}^{m\times m}$ is called symmetric if $\mathbf{X} = \mathbf{X}^T$.
$\mathbf{0} = \big(0, \ldots, 0\big)^T$	A vector of zero entries.

$\left(\mathbf{v}^T, \mathbf{w}^T\right)^T$	The vector of length $d + d'$ obtained by concatenating the entries of vector $\mathbf{v} \in \mathbb{R}^d$ with the entries of $\mathbf{w} \in \mathbb{R}^{d'}$.
$\text{span}\{\mathbf{B}\}$	The span of a matrix $\mathbf{B} \in \mathbb{R}^{a \times b}$, which is the subspace of all linear combinations of columns of $\mathbf{B}$, $\text{span}\{\mathbf{B}\} = \left\{\mathbf{Ba} : \mathbf{a} \in \mathbb{R}^b\right\} \subseteq \mathbb{R}^a$.
$\mathbb{S}_+^d$	The set of all positive semi-definite (psd) matrices of size $d \times d$.
$\det(\mathbf{C})$	The determinant of the matrix $\mathbf{C}$.

Probability Theory

$\mathbb{E}_p\{f(\mathbf{z})\}$	The expectation of a function $f(\mathbf{z})$ of a RV $\mathbf{z}$ whose probability distribution is $p(\mathbf{z})$. If the probability distribution is clear from context we just write $\mathbb{E}\{f(\mathbf{z})\}$.
$p(\mathbf{x}, y)$	A (joint) probability distribution of a RV whose realizations are data points with features $\mathbf{x}$ and label y.
$p(\mathbf{x}\|y)$	A conditional probability distribution of a RV $\mathbf{x}$ given the value of another RV y [41, Sec. 3.5].
$p(\mathbf{x}; \mathbf{w})$	A parametrized probability distribution of a RV $\mathbf{x}$. The probability distribution depends on a parameter vector $\mathbf{w}$. For example, $p(\mathbf{x}; \mathbf{w})$ could be a multivariate normal distribution of a Gaussian RV $\mathbf{x}$ with the parameter vector $\mathbf{w}$ given by the entries of the mean vector $\mathbb{E}\{\mathbf{x}\}$ and the covariance matrix $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.
$\mathcal{N}(\mu, \sigma^2)$	The probability distribution of a scalar normal ("Gaussian") RV $x \in \mathbb{R}$ with mean (or expectation) $\mu = \mathbb{E}\{x\}$ and variance $\sigma^2 = \mathbb{E}\big\{(x-\mu)^2\big\}$.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$	The multivariate normal distribution of a vector-valued Gaussian RV $\mathbf{x} \in \mathbb{R}^d$ with mean (or expectation) $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\}$ and covariance matrix $\mathbf{C} = \mathbb{E}\big\{\big(\mathbf{x} - \boldsymbol{\mu}\big)\big(\mathbf{x} - \boldsymbol{\mu}\big)^T\big\}$.

Machine Learning

r	An index $r = 1, 2, \ldots,$ that enumerates data points.
m	The number of data points in (the size of) a dataset.
$\mathcal{D}$	A dataset $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ is a list of individual data points $\mathbf{z}^{(r)}$, for $r = 1, \ldots, m$.
d	Number of features that characterize a data point.
x_j	The jth feature of a data point. The first feature of a given data point is denoted x_1, the second feature x_2 and so on.
$\mathbf{x}$	The feature vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ of a data point whose entries are the individual features of a data point.
$\mathcal{X}$	The feature space $\mathcal{X}$ is the set of all possible values that the features $\mathbf{x}$ of a data point can take on.
$\mathbf{z}$	Beside the symbol $\mathbf{x}$, we sometimes use $\mathbf{z}$ as another symbol to denote a vector whose entries are features of a data point. We need two different symbols to distinguish between "raw" or "original" and learnt features [25, Ch. 9].
$\mathbf{x}^{(r)}$	The feature vector of the rth data point within a dataset.
$x_j^{(r)}$	The jth feature of the rth data point within a dataset.
$\mathcal{B}$	A mini-batch (subset) of randomly chosen data points.
B	The size of (the number of data points in) a mini-batch.

y	The label (quantity of interest) of a data point.
$y^{(r)}$	The label of the rth data point.
$\big(\mathbf{x}^{(r)}, y^{(r)}\big)$	The features and label of the rth data point.
$\mathcal{Y}$	The label space $\mathcal{Y}$ of a ML method consists of all potential label values that a data point can have. We often use label spaces that are larger than the set of different label values arising in a give dataset (e.g., a training set). We refer to a ML problems (methods) using a numeric label space, such as $\mathcal{Y} = \mathbb{R}$ or $\mathcal{Y} = \mathbb{R}^3$, as regression problems (methods). ML problems (methods) that use a discrete label space, such as $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{$"cat","dog","mouse"$\}$ are referred to as classification problems (methods).
α	learning rate (step-size) used by gradient-based methods.
$h(\cdot)$	A hypothesis map that reads in features $\mathbf{x}$ of a data point and delivers a prediction $\hat{y} = h(\mathbf{x})$ for its label y.
$\mathcal{Y}^{\mathcal{X}}$	Given two sets $\mathcal{X}$ and $\mathcal{Y}$, we denote by $\mathcal{Y}^{\mathcal{X}}$ the set of all possible hypothesis maps $h : \mathcal{X} \to \mathcal{Y}$.
$\mathcal{H}$	A hypothesis space or model used by a ML method. The hypothesis space consists of different hypothesis maps $h : \mathcal{X} \to \mathcal{Y}$ between which the ML method has to choose .

B^2	The squared bias of a learnt hypothesis $\hat{h}$ delivered by a ML algorithm that is fed with data points which are modelled as realizations of RVs. If data is modelled as realizations of RVs, also the delivered hypothesis $\hat{h}$ is the realization of a RV.
V	The variance of the (parameters of the) hypothesis delivered by a ML algorithm. If the input data for this algorithm is interpreted as realizations of RVs, so is the delivered hypothesis a realization of a RV.
$L\left((\mathbf{x}, y), h\right)$	The loss incurred by predicting the label y of a data point using the prediction $\hat{y} = h(\mathbf{x})$. The prediction $\hat{y}$ is obtained from evaluating the hypothesis $h \in \mathcal{H}$ for the feature vector $\mathbf{x}$ of the data point.
E_v	The validation error of a hypothesis h, which is its average loss incurred over a validation set.
$\widehat{L}\big(h\|\mathcal{D}\big)$	The empirical risk or average loss incurred by the predictions of hypothesis h for the data points in the dataset $\mathcal{D}$.
E_t	The training error of a hypothesis h, which is its average loss incurred over a training set.
t	A discrete-time index $t = 0, 1, \ldots$ used to enumerate a sequence to sequential events ("time instants").
t	An index that enumerates learning tasks within a multi-task learning problem.

λ	A regularization parameter that controls the amount of regularization.
$\lambda_j\big(\mathbf{Q}\big)$	The jth eigenvalue (sorted either ascending or descending) of a psd matrix $\mathbf{Q}$. We also use the shorthand λ_j if the corresponding matrix is clear from context.
$\sigma(\cdot)$	The activation function used by an artificial neuron within an ANN.
$\mathcal{R}_{\hat{y}}$	A decision region within a feature space.
$\mathbf{w}$	A parameter vector $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ whose entries are parameters of a model. These parameters could be feature weights in linear maps, the weights in ANNs or the thresholds used for splits in decision trees.
$h^{(\mathbf{w})}(\cdot)$	A hypothesis map that involves tunable model parameters $w_1, \ldots, w_d$ which are stacked into the vector $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$.
$\nabla f(\mathbf{w})$	The gradient of a differentiable real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ is the vector $\nabla f(\mathbf{w}) = \big(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_d}\big)^T \in \mathbb{R}^d$ [22, Ch. 9].
$\phi(\cdot)$	A feature map $\phi : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}' := \phi\big(\mathbf{x}\big) \in \mathcal{X}'$.

Federated Learning

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Empirical graph whose nodes $i \in \mathcal{V}$ carry local datasets and local models.
$i \in \mathcal{V}$	A node in the empirical graph that represents a local dataset and a corresponding local model. It might also be useful to think of node i as a small computer that can collect data and execute computations to train ML models.
$\mathcal{D}^{(i)}$	The local dataset $\mathcal{D}^{(i)}$ at node $i \in \mathcal{V}$.
m_i	The number of data points (sample size) contained in the local dataset $\mathcal{D}^{(i)}$ at node $i \in \mathcal{V}$.
$\mathcal{N}^{(i)}$	The neighbourhood of the node i in an empirical graph is the set of other nodes that are connected by end edge with i.
$\mathbf{x}^{(i,r)}$	The feature vector of the r-th data point in the local dataset $\mathcal{D}^{(i)}$.
$y^{(i,r)}$	The label of the r-th data point in the local dataset $\mathcal{D}^{(i)}$ at node $i \in \mathcal{V}$.
$L^{(\mathrm{d})}\big(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\big)$	The loss incurred by a "external" hypothesis h' on a data point with features $\mathbf{x}$ and predicted label $h\big(\mathbf{x}\big)$ that is obtained from some local hypothesis.