# Machine Learning

Questions and Answers

Daniel Zahnd

September 16, 2024 - October 22, 2024

## Contents



## 1 Preliminaries

Usually, one deals with datasets containing of data $x \in \mathbb{R}^n$ associated to other data $y$ with $n \in \mathbb{N}$. The data $x$ can be scalar of vectorial, whereas the data $y$ can also be a scalar or vector. The data $x$ is either $y \in \mathbb{R}^q$ with $q \in \mathbb{R}$ or a number of a class $\{1, \ldots, K\}$ for $K \in \mathbb{N}$. In the case where e.g. $y \in \mathbb{R}$, one speaks of a regression problem, whereas in the case where $y \in \{1, \ldots, K\}$ are classes, one speaks of a classification task.

Usually, one has a dataset containing of various pairs of $x$ and $y$; to distinguish an instance of data from components of a vector, one writes $(x^{(i)}, y^{(i)})$ for an instance of a dataset containing

of $m \in \mathbb{N}$ instances of data. For components of an instance of data $x^{(i)} \in \mathbb{R}^n$, one writes

$$x^{(i)} = \begin{pmatrix} x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \in \mathbb{R}^n. \tag{1.1}$$

For a whole dataset, one can write

$$\{(x^{(i)}, y^{(i)})\}_{i=1,\dots,m} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}. \tag{1.2}$$

The goal in machine learning is thus to find a function $h$ with associated tunable parameters $\theta$, that predicts $y$ based on $x$; hence one can write

$$h_\theta(x) = y. \tag{1.3}$$

The ultimate goal usually is to determine the parameters (weights) $\theta$; whereas for the shape or nature of the function $h$, one can usually make assumptions. The $h$ is reminiscent of "hypothesis" and thus of "hypothesis" function.

## 1.1 Probability measures

### 1.1.1 Random variable

A random variable is some quantity $x$, which can take a random value. Those random values follow a certain probability distribution, which is determined by the underlying process constituting the random variable.If the probability distribution of a random variable is known, the probability density $p(x)$ can be written down for the continuous and discrete cases as

$$\int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1 \qquad \text{and} \qquad \sum_i p(x_i) = 1. \tag{1.4}$$

### 1.1.2 Expectation value

The expectation value $\mathbb{E}(x) \doteq \bar{x}$ of a random variable $x$ for both the continuous and discrete case is defined as

$$\mathbb{E}(x) = \int_{-\infty}^{\infty} x p(x)\,\mathrm{d}x \qquad \text{and} \qquad \mathbb{E}(x) = \sum_i x_i p(x_i). \tag{1.5}$$

### 1.1.3 Variance, standard deviation and covariance

The variance $\mathbb{V}(x)$ of a continuous or discrete random variable $x$ can be calculated by means of

$$\mathbb{V}(x) = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x)\,\mathrm{d}x \qquad \text{and} \qquad \mathbb{V}(x) = \sum_i (x_i - \bar{x})^2 p(x_i). \tag{1.6}$$

The standard deviation $\sigma_x$ is defined as the square root of the variance, hence

$$\sigma_x = \sqrt{\mathbb{V}(x)}. \tag{1.7}$$

Let now $x_1, \dots, x_n$ be random variables with associated probability densities $p(x_j), j \in \{1, \dots, n\}$ and joint probability densities $p(x_i, x_j)$. Let furthermore be $\bar{x}_i = \mathbb{E}(x_i)$. The covariance $\mathbb{K}(x_i, x_j)$ of two random variables for the continuous and discrete case is defined as

$$\mathbb{K}(x_i, x_j) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_i - \bar{x}_i)(x_j - \bar{x}_j) p(x_i, x_j)\,\mathrm{d}x_i\,\mathrm{d}x_j \tag{1.8}$$

and

$$\mathbb{K}(x_i, x_j) = \sum_{k,l} (x_{i_k} - \bar{x}_i)(x_{j_l} - \bar{x}_j) p(x_{i_k}, x_{j_l}). \tag{1.9}$$

In the context of covariance, one usually also defines the correlation coefficient $\rho(x_i, x_j)$ between two random variables as

$$\rho(x_i, x_j) = \frac{\mathbb{K}(x_i, x_j)}{\sqrt{\mathbb{V}(x_i)\mathbb{V}(x_j)}} = \frac{\mathbb{K}(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}. \tag{1.10}$$

## 1.2 Preliminary questions

1. **Q** — How is the uniform distribution defined?

**A** — Consider a continuous random variable $x$. The uniform probability distribution is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{U}(x; a, b) = \begin{cases} \frac{1}{b-a} & , a \leq x \leq b \\ 0 & , \text{otherwise} \end{cases} \tag{1.11}$$

where $a$ and $b$ define the domain of the distribution. The expectation value and the variance of the uniform distribution are given by

$$\mathbb{E}(x) = \frac{a+b}{2}, \quad \mathbb{V}(x) = \frac{1}{12}(b-a)^2. \tag{1.12}$$

The uniform distribution is used for processes with no prior knowledge about the probabilities of events. It is furthermore used to model processes, where all outcomes are equally likely to happen.

2. **Q** — How is the standard normal distribution defined?

**A** — Consider a continuous random variable $x$. The standard normal probability distribution (Gaussian) is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.13}$$

where $\mu$ defines the expectation value and $\sigma$ the variance the distribution. The expectation value and the variance of the uniform distribution are given by

$$\mathbb{E}(x) = \mu, \quad \mathbb{V}(x) = \sigma^2. \tag{1.14}$$

The normal distribution is used for processes, which are influenced by an additive effect of a large number of different and independent influences modelled by arbitrary probability distributions. It pertains - with the uniform distribution - to the two default probability distributions to use, where little prior information about the modelled processes is available.

3. **Q** — How is the Bernoulli distribution defined?

**A** — Consider a discrete random variable $x$. The Bernoulli probability distribution is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{B}(x; p) = \begin{cases} p & , x = 1 \\ 1-p & , x = 0 \end{cases}, \tag{1.15}$$

where $0 \leq p \leq 1$ is the probability of success (i.e., $x = 1$). The random variable $x$ takes on only two possible values: 1 (success) or 0 (failure).

The expectation value and variance of the Bernoulli distribution are given by

$$\mathbb{E}(x) = p, \quad \mathbb{V}(x) = p(1-p). \tag{1.16}$$

The Bernoulli distribution models a single trial of an experiment where there are exactly two possible outcomes: success with probability $p$ and failure with probability $1-p$. It is a fundamental building block of the binomial distribution, which models a sequence of independent Bernoulli trials.

4. **Q** — How is the binomial distribution defined?

**A** — Consider a discrete random variable $x$. The binomial probability distribution is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{B}(x; n, p) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} & , x \in \{1, \dots, n\} \\ 0 & , \text{otherwise} \end{cases}, \tag{1.17}$$

where $n$ denotes the number of trials of an experiment and $p$ is the respective probability of success or failure of the outcome. The expectation value and the variance of the binomial distribution are given by

$$\mathbb{E}(x) = np, \quad \mathbb{V}(x) = np(1-p). \tag{1.18}$$

The binomial distribution is used to model processes that model a series of identical and independent experiments with exactly two possible outcomes, success or failure.

5. **Q** — How is the Poisson distribution defined?

**A** — Consider a discrete random variable $x$. The Poisson probability distribution is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{P}(x; \lambda) = \begin{cases} \frac{\lambda^x}{x!} e^{-\lambda} & , x \in \mathbb{N}_0 \\ 0 & , \text{otherwise} \end{cases}, \tag{1.19}$$

where $\lambda > 0$ defines both the expectation value and the variance. The expectation value and the variance of the Poisson distribution are given by

$$\mathbb{E}(x) = \lambda, \quad \mathbb{V}(x) = \lambda. \tag{1.20}$$

The Poisson distribution is - similarly to the binomial distribution - used to model processes that model a series of identical and independent experiments with exactly two possible outcomes, success or failure, but where the probability $p$ of success behaves as $p \to 0$ and where the number of trials $n$ behaves as $n \to \infty$.

6. **Q** — How is the Poisson distribution defined?

**A** — Consider a discrete random variable $x$. The Poisson probability distribution is defined by the probability density $p(x)$ as

$$p(x) = \mathcal{P}(x; \lambda) = \begin{cases} \frac{\lambda^x}{x!} e^{-\lambda} & , x \in \mathbb{N}_0 \\ 0 & , \text{otherwise} \end{cases}, \tag{1.21}$$

where $\lambda > 0$ defines both the expectation value and the variance. The expectation value and the variance of the Poisson distribution are given by

$$\mathbb{E}(x) = \lambda, \quad \mathbb{V}(x) = \lambda. \tag{1.22}$$

The Poisson distribution is - similarly to the binomial distribution - used to model processes that model a series of identical and independent experiments with exactly two possible outcomes, success or failure, but where the probability $p$ of success behaves as $p \to 0$ and where the number of trials $n$ behaves as $n \to \infty$.

7. **Q** — What is the popular definition of machine learning authored by Arthur Samuel in 1959?

**A** — According to Samuel, machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. Explicitly programmed here means, that everything would be hard-coded, instead of rule-based programming.

8. **Q** — What is another popular definition of machine learning authored by Tom Michel in 1999?

**A** — According to Michel, a well-posed machine learning problem may be described as follows: A computer program is said to learn from experience $E$ with respect to some tasks $T$ and performance measure $P$, if its performance at tasks $T$ as measured by performance $P$ improves with experience $E$.

9. **Q** — Broadly speaking, which three types of machine learning algorithms are there?

**A** — Broadly speaking, there are three different categories of machine learning algorithms:

(1) Supervised learning: A machine learns how to make predictions about a specific target of interest, given some observations.

(2) Unsupervised learning: A machine learns how to find useful structures and patterns in given data by itself.

(3) Reinforcement learning: A machine has the ability to act and thus influence its own observations, thereby learning to make predictions to achieve a given goal.

An example of supervised learning would be a classification task; that is to say to assign data to two or more given classes of objects depending on one or more variables. One could for example do this by means of linear or polynomial regression.

In comparison to supervised learning, unsupervised learning would for example try to separate data into two or more classes of objects, depending on what makes sense to the algorithm.

Reinforcement learning finally is about problems, where a sequence of decisions over time is required, where the basic idea is to implement a reward function as supervision to give the model a way to improve itself.

10. **Q** — How does gradient descent work; and what is the difference between stochastic gradient descent and just gradient descent?

**A** — Gradient descent is one of the most useful and foundational techniques in machine learning; basically, it is about minimizing a loss function $J(\theta)$ with respect to weights (parameters) $\theta$ which belong to the chosen model.

Hence, let now $J(\theta)$ be the loss function of a model, where $\theta = (\theta_1, \ldots, \theta_n)^\top$ are the parameters of the model, which need to be optimized. The loss function might take a form as given in

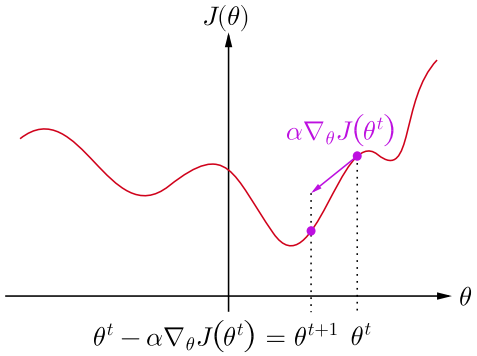


**Figure 1:** Visualization of the gradient descent routine. Note, that $\theta$ is generally a vector; thus, the figure does only represent the general case in the case where $\theta$ is a scalar. If $\theta$ would consist of two components, the figure would be still visualizable in two dimensions on paper, but for higher dimensions in $\theta$, gradient descent may not be visualizable anymore on paper.

fig. 1. Now, $\theta^t$ denotes the weights at iteration step $t \in \{1, \dots, T\}$ for a total iteration time $T$. Well, gradient descent updates the weights as

$$\theta^{t+1} = \theta^t - \alpha\nabla_\theta J(\theta^t), \tag{1.23}$$

where the gradient is taken with respect to the weights $\theta$ and is evaluated at $\theta^t$, i.e.

$$\nabla_\theta = \sum_{j=1}^{n} e_j \partial_{\theta_j}. \tag{1.24}$$

In the case, where no confusion arises to which respect a gradient is taken, the subscript can also be left away. The parameter $\alpha$ is called the learning rate in machine learning and it has to be chosen such, that the algorithm does find the global, instead of just a local minimum of $J(\theta)$.

11. **Q** — What is the rationale for the learning rate $\alpha$ in gradient descent?

**A** — Consider fig. 2; it shows a curve $J(\theta)$ and a quadratic function $g(\theta) = \|\theta - \theta^t\|$. Now, one can Taylor expand the function $J(\theta)$ around an iteration step $\theta^t$ quite easily to first order in $\theta$

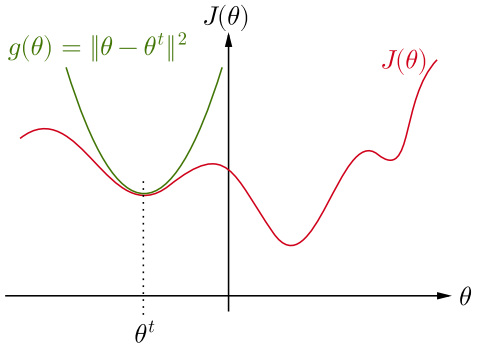


**Figure 2:** Rationale behind the gradient descent routine and the learning rate $\alpha$.

as

$$J(\theta) \approx J(\theta^t) + \nabla_\theta J(\theta^t)^\top (\theta - \theta^t). \tag{1.25}$$

Now, the second order term would involve the Hessian, which can be quite complicated. However, one can always add a quadratic term of the form $g(\theta)$, which is scaled by some parameter $\varepsilon$, such

that the resulting approximation of $J(\theta)$ always stays above the exact function; the added term with $\varepsilon$ thus approximates the Hessian. One hence has

$$J(\theta) \approx J(\theta^t) + \nabla_\theta J(\theta^t)^\top (\theta - \theta^t) + \frac{1}{2\varepsilon}\|\theta - \theta^t\|^2. \tag{1.26}$$

In order to find a minimum of this expression, one has to compute the gradient of the expression with respect to $\theta$, set it to zero and solve for $\theta$. From this,

$$\nabla_\theta J(\theta) \approx 0 + \nabla_\theta J(\theta^t) + \frac{1}{\varepsilon}(\theta - \theta^t) \stackrel{!}{=} 0 \quad \Leftrightarrow \quad \theta = \theta^t - \varepsilon \nabla_\theta J(\theta^t) \tag{1.27}$$

follows. This is nothing but gradient descent, where $\varepsilon = \alpha$ is the learning rate. As one can see, the width $(2\varepsilon)^{-1}$ of the added quadratic term to the first order Taylor expansion of $J(\theta)$ is the inverse of the learning rate. If the learning rate is large, the width of the added quadratic term is very narrow.

# 2 Supervised learning

## 2.1 Linear regression

### 2.1.1 Least mean squares

12. **Q** — How can a most simple machine learning model performing affine regression be implemented?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \mathbb{R}$. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset. Now, for affine regression, we can suggest a hypothesis function

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n, \tag{2.1}$$

which, defining $x \doteq (1, x_1, x_2, \dots, x_n)^\top$, can be written more compactly as

$$h_\theta(x) = \theta^\top x. \tag{2.2}$$

As a loss function to minimize, it makes sense to take the Euclidean norm as a metric to measure "closeness" of the prediction $h_\theta(x^{(i)})$ to the actual value $y^{(i)}$, thus

$$J(\theta) \doteq \frac{1}{2}\sum_{i=1}^{m} \|h_\theta(x^{(i)}) - y^{(i)})\|^2 = \frac{1}{2}\sum_{i=1}^{m} \|\theta^\top x^{(i)} - y^{(i)})\|^2. \tag{2.3}$$

The update rule for this regression task would be thus given by

$$\theta^{t+1} = \theta^t - \alpha \nabla_\theta J(\theta) = \theta^t - \alpha \sum_{i=1}^{m} \left(\theta^\top x^{(i)} - y^{(i)}\right) x^{(i)}, \tag{2.4}$$

where $\alpha \in \mathbb{R}$ is a suitably chosen learning rate. The proposed algorithm is called a least mean squares algorithm. This algorithm is equivalent to stating the optimal model parameters $\hat{\theta}$ as

$$\hat{\theta} = \arg\min_\theta \left( \sum_{i=1}^{m} \left[\theta^\top x^{(i)} - y^{(i)}\right]^2 \right). \tag{2.5}$$

13. **Q** — How can the least mean squares method be given in matrix form in the general case?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \mathbb{R}$. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset. Define a matrix $X$, such that $(x^{(i)})^\top$ is the $i$-th row and a matrix $Y$, such that $y^{(i)}$ is also the $i$-th row, namely

$$X \doteq \begin{pmatrix} 1 & (x^{(1)})^\top \\ \vdots & \vdots \\ 1 & (x^{(m)})^\top \end{pmatrix}, \qquad Y \doteq \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}. \tag{2.6}$$

Hereby, a column of ones was added to the matrix $X$ to account for the shift parameter of an affine functino. Given a parameter vector $\theta^\top = (\theta_0, \theta_1, \dots, \theta_n)$, one can then write down the hypothesis function $h_\theta(X)$ as

$$h_\theta(X) = X\theta \approx Y. \tag{2.7}$$

We thus write down the loss function as

$$J(\theta) = \frac{1}{2}\|X\theta - Y\|^2 = \frac{1}{2}(X\theta - Y)^\top (X\theta - Y). \tag{2.8}$$

Taking the gradient $\nabla_\theta$ of this expression leads to

$$\nabla_\theta J(\theta) = \theta^\top X^\top X - (X^\top Y)^\top. \tag{2.9}$$

Setting this expression to zero yields the closed-form solution

$$\theta^\star = (X^\top X)^{-1} X^\top Y \tag{2.10}$$

to the optimization problem

$$\theta^\star = \arg\min_\theta \left[ J(\theta) \right]. \tag{2.11}$$

14. **Q** — There is a closed form solution to the least mean squares problem? Why is it however not heavily used in practice?

**A** — The reason behind this is that the closed-form solution to the least mean squares problem involves the calculation of inverse matrices. The inverse matrix to be inverted is of size $m \times m$, where $m \in \mathbb{N}$ is the amount of data instances. Calculating the inverse of a matrix is computationally very demanding, especially for large matrices, as the calculation time goes with $m^2$.

Therefore, when working with large datasets, the closed-form solution is not the best option; rather, one used the gradient descent routine, which is much more efficient with large datasets, because no inverse matrices have to be calculated.

15. **Q** — How is the locally weighted affine regression method defined? What is the main difference to the normal affine regression method?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \mathbb{R}$. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset. Now, the optimal parameters $\hat{\theta}$ for affine regression are found by means of performing

$$\hat{\theta} = \arg\min_\theta \left( \sum_{i=1}^{m} \left[ \theta^\top x^{(i)} - y^{(i)} \right]^2 \right). \tag{2.12}$$

Now, the locally weighted affine regression takes as an input a query vector $x$ and uses a weighting function $w_i(x)$ defined by the exponential

$$w_i(x) = e^{-\frac{(x - x^{(i)})^2}{2\tau^2}}, \quad \tau \in \mathbb{R} \tag{2.13}$$

to weigh the samples $x^{(i)}$ close to the query more than samples further away from it. This weighting function is then multiplied by the objective, thus one has

$$\hat{\theta} = \arg\min_{\theta} \left( \sum_{i=1}^{m} w_i(x) \left[ \theta^\top x^{(i)} - y^{(i)} \right]^2 \right) \tag{2.14}$$

for the optimal parameters $\hat{\theta}$ of the locally weighted affine regression model. Thereby, also the main difference to the normal affine regression method is evident; the locally weighted method takes as an input a query vector $x$ which requires a new optimization of the model parameters each time a new (different) query is made; the normal affine regression method however requires to find the optimal model parameters only once, independent of what the query vector $x$ will be.

### 2.1.2 Probabilistic interpretation

16. **Q** — Which are three often used assumptions in machine learning?

**A** — The three main assumptions are:

(1) Feature data $x$ and associated data $y$ (related by a model $y = f(x;\theta)$) are modeled by means of a parametric probability density, that is to say, one assumes, that the data follows a probability density

$$p = p(y|x;\theta), \tag{2.15}$$

where $\theta$ are the model parameters.

(2) The data is assumed to be independent and identically distributed (IID). This assumption allows to write a joint probability distribution $p(y^{(i)},\dots,y^{(m)}|x^{(1)},\dots,x^{(m)};\theta)$ as a simple product, namely

$$p(y^{(i)},\dots,y^{(m)}|x^{(1)},\dots,x^{(m)};\theta) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta). \tag{2.16}$$

(3) The maximum likelihood approach can be used to identify the model parameters $\theta$ for the model $y = f(x;\theta)$, that is to say the optimal model parameters are found by means of

$$\hat{\theta} = \arg\max_{\theta} \left( \sum_{i=1}^{m} \ln \left[ p(y^{(i)}|x^{(i)};\theta) \right] \right). \tag{2.17}$$

17. **Q** — How and under which assumptions can it be shown, that Gaussian maximum likelihood and least mean squares yield the same result?

**A** — Assume, that the data $y$ is given by a linear model of the feature data $x$, where Gaussian noise is added. That is to say, one has a model

$$y \approx h_\theta(x) = \theta^\top x + \eta, \quad p(\eta) = \mathcal{N}(\eta;\mu=0,\sigma=1). \tag{2.18}$$

The conditional probability for $y$ given $x$ can thus be written as

$$p(y|x;\theta,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\theta^\top x)^2}{2\sigma^2}}, \tag{2.19}$$

where we have made the assumption that the expectation value for $y \approx h_\theta(x)$ is precisely the affine model $\theta^\top x$, thus $\mathbb{E}[h_\theta(x)] = \theta^\top x$; this is equivalent to stating, that we expect the data $y$ to

follow an affine model of $x$. Now, we have samples $\{(x^{(i)}, y^{(i)})\}_{i=1,\dots,m}$. The maximum likelihood approach with the given assumptions thus yields

$$p(y^{(1)}, \dots, y^{(m)} | x^{(1)}, \dots, x^{(m)}; \theta, \sigma) = \prod_{i=1}^{m} p(^{(i)} | x^{(i)}; \theta, \sigma) \tag{2.20}$$

because of the IID assumption. The optimal model parameters $\hat{\theta}$ are thus given by means of

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta,\sigma} \left( \sum_{i=1}^{m} \ln \left[ p(y^{(i)} | x^{(i)}; \theta, \sigma) \right] \right) \\ &= \arg\max_{\theta,\sigma} \left( \ln \left[ \frac{1}{\sqrt{2\pi}\sigma} \right] - \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left[ y^{(i)} - \theta^\top x^{(i)} \right]^2 \right) \\ &= \arg\min_{\theta} \left( \sum_{i=1}^{m} \left[ \theta^\top x^{(i)} - y^{(i)} \right]^2 \right), \end{aligned} \tag{2.21}$$

which is exactly the equation we obtained for the affine regression model with the least squares method. Note, that in the above last step we used $\sigma = 1$.

18. **Q** — How is the maximum likelihood approach for optimization problems defined?

**A** — Suppose, that one has data $y$ associated to other data $x$ by means of a model

$$y \approx h_\theta(x), \tag{2.22}$$

that depends on some model parameters $\theta$, which are to be found. What has to be assumed for the model $h_\theta(x)$ is that $y|x$ follows a certain probability density $p(y|x)$. Suppose now, that one has samples

$$\{(x^{(i)}, y^{(i)})\}_{i=1,\dots,m} \tag{2.23}$$

that are independently and identically distributed, one can find the model parameters $\theta$ by means of the so-called maximum likelihood approach. One calculates

$$p(y^{(1)}, \dots, y^{(m)} | x^{(1)}, \dots, x^{(m)}; \theta) = \prod_{i=1}^{m} p(y^{(i)} | x^{(i)}; \theta). \tag{2.24}$$

Hereby, the identically and independently distributed (IID) assumption has been used to write the joint probability as a product. Now, in order to find the model parameters $\theta$, this probability has to be maximized, since it should be highly likely to get precisely the samples $\{(x^{(i)}, y^{(i)})\}_{i=1,\dots,m}$. Therefore, the model parameters $\hat{\theta}$ are given by

$$\hat{\theta} = \arg\max_{\theta} \left( \sum_{i=1}^{m} \ln \left[ p(y^{(i)} | x^{(i)}; \theta) \right] \right). \tag{2.25}$$

Taking the logarithm of the probability renders the product as a sum, without changing the argument of maximal probability in $\theta$.

## 2.2 Classification and logistic regression

19. **Q** — How is the logistic regression algorithm defined?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \{0, 1\}$; this is to say, we have a binary classification task at hand. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset. We want

to find a model, which classifies the data $x$ into either the category $y = 0$ or $y = 1$. For this purpose, one can propose the hypothesis function

$$p(y = 1|x; \theta) \approx h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}, \quad p(y = 0|x; \theta) \approx 1 - h_\theta(x), \tag{2.26}$$

where $\theta^\top = (\theta_0, \theta_1, \ldots, \theta_n)$ and $x = (1, x_1, \ldots, x_n)^\top$. Now, one can rewrite

$$p(y = 1|x; \theta) = \phi, \qquad p(y = 0|x; \theta) = 1 - \phi \tag{2.27}$$

and thus in general

$$p(y|x; \theta) = \phi^y (1 - \phi)^{1-y}. \tag{2.28}$$

The maximum likelihood approach now yields for the optimal model parameters $\hat{\theta}$

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta \left( \sum_{i=1}^{m} \ln \left[ p(y^{(i)}|x^{(i)}; \theta) \right] \right) \\ &= \arg\max_\theta \left( \sum_{i=1}^{m} \left[ y^{(i)} \ln \left( \frac{1}{1 + e^{-\theta^\top x^{(i)}}} \right) + \left( 1 - y^{(i)} \right) \ln \left( \frac{e^{-\theta^\top x^{(i)}}}{1 + e^{-\theta^\top x^{(i)}}} \right) \right] \right). \end{aligned} \tag{2.29}$$

With the resulting model, one can then query for an $x$, with which probability it belongs to the class $y = 1$ or $y = 0$ by means of calculating

$$p(y = 1|x; \theta) = h_\theta(x), \qquad p(y = 0|x; \theta) = 1 - h_\theta(x). \tag{2.30}$$

## 2.3 Generalized linear models

20. **Q** — Given some function $f(y)$ and a probability density of $y$ given by $p(y|x)$, where $x$ is some other data. What is the expectation value $\mathbb{E}[f(y)|x]$?

**A** — The expectation value of a continuous random variable $y$ with probability density $p(y)$ is defined as

$$\mathbb{E}(y) = \int_{\mathbb{R}} y p(y) \, \mathrm{d}y. \tag{2.31}$$

Given some conditional probability density $p(y|x)$ and a function $f(y)$, the expectation value $\mathbb{E}[f(y)|x]$ is given by

$$\mathbb{E}[f(y)|x] = \int_{\mathbb{R}} f(y) p(y|x) \, \mathrm{d}y, \tag{2.32}$$

since

$$\int_{\mathbb{R}} p(y|x) \, \mathrm{d}y = 1 \quad \text{and} \quad \mathbb{E}[y|x] = \int_{\mathbb{R}} y p(y|x) \, \mathrm{d}y. \tag{2.33}$$

21. **Q** — What is the exponential family of probability densities and why is it important for generalized linear models?

**A** — A probability density $p(y|x; \eta)$ is said to belong to the exponential family, if it can be written in the form

$$p(y|x; \eta) = b(y) e^{\eta^\top T(y) - a(\eta)}, \tag{2.34}$$

where

(1) $\eta$ is called the natural parameter,

(2) $T(y)$ is called the sufficient statistic (oftentimes $T(y) = y$) and where

(3) $a(\eta)$ is the logarithmic partition function, essentially playing the role of a normalizing constant.

Recall, that $y$ is related to $x$ by means of a hypothesis function $h_\eta(x)$; that is why $x$ appears in the conditional probability $p(y|x;\eta)$. The parameters to optimize for the family of these distributions are $\eta$.

22. **Q** — What is the general framework of a generalized linear model?

**A** — There are three properties, by which a generalized linear model can be defined:

(1) The parameters $\eta$ to be optimized must be of the form $\eta = \theta^\top x = \eta(\theta)$.

(2) The probability density $p(y|x;\theta)$ (where $\theta = \theta(\eta)$) used to model the data must belong to the exponential family; therefore, one has to be able to rewrite $p(y|x;\theta)$ as $p(y|x;\theta) = b(y)e^{\eta^\top T(y)-a(\eta)}$ with some $b(y)$, $T(y)$ and $a(\eta)$.

(3) The hypothesis function $h_\theta(x)$ modeling $y$ must be of the form $h_\theta(x) = \mathbb{E}(T(y)|x)$.

23. **Q** — How is affine regression derived from a generalized linear model?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \mathbb{R}$. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset.

In order to derive a generalized linear model, we can perform the three key steps. First, we try to choose a probability density $p(y|x)$, which is suitable for the problem and check, if the chosen density belongs to the exponential family. For the sake of simplicity, $x \in \mathbb{R}$ and $y \in \mathbb{R}$ and $\sigma = 1$ for the moment. The normal distribution reads in this case as

$$p(y|x;\mu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2}}. \tag{2.35}$$

Rearranging terms and defining $\mu \doteq \eta$, $T(y) \doteq y$, $a(\eta) \doteq \eta^2 = \mu^2$ and $b(y) \doteq (2\pi)^{-1/2} e^{-y^2/2}$, one can rewrite the above probability density as

$$p(y|x;\eta) = b(y)e^{\eta^\top T(y)-a(\eta)}, \tag{2.36}$$

which proves that the normal distribution belongs to the exponential family. Thus, requirement (2) generalized linear (GLMs) is satisfied. Now, the third requirement of GLMs is that the hypothesis function is given by

$$h_\theta(x) = \mathbb{E}(T(y)|x). \tag{2.37}$$

In our case we have $T(y) = y$ and hence

$$h_\theta(x) = \mathbb{E}(T(y)|x) = \mathbb{E}(y|x) = \int_{\mathbb{R}} y p(y|x;\eta)\,\mathrm{d}y = \eta = \mu. \tag{2.38}$$

Requirement (1) of GLMs now requires us to choose $\eta = \theta^\top x$, which gives us with $h_\theta(x) = \eta$ the hypothesis function

$$h_\theta(x) = \theta^\top x \tag{2.39}$$

to implement, which is exactly the hypothesis function for affine regression.

24. **Q** — How is logistic regression derived from a generalized linear model?

**A** — Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \{0, 1\}$. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1, \dots, m\}$ be a training dataset.

In order to derive a generalized linear model, we can perform the three key steps. First, we try to choose a probability density $p(y|x)$, which is suitable for the problem and check, if the chosen density belongs to the exponential family. We choose a Bernoulli distribution of the form

$$p(y|x;\phi) = \phi^y(1-\phi)^{1-y}. \tag{2.40}$$

Rearranging terms and defining $b(y) = 1$ (unity matrix), $\eta \doteq \ln\left(\frac{\phi}{1-\phi}\right)$ and $a(\eta) \doteq -\ln(1-\phi) = \ln(1+e^{\eta})$ and $T(y) \doteq y$, one can rewrite the above probability density as

$$p(y|x;\eta) = b(y)e^{\eta^\top T(y) - a(\eta)}, \tag{2.41}$$

which proves that the Bernoulli distribution belongs to the exponential family. Thus, requirement (2) generalized linear (GLMs) is satisfied. Note, that $\eta$ relates to $\phi$ as

$$\phi(\eta) = \frac{1}{1+e^{-\eta}}. \tag{2.42}$$

Now, the third requirement of GLMs is that the hypothesis function is given by

$$h_\theta(x) = \mathbb{E}(T(y)|x). \tag{2.43}$$

In our case we have $T(y) = y$ and hence

$$h_\theta(x) = \mathbb{E}(T(y)|x) = \mathbb{E}(y|x) = \sum_{y\in\{0,1\}} yp(y|x;\eta) = \phi. \tag{2.44}$$

Requirement (1) of GLMs now requires us to choose $\eta = \theta^\top x$, which gives us with $h_\theta(x) = \phi$ the hypothesis function

$$h_\theta(x) = \frac{1}{1+e^{-\theta^\top x}} \tag{2.45}$$

to implement, which is exactly the hypothesis function for logistic regression.

25. **Q** — How is logistic regression for $k \in \mathbb{N}$ classes rather than just for two classes defined?

**A** — Logistic regression for multiple classes is also known as multinomial logistic regression or softmax regression. This is a widely used technique in machine learning to cluster data into multiple classes.

Softmax regression can be derived based on the framework of GLMs. Let $x \in \mathbb{R}^n$ be a feature vector and $y \in \{1,\dots,k\}$, where each index for $y$ refers to a different class. Let furthermore $(x^{(i)}, y^{(i)})$ with $i \in \{1,\dots,m\}$ be a training dataset. Now, the probability density for the multinomial case can be generalized from the binomial case, which is in that case the Bernoulli probability density. Generalized to the multinomial case, the probability density reads as

$$p(y|x;\theta) = \phi_1^{\mathbb{I}\{y=1\}} \cdot \dots \cdot \phi_k^{\mathbb{I}\{y=k\}}, \tag{2.46}$$

where

$$\mathbb{I}\{y=i\} = \begin{cases} 1, & y = i \\ 0, & \text{otherwise} \end{cases}, \tag{2.47}$$

is the indicator function and where $\phi_i = \phi_i(\theta)$ are functions of the model parameters $\theta$. One can hence write the individual probability densities $\phi_i$ as

$$\phi_i = p(y=i|x;\theta). \tag{2.48}$$

Now, concerning the $\phi_i$, one can state that they need to add up to 1, since this is a necessary condition on probability densities; hence we have

$$\sum_{i=1}^{k} \phi_i = 1, \qquad \phi_k = 1 - \sum_{i=1}^{k-1} \phi_i. \tag{2.49}$$

Since logistic regression is also in its multinomial form a linear model, one needs to verify the three key conditions for GLMs. First, we check, if the above proposed probability density is part of the exponential family. Towards this end, we define the function $T(y)$ as a $k-1$ dimensional vector, that for $y = i$ has a one in row $i$ but zeros in all other rows, hence

$$[T(y)]_i = \mathbb{I}\{y = i\}, \quad i \in \{1, \dots, k-1\} \quad \Leftrightarrow \quad T(y) = \begin{pmatrix} \mathbb{I}\{y = 1\} \\ \vdots \\ \mathbb{I}\{y = k-1\} \end{pmatrix}. \tag{2.50}$$

With this function, we can write

$$\begin{aligned} p(y|x;\theta) &= \phi_1^{\mathbb{I}\{y=1\}} \cdot \dots \cdot \phi_k^{\mathbb{I}\{y=k\}} \\ &= \phi_1^{[T(y)]_1} \cdot \dots \cdot \phi_{k-1}^{[T(y)]_{k-1}} \phi_k^{1-\sum_{j=1}^{k-1}[T(y)]_j} \\ &= \exp\left( [T(y)]_1 \ln(\phi_1) + \dots + [T(y)]_{k-1} \ln(\phi_{k-1}) + \left(1 - \sum_{j=1}^{k-1} [T(y)]_j\right) \ln(\phi_k) \right) \\ &= \exp\left( [T(y)]_1 \ln\left(\frac{\phi_1}{\phi_k}\right) + \dots + [T(y)]_{k-1} \ln\left(\frac{\phi_{k-1}}{\phi_k}\right) + \ln(\phi_k) \right). \end{aligned} \tag{2.51}$$

Defining a vector $\eta$ of $k-1$ elements as

$$\eta^\top \doteq \left( \ln\left[\frac{\phi_1}{\phi_k}\right], \dots, \ln\left[\frac{\phi_{k-1}}{\phi_k}\right] \right), \tag{2.52}$$

and furthermore defining the $k$-th element as $\eta_k \doteq 0$, we can write $p(y|x;\theta)$ finally as

$$p(y|x;\theta) = e^{\eta^\top T(y) - (-\ln[\phi_k])}. \tag{2.53}$$

If it can be shown, that $-\ln(\phi_k)$ can be written as a function $a(\eta)$ of $\eta$, it is verified that the proposed probability density $p(y|x;\theta)$ belongs to the exponential family. To this end, one calculates

$$e^{\eta_i} = \frac{\phi_i}{\phi_k} \quad \Leftrightarrow \quad \phi_i = \phi_k e^{\eta_i}, i \in \{1, \dots, k\}. \tag{2.54}$$

Recall now, that the $\phi_i$ must sum up to 1; hence we have

$$1 = \sum_{j=1}^{k} \phi_j = \phi_k \sum_{j=1}^{k} e^{\eta_j} \quad \Leftrightarrow \quad \phi_k = \frac{1}{\sum_{j=1}^{k} e^{\eta_j}} \quad \Leftrightarrow \quad \phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}. \tag{2.55}$$

The last expression is called the softmax function and is heavily used in deep learning as a so-called activation function for classification tasks. Assume, that a node in a deep neural network has $k$ inputs $\{\eta_1, \dots, \eta_k\}$. Acting with the softmax function on these inputs renders every input to a probability $\phi_i = e^{\eta_i} \left(\sum_{j=1}^{k} e^{\eta_j}\right)^{-1}$ of belonging to the $i$-th class. Note now, that it has been shown that $-\ln(\phi_k)$ can be written as a function of $\eta$. Hence, if $I^\top \doteq (1, \dots, 1)$ is a vector of $k-1$ entries which are all ones, we can define

$$a(\eta) \doteq -\ln(\phi_k) = -\ln\left(\frac{1}{\sum_{j=1}^{k} e^{\eta_j}}\right) = -\ln\left(\frac{1}{I^\top \eta}\right). \tag{2.56}$$

Furthermore defining $b(y) = b \doteq 1$, the probability density $p(y|x;\theta)$ can indeed be written in the form

$$p(y|x;\theta) = b(y)e^{\eta^\top T(y) - a(\eta)} \tag{2.57}$$

which proves, that said probability density belongs to the exponential family. So we can proceed with setting $\eta \doteq \theta x$, where $\theta$ now is a matrix

$$\theta = \begin{pmatrix} \theta_1^\top \\ \vdots \\ \theta_{k-1}^\top \end{pmatrix} \quad \text{such that} \quad \eta_i = \theta_i^\top x \tag{2.58}$$

consisting of vectors $\theta_1, \ldots, \theta_{k-1}$, each of dimension $n$. What remains is to calculate the hypothesis function $h_\theta(x)$. So we evaluate

$$h_\theta(x) = \mathbb{E}[T(y)|x] = \sum_{i=1}^{k} T(y=i)p(y=i|x;\theta) = \sum_{i=1}^{k} T(y=i)\phi_i = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{k-1} \end{pmatrix}. \tag{2.59}$$

For every $\phi_i$ for $i \in \{1, \ldots, k-1\}$ we have

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} = \frac{e^{\theta_i^\top x}}{1 + \sum_{j=1}^{k-1} e^{\theta_j^\top x}}, \tag{2.60}$$

where $\eta_k = 0$ and where $\phi_k = 1 - \sum_{j=1}^{k-1} \phi_j$.

So finally, one can achieve the optimal model parameters $\hat{\theta}$ by the maximum likelihood approach as

$$\begin{aligned}
\hat{\theta} &= \arg\max_\theta \Big( p(y^{(1)}, \ldots, y^{(m)} | x^{(1)}, \ldots, x^{(m)}; \theta) \Big) \overset{\text{IID}}{=} \arg\max_\theta \left( \sum_{i=1}^{m} \ln \Big[ p(y^{(i)}|x^{(i)};\theta) \Big] \right) \\
&= \arg\max_\theta \left( \sum_{i=1}^{m} \ln \Big[ \phi_1^{\mathbb{I}\{y^{(i)}=1\}} \cdot \ldots \cdot \phi_k^{\mathbb{I}\{y^{(i)}=k\}} \Big] \right) \\
&= \arg\max_\theta \left( \sum_{i=1}^{m} \sum_{j=1}^{k} \mathbb{I}\{y^{(i)} = j\} \ln(\phi_j) \right) \\
&= \arg\max_\theta \left( \sum_{i=1}^{m} \left[ \sum_{j=1}^{k-1} \mathbb{I}\{y^{(i)} = j\} \ln \left( \frac{e^{\theta_j^\top x^{(i)}}}{1 + \sum_{l=1}^{k-1} e^{\theta_l^\top x^{(i)}}} \right) + \mathbb{I}\{y^{(i)} = k\} \ln \left( 1 - \sum_{u=1}^{k-1} \phi_u \right) \right] \right)
\end{aligned} \tag{2.61}$$

The final expression with everything plugged in reads as

$$\arg\max_\theta \left( \sum_{i=1}^{m} \left[ \sum_{j=1}^{k-1} \mathbb{I}\{y^{(i)} = j\} \ln \left( \frac{e^{\theta_j^\top x^{(i)}}}{1 + \sum_{l=1}^{k-1} e^{\theta_l^\top x^{(i)}}} \right) + \mathbb{I}\{y^{(i)} = k\} \ln \left( 1 - \sum_{u=1}^{k-1} \left[ \frac{e^{\theta_u^\top x^{(i)}}}{1 + \sum_{l=1}^{k-1} e^{\theta_l^\top x^{(i)}}} \right] \right) \right] \right), \tag{2.62}$$

which can be maximized (or minimized with negative sign) with respect to the model parameters $\theta$.

## 2.4 Generative learning and naive Bayes

26. **Q** — What is the difference between a discriminative and generative approach in machine learning? Explain by using the Bayes theorem!

**A** — Text.

27. **Q** — What is generative learning?

**A** — Text.

28. **Q** — How does Gaussian discriminative analysis (GDA) work?

**A** — Text.

29. **Q** — How does the naive Bayes approach work?

**A** — Text.

30. **Q** — What is Laplace smoothing in the context of the naive Bayes approach?

**A** — Text.

31. **Q** — How does the multinomial event model work?

**A** — Text.

### 2.5 Support vector machines

32. **Q** — What is the general idea behind support vector machines?

**A** — Text.

A note concerning the name of support vector machines: The term "vector" comes from the inputs $x$, which are to be classified to belong either to the class $y = -1$ (negatives) or to the class $y = +1$ (positives). As a matter of fact, mostly the samples $x^{(i)}$ closest to the decision boundary actually decide on the location of it; only those vectors support the decision on the decision boundary - this is where the term "support" comes from.

33. **Q** — How is a support vector machine mathematically described?

**A** — Text.

## 3 Decision trees and ensembles

34. **Q** — ?

**A** — Text.

## 4 Ensemble boosting: Regularization and model selection

35. **Q** — ?

**A** — Text.

# 5 Unsupervised learning

## 5.1 Clustering and k-means

36. **Q** — ?

**A** — Text.

## 5.2 EM and factor analysis

37. **Q** — ?

**A** — Text.

## 5.3 PCA and ICA

38. **Q** — ?

**A** — Text.

# 6 Reinforcement learning

39. **Q** — ?

**A** — Text.