A Dictionary of Machine Learning

Comments are warmly welcome at `alex.jung@aalto.fi`.

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

March 14, 2024

Abstract

This dictionary of machine learning evolved during the design and implementation of the courses CS-E3210 **Machine Learning: Basic Principles**, CS-C3240 **Machine Learning**, CS-E4800 **Artificial Intelligence**, CS-EJ3211 **Machine Learning with Python**, CS-EJ3311 **Deep Learning with Python**, CS-E4740 **Federated Learning** and CS-E407507 **Human-Centered Machine Learning** offered to Finnish university students via **Aalto University** `https://www.aalto.fi/en`, to Finnish adult learners via **The Finnish Institute of Technology (FITech)** `fitech.io` and to international students via the **European University Alliance Unite!** `https://www.aalto.fi/en/unite`.

*please cite as: A. Jung, "A Dictionary of Machine Learning," Aalto University, 2023.

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 [1].
$\mathbb{R}_+$	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.
$\mathbf{I}_d, \mathbf{I}$	A square identity matrix of size $d \times d$. If the size is clear from the context, we drop the subscript.
$\mathbb{R}^d$	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 with its jth entry being 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 $\mathbf{x} \in \mathbb{R}^d$ [**?**]. 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 $\mathbf{X} = \mathbf{X}^T$.
$\mathbf{0} = \big(0, \ldots, 0\big)^T$	The vector in $\mathbb{R}^d$ with each entry equal to zero.
$\mathbf{1} = \big(1, \ldots, 1\big)^T$	The vector in $\mathbb{R}^d$ with each entry equal to one.

$\left(\mathbf{v}^T, \mathbf{w}^T\right)^T$	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 $\mathbf{C}$.
$\mathbf{A} \otimes \mathbf{B}$	The Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ [2].

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 [3, 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 with the parameter vector $\mathbf{w}$ given by the entries of te 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 [4, 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.
$\left(\mathbf{x}^{(r)}, y^{(r)}\right)$	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 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).
η	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 .
$d_{\text{eff}}(\mathcal{H})$	The effective dimension of a hypothesis space $\mathcal{H}$.

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.

α	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 artificial neural network (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 $w_1, \ldots, w_d$, 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 $\nabla f(\mathbf{w}) = \big(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_d}\big)^T \in \mathbb{R}^d$ [5, 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.
$\mathcal{G}^{(\mathcal{C})}$	The induced sub-graph of $\mathcal{G}$ using the nodes in $\mathcal{C} \subseteq \mathcal{V}$.
$\mathbf{L}^{(\mathcal{G})}$	The Laplacian matrix of a graph $\mathcal{G}$.
$\mathbf{L}^{(\mathcal{C})}$	The Laplacian matrix of the induced graph $\mathcal{G}^{(\mathcal{C})}$.
$\mathcal{D}^{(i)}$	The local dataset $\mathcal{D}^{(i)}$ at node $i \in \mathcal{V}$ of an empirical graph.
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.
$\mathbf{x}^{(i,r)}$	The features 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 $\mathcal{D}^{(i)}$.

$L^{(\mathrm{d})}\left(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\right)$	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.
$\mathrm{stack}\big\{\mathbf{w}^{(i)}\big\}_{i=1}^{n}$	The vector $\left(\big(\mathbf{w}^{(1)}\big)^{T}, \ldots, \big(\mathbf{w}^{(n)}\big)^{T}\right)^{T} \in \mathbb{R}^{dn}$ that is obtained by vertically stacking the local model parameters $\mathbf{w}^{(i)} \in \mathbb{R}^{d}$.

Glossary

0/1 loss The 0/1 loss $L\left((\mathbf{x}, y), h\right)$ measures the quality of a classifier $h(\mathbf{x})$ that delivers a prediction $\hat{y}$ (e.g., via thresholding (1)) for a label y of a data point. It is equal to 0 if the prediction is correct, i.e., $L\left((\mathbf{x}, y), h\right) = 0$ when $\hat{y} = y$. It is equal to 1 if the prediction is wrong, $L\left((\mathbf{x}, y), h\right) = 1$ when $\hat{y} \neq y$.

k-fold cross-validation (k-fold CV) k-fold cross-validation is a method for learning and validating a hypothesis using a given dataset. This method first divides the dataset evenly into k subsets or "folds" and then executes k repetitions of training and validation. Each repetition uses a different fold as the validation set and the remaining $k-1$ folds as a training set. The final output is the average of the validation errors obtained from the k repetitions.

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).

accuracy Consider data points characterized by features $\mathbf{x} \in \mathcal{X}$ and a categorical label y which takes on values from a finite label space $\mathcal{Y}$. The accuracy of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$, when applied to the data points in a dataset $\mathcal{D} = \left\{\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}$ is then defined as $1 - (1/m) \sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), h\right)$ using the 0/1 loss.

activation function Each artificial neuron within an 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).

Application Programming Interface (API) An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another software program that implements that API.

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

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.

backdoor A backdoor attack refers to the intentional manipulation of the training process underlying a ML method. The manipulation can be implemented by perturbing the training set (data poisoning) or optimization algorithm used by an empirical risk minimization (ERM) based method. The goal of a backdoor attack is to nudge the learnt hypothesis towards specific predictions for a certain range of feature values. This range of feature values serves as a key (or trigger) to unlock a "backdoor" in the sense of delivering anomolous predictions. These anomouls predictions are known to the attacker but cannot be inferred (easily) from the clean training set or optimization algorithm.

bagging bagging (or "bootstrap aggregation") is a generic technique to improve (the robustness of) a given ML method. The idea is to use the bootstrap to generate perturbed copy of a given training set and then apply the original ML method to learn a separate hypothesis for each perturbed copy of the training set. The resulting set of hypotheses is then used to predict the label of a data point by combining or aggregating the individual predictions of each individual hypothesis. For hypotheses that deliver numeric label values (regression methods) this aggregation could be implemented by computing the average of individual predictions.

baseline A reference value or benchmark for the average loss incurred by a hypothesis when applied to the data points generated in a specific ML application. Such a reference value might be obtained from human performance (e.g., error rate of dermatologists diagnosing cancer from visual inspection of skin areas) or other ML methods ("competitors")

batch A set of randomly selected data points out of a (typically very large) dataset.

Bayes estimator A hypothesis h whose Bayes risk is minimal [6].

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 [6].

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 $B^2 := \big\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\big\|_2^2$.

bootstrap Consider a probabilistic model that interprets a given set of data points $\mathcal{D} = \big\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big\}$ as realizations of independent and identically distributed (i.i.d.) RVs with a common probability distribution $p(\mathbf{z})$. The bootstrap uses the histogram of $\mathcal{D}$ as the underlying probability distribution $p(\mathbf{z})$.

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.

classifier A classifier is a hypothesis (map) $h(\mathbf{x})$ that is used to predict a discrete-valued label. Strictly speaking, a classifier is a hypothesis $h(\mathbf{x})$ that delivers values from a finite set $\mathcal{Y}$ (referred to as the label space). However, we are sometimes sloppy and use the term classifier also for a hypothesis that delivers a real number which is quantized, i.e., compared against a threshold, to obtain the predicted label value. For example, in a binary classification problem with label values $y \in \{-1, 1\}$, we refer to a hypothesis $h(\mathbf{x})$ as classifier if it is used to compute a predicted

label according to

$$\hat{y} = 1 \text{ for } h(\mathbf{x}) \geq 0, \text{ and } \hat{y} = -1 \text{ otherwise.} \tag{1}$$

cluster A cluster within an empirical graph is a subset of nodes that carry local datasets with (nearly) identical statistical distributions. Clustered federated learning (FL) aims at identying the clusters and use the pooled local darasets to train a separate model for each cluster.

cluster A cluster is a subset of data points that are more similar to each other than to the data points outside the cluster. The notion and measure of similarity between data points is a design choice. If data points are characterized by numeric feature vectors in some Euclidean space we can define the similarity between two data points via the Euclidean distance between their feature vectors.

clustered federated learning (CFL) Clustered FL assumes that local datasets clusters. The local datasets belonging to the same cluster have similar statistical properties.

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.

clustering assumption The clustering assumption postulates that data points in a dataset form a (small) number of groups or clusters. Data points in the same cluster are more similar with each other than with those outside the cluster [7]. We obtain different clustering methods by using different notions of similarity between data points.

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 gradient descent (GD).

condition number The condition number $\kappa(\mathbf{Q}) \geq 1$ of a psd matrix $\mathbf{Q}$ is the ratio $\lambda_{\max}/\lambda_{\min}$ between the largest $\lambda_{\max}$ and the smallest $\lambda_{\min}$ eigenvalue of $\mathbf{Q}$. One example for such a matrix $\mathbf{Q}$ is obtained from the feature matrix $\mathbf{X}$ of a dataset, $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$. From a computational perspective, we prefer $\mathbf{Q}$ to have a condition number close to 1.

confusion matrix Consider data points characterized by features $\mathbf{x}$ and label y having value $c \in \{1, \ldots, k\}$. The confusion matrix is $k \times k$ matrix with rows representing different values c of the true label of a data point. The columns of a confusion matrix correspond to different

values c' delivered by a hypothesis $h(\mathbf{x})$. The (c, c')-th entry of the confusion matrix is the fraction of data points with label $y = c$ and predicted label $\hat{y} = c'$.

connected graph An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected if we cannot find a (non-empty) subset $\mathcal{V}' \subset \mathcal{V}$ with no edges leaving $\mathcal{V}'$.

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 [8].

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

data A (indexed) set of data points.

data augmentation Data augmentation methods add synthetic data points to an existing set of data points. These synthetic data points might be obtained by perturbations (adding noise) or transformations (rotations of images) of the original data points.

data point A data point is any object that conveys information [9]. 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.

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 [10, 11].

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 [12]. By using indexed lists of data points we avoid some of the challenges arising in concept of an abstract set.

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.

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.

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 $h(\mathbf{x})$.

deep net We refer to an ANN with a (relatively) large number of hidden layers as a deep ANN or "deep net". Deep nets are used to represent the hypothesis spaces of deep learning methods [13].

degree of belonging A number that indicates the extend by which a data point belongs to a cluster. The degree of belonging can be interpreted as a soft cluster assignment. Soft clustering methods typically represent the degree of belonging by a real number in the interval $[0, 1]$. The boundary values 0 and 1 correspond to hard cluster assignments.

denial-of-service attack A denial-of-service attack uses data poisoning to steer the training of a local (client) model such that it performs poorly for typical data points

density-based spatial clustering of applications with noise (DBSCAN) A clustering algorithm for data points that are characterized by numeric feature vectors. Similar to k-means and soft clustering via GMM also DBSCAN uses the Euclidean distances between feature vectors to determine the clusters. However, in contrast to these other clustering methods, DBSCAN uses a different notion of similarity between data points. In particular, DBSCAN considers two data points as similar if they are "connected" via a sequence (path) of close-by intermediate data points. Thus, DBSCAN might consider two data points as similar (and therefore belonging to the same cluster) even if their feature vectors have a large Euclidean distance.

differentiable A function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable if it has a gradient $\nabla f(\mathbf{x})$ everywhere (for every $\mathbf{x} \in \mathbb{R}^d$) [5].

differential privacy Consider an algorithm used to compute some output based on a dataset. Differential privacy is a precise measure of this algorithm's privacy leakage. Roughly speaking, the algorithm is differentially private if the probability distribution of its output does not allow it to infer if the dataset contains a specific data point.

dimensionality reduction Dimensionality reduction methods map (typically many) raw features to a (relatively small) set of new features. These methods can be used to visualize data points by learning two features that can be used as the coordinates of a depication in a scatterplot.

discrepancy Consider a FL application with networked data represented by an empirical graph. FL methods use a discrepancy measure to compare

hypothesis maps from local models at nodes i, i' connected by an edge in the empirical graph.

edge weight Each edge $\{i, i'\}$ of an empirical graph is assigned a non-negative edge weight $A_{i,i'} \geq 0$. A zero weight $A_{i,i'} = 0$ indicates the absence of an edge between nodes $i, i' \in \mathcal{V}$.

effective dimension The effective dimension $d_{\text{eff}}(\mathcal{H})$ of an infinite hypothesis space $\mathcal{H}$ is a measure of its size. Loosely speaking, the effective dimension is equal to the number of "independent" tunable parameters of the model. These parameters might be the coefficients used in a linear map or the weights and bias terms of an ANN.

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 $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$.

eigenvalue decomposition The eigenvalue decomposition for a square matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a factorization of the form

$$\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{-1}.$$

The columns of the matrix $\mathbf{V} = \big(\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(d)}\big)$ are the eigenvectors of the matrix $\mathbf{V}$. The diagonal matrix $\mathbf{\Lambda} = \operatorname{diag}\big\{\lambda_1, \ldots, \lambda_d\big\}$ contains the eigenvalues λ_j corresponding to the eigenvectors $\mathbf{v}^{(j)}$. Note that the above decomposition exists only if the matrix $\mathbf{A}$ is diagonalizable.

eigenvector An eigenvector of a matrix $\mathbf{A}$ is a non-zero vector $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$ with some eigenvalue λ.

empirical graph Empirical graphs represent collections of local datasets and corresponding local models [7]. 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.

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.

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.

estimation error Consider data points with feature vectors $\mathbf{x}$ and label y. In some applications we can model the relation between features and label of a data point as $y = \bar{h}(\mathbf{x}) + \varepsilon$. Here we used some true hypothesis $\bar{h}$ and a noise term ε which might represent modelling or labelling errors. The estimation error incurred by a ML method that learns a hypothesis $\widehat{h}$, e.g., using ERM, is defined as $\widehat{h} - \bar{h}$. For a parametrized hypothesis space, consisting of hypothesis maps that are determined by a parameter vector $\mathbf{w}$, we define the estimation error in terms of parameter vectors as $\Delta\mathbf{w} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$. first

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 $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ [5].

expectation Consider a numeric RV $\mathbf{x}$ with probability distribution $p(\mathbf{x})$. The expectation of $\mathbf{x}$ is defined as the integral $\mathbb{E}\{\mathbf{x}\} := \int \mathbf{x} p(\mathbf{x})$ (if this integral is well-defined!) [5, 14, 15].

expectation maximization Expectation maximization is a generic technique for estimating the parameters of a probabilistic model (a parametrized probability distribution) $p(\mathbf{z}; \mathbf{w})$ from data [16–18]. Expectation maximization delivers an approximation to the maximum likelihood estimate for the model parameters $\mathbf{w}$.

expert ML aims at learning a hypothesis h that accurately predicts the label of a data point based on its features. We measure the prediction error using some loss function. Ideally we want to find a hypothesis that incurs minimum loss. One approach to make this goal precise is to use the i.i.d. assumption and use the resulting Bayes risk as the benchmark level for the (average) loss of a hypothesis. Alternatively, we might know a reference or benchmark hypothesis h' which might be obtained by some existing ML method. We can then compare the loss incurred by h against the loss incurred by h'. Such a reference or baseline hypothesis h' is referred to as an expert. Note that an expert might deliver very poor predictions. We typically compare against many different experts and aim at incurring not much more loss than the best among those experts (this is known as regret minimization) [19, 20]. first

explainability We can define the (subjective) explainability of a ML method

as the level of predictability (or lack of uncertainty) of the predictions delivered to a specific human user. Quantitative measures for the (subjective) explainability can be obtained via probabilistic models for the data fed into the ML method. In particular, we can then identify explainability of a ML method via the (differential) entropy of its predictions [21–23].

explainable empirical risk minimization An instance of structural risk minimization that adds a regularization term to the training error in ERM. The regularization term is chosen to favour hypothesis maps that are intrinsically explainable for a specific user. This user is characterized by user signals that she provides for the data points in a training set.

explainable machine learning Explainable ML methods aim at complementing predictions with explanations for how the prediction has been obtained.

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" [24–26]. However, this book makes consequent use of the term features for low-level properties of data points that can be measured easily.

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.

feature matrix Consider a dataset $\mathcal{D}$ of m data points that are characterized by feature vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$. It is convenient to collect these feature vectors into a feature matrix $\mathbf{X} := \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\big)$.

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.

federated averaging (FedAvg) Federated averaging is an iterative FL algorithm that alternates between local model trainings and averaging the resulting local model parameters. Different variants of this algorithm are obtained by different techniques for the model training. The authors of [27] consider federated averaging methods where the local model training is implemented by running several GD steps

federated learning (FL) Federated learning is an umbrella term for ML methods that train models in a collaborative fashion using decentralized

data and computation.

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

flow-based clustering Flow-basted clustering groups the nodes of an undirected graph by applying k-means clustering to node-wise feature vectors. These feature vectors are built from flows between carefully selected source and destination nodes [28].

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 $c = 1, \ldots, k$.

Gaussian random variable A Gaussian RV $\mathbf{x} \in \mathbb{R}^d$ with a multivariate normal distribution. The special case of d corresponds to a scalar Gaussian RV [3, 29, 30].

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.

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

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 $\nabla f(\mathbf{w})\big|_{\mathbf{w}'}$ [5].

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

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 $\nabla f(\mathbf{w}')$, $\mathbf{w}' \mapsto \mathbf{w}' - \eta \nabla f(\mathbf{w}')$.

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.

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 [31]. 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 $e \in \mathcal{E}$.

GTV minimization GTV minimization is an instance of regularized empirical risk minimization (RERM) using the generalized total variation (GTV) of local model parameters as a regularizer.

hard clustering Hard clustering refers to the task of partitioning a given set of data points into (few) non-overlapping clusters. Each data point is assigned to one specific cluster.

high-dimensional regime The high-dimensional regime of ERM is characterized by the effective dimension of the model being larger than the sample size, i.e., the number of (labeled) data points in the training set. For example, linear regression methods operate in the high-dimensional regime whenever the number d of features used to characterize data points exceeds the number of data points in the training set. Another example for ML methods that operate in the high-dimensional regime is deep learning with large ANNs, having more tunable weights (and bias terms) than the number of data points in the training set. High-dimensional statistics is a recent main thread of probability theory that studies the behavior of ML methods in the high-dimensional regime [32, 33].

Hilbert space A Hilbert space is a linear vector space that is equipped with an inner product between pairs of vectors. One important example for

a Hilbert space is the Euclidean spaces $\mathbb{R}^d$, for some dimension d, which consists of Euclidean vectors $\mathbf{u} = \big(u_1, \ldots, u_d\big)^T$ along with the inner product $\mathbf{u}^T\mathbf{v}$.

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

$$L\left((\mathbf{x}, y), h\right) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{2}$$

A regularized variant of the hinge loss is used by the support vector machine (SVM) [34] to learn a linear classifier with maximum margin between the two classes (see Figure 1).

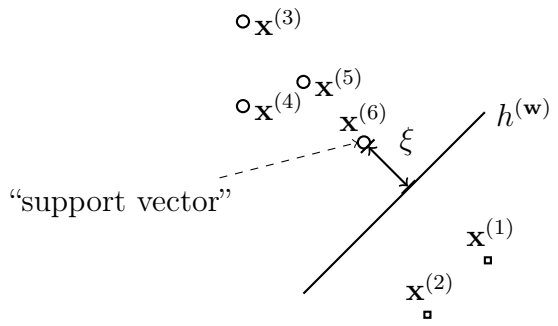

Figure 1: The SVM learns a hypothesis (or classifier) $h^{(\mathbf{w})}$ with minimum average soft-margin hinge loss. Minimizing this loss is equivalent to maximizing the margin ξ between the decision boundary of $h^{(\mathbf{w})}$ and each class of the training set.

histogram Consider a dataset $\mathcal{D}$ that consists of m data points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ that belong to some cell $[-U, U] \times \ldots \times [-U, U] \subseteq \mathbb{R}^d$ with side length

U. We partition this cell evenly into smaller elementary cells with side length Δ. The histogram of $\mathcal{D}$ assigns each elementary cell to the corresponding fraction of data points in $\mathcal{D}$ that fall into this elementary cell.

horizontal FL Horizontal FL refers to applications with local datasets that are constituted by different data points but using the same features to characterize them [**?**]. One example for horizontal FL is numerical weather prediction using a network of weather (observation) stations. Each weather station measures the same quantities such as daily temperature, air pressure and precipitation. However, different weather stations measure the characteristics or features of different spatio-temporal regions (each such region being a separate data point).

Huber loss The Huber loss is a mixture of the squared error loss and the absolute value of the prediction error.

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.

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 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.

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.

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

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 [21].

kernel Consider data points characterized by features $\mathbf{x} \in \mathcal{X}$ with values in some arbirtrary feature space. A kernel is a map that assigns each pair of feature values $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ a real number. This number measures the similarity between $\mathbf{x}$ and $\mathbf{x}'$. For more details about kernels and the resulting kernel methods, we refer to the literature [34, 35].

Kullback-Leibler divergence The Kullback–Leibler divergence is a quantitive measure for how much one probability distribution is different from another probability distribution [9].

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" [24–26].

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"$\}$

labeled datapoint A data point whose label is known or has been determined by some means (might require human experts).

Laplacian matrix The geometry or structure of a graph $\mathcal{G}$ can be analyzed using the properties of special matrices that are associated with $\mathcal{G}$. One such matrix is the graph Laplacian matrix $\mathbf{L}$ which is defined for an undirected and weighted graph (e.g., the empirical graph of networked data) [36, 37].

law of large numbers The law of large numbers refers to the convergence of the average of an increasing (large) number of i.i.d. RVs to the mean (or expectation) of their common probability distribution. Different instances of the law of large numbers are obtained using different notions of convergence.

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

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.

least absolute deviation regression Least absolute deviation regression uses the average of the absolute precondition errors to find a linear hypothesis.

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 $\mathbf{w}$.

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 $\mathbb{R}^d$.

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

$$\mathcal{H}^{(d)} := \left\{ h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\}. \tag{3}$$

Note that (3) 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).

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).

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.

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 $i, i' \in \mathcal{V}$.

logistic loss Consider a data point that is characterized by the features $\mathbf{x}$ and a binary label $y \in \{-1, 1\}$. We use a real-valued hypothesis h to predict the label y solely from the features $\mathbf{x}$. The logistic loss incurred by a specific hypothesis h is defined as

$$L\left((\mathbf{x}, y), h\right) := \log(1 + \exp(-yh(\mathbf{x}))). \tag{4}$$

. Carefully note that the expression (4) for the logistic loss applies only for the label space $\mathcal{Y} = \{-1, 1\}$ and using the thresholding rule (1).

logistic regression Logistic regression learns a linear hypothesis map (classifier) $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ to predict a binary label y based on numeric features $\mathbf{x}$ of a data point. The quality of a linear hypothesis map is measured using its average logistic loss on some labeled data points (the training set).

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.

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 $h \in \mathcal{H}$.

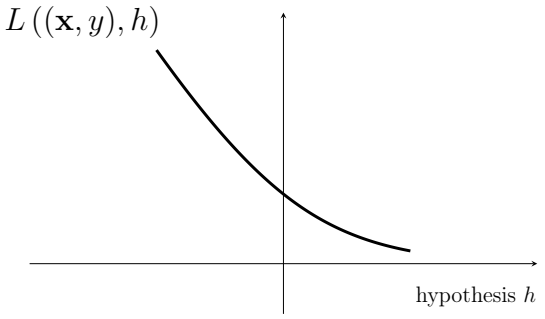

Figure 2: Some loss function $L\left((\mathbf{x}, y), h\right)$ for a fixed data point, with feature vector $\mathbf{x}$ and label y, and varying hypothesis h. ML methods try to find (learn) a hypothesis that incurs minimum loss.

maximum Given a set of real numbers, the maximum is the largest of those numbers.

maximum likelihood Consider data points $\mathcal{D} = \big\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big\}$ that are interpreted as realizations of i.i.d. RVs with a common probability

distribution $p(\mathbf{z};\mathbf{w})$ which depends on a parameter vector $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Maximum likelihood methods aim at finding a parameter vector $\mathbf{w}$ such that the probability (density) $p(\mathcal{D};\mathbf{w}) = \prod_{r=1}^{m} p(\mathbf{z}^{(r)};\mathbf{w})$ of observing the data is maximized. Thus, the maximum likelihood estimator is obtained as a solution to the optimization problem $\max_{\mathbf{w}\in\mathcal{W}} p(\mathcal{D};\mathbf{w})$.

mean The expectation $\mathbb{E}\{\mathbf{x}\}$ of a numeric RV $\mathbf{x}$.

mean squared estimation error Consider a ML method that uses a parametrized hypothesis space. For a given training set, whose data points are interpreted as realizations of RVs, the ML method learns the parameters incurring the estimation error $\Delta\mathbf{w}$. The mean squared estimation error is defined as the expectation $\mathbb{E}\{\|\Delta\mathbf{w}\|^2\}$ of the squared Euclidean norm of the estimation error.

metric A metric refers to a loss function that is used solely for the final performance evaluation of a learnt hypothesis. The metric is typically a loss function that has a "natural" interpretation (such as the 0/1 loss) but is not a good choice to guide the learning process, e.g., via ERM. For ERM, we typically prefer loss functions that depend smoothly on the (parameters of the) hypothesis. Examples for such smooth loss functions include the squared error loss (5) and the logistic loss (4).

minimum Given a set of real numbers, the minimum is the smallest of those numbers.

missing data By missing data, we refer to a situation where some feature values of a subset of data points are unknown. Data imputation techniaues

aim at estimating (predicting) these missing feature values [38].

model We use the term model as a synonym for hypothesis space

model parameters Model parameters are numbers that select a hypothesis map out of a hypothesis space.

multi-label classification Multi-label classification problems and methods involve data points that are characterized by several individual labels.

multivariate normal distribution The multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ is an important family ofprobability distributions for a continuous RV $\mathbf{x} \in \mathbb{R}^d$ [3, 30, 39]. 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

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

mututal information The mutual information $I(\mathbf{x}; y)$ between two RVs $\mathbf{x}$, y defined on the same probability space is given by

$$I(\mathbf{x}; y) := \mathbb{E}\left\{\log \frac{p(\mathbf{x}, y)}{p(\mathbf{x})p(y)}\right\}.$$

It is a measure for how well we can estimate y based solely from $\mathbf{x}$. A large value of $I(\mathbf{x}; y)$ means that y can be well predicted solely from $\mathbf{x}$. The prediction could be obtained by a hypothesis learnt by a ML method.

nearest neighbour Nearest neighbour methods learn a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ whose function value $h(\mathbf{x})$ is solely determined by the nearest neighbours in the feature space $\mathcal{X}$

neibbours The neighbours of a node $i \in \mathcal{V}$ within an empirical graph are those nodes $i' \in \mathcal{V}$ that are connected by an edge with i.

neighbourhood The neighbourhood of a node $i \in \mathcal{V}$ within an empirical graph are those nodes $i' \in \mathcal{V} \setminus \{i\}$ that are connected to i by an edge with i.

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.

networked exponential families A networked collection of exponential families having a separaate parameter vector for each node of the network. These parameter vectors are coupled via the network structure.

networked federated learning Networked federated learning refers to methods that learn personalized models in a distributed fashion from local datasets that are related by an intrinsic network structure.

networked model A networked model over an empirical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ assigns a local model (hypothesis space) to each node $i \in \mathcal{V}$ of the empirical graph $\mathcal{G}$.

node degree The degree of a node i in an undirected empirical graph is the number of its neighbours.

non i.i.d. data A dataset consisting of data points that cannot be well modelled as realizations of i.i.d. RVs.

non-i.i.d. See non-i.i.d. data.

non-smooth We refer to a function as non-smooth if it is not smooth [40].

norm A norm is a function that maps each element (vector) of a linear vectorspace to a non-negative number. This function must be homogeneous, definite and satisfy the triangle inequality.

objective function An objective function is a map that assigns each possible value of an optimization variable, such as the parameters $\mathbf{w}$ of a hypothesis $h^{(\mathbf{w})}$, to an objective value $f(\mathbf{w})$. The objective value $f(\mathbf{w})$ could by the the risk or the empirical risk of a hypothesis $h^{(\mathbf{w})}$.

outlier Many ML methods are motivated by the i.i.d. assumption which interprets data points as realizations of i.i.d. RVs with a common probability distribution. The i.i.d. assumption is useful for applications where the statistical properties of the data generation process are stationary (time-invariant). However, in some applications the data consists of a majority of "regular" data points that conform with an i.i.d. assumption and a small number of data points that have fundamentally different statistical properties compared to the regular data points. We refer to a data point that substantially deviates from the statistical properties of the majority of data points as an outlier. Different methods for outlier detection use different measures for this deviation.

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.

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.

polynomial regression Polynomial regression aims at learning a polynomial hypothesis map to predict a numeric label based on numeric features of a data point. For data points characterized by a single numeric features, polynomial regression uses the hypothesis space $\mathcal{H}_d^{(\text{poly})} := \{h(x) = \sum_{j=0}^{d-1} x^j w_j\}$. The quality of a polynomial 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).

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 $\mathbf{x} \in \mathbb{R}^d$.

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.

predictor A predictor is a hypothesis whose function values are numeric, such as real numbers. Given a data point with features $\mathbf{x}$, the predictor

value $h(\mathbf{x}) \in \mathbb{R}$ is used as a prediction (estimate/guess/approximation) for the true numeric label $y \in \mathbb{R}$ of the data point.

principal component analysis (PCA) Principal component analysis determines a given number of new features that are obtained by a linear transformation (map) of the raw features.

privacy leakage Consider a (ML or FL) system that processes a local dataset $\mathcal{D}^{(i)}$ and shares data, such as the predictions obtained for new data points, with other parties. Privacy leakage arises if the shared data carries information about a private (sensitive) feature of a data point (which might be a human) of $\mathcal{D}^{(i)}$. The amount of privacy leakage can be measured via mutual information using a probabilistic model for the local dataset.

privacy protection Privacy protection aims at avoiding (or minimizing) the privacy leakage occuring within data processing systems (such as ML or FL methods).

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 [6].

probabilistic PCA Probabilistic principal component analysis (PCA) (PPCA) extends basic PCA by using a probabilistic model for data points. The probabilistic model of PPCA reduces the task of dimensionality reduction to an estimation problem that can be solved using expectation

maximization (EM) methods.

probability We assign a probability value, typically chosen in the interval $[0,1]$, to each event that might occur in a random experiment [3, 14, 15, 41].

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 $p(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [3, 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 $p(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [3, Ch. 3].

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 [15, 30].

projected GD Projected GD extends basic GD for unconstrained opti-

mization to handle constraints on the optimization variable (model parameters). A single iteration of projected GD consists of first taking a gradient step and then projecting the result back into a constrain set.

proximable A Convex function for which the proximity operator can be computed efficiently are sometimes referred to as "proximable" or "simple" [42].

proximal operator Given a convex function and a vector $\mathbf{x}$, we define its proximity operator as

$$\mathbf{prox}_{L_i(\cdot),2\alpha}(\mathbf{w}'') := \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} f(\mathbf{w}) + (\rho/2)\left\|\mathbf{w}-\mathbf{w}'\right\|_2^2 \text{ with } \rho > 0.$$

Convex functions for which the proximity operator can be computed efficiently are sometimes referred to as "proximable" or "simple" [42].

quadratic function A quadratic function $f(\mathbf{w})$, reading in a vector $\mathbf{w} \in \mathbb{R}^d$ as its argument, is such that

$$f(\mathbf{w}) = \mathbf{w}^T\mathbf{Q}\mathbf{w} + \mathbf{q}^T\mathbf{w} + a,$$

with some matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$, vector $\mathbf{q} \in \mathbb{R}^d$ and scalar $a \in \mathbb{R}$.

random forest A random forest is a set (ensemble) of different decision trees. Each of these decision trees is obtained by fitting a perturbed copy of the original dataset.

random variable (RV) A random variable is a mapping from a probability space $\mathcal{P}$ to a value space [15]. 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 [15, 30].

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 $\mathcal{N}$ [5, 14, 15]. A realization of x is any element $a' \in \mathcal{N}$ such that there is an element $\omega' \in \mathcal{P}$ with $x(\omega') = a'$.

rectified linear unig (ReLU) The rectified linear unit or "ReLU" is a popular choice for the activation function of a neuron within an ANN. It is defined as $g(z) = \max\{0, z\}$ with z being the weighted input of the neuron.

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

regret The regret of a hypothesis h relative to another hypothesis h', which serves as a reference (or baseline), is the difference between the loss incurred by h and the loss incurred by h' [19]. The baseline hypothesis h' is also referred to as an expert.

regularization Regularization techniques modify the ERM principle such that the learnt hypothesis performs well (generalizes) beyond the train-

ing 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.

regularized empirical risk minimization Synoym for SRM.

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 $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [4, 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 $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [4, Ch. 3].

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 $\alpha\|\mathbf{w}\|_2^2$ with a regularization parameter $\alpha > 0$. It can be shown that the effect of adding to $\alpha\|\mathbf{w}\|_2^2$ 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.

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.

sample A finite sequence (list) of data points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(r)}$ that is obtained or interpreted as the realizations of m i.i.d. RVs with the common probability distribution $p(\mathbf{z})$. The length m of the sequence is referred to as the sample size.

sample covariance matrix The sample covariance matrix $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{d \times d}$ for a given set of feature vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ is defined as

$$\widehat{\mathbf{\Sigma}} = (1/m) \sum_{r=1}^{m} (\mathbf{x}^{(r)} - \widehat{\mathbf{m}})(\mathbf{x}^{(r)} - \widehat{\mathbf{m}})^T.$$

Here, we used the sample mean $\widehat{\mathbf{m}}$.

sample mean The sample mean $\mathbf{m} \in \mathbb{R}^{d \times d}$ for a given set of feature vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ is defined as

$$\mathbf{m} = (1/m) \sum_{r=1}^{m} \mathbf{x}^{(r)}.$$

sample size The number of individual data points contained in a dataset that is obtained from realizations of i.i.d. RVs.

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

semi-supervised learning Semi-supervised learning methods use (large amounts of) unlabeled data points to support the learning of a hypothesis from (a small number of) labeled data points [7].

similarity graph Some applications generate data points that are related by a domain-specific notion of similarity. These similarities can be represented conveniently using a similarity graph $\mathcal{G} = \big(\mathcal{V} := \{1, \ldots, m\}, \mathcal{E}\big)$. The node $r \in \mathcal{V}$ represents the r-th data point. Two nodes are connected by an undirected edge if the corresponding data points are similar.

smooth We refer to a real-valued function as smooth if it is differentiable and its gradient is continuous [40, 43]. 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.,

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

soft clustering Soft clustering refers to the task of partitioning a given set of data points into (few) overlapping clusters. Each data point is assigned to several different clusters with varying degree of belonging. Soft clustering amounts to determining such a degree of belonging (or soft cluster assignment) for each data point and each cluster.

spectogram The spectogram of a time signal, e.g., an audio recording, characterizes the time-frequency distribution of the signal. Loosely speaking, the spectogram quantities the signal strength at a specific time and frequency.

spectral clustering Spectral clustering groups the nodes of an undirected graph by applying k-means clustering to node-wise feature vectors. These feature vectors are built from the eigenvectors of the graph Laplacian matrix [28, 36].

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

$$L\left((\mathbf{x}, y), h\right) := \big(y - \underbrace{h(\mathbf{x})}_{=\hat{y}}\big)^2. \tag{5}$$

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.

step size Many ML methods use iterative optimization methods (such as gradient-based methods) to construct a sequence of increasingly accurate hypothesis maps $h^{(1)}, h^{(2)}, \ldots$. The kth iteration of such an algorithm starts from the current hypothesis $h^{(k)}$ and tries to modify it to obtain an improved hypothesis $h^{(k+1)}$. Iterative algorithms often use a step size (hyper-) parameter. The step size controls the amount by which a single iteration can change or modify the current hypothesis. Since the overall goal of such iteration ML methods is to learn a (approximately) optimal hypothesis we refer to a step size parameter also as a learning rate.

stochastic block model The stochastic block model (SBM) is a (family) probabilistic generative model for an undirected graph $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}\big)$

with given nodeset $\mathcal{V}$. In its most basic forms, the SBM amounts to generating a graph by first randomly assigning labels $c_i \in \{1, \ldots, k\}$ to each node $i \in \mathcal{V}$ of the graph. A pair of different nodes in the graph is connected by edge with probabailiy $p_{i,j}$ that depends solely on the labels c_i, c_j.

stochastic gradient descent Stochastic GD is obtained from GD by replacing the gradient of the objective function with a noisy (or stochastic) estimate.

stopping criterion Many ML methods use iterative algorithms that construct a sequence of model parameters (such as the weights of a linear map or the weights of an ANN) that (hopefully) converge to an optimal choice for the model parameters. In practice, given finite computational resources, we need to stop iterating after a finite number of times. A stopping criterion is any well-defined condition required for stopping iterating.

strongly convex A continuously differentiable real-valued function $f(\mathbf{x})$ is strongly convex with coefficient σ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + (\sigma/2)\left\|\mathbf{y} - \mathbf{x}\right\|_2^2$ [40], [44, Sec. B.1.1.].

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.

subgradient For a real-valued function $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, a vector $\mathbf{a}$ such that $f(\mathbf{w}) \geq f(\mathbf{w}') + \big(\mathbf{w} - \mathbf{w}'\big)^T \mathbf{a}$ is referred to as a subgradient of f at $\mathbf{w}'$ [45, 46].

subgradient descent Subgradient descent is a generalization of GD that does not require differentiablity of the function to be minimized. This generalization is obtained by replacing the concept of a gradient with that of a sub-gradient. Similar to gradients, also sub-gradients allow to construct local approximations of an objective function. The objective function might be the empirical risk $\widehat{L}\big(h^{(\mathbf{w})}\big|\mathcal{D}\big)$ viewed as a function of the model parameters $\mathbf{w}$ that select a hypothesis $h^{(\mathbf{w})} \in \mathcal{H}$.

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 (2).

test set A set of data points that have neither been used in a training set to learn parameters of a model nor in a validation set to choose between different models (by comparing validation errors).

total variation Consider some empirical graph whose nodes carry (personalized) local model parameters. The total variation measures the discrepancies between across edges $\{i, i'\}$.

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.

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).

trustworthiness Beside the computational aspects and statistical aspects, a third main design aspect for ML methods is their trustworthiness. The European Union has put forward seven key requirements for trustworthy AI systems (that typically build on ML methods) [47]: **human agency and oversight**, **technical robustness and safety**, **privacy and data governance**, **transparency**, **diversity non-discrimination and fairness**, **societal and environmental well-being**, **accountability**.

underfitting Consider a ML method that uses ERM to learn a hypothesis with minimum empirical risk on a given training set. Such a method is "underfitting" the training set if it is not able to learn a hypothesis with sufficiently small empirical risk on the training set. If a method is underfitting it will typically also not be able to learn a hypothesis with a small risk.

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 $\mathcal{D}$.

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.

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 [4, 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).

Vapnik–Chervonenkis (VC) dimension The VC dimension of an infinite hypothesis space is a widely-used measure for its size. We refer to [48] for a precise definition of VC dimension as well as a discussion of its basic properties and use in ML.

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 $\mathbb{E}\big\{\big\|\mathbf{x}-\mathbb{E}\{\mathbf{x}\}\big\|_2^2\big\}$.

vertical FL Vertical FL refers to applications with local datasets that are constituted by the same data points but characterizing them with different features [49]. For example, different healthcare providers might all contain information about the same population of patients. However, different healthcare providers collect different measurements (blood values, electrocardiography, lung x-ray) for the same patients.

weights We use the term weights synonymously for a finite set of parameters within a model. For example, the linear model consists of all linear maps $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ that read in a feature vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ of a data point. Each specific linear map is characterized by specific choices for the parameters for weights $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$.

zero-gradient condition Consider the unconstrained optimization problem $\min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w})$ with a smooth and convex objective function $f(\mathbf{w})$. A necessary and sufficient condition for a vector $\widehat{\mathbf{w}} \in \mathbb{R}^d$ to solve this problem is that the gradient $\nabla f\big(\widehat{\mathbf{w}}\big)$ is the zero-vector,

$$\nabla f\big(\widehat{\mathbf{w}}\big) = \mathbf{0} \Leftrightarrow f\big(\widehat{\mathbf{w}}\big) = \min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w}).$$

References

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

[2] G. Golub and C. van Loan, "An analysis of the total least squares problem," *SIAM J. Numerical Analysis*, vol. 17, no. 6, pp. 883–893, Dec. 1980.

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

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

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

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

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

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

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

[10] 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.

[11] 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.

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

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

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

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

[16] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer, 2001.

[18] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, ser. Foundations and Trends in Machine Learning. Hanover, MA: Now Publishers, 2008, vol. 1, no. 1–2.

[19] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games.* New York, NY, USA: Cambridge University Press, 2006.

[20] E. Hazan, *Introduction to Online Convex Optimization.* Now Publishers Inc., 2016.

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

[22] A. Jung, "Explainable empiricial risk minimization," *submitted to IEEE Sig. Proc. Letters (preprint: https://arxiv.org/pdf/2009.01492.pdf)*, 2020.

[23] J. Chen, L. Song, M. Wainwright, and M. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *Proc. 35th Int. Conf. on Mach. Learning*, Stockholm, Sweden, 2018.

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

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

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

[27] 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. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[28] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Flow-based clustering and spectral clustering: A comparison," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 1292–1296.

[29] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York: Mc-Graw Hill, 2002.

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

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

[32] M. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint.* Cambridge: Cambridge University Press, 2019.

[33] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data.* New York: Springer, 2011.

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

[35] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, Dec. 2002.

[36] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[37] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neur. Inf. Proc. Syst.*, 2001.

[38] K. Abayomi, A. Gelman, and M. A. Levy, "Diagnostics for multivariate imputations," *Journal of The Royal Statistical Society Series C-applied Statistics*, vol. 57, pp. 273–291, 2008.

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

[40] 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

[41] O. Kallenberg, *Foundations of modern probability.* New York: Springer, 1997.

[42] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Opt. Th. and App.*, vol. 158, no. 2, pp. 460–479, Aug. 2013.

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

[44] D. P. Bertsekas, *Convex Optimization Algorithms.* Athena Scientific, 2015.

[45] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization.* Athena Scientific, 2003.

[46] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, June 1999.

[47] E. Commission, C. Directorate-General for Communications Networks, and Technology, *The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self assessment.* Publications Office, 2020.

[48] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning – from Theory to Algorithms.* Cambridge University Press, 2014.

[49] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, *Vertical Federated Learning.* Cham: Springer International Publishing, 2020, pp. 69–81. [Online]. Available: https://doi.org/10.1007/978-3-031-01585-4_5