

INSTITUT FÜR INFORMATIK
der Ludwig-Maximilians-Universität München

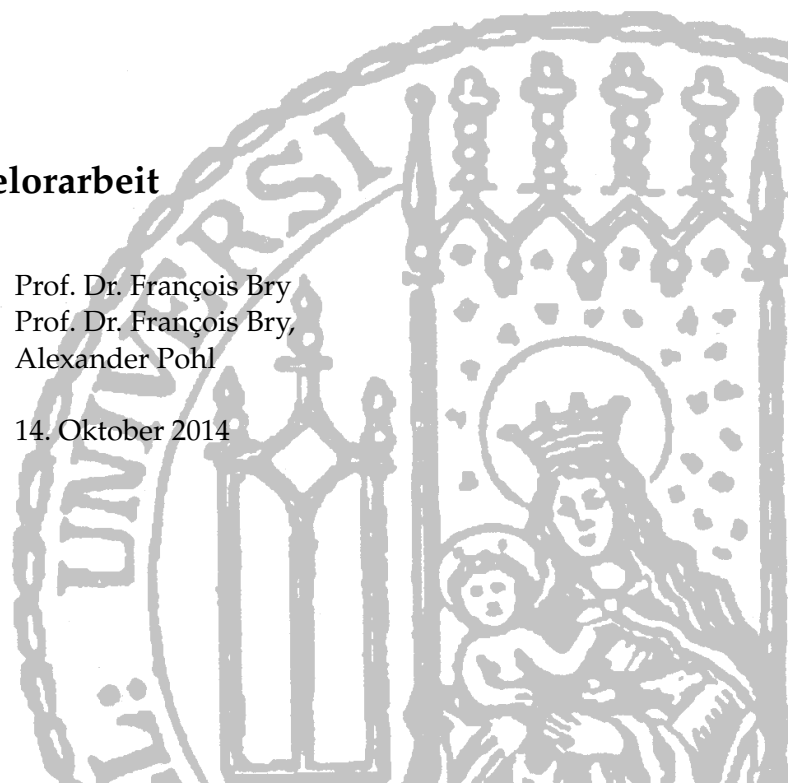
ENTWICKLUNG EINES
STOCHASTISCHEN VERFAHRENS
ZUR NACHVERFOLGUNG
STUDENTISCHER QUIZLEISTUNGEN
IN BACKSTAGE

Werner Hoffmann

Bachelorarbeit

Aufgabensteller Prof. Dr. François Bry
Betreuer Prof. Dr. François Bry,
Alexander Pohl

Abgabe am 14. Oktober 2014



Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit nach einer Idee von Alexander Pohl selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe

München, den 14. Oktober 2014

Werner Hoffmann

Abstract

Students in lectures today are in an environment where little feedback is given. Technology is already used by some Universities to support their lectures, displaying presentations and offering interaction platforms to Students. Is it possible to use this existing infrastructure to also offer feedback to individual Students?

We put this question to the test using „Backstage“ software. Backstage is a back-channel for lectures used by the LMU for showing slides, providing question and answer applications to students, and also offering the Lecturer the ability to set quizzes for the students. This thesis investigates how Backstage could be extended with a feedback function based upon the activity of students on Backstage. The thesis investigates issues with the existing data, that could influence the quality of the feedback that the students receive. We evaluate if the Bayesian Knowledge Tracing, a well-known technique in Educational Data Mining, is suitable for a feedback system in Backstage. We worked on a concept that is based on Bayesian Knowledge Tracing and the used Hidden Markov Model is modified.

The parameter found necessary are described in order to offer reliable feedback, and how each parameter should be chosen. A prototype was developed in order to test the results of specific chosen parameters.

Zusammenfassung

Studenten in Vorlesungen befinden sich in einer feedbackarmen Umgebung. Technologie Unterstützung wird bereits jetzt von manchen Universitäten verwendet um Folien anzuzeigen und für Kommunikation zwischen Studenten. Wäre es nicht möglich die vorhandene Infrastruktur dazu zu nutzen einem Studenten ein individuelles Feedback zu geben?

Diese Frage wird am Beispiel der Software Backstage getestet. Backstage ist eine vorlesungsbegleitende Software der LMU, auf der der Dozent Folien anzeigen kann, die zur Kommunikation von Studenten untereinander verwendet wird und in der auch Quiz¹ vom Dozenten gestellt werden können. Die Idee dieser Arbeit ist es herauszufinden, ob sich diese Plattform eignet den Studenten hilfreiches Feedback zu geben. Im Rahmen der Arbeit werden Problematiken im Zusammenhang mit den vorhandenen Daten angesprochen, die die Qualität des Feedbacks beeinflussen können. Wir haben evaluiert, ob das im Educational Data Mining häufig verwendete Verfahren des Bayesian Knowledge Tracing für ein Feedbacksystem geeignet sein könnte. Es wurde ein Konzept ausgearbeitet, dass auf einem Bayesian Knowledge Tracing basiert, bei dem das verwendete Hidden Markov Model angepasst wurde.

Hierzu wurde herausgearbeitet, welche Parameter benötigt werden und was bei deren Wahl zu beachten ist. Es wurde ein Prototyp einer Implementierung des Systems angefertigt um die Auswirkungen der gewählten Parameter testen zu können.

¹Der Plural von „Quiz“ wird dem Duden folgend mit „Quize“ verwendet.
<http://www.duden.de/rechtschreibung/Quiz>.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Modell zur Teilnahme an Audience Response Systemen	1
1.2 Ziele der Arbeit	2
1.3 Backstage	2
1.3.1 Backchannel Kommunikation	2
1.3.2 Audience Response System	3
1.3.3 Ablauf einer Quizrunde	3
2 Modell für die studentische Teilnahme an Audience Response Quiz . .	7
2.1 Datengrundlage	7
2.1.1 Kategorisierung der Fragen	8
2.1.1.1 Kategorien nach Alison King	8
2.1.1.2 Kategorien nach Itemschwierigkeit	9
2.1.2 Anzahl der Kategorien	9
2.2 Bayesian Knowledge Tracing	9
2.2.1 Modell	10
2.2.2 Berechnung	12
2.2.3 Beispiel	13
2.3 HMM als Generalisierung von BKT	14
2.3.1 Aufbau eines HMM	15
2.3.2 Beispiel eines HMM	16
2.3.3 BKT als HMM aufgebaut	17
2.4 Backstage-Feedback-System	19
2.4.1 Zeitschritt	19
2.4.2 Versteckte Zustände	19
2.4.3 Übergangsmatrix zwischen den versteckten Zuständen	20
2.4.4 Beobachtungen	21
2.4.5 Beobachtungsmatrix	21
2.4.5.1 Constraints	23
2.4.6 Startverteilung	23
2.4.7 Graphische Darstellung des Backstage Feedback Systems	24

3 Mögliche Implementierung	25
3.1 Architektur: Mögliche Anbindung an Backstage	25
3.2 möglicher Zeitpunkt der Berechnung	25
4 Evaluierung	27
4.1 Evaluierungsimplementierung	27
4.1.1 Konfiguration	27
4.1.2 Beschreibung	28
4.1.3 Oberfläche für Tests	28
4.2 Evaluierung anhand synthetischer Daten	29
4.2.1 Student, der sich nicht an Quiz beteiligt	29
4.2.2 Student, der <i>eher</i> gut ist	29
4.2.3 Student der sich verbessert	31
4.3 Ergebnis der Evaluierung	32
5 Fazit	33
5.1 Zusammenfassung der Arbeit	33
5.2 Vorteile gegenüber dem BKT	33
5.3 Grenzen des Modells	34
5.4 Weiterführende Arbeiten	34
6 Anhang	37
6.1 Implementierung der Startverteilung	37
6.2 Implementierung der Beobachtungsmatrix	37
6.3 Implementierung der Übergangsmatrix	38
6.4 Konfigurationsdateisdatei	39
Literaturverzeichnis	41

Abbildungsverzeichnis

1.1 Quiz, das gerade aktiv ist (Studentenansicht)	4
1.2 Quiz, das gerade aktiv ist (Dozentenansicht)	4
1.3 Ergebnis eines Quiz	5
2.1 Intuitive Annahme zum Zusammenhang von Itemschwierigkeit und Einteilung nach King	10
2.2 Modell eines BKT	11
2.3 Instanz eines BKT mit einer Reihe von Beobachtungen	13
2.4 Beispiel Hidden Markov Model	17
2.5 Darstellung des Backstage Feedback Systems	24
3.1 Technische Architektur	26
4.1 Testoberfläche	29
6.1 Beobachtungsmatrix in der derzeitigen Implementierung	38

Tabellenverzeichnis

2.1 Beobachtungsmatrix Bayesian Knowledge Tracing (BKT)	11
2.2 Eingabedaten BKT. Die hier verwendeten Werte stammen aus einem MOOC zum Thema Big Data in Education.	14
2.3 Beispiel BKT	14
2.4 Beispiel Hidden Markov Model (HMM) (R=Regen; S=Sonne)	18
2.5 Beobachtungsmatrix HMM	22
4.1 Codierung für Beobachtungen	28
4.2 Aufbau Tabellen	29
4.3 Student ohne Quizteilnahme	30
4.4 Student, der <i>eher</i> gut ist	31
4.5 Student der sich verbessert	31
6.1 Startverteilung	37
6.2 Beobachtungsmatrix implementiert	38
6.3 Übergangsmatrix implementiert	39

Abkürzungsverzeichnis

ARS	Audience Response System
BFS	Backstage Feedback System
BKT	Bayesian Knowledge Tracing
EDM	Educational Data Mining
HMM	Hidden Markov Model

In diesem Kapitel wird beschrieben, in welcher Umgebung die Daten entstehen und was das Ziel der Arbeit ist.

1.1 Modell zur Teilnahme an Audience Response Systemen

Das Umfeld einer universitären Vorlesung ist eine feedbackarme Umgebung. Von Studenten wird viel selbstgesteuertes Lernen verlangt. Ob sie ihr Lernziel erfüllen, erfahren sie meist erst am Ende des Semesters durch die Abschlussklausur. Die Lehrmethode „Vorlesung“ sieht kein Feedback vor. Der Dozent steht vor den Studenten und vermittelt den Unterrichtsstoff. Besonders in Massenvorlesung kommt auf jeden Dozenten eine große Anzahl von Studenten²³. Die Vorlesung ist gekennzeichnet durch Armut an Feedback und einer passiven Grundhaltung von Studenten. Daher stellt sich die Frage, wie man einem Studenten Feedback geben kann? Wie ist es möglich, dass ein Student bereits während einer Vorlesung einen Eindruck davon bekommen kann, ob er das Lernziel erreicht? Es ist nötig, dem Studenten bereits während des Semesters ein Feedback zu geben. Dies kann aufgrund der Vielzahl der Studenten und der Belastung des Lehrpersonals nur automatisiert erfolgen. Das Feedback soll aus der Teilnahme an einem Audience Response System (ARS) gegeben werden. Bei Audience Response Systemen handelt es sich um die Möglichkeit eines Dozenten, eine Frage an die Studenten zu stellen, die Studenten beantworten diese über eine technische Einrichtung (z.B. Schalter, Smartphones) [8]. Als ARS soll hier Backstage dienen, das im nächsten Kapitel (1.3) genauer erläutert wird.

²<http://www2.pms.ifi.lmu.de/erlebt/?p=10342>.

³<http://www.sueddeutsche.de/muenchen/lmu-und-tu-in-muenchen-wettstreit-der-universitaeten-1.1639995>.

1.2 Ziele der Arbeit

Ziel dieser Arbeit ist es, einen Ansatz zu konzipieren, bei dem mit Hilfe der in Backstage generierten Daten, aus den gestellten Quiz, ein Feedback für Studenten generiert wird. Hierzu wird im Forschungsgebiet des Educational Data Mining (EDM) gearbeitet. EDM beschäftigt sich damit, wie mit Daten die im Umfeld der Lehre und des Lernens gewonnen werden, diese verbessert werden kann [15].

Es soll mit Hilfe eines stochastischen Prozesses versucht werden, ein Modell über die Teilnahme eines Studenten an vorlesungsbegleitenden Quiz zu erstellen. Ein stochastischer Prozess ist ein zeitlicher Verlauf von zufälligen Beobachtungen⁴. Dieses Modell ermöglicht es, die Lernkurve eines Studenten einzuschätzen. Zudem ist es möglich, eine Prognose für die Zukunft zu erstellen.

1.3 Backstage

Diese Arbeit soll auf Daten von Backstage basieren. Backstage stellt eine Umgebung für Studenten und Dozenten zur Verfügung, auf der sie sich austauschen und kommunizieren können. Zudem ist es dem Dozenten möglich, Quiz zu stellen. Backstage wurde konzipiert um die Studenten zu aktivieren und zu motivieren. Mit Backstage kann der Dozent die Folien anzeigen und der Student hat unter anderem die Möglichkeit, diese zu kommentieren [5, 7, 11, 12].

Backstage liefert die Daten die für diese Arbeit verwendet werden sollen.

1.3.1 Backchannel Kommunikation

Backstage wird in sehr großen Vorlesungen eingesetzt. Bei diesen ist es für den einzelnen Studenten schwer, seine Fragen (die durchaus vorhanden sind) anzubringen. Meist ist die Frage eines Studenten nicht nur bei ihm vorhanden, sondern auch bei zahlreichen anderen Studenten.⁵ Das Klären einer Frage kostet jedoch Zeit und diese ist in solchen Veranstaltungen nicht für alle Fragen verfügbar. Aus diesem Grund wurde Backstage entwickelt, um so den Studenten eine Möglichkeit zu geben sich Fragen gegenseitig zu beantworten. Backstage dient zudem als Vorlesungsmedium. Backstage zeigt die Folien an und dazu die von den Studenten gestellten Fragen und Kommentare.

In Backstage annotieren die Studenten mit ihren Kommentaren den Foliensatz der Vorlesung. Die Kommentare sind durch Kategorien typisiert, damit jeder andere Student schnell erkennen kann, ob es sich hierbei um eine Frage, eine Anmerkung oder einen fachfremden Kommentar handelt.

⁴https://de.wikipedia.org/wiki/Stochastischer_Prozess.

⁵<http://backstage.pms.ifi.lmu.de/>.

1.3 Backstage

1.3.2 Audience Response System

Der Dozent hat die Möglichkeit, in Backstage Quizrunden durch das sogenannte Audience Response System (ARS) einzubauen. Diese Quiz werden von den Studenten beantwortet, und wenn der Dozent die Quizrunde beendet, wird die richtige Antwort angezeigt, sowie eine Statistik über die abgegebenen Ergebnisse. Der Student bekommt so zeitnah Rückmeldung über seinen aktuellen Wissensstand. Außerdem dient dies der Aktivierung der Studenten, die sonst passiv die Vorlesung verfolgen [5].

Das Audience Response System (ARS), wie es in Backstage umgesetzt ist, ist ein System mit dem der Dozent verschiedene Typen von Quiz stellen kann. Offene Fragen sind möglich, sowie Single und Multiple Choice Fragen. Die *Audience*, in unserem Fall also die Studenten, können diese Fragen beantworten. Der Student bekommt eine Auswahl an Antworten und versucht dann die richtige(n) auszuwählen oder, im Fall einer offenen Frage, die Antwort in einem Textfeld einzugeben.

1.3.3 Ablauf einer Quizrunde

Der Dozent startet ein Quiz. In Abbildung 1.1 ist eine Multiple Choice Frage aus Sicht eines Studenten zu sehen. Während eines aktiven Quiz ist es in Backstage nicht möglich, Fragen zu stellen oder andere Kommentare zu erstellen. Auch ist es während dieser Zeit nicht möglich in den Folien zu blättern. Dafür sind während dieser Zeit die Felder zur Abstimmung aktiv.

Der Dozent sieht, wie viele Personen bereits abgestimmt haben, wie die aktuelle Verteilung der Antworten ist und wie der Verlauf der Antworten war (siehe Abbildung 1.2).

Nachdem der Dozent die Quizrunde beendet hat wird das Ergebnis quantitativ ausgewertet. Es wird visuell dargestellt, welche Antwort die Richtige ist und wie viel Prozent der Studenten welche Antwort ausgewählt haben (siehe Abbildung 1.3).

Danach kann der Dozent mit der nächsten Folie die Vorlesung fortsetzen. Ebenso sind Diskussionen zur Frage sowie zu den Antwortmöglichkeiten möglich. Der Student hat jederzeit (auch in der Nachbereitung des Stoffes) die Möglichkeit, die Quiz sowie deren Antworten und Kommentare zu sehen.

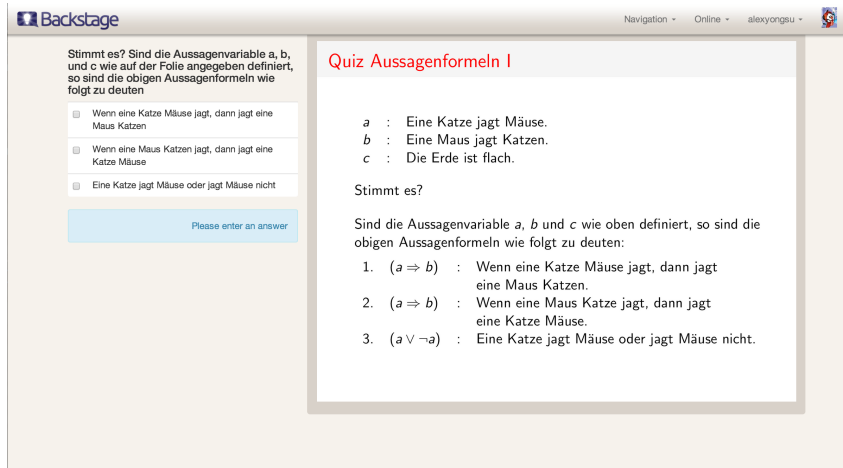


Abbildung 1.1: Quiz, das gerade aktiv ist (Studentenansicht)

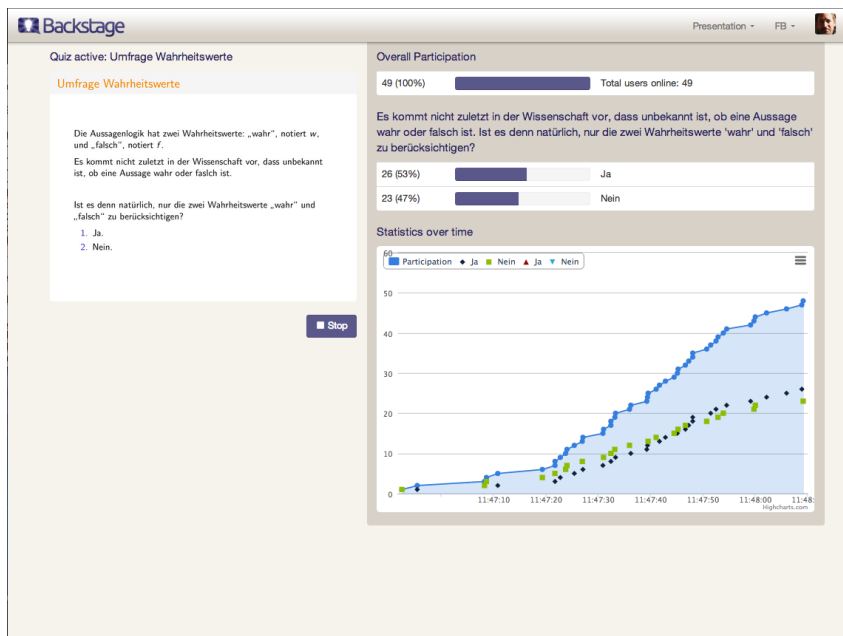


Abbildung 1.2: Quiz, das gerade aktiv ist (Dozentenansicht)

1.3 Backstage

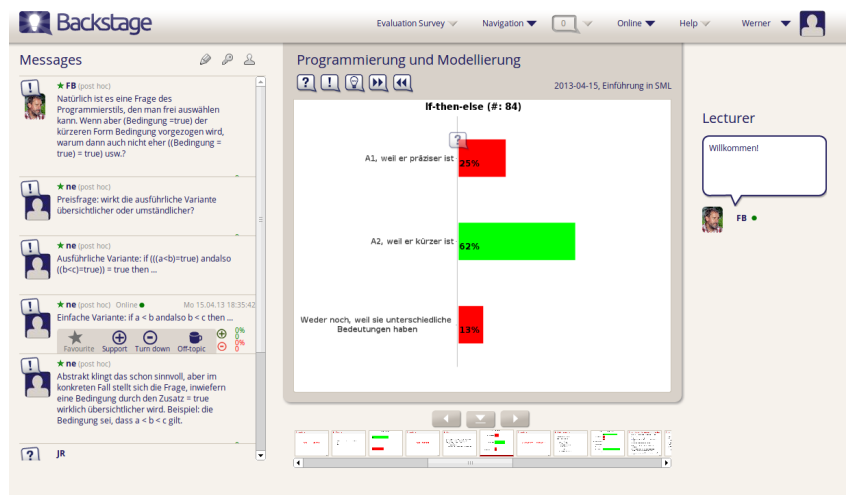


Abbildung 1.3: Ergebnis eines Quiz

Modell für die studentische Teilnahme an Audience Response Quiz

In diesem Kapitel soll ein stochastischer Prozess dazu verwendet werden, eine Aussage über einen Studenten zu machen. Stochastische Prozesse dienen der mathematischen Beschreibung von Ereignissen, die dem Zufall unterliegen und zeitlich geordnet sind. Die Antworten eines Studenten sind nicht zufällig, dem eigentlichen Wortsinn nach jedoch aus Sicht des Systems nicht vorhersehbar.

Um einen stochastischen Prozess anwenden zu können, müssen die gesammelten Daten in Kategorien eingeteilt werden, es müssen Zustände definiert werden, in denen sich ein Student befinden kann. Ziel ist es dann, das Modell so anzupassen, dass es die Daten möglichst genau vorhersagt. Es wird also angenommen, der Zustand eines Studenten sei bekannt und daraus wird die nächste Quizantwort generiert. Dieser sog. Generator wird dann umgekehrt benutzt, indem man auswertet, welcher Zustand als nächstes erzeugt werden würde und diesen als Vorhersage verwendet. Man nimmt also die Liste der Quizergebnisse und berechnet, wie wahrscheinlich welcher Zustand ist. Dies wird im folgenden noch einmal ausführlich dargestellt.

2.1 Datengrundlage

Quiz bieten die Möglichkeit, eine qualitative Aussage über die Leistungen eines Studenten objektiv vorzunehmen und maschinell auszuwerten.

Die Daten der Quiz eignen sich in besonderem Maße für eine automatisierte Auswertung. Die Daten benötigen keine zusätzliche Transformation. Anders als bei den Daten aus der Backchannel-Kommunikation (Fragen, Chat, etc.) eine zusätzliche Transformation nötig ist, sollen sie in dieser Arbeit nicht weiter berücksichtigt werden. Der Datenbestand der Quizantworten ist unseren Erfahrungen nach durchwegs hoch. Ein sehr großer Teil der zu diesem Zeitpunkt in Backstage angemeldeten Studenten nimmt auch an den Quiz teil. Ebenso spielen unseren Erfah-

2 Modell für die studentische Teilnahme an Audience Response Quiz

rungen nach Charaktereigenschaften der Studenten bei der Teilnahme an Quiz eine geringere Rolle, wie beispielsweise bei Kommentaren und anderen Postings.

Zu jedem aktiven Studenten steht unserer Erfahrung nach eine etwa vergleichbare Datenbasis zur Verfügung.

Es können hier verschiedene Arten von Daten unterschieden werden: Quiz-bezogene Daten: mögliche Antworten, richtige Antworten; Quizrunden-bezogene Daten: Anzahl der Quizantworten, Zeitraum/Dauer der Quizrunde, Ergebnis der Quizrunde und Studenten-bezogene Daten: Auswahl der Quizantwort, Korrektheit der Antwort.

Zu Beginn des Semesters melden sich viele Studenten erst in Backstage an. Manche haben bereits einen Account. Hier stellt sich die Frage, ob Daten aus einer anderen Vorlesung für ein Feedbacksystem in dieser Vorlesung verwendet werden dürfen/sollen. Hier haben wir uns klar dagegen ausgesprochen. Die Vorlesungen sind zu unterschiedlich, als dass man Ergebnisse einfach übertragen könnte. Das Feedback soll für jeden Studenten und jede Vorlesung unabhängig voneinander generiert und angezeigt werden.

2.1.1 Kategorisierung der Fragen

Für die Verwendung des Modells ist es nötig, die Fragen in Kategorien einzuteilen. Im einfachsten Fall könnte dies richtig und falsch sein. Die Antwort eines Studenten auf ein Quiz ist unserer Meinung nach jedoch mehr als nur richtig oder falsch. Um mehr Aussagekraft in das Modell zu bringen, haben wir uns dazu entschlossen, zusätzlich die Frage zu betrachten. Durch die Einteilung der Frage in eine Kategorie lässt sich bereits die Aussagekraft einer Antwort erhöhen.

Die Fragen in Kategorien einzuteilen wäre über verschiedene Möglichkeiten denkbar.

2.1.1.1 Kategorien nach Alison King

Naheliegender war es hier ein Kategoriensystem zu verwenden, das in der Pädagogik verbreitet ist. Die Pädagogin Alison King hat eine Theorie entwickelt [9], die es erlaubt Fragen zu kategorisieren.

Alison King hat herausgearbeitet, dass es 3 Typen von Fragen gibt. Zum ersten die *Faktenfrage*, dann *Verständnisfrage* und *Integrationsfrage*. Diese Typen der Fragen können nach ihrer Elaboriertheit geordnet werden, wobei die Integrationsfragen am höchsten sind, also auch am anspruchsvollsten, und Faktenfragen sind am niedrigsten, also am einfachsten zu lösen. Diese Kategorisierung würde sich für unser Modell durchaus eignen, da hierdurch eine genauere Qualifizierung der Fragen möglich ist. Diese Qualifizierung könnte auf einer sehr sachlichen Ebene gemacht

2.2 Bayesian Knowledge Tracing

werden. Es gibt klare Kriterien, welche Frage in welche Kategorie gehört.

2.1.1.2 Kategorien nach Itemschwierigkeit

Die Itemschwierigkeit gibt den Anteil der Studenten an, die ein Quiz richtig beantwortet haben. Die Itemschwierigkeit wird im einfachsten Fall mit n/N berechnet, wobei n die Anzahl der Studenten ist, die dieses Quiz richtig beantwortet haben und N ist die Anzahl aller in Backstage für diese Vorlesung angemeldeten Studenten. Um die Kategorisierung steuern zu können, haben wir uns zusätzlich einen *Glaubensfaktor* g überlegt (wobei $g > 0$). Die Formel zur Berechnung zur Itemschwierigkeit lautet dann $\frac{n \times g}{N \times g}$. Je höher g ist, desto weniger Glauben wird dem Wert n geschenkt. Ein $g > 1$ nähert den Wert der Itemschwierigkeit dem Wert 1 an, verschiebt die Verteilung also nach oben, während ein Wert von $g < 1$ die Verteilung nach unten verschiebt.

Intuitiv könnte man annehmen, dass zwischen der Itemschwierigkeit und der Einteilung nach King ein Zusammenhang besteht. Nämlich, dass je höher die Kategorie nach King ist, desto höher ist auch die Itemschwierigkeit. Siehe Abbildung 2.1. Der Zusammenhang konnte in einer formativen Evaluation von Quizantworten, die mit Backstage gesammelt wurden, nicht bestätigt werden. Es wurde festgestellt, dass einige Integrationsfragen in ihrem Fall besser beantwortet wurden als Faktenfragen. Unserer Vermutung nach liegt das daran, dass die Antwortmöglichkeiten nicht berücksichtigt wurden. Die Fragen lassen sich zwar nach King kategorisieren, nicht jedoch die Quiz. Bei den Quiz spielen die Antwortmöglichkeiten ebenfalls eine Rolle. Man könnte sagen, dass die Antwortmöglichkeiten eine zusätzliche Dimension bei dieser Einteilung darstellen.

Daher haben wir uns entschieden in dieser Arbeit die Itemschwierigkeit zu verwenden.

2.1.2 Anzahl der Kategorien

Um eine Differenzierung der Fragen herzustellen, war es nötig, mehr als eine Kategorie zu haben. Wir haben uns hier für drei Kategorien entschieden, da hier der Aufwand der Einteilung der Fragen, sowie der Mehrwert für die Evaluation in einem gesunden Verhältnis standen. Die Kategorien sind mit *leicht*, *mittel* und *schwer* zudem leicht zu verstehen.

2.2 Bayesian Knowledge Tracing

Eine im Bereich des EDM sehr oft verwendete Technik um die Leistung von Studenten zu bewerten, ist Bayesian Knowledge Tracing (BKT). Diese Technik wurde als Komponente für intelligente Tutor-Systeme entwickelt.

Im Jahr 1995 haben Albert Corbett und John Anderson ein System [3] entwickelt, das sich etabliert hat und heute in zahlreichen Variationen vorliegt [10]. Dieses Sys-

2 Modell für die studentische Teilnahme an Audience Response Quiz

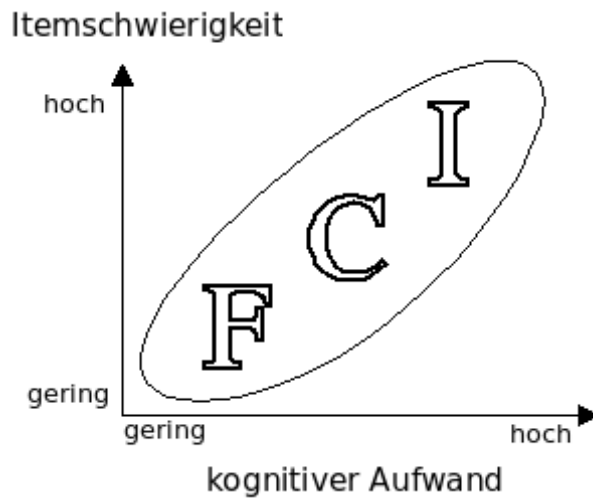


Abbildung 2.1: Intuitive Annahme zum Zusammenhang von Itemschwierigkeit und Einteilung nach King

tem ist heute unter dem Namen BKT bekannt. Das BKT ist eingebunden in ein Tutoringsystem. Dieses funktioniert folgendermaßen: Ein Student lernt durch das Beantworten von Quiz oder durch Informationseinheiten zwischen den Quiz. Erst wenn er „genügend“ Quiz richtig beantwortet hat, bekommt er neue Lerninhalte und kann fortfahren. Tutoring-Systeme gab es zu dieser Zeit bereits, das neue an Corbett und Anderson war, dass sie die Entscheidung, wann ein Student zur nächsten Einheit vorrücken darf, nicht auf feste Regeln (z.B. nach fünf richtigen Antworten in Folge), sondern auf ein stochastisches Modell gestützt haben. Bei diesem werden die Faktoren Raten (*Guess*), also Wissen ist nicht vorhanden, die Antwort aber trotzdem richtig, sowie Verrutschen und Flüchtigkeitsfehler (*Slip*), also das Wissen ist vorhanden, die Antwort jedoch falsch, mit einbezogen.

2.2.1 Modell

Um das Modell des BKT zu verwenden, werden neben den bereits angesprochenen Parametern für *Slip* und *Guess* noch weitere Werte benötigt. Die beiden Komplementäre zu *Guess* und *Slip*, die sich jedoch durch $1 - \text{Wahrscheinlichkeit}$ berechnen lassen. Außerdem wird die Startwahrscheinlichkeit, also für Level 0 ($P(L_0)$ ⁶), der angibt ob ein Student etwas bereits zu Beginn des Quiz wusste, benötigt. Zuletzt noch die Transferwahrscheinlichkeit $P(T)$, dass der Student etwas nicht wusste und durch diese Einheit gelernt hat. Die Wahrscheinlichkeit, dass ein Transfer in die andere Richtung stattfindet, ist nicht nötig, da das System annimmt, dass nichts vergessen wird.

Die möglichen Beobachtungen sind zwei Werte: Antwort eines Studenten in einem Quiz ist richtig (CORRECT) oder für falsche Antworten (FALSE). Außerdem gibt

⁶ $P(\dots)$ bedeutet „Wahrscheinlichkeit, dass ...“

2.2 Bayesian Knowledge Tracing

es zwei Zustände, in denen sich ein Student befinden kann: NOT KNOWLEDGE, der aussagt, dass der Student das Wissen dieser Lerneinheit (noch) nicht hat und KNOWLEDGE, der dafür steht, dass der Student das benötigte Wissen hat. Ein Student beginnt immer bei NOT KNOWLEDGE und muss so lange Quiz beantworten, bis er in den Zustand KNOWLEDGE kommt. Dann hat er diese Lerneinheit bestanden und rückt zur nächsten vor. Danach wird der BKT zurück gesetzt. Hieraus ergibt sich auch der Grund, warum es keine Übergangswahrscheinlichkeit zwischen dem Zustand KNOWLEDGE zu NOT KNOWLEDGE gibt: die Berechnung endet sobald einmal der Zustand KNOWLEDGE erreicht wurde.

Der probabilistische Zusammenhang zwischen den beobachteten Quizantworten (richtig beantwortet, falsch beantwortet, geraten, Flüchtigkeitsfehler) und dem Transfer (Wissen vorhanden/nicht vorhanden) kann wie folgt in einer Tabelle ("Beobachtungsmatrix") zusammengefasst werden.

Tabelle 2.1: Beobachtungsmatrix BKT

Beobachtung	KNOWLEDGE	NOT KNOWLEDGE
CORRECT	$1 - P(G)$	$P(G)$
FALSE	$P(S)$	$1 - P(S)$

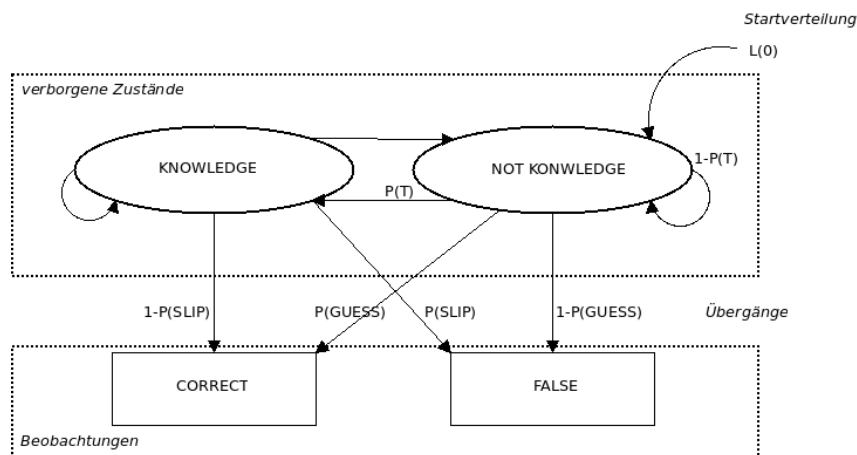


Abbildung 2.2: Modell eines BKT

Das BKT-Modell ist ein vereinfachtes Hidden-Markov-Modell, bei dem bspw. für die Entscheidungsfindung uninteressante Zustandsübergänge nicht weiter berücksichtigt werden. Mit diesem Modell wird versucht anhand von Beobachtungen (Student gibt richtige/ falsche Antwort auf eine Frage) herauszufinden welchen Wert der versteckte Zustand KNOWLEDGE hat. Der Zustand ist deshalb versteckt, da es keine direkte Möglichkeit gibt, ihn zu messen oder zu berechnen. Es kann für den versteckten Zustand nur eine Wahrscheinlichkeit aufgrund von Beobachtungen gegeben werden.

Eine Problematik ist bei diesem Verfahren besonders hervor zu heben. Es werden

2 Modell für die studentische Teilnahme an Audience Response Quiz

Werte für die Parameter benötigt. Wie diese berechnet werden können wird in zahlreichen Publikationen angesprochen [3, 16, 2, 6].

In der Abbildung 2.2 ist zu sehen, wie ein BKT aufgebaut ist. Es gibt zwei versteckte Zustände, die die Wahrscheinlichkeiten angeben, dass ein Student eine bestimmte Fähigkeit hat oder nicht hat. Diese Zustände werden oft auch als *versteckte Zustände* oder *latente Variablen* bezeichnet. Dann werden eine Reihe von Tests durchgeführt, die Ergebnisse dieser Tests dienen als Beobachtungen und beeinflussen die Wahrscheinlichkeiten der versteckten Zustände. Da die Zustände redundant sind wird meist nur auf den Zustand Wissen vorhanden zurückgegriffen. In dieser Abbildung werden vollständigkeithalber auch die Kanten mit abgebildet, die nicht definiert sind (unbeschriftet).

BKT wurde entwickelt, um zu prüfen, ob ein Student eine gewisse Lerneinheit erfüllt hat und bereit ist zur nächsten vorzurücken. Die Einteilung des Stoffes in Lerneinheiten ist ebenfalls nicht trivial⁷.

2.2.2 Berechnung

Die Formel zur Berechnung der Wahrscheinlichkeit, dass etwas gelernt wurde lautet [4]:

$$P(L_n) = P(L_{n-1} | \text{Beobachtung}) + (1 - P(L_{n-1} | \text{Beobachtung})) \times P(T)$$

also die Wahrscheinlichkeit, dass es im Schritt davor bereits bekannt war und die Wahrscheinlichkeit, dass es im Schritt davor noch nicht bekannt war jedoch durch den Schritt gelernt wurde. Die Werte für den Vorgängerschritt werden jeweils in Abhängigkeit von der Beobachtung betrachtet.

Die Formeln zur Berechnung von $P(L_{n-1} | \text{Beobachtung})$ beruhen auf dem Satz von Bayes [1].

Bei Beobachtung eines **richtigen** Ergebnisses:

$$P(L_{n-1} | \text{richtig}) = \frac{P(L_{n-1}) \times (1 - P(S))}{P(L_{n-1}) \times (1 - P(S)) + (1 - P(L_{n-1})) \times P(G)}$$

also die Wahrscheinlichkeit, dass der Student es bereits vorher wusste und nicht verrutscht ist, über der Wahrscheinlichkeit, dass er es bereits vorher wusste und nicht verrutscht ist plus der Wahrscheinlichkeit, dass der Student es nicht wusste und geraten hat. Der Zähler gibt an, wie hoch die Wahrscheinlichkeit ist, dass der Student es tatsächlich weiß, während der Nenner angibt, wie hoch die Wahrscheinlichkeit ist, dass der Student die richtige Antwort gegeben hat.

Bei Beobachtung eines **falschen** Ergebnisses:

$$P(L_{n-1} | \text{falsch}) = \frac{P(L_{n-1}) \times P(S)}{P(L_{n-1}) \times P(S) + (1 - P(L_{n-1})) \times (1 - P(G))}$$

⁷http://www.learnlab.org/research/wiki/index.php/Knowledge_component.

2.2 Bayesian Knowledge Tracing

Wie oben nur für die Beobachtung eines falschen Ergebnisses beim Studenten.

Beim BKT werden die Zustände immer nur für den Zustand KNOWLEDGE berechnet, da die Werte so sind, dass NOT KNOWLEDGE das Komplement dazu ist. $1 - \text{KNOWLEDGE} = \text{NOT KNOWLEDGE}$. Aus dieser Vereinfachung heraus wird der Zustand NOT KNOWLEDGE auch in Illustrationen häufig nicht gezeichnet und berücksichtigt. In diesem Zusammenhang wird oft die Formulierung „der Student befindet sich im Zustand ...“ verwendet. Diese bezieht sich auf den Zustand, der wahrscheinlicher ist. Der Student ist nie zu 100% in einem Zustand, er ist immer zu einer bestimmten Wahrscheinlichkeit in beiden. Da üblicherweise nur der Zustand KNOWLEDGE überhaupt betrachtet wird, befindet sich der Student in diesem Zustand, wenn die Wahrscheinlichkeit dieses Zustandes >0.5 ist. Damit folgt automatisch, dass der Zustand NOT KNOWLEDGE < 0.5 ist und somit die Wahrscheinlichkeit für den Zustand KNOWLEDGE größer ist.

2.2.3 Beispiel

Bei BKT ist es notwendig, das Wissen in Lerneinheiten einzuteilen. Ein Student hat zu einer Lerneinheit folgende Ergebnisse in einer Reihe von Tests: **FALSE- FALSE-CORRECT - FALSE - CORRECT - ... - CORRECT**

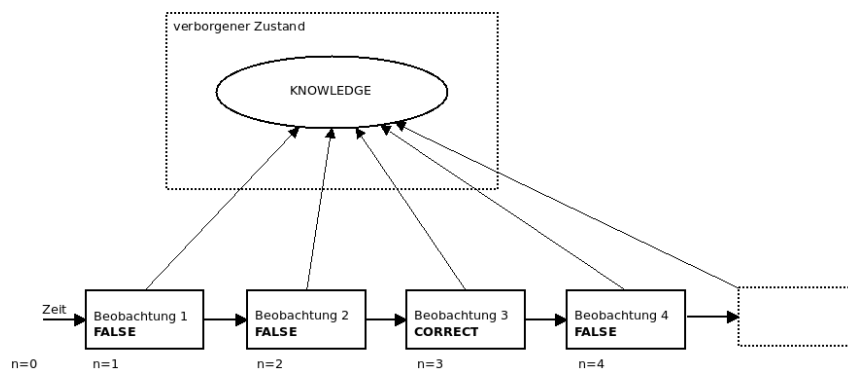


Abbildung 2.3: Instanz eines BKT mit einer Reihe von Beobachtungen

Ab wann kann davon ausgegangen werden, dass der Student diese Lerneinheit gelernt hat?

Der Student hat für jede Lerneinheit immer genau einen der Zustände: KNOWLEDGE oder NOT KNOWLEDGE. Das stochastische Modell funktioniert so, dass es eine Vorhersage für den Studenten macht, ob dieser die nächste Frage CORRECT beantworten wird. Ist diese Vorhersage positiv, wird die Lerneinheit als gelernt betrachtet.

Die Werte aus diesem Beispiel stammen aus einem MOOC zum Thema Big Data in Education⁸ und sollen lediglich als Beispiel dienen. Die Anpassung des Modells auf die tatsächlichen Werte, ist eines der größten Probleme des BKT [16].

⁸<https://class.coursera.org/bigdata-edu-001/lecture/69>

Tabelle 2.2: Eingabedaten BKT. Die hier verwendeten Werte stammen aus einem MOOC zum Thema Big Data in Education.

Variable	Wert (beispielhaft)	Bedeutung
$P(L_0)$	0,4	Student hat das Wissen bereits vor dieser Lerneinheit Wahrscheinlichkeit, dass der Student das Wissen bei einmaliger Testmöglichkeit dieser Lerneinheit gelernt hat
$P(T)$	0,1	
$P(G)$	0,3	Wahrscheinlichkeit, dass ein Student bei richtiger Antwort nur geraten hat (Guess)
$P(S)$	0,2	Wahrscheinlichkeit, dass ein Student bei falscher Antwort nur verrutscht ist (Slip)

Tabelle 2.3: Beispiel BKT

n	Beobachtung	$P(L_{n-1})$	$P(L_n \text{Beobachtung})$	$P(L_n)$
1	FALSE	0,40	$\frac{0,40 \times 0,30}{0,40 \times 0,30 + 0,60 \times 0,80} = 0,20$	$0,20 + 0,80 \times 0,10 = 0,24$
2	FALSE	0,24	$\frac{0,24 \times 0,30}{0,24 \times 0,30 + 0,76 \times 0,80} = 0,10$	$0,10 + 0,90 \times 0,10 = 0,19$
3	CORRECT	0,19	$\frac{0,19 \times 0,70}{0,19 \times 0,70 + 0,81 \times 0,20} = 0,45$	$0,45 + 0,55 \times 0,10 = 0,50$
4	FALSE	0,50	$\frac{0,50 \times 0,30}{0,50 \times 0,30 + 0,50 \times 0,80} = 0,27$	$0,27 + 0,73 \times 0,10 = 0,34$
5	CORRECT	0,34	$\frac{0,34 \times 0,70}{0,34 \times 0,70 + 0,66 \times 0,20} = 0,64$	$0,64 + 0,36 \times 0,10 = 0,67$
6	CORRECT	0,67	$\frac{0,67 \times 0,70}{0,67 \times 0,70 + 0,33 \times 0,20} = 0,87$	$0,87 + 0,13 \times 0,10 = 0,88$
7	CORRECT	0,88	...	

Ab wann würde der BKT vorhersagen, dass ein Student für diese Lerneinheit das nötige Wissen erreicht hat? Dies hängt in diesem Beispiel von der Rechengenauigkeit ab. Es muss die Wahrscheinlichkeit für $P(L_n) > 50\%$ sein. Dies wäre bei größerer Rechengenauigkeit bereits bei Schritt erreicht, hier jedoch erst mit dem fünften Schritt.

Warum müssen gefühlt nur so wenige Fragen richtig beantwortet werden? Zum einen hängt dies sehr stark von den gewählten Parametern ab. Zum anderen nimmt man an, dass mit jeder Frage, mit der sich der Student beschäftigt, sein Wissen steigt (dargestellt durch $P(T)$) unabhängig davon, ob er die Frage richtig oder falsch beantwortet, daher wird $P(T)$ noch addiert. Auch die Erkenntnis, dass eine Antwort auf eine Frage falsch ist, ist neues Wissen.

2.3 HMM als Generalisierung von BKT

Für die Verwendung in Backstage stellt BKT keine zufriedenstellende Lösung dar. Zum einen ist es nicht möglich, aufgrund der Daten Lerneinheiten zu bilden. Die Lerneinheiten müssten vom Lehrenden gebildet werden und dies würde für ihn einen zusätzlichen Aufwand bedeuten. Um die Benutzerfreundlichkeit nicht einzuschränken ist dies in Backstage nicht gewünscht.

2.3 HMM als Generalisierung von BKT

Bei BKT werden alle Vorgänge, die einen Übergang von KNOWLEDGE in NOT KNOWLEDGE definieren nicht betrachtet, da dieser in BKT nicht existiert. Sobald der Student einmal den Zustand KNOWLEDGE erreicht hat, hat er diese Lerneinheit bestanden und rückt zur nächsten vor. Der BKT wird zurückgesetzt auf die Startwerte. In einem Feedbacksystem ist es nötig, dass sich Studenten sowohl verbessern wie auch verschlechtern können.

Zudem hat der Student nicht die Möglichkeit, mehrere Aufgaben zu einer Lerneinheit zu absolvieren. Der Dozent stellt eine Reihe von vorher definierten Fragen im Laufe einer Vorlesung. Wie viele der Student davon richtig beantwortet, beeinflusst den Fortgang der Veranstaltung nur indirekt, und die Anzahl der gestellten Quiz gar nicht.

Der dritte Grund, warum sich BKT für ein Feedbacksystem in Backstage nicht eignet ist, dass BKT kein Feedbacksystem ist, sondern ein intelligenter Tutor. Das System ist dafür ausgelegt, eine binäre Entscheidung zu treffen, darf der Student zur nächsten Lerneinheit vorrücken oder nicht. Als Feedbacksystem ist das ungeeignet. Ein Student braucht mehrere verschiedene mögliche Zustände, in denen er sich befinden kann und von denen er seinen Erfolg ablesen kann. Die Aussagekraft des Feedbacks muss möglichst hoch sein um den Studenten einen echten Mehrwert zu liefern.

Das Hidden Markov Model (HMM) ist ein in verschiedenen Bereichen verwendetes Model, es ist insbesondere sehr weit verbreitet bei Sprach- und Schrifterkennung [14, 17]. Was all diese Gebiete gemeinsam haben ist, dass sie etwas bestimmen möchten, was sie nur durch Indizien erkennen können. Im Fall von Spracherkennung sind es die Daten über Schallwellen, die zu Silben interpretiert werden, und im Fall von Schrifterkennung sind es Pixel in unterschiedlichen Farben, die zu Buchstaben interpretiert werden.

Das besondere am HMM ist, dass es auf diesem Modell mehrere Formen der Berechnung gibt. Es kann zum einen aus einer Liste an Beobachtungen der wahrscheinlichste Zustand berechnet werden. Man könnte jedoch auch, aus gegebenen Endzuständen (z.B. Klausurnoten), wahlweise die Startverteilung oder die Beobachtungsmatrix neu berechnen. Ebenso kann aus dem Endzustand die Liste der wahrscheinlichsten Beobachtungen generiert werden.

2.3.1 Aufbau eines HMM

Für ein HMM werden folgende Daten benötigt:

- versteckte Zustände
- Übergangsmatrix zwischen den versteckten Zuständen
- Beobachtungen

2 Modell für die studentische Teilnahme an Audience Response Quiz

- Beobachtungsmatrix, die angibt, wie eine Beobachtung die Werte der versteckten Zustände verändert
- Startverteilung

Diese Variablen werden im folgenden erklärt [13].

Versteckte Zustände werden für latente Variablen verwendet. Also für Variablen, die nicht direkt gemessen werden können. Die versteckten Zustände stellen eine Wahrscheinlichkeitsverteilung dar. Die Summe der Werte der versteckten Zustände ist daher immer Eins.

Die *Übergangsmatrix* enthält die Übergangswahrscheinlichkeiten zwischen den versteckten Zuständen. Die Größe der Übergangsmatrix ist quadratisch in der Anzahl der versteckten Zustände. Hier werden die Wahrscheinlichkeiten definiert, dass zu einem Schritt ein *plötzlicher* Übergang von einem versteckten Zustand zu einem anderen versteckten Zustand stattfindet. Die Übergangsmatrix repräsentiert, in welchen Zustand das Modell strebt.

Die *Beobachtungen* sind eine Liste an möglichen Daten, die gemessen werden können.

Die *Beobachtungsmatrix* gibt an, wie eine gemachte Beobachtung sich auf die Werte der versteckten Zustände auswirkt.

Die *Startverteilung* gibt an, wie die Wahrscheinlichkeitsverteilung der versteckten Zustände zu Beginn der Beobachtungen ist.

2.3.2 Beispiel eines HMM

Zur Veranschaulichung des HMM wird oft das *Gefangenen-Problem* verwendet, das auch hier zur Veranschaulichung herangezogen werden soll ⁹. Ein Gefangener sitzt in einem Verlies und kann nicht nach draußen sehen. Er hat keine Möglichkeit das Wetter zu erfahren, würde es dennoch gerne wissen. Daher beobachtet er die Schuhe des Wärters, ob diese schmutzig oder sauber sind. Zudem verwendet er das Wissen über das Wetter, das er bereits hat.

Die für den Gefangenen nicht zugänglichen Zustände sind *Sonne* und *Regen*.

Der Gefangene weiß, dass auf einen Regentag zu 50% ein sonniger Tag und auf einen sonnigen Tag zu 70% ein Regentag folgt.

Die Liste der möglichen Beobachtungen, die der Gefangene in seiner Situation machen kann sind {*Schuhe des Wärters sauber*; *Schuhe des Wärters schmutzig*}.

⁹http://de.wikipedia.org/wiki/Hidden_Markov_Model#Gefangener_im_Verlies

2.3 HMM als Generalisierung von BKT

Die möglichen Beobachtungen werden mit Hilfe der Beobachtungsmatrix auf die versteckten Zustände abgebildet. Wenn es draußen regnet sind die Schuhe des Wärters zu 90% schmutzig (und zu 10% sauber). Da der Wärter nicht besonders gründlich ist, sind seine Schuhe auch bei sonnigem Wetter zu 60% schmutzig (und zu 40% sauber).

Beobachtung	Sonne	Regen
Schuhe sauber	40	10
Schuhe schmutzig	60	90

Der Gefangene glaubt sich daran zu erinnern, dass am Tag seiner Gefangennahme sonniges Wetter war. Er ist sich jedoch nicht mehr ganz sicher.

Sonne	Regen
80	20

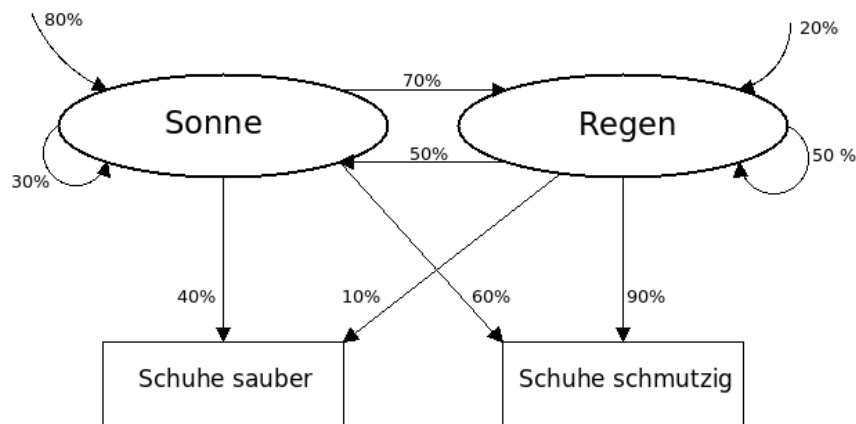


Abbildung 2.4: Beispiel Hidden Markov Model Quelle: http://de.wikipedia.org/wiki/Datei:Hidden_markov_model.svg

Um eine Berechnung durchführen zu können, ist eine Reihe von Beobachtungen nötig. z.B. **schmutzig; schmutzig; sauber** für die ersten 3 Tage nach seiner Gefangennahme.

Mit einem HMM sind verschiedene Berechnungen möglich. Wir haben hier ein fertiges Modell und eine Reihe von Beobachtungen gegeben und können daraus eine Vorhersage für die nächste Beobachtung machen. Hierzu wird der Forward-Algorithmus verwendet.

Man spricht davon, dass die versteckten Zustände die Beobachtungen generieren, also z.B. durch den Regen werden dreckige Schuhe generiert. Beispiel siehe Tabelle 2.4

2.3.3 BKT als HMM aufgebaut

BKT kann als HMM aufgefasst werden, daher soll hier kurz gezeigt werden, wie dieser HMM aufgebaut ist. Im BKT gibt es zwei versteckte Zustände, wobei nur ei-

2 Modell für die studentische Teilnahme an Audience Response Quiz

Tabelle 2.4: Beispiel HMM (R=Regen; S=Sonne)

n	Beobachtung	$P(S_{n-1})$	$P(S_n \text{Beobachtung})$	$P(S_n)$	$P(R_{n-1})$	$P(R_n \text{Beobachtung})$	$P(R_n)$	Ergebnis
1	schmutzig	0,80	$\frac{0,80 \times 0,60}{0,80 \times 0,60 + 0,20 \times 0,90} = 0,73$	$0,73 \times 0,30 + 0,27 \times 0,50 = \mathbf{0,35}$	0,20	$\frac{0,20 \times 0,90}{0,20 \times 0,90 + 0,80 \times 0,60} = 0,27$	$0,27 \times 0,50 + 0,73 \times 0,70 = \mathbf{0,65}$	Regen
2	schmutzig	0,35	$\frac{0,35 \times 0,60}{0,35 \times 0,60 + 0,65 \times 0,90} = 0,26$	$0,26 \times 0,30 + 0,74 \times 0,50 = \mathbf{0,45}$	0,65	$\frac{0,65 \times 0,90}{0,65 \times 0,90 + 0,35 \times 0,60} = 0,74$	$0,74 \times 0,50 + 0,26 \times 0,70 = \mathbf{0,55}$	Regen
3	sauber	0,45	$\frac{0,45 \times 0,40}{0,45 \times 0,40 + 0,55 \times 0,10} = 0,77$	$0,77 \times 0,30 + 0,23 \times 0,50 = \mathbf{0,35}$	0,55	$\frac{0,55 \times 0,10}{0,55 \times 0,10 + 0,45 \times 0,40} = 0,23$	$0,77 \times 0,70 + 0,23 \times 0,65 = \mathbf{0,65}$	Regen

2.4 Backstage-Feedback-System

ner berechnet wird *KNOWLEDGE*. Der zweite versteckte Zustand ist *NOT KNOWLEDGE* und wird bei der Berechnung außen vor gelassen, da er in BKT nicht von Relevanz ist.

Es gibt nur einen relevanten Übergang zwischen den versteckten Zuständen, von *NOT KNOWLEDGE* zu *KNOWLEDGE*. Dieser wird im BKT als Transferwahrscheinlichkeit ($P(T)$) bezeichnet. Die Richtung von *KNOWLEDGE* zu *NOT KNOWLEDGE* ist nicht relevant, da wenn einmal der Zustand *KNOWLEDGE* erreicht wurde, wird das BKT beendet.

Die im BKT möglichen Beobachtungen sind {CORRECT; FALSE}.

Die Beobachtungsmatrix besteht aus vier Werten, den Werten $P(G)$ (Guess) und $P(S)$ (Slip) und dem Wert von Eins minus $P(G)$ bzw $P(S)$.

Die Startverteilung ist im BKT immer so gewählt, dass *NOT KNOWLEDGE* überwiegt.

2.4 Backstage-Feedback-System

Das Backstage Feedback System gibt einem Studenten eine Rückmeldung über seine Performance in den Quiz der Vorlesung. Er bekommt dadurch eine Einschätzung, wie er im Vergleich zu seinen Kommilitonen steht. Dieses Feedback hat einen Mehrwert gegenüber einer einfachen Statistik, über die Anzahl der richtigen und falschen Antworten, da in diesem System die Fragen gewichtet werden. Zudem ist eine Statistik auf die Vergangenheit gerichtet, während das BFS auch als ein Prognose System für die Zukunft verwendet werden kann.

2.4.1 Zeitschritt

Dieses System ist ein dynamisches System, das von Zeitschritten abhängt. Jede gemachte Beobachtung ist ein Zeitschritt. Es stellt sich die Frage, wie in einem System in dem Fragen nicht zwingend beantwortet werden müssen, ein Zeitschritt definiert wird. Es wäre zum einen möglich für jeden Studenten eine individuelle Zeitleiste anzulegen, die mit jedem von ihm beantwortetem Quiz einen Schritt weiter bewegt. Zum anderen könnte man eine globale Zeitleiste anlegen und die Beobachtung, dass man keine Beobachtung gemacht hat, mit einbeziehen. Frei nach dem Axiomen von Paul Watzlawick „Man kann nicht nicht kommunizieren“¹⁰ besteht auch die Möglichkeit, eine fehlende Teilnahme an einem Quiz als Beobachtung zu sehen. In diesem Modell soll hierfür eine Beobachtung SKIP eingeführt werden, die anzeigt, dass der Student dieses Quiz ausgelassen hat.

2.4.2 Versteckte Zustände

Dies sind hier die möglichen Zustände, in denen sich ein Student befindet. Um eine höhere Aussagekraft als beim BKT zuzulassen, muss es mehr als zwei sein. Diese Werte sind Zustände eines Zustandsmodells, die den Wissensstand des Studenten repräsentieren. Für diese Anwendung wurden drei Stufen zur Qualifizierung des

¹⁰<http://www.paulwatzlawick.de/axiome.html>

2 Modell für die studentische Teilnahme an Audience Response Quiz

Studenten gewählt. Um noch übersichtlich arbeiten zu können wurden vorerst nur drei Zustände gewählt.

- Der Student hat keine Probleme der Vorlesung zu folgen (NO PROBLEMS).
- Der Student kann der Vorlesung teilweise folgen (SOME PROBLEMS).
- Der Student kann der Vorlesung nicht oder nur ungenügend folgen (MANY PROBLEMS).

Jeder dieser 3 Zustände hat einen Wert, zu welcher Wahrscheinlichkeit sich der Student gerade in diesem befindet.

Der Zustand mit dem höchsten Wert ist derjenige der am Ende der Berechnung dem Studenten als sein aktueller Zustand angezeigt wird.

2.4.3 Übergangsmatrix zwischen den versteckten Zuständen

Das HMM sieht vor, dass für jeden Zeitschritt, auch ein Übergang zwischen den versteckten Zuständen stattfindet. Hier stellt sich die Frage, wie die natürliche Entwicklung des Wissens eines Studenten ist? Diese Frage ist nicht ganz einfach zu beantworten, vor allem da wir im Kontext der Lehre arbeiten. Ziel der Vorlesung ist es, dass jeder Student zu Beginn kaum Wissen zu einem Thema hat und zum Ende der Vorlesung alles Gelehrte wiedergeben kann. Dies ist in der Praxis jedoch meist nicht der Fall.

Diese Frage zu beantworten ist äußerst anspruchsvoll. Es gibt keinen Zustand, in den der Student unweigerlich strebt. Es wird zu Beginn der Vorlesung eine Verteilung der Wissenszustände geben und auch am Ende, jedoch am Ende auf einem anderem Niveau. Der Wissensstand ist in diesem Modell immer relativ zum möglichen Wissen, verändert sich also auch mit jeder Veranstaltung.

Die Beantwortung dieser Frage findet daher auf einer pädagogischen Ebene statt. Zu Beginn der Vorlesung wird unterstellt, dass jeder Student das nötige Wissen hat (Vorwissen). Siehe hierzu auch Kapitel 2.4.6. Im Laufe der Zeit wird unterstellt, dass der Student schlechter wird, außer er kann durch die Teilnahme an den Quiz das Gegenteil *beweisen*.

Dies führt dann auch zu dem für den Studenten gut nachvollziehbaren Ergebnis, dass er am Ende der Vorlesung ohne Teilnahme an den Quiz auf einem sehr guten Zustand startet und bei mangelnder Teilnahme schlechter wird. Eine positive Entwicklung, wie bei BKT verwendet, empfiehlt sich in unserem System nicht. Bei BKT ist es so, dass immer eine positive Entwicklung stattfindet, auch wenn die Frage falsch beantwortet wird. Dies funktioniert bei BKT, da dort die Fragen immer zu einer Lerneinheit gestellt werden, bis diese gelernt sind, mit jeder Frage also das Wissen zu diesem Thema steigt. Auch das Erlernen, dass eine Antwort falsch ist, ist wieder neues Wissen. Würden wir dies in BFS genauso machen, tritt der Effekt auf, dass Studenten, die sich verbessern möchten, keine Antworten mehr abgeben. Dies

2.4 Backstage-Feedback-System

ist möglich, da die Beantwortung von Quiz in Backstage keine Pflicht ist, anders als in BKT-Systemen.

2.4.4 Beobachtungen

Als mögliche Beobachtungen werden die Quizergebnisse verwendet. Die Beobachtungen haben zwei Dimensionen:

- Das **Ergebnis** des Studenten in den Kategorien Richtig und Falsch.
- Sowie die **Schwierigkeit** des Quiz in Kategorien. Wie die Kategorisierung stattfindet wurde in Kapitel 2.1.1 bereits erklärt.

Zudem gibt es noch einen zusätzlichen Zustand der stellvertretend für keine Beobachtung steht. Für diesen Studenten liegt also zu diesem Quiz keine Antwort vor. Näheres hierzu findet sich in Kapitel 2.4.1.

Für jedes vom Dozenten veranstaltete Quiz wird bei jedem Studenten eine Beobachtung hinzugefügt. Die Beobachtungen sind eine Liste an möglichen Daten, die gemessen werden können.

Daraus ergeben sich folgende 7 möglichen Beobachtungen:

- EASY_TRUE: Der Student hat eine einfache Quizfrage richtig beantwortet.
- MEDIUM_TRUE: Der Student hat eine mittelschwere Quizfrage richtig beantwortet.
- HEAVY_TRUE: Der Student hat eine schwere Quizfrage richtig beantwortet.
- EASY_FALSE: Der Student hat eine einfache Quizfrage falsch beantwortet.
- MEDIUM_FALSE: Der Student hat eine mittelschwere Quizfrage falsch beantwortet.
- HEAVY_FALSE: Der Student hat eine schwere Quizfrage falsch beantwortet.
- SKIP: Der Student hat eine Quizfrage nicht beantwortet.

2.4.5 Beobachtungsmatrix

Diese Matrix gibt an, wie eine gemachte Beobachtung sich auf die Werte der einzelnen Zustände auswirkt. Es gibt sieben mögliche Beobachtungen in der Nachverfolgung der studentischen Leistungen. Diese möglichen Beobachtungen müssen mit Hilfe einer Tabelle auf die 3 versteckten Zustände abgebildet werden.

Hier soll allgemein gelten,

- eine schwere Frage richtig zu beantworten ist unwahrscheinlicher als eine leichte Frage

2 Modell für die studentische Teilnahme an Audience Response Quiz

- je niedriger die Wahrscheinlichkeit ist eine Frage richtig zu beantworten, desto eher führt sie zu einem besseren Zustand
- eine schwere Frage falsch zu beantworten ist wahrscheinlicher als eine einfache Frage falsch zu beantworten.

Die Beobachtungsmatrix hat folgenden Aufbau, dessen Werte es nun zu befüllen gilt:

Tabelle 2.5: Beobachtungsmatrix HMM

Beobachtung	NO PROBLEMS	SOME PROBLEMS	MANY PROBLEMS
EASY_TRUE	a	b	c
MEDIUM_TRUE	d	e	f
HEAVY_TRUE	g	h	i
EASY_FALSE	j	k	l
MEDIUM_FALSE	m	n	o
HEAVY_FALSE	p	q	r
SKIP	s	t	u

2.4 Backstage-Feedback-System

2.4.5.1 Constraints

Aus den oben genannten Grundsätzen folgen diese Constraints:

- Monotonie:
 - Es ist wahrscheinlicher eine Frage der Kategorie EASY richtig zu beantworten als eine Frage der Kategorie MEDIUM, daher muss sich bei einer richtigen Antwort auf eine Frage der Kategorie MEDIUM das Ergebnis mehr verbessern als bei einer richtigen Antwort auf eine Frage der Kategorie EASY
 - Entsprechendes gilt von MEDIUM zu HEAVY und von EASY zu HEAVY
 - Für den Bereich FALSE ist es genau umgekehrt. Eine Frage der Kategorie EASY falsch zu beantworten ist schlimmer als eine der Kategorie MEDIUM
 - Dies gilt analog für die Kategorien MEDIUM_FALSE und HEAVY_FALSE.
- Robustheit
 - Sind alle Antworten TRUE muss der Student zu jedem Zeitpunkt im Zustand NO PROBLEMS sein
 - Es sollen keine Sprünge möglich sein, also es soll nie möglich sein, durch ein Quiz vom Zustand NO PROBLEMS in den Zustand MANY PROBLEMS und umgekehrt zu wechseln
 - Das System soll eine gewisse Stabilität erhalten, also nicht das letzte Quiz Ergebnis wiedergeben, sondern es soll für den Studenten etwas mühsam sein, sich von einem Zustand in den nächsten hoch zu arbeiten.
- Symmetrie
 - Mit einer Antwort aus dem Bereich FALSE darf sich das Ergebnis nie verbessern und mit einer Antwort aus dem Bereich TRUE nie verschlechtern.

Unter Beachtung all dieser Prinzipien ist der Spielraum, in dem die Werte für die Variablen a bis u aus Tabelle 2.5 liegen, bereits eingeschränkt.

Wie die einzelnen Werte in dieser Implementierung gewählt wurden, findet sich in Kapitel 6.2.

2.4.6 Startverteilung

Zu Beginn der Vorlesung wird unterstellt, dass jeder Student das nötige Wissen hat (Vorwissen). Dies stellt eine Annahme dar, die nicht auf alle Studenten zutreffen wird. Diese Annahme wurde teilweise auch aus pädagogischen Gründen getroffen, um gleich zu Anfang eine Motivation zu setzen. Die Studenten sollen

2 Modell für die studentische Teilnahme an Audience Response Quiz

motiviert sein, ihren Anfangs hohen Status zu behalten und nicht abzufallen (Reinforcement).

Wie in Kapitel 2.4.3 bereits angesprochen, sind Werte, die sich auf die versteckten Zustände beziehen, nur schwer zu ermitteln. Es handelt sich um latente Variablen, die nicht direkt ermittelt werden können. Daher ergeben sich auch für die Berechnung der Startverteilung gewisse Probleme.

Die Startverteilung ist in erster Linie für den Beginn des Semesters von Bedeutung, nach den ersten Quiz spielt sie keine Rolle mehr.

Die Ermittlung der Werte für die Startverteilung könnte am ehesten mit einem Eignungstest oder einem ähnlichen kurzen Test am Anfang der Vorlesung berechnet werden. Da der Aufwand hierfür jedoch viel zu hoch wäre, ist von dieser Lösung in der Praxis wenig zu halten. Daher wäre es wohl am einfachsten eine Gleichverteilung anzunehmen. Also jeder Student hat zu Beginn 33% in jedem Zustand. Dies ist jedoch nur eine willkürliche Verteilung, wie es jede andere auch wäre.

Details zur Umsetzung der Startverteilung sind in Kapitel 6.1 zu finden.

2.4.7 Graphische Darstellung des Backstage Feedback Systems

In Abbildung 2.5 ist zu sehen, wie das fertige Backstage Feedback System aussieht. In dieser Darstellung wird klar, wie die Komplexität des Systems mit jedem zusätzlichen versteckten Zustand und jeder zusätzlichen möglichen Beobachtung steigt.

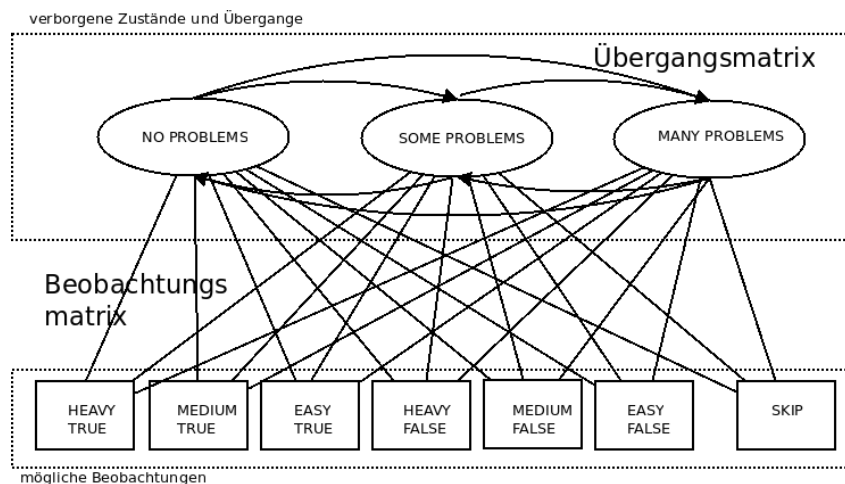


Abbildung 2.5: Darstellung des Backstage Feedback Systems

Mögliche Implementierung

In diesem Kapitel soll beschrieben werden, wie eine produktive Implementierung des Systems gemacht werden könnte.

3.1 Architektur: Mögliche Anbindung an Backstage

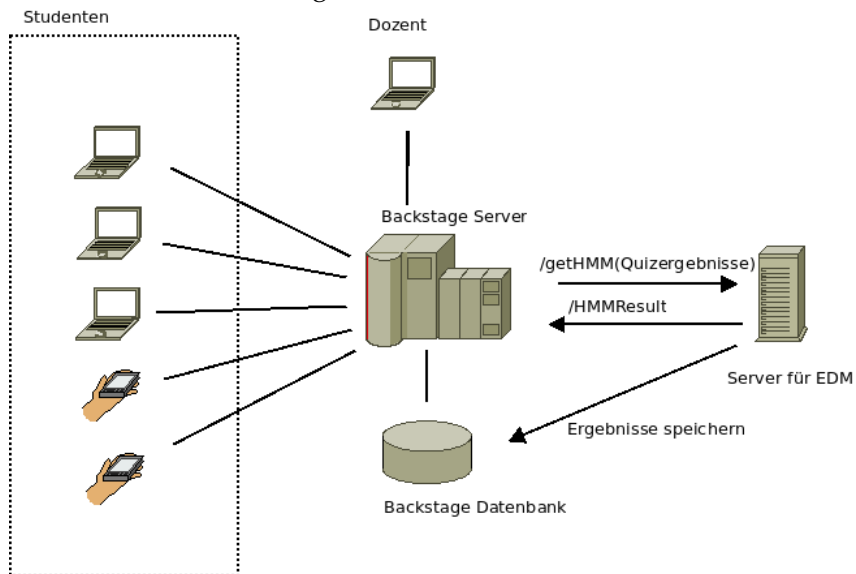
Bei der Implementierung des Systems ist es wichtig, dass durch den Rechenaufwand, den der Data-Mining Prozess benötigt, keinesfalls die Leistung und Erreichbarkeit von Backstage leidet. Obwohl das letztendlich implementierte System auf einem HMM beruht, der äußerst sparsam mit Rechnerzeit umgeht, wurde dieses Ziel beibehalten um es für eventuell andere EDM Anwendungen bereits als Plattform zur Verfügung zu stellen. Der Datamining Prozess sollte hierzu auf einen eigenen Server ausgelagert werden. Mit diesem kommuniziert der Backstageserver über eine HTTP-Schnittstelle. Siehe Abbildung 3.1

3.2 möglicher Zeitpunkt der Berechnung

Algorithmen aus dem Bereich des Data-Mining sind oft sehr laufzeit-intensiv, daher stellt sich die Frage wie oft soll diese Berechnung durchgeführt werden und vor allem wann. Hierfür gäbe es verschiedene Möglichkeiten: Es wäre denkbar, nach jeder Vorlesungseinheit vom Dozenten gesteuert das Feedbacksystem zu starten. Die Ergebnisse wären dann nach dem Ende der Berechnung für jeden Studenten sichtbar. Hier wäre es auch möglich einen festen Zeitpunkt zu wählen (jeden Montag morgen/ jeden morgen/ alle 2 Stunden etc.).

Eine automatische Auswertung nach jedem Quiz erschien uns nicht als sinnvoll, da hier nur unnötig Rechenlast generiert wird.

Abbildung 3.1: Technische Architektur



Als beste Lösung erschien es uns, die Daten für jeden Studenten auf Anfrage zu berechnen und die für den Dozenten als Anfrage an alle Studenten zu stellen. Um Rechenzeit zu sparen werden nicht nur die Ergebnisse ausgegeben, sondern auch die Zwischenergebnisse gespeichert, so dass keine Berechnung mehrmals durchgeführt werden muss. Es werden jeweils die Daten der bisherigen Berechnung verwendet und die Beobachtungen nach dem gespeicherten Zeitstempel berücksichtigt.

In diesem Kapitel wird erklärt, wie das Backstage Feedback System bisher evaluiert wurde.

4.1 Evaluierungsimplementierung

Ein Teil der in dieser Arbeit beschriebenen Verfahren wurde implementiert. Die Implementierung ist nicht für einen produktiven Einsatz in Backstage gedacht. Sie soll lediglich eine Möglichkeit bieten, ein Gefühl für die gewählten Werte zu bekommen.

4.1.1 Konfiguration

Um die Konfiguration des Systems so einfach wie möglich zu machen, sind alle Werte die für den HMM benötigt werden nicht fest im Programm codiert, sondern liegen in einer Textdatei. So können sie sobald es eine Studie oder Erfahrungswerte gibt möglichst einfach angepasst werden.

Das Einlesen der Textdatei geschieht einmalig bei der ersten Verwendung des Systems. Ändern sich die Daten, kann entweder das System neu gestartet werden oder über einen HTTP-Request das erneute einlesen der Datei angestoßen werden.

Die für den Test erstellte Konfigurationsdatei befindet sich im Anhang.

4.1.2 Beschreibung

Die Implementierung wurde in Java¹¹ erstellt. Als Web Framework diente Play!¹², sowie für die Umsetzung des HMM wurde die Bibliothek JAHMM¹³ verwendet.

Der Anwender übergibt dem *EDM-Server* eine Liste an Beobachtungen.

Um die Performance zu steigern und Rechenleistung zu sparen, ist es möglich, ebenfalls eine Startverteilung zu übergeben. (Diese ist das Ergebnis der vorherigen Berechnung). Wird keine Startverteilung übergeben, verwendet der *EDM-Server* die Startverteilung, die ihm aus dem HMM bekannt ist. Wird hingegen eine Startverteilung übergeben, wird diese verwendet.

4.1.3 Oberfläche für Tests

Um das Testen der Werte möglichst einfach zu machen, wurde eine Weboberfläche geschrieben, die genau die Funktionalität bietet, wie sie auch dem Backstageserver zur Verfügung steht. Sie ist bewusst einfach gehalten, da sie nur zum Testen dient (siehe Abbildung 4.1).

Die ersten drei Eingabefelder sind optional für die Startverteilung. Hier können Werte von 0 bis 100 eingegeben werden, diese werden als Prozentangabe interpretiert. Die Summe der drei Werte muss 100 ergeben. Das letzte Eingabefeld ist für eine oder mehrere Beobachtungen gedacht. Die Beobachtungen werden hier in Kleinbuchstaben codiert (siehe Tabelle 4.1).

Tabelle 4.1: Codierung für Beobachtungen

Beobachtung	Codierung
SKIP	z
EASY_FALSE	d
MEDIUM_FALSE	e
HEAVY_FALSE	f
EASY_TRUE	a
MEDIUM_TRUE	b
HEAVY_TRUE	c

Eine Aufbereitung der Rückgabewerte findet nicht statt. Diese können in der Konsole betrachtet werden. Hier werden zudem auch alle Zwischenergebnisse ausgegeben.

¹¹<http://www.java.com/>

¹²<http://www.playframework.com/>.

¹³<https://code.google.com/p/jahmm/>.

4.2 Evaluierung anhand synthetischer Daten

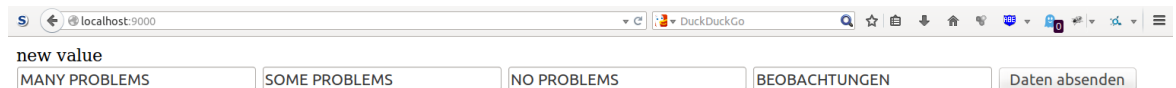


Abbildung 4.1: Testoberfläche

4.2 Evaluierung anhand synthetischer Daten

Um zu zeigen, wie sich das System verhält ein paar ausgewählte Beispiele. Diese sind etwas eingänglicher als eine rein theoretische Beschreibung.

In den folgenden Tabellen bedeuten die Überschriften jeweils:

Tabelle 4.2: Aufbau Tabellen

Zeile	Dies ist zum einen eine laufende Nummerierung und andererseits der Wert, wie oft die Übergangsmatrix angewandt wurde.
Beobachtung	Dies ist die Beobachtung, die zu diesem Zeitpunkt gemacht wurde.
Wert für <i>MANY</i>	Dies ist der Wert, den der Algorithmus für <i>der Student hat viele Probleme der Vorlesung zu folgen</i> berechnet
Wert für <i>SOME</i>	Dies ist der Wert, den der Algorithmus für <i>der Student hat einige Probleme der Vorlesung zu folgen</i> berechnet
Wert für <i>NO</i>	Dies ist der Wert, den der Algorithmus für <i>der Student hat keine Probleme der Vorlesung zu folgen</i> berechnet
Ergebnis	Dies ist die Spalte mit dem Höchsten Wert und damit das was dem Studenten als sein Feedback angezeigt wird

4.2.1 Student, der sich nicht an Quiz beteiligt

Hier wird ein Student simuliert, der sich nicht an den Quiz beteiligt. Die Veränderungen der Werte in dieser Tabelle beruhen auf der Übergangsmatrix. Die Beobachtungsmatrix verhält sich bei der Beobachtung *SKIP* neutral.

Auswertung siehe Tabelle 4.3. Hier ist zu sehen, dass beim zehnten Quiz, an dem sich der Student nicht beteiligt, sein Feedback auf *SOME PROBLEMS* herunter gestuft wird und nach weiteren elf Quiz auf *MANY PROBLEMS*.

4.2.2 Student, der *eh*er gut ist

Das Beispiel in Tabelle 4.4 soll dazu dienen zu zeigen, wie sich der Verlauf des Zustandes mit ein paar Antworten verändert und die Stabilität des Systems darzustellen.

Tabelle 4.3: Student ohne Quizteilnahme

Zeile	Beobachtung	MANY	SOME	NO	Ergebnis
0	SKIP	0,300000	0,300000	0,400000	NO PROBLEMS
1	SKIP	0,303000	0,305000	0,392000	NO PROBLEMS
2	SKIP	0,306050	0,309790	0,384160	NO PROBLEMS
3	SKIP	0,309147	0,314375	0,376476	NO PROBLEMS
4	SKIP	0,312291	0,318761	0,368947	NO PROBLEMS
5	SKIP	0,315479	0,322952	0,361568	NO PROBLEMS
6	SKIP	0,318708	0,326954	0,354336	NO PROBLEMS
7	SKIP	0,321978	0,330771	0,347250	NO PROBLEMS
8	SKIP	0,325286	0,334408	0,340305	NO PROBLEMS
9	SKIP	0,328630	0,337870	0,333499	SOME PROBLEMS
10	SKIP	0,33200	0,341162	0,326829	SOME PROBLEMS
11	SKIP	0,335420	0,344286	0,320292	SOME PROBLEMS
12	SKIP	0,338860	0,347249	0,313886	SOME PROBLEMS
13	SKIP	0,342335	0,350055	0,307608	SOME PROBLEMS
14	SKIP	0,345836	0,352706	0,301456	SOME PROBLEMS
15	SKIP	0,349363	0,355208	0,295427	SOME PROBLEMS
16	SKIP	0,352915	0,357565	0,289519	SOME PROBLEMS
17	SKIP	0,356491	0,359780	0,283728	SOME PROBLEMS
18	SKIP	0,360088	0,361856	0,278054	SOME PROBLEMS
19	SKIP	0,363707	0,363799	0,272493	SOME PROBLEMS
20	SKIP	0,367345	0,365611	0,267043	MANY PROBLEMS
21	SKIP	0,371001	0,367296	0,261702	MANY PROBLEMS
22	SKIP	0,374674	0,368857	0,256468	MANY PROBLEMS
23	SKIP	0,378363	0,370297	0,251338	MANY PROBLEMS
24	SKIP	0,382066	0,371621	0,246312	MANY PROBLEMS
25	SKIP	0,385782	0,372831	0,241385	MANY PROBLEMS
26	SKIP	0,389510	0,373931	0,236558	MANY PROBLEMS
27	SKIP	0,393250	0,374922	0,231827	MANY PROBLEMS
28	SKIP	0,396999	0,375810	0,227190	MANY PROBLEMS
29	SKIP	0,400757	0,376595	0,222646	MANY PROBLEMS
30	SKIP	0,404523	0,377282	0,218193	MANY PROBLEMS
31	SKIP	0,408296	0,377874	0,213829	MANY PROBLEMS
32	SKIP	0,412074	0,378371	0,209553	MANY PROBLEMS
33	SKIP	0,415858	0,378779	0,205362	MANY PROBLEMS

4.2 Evaluierung anhand synthetischer Daten

Dies sind die Beobachtungen für einen rein fiktiven Studenten: *SEQUENCE* = [MEDIUM_TRUE, EASY_TRUE, HEAVY_TRUE, MEDIUM_FALSE, MEDIUM_TRUE, HEAVY_FALSE, HEAVY_TRUE]

Obwohl der Student zwei Fragen (MEDIUM_FALSE und HEAVY_FALSE) falsch beantwortet hat, ist er immer noch im Zustand *hat keine Probleme der Vorlesung zu folgen*.

Tabelle 4.4: Student, der *eher* gut ist

Zeile	Beobachtung	MANY	SOME	NO	Ergebnis
0	MEDIUM_TRUE	0,106508	0,301775	0,591715	NO PROBLEMS
1	EASY_TRUE	0,048504	0,309482	0,642012	NO PROBLEMS
2	HEAVY_TRUE	0,009254	0,229025	0,761719	NO PROBLEMS
3	MEDIUM_FALSE	0,036474	0,397516	0,566009	NO PROBLEMS
4	MEDIUM_TRUE	0,011561	0,327860	0,660578	NO PROBLEMS
5	HEAVY_FALSE	0,028151	0,480611	0,491237	NO PROBLEMS
6	HEAVY_TRUE	0,006307	0,371773	0,621919	NO PROBLEMS

4.2.3 Student der sich verbessert

In Tabelle 4.5 ist zu sehen, wie sich das System bei einem Anfangs eher schlechten Studenten verhält, der im Laufe der Zeit besser wird. Hier wurde untersucht, wie lange es dauert bis sich ein schlechtes Ergebnis wieder verbessern lässt. *SEQUENCE* = [EASY_TRUE, EASY_FALSE, MEDIUM_FALSE, EASY_TRUE, HEAVY_FALSE, MEDIUM_TRUE, EASY_TRUE, MEDIUM_TRUE, EASY_TRUE, MEDIUM_TRUE, MEDIUM_TRUE, HEAVY_TRUE]. Der Student ist nach drei falschen Ergebnissen knapp im Zustand MANY PROBLEMS und kann sich mit einem richtigen Quiz wieder in SOME PROBLEMS verbessern, um dann nach sechs weiteren richtigen Quiz wieder im Zustand NO PROBLEMS zu sein.

Tabelle 4.5: Student der sich verbessert

Zeile	Beobachtung	MANY	SOME	NO	Ergebnis
0	EASY_TRUE	0,151898	0,341772	0,506329	NO PROBLEMS
1	EASY_FALSE	0,418866	0,382883	0,198250	MANY PROBLEMS
2	MEDIUM_FALSE	0,632310	0,297938	0,069751	MANY PROBLEMS
3	EASY_TRUE	0,431296	0,452686	0,116017	SOME PROBLEMS
4	HEAVY_FALSE	0,532035	0,412445	0,055518	MANY PROBLEMS
5	MEDIUM_TRUE	0,278826	0,603279	0,117894	SOME PROBLEMS
6	EASY_TRUE	0,148147	0,701634	0,150218	SOME PROBLEMS
7	MEDIUM_TRUE	0,056522	0,720031	0,223445	SOME PROBLEMS
8	EASY_TRUE	0,028638	0,725330	0,246031	SOME PROBLEMS
9	MEDIUM_TRUE	0,011619	0,663154	0,325226	SOME PROBLEMS
10	MEDIUM_TRUE	0,005659	0,582534	0,411805	SOME PROBLEMS
11	HEAVY_TRUE	0,002262	0,461007	0,536729	NO PROBLEMS

4.3 Ergebnis der Evaluierung

Die Evaluierung hat gezeigt, dass das BFS ein Ergebnis liefert, dass der Intuition entspricht. Es ist möglich, die Ergebnisse zu verstehen, auch wenn man das dahinter liegende System nicht kennt. Dies ist so gewünscht, um die Akzeptanz bei den Studenten zu erhöhen. Die Ergebnisse entsprachen dem, was wir erwartet hatten.

In diesem Kapitel wird dargestellt, welche Ergebnisse diese Arbeit erbrachte und wie sie möglicherweise weitergeführt werden könnte.

5.1 Zusammenfassung der Arbeit

Diese Arbeit hat gezeigt, wie man für den digitalen Backchannel Backstage ein Feedbacksystem basierend auf der studentischen Quizteilnahme erstellen könnte. Es gibt bereits Arbeiten in verwandten Gebieten des Educational Data Mining (EDM), auf die man hierfür aufbauen kann [4, 18], deren Eignung für ein Feedbacksystem in Backstage keineswegs offensichtlich ist und folglich noch untersucht werden sollte. Es zeigt sich, dass das Setzen der Parameter äußerst komplex und aufwendig ist. Wie gut mit ungenauen Parametern gearbeitet werden kann ist bisher noch nicht ersichtlich.

5.2 Vorteile gegenüber dem BKT

Um den speziellen Anforderungen eines Feedbacksystems für Backstage gerecht zu werden, wurde das Backstage Feedback System (BFS) entwickelt. Der größte Unterschied zwischen dem BKT und dem BFS liegt darin, dass das BFS *Vergessen* zulässt. In der Praxis ist es so, dass Wissen das nicht regelmäßig verwendet wird, auch wieder vergessen wird. Daher muss der Student im BFS ständig zeigen, dass er gerade auf dem Laufenden ist. Im BKT wird einmal Gelerntes immer als korrekt angenommen. Im BFS muss der Student regelmäßig beweisen, dass er wirklich gut ist.

Der zweite große Unterschied ist, dass BFS ein Feedback mit mehreren Zuständen gibt und eben nicht nur ein positives oder negatives Feedback.

BKT wird in tutoriellen Systemen verwendet, um zu überprüfen, ob ein Student die notwendigen Lernleistungen erbracht hat um Zugang zur nächsten Lerneinheit zu erhalten. Hierfür ist eine dichotome Entscheidung („ja“ / „nein“) ausreichend. BFS versucht im Gegensatz zu BKT, eine mehrstufige Rückmeldung geben zu können, ob ein Student eine Lerneinheit gar nicht, gut, oder sehr gut abgeschlossen hat. Im BFS ist es nicht nötig, dass eine Entscheidung getroffen wird.

5.3 Grenzen des Modells

Wie schon angesprochen hat dieses Feedbacksystem Grenzen. Diese liegen insbesondere an einer nicht ausreichenden Menge an Daten. Es wäre daher denkbar, dass das Feedbacksystem erst ab einer gewissen Genauigkeit, also einer gewissen Anzahl an Daten aktiviert wird. Diese Datenmenge könnte über verschiedene Parameter bestimmt werden. Zum Beispiel könnte man pauschal keine Berechnung durchführen, so lange die Anzahl der Ergebnisse einen gewissen Grenzwert nicht überschritten hat. Es wäre auch möglich eine prozentuale Grenze einzuführen. Nur Studenten, die mehr als 30% der Quiz beantwortet haben, bekommen ein Feedback angezeigt. Zudem wäre es möglich eine Kombination aus beiden Techniken zu verwenden.

Alternativ wäre es möglich, auch eine komplett andere Herangehensweise zu wählen. Da wir in dieser Arbeit bereits mit HMM arbeiten, wäre es naheliegend auch einen HMM zu bauen, der die Entscheidung trifft, ob bereits genug Daten für ein Feedback vorliegen. Dieses Konzept wurde nicht ausgearbeitet. Der Umfang würde vermutlich eine eigene Arbeit ergeben.

5.4 Weiterführende Arbeiten

Im Laufe der Arbeit wurden bereits einige Aspekte angesprochen wie diese Arbeit weitergeführt werden könnte. Hier noch mal eine Zusammenfassung der möglichen Themen.

Im Rahmen von Human Computation wäre es möglich, dass die Einschätzung der Fragen, also die Kategorie in der diese eingeordnet werden (siehe Kapitel 2.1.1) von den Studenten gemacht wird. Hier wäre ein großer Vorteil, dass somit Gesamtleistungen der Studenten besser gewürdigt werden könnten. Eine schwere Frage, die alle Studenten richtig beantwortet haben, wird bei diesem System abgewertet und wie eine einfache Frage bewertet.

Ein weiterer interessanter Gedanke wäre, das System adaptiv (selbstlernend) zu machen. Der Student hat die Möglichkeit regelmäßig (oder einmalig) eine Selbsteinschätzung abzugeben und basierend auf dieser werden die Werte für das HMM angepasst. Dies könnte das System dazu bringen sich eher mit der Meinung der Studenten zu decken. Dies birgt jedoch die Gefahr, dass hier verhindert wird, dass das System etwas erkennt, was dem Studenten selbst noch gar nicht bewusst ist.

5.4 Weiterführende Arbeiten

Das System würde bestraft werden, wenn es ein präziseres Feedback gibt, als es die Selbsteinschätzung ist.

Die Darstellung des berechneten Ergebnisses muss noch konzipiert werden. Bisher ist lediglich eine einfache Textausgabe in der Konsole implementiert. Diese könnte mit Mitteln aus dem Bereich Gamification aufgewertet werden. Erste Ideen hierfür sind z.B. die Anzeige durch eine Art Wettervorhersage zu visualisieren. Eine gute Performance führt zu einer *sonnigen Vorhersage* und entsprechend bei schlechterem Ergebnis *Wolken* bis *Gewitter*. Hier sind auch zahlreiche andere Darstellungsformen denkbar.

Sollte auf diesem Gebiet weitergearbeitet werden ist es unerlässlich die Werte zu evaluieren.

Ein sehr vielversprechendes Thema öffnet sich, wenn das Feedbacksystem im Einsatz ist. Dann könnte mit Hilfe des Feedbacksystems das Audience Response System verbessert werden. Es wäre gegebenenfalls möglich verschiedene Varianten einer Frage zu stellen und jeder Student bekommt eine auf ihn zugeschnittene Frage.

Informationen, wie die Werte für die Evaluierung gewählt wurden.

6.1 Implementierung der Startverteilung

Das Ergebnis, wie es derzeit implementiert ist, ist in Tabelle 6.1 zu sehen.

Diese Startverteilung ist der Grund, warum der Student mehrere Male am Quiz nicht teilnehmen kann (Beobachtung *SKIP*) und trotzdem noch im Zustand *NO PROBLEMS* ist.

Tabelle 6.1: Startverteilung

Variable	Wert
NO PROBLEMS	0,4
SOME PROBLEMS	0,3
MANY PROBLEMS	0,3

6.2 Implementierung der Beobachtungsmatrix

Die Implementierung wurde mit folgenden Werten vorgenommen. Diese basieren auf den in Kapitel 2.4.5.1 genannten Constraints.

Die Werte für *NO PROBLEMS* und *MANY PROBLEMS* sind identisch nur in umgekehrter Reihenfolge.

Die Tabelle ist so zu lesen: Wenn sich der Student im Zustand *NO PROBLEMS* befindet, dann ist die nächste Beobachtung zu 4% *EASY_FALSE* und zu 27% *HEAVY_TRUE*.

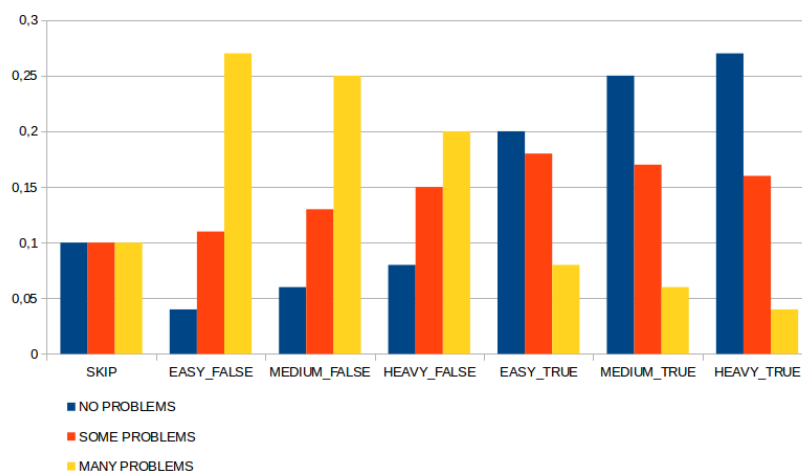
Tabelle 6.2: Beobachtungsmatrix implementiert

Beobachtung	NO PROBLEMS	SOME PROBLEMS	MANY PROBLEMS
SKIP	0,10	0,10	0,10
EASY_FALSE	0,04	0,11	0,27
MEDIUM_FALSE	0,06	0,13	0,25
HEAVY_FALSE	0,08	0,15	0,20
EASY_TRUE	0,20	0,18	0,08
MEDIUM_TRUE	0,25	0,17	0,06
HEAVY_TRUE	0,27	0,16	0,04

Wir verwenden diesen Prozess jedoch in die andere Richtung. Der Student hat keinen Einfluss auf die Schwierigkeit der Frage und somit der vollen Beobachtung. Wir verwenden die Beobachtung als gegeben und berechnen daraus den wahrscheinlichsten Zustand.

In Abbildung 6.1 ist der Verlauf der Werte für die verschiedenen Beobachtungen noch einmal anschaulich dargestellt.

Abbildung 6.1: Beobachtungsmatrix in der derzeitigen Implementierung



6.3 Implementierung der Übergangsmatrix

Die Übergangsmatrix hat in dem von uns gewählten System kaum eine Auswirkung auf das Gesamtergebnis. Die Wahrscheinlichkeit, dass der Student im aktuellen Zustand bleibt ist mit 98% bzw 99% die Höchste, dass der Student um eine Stufe absteigt liegt bei 1% bzw 2%. Dies soll darstellen, dass der Student im Laufe der Zeit Wissen wieder vergisst, also muss er mit korrekten Antworten das Gegenteil zeigen. Um ein Gefühl für diese Werte zu bekommen folgendes Beispiel, das

6.4 Konfigurationsdateisdatei

gilt falls es keine Beobachtungen gäbe. Nach den ungefähr ersten 7 Quiz, an denen er nicht teilnimmt wird sein Rating von *NO PROBLEMS* auf *SOME PROBLEMS* herabgestuft. Nach weiteren 11 Quiz wird sein Zustand auf *kann der Vorlesung nicht Folgen* gesetzt. Gedanke hinter diesen Werten sind, dass die ersten sieben Quiz ca. zwei Vorlesungen entsprechen dürften. In dieser Zeit sollten die meisten ihren Zugang zu Backstage haben. Siehe auch das Beispiel in Tabelle 4.3 in der dieser Verlauf genau dargestellt wird.

Tabelle 6.3: Übergangsmatrix implementiert

	NO PROBLEMS	SOME PROBLEMS	MANY PROBLEMS
NO PROBLEMS	0,98	0,02	0,00
SOME PROBLEMS	0,00	0,99	0,01
MANY PROBLEMS	0,00	0,00	1,00

6.4 Konfigurationsdateisdatei

Dies ist die Konfigurationsdateisdatei, die für die Evaluierung in Kapitel 4 verwendet wurde.

```
#This file is to declare the variables
#for the educational data mining process
#
#Lines started with "#" are comments
#
#HM Feedbacksystem
####
# startpropability
#not able to follow lecture
pOfMANY 0.3
#partially able to follow lecture
pOfSOME 0.3
#is fully able to follow lecture
pOfNO 0.4
####
# probability of transition from state i to state j
#
pOfMANYFollowMANY 1
pOfMANYFollowSOME 0
pOfMANYFollowNO 0
#
pOfSOMEFollowMANY 0.01
pOfSOMEFollowSOME 0.99
pOfSOMEFollowNO 0.0
#
```

```
pOfNOFollowMANY 0
pOfNOFollowSOME 0.02
pOfNOFollowNO 0.98
#
####
# probability distribution of an observation ,
#parametrized on observation
# observations are in the order
# EASY_TRUE, MEDIUM_TRUE, HEAVY_TRUE,
#EASY_FALSE, MEDIUM_FALSE, HEAVY_FALSE, SKIP;
pOfMANYDependent 0.08 0.06 0.04 0.27 0.25 0.2 0.10
pOfSOMEDependent 0.18 0.17 0.16 0.11 0.13 0.15 0.10
pOfNODependent 0.2 0.25 0.27 0.04 0.06 0.08 0.10
```

Literaturverzeichnis

- [1] Ryan Baker, Albert Corbett, Sujith Gowda, Angela Wagner, Benjamin MacLaren, Linda Kauffman, Aaron Mitchell, and Stephen Giguere, *Contextual slip and prediction of student performance after use of an intelligent tutor*, User Modeling, Adaptation, and Personalization, 2010, pp. 52–63.
- [2] Joseph Beck, *Difficulties in inferring student knowledge from observations (and why you should care)*, Educational Data Mining: Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education, 2007, pp. 21–30.
- [3] Albert Corbett and John Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge*, User Modeling and User-Adapted Interaction (1994), no. 4, 253–278.
- [4] Albert Corbett and Akshat Bhatnagar, *Student modeling in the act programming tutor: Adjusting a procedural learning model with declarative knowledge*, User Modeling, Springer, 1997, pp. 243–254.
- [5] Vera Gehlen-Baum, Alexander Pohl, Armin Weinberger, and François Bry, *Backstage – designing a backchannel for large lectures*, Proceedings of the European Conference on Technology Enhanced Learning, Saarbrücken, Germany (18-21 September 2012), 2012.
- [6] Yue Gong, Joseph Beck, and Neil Heffernan, *Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures*, Intelligent Tutoring Systems (Vincent Aleven, Judy Kay, and Jack Mostow, eds.), Lecture Notes in Computer Science, vol. 6094, Springer Berlin Heidelberg, 2010, pp. 35–44.
- [7] Julia Hadersberger, Alexander Pohl, and François Bry, *Discerning Actuality in Backstage – Comprehensible Contextual Aging*, Proceedings of the European Conference on Technology Enhanced Learning, Saarbrücken, Germany (18-21 September 2012), 2012.

- [8] Robin H Kay and Ann LeSage, *Examining the benefits and challenges of using audience response systems: A review of the literature*, *Computers & Education* **53** (2009), no. 3, 819–827.
- [9] Alison King, *Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain*, *American educational research journal* Vol. 31 (1994), no. 2, 338–368.
- [10] Zachary Pardos and Neil Heffernan, *Kt-idem: Introducing item difficulty to the knowledge tracing model*, *User Modeling, Adaption and Personalization*, Springer, 2011, pp. 243–254.
- [11] A. Pohl, F. Bry, J. Schwarz, and M. Gottstein, *Sensing the classroom: Improving awareness and self-awareness of students in Backstage*, 15th. International Conference on Interactive Collaborative Learning (ICL), 2012, pp. 1–8.
- [12] Alexander Pohl, Vera Gehlen-Baum, and Francois Bry, *Enhancing the digital backchannel backstage on the basis of a formative user study*, *International Journal of Emerging Technologies in Learning (iJET)* (2012), pp–33.
- [13] Lawrence Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, *Proceedings of the IEEE* Vol. 77 (1989), no. 2, 257–286.
- [14] Lawrence Rabiner and Biing-Hwang Juang, *An introduction to hidden Markov models*, *ASSP Magazine, IEEE* Vol. 3 (1986), no. 1, 4–16.
- [15] Cristóbal Romero and Sebastián Ventura, *Educational data mining: a review of the state of the art*, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **40** (2010), no. 6, 601–618.
- [16] Brett van de Sande, *Properties of the bayesian knowledge tracing model*, *JEDM-Journal of Educational Data Mining*, Vol. 5 (2013), no. 2, 1–10.
- [17] Varga and Moore, *Hidden markov model decomposition of speech and noise*, *Acoustics, Speech, and Signal Processing, IEEE*, 1990, pp. 845–848.
- [18] Amelia Zafra and Sebastián Ventura, *Predicting student grades in learning management systems with multiple instance genetic programming.*, *International Working Group on Educational Data Mining* (2009), 309–319.