# What is Machine Learning?

Scribes: Jose Rodriguez and Basanta Pahari

## 1 Introduction

Machine Learning is centered around the concepts of Data, Models and Learning. It originated in the computer science community [3] and can be roughly described as the field of study concerned with the development of automated methods for learning good models from data. Here the expression ”good model” refers to the predictive power of the method. One guiding principle of Machine Learning is that a model is as good as its ability to describe unseen data.

In this chapter, we illustrate the main ideas from machine learning by adapting some material from [1]. Even though the data that can be considered using its methods can be very general and include objects such as texts and chemical formulas, below we consider the situation where data is in numerical form.

Let $\mathscr{X} = \{x_1, \ldots, x_N\}$ where each $x_n \in \mathbb{R}^D$. We refer to any $x_n \in \mathscr{X}$ as an *example* or *data point*. In matrix form, we can write

$$\mathscr{X} = \begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{N,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,D} & x_{D,2} & \cdots & x_{N,D} \end{bmatrix} \in \mathbb{R}^{D \times N},$$

where each column is a data point or example and each row represents a particular *feature* or *attribute*. In the typical supervised learning setting, to each data point $x_n \in \mathbb{R}^D$, we associate a label $y_n \in \mathbb{R}$.

Published: January 23, 2020

### *1.1 Learning a model*

We can formalize the problem of *learning* as the problem of finding a *predictor* that assigns a data point to a label. In the machine learning literature, we can identify two main approaches to handle the predictor: it can be either a $(1)$ *function* or $(2)$ a *probabilistic model*.

- Case $(1)$: the predictor is a function. Here the predictor function $f$ maps a feature vector $x$ to a label $y$. For example, we can choose $f : \mathbb{R}^D \to \mathbb{R}$ of the form $f(x) = \theta^t x + \theta_0$ where $\theta \in \mathbb{R}^D$ and $\theta_0 \in \mathbb{R}$ are the parameters of the predictor.
- Case (2): the predictor is a probabilistic model. For instance, the predictor is a probability density function with finitely many parameters such as the normal distribution.

The goal of learning is to determine the parameters of a predictor such that it performs well on unseen data. We can identify three (algorithmic) phases in the learning process.

- (1) Prediction or inference. During this phase, the predictor is applied to unseen data (= test data) to generate an outcome.
- (2) Training or parameter estimation. During this phase, We adjust the parameters of the predictor based on training data. To perform this task, we need a measure of quality to control the performance of the predictor. For that, there are two main strategies depending on the predictor being a function or a probabilistic model. In the first case, a widely used strategy consists in using *Empirical Risk Minimization*. In the second case, another widely used method is *Maximum Likelihood Estimation.*
- (3) Parameter tuning or model selection. As part of the learning process, we need make high-level decisions about the structure of the predictor. Parameter tuning or model selection is concerned about making this decision. Recall that the end goal of the learning process is for the predictor to perform well on unseen data. This goal typically entails to make appropriate choices on the structure of the predictor.

Below, we briefly describe some general ideas about carrying over the learning process in the two cases where the predictor is a function or a probability model.

## 2 Predictor as a function

Assume we are given a set of $N$ samples $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^D$ and the corresponding (scalar) labels $Y = \{y_1, \ldots, y_N\} \subset \mathbb{R}$. We want to find a predictor of the form $f(\cdot;\theta) : \mathbb{R}^D \to \mathbb{R}$, with a parameter vector $\theta$ (we assume for simplicity that $\theta$ is finite dimensional) having the property that $f(x_n, \theta)$ provides a good approximation of the labels $y_n$, for all $n = 1, \ldots, N$. For example, we can choose our predictor to be a linear function $f(x, \theta) = \theta^t x$ where $\theta \in \mathbb{R}^D$.

## 2.1 Training

The goal of the training, is to exploit the data pairs $\{(x_1, y_1), \dots (x_N, y_N)\}$ in order to determine the best parameters $\theta$ of our predictor $f(\cdot, \theta)$; that is, training has the aim to possibly identify the value of the parameter $\theta^*$ giving the best fit of $f$ to the data. Here the notion of 'best fit' is determined by the introduction of an appropriate *loss function* $L(y_n, f(x_n, \theta))$ that takes the ground truth $y_n$ and the prediction $f(x_n, \theta)$ as input and outputs a non-negative number (*the loss*) measuring the error made in this particular prediction. The goal of training is to find the parameter $\theta^*$ that minimizes the loss $L$.

## 2.2 Empirical risk minimization

We assume that the example pairs $(x_1, y_1), \dots, (x_n, y_n)$ are independent and identically distributed (*i.i.d.*). This means that any two data points $(x_i, y_i)$ and $(x_j, y_j)$ are statistically indepent of each other and that they are drawn from the (unknown) distribution. This assumption implies that the empirical mean is a good estimate of the population mean. Hence we define the average loss on the training data as the *empirical risk*

$$R_{emp}(f, X, Y) = \frac{1}{N} \sum_{n=1}^{N} L(y_n, f(x_n; \theta)).$$

*Example 1 (Least-squares linear regression).* Let $(x_1, y_1), \dots, (x_N, y_N)$ be a collection of pairs in $\mathbb{R}^D \times \mathbb{R}$. We want to find a predictor function of the form

$$.f(x; \theta, \theta_0) = \theta^t x + \theta_0, \quad x \in \mathbb{R}^D, \theta \in \mathbb{R}^D, \theta_0 \in \mathbb{R}. \tag{1}$$

Observe that we can map any $x = (x^1, \dots, x^D)^t \in \mathbb{R}^D$ to $\tilde{x} = (1, x^{(1)}, \dots, x^{(D)})^t \in \mathbb{R}^{D+1}$ and similarly any $\theta = (\theta^{(1)}, \dots, \theta^{(D)})^t \in \mathbb{R}^D$ to $\tilde{\theta} = (\theta_0, \theta^{(1)}, \dots, \theta^{(D)})^t \in \mathbb{R}^{D+1}$. Using this map, we can redefine the affine function $f$ in (1) as the linear function $\tilde{f} : \mathbb{R}^{D+1} \to \mathbb{R}$ given by

$$\tilde{f}(\tilde{x}, \tilde{\theta}) = \tilde{\theta}^t \tilde{x}.$$

To find the best parameter $\tilde{\theta}$ of the predictor $\tilde{f}(\cdot, \tilde{\theta})$ for the given data we can use the squared loss function $L(y, \tilde{f}(x, \tilde{\theta})) = (y - \tilde{f}(\tilde{x}; \tilde{\theta}))^2$. Under these assumptions, the empirical risk becomes

$$R_{emp}(\tilde{f}, X, Y) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \tilde{f}(\tilde{x}_n, \tilde{\theta}))^2. = \frac{1}{N} \sum_{n=1}^{N} (y_n - \tilde{\theta}^t \tilde{x})^2.$$

Hence, to minimize the empirical risk we solve the expression

$$\cdot \min_{\tilde{\theta}\in\mathbb{R}^{D+1}} \frac{1}{N}\sum_{n=1}^{N}(y_n - \tilde{\theta}^t\tilde{x})^2 = \min_{\tilde{\theta}\in\mathbb{R}^{D+1}} \frac{1}{N}\|Y - \tilde{X}\tilde{\theta}\|^2, \tag{2}$$

where $Y$ is the vector containing the labels $y_n$ as entries and $\tilde{X}$ containing the data points $\tilde{x}_n$ as columns. The minimization problem (2) is known as the least-square linear regression problem.

As we observed above, we are not satisfied in a predictor that only performs well on the training data. Rather, we seek to find a predictor that performs well on unseen data. More precisely, we are interested in finding a predictor function $f(\cdot,\theta)$ that minimizes the *expected risk*

$$R_{true}(f) = \mathbb{E}_{x,y}[L(y, f(x))]$$

where $y$ is the label of $x$ and $f(x)$ is the prediction. In the expression above, the expectation $E_{x,y}$ is taken over the infinite set of all possible data and labels. Clearly, the notion of expected risk leads to a number of practical questions, most notably: how do we estimate expected risk from finite data? How do we manage our training procedure to ensure that we can control the expected risk?

It turns out that empirical risk minimization may lead to *overfitting*. This is the situation where the predictor fits closely the training but does not generalize well on new data, i.e., $R_{emp}$ underestimates $R_{true}$. How to address this situation is a major challenge in machine learning application.

One strategy that can be used to avoid overfitting is *regularization* consisting in finding a compromise between accurate solution of empirical risk minimization and the size or complexity of the model. In other words, it is a method that discourages complex or extreme solutions to an optimization problem in favor of simpler ones.

*Example 2 (Regularized least squares).* It is known if the number of variables does not exceed the number of data, the regression model may suffers from poor generalization. In this case, a simple regularization strategy is to add a penalty term involving the parameter $\theta$:

$$min_{\theta\in\mathbb{R}^D}\frac{1}{N}\|Y - \tilde{X}\tilde{\theta}\|^2 + \lambda\|\tilde{\theta}\|^2.$$

The term $\|\theta\|^2$ is called a regularizer and $\lambda$ is the regularization parameter. The effect of the regularization is to force the solution to be 'sparse' in some way or to reflect other prior knowledge about the problem such as information about correlations between features

**Bibliographical note.**. The original development of empirical risk minimization is due to Vapnik [4, 5] and was framed in a rather heavily theoretical language. The area of study that developed following this work is called *statistical learning theory* [2].

**Note** from [1].: ”Thinking about empirical risk minimization as 'probability free' is incorrect. There is an underlying unknown probability distribution $p(x,y)$ that governs the data generation. However, the approach of empirical risk minimization is agnostic to that choice of distribution. This is in contrast to standard statistical approaches that explicitly require the knowledge of $p(x,y)$. Furthermore, since the distribution is a joint distribution on both examples $x$ and labels $y$, the labels can be nondeterministic. In contrast to standard statistics we do not need to specify the noise distribution for the labels $y$."

## 3 Predictor as a probability model

We briefly sketch the general ideas that are applied when the predictor is a probability model. There is a parallel with the methodology we presented above where risk minimization becomes parameter estimation, loss function becomes likelihood and finally and regularization becomes a prior.

### *3.1 Maximum likelihood estimation*

We want to determine the parameters of the probability model that fit data well. To this end, we apply the notion of *likelihood* function. Hence, we assume that data are random variables associated with a probability density function $p(x;\theta)$ parametrized by $\theta$. We define the *negative log-likelihood* by

$$\mathscr{L}_x(\theta) = -\log p(x|\theta).$$

Note that, in this expression, $\theta$ is the variable (it is the quantity we want do find, for the given data) and $x$ is fixed. To find the value of the parameter vector $\theta$ that best fit the data, we want to maximize the likelihood and this is achieved by minimizing the function $\mathscr{L}_x(\theta)$ with respect to $\theta$.

In the example below, we illustrate this approach using a normal probability model.

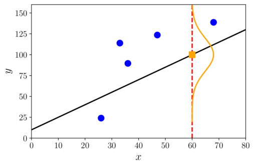

**Fig. 1** The uncertainty of the data is described using a Gaussian model.

*Example 3 (Gaussian distribution).* In this example, we assume that we can explain data uncertainty using a Gaussian probability model where the uncertainty of any observation is a Gaussian noise with zero mean and fixed variance $\sigma^2$. In addition, we also assume a linear model $\theta^T x_n$ for prediction. That is, we assume that for any observations $(x_n, y_n)$

$$p(y_n|x_n, \theta) = N(y_n, \theta^T x_n, \sigma^2). \tag{3}$$

This is illustrated in Fig. 1

We are given a set of examples $(x_1, y_1), \ldots, (x_N, y_N)$ that are independent and identically distributed (idd). Independence implies that the likelihood of the whole data set ($Y = \{y_1, \ldots, y_N\}$ and $X = \{x_1, \ldots, x_N\}$) factorizes into a product of the likelihoods of each individual example

$$P(Y|X, \theta) = \Pi_{n=1}^{N} p(y_n|x_n, \theta).$$

The property of being ‘identically distributed’ means that each term in the former product is the same distribution given by (3). Hence, we compute the negative log-likelihood as

$$\begin{aligned}
\mathscr{L}(\theta) &= -\sum_{n=1}^{N} \log p(y_n|x_n, \theta) \\
&= -\sum_{n=1}^{N} \log N(y_n|\theta^T x_n, \sigma^2) \\
&= -\sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y_n - \theta^T x_n}{2\sigma^2}\right) \\
&= \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \theta^T x_n)^2 - \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}}.
\end{aligned}$$

Solving the maximum likelihood estimation is equivalent to minimizing $\mathscr{L}(\theta)$. From the expressions above it is clear that this minimization is equivalent to solving the least-square regression problem.

*Remark 1.* Unlike the simple example above, it is not possible to derive a closed solution of the maximum likelihood estimation using general (non-Gaussian) probability models.

## References

1. Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press Cambridge, 2019.
2. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
3. B Marr. A short history of machine learning every manager should read. forbes. *Forbes. http://tinyurl. com/gslvr6k*, 2016.
4. Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2000.

5. Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.