Automatisiertes Komponieren mit Hilfe von Markov Ketten

Masterarbeit von

Philipp Beyer
[redacted]
Studiengang: Lehramt Mathematik
[redacted]

Karlsruher Institut für Technologie (KIT)
Steinbuch Centre for Computing (SCC)

1. Prüfer:	Prof. Dr. Martin Frank
2. Prüferin:	Dr. Ingrid Lenhardt
Betreuende Mitarbeiterin:	Stephanie Hofmann
Abgabetermin:	02.11.2021

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einleitung

„In schöner Musik steckt meist Mathematik"

Mit diesem Titel warb 2019 eine Ausstellung in Heidelberg zum Thema Zusammenhang zwischen Mathematik und Musik. Bereits die Pythagoräer erkannten harmonische Zahlenverhältnisse für die Musik und zeigten, dass harmonisch klingende Noten mathematischen Regeln gehorchen (vergleiche mySciene, 2019). Mit Hilfe solcher mathematischer Regeln, die man aus bekannten Liedern ableiten kann, können Vorhersageregeln für die Komposition neuer Musik entwickelt werden. Eine wesentliche Rolle spielt hierbei die Künstliche Intelligenz (KI), die es ermöglicht bekannte Lieder einzulesen und Vorhersageregeln zu entwickeln. So wurde z.B. Beethovens unvollendete 10. Sinfonie mit Hilfe von KI vollendet (vergleiche FAZ, 2021).

Die Mathematik hinter der automatischen Komposition kann auch auf Schulniveau reduziert werden und den Schülerinnen und Schülern[1] als Anwendungsbeispiel für Mathematik im Alltag aufgezeigt werden. Durch solche Alltagsbeispiele kann die gesellschaftliche Bedeutung von Mathematik gesteigert werden.

Ziel dieser Abschlussarbeit ist die Entwicklung eines Vorhersagemodells auf Basis von Markov Ketten zur automatischen Komposition von Musik als Lernmodul für Schüler. Hierbei werden sowohl die mathematisch-fachlichen Hintergründe, als auch die didaktischen und methodischen Prinzipien, die dem Lernmodul zu Grunde liegen, betrachtet.

[1] Aus Gründen der besseren Lesbarkeit wird im Folgenden auf die gleichzeitige Verwendung weiblicher und männlicher Sprachformen verzichtet und das generische Maskulinum verwendet. Sämtliche Personenbezeichnungen gelten gleichermaßen für beide Geschlechter.

2. Didaktischer Hintergrund

In diesem Kapitel wird das CAMMP Projekt, in dem dieses Lernmodul entwickelt wurde, vorgestellt und die didaktischen Theorien, die bei der Entwicklung des Lernmoduls relevant waren, erläutert.

2.1. CAMMP

CAMMP steht für „Computational and Mathematical Modeling Program“ (Computergestütztes Mathematisches Modellierungsprogramm) und ist ein außerschulisches Angebot des Karlsruher Institut für Technologie (KIT), sowie ein Schülerlabor der RWTH Aachen. Gegründet wurde CAMMP im Jahr 2011 von Prof. Dr. Ahmed Ismail, Dr. Nicole Faber und Prof. Dr. Martin Frank in Aachen. Mit dem Wechsel von Prof. Frank an das KIT wurde ab 2017 in Karlsruhe ein zweiter Standort aufgebaut.

Das Angebot von CAMMP richtet sich an Schüler verschiedener Altersgruppen, die reale Probleme aus Alltag, Industrie und Forschung untersuchen. Mit Hilfe mathematischer Modellierung und Computereinsatz sollen Problemlösekompetenzen spielerisch vermittelt und gefördert werden. Des Weiteren wird die Bedeutung von Mathematik und Simulationswissenschaften für die gesamte Gesellschaft verdeutlicht (vergleiche CAMMP, 2021a).

Für das Angebot von CAMMP gibt es zwei verschiedene Veranstaltungsformate. Zum Einen kann das Angebot im Rahmen eines CAMMP days wahrgenommen werden. Dieser kann am KIT, an der RWTH Aachen oder in einer Schule vor Ort oder auch online auf einer Plattform durchgeführt werden. Schüler aus der Mittel- und Oberstufe bearbeiten in Kleingruppen eine Problemstellung und erhalten Unterstützung von wissenschaftlichen Mitarbeitern (vergleiche CAMMP, 2021b). Die zweite Veranstaltungsform ist die CAMMP week, eine Woche in einer Jugendherberge zum Thema computergestützte mathematische Modellierung. Gruppen aus Schülern aus der Oberstufe forschen fünf Tage an einer individuellen Aufgabenstellung und präsentieren ihre Ergebnisse bei der Abschlussveranstaltung (vergleiche CAMMP, 2021c).

2.2. Mathematische Modellierung

Die mathematische Modellierung ist, wie im Namen zu erkennen, einer der zentralen Bestandteile der Lernmodule von CAMMP. Reale Fragestellungen weisen häufig einen sehr hohen Komplexitätsgrad auf und können deshalb nur schwer beantwortet werden. Die Mathematisierung dieser Fragestellungen ermöglicht es Vereinfachungen und Approximationen vorzunehmen und damit die Komplexität zu reduzieren.

2.2.1. Mathematisches Modell

Ein mathematisches Modell ist eine vereinfachte Darstellung eines Ausschnitts der Welt, das nach Greefrath (2018) die folgenden Eigenschaften erfüllt.

Isolation Ein Modell betrachtet den Ausschnitt der Welt isoliert.

Entsprechung Ein Modell ist eine vollständige und konsistente Menge von mathematischen Strukturen, die dem Ausschnitt der Welt entsprechen.

Vereinfachung Ein Modell berücksichtigt nur Teilaspekte der Realität und stellt diese vereinfacht dar.

Anwendung von Mathematik Auf das Modell können mathematische Methoden angewendet werden.

Aus diesen Eigenschaften folgt jedoch kein eindeutiges Modell, je nach Problemstellung muss die Realität zweckmäßig vereinfacht werden. Unterschiedliche Probleme erfordern deshalb häufig eine neue Modellierung, auch wenn der gleiche Gegenstand betrachtet wird.

2.2.2. Modellierungskreislauf

Der gesamte Prozess der mathematischen Modellierung wird häufig als Modellierungskreislauf veranschaulicht. Dies ist eine idealisierte Darstellung des Prozesses und somit selbst wieder ein Modell, entsprechend existieren verschiedene Ansätze mit unterschiedlichen Phasen der Modellierung. Die Ansätze unterscheiden sich hauptsächlich in der

Komplexität der Mathematisierung. Gemeinsam haben jedoch nahezu alle Ansätze die Unterteilung in die Dimensionen Realität und Mathematik.

Modellierungskreislauf nach Blum und Leiß

Eines der bekanntesten Modelle ist der Modellierungskreislauf nach Blum und Leiß (2005), siehe Abbildung 2.1, der im Rahmen des DISUM-Projekts entwickelt wurde. Das Modell wurde unter kognitiven Gesichtspunkten erstellt und kann dadurch genau beschreiben, wie Lernende mit Modellierungsaufgaben umgehen. Als erster Schritt steht deshalb das Situationsmodell, das die mentale Darstellung der Situation durch das Individuum beschreibt (vergleiche Hankeln, 2018; Ferri et al., 2013).

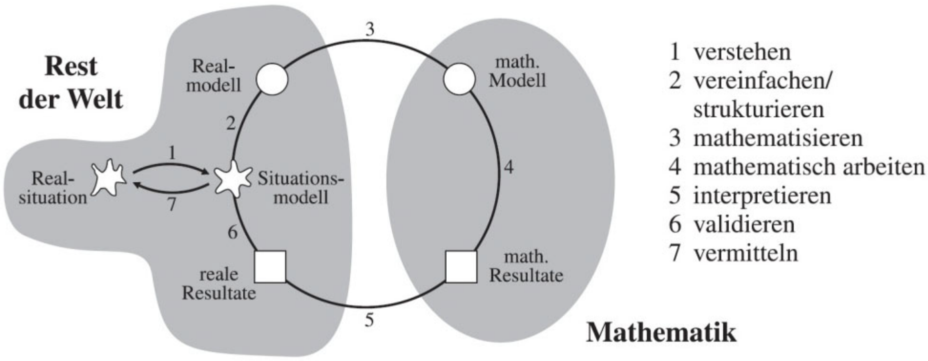


Abbildung 2.1.: Modellierungskreislauf nach Blum und Leiß
Quelle: Ferri et al. (2013)

Modellierungskreislauf von CAMMP

Der Modellierungskreislauf nach Blum und Leiß ist auf Grund der hohen Komplexität als Hilfestellung für die Schüler für den Prozess der Modellierung weniger geeignet. Von CAMMP wird deshalb ein an Blum (1985) angelehnter vierschrittiger Modellierungskreislauf, siehe Abbildung 2.2, verwendet.

Das reale Problem wird zunächst in der Realität vereinfacht und im zweiten Schritt mathematisch beschrieben, so dass es mit mathematischen Methoden gelöst werden kann. Die so berechnete mathematische Lösung wird im vierten Schritt interpretiert und validiert. Wenn das Ergebnis nicht plausibel oder zufriedenstellend ist, muss das Modell verbessert, z.B. durch andere Annahmen oder Vereinfachungen, und der Kreislauf erneut durchlaufen werden (vergleiche Ferri et al., 2013).

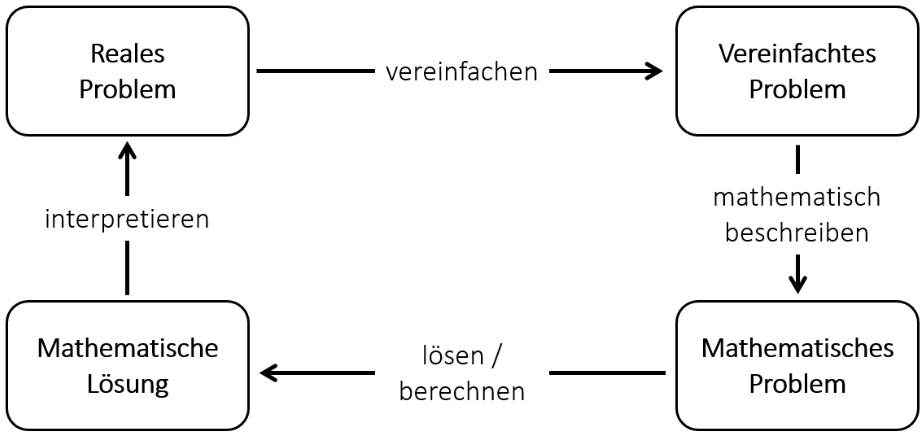


Abbildung 2.2.: Modellierungskreislauf von CAMMP

2.2.3. Mathematisches Modellieren mit digitalen Werkzeugen

Mit Hilfe digitaler Werkzeuge kann der Modellierungsprozess weiter verbessert werden. Das Potential der Werkzeuge kann dabei in verschiedenen Schritten des Modellierungskreislaufs unterstützend ausgenutzt werden. Der Modellierungskreislauf nach Kaiser, Blum, Ferri und Greefrath, siehe Abbildung 2.3, zeigt in Anlehnung an den Modellierungskreislauf nach Blum und Leiß wie die einzelnen Schritte durch digitale Werkzeuge erweitert werden können.

Insbesondere das Berechnen von numerischen oder algebraischen Ergebnissen kann mit digitalen Werkzeugen deutlich vereinfacht werden oder die Berechnung erst ermöglichen. Dies ist aber mit dem Risiko verbunden, dass die Schüler weniger Verständnis für den Sinn und die Funktion von Algorithmen ausbilden.
Eine weitere Einsatzmöglichkeit ist die Visualisierung von mathematischen Objekten für einen Darstellungswechsel. Hierbei kann die Visualisierung sowohl im Modellierungs- als auch im Problemlöseprozess sinnvoll eingesetzt werden. Das direkte Feedback digitaler Werkzeuge ermöglicht damit zudem erkundendes und experimentelles Arbeiten, z.B. mit Simulationen. Einerseits können hierdurch kognitive Hürden überwunden werden, andererseits kann es auch zur Beschleunigung des Arbeitens führen und somit besteht die Gefahr, dass keine tiefergehende Reflexion stattfindet und das mathematische Arbeiten nur auf Versuch und Irrtum basiert (vergleiche Hankeln, 2018).

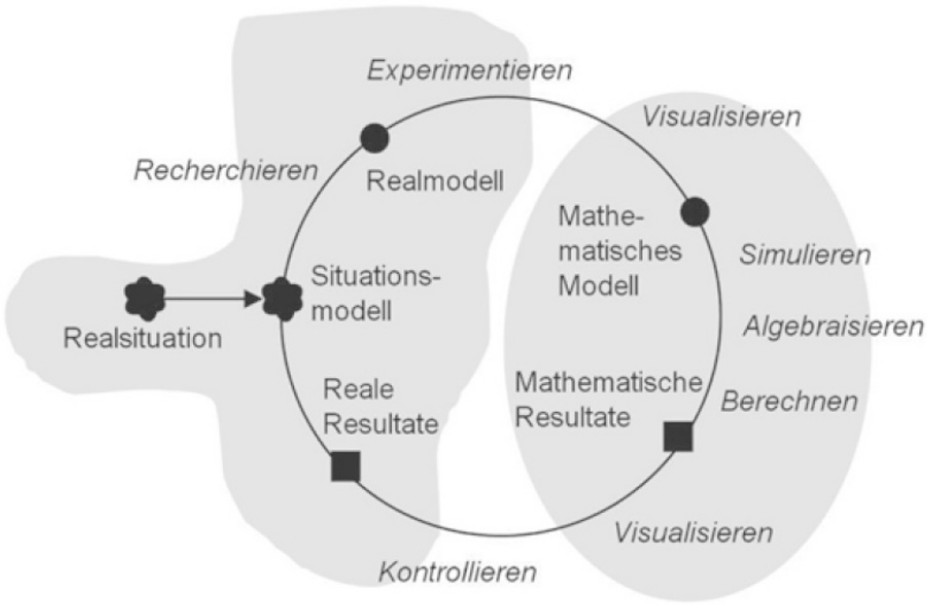


Abbildung 2.3.: Modellierungskreislauf nach Kaiser, Blum, Ferri und Greefrath
Quelle: Hankeln (2018)

Im Rahmen von CAMMP wird der computerunterstützte Modellierungskreislauf als Modellierungsspirale, siehe Abbildung 2.4, veranschaulicht, die an die Solution Helix of Math der Initiative Computer-Based Math angelehnt ist (vergleiche Frank et al., 2018). Die Modellierungsspirale erweitert den Modellierungskreislauf um die Dimension des Lösungsfortschritts und zeigt die Annäherung an eine optimale Lösung.

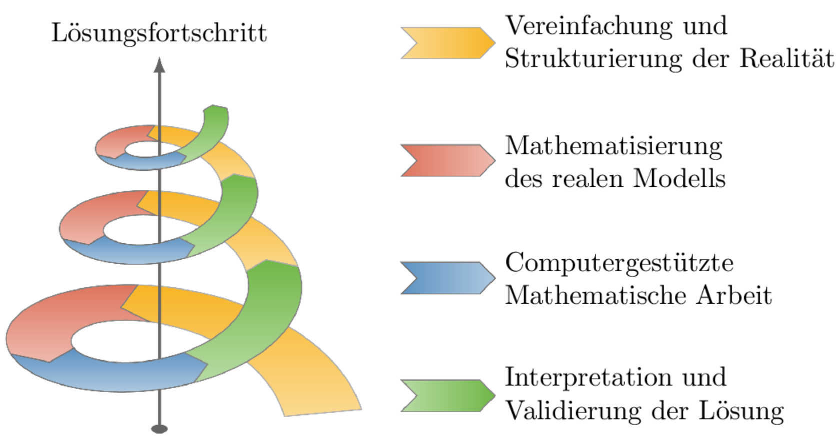


Abbildung 2.4.: Modellierungsspirale von CAMMP
Quelle: Frank et al. (2018)

2.2.4. Modellierungskompetenz

In der Vereinbarung der Kultusministerkonferenz über die „Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife“ ist die mathematische Modellierung eine der sechs allgemeinen mathematischen Kompetenzen, die das Spektrum des mathematischen Arbeitens erfassen (vergleiche Kultusministerkonferenz, 2012). Dementsprechend ist die mathematische Modellierung auch im Bildungsplan des Gymnasiums für das Fach Mathematik als eine der fünf prozessbezogenen Kompetenzen verankert (vergleiche Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016).

> „Hier geht es um den Wechsel zwischen Realsituationen und mathematischen Begriffen, Resultaten oder Methoden. Hierzu gehört sowohl das Konstruieren passender mathematischer Modelle als auch das Verstehen oder Bewerten vorgegebener Modelle. Typische Teilschritte des Modellierens sind das Strukturieren und Vereinfachen gegebener Realsituationen, das Übersetzen realer Gegebenheiten in mathematische Modelle, das Interpretieren mathematischer Ergebnisse in Bezug auf Realsituationen und das Überprüfen von Ergebnissen im Hinblick auf Stimmigkeit und Angemessenheit bezogen auf die Realsituation. Das Spektrum reicht von Standardmodellen (z. B. bei linearen Zusammenhängen) bis zu komplexen Modellierungen.“
> - (Kultusministerkonferenz, 2012, S. 15)

Die hier genannten Teilschritte bzw. Teilkompetenzen der Modellierung werden jeweils vom Modellierungskreislauf von CAMMP abgedeckt, was die Eignung des Kreislaufs als Orientierungshilfe für die Schüler nochmals verdeutlicht. Die Aufteilung der Modellierung in Teilkompetenzen reduziert zudem die Komplexität der Problemstellung, was für den Aufbau von Modellierungskompetenz förderlich ist (vergleiche Hankeln, 2018; Ferri et al., 2013).

2.3. Prinzip der minimalen Hilfe

Das Prinzip der minimalen Hilfe wurde bereits in den 60er Jahren vom Didaktiker Hans Aebli eingeführt und in den 70er Jahren vom Mathematikdidaktiker Friedrich Zech weiterentwickelt. Das Ziel des Prinzips ist es, den Schülern nur so viel Hilfe zu geben, wie sie von sich aus einfordern. Die Schüler sollen in ihrem Arbeitsprozess also nicht gestört werden und erst Hilfe erhalten, wenn sie an einem Punkt nicht mehr weiterkommen. Das

Ausmaß der Hilfestellungen soll dabei über eine fünfstufige Skala gesteuert werden, so dass zunächst nur kleine Hilfen gegeben werden und die Schüler nicht direkt zur Lösung geführt werden (vergleiche Zech, 2002). Die fünf Stufen der Hilfestellungen werden im Folgenden mit Beispielen des Deutsches Zentrum für Lehrerbildung kurz vorgestellt.

Ebene 1: Motivationshilfe
Die Schüler werden mit positiven Zusprüchen motiviert weiter an der Aufgabe zu arbeiten. Motivationshilfen sind z.B. „Du schaffst das!“ oder „So eine ähnliche Aufgabe hast du schon einmal gelöst.“

Ebene 2: Rückmeldehilfe
Die Schüler erhalten Feedback über ihre bisher geleistete Arbeit. Rückmeldehilfen sind z.B. „Du bist auf dem richtigen Weg.“ oder „In der Berechnung ist noch ein Fehler.“

Ebene 3: Allgemeine strategische Hilfe
Die Schüler erhalten Hinweise zu allgemeinen Lösungsstrategien. Solche Hinweise sind z.B. „Lies dir die Aufgabenstellung nochmal durch.“ oder „Probiere einen anderen Weg.“

Ebene 4: Inhaltsorientierte strategische Hilfe
Die Schüler erhalten eine inhaltliche Rückmeldung und Vorschläge für die Vorgehensweise. Solche Rückmeldungen sind z.B. „Versuche das Problem graphisch zu lösen.“ oder „Vielleicht hilft es dir, wenn du die Zahlen sortierst.“

Ebene 5: Inhaltliche Hilfe
Die Schüler erhalten konkrete inhaltliche Rückmeldungen und Vorschläge für weitere Schritte. Solche Rückmeldungen sind z.B. „Wie ist die Formel für ... definiert?“ oder „Welche Regelmäßigkeit kannst du bei den sortierten Zahlen entdecken?“

2.4. Darstellungsebenen nach Bruner

Nach dem Psychologen Jerome Bruner verläuft die Denkentwicklung auf drei verschiedenen Darstellungsebenen, die sich gegenseitig ergänzen. Insbesondere der Wechsel zwischen den Ebenen ist für den Lernprozess entscheidend (vergleiche Sill, 2019).

Die enaktive Ebene beschreibt das Handeln am konkreten Objekt. Die Schüler erfassen die Umwelt durch konkrete oder vorgestellte Handlungen. Dadurch kann auch die Arbeit am Computer mit interaktiven Programmen als virtuell-enaktiv bezeichnet werden. Am Beispiel der Addition könnte diese Ebene durch das Zusammenfügen von zwei Mengen, z.B. Äpfeln, dargestellt werden.
Die ikonische Ebene beschreibt die bildliche Darstellung der Sachverhalte. Die Schüler erfassen die Umwelt in konkreten oder vorgestellten Bildern. Die Addition kann so auch zeichnerisch dargestellt und gelöst werden.
Die symbolische Ebene beschreibt das Arbeiten auf abstrakter Ebene. Die Schüler erfassen die Umwelt mit Sprachen oder Symbolen, für die es anwendbare Regeln gibt. Für die Addition bedeutet dies z.B. die Notation $6 + 3 = 9$.

3. Mathematischer und theoretischer Hintergrund

In diesem Kapitel werden die mathematischen und fachlichen Hintergründe der automatisierten Komposition mit Markov Ketten beschrieben. Die Inhalte gehen über das für die Dozenten erstellte Basic Paper, siehe Anhang C, hinaus.

3.1. Algorithmische Kompositionsverfahren

Bereits seit mehreren hundert Jahren beschäftigen sich Musiker und Mathematiker mit Verfahren, die die Komposition musikalischer Stücke vereinfachen oder komplett übernehmen können. Wenn diese Verfahren automatisch ablaufen und durch einen mathematischen Prozess beschrieben werden können, dann spricht man von algorithmischer Komposition. Für die Umsetzung existieren unterschiedliche Methoden, die im Folgenden kurz vorgestellt werden. Die algorithmische Komposition mit Markov Ketten wird in Abschnitt 3.2.2 ausführlich thematisiert.

3.1.1. Geschichtlicher Hintergrund

Die grundsätzlichen Methoden zur Entwicklung algorithmischer Kompositionsverfahren sind heute noch die gleichen wie früher. Durch Analysieren von Notenverläufen, harmonischen Strukturen und rythmischen Mustern kann man die Strukturen von Musik mathematisch erfassen. Die folgende Auflistung zeigt einen groben Überblick über den historischen Verlauf der entwickelten Verfahren (vergleiche Roads, 1996; Nierhaus, 2009).

1026: Der Benediktinermönch und Musiktheoretiker Guido von Arezzo entwickelt ein Verfahren zur automatischen Generierung von Melodien aus Textmaterial. Hierbei werden Buchstaben und Silben auf Tonhöhen und melodische Phrasen abgebildet, siehe Abbildung 3.1. Guido von Arezzo gilt deshalb auch als Vater der Solmisation.

1650: Der Jesuit und Universalgelehrte Athanasius Kircher entwickelt mit der Komponiermaschine ein mechanisches System zur automatischen Komposition. Die Komponiermaschine besteht aus Täfelchen mit vierzeiligen Zahlenspalten, wobei die Zahlen jeweils Tönen zugeordnet sind. Die Benutzer entnehmen dem Versmaß des Textes entsprechende Tafeln und fügen willkürlich gewählte Segmente aneinander. Die Repräsentation von Tonhöhen durch Zahlen war ein deutlicher Fortschritt für die Klasse der möglichen Kompositionen. Des Weiteren zeigt das von Kircher angewendete Prinzip eine Anwendung von Tonklassen, die seit dem 20. Jahrhundert eine wesentliche Rolle in der Musikproduktion und -analyse spielen.

18. Jhdt: Zum Ende des 18. Jahrhunderts entwickeln verschiedene Komponisten und Musiktheoretiker musikalische Würfelspiele. In einer Tabelle stehen vorkomponierte Takte, durch Würfeln oder Kartenziehen wird eine Zufallszahl erzeugt, die eine Zeile aus einer Tabelle bestimmt. Die Spalten der Tabelle werden nach der Reihenfolge der Würfe ausgewählt. Mit dieser Methode wurden meistens Walzer, Polonaisen oder Menuetten komponiert. Das bekannteste Spiel stammt von Wolfgang Amadeus Mozart mit seiner „Anleitung zum Componieren von Walzern vermittels zweier Würfel", siehe Abbildung 3.2.

1957: Die amerikanischen Komponisten Lejaren Hiller und Leonard Issacson programmierten den ILLIAC I Computer, der die Illiac Suite komponierte, das erste vollständig von einem Computer geschriebene Stück.

1960: Der russische Wissenschaftler R. Kh . Zaripov veröffentlicht die erste Arbeit über algorithmische Komposition mithilfe eines Computers.

1980: Der amerikanische Wissenschaftler und Komponist David Cope entwickelt die Software EMI (Experiments in Musical Intelligence). Das System analysiert existierende Musikstücke und generiert darauf basierende neue Musikstücke.

Abbildung 3.1.: Zuordnung von Vokalen auf Tonhöhen nach Guido von Arezzo
Quelle: Nierhaus (2009)

WOLFGANG AMADEUS MOZART

Musikalisches Würfelspiel

Table of Measure Numbers

Part One

	I	II	III	IV	V	VI	VII	VIII
2	96	22	141	41	105	122	11	30
3	32	6	128	63	146	46	134	81
4	69	95	158	13	153	55	110	24
5	40	17	113	85	161	2	159	100
6	148	74	163	45	80	97	36	107
7	104	157	27	167	154	68	118	91
8	152	60	171	53	99	133	21	127
9	119	84	114	50	140	86	169	94
10	98	142	42	156	75	129	62	123
11	3	87	165	61	135	47	147	33
12	54	130	10	103	28	37	106	5

Part Two

	I	II	III	IV	V	VI	VII	VIII
2	70	121	26	9	112	49	109	14
3	117	39	126	56	174	18	116	83
4	66	139	15	132	73	58	145	79
5	90	176	7	34	67	160	52	170
6	25	143	64	125	76	136	1	93
7	138	71	150	29	101	162	23	151
8	16	155	57	175	43	168	89	172
9	120	88	48	166	51	115	72	111
10	65	77	19	82	137	38	149	8
11	102	4	31	164	144	59	173	78
12	35	20	108	92	12	124	44	131

Abbildung 3.2.: Musikalisches Würfelspiel von Wolfgang Amadeus Mozart
Quelle: Roads (1996)

3.2. Algorithmische Komposition mit Markov Ketten

3.2.1. Markov Ketten

Markov Ketten, nach Andrei A. Markov (1856–1922), stellen einen stochastischen Prozess dar, der dadurch charakterisiert ist, dass das zukünftige Verhalten des Prozesses nur vom aktuellen Zustand abhängt und nicht von der Vergangenheit. Dies ermöglicht die Modellierung dynamischer Systeme in verschiedenen Gebieten, wie z.B. in der Produktionsplanung, beim Qualitätsmanagement oder in der Informationsverarbeitung. Die folgende Beschreibung basiert auf Aldous & Fill (2014), Norris (1997) und Kersting & Wakolbinger (2011).

Definition 3.1. *Eine Folge $X = \{X_k\}_{k \in \mathbb{N}}$ von Zufallsvariablen $X_k : \Omega \to S$ heißt* stochastischer Prozess *im Zustandsraum S.*

Die Zufallsvariable X_k beschreibt den Zustand des Systems zum Zeitpunkt k, dies kann z.B. ein Lagerbestand oder die Größe einer Population sein.

Definition 3.2. *Ein stochastischer Prozess $X = \{X_k\}_{k \in \mathbb{N}}$ in einem endlichen oder abzählbar unendlichen Zustandsraum S heißt* Markov Kette erster Ordnung*, wenn die Markovbedingung*

$$\mathbb{P}(X_{k+1} = i_{k+1} | X_0 = i_0, ..., X_k = i_k) = \mathbb{P}(X_{k+1} = i_{k+1} | X_k = i_k) = p_{i_k i_{k+1}}$$

für alle $k \in \mathbb{N}$ und für alle Zustände $i_k \in S$ gilt. Die p_{ij} heißen Übergangswahrscheinlichkeiten *von Zustand i auf Zustand j.*

Die Wahrscheinlichkeitsverteilung des Zustands zur Zeit $k+1$ ist also nur vom Zustand zur Zeit k abhängig, jedoch nicht von den vorherigen Zuständen $k' < k$.
Die Übergangswahrscheinlichkeiten werden üblicherweise in einer stochastischen Matrix, der Übergangsmatrix, dargestellt.

Definition 3.3. *Eine Matrix $P = (p_{ij})_{i,j \in S}$ heißt* stochastische Matrix*, falls $p_{ij} \geq 0$ ist und für alle $i \in S$ die Zeilensumme $\sum_{j \in S} p_{ij} = 1$ ist.*

Definition 3.4. *Sei (X_k) eine Markov Kette mit Übergangsmatrix $P = (p_{ij})$.*

a) Ein Zustand $j \in S$ ist von einem Zustand $i \in S$ erreichbar *(kurz: $i \rightsquigarrow j$), falls es ein $n \in \mathbb{N}$ gibt mit $p_{ij}^n > 0$. Der Zustand j kann also in n Schritten vom Zustand i erreicht werden.*

b) Ein Zustand $i \in S$ kommuniziert *mit einem Zustand $j \in S$, falls sowohl $i \rightsquigarrow j$ als auch $j \rightsquigarrow i$ gilt.*

c) Die Markov Kette bzw. die zugehörige Übergangsmatrix heißen irreduzibel, *falls alle Zustände miteinander kommunizieren.*

Analog zur Markov Kette erster Ordnung können auch Markov Ketten höherer Ordnung betrachtet werden.

Definition 3.5. *Ein stochastischer Prozess $X = \{X_k\}_{k\in\mathbb{N}}$ in einem endlichen oder abzählbar unendlichen Zustandsraum S heißt* Markov Kette n-ter Ordnung, *wenn die Markovbedingung*

$$\mathbb{P}(X_{k+1} = i_{k+1}|X_0 = i_0, ..., X_k = i_k) = \mathbb{P}(X_{k+1} = i_{k+1}|X_{k-n+1} = i_{k-n+1}, ..., X_k = i_k)$$

für alle $k \in \mathbb{N}$ und für alle Zustände $i_k \in S$ gilt.

Die Wahrscheinlichkeitsverteilung des Zustands zur Zeit $k+1$ ist also von den n vorherigen Zuständen abhängig.
In den folgenden Beispielen werden nur Markov Ketten erster Ordnung berücksichtigt.

Beispiel 3.6. Wettervorhersage
Die Wettervorhersage lässt sich in einem vereinfachten Modell als Markov Kette beschreiben. Das Wetter kann hier die Zustände $S = \{Sonne, Wolken, Regen\}$ annehmen. Die Wahrscheinlichkeiten für Wetteränderungen sind durch die stochastische Matrix P gegeben.

$$P = \begin{array}{c} \\ Sonne \\ Wolken \\ Regen \end{array} \begin{array}{c} \begin{array}{ccc} Sonne & Wolken & Regen \end{array} \\ \begin{pmatrix} 0,5 & 0,25 & 0,25 \\ 0,5 & 0,25 & 0,25 \\ 0 & 0,25 & 0,75 \end{pmatrix} \end{array}$$

Markov Ketten mit wenigen Zuständen können auch wie in Abbildung 3.3 übersichtlich als Graph dargestellt werden.

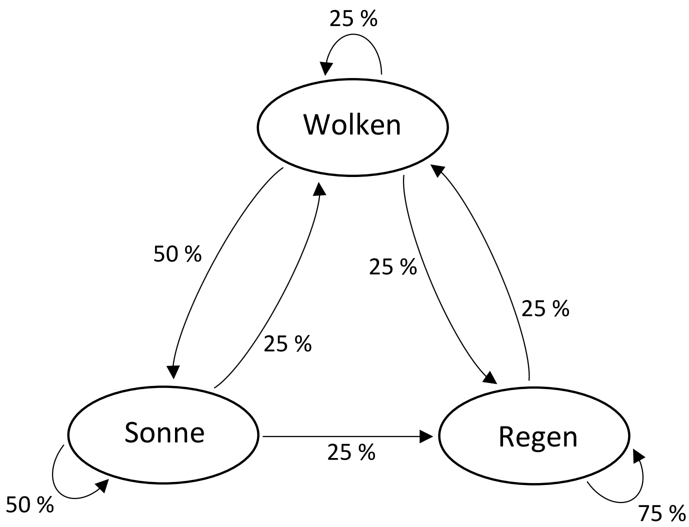


Abbildung 3.3.: Übergangsgraph der Wettervorhersage

Beispiel 3.7. Random Walk
Ein Random Walk, oder auf deutsch zufällige Irrfahrt, ist ein mathematisches Modell für eine Verkettung zufälliger Bewegungen. Wir betrachten im Folgenden den eindimensionalen Fall, bei dem sich ein Teilchen mit gleicher Wahrscheinlichkeit nach vorne oder hinten bewegen kann, siehe Abbildung 3.4.

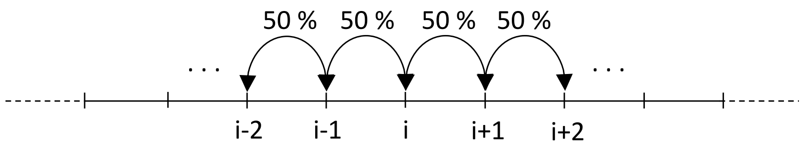


Abbildung 3.4.: Übergangsgraph des Random Walks

Hieraus ergibt sich die Übergangsmatrix P.

$$P = \begin{array}{cc} & \begin{array}{ccccccc} \cdots & i-2 & i-1 & i & i+1 & i+2 & \dots \end{array} \\ \begin{array}{c} \vdots \\ i-2 \\ i-1 \\ i \\ i+1 \\ i+2 \\ \vdots \end{array} & \begin{pmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0,5 & 0 & 0 & 0 & \cdots \\ \cdots & 0,5 & 0 & 0,5 & 0 & 0 & \cdots \\ \cdots & 0 & 0,5 & 0 & 0,5 & 0 & \cdots \\ \cdots & 0 & 0 & 0,5 & 0 & 0,5 & \cdots \\ \cdots & 0 & 0 & 0 & 0,5 & 0 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \end{array}$$

Beispiele für hieraus entstehende Bewegungen sind in Abbildung 3.5 zu sehen. Hier starten zehn Teilchen zur gleichen Zeit im Zustand 0 und führen dann unabhängig voneinander 200 Bewegungen aus.

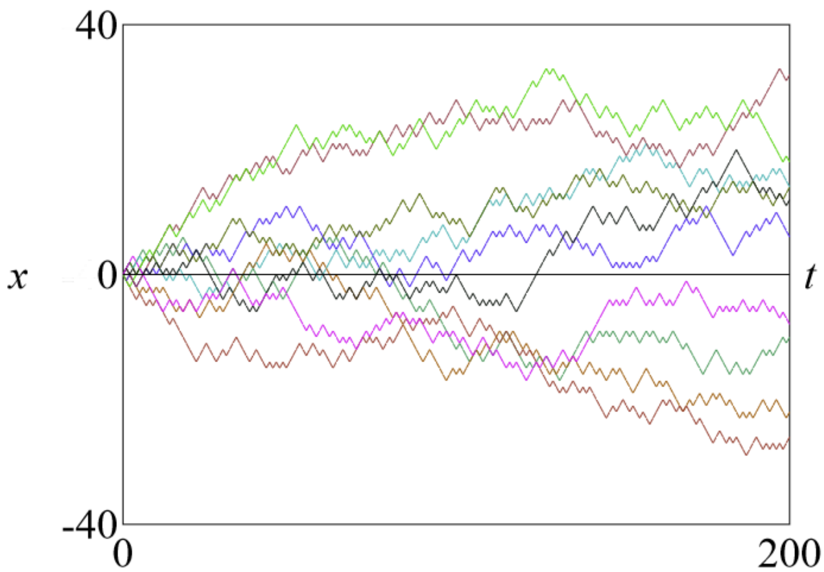


Abbildung 3.5.: Beispiele für den Verlauf des Random Walks

3.2.2. Musikkomposition mit Markov Ketten

Markov Ketten für die algorithmische Komposition zu verwenden ist eine sehr verbreitete und gut geeignete Methode. Die Noten definieren den Zustandsraum S und die Übergangsmatrix kann entsprechend der Notenübergänge aus einer Beispielsequenz erzeugt werden.

Schritt 1: Abzählen der Notenübergänge

Die Übergangswahrscheinlichkeiten werden mit der Maximum-Likelihood-Methode berechnet. Hierfür wird im ersten Schritt die Häufigkeit h_{ij}, also das Auftreten von Zustand $j \in S$ zum Zeitpunkt k, wenn Zustand $i \in S$ zum Zeitpunkt $(k-1)$ aufgetreten ist, für alle $i, j \in S$ und alle k aus der Beispielsequenz, abgezählt. Die so berechneten Häufigkeiten h_{ij} definieren die Matrix der absoluten Häufigkeiten H.

Beispiel 3.8. *Für die Beispielsequenz in Abbildung 3.6 ergibt sich die Matrix H.*

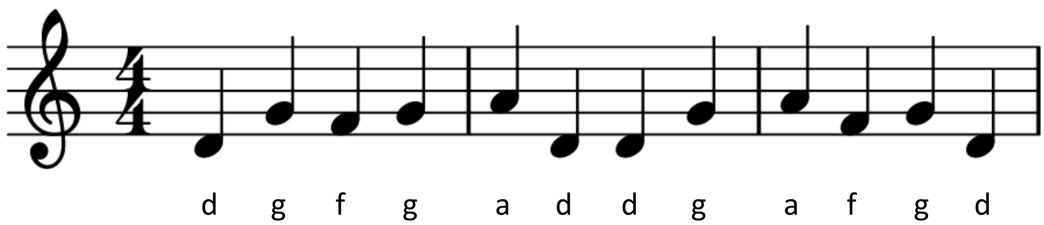


Abbildung 3.6.: Beispielsequenz

$$H = \begin{array}{c c} & \begin{array}{cccc} a & d & f & g \end{array} \\ \begin{array}{c} a \\ d \\ f \\ g \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 2 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Schritt 2: Berechnung der Übergangsmatrix

Aus dieser Matrix kann im zweiten Schritt die Übergangsmatrix $P = (p_{ij})$ berechnet werden. Für die einzelnen Einträge gilt:

$$p_{ij} = \frac{h_{ij}}{\sum\limits_{k \in S} h_{ik}} \tag{3.1}$$

Beispiel 3.9. *Mit Hilfe der Formel 3.1 ergibt sich aus der Matrix H die Übergangsmatrix P.*

$$P = \begin{array}{c} \\ a \\ d \\ f \\ g \end{array} \begin{array}{c} \begin{array}{cccc} a & d & f & g \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 1/3 & 0 & 2/3 \\ 0 & 0 & 0 & 1 \\ 1/2 & 1/4 & 1/4 & 0 \end{pmatrix} \end{array}$$

Schritt 3: Erzeugung einer neuen Sequenz

Für die Erzeugung einer neuen Sequenz kann eine beliebige Note aus dem Zustandsraum S als Anfangsnote ausgewählt werden. Die zweite Note wird über die durch die erste Note gegebene Wahrscheinlichkeitsverteilung bestimmt. Die dritte Note wird dementsprechend über die durch die zweite Note gegebene Wahrscheinlichkeitsverteilung bestimmt. Auf diese Weise entsteht eine neue Sequenz mit beliebiger Länge.

Beispiel 3.10. *Als Anfangszustand wird die Note „a“ ausgewählt, entsprechend ist die Wahrscheinlichkeitsverteilung $(0, 1/2, 1/2, 0)$ für die zweite Note (a, d, f, g) gegeben. Ein von der Verteilung abhängiger Zufallswert bestimmt die Note „f“ als zweiten Zustand. Entsprechend der Übergangsmatrix kann der dritte Zustand nur die Note „g“ sein. Auf diese Weise ist die Sequenz in Abbildung 3.7 entstanden.*

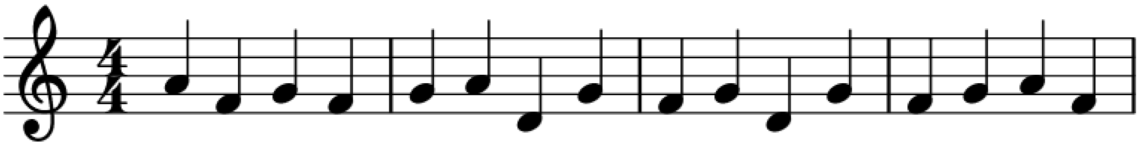


Abbildung 3.7.: Beispiel für eine neu erzeugte Sequenz

3.2.3. Computergestützte Komposition mit Markov Ketten

Für größere Beispielsequenzen als in Beispiel 3.8 ist ein computergestütztes Einlesen der Trainingsdaten sinnvoll. Die Trainingsdaten werden hier als MIDI Dateien eingelesen und mit JuliaMusic analysiert. Die Berechnung der Markov Kette erfolgt nun analog zu den in 3.2.2 beschriebenen Schritten.

MIDI Technologie

MIDI steht für Musical Instrument Digital Interface, oder auf deutsch „Digitale Schnittstelle für Musikinstrumente“, und wurde bereits in den frühen 80er Jahren entwickelt und wird auch heute noch weiterentwickelt. Es ist das in der Musikindustrie übliche Technologieprotokoll, das die Kommunikation zwischen Computern, Instrumenten und anderer Hardware ermöglicht. MIDI Dateien übertragen dabei keine eigentlichen Audiosignale, sondern nur Daten in Form von Anweisungen, wie eine Maschine die MIDI-Note wiedergeben soll (vergleiche Stotz, 2019). Jede MIDI-Note enthält Anweisungen mit den folgenden Informationen:

- *Note-on / Note-off*: Von wann bis wann die Note gespielt wird
- *Note*: Welche Note gespielt wird
- *Velocity*: Anschlagstärke

JuliaMusic

JuliaMusic ist eine Sammlung von Paketen, die in der Programmiersprache Julia geschrieben sind und für das Lesen, Manipulieren und Speichern von Musikdaten verwendet werden können. Für die hier benötigten Funktionen des Lesens und Schreibens der

MIDI Dateien wird insbesondere das Paket MIDI.jl verwendet. Dieses Paket ermöglicht es, die Rohdaten der MIDI Dateien intuitiv und für Menschen lesbar darzustellen (vergleiche Datseris & Hobson, 2019). Abbildung 3.8 zeigt die Darstellung einer eingelesenen MIDI Datei mit MIDI.jl.

```
291 Notes with tpq=480
 Note B3  | vel = 122 | pos = 39840, dur = 195 | channel 3
 Note E4  | vel = 122 | pos = 40080, dur = 216 | channel 3
 Note F♯4 | vel = 122 | pos = 40320, dur = 413 | channel 3
 Note B4  | vel = 122 | pos = 40800, dur = 425 | channel 3
 Note B4  | vel = 122 | pos = 41280, dur = 429 | channel 3
 Note B4  | vel = 122 | pos = 41760, dur = 727 | channel 3
 Note B3  | vel = 122 | pos = 43919, dur = 216 | channel 3
   ⋮
 Note E4  | vel = 122 | pos = 313440, dur = 240 | channel 3
 Note B3  | vel = 122 | pos = 313680, dur = 437 | channel 3
 Note B3  | vel = 122 | pos = 314160, dur = 240 | channel 3
 Note B3  | vel = 122 | pos = 314760, dur = 107 | channel 3
 Note E4  | vel = 122 | pos = 314880, dur = 219 | channel 3
 Note F♯4 | vel = 122 | pos = 315120, dur = 108 | channel 3
 Note F♯4 | vel = 122 | pos = 315240, dur = 2519 | channel 3
```

Abbildung 3.8.: Mit MIDI.jl eingelesene MIDI Datei

3.3. Bedingte Wahrscheinlichkeit

Betrachtet man einen einzelnen Übergang einer Markov Kette vom Zustand i in den Zustand j, dann ist die Übergangswahrscheinlichkeit p_{ij} die bedingte Wahrscheinlichkeit sich zum Zeitpunkt $(k+1)$ in Zustand j zu befinden, gegeben, dass man zum Zeitpunkt k in Zustand i ist.

Definition 3.11. *Es seien* $(\Omega, \mathbb{P})$ *ein endlicher Wahrscheinlichkeitsraum und* $A \subset \Omega$ *mit* $\mathbb{P}(A) > 0$*. Dann heißt*

$$\mathbb{P}_A(B) := \mathbb{P}(B|A) := \frac{\mathbb{P}(\mathbb{B} \cap \mathbb{A})}{\mathbb{P}(A)}, \quad B \subset \Omega$$

die bedingte Wahrscheinlichkeit von B unter der Bedingung A. *(nach Henze, 2016)*

4. Didaktisch-methodisches Konzept

In diesem Kapitel wird das für das Lernmodul zugrunde liegende didaktisch-methodische Konzept vorgestellt. Dies umfasst einen Überblick über die Materialien und den Ablauf des Lernmoduls.

4.1. Ziele des Lernmoduls

Das entwickelte Lernmodul fördert verschiedene curriculare sowie außercurriculare Kompetenzen der Schüler. Das primäre Ziel ist das Entwickeln eines Grundverständnisses für empirische Vorhersagemodelle am Beispiel der Musikkomposition. Die Schüler lernen den Zusammenhang zwischen Mathematik und Musik kennen und können somit die Mathematik in der alltäglichen Lebenswelt anwenden. Hierdurch soll den Schülern die Bedeutung der Mathematik für die gesamte Gesellschaft verdeutlicht und das Interesse an Mathematik gefördert werden.

Das Vorhersagemodell wird auf der Basis von Markov Ketten in einer didaktisch reduzierten Version von den Schülern schrittweise selbst entwickelt. Die Schüler greifen hierzu auf größtenteils bekannte inhaltliche Kompetenzen aus dem Bereich der Stochastik zurück und können diese in einem außerschulischen Kontext anwenden und damit wiederholen und vertiefen.

Des Weiteren verfolgt das Lernmodul das Ziel die prozessbezogene Kompetenz des Modellierens zu fördern. Hierbei soll insbesondere der Modellierungskreislauf von CAMMP, siehe Abschnitt 2.2.2, helfen, der von den Schülern größtenteils eigenständig durchlaufen und in den Plenumsdiskussionen besprochen wird. Durch das schrittweise durchlaufen des Modellierungskreislaufs wird die Modellierungskompetenz besonders gefördert, vergleiche Abschnitt 2.2.4.

Durch die Automatisierung der Musikkomposition im Laufe des Lernmoduls lernen die Schüler zusätzlich auch etwas über Maschinelles Lernen, insbesondere wie das zu Grunde liegende statistische Modell aufgebaut wird und anhand dieses Modells Entscheidungen getroffen werden.

Ferner lernen die Schüler auch die Bedeutung digitaler Werkzeuge für die Modellierung kennen, die die Automatisierung der Musikkomposition erst ermöglichen.

4.2. Didaktische Reduktion

Markov Ketten, wie sie in Abschnitt 3.2.1 definiert wurden, sind für Schüler zu komplex und werden deshalb nur indirekt thematisiert, indem die dahinter steckenden mathematischen Schritte nacheinander ausgeführt werden. Die Markov Ketten reduzieren sich dadurch auf die bereits bekannten Themen absolute und relative Häufigkeiten, sowie bedingte Wahrscheinlichkeiten. Auf den Begriff Markov Ketten kann dadurch komplett verzichtet werden.

Da die Behandlung von Matrizen kein Bestandteil des Bildungsplans des Landes Baden-Württemberg ist, sind diese vermutlich für die meisten Schüler noch unbekannt. Die benötigten Matrizen werden deshalb im Lernmodul als Tabellen bezeichnet. Da die Matrizen für die Schüler nur zum Speichern der absoluten und relativen Häufigkeiten verwendet werden, kann auf die Einführung des Begriffs problemlos verzichtet werden.

4.3. Notwendige mathematische Voraussetzungen und curriculare Einordnung

Für das entwickelte Lernmodul spielt die Stochastik eine große Rolle. Zum Bearbeiten der Aufgaben benötigen die Schüler deshalb die folgenden inhaltsbezogenen Kompetenzen:

- absolute und relative Häufigkeit bestimmen
- bedingte Wahrscheinlichkeit erklären

- Wahrscheinlichkeitsverteilung einer Zufallsgröße angeben und im Sachverhalt interpretieren

Alle benötigten inhaltlichen Kompetenzen sind im Bildungsplan jeweils unter der Leitidee Daten und Zufall zu finden. Nach dem Bildungsplan werden die absolute und relative Häufigkeit bereits in der 7. und 8. Klasse behandelt, die bedingte Wahrscheinlichkeit und Wahrscheinlichkeitsverteilungen erst in der 9. und 10. Klasse (vergleiche Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016). In der Praxis werden die Themen aber meistens in der 9. Klasse behandelt, weshalb das Lernmodul für Schüler ab der 10. Klasse geeignet ist.

Neben den zuvor genannten inhaltlichen Kompetenzen werden auch die prozessbezogenen Kompetenzen „Probleme lösen“, „Modellieren“ sowie „mit symbolischen, formalen und technischen Elementen der Mathematik umgehen“ benötigt und gefördert.

4.4. Julia und Jupyter Notebook

Die digitalen Arbeitsblätter des Lernmoduls wurden als Jupyter Notebook Dokumente erstellt, wobei für die Programmierung die höhere Programmiersprache Julia verwendet wurde. Julia wurde hauptsächlich für numerisches und wissenschaftliches Rechnen entwickelt und zeichnet sich durch ihre schnelle Ausführungsgeschwindigkeit aus. Für dieses Lernmodul ist insbesondere die effiziente Implementierung zur Generation von Zufallszahlen von Bedeutung (vergleiche Wikipedia, 2021a). Auf Grund der leicht verständlichen Syntax eignet sich Julia sehr gut für erste Erfahrungen im Programmieren, wodurch sich auch Schüler ohne Programmierkenntnisse schnell einarbeiten können.

Die Jupyter Notebook Dokumente bestehen aus Eingabe- und Ausgabezeilen, die Code, Text und Plots enthalten können (vergleiche Wikipedia, 2021b). Die Dokumente sind so gestaltet, dass die Aufgaben in den Codefeldern beantwortet werden. Die Codefelder sind als Art Lückentext gestaltet, so dass die Schüler nur einzelne Wörter, Zahlen oder Formeln zum vorgegebenen Code hinzufügen müssen. Die Lösung in Codefeldern ermöglicht neben der Ausgabe von Graphen, Text oder Tabellen zudem ein Rückmeldesystem nach dem Prinzip der minimalen Hilfe, siehe Abschnitt 2.3. Die Schüler erfahren somit sofort, ob ihre Lösung richtig ist und erhalten für typische Fehler Hilfestellungen als Rückmeldung.

Die Jupyter Notebooks liegen auf Servern des KIT, worauf die Schüler über den Multi-User-Server JupyterHub zugreifen können. Nach dem Download des entsprechenden Lernmoduls können die Schüler die Arbeitsblätter direkt im Browser bearbeiten, weitere Programme werden nicht benötigt. Diese Art der Bereitstellung der Arbeitsblätter ermöglicht eine ortsunabhängige Durchführung des Lernmoduls.

4.5. Ablauf des Lernmoduls

Das Lernmodul ist auf eine Dauer von 120 Minuten ausgelegt und besteht aus zwei Arbeitsphasen sowie drei Plenumsdiskussionen. Der zeitliche Ablauf wird nachfolgend dargestellt.

1. **Begrüßung und Einstiegspräsentation** *(20 Minuten)*
   Nach einer kurzen Begrüßung und Vorstellungsrunde wird die Problemstellung mit einer Präsentation eingeführt. Zunächst wird auf die Geschichte der algorithmischen Komposition eingegangen und dann der heutige Stand gezeigt. Anhand eines Beispiels wird ein computerkomponiertes Stück einem Stück von Bach direkt gegenüber gestellt, wobei die Schüler das Stück von Bach erraten sollen. Im Anschluss folgt eine Einführung in die mathematische Modellierung mit Hilfe des Modellierungskreislaufs.

2. **Erste Arbeitsphase** *(40 Minuten)*
   Die Schüler bilden Kleingruppen und bearbeiten das erste Arbeitsblatt. Anhand einer kurzen Beispielsequenz wird in kleinen Schritten ein Vorhersagemodell manuell erstellt.

3. **Zwischenpräsentation** *(10 Minuten)*
   Die einzelnen Schritte der Modellierung werden im Plenum besprochen und der Bezug der Schritte zum Modellierungskreislauf hergestellt. Im Anschluss wird der Weg zur Automatisierung der Komposition aufgezeigt und der MIDI Standard erklärt.

4. **Zweite Arbeitsphase** *(35 Minuten)*
   Die Schüler automatisieren das Vorhersagemodell und arbeiten mit einem echten Lied als Trainingsdaten. Anschließend wird das Modell in zwei Schritten erweitert. Zunächst wird neben der Tonhöhe auch die Tonlänge berücksichtigt und danach

anstatt einer vorherigen Note, die zwei vorherigen Noten für die Notenübergänge berücksichtigt.

5. **Abschlusspräsentation, Evaluation und Verabschiedung** *(15 Minuten)*
Zum Abschluss werden die Ergebnisse im Plenum diskutiert und der Modellierungskreislauf nochmals angepasst. Die Präsentation endet mit einem Beispiel für ein mit künstlicher Intelligenz komponiertes Lied im Beatles Stil, zur Veranschaulichung welches Potential computerkomponierte Lieder haben. Anschließend folgt die Evaluation des Lernmoduls und die Verabschiedung.

Die zeitlichen Angaben dienen jeweils nur als Orientierung und müssen an den Kurs angepasst werden. Die Sicherungsphasen der Zwischen- und Abschlusspräsentation sollten erst durchgeführt werden, wenn idealerweise alle Schüler oder bei sehr heterogenen Gruppen ein Großteil der Schüler die Arbeitsblätter bearbeitet hat. Hierdurch kann die Sicherung im Dialog mit den Schülern erfolgen, wodurch die Schüler ihr Vorgehen selbst reflektieren.

4.6. Vorstellung der Materialien

4.6.1. Einstiegspräsentation

Die Einstiegspräsentation, siehe Anhang A.1, stellt die Problemstellung dar und führt den Modellierungskreislauf ein.

Zu Beginn des Vortrags wird kurz „künstliche Intelligenz“ erläutert und die Bedeutung für die Gesellschaft anhand verschiedener Beispiele dargestellt. Über den in verschiedenen Bereichen des Alltags bereits weit verbreiteten Einsatz folgt der Übergang zur Zielsetzung des Lernmoduls: Mit Hilfe künstlicher Intelligenz ein Lied zu komponieren.

Mit dem Hinweis, dass algorithmische Kompositionen auch schon vor dem Informationszeitalter entwickelt wurden, werden Einschätzungen der Schüler gesammelt, seit wann algorithmische Kompositionsverfahren entwickelt werden und wann das erste von einem Computer komponierte Lied veröffentlicht wurde. Anschließend wird der geschichtliche Hintergrund der algorithmischen Komposition, vergleiche Abschnitt 3.1.1, vorgestellt. Der geschichtliche Hintergrund wird dabei in Anlehnung an die Fragen zur Einschätzung aufgeteilt. Zuerst werden die ohne Computer entwickelten Algorithmen vorgestellt, wobei auf das Musikalische Würfelspiel als stochastischer Algorithmus genauer eingegangen wird. Anschließend werden die ersten Meilensteine in der Entwicklung algorithmischer Kompositionsverfahren mit Computern vorgestellt.

Im Anschluss werden die Schüler über den aktuellen Stand der Entwicklung informiert. Anhand der Existenz kommerzieller Angebote und Wettbewerbe und der Fertigstellung von Beethovens unvollendeter 10. Sinfonie mit Hilfe von künstlicher Intelligenz, werden den Schülern verschiedene aktuelle Anwendungsmöglichkeiten und die Bedeutung für die Gesellschaft gezeigt. Zur Veranschaulichung, wie gut computergenerierte Musik inzwischen ist, wird eine Sequenz des „Orchestra of the Age of Enlightenment“ (Orchestra of the Age of Enlightenment, 2019) gezeigt, in der ein Stück von Bach und ein mit KI komponiertes Stück gegenübergestellt werden. Hierzu werden Einschätzungen der Schüler gesammelt, welches Stück von Bach war.

Nach der Einführung in die Theorie wird den Schülern die mathematische Modellierung anhand der vier Schritte des Modellierungskreislaufs vorgestellt und erläutert. Anschließend wird die Problemstellung im Modellierungskreislauf eingeordnet.

Zuletzt erhalten die Schüler Hinweise, wie sie auf die digitalen Arbeitsblätter zugreifen und diese bearbeiten können.

4.6.2. Arbeitsblatt 1

Das erste Arbeitsblatt, siehe Anhang B.1, führt schrittweise zur Bildung eines Vorhersagemodells für Notenübergänge. Hierfür wird das Modell manuell erstellt und berechnet, die Automatisierung wird erst auf dem zweiten Arbeitsblatt berücksichtigt.

Das Arbeitblatt beginnt mit dem Kennenlernen von Noten, so dass das Modell manuell erstellt werden kann. Da sich das kurze Beispiel, auf dem das Modell aufgebaut wird, nur auf vier unterschiedliche Noten beschränkt, sind hier auch bei Schülern die keinen Bezug zur Musik haben keine Schwierigkeiten zu erwarten. Durch das Arbeiten mit den Notenzeichen, die in Buchstaben übersetzt werden, arbeiten die Schüler auf der symbolischen Ebene, vergleiche Abschnitt 2.4.

In den folgenden Aufgaben werden die absoluten und relativen Häufigkeiten der Notenübergänge berechnet. Die Lösungen werden jeweils in die Lücken der Codefelder eingetragen, wodurch die Schüler nach dem Prinzip der minimalen Hilfe, vergleiche Abschnitt 2.3, direkt ein Feedback zu ihrer Lösung erhalten.

Im nächsten Schritt folgt der Übergang von einzelnen Übergangswahrscheinlichkeiten zur gesamten Darstellung in der Übergangstabelle. Der Übergang und das Verständnis für die Bedeutung der Tabelle wird mit einem Beispielbild, siehe Abbildung 4.1, vereinfacht. Die Werte der Übergangstabelle werden von den Schülern zur Überprüfung wieder in einem Codefeld eingegeben.

$$\begin{array}{cc} & \begin{array}{cccc} a & d & f & g \end{array} \\ \begin{array}{c} a \\ d \\ f \\ g \end{array} & \begin{pmatrix} ? & ? & \frac{1}{2} & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix} \end{array}$$

Abbildung 4.1.: Vorgegebene Übergangstabelle

Die berechnete Übergangstabelle bildet die Grundlage für das Vorhersagemodell. Im letzten Schritt müssen nur noch die Wahrscheinlichkeiten der Notenübergänge simuliert werden. Zur Simulation der bedingten Wahrscheinlichkeiten wird virtuell-enaktiv ein Urnenmodell mit Ziehen mit Zurücklegen betrachtet, siehe Abbildung 4.2. Die Schüler sollen in dieser Aufgabe die Wahrscheinlichkeitsverteilungen in den Urnen den entsprechenden Noten zuordnen.

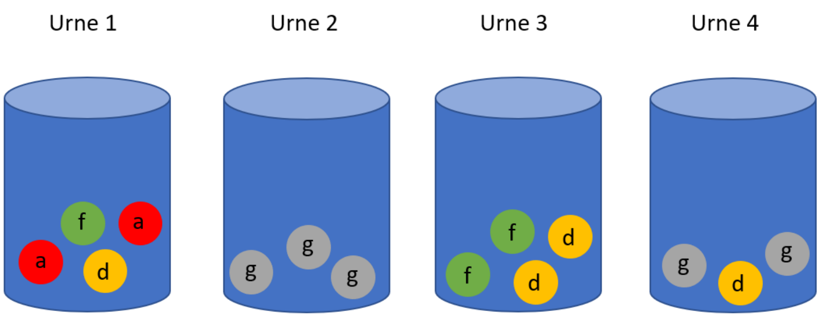


Abbildung 4.2.: Urnenmodell der Notenübergänge

Zum Abschluss des ersten Arbeitsblatts wird die Auswahl der nächsten Note von Julia ausgeführt und ein Lied generiert. Die Schüler können hierbei die erste Note und die Anzahl der Noten festlegen.

4.6.3. Zwischenpräsentation

Die Zwischenpräsentation, siehe Anhang A.3, dient der Zusammenfassung des bisherigen Vorgehens zur Modellbildung und als Einführung in die Automatisierung des Modells.

Die auf dem ersten Arbeitsblatt durchgeführten Schritte werden nacheinander in den Modellierungskreislauf eingeordnet und die Bedeutung für die Modellierung erläutert. Durch zusätzliche Fragen über das Vorgehen bei den Schritten des mathematischen Problems und der mathematischen Lösung sollen die Schüler zusätzlich zur Reflexion angeregt werden.

Im Anschluss werden die Schüler über das weitere Vorgehen zur Automatisierung der Komposition informiert, das sich in drei Schritte gliedert.

1. Echtes Lied als Grundlage einlesen
2. Automatische Berechnung der Übergangstabelle
3. Automatische Komposition eines neuen Lieds

Die Lieder werden im ersten Schritt als MIDI Dateien eingelesen. Da dieser Standard den meisten Schülern in der Regel nicht bekannt ist, werden die wichtigsten Grundlagen, siehe Abschnitt 3.2.3, kurz vorgestellt.

4.6.4. Arbeitsblatt 2

Auf dem zweiten Arbeitsblatt, siehe Anhang B.2, wird die Komposition automatisiert. Hierfür können die Schüler zunächst eines von fünf Liedern als Grundlage für die spätere Komposition auswählen. Da MIDI Dateien, wie in Abschnitt 3.2.3 beschrieben, nicht direkt Musik enthalten und sich beim Abspielen mit einem Interpreter entsprechend anders anhören als die bekannten Lieder, können die Schüler die Ausgangsdateien auch abspielen.

Die folgenden Aufgaben durchlaufen die in Abschnitt 3.2.2 beschriebenen Schritte über die Berechnung der Übergangstabelle zur Erzeugung einer neuen Sequenz. Auf Grund der komplexen und für Schüler unverständlichen Programmierung werden in diesem Abschnitt hauptsächlich Theoriefragen gestellt.

Für die neu generierte Sequenz können die Schüler wie auf dem ersten Arbeitsblatt die erste Note und die Anzahl der Noten festlegen und das Ergebnis anhören. Da in diesem ersten Modell nur die Notenhöhe und nicht die Notenlänge berücksichtigt wurde, hält sich die Ähnlichkeit mit dem Original noch in Grenzen.

Im nächsten Schritt wird das Modell durch die Berücksichtigung der Notenlänge erweitert. Da die Berechnung analog zum ersten Teil des zweiten Arbeitsblatts abläuft, wird hierauf nicht nochmal eingegangen und direkt die neue Sequenz generiert.

Zum Abschluss wird das Modell noch um die Berücksichtigung der zwei vorherigen Noten erweitert. Mit der ikonischen Betrachtung der hier entstehenden dreidimensionalen Übergangstabelle soll das Verständnis für das Modell gestärkt werden. Hierbei betrachten die Schüler nur vorgegebene Zeilen des Würfels, siehe Abbildung 4.3, und erläutern die Bedeutung. Nach Auswahl der ersten beiden Noten und der Anzahl der Noten können sich die Schüler das Ergebnis anhören.

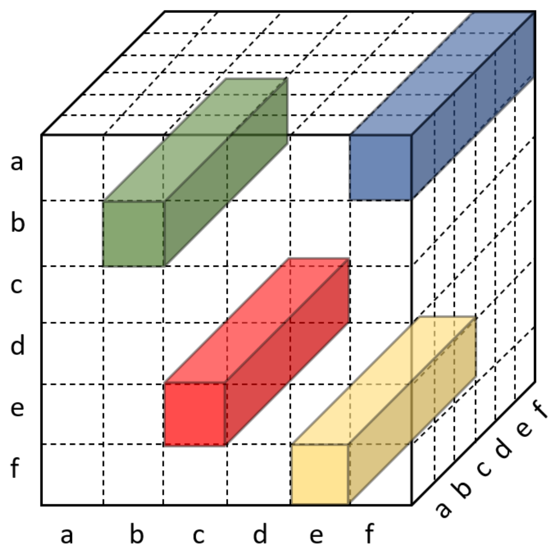


Abbildung 4.3.: 3-dim Übergangstabelle mit markierten Wahrscheinlichkeitsverteilungen

4.6.5. Abschlusspräsentation

In der Abschlusspräsentation, siehe Anhang A.5, werden die Ergebnisse und Möglichkeiten zur Erweiterung des Modells diskutiert.

Zunächst berichten die Schüler über die Qualität der im Lernmodul generierten Lieder und ob Ähnlichkeiten zu dem von ihnen gewählten Lied erkennbar sind. Dabei soll vor allem auch auf die Verbesserungen durch die Modellerweiterungen eingegangen werde. Da die MIDI Dateien der Lieder als Grundlage für eine neue Komposition jedoch unterschiedlich gut geeignet sind, sind hier verschiedene Meinungen zu erwarten.

Als nächstes werden die auf dem zweiten Arbeitsblatt durchgeführten Schritte in den Modellierungskreislauf eingeordnet, die sich durch die Automatisierung des Verfahrens im Vergleich zum ersten Arbeitsblatt etwas verändert haben. Hierbei wird nochmal die Übergangstabelle des erweiterten Modells, bei dem zwei vorherige Noten berücksichtigt werden, thematisiert und bei Bedarf erläutert.

Als Ausblick für weitere Verbesserungsmöglichkeiten des Modells werden Vorschläge der Schüler gesammelt und die Bedeutung für das mathematische Problem diskutiert.

Zum Abschluss des Lernmoduls, wird den Schülern ein mit künstlicher Intelligenz komponiertes Lied im Beatles Stil (Sony CSL, 2016) gezeigt, um das Potential computerkomponierter Lieder zu veranschaulichen.

4.6.6. Materialien für Dozenten

Zusätzlich zu den vorher beschriebenen Materialien stehen den Dozenten weitere Unterlagen zur Verfügung, die sie bei der Durchführung des Lernmoduls unterstützen. Hierzu zählen zum Einen das methodische Konzept, siehe Anhang D, mit einer Auflistung der benötigten Materialien und einem Verlaufsplan. Des Weiteren erhalten die Dozenten ein Basic Paper, siehe Anhang C, zur Einarbeitung in die mathematischen und fachlichen Hintergründe. Zusätzlich erhalten die Dozenten Notizen für die Präsentationen, siehe Anhänge A.2, A.4 und A.6.

5. Durchführung und Evaluation des Workshops

Da das Lernmodul auf Grund der Corona Pandemie nicht in Präsenz durchgeführt werden konnte, wurde es im Rahmen eines online Workshops erprobt. Die erste Durchführung fand am 18.10. statt. Eine zweite Durchführung konnte aus Zeitgründen nicht realisiert werden.

5.1. Beschreibung der Schülergruppe

Der Workshop wurde insbesondere über die Lernplattform Lern-Fair und math4MINT beworben, wodurch Teilnehmer aus ganz Deutschland gefunden werden konnten. Am Workshop haben acht Schüler und eine Schülerin teilgenommen, die alle ein Gymnasium besuchen. Bis auf zwei Schüler, die in der 10. Klasse sind, besuchen alle Teilnehmer die 11. Klasse. Obwohl es für den Workshop nicht erforderlich ist, ist ein Großteil der Teilnehmer selbst musikalisch aktiv und ein Teil davon hat sich auch schon mit der Komposition von Musik beschäftigt.

5.2. Beschreibung der Durchführung

Der Workshop wurde an einem Montag Nachmittag von 16 bis 18 Uhr durchgeführt. Für die Kommunikation wurde die Plattform Mattermost verwendet, die sowohl das Chatten als auch Videokonferenzen ermöglicht.

Da nicht alle Schüler pünktlich um 16 Uhr der Videokonferenz beigetreten waren, begann die Begrüßung mit einer 5-minütigen Verspätung. Durch technische Schwierigkeiten beim Anlegen der Accounts zum Zugriff auf die Lernmaterialien kam es nach der Einführungspräsentation nochmals zu einer 10-minütigen Verzögerung.

Die Schüler wurden schließlich in zwei Gruppen mit 4 bzw. 5 Personen eingeteilt um das erste Arbeitsblatt gemeinsam zu bearbeiten. Hierfür wurden den beiden Gruppen jeweils ein Videokonferenzraum für die Kommunikation zur Verfügung gestellt. Die beiden Gruppen unterschieden sich bei der Zusammenarbeit innerhalb der Gruppen deutlich. Während eine Gruppe jede Aufgabe zusammen diskutierte um eine gemeinsame Lösung zu finden, war die zweite Gruppe durch Einzelarbeit geprägt, die die Möglichkeit zur Zusammenarbeit nur in einzelnen Fällen nutzte. Probleme auf dem ersten Arbeitsblatt konnten jeweils mit dem Hinweis den Aufgabentext nochmal genau zu lesen behoben werden.

Die Zwischenpräsentation war durch viel Interaktion mit den Schülern geprägt. Die einzelnen Schritte der Modellierung konnten von den Schülern gut nachvollzogen werden.

Das zweite Arbeitsblatt musste auf Grund der Verzögerungen zu Beginn für manche Schüler etwas gekürzt werden. In diesen Fällen wurden die Aufgaben von Teil 4b) nicht bearbeitet. Schwierigkeiten bei der Bearbeitung gab es hauptsächlich beim Einsetzen der Formel in Schritt 3 von Teil 3b). Auch hier war ein Hinweis auf den Aufgabentext für die Lösung meistens ausreichend.

Die Plenumsdiskussion in der Abschlusspräsentation zeigte eine unterschiedliche Reaktion bei der Zufriedenheit mit dem Ergebnis. Während manche Schüler die Ähnlichkeit erkannten und vom Ergebnis überrascht waren, hatten andere Schüler größere Erwartungen an das Endergebnis.

Für die Durchführung der Evaluation musste am Ende etwa 10 Minuten überzogen werden, so dass der Workshop erst um 18:10 Uhr endete.

5.3. Evaluation

Die Evaluation wurde mit Hilfe einer Onlineumfrage durchgeführt und zusätzlich wurden Schüler während des Workshops nach Schwierigkeiten oder Problemen befragt.

5.3.1. Aufbau des Fragebogens

Der Fragebogen war thematisch in fünf Teile aufgeteilt, siehe Anhang E.1.

Persönliches
: Fragen nach dem Geschlecht, der Schulform, der Jahrgangsstufe und der Kurswahl

Schwierigkeitsgrad
: Fragen zur eigenständigen Lösbarkeit, dem benötigten Vorwissen, der Schwierigkeit der Aufgaben und dem Umgang mit Julia

Gestaltung
: Fragen zu den Präsentationen, der Aufgabenstellung und dem zeitlichen Ablauf

Motivation
: Fragen zum persönlichen Bezug zu Musik und dem Interesse am Thema

Offenes Feedback
: Möglichkeit für Lob, Kritik und Verbesserungsvorschläge

Bei den Fragen zu den Themen Schwierigkeitsgrad, Gestaltung und Motivation wurde jeweils die Zustimmung über die vier Antwortmöglichkeiten „trifft gar nicht zu“, „trifft eher nicht zu“, „trifft zum Teil zu “ oder „trifft voll zu “ abgefragt.

5.3.2. Auswertung der Evaluation

Die Ergebnisse der Umfrage aus dem ersten Teil wurden bereits in Abschnitt 5.1 besprochen und werden deshalb hier nicht weiter thematisiert.

Schwierigkeitsgrad

Die Schwierigkeit der Aufgaben hatte ein der Gruppe angemessenes Niveau, so dass für alle Teilnehmer das benötigte Vorwissen ausreichend war und die Aufgaben eigenständig bearbeitet werden konnten. Dies spiegelt sich auch in der Einschätzung der Schüler wider, dass die Aufgaben weder zu schwer, noch zu leicht waren. Schwierigkeiten hatte aber mehrere Schüler beim Umgang mit Julia.

Gestaltung

Die Präsentationen und die Plenumsdiskussionen wurden jeweils sehr positiv und für das Verständnis als sehr hilfreich beurteilt. Des Weiteren wurden die Aufgabenstellungen als verständlich und abwechslungsreich bewertet. Auf Grund der Verzögerungen zu Beginn des Workshops hatte auch kein Schüler Phasen in denen er nichts zu tun hatte. Ohne Verzögerung hätte es auf Grund der Heterogenität der Gruppe für manche Schüler vermutlich solche Phasen gegeben.

Motivation

Wie bereits in Abschnitt 5.1 erwähnt wurde, hatte ein Großteil der Schüler durch das Spielen eines Instruments einen persönlichen Bezug zur Musik. Nahezu alle Teilnehmer hatten vor dem Workshop auch schon von automatischer Komposition gehört und für fast alle konnte das Interesse an diesem Thema mit dem Workshop gesteigert werden.

Offenes Feedback

Im offenen Feedback wurden hauptsächlich die technischen Probleme zu Beginn des Workshops und teilweise der Umgang mit Julia kritisiert. Gelobt wurden hingegen die interaktiven Arbeitsblätter, die Zusammenarbeit in der Gruppe und das Thema an sich.

5.3.3. Fazit

Basierend auf meiner persönlichen Einschätzung und dem Ergebnis der Evaluation kann die Durchführung des Workshops als Erfolg bewertet werden. Die Schüler zeigten bei den Plenumsdiskussionen jeweils, dass sie die Thematik verstanden haben und bestätigen dies in der Evaluation. Die in 4.1 formulierten Ziele konnten somit erreicht werden.

Insgesamt wurde der Workshop mit der Schulnote 2,1 bewertet, wobei die Durchschnittsnote insbesondere durch eine 5 negativ beeinflusst wurde. Der Workshopleiter wurde mit der Schulnote 1,4 bewertet.

Da alle Teilnehmer einen starken Bezug zur Musik hatten, wäre eine weitere Durchführung mit Schülern ohne direkten Bezug zur Musik sinnvoll, um eventuelle Schwierigkeiten zu entdecken.

6. Ausblick

Der entwickelte Workshop bietet noch Möglichkeiten zur Weiterentwicklung, einige Ideen und Vorschläge werden in diesem Kapitel vorgestellt.

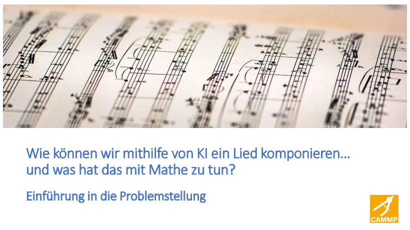
Wie in Kapitel 4.2 erläutert, wurden im Workshop Markov Ketten vollständig auf die bedingte Wahrscheinlichkeit reduziert. Ausgehend von den Erkenntnissen aus Teil 3 des zweiten Arbeitsblatts könnten diese aber in einem für die Kursstufe konzipierten Workshop thematisiert und weitere Anwendungsmöglichkeiten betrachtet werden. Des Weiteren könnte man hier Themen wie erreichbare und kommunizierende Noten oder irreduzible Übergangsmatrizen diskutieren, vergleiche Definition 3.4.

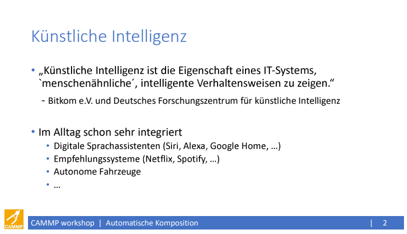
Die Tonhöhe und Tonlänge werden bisher als getrennte Zustände in eigenen Übergangsmatrizen betrachtet. Zur weiteren Verbesserung des Modells können die Tonhöhe und Tonlänge einer Note zusammen als Zustand erfasst werden. Hierzu können die Übergangsmatrizen aber nicht mehr als Matrizen programmiert werden, sondern müssen in Form von Dataframes gespeichert werden. Dies würde zudem die Berechnung von Markov Ketten höherer Ordnung ermöglichen.

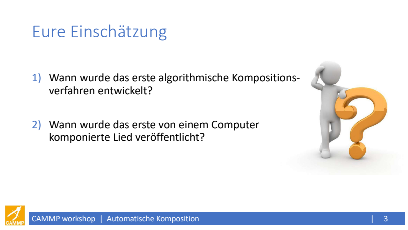
Durch den Bezug zu Künstlicher Intelligenz und Maschinellem Lernen bieten sich darüber hinaus auch Möglichkeiten den Workshop für Informatik Kurse aufzubereiten. Der für die Automatisierung verwendete Code wird im aktuellen Workshop auf Grund der Komplexität nur in Form von Pseudocode besprochen. Mit entsprechenden Vorkenntnissen könnten hier aber auch kleine Programmieraufgaben gestellt werden.

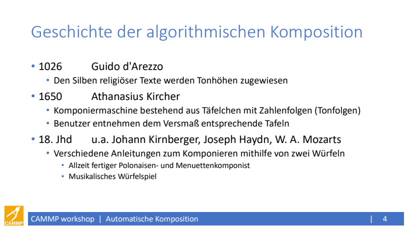
A. Präsentationen

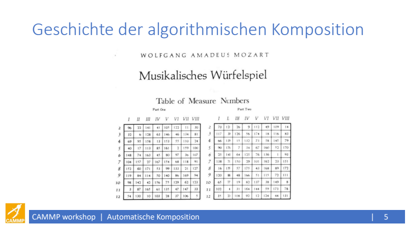
A.1. Einstiegspräsentation

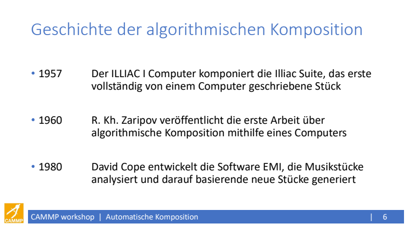




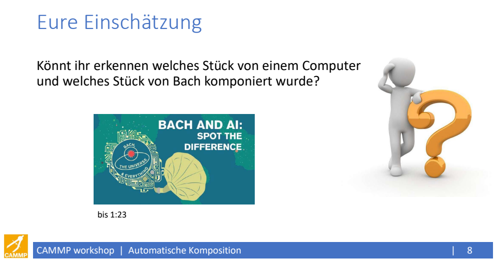


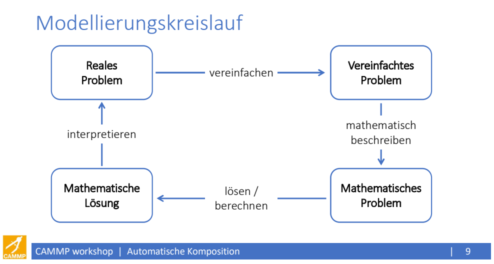


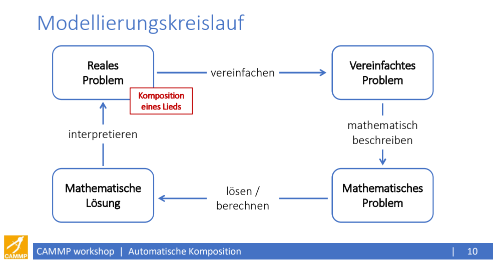


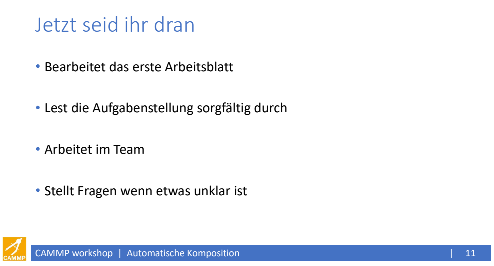


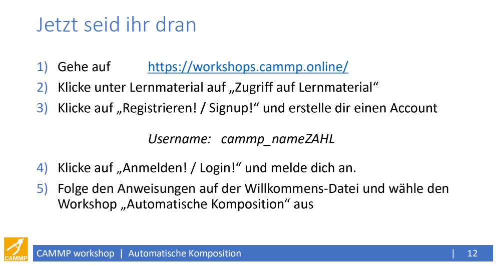

Automatische Komposition mit KI heute
KÜNSTLICHE INTELLIGENZ
Beethovens "Unvollendete" von KI vollendet
Mit Hilfe von Algorithmen haben Wissenschaftler die 10. Sinfonie zu Ende komponieren lassen. Sind Computerprogramme so kreativ wie ein Musikgenie?
SONG CONTEST 2021
AIVA
The Artificial Intelligence composing emotional soundtrack music
CAMMP workshop | Automatische Komposition | 7


Eure Einschätzung
Könnt ihr erkennen welches Stück von einem Computer und welches Stück von Bach komponiert wurde?
BACH AND AI:
SPOT THE DIFFERENCE
bis 1:23
CAMMP workshop | Automatische Komposition | 8


Modellierungskreislauf
Reales Problem
vereinfachen
Vereinfachtes Problem
mathematisch beschreiben
Mathematisches Problem
lösen / berechnen
Mathematische Lösung
interpretieren
CAMMP workshop | Automatische Komposition | 9


Modellierungskreislauf
Reales Problem
Komposition eines Lieds
vereinfachen
Vereinfachtes Problem
mathematisch beschreiben
Mathematisches Problem
lösen / berechnen
Mathematische Lösung
interpretieren
CAMMP workshop | Automatische Komposition | 10


Jetzt seid ihr dran
• Bearbeitet das erste Arbeitsblatt
• Lest die Aufgabenstellung sorgfältig durch
• Arbeitet im Team
• Stellt Fragen wenn etwas unklar ist
CAMMP workshop | Automatische Komposition | 11


Jetzt seid ihr dran
1) Gehe auf https://workshops.cammp.online/
2) Klicke unter Lernmaterial auf „Zugriff auf Lernmaterial“
3) Klicke auf „Registrieren! / Signup!“ und erstelle dir einen Account
Username: cammp_nameZAHL
4) Klicke auf „Anmelden! / Login!“ und melde dich an.
5) Folge den Anweisungen auf der Willkommens-Datei und wähle den Workshop „Automatische Komposition“ aus
CAMMP workshop | Automatische Komposition | 12

A.2. Notizen zur Einstiegspräsentation

Notizen Einstiegspräsentation

Folie 1: **Begrüßung**

- Begrüßung der Schüler und Vorstellung des Themas

Folie 2: **Künstliche Intelligenz**

- Je nachdem in welchem Fachbereich KI angewendet wird, existieren verschiedene Definitionen. Für die Anwendung in der Musikkomposition ist z.B. die Definition des Bitkom e.V. und Deutschen Forschungszentrum für künstliche Intelligenz passend.
- Künstliche Intelligenz ist bereits bei vielen Anwendungen im Alltag integriert. (Möglicher Hinweis auf CAMMP day zum Thema Netflix)

Folie 3: **Einschätzung der Schüler**

- Hinweis, dass schon vor dem Computeralter und künstlicher Intelligenz algorithmische Kompositionsverfahren entwickelt wurden
- Schüler geben eine zeitliche Einschätzung zur Entwicklung dieser Verfahren ab

Folie 4: **Geschichte der algorithmischen Komposition**

- 1026: Der Benediktinermönch und Musiktheoretiker Guido von Arezzo entwickelt ein Verfahren zur automatischen Generation von Melodien aus Textmaterial. Hierbei werden Buchstaben und Silben auf Tonhöhen und melodische Phrasen abgebildet
- 1650: Der Universalgelehrte Athanasius Kircher entwickelt mit der Komponiermaschine eine mechanisches System zur automatischen Komposition. Die Komponiermaschine besteht aus Täfelchen mit vierzeiligen Zahlenspalten, wobei die Zahlen jeweils Tönen zugeordnet sind. Die Benutzer entnehmen dem Versmaß des Textes entsprechende Tafeln und fügen willkürlich gewählte Segmente aneinander.
- 18. Jhdt.: Zum Ende des 18. Jahrhunderts entwickeln verschiedene Komponisten und Musiktheoretiker musikalische Würfelspiele. Mit dieser Methode wurden meistens Walzer, Polonaisen oder Menuetten komponiert. Die bekanntesten Würfelspiel sind der „Allzeit fertige Polonaisen und Menuettenkomponist“ von Kirnberger und das „Musikalische Würfelspiel“ von Mozart

Folie 5: **Musikalisches Würfelspiel**

- In einer Tabelle stehen vorkomponierte Takte, durch Würfeln oder Kartenziehen wird eine Zufallszahl erzeugt, die eine Zeile aus einer Tabelle bestimmt. Die Spalten der Tabelle werden nach der Reihenfolge der Würfe ausgewählt.

Folie 6: Geschichte der algorithmischen Komposition

- 1957: Die amerikanischen Komponisten Lejaren Hiller und Leonard Issacson programmierten den ILLIAC I Computer, der die Illiac Suite komponierte, das erste vollständig von einem Computer geschriebene Stück.
- 1960: Der russische Wissenschaftler R. Kh . Zaripov veröffentlicht die erste Arbeit über algorithmische Komposition mithilfe eines Computers.
- 1980: Der amerikanische Wissenschaftler und Komponist David Cope entwickelt die Software EMI (Experiments in Musical Intelligence). Das System analysiert existierende Musikstücke und generiert darauf basierende neue Musikstücke.

Folie 7: Automatische Komposition mit KI heute

- Im Oktober 2021 wurde Beethovens unvollendete 10. Sinfonie mit Hilfe von KI vollendet
- AIVA ist ein Beispiel für einen kommerziellen Anbieter von mit KI erzeugter Musik
- Seit 2020 findet ein jährlicher AI Song Contest statt, der vom Eurovision Song Contest inspiriert wurde

Folie 8: Bach und künstliche Intelligenz

- Video (0:00 – 1:23) zeigen und Schüler raten lassen, welches Stück von Bach ist
- Lösung: das zweite Stück ist von Bach

Folie 9: Modellierungskreislauf

- Für die Lösung der Problemstellung wird das Werkzeug der mathematischen Modellierung verwendet
- Ein betrachtetes reales Problem aus dem Alltag wird zunächst durch geeignete Annahmen vereinfacht. Das vereinfachte Problem kann dann mathematisch beschrieben werden, so dass das mathematische Problem mit mathematischen Methoden gelöst werden kann. Die mathematische Lösung muss schließlich in Bezug auf das reale Problem interpretiert werden.

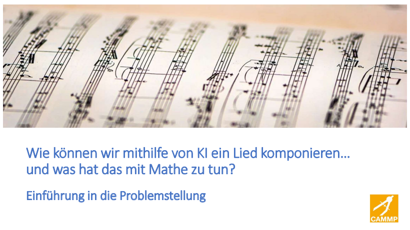
Folie 10: Modellierungskreislauf

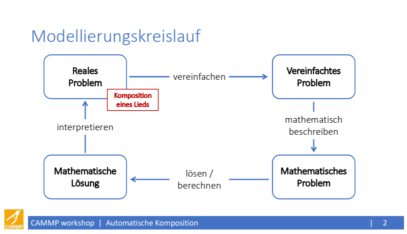
- In unserem realen Problem gehen wir von der Komposition eines Lieds aus.

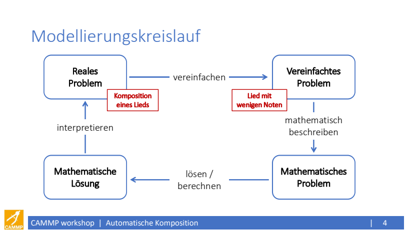
Folie 11: Hinweise für die Bearbeitung der Arbeitsblätter

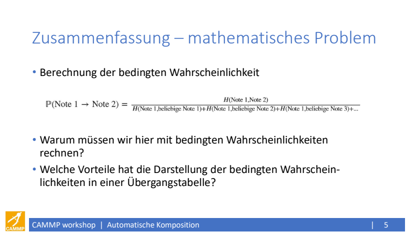
Folie 12: Anleitung für den Zugriff auf die Lernmaterialien

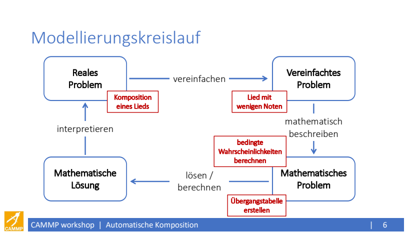
A.3. Zwischenpräsentation





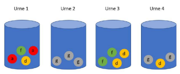






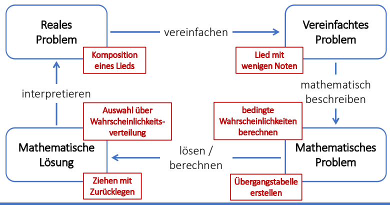
Zusammenfassung – mathematische Lösung

- Ziehen mit Zurücklegen



- Auswahl über Wahrscheinlichkeitsverteilung durch Julia
- Wie könnte man die Auswahl der nächsten Note noch simulieren?

Modellierungskreislauf



Automatisierung der Komposition

- Echtes Lied als Grundlage einlesen
- Übergangstabelle automatisch berechnen
- Neues Lied automatisch komponieren

MIDI Dateien

- Musical Instrument Digital Interface (digitale Schnittstelle für Musikinstrumente)
- ursprünglich zum Austausch von Daten zwischen elektronischen Musikinstrumenten
- MIDI Dateien enthalten digitale Informationen über Notenhöhe, Notenlänge und Notenposition
- eine Art Notenschrift in Tabellenform

```
Note B3  | vel = 122 | pos = 39840, dur = 195 | channel 3
Note E4  | vel = 122 | pos = 40080, dur = 216 | channel 3
Note F♯4 | vel = 122 | pos = 40320, dur = 413 | channel 3
```


Jetzt seid ihr dran

- Bearbeitet das zweite Arbeitsblatt
- Lest die Aufgabenstellung sorgfältig durch
- Arbeitet im Team
- Stellt Fragen wenn etwas unklar ist

A.4. Notizen zur Zwischenpräsentation

Notizen Zwischenpräsentation

Folie 2: Modellierungskreislauf

- Wiederholung des Modellierungskreislaufs mit der Problemstellung der Musikkomposition

Folie 3: Vereinfachtes Problem

- Das reale Problem wurde durch ein kurzes Lied auf wenige Noten vereinfacht

Folie 4: Modellierungskreislauf

- Ergänzung der Vereinfachung zum Modellierungskreislauf

Folie 5: Mathematisches Problem

- Das mathematische Problem wird durch die Berechnung der bedingten Wahrscheinlichkeiten definiert
- Warum bedingte Wahrscheinlichkeit? Wir interessieren uns nicht für die Auftretenswahrscheinlichkeit einer Note, sondern für die Wahrscheinlichkeit, dass eine Note auf eine bestimmte Note folgt
- Vorteile der Übergangstabelle: Übersichtliche Darstellung, Einfache Überprüfung auch Richtigkeit → Zeilensumme muss 1 ergeben

Folie 6: Modellierungskreislauf

- Ergänzung des mathematischen Problems zum Modellierungskreislauf

Folie 7: Mathematische Lösung

- Zur Lösung des mathematischen Problems wird die Wahrscheinlichkeitsverteilung durch Ziehen mit Zurücklegen aus der entsprechenden Urne simuliert
- Alternativ kann die Auswahl auch durch Julia erfolgen
- Andere Auswahlmöglichkeiten: Spielkarten, Würfel
  ACHTUNG: die entsprechenden Wahrscheinlichkeiten müssen auch dargestellt werden können, mit einem Würfel können z.B. nur die Wahrscheinlichkeiten 1/6, 1/3 und 1/2 simuliert werden

Folie 8: **Modellierungskreislauf**

- Ergänzung der mathematischen Lösung zum Modellierungskreislauf

Folie 9: **Schritte zur Automatisierung**

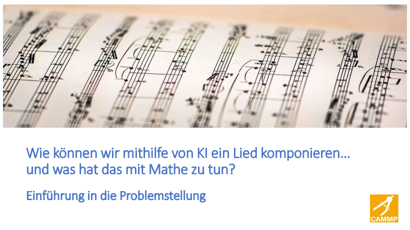
- Die auf dem ersten Arbeitsblatt durchgeführten Schritte sollen auf dem zweiten Arbeitsblatt in drei Schritten automatisiert werden
    1. Einlesen eines echten Lieds als Grundlage
    2. Automatische Berechnung der Übergangstabelle
    3. Automatische Komposition eines neuen Lieds

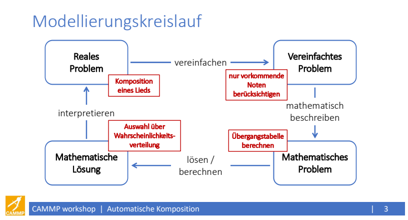
Folie 10: **MIDI Dateien**

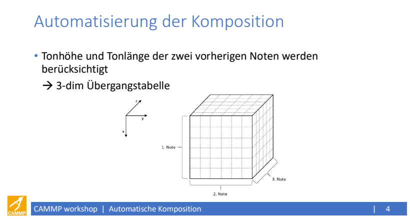
- Die Lieder werden als MIDI Dateien eingelesen
- MIDI steht für Musical Instrument Digital Interface (deutsch: digitale Schnittstelle für Musikinstrumente) und wurde ursprünglich zum Austausch von Daten zwischen elektronischen Musikinstrumenten verwendet
- MIDI Dateien enthalten nicht direkt Musik, sondern nur Anweisungen was zu tun ist. Sie MIDI Dateien enthalten digitale Informationen über Notenhöhe, Notenlänge und Notenposition
- MIDI Dateien sind eine Art Notenschrift in Tabellenform

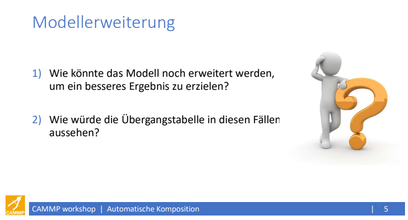
Folie 11: **Hinweise für die Bearbeitung der Arbeitsblätter**

A.5. Abschlusspräsentation









A.6. Notizen zur Abschlusspräsentation

Notizen Abschlusspräsentation

Folie 2: Feedback zu den automatisch generierten Liedern

- Die Schüler berichten über die Qualität der generierten Lieder und ob Ähnlichkeiten zu dem von ihnen gewählten Lied erkennbar sind
- Die Verbesserung des Ergebnisses durch die Modellerweiterung sollte erkennbar sein

Folie 3: Modellierungskreislauf

- Der Modellierungskreislauf wird an die Schritte der automatischen Komposition angepasst
- Das reale Problem wurde vereinfacht, indem nur die vorkommenden Noten berücksichtigt werden
- Das mathematische Problem wird auf die Berechnung der Übergangstabelle reduziert
- Die mathematische Lösung erfolgt durch die Auswahl über die Wahrscheinlichkeitsverteilung durch Julia

Folie 4: Übergangstabelle für zwei vorherige Noten

- Besprechung der 3-dim Übergangstabelle
- Die x-Richtung gibt die erste Note an, die y-Richtung gibt die zweite Note an und die z-Richtung gibt schließlich die Wahrscheinlichkeitsverteilung für die Noten, die auf die ersten beiden Noten folgen können, an

Folie 5: Modellerweiterung

- Plenumsdiskussion zu weiteren möglichen Erweiterungen zur Verbesserung des Modells
- Mögliche Erweiterungen:
    - noch mehr vorherige Noten berücksichtigen → Dimension der Übergangstabelle steigt entsprechen, so dass wir sie uns nicht bildlich vorstellen können
    - einzelne Ausschnitte des Lieds als Motiv betrachten und als Grundlage verwenden

Folie 6: Beispiel für ein mit KI komponiertes Lied

- Das Lied wurde nicht vollständig von KI komponiert. Von KI komponierte Ausschnitte wurde von einem Produzent gemischt und der von KI generierte Text von einem Sänger gesungen

B. Digitale Arbeitsblätter

B.1. Arbeitsblatt 1

Arbeitsblatt 1

Damit du dieses Arbeitsblatt bearbeiten kannst, musst du als erstes das folgende Codefeld ausführen. Klicke dazu in das nächste Codefeld und drücke auf den "Run"-Button (oder drücke Shift + Enter).

In [ ]:
```
# Hier nichts ändern
include("../code/loadFunctions1.jl");
```

Einleitung

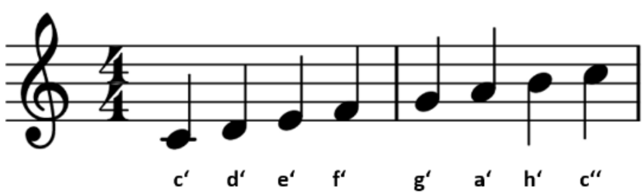
Ziel des Workshops ist es zu lernen, wie man auf Grundlage eines bekannten Lieds nur mit Hilfe der Wahrscheinlichkeitsrechnung ein neues ähnliches Lied komponieren kann. Hierfür müssen wir uns zunächst überlegen, wie man ein Lied modellieren kann. Bevor wir mit einem ganzen Lied beginnen, wollen wir die Funktionsweise der automatischen Komposition zunächst anhand eines kleinen Beispiels veranschaulichen.

Teil 1) Manuelle Komposition

Bevor wir die Komposition automatisieren, führen wir diese zunächst manuell durch um den Prozess der zufälligen Komposition zu veranschaulichen. Hierzu müssen wir ein paar Noten kennenlernen.

a) Noten kennenlernen

Das folgende Bild zeigt die Töne der C-Dur Tonleiter mit den Stammtönen C, D, E, F, G, A und H.



b) Notenfolge analysieren

Als Grundlage für unsere eigene Komposition dient das folgende kurze Musikstück.

Schreibe dir die Notenfolge auf.

Du kannst dir das Lied auch anhören, indem du das folgende Codefeld ausführst.

In [ ]:
```
#Hier nichts ändern
Komposition.inline_audioplayer("../data/bsp.wav")
```

Wir untersuchen jetzt wie oft eine bestimmte Note auf eine andere Note folgt. Hierzu betrachten wir zunächst die absoluten Häufigkeiten.

Notiere die absolute Häufigkeit der Übergänge indem du sie für `NaN` im folgenden Codelfeld einsetzt.

In [ ]:
```
# Übergang von "a" zu "d"
h1=   NaN;
# Übergang von "a" zu "f"
h2=   NaN;
# Übergang von "d" zu "d"
h3=   NaN;
# Übergang von "d" zu "g"
h4=   NaN;
# Übergang von "f" zu "g"
h5=   NaN;
# Übergang von "g" zu "a"
h6=   NaN;
# Übergang von "g" zu "d"
h7=   NaN;
# Übergang von "g" zu "f"
h8=   NaN;

#Hier nichts ändern
Komposition.checkA(h1,h2,h3,h4,h5,h6,h7,h8)
```

Mit Hilfe der absoluten Häufigkeiten sollen nun die relativen Häufigkeiten der Übergänge bestimmt werden.

(1) *Beschreibe wie man aus der absoluten Häufigkeit des Übergangs von der Note "a" zur Note "d" die relative Übergangswahrscheinlichkeit $\mathbb{P}(\mathrm{a} \to \mathrm{b})$ berechnen kann?*

▸ Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

---

(2) *Gib eine allgemeine Formel zur Berechnung der relativen Übergangswahrscheinlichkeit $\mathbb{P}(\mathrm{Note}\ 1 \to \mathrm{Note}\ 2)$ an.*

▸ Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

Berechne nun die Übergangswahrscheinlichkeiten für die einzelnen Übergänge. Notiere die Übergangswahrscheinlichkeiten indem du sie für `NaN` im folgenden Codelfeld einsetzt.

In [ ]:
```
# Übergangswahrscheinlichkeit von "a" zu "d"
p1=   NaN;
# Übergangswahrscheinlichkeit von "a" zu "f"
p2=   NaN;
# Übergangswahrscheinlichkeit von "d" zu "d"
p3=   NaN;
# Übergangswahrscheinlichkeit von "d" zu "g"
p4=   NaN;
# Übergangswahrscheinlichkeit von "f" zu "g"
p5=   NaN;
# Übergangswahrscheinlichkeit von "g" zu "a"
p6=   NaN;
# Übergangswahrscheinlichkeit von "g" zu "d"
p7=   NaN;
# Übergangswahrscheinlichkeit von "g" zu "f"
p8=   NaN;

#Hier nichts ändern
Komposition.checkB(p1,p2,p3,p4,p5,p6,p7,p8)
```

Für eine bessere Übersicht, insbesondere wenn später mit mehr Noten gearbeitet wird, können die Übergangswahrscheinlichkeiten auch in einer Tabelle dargestellt werden. Die Zeile gibt dabei die aktuelle Note und die Spalte die Folgenote an, die Einträge entsprechen den Übergangswahrscheinlichkeiten. Die Übergangswahrscheinlichkeit von "a" zu "f" ist $\frac{1}{2}$, wir tragen daher in der Zeile "a" und Spalte "f" die Übergangswahrscheinlichkeit ein und erhalten folgende **Übergangstabelle**:

$$
\begin{array}{c c}
 & \begin{array}{cccc} a & d & f & g \end{array} \\
\begin{array}{c} a \\ d \\ f \\ g \end{array} & \begin{pmatrix} ? & ? & ½ & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}
\end{array}
$$

Trage die fehlenden Übergangswahrscheinlichkeiten im folgenden Codefeld ein, indem du sie für die `NaN` einsetzt. Beachte dabei, dass es Übergänge gibt, die im Lied nicht auftauchen. Die Wahrscheinlichkeit für einen Übergang liegt daher bei 0.

In [ ]:

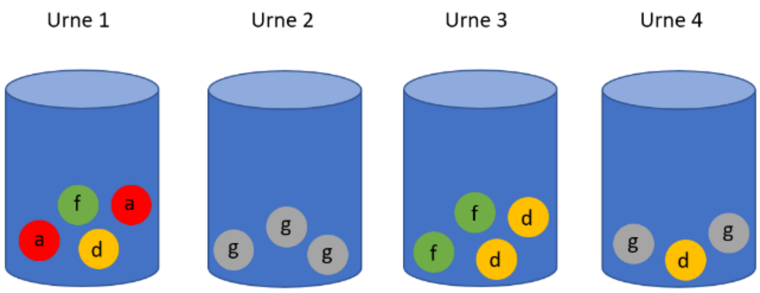
```
# Ersetze NaN
Tabelle =  [NaN NaN 1/2 NaN; # erste Zeile der Tabelle
           NaN NaN NaN NaN; # zweite Zeile der Tabelle
           NaN 0 NaN NaN; # dritte Zeile der Tabelle
           NaN NaN NaN 0]; # vierte Zeile der Tabelle

#Hier nichts ändern
Komposition.checkC(Tabelle)
```

Welche Bedingung muss jede Zeile der Übergangstabelle erfüllen und warum?

► Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

c) Über bedingte Wahrscheinlichkeiten Notenübergänge erzeugen

Mit den zuvor bestimmten Übergangswahrscheinlichkeiten wollen wir jetzt ein neues Lied komponieren, indem wir die unterschiedlichen Wahrscheinlichkeiten dafür nutzen die nächste Note zu bestimmen. Zur Simulation der Wahl der nächsten Note in Abhängigkeit von den Übergangswahrscheinlichkeiten betrachten wir das Urnenmodell. Je nachdem von welcher Note wir ausgehen, müssen wir eine andere Urne zur Bestimmung der nächsten Note wählen.



Bestimme welche Urne zu welcher Note gehört. Trage die entsprechende Note im folgenden Codefeld ein, indem du sie für die `NaN` einsetzt.

In [ ]:
```
# Ersetze NaN, die Anführungszeichen müssen stehen bleiben
Urne_1 =  "NaN";
Urne_2 =  "NaN";
Urne_3 =  "NaN";
Urne_4 =  "NaN";

#Hier nichts ändern
Komposition.checkD(Urne_1, Urne_2, Urne_3, Urne_4)
```

Mit dieser Methode kann durch Ziehen mit Zurücklegen eine neue Notenfolge und somit ein Lied komponiert werden. Man wählt eine beliebige erste Note und zieht aus der entsprechenden Urne eine Kugel. Die Kugel gibt die nächste Note und somit auch die nächste Urne, aus der gezogen wird, an.

Teil 2) Automatische Komposition

Die Auswahl der nächsten Note können wir mit Hilfe von Julia automatisieren und somit schneller ein längeres Lied komponieren. Wir benötigen hierzu wieder die Übergangstabelle, da sie für jede Note die Wahrscheinlichkeitsverteilung für die nächste Note enthält.

Wähle eine Note, mit der das Lied anfangen soll und trage sie im Feld "ersteNote" für `NaN` ein. Wähle anschließend die Anzahl der Noten, also wie lange das Lied werden soll und trage sie im Feld "anzahlNoten" für `NaN` ein.

In [ ]:
```
# Ersetze NaN, die Anführungszeichen müssen stehen bleiben
ersteNote =  "NaN";
anzahlNoten =  NaN;

#Hier nichts ändern
notenfolge = Komposition.folge(Tabelle; ersteNote, anzahlNoten)
```

Führe das folgende Codefeld aus, um dir das Ergebnis anzuhören.

In [ ]:
```
#Hier nichts ändern
Komposition.createmidi(notenfolge, anzahlNoten)
```

B.2. Arbeitsblatt 2

Arbeitsblatt 2

Damit du dieses Arbeitsblatt bearbeiten kannst, musst du als erstes das folgende Codefeld ausführen. Klicke dazu in das nächste Codefeld und drücke auf den "Run"-Button (oder drücke Shift + Enter).

In [ ]:
```
# Hier nichts ändern
include("../code/loadFunctions2.jl");
```

Teil 3) Mit einem echten Lied arbeiten

Wir wollen die automatische Komposition jetzt erweitern, indem wir ein echtes Lied als Grundlage für unsere Übergangstabelle verwenden und diese ebenfalls automatisch erzeugen lassen.

a) Noten einlesen

Zum Einlesen der Noten benutzen wir eine MIDI Datei, in der unter anderem die Notenhöhen und -längen gespeichert sind. Zur Auswahl stehen 5 verschiedene Lieder. Wähle eines der folgenden Lieder aus, auf Grundlage dessen ein neues Lied komponiert werden soll.

- Chainsmokers - Paris
- Ed Sheeran - Shape of you
- Juice Wrld - Lucid Dreams
- Luis Fonci - Despacito
- Marshmello - Alone

Trage den Liedtitel für `NaN` in das folgende Codefeld ein um dir die ersten 10 Noten des Lieds anzuzeigen. Der Titel muss in Kleinbuchstaben und ohne Leerzeichen eingetragen werden, also z.B. für "Ed Sheeran - Shape of you" muss "shapeofyou" eingetragen werden.

In [ ]:
```
# Ersetze NaN, die Anführungszeichen müssen stehen bleiben
liedtitel =  "NaN";

#Hier nichts ändern
notenfolge = Komposition.importieren(liedtitel);
Komposition.ersteNoten(notenfolge)
```

In der Notenfolge werden nicht die Namen der Noten, sondern die entsprechenden MIDI Nummern angezeigt. Das mittlere C ("C4") entspricht z.B. der Nummer 60. Die MIDI Version eines Lieds hört sich auch deutlich anders an, als die Original Version eines Lieds.

Führe das folgende Codefeld aus um die MIDI Datei abzuspielen.

In [ ]:
```
#Hier nichts ändern
Komposition.playmidi(liedtitel)
```

---

b) Übergangstabelle berechnen

Zum Berechnen der Übergangstabelle betrachten wir zunächst wieder die absolute Häufigkeit der Übergänge und stellen diese in einer Tabelle dar.

Von welchen Größen hängt die Anzahl der Spalten bzw. Zeilen der Übergangstabelle ab?

▸ Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

Führe das folgende Codefeld aus um die vorkommenden Noten anzuzeigen.

In [ ]:
```
#Hier nichts ändern
einzelneNoten=unique(notenfolge)
```

Nun wollen wir die Übergangstabelle schrittweise erzeugen und berechnen.

Schritt 1:
Wir erstellen zuerst eine leere Tabelle für die absolute Häufigkeit der Übergänge.

Ersetze die `NaN` mit der Anzahl der Zeilen und Spalten der Tabelle und führe das Codefeld aus um die leere Tabelle zu laden.

In [ ]:
```
#Ersetze die NaN
zeilen =  NaN;
spalten =  NaN;

Komposition.checkF(zeilen, spalten, einzelneNoten)
```

Schritt 2:
Nun wollen wir die Tabelle mit Werten füllen. Wir durchlaufen die Zeilen der Tabelle mit der Variable i und die Spalten der Tabelle mit der Variable j.

Wir durchlaufen mit der Variablen k die aufeinander folgenden Noten des Lieds. Definiere in Abhängigkeit von k eine Bedingung, mit der abgezählt werden kann, wie oft die Noten i und j aufeinander folgen.

▸ Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

Wir füllen die Tabelle, indem wir mit for-Schleifen jeden Eintrag der Tabelle nacheinander durchgehen und abzählen, wie oft die Note in der Spalte auf die Note in der Zeile folgt. Der Code, den wir hierfür verwenden, sieht vereinfacht dargestellt folgendermaßen aus:

```
for i in 1 : Anzahl Zeilen # i durchläuft die Zeilen der Tabelle
    for j in 1 : Anzahl Spalten # j durchläuft die Spalten der Tabelle
        for k in 2 : Anzahl Noten # k durchläuft nacheinander alle Noten
        des Lieds
            if Die Note an der Stelle k-1 entspricht der Note von Zeile i
            und die Note an der Stelle k entspricht der Note von Spalte j
                Dann erhöhe den Eintrag in Zeile i und Spalte j um 1
            end
        end
    end
end
```

Wenn du for-Schleifen und if-Bedingungen noch nicht kennst, kannst du dir den Tipp zu for-Schleifen und den Tipp zu if-Bedingungen anschauen.

Führe das folgende Codefeld aus um die leere Tabelle zu füllen.

In [ ]:
```
#Hier nichts ändern
H = Komposition.absTabelle(einzelneNoten, notenfolge)
```

Schritt 3:
Aus der Übergangstabelle H der absoluten Häufigkeiten soll nun die Übergangstabelle P der relativen Häufigkeiten berechnet werden. Aus dem ersten Arbeitsblatt kennen wir die Formel für die Übergangswahrscheinlichkeit:

$$\mathbb{P}(\text{Note 1} \to \text{Note 2}) = \frac{H(\text{Note 1,Note 2})}{H(\text{Note 1,beliebige Note 1})+H(\text{Note 1,beliebige Note 2})+H(\text{Note 1,beliebige Note 3})+\ldots}$$

Für die Tabelle bedeutet dies, dass jeder Eintrag durch die entsprechende Zeilensumme geteilt werden muss.

Wir erstellen zunächst wieder eine leere Tabelle und durchlaufen diese dann mit Hilfe von for-Schleifen um sie zu füllen.

Ersetze `NaN` mit der Formel für die Berechnung der Übergangswahrscheinlichkeit .

Hinweise:

- `T[i,j]` gibt den Wert in der i-ten Zeile und j-ten Spalte der Tabelle T an.
- Mit `sum(T,dims=2)[i]` wird die i-te Zeilensumme der Tabelle T berechnet.

In [ ]:
```
#Hier nichts ändern
P=fill(1/length(einzelneNoten), (length(einzelneNoten), length(einzelneNoten))); # leere Tabelle

#Ersetze NaN mit der Formel für die Berechnung der Übergangswahrscheinlichkeit
 for i in 1 : zeilen # i durchläuft die Zeilen der Tabelle
     for j in 1 : spalten # j durchläuft die Spalten der Tabelle
        P[i,j] =  NaN; # Formel für die Berechnung der Übergangswahrscheinlichkeit
    end
end

P = Komposition.checkG(H, P, einzelneNoten)
```

c) Automatische Komposition

Die Auswahl der nächsten Note wird jetzt wieder mit Julia ausgeführt.

Wähle eine Note, mit der das Lied anfangen soll und trage sie im Feld "ersteNote" für `NaN` ein. Wähle anschließend die Anzahl der Noten, also wie lange das Lied werden soll und trage sie im Feld "anzahlNoten" für `NaN` ein.

In [ ]:
```
#Ersetze NaN
ersteNote =  NaN;
anzahlNoten =  NaN;

#Hier nichts ändern
notenfolge_neu = Komposition.folge2(P; einzelneNoten, ersteNote, anzahlNoten)
```

Führe das folgende Codefeld aus, um dir das Ergebnis anzuhören.

In [ ]:
```
#Hier nichts ändern
Komposition.createmidi2(notenfolge_neu, anzahlNoten, liedtitel)
```

Was fällt dir im Bezug auf die Länge der Noten auf?

► Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

Teil 4) Modellerweiterung

Im Folgenden werden die für die automatische Komposition berücksichtigten Daten erweitert um ein besseres Ergebnis zu erzielen.

a) Notenlänge wird berücksichtigt

Zunächst berücksichtigen wir zusätzlich zur Notenhöhe auch die Notenlänge. Hierzu berechnen wir analog zur Übergangsmatrix der Notenfolge eine Übergangsmatrix für die Notenlänge und nutzen beide Übergangstabellen um ein neues Lied zu generieren.

Führe das folgende Codefeld aus, um dir das Ergebnis anzuhören.

In [ ]:
```
#Hier nichts ändern
Komposition.folge3(notenfolge_neu, liedtitel, ersteNote, anzahlNoten)
```

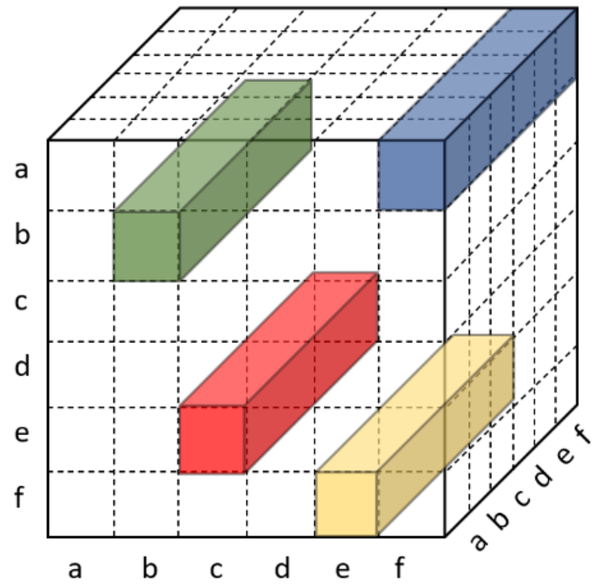
b) Zwei vorherige Noten werden berücksichtigt

Um ein noch besseres Ergebnis zu erhalten, berücksichtigen wir jetzt nicht nur eine vorausgehende Note, sondern zwei.

(1) *Überlege dir welche Form die Übergangstabelle in diesem Fall hat und was die Einträge der Übergangstabelle bedeuten.*

▸ Erst die Frage beantworten, dann hier klicken, um die Lösung zu sehen!

(2) *In der folgenden Übergangstabelle sind verschiedene Wahrscheinlichkeitsverteilungen farblich markiert. Die blaue Reihe zeigt z.B. die Wahrscheinlichkeitsverteilung für die Note, die auf "a" und "f" folgt. Ordne die anderen Verteilungen den Noten zu, indem du die entsprechenden Farben (rot, gelb, grün) im folgenden Codefeld für `NaN` einsetzt.*



In [ ]:

```
# Ersetze NaN, die Anführungszeichen müssen stehen bleiben
#Note 1 folgt auf "a" und "f"
Note_1 = "blau";
#Note 2 folgt auf "e" und "c"
Note_2 =  "NaN";
#Note 3 folgt auf "f" und "e"
Note_3 =  "NaN";
#Note 4 folgt auf "b" und "b"
Note_4 =  "NaN";

#Hier nichts ändern
Komposition.checkH(Note_2, Note_3, Note_4)
```

Wähle die ersten zwei Noten aus, mit denen das Lied anfangen soll und trage sie im Feld "ersteNote" und "zweiteNote" für `NaN` ein. Wähle anschließend die Anzahl der Noten, also wie lange das Lied werden soll und trage sie im Feld "anzahlNoten" für `NaN` ein.

In [ ]:

```
# Ersetze die NaN
ersteNote =  "NaN";
zweiteNote =  "NaN";
anzahlNoten =  NaN;

#Hier nichts ändern
Komposition.folge4(liedtitel, ersteNote, zweiteNote, anzahlNoten)
```

C. Basic Paper

Wie funktioniert eigentlich...
die automatische Komposition von Musik mit KI?

Lektion 1
Motivation

Lektion 2
Algorithmische Kompositionsverfahren

Lektion 3
Algorithmische Komposition mit Markov Ketten

Lektion 1 Motivation

„In schöner Musik steckt meist Mathematik“

Mit diesem Titel warb 2019 eine Ausstellung in Heidelberg zum Thema Zusammenhang zwischen Mathematik und Musik. Bereits die Pythagoräer erkannten harmonische Zahlenverhältnisse für die Musik und zeigten, dass harmonisch klingende Noten mathematischen Regeln gehorchen (vergleiche myScience, 2019). Mit Hilfe solcher mathematischer Regeln, die man aus bekannten Liedern ableiten kann, können Vorhersageregeln für die Komposition neuer Musik entwickelt werden. Eine wesentliche Rolle spielt hierbei die Künstliche Intelligenz, die es ermöglicht bekannte Lieder einzulesen und Vorhersageregeln zu entwickeln. So wurde z.B. Beethovens unvollendete 10. Sinfonie mit Hilfe von KI vollendet (vergleiche FAZ, 2021).

Solche Vorhersageregeln sollen auch in diesem Lernmodul entwickelt werden, um schließlich selbst ein neues Lied komponieren zu können. Die Schülerinnen und Schüler[1] werden hierfür schrittweise durch die mathematische Modellierung eines solchen Vorhersagemodells geführt und lernen somit neben den mathematischen Grundlagen der automatischen Komposition auch das Vorgehen bei der mathematischen Modellierung von Alltagssituationen.

Das vorliegende Paper bietet eine Einführung in die mathematischen und fachlichen Grundlagen der automatischen Komposition von Musik mit Markov Ketten.

[1] Aus Gründen der besseren Lesbarkeit wird im Folgenden auf die gleichzeitige Verwendung weiblicher und männlicher Sprachformen verzichtet und das generische Maskulinum verwendet. Sämtliche Personenbezeichnungen gelten gleichermaßen für beide Geschlechter.

Lektion 1
Motivation

Lektion 2
Algorithmische Kompositionsverfahren

Lektion 3
Algorithmische Komposition mit Markov Ketten

Lektion 2 Algorithmische Kompositionsverfahren

Bereits seit mehreren hundert Jahren beschäftigen sich Musiker und Mathematiker mit Verfahren, die die Komposition musikalischer Stücke vereinfachen oder komplett übernehmen können. Wenn diese Verfahren automatisch ablaufen und durch einen mathematischen Prozess beschrieben werden können, dann spricht man von algorithmischer Komposition.

Geschichtlicher Hintergrund

Die grundsätzlichen Methoden zur Entwicklung algorithmischer Kompositionsverfahren sind heute noch die gleichen wie früher. Durch Analysieren von Notenverläufen, harmonischen Strukturen und rythmischen Mustern kann man die Strukturen von Musik mathematisch erfassen. Die folgende Auflistung zeigt einen groben Überblick über den historischen Verlauf der entwickelten Verfahren (vergleiche Roads, 1996; Nierhaus, 2009).

1026: Der Benediktinermönch und Musiktheoretiker Guido von Arezzo entwickelt ein Verfahren zur automatischen Generierung von Melodien aus Textmaterial. Hierbei werden Buchstaben und Silben auf Tonhöhen und melodische Phrasen abgebildet, siehe Abbildung 1. Guido von Arezzo gilt deshalb auch als Vater der Solmisation.

1650: Der Jesuit und Universalgelehrte Athanasius Kircher entwickelt mit der Komponiermaschine eine mechanisches System zur automatischen Komposition. Die Komponiermaschine besteht aus Täfelchen mit vierzeiligen Zahlenspalten, wobei die Zahlen jeweils Tönen zugeordnet sind. Die Benutzer entnehmen dem Versmaß des Textes entsprechende Tafeln und fügen willkürlich gewählte Segmente aneinander. Die Repräsentation von Tonhöhen durch Zahlen war ein deutlicher Fortschritt für die Klasse der möglichen Kompositionen. Des Weiteren zeigt das von Kircher angewendete Prinzip eine Anwendung von Tonklassen, die seit dem 20. Jahrhundert eine wesentliche Rolle in der Musikproduktion und -analyse spielen.

18. Jhdt: Zum Ende des 18. Jahrhunderts entwickeln verschiedene Komponisten und Musiktheoretiker musikalische Würfelspiele. In einer Tabelle stehen vorkomponierte Takte, durch Würfeln oder Kartenziehen wird eine Zufallszahl erzeugt, die eine Zeile aus einer Tabelle bestimmt. Die Spalten der Tabelle werden nach der Reihenfolge der Würfe ausgewählt. Mit dieser Methode wurden meistens Walzer, Polonaisen oder Menuetten komponiert. Das bekannteste Spiel stammt von Wolfgang Amadeus Mozart mit seiner „Anleitung zum Componieren von Walzern vermittels zweier Würfel“, siehe Abbildung 2.

1957: Die amerikanischen Komponisten Lejaren Hiller und Leonard Issacson programmierten den ILLIAC I Computer, der die Illiac Suite komponierte, das erste vollständig von einem Computer geschriebene Stück.

1960: Der russische Wissenschaftler R. Kh . Zaripov veröffentlicht die erste Arbeit über algorithmische Komposition mithilfe eines Computers.

1980: Der amerikanische Wissenschaftler und Komponist David Cope entwickelt die Software EMI (Experiments in Musical Intelligence). Das System analysiert existierende Musikstücke und generiert darauf basierende neue Musikstücke.

Lektion 3 Algorithmische Komposition mit Markov Ketten

Zunächst werden in diesem Abschnitt Markov Ketten allgemein betrachtet. Anschließend wird näher auf die Musikkomposition mit Markov Ketten eingegangen.

Lektion 3.1 Markov Ketten

Markov Ketten, nach Andrei A. Markov (1856–1922), stellen einen stochastischen Prozess dar, der dadurch charakterisiert ist, dass das zukünftige Verhalten des Prozesses nur vom aktuellen Zustand abhängt und nicht von der Vergangenheit. Dies ermöglicht die Modellierung dynamischer Systeme in verschiedenen Gebieten, wie z.B. in der Produktionsplanung, beim Qualitätsmanagement oder in der Informationsverarbeitung. Die folgende Beschreibung basiert auf Aldous and Fill (2014), Norris (1997) und Kersting and Wakolbinger (2011).

Definition .1. *Eine Folge* $X = \{X_k\}_{k\in\mathbb{N}}$ *von Zufallsvariablen* $X_k : \Omega \to S$ *heißt* stochastischer Prozess *im Zustandsraum S.*

Die Zufallsvariable X_k beschreibt den Zustand des Systems zum Zeitpunkt k, dies kann z.B. ein Lagerbestand oder die Größe einer Population sein.

Definition .2. *Ein stochastischer Prozess* $X = \{X_k\}_{k\in\mathbb{N}}$ *in einem endlichen oder abzählbar unendlichen Zustandsraum S heißt* Markov Kette erster Ordnung*, wenn die Markovbedingung*

$$\mathbb{P}(X_{k+1} = i_{k+1}|X_0 = i_0, ..., X_k = i_k) = \mathbb{P}(X_{k+1} = i_{k+1}|X_k = i_k) = p_{i_k i_{k+1}}$$

für alle $k \in \mathbb{N}$ *und für alle Zustände* $i_k \in S$ *gilt. Die* p_{ij} *heißen* Übergangswahrscheinlichkeiten *von Zustand i auf Zustand j.*

Die Wahrscheinlichkeitsverteilung des Zustands zur Zeit $k+1$ ist also nur vom Zustand zur Zeit k abhängig, jedoch nicht von den vorherigen Zuständen $k' < k$.
Die Übergangswahrscheinlichkeiten werden üblicherweise in einer stochastischen Matrix, der Übergangsmatrix, dargestellt.

Definition .3. *Eine Matrix* $P = (p_{ij})_{i,j\in S}$ *heißt* stochastische Matrix*, falls* $p_{ij} \geq 0$ *ist und für alle* $i \in S$ *die Zeilensumme* $\sum_{j\in S} p_{ij} = 1$ *ist.*

Analog zur Markov Kette erster Ordnung können auch Markov Ketten höherer Ordnung betrachtet werden.

Definition .4. *Ein stochastischer Prozess* $X = \{X_k\}_{k\in\mathbb{N}}$ *in einem endlichen oder abzählbar unendlichen Zustandsraum S heißt* Markov Kette n-ter Ordnung*, wenn die Markovbedingung*

$$\mathbb{P}(X_{k+1} = i_{k+1}|X_0 = i_0, ..., X_k = i_k) = \mathbb{P}(X_{k+1} = i_{k+1}|X_{k-n+1} = i_{k-n+1}, ..., X_k = i_k)$$

für alle $k \in \mathbb{N}$ *und für alle Zustände* $i_k \in S$ *gilt.*

Die Wahrscheinlichkeitsverteilung des Zustands zur Zeit $k+1$ ist also von den n vorherigen Zuständen abhängig.
In den folgenden Beispielen werden nur Markov Ketten erster Ordnung berücksichtigt.

Lektion 1
Motivation

Lektion 2
Algorithmische Kompositionsverfahren

Lektion 3
Algorithmische Komposition mit Markov Ketten

Lektion 3.2 Musikkomposition mit Markov Ketten

Markov Ketten für die algorithmische Komposition zu verwenden ist eine sehr verbreitete und gut geeignete Methode. Die Noten definieren den Zustandsraum S und die Übergangsmatrix kann entsprechend der Notenübergänge aus einer Beispielsequenz erzeugt werden.

Schritt 1: Abzählen der Notenübergänge

Die Übergangswahrscheinlichkeiten werden mit der Maximum-Likelihood-Methode berechnet. Hierfür wird im ersten Schritt die Häufigkeit h_{ij}, also das Auftreten von Zustand $j \in S$ zum Zeitpunkt k, wenn Zustand $i \in S$ zum Zeitpunkt $(k-1)$ aufgetreten ist, für alle $i, j \in S$ und alle k aus der Beispielsequenz, abgezählt. Die so berechneten Häufigkeiten h_{ij} definieren die Matrix der absoluten Häufigkeiten H.

Beispiel .5. *Für die Beispielsequenz in Abbildung 3 ergibt sich die Matrix H.*

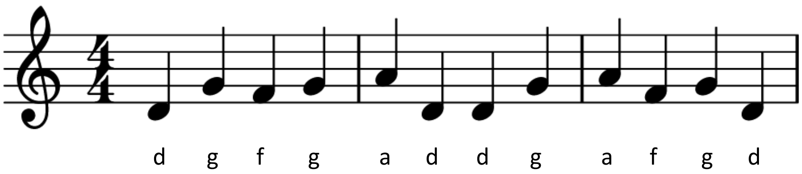


Abbildung 3: Beispielsequenz

$$H = \begin{array}{c c} & \begin{array}{cccc} a & d & f & g \end{array} \\ \begin{array}{c} a \\ d \\ f \\ g \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 2 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Schritt 2: Berechnung der Übergangsmatrix

Aus dieser Matrix kann im zweiten Schritt die Übergangsmatrix $P = p_{ij}$ berechnet werden. Für die einzelnen Einträge gilt:

$$p_{ij} = \frac{h_{ij}}{\sum\limits_{k \in S} h_{ik}} \tag{1}$$

Beispiel .6. *Mit Hilfe der Formel 1 ergibt sich aus der Matrix H die Übergangsmatrix P.*

$$P = \begin{array}{c c} & \begin{array}{cccc} a & d & f & g \end{array} \\ \begin{array}{c} a \\ d \\ f \\ g \end{array} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 1/3 & 0 & 2/3 \\ 0 & 0 & 0 & 1 \\ 1/2 & 1/4 & 1/4 & 0 \end{pmatrix} \end{array}$$

Lektion 1 Motivation
Lektion 2 Algorithmische Kompositionsverfahren
Lektion 3 Algorithmische Komposition mit Markov Ketten

Schritt 3: Erzeugung einer neuen Sequenz

Für die Erzeugung einer neuen Sequenz kann eine beliebige Note aus dem Zustandsraum S als Anfangsnote ausgewählt werden. Die zweite Note wird über die durch die erste Note gegebene Wahrscheinlichkeitsverteilung bestimmt. Die dritte Note wird dementsprechend über die durch die zweite Note gegebene Wahrscheinlichkeitsverteilung bestimmt. Auf diese Weise entsteht eine neue Sequenz mit beliebiger Länge.

Beispiel .7. *Als Anfangszustand wird die Note „a" ausgewählt, entsprechend ist die Wahrscheinlichkeitsverteilung* $(0, 1/2, 1/2, 0)$ *für die zweite Note* (a, d, f, g) *gegeben. Ein von der Verteilung abhängiger Zufallswert bestimmt die Note „f" als zweiten Zustand. Entsprechend der Übergangsmatrix kann der dritte Zustand nur die Note „g" sein. Auf diese Weise ist die Sequenz in Abbildung 4 entstanden.*

Abbildung 4: Beispiel für eine neu erzeugte Sequenz

Lektion 3.3 Computergestützte Komposition mit Markov Ketten

Für größere Beispielsequenzen als in Beispiel .5 ist ein computergestütztes Einlesen der Trainingsdaten sinnvoll. Die Trainingsdaten werden hier als MIDI Dateien eingelesen und mit JuliaMusic analysiert. Die Berechnung der Markov Kette erfolgt nun analog zu den in 3.2 beschriebenen Schritten.

MIDI Technologie

MIDI steht für Musical Instrument Digital Interface, oder auf deutsch „Digitale Schnittstelle für Musikinstrumente", und wurde bereits in den frühen 80er Jahren entwickelt und wird auch heute noch weiterentwickelt. Es ist das in der Musikindustrie übliche Technologieprotokoll, das die Kommunikation zwischen Computern, Instrumenten und anderer Hardware ermöglicht. MIDI Dateien übertragen dabei keine eigentlichen Audiosignale, sondern nur Daten in Form von Anweisungen, wie eine Maschine die MIDI-Note wiedergeben soll (vergleiche Stotz, 2019). Jede MIDI-Note enthält Anweisungen mit den folgenden Informationen:

- *Note-on / Note-off*: Von wann bis wann die Note gespielt wird
- *Note*: Welche Note gespielt wird
- *Velocity*: Anschlagstärke

JuliaMusic

JuliaMusic ist eine Sammlung von Paketen, die in der Programmiersprache Julia geschrieben sind und für das Lesen, Manipulieren und Speichern von Musikdaten verwendet werden können. Für die hier benötigten Funktionen des Lesens und Schreibens der MIDI Dateien wird insbesondere das Paket MIDI.jl verwendet. Dieses Paket ermöglicht es, die Rohdaten der MIDI Dateien intuitiv und für Menschen lesbar darzustellen (vergleiche Datseris and Hobson, 2019). Abbildung 5 zeigt die Darstellung einer eingelesenen MIDI Datei mit MIDI.jl.

Lektion 1
Motivation

Lektion 2
Algorithmische Kompositionsverfahren

Lektion 3
Algorithmische Komposition mit Markov Ketten

```
291 Notes with tpq=480
 Note B3  | vel = 122 | pos = 39840, dur = 195 | channel 3
 Note E4  | vel = 122 | pos = 40080, dur = 216 | channel 3
 Note F♯4 | vel = 122 | pos = 40320, dur = 413 | channel 3
 Note B4  | vel = 122 | pos = 40800, dur = 425 | channel 3
 Note B4  | vel = 122 | pos = 41280, dur = 429 | channel 3
 Note B4  | vel = 122 | pos = 41760, dur = 727 | channel 3
 Note B3  | vel = 122 | pos = 43919, dur = 216 | channel 3
  ⋮
 Note E4  | vel = 122 | pos = 313440, dur = 240 | channel 3
 Note B3  | vel = 122 | pos = 313680, dur = 437 | channel 3
 Note B3  | vel = 122 | pos = 314160, dur = 240 | channel 3
 Note B3  | vel = 122 | pos = 314760, dur = 107 | channel 3
 Note E4  | vel = 122 | pos = 314880, dur = 219 | channel 3
 Note F♯4 | vel = 122 | pos = 315120, dur = 108 | channel 3
 Note F♯4 | vel = 122 | pos = 315240, dur = 2519 | channel 3
```

Abbildung 5: Mit MIDI.jl eingelesene MIDI Datei

Lektion 3.4 Bedingte Wahrscheinlichkeit

Betrachtet man einen einzelnen Übergang einer Markov Kette vom Zustand i in den Zustand j, dann ist die Übergangswahrscheinlichkeit p_{ij} die bedingte Wahrscheinlichkeit sich zum Zeitpunkt $(k+1)$ in Zustand j zu befinden, gegeben, dass man zum Zeitpunkt k in Zustand i ist.

Definition .8. *Es seien* $(\Omega, \mathbb{P})$ *ein endlicher Wahrscheinlichkeitsraum und* $A \subset \Omega$ *mit* $\mathbb{P}(A) > 0$. *Dann heißt*

$$\mathbb{P}_A(B) := \mathbb{P}(B|A) := \frac{\mathbb{P}(\mathbb{B} \cap \mathbb{A})}{\mathbb{P}(A)}, \quad B \subset \Omega$$

die bedingte Wahrscheinlichkeit von B unter der Bedingung A. *(nach Henze, 2016)*

Lektion 1
Motivation

Lektion 2
Algorithmische Kompositionsverfahren

Lektion 3
Algorithmische Komposition mit Markov Ketten

Lektion Literatur

mySCience. *In schöner Musik steckt meist Mathematik*. `https://www.myscience.de/news/wire/in_schoener_musik_steckt_meist_mathematik-2019-tum`, 2019. letzter Aufruf: 30.10.2021.

FAZ. *Künstliche Intelligenz vollendet die 10. Sinfonie von Beethoven*. `https://www.faz.net/aktuell/feuilleton/buehne-und-konzert/von-ki-vollendete-sinfonie-von-beethoven-uraufgefuehrt-17577854.html`, 2021. letzter Aufruf: 30.10.2021.

Curtis Roads. *The Computer Music Tutorial*. MIT Press, Cambridge, 1996. ISBN 978-0-262-68082-0.

Gerhard Nierhaus. *Algorithmic Composition - Paradigms of Automated Music Generation*. Springer Science & Business Media, Berlin Heidelberg, 2009. ISBN 978-3-211-75540-2.

David Aldous & James Allen Fill. *Reversible Markov Chains and Random Walks on Graphs*. `https://www.stat.berkeley.edu/~aldous/RWG/book.pdf`, 2014. letzter Aufruf: 15.09.2021.

J. R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. doi: 10.1017/CBO9780511810633.

Götz Kersting & Anton Wakolbinger. *Elementare Stochastik*. Springer-Verlag, Berlin Heidelberg New York, 2011. ISBN 978-3-034-60414-7.

Dieter Stotz. *Computergestützte Audio- und Videotechnik - Multimediatechnik in der Anwendung*. Springer-Verlag, Berlin Heidelberg New York, 2019. ISBN 978-3-662-58873-4.

George Datseris & Joel Hobson. MIDI.jl: Simple and intuitive handling of midi data. *The Journal of Open Source Software*, 4(35):1166, mar 2019. doi: 10.21105/joss.01166. URL `https://doi.org/10.21105/joss.01166`.

Norbert Henze. *Stochastik für Einsteiger - Eine Einführung in die faszinierende Welt des Zufalls*. Springer-Verlag, Berlin Heidelberg New York, 2016. ISBN 978-3-658-14739-6.

Lektion 3 | Folie 5 **Seite 8**

D. Methodisches Konzept

Methodisches Konzept
für die Dozenten

Material:

Präsentationen:

- Folien_Einführung.pdf
- Notizen_Einführung.pdf
- Folien_Zwischen.pdf
- Notizen_Zwischen.pdf
- Folien_Abschluss.pdf
- Notizen_Abschluss.pdf

Jupyter Notebooks Ordner:

- code
- data
- figs
- help
- worksheets

Dauer:
120 Minuten mit 2 Arbeitsphasen und 3 Plenumsdiskussionen

Ziel:
Die Schüler lernen den Zusammenhang zwischen Mathematik und Musik kennen und entwickeln ein Vorhersagemodell für Notenübergänge, das für die automatische Komposition von Musik genutzt wird.

Verlaufsplan:

Phase	Dauer	Inhalt
Begrüßung und Einstiegspräsentation	20 Minuten	• Begrüßung • Vorstellungsrunde • Einführung in die Problemstellung • Einführung in die mathematische Modellierung
Erste Arbeitsphase	40 Minuten	• Bearbeitung 1. Arbeitsblatt
Zwischenpräsentation	10 Minuten	• Besprechung der einzelnen Schritte der Modellierung am Modellierungskreislauf • Einführung des MIDI Standards
Zweite Arbeitsphase	35 Minuten	• Bearbeitung 2. Arbeitsblatt
Abschlusspräsentation	15 Minuten	• Zusammenfassung der Ergebnisse • Evaluation • Verabschiedung

E. Evaluation

E.1. Fragebogen

Workshop Automatische Komposition

Seite 1

Liebe Schülerin, lieber Schüler,

vielen Dank für die Teilnahme an dem Workshop, den ich im Rahmen meiner Masterarbeit am KIT entwickelt habe. Um den Workshop noch weiter zu verbessern, würde ich gerne deine Meinung erfahren, wie dir der Workshop gefallen hat.
Die Umfrage ist natürlich anonym.

Vielen Dank für deine Rückmeldung
Philipp

Seite 2

Geschlecht

- ◯ weiblich
- ◯ männlich
- ◯ divers

Schulform *

- ◯ Hauptschule
- ◯ Realschule
- ◯ Gymnasium
- ◯ Gemeinschaftsschule
- ◯ andere Schulform

Jahrgangsstufe *

- ◯ 10
- ◯ 11
- ◯ 12
- ◯ 13
- ◯ andere

Kurswahl *

- [ ] Mathe Grundkurs
- [ ] Mathe Leistungskurs
- [ ] Musik Grundkurs
- [ ] Musik Leistungskurs

Wie hast du von dem Kursangebot erfahren? *

- ○ Lehrer*innen
- ○ Freunde
- ○ Eltern
- ○ Internet
- ○ Sonstiges

Seite 3

Schwierigkeitsgrad *

	trifft gar nicht zu	trifft eher nicht zu	trifft zum Teil zu	trifft voll zu		
Ich konnte die Aufgaben eigenständig lösen.	○	○	○	○	○	○
Mein mathematisches Vorwissen war ausreichend, um die Aufgaben zu bearbeiten.	○	○	○	○	○	○
Die Aufgaben waren insgesamt zu schwierig.	○	○	○	○	○	○
Die Aufgaben waren insgesamt zu einfach.	○	○	○	○	○	○
Ich habe des Öfteren Tipps zu den Aufgaben benötigt.	○	○	○	○	○	○
Der Umgang mit Julia fiel mir schwer.	○	○	○	○	○	○

Gestaltung *

	trifft gar nicht zu	trifft eher nicht zu	trifft zum Teil zu	trifft voll zu		
Neue Inhalte wurden in den Präsentationen gut erklärt.	○	○	○	○	○	○
Die Aufgabenstellungen waren klar und verständlich.	○	○	○	○	○	○
Die Aufgaben waren abwechslungsreich.	○	○	○	○	○	○
Die Hilfekarten waren hilfreich.	○	○	○	○	○	○
Ich hatte zwischen den Aufgabenblättern nichts zu tun.	○	○	○	○	○	○
Die Diskussionsphasen im Plenum haben mir beim Verständnis geholfen.	○	○	○	○	○	○

Motivation *

	trifft gar nicht zu	trifft eher nicht zu	trifft zum Teil zu	trifft voll zu		
Ich beschäftige mich in meiner Freizeit mit dem Thema Musik.	◯	◯	◯	◯	◯	◯
Ich spiele ein Instrument.	◯	◯	◯	◯	◯	◯
Ich habe mich schon einmal mit dem Thema Musikkomposition beschäftigt.	◯	◯	◯	◯	◯	◯
Ich habe vor dem Workshop schon von automatischer Komposition gehört.	◯	◯	◯	◯	◯	◯
Der Workshop hat mein Interesse an dem Thema gesteigert.	◯	◯	◯	◯	◯	◯
Ich würde diesen Workshop anderen weiterempfehlen.	◯	◯	◯	◯	◯	◯

Seite 4

Hat dir etwas an diesem Workshop absolut nicht gefallen?

Hat dir etwas an diesem Workshop besonders gut gefallen?

Hättest du gerne noch etwas anderes gesehen oder erfahren?

Ich gebe dem Workshop die Schulnote *

- ◯ 1 (sehr gut)
- ◯ 2 (gut)
- ◯ 3 (befriedigend)
- ◯ 4 (ausreichend)
- ◯ 5 (mangelhaft)
- ◯ 6 (ungenügend)

Ich gebe dem Workshopleiter die Schulnote *

- ◯ 1 (sehr gut)
- ◯ 2 (gut)
- ◯ 3 (befriedigend)
- ◯ 4 (ausreichend)
- ◯ 5 (mangelhaft)
- ◯ 6 (ungenügend)

» **Umleitung auf Schlussseite von Umfrage Online**

E.2. Ergebnis der Evaluation

Workshop Automatische Komposition

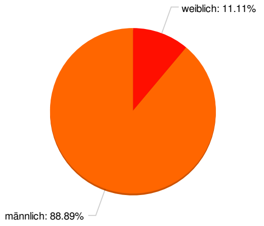
1. Geschlecht

Anzahl Teilnehmer: 9

1 (11.1%): weiblich

8 (88.9%): männlich

- (0.0%): divers



2. Schulform *

Anzahl Teilnehmer: 9

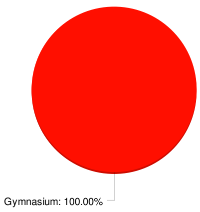
- (0.0%): Hauptschule

- (0.0%): Realschule

9 (100.0%): Gymnasium

- (0.0%): Gemeinschaftsschule

- (0.0%): Andere



3. Jahrgangsstufe *

Anzahl Teilnehmer: 9

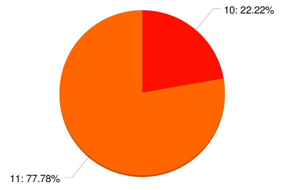
2 (22.2%): 10

7 (77.8%): 11

- (0.0%): 12

- (0.0%): 13

- (0.0%): Andere



4. Kurswahl *

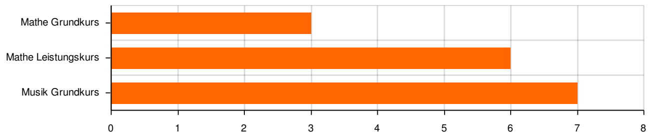
Anzahl Teilnehmer: 9

3 (33.3%): Mathe Grundkurs

6 (66.7%): Mathe Leistungskurs

7 (77.8%): Musik Grundkurs

- (0.0%): Musik Leistungskurs



5. Wie hast du von dem Kursangebot erfahren? *

Anzahl Teilnehmer: 9

2 (22.2%): Lehrer*innen

2 (22.2%): Freunde

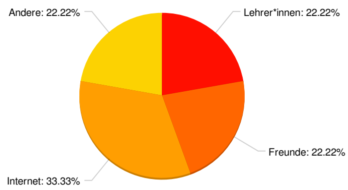
- (0.0%): Eltern

3 (33.3%): Internet

2 (22.2%): Andere

Antwort(en) aus dem Zusatzfeld:

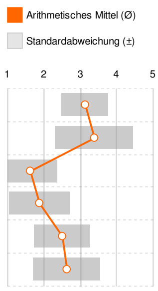
- KIT
- Lern-Fair



6. Schwierigkeitsgrad *

Anzahl Teilnehmer: 8

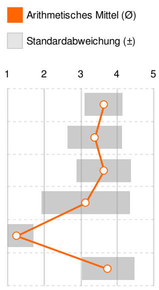
	trifft gar nicht zu (1)		trifft eher nicht zu (2)		trifft zum Teil zu (3)		trifft voll zu (4)		5. Spalte (5)		nicht beurteilbar (0)		
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Σ	Ø	±
Ich konnte die Aufgaben ...	-	-	1x	12,50	5x	62,50	2x	25,00	-	-	-	3,13	0,64
Mein mathematisches Vo...	1x	12,50	-	-	2x	25,00	5x	62,50	-	-	-	3,38	1,06
Die Aufgaben waren insg...	4x	50,00	3x	37,50	1x	12,50	-	-	-	-	-	1,63	0,74
Die Aufgaben waren insg...	3x	37,50	3x	37,50	2x	25,00	-	-	-	-	-	1,88	0,83
Ich habe des Öfteren Tip...	1x	12,50	2x	25,00	5x	62,50	-	-	-	-	-	2,50	0,76
Der Umgang mit Julia fie...	1x	12,50	2x	25,00	4x	50,00	1x	12,50	-	-	-	2,63	0,92



7. Gestaltung *

Anzahl Teilnehmer: 8

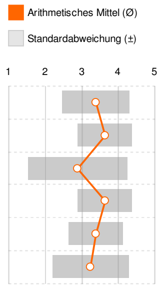
	trifft gar nicht zu (1)		trifft eher nicht zu (2)		trifft zum Teil zu (3)		trifft voll zu (4)		5. Spalte (5)		nicht beurteilbar (0)		
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Σ	Ø	±
Neue Inhalte wurden in d...	-	-	-	-	3x	37,50	5x	62,50	-	-	-	3,63	0,52
Die Aufgabenstellungen w..	-	-	1x	12,50	3x	37,50	4x	50,00	-	-	-	3,38	0,74
Die Aufgaben waren abwe..	-	-	1x	12,50	1x	12,50	6x	75,00	-	-	-	3,63	0,74
Die Hilfekarten waren hi...	1x	12,50	-	-	4x	50,00	1x	12,50	1x	12,50	1x	3,14	1,21
Ich hatte zwischen den A...	6x	75,00	2x	25,00	-	-	-	-	-	-	-	1,25	0,46
Die Diskussionsphasen im..	-	-	-	-	3x	37,50	4x	50,00	1x	12,50	-	3,75	0,71



8. Motivation *

Anzahl Teilnehmer: 8

	trifft gar nicht zu (1)		trifft eher nicht zu (2)		trifft zum Teil zu (3)		trifft voll zu (4)		5. Spalte (5)		nicht beurteilbar (0)		
	Σ	%	Σ	%	Σ	%	Σ	%	Σ	%	Σ	Ø	±
Ich beschäftige mich in m..	-	-	2x	25,00	1x	12,50	5x	62,50	-	-	-	3,38	0,92
Ich spiele ein Instrument.	-	-	1x	12,50	1x	12,50	6x	75,00	-	-	-	3,63	0,74
Ich habe mich schon einm.	2x	25,00	1x	12,50	1x	12,50	4x	50,00	-	-	-	2,88	1,36
Ich habe vor dem Worksh...	-	-	1x	12,50	1x	12,50	6x	75,00	-	-	-	3,63	0,74
Der Workshop hat mein I...	-	-	1x	12,50	3x	37,50	4x	50,00	-	-	-	3,38	0,74
Ich würde diesen Worksh...	-	-	3x	37,50	-	-	5x	62,50	-	-	-	3,25	1,04



9. Hat dir etwas an diesem Workshop absolut nicht gefallen?

Anzahl Teilnehmer: 6

- Julia war nicht immer leicht zu verstehen
- Ich hatte anfangs Schwierigkeiten zu meinem Arbeitsblatt zu kommen.
- Funktionalität von Jitsi und vorallem jupyterhub.
  Julia (lieber Java oder C++, notfalls Pyphon), Benutzungsfreundlichkeit der Arbeitsblätter und Funktionalität
- Die Hälfte der Zeit ging erstmal wegen technischen Problemen drauf
- Ich hätte gerne am Anfang bei der Präsentation ein bisschen mehr Zeit zum Reinkommen gehabt.
- Die technischen Probleme.

10. Hat dir etwas an diesem Workshop besonders gut gefallen?

Anzahl Teilnehmer: 7

- die interaktiven Arbeitsblätter waren cool, wenn auch teilweise schwierig zu bedienen.
- Interaktive Aufgabenstellungen, Erarbeitung in Gruppen
- hätte auf jeden Fall was werden können, der Inhalt der Blätter und das Thema waren sehr interessant
- Das Thema, ganz klar, es hat mir als Neuling in dem Gebiet gezeigt, wie man mit dem Computer Musik generieren kann, das war sehr interessant, natürlich auch n bisschen enttäuschend, weil man hohe Erwartungen hat.
- Mir hat gut gefallen, dass wir die Aufgaben in Gruppen und mit persönlichem Austausch bearbeiten konnten.
- Das Lied mit den Beatles.
- Wie offen alle miteinander umgegangen sind und wir kooperieren mussten, um die Aufgaben zu schaffen.

11. Hättest du gerne noch etwas anderes gesehen oder erfahren?

Anzahl Teilnehmer: 2

- -
- Weitere Beispiele.

12. Ich gebe dem Workshop die Schulnote *

Anzahl Teilnehmer: 8

4 (50.0%): 1 (sehr gut)

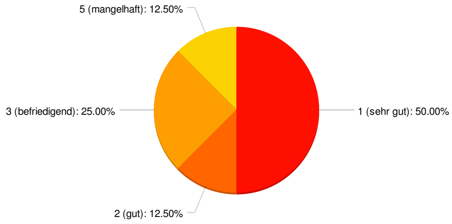
1 (12.5%): 2 (gut)

2 (25.0%): 3 (befriedigend)

- (0.0%): 4 (ausreichend)

1 (12.5%): 5 (mangelhaft)

- (0.0%): 6 (ungenügend)



13. Ich gebe dem Workshopleiter die Schulnote *

Anzahl Teilnehmer: 8

5 (62.5%): 1 (sehr gut)

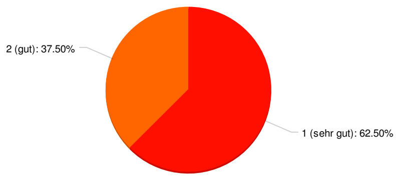
3 (37.5%): 2 (gut)

- (0.0%): 3 (befriedigend)

- (0.0%): 4 (ausreichend)

- (0.0%): 5 (mangelhaft)

- (0.0%): 6 (ungenügend)



Literaturverzeichnis

Rita Borromeo Ferri, Gilbert Greefrath, & Gabriele Kaiser. *Mathematisches Modellieren für Schule und Hochschule - Theoretische und didaktische Hintergründe*. Springer-Verlag, Berlin Heidelberg New York, 2013. ISBN 978-3-658-01580-0.

Corinna Hankeln. *Mathematisches Modellieren mit dynamischer Geometrie-Software - Ergebnisse einer Interventionsstudie*. Springer-Verlag, Berlin Heidelberg New York, 2018. ISBN 978-3-658-23339-6.

Martin Frank, Pascal Richter, Christina Roeckerath, & Sarah Schönbrodt. *Wie funktioniert eigentlich GPS? – ein computergestützter Modellierungsworkshop*, pages 137–163. Springer Fachmedien Wiesbaden, Wiesbaden, 2018. ISBN 978-3-658-21940-6.

Gerhard Nierhaus. *Algorithmic Composition - Paradigms of Automated Music Generation*. Springer Science & Business Media, Berlin Heidelberg, 2009. ISBN 978-3-211-75540-2.

Curtis Roads. *The Computer Music Tutorial*. MIT Press, Cambridge, 1996. ISBN 978-0-262-68082-0.

myScience. *In schöner Musik steckt meist Mathematik*. https://www.myscience.de/news/wire/in_schoener_musik_steckt_meist_mathematik-2019-tum, 2019. letzter Aufruf: 30.10.2021.

FAZ. *Künstliche Intelligenz vollendet die 10. Sinfonie von Beethoven*. https://www.faz.net/aktuell/feuilleton/buehne-und-konzert/von-ki-vollendete-sinfonie-von-beethoven-uraufgefuehrt-17577854.html, 2021. letzter Aufruf: 30.10.2021.

CAMMP. *Die Idee hinter CAMMP*. https://www.cammp.online/85.php, 2021a. letzter Aufruf: 15.09.2021.

CAMMP. *CAMMP day - mathematischer Modellierungstag.* https://www.cammp.online/116.php, 2021b. letzter Aufruf: 15.09.2021.

CAMMP. *CAMMP week - Modellierungswoche.* https://www.cammp.online/21.php, 2021c. letzter Aufruf: 15.09.2021.

Gilbert Greefrath. *Anwendungen und Modellieren im Mathematikunterricht - Didaktische Perspektiven zum Sachrechnen in der Sekundarstufe.* Springer-Verlag, Berlin Heidelberg New York, 2018. ISBN 978-3-662-57680-9.

Kultusministerkonferenz. *Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife.* https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2012/2012_10_18-Bildungsstandards-Mathe-Abi.pdf, 2012. letzter Aufruf: 20.10.2021.

Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan Mathematik.* https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/GYM/M, 2016. letzter Aufruf: 20.10.2021.

Friedrich Zech. *Grundkurs Mathematikdidaktik : theoretische und praktische Anleitungen für das Lehren und Lernen von Mathematik.* Beltz Pädagogik. Beltz, Weinheim, 10. aufl. edition, 2002. ISBN 978-3-407-25216-6.

Deutsches Zentrum für Lehrerbildung. *Prinzip der minimalen Hilfe.* https://pikas-mi.dzlm.de/node/347. letzter Aufruf: 24.10.2021.

Hans-Dieter Sill. *Grundkurs Mathematikdidaktik.* Ferdinand Schöningh, Parderborn, München, 2019. ISBN 978-3-825-25008-9.

David Aldous & James Allen Fill. *Reversible Markov Chains and Random Walks on Graphs.* https://www.stat.berkeley.edu/~aldous/RWG/book.pdf, 2014. letzter Aufruf: 15.09.2021.

J. R. Norris. *Markov Chains.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. doi: 10.1017/CBO9780511810633.

Götz Kersting & Anton Wakolbinger. *Elementare Stochastik.* Springer-Verlag, Berlin Heidelberg New York, 2011. ISBN 978-3-034-60414-7.

Dieter Stotz. *Computergestützte Audio- und Videotechnik - Multimediatechnik in der Anwendung*. Springer-Verlag, Berlin Heidelberg New York, 2019. ISBN 978-3-662-58873-4.

George Datseris & Joel Hobson. MIDI.jl: Simple and intuitive handling of midi data. *The Journal of Open Source Software*, 4(35):1166, mar 2019. doi: 10.21105/joss.01166. URL https://doi.org/10.21105/joss.01166.

Norbert Henze. *Stochastik für Einsteiger - Eine Einführung in die faszinierende Welt des Zufalls*. Springer-Verlag, Berlin Heidelberg New York, 2016. ISBN 978-3-658-14739-6.

Wikipedia. *Julia (Programmiersprache)*. https://de.wikipedia.org/wiki/Julia_(Programmiersprache), 2021a. letzter Aufruf: 03.10.2021.

Wikipedia. *Project Jupyter*. https://de.wikipedia.org/wiki/Project_Jupyter, 2021b. letzter Aufruf: 03.10.2021.

Orchestra of the Age of Enlightenment. *Bach vs AI: spot the difference*. https://www.youtube.com/watch?v=lv9W7qrYhbk, 2019. letzter Aufruf: 18.10.2021.

Sony CSL. *Daddy's Car: a song composed with Artificial Intelligence - in the style of the Beatles*. https://www.youtube.com/watch?v=LSHZ_b05W7o, 2016. letzter Aufruf: 18.10.2021.

Erklärung

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 02.11.2021