Bachelor Thesis, Mathematics

“Machine Learning using Bayesian statistics and Neural networks”

A thesis submitted in fulfillment of the requirements for the bachelor’s degree in Mathematics

Author: Lukas Prokop
Supervisor: 高山・信毅 ※
Supervisor: Bredies Kristian †

※ University of Kobe, Japan
† University of Graz, Austria

Version: August 14, 2017

Abstract

Machine Learning is a vivid research area. Machine Learning fundamentally changes the idea that programmers mechanically write programs in order to perform tasks such as classification, regression, clustering, density estimation, and model selection. In supervised learning, machines learn by observing test vectors and their desired output. They successively adapt their estimation of the input to return desired outputs for actual data. Validation data is used to verify whether this estimate performs good on input, the machine has not learned about. Recent efforts such as Google DeepMind's AlphaGo or Neural Algorithms of Artistic Style (by Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge) show the success of Machine Learning in real-world applications. This thesis will cover Bayesian and Neural Networks as two branches of Machine Learning.

Bayesian networks consider probabilities as quantification of belief. Bayes' Theorem is used as a very generic tool to establish a relation between the prior, a likelihood, and the posterior belief. For example, consider a set of random variables modelling symptoms of a disease. Probabilistic dependencies between these symptoms exist inherently. For a set of sample patients, the belief in experiencing certain symptoms and a certain disease is specified. The machine can now learn these input-output relationships. In the following for a new patient, the belief in symptoms is provided and the machine returns a belief whether the patient suffers from a certain disease.

Neural networks are another branch of Machine Learning. Neural networks consist of multiple layers of neurons. Input signals traverse these layers and using sophisticated algorithms, neuron weights are adjusted so that neurons learn which input signals belong to which output class. This research area is prominent since the 1970s, when Paul Werbos described the Backpropagation algorithm in is Ph.D. thesis.

This bachelor thesis sums up fundamental theorems and theoretical background of the aforementioned fields. Furthermore it covers technical details of my implementation to recognize mathematical expressions.

Keywords: Machine learning, Bayesian network, Bayesian statistics, Bayes' Theorem, probability theory, Neural networks

Abstract

抽象は日本語で仕上がます…...

Keywords: Machine learning, Bayesian statistics, Bayes’ Theorem, Probability theory, Neural networks

Acknowledgements

I would like to thank my advisor 高山先生 of Kobe University for his continued effort during my studies. Thank you for the valuable input and one year of academic support.

I would also like to thank my Austrian advisor Bredies Kristian for a final revision and grading at University of Graz.

During my year abroad in Japan, I met lot of new people and gained valuable experiences. But I wouldn't have taken the challenge without Martina. Thank you for staying with me and sharing your experiences continuously.

どうもありがとうございました。

All source codes are available at lukas-prokop.at/proj/bakk_kobe and published under terms and conditions of Free/Libre Open Source Software. This document was printed with XƎLATEX in the Andada typeface.

Contents

Chapter 1

Bayesian theory

Probability theory and statistics

σ-algebra

The following mathematical object is necessary in order to define probability properly:

Definition 1 Let X be a set. A σ-algebra is a set Y of subsets of X satisfying:

1. $\emptyset \in Y$
2. $Z \in Y \implies Z^C \in Y$ where Z^C denotes the complement of Z, $X \setminus Z$.
3. $(\bigcup_{i=1}^{\infty} Z_i) \in Y$ where $n \in \mathbb{N}$ and $Z_i \in Y$ where $i = 1, 2, \ldots$.

When X is a finite set, we may limit the third condition to a finite union.

Example 1 Let $X := \{a, b, c, d\}$ and $Z := \{\{a\}\}$. We extend Z to a σ-algebra Y:

$$Y = \{\{\}, \{a, b, c, d\}, \{a\}, \{b, c, d\}\}$$

The notion of the σ-algebra is essential to define the notion of the probability space and the random variable rigorously. In this thesis, our discussion is rigorous when X is a finite set. When X is an infinite set, we need a full general discussion of measure theory. Then our discussion will sometimes be intuitive or informal.

Basic definitions

Probability theory is concerned with random experiments and random phenomena. Probability in its basic form is the fraction of events with a certain outcome to the total number of events.

Definition 2 A *probability space* $(\Omega, \mathcal{A}, \mathbb{P})$ denotes the set of possible outcomes, a set of events, and a map from an element of $\mathcal{A}$ to a real value in $[0, 1]$. $\mathcal{A}$ is a σ-algebra. As elements of $\mathcal{A}$ are sets, we can apply set operations on them. If Ω is a finite space, *probability measure* $\mathbb{P}$ satisfies the following conditions:

$$\mathbb{P}[A] \geq 0 \text{ for } A \in \mathcal{A} \tag{1.1}$$

$$\mathbb{P}[A + B] = \mathbb{P}[A] + \mathbb{P}[B] \text{ for } A, B \in \mathcal{A} \text{ and } A \cap B = \emptyset \tag{1.2}$$

$$\mathbb{P}[\Omega] = 1 \tag{1.3}$$

Property 1.2 implies linearity of the probability measure (for mutually exclusive events A):

$$\mathbb{P}\left[\bigcup A\right] = \sum \mathbb{P}\left[A\right] \tag{1.4}$$

An *event* is any subset a of Ω, hence $a \in \mathcal{A}$. A $\mathbb{Z}$*-valued random variable* R is a map from Ω to $\mathbb{Z}$ such that $R^{-1} \in \mathcal{A}$ for any $z \in \mathbb{Z}$. A $\mathbb{R}$*-valued random variable* R is a map from Ω to $\mathbb{R}$ such that $R^{-1}((r, s]) \in \mathcal{A}$ for any real numbers $r < s$.

Example 2 A coin toss has two possible outcomes, heads (*h*) or tails (*t*). We consider two coin tosses. Then $\Omega = \{(h, h), (h, t), (t, h), (t, t)\}$ and $\mathcal{A}$ is the powerset of Ω (i.e. set of all subsets). Let $\mathbb{P}[A] = \#A/4$, the size of set A divided by 4. Let R be our random variable in $\mathbb{Z}$ defined as result of the first coin toss (1 represents head, 0 represents tails). Then $\mathbb{P}[R = 1] = \mathbb{P}[\{\omega \in \Omega \,|\, R(\omega) = 1\}] = \mathbb{P}[\{(h, h), (h, t)\}] = \frac{2}{4}$.

In the following, we will declare random variables, but won't specify the group explicitly. Either it is obvious from context (because of the numbers, we use) or our statements work for both groups.

Average Value and Expected Value

Definition 3 Let Ω be a finite set and R be an E-valued random variable where E is $\mathbb{Z}$ or $\mathbb{R}$. The *average value* $\overline{R}$ of a random variable R is defined as,

$$\overline{R} := \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} R(a) \tag{1.5}$$

If all outcomes of the sample space Ω are considered, we call $\overline{R}$ the *population mean* (denoted μ), otherwise *sample mean*.

We define the *expected value* of a random variable R as follows and denote it by $\mathbb{E}$.

$$\begin{aligned}\mathbb{E}[R] &:= \mu = \sum_{a\in\mathcal{A}} \mathbb{P}[a]\cdot R(a) \qquad (1.6)\\ &= \sum_{z\in E} \mathbb{P}[R=z]\cdot z\end{aligned}$$

where $\mathbb{P}[R = z]$ is the probability of the random variable R taking the value z. In other words, $\mathbb{P}[R = z] := \mathbb{P}[A_z]$ with $A_z := \{e \in \Omega \mid R(e) = z\}$.

Let R and S be two random variables and $c \in \mathbb{R}$. The following properties are satisfied:

$$\begin{aligned}
\mathbb{E}[c] &:= c && (1.7)\\
\mathbb{E}[R+c] &:= \sum_{a\in\mathcal{A}} (\mathbb{P}[R=a]\cdot(a+c))\\
&= \sum_{a\in\mathcal{A}} (\mathbb{P}[R=a]\cdot a) + \sum_{a\in\mathcal{A}} (\mathbb{P}[R=a]\cdot c)\\
&= \mathbb{E}[R] + c\cdot\sum_{a\in\mathcal{A}} \mathbb{P}[R=a]\\
&= \mathbb{E}[R] + c\cdot 1 = \mathbb{E}[R]+c && (1.8)\\
\mathbb{E}[R+S] &:= \sum_{a\in(\mathcal{A}_R\cup\mathcal{A}_S)} \begin{cases}\mathbb{P}[R=a]\cdot a & \text{if } a\in A_R\\ \mathbb{P}[S=a]\cdot a & \text{if } a\in A_S\end{cases}\\
&= \sum_{a\in\mathcal{A}_R} \mathbb{P}[R=a]\cdot a + \sum_{a\in\mathcal{A}_S} \mathbb{P}[S=a]\cdot a\\
&= \mathbb{E}[R]+\mathbb{E}[S] && (1.9)\\
\mathbb{E}[c\cdot R] &:= \sum_{a\in\mathcal{A}} \mathbb{P}[R=a]\cdot(c\cdot a)\\
&= c\cdot\sum_{a\in\mathcal{A}} \mathbb{P}[R=a]\cdot a = c\cdot\mathbb{E}[R] && (1.10)
\end{aligned}$$

Equivalence of the continuous probability model

Definition 4 Let R be an $\mathbb{R}$-valued random variable and f be a continuous function defined in $(-\infty,\infty) \subseteq \mathbb{R}$. Let f satisfy the following properties:

$$\mathbb{P}[R\le y] := \int_{-\infty}^{y} f(x)\,dx \qquad \mathbb{P}[z\le R\le y] := \int_{z}^{y} f(x)\,dx \qquad \mathbb{P}[z\le R] := \int_{z}^{\infty} f(x)\,dx$$

This establishes a relation between function f and random variable R. Because R satisfies properties 1.1 and 1.3 of probability measures, f also satisfies:

$$f(x)\ge 0 \qquad \forall x\in(-\infty,\infty) \qquad (1.11)$$

$$1 = \int_{-\infty}^{\infty} f(x)\,dx \tag{1.12}$$

f is called a *Probability Density Function* (PDF).

Example 3

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The normal distribution function depending on parameters μ and σ^2 is one example for a probability density function. The function is introduced in Section 1.2.1 in detail.

We have seen that the continuous model follows equivalent properties. The same is true for the expected value.

Definition 5 The *expected value* in the continuous model is defined as,

$$\mathbb{E}[R] := \int_{-\infty}^{\infty} \left(\mathbb{P}[R = x] \cdot x\right)\, dx = \int_{\mathbb{R}} \left(\mathbb{P}[R = x] \cdot x\right)\, dx \tag{1.13}$$

where $z \in E$ where E is $\mathbb{Z}$ or $\mathbb{R}$ (group of the random variable).

Proof 1 Let R and S be two random variables and $c \in \mathbb{R}$. The expected value satisfies:

$$\begin{aligned}
\mathbb{E}[c] &:= c && (1.14)\\
\mathbb{E}[R + c] &:= \int_{\mathbb{R}} \mathbb{P}[R = x] \cdot (x + c)\, dx \\
&= \int_{\mathbb{R}} \left(\mathbb{P}[R = x] \cdot x + \mathbb{P}[R = x] \cdot c\right)\, dx \\
&= \int_{\mathbb{R}} \mathbb{P}[R = x] \cdot x\, dx + c \cdot \int_{\mathbb{R}} \mathbb{P}[R = x]\, dx \\
&= \mathbb{E}[R] + c \cdot 1 = \mathbb{E}[R] + c && (1.15)\\
\mathbb{E}[R + S] &:= \int_{\mathbb{R}} \left(\mathbb{P}[R = x] \cdot x + \mathbb{P}[S = x] \cdot x\right)\, dx \\
&= \int_{\mathbb{R}} \mathbb{P}[R = x] \cdot x\, dx + \int_{\mathbb{R}} \mathbb{P}[S = x] \cdot x\, dx \\
&= \mathbb{E}[R] + \mathbb{E}[S] && (1.16)\\
\mathbb{E}[c \cdot X] &:= \int_{\mathbb{R}} \mathbb{P}[R = x] \cdot (x \cdot c)\, dx \\
&= c \cdot \int_{\mathbb{R}} \mathbb{P}[R = x] \cdot x\, dx = c \cdot \mathbb{E}[X] && (1.17)
\end{aligned}$$

Because $\mathbb{P}$ and $\mathbb{E}$ provide the same properties in the discrete and continuous case, we often do not distinguish between these cases. The statements hold true for $\mathbb{Z}$-valued as well as $\mathbb{R}$-valued random variables.

Variance and standard deviation

Definition 6 *Variance* quantifies how strong values are spread out from $\mathbb{E}[R]$:

$$\sigma^2 := \mathbb{E}\left[(R - \mathbb{E}[R])^2\right]$$

Considering the entire population, the variance can also quantify over the population mean μ:

$$\sigma^2 = \mathbb{V}[R] := \mathbb{E}[(R-\mu)^2]$$

In the discrete case, this is equivalent to,

$$\mathbb{V}[R] = \mathbb{E}\left[\sum_{a\in\mathcal{A}} (\mathbb{P}[R=a]\cdot(a-\mu))^2\right]$$

and in the continuous case, we have:

$$\mathbb{V}[R] = \mathbb{E}\left[\int_{\mathbb{R}} (\mathbb{P}[R=x]\cdot(x+\mu))^2 \; dx\right]$$

Definition 7 The *standard deviation* is defined as its second root:

$$\sigma = \sqrt{\mathbb{V}[R]}$$

Covariance

Covariance measures the joint variability of two given random variables. It is defined as:

$$\begin{aligned}\text{Cov}[X,Y] &= \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])] && (1.18)\\ &= \mathbb{E}[XY - Y\cdot\mathbb{E}[X] - X\cdot\mathbb{E}[Y] + \mathbb{E}[X]\mathbb{E}[Y]] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\cdot\mathbb{E}[Y] && (1.19)\end{aligned}$$

If X and Y are independent (compare with Section 1.1.10), then the covariance is zero.

$$\text{Cov}[X,Y] = \mathbb{E}[XY] - \mathbb{E}[X]\cdot\mathbb{E}[Y] = \mathbb{E}[X]\cdot\mathbb{E}[Y] - \mathbb{E}[X]\cdot\mathbb{E}[Y] = 0 \tag{1.20}$$

We will exploit the following properties:

$$\text{Cov}[X,Y] = \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])] = \mathbb{E}[(Y-\mathbb{E}[Y])(X-\mathbb{E}[X])] = \text{Cov}[Y,X] \tag{1.21}$$

$$\text{Cov}[X,X] = \mathbb{E}[(X-\mathbb{E}[X])^2] = \mathbb{V}[X] \tag{1.22}$$

Let X be a set of n independent variables $X_{1\le i\le n}$. Then it holds that:

$$\mathbb{V}\left[\sum_{i=1}^{n} X_i\right] = \mathbb{E}\left[\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]\right)^2\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} (X_i - \mathbb{E}\left[X_i\right])\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n} (X_i - \mathbb{E}[X_i]) \cdot \sum_{j=1}^{n} (X_i - \mathbb{E}[X_i])\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{n} \left(\sum_{i=1}^{n} (X_i - \mathbb{E}[X_i])\right) (X_j - \mathbb{E}[X_j])\right] \\
&= \mathbb{E}\left[\sum_{i,j\in[1,n]} (X_i - \mathbb{E}[X_i])\,(X_j - \mathbb{E}[X_j])\right] \\
&= \sum_{i,j\in[1,n]} \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] \\
&= \sum_{i,j\in[1,n]} \mathsf{Cov}[X_i, X_j] \\
&= \sum_{\substack{i\neq j \\ i,j\in[1,n]}} \mathsf{Cov}[X_i, X_i] + \sum_{\substack{i\neq j \\ i,j\in[1,n]}} \mathsf{Cov}[X_i, X_j] \\
&= \sum_{i=1}^{n} \mathbb{V}[X_i] + \sum_{\substack{i\neq j \\ i,j\in[1,n]}} \mathsf{Cov}[X_i, X_j] \\
&= \sum_{i=1}^{n} \mathbb{V}[X_i] + 2 \cdot \sum_{1\leq i<j\leq n} \mathsf{Cov}[X_i, X_j] \qquad (1.23)
\end{aligned}
$$

Law of Large Numbers

The Law of Large Numbers stresses the practical importance of the expected value.

Theorem 1 (Law of Large Numbers) First, we define the notion of the average value over a sample A_i of size n:

$$\overline{R}_n := \frac{1}{n}\sum_{i=0}^{n} A_i$$

Then, the Law of Large Numbers states that,

$$\lim_{n\to\infty} \overline{R}_n = \mathbb{E}[R]$$

In order to prove this theorem, we use Chebyshev's Inequality, the Weak Law of Large numbers, Borel-Cantelli Lemma and the Strong Law of Large Numbers. The latter is considered equivalent to the Law of Large numbers [1]. Our proof structure is based on Craig A. Tracy's [14][2].

Chebyshev's Inequality

Consider the continuous case. Let R be a random variable, f be a PDF over R, $|\cdot|$ be a norm, $a \in \mathbb{R}_{\geq 0}$, $p \in \mathbb{N}$ and let $\mathbb{E}[R^p]$ be defined as follows:

$$\mathbb{E}[R^p] = \int\limits_{\mathbb{R}} x^p \cdot f(x)\,dx \geq \int\limits_{x \geq a} x^p \cdot f(x)\,dx \geq a^p \int\limits_{x \geq a} f(x)\,dx = a^p \cdot \mathbb{P}[R \geq a]$$

The discrete case follows immediately. This concludes the correctness of the following theorem:

Theorem 2 (Chebyshev's Inequality Theorem) Let R be a random variable, $a \in \mathbb{R}_{\geq 0}$ and $p \in \mathbb{N}$ is arbitrary. Assume $\mathbb{E}[R^p] < \infty$. Then it holds that,

$$\mathbb{P}[R \geq a] \leq \frac{1}{a^p}\mathbb{E}[R^p]$$

Weak Law of Large Numbers

The next theorem is called Weak Law of Large Numbers.

Theorem 3 (Weak Law of Large Numbers, Bernoulli's Theorem) Let R_i be a sequence of independent and identically distributed random variables (see section 1.2.3 for a definition of i.i.d.) with common mean μ and variance σ^2. Let

$$S_n := \sum_{i=1}^{n} R_i \qquad\qquad T_n := \frac{S_n}{n} - \mu$$

Then for any $\varepsilon > 0$,

$$\lim_{n\to\infty} \mathbb{P}[T_n \geq \varepsilon] = 0$$

Proof 2 First, we determine the expected values.

$$\mathbb{E}[S_n] = \mathbb{E}\left[\sum_{i=1}^{n} R_i\right] = \sum_{i=1}^{n}\mathbb{E}[R_i] = \sum_{i=1}^{n}\mu = n\cdot\mu \qquad (1.16) \qquad (1.24)$$

$$\mathbb{E}[T_n] = \mathbb{E}\left[\frac{S_n}{n} - \mu\right] = \frac{\mathbb{E}\left[S_n\right]}{\mathbb{E}\left[n\right]} - \mathbb{E}\left[\mu\right] \qquad (1.16)$$

[1] Depending on your requirements of certainty, the Weak Law of Large numbers might be already considered equivalent to the Law of Large Numbers (Theorem 1), but we look for the Strong Law of Large Numbers in this thesis.

[2] Please recognize that there is a small typographical error on page 3. "$S_n(\omega) = 1$ for every n" should be "$X_n(\omega) = 1$ for every n".

$$= \frac{n \cdot \mu}{n} - \mu = 0 \qquad \text{(1.24), (1.17)} \tag{1.25}$$

We also need a result regarding the variance. In the continuous and discrete case, it holds that $a^2 \cdot \mathbb{V}[X] = \mathbb{V}[a \cdot X]$:

$$\begin{align*}
a^2 \cdot \mathbb{V}[X] &= a^2 \cdot \int (x-\mu)^2 \cdot f(x)\, dx = \int (x \cdot a - \mu \cdot a)^2 \cdot f(x)\, dx \\
a^2 \cdot \mathbb{V}[X] &= a^2 \cdot \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2 = \sum_{i=1}^{n} p_i \cdot a^2 \cdot (x_i - \mu)^2 = \sum_{i=1}^{n} p_i \cdot (x_i \cdot a - \mu \cdot a)^2 \tag{1.26}
\end{align*}$$

The relation $\mathbb{E}[X^2] = \mathbb{V}[X] + \mathbb{E}[X]^2$ holds as well,

$$\begin{align*}
\mathbb{V}[X] &= \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2] - \mathbb{E}[2X\mu] + \mathbb{E}[\mu^2] && \text{(1.16)} \\
&= \mathbb{E}[X^2] - 2 \cdot \mathbb{E}\left[X \cdot \mathbb{E}[X]\right] + \mathbb{E}\left[\mathbb{E}[X]^2\right] && \text{(1.17)} \\
&= \mathbb{E}[X^2] - 2 \cdot \mathbb{E}[X]^2 + \mathbb{E}[X]^2 && \text{(1.17)} \\
&= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \tag{1.27}
\end{align*}$$

We use this result to prove $\mathbb{V}[T_n] = \frac{\sigma^2}{n}$.

$$\begin{align*}
\mathbb{V}[T_n] &= \mathbb{E}[T_n^2] - \mathbb{E}[T_n]^2 = \mathbb{E}[T_n^2] - 0^2 = \mathbb{E}[T_n^2] && \text{(1.27)(1.25)} \\
&= \mathbb{E}\left[\left(\frac{S_n}{n} - \mu\right)^2\right] = \mathbb{E}\left[\left(\frac{S_n}{n}\right)^2 - 2\frac{S_n}{n}\mu + \mu^2\right] \\
&= \mathbb{E}\left[\left(\frac{S_n}{n}\right)^2\right] - 2 \cdot \mathbb{E}\left[\frac{\mu}{n} S_n\right] + \mathbb{E}\left[\mu^2\right] && \text{(1.16)(1.17)} \\
&= \mathbb{V}\left[\frac{S_n}{n}\right] + \mathbb{E}\left[\frac{S_n}{n}\right]^2 - 2\mu \cdot \mathbb{E}\left[\frac{S_n}{n}\right] + \mu^2 && \text{(1.27)(1.14)} \\
&= \frac{1}{n^2} \cdot \mathbb{V}\left[S_n\right] + \left(\frac{1}{n} \cdot \mathbb{E}\left[S_n\right]\right)^2 - \frac{2\mu}{n} \cdot (n \cdot \mu) + \mu^2 && \text{(1.17)(1.26)} \\
&= \frac{1}{n^2} \cdot \left(\mathbb{V}\left[S_n\right] + \mathbb{E}[S_n]^2\right) - 2\mu^2 + \mu^2 \\
&= \frac{1}{n^2}\left(\mathbb{V}\left[\sum_{i=1}^{n} R_i\right] + \mathbb{E}[S_n]^2\right) - \mu^2 \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \mathbb{V}[R_i] + 2 \sum_{\substack{i<j \\ i,j=1}}^{n} \text{Cov}[R_i, R_j] + \mathbb{E}[S_n]^2\right) - \mu^2 && \text{(1.23)} \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \sigma^2 + 0 + (n \cdot \mu)^2\right) - \mu^2 && \text{(1.24)(1.20)} \\
&= \frac{1}{n^2}\left(n \cdot \sigma^2 + n^2 \cdot \mu^2\right) - \mu^2 = \frac{\sigma^2}{n} + \mu^2 - \mu^2 = \frac{\sigma^2}{n} \tag{1.28}
\end{align*}$$

Furthermore, the following equation holds:

$$\mathbb{V}[T_n] = \mathbb{E}\left[(T_n - \mathbb{E}[T_n])^2\right] = \mathbb{E}\left[(T_n - 0)^2\right] = \mathbb{E}\left[T_n^2\right] \tag{1.29}$$

Now we can apply Chebyshev's Inequality Theorem ($R = T_n, a = \varepsilon \in \mathbb{R}, p = 2$):

$$\mathbb{P}[|T_n| \geq \varepsilon] \leq \frac{1}{\varepsilon^2}\mathbb{E}\left[|T_n|^2\right] = \frac{1}{\varepsilon^2}\mathbb{V}\left[|T_n|\right] = \frac{1}{\varepsilon^2}\frac{\sigma^2}{n} \qquad (1.29) \qquad (1.30)$$

For any $\varepsilon > 0$ with $n \to \infty$, it holds that

$$\mathbb{P}[|T_n| \geq \varepsilon] \to 0$$

$$\Leftrightarrow \forall \varepsilon > 0 : \lim_{n\to\infty} \mathbb{P}[|T_n| \geq \varepsilon] = 0$$

This concludes the proof of the Weak Law of Large Numbers. In order to finish our proof of the Strong Law of Large numbers, we will use the Borel-Cantelli Lemma, an important result of measure theory.

Borel-Cantelli Lemma

Lemma 1 (Borel-Cantelli Lemma) Let R_i with $1 \leq i < \infty$ be a sequence of events. Assume the sum of these probabilities is finite, then it holds that:

$$\sum_{i=1}^{\infty} \mathbb{P}[R_i] < \infty \implies \mathbb{P}\left(\limsup_{i\to\infty} R_i\right) = 0 \qquad (1.31)$$

So the probability, that the occuring event is an event which occurs infinitely often, is 0.

Proof 3 Please consider, that the limsup is defined as,

$$\limsup_{i\to\infty} R_i := \bigcap_{j=1}^{\infty} \bigcup_{i\geq j}^{\infty} R_i \qquad (1.32)$$

The condition requires, that $\sum_{i=1}^{\infty} \mathbb{P}[R_i] < \infty$. This statement is equivalent to

$$\inf_{j\geq 1} \sum_{i=j}^{\infty} \mathbb{P}[R_i] = 0 \qquad (1.33)$$

We can make our final conclusion to prove the Borel-Cantelli Lemma:

$$\begin{aligned} \mathbb{P}\left[\limsup_{i\to\infty} R_i\right] &= \mathbb{P}\left[\bigcap_{j=1}^{\infty} \bigcup_{i=j}^{\infty} R_i\right] && (1.32) \\ &\leq \inf_{j\geq 1} \mathbb{P}\left[\bigcup_{i=j}^{\infty} R_i\right] && (1.4) \end{aligned}$$

$$\begin{aligned}&\leq \inf_{j\geq 1}\sum_{i=j}^{\infty}\mathbb{P}\left[R_i\right]\\&= 0 \qquad (1.33) \qquad (1.34)\end{aligned}$$

The result of an intersection of elements creates a set, which is an actual subset in any of these sets. However, the infimum is not necessarily an element of the set. Hence, an inequality is introduced in the second line.

Strong Law of Large Numbers

Theorem 4 (Strong Law of Large Numbers) Assume the definitions of Theorem 3. Therefore, $R_1, R_2, \ldots$ is an infinite sequence of independent random variables with a common distribution ($\mu = \mathbb{E}[R_j]$, $\sigma^2 = \mathbb{V}[R_j]$). S_n and T_n are defined. Now consider event $\mathcal{E}$:

$$\mathcal{E} = \left\{\omega \in \Omega : \lim_{n\to\infty}\frac{S_n(\omega)}{n} = \mu\right\}$$

Then it holds that

$$\mathbb{P}[\mathcal{E}] = 1$$

Proof 4 The following proof assumes $\sigma^2 = \mathbb{E}[R_j^2] < \infty$ and $\mathbb{E}[R_j^4] < \infty$. This restriction makes our proof easier, but the less restricted case $\mathbb{E}[R_j] < \infty$ suffices as assumption (but this is not proven in this thesis).

Without loss of generality we assume $\mu = 0$.
If $\mu = 0$ is not satisfied, we consider $P_j \coloneqq R_j - \mu$ instead.

Now, we want to give a brief outline of the proof. If it holds that,

$$\lim_{n\to\infty}\frac{S_n(\omega)}{n} \neq 0$$

then $\exists \varepsilon \in \mathbb{R}$ with $\varepsilon > 0$ such that for infinitely many n

$$\frac{S_n(\omega)}{n} > \varepsilon$$

So to prove the theorem, we will prove that for every $\varepsilon > 0$,

$$\mathbb{P}[S_n > n \cdot \varepsilon \text{ infinitely often}] = 0$$

In the following, this reveals that

$$\mathbb{P}[\mathcal{E}] = \mathbb{P}\left[\frac{S_n}{n} = 0\right] = 1$$

proving Theorem 4. Hence condition $\frac{S_n}{n} = 0$ holds with probability 1.

First of all, we define

$$A_n = \{\omega \in \Omega : S_n \geq n \cdot \varepsilon\}$$

and look at $\mathbb{P}[A_n]$ using the Chebyshev inequality (Theorem 2) with $p = 4$ and $a = n \cdot \varepsilon$:

$$\mathbb{P}[S_n \geq (n \cdot \varepsilon)] \leq \frac{1}{(n \cdot \varepsilon)^4} \mathbb{E}[S_n^4]$$

We must determine $\mathbb{E}[S_n^4]$ which equals to

$$\begin{aligned} \mathbb{E}\left[\sum_{1 \leq i,j,k,l \leq n} R_i R_j R_k R_l\right] &= \mathbb{E}\left[\sum_{1 \leq i}^{n} \sum_{1 \leq j}^{n} \sum_{1 \leq k}^{n} \sum_{1 \leq l}^{n} R_i R_j R_k R_l\right] \\ &= \mathbb{E}\left[(R_1^4 + \ldots + R_1 R_n^3) + (R_2 \ldots) + (R_3 \ldots) + (R_n \ldots + R_n^4)\right] \end{aligned}$$

Because $\mathbb{E}[R_i] = 0$, we can remove all terms containing R_j of degree 1 for any j. These are terms of the structure (assuming i, j, k and l distinct),

$$\mathbb{E}[R_i^3 R_j],\ \mathbb{E}[R_i^2 R_j R_k],\ \mathbb{E}[R_i R_j R_k R_l]$$

Remember that multiplication of expected values (compare with Equation 1.20) applies here. The non-zero terms are $\mathbb{E}[R_i^4]$ and $\mathbb{E}[R_i^2 R_j^2] = \left(\mathbb{E}[R_i^2]\right)^2$. Now, we need to quantify the occurences of these non-zero terms. $\mathbb{E}[R_i^4]$ occurs n times. Terms $\mathbb{E}[R_i^2 R_j^2]$ occur $3n \cdot (n-1)$ times, as there are $\frac{(n-1) \cdot n}{2}$ ways to choose 2 indices and 6 ways to find $R_i^2 R_j^2$. In conclusion, we determined,

$$\mathbb{E}\left[S_n^4\right] = n \cdot \mathbb{E}\left[R_1^4\right] + 3n \cdot (n-1) \cdot \sigma^4 = n \cdot \left(\mathbb{E}\left[R_1^4\right] + 3n \cdot \sigma^4 - 3\sigma^4\right)$$

In this expression, n is our only constant occuring with polynomial degree 2. So for any n sufficiently large, there exists $C \in \mathbb{R}$ such that

$$\begin{aligned} 3\sigma^4 n^2 + \left(\mathbb{E}\left[R_1^4\right] - 3\sigma^4\right) \cdot n \leq C \cdot n^2 \qquad (1.35) \\ \Rightarrow \mathbb{E}\left(S_n^4\right) \leq C n^2 \end{aligned}$$

We return back to Chebyshev's inequality

$$\mathbb{P}[S_n \geq (n \cdot \varepsilon)] \leq \frac{1}{(n \cdot \varepsilon)^4} \mathbb{E}[S_n^4] \leq \frac{C \cdot n^2}{\varepsilon^4 \cdot n^2 \cdot n^2}$$

It follows that, there exists some n_0 such that Equation (1.35) is satisfied. With this approach, we skip a finite number of elements of the sum. This does not affect its convergence or divergence.

$$\sum_{n \geq n_0} \mathbb{P}\left[S_n \geq n \cdot \varepsilon\right] \leq \sum_{n \geq n_0} \frac{C}{\varepsilon^4 n^2} < \infty$$

Therefore, the conditions to apply the Borel-Cantelli Lemma (Lemma 1) are satisfied. For every $\varepsilon > 0$ it holds that,

$$\mathbb{P}\left[S_n \geq n\varepsilon \text{ infinitely often}\right] = 0$$

Marginalization

Definition 8 Let R and S be two random variables. Then we define,

$$\mathbb{P}[R = r, S = s] := \mathbb{P}[\{\omega \in \Omega \mid R(\omega) = r \wedge S(\omega) = s\}]$$

This definition enables us to define *marginalization*:

$$\mathbb{P}[R = r] = \sum_{s \in S} \mathbb{P}[R = r, S = s]$$

Example 4 Please consider our coin tossing example again. Let R be our random variable in $\mathbb{Z}$ defined as result of the first coin toss (1 represents head, 0 represents tails). $\mathbb{Z}$-valued random variable S is defined as result of the second coin toss. Then

$$\mathbb{P}[R = 1, S = 0] = \mathbb{P}[\{\omega \in \Omega \mid R(\omega) = 1 \wedge S(\omega) = 0\}] = \mathbb{P}[\{(h, t)\}] = \frac{1}{4}$$

Marginalization applies, if we query $\mathbb{P}[R = 1]$:

$$\mathbb{P}[R = 1] = \mathbb{P}[R = 1, S = 0] + \mathbb{P}[R = 1, S = 1] = \mathbb{P}[\{(h, t)\}] + \mathbb{P}[\{(h, h)\}] = \frac{1}{4} + \frac{1}{4} = \frac{2}{4}$$

Joint distribution

Definition 9 Let R and S be two random variables. Joint distribution is given by the following definition:

$$\mathbb{P}[R = r, S = s] = \mathbb{P}[R = r] \cdot \mathbb{P}[S = s]$$

It is important to recognize, that we assume conditional independence of events as defined in Section 1.1.10.

Example 5 In our coin tossing example, we have that:

$$\mathbb{P}[R = 1, S = 0] = \mathbb{P}[\{(h, t), (h, h)\}] \cdot \mathbb{P}[\{(t, t), (h, t)\}] = \frac{2}{4} \cdot \frac{2}{4} = \frac{1}{2}$$

In the Marginalization example, we used the same query and used a different approach to get the same result.

independence

Definition 10 We assume Ω is a finite set and consider the probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Two $\mathbb{Z}$-valued random variables R and S are called *(mutually) independent* if

$$\mathbb{P}[R = r, S = s] = \mathbb{P}[R = r] \cdot \mathbb{P}[S = s]$$

for any $r, s \in \mathbb{Z}$ (see, e.g., [10, p. 27], [8, p. 143]).

Let A be an element of the σ-algebra $\mathcal{A}$. We denote by 1_A the indicator function of A defined as

$$1_A(a) = \begin{cases} 1 & \text{if } a \in A \\ 0 & \text{if } a \notin A \end{cases}$$

The indicator function 1_A defines a $\mathbb{Z}$-valued random variable. Let A and B be elements of the σ-algebra $\mathcal{A}$. The sets (events) A and B are called independent when the random variables 1_A and 1_B are independent.

> **Example 6** Consider our previous coin tossing example (compare with Example 2). Let us denote coin tosses ij, where ij are the results of the first and second coin toss respectively. Define the random variable R as
>
> $$R(00) = 0, R(01) = 0, R(10) = 1, R(11) = 1$$
>
> and define the random variable S (intuitively the result of the second toss) as
>
> $$S(00) = 0, S(01) = 1, S(10) = 0, S(11) = 1$$
>
> Then we have
>
> $$\mathbb{P}(R = 0, S = 0) = \mathbb{P}(R = 0, S = 1) = \mathbb{P}(R = 1, S = 0) = \mathbb{P}(R = 1, S = 1) = 1/4$$
>
> Since $\mathbb{P}(R = i) = \frac{1}{2}$ and $\mathbb{P}(S = j) = \frac{1}{2}$, we can see $\mathbb{P}(R = i, S = j) = \mathbb{P}(R = i) \cdot \mathbb{P}(S = j)$. Therefore R and S are independent random variables. The elements of σ algebra $A = \{01, 11\} = R^{-1}(1)$ and $B = \{10, 11\} = S^{-1}(1)$ are independent. Intuitively speaking, the event getting 1 at the first toss and the event getting 1 at the second toss are independent.

Let R, S, T be $\mathcal{Z}$-valued random variables. The random variables R and S are conditionally independent under the given random variable T when

$$\mathbb{P}[R = r, S = s \,|\, T = t] = \mathbb{P}[R = r \,|\, T = t] \cdot \mathbb{P}[S = s \,|\, T = t]$$

for any $r, s, t \in \mathcal{Z}$ (see, e.g., [7, p. 3.1]). Let

$$A_r = R^{-1}(r) \subset \Omega \qquad B_s = S^{-1}(s) \subset \Omega \qquad C_t = T^{-1}(t) \subset \Omega$$

Then $\mathbb{P}[A_r], \mathbb{P}[B_s], \mathbb{P}[C_t]$ can be regarded as probability distribution functions associated to R, S, T respectively. We note that we have

$$\begin{aligned} \mathbb{P}[R = r, S = s \,|\, T = t] &= \frac{\mathbb{P}[A_r \cap B_s \cap C_t]}{\mathbb{P}[C_t]} \\ \mathbb{P}[R = r \,|\, T = t] &= \frac{\mathbb{P}[A_r \cap C_t]}{\mathbb{P}[C_t]} \\ \mathbb{P}[S = r \,|\, T = t] &= \frac{\mathbb{P}[B_r \cap C_t]}{\mathbb{P}[C_t]} \end{aligned}$$

by the definition of the conditional probability.

ToDo: definition 3+ variables case

ToDo: mutual independence $\neq$ pairwise independence

Bayes' Theorem

Theorem 5 (Bayes' Theorem) Let A and B be two events and $\mathbb{P}[B] \neq 0$. Then:

$$\mathbb{P}[A|B] = \frac{\mathbb{P}[B|A] \cdot \mathbb{P}[A]}{\mathbb{P}[B]}$$

Proof 5 In section 1.1.10, we showed the following relation between marginal and conditional probability:

$$\mathbb{P}[A, B] = \mathbb{P}[B|A] \cdot \mathbb{P}[A] = \mathbb{P}[A|B] \cdot \mathbb{P}[B]$$

Bayes' theorem follows immediately:

$$\frac{\mathbb{P}[B|A] \cdot \mathbb{P}[A]}{\mathbb{P}[B]} = \mathbb{P}[A|B]$$

Bayes' Theorem is fundamental to theory we will cover in the following. $\mathbb{P}[A]$ is called *prior probability* and $\mathbb{P}[A|B]$ is called *posterior probability* in the Bayesian interpretation. The names derive from the fact, that $\mathbb{P}[A]$ is known beforehand in most applications and $\mathbb{P}[A|B]$ is the degree of belief in A after B happened.

Probability distributions

Probability distributions are templates for probability density functions satisfying the criteria mentioned in Section 1.1.4. They are parameterized by one or more variables and can be continuous or discrete.

Normal distribution

ToDo: definition

ToDo: visualization

Gaussian distribution

ToDo: definition

ToDo: visualization

Independent and identically distributed

ToDo: definition

ToDo: illustrative example

Graphical models

ToDo: Show symmetry, decomposition, weak union and contraction, via Dawid et al.

Example: Polynomial curve fitting problem

The problem

ToDo: visualization

In the following, we introduce the curve fitting problem similar to [3, p. 4 ff.]. The problem is defined as follows:

Problem 1 (Polynomial curve fitting problem) Consider a polynomial of arbitrary degree.

Given $x = (x_1, \ldots, x_n)^N$ as a vector of N x-values and $t = (t_1, \ldots, t_n)^N$ as the corresponding y-values drawn from the polynomial. Furthermore let $E(w)$ be an error function for given polynomial coefficients w.

Find a polynomial with coefficients w which approximates values t minimizing $E(w)$.

The degree of the polynomial is unknown on purpose. *Model selection* is a branch of Machine Learning dedicated to finding appropriate models for given problems. So for polynomial degree choice for our curve fitting problem, we refer to research literature in Model Selection. **ToDo:** provide useful references for Curve Fitting

Popular error functions include

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n, w) - t_n\right)^2 \qquad \text{(Mean squared error, MSE)}$$

$$E(w) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2} \qquad \text{(Root mean square, RMS)}$$

$$E(w) = \frac{1}{N}\sum_{n=1}^{N}(y(x_n, w) - t_n) \qquad \text{(Mean signed deviation, MSD)}$$

Overfitting

Machine Learning distinguishes between a *training* and *validation* dataset as input. It uses the training set to learn which output is desired for some given input. Therefore all elements of the training set are labelled such that the error in the output can be quantified. *Overfitting* describes the situation, when the learning algorithm approximates the output with little error, but input from the validation set (which contains different inputs) is computed with high error. So the algorithm perfectly adapted itself to recognize the training data, but performs badly for any other input.

ToDo: visualization

Regularization as countermeasure

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2 + \frac{\lambda}{2}\left|w\right|^2$$

We now model the problem from a probabilistic view:

Maximum Likelihood Estimator

The Maximum Likelihood Estimator (MLE) is a technique to estimate the parameters of a probability distribution. It maximizes the likelihood that the given data actually occurs.

ToDo: Illustrate that the Curve Fitting problem is considered Bayesian here

Theorem 6 Consider input data x, mean μ and variance σ^2:

$$\ln \mathbb{P}[x|\mu, \sigma^2] = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

Then $\mu_{\text{ML}} = \frac{1}{N} \cdot \sum_{n=1}^{N} x_n$ for maximized μ and
$\sigma_{\text{ML}} = \frac{1}{N} \cdot \sum_{n=1}^{N} (x_n - \mu_{\text{ML}})^2$ for maximized σ^2

So we want to determine the 2 parameters of a Gaussian distribution, namely μ and σ^2, in the maximum likelihood case. We begin with μ:

Proof 6 1. Derive $\ln \mathbb{P}[x|\mu, \sigma^2]$ for μ

$$\begin{aligned}\frac{\partial}{\partial \mu} \ln \mathbb{P}[x|\mu, \sigma^2] &= \frac{\partial}{\partial \mu} \left(-\frac{1}{2\sigma^2} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \right) \\ &= \frac{\partial}{\partial \mu} \left(-\frac{1}{2\sigma^2} \cdot \sum_{n=1}^{N} (x_n^2 - 2x_n\mu + \mu^2) - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \right) \\ &= -\frac{1}{2\sigma^2} \cdot \sum_{n=1}^{N} (-2x_n + 2\mu) \\ &= -\frac{1}{\sigma^2} \cdot \sum_{n=1}^{N} (\mu - x_n)\end{aligned}$$

2. Set result zero

$$0 = -\frac{1}{\sigma^2} \cdot \sum_{n=1}^{N} (\mu - x_n) = \sum_{n=1}^{N} (\mu - x_n) = N \cdot \mu - \sum_{n=1}^{N} x_n$$

$$\Longrightarrow \mu_{\text{ML}} = \frac{1}{N} \cdot \sum_{n=1}^{N} x_n \qquad \text{commonly called “sample mean”}$$

We continue with σ^2 and use the same approach:

Proof 7 1. Derive $\ln \mathbb{P}[x|\mu, \sigma^2]$ for σ^2

$$\begin{aligned}\frac{\partial}{\partial \sigma^2} \ln \mathbb{P}[x|\mu, \sigma^2] &= \frac{\partial}{\partial \sigma^2} \left(-\frac{1}{2\sigma^2} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \right) \\ &= \frac{1}{2\sigma^4} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 - \frac{N}{2} \cdot \frac{1}{\sigma^2} \\ &= \frac{1}{2\sigma^2} \left(\frac{1}{\sigma^2} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 - N \right)\end{aligned}$$

2. Set result zero

$$0 = \frac{1}{2\sigma^2} \left(\frac{1}{\sigma^2} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 - N \right)$$

$$
\begin{aligned}
N \cdot \sigma^2 &= \sum_{n=1}^{N} (x_n - \mu)^2 \\
\sigma^2_{\text{ML}} &= \frac{1}{N} \cdot \sum_{n=1}^{N} (x_n - \mu)^2 \qquad \text{commonly called "sample variance"}
\end{aligned}
$$

And now we derive the precision parameter β in the maximum likelihood case:

Theorem 7 Given

$$\ln \mathbb{P}[t|x, w, \beta] = -\frac{\beta}{2} \cdot \sum_{n-1}^{N} (y(x_n, w) - t_n)^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

then find

$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N} \cdot \sum_{n=1}^{N} (y(x_n, w_{\text{ML}}) - t_n)^2$$

by maximizing β

Proof 8 1. Derive $\ln \mathbb{P}[t|x, w, \beta]$ with β

$$
\begin{aligned}
\frac{\partial}{\partial \beta} \ln \mathbb{P}[t|x, w, \beta] &= \frac{\partial}{\partial \beta} \left(-\frac{\beta}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \right) \\
&= -\frac{1}{2} \cdot \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 + \frac{N}{2} \cdot \frac{1}{\beta}
\end{aligned}
$$

2. Set result zero

$$
\begin{aligned}
0 &= -\frac{1}{2} \cdot \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 + \frac{N}{2\beta} \\
\frac{N}{\beta} &= \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 \\
\frac{1}{\beta_{\text{ML}}} &= \frac{1}{N} \cdot \sum_{n=1}^{N} (y(x_n, w) - t_n)^2
\end{aligned}
$$

The maximum of the logarithm of an expression corresponds to the minimum of the negative logarithm of the same expression. **ToDo:** so why do we minimize and not maximize?

$$-\log \mathbb{P}[\omega|x, t, \alpha, \beta] \propto -\log \left[\mathbb{P}[t|x, \omega, \beta] \cdot \mathbb{P}[\omega|\alpha] \right] \tag{1.36}$$

due to proportionality, $\exists c \in \mathbb{R}$ such that

(1.37)

$$= -\log \mathbb{P}[t|x,\omega,\beta] - \log \mathbb{P}[\omega|\alpha] - \log c \tag{1.38}$$

insert formula Bishop 1.62

$$\tag{1.39}$$

$$= \frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n\right)^2 - \frac{N}{2}\ln\beta + \frac{N}{2}\ln 2\pi - \log\left(\left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}} \cdot \exp\left(-\frac{\alpha}{2}\omega^T\omega\right)\right) - \log c$$

$$\tag{1.40}$$

$$= \frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n\right)^2 - \frac{N}{2}\ln\beta + \frac{N}{2}\ln 2\pi - \frac{M+1}{2}\log\alpha + \frac{M+1}{2}\log 2\pi + \frac{\alpha}{2}\omega^T\omega - \log c$$

$$\tag{1.41}$$

Let f be any function with a minimum. Then $\arg\min_\omega f(\omega) = \arg\min_\omega c \cdot f(\omega) + a$ for any $c, a \in \mathbb{R}$. This applies also to our case:

$$\begin{aligned}
\arg\min_\omega -\log \mathbb{P}[\omega|x,t,\alpha,\beta] &= \arg\min_\omega \left(\frac{\beta}{2}\cdot\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n\right)^2 + \frac{\alpha}{2}\omega^T\omega\right) && (1.42)\\
&= \arg\min_\omega \beta\left(\frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n\right)^2 + \frac{\frac{\alpha}{\beta}}{2}\omega^T\omega^T\right) && (1.43)\\
&= \arg\min_\omega \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n\right)^2 + \frac{\frac{\alpha}{\beta}}{2}\omega^T\omega^T && (1.44)\\
& && (1.45)
\end{aligned}$$

Hence, the coefficients maximizing the probability that the coefficients correspond to our model parameters (x,t,α,β) are given in the last line. Considering we determine the best coefficients by minimizing the error function, it is justified to consider these coefficients as optimum. Let $\lambda = \frac{\alpha}{\beta}$, then ...

$$\tilde{E}(\omega) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\omega)-t_n^2\right)^2 + \frac{\lambda}{2}\|\omega\|^2$$

Chapter 2

Neural networks

Neural networks are our tool of choice for our implementation project. We want to recognize mathematical expressions and also consider Kanji recognition. Challenges for these applications are especially ambiguity of handwriting and implicit conventions applied to notation. As a result, the quality of such applications varies greatly. This particular implementation cannot compete with industrial applications due to time constraints for this thesis.

Recognizing handwriting recorded with a digitizer as a time sequence of pen coordinates is known as *online character reorganization*. As opposed to *offline handwritten character recognition* dealing with the scanned handwritten document [11]. Our implementation only covers online recognition.

Structure of neural networks

The definitions in this chapter are based on Christopher M. Bishop's book *Pattern Recognition and Machine Learning* [3]. The definition of neural networks is based on Bishop's definition of the multilayer perceptron on page 227 ff. as a result of considerations of linear models. The linear models deal with the following problems:

> **Problem 2 (Regression problem)** Given a training data comprising N observations $\{x_n\}$, where $n = 1, \ldots, N$ together with corresponding target values $\{t_n\}$, the goal is to predict the value of t for a new value of x. [3, p. 138]

> **Problem 3 (Classification problem)** Take an input vector x and assign it to one of K discrete class $\mathcal{C}_k$ where $k = 1, \ldots, K$. [3, p. 179]

The models are based on linear combinations of fixed nonlinear basis functions $\varphi_j(x)$:

$$y(x, w) = f\left(\sum_{j=1}^{M} w_j \varphi_j(x)\right) \tag{2.1}$$

In this case, x represents the input vector, w represents so-called ***weights***, $\varphi_j(x)$ denotes the j-th fixed nonlinear basis function and $f(\cdot)$ is a nonlinear activation function. In the case of regression, $f(\cdot)$ is the identity function.

ToDo: Explain layers

ToDo: Discuss the properties of a neural network of two inputs, one hidden layer, and one output

Neural networks in practice

Curve Fitting Problem in Neural Networks

ToDo: Illustrate the exercises given by Takayama-sensei

Backpropagation

ToDo: Explain the algorithm

Gradient Descent

ToDo: Explain the algorithm and exercises

ToDo: Give convergence proof

Google TensorFlow

Google TensorFlow [1] is an open-source software library for Machine Intelligence. It provides users a choice between a high-level and a low-level API. This enables them to implement a wide variety of Machine Learning applications. We will list some basic application classes distinguished in Machine Learning:

Supervised learning labelled data is available used in training phase, validation possible

- **regression** output values are continuous
- **classification** output values are discrete

Unsupervised learning labelled data unavailable, validation impossible

- **clustering** find groups of similar features
- **density estimation** find distribution of data within input space

For our application, TensorFlow is the tool of choice. We used the Python API to implement our handwriting recognition system.

Terminology

A *glyph* is a visual unit drawn on any surface visible to a potential reader. A *character* is the perceived unit of writing. *Unicode* [6] is an encoding covering most of the world's writing systems. A *Unicode code point* is a unique number assigned to a character.

A problem statement

The problem of handwriting recognition dates back to the early days of machine learning even before 1960 [2]. Since the beginning, it was considered a more convenient and natural way to input text into a machine. This is because most people learn to write with pen and paper before they learn to type text on a keyboard. The requirement of a keyboard for typing is impractical for mobile applications. Instead, the industry developed devices, where a pen or your finger is used as an input interface on a surface tracking your motion. Handwriting recognition has gained attention on handheld devices such as personal digital assistants (PDA). With the recent advent of smartphones after the year 2000, new input devices were used for handwriting. The screens got large enough such that the user can draw the desired characters and applications recognize the glyphs. Even though the quality has improved tremendously since the 1960s, most users still prefer a virtual keyboard over character drawing interfaces.

The MNIST dataset [12] is the most famous dataset to begin an application with. Extremely high accuracies have been obtained [13] to recognize the digits given in this dataset. However, there are many other datasets consisting of less examples and which can be considered more difficult. The aforementioned paper [13] considers Thai, Bangla, and Latin scripts, which employs new challenges considering a high variability in the shapes, strokes, curls, and concavities.

They collected a Thai handwritten script dataset containing 24,045 character images in total from various writers. Figure 2.1 shows examples for the Thai, Bangla, and Latin scripts. Recognize that the Bangla numeral 4 (Example b, row 2, column 5 in Figure 2.1) and the Latin digit 8 (Example b, row 3, column 9) are written the same way. It follows immediately that recognition software must not only recognize individual characters, but has to use context information to determine which script is used. Accuracies above 95 %, 88 % and 60 % for Latin, Thai, and Bangla scripts, respectively using support vector machines.

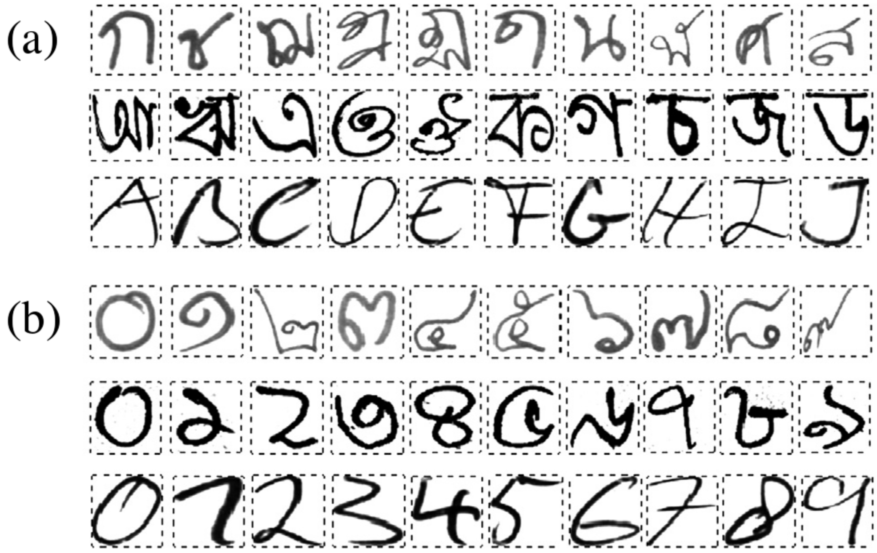

Figure 2.1: Thai, Bangla, and Latin handwriting example shown in the first, second and third row, respectively. Sample (a) shows handwritten characters and (b) shows handwritten digits.

In this thesis, we want to focus on two tasks. The first task is a generic version of the second task.

1. Given the stroke traces of drawn characters, recognize the syntax and semantics of the characters and forward the data to a domain-specific application in an appropriate format.
2. Given the stroke traces of mathematical characters, recognize the syntax and semantics of the characters and return its representation in mathematical notation formats such as LaTeX or MathML [9].

has gained a lot of momentum since the increase of the Internet [4].

ToDo: Add the wonderful pathological examples mentioned in the papers, I read

Current state

ToDo: Sum up the papers, you read

ToDo: Show some existing implementations

Implementation

Our application implements 5 stages. Namely,

segmentation splitting into individual characters,

recognition given a drawn character, return the corresponding Unicode point,

annotation of spatial information spatial relationship between characters is added,

correction using the context information, we apply corrections to the recognized characters,

output format we generate the desired MathML/LaTeX output.

Segmentation

We decided in favor of a very simple implementation for this step. Every stroke is considered as individual character unless it intersects with another stroke. Two or more intersecting strokes are considered as one character. This unit is passed on to the next stage.

Recognition

First, we define the glyphs, we want to recognize. In general, mathematical notation can be intermingled with text, but we restrict our recognition system to the latin script combined with mathematical symbols. This restriction might be insufficient for other recognition applications. But the task to recognize text in languages such as Japanese, Arabic and Hewbrew goes beyond the scope of the problem tackled in this thesis.

We define four properties:

1. We want to recognize mathematical symbols, latin script and numbers. We use Unicode points to reference these characters.

2. A character X is *visually similar* to another character Y, if at least one of the following conditions is met [1]:
   - if X and Y are commonly written in the same stroke layout by handwriters,
   - X is a larger version of Y or vice versa,
   - X has the same layout like Y and only distinguishes itself by the position within the character boundaries.
3. We want to recognize most characters individually. Some characters are visually difficult to distinguish. Two or more visually similar glyphs are represented in a *character group* (a set of Unicode points). We distinguish the characters later context-sensitively.
4. A character group is represented by its *representative* (one Unicode point), i.e. the member with the smallest Unicode point value.

In order to retrieve the list of character groups and representatives, we using the following algorithm:

1. We consider
   - the Latin alphabet of the Basic Latin block (U+0041-U+005A, U+0061-U+007A),
   - the Mathematical Operators block (U+2200-U+22FF),
   - the Supplemental Mathematical Operators block (U+2A00-U+2AFF),
   - extended by Hindu-Arabic digits, and
   - some more characters considered important for mathematical character recognition.

   Our sources specify 706 Unicode points.
2. These characters are taken and Unicode-normalized [5] using the form NFKD (decomposition by compatibility followed by recomposition respecting canonical equivalence). One exception is (FORKING). This character is not normalized (see below). The resulting characters from the normalization constitute our new list of Unicode points. 8 characters get lost.
3. A custom list defines visually similar characters. Thus some characters are merged into the same character group. 648 Unicode point representatives are left.

This process is required, because character properties such as size are relative. Therefore we defined visual similarity, character classes and their representatives above. Consider, for example, a glyph drawn to represent digit 2. We cannot determine whether it corresponds to 2 (DIGIT

[1] Visual similarity as defined above is a reflexive, symmetric and transitive binary relation (hence, an equivalence relation).

TWO) or $_2$ (SUBSCRIPT TWO), because size is only difference between the characters. Size vastly depends on the size of the drawing area. As such we want to recognize both inputs as `2` (DIGIT TWO) and use context-sensitive information later on to potentially replace it with unicode point $_2$ (SUBSCRIPT TWO). Unicode normalization provides us a standardized way to normalize the characters.

We want to illustrate this process with examples.

- The character `A` (LATIN CAPITAL LETTER A) is passed through the whole algorithm and is added in its original form.
- The character (ROMAN NUMERAL TWO) is normalized to `II` (LATIN CAPITAL LETTER I, LATIN CAPITAL LETTER I). Because these characters are equivalent to the existing individual symbol `I` (LATIN CAPITAL LETTER I), nothing is actually changed in the database by adding (ROMAN NUMERAL TWO).
- However, (LARGE TRIPLE VERTICAL BAR OPERATOR) is not decomposed into three individual code points.
- Sometimes the opposite process takes place. Character (FORKING) is identified as one unicode point point. But normalization returns two characters. Namely, (NONFORKING) and / (COMBINING LONG SOLIDUS OVERLAY). This exception has been removed explicitly in step 2 of the algorithm above.
- The character (EQUAL TO BY DEFINITION) is a combination of the character = (EQUALS SIGN) and `def` (LATIN SMALL LETTER D, LATIN SMALL LETTER E, LATIN SMALL LETTER F). Symbol = (EQUALS SIGN) itself is a composition of two characters – (HYPHEN-MINUS). However, Unicode normalization retains the same character.

Because of this process, a hierarchical structure of glyph components is implied. A glyph can be decomposed if two or more non-intersecting strokes are part of a glyph and those strokes occur in at least one other glyph.

ToDo: describe stroke simplification algorithm

ToDo: Describe the database we used to compare characters with

Annotation

ToDo: describe how the spatial information is encoded

Correction

ToDo: describe how correction is done

ToDo: HMM desired, but was too complex for this thesis

Output format

ToDo: Show MathML/LaTeX example

Appendices

Appendix A

Python program illustrating the Law of Large numbers

In applications of probability theory to computer simulations, random variables stand for random numbers. The following python code is a simulation to check that the Law of Large Numbers (see Section 1.1.7) holds.

```python
import random

n = 1000
m = 50

s = 0
for _ in range(n + 1):
    # add a random value from `[0, 2m]` of uniform distribution
    s += random.randint(0, 2*m)

print((s / n) - m)
```

The program is expected to print a value close to 0 as the expected value of n uniformly distributed random values from $[0, 2m]$ is $2m/n$.

python also provides a `statistics` module with a function computing the mean value:

```python
import random
import statistics

n = 1000
m = 50

# compute the mean value of `n` random values from `[0, 2m]`
mean = statistics.mean(random.randint(0, 2*m) for _ in range(n + 1))

print(mean - m)
```

Index

Bibliography

[1] Martín Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).

[2] C. E. G. Bailey. “Introductory lecture on character recognition”. In: *Proceedings of the IEE - Part B: Electronic and Communication Engineering* 106.29 (1959), pp. 444–445. issn: 0369-8890. doi: `10.1049/pi-b-2.1959.0286`.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. isbn: 0387310738.

[4] Kam-Fai Chan and Dit-Yan Yeung. “Mathematical expression recognition: a survey”. In: *International Journal on Document Analysis and Recognition* 3.1 (2000), pp. 3–15.

[5] Unicode Consortium. *UAX #15: Unicode Normalization Forms*. `http://www.unicode.org/reports/tr15/`. [Online; accessed 02-August-2016]. 2017.

[6] Unicode Consortium. *Unicode Consortium*. `http://www.unicode.org/`. [Online; accessed 02-August-2016]. 2017.

[7] A Philip Dawid. “Conditional independence in statistical theory”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1979), pp. 1–31.

[8] Charles Miller Grinstead and Laurie Snell James. *Introduction to probability*. American Mathematical Society, 2012.

[9] The W3C Math Working Group. *W3C Math Home*. since 1996. url: `https://www.w3.org/TR/MathML3/`.

[10] Kiyosi Itô. *Introduction to probability theory*. Cambridge University Press, 1984.

[11] Ashok Kumar and Pradeep Kumar Bhatia. “Offline handwritten character recognition using improved back-propagation algorithm”. In: (2013).

[12] Yann LeCun. “The MNIST database of handwritten digits”. In: *http://yann. lecun. com/exdb/mnist/* (1998).

[13] Olarik Surinta et al. “Recognition of handwritten characters using local gradient feature descriptors”. In: *Engineering Applications of Artificial Intelligence* 45 (2015), pp. 405–414.

[14] Craig A. Tracy. *Mathematics 135A, Fall 2008*. url: https://www.math.ucdavis.edu/~tracy/courses/math135A/UsefullCourseMaterial/lawLargeNo.pdf (visited on 12/13/2016).