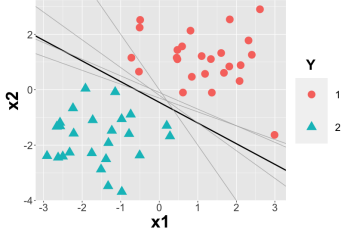
Introduction to Machine Learning

Slides created by Dr Rong Qu, Dr John Drake and Dr Huan Jin

Introduction to Machine Learning

ML-Basics: In a Nutshell



Learning goals

- Understand fundamental goal of supervised machine learning
- Know concepts of task, model, parameter, learner, loss function.

WHAT IS ML?

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."
Tom Mitchell, Carnegie Mellon University, 1998

$\Rightarrow$ 99 % of this lecture is about **supervised learning**:

Training

Prediction

Introduction to Machine Learning

ML-Basics: What is Machine Learning?

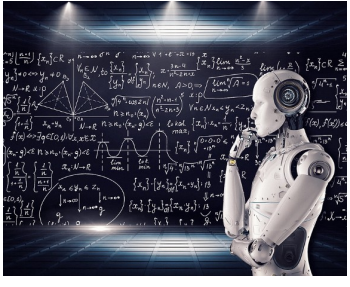


Image via www.vpnsrus.com

Learning goals

Understand basic terminology of and connections between ML, AI, DL

Know the main directions of ML: Supervised, Unsupervised and Reinforcement Learning

MACHINE LEARNING IS CHANGING OUR WORLD

Search engines learn what you want

Recommender systems learn your taste in books, music, movies,...

Algorithms do automatic stock trading

Google Translate learns how to translate text

Siri learns to understand speech

DeepMind beats humans at Go

Cars drive themselves

Smart-watches monitor your health

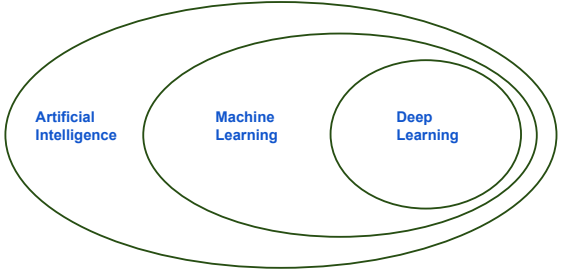
Election campaigns use algorithmically targeted ads to influence voters

Data-driven discoveries are made in physics, biology, genetics, astronomy, chemistry, neurology,...

...

THE WORLD OF ARTIFICIAL INTELLIGENCE

... and the connections to Machine Learning and Deep Learning



Many people are confused what these terms actually mean.

And what does all this have to do with statistics?

ARTIFICIAL INTELLIGENCE

AI is a general term for a very large and rapidly developing field.

There is no strict definition of AI, but it’s often used when machines are trained to perform on tasks which until that time could only be solved by humans or are very difficult and assumed to require "intelligence".

AI includes machine learning, natural language processing, computer vision, robotics, planning, search, game playing, intelligent agents, and much more.

MACHINE LEARNING

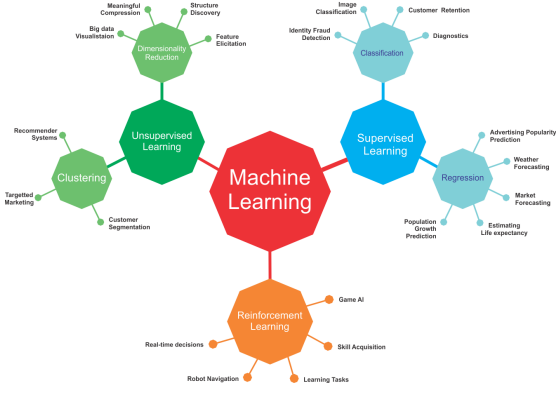


Image via `https://www.oreilly.com/library/view/java-deep-learning/9781788997454/assets/899ceaf3-c710-4675-ae99-33c76cd6ac2f.png`

Mathematically well-defined and solves reasonably narrow tasks.

ML algorithms usually construct predictive/decision models from data, instead of explicitly programming them.

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
Tom Mitchell, Carnegie Mellon University, 1998

DEEP LEARNING

DL is a subfield of ML which studies neural networks.

Artificial neural networks (ANNs) might have been (roughly) inspired by the human brain, but they are simply a certain model class of ML.

ANNs have been studied for decades. DL uses more layers, specific neurons were invented for images and tensors and many computational improvements allow training on large data.

DL can be used on tabular data, but typical applications are images, texts or signals.

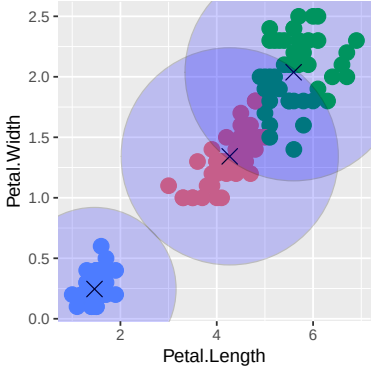
The last 10-15 years have produced remarkable results and imitations of human ability, where the result looked intelligent.

UNSUPERVISED LEARNING

Data without labels *y*

Search for patterns within the inputs *x*

Unsupervised as there is no “true” output we can optimize against



Dimensionality reduction (PCA, Autoencoders ...);
compress information in $\mathcal{X}$

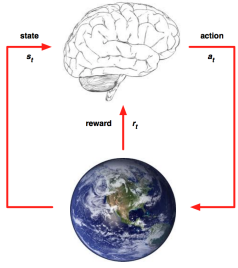
Clustering: group similar observations

Outlier detection, anomaly detection

Association rules

REINFORCEMENT LEARNING

RL is a general-purpose framework for AI. At each time step an *agent* interacts with *environment*. It: observes state; receives reward; executes action.

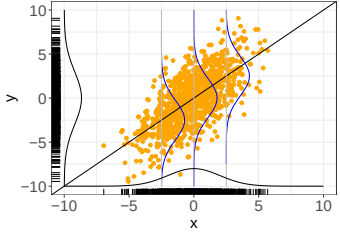


Goal: Select actions to maximize future reward.

Reward signals may be sparse, noisy and delayed.

Introduction to Machine Learning

ML-Basics: Data



Learning goals

- Understand structure of tabular data in ML
- Understand difference between target and features
- Understand difference between labeled and unlabeled data
- Know concept of data-generating process

IRIS DATA SET

Introduced by the statistician Ronald Fisher and one of the most frequently used toy examples.

- Classify iris subspecies based on flower measurements.
- 150 iris flowers: 50 versicolor, 50 virginica, 50 setosa.
- Sepal length / width and petal length / width in [cm].

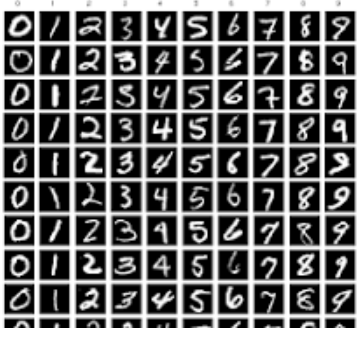
Source: `https://rpubs.com/vidhividhi/irisdataeda`

Word of warning: "iris" is a small, clean, low-dimensional data set, which is very easy to classify; this is not necessarily true in the wild.

MINST DATA SET

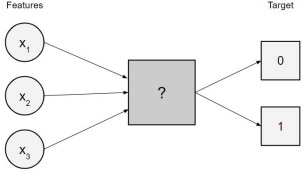
Images of handwritten digit

- Each image contains 28*28 =784 pixels.
- Each image has a label of a digit between 0 and 9.
- 55000 images in the dataset.



DATA IN SUPERVISED LEARNING

- The data we deal with in supervised learning usually consists of observations on different aspects of objects:
  - **Target**: the output variable / goal of prediction
  - **Features**: measurable properties that provide a concise description of the object
- We assume some kind of relationship between the features and the target, in a sense that the value of the target variable can be explained by a combination of the features.



Features x				Target y
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.3	3.0	1.1	0.1	setosa
5.0	3.3	1.4	0.2	setosa
7.7	3.8	6.7	2.2	virginica
5.5	2.5	4.0	1.3	versicolor

ATTRIBUTE TYPES

- Both features and target variables may be of different data types
  - **Numerical** variables can have values in $\mathbb{R}$
  - **Integer** variables can have values in $\mathbb{Z}$
  - **Categorical** variables can have values in $\{C_1, ..., C_g\}$
  - **Binary** variables can have values in $\{0, 1\}$
- For the **target** variable, this results in different tasks of supervised learning: *regression* and *classification*.
- Most learning algorithms can only deal with numerical features, although there are some exceptions (e.g., decision trees can use integers and categoricals without problems). For other feature types, we usually have to pick or create an appropriate **encoding**, i.e., cast them to numerical values.
- If not stated otherwise, we assume numerical features.

ENCODING FOR CATEGORICAL FEATURES

- We expand the representation of a feature x with k mutually exclusive categories from a scalar to a length-$\tilde{k}$ vector with at most one element being 1, and 0 otherwise: $\boldsymbol{o}(x) = [\mathbb{I}(x = j)]_{j=1,2,\ldots,\tilde{k}} \in \{0, 1\}^{\tilde{k}}$.
- Each entry of $\boldsymbol{o}(x)$ is treated as a separate feature.
- Two popular ways to do this are
  - **One-hot encoding**: $\tilde{k} = k$ dummies, so *exactly one* element is 1 ("hot"). E.g., $x \in \{a, b, c\} \mapsto \boldsymbol{o}(x) = (x_a, x_b, x_c)$, with $x_a = x_b = 0, x_c = 1$ and $\boldsymbol{o}(x) = (0, 0, 1)$ for $x = c$.
  - **Dummy encoding**: $\tilde{k} = k - 1$ dummies, so *at most one* element is 1, cutting the redundancy of one-hot encoding (necessary for learners that require non-singular input matrices, such as in linear regression). E.g., $x \in \{a, b, c\} \mapsto \boldsymbol{o}(x) = (x_a, x_b)$ for reference category c, with $x_a = x_b = 0$ and $\boldsymbol{o}(x) = (0, 0)$ for $x = c$.
- For features with a natural **order** in their categories we resort to encodings that reflect this ordinality, e.g., a sequence of integer values.

OBSERVATION LABELS

- We call the entries of the target column **labels**.
- We distinguish two basic forms our data may come in:
  - For **labeled** data we have already observed the target
  - For **unlabeled** data the target labels are unknown

	Features x				Target y
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
labeled data	4.3	3.0	1.1	0.1	setosa
	5.0	3.3	1.4	0.2	setosa
	7.7	3.8	6.7	2.2	virginica
	5.5	2.5	4.0	1.3	versicolor
unlabeled data	5.9	3.0	5.1	1.8	?
	4.4	3.2	1.3	0.2	?

NOTATION FOR DATA

In formal notation, the data sets we are given are of the following form:

$$\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)} \right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)} \right) \right) \in (\mathcal{X} \times \mathcal{Y})^n.$$

We call

- $\mathcal{X}$ the input space with $p = \dim(\mathcal{X})$ (for now: $\mathcal{X} \subset \mathbb{R}^p$),
- $\mathcal{Y}$ the output / target space,
- the tuple $\left(\mathbf{x}^{(i)}, y^{(i)} \right) \in \mathcal{X} \times \mathcal{Y}$ the i-th observation,
- $\mathbf{x}_j = \left(x_j^{(1)}, \ldots, x_j^{(n)} \right)^\top$ the j-th feature vector.

We denote

- $(\mathcal{X} \times \mathcal{Y})^n$, i.e., the set of all data sets of size n, as $\mathbb{D}_n$,
- $\bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n$, i.e., the set of all finite data sets, as $\mathbb{D}$.

So we have observed n objects, described by p features.

DATA-GENERATING PROCESS

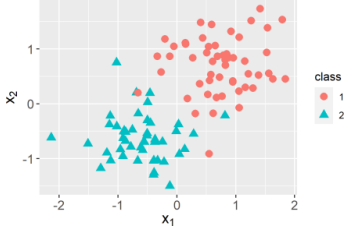
- We assume the observed data $\mathcal{D}$ to be generated by a process that can be characterized by some probability distribution

$$\mathbb{P}_{xy},$$

  defined on $\mathcal{X} \times \mathcal{Y}$.
- We denote the random variables following this distribution by lowercase **x** and y.
- It is important to understand that the true distribution is essentially **unknown** to us. In a certain sense, learning (part of) its structure is what ML is all about.

Introduction to Machine Learning

ML-Basics: Supervised and Unsupervised Tasks



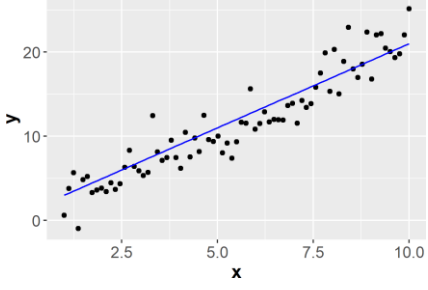
Learning goals

- Know the difference between supervised and unsupervised learning tasks;
- Know definition and examples of supervised tasks
- Know definition and examples of unsupervised tasks

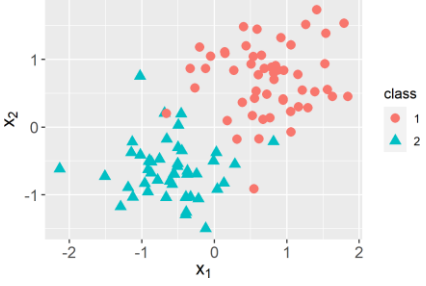
TASKS: Supervised learning tasks

- Supervised tasks are data situations where learning the functional relationship between inputs (features) and output (target) is useful.
- The two most basic tasks are regression and classification, depending on whether the target is numerical or categorical.

Regression: Our observed labels come from $\mathsf{Y} \subseteq \mathsf{R}$.



Classification: Observations are categorized: $y \in \mathsf{Y} = \{C_1, \ldots, C_g\}$.



REGRESSION EXAMPLE: HOUSE PRICES

Predict the price for a house in a certain area

Features x				Target y
square footage of the house	number of bedrooms	swimming pool (yes/no)	...	house price in US$
1,180	3	0	...	221,900
2,570	3	1	...	538,000
770	2	0	...	180,000
1,960	4	1	...	604,000

REGRESSION EXAMPLE: LENGTH-OF-STAY

Predict days a patient has to stay in hospital at time of admission

Features x					Target y
diagnosis category	**admission type**	**gender**	**age**	**...**	**Length-of-stay in the hospital in days**
heart disease	elective	male	75	...	4.6
injury	emergency	male	22	...	2.6
psychosis	newborn	female	0	...	8
pneumonia	urgent	female	67	...	5.5

CLASSIFICATION EXAMPLE: RISK CATEGORY

Predict one of five risk categories for a life insurance customer to determine the insurance premium

Features x				Target y
job type	age	smoker	...	risk group
carpenter	34	1	...	3
stuntman	25	0	...	5
student	23	0	...	1
white-collar worker	39	0	...	2

Unsupervised learning tasks

Unsupervised learning: given only samples x of the data, infers a function f such that y = f(x) describes the hidden structure of the unlabeled data - more of an exploratory/descriptive data analysis

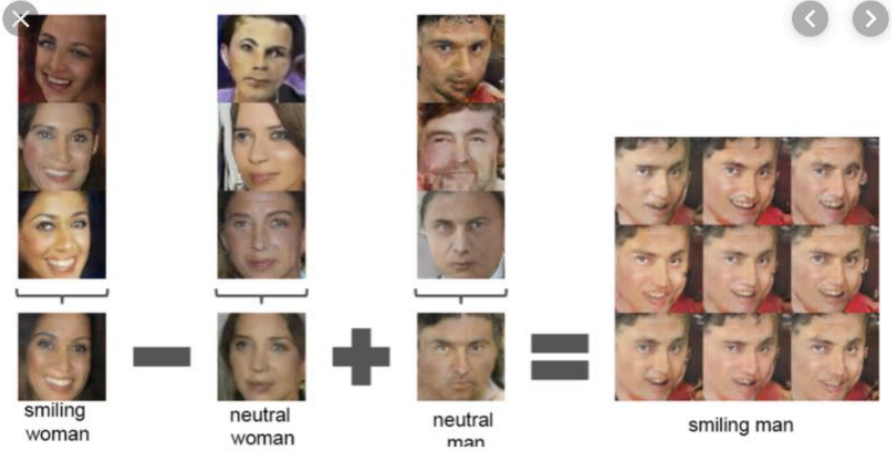
- Clustering: Learn any intrinsic structure hidden in the data
- Dimensional Reduction: y is continuous. Discover a lower- dimensional surface on which the data lives
- Anomaly and novelty detection: Find some unusual and interesting datapoints.

SUPERVIDED VS. UNSUPERVISED

Supervised	Un-supervised
y = F(x): function	**y = ?**: no function
D: labeled training set	**D**: unlabeled data set
Learn: G(x): model trained to predict labels of new cases	**Learn:** ?
Goal: $E[(F(x)-G(x))^2] \approx 0$	**Goal:** ?

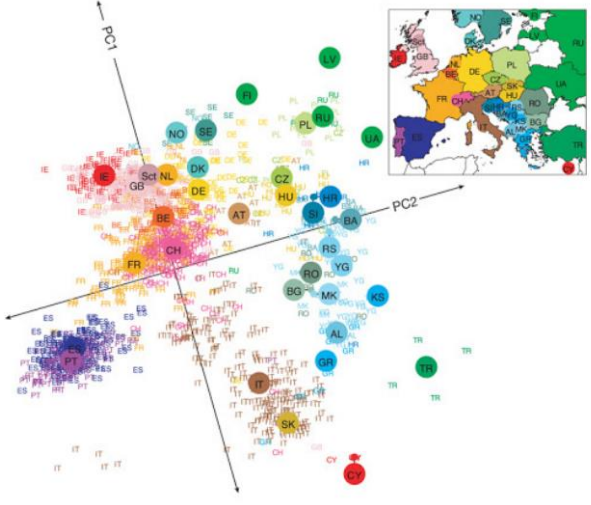
Unsupervised EXAMPLE: Human Faces

Modern unsupervised algorithm based on deep learning uncover structure in human face datasets.



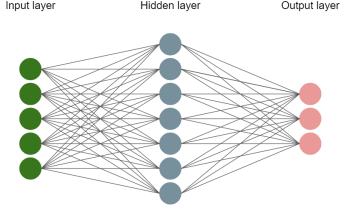
Unsupervised EXAMPLE: DNA analysis

Dimensionality reduction applied to DNA reveal the geography of European countries.



Introduction to Machine Learning

ML-Basics: Models & Parameters



Learning goals

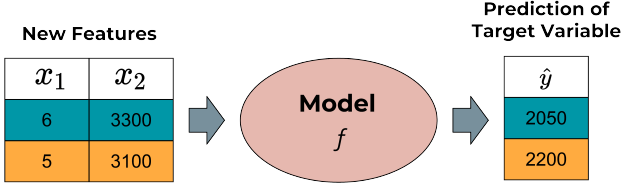
- Understand that an ML model is simply a parametrized curve
- Understand that the hypothesis space lists all admissible models for a learner
- Understand the relationship between the hypothesis space and the parameter space

WHAT IS A MODEL?

- A **model** (or **hypothesis**)

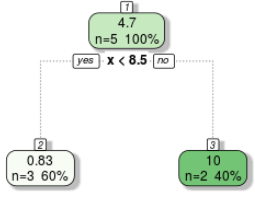
$$f : \mathcal{X} \to \mathbb{R}^g$$

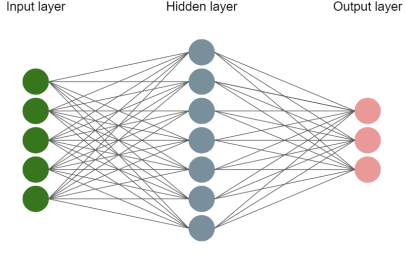
  is a function that maps feature vectors to predicted target values.
- In conventional regression: $g = 1$; for classification g is the number of classes, and output vectors are scores or class probabilities (details later).



WHAT IS A MODEL? / 2

- f is meant to capture intrinsic patterns of the data, the underlying assumption being that these hold true for *all* data drawn from $\mathbb{P}_{xy}$.
- It is easily conceivable how models can range from super simple (e.g., linear, tree stumps) to very complex (e.g., deep neural networks) and there are infinitely many choices how we can construct such functions.





- In fact, ML requires **constraining** f to a certain type of functions.

HYPOTHESIS SPACES

- Without restrictions on the functional family, the task of finding a "good" model among all the available ones is impossible to solve.
- This means: we have to determine the class of our model *a priori*, thereby narrowing down our options considerably. We could call that a **structural prior**.
- The set of functions defining a specific model class is called a **hypothesis space** $\mathcal{H}$:

$$\mathcal{H} = \{f : f \text{ belongs to a certain functional family}\}$$

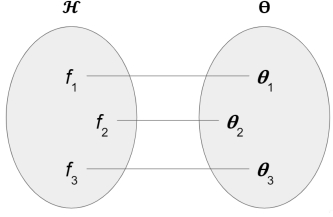
PARAMETRIZATION

- All models within one hypothesis space share a common functional structure. We usually construct the space as **parametrized family of curves**.
- We collect all parameters in a **parameter vector** $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)$ from **parameter space** Θ.
- They are our means of fixing a specific function from the family. Once set, our model is fully determined.
- Therefore, we can re-write $\mathcal{H}$ as:

$$\mathcal{H} = \{f_{\boldsymbol{\theta}} : \quad f_{\boldsymbol{\theta}} \text{ belongs to a certain functional family parameterized by } \boldsymbol{\theta}\}$$

PARAMETRIZATION / 2

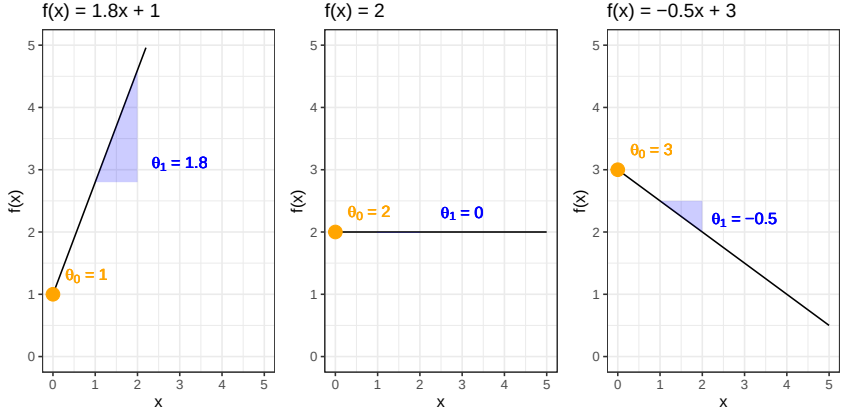
- This means: finding the optimal model is perfectly equivalent to finding the optimal set of parameter values.
- The relation between optimization over $f \in \mathcal{H}$ and optimization over $\boldsymbol{\theta} \in \Theta$ allows us to operationalize our search for the best model via the search for the optimal value on a d-dimensional parameter surface.



- $\boldsymbol{\theta}$ might be scalar or comprise thousands of parameters, depending on the complexity of our model.

EXAMPLE: UNIVARIATE LINEAR FUNCTIONS

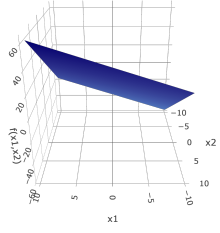
$$\mathcal{H} = \{f : f(\mathbf{x}) = \theta_0 + \theta_1 x, \boldsymbol{\theta} \in \mathbb{R}^2\}$$



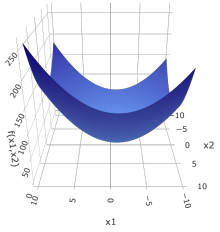
EXAMPLE: BIVARIATE QUADRATIC FUNCTIONS

$$\mathcal{H} = \{f : f(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2, \boldsymbol{\theta} \in \mathbb{R}^6\},$$

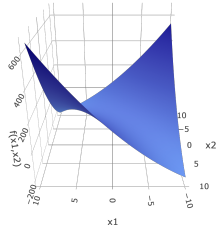
$f(x) = 3 + 2x_1 + 4x_2$



$f(x) = 3 + 2x_1 + 4x_2 +$
$+ 1x_1^2 + 1x_2^2$



$f(x) = 3 + 2x_1 + 4x_2 +$
$+ 1x_1^2 + 1x_2^2 + 4x_1 x_2$



IDEA OF SUPERVISED LEARNING

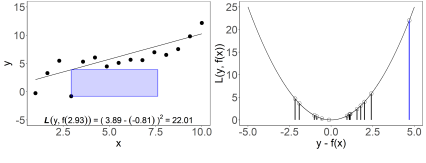
Goal: Automatically identify the fundamental functional relation in the data that maps an object's features to the target.

- **Supervised** learning means we make use of *labeled* data for which we observed the outcome.
- We use the labeled data to learn a model f.
- Ultimately, we use our model to compute predictions for **new** data whose target values are unknown.

The algorithm for finding our *f* is called **learner**. It is also called **learning algorithm**.

Introduction to Machine Learning

ML-Basics: Losses & Risk Minimization



Learning goals

- Know the concept of loss
- Understand the relationship between loss and risk
- Understand risk minimization

HOW TO EVALUATE MODELS

- When training a learner, we optimize over our hypothesis space, to find the function which matches our training data best.
- This means, we are looking for a function, where the predicted output per training point is as close as possible to the observed label.

Features x		Target y		Prediction $\hat{y}$
People in Office (Feature 1) x_1	Salary (Feature 2) x_2	Worked Minutes Week (Target Variable)	$\overset{?}{\approx}$	Worked Minutes Week (Target Variable)
4	4300 €	2220		2588
12	2700 €	1800		1644
5	3100 €	1920		1870

$\mathcal{D}_{\text{train}}$

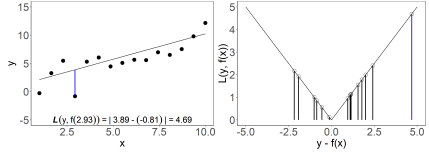
- To make this precise, we need to define now how we measure the difference between a prediction and a ground truth label pointwise.

LOSS

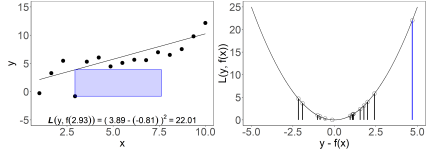
The **loss function** $L(y, f(\mathbf{x}))$ quantifies the "quality" of the prediction $f(\mathbf{x})$ of a single observation $\mathbf{x}$:

$$L : \mathcal{Y} \times \mathbb{R}^g \to \mathbb{R}.$$

In regression, we could use the absolute loss $L(y, f(\mathbf{x})) = |f(\mathbf{x}) - y|$;



or the L2-loss $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$:



RISK OF A MODEL

- The (theoretical) **risk** associated with a certain hypothesis $f(\mathbf{x})$ measured by a loss function $L(y, f(\mathbf{x}))$ is the **expected loss**

$$\mathcal{R}(f) := \mathbb{E}_{xy}[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x})) \, \mathrm{d}\mathbb{P}_{xy}.$$

- This is the average error we incur when we use f on data from $\mathbb{P}_{xy}$.
- Goal in ML: Find a hypothesis $f(\mathbf{x}) \in \mathcal{H}$ that **minimizes** risk.

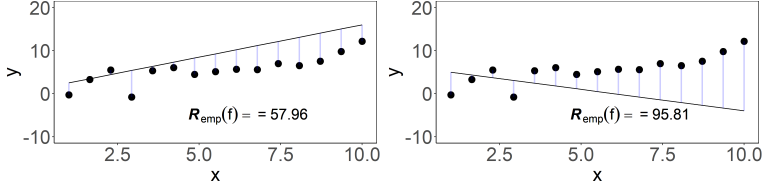
EMPIRICAL RISK

To evaluate, how well a given function f matches our training data, we now simply sum-up all f's pointwise losses.

$$\mathcal{R}_{\text{emp}}(f) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$$

This gives rise to the **empirical risk function** which allows us to associate one quality score with each of our models, which encodes how well our model fits our training data.

$$\mathcal{R}_{\text{emp}} : \mathcal{H} \to \mathbb{R}$$



EMPIRICAL RISK MINIMIZATION

The best model is the model with the smallest risk.

If we have a finite number of models f, we could simply tabulate them and select the best.

Model	$\theta_{intercept}$	θ_{slope}	$\mathcal{R}_{\text{emp}}(\theta)$
f_1	2	3	194.62
f_2	3	2	127.12
f_3	6	-1	95.81
f_4	1	1.5	57.96

WHAT COMES NEXT

We will deal with **supervised learning** for regression and classification: predicting labels *y* based on features *x*, using patterns that we learned from labeled data.

First, we will go through fundamental concepts in supervised ML:

- Regression
- Classification
- Neural network