Maschinelles Lernen mit Bayes'schen Programmen

Maria Matveev

Geboren am 28. August 1999 in Tscheljabinsk, Russische Föderation

13. August 2020

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Zweitgutachter: Dr. Bastian Bohn

Institut für Numerische Simulation

Betreuer am Fraunhofer SCAI: Dr. Sebastian Mayer

Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Inhaltsverzeichnis

Einleitung **v**

1. Grundlegendes **1**
1.1. Multivariate Modellierung . . . 2
1.2. Bayes'sche Statistik . . . 5
1.2.1. Bayes'scher Ansatz im Begriffslernen . . . 5
1.2.2. Bayes'sche Statistische Modelle . . . 7
1.3. Stichprobenziehen . . . 10
1.4. Begriffslernen . . . 12

2. Maschinelles Lernen mit Bayes'schen Programmen **15**
2.1. Aufgaben auf dem Omniglot-Datensatz . . . 16
2.2. Rahmenwerk . . . 16
2.2.1. Spezifikation des Problems auf drei Ebenen . . . 16
2.2.2. Spezifikation der multivariaten Wahrscheinlichkeitsverteilungen . . . 18
2.2.3. Lernen der Modellparameter . . . 20
2.2.4. Parsing der Daten . . . 21
2.2.5. Bewertung der Parses und Lösen der Aufgaben . . . 21
2.2.6. Zusammenfassung des Rahmenwerks . . . 22
2.2.7. Kernprinzipien . . . 22
2.3. Modellbewertung . . . 23
2.3.1. Diskretisierung der Verteilungen . . . 23
2.3.2. Klassifkation anhand eines Beispiels . . . 25
2.3.3. Generierung neuer Beispiele . . . 28
2.3.4. Erfinden neuer Konzepte . . . 29
2.4. Anwendung . . . 29
2.4.1. Anwendung in der Mensch-Maschine-Interaktion . . . 29
2.4.2. Andere Anwendungen . . . 31

3. Maschinelles Lernen mit Bayes'schen Programmen auf dem Omniglot-Datensatz **33**
3.1. Problemstellung . . . 33
3.1.1. Datensatz . . . 33
3.1.2. Kognitionswissenschaftliche Relevanz . . . 33
3.2. Anwendung des Rahmenwerks . . . 34
3.2.1. Spezifikation der Buchstaben auf drei Ebenen . . . 34
3.2.2. Spezifikation der multivariaten Verteilungen eines Buchstabens . . . 37
3.2.3. Lernen der Modellparameter vom Buchstabenmodell . . . 38

3.2.4. Parsing eines Buchstabens 40
3.2.5. Bewertung der Zeichenanweisungen 44
3.3. Auswertung 45
3.3.1. Modifikationen 45
3.3.2. Alternative Modelle für die Aufgaben 46
3.3.3. Verhaltensstudien und visuelle Turing-Tests 46
3.3.4. Vergleich der Ergebnisse 47

4. Schluss **49**

A. Grundlagen der Wahrscheinlichkeitstheorie **51**

B. Übersetzungen **56**

C. Weitere Algorithmen **57**

D. Weitere Abbildungen und Tabellen **61**

Literatur **65**

Einleitung

Can machines think?, zu deutsch *Können Computer denken?* Mit dieser Frage beginnt Alan Turing 1950 seinen Aufsatz, in dem er unter anderem sein berühmtes *imitation game*, oft auch Turing-Test genannt, veröffentlicht [Tur50]. In diesem Gedankenexperiment wird der Versuchsperson ein digitaler Kommunikationsweg zu zwei Räumen gegeben. Ihr wird erklärt, dass aus einem der Räume ein Mensch antwortet, und aus dem anderen die Antworten von einem Computer berechnet werden. Die Aufgabe der Versuchsperson ist herauszufinden, in welchem der Räume ein Mensch, und in welchem die Künstliche Intelligenz sitzt. Turing argumentiert, dass der Computer, der erfolgreich diesen Test besteht, eine dem Menschen ähnliche Intelligenz besitzt. Es gibt viel Kritik zu diesem Gedankenexperiment. Etwa argumentiert Searle in seinem Gedankenexperiment zum *Chinesischen Zimmer*, dass die Befolgung von Syntax lange nicht das Verstehen von Semantik impliziert [Sea80]. Unabhängig davon ist der Turing-Test und die Ununterscheidbarkeit zwischen den Antworten von Mensch und Computer eine oft verwendete Messlatte, um Verfahren des Maschinellen Lernens zu evaluieren.

In dieser Arbeit beschäftigen wir uns mit dem Rahmenwerk *Maschinelles Lernen mit Bayes'schen Programmen.*
Erstmals umfassend eingeführt wurde das Rahmenwerk im Jahr 2014 in der Dissertation von Brenden Lake 2014 [Lak14], anschließend wurde es es bis zur renommierten Publikation in der Fachzeitschrift Science, *Human-level concept learning through probabilistic program induction* [LST15] im Jahr 2015 überarbeitet. Die Kognitionswissenschaftler modellierten die Art und Weise, wie Menschen Buchstaben verstehen probabilistisch. Das gelernte Modell soll dann verschiedene generative Aufgaben lösen. Unter anderem soll es ein weiteres Beispiel eines unbekannten Buchstabens, der durch ein einziges Bild gegeben war, zeichnen. Diese Aufgabe ist in Abbildung 1 dargestellt.

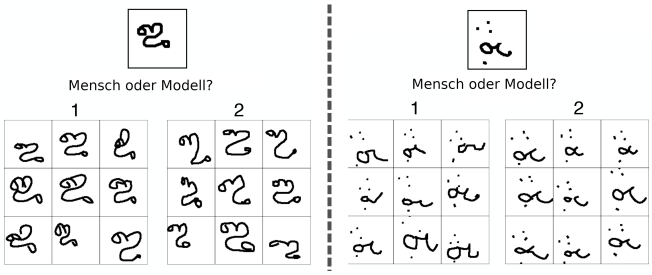

Abbildung 1.: Dargestellt wird in einem der 3×3 Felder die Antwort von Menschen, im anderen vom Modell auf die Aufgabe, ein weiteres Beispiel von einem unbekannten Buchstaben zu zeichnen (links 1, rechts 2 vom Modell generiert). Abbildung modifiziert aus [LST19].

In dieser für Menschen einfachen Aufgabe steckt die Schwierigkeit, zwischen Schlüsselcharakteristika und erlaubter Variabilität in der Zeichnung eines Buchstabens zu unterscheiden. Viele Modelle, die auf Mustererkennung beruhen, scheitern daher an dieser Aufgabe [LST19]. Das vorgestellte Modell sieht Buchstaben als Folge der Zeichnung auf Papier an und zerlegt das Bild durch ein sogenanntes *Parsing* in Striche und ihre Relationen zueinander. Auf diese Weise gelingt es, eine Wahrscheinlichkeitsverteilung auf die Variabilität und entscheidenden Charakteristika eines Buchstabens zu legen und damit Buchstaben untereinander zu vergleichen. Die Parameter der Wahrscheinlichkeitsverteilungen werden auf einigen Alphabeten gelernt. Diese Erfahrung wird dann im gelernten Modell auf unbekannte Buchstaben übertragen. Zu den Kernideen gehört außerdem, Konzepte als probabilistische Programme aufzufassen.
Schließlich wird in visuellen Turing-Tests die Antwort des Modells in verschiedenen Aufgaben mit denen von menschlichen Probanden verglichen. Bemerkenswerterweise wurde der Turing-Test in dem Sinne bestanden, dass die Versuchspersonen nicht zwischen Antwort von Mensch und Modell unterscheiden konnten. Eine naheliegende Folgerung aus diesem Ergebnis ist, dass die Ausgabe des Modells auf dem Level menschlicher Intelligenz ist.

Dies ist insbesondere eindrucksvoll, da einige Wissenschaftler argumentieren, dass die Grundkonzepte, die zum Verstehen eines Buchstabens benötigt werden, eben die sind, in denen der Mensch dem Computer überlegen ist. So behauptete Hofstadter bereits 1985 : „The central problem of Artificial Intelligence is the question: 'what is the letter a'?“, zu deutsch ungefähr „Die zentrale Aufgabe von Künstlicher Intelligenz ist zu erkennen 'was ist ein a?'“[Hof85].

Obwohl sich die Ursprungspublikationen [Lak14] und [LST15] mathematischer Sprache bedienen, wird der Modellierungsprozess einzig am Buchstabendatensatz *Omniglot* [Age20] erklärt. Wir arbeiten die wesentlichen Schritte des Verfahrens mathematisch auf und beweisen die verwendeten Approximationen. Darüber hinaus diskutieren wir die Anwendbarkeit des Rahmenwerks auf andere Konzepte. Obwohl vor viereinhalb Jahren veröffentlicht, gab es bisher nur eine kleine Anzahl von Anwendungen auf andere Daten, deren Ergebnisse suggerieren, dass die Modelle die Konzepte unterhalb des Levels menschlicher Intelligenz lernen.

Aufbau der Arbeit

Die Arbeit ist in vier Kapitel unterteilt. Im Kapitel 1 arbeiten wir die theoretischen Grundlagen des Rahmenwerks auf. Insbesondere gehen wir auf multivariate Modellierung, bayes'sche Statistik, Stichprobenziehen und Begriffslernen ein. Wir beschreiben in Abschnitt 1.2.1 an einem einfachen Beispiel, wie der bayes'sche Gedanke in das Lernen von Konzepten oder Begriffen einspielt. Wir gehen davon aus, dass die Leserin und der Leser mit wahrscheinlichkeitstheoretischen Grundlagen vertraut ist, fassen jedoch im Anhang A die wesentlichen Definitionen und Sätze zur Wahrscheinlichkeitstheorie zusammen.
Im Kapitel 2 stellen wir das Rahmenwerk des *Maschinellen Lernen mit Bayes'schen Programmen* vor und beweisen die verwendeten Approximationen. In Abschnitt 2.4 schlagen wir weitere Anwendungen des Rahmenwerks insbesondere für die Mensch-Maschine-Interaktion vor.

Wir stellen in Kapitel 3 die Anwendung dieses Rahmenwerks auf dem Omniglot-Datensatz vor. Dabei orientieren wir uns zwar an den Publikationen [Lak14] und [LST15], stellen das Modell aber insbesondere anhand unserer wesentlichen Schritte vor, die wir im vorherigen Kapitel ausgearbeitet haben.
Zuletzt geben wir in Kapitel 4 eine kurze Zusammenfassung, eine Diskussion des Modells und einen Ausblick.

Eigener Beitrag

Folgende wesentliche Beiträge finden sich in dieser Arbeit:

- Ausarbeitung der wesentlichen Schritte des Rahmenwerks als allgemeines Algorithmus'schema
- Beweis der verwendeten Approximationen samt Verallgemeinerungen
- Strukturierung der Anwendung des Rahmenwerks auf dem Buchstabendatensatz *Omniglot* anhand der identifizierten wesentlichen Schritte
- Diskussion von Parameterwahlen des Modells aus [LST15] samt
- Durchführung von informellen Experimenten am Programmcode des Modells und Daten in [Lak15] und [Lak19]
- Diskussion der Anwendbarkeit des Rahmenwerks auf andere Datensätze und in der Mensch-Maschine-Interaktion.

Danksagung

Ich möchte mich an dieser Stelle gerne für die Unterstützung bedanken, die ich erfahren durfte. Zunächst bedanke ich mich herzlich bei Professor Jochen Garcke für die umfassende Betreuung dieser Bachelorarbeit. Die Ausarbeitung wurde während zahlreicher Einschränkungen durch die COVID-19-Pandemie verfasst, und ich möchte betonen, wie gut Herr Garcke die Betreuung trotz des ungewöhnlichen ausschließlich digitalen Weges meisterte. Des weiteren möchte ich mich bei Dr. Bastian Bohn für die Übernahme der Zweitkorrektur bedanken. Meinem Betreuer am Fraunhofer SCAI, Dr. Sebastian Mayer danke ich für die Themenanregung und die vielen Diskussionen über Anwendungsmöglichkeiten des in dieser Bachelorarbeit vorgestellten Rahmenwerks. Am Fraunhofer SCAI bedanke ich mich im besonderen auch bei Kathrin Viertel und Moritz Wolter, sowie bei den anderen Kolleginnen und Kollegen für ihre Unterstützung. Professor Ilya Pavlyukevich von der Universität Jena nahm sich mehrere Male Zeit, einige meiner Fragen zur bayes'schen Statistik zu klären, auch ihm danke ich.
Nicht zuletzt gilt mein herzlicher Dank meinen Freunden und meiner Familie.

Kapitel 1.

Grundlegendes

In diesem Kapitel betrachten wir einige theoretische Grundlagen. Dabei gehen wir auf multivariate Modellierung, bayes'sche Statistik, Stichprobenziehen, kognitionswissenschaftliche Konzepte, sowie auf die probabilistische Modellierung dieser beim Maschinellen Lernen ein.
Dabei setzen wir einige wahrscheinlichkeitstheoretische Grundlagen voraus. Die wesentlichen Definitionen und Sätze sind im Anhang in Abschnitt A zusammengefasst. Wir wiederholen daraus den wichtigen Satz von Bayes. Dieser ist nach Thomas Bayes benannt, der auch Namensgeber der bayes'schen Statistik ist. Im Folgenden werden wir einige Variationen des Satz von Bayes kennenlernen.

Satz 1. *Sei* $\{\Omega, \mathscr{A}, \mathbb{P}\}$ *ein Wahrscheinlichkeitsraum und* $\mathcal{D}, h \in \mathscr{A}$ *Ereignisse mit* $\mathbb{P}(\mathcal{D}) > 0, \mathbb{P}(h) > 0$. *Dann gilt*

$$\mathbb{P}(h|\,\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}\,|h)\,\mathbb{P}(h)}{\mathbb{P}(\mathcal{D})}.$$

Bemerkung 2. Wir leiten diese Aussage im Anhang A unter Satz 66 her.

Notationen

Wir bezeichnen wie üblich die reellen Zahlen mit $\mathbb{R}$, die ganzen Zahlen mit $\mathbb{Z}$, den Körper mit zwei Elementen mit $\mathbb{Z}_2$ und die natürlichen Zahlen mit $\mathbb{N}$. Die Null zählen wir zu den natürlichen Zahlen dazu. Schließen wir die Null aus, so schreiben wir $\mathbb{N}^+$. Die positiven reellen Zahlen bezeichnen wir mit $\mathbb{R}^+$, die nichtnegativen mit $\mathbb{R}_0^+$. Für eine endliche Menge M bezeichnen wir mit $|M|$ ihre Kardinalität und mit $\mathcal{P}(M)$ ihre Potenzmenge. Wir schreiben $\mathbb{1}_M$ für die Indikatorfunktion der Menge M.
Wir schreiben für eine Zufallsvariable X für die Wahrscheinlichkeitsfunktion oder Wahrscheinlichkeitsdichte $\rho_X(x)$, wobei wir oft nur $\rho(x)$ oder ρ_X schreiben, wenn die zugehörige Zufallsvariable klar ist. Übliche Verteilungen aus der Literatur stellen wir in Beispiel 75 im Anhang A vor. Dabei schreiben wir δ für das Dirac-Maß, $Ber(p)$ für die Bernoulliverteilung zu Parameter p mit $0 \leq p \leq 1$, $\mathcal{U}(a, b)$ für die Gleichverteilung auf dem Intervall (a, b), $\mathcal{N}(\mu, \sigma^2)$ für die Normalverteilung mit Erwartungswert μ und Varianz σ^2, $Exp(a)$ für die Exponentialverteilung zu Parameter $a > 0$ und $\mathcal{G}(p, b)$ für die Gammaverteilung zu inversem Skalenparameter $b > 0$ und Formparameter $p > 0$. Ist eine Zufallsvariable durch ihre Verteilung gegeben, schreiben wir $\sim$, zum Beispiel $X \sim \mathcal{N}(\mu, \sigma^2)$. Zuletzt bezeichnen wir mit $\propto$ Proportionalität.

1.1. Multivariate Modellierung

Besonders wichtig für unsere Untersuchungen werden mehrdimensionale Zufallsvektoren sein. Um die multivariate Verteilung mehrerer Zufallsvariablen zu definieren, benötigen wird eine σ-Algebra und ein Produktmaß. Im Folgenden, insbesondere bei den bedingten Dichten, orientieren wir uns an [Ebe18, § 2.2].

Definition 3. Seien $(\Omega_i, \mathscr{A}_i, \mu_i)$ Wahrscheinlichkeitsräume. Wir definieren die **Produkt-σ-Algebra** $\bigotimes_{i=1}^n \mathscr{A}_i$ über dem kartesischen Produkt der Mengen $\bigtimes_{i=1}^n \Omega_i$ über den Erzeuger $\{A_1 \times \ldots \times A_n | A_i \in \mathscr{A}\}$. Auf dieser Menge $\bigotimes_{i=1}^n \mathscr{A}_i$ definieren wir das **Produktmaß** μ über ihren Wert auf dem Erzeuger durch

$$\mu(A_1 \times \ldots \times A_n) = \prod_{i=1}^n \mu_i(A_i) \qquad \text{für } A_i \in \mathscr{A}_i\,.$$

Bemerkung 4. Es lässt sich durch Nachrechnen zeigen, dass der definierte Erzeuger durchschnittsstabil ist und daher das definierte Maß eindeutig ist. Ein Beweis dieser Aussage findet sich etwa in [Bov20, Satz 2.15].

Bemerkung 5. Mit dem Satz von Carathéodory können wir allgemein die Existenz und Eindeutigkeit von Produktmaßen zeigen, wie etwa in [Kle13, § 14].

Beispiel 6. Ein Beispiel für eine Produkt-σ-Algebra ist die Borel'sche Algebra $\mathcal{B}(\mathbb{R}^n) = \bigotimes_{i=1}^n \mathcal{B}(\mathbb{R})$. Ein alternativer Erzeuger der Menge ist durch alle Mengen der Form $\{(-\infty, c_1] \times \ldots \times (-\infty, c_n]\}$ für $c_1, \ldots, c_n \in \mathbb{R}$ gegeben.

Auf dem dadurch definierten messbaren Raum können wir nun wie schon vorher von Zufallsvariablen sprechen.

Definition 7. Seien $(\Omega, \mathscr{A}_i, \mu_i)$ Wahrscheinlichkeitsräume und $X_i : \Omega \to \mathbb{R}$ für $i = 1, \ldots n$ messbare Funktionen. Die **multivariate Verteilung** dieser Zufallsvariablen auf der Produkt-σ-Algebra $\mathcal{B}(\mathbb{R}^n)$ ist durch die Funktion

$$\mathbb{P}_X((A_1, \ldots, A_n)) := \mathbb{P}(\{\omega = (\omega_1, \ldots, \omega_n) | X_i(\omega_i) \in \mathscr{A}_i\})$$

gegeben. $X = (X_1, \ldots, X_n)$ nennen wir **multivariate Zufallsvariable**.

Definition 8. Sei $X : \Omega \to \mathbb{R}^n$ multivariate Zufallsvariable. Falls es eine integrierbare Funktion $\rho_X : \mathbb{R}^n \to [0, \infty)$ gibt, so dass für $A \in \mathcal{B}(\mathbb{R}^n)$ gilt, dass

$$\mathbb{P}(X \in A) = \int_A \rho_X(x)\ d^n x,$$

so heißt ρ_X **Wahrscheinlichkeitsdichte** von X. Ist X diskret, so ist die **Wahrscheinlichkeitsfunktion** durch $\rho_X(x_1, \ldots, x_n) = \mathbb{P}(X_1 = x_1, \ldots, X_n = x_n)$ definiert.

Bemerkung 9. Wir lassen den Index oft weg, wenn klar ist, worauf sich die Argumente beziehen. So sprechen wir von $\rho(x, y)$ statt $\rho_{X,Y}(x, y)$.

Betrachten wir nun den Spezialfall, dass die gemeinsame Verteilung der Zufallsvariablen $X : \Omega \to \mathbb{R}^n$ und $Y : \Omega \to \mathbb{R}^m$ und absolut stetig mit Dichte ρ ist. Das heißt

$$\mathbb{P}(X \in A, Y \in B) = \int_A \int_B \rho(x, y)\ dy\ dx \qquad \text{für } A \in \mathcal{B}(\mathbb{R}^n), B \in \mathcal{B}(\mathbb{R}^m).$$

Definition 10. Wir nennen die Verteilung $\rho(x)$ von X **Randverteilung** und meinen damit

$$\mathbb{P}(X \in A) := \mathbb{P}(X \in A, Y \in \mathcal{B}(\mathbb{R}^m)).$$

Lemma 11. *Die Randverteilungen von X und Y sind ebenfalls absolut stetig und haben die Dichten*

$$\rho(x) = \int_{\mathbb{R}^m} \rho(x,y)\ dy \qquad \text{und}$$
$$\rho(y) = \int_{\mathbb{R}^n} \rho(x,y)\ dx.$$

Beweis. Die Aussage folgt aus dem Satz von Fubini. □

In der bayes'schen Statistik bedingen wir oft auf Zufallsvariablen. In diesem Zusammenhang benötigen wir eine bedingte Dichte.

Definition 12. Die Funktion $\rho_{Y|X} : \mathbb{R}^m \times \mathbb{R}^y \to [0,\infty)$ definiert durch

$$\rho_{Y|X}(y|x) = \begin{cases} \frac{\rho_{X,Y}(x,y)}{\rho_X(x)} & \text{für } \rho_X(x) \neq 0 \\ \rho_Y(y) & \text{für } \rho_X(x) = 0 \end{cases}$$

heißt **bedingte Dichte von Y gegeben X**. Wieder schreiben wir nur $\rho(y|x)$ statt $\rho_{Y|X}(y|x)$ wenn die zugehörigen Zufallsvariablen klar sind.

Bemerkung 13. Entsprechend definieren wir die bedingte Dichte von mehreren mehrdimensionalen Zufallsvariablen. Seien $X_i : \Omega \to \mathbb{R}^{n_i}$ für $i = 1,\dots,n$ und $Y_j : \Omega \to \mathbb{R}^{m_j}$ für $j = 1,\dots,m$ gegeben. Wir setzen $N = \sum_{i=1}^n n_i$ und $M = \sum_{i=1}^m m_i$ und definieren die zusammengesetzten Zufallsvariablen $X \to \mathbb{R}^N$ durch $X = (X_1,\dots,X_n)$ und $Y \to \mathbb{R}^M$ durch $Y = (Y_1,\dots,Y_m)$. Entsprechend können wir die Definition 12 auf X und Y anwenden und erhalten

$$\rho(x_1, x_2, \dots x_n | y_1, y_2, \dots y_m) := \rho(x|y).$$

Siehe hierzu auch die Kettenregeln im Satz 18.

Definition 14. Dabei bezeichnet $\rho(x)$ die **A-priori Verteilung**, $\rho(x|y)$ die **A-posteriori Verteilung**, $\rho(y|x)$ die **Plausibilität** oder **Likelihood**.

Korollar 15. *Es gilt*

$$\rho(x,y) = \rho(x|y)\rho(y) = \rho(y|x)\rho(x).$$

Korollar 16. *Insbesondere gilt für X, Y mit absolut stetiger gemeinsamer Verteilung*

$$\rho(x) = \int_{\mathbb{R}^m} \rho(x|y)\rho(y)\ dy.$$

Beweis. Dies folgt direkt aus Lemma 11 und Korollar 15, denn

$$\rho(x) \overset{Lemma\ 11}{=} \int_{\mathbb{R}^m} \rho(x,y)\ dy \overset{Kor.15}{=} \int_{\mathbb{R}^m} \rho(x|y)\rho(y)\ dy.$$

□

Dies kann als kontinuierliches Äquivalent zum Satz der totalen Wahrscheinlichkeit aufgefasst werden kann, siehe Satz 68.
Für Zufallsvariablen gilt dann die folgende Formulierung des Satz von Bayes.

Satz 17. *Seien $X : \Omega \to \mathbb{R}^n$ und $Y : \Omega \to \mathbb{R}^m$ Zufallsvariablen mit absolut stetiger Dichte $\rho(x, y)$ und es gelte $\rho(y) > 0$. Dann gilt*

$$\rho(x|y) = \frac{\rho(x,y)}{\int_{\mathbb{R}^n} \rho(x,y)\ dx} = \frac{\rho(y|x)\rho(x)}{\int_{\mathbb{R}^n} \rho(y|x)\rho(x)\ dx}.$$

Beweis. Es gilt

$$\rho(x|y) \overset{Def.\ 12}{=} \frac{\rho(x,y)}{\rho_Y(y)} \overset{Lemma\ 11}{=} \frac{\rho(x,y)}{\int_{\mathbb{R}^n} \rho(x,y)\ dx} \overset{Kor.15}{=} \frac{\rho(y|x)\rho(x)}{\int_{\mathbb{R}^n} \rho(y|x)\rho(x)\ dx}.$$

□

Arbeiten wir mit mehr als einer Zufallsvariable, so können wir diese gemeinsame Dichte mit der Kettenregel ausmultiplizieren.

Satz 18. *Es gilt für $d \in \mathbb{N}$ und die Zufallsvariablen $X_1, \dots X_d$*

$$\rho(x_1, x_2, \dots, x_d) = \rho(x_1) \cdot \rho(x_2|x_1) \cdot \rho(x_3|x_2, x_1) \dots \rho(x_d|x_{d-1}, \dots, x_2, x_1).$$

Beweis. Dies beweisen wir induktiv.
Induktionsanfang: d=2 Es ist nach Korollar 15

$$\rho(x_1, x_2) = \rho(x_1|x_2)\rho(x_1).$$

Induktionsschritt: $d \curvearrowright d+1$. Die Aussage gelte für d. Seien $X_1, \dots, X_{d+1}$ beliebige Zufallsvariablen $X_i : \Omega \to \mathbb{R}^{n_i}$ mit $i = 1, \dots, d+1$. Wir definieren die Zufallsvariablen $X'_1, \dots, X'_d$ durch

$$X'_i = \begin{cases} X_i : \Omega \to \mathbb{R}^{n_i} & i = 1, \dots, d-1 \\ (X_d, X_{d+1}) : \Omega \to \mathbb{R}^{n_d + n_{d+1}} & i = d. \end{cases}$$

Dann gilt nach Induktionsvoraussetzung (IV):

$$\begin{aligned} & \rho(x_1, x_2, \dots, x_d, x_{d+1}) \\ &= \rho(x'_1, x'_2, \dots, x'_d) \\ &\overset{IV}{=} \rho(x'_1) \cdot \rho(x'_2|x'_1) \cdot \rho(x'_3|x'_2, x'_1) \dots \rho(x'_d|x'_{d-1}, \dots, x'_2, x'_1) \\ &= \rho(x_1) \cdot \rho(x_2|x_1) \cdot \rho(x_3|x_2, x_1) \dots \rho(x_d, x_{d+1}|x_{d-1}, \dots, x_2, x_1) \\ &\overset{Kor.15}{=} \rho(x_1) \cdot \rho(x_2|x_1) \dots \rho(x_d|x_{d-1}, \dots, x_2, x_1) \cdot \rho(x_{d+1}|x_d, \dots, x_2, x_1). \end{aligned}$$

Folglich gilt die Kettenregel tatsächlich. □

1.2. Bayes'sche Statistik

In der Statistik gibt es zwei grundlegende Paradigmen: den frequentistischen und den bayes'schen Ansatz. Beim ersteren wird Wahrscheinlichkeit als Grenzwert der relativen Häufigkeit von Ereignissen in vielen (wiederholbaren) Zufallsexperimenten gesehen. Bayes'sche Statistik dagegen befasst sich mit der Wahrscheinlichkeit als Maß der Glaubwürdigkeit [Jay03].

Gegeben sind diskrete Daten $\mathcal{D}$ und ein Raum möglicher Hypothesen $\mathcal{H}$ mit den möglichen Modellen, die den Daten unterliegen könnten. Die A-priori Verteilung $\rho(h)$ für $h \in \mathcal{H}$ modelliert dabei das Wissen über die Daten.
Das Ziel ist, die passendste Hypothese zu beobachteten Daten zu schätzen. Da $\rho(\mathcal{D})$ als konstant betrachtet wird und damit nach Satz 1

$$\rho(h|\,\mathcal{D}) \propto \rho(\mathcal{D}\,|h)\rho(h)$$

gilt, genügt es, diesen Wert zu maximieren.

Definition 19. Zwei verbreitete Punktschätzer sind der **Maximum Likelihood Estimator MLE**

$$\widehat{h}^{MLE} = \arg\max_h \rho(\mathcal{D}\,|h)$$

und der **Maximum a posteriori**-Schätzer *MAP*

$$\widehat{h}^{MAP} = \arg\max_h \rho(\mathcal{D}|h)\rho(h).$$

Der MAP-Schätzer bezieht das a-priori Wissen über den Prozess in die Schätzung mit ein. MLE ist ein Spezialfall von MAP mit uniformer A-priori Verteilung [Mur12].

1.2.1. Bayes'scher Ansatz im Begriffslernen

An einem Beispiel aus der Dissertation von J. B. Tenenbaum wollen wir zeigen, wie diese Auffassung mit dem Begriffslernen zusammenhängt. Dabei orientieren wir uns an [Mur12] und [Ten99]. Diese beschreiben eine Aufgabe *Number Game* aus dem Bereich des Begriffslernens und modellieren sie probabilistisch.

Beispiel 20. Im folgenden Beispiel *Number Game* zeigen wir anhand einer einfachen Aufgabenstellung aus dem Begriffslernen, wie der bayes'sche Gedanke in die Modellierung einfließt.
Gegeben sei eine Menge $M = \{1, \ldots, 100\}$ und eine Teilmenge dieser Menge $\mathcal{D}$. Die Aufgabe ist, eine Konzept zu finden, zu dem diese Daten gehören, zum Beispiel alle geraden Zahlen oder alle Dreierpotenzen. Dazu betrachten wir zuerst alle mit den Daten konsistenten Hypothesen. Gehört etwa die Zahl 3 zu $\mathcal{D}$, kann die Hypothese nicht mehr *alle Zweierpotenzen* heißen. Aus allen Hypothesen wählen wir nun die aus, die am besten passt.
Der bayes'sche Ansatz ist nun, auf allen Hypothesen eine A-priori Verteilung zu definieren. Die verschiedenen Hypothesen treten nur mit einer gewissen Wahrscheinlichkeit

auf. Insbesondere werden einige Hypothesen als wahrscheinlicher als andere gesehen. Nur wenn die wahrscheinlicheren Hypothesen ausgeschlossen wurden, etwa da durch mehr Datenpunkte sie nicht mehr konsistent mit den Daten sind, kann die unwahrscheinlichere Hypothese gewählt werden. Wie beziehen also unser Wissen über die Welt in das Schätzen der Hypothese mit ein.

Formulieren wir diese Situation formal. Ein arithmetisches Konzept C auf M ist eine Teilmenge $C \subset M$. Gegeben positive Beispiele $\mathcal{D} \subset M$ ist die Aufgabe für $x \in M \setminus \mathcal{D}$ zu schätzen, ob $x \in C$. In der Dissertation stellte Tenenbaum diese Aufgabe verschiedenen Probanden und modellierte sie. Er konnte nachweisen, dass die Modellierung gut die durchgeführte Versuchsstudie beschreibt, was dafür spricht, dass Menschen Konzepte in diesem bayes'schen Rahmenwerk verstehen.
Wir stellen einen Hypothesenraum $\mathcal{H}$ mit allen möglichen arithmetischen Konzepten auf, zum Beispiel $\mathcal{H} = \mathcal{P}(M)$ oder die Intervalle und Potenzen

$$\mathcal{H} = \{\{n, n+1, \ldots, m\} | 1 \leq n \leq m \leq 100\} \cup \{\{n^p | 0 \leq p, n^p \leq 100\} | n \in M\}.$$

Nach Angabe der positiven Beispiele $\mathcal{D}$ werden nur noch alle Hypothesen betrachtet, die mit der gesehen Menge konsistent sind.

Wir treffen die starke Annahme, das der Datensatz $\mathcal{D}$ mit N Elementen uniform zufällig mit Zurücklegen aus dem Konzept C gezogen wurde. Das bedeutet für eine Hypothese h, dass

$$\rho(\mathcal{D}\,|h) = \Big(\frac{1}{|h|}\Big)^N$$

ein plausibler Likelihood ist, da damit die einfachste konsistente Hypothese den höchsten Likelihood hat.

Betrachten wir ein konkretes, stark vereinfachtes Zahlenbeispiel. Seien die folgenden drei Hypothesen gegeben, wobei wir uniforme A-priori Verteilung $\rho(h)$ annehmen, also $\rho(h_2) = \rho(h_{gerade}) = \rho(h_{ungerade}) = \frac{1}{3}$ gilt:

$$\begin{aligned}
h_2 &= \{1, 2, 4, 8, 16, 32, 64\}; & |h_2| &= 7; & \rho(h_2) &= \frac{1}{3} \\
h_{gerade} &= \{2k | k = 1, \ldots, 50\}; & |h_{gerade}| &= 50; & \rho(h_{gerade}) &= \frac{1}{3} \\
h_{ungerade} &= \{2k+1 | k = 0, \ldots, 50\}; & |h_{ungerade}| &= 50; & \rho(h_{ungerade}) &= \frac{1}{3}.
\end{aligned}$$

Nach Betrachten der Menge $\mathcal{D} = \{8, 16, 2, 64\}$, sind nur noch die Hypthesen h_2 und h_{gerade} mit den Daten $\mathcal{D}$ konsistent. Da die gesehenen Zahlen nur Zweierpotenzen sind, wäre es ein *verdächtiger Zufall* (englisch: *suspicious coincidence*), wenn das unterliegende Konzept h_{gerade} wäre. Die Modellierung unterstützt das, da

$$\begin{aligned}
\widehat{h}^{MAP} &= \arg\max_h \rho(\mathcal{D}|h)\rho(h) &&= h_2 \\
\widehat{h}^{MLE} &= \arg\max_h \rho(\mathcal{D}\,|h) &&= h_2.
\end{aligned}$$

Bisher haben wir unser Wissen über die Welt nicht in die Modellierung eingebracht. Sei nun noch zusätzlich eine konsistente, aber *unnatürliche* Hypothese $h_{unnatuerlich}$ gegeben.

Wir gewichten diese Unnatürlichkeit mit einer geringen A-priori Verteilung. Unsere Hypothesenraum enthalte im Folgenden diese drei konsistenten Hypothesen zur Menge $\mathcal{D} = \{8, 16, 2, 64\}$:

$$\begin{aligned} h_2 &= \{1, 2, 4, 8, 16, 32, 64\}; & |h_2| &= 7; & \rho(h_2) &= \frac{4}{9} \\ h_{gerade} &= \{2k | k = 1, \ldots, 50\}; & |h_{gerade}| &= 50; & \rho(h_{gerade}) &= \frac{4}{9} \\ h_{unnatuerlich} &= \{2, 8, 16, 64, 19\}; & |h_{unnatuerlich}| &= 5; & \rho(h_{unnatuerlich}) &= \frac{1}{9}. \end{aligned}$$

Entsprechend geben die Schätzer

$$\begin{aligned} \widehat{h}^{MAP} &= \arg\max_h \rho(\mathcal{D}|h)\rho(h) &&= h_2 \\ \widehat{h}^{MLE} &= \arg\max_h \rho(\mathcal{D}\,|h) &&= h_{unnatuerlich}. \end{aligned}$$

Wählen wir die kleinste mit den Daten konsistente Hypothese aus, so maximieren wir $\widehat{h}^{MLE} = \arg\max_h \rho(\mathcal{D}\,|h) = h_{unnatuerlich}$. In diesen Schätzer geht das Wissen über die Welt $\rho(h)$ nicht mit ein. Doch die gefundene Hypothese $h_{unnatuerlich}$ ist unnatürlich, was wir durch eine kleine A-priori Verteilung $\rho(h_{unnatuerlich}) = \frac{1}{9}$ modelliert haben. Entsprechend ist die A-posteriori Verteilung nach Betrachtung des Datensatzes $\mathcal{D}$ dennoch klein, auch wenn der Likelihood $\rho(\mathcal{D}\,|h_{unnatuerlich}) = \frac{1}{5}$ der höchste unter allen möglichen Hypothesen ist. Insgesamt beziehen wir in dem bayes'schen Ansatz unser Wissen über den Hypothesenraum zusätzlich in die Schätzung ein.

Bemerkung 21. Im Maschinellen Lernen mit Bayes'schen Programmen werden wir die A-priori Wahrscheinlichkeit anhand eines Datensatzes trainieren. Dadurch lernt der Algorithmus automatisch, welche Hypothesen unnatürlich sind, und welche natürlich.

Bemerkung 22. Da im Beispiel 20 mit einem diskreten Wahrscheinlichkeitsraum gearbeitet wird, hätten wir statt von der Wahrscheinlichkeitsfunktion ρ auch von dem Wahrscheinlichkeitsmaß $\mathbb{P}$ sprechen können, und statt der A-priori Verteilung von der sogenannten A-priori Wahrscheinlichkeit.

1.2.2. Bayes'sche Statistische Modelle

Wir betrachten im Folgenden nicht mehr einen reinen Hypothesenraum und $\mathscr{A}$ als Menge der möglichen Konzepte, sondern ein statistisches Modell als Familie von Wahrscheinlichkeitsverteilungen. Dabei interessieren uns vor allem identifizierbare statistische Modelle.

Definition 23. Ein **statistisches Modell** ist ein Tripel $\mathcal{M} = (\Omega, \mathscr{A}, (P_\theta)_{\theta\in\Theta})$ mit Parameterraum Θ, Messraum $(\Omega, \mathscr{A})$ und Familie von Wahrscheinlichkeitsverteilungen auf dem Stichprobenraum $(\Omega, \mathscr{A})$. Falls $\Theta \subset \mathbb{R}^m$ sprechen wir von einem **parametrischen Modell**, sonst von einem **nichtparametrischem Modell**.

Definition 24. Ein statistisches Modell heißt **identifizierbar**, falls

$$\theta_1 \neq \theta_2 \Rightarrow \mathbb{P}_{\theta_1} \neq \mathbb{P}_{\theta_2}$$

für alle $\theta_1, \theta_2 \in \Theta$ gilt.

Wir haben Beobachtungen/Messungen gegeben, die wir stochastisch durch unabhängig identisch verteilte Zufallsvariablen $X_1, \dots, X_n : \Omega \to \mathbb{R}^k$ mit unbekannter Wahrscheinlichkeitsverteilung $\mathbb{P}_\theta$ modellieren. Gegeben eine Realisierung der Zufallsvariablen $I_1, \dots, I_n$ wollen wir den Parametervektor θ schätzen. Dabei wird θ als unbekannt, aber deterministisch angenommen. Falls Θ diskret ist, können wir wieder maximieren

$$\arg\max_\theta \mathbb{P}_\theta(I_1, \dots, I_n) \overset{I_i \text{ unabhängig}}{=} \arg\max_\theta \prod_{i=1}^n \mathbb{P}_\theta(I_i).$$

Dies entspricht dem Maximum Likelihood Estimator [HTF09].
Wir schränken uns dabei oft auf den Fall ein, dass es eine Dichte ρ gibt, so dass die Beobachtungen gemäß der Dichte $\rho(I|\theta)$ verteilt sind. Daher können wir statt von einer Familie von Wahrscheinlichkeitsmaßen $(P_\theta)_{\theta\in\Theta}$ auch von dem Likelihood $\rho(I|\theta)$ sprechen.

Betrachten wir im Folgenden die bayes'sche Situation, wie sie etwa in den Büchern [Gel+13] und [Rob07] dargestellt wird.
Wir nehmen an, dass zu dem Parameter θ eine A-priori Verteilung gegeben ist. Eine alternative Sichtweise ist, θ als Realisierung eines Zufallsvektors $\hat{\theta} : \Omega \to \Theta$ zu sehen. Wir setzen eine A-priori Verteilung des Parameters voraus, eine angenommene Verteilung vor Messung der Realisierungen von $X_1, \dots, X_n$. Dies führt zum Begriff des bayes'schen statistischen Modells.

Definition 25. Sei $\Theta \subset \mathbb{R}^m$. Ein **bayes'sches statistisches Modell** ist ein statistisches Modell mit einem Likelihood $\rho(x|\theta)$ und einer A-priori Verteilung auf dem Parameterraum $\rho(\theta)$.

Mit Satz 17 und Korollar 16 gilt dann als A-posteriori Verteilung unseres Parameters nach Messung der Daten I:

$$\rho(\theta|I) = \frac{\rho(I|\theta)\rho(\theta)}{\int_\Theta \rho(I|\theta)\rho(\theta)\ d\theta}.$$

Uns interessiert nun, wie ein weiterer Datenpunkt $I^{(new)}$ aussieht, nach dem wir I gesehen haben.

Satz 26. *Es gilt*

$$\rho(I^{(new)}|I) = \int_\Theta \rho(I^{(new)}|\theta)\rho(\theta|I)\ d\theta.$$

Beweis. Es gilt

$$\begin{aligned}\rho(I^{(new)}|I) &\overset{Lemma\ 11}{=} \int_\Theta \rho(I^{(new)}, \theta|I)\ d\theta \\ &\overset{Kor.\ 15}{=} \int_\Theta \rho(I^{(new)}|\theta, I)\rho(\theta|I)\ d\theta \\ &= \int_\Theta \rho(I^{(new)}|\theta)\rho(\theta|I)\ d\theta.\end{aligned}$$

Dabei folgt die letzte Gleichheit aus der Unabhängigkeit der Beobachtungen. □

Eine mögliche Darstellung von bayes'schen statistischen Modellen ist, eine sogenannte bedingte Hierarchie in der A-priori Verteilung $\rho(\theta)$ einzuführen. Dies führt zu der Definition eines *hierarchischen bayes'schen Modells* [Vid04] [Gel+13].

Definition 27. Gegeben sei ein bayes'sches statistisches Modell über den Likelihood $\rho(I|\theta)$ und die A-priori Verteilung $\rho(\theta)$. Seien außerdem Parameterräume $\Theta_1, \ldots, \Theta_n \subset \mathbb{R}^{n_i}$ gegeben. Kann die A-priori Verteilung $\rho(\theta)$ in bedingte Verteilungen $\rho(\theta|\theta_1), \rho(\theta_1|\theta_2), \ldots,$ $\rho(\theta_{n-1}|\theta_n)$ und die Randverteilung $\rho(\theta_n)$ zerlegt werden, so dass gilt

$$\rho(\theta) = \int_{\Theta_1 \times \ldots \times \Theta_n} \rho(\theta|\theta_1)\rho(\theta_1|\theta_2)\ldots\rho(\theta_{n-1}|\theta_n)\rho(\theta_n)\ d\theta_1 \ldots\ d\theta_n,$$

so nennen wir das Modell **Hierarchisches Bayes'sches Modell**. Die Parameter θ_i bezeichnen wir als **Hyperparameter** des Modells. Wir sagen, dass das Modell $n+1$ **Stufen** beziehungsweise $n+2$ **Ebenen** hat und visualisieren das Modell durch

$$\theta_n \to \ldots \to \theta_1 \to \theta \to I.$$

Wir nehmen also an, dass stets

$$\rho(I|\theta, \theta_1, \ldots) = \rho(I|\theta)$$

gilt und gleichermaßen

$$\begin{aligned} \rho(\theta_i|\theta, I) &= \rho(\theta_i|\theta) \quad \text{und} \\ \rho(\theta_i|\theta_{i+1}, \ldots \theta_n) &= \rho(\theta_i|\theta_{i+1}). \end{aligned}$$

Nach der Kettenregel aus Satz 18 wissen wir

$$\rho(I, \theta, \theta_1, \ldots, \theta_n) = \rho(I|\theta, \theta_1, \ldots, \theta_n)\rho(\theta|\theta_1, \ldots, \theta_n)\ldots\rho(\theta_{n-1}|\theta_n)\rho(\theta_n).$$

Mit der in der Definition 27 eingeführten sogenannten bedingten Hierarchie in der A-priori Verteilung folgt

$$\rho(I, \theta, \theta_1, \ldots, \theta_n) = \rho(I|\theta)\rho(\theta|\theta_1)\rho(\theta_1|\theta_2)\ldots\rho(\theta_{n-1}|\theta_n)\rho(\theta_n).$$

Vorteile der Betrachtung ist, dass nur benachbarte bedingte Verteilungen definiert werden müssen. Dies erleichtert etwa das bedingte Stichprobenziehen. Wir können durch die Ebenen eine Hierarchie in den Daten repräsentieren. Dies erlaubt das Vergleichen von gruppierten Daten. Ein Beispiel ist etwa die Analyse der Ergebnisse eines Mathematiktests von Studierenden verschiedener Universitäten in verschiedenen Studienfächern. Wir können hier die Studienfächer und Universitäten auf verschiedenen Ebenen modellieren [Gel+13].

Beim *Maschinellen Lernen mit bayes'schen Programmen* werden wir insbesondere hierarchische bayes'sche Modelle mit zwei Stufen betrachten. Grafisch haben wir

$$\psi \to \tilde{\theta} \to I.$$

Wir benennen die Ebenen mit $I, \tilde{\theta}, \psi$. Nach Modellierung gilt dann

$$\rho(I|\,\tilde{\theta}) = \rho(I|\,\tilde{\theta}, \psi).$$

Damit können wir die folgende Proportionalität beweisen.

Satz 28. *Es gilt:*

$$\rho(\psi, \tilde{\theta}\,|I) \propto \rho(I|\,\tilde{\theta})\rho(\tilde{\theta}\,|\psi)\rho(\psi).$$

Beweis. Es ist nach Satz von Bayes 1

$$\rho(\tilde{\theta}, \psi) = \rho(\tilde{\theta}\,|\psi)\rho(\psi)$$

und

$$\begin{aligned}\rho(\psi, \tilde{\theta}, I) =& \rho(I|\,\tilde{\theta}, \psi)\rho(\tilde{\theta}, \psi)\\ =& \rho(I|\,\tilde{\theta})\rho(\tilde{\theta}, \psi).\end{aligned}$$

Die Aussage folgt direkt, da

$$\rho(\psi, \tilde{\theta}\,|I)\rho(I) = \rho(\psi, \tilde{\theta}, I).$$

□

1.3. Stichprobenziehen

Stichprobenziehen ist eine wichtige Methode, um hochdimensionale Räume zu approximieren. Wir stellen daher kurz den Metropolis-Hastings-Algorithmus als Spezialfall sogenannter *Markov Chain Monte Carlo*-Methoden vor, kurz MCMC.
Wir orientieren uns im Folgenden an [Gel+13] sowie [Cyn12]. Insgesamt möchten wir eine A-posteriori Wahrscheinlichkeit $\rho(\theta|x)$ approximieren. Im Hinblick auf die vorgestellten hierarchischen bayes'schen Modelle, kann man dann stufenweise die Parameter approximieren.

Beispiel 29. Wir möchten anhand der *Monte-Carlo-Integration* den ursprünglichen Grund der Monte-Carlo Methoden verdeutlichen.
Angenommen, wir haben ein Integral der Form $\int_a^b g(x)\ dx$. Zerlegen wir g in eine Funktion f und eine Dichte ρ mit $\int_a^b \rho(x)\ dx = 1$, so steckt in der Integralberechnung der Erwartungswert von f über ρ

$$\int_a^b g(x)\ dx = \int_a^b f(x) \cdot \rho(x)\ dx = \mathbb{E}_{\rho(x)}(f(x)).$$

Nach dem Gesetz der großen Zahlen [Bov20, § 6] gilt dann für unabhängig identisch verteilte Realisierungen $x_1, \ldots, x_n$ der Zufallsvariable, die durch ρ definiert wird eine fast sichere Konvergenz und damit die Approximation

$$\int_a^b h(x)\ dx = \mathbb{E}_{\rho(x)}(f(x)) \overset{n\to\infty}{\approx} \frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

In der bayes'schen Sichtart betrachten wir eine Zerlegung von $\int_\Theta g(\theta)\ d\theta$ in Likelihood $\rho(x|\theta)$ und A-priori Verteilung $\rho(\theta)$. Dann haben wir für θ_i unabhängig identisch verteilte Stichproben aus $\rho(\theta_i|x)$ für $n \to \infty$ fast sichere Konvergenz

$$\int_\Theta f(\theta)\rho(x|\theta)\rho(\theta)\ d\theta \overset{n\to\infty}{\approx} \frac{1}{n}\sum_{i=1}^{n} f(\theta_i)\rho(x|\theta_i).$$

Insbesondere möchten wir aus der A-posteriori Verteilung $\rho(\theta|x)$ Stichproben ziehen. Diese approximieren wir, in dem wir die Stichproben iterativ bedingt auf den letzten Wert ziehen. Unter gewissen Voraussetzungen wird dadurch die richtige Verteilung approximiert.
Um dies formal einzuführen, betrachten wir einige Definitionen, beginnend mit der Definition des stochastischen Kerns.

Definition 30. Ein **stochastischer Kern** K über einem messbaren Raum $\{\Omega, \mathscr{A}\}$ ist eine Funktion $K : \Omega \times \mathscr{A} \to [0, 1]$ so dass $K(\theta|\cdot)$ für alle $\theta \in \Theta$ ein Wahrscheinlichkeitsmaß ist und $K(\cdot|A)$ für alle $A \in \mathscr{A}$ eine messbare Funktion ist. Ein stochastischer Kern ist **absolut stetig**, falls es eine Dichte k gibt, so dass für alle $\theta \in \Theta$ gilt

$$K(\theta|A) = \int_A k(\theta, x)\ dx.$$

Nun können wir eine kontinuierliche absolut stetige Markovkette definieren.

Definition 31. Eine **kontinuierliche Markovkette** ist eine Folge $\theta_0, \theta_1, \ldots$ von Zufallsvariablen, so dass gilt

$$\mathbb{P}(\theta_t \in A|\theta_{t-1}, \ldots, \theta_1) = \mathbb{P}(\theta_t \in A|\theta_{t-1}) := K(\theta_t|A).$$

Das Ziel wird sein, eine Markovkette zu konstruieren, deren *stationäre Verteilung* eben die A-posteriori Verteilung $\rho(\theta|x)$ ist.

Definition 32. Die **stationäre Verteilung** einer Markovkette ist Π, falls gilt

$$\Pi(A) = \int_\Theta K(\theta|A)\ \Pi(d\theta).$$

Man kann zeigen, dass eine Verteilung π, für die Gleichung

$$K(\theta|\theta')\pi(\theta, x) = K(\theta'|\theta)\pi(\theta')$$

gilt, stationär ist.

In Algorithmus 1 ist der bekannte Metropolis-Hastings-Algorithmus skizziert. J ist dabei die sogenannte *Vorschlagsverteilung.*

Algorithmus 1 Metropolis-Hasting-Algorithmus, um die A-posteriori Verteilung $\rho(\theta|x)$ zu approximieren. Algorithmus aus [Cyn12].

procedure MetropolisHastings(x)
  Wähle Startpunkt θ_0 mit $\rho(\theta_0|x) > 0$.
  for $t = 1, \ldots, N$ **do**
    Ziehe θ^* aus der Vorschlagsverteilung $\theta^* \sim J(\theta^*|\theta_{t-1})$
    $\alpha(\theta_{t-1}, \theta^*) \leftarrow \min\left(\frac{\rho(\theta^*|x)J(\theta^*|\theta_{t-1})}{\rho(\theta_{t-1}|x)J(\theta_{t-1}|\theta^*)}, 1\right)$
    $\theta_t \leftarrow \begin{cases} \theta^* & \text{mit Wahrscheinlichkeit } \alpha(\theta_{t-1}, \theta^*) \\ \theta_{t-1} & \text{sonst} \end{cases}$
  end for
end procedure

Der Algorithmus erzeugt eine Markovkette, da die Vorschlagsverteilung J stets nur vom vorhergehenden Wert θ_{t-1} abhängt. Dabei wird ein Vorschlag θ^* nur mit Wahrscheinlichkeit $\frac{\rho(\theta^*|x)}{\rho(\theta_{t-1}|x)}$ angenommen, wenn die Wahrscheinlichkeit kleiner ist, also $\rho(\theta^*|x) < \rho(\theta_{t-1}|x)$ gilt. Hat der Vorschlag θ^* höhere A-priori Verteilung $\rho(\theta^*|x) \geq \rho(\theta_{t-1}|x)$, so wird er stets angenommen. Der Vorteil dieser Sichtart ist, dass wir die Verteilungen nur bis auf eine multiplikative Konstante kennen müssen. Es gilt dann der folgende Satz.

Satz 33. *Angenommen J ist so gewählt, dass die erzeugte Markovkette $\theta_0, \theta_1, \ldots$ eine eindeutige stationäre Verteilung hat, dann ist diese eben $\rho(\theta|x)$.*

Beweis. Der stochastische Kern der erzeugten Markovkette ist nach Definition

$$K(\theta|\theta') = J(\theta, \theta')\alpha(\theta, \theta').$$

Wir möchten, dass $\rho(\theta|x)$ die stationäre Verteilung ist und damit für alle $\theta, \theta' \in \Theta$ gilt:

$$K(\theta|\theta')\rho(\theta, x) = K(\theta'|\theta)\rho(\theta'|x).$$

Dies können wir durch Umformung zeigen, siehe etwa [Cyn12, § 6.2.3]. □

Für weitergehende Betrachtungen verweisen wir auf [Gel+13, § 11]. Dort wird erörtert, wie die Vorschlagsverteilung gewählt werden kann sowie weitere Algorithmen vorgestellt. Insbesondere gibt es eine ganze Klasse an sogenannten MCMC-Algorithmen.

1.4. Begriffslernen

Ein Ziel von maschinellem Lernen ist es, die Funktionsweise menschlicher Intelligenz zu verstehen. Der Ansatz, den wir beim Maschinellen Lernen mit Bayes'schen Programmen anwenden, versucht die kognitionswissenschaftlichen Konzepte probabilistisch zu modellieren.

Ein Teil menschlicher Kognition ist das *Begriffslernen* (englisch: *concept learning*). Dies bezieht sich auf die Art und Weise, wie Menschen die Welt in Kategorien unterteilen. In der Begriffsbildung wird zwischen Eigenschaftsbegriffen und Erklärungsbegriffen unterschieden [Eck91]. Aufgrund von Erfahrungen können Menschen positive Beispiele einer Kategorie von negativen unterscheiden. Empirische Untersuchungen haben gezeigt, dass positive Beispiele für das Lernen eines Begriffs ausreichen [TX07].
Mathematisch wollen wir also eine zweiwertige Funktion lernen, die eins ausgibt, falls das Objekt zu der Kategorie gehört, und null falls nicht. Es gibt auch Abstufungen mit unscharfer Logik, doch meist wird vom diskreten Fall ausgegangen. Begriffe werden außerdem durch Beschreibungen/Attribute getrennt, und umgekehrt kann von Beschreibungen auf das Konzept geschlossen werden.

Ein Beispiel aus [TX07] bezieht sich auf das Lernen des unbekannten Wortes *Hund*. Angenommen, ein Kleinkind hat das Wort *Hund* noch nie gehört und geht mit seinen Eltern spazieren und ein Dalmatiner „Max“ rennt vorbei, und die Mutter sagt: „Schau, da ist ein Hund.“ Der Hypothesenraum ist groß: so könnte sie sich auf allgemein Tiere, rennende Tiere, Hunde, Dalmatiner, alle Hunde außer Schäferhunde, Hunde namens

Max, die vordere Hälfte eines rennenden Hundes und mehr beziehen. Dennoch schaffen selbst Kleinkinder beeindruckende Verallgemeinerungen und können den Hypothesenraum an nur wenigen positiven Beispielen in Konzepte zu unterteilen. Dabei eliminieren sie *unnatürliche* Hypothesen, und haben eine natürliches Gespür für Assoziationen. Von klein auf sind Kinder in der Lage, Erfahrungen zu übertragen. Insbesondere kann die innere Logik und Kausalität eines Begriffes auf unbekannte Begriffe übertragen werden. Ist das Wort *Katze* gelernt, so wird der Begriff *Hund* deutlich schneller gelernt und abgegrenzt. Insbesondere werden abstrakte Konzepte wiederverwendet.

Zu den Schlüsselideen der kindlichen Grundeinstellung gehören *intuitive Physik* und *intuitive Psychologie.* Kleinkinder wissen intuitiv, dass Objekte über die Zeit bestehen, sich nur in einem gewissen Maß verändern und das Objekt von verschiedenen Blickwinkeln beobachtet dasselbe bleibt und einer gewissen Kausalität unterliegt. Diese Intuition erlaubt, viele physikalisch unmögliche Theorien auszuschließen und den Hypothesenraum deutlich zu verkleinern. Intuitive Psychologie postuliert, dass Kinder bereits von einem mentalen Zustand bei anderen Personen ausgehen. So haben Menschen Ziele und Vorstellungen von der Welt. Dieser Sachverhalt ermöglicht Kindern eine Unterteilung der Welt in gut und böse [LUS16]. In Kapitel 1.2.2 haben wir bereits eine bayes'sche Modellierung des Hypothesenraums kennengelernt, und wie durch diese probabilistische Auffassung natürliche von unnatürliche Konzepte unterschieden werden können.

Ein weiterer Aspekt ist die Verwendung von *Modellen* von Begriffen. Modelle sind eine Repräsentation der Wirklichkeit und unterliegen einer Kausalität. Als besonders wichtig zählen dabei generative Modelle [LUS16]. Im Gegensatz dazu gibt es beim maschinellen Lernen einen Trend zu modellfreien Ansätzen. Konzepte werden als Muster gesehen, und das Lernen dient der Mustererkennung. Insbesondere werden Eigenschaften (englisch: *features*) aus den Daten extrahiert und diese zur Klassifikation gegeneinander abgeglichen. Mit diesem Ansatz können bemerkenswerte Resultate erzielt werden, wie die zahlreichen Fortschritte im *Deep Learning* zeigen. Bereits 2015 konnte der Ansatz, sehr tiefe Netzwerke zur Bilderkennung zu lernen, das durchschnittliche Level menschlicher Fähigkeiten überschreiten [He+15]. Diese Herangehensweise benötigt sehr viele Daten und sehr viel Rechenleistung, doch zeigt sich in der Mustererkennung sehr erfolgreich. Im Gegensatz zu diesen Modellen können Menschen komplexe Konzepte an einer relativ kleinen Anzahl an Beispielen lernen sowie die gelernten Konzepte gut auf unbekannte Situationen übertragen. Dieser Transfer und das Trainieren mit einer kleinen Datenmenge funktioniert im Deep Learning in der Regel nicht. Durch die probabilistische Modellierung versuchen wir in Kapitel 2 und 3 durch eine kleine Datenmenge mithilfe von Erfahrungsübertrag unbekannte Buchstaben zu lernen und auch generative Aufgaben auf dem Datensatz zu lösen.

Neben dem Vergleich und der Unterscheidung von positiven und negativen Beispielen einer Kategorie sind auch generative Aufgaben wichtige Probleme im Bereich des Begriffslernens. Ein Konzept ermöglicht Handlung, Vorstellung, Kommunikation und Erklärung von Sachverhalten [Lak14, § 1.4]. Sehen Probanden etwa zum ersten Mal einen *Segway*, so können sie in der Regel problemlos verschiedene Aufgaben zu diesem Konzept lösen. Sie können das Fahrzeug in Teile zerlegen, das Konzept auf andere Situationen, wie das Fahren unter Wasser oder in den Bergen anpassen und neue Segways skizzieren. Darin stecken zwei wichtige Kernkonzepte: Zum einen spielt die *Kompositionalität* eine

wichtige Rolle. Das Segway wird in Teile zerlegt und das vorhandene Wissen über diese Teile, zum Beispiel, dass Räder rollen, fließt in das Lernen des neuen Konzeptes Segway mit ein. Das zweite Konzept ist das Verallgemeinern von *Skizzen*. Insbesondere wird der Zusammenhang zwischen verschiedenen Darstellungen, wie etwa einem Foto, Video, physikalischen Objekt oder einer skizzenhaften Zeichnung etwa des Segways erkannt.

In unserer Herangehensweise, dem *Maschinellem Lernen mit Bayes'schen Programmen*, das wir in Kapitel 2 vorstellen und in Kapitel 3 auf dem Buchstabendatensatz anwenden, lernt das Programm ein Modell des Konzeptes. Dieses Modell wird durch eine Verteilung im Gesamtraum repräsentiert. Die A-priori Verteilung wird in der ersten Phase aus dem Datensatz gelernt. Schließlich wird diese abstrakte Erfahrung auf unbekannte Konzepte übertragen. Ein weiterer wichtiger Aspekt ist die Kompositionalität. Das Konzept wird als Menge von Teilen und ihren Relationen untereinander gesehen.
In Abschnitt 3.1.2 stellen wir die kognitionswissenschaftliche Relevanz konkret des Begriffs *Buchstabe* heraus.

Kapitel 2.

Maschinelles Lernen mit Bayes'schen Programmen

In diesem Kapitel stellen wir das Rahmenwerk von *Maschinellem Lernen mit Bayes'schen Programmen* vor. Dieses wurde von Brenden M. Lake in seiner Dissertation 2014 auf dem Buchstabendatensatz *Omniglot* eingeführt [Lak14] und erlangte durch die renommierte Publikation *Human-level concept learning through probabilistic program induction* [LST15] in der Fachzeitschrift Science im Jahr 2015 Bekanntheit.
Konzepte werden auf drei Ebenen beschrieben, um die Schlüsselcharakteristika, erlaubte Variabilität und schließlich die Realisierung im Bildraum zu modellieren. Jede der Ebenen wird als probabilistisches Programm dargestellt, in dem die einzelnen Modellparameter gemäß vorher gelernten Wahrscheinlichkeitsverteilungen gezogen werden. Dabei gehen wir von einer bedingten Hierarchie in der A-priori Verteilung wie in Definition 27 aus. Um die Parameter zu schätzen wird ein sogenannter Parser im Bildraum definiert, der wahrscheinliche Vorschläge für die beiden oberen Ebenen liefert.
Das Modell wurde auf einem Buchstabendatensatz, Omniglot entwickelt und schließlich in visuellen Turing-Tests verifiziert. Bemerkenswerterweise konnten fast alle Probanden nicht zwischen der Antwort von Mensch und Modell unterscheiden. Dies suggeriert, dass das Begriffslernen auf einem Level menschlicher Intelligenz geschieht. Gleichzeitig beruht diese Leistung sehr auf der kognitionswissenschaftlichen Rekonstruktion der Art und Weise, wie Menschen Buchstaben verstehen. Daher lässt sich das Verfahren nicht pauschal auf andere Konzepte übertragen, da sie auf der sehr präzisen Modellierung des Konzeptes beruht.
Dennoch ist die Herangehensweise auch bei anderen Problemen interessant. Die innere Logik des Schemas erlaubt eine skizzenhafte Beschreibung von Konzepten, wie Menschen sie ständig benutzen. Wir werden in Abschnitt 2.4 Anwendungen insbesondere für die Mensch-Maschine-Interaktion darstellen.
In diesem Kapitel arbeiten wir aus der Publikation [LST15] die wesentlichen Schritte des Rahmenwerks aus. Wir formulieren die Kernaussagen und beweisen sie. Später in Kapitel 3 werden wir die identifizierten wesentlichen Schritte auf den Omniglot-Datensatz anwenden. Wir beginnen damit, die Aufgaben, wie sie von dem Modell zu lösen sind, im ersten Abschnitt 2.1 vorzustellen.

2.1. Aufgaben auf dem Omniglot-Datensatz

In Kapitel 3 werden wir das Rahmenwerk des Maschinellen Lernen mit Bayes'schen Programmen auf einen Datensatz von Buchstaben, genannt *Omniglot* anwenden. Das gelernte Modell eines Buchstabens soll verschiedene Aufgaben auf dem Level menschlicher Intelligenz lösen, die in Abbildung 2.1 dargestellt sind. Die Aufgaben sind die Klassifikation unbekannter Buchstaben anhand eines Beispiels (*One-shot-classification*), die Zerlegung von Buchstaben in Teile (*Parsing*), die Generierung weiterer Beispiele eines unbekannten Buchstabens und das Erfinden neuer Buchstaben.

2.2. Rahmenwerk

In diesem Abschnitt stellen wir das Rahmenwerk des *Maschinellen Lernen mit Bayes'schen Programmen* vor. Dazu sind die folgenden fünf Schritte wesentlich:

i) Spezifikation des Problems auf drei Ebenen.

ii) Spezifikation der multivariaten Wahrscheinlichkeitsverteilungen.

iii) Lernen der Modellparameter.

iv) Parsing der Daten.

v) Bewertung der Parses und Lösen der Aufgaben.

Diese fünf Schritte gehen wir nun durch und erstellen so sukzessive das gesamte Rahmenwerk. In Abschnitt 2.2.6 fassen wir diese Schritte des Rahmenwerks zusammen. Konkret setzen wir die Schritte im Kapitel 3 auf dem Buchstabendatensatz um.

Bemerkung 34. An dieser Stelle möchten wir kurz den Namen *Maschinelles Lernen mit Bayes'schen Programmen* kommentieren. In der Veröffentlichung sprechen Lake et al von *Hierarchial Bayesian Program Learning* [Lak14] beziehungsweise *Bayesian Program Learning* [LST15]. Die gewählte freie Übersetzung liegt daher nahe.
2014 wurde von Bessiere et al ein Buch mit dem Titel *Bayesian Programming* veröffentlicht [Bes+13]. Dort zeigen sie die von Jaynes in [Jay03] skizzierte Idee auf, Computerprogramme nicht mit Boolescher Logik sondern über Wahrscheinlichkeitsverteilungen zu konstruieren. In diesem Kontext wird auch ein sogenannter *Bayes'scher Computer* diskutiert. Die im Buch vorgestellte probabilistische Modellierung hat Überschneidungen zu unserem Modell und stellt einen algebraischen Formalismus für etwa sogenannte bayes'sche Netze oder auch hierarchische bayes'sche Modelle dar. In dieser Ausarbeitung beschränken wir uns jedoch auf die Bedeutung von *Bayes'schen Programmen*, wie sie von Lake et al. in [Lak14] und [LST15] bezeichnet werden: Programme, die multivariaten Verteilungen entsprechen.

2.2.1. Spezifikation des Problems auf drei Ebenen

Das Konzept, das modelliert werden soll, wird auf drei Ebenen hierarchisch beschrieben. Dies führt zu einem hierarchischen bayes'schen Modell $\psi \to \tilde{\theta} \to I$ wie in Abschnitt 1.2.2, wobei wir die Daten I, Parameter $\tilde{\theta}$ und Hyperparameter ψ betrachten. Damit folgt

$$\rho(I|\,\tilde{\theta}, \psi) = \rho(I|\,\tilde{\theta}) \quad \text{und} \quad \rho(\psi|\,\tilde{\theta}, I) = \rho(\psi|\,\tilde{\theta}).$$

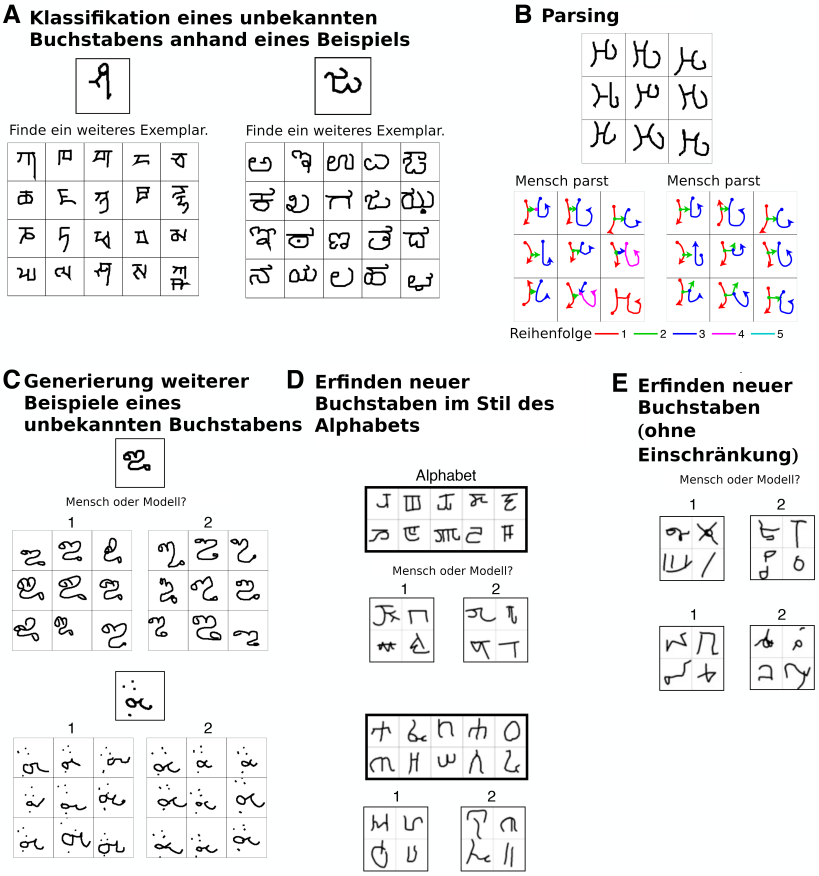

Abbildung 2.1.: Dargestellt werden die fünf Aufgaben, die auf dem Omniglot-Datensatz gestellt sind und vom selben Modell gelöst werden. Dazu gehören:
A Die Klassifikation anhand eines Beispiels, wobei ein unbekannter Buchstabe innerhalb der 20 Zeichen eines unbekannten Alphabets erkannt werden soll.
B Das Schließen einer plausiblen Zeichenanweisung. Dargestellt werden die Motorprogramme von verschiedenen menschlichen Probanden links sowie des Modells rechts, gegeben jeweils ein binäres Bild des Buchstabens.
C Die Generierung weiterer Beispiele eines unbekannten Buchstabens. Dabei hat jeweils eins der 3×3 Felder ein Mensch, und eins das Modell generiert.
D Das Erfinden vier neuer Buchstaben, die in den Stil der gegebenen zehn Buchstaben passen. Zu sehen ist auf den 2×2 Feldern jeweils die Ausgabe des Modells oder die erfundenen Buchstaben menschlicher Probanden.
E Das Erfinden vier neuer Buchstaben ohne Einschränkungen.
Dabei sind die vom Modell generierten Felder jeweils zeilenweise:
bei **C** 1,2; bei **D** 2,2; bei **E** 2,2. Abbildung modifiziert aus [LST19].

Die Forderung aus der Definition gilt mit Korollar 1.1 immer

$$\rho(\tilde{\theta}) = \int_{\Psi} \rho(\tilde{\theta}\,|\psi)\rho(\psi)\; d\psi.$$

Das Schema ψ ist eine Menge von *Teilen* $S = \{S_1, \ldots, S_\kappa\}$ und ihren *Relationen* $R = \{R_1, \ldots, R_\kappa\}$ zueinander. Dabei bestehen die Teile aus Primitiven. *Primitive* sind die elementaren Grundbestandteile und sind in einem Verzeichnis geordnet. Das variierte Schema $\tilde{\theta}$ enthält gestörte/variierte Schemaparameter. Zusätzlich dazu beschreibt $\tilde{\theta}$ die globale Bildtransformation. Das Bild I kommt deterministisch aus diesem variierten Schema $\tilde{\theta}$ hervor.

Notation 35. Im Folgenden werden wir mit $\psi \in \Psi$ stets das Schema, mit $\tilde{\theta} \in \Theta$ stets das variierte Schema und mit I das umgesetzte variierte Schema im Bildraum bezeichnen.

Beispiel 36. Zum Verständnis gehen wir diese Begriffe am Buchstabendatensatz Omniglot durch. Buchstaben werden als die Folge ihrer Zeichnung auf Papier gesehen. Die *Teile* sind die Striche zwischen dem Ansetzen des Stiftes bis zum Absetzen dieses. Pausen, wie starke Richtungswechsel bei der Zeichnung, unterteilen die Striche in die *Primitive*, die durch kubische Splines modelliert werden. Ein Strich steht zu den vorhergehenden in *Relation*. Es gibt vier Möglichkeiten: der neue Strich kann am Anfang eines vorherigen Strichs angesetzt sein, am Ende, entlang des Strichs oder unabhängig davon. Der Ablauf ist in Abbildung 2.2 schematisch dargestellt.

Bemerkung 37. Wir möchten kurz die Anzahl der Ebenen diskutieren, die das hierarchische Modell hat. Die gewählte Zahl *drei* bietet sich an, da wir damit das Schema, das variierte Schema und die Daten darstellen können. Unabhängig davon modellieren wir Schema ψ und variiertes Schema $\tilde{\theta}$ jeweils mehrstufig, denn diese bestehen aus mehreren Parametern, die wiederum bedingte Abhängigkeiten voneinander haben. Auf die jeweiligen multivariaten Verteilungen von ψ und $\tilde{\theta}$ gehen wir im nächsten Abschnitt 2.2.2 ein. Eine vierte Ebene über dem Schema ψ kann etwa verschiedene Schemas untereinander gruppieren. Für den Buchstabendatensatz wären das etwa Buchstaben desselben Stils, die zum selben Alphabet gehören. Dieser Gedanke wird in [Lak14] beschrieben. Eine mögliche fünfte Ebene wäre, diese Stilgruppen wiederum zu gruppieren.
In der Praxis haben hierarchische Modelle aber selten mehr als zwei Stufen [Gel+13], was zu einem Modell mit drei Ebenen wie bei uns passt. In der Praxis eignet es sich besonders, zwischen sogenannter informierter und uninformierter A-priori Verteilung zu unterscheiden. Dabei werden die strukturellen, objektiven Kriterien in einer anderen Ebene als die subjektiven Informationen modelliert. In einer gewissen Art und Weise setzen wir dies in der Modellierung um: wir unterscheiden zwischen Schema und den erlaubten Variationen.

2.2.2. Spezifikation der multivariaten Wahrscheinlichkeitsverteilungen

Die drei Ebenen $\psi, \tilde{\theta}, I$ werden probabilistisch modelliert. Dazu definieren wir die Verteilung $\psi = \{\kappa, S, R\}$ durch die Verteilungen der Teile $S = \{S_1, \ldots, S_\kappa\}$ und Relationen $R = \{R_1, \ldots, R_\kappa\}$. Wir definieren die folgende multivariate Verteilung für das Schema ψ:

$$\rho(\psi) = \rho(\kappa) \prod_{i=1}^{\kappa} \rho(S_i)\rho(R_i|S_1, \ldots, S_{i-1}). \tag{2.1}$$

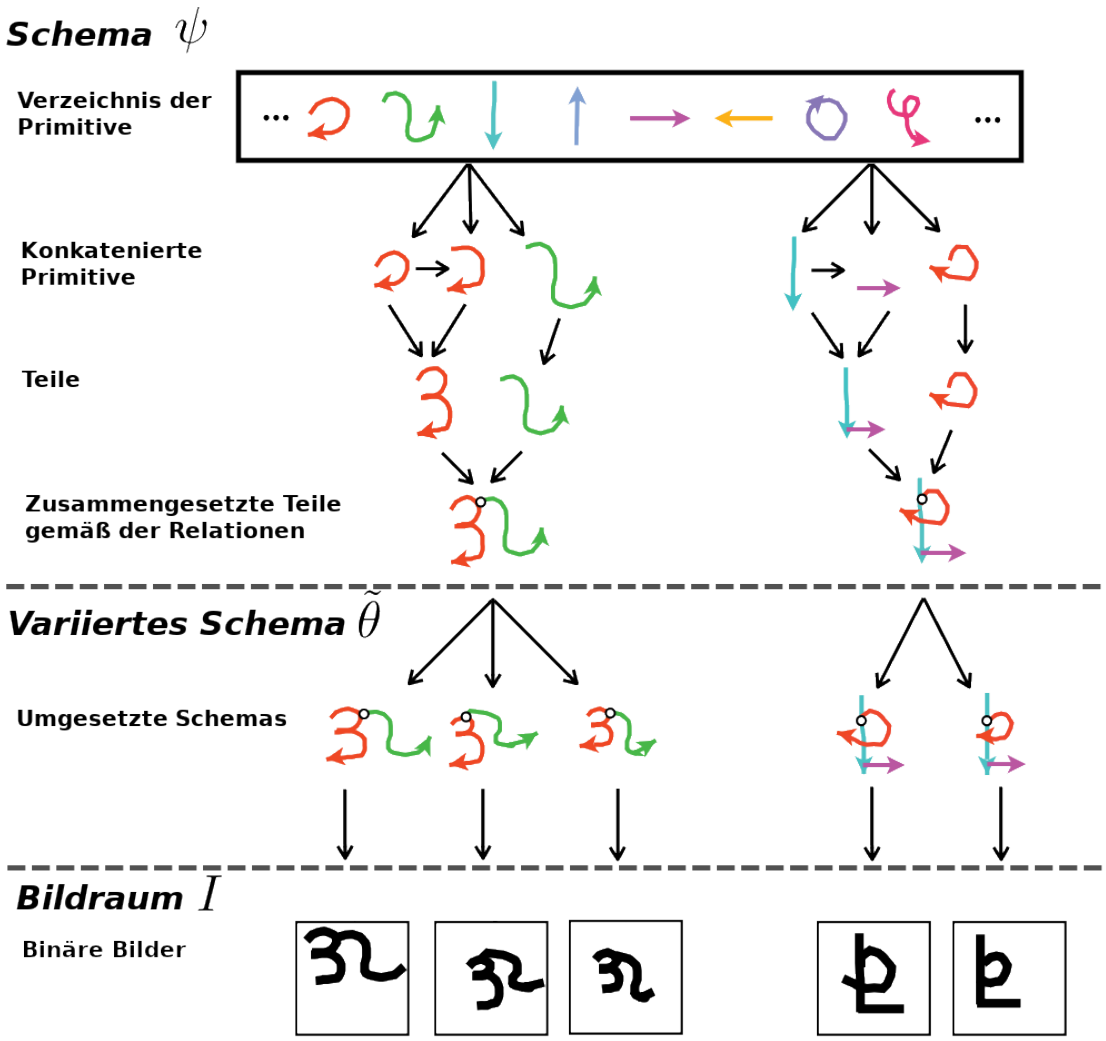

Abbildung 2.2.: Dargestellt wird das Konzept von zwei Buchstabens auf den drei Ebenen. Das Schema ψ besteht aus Primitiven, Teilen und ihren Relationen. Im oberen Kasten ist ein Ausschnitt aus dem Verzeichnis der Primitive dargestellt. Diese werden zu Teilen konkateniert. Die Teile, entsprechend der Relationen zusammengesetzt, bilden das Buchstabenschema.
Das variierte Schema $\tilde{\theta}$ setzt erlaubte Variabilität der skizzenhaften Beschreibung des Buchstabenschmas um. Etwa geht beim rechten Buchstaben der rote Kringel einmal deutlich über den senkrechten Strich, und einmal nicht.
Bei der Transformation in den Bildraum I kommt zusätzlich eine globale Bildtransformation dazu. Das variierte Schema, durch die variierten Teile und Relationen gegeben, wird deterministisch als binäres Bild umgesetzt. Abbildung modifiziert aus [LST15].

Die Verteilungen $\rho(S_i)$ und $\rho(R_i|S_1,\ldots,S_{i-1})$ sind dabei problemspezifisch und unter Beachtung der Primitive definiert.
Das variierte Schema $\tilde{\theta}$ enthält gestörte Äquivalente der Teile $\tilde{S} = \{\tilde{S}_1,\ldots,\tilde{S}_\kappa\}$ und Relationen $\tilde{R} = \{\tilde{R}_1,\ldots,\tilde{R}_\kappa\}$. Zusätzlich dazu enthält das variierte Schema $\tilde{\theta}$ noch eine Bildtransformationen $\tilde{B}$. Wir definieren für das variierte Schema $\tilde{\theta} = \{\tilde{S},\tilde{R},\tilde{B}\}$ die multivariate Verteilung

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{B}|\tilde{S},\tilde{R},\psi)\cdot\prod_{i=1}^{\kappa}\rho(\tilde{R}_i|R_i)\rho(\tilde{S}_i|S_i). \tag{2.2}$$

Wieder sind die Verteilungen von $\rho(\tilde{B}|\tilde{S},\tilde{R},\psi), \rho(\tilde{R}_i|R_i)$ und $\rho(\tilde{S}_i|S_i)$ problemspezifisch. Schließlich kommt I deterministisch aus $\tilde{\theta}$ hervor. Sei f diese deterministische Funktion. Damit erhalten wir

$$\rho(I|\,\tilde{\theta}^{(T)}) = \begin{cases} 1 & \text{für } f(\tilde{\theta}^{(T)}) = I \\ 0 & \text{sonst.} \end{cases} \tag{2.3}$$

Diese Verteilungen werden jeweils als probabilistisches Programm aufgefasst. Dies ermöglicht, neue Konzepte zu erfinden. Im Algorithmus 2 ist das zugehörige Programm zu der Gleichung 2.1 zu sehen. Mit $x \leftarrow \rho(x)$ ist im Folgenden stets gemeint, dass x einen gezogenen Wert gemäß der Verteilung $\rho(x)$ zugewiesen bekommt. Entsprechend gibt es auch Programme zu der Gleichung 2.2 beziehungsweise 2.3. Diese sind im Anhang C unter Algorithmus 4 und 5 zu finden, wobei die letztere der Umsetzung der deterministischen Funktion f entspricht.

Algorithmus 2 Generierung eines Schemas als Programm auf Basis der multivariaten Verteilung

$$\rho(\psi) = \rho(\kappa)\prod_{i=1}^{\kappa}\rho(S_i)\rho(R_i|S_1,\ldots,S_{i-1}).$$

```
1:  procedure GeneriereSchema
2:      κ ← ρ(κ)                          ▷ Ziehe Anzahl der Teile.
3:      for i = 1,...,κ do
4:          S_i ← ρ(S_i|i)                ▷ Ziehe das i-te Teil.
5:          R_i ← ρ(R_i|S_1,...,S_{i-1})  ▷ Ziehe Relation.
6:      end for
7:      S ← {S_1,...,S_κ}
8:      R ← {R_1,...,R_κ}
9:      ψ ← {κ,R,S}
10:     return ψ                          ▷ Fertig erstelltes Schema.
11: end procedure
```

2.2.3. Lernen der Modellparameter

Gegeben ist ein Datensatz von vielen verschiedenen Konzepten. Diese liegen jeweils als Schemas und variierte Schemas vor. Konkret haben wir für ein Konzept k immer mehrere Schemas $\psi_k^{(i)}$ gegeben und pro Schema $\psi_k^{(i)}$ wiederum verschiedene variierte Schema $\tilde{\theta}_k^{(i,j)}$. Anhand dieser sollen die Verteilungen von ψ und $\tilde{\theta}\,|\psi$ gelernt werden. Das bedeutet

bezüglich der Gleichung 2.1, dass $\rho(\kappa), \rho(S_i)$ und $\rho(R_i|S_1, \ldots, S_{i-1})$ gelernt werden. Diese Verteilungen bestimmen wir aus den verschiedenen Schemas $\psi_k^{(i)}$. Aus Gleichung 2.2 sollen schließlich $\rho(\tilde{B}|\tilde{S}, \tilde{R}, \psi), \rho(\tilde{R}_i|R_i)$ und $\rho(\tilde{S}_i|S_i)$ bestimmt werden. Dazu wird die Variabilität zwischen den verschiedenen variierten Schema $\tilde{\theta}_k^{(i,j)}$ zu festem Schema $\psi_k^{(i)}$ verglichen.
Das Lernen der Parameter kann durch Expertenwissen, (geglättete) empirische Werte aus den Daten, Clusteranalysen wie sogenannte Gauß'sche Mischmodelle [HTF09] oder ähnliche Methoden passieren.

Beispiel 38. Wozu das Lernen der Modellparameter dient, zeigen wir am Beispiel von Buchstaben. Das Modell soll später die essentiellen Schlüsselcharakteristika und erlaubte Variabilität innerhalb unbekannter Buchstaben erkennen. Dazu vergleicht er die Freiheitsgrade für Schlüsselcharakteristika und Variabilität für den Datensatz, um das Wissen dann später auf unbekannte Buchstaben zu übertragen.
Sieht das Modell schließlich einen unbekannten Buchstaben, sagen wir ein $\mathcal{F}$, so erkennt es, dass in der Zeichnung ein gewisser Spielraum ist. Etwa können die beiden waagerechten Striche etwas länger, kürzer oder schräger sein. Gleichzeitig müssen die Schlüsselcharakteristika erhalten werden. Insbesondere dürfen die beiden waagerechten Striche nicht so schräg werden, dass sie sich berühren und wir ein $\mathcal{P}$ erhalten. Dabei hat das Modell niemals ein $\mathcal{F}$ oder $\mathcal{P}$ gesehen, wohl aber andere Buchstaben (zum Beispiel das japanische, hebräische und burmesische Alphabet). Diese Übertragung von Erfahrungen wenden Menschen ständig an und ist ein wichtiges Kernprinzip vom Maschinellen Lernen mit Bayes'schen Programmen.

2.2.4. Parsing der Daten

Das Objekt ist über drei Ebenen $\psi, \tilde{\theta}, I$ beschrieben. Gleichzeitig sind alle Aufgaben im Bildraum zu lösen. Das bedeutet, dass wir gegeben I passende ψ und $\tilde{\theta}$ finden müssen, so dass $f(\tilde{\theta}) = I$ gilt.
Dies nennen wir *Parsing*. Der Parser schlägt gegeben I für das Schema und variierte Schema Vorschläge $\psi^{[1]}, \tilde{\theta}^{[1]}, \ldots, \psi^{[K]}, \tilde{\theta}^{[K]}$ vor. Für diese Paare von Vorschlägen ist jeweils $\rho(\psi^{[i]}|\, \tilde{\theta}^{[i]})$ hoch und es gilt jeweils $f(\tilde{\theta}^{[i]}) \approx I$. Diese Vorschläge beschreiben eine mögliche Zerlegung des Bildes in die Teile und Relationen. Um die Zerlegung zu verbessern wollen wir, gegeben $\tilde{\theta}^{[i]}$, passende Vorschläge für ψ bekommen. Das entspricht der Stichprobenentnahme aus $\rho(\psi|\, \tilde{\theta}^{[i]})$ und führt zu N Vorschlägen $\psi^{[i,1]}, \ldots, \psi^{[i,N]}$.
Um das variierte Schema $\tilde{\theta}$ optimaler Bewertung zu finden werden die Vorschläge des Parsers respektive des Bildes optimiert. Die Optimierung kann auch bezüglich eines anderen Bildes stattfinden.
Das Parsing erlaubt, dass wir Bilder auf der Ebene des Schemas vergleichen. So können die Schlüsselcharakteristika der Konzepte verglichen werden, ohne dass das erlaubte Maß an Variabilität zwischen den umgesetzten Objekten beachtet wird.

2.2.5. Bewertung der Parses und Lösen der Aufgaben

Anhand der Vorschläge des Parsers möchten wir $\rho(\psi, \tilde{\theta}\,|I)$ berechnen können. Dafür diskretisieren wir diese Verteilung an den Vorschlägen, die der Parser liefert. Dazu werden wir in Abschnitt 2.3.1 eine Approximation herleiten.

Das Modell soll dann die folgenden Aufgaben lösen können, die in Abbildung 2.1 für den Omniglot-Datensatz dargestellt sind:

i) Die Zerlegung von Konzepten in ihre Teile und Relationen (**B** in der Abbildung 2.1).

ii) Die Klassifikation anhand eines Beispiels eines unbekannten Buchstabens (**A**). Dafür müssen wir für zwei verschiedene Bilder $I^{(1)}$ und $I^{(2)}$ den Term $\rho(I^{(1)}|I^{(2)})$ auswerten.

iii) Das Generieren weiterer Beispiele des Konzeptes anhand eines Bildes (**C**). Dazu müssen wir aus $\rho(I^{(2)}, \tilde{\theta}^{(2)}\,|I^{(1)})$ Stichproben ziehen.

iv) Das Erfinden eines neuen Konzepts (**E**). Das entspricht dem Ziehen einer Stichprobe aus $\rho(\psi)$ und $\rho(\tilde{\theta}\,|\psi)$. Da wir die multivariate Verteilung als Programm dargestellt haben, ist das äquivalent zur Ausführung der probabilistischen Programme.

Auf Abbildung 2.1 ist zusätzlich noch das Erfinden eines neues Konzepts, das zu einer Klasse von Konzepten passt gegeben (**D**). Darauf gehen wir nicht im Detail ein, kommentieren es aber im Abschnitt 2.3.3.

2.2.6. Zusammenfassung des Rahmenwerks

Im Schema 3 sind die allgemeinen Bestandteile des Rahmenwerks von Maschinellen Lernen mit Bayes'schen Programmen dargestellt. Diese Schritte führen wir in Kapitel 3 ausführlich auf dem Omniglot-Datensatz aus.

2.2.7. Kernprinzipien

Drei Kernprinzipien sind im Design des Algorithmus'schema 3 besonders wichtig. Das sind:

Kompositionalität Ein Objekt besteht aus verschiedenen Teilen. Die erlaubten Teile und ihre Relationen werden gelernt. Insbesondere wird ein Verzeichnis der möglichen Primitive gelernt. Beim Parsing wird das Objekt in diese elementaren Bestandteile zerlegt. Dadurch ist das Konzept hierarchisch aufgebaut.

Kausalität Die Konstruktion des Modells beachtet die unterliegende Kausalität innerhalb eines Konzepts. Zum Beispiel wird ein Buchstabe als die Folge der Zeichnung, zum Beispiel durch einen Stift auf Papier, gesehen. Dieser generative Prozess wird im Parsing nachvollzogen. Diese Betrachtung ermöglicht Variabilität innerhalb dieses generativen Prozesses.

Übertragen von Erfahrungen Zuerst wird anhand eines Datensatzes die Freiheitsgrade der Schlüsselcharakteristika sowie der Variabilität in der Umsetzung dieser gelernt. Beim Betrachten neuer Bilder wird diese Erfahrung auf ein unbekanntes Konzept übertragen.

Algorithmus'schema 3 Allgemeines Schema mit den wesentlichen Schritten von *Maschinellem Lernen mit Bayes'schen Programmen.*

procedure MASCHINELLES LERNEN MIT BAYES'SCHEN PROGRAMMEN
    Start Spezifiziere die drei Ebenen des Objekts.
        Identifiziere die Teile, aus denen das Objekt besteht.
        Identifiziere die Primitive, aus denen Teile bestehen.
        Identifiziere die Relationen, in denen die Teile zueinander stehen.
    Ende
    Start Modelliere die drei Ebenen probabilistisch.
        Spezifiziere $\rho(\psi)$ auf Basis der Primitive, Teile und Relationen.
        Spezifiziere $\rho(\tilde{\theta})$ auf Basis von ψ und der globalen Bildtransformation.
        Beschreibe den deterministischen Übergang von $\tilde{\theta}$ in den Bildraum I.
        Schreibe diese drei Wahrscheinlichkeitsverteilungen als Programm auf.
    Ende
    Start Lerne die Parameter Wahrscheinlichkeitsverteilungen.
        Lerne anhand des Datensatzes die Parameter von $\rho(\psi)$ und $\rho(\psi|\,\tilde{\theta})$.
    Ende
    Start Parsing der Daten.
        Definiere einen Parser, um gegeben I passende $\psi, \tilde{\theta}$ vorzuschlagen.
        Spezifiziere, wie du gegeben $\tilde{\theta}$ passende ψ vorschlagen kannst.
    Ende
    Start Bewertung der Parses und Lösen der Aufgaben.
        Spezifiziere die Schätzung von $\rho(\psi, \tilde{\theta}\,|I)$ anhand der Vorschläge des Parsers.
        Löse anhand der Bewertung die Aufgaben.
    Ende
end procedure

2.3. Modellbewertung

Fassen wir die Aussagen aus Abschnitt 2.2 einmal zusammen. Wir haben ein hierarchisches bayes'sches Modell mit drei Ebenen $\psi \to \tilde{\theta} \to I$. Darauf gelten folgende zwei Aussagen. wobei $\tilde{\theta}^{(T)}$ wieder ein variiertes Schema mit $I = f(\tilde{\theta}^{(T)})$ ist:

$$\begin{aligned}\rho(\psi|\,\tilde{\theta}, I) =& \rho(\psi|\,\tilde{\theta})\\ \rho(I|\,\tilde{\theta}, \psi) =& \rho(I|\,\tilde{\theta}).\end{aligned}$$

Außerdem gehen wir davon aus, dass wir die Terme $\rho(\psi), \rho(\tilde{\theta}\,|\psi)$ und $\rho(I|\,\tilde{\theta})$ für konkrete Werte für $\psi, \tilde{\theta}, I$ auswerten können. Das Parsing erlaubt uns, Stichproben aus $\rho(\tilde{\theta}\,|I)$ und $\rho(\psi|\,\tilde{\theta})$ zu ziehen.

2.3.1. Diskretisierung der Verteilungen

Wir möchten verschiedene Verteilungen wie $\rho(I)$ und $\rho(\psi, \tilde{\theta}\,|I)$ ausrechnen können. Diese Terme sind kompliziert, und wir möchten sie durch die Vorschläge des Parsers approximieren.

Dieser Parser liefere uns K Vorschläge für ψ und $\tilde{\theta}$, was zu $\psi^{[1]}, \tilde{\theta}^{[1]}, \ldots, \psi^{[K]}, \tilde{\theta}^{[K]}$ führt.

Dabei gilt jeweils $f(\tilde{\theta}^{[i]}) \approx I$, wobei wir eine sehr kleine Fehlerschranke erlauben.
Um die Variabilität von den Schlüsselcharakteristika zu unterscheiden, ziehen wir von jeden der Vorschläge $\tilde{\theta}^{[i]}$ jeweils N weitere Vorschläge für ψ aus $\rho(\psi|\,\tilde{\theta}^{[i]})$. Dies führt zu insgesamt $K \cdot N$ Vorschlägen für ψ, eben $\psi^{[1,1]}, \ldots, \psi^{[1,N]}, \ldots\ldots, \psi^{[K,1]}, \ldots, \psi^{[K,N]}$. Wir nehmen an, dass unsere Vorschläge den gesamten Raum hinreichend gut approximieren, siehe dazu auch Bemerkung 39. Konkret fordern wir, dass die gefundenen variierten Schemas alle $\tilde{\theta}$ mit $f(\tilde{\theta}) = I$ sind, also

$$\rho(I|\,\tilde{\theta}) \approx \sum_{i=1}^{K} \rho(I|\,\tilde{\theta}^{[i]})\delta(\tilde{\theta} - \tilde{\theta}^{[i]}) \tag{2.4}$$

gilt, sowie wir alle wahrscheinlichen ψ gefunden haben, also gilt

$$\rho(\tilde{\theta}^{[i]}, \psi) = \rho(\tilde{\theta}^{[i]}\,|\psi)\rho(\psi) \approx \frac{1}{N}\sum_{j=1}^{N} \rho(\tilde{\theta}^{[i]}\,|\psi^{[i,j]})\rho(\psi^{[i,j]})\delta(\psi - \psi^{[i,j]}) \tag{2.5}$$

$$\approx \rho(\tilde{\theta}^{[i]}\,|\psi^{[i]})\rho(\psi^{[i]})\frac{1}{N}\sum_{j=1}^{N} \delta(\psi - \psi^{[i,j]}). \tag{2.6}$$

Dann gilt der folgende Satz 40, womit wir $\rho(\psi, \tilde{\theta}\,|I)$ ausrechnen können.

Bemerkung 39. Die beiden Forderungen, dass unsere gefundenen $\tilde{\theta}$ und ψ den Raum approximieren, sind sehr stark. Um dies zu gewährleisten, werden wir im Parsing die gefundenen Vorschläge stark optimieren. Für $\rho(\psi^{[i]}, \tilde{\theta}^{[i]}, I)$ mit hoher Wahrscheinlichkeit müssen die restlichen Möglichkeiten für ψ und $\tilde{\theta}$ eine so kleine Wahrscheinlichkeit haben, dass die Diskretisierung der Integrale legitim ist. Für die Gleichung 2.4 ist

$$\rho(I) = \int_{\Theta} \rho(I|\,\tilde{\theta})\rho(\tilde{\theta})\; d\,\tilde{\theta} = \int_{\Theta} \mathbb{1}_{f(\tilde{\theta})=I}\,\rho(\tilde{\theta})\; d\,\tilde{\theta},$$

wobei $\rho(I|\,\tilde{\theta})$ nur nicht null ist, falls $f(\tilde{\theta}) = I$ gilt. Wir schätzen also, dass wir alle wahrscheinlichen $\tilde{\theta}$ gefunden haben. Für die zweite Gleichung zu $\rho(\tilde{\theta}, \psi)$ gilt, dass

$$\rho(\tilde{\theta}) = \int_{\psi} \rho(\tilde{\theta}\,|\psi)\rho(\psi)\; d\psi.$$

Satz 40. *Es gelten die Voraussetzungen und Bezeichnungen wie oben. Dann gilt für* $\rho(\psi, \tilde{\theta}\,|I)$ *die Approximation*

$$\rho(\psi, \tilde{\theta}\,|I) \approx \frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N} \omega_i \delta(\tilde{\theta} - \tilde{\theta}^{[i]})\delta(\psi - \psi^{[i,j]}),$$

wobei

$$\omega_i \propto \tilde{\omega_i} = \rho(\psi^{[i]}, \tilde{\theta}^{[i]}, I) \qquad \textit{unter der Nebenbedingung, dass} \quad \sum_{i=1}^{n} \omega_i = 1.$$

Beweis. Nach Satz 26 gilt zunächst

$$\rho(\psi, \tilde{\theta}\,|I) = \frac{\rho(\psi, \tilde{\theta}, I)}{\rho(I)} = \frac{\rho(I|\,\tilde{\theta})\rho(\tilde{\theta}\,|\psi)\rho(\psi)}{\rho(I)}.$$

Damit folgt direkt

$$\begin{aligned}
\rho(\psi, \tilde{\theta}\,|I) &= \frac{\rho(I|\,\tilde{\theta})\rho(\tilde{\theta}\,|\psi)\rho(\psi)}{\rho(I)} \\
&\overset{Gl.\ 2.4}{\approx} \frac{\sum_{i=1}^{K} \rho(I|\,\tilde{\theta}^{[i]})\delta(\tilde{\theta}-\tilde{\theta}^{[i]})\rho(\tilde{\theta}^{[i]}\,|\psi)\rho(\psi)}{\rho(I)} \\
&\overset{Gl.\ 2.5}{\approx} \frac{\sum_{i=1}^{K} \rho(I|\,\tilde{\theta}^{[i]})\delta(\tilde{\theta}-\tilde{\theta}^{[i]})\frac{1}{N}\sum_{j=1}^{N} \rho(\tilde{\theta}^{[i]}\,|\psi^{[i,j]})\rho(\psi^{[i,j]})\delta(\psi-\psi^{[i,j]})}{\rho(I)} \\
&\overset{Gl.\ 2.6}{\approx} \frac{\sum_{i=1}^{K} \rho(I|\,\tilde{\theta}^{[i]})\delta(\tilde{\theta}-\tilde{\theta}^{[i]})\frac{1}{N}\rho(\tilde{\theta}^{[i]}\,|\psi^{[i]})\rho(\psi^{[i]})\sum_{j=1}^{N} \delta(\psi-\psi^{[i,j]})}{\rho(I)} \\
&\approx \frac{\frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N} \rho(\psi^{[i]}, \tilde{\theta}^{[i]}, I)\delta(\tilde{\theta}-\tilde{\theta}^{[i]})\delta(\psi-\psi^{[i,j]})}{\rho(I)}.
\end{aligned}$$

Die Aussage folgt für ω_i mit

$$\omega_i \propto \tilde{\omega}_i = \rho(\psi^{[i]}, \tilde{\theta}^{[i]}, I) \qquad \text{sowie} \qquad \sum_{i=1}^{N} \omega_i = 1.$$

□

Definition 41. Wir definieren Q als Approximation von $\rho(\psi, \tilde{\theta}\,|I)$

$$Q(\psi, \tilde{\theta}, I) = \sum_{i=1}^{K} \omega_i \delta(\tilde{\theta}-\tilde{\theta}^{[i]}) \frac{1}{N} \sum_{j=1}^{N} \delta(\psi-\psi^{[ij]}).$$

Damit haben wir eine Berechnungsmöglichkeit für $\rho(\psi, \tilde{\theta}\,|I)$ gefunden, die wir brauchen, um gefundene Parameter überhaupt bewerten zu können. Es folgt mit der Beschreibung der $\tilde{\omega}_i$ das folgende Korollar für $\rho(I)$.

Korollar 42. *Es gilt in der Situation von oben für $\rho(I)$ die Approximation*

$$\rho(I) \approx \sum_i \tilde{\omega}_i = \sum_i \rho(\psi^{[i]}, \tilde{\theta}^{[i]}, I).$$

Beweis. Aus dem Beweis von der eben bewiesenen Approximation in Satz 40 gilt direkt:

$$1 = \sum_{i=1}^{N} \omega_i = \sum_{i=1}^{N} \frac{\tilde{\omega}_i}{\rho(I)} \iff \rho(I) = \sum_{i=1}^{N} \tilde{\omega}_i.$$

□

2.3.2. Klassifkation anhand eines Beispiels

Nun möchten wir die Aufgaben lösen. Wir beginnen mit der Klassifikation eines unbekannten Konzepts anhand eines Beispiels.
Gegeben sei ein Testbild $I^{(T)}$ und C Bilder von unbekannten Konzepten $I^{(c)}$, $c = 1, \ldots, C$,

wobei eines davon auch eine Realisierung von $I^{(T)}$ darstellt. Wir suchen den passendsten Index c. Dies entspricht Aufgabe **A** von Abbildung 2.1. Wir schätzen also

$$\arg\max_c \rho(I^{(T)}|I^{(c)}), \tag{2.7}$$

wobei $\rho(I^{(T)}|I^{(c)})$ konkrete Zahlenwerte für die C Klassenbilder $I^{(1)}, \ldots, I^{(C)}$ sind.
In der Implementierung des Algorithmus werden wir nicht nur $\rho(I^{(T)}|I^{(c)})$ betrachten, sondern auch $\rho(I^{(c)}|I^{(T)})$. Bevor wir die dafür benötigte äquivalente Darstellung von Gleichung 2.7 in Lemma 42 beweisen, betrachten wir erst, wie wir $\rho(I^{(T)}|I^{(c)})$ schätzen können. Dabei bezeichnen im Folgenden die hochgestellten Indizes, ob wir uns auf eines der Klassenbilder $\psi^{(c)}, \tilde{\theta}^{(c)}, I^{(c)}$ beziehungsweise auf das Testbild $\psi^{(T)}, \tilde{\theta}^{(T)}, I^{(T)}$ beziehen.

Satz 43. *Es gilt*

$$\rho(I^{(T)}|I^{(c)}) \approx \sum_{i=1}^{K} \omega_i^{(c)} \max_{\tilde{\theta}^{(T)}} \Big(\rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \frac{1}{N} \sum_{j=1}^{N} \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)[ij]})\Big).$$

Beweis. Wir können den Term $\rho(I^{(T)}, \tilde{\theta}^{(T)}, \tilde{\theta}^{(c)}, \psi^{(c)}|I^{(c)})$ mit Korollar 15 und Q aus Definition 41 umformen zu

$$\begin{aligned}
\rho(I^{(T)}, \tilde{\theta}^{(T)}, \tilde{\theta}^{(c)}, \psi^{(c)}|I^{(c)}) &= \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \cdot \rho(\tilde{\theta}^{(T)}, \tilde{\theta}^{(c)}, \psi^{(c)}|I^{(c)}) \\
&= \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \cdot \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot \rho(\tilde{\theta}^{(c)}, \psi^{(c)}|I^{(c)}) \\
&\approx \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \cdot \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot Q(\tilde{\theta}^{(c)}, \psi^{(c)}, I^{(c)}).
\end{aligned}$$

Über die kontinuierliche Form vom Satz der totalen Wahrscheinlichkeit in Lemma 16 folgt, siehe auch [Lak14]:

$$\begin{aligned}
&\rho(I^{(T)}|I^{(c)}) \\
&\overset{16}{=} \int_{\Theta^{(T)}} \int_{\Psi^{(c)}} \int_{\Theta^{(c)}} \rho(I^{(T)}, \tilde{\theta}^{(T)}, \tilde{\theta}^{(c)}, \psi^{(c)}|I^{(c)})\; d\,\tilde{\theta}^{(c)}\; d\psi^{(c)}\; d\,\tilde{\theta}^{(T)} \\
&\approx \int_{\Theta^{(T)}} \int_{\Psi^{(c)}} \int_{\Theta^{(c)}} \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \cdot \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot Q(\tilde{\theta}^{(c)}, \psi^{(c)}, I^{(c)})\; d\psi^{(c)}\; d\,\tilde{\theta}^{(c)}\; d\,\tilde{\theta}^{(T)} \\
&= \int_{\Theta^{(T)}} \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \Big[\int_{\Psi^{(c)}} \int_{\Theta^{(c)}} \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot Q(\tilde{\theta}^{(c)}, \psi^{(c)}, I^{(c)})\; d\psi^{(c)}\; d\,\tilde{\theta}^{(c)} \Big]\; d\,\tilde{\theta}^{(T)}\,.
\end{aligned}$$

Nach der Diskretisierung von Q gilt für festes $\tilde{\theta}^{(T)}$ für Stichproben $\psi^{(c)[ij]}$

$$\begin{aligned}
&\int_{\Psi^{(c)}} \int_{\Theta^{(c)}} \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot Q(\tilde{\theta}^{(c)}, \psi^{(c)}, I^{(c)})\; d\psi^{(c)}\; d\,\tilde{\theta}^{(c)} \\
&= \int_{\Psi^{(c)}} \int_{\Theta^{(c)}} \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)}) \cdot \frac{1}{N} \sum_{i=1}^{K} \sum_{j=1}^{N} \omega_i^{(c)} \delta(\tilde{\theta} - \tilde{\theta}^{(c)[i]}) \delta(\psi - \psi^{(c)[i,j]})\; d\psi^{(c)}\; d\,\tilde{\theta}^{(c)} \\
&= \sum_{i=1}^{K} \frac{1}{N} \cdot \omega_i^{(c)} \cdot \sum_{j=1}^{N} \rho(\tilde{\theta}^{(T)}\,|\psi^{(c)[ij]}).
\end{aligned}$$

Wegen der Dirac-Maße $\delta(\tilde{\theta} - \tilde{\theta}^{(c)[i]})\delta(\psi - \psi^{(c)[i,j]})$ ist der Term im Doppelintegral entweder Null oder $\frac{1}{N} \cdot \omega_i^{(c)} \cdot \rho(\tilde{\theta}^{(T)} | \psi^{(c)[i,j]})$.
Damit können wir $\rho(I^{(T)}|I^{(c)})$ approximieren via

$$\begin{aligned}
\rho(I^{(T)}|I^{(c)}) &\approx \int_{\Theta^{(T)}} \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \Big[\sum_{i=1}^{K} \frac{1}{N} \cdot \omega_i^{(c)} \cdot \sum_{j=1}^{N} \rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]}) \Big] \, d\,\tilde{\theta}^{(T)} \\
&= \sum_{i=1}^{K} \omega_i^{(c)} \int_{\Theta^{(T)}} \rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \frac{1}{N} \sum_{j=1}^{N} \rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]}) \, d\,\tilde{\theta}^{(T)} \\
&\approx \sum_{i=1}^{K} \omega_i^{(c)} \max_{\tilde{\theta}^{(T)}} \Big(\rho(I^{(T)}|\,\tilde{\theta}^{(T)}) \frac{1}{N} \sum_{j=1}^{N} \rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]}) \Big).
\end{aligned}$$

Die letzte Approximation folgt dabei aus kleinen Werten für $\rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]})$ für unpassende $\tilde{\theta}^{(c)}$. Das Maximum wird dabei per Re-Optimierung n von $\theta^{(T)}$ besteht. Dazu werden die kontinuierlichen Variablen der K Parses von $I^{(T)}$ über einen Gradientenaufstieg [Jos19, § 6.2] maximiert. □

Bemerkung 44. In dieser Gleichung kommt der Term $\rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]})$ vor. Wir müssen also zusätzlich die Parameter von $\tilde{\theta}^{(T)}$ bezüglich $\rho(\tilde{\theta}^{(T)} | \psi^{(c)[ij]})$ re-optimieren. Diese Re-Optimierung ist ein Knackpunkt, wieso die Klassifikation eines unbekannten Buchstabens anhand eines Beispiels so gut funktioniert. Es wird versucht, ein variiertes Schema $\tilde{\theta}^{(T)}$ zu finden, dass zu den möglichen Schemas $\psi^{(c)[ij]}$ passt.
Im Beispiel der Buchstaben beschreibt das Schema eine Zeichenanweisung. Es wird folglich versucht, das Bild, mit dem verglichen wird, mit der Zeichenanweisung des ersten Bilds zu zeichnen. Der Wert $\rho(I^{(T)}|I^{(c)})$ approximiert, wie gut dies funktioniert.

Bemerkung 45. In dem wir sowohl die Zeichenanweisung von $I^{(c)}$ bezüglich $I^{(T)}$ re-optimieren, als auch andersherum, verbessern wir die Approximation. Dies haben auch Pilotversuche von Lake in [Lak14] gezeigt. Daher verwendet die Implementierung folgende äquivalente Formulierung von $\arg\max_c \rho(I^{(T)}|I^{(c)})$, die wir in Lemma 46 beweisen.

Lemma 46. *Es ist*

$$\arg\max_c \rho(I^{(T)}|I^{(c)}) = \arg\max_c \ln[\frac{\rho(I^{(c)}|I^{(T)})}{\rho(I^{(c)})} \rho(I^{(T)}|I^{(c)})].$$

Beweis. Wir formen die Gleichung um:

$$\begin{aligned}
\arg\max_c \rho(I^{(T)}|I^{(c)}) &= \arg\max_c \ln \rho(I^{(T)}|I^{(c)}) \\
&= \arg\max_c \ln[\rho(I^{(T)}|I^{(c)})]^2 \\
&= \arg\max_c \ln[\frac{\rho(I^{(c)}|I^{(T)})}{\rho(I^{(c)})} \rho(I^{(T)}) \rho(I^{(T)}|I^{(c)})] \\
&= \arg\max_c \ln[\frac{\rho(I^{(c)}|I^{(T)})}{\rho(I^{(c)})} \rho(I^{(T)}|I^{(c)})].
\end{aligned}$$

Da die Funktionen ln und $x \mapsto x^2$ streng monoton wachsend auf $(0, \infty)$ sind und ρ nichtnegativ ist, können wir die ersten beiden Umformungsschritte machen. Der dritte Schritt gilt aufgrund von Korollar 15. Der letzte Schritt liegt daran, dass von wir über $c = 1, \ldots, C$ maximieren und der Ausdruck daher nicht von $\rho(I^{(T)})$ abhängt. □

Den Term

$$\arg\max_c \ln[\frac{\rho(I^{(c)}|I^{(T)})}{\rho(I^{(c)})}\rho(I^{(T)}|I^{(c)})]$$

können wir durch Korollar 42 sowie doppelte Anwendung von Satz 43 bestimmen. Da wir dadurch sowohl $\rho(\tilde{\theta}^{(T)}\,|\psi^{(c)[ij]})$ als auch $\rho(\tilde{\theta}^{(c)}\,|\psi^{(T)[ij]})$ betrachten, und für das Bestimmen des Maximums in der Formel aus Satz 43 optimieren, ist diese äquivalente Betrachtung stabiler.

2.3.3. Generierung neuer Beispiele

Eine weitere Aufgabe ist die Generierung neuer Beispiele eines unbekannten Konzepts. Dies entspricht Part **C** der Abbildung 2.1.
Gegeben Bild $I^{(1)}$ suchen wir ein weiteres Beispiels des abgebildeten Konzepts $I^{(2)}$. Dies entspricht dem Ziehen einer Stichprobe gemäß $\rho(I^{(2)}, \tilde{\theta}^{(2)}\,|I^{(1)})$. Dazu schreiben wir den Term wieder um.

Satz 47. *Es gilt*

$$\rho(I^{(2)}, \tilde{\theta}^{(2)}\,|I^{(1)}) = \sum_{i=1}^{K}\sum_{j=1}^{N}\omega_i^{(1)}\frac{1}{N}\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\psi^{(1)[ij]}).$$

Beweis. Wir integrieren wieder mit Korollar 16 und wenden wiederholt Korollar 15 an:

$$\begin{aligned}
&\rho(I^{(2)}, \tilde{\theta}^{(2)}\,|I^{(1)})\\
=&\int_{\Psi^{(1)}}\int_{\Theta^{(1)}}\rho(I^{(2)}, \tilde{\theta}^{(2)}, \tilde{\theta}^{(1)}, \psi^{(1)}|I^{(1)})\;d\psi^{(1)}\;d\,\tilde{\theta}^{(1)}\\
=&\int_{\Psi^{(1)}}\int_{\Theta^{(1)}}\rho(I^{(2)}, \tilde{\theta}^{(2)}\,|\,\tilde{\theta}^{(1)}, \psi^{(1)})\rho(\tilde{\theta}^{(1)}, \psi^{(1)}|I^{(1)})\;d\psi^{(1)}\;d\,\tilde{\theta}^{(1)}\\
=&\int_{\Psi^{(1)}}\int_{\Theta^{(1)}}\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\,\tilde{\theta}^{(1)}, \psi^{(1)})\rho(\tilde{\theta}^{(1)}, \psi^{(1)}|I^{(1)})\;d\psi^{(1)}\;d\,\tilde{\theta}^{(1)}\\
\approx&\int_{\Psi^{(1)}}\int_{\Theta^{(1)}}\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\psi^{(1)})Q(\tilde{\theta}^{(1)}, \psi^{(1)}, I^{(1)})\;d\psi^{(1)}\;d\,\tilde{\theta}^{(1)}\\
=&\int_{\Psi^{(1)}}\int_{\Theta^{(1)}}\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\psi^{(1)})\frac{\omega_i^{(1)}}{N}\sum_{i=1}^{K}\sum_{j=1}^{N}\delta(\tilde{\theta}-\tilde{\theta}^{(1)[i]})\delta(\psi-\psi^{(1)[ij]})\;d\psi^{(1)}\;d\,\tilde{\theta}^{(1)}\\
=&\frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{N}\omega_i^{(1)}\cdot\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\psi^{(1)[ij]}).
\end{aligned}$$

Wie im Beweis von Satz 43 ist im letzten Schritt wegen der Dirac-Maße $\delta(\tilde{\theta}-\tilde{\theta}^{(1)[i]})\delta(\psi-\psi^{(1)[i,j]})$ der Term im Doppelintegral entweder Null oder $\frac{1}{N}\cdot\omega_i^{(1)}\cdot\rho(I^{(2)}|\,\tilde{\theta}^{(2)})\rho(\tilde{\theta}^{(2)}\,|\psi^{(1)[ij]})$. □

Insgesamt können wir mit dieser Formel aus Satz 47 ein weiteres Beispiel eines unbekannten Konzepts generieren. Um mit $I^{(2)}$ zu vergleichen, werden von $I^{(1)}$ insgesamt K Parses gesammelt. Diese haben K verschiedene Bewertungen gemäß Q. Anstatt dann direkt $\rho(I^{(2)}, \tilde{\theta}^{(2)} | I^{(1)})$ mit der Formel aus Satz 47 zu berechnen, werden die Parses noch umgeordnet, um eine größere Vielfalt zu erzeugen. Konkret werden die K Schemas $\tilde{\theta}^{[i]}$ nach der Bewertung sortiert, was in einer Permutation σ gespeichert wird, wobei $\sigma(1)$ zu der besten Bewertung $\omega_i^{(1)}$ korrespondiert. Dann wird die Stichprobe gemäß folgender modifizierten Formel gezogen.

$$\rho(I^{(2)}, \tilde{\theta}^{(2)} | I^{(1)}) \approx \frac{1}{\sum_{k=1}^{K} \frac{1}{\sigma(k)}} \frac{1}{N} \sum_{i=1}^{K} \frac{1}{\sigma(i)} \sum_{j=1}^{N} \rho(I^{(2)} | \tilde{\theta}^{(2)}) \rho(\tilde{\theta}^{(2)} | \psi^{(1)[ij]}).$$

Diese Herangehensweise mindert Dominanzeffekte des am besten bewerteten Parses ab. Dieser wird nicht mehr so stark gegenüber den schlechter bewerteten Parses bevorzugt, was zu vielfältigeren Zeichenanweisungen führt.

2.3.4. Erfinden neuer Konzepte

Zuletzt betrachten wir die Aufgabe, neue Konzepte zu erfinden. Part **E** von Abbildung 2.1 bezieht sich dabei auf das Erfinden ohne Einschränkung. In Part **D** sind bereits einige Konzepte gegeben. Gesucht ist ein weiteres Konzept, das in den Stil der gegebenen Konzepte passt.

Das Erfinden eines Konzeptes entspricht in unserer Modellierung dem Ziehen von Stichproben aus $\rho(\psi)$. Dies entspricht schlicht der Ausführung des Programms aus Algorithmus 2. Dies führt zu einem Schema ψ. Dieses realisieren wir, in dem wir $\tilde{\theta}$ durch Stichprobenziehen aus $\rho(\tilde{\theta} | \psi)$ bekommen und das Bild I deterministisch aus $\tilde{\theta}$ erzeugen.

Bei der zweiten Aufgabe wollen wir noch den Stil von gegebenen Konzepten $I^{(1)}, \ldots, I^{(J)}$ beachten. Dies entspricht tatsächlich einer weiteren Stufe des hierarchischen bayes'schen Modells A, der wiederum Hyperparameter zu ψ ist. Sie modelliert die Variabilität und Ähnlichkeit innerhalb eines Alphabetes. In Abhängigkeit von A wird dann ein Schema ψ gezogen.
Um die Aufgabe zum Erfinden eines neuen Konzepts im Stil unbekannter Buchstaben zu lösen, wollen wir insgesamt eine Stichprobe aus $\rho(I^{(J+1)} | I^{(1)}, \ldots, I^{(J)})$ ziehen, um einen weiteren Buchstaben des Konzepts $I^{(J+1)}$ zu erhalten. Weiterführende Behandlungen sind etwa in [Lak14, § 5] zu finden und würden den Rahmen dieser Arbeit sprengen.

2.4. Anwendung

In diesem Abschnitt möchten wir die Anwendung des vorgestellten Rahmenwerks diskutieren. Dabei gehen wir zuerst auf Anwendungsmöglichkeiten in der Mensch-Maschine-Interaktion ein, und anschließend auf bereits umgesetzte Anwendungen aus der Literatur.

2.4.1. Anwendung in der Mensch-Maschine-Interaktion

Betrachten wir einen anderen Begriff: die skizzenhafte Beschreibung von Häusern. Der generative Aufbau durch Primitive, Teile und Relationen, die iterativ aneinander gefügt

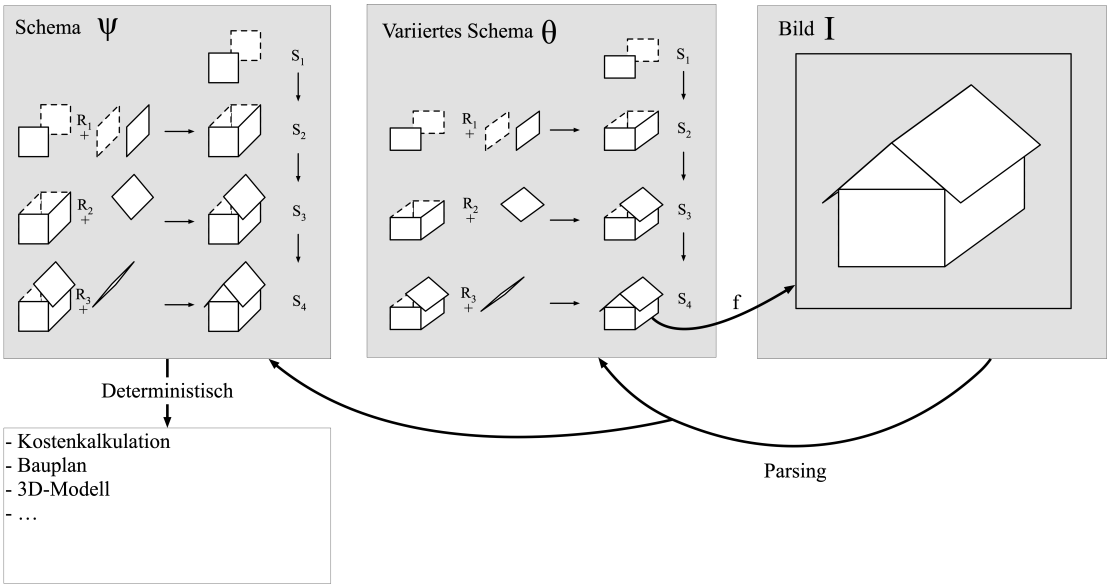

Abbildung 2.3.: Dargestellt wird das Konzept von einem Haus auf den drei Ebenen. Das Schema ψ besteht aus Primitiven, Teilen und ihren Relationen. Die Primitive sind Wandflächen. Teile bestehen aus parallelen Wandflächen. Diese stehen in Relationen zueinander, die durch einen Winkel und die entsprechende Kante, die zusammengeklebt wird, beschrieben wird.
Das variierte Schema $\tilde{\theta}$ setzt erlaubte Variabilität der skizzenhaften Beschreibung des Hauses um. Hier ist die Vorderwand nicht mehr quadratisch, sondern etwas niedriger. Schließlich geht das variierte Schema durch f deterministisch in den Bildraum.
Aus dem Schema können deterministisch Eigenschaften, wie ein Bauplan oder 3D-Modell bestimmt werden. Durch das Parsing kann so von der Skizze I auf Eigenschaften des dargestellten Objekts inferiert werden.

werden, ist schematisch in Abbildung 2.3 dargestellt.
Das Bild I rechts in der Abbildung stellt eine Skizze von einem Haus dar. Menschen können sehr gut von dieser skizzenhaften Beschreibung auf Eigenschaften des Hauses schließen. Beispiele für Eigenschaften sind eine Kostenkalkulation für den Bau, ein Bauplan oder ein 3D Modell.

Die Logik vom Rahmenwerk *Maschinelles Lernen mit Bayes'schen Programmen* erlaubt mit dem Parsing eine Zerlegung der Bilder I in die Teile S und Relationen R, wobei wir zwischen Variabilität und Schlüsselcharakteristika unterscheiden können. Aus dem Schema ψ können mit der deterministischen Funktion aus dem Schema verschiedene Kennwerte berechnet werden. Insbesondere kann aus der *skizzenhaften Beschreibung* auf präzise Eigenschaften geschlossen werden.
Dies hat klare Anwendungen in der Mensch-Maschine-Interaktion. Vorstellbar ist zum Beispiel eine Bildersuche im Internet aufgrund einer Skizze. Anstatt wie heute üblich mit Schlagworten oder ähnlichen Bildern das Objekt zu suchen, könnte die Suche durch die beschriebene Technik auf Basis der gezeichneten Skizze stattfinden. Darüber hinaus

sind Anwendungen in Modedesign, Architektur und Computerlinguistik denkbar. Zum Beispiel könnte aus der Skizze eines Kleidungsstücks das Schnittmuster bestimmt werden. Insbesondere ist in der Anwendung interessant, dass von dem Schema deterministisch auch andere Eigenschaften oder Darstellungen berechnet werden können. Im Omniglot-Beispiel wäre etwa eine kalligrafische Übersetzung des Buchstabens denkbar. Der durch ein Bild gegebene Buchstabe kann in einen anderen Schreibstil übersetzt werden. Bemerkenswerterweise würde diese Veränderung für unbekannte Buchstaben funktionieren. Zwar können heute durch Depp Learning bereits Malstile von Bildern kopiert werden [LW16], doch bei Skizzen ist die Besonderheit, dass das dargestellte Objekt nicht detailreich, sondern nur schematisch dargestellt wird, und so viel auf Eigenschaften, die nicht gezeichnet, aber impliziert sind, geschlossen werden muss.

Menschen denken viel in Skizzen [Eck91] und sind Meister darin, aus der skizzenhaften Beschreibung zu abstrahieren. Maschinen können das im Allgemeinen sehr schlecht. Die hierarchische Darstellung im Maschinellen Lernen mit Bayes'schen Programmen bringt eine menschenähnliche Denkweise mit sich. Da zwischen Schlüsselcharakteristika und Variabilität unterschieden werden kann, werden sehr gute Resultate in den Aufgaben aus dem Bereich des Begriffslernen erzielt. Gleichzeitig ist das Modell sehr restriktiv und die Modellierung und das Parsing benötigt viel Expertenwissen, damit die Approximationen aus Abschnitt 2.3 überhaupt legitim sind.

2.4.2. Andere Anwendungen

Das vorangegangene Kapitel 2 suggeriert, dass das konstruierte Modell für verschiedene Anwendungen geeignet ist. Die Daten sollten aus einem iterativ generierenden Prozesses entstanden sein, damit die Zerlegung des Schemas ψ in die Teile und Relationen anwendbar ist. In [Reb16] beginnt Rebo, allgemeine Schritte aus dem Rahmenwerk auszuarbeiten, der Gedankengang wird jedoch nicht zu ende geführt.
Anwendungsmöglichkeiten sind etwa verschiedene visuelle Konzepte, bei denen eine skizzenhafte Beschreibung gewählt wird. So wenden Meng-Zhen et al das Rahmenwerk auf sich bewegende Strichmännchen an. Die Primitive, Teile und Relationen des Strichmännchen-Skeletts sind durch sogenannte Gelenkpunkte fest bestimmt [CTH17].
Unter anderem vom gleichen Autor Lake wird in [LLT14] das Konzept von gesprochenen Wörtern untersucht. So wird anhand der Spektogramme von ausgesprochenen Wörtern versucht, unbekannte Wörter anhand der Aussprache zu vergleichen. Konkret lernt das Modell die Parameter der Verteilungen an Spektogrammen einmal an gesprochenen Wörtern in Japanisch und einmal in Englisch. Später soll das Modell gegeben eine Testaufnahme und einige Klassenaufnahmen von einem japanischen Wort diejenige Aufnahme identifizieren, die dem ausgesprochenen Wort der Testaufnahme entspricht. Dabei wird einmal der Klassifikationsfehler auf den gesprochen Wörtern von gleichgeschlechtlichen Sprechern, und einmal mit sowohl weiblichen als auch männlichen Sprechern evaluiert. Probanden hatten bei dieser Aufgabe eine Fehlerquote von 2.6% beziehungsweise 2.9%. Das Modell, dass auf japanischen Wörtern trainierte, hatte einen Klassifikationsfehler von 7.5%, wenn das Geschlecht über die Aufnahmen nicht variiert wird, gegenüber von 21.8%, wenn die Aufnahmen von beiden Geschlechtern stammten. Das Modell, dass auf englischen Wörtern trainierte, hatte einen Klassifikationsfehler von 16.8% beziehungsweise von 34.5%. Die gewählte Repräsentation der Daten konnte folglich nicht das Level

menschlicher Intelligenz erreichen und zwischen Schlüsselcharakteristika und Variabilität unterschieden. Außerdem werden generative Aufgaben durch das Modell gelöst. Dabei ist das Ergebnis relativ schlecht und die Modellausgabe wird nur durch das Einfügen von starken Störgeräuschen in die Aufnahmen der menschlichen Probanden schwerer von dieser unterscheidbar [LLT14].
Eine abstraktere Herangehensweise wählten Overlan et al in der Publikation [OJP17]. Sie betrachteten abstrakte Konzepte, die aus zusammengesetzten geometrischen Figuren bestehen. Die Aufgabe war, neue Konzepte, die zu gegebenen Figuren passen, zu generieren. Dies erfordert eine starke Eingrenzung des Hypothesenraums anhand weniger Beispiele. Im sogenannten *program synthesis* in [ELT15] wird das Programm, das die multivariate Wahrscheinlichkeitsverteilung beschreibt, aus den Daten gelernt. Ellis untersucht in [EST16] und [Ell+18] auch Möglichkeiten, aus diesen Räumen möglicher Programme Stichproben zu ziehen. Diese Denkart verallgemeinert die hierarchischen bayes'schen Modelle aus dem vorgestellten Rahmenwerk.

Kapitel 3.

Maschinelles Lernen mit Bayes'schen Programmen auf dem Omniglot-Datensatz

In diesem Kapitel stellen wir die Anwendung von Maschinellem Lernen mit Bayes'schen Programmen auf dem Omniglot-Datensatz vor. Dafür stellen wir zuerst die Problemstellung vor, wenden das Rahmenwerk wie aus Abschnitt 2.2 auf den Datensatz an, betrachten ein modifiziertes Modell und werten das Modell aus.

3.1. Problemstellung

In diesem Abschnitt wird der Datensatz Omniglot und die kognitionswissenschaftliche Relevanz vorgestellt. Die Aufgaben, die auf dem Datensatz zu lösen sind, wurden bereits in Abschnitt 2.1 erklärt.

3.1.1. Datensatz

Der Datensatz besteht aus 50 verschiedenen Alphabeten mit insgesamt 1623 Buchstaben aus historischen, aktuellen und künstlich ausgedachten Schriften, wie Hebräisch, Latein oder dem Alphabet aus der TV-Serie Futurama.
Die Buchstaben stammen aus der Sammlung einer digitalen Enzyklopädie über verschiedene Schriftsysteme, `omniglot.com` [Age20]. Ausgehend von diesem Datensatz wurden die Buchstaben in prozedurale Form umgewandelt. Dabei sahen Probanden die gedruckten Buchstaben. In ein Feld darunter sollten die Probanden die Buchstaben per geklickten Mauszeiger auf den Bildschirm zeichnen, was in einer Liste von $x, y, time$ gespeichert wurde. Jeder Buchstabe liegt zusätzlich als binäres Bild vor und wurde mehrmals von verschiedenen Teilnehmern gezeichnet. Beispiele von Buchstaben des Datensatzes sind in Abbildung D.1 im Anhang zu finden.

3.1.2. Kognitionswissenschaftliche Relevanz

1985 schrieb D. Hofstadter „The central problem of Artificial Intelligence is the question: 'what is the letter a'?", zu deutsch ungefähr „Die zentrale Aufgabe von Künstlicher Intelligenz ist zu erkennen 'was ist ein a?'"[Hof85]. Er argumentierte, dass das Verstehen und Einsetzen derer Grundkonzepte, die Menschen für die Erkennung von Buchstaben

benutzen, die fundamentalen Meilensteine auf dem Weg zu Künstlicher Intelligenz seien [LUS16].
Unser Ansatz versucht Teile dieser Fragen zu beantworten. Was ist das Konzept *Buchstabe*, was ist ein *a*? Ein Buchstabe ist eine Zusammensetzung von - geraden oder gekrümmten - Strichen, die so eine Bedeutung haben. Das Bild eines Buchstabens ist die Darstellung des Konzeptes und innerhalb dieses Konzeptes gibt es verschiedene erlaubte Freiheitsgrade, etwa längere oder kürzere Striche.
Eng mit dem menschlichen Verständnis des Buchstaben verbunden ist der unterliegende generative Prozess. Der Buchstabe ist durch den stückweise glatten Zug eines Stiftes auf Papier entstanden, dessen Striche sich nach festem Schema kreuzen. Das Modell identifiziert später eine plausible Zeichenanweisung, und erkennt, welche Schlüsselcharakteristika den Buchstaben ausmachen, und welche Variabilität erlaubt ist. Dazu lernt es erst am Datensatz, was ein Buchstabe ist und überträgt später diese Erfahrung auf unbekannte Buchstaben. Es gibt empirische Beweise dafür, dass für das menschliche Verständnis dieser generative Prozess wichtig ist. Über zehn dieser Experimente über den generativen Prozess an Buchstaben sind in [Lak14, § 2.1] beschrieben. Etwa sollten zwei Probandengruppen Buchstaben nach einer Anleitung zeichnen, wobei zwischen den Gruppen die Richtung des letzten Striches variierte. Später sollten sie den Buchstaben nochmal zeichnen. Fast alle malten den letzten Strich dabei in der Richtung der vorher gelernten Anleitung [BF88].
Wie in Abschnitt 1.4 beschrieben, funktioniert das menschliche Denken sehr kompositional. Konzepte sind hierarchisch angeordnet und komplizierte Konzepte sind auf Basis von einfacheren definiert. Wir übernehmen diesen Gedanken, in dem wir den Buchstaben aus den Strichen zusammensetzen, die untereinander in Relation stehen.

3.2. Anwendung des Rahmenwerks

Im Folgenden wenden wir das Rahmenwerk aus Abschnitt 2.2 an. Auch wenn einige Abschnitte zusammengelegt werden könnten, werden wir uns an die Iterationsreihenfolge aus dem Algorithmus'schema 3 halten, um diese Grundschritte identifizierbar zu lassen.

3.2.1. Spezifikation der Buchstaben auf drei Ebenen

Wir möchten das Konzept eines Buchstabens als hierarchisches bayes'sches Modell sehen

$$\psi \to \tilde{\theta} \to I.$$

Hierbei beschreibt wie in Abschnitt 2.2.1 ψ das Schema, $\tilde{\theta}$ das variierte Schema und I das Bild des Buchstabens.
Ein Buchstabenschema ψ besteht aus κ Strichen und ihren Relationen zueinander. Ein Strich steht für die gemalte Linie von Stift-ansetzen bis Stift-absetzen und wird durch n konkatenierte Teilstriche modelliert. Die Teilstriche sind kubische Splines und werden durch Kontrollpunkte, Größe und Index im Verzeichnis beschrieben. Die möglichen Relationen des i-ten Strichs zu den vorherigen Strichen sind $start, ende, entlang$ und $frei$. Diese Modellierung ist in der Abbildung 2.2 im Abschnitt 2.2.1 dargestellt.

Sprechen wir diese Modellierung einmal formal durch:

Ein Buchstabenschema $\psi \in \Psi$ besteht aus einer Menge von κ Strichen $S = \{S_1, \ldots, S_\kappa\}$. Jeder Strich ist eine Menge von n_i Teilstrichen $S_i = \{s_{i1}, s_{i2}, \ldots, s_{in_i}\} \in \Omega_S(\kappa, n_1)$, die jeweils durch einen kubischen Spline modelliert sind. Der kubische Spline ist durch fünf Kontrollpunkte x_{ij}, einer relativen Größe y_{ij} sowie einem Index z_{ij} spezifiziert. Dieser verweist in das Verzeichnis aller möglichen Grundformen und wird später für die Bewertung benötigt. Die Relationen zwischen den Strichen sind in der Menge $R = \{R_1, \ldots, R_\kappa\} \in \Omega_R(\kappa, S)$ festgelegt. R_i beschreibt dabei die Relation des i-ten Striches zu den vorangegangenen Strichen $S^{(i-1)} = \{S_1, \ldots, S_{i-1}\}$. Die vier möglichen Relationen sind dabei *frei*, *start*, *ende* und *entlang* und werden durch spezifische Parameter beschrieben. Bei *start* beziehungsweise *ende* ist der Strich am Anfangs- oder Endpunkt des u_i-ten Strichs angehangen. Bei der Relation *entlang* spezifizieren wir, dass der Strich an Splinekoordinate τ_i des v_i-ten Teilstrichs des u_i-ten Strichs angehangen wird. Die Relation *frei* spezifiziert einen Startpunkt L_i in Abhängigkeit der Nummer des Strichs J_i. Damit gilt:

$$\Psi = \Big\{\psi = (\kappa, S, R) | \kappa \in \mathbb{N}, S \in \bigtimes_{i=1}^{\kappa} \Omega_S(\kappa, n_i), R \in \bigtimes_{i=1}^{\kappa} \Omega_R(\kappa, S^{(i-1)})\Big\}$$

$$\Omega_S(\kappa, n) = \Big\{S = \{s_1, s_2, \ldots, s_n\} | n \in \mathbb{N}^+; \quad \forall i \in \{1, 2, \ldots n\} : s_i \in \Omega_s\Big\}$$

$$\Omega_s = \Big\{s = \{x, y, z\} | x \in \mathbb{R}^{10}; \quad y \in \mathbb{R}; z \in \mathbb{N}\Big\}$$

$$\begin{aligned}\Omega_R(\kappa, S) \in \Big\{ & \{R = \{\zeta = frei, J_i, L_i\} | J_i \in \mathbb{N}; \quad L_i \in \mathbb{R}^2\}, \\ & \{R = \{\zeta = start, u_i\} | u_i \in \{1, \ldots, i-1\}\}, \\ & \{R = \{\zeta = ende, u_i\} | u_i \in \{1, \ldots, i-1\}\}, \\ & \{R = \{\zeta = entlang, u_i, v_i, \tau_i\} | u_i \in \{1, \ldots, i-1\}; v_i \in \{1, \ldots, n_{u_i}\}; \tau_i \in \mathbb{R}\}\Big\}.\end{aligned}$$

Das variierte Schema $\tilde{\theta}$ enthält variierte Versionen der einzelnen Parameter und die Bildtransformation $\tilde{B}$. Dabei bezeichnet die Tilde ˜ jeweils die Zugehörigkeit zu $\tilde{\theta}$ und betont gegebenenfalls den Bezug zum entsprechenden Parameter auf der Ebene des Schemas ψ.

Diese Parameter in $\tilde{\theta}$ sind die variierten Striche $\tilde{S} = \{\tilde{S}_1, \ldots, \tilde{S}_\kappa\}$ mit $\tilde{S}_i \in \Omega_S(\kappa, n)$ und Relationen $\tilde{R} = \{\tilde{R}_1, \ldots, \tilde{R}_\kappa\}$ mit $\tilde{R}_i \in \Omega_R(\kappa, \tilde{S})$. Zusätzlich dazu wird die Bildtransformation $\tilde{B} \in \Omega_{\tilde{B}}$ beschrieben. Dazu gehört die Startposition des ersten Striches $\tilde{L}$ und eine globale Transformation $\tilde{A}$ innerhalb des Bildes. Die vierdimensionale Zufallsvariable kodiert in den ersten beiden Koordinaten eine Skalierung bezüglich der Achsen und in den letzten beiden die Translation innerhalb des Bildes.
Außerdem werden verschiedene Störungen des Bildes kodiert, einmal mögliches Pixelrauschen $\tilde{\epsilon}$, welches Pixel des binären Bildes kippt. Der Filterparameter $\tilde{\sigma_b}$ führt zu einer gewollten Unschärfe im Bild. Insgesamt gilt

$$\Omega_{\tilde{B}} = \Big\{\tilde{B} = (\tilde{A}, \tilde{L}, \tilde{\sigma_b}, \tilde{\epsilon}) |\ \tilde{A} \in \mathbb{R}^4,\ \tilde{L} \in \mathbb{R}^2,\ \tilde{\sigma_b} \in \mathbb{R},\ \tilde{\epsilon} \in [0, 1]\Big\}.$$

Beim Zeichnen des binären Bildes $I \in \mathbb{Z}_2^{105\times105}$ wird das variierte Schema $\tilde{\theta}$ iterativ abgearbeitet. Die Striche werden der Reihe nach gezeichnet, wobei dem bisherigen Bild jeweils iterativ der nächste Teilstrich angehangen wird. Dabei wird die Trajektorie des Teilstrichs deterministisch aus den Kontrollpunkten $\tilde{x}_{ij}$, relativer Größe $\tilde{y}_{ij}$ und Startposition $\tilde{L}_{ij}$ bestimmt. Die Startposition $\tilde{L}_{ij}$ ist entweder per $\tilde{L}$ für den ersten Strich gegeben, oder wird aus der Relation und dem bisher gezeichneten Bild berechnet.
Dann wird die Trajektorie $\tilde{T}_{ij} = h(\tilde{L}_{ij}, \tilde{x}_{ij}, \tilde{y}_{ij})$ aus den skalierten Kontrollpunkten des kubischen Splines $\tilde{y}_{ij} \cdot \tilde{x}_{ij} \in \mathbb{R}^{10}$ berechnet, auf die richtige Startkoordinate $\tilde{L}_{ij}$ verschoben und durch Faltung mit dem Filter $b[\frac{a}{12}, \frac{a}{6}, \frac{a}{12}, 1-a, \frac{a}{6}, \frac{a}{12}, \frac{a}{6}, \frac{a}{12}]$ für $a = 0.5$, $b = 6$ und Diskretisierung der Werte verdickt. Dies führt zu einem binären Bild I der Größe 105×105.

Insgesamt haben wir die folgende Notation für die Beschreibung des Buchstabens auf den drei Ebenen $\psi, \tilde{\theta}, I$:

$\psi = \{\kappa, S, R\}$	Buchstabenschema
$\kappa \in \mathbb{N}^+$	Anzahl der Striche
$S = \{S_1, \ldots, S_\kappa\}$	Menge der Striche
$R = \{R_1, \ldots, R_\kappa\}$	Menge der Relationen
$S^{(i)} = \{S_1, \ldots, S_i\}$	Menge der ersten i Striche
$S_i = \{s_{i1}, s_{i2}, \ldots, s_{in_i}\} \in \Omega_S(\kappa, n_i)$	Menge der Teilstriche des i-ten Strichs
$n_i \in \mathbb{N}^+$	Anzahl der Teilstriche des i-ten Strichs
$s_{ij} = \{x_{ij}, y_{ij}, z_{ij}\} \in \Omega_s$	Beschreibung des j-ten Teilstrichs des i-ten Strichs
$x_{ij} \in \mathbb{R}^{10} = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$	Koordinaten der 5 Kontrollpunkte des Teilstrichs
$y_{ij} \in \mathbb{R}^+$	relative Größe des Teilstrichs
$z_{ij} \in \mathbb{N}$	Index des Teilstrichs im Verzeichnis der Grundformen
$R_i \in \Omega_R(\kappa, S^{(i-1)})$	Relation des i-ten Strichs zu $S^{(i-1)}$
$J_i \in \mathbb{N}$	Startpositionsparameter für die Relation $frei$
$L_i \in \mathbb{R}^2$	Startkoordinate des Strichs für die Relation $frei$
$u_i \in \{1, 2, \ldots, i-1\}$	Index des Strichs, auf das sich die Relation bezieht
$v_i \in \{1, 2, \ldots, n_{u_i}\}$	Index des Teilstrichs auf dem u_i-tem Strich
$\tau_i \in \mathbb{R}$	relative Koordinate auf dem Teilstrich
$\tilde{\theta} = \{\tilde{S}, \tilde{R}, \tilde{B}\}$	variiertes Schema
$\tilde{B} = (\tilde{A}, \tilde{L}, \tilde{\sigma_b}, \tilde{\epsilon}) \in \Omega_{\tilde{B}}$	Bildtransformationsparameter
$\tilde{L} \in \mathbb{R}^2$	Startposition des ersten Strichs
$\tilde{A} \in \mathbb{R}^4$	Bildtransformation
$\tilde{\sigma_b} \in \mathbb{R}$	Filterparameter
$\tilde{\epsilon} \in [0, 1]$	Störung der Pixel
$T_{ij} \subset \mathbb{Z}_2^{105\times105}$	Trajektorie des j-ten Teilstrichs des i-ten Strichs
$I \in \mathbb{Z}_2^{105\times105}$	binäres Bild.

Bemerkung 48. Der fertige Buchstabe ist durch seine Striche und Relationen beschrieben. In einer gewissen Weise sehen wir den Buchstaben als Graphen. Zum einen ist eine gewisse Graphstruktur durch die Teile und Relationen gegeben, wobei die Knotenmenge durch die Anfangs- und Endpunkte der Striche gebildet wird. Zum anderen ist jeder Strich selbst durch die Konkatenierung der Teilstriche entstanden. Zusätzlich dazu haben wir noch Informationen über die zweidimensionalen Koordinaten der Zeichnung des Graphens und den generativen Prozess, heißt einer Reihenfolge der Kanten.
Eine Alternative, den fertigen Buchstaben zu speichern wäre eine $\mathbb{Z}_2^{105\times 105}$ Matrix. In der gewählten Beschreibung ist jedoch implizit die Zeichenanweisung in zwei Weisen enthalten, einmal als abstraktes Schema und einmal mit konkreten Koordinatenwerten.

3.2.2. Spezifikation der multivariaten Verteilungen eines Buchstabens

Nun modellieren wir die drei Ebenen $\psi, \tilde{\theta}, I$ probabilistisch. Dazu legen wir die multivariaten Wahrscheinlichkeitsverteilungen von $\psi = \{\kappa, S, R\}$ und $\tilde{\theta}\,|\psi$ mit $\tilde{\theta} = \{\tilde{S}, \tilde{R}, \tilde{B}\}$ fest.
Es gilt zunächst für ψ wie in Gleichung 2.1 im Abschnitt 2.2.2

$$\rho(\psi) = \rho(\kappa) \prod_{i=1}^{\kappa} \rho(S_i)\rho(R_i|S_1, \ldots, S_{i-1}).$$

Dabei setzen wir für die multivariate Verteilung eines Strichs $S_i = \{s_{i1}, \ldots s_{in_i}\}$:

$$\rho(S_i) = \rho(z_i) \prod_{j=1}^{n_i} \rho(x_{ij}|z_{ij})\rho(y_{ij}|z_{ij}) \quad \text{mit} \tag{3.1}$$

$$\rho(z_i) = \rho(z_{i1}) \prod_{j=2}^{n_i} \rho(z_{ij}|z_{i(j-1)}). \tag{3.2}$$

Die bedingten Verteilungen $\rho(x_{ij}|z_{ij})$ und $\rho(x_{ij}|z_{ij})$ sind definiert durch

$$\rho(x_{ij}|z_{ij}) \sim \mathcal{N}(\mu_{z_{ij}}, \Sigma_{z_{ij}}) \tag{3.3}$$

$$\rho(y_{ij}|z_{ij}) \sim \mathcal{G}(\alpha_{z_{ij}}, \beta_{z_{ij}}), \tag{3.4}$$

wobei $\mu_{z_{ij}}, \Sigma_{z_{ij}}, \alpha_{z_{ij}}$ und $\beta_{z_{ij}}$ Parameter der Verteilungen sind. Die weiteren Verteilungen in $\rho(\psi)$ werden aufgrund der empirischen Häufigkeiten festgelegt.

Bemerkung 49. Die definierten Wahrscheinlichkeitsverteilungen spiegeln das Expertenwissen über die gewünschten Zeichenanweisungen wieder. Die Gleichung 3.2 stellt eine Markovkette erster Ordnung dar [Kle13, § 17] Dadurch können Winkel und sich wiederholende Strukturen kodiert werden. Die Verteilung eines Striches in Gleichung 3.1 lässt sich durch die Teilstriche, die durch Kontrollpunkte x_{ij}, relativer Größe y_{ij} und den Index z_{ij} gegeben sind definieren. Da der Index bereits einen kubischen Spline beschreibt, erklären sich die bedingten Verteilungen $\rho(x_{ij}|z_{ij})$ und $\rho(y_{ij}|z_{ij})$ in Gleichungen 3.3 und 3.4, die nur eine geringe Abweichung zulassen.

Wir legen fest, dass die folgende multivariate Wahrscheinlichkeitsverteilung für $\tilde{\theta} = \{\tilde{S}, \tilde{R}, \tilde{B}\}$ mit $\tilde{B} = (\tilde{A}, \tilde{L}, \tilde{\sigma_b}, \tilde{\epsilon})$ gilt:

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{L}|\tilde{S}, \tilde{R}, \tilde{A}, \tilde{\sigma_b}, \tilde{\epsilon}, \psi) \cdot \prod_{i=1}^{\kappa} \rho(\tilde{R}_i|R_i)\rho(\tilde{x}_i|x_i)\rho(\tilde{y}_i|y_i)\rho(\tilde{A}, \tilde{\sigma_b}, \tilde{\epsilon}).$$

Dabei sind viele der Variablen schlichtweg variierte Varianten der zugehörigen Variable auf der Ebene des Schemas. So gilt

$$\begin{aligned}\rho(\tilde{x}_{ij}|x_{ij}) &\sim \mathcal{N}(\mu_{x_{ij}}, \sigma_x^2)\\ \rho(\tilde{y}_{ij}|y_{ij}) &\sim \max(0, \mathcal{N}(y_{ij}, \sigma_y^2))\\ \rho(\tilde{\tau}_i|\tau_i) &\sim \min(0, \max(\mathcal{N}(\tau_i, \sigma_\tau^2), 1)).\end{aligned}$$

Für die Relationen außer $\zeta = \textit{entlang}$ gelte $\rho(\tilde{R}_i|R_i) = \delta(\tilde{R}_i - R_i)$. Bei der Relation *entlang* wird die relative Befestigungskoordinate τ_i variiert. Wieder sind $\mu_{x_{ij}}, \sigma_x^2, \sigma_y, \sigma_\tau$ Parameter der Verteilungen.
Für die Startposition $\tilde{L}_i$ gilt

$$\rho(\tilde{L}_i|\tilde{R}_i, T_1, \ldots, \tilde{T}_{i-1}) \sim \mathcal{N}(g(\tilde{R}_i, \tilde{T}_1, \ldots, \tilde{T}_{i-1}), \Sigma_L)$$

für die Funktion g, die die Koordinaten des nächsten Startpunktes gegeben bisherige Trajektorien $\tilde{T}_1, \ldots, \tilde{T}_{i-1}$ und entsprechende Relation $\tilde{R}_i$ berechnet. Bei der Relation $frei$ wird entsprechend das $\tilde{L}_i$ aus der Spezifikation der Relation gewählt. Bei *ende* und *start* wählen wir als Startpunkt die jeweilige Koordinate des End- oder Anfangspunktes wie in der Relation angegeben und bei $\zeta_i = \textit{entlang}$ wird der Startpunkt auf dem entsprechenden Teilstrich gemäß $\tilde{\tau}$ bestimmt.

Für die Störung des Bildes werden die entsprechenden Verteiler wie folgt festgelegt, wobei $\Sigma_{\tilde{A}}$ wieder ein Parameter ist:

$$\begin{aligned}\rho(\tilde{A}) &\sim \mathcal{N}([1, 1, 0, 0], \Sigma_{\tilde{A}})\\ \rho(\tilde{\sigma_b}) &\sim \mathcal{U}(0.5, 16)\\ \rho(\tilde{\epsilon}) &\sim \mathcal{U}(0.0001, 0.5).\end{aligned}$$

Bemerkung 50. Die definierten Wahrscheinlichkeitsverteilungen passen zu der Intuition, dass $\tilde{\theta}$ eine Variation von dem Schema ψ darstellt. Die Normalverteilung wird oft verwendet, um Störungen von Größen zu modellieren [Mes19].

Im Anhang C sind die Algorithmen zu diesen Verteilungen unter Alogrithmus 6, 7, 8 und 9 zu finden.

3.2.3. Lernen der Modellparameter vom Buchstabenmodell

Die Wahrscheinlichkeitsverteilungen der einzelnen Parameter werden jeweils anhand des Datensatzes gelernt. Dazu werden in einer Experimentreihe 30 Alphabete und in einer anderen 5 Alphabete verwendet.
Da der Datensatz aus Listen $x, y, time$ der gezeichneten Bilder besteht, kann der Buchstabe leicht in Striche zerlegt werden. Ein Strich ist die gemalte Trajektorie zwischen dem Ansetzen und Absetzen des Stiftes. Die identifizierten Striche werden auf $50ms$ lange Zeitschritte normiert. Die fehlenden Werte werden durch lineare Interpolation approximiert. Innerhalb einer Zeichnung eines Strichs werden Pausen markiert, wenn der

Stift weniger als einen Pixel innerhalb eines Zeitschritts bewegt wird. Dadurch können die Striche S_i und Teilstriche s_{ij} direkt aus den Daten abgelesen werden. Die Teilstriche werden normiert. Die normierten Teilstriche haben im zweidimensionalen Koordinatenraum Durchschnitt 0 und sind standardisiert, so dass sie entlang der Gerade größter Ausbreitung die selbe Größe haben und zwischen zwei Punkten genau einen Pixel Abstand ist. Kurze Teilstriche mit einer Länge unter 10 Pixel werden entfernt. Auf diese Weise werden insgesamt 55.000 Teilstriche aus den Daten extrahiert, die jeweils durch einen kubischen Spline approximiert werden. Jeder Spline ist dann durch 5 Kontrollpunkte $\tilde{x}_i \in \mathbb{R}^{(10)}$ und die relative Größe dieses Splines im Bildraum $\tilde{y}_i$ dargestellt, wobei diese doppelt gewichtet wird. Diese 55.000 Punkte im $\mathbb{R}^{12}$ werden durch eine Gauß'sche Mischverteilung in 1250 Cluster eingeteilt. Jeder dieser Cluster bildet jeweils einen Eintrag im Verzeichnis aller möglichen Primitive. Ein Index z wird jedem Cluster zugeteilt.
Für jeden Cluster werden die Variabilität $\mu_z, \Sigma_z, \alpha_z$ und β_z durch einen Maximum Likelihood Estimator geschätzt. Aus den geglätteten empirischen Werten werden die Werte für $\rho(z_{ij}|z_{i(j-1)})$ geschätzt.

Die Verteilungen $\rho(\kappa)$ der Anzahl der Striche κ und $\rho(n_i|\kappa)$ werden aus den empirischen Daten abgelesen. Die empirischen Häufigkeiten sind in Abbildung D.2 im Anhang D in Balkendiagrammen dargestellt. Wie aus der Abbildung leicht zu entnehmen ist, kommen besonders oft Buchstaben mit ein oder zwei Strichen vor. Die Häufigkeit einer bestimmten Anzahl κ von Strichen im Buchstabe sinkt mit steigendem κ. Deutlich wird außerdem, dass Buchstaben mit vielen Strichen wenige Teilstriche haben, während Buchstaben mit wenigen Strichen oft mehr Teilstriche besitzen.
In derselben Darstellung ist die empirische Verteilung der Startposition $\tilde{L}$ innerhalb des Bildrahmens dargestellt. Dabei werden die Startkoordinaten des ersten Strichs, zweiten Strichs und der weiteren Striche gezählt. Bei der Zeichnung der meisten Buchstaben wird zuerst oben links in der Ecke angesetzt.
Die Relationen werden aufwendig ausgerechnet [LST13] und die empirische Häufigkeit der Relationen ermittelt. Demnach ist $\rho(\zeta_i = entlang) = 50\%$, $\rho(\zeta_i = start) = 5\%$, $\rho(\zeta_i = ende) = 11\%$ und $\rho(\zeta_i = frei) = 34\%$.
Die erlaubte Variabilität wird zwischen verschiedenen Bildern desselben Buchstabens verglichen. Dabei sollten Probanden die Buchstaben nach einer festgelegten Zeichenanweisung zeichnen. Aus diesen Bildern werden mit Maximum Likelihood Estimation die Parameter $\sigma_x, \sigma_y, \sigma_\tau$ für die Variabilität zwischen den Zeichnungen desselben Buchstaben geschätzt.
Auch die Parameter für die globale Transformation innerhalb des Bildrahmens $\rho(\tilde{A})$ werden berechnet. Dazu wird der Schwerpunkt und Anzahl der schwarzen Pixel gezählt. Pro Buchstabe wird die Variabilität verglichen, wobei vorher die einzelnen Buchstabengruppen auf gemeinsamen Schwerpunkt und relative Größe transformiert werden. Dadurch kann σ_A aus den empirischen Verteilungen innerhalb der Gruppe eines Buchstabens für alle Buchstaben bestimmt werden.

Bemerkung 51. Später in Abschnitt 3.3.4 werden wir das Modell zu einer modifizierten Version ohne Erfahrungsübertrag vergleichen. Damit beschreiben wir eine Modellarchitektur, die über dieselbe multivariate Verteilung über Teile, Relationen und Primitive aufgebaut ist, wobei die Verteilungen angepasst und nicht mehr aus den Daten gelernt werden. Durch unsere aufwendige Konstruktion der Modelle ist die Ausgabe dieses mo-

difizierten Modells für die Klassifikationsaufgaben und die generativen Aufgabe zwar schlechter, aber dennoch passabel. Das zeigt, dass das Lernen der Parameter essentiell für die Leistung auf der Höhe menschlicher Intelligenz ist, aber die komplexe Modellstruktur bereits einen großen Teil beisteuert.

3.2.4. Parsing eines Buchstabens

Es ist notwendig, aus einem Bild I geeignete Parameter $\psi, \tilde{\theta}$ zu schätzen. Nur damit kann später Q aus Definition 41 berechnet werden, und verschiedene Bilder verglichen werden. Wir müssen demnach eine Zeichenanweisung aus dem Bild schätzen, die beschreibt, wie der Buchstabe gemalt wurde. Dies tun wir durch eine stochastische Irrfahrt auf dem Buchstabengraphen. Als Buchstabengraphen bezeichnen wir den Graphen, den wir aus dem Bild schätzen, wobei wir die Linien des Buchstabens als Kanten und die Schnittpunkte dieser als Knoten sehen.
Der Prozess, aus dem Bild einen Graphen zu schätzen, ist in Abbildung 3.1 dargestellt. Zuerst werden die Linien des Originalbildes (a) verdünnt, was in einem verdünnten Bild (b) resultiert [S L92]. Um die Knoten des Graphens zu schätzen, werden kritische Punkte identifiziert, die nicht genau zwei gefärbte Nachbarpixel haben. Wie bei (c) dargestellt, kann es aber beim Verdünnungsprozess zu zu vielen identifizierten Knoten kommen. Diese kritischen Punkte werden durch ein Überschneidungskriterium zusammengelegt. Dabei werden Kreise um die kritischen Punkte gelegt, so dass die Fläche des Kreises innerhalb des Strichs aus dem Originalbild ist. Sich überschneidende Kreise werden zusammengelegt [LS90].
Schließlich wird auf dem Graphen eine stochastische Irrfahrt ausgeführt, bis alle Kanten

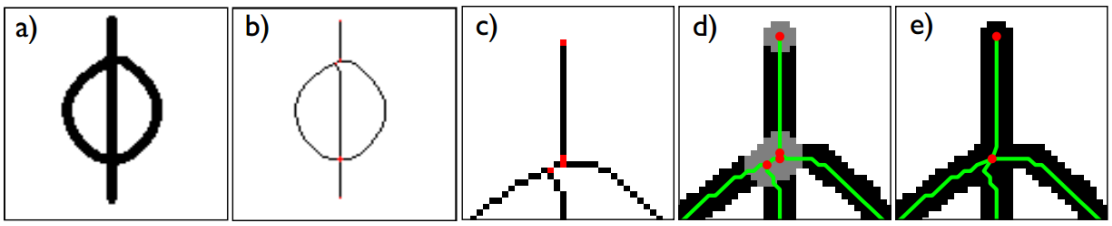

Abbildung 3.1.: Dargestellt ist das Originalbild (a), der verdünnte Buchstabe (b), die identifizierten kritischen Punkte in rot bei (c), die Kreise des Überschneidungskriteriums bei (d) und der fertig extrahierte Graph bei (e). Abbildung entnommen aus [LST15].

mindestens einmal besucht wurden. Der Stochastische Irrläufer läuft dabei von Knoten zu Knoten, und kann zwischendurch absetzen. Am Anfang und nach dem Absetzen wird der angesteuerte Knoten proportional zu $\frac{1}{b^\gamma}$ gewählt, wobei b die Anzahl der bisher unbesuchten Kanten ab diesem Knoten bezeichnet und γ eine Realisierung einer Zufallsvariable pro Irrfahrt ist und das Maß der Entropie kodiert.
Für einen Knoten gibt es die verschiedenen Möglichkeiten, weiter zu traversieren oder den Stift abzusetzen. Die Wahrscheinlichkeit einer Aktion A ist dabei proportional zu

$$\rho(A) \propto \exp(-\lambda \sigma_A).$$

Dabei kodiert σ_A eine Bewertung der Aktion. Ist die Kante unbesucht, ist dies schlicht der lokale Winkel. Eine Strafe von 90° wird zu dem Winkel addiert, falls die Kante bereits abgelaufen wurde. Das Absetzen des Stiftes wird mit 45° kodiert. Der Entscheidungsprozess ist in Abbildung 3.2 dargestellt.

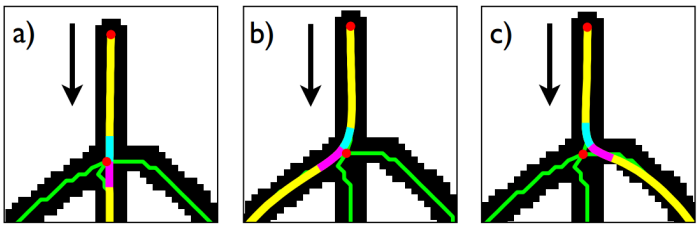

Abbildung 3.2.: Dargestellt wird ein Entscheidungsprozess bei der stochastischen Irrfahrt. Von oben kommend, gibt es an dem Knoten drei Möglichkeiten weiter nach unten zu traversieren. Die Wahrscheinlichkeit einer Aktion hängt dabei vom lokalen Winkel ab, der bei (a) 0° ist, bei (b) 28° und (c) 47° ist. Bevorzugt wird ein kleiner Winkel. Weitere Möglichkeiten nach Traversierung des oberen Strichs wären das erneute Abfahren des senkrechten Teilstrichs oben sowie das Absetzen des Stiftes an dem Knoten. Abbildung entnommen aus [LST15].

Auf diese Art und Weise werden verschiedene Traversen des Graphens erzeugt, bis mindestens 150 Durchläufe oder 100 paarweise verschiedene Striche erzeugt worden sind. Nun müssen die gefundenen Striche noch in Teilstriche zerlegt werden, bevor wir die Modellbewertung aus Kapitel 2.3.1 anwenden können. Dazu werden greedy (gierig) [Cor+09, § 16] der Zeichenanweisung des Strichs Pausen hinzugefügt, weggenommen und verschoben.

Lemma 52. *Angenommen es gilt die Approximation* $\rho(\tilde{y}_{ij}|y_{ij}) \approx \delta(\tilde{y}_{ij} - y_{ij})$. *Für die Bewertung von* $\rho(\tilde{x}_i, \tilde{y}_i, z_i)$ *schätzen wir ab:*

$$\rho(\tilde{x}_i, \tilde{y}_i, z_i) = \rho(z_i) \prod_{j=1}^{n_i} \rho(\tilde{y}_{ij}|y_{ij})\rho(y_{ij}|z_{ij}) \int \rho(\tilde{x}_{ij}|x_{ij})\rho(x_{ij}|z_{ij})\ dx_{ij}.$$

Beweis. Wir haben nach Satz 18

$$\rho(\tilde{x}_i, \tilde{y}_i, z_i) = \rho(z_i) \cdot \rho(\tilde{y}_i|z_i) \cdot \rho(\tilde{x}_i|\tilde{y}_i, z_i).$$

Dabei gilt nach Unabhängigkeit der y_{ij} für feste i

$$\rho(\tilde{y}_i|z_i) = \rho(\tilde{y}_{i1}, \ldots, \tilde{y}_{in_i}|z_{i1}, \ldots, z_{in_i}) = \prod_{j=1}^{n_i} \rho(\tilde{y}_{ij}|z_{ij}) \approx \prod_{j=1}^{n_i} \rho(\tilde{y}_{ij}|y_{ij})\rho(y_{ij}|z_{ij}).$$

Die letzte Approximation folgt dabei aus und damit

$$\rho(\tilde{y}_{ij}|z_{ij}) = \int \rho(\tilde{y}_{ij}|y_{ij})\rho(y_{ij}|z_{ij})\ dy_{ij} \approx \rho(\tilde{y}_{ij}|y_{ij})\rho(y_{ij}|z_{ij}).$$

Wegen Satz 16 gilt

$$\rho(\tilde{x}_i|\tilde{y}_i, z_i) = \rho(\tilde{x}_i|z_i) = \int \rho(\tilde{x}_{ij}|x_{ij})\rho(x_{ij}|z_{ij})\ dx_{ij}.$$

Insgesamt folgt daher

$$\rho(\tilde{x}_i, \tilde{y}_i, z_i) = \rho(z_i) \prod_{j=1}^{n_i} \rho(\tilde{y}_{ij}|y_{ij})\rho(y_{ij}|z_{ij}) \int \rho(\tilde{x}_{ij}|x_{ij})\rho(x_{ij}|z_{ij})\ dx_{ij}.$$

□

Mithilfe dieser Bewertung können wir die Striche in Teilstriche zerlegen, diese entsprechend des Verzeichnis' klassifizieren und dann bewerten. Dabei darf das berechnete Bild aus dem gefundenen $\tilde{\theta}$ von I pro Teilstrich nicht mehr als 3 Pixel abweichen. Dies führt zu sehr guten $2K$ Zeichenanweisungen für unser Bild I. Diese $\tilde{\theta}$ werden noch respektive $\tilde{L}, \tilde{\tau}, \tilde{x}, \tilde{y}, \tilde{\epsilon}, \tilde{\sigma}_b$ mit Gradientenabstieg [Jos19, § 6.2] optimiert, während die diskreten Variablen konstant gehalten werden. Aus diesen werden die besten K Zeichenanweisungen $\tilde{\theta}^{[1]}, \psi^{[1]}, \ldots, \tilde{\theta}^{[K]}, \psi^{[K]}$ gewählt. Für jeden dieser K Parses bestimmen wir per Stichprobenentnahme aus $\rho(\psi|\,\tilde{\theta}^{[i]})$ insgesamt N Parses.

Bemerkung 53. Bei dem Parsing wird also der generative Prozess des Buchstabens nachvollzogen. Der stochastische Irrläufer wählt dabei Aktionen, die einer plausiblen Zeichnung entsprechen. So werden gerade Linien vor strikten Richtungsänderungen oder dem doppelten Nachfahren von Strecken bevorzugt. Da der Graph relativ klein ist, und wir im Prozess insgesamt 150 Durchläufe oder 100 paarweise verschiedene Striche generieren, können wir davon ausgehen, dass wir ungefähr alle möglichen Zeichenanweisungen gefunden haben. Die Zerlegung in Striche erfolgt greedy, folglich ist die gefundene Wahrscheinlichkeit der Zerlegung nahe des Optimums.
Nun haben wir gute Zeichenanweisung. Um den strikten Voraussetzungen an die Approximation in Satz 40 gerecht zu werden, muss die Wahrscheinlichkeit der $\rho(I, \tilde{\theta}^{[i]}, \psi^{[i]})$ und $\rho(\tilde{\theta}^{[i]}\,|\psi^{[ij]})$ möglichst groß sein. Durch die Optimierung wird dies in den meisten Fällen gewährleistet.

Wir verdeutlichen diesen Prozess einmal am Beispiel des Buchstabens ϕ. In Abbildung 3.3 sind sechs verschiedene Parses des Buchstabens ϕ samt Logarithmus der Bewertung Q dargestellt. Wir sehen, dass die Optimierung einen großen Einfluss auf die Bewertung hat und auch den Rang der verschiedenen Parses beeinflusst. Dabei zeigen die dargestellten Parses plausible Zeichenanweisungen.
In Abbildung 3.4 sind die Re-Optimierungen der Zeichenanweisungen auf der Ebene des variierten Schemas dargestellt. Durch globale Bildtransformation und Veränderung der weiteren Parameter wird die Zeichenanweisung respektive des Trainingsbilds optimiert. Diese Re-Optimierung führen wir links bei dem Vergleich zweier unbekannter Bilder von verschiedenen Buchstaben samt Berechnung von $\rho(I^{(T)}|I^{(c)})$ wie in Gleichung 2.7 durch, wobei wir die in Satz 43 bewiesene Approximation benutzen. Rechts ist die Re-Optimierung der Zeichenanweisungen für zwei gleiche Buchstaben abgebildet. Bei den Ergebnissen wie in Abschnitt 3.3 wurde $K = 5$ und $N = 10$ gewählt.

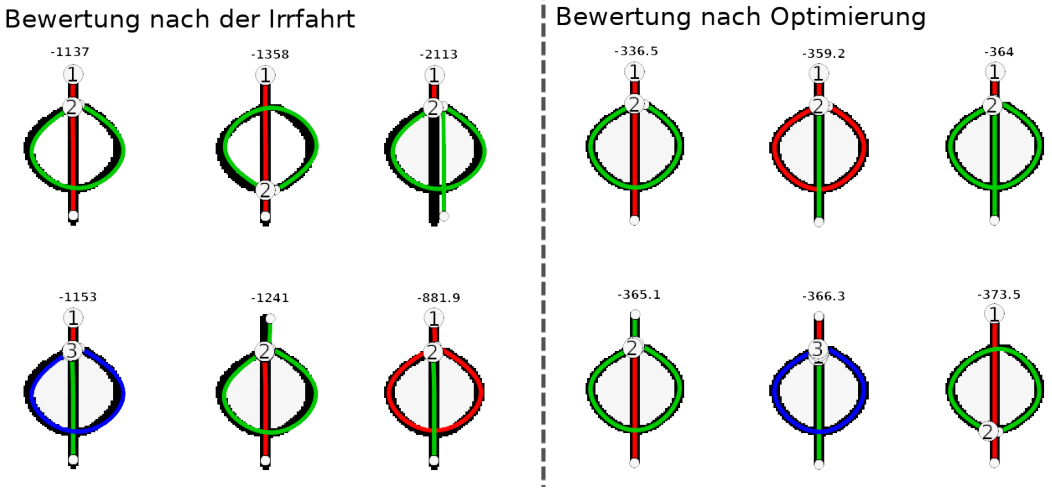

Abbildung 3.3.: Zu sehen sind sechs verschiedene Parses des Buchstabens ϕ samt der Bewertung Q aus Definition 41 vor und nach der Optimierung. Die Farben differenzieren die verschiedenen Striche, wobei der erste Strich in rot, der zweite in grün und der dritte in blau dargestellt wird. Die weißen Kreise mit Zahl i kodieren jeweils den Startpunkt des i-ten Strichs. Wie an der Zeichenanweisung oben links zu sehen ist, wird durch die Optimierung die Bewertung deutlich verbessert, dort von -1137 auf -336.5. Die Bewertung ist nicht zwischen 0 und 1, da der natürliche Logarithmus der Approximation Q genommen wird. Abbildung generiert aus dem leicht modifizierten Code aus [Lak15] mit $K = 6$ und $N = 10$.

Bemerkung 54. Der gewählte Wert von K und N beeinflusst die Approximation von Q aus Definition 41. Je größer K und N, desto mehr Schemas und variierte Schemas werden bei der Diskretisierung berücksichtigt. Gleichzeitig erhöht sich der Rechenaufwand deutlich. Informelle Experimente zeigen, dass die Erhöhung von K oder N über die in der Publikation gewählten Werte keinen deutlichen Unterschied in den Generierungsaufgaben sichtbar macht. Erniedrigen wir die Werte, so funktioniert die Re-Optimierung schlechter, da deutlich weniger Freiheitsgrade in der Zeichnung des Buchstabens erfasst werden. Im Extremfall $K = 1$ werden zwei Bilder mit unterschiedlichen besten Zeichenanleitungen nicht als zugehörig erkannt. Die Wahl der Parameter ist folglich ein Zwiespalt zwischen Rechenaufwand und Genauigkeit. Für das Finden von K Parses und der Optimierung dieser werden auf einem Computer mit Linux (Ubuntu 18.04 mit einem 5.4.0 Kernel) und einem Intel® Core™ i5-8250U CPU @ 1.60GHz x 8 Prozessor mit 16 GB RAM die Rechenzeit wie in folgender Tabelle 3.1 benötigt.

	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$
∅-Zeit	$66.00s$	$129.82s$	$203.95s$	$266.87s$	$333.54s$	$399.75s$	$492.80s$

Tabelle 3.1.: Darstellung der durchschnittlichen Rechenzeiten für die Optimierung von K Parses am Buchstaben ϕ aus Abbildung 3.3. Daten generiert durch modifizierten Code aus [Lak15], genaue Ergebnisse im Anhang in Tabelle D.1.

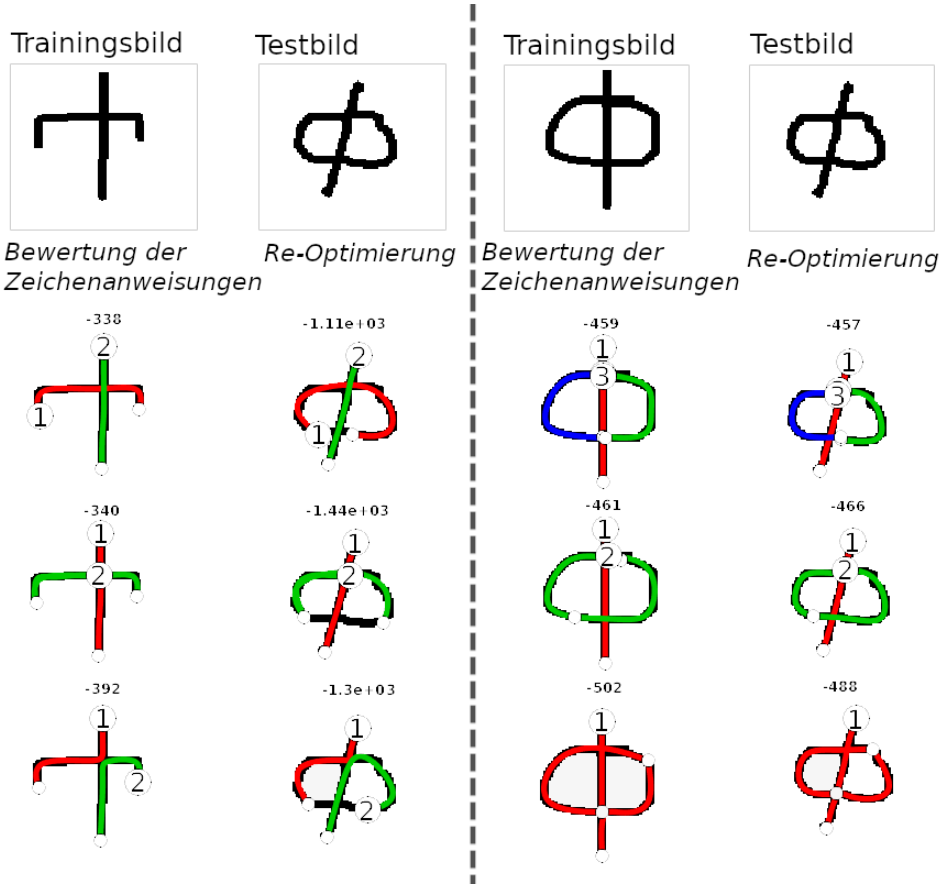

Abbildung 3.4.: Zu sehen sind die Ergebnisse von zwei Re-Optimierungen. Links wird diese auf zwei verschiedenen Buchstaben durchgeführt. Gegeben die Zeichenanweisung für das Trainingsbild, wird dieses an das Testbild optimiert. Dargestellt werden dabei jeweils drei ausgewählte Bewertungen und ihre Re-Optimierungen. Da Test- und Trainingsbild verschieden sind, ist die Bewertung dennoch schlecht. Wir sehen deutlich, dass die *Arme* des Buchstabens oben links länger werden und sich an das ϕ annähern. Rechts wird die Zeichenanweisung auf ein anderes Bild desselben Buchstabens ϕ optimiert. Dies klappt sehr gut, wie die sehr gute Bewertung des re-optimierten Bildes, -459 zu -457 oben im rechten Bereich zeigt. Abbildung generiert aus dem leicht modifizierten Code in [Lak15] mit $K = 5$ und $N = 10$.

3.2.5. Bewertung der Zeichenanweisungen

Die Bewertung der Zeichenanweisungen und das Lösen der Aufgaben wird wie in Abschnitt 2.2.5 und 2.3 beschrieben umgesetzt. Dabei wird bei der Klassifikationsaufgabe der zweiseitige Ansatz wie in Lemma 46 verwendet.

3.3. Auswertung

In diesem Kapitel werden wir die Modellarchitektur auswerten. Dazu stellen wir verschiedene Modifikationen vor, alternative Modelle, um die Aufgaben zu lösen und die Visuellen Turing-Tests, in denen die Probanden die Leistung des Modells abschätzen sollten. Anschließend vergleichen wir die Ergebnisse der Modelle.

3.3.1. Modifikationen

Um die komplexe Modellarchitektur zu prüfen, vergleichen wir das Modell mit verschiedenen Vereinfachungen. Wir stellen drei Vereinfachungen vor, bei denen die Kompositionalität beziehungsweise der Erfahrungsübertrag auf der Ebene des Schemas ψ oder variierten Schemas $\tilde{\theta}$ gemindert wird.

Bei der ersten Vereinfachung wird die Anzahl der Striche fest auf $\kappa = 1$ gesetzt, jeder Buchstabe wird also „ohne Absetzen“ gezeichnet. Wir stellen den Buchstaben durch ein *Ein-Strich-Modell* dar. Das Schema $\psi = \{n, X\}$ besteht nicht mehr aus einer Menge von Strichen und ihren Relationen, sondern nur noch aus den n Kontrollpunkten X eines kubischen B-Splines. Insgesamt ist die multivariate Verteilung zu $\psi = \{n, X\}$ durch

$$\rho(\psi) = \rho(n)\rho(X_1)\prod_{i=2}^{n}\rho(X_i|X_{i-1})$$

gegeben. Der Startpunkt X_1 hat die gleiche Verteilung wie die der Startposition $\tilde{L}$ im Ursprungsmodell. Die Verteilungen der weiteren Kontrollpunkte werden aus einer Markovkette gezogen. Um die Parameter zu lernen, werden die Buchstaben aus dem Datensatz durch zusätzliche verbindende gerade Striche in Ein-Strich-Modelle umgewandelt.
Das variierte Schema $\tilde{\theta}$ besteht aus den variierten Kontrollpunkten $\tilde{X}$ und der globalen Bildtransformation $\tilde{B}$ mit der gleichen Verteilung wie im Ursprungsmodell. Die Verteilung für $\tilde{\theta} = \{\tilde{X}, \tilde{B}\}$ ist damit

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{L}|\tilde{X}, \tilde{A}, \tilde{\sigma_b}, \tilde{\epsilon}, \psi) \cdot \prod_{i=1}^{n}\rho(\tilde{X}_i|X_i) \cdot \rho(\tilde{A}, \tilde{\sigma}_b, \tilde{\epsilon}).$$

Das Programm zu der modifizierten Verteilung von ψ und $\tilde{\theta}\,|\psi$ ist in den Algorithmen 10 und 11 im Anhang C in Pseudocode dargestellt.

Bei der zweiten und dritten Vereinfachung werden die Modellparameter nicht mehr gelernt, sondern uniform zufällig aus einem größerem Bereich gezogen, was die Wichtigkeit des Erfahrungsübertrages betont. Dabei werden auf der Ebene des Schemas ψ die folgenden Wahrscheinlichkeitsverteilungen angepasst:

$$\begin{aligned} \rho(x_{ij}|z_{ij}) \sim & \mathcal{U}(0, 105)^{10} && \text{statt} \quad \sim \mathcal{N}(\mu_{z_{ij}}, \Sigma_{z_{ij}}) \\ \rho(y_{ij}|z_{ij}) \sim & \mathcal{U}(0, 105) && \text{statt} \quad \sim \mathcal{G}(\alpha_{z_{ij}}, \beta_{z_{ij}}). \end{aligned}$$

Als einzige Relation wird $\zeta_i = frei$ erlaubt .

Auf der Ebene des variierten Schema $\tilde{\theta}$ wird die Wahrscheinlichkeitsverteilung wie folgt angepasst. Die Startposition $\tilde{L}$ werden nicht mehr aus der empirischen Verteilung gezogen, sondern aus einer stetigen Gleichverteilung auf dem Bildraum der 105×105 großen binären Bilder.
Die Parameter für die Transformationen $\Sigma_A, \sigma_\tau, \sigma_x, \sigma_y$ werden durch dreimal größere Werte ersetzt, um mehr Variabilität zu erlauben.
Im Anhang D sind unter Abbildung D.4 und D.3 Ausgaben der drei modifizierten Modelle für die Aufgabe zum Generieren weiterer Beispiele eines unbekannten Buchstabens und zum Erfinden eines Buchstabens ohne Einschränkungen dargestellt. Es wird deutlich, dass die Kompositionalität und der Erfahrungsübertrag durch die gelernten Verteilungen wichtig sind. Dies werden wir auch an den Ergebnissen in Abschnitt 3.3.4 sehen.

3.3.2. Alternative Modelle für die Aufgaben

Außerdem wird unser Modell für den Datensatz mit drei anderen Modellen, *Deep Siamese Convnet*, *Deep Convnet* und *Hierarchichal Deep* verglichen. Für das konvolutionale neuronale Netz *Deep Convnet* wird eine Netzwerkarchitektur angepassst an [Lec+98] gewählt. Das konvolutionale siamesische neuronale Netz *Deep Siamese Convnet* wird in einer ähnlichen Architektur wie in [KZS05] und [Koc15] auf den Daten gelernt. Das *Hierarchichal Deep* stellt eine Erweiterung einer sogenannten hierarchischen Boltzmannmaschine dar und ist ausführlich in [STT13] beschrieben. Einige der Paper beziehen sich auf den berühmten MNIST-Datensatz. Um die Architektur entsprechend nutzen zu können, werden die Bilder auf 28×28 Pixel skaliert. Die weitere Vorverarbeitung und vorgenommene Anpassungen sind unter [LST15, § S4.4 und S4.5] zu finden. Viele weitere Modelle, die Aufgaben auf dem Omniglot-Datensatz lösen, sind ferner unter [LST19] aufgelistet.

3.3.3. Verhaltensstudien und visuelle Turing-Tests

In einer Verhaltensstudie mit über 500 Teilnehmern wurde das trainierte Modell schließlich überprüft.
In der ersten Phase sollten verschiedene Probanden die Aufgaben wie in Abbildung 2.1 dargestellt lösen. So sollte in **A** bei der Klassifikation der passendste Buchstabe ausgewählt werden. Bei den generativen Aufgaben **C** zur Generierung eines weiteren Beispiels, **D** zum Erfinden eines neuen Konzepts passend zu gegebenen Buchstaben und **E** zum Erfinden eines Buchstabens ohne Einschränkungen musste der Buchstabe jeweils in maximal drei Sekunden geschrieben werden, um zu vermeiden, dass zu akribisch abgezeichnet wird, was nicht dem gewünschten Szenario entspricht.

In visuellen Turing-Tests wird überprüft, wie unterscheidbar die generativen Aufgaben von Modell und Menschen gelöst worden sind. Bei den Aufgaben **C**, **D**, **E** werden den Probanden zwei 2×2 beziehungsweise 3×3 Felder gezeigt. Sie sollten entscheiden, welches der Felder vom Computerprogramm und welches von Menschen gezeichnet wurde. Insgesamt sah jeder Proband 50 Felder. Alle zehn beantworteten Vergleiche wird den Versuchspersonen ihre bisherige Leistung dargestellt. In einer Modifikation von Aufgabe **C** zur dynamischen Generierung neuer Beispiele eines Buchstabens sahen die Probanden Videos der Zeichnungen von Mensch und Computer. Die Dauer der Striche wurde dabei

normalisiert.

Probanden werden durch ihr Identifizierungslevel, kurz *ID-Level*, verglichen. Die Kennzahl beschreibt, wie viel Prozent der Antworten richtig waren. Sind Mensch und Modell ununterscheidbar, ist das Level bei 50%. Ist das ID-Level bei 0% oder 100% spricht dies für ein schlechtes Modell, das deutlich unterscheidbare Bilder im Vergleich zu Menschen produziert. Mehr Details zu den visuellen Turing-Tests sind unter [LST15, § S5] zu finden.

3.3.4. Vergleich der Ergebnisse

In der Abbildung 3.5 sind die Ergebnisse für die Aufgaben aus Abbildung 2.1 dargestellt. Bei der Klassifikationsaufgabe (**A**) über unbekannte Buchstaben anhand eines Beispiels ist das Modell mit nur 3.3% Fehler bemerkenswerterweise erfolgreicher als der Mensch mit 4.9% Fehlerquote. Bei der Generierung eines weiteren Beispiels (**C**) konnten die Probanden nicht zwischen Mensch und Modell unterscheiden, was das ID-Level von 52% beweist. Auch bei dem Erfinden von neuen Buchstaben zeigt das erzielte ID-Level von 49% für die Aufgabe (**D**) im Stil eines Alphabetes, und 51% für das Erfinden ohne Einschränkungen (**E**), die Ununterscheidbarkeit der Modellausgabe zu der menschlicher Probanden. Sollten die Probanden die neuen Beispiele nicht anhand der statischen Bilder, sondern anhand von normierten Videos bewerten, lag das ID-Level bei 59%.
Die Experimente mit der modifizierten Modellarchitektur, wo auf Erfahrungsübertrag oder Kompositionalität verzichtet wird zeigen, dass die Kernprinzipien in der Tat essentiell für die Leistung des Modells sind.
Wir betonen, dass die fünf Aufgaben alle vom gleichen Modell gelöst worden sind, und dennoch eine Leistung auf der Ebene menschlicher Intelligenz erzielen.

Bemerkung 55. In die Konstruktion der Modellarchitektur ist sehr stark das kognitionswissenschaftliche Verständnis wie Menschen Buchstaben erkennen, eingeflossen. Insbesondere wird durch die generative Darstellung über die Zeichenanweisung die innere Kausalität eines Buchstabenkonzepts umgesetzt. Das Einfließen des Expertenwissens wird auch in den Ergebnissen im besonderen deutlich. Für die Leistung auf dem Level menschlicher Intelligenz sind die Kernprinzipien Erfahrungsübertrag und Kompositionalität essentiell. Die Ergebnisse des modifizierten Modells zeigen aber, dass allein die Modellbeschreibung bereits zu 11% und 14% Klassifikationsfehler bei der Aufgabe **A** aus Abbildung 2.1 führen. Im Vergleich dazu ist der Klassifikationsfehler mit 3.3% für das Modell zwar deutlich besser, dennoch zeigt der kleine zweistellige Fehler bei den Modifikationen, dass zum einen das Lernen der Wahrscheinlichkeitsverteilungen und zum anderen der kompositionale Aufbau der Schemas nur einen relativ geringen Teil in die Leistung des Modells einspielen.
Auch bei den generativen Aufgaben ist das ID-Level mit etwa 64% und 68% beim Erfinden unbekannter Buchstaben recht gut. Das Lernen der genauen Parameter der Wahrscheinlichkeitsverteilungen ist wichtig, damit die Modellausgabe ununterscheidbar ist, doch auch ohne diesen Erfahrungsübertrag generiert das Modell relativ gute Bilder, weil der Aufbau der Zeichenanweisung so geschickt gewählt ist.

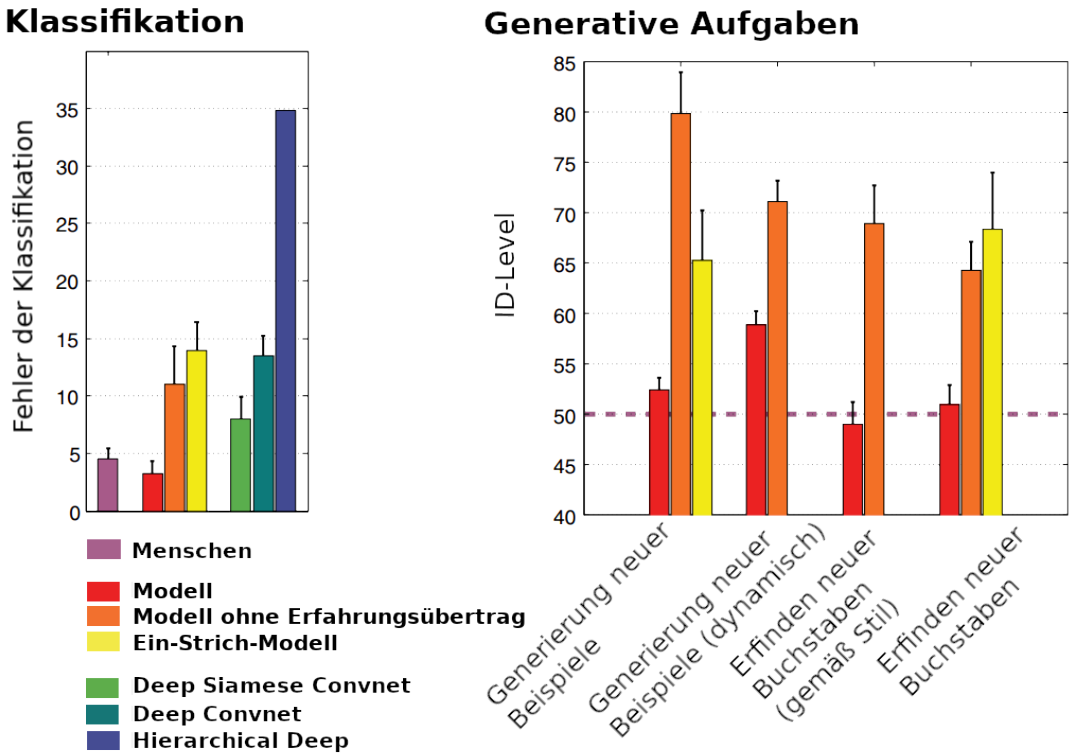

Abbildung 3.5.: Im Diagramm links ist der Fehler der Klassifikation anhand eines Beispiels, in Abbildung 2.1 **A**, dargestellt. Wir sehen, dass das beschriebene Modell (rot) mit 3.3% Fehler eine bessere Klassifikation als der durchschnittliche Fehler der menschlichen Probanden mit 4.9% erzielte. Die Kerneigenschaften Erfahrungsübertrag (orange) und Kompositionalität (gelb) spielen eine wichtige Rolle für den niedrigen Fehler, wie die Ergebnisse der modifizierten Modellarchitektur mit 11% und 14% Klassifikationsfehler. Bei dem Modell ohne Erfahrungsübertrag werden hierbei nur die Verteilungen auf der Ebene des variierten Schemas angepasst. Auch im Vergleich zu den anderen Ansätzen aus Abschnitt 3.3.2, Deep Siamese Convnet, Deep Convnet und Hierarchial Deep führt der Ansatz mit Bayes'schen Programmen zu einem deutlich kleinerem Klassifikationsfehler zeigen. Dieser liegt bei 8%, 13.5% und 34.8%.
Rechts werden die ID-Level auf den generativen Aufgaben dargestellt. Deutlich kann verifiziert werden, dass die Kernprinzipien Kompositionalität und Erfahrungsübertrag essentiell für die Qualität der neu generierten Bilder sind. So ist bei der Generierung neuer Beispiele das ID-Level des Modells bei 52%, des Modells ohne Erfahrungsübertrag (nur variiertes Schema) bei 65% und beim Ein-Strich-Modell bei 80%. Bei dem Erfinden neuer Buchstaben ohne Einschränkungen liegen die ID-Level bei 51%, 64% und 69%. Bei der dynamischen Generierung liegen diese Werte bei 59% und 71% für das Modell ohne Erfahrungsübertrag. Bei dem Erfinden neuer Buchstaben im Stile eines Alphabetes liegen die ID-Level bei 49% und 69%.
Abbildung modifiziert aus [LST15].

Kapitel 4.

Schluss

Die Einleitung begannen wir mit der Frage *Can machines think?*, zu deutsch *Können Computer denken?* [Tur50]. In dieser Ausarbeitung haben wir ein Rahmenwerk kennengelernt, dass „wie ein Mensch denkt“.

Das *Maschinelle Lernen mit Bayes'schen Programmen* stellt eine Modellierungsmöglichkeit für generative kompositionale Konzepte dar. Am Buchstabendatensatz *Omniglot* entwickelt [LST15], konnte diese Modellierung die herkömmlichen Deep Learning Ansätze in fünf Aufgaben aus dem Bereich des Begriffslernens weit übertreffen. Menschen können herausragend Begriffe anhand weniger Beispiele lernen, insbesondere, wenn sie bereits Erfahrung mit ähnlichen Konzepten hatten [Eck91], [LUS16]. Sogenannte künstliche neuronale Netze mit mehr als hundert Schichten wie etwa in [He+15] erzielen zwar beeindruckende Ergebnisse, benötigen allerdings sehr viele Daten und Rechenaufwand, um ein einziges Konzept zu lernen. Im Gegensatz dazu benötigt die vorgestellte Modellierung nur wenige Daten, und kann die gelernte Erfahrung sehr gut auf unbekannte Begriffe übertragen. Gleichzeitig fließt in die Modellierung sehr viel Expertenwissen hinein. Die überragenden Ergebnisse auf dem Buchstabendatensatz, etwa der kleinere Klassifikationsfehler mit 3.3% als der des Durchschnitts der menschlichen Probanden mit 4.9%, sind in großen Teilen durch die Eingrenzung der Möglichkeiten durch die Modellarchitektur geschuldet. Dies zeigen insbesondere die Ergebnisse der Modellmodifikation ohne Erfahrungsübertrag, bei der ohne gelernte Parameter der multivariaten Verteilungen bereits ein Klassifikationsfehler von nur 11% erreicht wird.

Dennoch ist die Modellierung samt Unterscheidung von Schlüsselcharakteristika zu Variabilität für andere Begriffe interessant. Die innere Logik des vorgestellten Rahmenwerks in Kapitel 2 erlaubt eine skizzenhafte Beschreibung von Konzepten. Die identifizierten wesentlichen Schritte und Modellierungsaspekte zeigen das Potential, dass auf anderen Daten ähnlich gute Ergebnisse erzielt werden können. Dafür wird insbesondere eine genügend gute Repräsentation benötigt, bei welcher das Verständnis menschlicher Intelligenz durch Kognitionswissenschaftlern eine wichtige Rolle spielt.
Die vorgestellte Herangehensweise ist insbesondere für Mensch-Maschine-Interaktion interessant, wie wir am Beispiel von Häuserbeschreibungen gezeigt haben. Die Anwendungen auf andere Datensätze in der Literatur, wie auf sich bewegende Strichmännchen [CTH17] oder ausgesprochene Wörter [LLT14] zeigen auf, dass die Herangehensweise einer sehr sorgfältigen Modellierung bedarf. Insbesondere wurden auf anderen Daten

nur mittelmäßige Ergebnisse erzielt, was vermutlich an einer ungeeigneten Modellkonstruktion liegt. Diese Schwäche versucht etwa Ellis durch das Lernen der Programme [ELT15] zu umgehen. Dennoch wird für die Repräsentation der Konzepte im Modell viel Expertenwissen benötigt und das Rahmenwerk lässt sich nur schwer auf neue Begriffe über die Buchstaben hinaus übertragen.

Das Modell für die Buchstaben, dass in Kapitel 3 vorgestellt wird, stellt Buchstaben durch ihre Zeichenanweisung dar. Dies überzeugt neben dem geringen Klassifikationsfehler auch in visuellen Turing-Test, bei denen Probanden nicht zwischen Modellausgabe und den generierten Bildern durch Menschen unterscheiden konnten. Zu den Schwächen des Modells gehört aber, dass repetitive Strukturen nur in einem gewissen Maß erkannt werden (nur durch die Markovkette der Wahl der Primitive). Außerdem geht die Symmetrie und Parallelität innerhalb eines Buchstabens nicht in die Modellierung mit ein und optionale Elemente, wie der Querstrich bei einer Sieben werden nicht erkannt. Trotz der identifizierten Schwächen ist die Leistung des Modells auf dem Level menschlicher Intelligenz.

Im Hinblick auf sogenannte erklärbare künstliche Intelligenz (englisch: *explainable AI*, siehe etwa [Xu+19]) stellt das vorgestellte *Maschinelle Lernen mit Bayes'schen Programmen* einen denkbaren Ansatz dar. Erklärbare künstliche Intelligenz beschäftigt sich damit, auf welche Weise Algorithmen aus dem Bereich der künstlichen Intelligenz zu ihren Ergebnissen kommen. In der vorgestellten Modellierung lässt sich dies genau nachvollziehen, was eine breite Anwendung in relevanten Gebieten wie der Medizin denkbar macht. Darüber hinaus können wir nach dem Stand kognitionswissenschaftlicher Forschung sagen, dass bei Anwendung auf dem Buchstabendatensatz das Modell tatsächlich „wie ein Mensch denkt".

Anhang A.

Grundlagen der Wahrscheinlichkeitstheorie

Im Folgenden tragen wir kurz wichtige Definitionen aus der Wahrscheinlichkeitstheorie zusammen, deren Verständnis essentiell für die vorliegenden Betrachtungen sind. Dabei orientieren wir uns am Skript zur Vorlesung *Einführung in die Wahrscheinlichkeitstheorie*, das von Professor Bovier an der Universität Bonn im Wintersemester 2019/20 gehalten wurde und unter [Bov20] zu finden ist, sowie am Skript [Ebe18] und den Büchern [Kle13] und [Cza11].

Wir beginnen mit Definitionen zum Grundbegriff σ-Algebra.

Definition 56. Sei Ω eine Menge. Eine Teilmenge der Potenzmenge $\mathscr{A} \subset \mathcal{P}(\Omega)$ heißt $\sigma-$**Algebra** falls gilt:

i) Es gilt $\Omega \in \mathscr{A}$.

ii) Für $A \in \mathscr{A}$ ist $A^c = \Omega \backslash A \in \mathscr{A}$.

iii) Seien $A_i \in \mathscr{A}$ mit $i \in I$ für I abzählbar. Dann ist $\bigcup_{i \in I} \in \mathscr{A}$.

Elemente einer σ-Algebra $A \in \mathscr{A}$ nennen wir **Ereignisse**. Das Paar $(\Omega, \mathscr{A})$ heißt **messbarer Raum**.

Beispiel 57. Ein Beispiel für eine σ-Algebra ist die Potenzmenge $\mathcal{P}(A)$ einer Menge A.

Definition 58. Sei $\mathscr{E} \subset \mathcal{P}(\Omega)$ für eine σ-Algebra $\mathscr{A}$. Wir nennen $\mathscr{E}$ einen **Erzeuger** von $\mathscr{A}$, falls $\mathscr{A}$ die kleinste σ-Algebra ist, die $\mathscr{E}$ enthält.

Definition 59. Sei Ω eine beliebige Menge. Eine Teilmenge der Potenzmenge $\emptyset \neq \mathscr{B} \subset \mathcal{P}(\Omega)$ heißt **durchschnittsstabil**, falls für alle $A, B \in \mathscr{B}$ gilt, dass $A \cap B \in \mathscr{B}$.

Aufbauend auf dem Begriff der σ-Algebra können wir die Definitionen von Wahrscheinlichkeitsmaß, Wahrscheinlichkeitsraum, bedingte Wahrscheinlichkeit und Unabhängigkeit angeben.

Definition 60. Gegeben sei eine σ-Algebra $\mathscr{A}$ über Ω. Ein **Wahrscheinlichkeitsmaß** ist eine Funktion $\mathbb{P} : \mathscr{A} \to \mathbb{R}_0^+$, so dass die Kolmogorovschen Axiome erfüllt sind:

i) $\mathbb{P}(\Omega) = 1$, $\mathbb{P}(\emptyset) = 0$,

ii) Für paarweise disjunkte Mengen $A_i \in \mathscr{A}$ gilt $\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$.

Definition 61. Ein **Wahrscheinlichkeitsraum** $(\Omega, \mathscr{A}, \mathbb{P})$ ist ein Tripel aus Ergebnismenge Ω, σ-Algebra $\mathscr{A}$, und Wahrscheinlichkeitsmaß $\mathbb{P}$.

Im Folgenden fixieren wir einen Wahrscheinlichkeitsraum $(\Omega, \mathscr{A}, \mathbb{P})$.

Definition 62. Seien $A, B \in \mathscr{A}$ mit $\mathbb{P}(B) > 0$. Dann ist die **bedingte Wahrscheinlichkeit** von A gegeben B

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Bemerkung 63. Die eingeschränkte Menge $\mathscr{A} \cap B = \{A \cap B, A \in \mathscr{A}\}$ ist eine σ-Algebra und die bedingte Wahrscheinlichkeit $\mathbb{P}(\cdot|B) : \mathscr{A} \cap B \to \mathbb{R}_0^+$ definiert ein Wahrscheinlichkeitsmaß auf dieser Menge. Ein Beweis findet sich etwa in [Bov20, Satz 3.2].

Definition 64. Die Ereignisse A, B heißen **unabhängig**, falls gilt

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B).$$

Bemerkung 65. Falls $\mathbb{P}(B) > 0$ ist dies äquivalent zu

$$\mathbb{P}(A|B) = \mathbb{P}(A).$$

Mit Definition 62 können wir den *Satz von Bayes*, der zentral in der bayes'schen Statistik ist, benennen. Wir werden ihn in fast allen folgenden Betrachtungen benutzen. In seiner einfachsten Form lautet er wie folgt.

Satz 66. *Seien $\mathcal{D}, h \in \mathscr{A}$ Ereignisse mit $\mathbb{P}(\mathcal{D}) > 0, \mathbb{P}(h) > 0$. Dann*

$$\mathbb{P}(h|\,\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}\,|h)\,\mathbb{P}(h)}{\mathbb{P}(\mathcal{D})}.$$

Beweis. Definition 62 umgeformt gibt $\mathbb{P}(\mathcal{D} \cap h) = \mathbb{P}(\mathcal{D}\,|h)\,\mathbb{P}(h)$. Damit gilt

$$\mathbb{P}(h|\,\mathcal{D}) \overset{Def.\ 62}{=} \frac{\mathbb{P}(\mathcal{D} \cap h)}{\mathbb{P}(\mathcal{D})} = \frac{\mathbb{P}(\mathcal{D}\,|h)\,\mathbb{P}(h)}{\mathbb{P}(\mathcal{D})}.$$

□

Bemerkung 67. Diesen Satz haben wir auch in den Grundlagen unter Satz 1 vorgestellt.

Der Nenner $\mathbb{P}(\mathcal{D})$ von Satz 1 lässt sich mit dem Satz der totalen Wahrscheinlichkeit umschreiben.

Satz 68. *Sei $h_n \in \mathscr{A}, n \in \mathbb{N}$ eine Folge von paarweise disjunkten Mengen mit $\bigcup_{n\in\mathbb{N}} h_n = \Omega$. Für alle $\mathcal{D} \in \mathscr{A}$ gilt dann*

$$\sum_{n\in\mathbb{N}} \mathbb{P}(\mathcal{D}\,|h_n)\,\mathbb{P}(h_n) = \mathbb{P}(\mathcal{D}).$$

Beweis. Es gilt

$$\sum_{n\in\mathbb{N}} \mathbb{P}(\mathcal{D}\,|h_n)\,\mathbb{P}(h_n) \overset{Def.\ 62}{=} \sum_{n\in\mathbb{N}} \mathbb{P}(\mathcal{D} \cap h_n) = \mathbb{P}(\mathcal{D} \cap \bigcup_{n\in\mathbb{N}} h_n) = \mathbb{P}(\mathcal{D} \cap \Omega) = \mathbb{P}(\mathcal{D}).$$

□

Der Begriff der Zufallsvariable auf einem Wahrscheinlichkeitsraum ermöglicht, über der gleichen Grundmenge mehrere Größen zu untersuchen.

Definition 69. Eine Abbildung $X : \Omega \to \mathbb{R}$ heißt **Zufallsvariable**, falls sie messbar ist, das heißt $X^{-1}(B) = \{\omega \in \Omega | X(\omega) \in B\} \in \mathscr{A}$ für alle $B \in \mathcal{B}(\mathbb{R})$ gilt. Eine Zufallsvariable heißt **diskret**, falls sie nur abzählbar viele verschiedene Werte annimmt.
Das Wahrscheinlichkeitsmaß $\mathbb{P}_X(B) := \mathbb{P}(\{\omega \in \Omega | X(\omega) \in B\}) = \mathbb{P}(X^{-1}(B))$ für $B \in \mathcal{B}(\mathbb{R})$ heißt **Verteilung** von X.

Bemerkung 70. Die **Borel'sche σ-Algebra** $\mathcal{B}(\mathbb{R})$ ist die σ-Algebra über $\mathbb{R}$, die von allen Mengen der Form $(-\infty, c]$ mit $c \in \mathbb{R}$ erzeugt wird.

Aufbauend auf dem Begriff der Zufallsvariable können wir die Wahrscheinlichkeitsfunktion und Wahrscheinlichkeitsdichte definieren.

Definition 71. Sei $X : \Omega \to \mathbb{R}$ eine diskrete Zufallsvariable, die die Werte $\{x_i\}_{i \in I}$ mit abzählbarer Indexmenge I annimmt. Die Funktion $\rho_X : \Omega \to [0, 1]$, definiert durch

$$\rho_X(x) = \begin{cases} \mathbb{P}(X = x_i) & \text{es gibt } i \in I \text{ mit } x = x_i \\ 0 & x \notin \{x_i\}_{i \in I} \end{cases}$$

heißt **Wahrscheinlichkeitsfunktion von X**.

Definition 72. Wir fixieren eine Zufallsvariable X. Dann heißt X **absolut stetig**, falls es $\rho_X : \mathbb{R} \to \mathbb{R}_0^+$ Lebesgue-integrierbar mit $\int_{\mathbb{R}} \rho_X(x)\, dx = 1$ gibt, so dass

$$\mathbb{P}(X \in B) = \int_B \rho_X(x)\, dx$$

für alle $B \in \mathcal{B}(\mathbb{R})$ gilt. In diesem Fall heißt ρ_X **Wahrscheinlichkeitsdichte** oder nur **Dichte** von X.

Bemerkung 73. Meistens schreiben wir ρ_X oder nur $\rho(x)$, wenn die zugehörige Zufallsvariable klar ist.

Definition 74. Der **Erwartungswert** einer diskreten Zufallsvariable X mit Werten $\{x_i\}_{i \in I}$ für eine abzählbare Indexmenge I ist

$$\mathbb{E}(X) = \sum_{i \in I} x_i \cdot \mathbb{P}(X = x_i),$$

falls diese Reihe absolut konvergent ist, das heißt

$$\sum_{i \in I} |x_i| \cdot \mathbb{P}(X = x_i) < \infty.$$

Für eine absolut stetige Zufallsvariable X mit Dichte ρ ist der Erwartungswert durch

$$\mathbb{E}(X) = \int_{\mathbb{R}} x\rho(x)\, dx$$

definiert, sofern

$$\int_{\mathbb{R}} |x|\rho(x)\, dx < \infty.$$

Sollte Summe oder Integral nicht absolut konvergieren, so sagen wir, dass der Erwartungswert nicht existiert.

Beispiel 75. Einige wichtige Verteilungen sind die folgend vorgestellten, siehe auch [Bov20, § 2.3] und [Cza11, § 1.2].

- Das **Dirac-Maß** $\delta_t(A) = \mathbb{1}_A(t)$ für $A \in \mathscr{A}$, wobei $\mathbb{1}_A$ die Indikatorfunktion der Menge A sei.
- Die **Bernoulli-Verteilung** $Ber(p)$ mit

  $$\mathbb{P} = p\delta_1 + (1-p)\delta_0$$

  für ein $0 \leq p \leq 1$.
- Die **Gleichverteilung** $\mathcal{U}(a,b)$ auf dem Intervall (a,b) für $a < b$ mit Verteilung

  $$\rho(x) = \frac{1}{b-a}\,\mathbb{1}_{(a,b)(x)}\,.$$

- Die **Normalverteilung** $\mathcal{N}(\mu, \sigma^2)$ mit Dichte

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

  Erwartungswert $\mu \in \mathbb{R}$ und Varianz σ^2 mit $\sigma > 0$.
- Die **Exponential-Verteilung** $Exp(a)$ mit Dichte

  $$\rho(x) = a \cdot \mathrm{e}^{-a \cdot x}\,\mathbb{1}_{[0,\infty)}(x)$$

  für $a > 0$.
- Die **Gamma-Verteilung** $\mathcal{G}(p,b)$ mit inversem Skalenparameter $b > 0$ und Formparameter $p > 0$. Die Dichte

  $$\gamma(x) = \begin{cases} \frac{b^p}{\Gamma(p)} x^{p-1} e^{-bx} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

  beschreibt die Verteilung, wobei die Funktion Γ die Gammafunktion notiert.
- Die **empirische Verteilung** für Daten $\mathcal{D}$, definiert durch $\rho(x_i) = \mathbb{P}(x_i) = \frac{\#\{x_i\}}{|\mathcal{D}|}$ wobei $\#\{x_i\}$ die Anzahl der gemessenen Werte x_i in den Daten $\mathcal{D}$ ist.

Zuletzt möchten wir einen Konvergenzbegriff einführen und das sogenannte Gesetz der großen Zahlen ohne Beweis einführen.

Definition 76. Sei X eine Zufallsvariable und X_n eine Folge von Zufallsvariablen. X_n konvergiert **fast sicher** gegen X, falls gilt

$$\mathbb{P}(\lim_{n\to\infty} X_n = X) = \rho(\{\omega \in \Omega | \lim_{n\to\infty} X_n(\omega) = X(\omega)\}) = 1.$$

Damit können wir das *Starke Gesetz der großen Zahlen* zitieren.

Satz 77. *Seien X_n unabhängig, identisch verteilte, integrierbare Zufallsvariablen, so dass der Erwartungswert $\mathbb{E}(X_1)$ existiert. Dann gilt*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_i = \mathbb{E}(X_1) \quad \textit{fast sicher.}$$

Beweis. Ein Beweis findet sich etwa in [Bov20, § 6]. □

Anhang B.

Übersetzungen

In der vorliegenden Arbeit wurde auf größtenteils englische Literatur zurückgegriffen und viele Begriffe zum Teil frei in das Deutsche übersetzt. Eine Übersicht der Übersetzungen findet sich in der folgenden Tabelle.

Deutsch	Englisch
Maschinelles Lernen mit Bayes'schen Programmen	Bayesian Program Learning
A-priori Verteilung	Prior
A-posteriori Verteilung	Posterior
Likelihood/Plausibilität	Likelihood
Schätzen	Inference
Begriffslernen	Concept Learning
Schema	Type Level
Variiertes Schema	Token Level
Buchstabe	Character
Kompositionalität	Compositionality
Kausalität	Causality
Übertragen von Erfahrungen	Learning to Learn
Strich	Stroke
Teilstrich	Substroke
Grundformen	Primitives
Parsing	Parsing
Zeichenanweisung	Motor Program
Gauß'sche Mischverteilung	Gaussian Mixture Modell
Stochastische Irrfahrt	Random Walk
Stochastischer Irrläufer	Random Walker

Anhang C.

Weitere Algorithmen

Algorithmus 4 Generierung eines variierten Schemas $\tilde{\theta}$ aus der multivariaten Verteilung, gegeben ψ

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{B}|\tilde{S},\tilde{R},\psi)\cdot\prod_{i=1}^{\kappa}\rho(\tilde{R}_i|R_i)\rho(\tilde{S}_i|S_i).$$

procedure GeneriereVariiertesSchema(ψ)
  for $i = 1,\ldots,\kappa$ **do**
    $\tilde{S}_i \leftarrow \rho(\tilde{S}_i|s_i)$ ▷ Variierte Teile.
    $\tilde{R}_i \leftarrow \rho(\tilde{R}_i|R_i)$ ▷ Variierte Relation.
  end for
  $\tilde{S} \leftarrow \{\tilde{S}_1,\ldots,\tilde{S}_\kappa\}$
  $\tilde{R} \leftarrow \{\tilde{R}_1,\ldots,\tilde{R}_\kappa\}$
  $\tilde{B} \leftarrow \rho(\tilde{B}|\tilde{S},\tilde{R},\psi)$ ▷ Bildtransformation.
  $\tilde{\theta} \leftarrow \{\tilde{S},\tilde{R},\tilde{B}\}$
  return $\tilde{\theta}$ ▷ Generiertes variiertes Schema.
end procedure

Algorithmus 5 Generierung eines Bildes I aus der multivariaten Verteilung, gegeben $\tilde{\theta}$. Es gilt dann

$$\rho(I|\,\tilde{\theta}^{(T)}) = \begin{cases} 1 & \text{für } f(\tilde{\theta}^{(T)}) = I \\ 0 & \text{sonst.} \end{cases}$$

procedure GeneriereBild($\tilde{\theta}$)
  $I \leftarrow f(\tilde{\theta})$ ▷ Generiere deterministisch das Bild.
  return I
end procedure

Algorithm 6 Generierung eines Schemas ψ mit der multivariaten Verteilung

$$\rho(\psi) = \rho(\kappa) \prod_{i=1}^{\kappa} \rho(S_i)\rho(R_i|S_1, \ldots, S_{i-1}).$$

Algorithmus frei aus [LST15].

procedure GENERIERESCHEMA
  $\kappa \leftarrow \rho(\kappa)$ ▷ Ziehe Anzahl der Striche.
  for $i = 1, \ldots, \kappa$ **do**
    $n_i \leftarrow \rho(n_i|\kappa)$ ▷ Ziehe Anzahl der Teilstriche.
    $S_i \leftarrow$ GENERIERESTRICH(i, n_i) ▷ Ziehe Strich.
    $\zeta_i \leftarrow \rho(\zeta_i)$ ▷ Ziehe Typ der Relation.
    $R_i \leftarrow \rho(R_i|\zeta_i, S_1, \ldots, S_{i-1})$ ▷ Ziehe Relation.
  end for
  $\psi \leftarrow \{\kappa, R, S\}$
  return ψ ▷ Schema.
end procedure

Algorithmus 7 Generierung des i-ten Striches mit n_i Teilstrichen. Die Verteilung ist

$$\rho(S_i) = \rho(z_i) \prod_{j=1}^{n_i} \rho(x_{ij}|z_{ij})\rho(y_{ij}|z_{ij}) \quad \text{mit} \quad \rho(z_i) = \rho(z_{i1}) \prod_{j=2}^{n_i} \rho(z_{ij}|z_{i(j-1)}).$$

Algorithmus frei aus [LST15].

procedure GENERIERESTRICH(i, n_i)
  $z_{i1} \leftarrow \rho(z_{i1})$ ▷ Ziehe ersten Teilstrich.
  for $j = 2, \ldots, n_i$ **do**
    $z_{ij} \leftarrow \rho(z_{ij}|z_{i(j-1)})$ ▷ Ziehe restliche Teilstriche.
  end for
  for $j = 1, \ldots, n_i$ **do**
    $x_{ij} \leftarrow \rho(x_{ij}|z_{ij})$ ▷ Ziehe Kontrollpunkte des j-ten Teilstrichs.
    $y_{ij} \leftarrow \rho(y_{ij}|z_{ij})$ ▷ Ziehe relative Größe des Teilstriches.
    $s_{ij} \leftarrow \{x_{ij}, y_{ij}, z_{ij}\}$
  end for
  $S_i \leftarrow \{s_{i1}, \ldots, s_{in_i}\}$ ▷ Definition des i-ten Strichs.
  return S_i ▷ i-ter Strich.
end procedure

Algorithmus 8 Generierung eines variierten Schemas $\tilde{\theta}$ aus ψ. Die Verteilung ist

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{L}|\tilde{S}, \tilde{R}, \tilde{A}, \tilde{\sigma_b}, \tilde{\epsilon}, \psi) \cdot \prod_{i=1}^{\kappa} \rho(\tilde{R}_i|R_i)\rho(\tilde{x}_i|x_i)\rho(\tilde{y}_i|y_i)\rho(\tilde{A}, \tilde{\sigma}_b, \tilde{\epsilon}).$$

Algorithmus frei aus [LST15].

procedure GeneriereVariiertesSchema(ψ)
  for $i = 1, \ldots, \kappa$ **do**
    $\tilde{R} \leftarrow \rho(\tilde{R})$ ▷ Relation bleibt erhalten.
    if $\tilde{\zeta} = entlang$ **then**
      $\tilde{\tau}_i \leftarrow \rho(\tilde{\tau}_i|\tau_i)$ ▷ Variabilität, an welcher Stelle der neue Strich anfängt.
    end if
    for $j = 1, \ldots, n_i$ **do**
      $\tilde{x}_{ij} \leftarrow \rho(\tilde{x}_{ij}|x_{ij})$ ▷ Variabilität der Kontrollpunkte.
      $\tilde{y}_{ij} \leftarrow \rho(\tilde{y}_{ij}|y_{ij})$ ▷ Variabilität der relativen Größe.
    end for
  end for
  $\tilde{A} \leftarrow \rho(\tilde{A})$ ▷ Ziehe globale Transformation.
  $\tilde{\epsilon} \leftarrow \rho(\tilde{\epsilon})$ ▷ Ziehe Pixelfehler.
  $\tilde{\sigma_b} \leftarrow \rho(\tilde{\sigma}_b)$ ▷ Ziehe Unschärfe.
  $\tilde{B} \leftarrow (\tilde{A}, \tilde{L}, \tilde{\sigma_b}, \tilde{\epsilon})$
  $\tilde{\theta} \leftarrow \{\tilde{S}, \tilde{R}, \tilde{B}\}$
  return $\tilde{\theta}$ ▷ Variiertes Schema.
end procedure

Algorithmus 9 Generierung des Bildes I aus $\tilde{\theta}$ für den Omniglot-Datensatz.

procedure GeneriereBild($\tilde{\theta}$)
  for $i = 1, \ldots, \kappa$ **do**
    for $j = 1, \ldots, n_i$ **do**
      $\tilde{T}_{ij} \leftarrow h(\tilde{L}_{ij}, \tilde{x}_{ij}, \tilde{y}_{ij})$▷ Konstruiere die Trajektorie des j-ten Teilstrichs des i-ten Strichs.
    end for
    $\tilde{T}_i \leftarrow \{\tilde{T}_{i1}, \ldots, \tilde{T}_{in_i}\}$
  end for
  $\tilde{T} \leftarrow \{\tilde{T}_1, \ldots, \tilde{T}_\kappa\}$
  $I \leftarrow \tilde{f}(\tilde{T}, \tilde{A}, \tilde{\sigma}_b, \tilde{\epsilon})$ ▷ Wende Filter, Unschärfe und Bildstörung an.
  return T ▷ Erstelltes binäres Bild.
end procedure

Algorithmus 10 Generierung eines Buchstabenschemas mit genau einem Strich aus der Verteilung

$$\rho(\psi) = \rho(n)\rho(X_1)\prod_{i=2}^{n}\rho(X_i|X_{i-1}).$$

Algorithmus aus [LST15].

procedure GeneriereEinStrichModell
  $n \leftarrow \rho(n)$ ▷ Ziehe Anzahl der Teilstriche.
  $X_1 \leftarrow \rho(X_1)$ ▷ Ziehe ersten Kontrollpunkt.
  for $i = 2, \ldots, n$ **do**
    $X_1 \leftarrow \rho(X_i|X_{i-1})$ ▷ Ziehe den nächsten Kontrollpunkt.
  end for
  $X \leftarrow \{X_1, \ldots, X_n\}$
  $\psi \leftarrow \{n, X\}$
  return ψ ▷ Buchstabenschema.
end procedure

Algorithmus 11 Generierung eines variierten Schemas $\tilde{\theta}$ aus ψ für das Ein-Strich-Modell. Die Verteilung ist

$$\rho(\tilde{\theta}\,|\psi) = \rho(\tilde{L}|\tilde{X}, \tilde{A}, \tilde{\sigma_b}, \tilde{\epsilon}, \psi) \cdot \prod_{i=1}^{n}\rho(\tilde{X}_i|X_i) \cdot \rho(\tilde{A}, \tilde{\sigma}_b, \tilde{\epsilon}).$$

procedure GenerierevariiertesSchema(ψ)
  for $i = 1, \ldots, n$ **do**
    $\tilde{X}_i \leftarrow \rho(\tilde{X}_i|X_i)$ ▷ Variabilität der Kontrollpunkte.
  end for
  $\tilde{A} \leftarrow \rho(\tilde{A})$ ▷ Ziehe globale Transformation.
  $\tilde{\epsilon} \leftarrow \rho(\tilde{\epsilon})$ ▷ Ziehe Pixelfehler.
  $\tilde{\sigma_b} \leftarrow \rho(\tilde{\sigma_b})$ ▷ Ziehe Unschärfe
  $\tilde{B} \leftarrow (\tilde{A}, \tilde{L}, \tilde{\sigma_b}, \tilde{\epsilon})$
  $\tilde{\theta} \leftarrow \{\tilde{X}, \tilde{B}\}$
  return $\tilde{\theta}$ ▷ Variiertes Schema.
end procedure

Anhang D.

Weitere Abbildungen und Tabellen

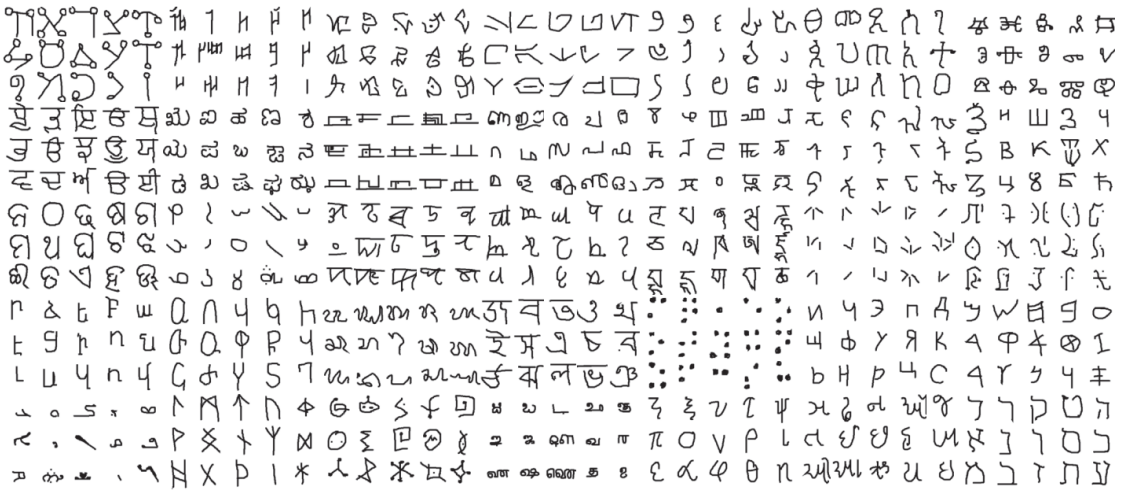

Abbildung D.1.: Einige Buchstaben aus dem Omniglot-Datensatz sind geordnet nach Alphabeten dargestellt. Die Buchstaben in den 3×5-Blöcken gehören jeweils zu einem Alphabet.
Zu sehen ist etwa das kyrillische Alphabet (rechts, zweites Alphabet von oben) oder die Braille-Schrift (fünfte Spalte von links, zweite Zeile von unten).
Jeder Buchstabe wurde mehrmals von Probanden gezeichnet und als Video durch eine Liste $x, y, time$ dargestellt. Abbildung entnommen aus [LST15].

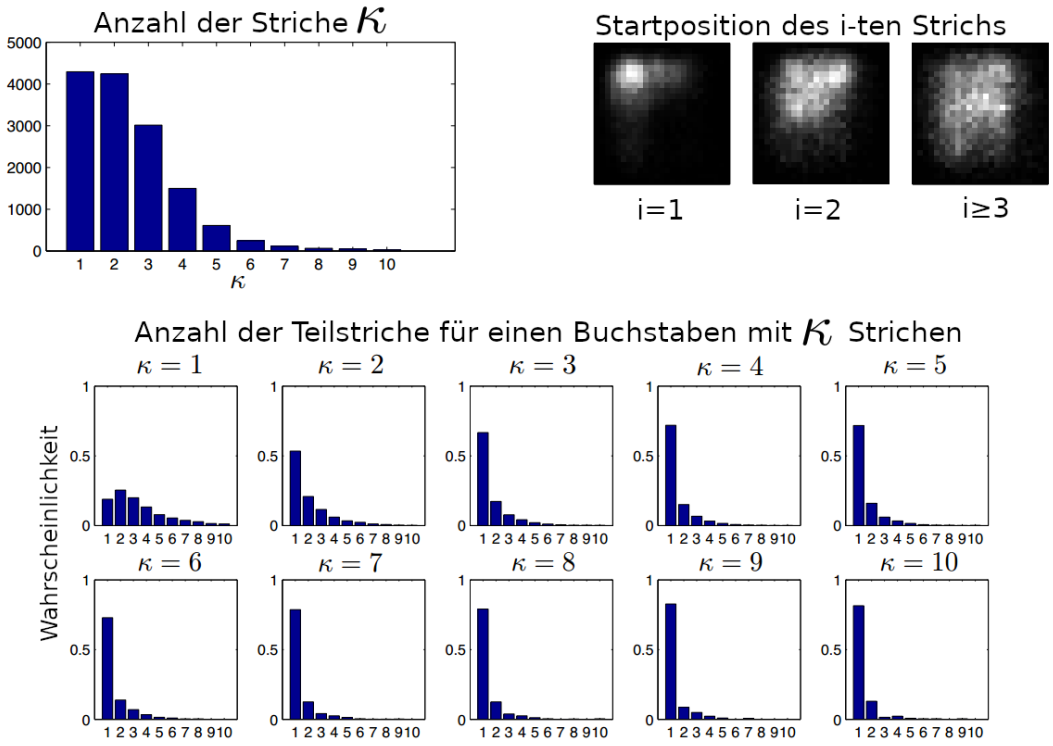

Abbildung D.2.: Dargestellt werden einige empirische Verteilungen, die aus den Daten entnommen wurden. Oben links ist im Balkendiagramm ist die Häufigkeit von κ dargestellt. Dabei bezeichnet κ die Anzahl der Male, die der Stift bei der Zeichnung des Buchstabens angesetzt wurde. Die 10 Balkendiagramme unten entsprechen wiederum der Wahrscheinlichkeit aus den empirischen Häufigkeiten für die Anzahl n_i der Teilstriche innerhalb eines Striches in Abhängigkeit der Anzahl der Striche κ. Oben rechts ist die empirische Verteilung der Startposition $\tilde{L}$ innerhalb des Bildrahmens dargestellt. Helle Werte kodieren dabei eine große, dunkle eine niedrige Wahrscheinlichkeit. Die drei Bilder korrespondieren zu den Startkoordinaten des ersten, zweiten oder der weiteren Striche.
Abbildung modifiziert aus [LST15].

	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$
Durchlauf 1	60.83s	131.96s	177.41s	267.14s	323.51s	423.68s	508.22s
Durchlauf 2	66.98s	131.42s	221.11s	272.20s	346.71s	410.93s	368.51s
Durchlauf 3	67.05s	125.62s	202.15s	295.06s	335.57s	388.09s	476.60s
Durchlauf 4	64.89s	125.10s	188.09s	291.63s	316.90s	414.04s	503.37s
Durchlauf 5	62.29s	129.80s	214.31s	272.42s	359.23s	404.47s	520.14s
Durchlauf 6	62.50s	135.11s	213.48s	277.69s	347.41s	424.16s	502.34s
Durchlauf 7	63.08s	123.47s	206.25s	212.54s	322.68s	362.11s	561.01s
Durchlauf 8	62.58s	133.31s	195.22s	272.07s	319.01s	381.18s	494.03s
Durchlauf 9	74.10s	129.88s	199.64s	264.94s	335.01s	397.33s	489.66s
Durchlauf 10	65.69s	132.56s	221.84s	242.95s	329.37s	391.53s	504.12s
Durchschnitt	65.00s	129.82s	203.95s	266.86s	333.54s	399.75s	492.80s

Tabelle D.1.: Darstellung der exakten Rechenzeiten für die Optimierung von K Parses am Beispielbild ϕ aus Abbildung 3.3. Gerechnet wurde auf einem Computer mit Linux (Ubuntu 18.04 mit einem 5.4.0 Kernel) und einem Intel® Core™ i5-8250U CPU @ 1.60GHz x 8 Prozessor mit 16 GB RAM.
Daten generiert durch modifizierten Code aus [Lak15]. Zusammengefasste Ergebnisse sind in Tabelle 3.1 zu finden.

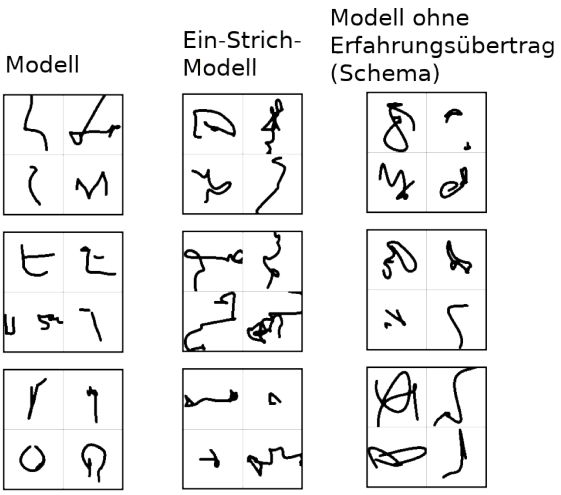

Abbildung D.3.: Dargestellt werden drei Ausgaben vom Modell, Ein-Strich-Modell und dem Modell ohne Erfahrungsübertrag (Schema) auf die Aufgabe, vier Buchstaben zu erfinden. Zu sehen ist deutlich, dass die Ausgabe des Modells realistischer wirkt. Die Ausgaben der modifizierten Modelle wirken nicht einer klaren Zeichenanweisung folgend und die Strichteile sind zu nah aneinander, sodass die Ausgabe etwas wie „Gekritzel“ und nicht wie ein Buchstabe wirkt.
Abbildung modifiziert aus [LST15].

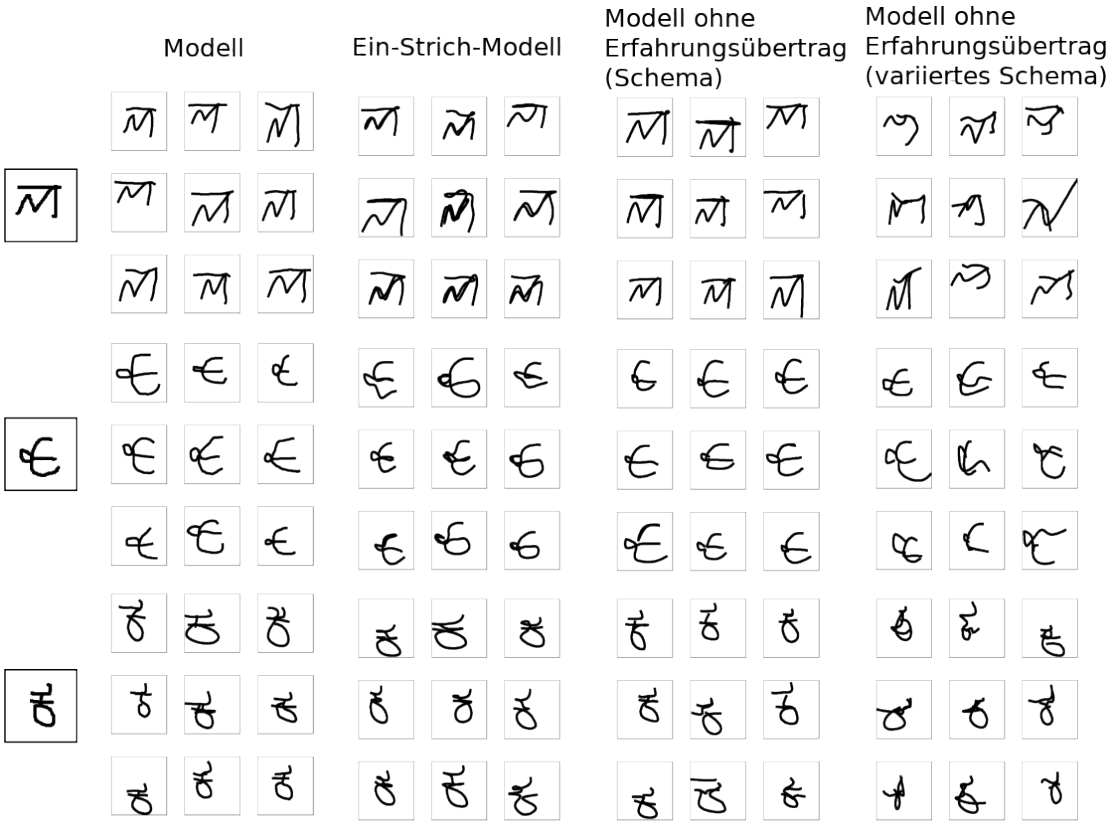

Abbildung D.4.: Dargestellt werden drei Ausgaben vom Modell, Ein-Strich-Modell und zwei Modellen ohne Erfahrungsübertrag (Schema oder variiertes Schema) auf die Aufgabe, zu einem Beispielbild eines unbekannten Buchstabens ein neues Beispiel zu generieren. Zu sehen ist deutlich, dass die Ausgabe des Modells die Schlüsselcharakteristika und erlaubte Variabilität besser umsetzt. Das Ein-Strich-Modell scheint die Striche doppelt zu malen oder unnatürliche Verbindungen einzufügen. Das Modell, dass auf der Ebene des Schemas keinen Erfahrungsübertrag hat, scheint recht realistische Bilder zu produzieren. Beim genauen hinsehen wird aber deutlich, dass die generierten Buchstaben Lücken aufzuweisen scheinen. Dies liegt daran, dass die einzige erlaubte Relation $frei$ ist. Wird auf der Ebene des variierten Schemas der Erfahrungsübertrag verringert, in dem die erlaubten Varianzen der Normalverteilungen verdreifacht wurden und der Startpunkt aus der stetigen Gleichverteilung auf dem Bildraum gezogen wurde, so wirken die generierten Beispiele unnatürlich. Die Schlüsselcharakteristika sind nicht mehr in jeder Abbildung genug umgesetzt.
Abbildung modifiziert aus [LST15].

Literatur

[Age20] Simon Ager. *Omniglot, the online encyclopedia of writing systems & languages*. Webseite. https://www.omniglot.com/; zuletzt abgerufen am 01.07. 2020.

[Bes+13] Pierre Bessiere u. a. *Bayesian Programming*. Chapman und Hall/CRC, 2013. ISBN: 9781439880326.

[BF88] Mary K. Babcock und Jennifer J. Freyd. „Perception of Dynamic Information in Static Handwritten Forms". In: *The American Journal of Psychology* 101.1 (1988), S. 111–130. ISSN: 00029556. URL: http://www.jstor.org/stable/1422797.

[Bov20] Anton Bovier. „Einführung in die Wahrscheinlichkeitstheorie". Wintersemester 2019/20, Stand 24. Januar 2020. Bonn, 2020. URL: https://www.dropbox.com/s/wllhl2v4ccv2x7f/wt-new.pdf?dl=0.

[Cor+09] Thomas Cormen u. a. *Introduction to Algorithms*. 3. Aufl. The MIT Press, 2009. ISBN: 9780262033848.

[CTH17] Cheng, Meng-Zhen, Tang, Quan-Hua und Huang, Long-Jun. „Motion Learning Based on Bayesian Program Learning". In: *ITM Web Conf.* 12 (2017). DOI: 10.1051/itmconf/20171205011. URL: https://doi.org/10.1051/itmconf/20171205011.

[Cyn12] Ben Letham und Cynthia Rudin. „Probabilistic Modeling and Bayesian Analysis". Frühlingssemster 2012. MIT, 2012. URL: https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15_097S12_lec15.pdf.

[Cza11] Thorsten Czado Claudia und Schmidt. *Mathematische Statistik*. Statistik und ihre Anwendungen. Springer, Berlin, Heidelberg, 2011. ISBN: 9783642172601.

[Ebe18] Andreas Eberle. „Einführung in die Wahrscheinlichkeitstheorie". Wintersemester 2017/18, Stand 10. Februar 2018. Bonn, 2018. URL: https://wt.iam.uni-bonn.de/fileadmin/WT/Inhalt/people/Andreas_Eberle/Wtheorie17/Wtheorie1718.pdf.

[Eck91] Thomas Eckes. *Psychologie der Begriffe: Strukturen des Wissens und Prozesse der Kategorisierung*. Hogrefe, 1991. ISBN: 9783801704315.

[Ell+18] Kevin Ellis u. a. „Learning Libraries of Subroutines for Neurally–Guided Bayesian Program Induction". In: *Advances in Neural Information Processing Systems 31*. Hrsg. von S. Bengio u. a. Curran Associates, Inc., 2018, S. 7805–7815. URL: http://papers.nips.cc/paper/8006-learning-libraries-

of-subroutines-for-neurallyguided-bayesian-program-induction.pdf.

[ELT15] Kevin Ellis, Armando Lezama und Joshua Tenenbaum. „Unsupervised learning by program synthesis“. In: (Dez. 2015).

[EST16] Kevin Ellis, Armando Solar-Lezama und Josh Tenenbaum. „Sampling for Bayesian Program Learning“. In: *Advances in Neural Information Processing Systems 29*. Hrsg. von D. D. Lee u. a. Curran Associates, Inc., 2016, S. 1297–1305. URL: http://papers.nips.cc/paper/6082-sampling-for-bayesian-program-learning.pdf.

[Gel+13] Andrew Gelman u. a. *Bayesian Data Analysis*. 3. Aufl. Texts in Statistical Science. Chapman und Hall/CRC, 2013. ISBN: 9781439840955.

[He+15] Kaiming He u. a. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[Hof85] D. Hofstadter. *Metamagical themas: Questing for the essence of mind and pattern*. Basic Books, 1985. ISBN: 9781299605220.

[HTF09] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer-Verlag New York, 2009. ISBN: 9780387848570.

[Jay03] E. T. Jaynes. *Probability Theory - The Logic of Science*. Cambridge University Press, 2003. ISBN: 9780511790423.

[Jos19] Florian Jarre und Josef Stoer. *Optimierung - Einführung in mathematische Theorie und Methoden*. 2. Aufl. Masterclass. Springer Spektrum, 2019. ISBN: 9783662588543.

[Kle13] Achim Klenke. *Wahrscheinlichkeitstheorie*. 3. Aufl. Masterclass. Springer Spektrum, 2013. ISBN: 9783642360176.

[Koc15] Gregory Koch. „Siamese Neural Networks for One-Shot Image Recognition“. Masterarbeit. University of Toronto, 2015.

[KZS05] Gregory Koch, Richard Zemel und Ruslan Salakhutdinov. „Siamese neural networks for one-shot image recognition“. In: Bd. 2. ICML Deep Learning Workshop. 2005.

[Lak14] Brenden Lake. „Towards more human-like concept learning in machines: Compositionality, causality, and learning-to-learn“. Diss. Massachusetts Institute of Technology (MIT), 2014.

[Lak15] Brenden Lake. *BPL model for one-shot learning*. https://github.com/brendenlake/BPL. 2015.

[Lak19] Brenden Lake. *Omniglot data set for one-shot learning*. https://github.com/brendenlake/omniglot. 2019.

[Lec+98] Y. Lecun u. a. „Gradient-based learning applied to document recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324.

[LLT14] Brenden Lake, C.Y Lee und J.B. Tenenbaum. „One-shot learning of generative speech concepts“. In: *Proceedings of the 36th Annual Conference of the Cognitive Science Society*. 2014.

[LS90] Chia-Wei Liao und Jun S.Huang. „Stroke segmentation by bernstein-bezier curve fitting“. In: *Pattern Recognition* 23.6 (5 1990), S. 475–484.

[LST13] Brenden M. Lake, Ruslan Salakhutdinov und Joshua B. Tenenbaum. „One-shot learning by inverting a compositional causal process“. In: *Advances in Neural Information Processing Systems* 26 (2013).

[LST15] Brenden M. Lake, Ruslan Salakhutdinov und Joshua B. Tenenbaum. „Human-level concept learning through probabilistic program induction“. In: *Science* 350.6266 (2015), S. 1332–1338. ISSN: 0036-8075. DOI: 10.1126/science.aab3050. eprint: https://science.sciencemag.org/content/350/6266/1332.full.pdf. URL: https://science.sciencemag.org/content/350/6266/1332.

[LST19] Brenden M. Lake, Ruslan Salakhutdinov und Joshua B. Tenenbaum. *The Omniglot challenge: a 3-year progress report*. 2019. arXiv: 1902.03477 [cs.AI].

[LUS16] Brenden M. Lake, Tomer D. Ullman und Joshua B. Tenenbaum und Samuel J. Gershman. „Building Machines That Learn and Think Like People“. In: *CoRR* abs/1604.00289 (2016). arXiv: 1604.00289. URL: http://arxiv.org/abs/1604.00289.

[LW16] Chuan Li und Michael Wand. „Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks“. In: *CoRR* abs/1604.04382 (2016). arXiv: 1604.04382. URL: http://arxiv.org/abs/1604.04382.

[Mes19] Gaby Messer M. und Schneider. *Statistik - Theorie und Praxis im Dialog*. Springer-Lehrbuch. Springer Spektrum, 2019. ISBN: 9783662593387.

[Mur12] Kevin P. Murphy. *Machine Learning A Probabilistic Perspective*. Adaptive computation and machine learning series. The MIT Press, 2012. ISBN: 9780262018029.

[OJP17] Matthew C. Overlan, Robert A. Jacobs und Steven T. Piantadosi. „Learning abstract visual concepts via probabilistic program induction in a Language of Thought“. In: *Cognition* 168 (2017), S. 320–334. ISSN: 0010-0277. DOI: https://doi.org/10.1016/j.cognition.2017.07.005. URL: http://www.sciencedirect.com/science/article/pii/S0010027717302020.

[Reb16] Maxwell Rebo. *Generalizing BPL*. https://github.com/MaxwellRebo/PyBPL. 2016.

[Rob07] Christian Robert. *The Bayesian Choice: From Decision Theoretic Foundations to Computational Implementation*. Springer Texts in Statistics. Springer Science+Buisness Media, Jan. 2007. ISBN: 9780387715988.

[S L92] L. Lam und S. Lee und C. Suen. „Thinning methodologies-a comprehensive survey“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.9 (1992), S. 869–885.

[Sea80] John R. Searle. „Minds, brains, and programs“. In: *Behavioral and Brain Sciences* 3.3 (1980), S. 417–424. DOI: 10.1017/S0140525X00005756.

[STT13] R. Salakhutdinov, J. B. Tenenbaum und A. Torralba. „Learning with Hierarchical-Deep Models“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), S. 1958–1971.

[Ten99] Joshua B. Tenenbaum. „A Bayesian framework for concept learning“. Diss. Massachusetts Institute of Technology (MIT), 1999.

[Tur50] Alan M. Turing. „Computing Machinery and Intelligence“. In: *Mind* 59.October (1950), S. 433–60. DOI: 10.1093/mind/LIX.236.433.

[TX07] Joshua B. Tenenbaum und Fei Xu. „Word learning as Bayesian inference“. In: *Psychological Review* 116 (2007).

[Vid04] Brani Vidakovic. „Bayesian Statistics: Handouts“. Handout zum Kurs Bayesian Statistics for Engineers. Georgia Institute of Technology, 2004. URL: https://www2.isye.gatech.edu/isyebayes/handouts.html.

[Xu+19] Feiyu Xu u. a. „Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges“. In: *Natural Language Processing and Chinese Computing*. Hrsg. von Jie Tang u. a. Cham: Springer International Publishing, 2019, S. 563–574. ISBN: 9783030322366.

Abbildungsverzeichnis

1. Beispiel der Aufgabe *Generieren eines weiteren Beispiels eines unbekannten Buchstabens* auf dem Omniglot-Datensatz v

2.1. Aufgaben auf dem Omniglot-Datensatz aus dem Bereich des Begriffslernen 17
2.2. Darstellung der Zusammensetzung eines Buchstabens aus Primitiven, Strichen, Relationen. 19
2.3. Skizzenhafte Beschreibung eines Hauses durch Teile und ihre Relationen sowie Übersetzung in den Bildraum. 30

3.1. Umwandlung des Bildes in einen Graphen 40
3.2. Darstellung der verschiedenen Traversier-Möglichkeiten in der stochastischen Irrfahrt auf dem Buchstabenskelett. 41
3.3. Darstellung der verschiedenen gefunden Parses auf dem Buchstaben ϕ vor und nach Optimierung 43
3.4. Darstellung der Re-Optimierung bei zwei verschiedenen beziehungsweise zwei gleichen Buchstaben 44
3.5. Auswertung des Modells im Vergleich zu Menschen und anderen Modellen auf dem Omniglot-Datensatz. 48

D.1. Omniglot-Datensatz 61
D.2. Darstellung einiger gelernter empirischer Verteilungen aus dem Omniglot-Datensatz 62
D.3. Vergleich von Ausgaben des Modelles mit den Modifikationen für die Aufgabe *Buchstaben erfinden* 63
D.4. Vergleich von Ausgaben des Modelles mit den Modifikationen für die Aufgabe *Generierung eines weiteren Beispiels eines unbekannten Buchstabens* 64

Algorithmenverzeichnis

1. Metropolis-Hastings-Algorithmus um Stichproben aus der A-posteriori Verteilung zu ziehen 11

2. Darstellung eines Schemas ψ als Programm auf Basis der multivariaten Verteilung $\rho(\psi)$ 20
3. Allgemeines Schema mit den wesentlichen Schritten von *Maschinellem Lernen mit Bayes'schen Programmen.* 23

4. Darstellung eines variierten Schemas $\tilde{\theta}$ als Programm auf Basis der multivariaten Verteilung $\rho(\tilde{\theta}\,|\psi)$ 57
5. Deterministische Generierung des Bildes I aus dem variierten Schema $\tilde{\theta}$. 57
6. Darstellung des Schemas ψ als Programm für den Omniglot-Datensatz . . 58
7. Darstellung des Striches S als Programm für den Omniglot-Datensatz . . 58
8. Darstellung des variierten Schemas $\tilde{\theta}$ als Programm für den Omniglot-Datensatz 59
9. Deterministische Generierung des Bildes I aus dem variierten Schema $\tilde{\theta}$ für den Omniglot-Datensatz 59
10. Darstellung eines Schemas ψ als Programm auf Basis der multivariaten Verteilung aus dem Ein-Strich-Schema 60
11. Darstellung eines variierten Schemas $\tilde{\theta}$ als Programm auf Basis der multivariaten Verteilung aus dem Ein-Strich-Schema 60

Tabellenverzeichnis

3.1. Darstellung der durchschnittlichen Rechenzeiten für die Optimierung von K Parses 43

D.1. Exakte Rechenzeiten für die Optimierung von K Parses am Beispielbild ϕ 63