

# Transfer von Ergebnissen aus dem Gebiet der Künstlichen Intelligenz in Spielen in die Medizinische Informatik zur Anwendung in entscheidungsunterstützenden Systemen

## Transferring results from the field of artificial intelligence in games to medical informatics for application in decision support systems

### Abstract

This paper considers the transfer of approaches from the field of *artificial intelligence in games* to the field of *medical informatics* to be used in decision support systems. The most relevant of the developed approaches will be roughly introduced and it will be considered how these approaches can be applied in medical informatics in the context of decision support systems. Moreover, it will be considered which of the approaches are eligible in this context. Apart from placing the approaches in the context of decision support systems, the paper thereby also provides an example for the interplay of different fields of research.

**Keywords:** artificial intelligence, machine learning, decision support, comprehensibility

### Zusammenfassung

Dieser Artikel behandelt den Transfer von Ansätzen aus dem Gebiet der *Künstlichen Intelligenz in Spielen* in die *Medizinische Informatik* zum Einsatz im Bereich entscheidungsunterstützender Systeme. Die wesentlichen in den entsprechenden Vorarbeiten entstandenen Ansätze werden zunächst grob eingeführt und es wird beleuchtet, wie diese in der Medizinischen Informatik im Kontext von entscheidungsunterstützenden Systemen Anwendung finden können. Darüber hinaus wird dargelegt, welche der in den Vorarbeiten entstandenen Ansätze für diesen Kontext geeignet sind. Damit liefert der Artikel neben der Einordnung der Ansätze zum Einsatz im Bereich der entscheidungsunterstützenden Systeme auch ein Beispiel für das Zusammenspiel unterschiedlicher Forschungsfelder.

**Schlüsselwörter:** Künstliche Intelligenz, maschinelles Lernen, Entscheidungsunterstützung, Nachvollziehbarkeit

## 1 Einleitung

### 1.1 Vorbemerkungen

Dieser Artikel bezieht sich auf Ansätze, die im Rahmen von [1] und damit in Zusammenhang stehenden Vorarbeiten entstanden sind. Obgleich diese in einem sehr unterschiedlichen Forschungsfeld entstanden sind, konnten einige der zentralen daraus resultierenden Ergebnisse in die Medizinische Informatik zur Nutzung im Bereich der entscheidungsunterstützenden Systeme transferiert werden.

Zu den zwei Hauptbeiträgen der Vorarbeiten gehören in diesem Zusammenhang:

- Die Entwicklung von Verfahren zur Extraktion von menschenlesbarem Wissen aus lernenden Systemen, die eigenständig im Kontext von Spielen agieren.
- Die Implementierung der wichtigsten dieser Verfahren in der INTEKRATOR Toolbox [2], mit deren Hilfe die Verfahren auch in anderen Bereichen (wie z.B. in der Medizin) genutzt werden können: Mithilfe der INTEKRATOR Toolbox ist es möglich, explizites und für Menschen zugängliches Wissen aus medizinischen und anderen Daten zu lernen.

### Daan Apeldoorn<sup>1</sup>

<sup>1</sup> Universitätsmedizin der Johannes Gutenberg-Universität Mainz, Institut für Medizinische Biometrie, Epidemiologie und Informatik (IMBEI), Medizinische Informatik, Mainz, Deutschland

Im vorliegenden Artikel wird auf den Transfer der Verfahren eingegangen, die im Zuge der vorangegangenen Arbeiten im Kontext von Spielen entwickelt wurden. Es wird im Folgenden beschrieben, wie die entwickelten Verfahren in der Medizinischen Informatik im Bereich entscheidungsunterstützender Systeme Anwendung finden können und damit gleichzeitig auch ein Beispiel für das Zusammenspiel unterschiedlicher Forschungsfelder gegeben. Zu den Hauptbeiträgen des vorliegenden Artikels gehören:

- Die Erläuterung der Ursprünge von einigen der INTEKRATOR Toolbox [2] zu Grunde liegenden Verfahren und deren Zusammenhang mit dem Gebiet der Künstlichen Intelligenz in Spielen
- Die Einordnung, welche Ansätze und Verfahren aus den Vorarbeiten Potential für die Medizinische Informatik im Zusammenhang mit entscheidungsunterstützenden Systemen aufweisen

Darüber hinaus werden einige wesentliche Aspekte der Benutzungsschnittstelle der INTEKRATOR Toolbox überblicksartig zusammengefasst (eine ausführlichere Fassung findet sich in [1] und [2]).

## 1.2 Zusammenhänge zwischen den unterschiedlichen Forschungsfeldern

Welche Zusammenhänge lassen sich zwischen zwei auf den ersten Blick so unterschiedlichen Forschungsfeldern wie der *Künstlichen Intelligenz in Spielen* und der *Medizinischen Informatik* herstellen?

In [1] wurden Verfahren zur Extraktion von explizitem Wissen aus lernenden Agenten (d.h. eigenständig agierenden Systemen) im Kontext von Spielen entwickelt (z.B. [3], [4]). Eines der Ziele besteht dabei darin, für Menschen nachvollziehbar zu machen, was ein solches lernendes System im Rahmen eines Spiels gelernt hat (ähnlich wie vereinfacht in Abbildung 1 dargestellt). Aus den ursprünglich für Agenten in Spielen entwickelten Ansätzen ist die INTEKRATOR Toolbox [2] entstanden, ein System, mit dessen Hilfe Wissen auch aus medizinischen Daten expliziert und Menschen zugänglich gemacht werden kann. Das resultierende System verbindet Techniken des maschinellen Lernens und der Wissensrepräsentation (einem Teilgebiet der Künstlichen Intelligenz, KI) und dient nicht nur als Erklärungswerkzeug, sondern kann auch genutzt werden, um automatisiert Expertensysteme aus Daten zu erzeugen (siehe [5]). An ein so erzeugtes System lassen sich dann Anfragen stellen, um Schlussfolgerungen zu erhalten (beispielsweise zur Empfehlung von Therapien bei gegebenen Patienteneigenschaften). Dabei können zusätzlich Erklärungen geliefert werden, wie diese Schlussfolgerungen zustande kommen.

## 2 Methoden

Selbständig lernende Systeme haben häufig einen Blackbox-artigen Charakter: Hat ein System ein bestimm-

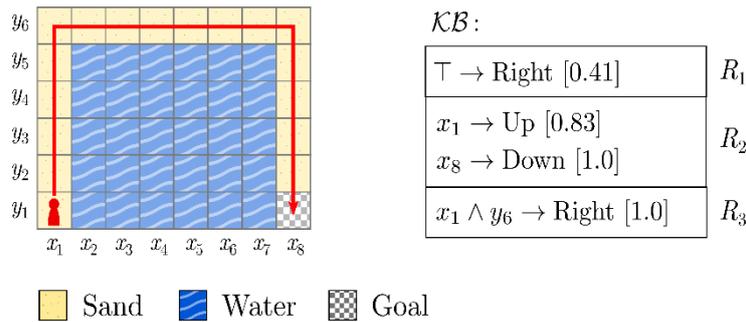
tes Verhalten gelernt, ist es oft schwierig, das gelernte Wissen in kompakter, menschenlesbarer Form zu explizieren. In kritischen Anwendungen (wie etwa in der Medizin) kann diese Problematik dazu führen, dass die mangelnde Vertrauenswürdigkeit solcher Systeme ihrem Einsatz im Wege steht. Methoden der Wissensrepräsentation [6] (ein traditionelles Teilgebiet der KI) können hier Abhilfe schaffen, da bei wissensbasierten Systemen regelhaftes Wissen zumeist in expliziter Form vorliegt und verarbeitet wird: So können beispielsweise einzelne oder mehrere Regeln identifiziert werden, auf Basis derer eine Entscheidung getroffen wird (etwa im Gegensatz zu neuronalen Netzen, bei denen das Wissen implizit in zahlreichen numerischen Gewichten enthalten ist, die nur schwer Rückschlüsse auf deren inhaltliche Bedeutung zulassen, und somit substantielle Erklärungen für eine Entscheidung häufig schwer oder gar nicht geliefert werden können). Die Funktionsweise unterscheidet sich somit auch grundsätzlich von Large Language Models (LLMs) und damit verbundenen Chat-Anwendungen: Insbesondere existiert hier auch kein Dialogsystem und Erklärungen zu Inferenzen werden auf Basis von nachvollziehbaren Regeln mit annotierten (bedingten) Wahrscheinlichkeiten geliefert.

Während etwa in [7] überblicksartig verschiedene Transparenzbezogene Ansätze im Kontext von maschinellen Lernverfahren vorgestellt werden (auch im Hinblick auf medizinische Nutzung, vgl. [7], S. 140), sind in [1] Verfahren zur automatisierten Gewinnung von verständlichem explizitem Wissen aus lernenden Agenten in Spielen entstanden: Ein solches selbständig handelndes System kann beispielsweise mit Methoden des maschinellen Lernens ein bestimmtes Verhalten erlernen, um ein Spiel zu gewinnen. Danach lässt sich die Frage stellen, wie sich das gelernte Wissen am besten in Form von Regeln explizieren lässt, sodass ein Mensch nachvollziehen kann, was gelernt wurde. Die INTEKRATOR Toolbox [2] ermöglicht es, einige der in diesem Rahmen entstandenen Verfahren auf einfache Art und Weise auch auf medizinische Datensätze anzuwenden [5].

Im Folgenden werden die zentralen Konzepte und Verfahren kurz beleuchtet (Abschnitt 2.1), bevor näher auf die INTEKRATOR Toolbox eingegangen wird (Abschnitt 2.2).

### 2.1 Lernen und Repräsentieren von Wissen in Form von Regeln mit Ausnahmen

Auch wenn die Darstellung von explizitem Wissen durch Regeln grundsätzlich ein gewisses Maß an Transparenz gewährleistet (da im Prinzip jede einzelne Regel nachvollzogen werden kann), stößt auch dieser Ansatz an seine Grenzen, wenn die Regelmenge (wie oft in realen Anwendungen) eine gewisse Größe übersteigt. Eine Möglichkeit, das Wissen kompakter (und damit grundsätzlich für den Menschen besser erfassbar) darzustellen, ist die Darstellung in Form von (Default-)Regeln mit Ausnahmen (vgl. etwa [8]): „Normalerweise gilt ..., es sei denn ...“. Die



**Abbildung 1: Gelerntes Wissen eines Agenten**

Ein Agent hat mit Methoden des maschinellen Lernens das Verhalten gelernt, die rote Spielfigur um das Wasser zu navigieren (linker Teil). Dieses Verhalten lässt sich in kompakter Form durch mehrstufige Regeln mit Ausnahmen in einer Wissensbasis ( $KB$ ) darstellen (rechter Teil). Das Wissen lässt sich dabei von oben nach unten wie folgt lesen: „Normalerweise (siehe Ebene  $R_1$ ) sollte der Agent die Spielfigur nach *rechts* bewegen, es sei denn (siehe Ebene  $R_2$ ), sie befindet sich an  $x_1$ , dann sollte der Agent sie nach *oben* bewegen oder, wenn sie sich bei  $x_8$  befindet, dann sollte er sie nach *unten* bewegen. Befindet sich die Spielfigur (siehe Ebene  $R_3$ ) an Position  $x_1$  und  $y_6$ , sollte er sie nach *rechts* bewegen“. Die an den Regeln annotierten Gewichte stellen dabei die bedingten Wahrscheinlichkeiten der Konklusionen, gegeben die Prämissen der Regeln, dar.

Quellen: Bearbeitung von Figure 3.1 in [1], welche auf Figure 1 in [12] beruht, Nutzung mit Erlaubnis von IEEE. Linker Teil nach [15], © Richard S. Sutton and Andrew G. Barto, CC BY-NC-ND 4.0 (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Darstellung von Wissen in Form von Regeln mit Ausnahmen findet sich bereits in frühen Arbeiten, wie z.B. im Rahmen der von Reiter entworfenen Default-Logik [8] oder auch in moderneren Ansätzen wie der Antwortmenprogrammierung [9].

Der im Rahmen von [1] entstandene Ansatz der Wissensrepräsentation führt das Konzept der Regeln mit Ausnahmen über mehrere Abstraktionsebenen fort. Das resultierende Wissen ist dann in Form von Regeln mit Ausnahmen (und Ausnahmen von Ausnahmen, etc.) organisiert. Abbildung 1 zeigt ein einfaches Beispiel eines lernenden Agenten zusammen mit dem extrahierten Wissen, welches das gelernte Verhalten des Agenten repräsentiert.

Zur automatisierten Gewinnung der entwickelten Wissensdarstellungen aus Sensordaten (wie im rechten Teil von Abbildung 1) wurden in [1] zwei unterschiedliche Algorithmen vorgestellt. Zudem wurde ein Verfahren entwickelt, mit dessen Hilfe solche Darstellungen auch aus numerischen Daten gewonnen werden können. Einige dieser Verfahren wurden in der INTEKRATOR Toolbox implementiert und sind somit auch auf medizinische sowie andere Daten anwendbar.

Das in der INTEKRATOR Toolbox verfügbare Verfahren zum Lernen einer Wissensdarstellung wie im rechten Teil von Abbildung 1 geht von einem Datensatz mit  $n$  Income- und einer Outcome-Spalte aus. Der Lernalgorithmus versucht dann mit möglichst wenigen Regeln und möglichst wenigen Ausnahmen die unterschiedlichen Ausprägungen in der Outcome-Spalte mit den in der Income-Spalte auftretenden Ausprägungen zu erklären. Bei deterministischen Daten (d.h. Daten, bei denen ein und dieselbe Konstellation von Income-Ausprägungen auch immer nur zu ein und derselben Outcome-Ausprägung führt) ist der Algorithmus vollständig – ein entsprechender Vollständigkeitsbeweis befindet sich in [1].

Für das Lernen von Wissen aus numerischen Daten, wird zunächst spaltenweise ein Clustering-Verfahren als unüberwachtes maschinelles Lernverfahren auf die nume-

rischen Daten angewendet. Die resultierenden Cluster werden dann als Ausprägungen der einzelnen Spalten aufgefasst. Diese Ausprägungen können dann vom zuvor beschriebenen Lernalgorithmus in Prämissen und Konklusionen gelernter Regeln verwendet werden. Bei dem Clustering-Verfahren handelt es sich um ein  $k$ -means-Clustering (z.B. [10]). Dieses kann mit einer vom Anwender vorgegebenen maximalen Anzahl  $k$  an Clustern gestartet werden (sofern eine Unterteilung a priori gewünscht bzw. bekannt ist); alternativ kann die Anzahl der Cluster auch automatisiert aus den Daten ermittelt werden. Der entsprechende Algorithmus beginnt in jedem Fall mit  $k=2$  Clustern und erhöht  $k$  dann sukzessive solange, bis leere Cluster entstehen – das letzte Clustering, welches noch keine leeren Cluster enthält, wird dann für die Erzeugung der Ausprägungen genutzt. Die initialen Zentroiden der Cluster werden dabei äquidistant über den Wertebereich einer entsprechenden Datenspalte verteilt. Die INTEKRATOR Toolbox bietet zu den aus Clustern gewonnenen Ausprägungen eine Option, mittels derer sich Informationen zu den Clustern (finale Zentroiden, Anzahl der ermittelten Cluster, prozentuale Abdeckung der Daten einzelner Cluster, etc.) an den Ausprägungen annotieren lassen. (Eine detaillierte Beschreibung der Lernverfahren findet sich in [1].)

## 2.2 Verallgemeinerung der Verfahren und Transfer in die Medizinische Informatik für den Einsatz in entscheidungsunterstützenden Systemen

Die INTEKRATOR Toolbox [2] soll es ermöglichen, die in Abschnitt 2.1 beschriebenen Verfahren unkompliziert und flexibel einsetzen zu können. Dazu wurden die zentralen entstandenen Verfahren in einer leichtgewichtigen Java-basierten Open-Source-Software implementiert, die

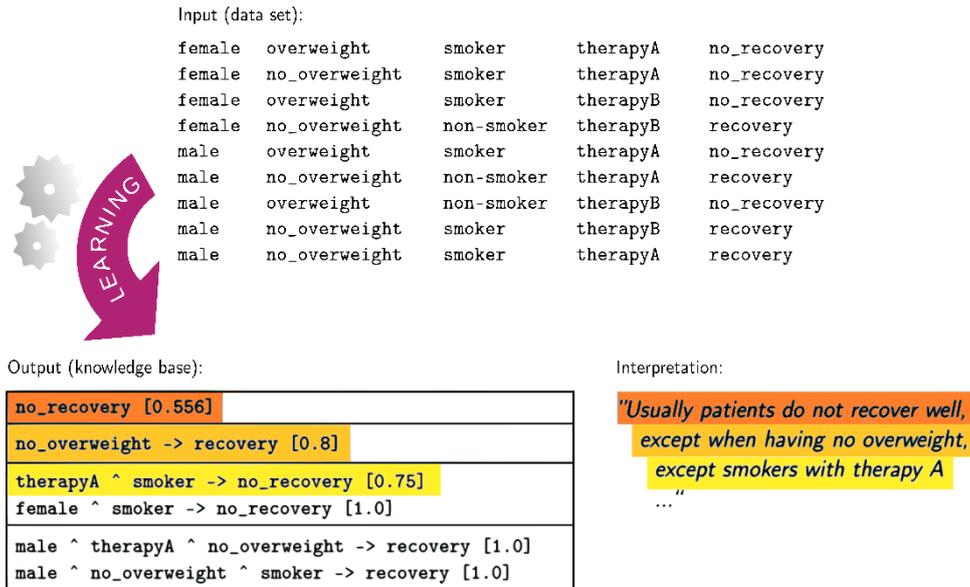


Abbildung 2: Wissen Lernen mit INTEKRATOR

Anstelle der Sensordaten und des Verhaltens eines Agenten tritt hier ein exemplarischer medizinischer Datensatz (oben). Durch das Anwenden des zuvor entwickelten Lernalgorithmus wird eine Wissensbasis gewonnen, die den Datensatz in Form von Regeln mit Ausnahmen repräsentiert (links unten). Diese lässt sich von oben nach unten lesen (rechts unten). Ein Beweis für die Vollständigkeit des Lernalgorithmus findet sich in [1].

Quelle: Bearbeitung von Figure 2 in [5], CC BY-NC 4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>)

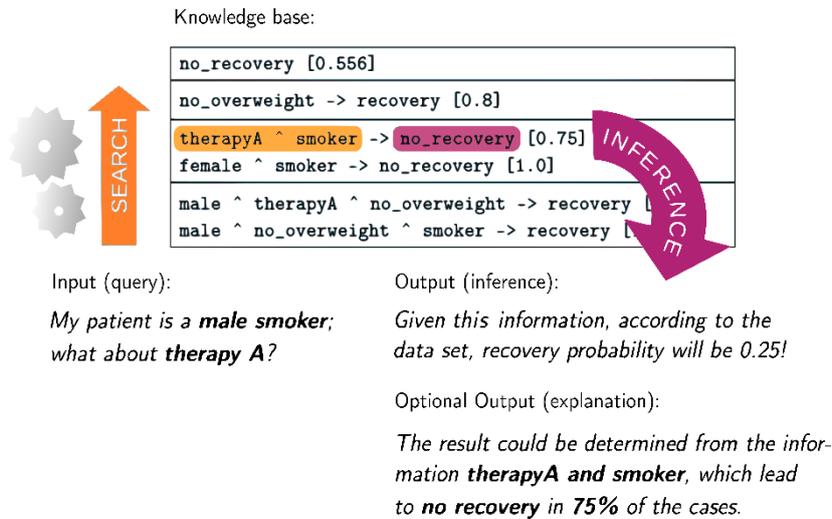


Abbildung 3: Inferenz mit INTEKRATOR

Aus einer gelernten Wissensbasis (siehe Abbildung 2) lassen sich bei gegebenen Informationen Schlussfolgerungen ableiten. Diese enthalten insbesondere auch die Regel(n), von denen die Schlussfolgerungen abgeleitet wurden, als Erklärung.

Quelle: Bearbeitung von Figure 3 in [5], CC BY-NC 4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>)

sowohl als Konsolenanwendung als auch als Programmierbibliothek eingesetzt werden kann. Damit eignet sich die Software beispielsweise auch dazu, die Verfahren in Webanwendungen einzusetzen. In [5] wurde die INTEKRATOR Toolbox zum automatisierten Lernen von Wissen für medizinische Experten-Systemen vorgeschlagen: Anstelle der manuellen Modellierung von Expertenwissen ist es damit möglich, explizites menschenlesbares Wissen direkt aus medizinischen Daten zu lernen. Eine so gelernte Wissensbasis kann dann nicht nur inspiziert werden, um Zusammenhänge in den Daten zu verstehen, sondern es lassen sich mittels INTEKRATOR auch Anfra-

gen an das gelernte Wissen stellen. Damit lassen sich automatische Schlussfolgerungen einschließlich zugehöriger Erklärungen erhalten. Abbildung 2 und Abbildung 3 verdeutlichen dies.

## 2.3 Überblick über die Benutzungsschnittstelle der INTEKRATOR Toolbox

Die Funktionalität der INTEKRATOR Toolbox [2] ist sowohl als JAVA-Bibliothek als auch als Konsolenanwendung

zugänglich. Über letzteren Weg kann sie auch von externen Programmen aus aufgerufen werden (wobei die Ausgabe vom aufrufenden Programm weiterverarbeitet werden kann). Damit ist es möglich, Wissensbasen aus Daten zu lernen, Inferenzanfragen an die so gelernten Wissensbasen zu stellen oder diese auch (bedingt) nachträglich zu revidieren. Darüber hinaus können gelernte (sowie auch manuell modellierte) Wissensbasen gegen einen Datensatz (oder eine Teilmenge davon) evaluiert werden, um die Qualität einer Wissensbasis beurteilen zu können. Im Folgenden wird auf die Benutzungsschnittstelle für die grundsätzlichen Funktionalitäten eingegangen. Eine detailliertere Beschreibung (insbesondere der Syntax für die Nutzung als Konsolenanwendung) findet sich in [1] (Anhang A) sowie in [2]; weitere Details können auch der Dokumentation des Quelltextes entnommen werden (JAVADOC).

### 2.3.1 Lernen und Revision von Wissensbasen

Um eine Wissensbasis aus einem Datensatz zu lernen, muss dieser dem Lernmodul als Eingabe in Form von  $n$  Income- und einer Outcome-Spalte geliefert werden. Es wird dann eine Wissensbasis mit höchstens  $n+1$  Ebenen zurückgegeben. Für das Lernmodul der INTEKRATOR Toolbox existieren dabei im Wesentlichen folgende Optionen:

- *Kompaktheit der gelernten Wissensbasis:* Es kann festgelegt werden, ob nur für den Eingabedatensatz relevante Regeln repräsentiert werden sollen, oder ob auf jeden Fall eine Regel auf der allgemeinsten Ebene bzw. *alle* aus den Daten gelernten Regeln geliefert werden sollen (auch wenn diese zur Beantwortung von Inferenzanfragen aus dem Datensatz, aus dem sie gelernt wurden, nicht benötigt werden).
- *Diskretisierung:* Das Abbilden von numerischen Daten auf für die Regelbildung verwendbare Symbole kann vollständig automatisiert durch Clustering oder aber auch durch die manuelle Vorgabe der Anzahl und Namen von Clustern innerhalb einer bestimmten Spalte eines Datensatzes erfolgen.
- *Vorselektion:* Bei höher-dimensionalen Datensätzen kann eine Vorselektion für die Regelbildung potentiell relevanter Income-Spalten erfolgen, um die Berechnungszeit zu reduzieren. Die Vorselektion kann automatisiert oder halb-automatisiert (durch Vorgabe einer gewünschten Anzahl Spalten) erfolgen.
- *Sampling:* Es kann festgelegt werden, ob die Wissensbasis nur aus einer Stichprobe des Eingabedatensatzes gelernt werden soll, was ebenfalls zur Verringerung der Berechnungszeit beitragen kann.

Gelernte Wissensbasen lassen sich nachträglich mittels des Revisionsmoduls mit neuen Regeln revidieren. Dabei ist zu beachten, dass potentiell ungewollte Nebeneffekte auf bereits enthaltenes Wissen nur dann umgangen werden können, wenn die Revision mit einer Regel mit Prämissenlänge  $n$  auf der untersten Ebene erfolgt (bei  $n$  Income-Spalten also auf Ebene  $n+1$ ). (Dies ist der Fall,

da sich die neue Regel dann nur auf die spezielle Konstellation der  $n$  Merkmalsausprägungen bezieht.)

### 2.3.2 Inferenzanfragen und Evaluation von Wissensbasen

An eine gelernte (oder auch manuell modellierte) Wissensbasis lassen sich Inferenzanfragen stellen, indem eine Eingabe bestehend aus einer oder mehreren unterschiedlichen Merkmalsausprägungen an das Inferenzmodul geliefert wird. Das Inferenzmodul gibt daraufhin die entsprechende Ergebnisausprägung mit einer annotierten bedingten Wahrscheinlichkeit zurück. Optional ist es möglich, sich zusätzlich die Regel(n) ausgeben zu lassen, auf Basis derer die Entscheidung für das zurückgegebene Inferenzergebnis getroffen wurde.

Darüber hinaus lässt sich eine Wissensbasis gegen einen Datensatz evaluieren, um die Qualität der Wissensbasis beurteilen zu können. Das Check-Modul der INTEKRATOR Toolbox prüft dann für eine gegebene Wissensbasis und einen Datensatz, für welchen Anteil der im Datensatz enthaltenen Datenzeilen der korrekte Outcome aus den Income-Merkmalen geschlussfolgert werden kann. Auf diese Weise kann auch eine aus einem Trainingsdatensatz gelernte Wissensbasis gegen einen Testdatensatz evaluiert werden (etwa durch entsprechendes vorheriges Teilen des Datensatzes).

## 3 Ergebnisse

In [1] sind Verfahren zur Gewinnung von explizitem verständlichen Wissen aus lernenden Systemen sowie deren Nutzung innerhalb der lernenden Systeme selbst entstanden. Die relevantesten Ergebnisse werden im Folgenden kurz hinsichtlich ihrer Anwendbarkeit in medizinischen Anwendungen – insbesondere in entscheidungsunterstützenden Systemen – beleuchtet.

Zur Repräsentation des gewonnenen Wissens wurde ein Wissensrepräsentationsformalismus beschrieben, welcher insbesondere auf lernende Agenten zugeschnitten und dabei verständlich ist (hinsichtlich der Verständlichkeit, siehe auch die gemeinsame Arbeit [11]). Unterschiedliche Lernalgorithmen ermöglichen die automatisierte Gewinnung solcher Repräsentationen aus lernenden Systemen. Durch das Auffassen von Spalten eines Datensatzes als sensorische Daten eines Agenten, lässt sich der Ansatz und seine Verständlichkeit auch für medizinische Datensätze nutzbar machen. Durch den in [1] entwickelten Inferenzalgorithmus lässt sich das so gelernte explizite Wissen auch ähnlich wie ein Expertensystem betrachten und anfragen.

Auf Basis des aus Daten gewonnenen Wissens wurde im Rahmen einer gemeinsamen Arbeit (siehe [12]) zudem ein Maß für die subjektiv empfundene Schwierigkeit von Computerspielen entwickelt. Diese folgt dem Grundgedanken, dass Spiele, zu deren Bewältigung Wissen mit zahlreichen Ausnahmen und Ausnahmen von Ausnahmen, etc. nötig ist, als schwieriger empfunden werden als sol-

che, für die ein weniger ausnahmebehaftetes Wissen hinreichend ist. Obgleich dieser Ansatz grundsätzlich auch auf aus medizinischen Daten gelerntes Wissen angewendet werden kann, scheint eine sinnvolle Anwendung in diesem Bereich weniger offensichtlich.

Auch die entstandenen Agentenmodelle, die die Ausnutzung von gelerntem expliziten Wissen während eines (bestärkenden) Lernprozesses zulassen, scheinen zunächst keinen offensichtlichen Nutzen für medizininformatische Anwendungen erkennen zu lassen. Allerdings ist hier das Agentenmodell zum Lernen von sogenannten Vorwärtsmodellen hervorzuheben (siehe auch die gemeinsamen Arbeiten [3], [13]), d.h. solche Modelle, die es ermöglichen, von dem Zustand eines Agenten und einer Aktion auf einen Folgezustand zu schließen. Hier wären beispielsweise Anwendungen für medizinische Datensätze denkbar, bei denen der Zustand eines Patienten oder einer Erkrankung nach einer oder mehrerer subsequenter Behandlungen (Aktionen) von Interesse ist.

Als weiteres Ergebnis lässt sich hier abschließend die INTEKRATOR Toolbox [2] anführen. Diese ermöglicht es, die zentralen Ansätze und Verfahren in einfacher Art und Weise auf medizinische (sowie auch andere) Datensätze anzuwenden. In [14] wird sie etwa für weitere Forschung im Zusammenhang mit Empfehlungen zur Entscheidungsunterstützung bei Krebstherapien eingesetzt.

## 4 Diskussion und Fazit

Der Transfer von Verfahren aus Forschungsbereichen wie der Künstlichen Intelligenz kann es ermöglichen, Potentiale für die Medizinische Informatik (z.B. für entscheidungsunterstützende Systeme) zu erschließen. Aus den hier betrachteten Vorarbeiten ist dabei ein System entstanden, welches potentiell auch praktischen Mehrwert für den Einsatz in medizinischen Anwendungen hat. Im Gegensatz zu Ansätzen auf Basis von neuronalen Netzen und Deep Learning, ermöglicht die INTEKRATOR Toolbox das Gelernte zu explizieren und Erklärungen für Schlussfolgerungen zu liefern. Für den Einsatz von KI-Verfahren in der Medizin kann dies ein Vorteil sein, der das Vertrauen in solche KI-Verfahren und das Zusammenspiel zwischen Mensch und Maschine verbessern kann. Als Limitation des aus den betrachteten Vorarbeiten entstandenen Systems lässt sich hier insbesondere anführen, dass eine Evaluation an Realdaten noch aussteht. Als erster Ansatz dazu kann [14] (eine Anwendung auf semi-synthetische medizinische Daten) verstanden werden.

Auch darüber hinaus bietet das entstandene System noch viel Raum für weitere Forschung, Verbesserungen und Weiterentwicklungen: So werden aktuell Datenschutzaspekte näher betrachtet (welche etwa auch für einige sich zum Teil noch in der Entwicklung befindenden Richtlinien zur KI-Regulation relevant sein könnten) und zukünftig sollen weitere Verbesserungen und Erweiterungen vorgenommen werden.

## Anmerkungen

### Interessenkonflikte

Der Autor erklärt, dass er keine Interessenkonflikte in Zusammenhang mit diesem Artikel hat.

### Danksagung

Teile weiterer Forschung im Kontext dieser Arbeit werden durch die Carl-Zeiss-Stiftung finanziert: P2021-02-014 (TOPML-Projekt).

## Literatur

1. Apeldoorn D. Comprehensible Knowledge Base Extraction for Learning Agents - Practical Challenges and Applications in Games. Dissertation at the Department of Computer Science of TU Dortmund University. Mainz: Universitätsbibliothek der Johannes Gutenberg-Universität Mainz; 2023. DOI: 10.25358/openscience-9303
2. Apeldoorn D. InteKRator Toolbox. GitLab repository; [visited 2023 Mar 31]. Available from: [https://gitlab.com/dapel1/intekrator\\_toolbox/](https://gitlab.com/dapel1/intekrator_toolbox/)
3. Apeldoorn D, Dockhorn A. Exception-Tolerant Hierarchical Knowledge Bases for Forward Model Learning. *IEEE Transactions on Games*. 2021;13(3):249-62. DOI: 10.1109/TG.2020.3008002
4. Apeldoorn D, Kern-Isberner G. Towards an Understanding of What is Learned: Extracting Multi-Abstraction-Level Knowledge from Learning Agents. In: Rus V, Markov Z, editors. *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2017)*; 2017 May 22-24; Marco Island, FL, USA. Palo Alto: AAAI Press; 2017. p. 764-7.
5. Apeldoorn D, Panholzer T. Automated Creation of Expert Systems with the InteKRator Toolbox. *Studies in Health Technology and Informatics*. Stud Health Technol Inform. 2021;283:46-55. DOI: 10.3233/SHTI210540
6. Beierle C, Kern-Isberner G. *Methoden wissensbasierter Systeme – Grundlagen, Algorithmen, Anwendungen*. 4th ed. Wiesbaden: Vieweg+Teubner; 2008.
7. Holzinger A. Explainable AI (ex-AI). *Informatikspektrum*. 2018;41:138-43. DOI: 10.1007/s00287-018-1102-5
8. Reiter R. A Logic for Default Reasoning. *Artificial Intelligence*. 1980;13(1-2):81-132. DOI: 10.1016/0004-3702(80)90014-4
9. Brewka G, Eiter T, Truszczynski M. Answer Set Programming at a Glance. *Commun ACM*. 2011;54(12):92-103. DOI: 10.1145/2043174.2043195
10. Steinley D. K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*. 2006;59(1):1-34. DOI: 10.1348/000711005X48266
11. Krüger C, Apeldoorn D, Kern-Isberner G. Comparing Answer Set Programming and Hierarchical Knowledge Bases Regarding Comprehensibility and Reasoning Efficiency in the Context of Agents. In: *Proceedings of the 30th International Workshop on Qualitative Reasoning (QR 2017) at International Joint Conference on Artificial Intelligence (IJCAI 2017)*; 2017 Aug 21; Melbourne, Australia. Evanston, IL: Northwestern University; 2017.

12. Apeldoorn D, Volz V. Measuring Strategic Depth in Games Using Hierarchical Knowledge Bases. In: 2017 IEEE Conference on Computational Intelligence and Games (CIG'17); 2017 Aug 22-25; New York, USA. Piscataway: IEEE; 2017. p. 9-16. DOI: 10.1109/CIG.2017.8080409
13. Dockhorn A, Apeldoorn D. Forward Model Approximation for General Video Game Learning. In: Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG'18); 2018 Aug 14-17; Maastricht, Netherlands. Piscataway: IEEE; 2018. p. 425-32. DOI: 10.1109/CIG.2018.8490411
14. Thevapalan A, Apeldoorn D, Kern-Isberner G, Meyer RG, Nietzke M, Panholzer T. Comparison and Incorporation of Reasoning and Learning Approaches for Cancer Therapy Research. Stud Health Technol Inform. 2023;307:161-71. DOI: 10.3233/SHTI230709
15. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, Massachusetts: The MIT Press; 2018.

**Korrespondenzadresse:**

Dr. Daan Apeldoorn  
Universitätsmedizin der Johannes Gutenberg-Universität  
Mainz, Institut für Medizinische Biometrie, Epidemiologie  
und Informatik (IMBEI), Medizinische Informatik, 55101  
Mainz, Deutschland, Tel.: +49 6131 17-5062  
daan.apeldoorn@uni-mainz.de

**Bitte zitieren als**

Apeldoorn D. Transfer von Ergebnissen aus dem Gebiet der Künstlichen Intelligenz in Spielen in die Medizinische Informatik zur Anwendung in entscheidungsunterstützenden Systemen. GMS Med Inform Biom Epidemiol. 2024;20:Doc10. DOI: 10.3205/mibe000266, URN: urn:nbn:de:0183-mibe0002661

**Artikel online frei zugänglich unter**

<https://doi.org/10.3205/mibe000266>

**Veröffentlicht:** 09.04.2024

**Copyright**

©2024 Apeldoorn. Dieser Artikel ist ein Open-Access-Artikel und steht unter den Lizenzbedingungen der Creative Commons Attribution 4.0 License (Namensnennung). Lizenz-Angaben siehe <http://creativecommons.org/licenses/by/4.0/>.