



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

Ausarbeitung Anwendungen 2  
Sommersemester 2011  
Andreas Basener

Erlernen und Erkennen von  
Verhaltensmustern in Sensordaten

## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>3</b>
1.1 Rückblick . . . . .	3
<b>2 Related Works</b>	<b>3</b>
2.1 BehaviourScope . . . . .	4
2.1.1 Struktur der Events . . . . .	5
2.1.2 zeitliche Abstraktion . . . . .	5
2.1.3 Apriori-Algorithmus . . . . .	6
2.2 Aware Home . . . . .	7
2.2.1 n-Grams . . . . .	7
2.2.2 Aktivitätsgraph . . . . .	8
2.2.3 Aktivitätsklassen . . . . .	9
2.3 CASAS . . . . .	9
2.3.1 Activity Discovery Method . . . . .	10
2.3.2 Clustering . . . . .	10
2.3.3 Tracking . . . . .	11
<b>3 Einordnung und Abgrenzung</b>	<b>12</b>
3.1 Sensordaten . . . . .	12
3.2 Fehlerrate . . . . .	12
<b>Literatur</b>	<b>13</b>

# 1 Einführung

In dieser Ausarbeitung für die Veranstaltung „Anwendungen 2“ werde ich, nach einem kurzen Rückblick auf Anwendungen 1, verschiedene Ansätze vorstellen, die sich mit dem Thema Erlernen und Erkennen von Verhaltensmustern in Sensordaten beschäftigen.

## 1.1 Rückblick

In meiner Ausarbeitung zu der Veranstaltung Anwendungen 1 ([Basener, 2011](#)) habe ich einen Einblick in das Thema „Drahtlose Sensornetzwerke im Kontext Ambient Assisted Living“ gegeben.

Damit auf die Aktivitäten von Bewohnern reagiert werden kann, werden Sensordaten benötigt. Diese Sensordaten können von einem drahtlosen Sensornetzwerk bereitgestellt werden. Die einzelnen Sensorknoten überwachen die Umwelt mit unterschiedlichen Sensoren, z.B. Temperatur- oder Bewegungssensoren, und übertragen sie drahtlos an Rechner, die die Daten auswerten.

Aus den gewonnenen Daten können nun Rückschlüsse auf das Verhalten der Bewohner gezogen werden. Wenn das Verhalten des Bewohners erkannt wird, kann sich die Umwelt auf ihn einstellen. Das kann dazu genutzt werden, den Komfort zu erhöhen, aber auch um Gefahrensituationen zu erkennen und zu entschärfen. Liegt der Bewohner z.B. bereits im Bett und schläft, ist es wünschenswert, wenn der vergessene Herd in der Küche ausgeschaltet wird.

Im weiteren Verlauf meiner Ausarbeitung werde ich mich mit der Fragestellung auseinandersetzen, wie in den Sensordaten Verhaltensmuster gefunden und erkannt werden können.

## 2 Related Works

Einzelne Sensordaten enthalten kaum sinnvolle Informationen. Zwar gibt ein einzelner Wert, z.B. eines Temperatursensors, Auskunft über seinen aktuellen Zustand. Daraus lässt sich aber nicht erkennen, ob dieser Zustand z.B. über einen längeren Zeitraum stabil ist oder ob der Sensorwert evtl. in einem Zusammenhang mit einem anderen Sensor steht.

Es müssen somit mehrere Sensorinformationen in einen Zusammenhang gebracht werden. Damit ergibt sich die nächste Fragestellung: *Welche Zusammenhänge existieren zwischen Sensorinformationen?*

Eine Möglichkeit besteht darin, sich vorher intensive Gedanken zu machen, welche Zusammenhänge bestehen können und welche Verhaltensmuster vom Bewohner erwartet werden.

Das bedeutet aber einen hohen manuellen Aufwand und setzt voraus, dass sich der Bewohner erwartungsgemäß verhält.

Eine elegantere Lösung ist es, die gewonnenen Sensordaten zu nutzen, um darin nach Verhaltensmustern zu suchen.

Nachfolgend werden drei Projekte vorgestellt, die sich bereits mit dem Erlernen und Erkennen von Verhaltensmustern in Sensordaten beschäftigt haben.

## 2.1 BehaviourScope

Das Projekt BehaviourScope<sup>1</sup> der Universität Yale in den USA wurde im Zeitraum 2006 bis 2009 im Enelab durchgeführt. An diesem Projekt waren u.a. Athanasios Bamis, Jia Fang, Andreas Savvides und Dimitrios Lymberopoulos beteiligt, die sich mit Verhaltensmustern in Sensornetzwerken beschäftigt haben.

In diesem Projekt wurde u.a. untersucht, wie aus zeitkontinuierlichen Events Informationen über Aktivitäten der Bewohner gewonnen werden können.

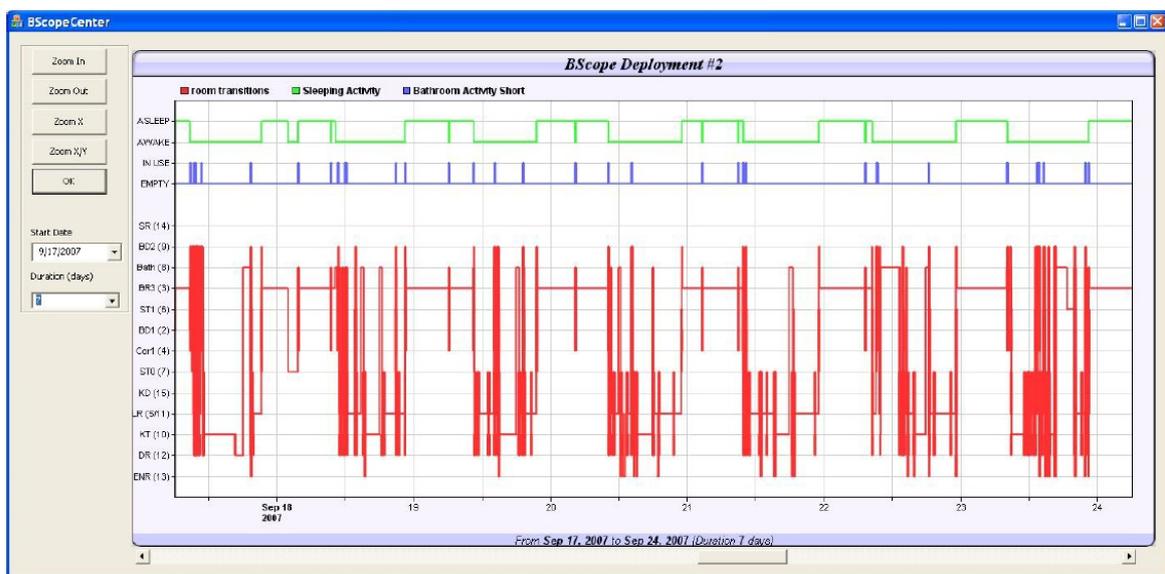


Abbildung 1: mehrere Eventstreams (Lymberopoulos u. a., 2008)

<sup>1</sup><http://bscope.eng.yale.edu/>

### 2.1.1 Struktur der Events

Ein Event ist als ein Tupel definiert (Lymberopoulos u. a., 2008, S. 3).

$$\text{Eventtupel} = \{P, T, D\}$$

Wobei  $P$  die ID des Sensors ist,  $T$  der Zeitstempel und  $D$  die Dauer des Events in Minuten. Ein Ausschnitt aus einem Eventstream kann wie folgt angegeben werden:

$$E(T_{\text{start}}, T_{\text{stop}}) = \{(P_j^i, T_j^i, D_j^i) | (P_j^i, T_j^i, D_j^i) \in S^i, T_{\text{start}} \leq T_j^i \leq T_{\text{stop}} \forall i, j\}$$

$T_{\text{start}}$  und  $T_{\text{stop}}$  geben die obere bzw. untere Schranke für das Zeitintervall an. Alle Eventtupel, die innerhalb des Intervalls liegen, werden so zu einem Block zusammengefasst.

Ein Beispiel, wie ein Intervall aussehen kann ist folgendes:

$$E(10\text{pm}, 10\text{am}) = \langle \{\text{Bed}, 11:00\text{pm}, 300\}, \{\text{Bath}, 4:00\text{am}, 5\}, \{\text{Bed}, 4:05\text{am}, 300\}, \\ \{\text{Bath}, 9:05\text{pm}, 10\}, \{\text{Kitchen}, 09:15\text{pm}, 30\} \rangle$$

### 2.1.2 zeitliche Abstraktion

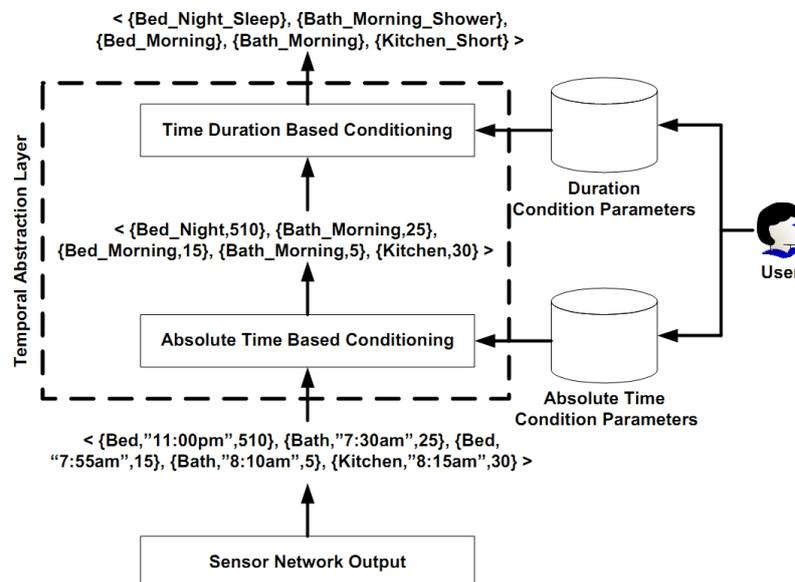


Abbildung 2: zeitliche Abstraktion (Lymberopoulos u. a., 2008)

Da es unwahrscheinlich ist, dass der Bewohner die gleichen Aktionen zu exakt der gleichen Zeit bzw. im exakt gleichen Zeitraum ausführt, wird noch ein Mechanismus benötigt, um

Abweichungen auszugleichen. Hierzu werden die Eventtupel auf fest definierte Aktionen abstrahiert, die eine einfache zeitliche Einordnung ermöglichen (s. Abbildung 2).

Zuerst wird der Zeitstempel eines Sensortupels  $T_0$  auf eine absolute Zeitangabe abgebildet. Z.B. wird die Zeitangabe *11:00pm* aus dem Tupel  $T_0 = \{\text{Bed}, 11:00\text{pm}, 510\}$  mit der ID *Bed* zu *Bed\_Night* verknüpft. Daraus ergibt sich ein neues Tupel  $T_1 = \{\text{Bed}_\text{Night}, 510\}$ .

Anschließend wird mittels der Dauer auf die eigentliche Aktion geschlossen. Aus dem Tupel  $T_1$  entsteht das endgültige Tupel  $T_2 = \{\text{Bed}_\text{Night}_\text{Sleep}\}$

Wie die zeitlichen Parameter zu interpretieren sind, wird vom Benutzer manuell vorgegeben.

### 2.1.3 Apriori-Algorithmus

Um wiederkehrende Verhaltensmuster in den Eventstreams zu erkennen, wird im BehaviorScope-Projekt der Apriori-Algorithmus verwendet. In Listing 1 ist der Apriori-Algorithmus aufgeführt.

**Apriori** ( $D$ )

**Eingabe:** Datenbasis  $D$

**Ausgabe:** Menge häufiger Itemmengen

```

 $L_1 := \{\text{häufige 1-Itemmenge}\}$ 
 $k := 2$ 
while  $L_{k-1} \neq \emptyset$  do
   $C_k := \text{AprioriGen}(L_{k-1})$  //neue Kandidatenmenge
  for all Transaktionen  $t \in D$  do
     $C_t := \{c \in C_k \mid c \subseteq t\}$  //in  $t$  enthaltene Kandidatenmenge
    for all Kandidaten  $c \in C_t$  do
       $c.\text{count} := c.\text{count} + 1$ 
    end for
  end for
   $L_k := \{c \in C_k \mid c.\text{count} \geq |D| * \text{minsupp}\}$ 
   $k := k + 1$ 
end while
return  $\bigcup_k L_k$ 

```

Listing 1: Apriori-Algorithmus (Beierle und Kern-Isberner (2006), S. 148)

Durch den Apriori-Algorithmus werden die am häufigsten vorkommenden Eventsequenzen gefunden. Dabei wird mit einem einzelnen Event begonnen und dieser iterativ um weitere Events erweitert. Ob eine erweiterte Sequenz übernommen wird, hängt davon ab, ob die in

dieser Sequenz enthaltenen Subsequenzen eine Mindesthäufigkeit aufweisen.

Eine detaillierte Beschreibung des Apriori-Algorithmus ist in (Beierle und Kern-Isberner, 2006, S. 147 f.) zu finden.

Die so gefundenen Sequenzen bilden die Basis für die Aktivitätenerkennung in neuen Sensordaten.

## 2.2 Aware Home

Das Aware-Home-Projekt<sup>2</sup> des Georgia Institute of Technology in den USA beschäftigt sich seit 1999 mit der Verknüpfung von moderner Informationstechnologie mit dem alltäglichen Wohnen.

In diesem Projekt sind u.a. Brian Jones, Nachiket Ramanujam, Lana Yarosh und Tae-jung Yun aktiv, die sich mit dem Erlernen von Verhaltensmustern beschäftigt haben.

Die Verhaltensmustererkennung basiert in diesem Projekt auf Verhaltensklassen und deren Verknüpfung in Graphen.

In diesem Verfahren wird davon ausgegangen, dass einzelne Aktivitäten klar als solche angegeben sind. Eine Aktivität besteht dabei aus mehreren diskreten Events. Diese Aktivitäten werden nun auf deren Eigenschaften hin untersucht. Ähnliche Aktivitäten werden zu Aktivitätsklassen zusammengefasst. Mit diesen Klassen können dann neue Aktivitäten eingeordnet werden.

Der genaue Ablauf des in Abbildung 3 dargestellte Frameworks wird in den nachfolgenden Abschnitten beschrieben:

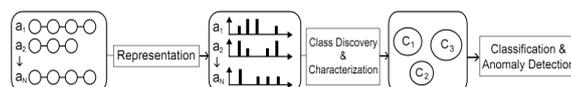


Abbildung 3: Framework (Hamid, 2008)

### 2.2.1 n-Grams

Die Aktivitäten stellen Sequenzen von diskreten Events dar. Über diese Sequenzen können Fenster mit der Größe  $n$  geschoben werden, die  $n$ -Grams genannt werden. Es werden sämtliche  $n$ -Grams einer Aktivität erstellt. Aus den gefundenen  $n$ -Grams wird dann ein Histogramm erstellt, das angibt, welches  $n$ -Gram wie häufig vorkommt. In Abbildung 4 ist zu

<sup>2</sup><http://awarehome.imtc.gatech.edu/>

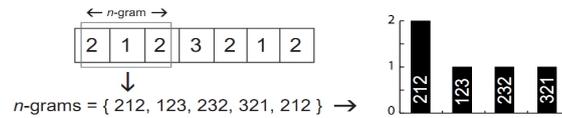


Abbildung 4: n-grams (Hamid, 2008)

sehen, wie ein n-Gram und das dazugehörige Histogramm für die Größe 3 aussehen können.

### 2.2.2 Aktivitätsgraph

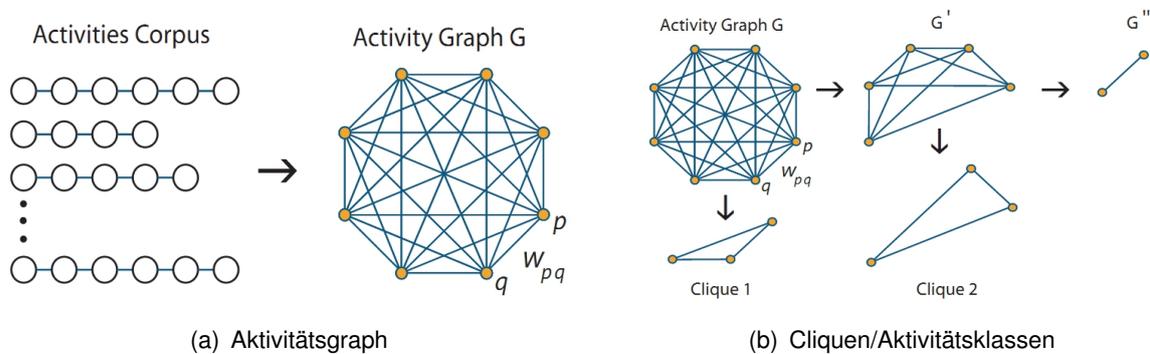


Abbildung 5: (Hamid, 2008)

Mit Hilfe der n-Grams und der daraus berechneten Histogramme wird nun ein ungerichteter und gewichteter Aktivitätsgraph erstellt. Dazu wird für alle Aktivitäten berechnet, wie sehr sie den anderen Aktivitäten ähneln.

Für die Ähnlichkeit zweier Aktivitäten wird untersucht, ob die n-Grams nur in einer Aktivität vorkommen oder in beiden. Kommt ein n-Gram nur in einer Aktivität vor, ergibt sich daraus ein Unterschied. Kommt jedoch ein n-Gram in beiden Aktivitäten vor, wird untersucht, wie häufig das n-Gram jeweils vorkommt. Die genaue Formel ist in (Hamid, 2008, S. 31) in der Gleichung (1) angegeben.

Aus den ermittelten Ähnlichkeiten der Aktivitäten kann ein Aktivitätsgraph erstellt werden. Abbildung 5(a) veranschaulicht dies. Die Knoten sind hierbei die Aktivitäten, die Kanten geben die Ähnlichkeit zwischen den einzelnen Aktivitäten wieder.

### 2.2.3 Aktivitätsklassen

Im anschließenden Schritt wird der ermittelte Aktivitätsgraph reduziert. Hierbei werden ähnliche Knoten zu Aktivitätsklassen zusammengefasst. Dazu werden in dem Aktivitätsgraphen iterativ maximale Cliques gesucht. Um den Rechenaufwand klein zu halten, werden hierfür dominante Sets verwendet. Die genaue Berechnung ist in (Hamid, 2008, S. 33ff.) zu finden.

Jetzt können neue Aktivitätsinstanzen den berechneten Aktivitätsklassen zugeordnet werden. Dazu wird die neue Aktivität mit allen in einer Aktivitätsklasse enthaltenen Aktivitäten verglichen und die Ähnlichkeiten berechnet. Die aufsummierten und gewichteten Ähnlichkeiten zu den einzelnen Aktivitäten einer Klasse geben dann die Ähnlichkeit der neuen Aktivität zu der gesamten Aktivitätsklasse an (Abb. 5(b)).

## 2.3 CASAS

Das CASAS-Smart-Home-Projekt<sup>3</sup> der Washington State University in den USA besteht seit 2005. Diane J. Cook, Lawrence Holder, Behrooz Shirazi, Maureen Schmitter-Edgecombe und Teddy Yap beschäftigen sich im Rahmen dieses Projektes mit dem Erkennen und Erlernen von Verhaltensmustern in Sensordaten.

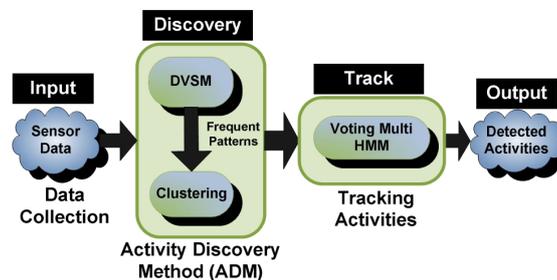


Abbildung 6: Ablauf der Aktivitätenerkennung (Rashidi u. a., 2011)

Dort werden die Sensordaten mit Hilfe der Activity Discovery Method (ADM) untersucht. In der ADM werden häufig wiederkehrende Muster in den Sensordaten erkannt und abgespeichert. Damit das anschließende Erkennen von Mustern besser funktioniert, werden die erkannten Muster zu Clustern zusammengefügt. Mit diesen Clustern wird dann die eigentliche Mustererkennung für neue Sensordaten durchgeführt.

<sup>3</sup><http://ailab.wsu.edu/casas/>

### 2.3.1 Activity Discovery Method

Die Activity Discovery Method (ADM) besteht aus zwei Teilen. Im ersten Teil werden die eingehenden Sensordaten auf Muster untersucht. Dies geschieht mit Hilfe des Discontinuous Varied-order Sequential Miners (DVSM). Im zweiten Teil werden ähnliche Verhaltensmuster zu Clustern zusammengefasst.



Abbildung 7: Eventstreams (Rashidi u. a., 2011)

Der DVSM arbeitet folgendermaßen: Zuerst werden alle Einzelevents eines Datensatzes  $D$  auf ihre Häufigkeit hin geprüft. Die häufigsten Events werden in einem neuen reduzierten Datensatz  $D_r$  zusammengefasst. Welche die häufigsten Events sind, wird von dem Benutzer durch den Wert  $\alpha$  vorgegeben. Über  $D_r$  wird nun ein Fenster der Größe 2 geschoben, um so Muster mit der Länge 2 zu erhalten. Anschließend werden die erkannten Muster um ihre Prä- und Suffixe erweitert.

Für jedes erkannte Muster werden zusätzlich häufig vorkommende Variationen gesucht. Diese Variationen werden mit Hilfe der Levenshtein-Distanz berechnet (Merkl und Waack, 2005, S. 93ff.). Wie in Abbildung 7 zu sehen ist, kann ein Muster  $\{a, b, c\}$  neben seiner einfachen Form noch in anderen Varianten auftauchen. So können einzelne Events vertauscht sein ( $\{a, c, b\}$ ), oder aber fremde Events tauchen innerhalb einer Variante auf ( $\{a, b, e, c\}$ ). Zudem kann auch beides gleichzeitig eintreten ( $\{a, c, e, b\}$ ). Details sind in (Rashidi u. a., 2011, S. 4f.) in den Formeln (1) bis (9) beschrieben.

### 2.3.2 Clustering

Im nächsten Schritt der ADM werden die entdeckten Muster zu Clustern zusammengefasst. Zu diesem Zweck werden die einzelnen Events um zusätzliche Informationen erweitert. Diese Zusatzinformationen können der Typ eines Sensors oder die Dauer eines Events sein. Diese erweiterten Events bilden Zustände, die für das Clustering verwendet werden. Dazu wird, ähnlich wie in der DVSM, die Distanz zwischen den Zuständen berechnet (Rashidi u. a., 2011, S. 5f.).

Zustände, die sich ähneln, werden dann zu Clustern zusammengefasst. Der Zustand mit den geringsten Distanzen zu den anderen Zuständen eines Clusters wird als Clusterzentrum verwendet. Wie viele Cluster erzeugt werden, wird vom Benutzer vorgegeben. Die erzeugten Cluster spiegeln die Aktivitäten wieder, die zukünftig erkannt werden können.

### 2.3.3 Tracking

Nachdem aus den Sensordaten entsprechende Verhaltensmuster gefunden wurden, wird im Tracking das erlernte Modell auf neue Sensordaten angewendet. Um herauszufinden, zu welchen Aktivitäten die neuen Sensordaten gehören, werden Hidden-Markov-Modelle (HMM) (Fink, 2003) verwendet.

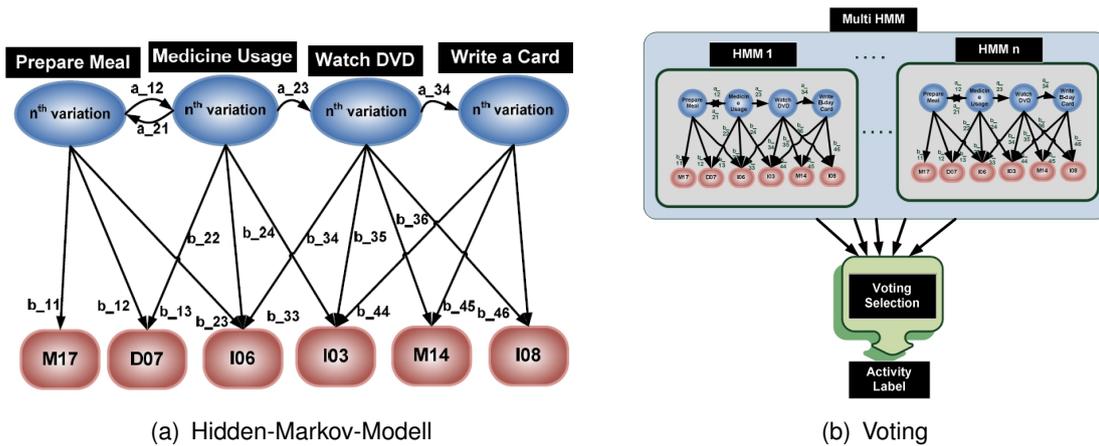


Abbildung 8: (Rashidi u. a., 2011)

Die Aktivitäten bilden hierbei die versteckten Zustände des HMM.

Damit nun die richtige Aktivität erkannt werden kann, werden für alle  $n$  Variationen  $n$  HMM aufgestellt. Neue Sensordaten werden in alle HMM eingegeben und jeweils die wahrscheinlichste Aktivität wird berechnet. Danach wird die Aktivität ausgewählt, die in allen HMM am häufigsten als aktuell wahrscheinlichste Aktivität berechnet wurde (Rashidi u. a., 2011, S. 6f.).

## 3 Einordnung und Abgrenzung

### 3.1 Sensordaten

Die in Kapitel 2 vorgestellten Arbeiten haben verschiedene Vorbedingungen und Ansätze, um Muster in Sensordaten zu finden und zu erkennen. Im BehaviourScope-Projekt muss der Benutzer manuell angeben, wie die zeitlichen Parameter der Eventtupel zu interpretieren sind, im Aware-Home-Projekt müssen die Aktivitäten explizit angegeben werden und im CASAS-Projekt muss angegeben werden, wie viele Aktivitäten erkannt werden sollen. Das widerspricht dem Gedanken, das Erlernen und Erkennen der Verhaltensmuster automatisch ablaufen zu lassen.

Das LivingPlace-Projekt der HAW hat seine eigenen Voraussetzungen, die eine direkte Umsetzung der vorgestellten Arbeiten weiter erschweren. Dazu gehört u.a., dass manche Teilprojekte momentan erst entstehend bzw. erst geplant sind. Daraus ergibt sich, dass noch keine verwertbaren Sensordaten vorhanden sind, die ein Erlernen und Erkennen von Verhaltensmustern zulassen.

Im Rahmen des CASAS-Projektes der Washington State University werden Sensoraufzeichnungen zur Verfügung gestellt<sup>4</sup>. Diese sind aber speziell auf deren Projekte abgestimmt und spiegeln kaum die Bedürfnisse für das Living-Place wider. Um eine brauchbare Basis an Sensordaten zur Verfügung zu stellen, entwickle ich im Rahmen der „Masterprojekt 1“-Veranstaltung einen Simulator für das Living-Place, mit dem Tagesabläufe simuliert werden können und die dazu passenden Sensordaten generiert werden.

### 3.2 Fehlerrate

Die Fehlerraten der vorgestellten Verfahren liegen nicht bei 0%. Da die Verfahren auf unterschiedlichen Ausgangsdaten basieren und unterschiedliche Fehlererkennungen verwenden, verzichte ich auf einen direkten Vergleich der Fehlerrate.

Es ist aber abzulesen, dass bei einer Umsetzung eines geeigneten Verfahrens für das Living-Place mit Fehlern beim Lernen und Erkennen von Verhaltensmustern zu rechnen ist. In unkritischen Situationen, wie Kaffee kochen, ist eine fehlerhafte Erkennung wohl zu verschmerzen. In medizinischen Notsituationen muss das Verfahren jedoch zuverlässig funktionieren.

---

<sup>4</sup><http://ailab.wsu.edu/casas/datasets.html>

## Literatur

- [Bamis u. a. 2010] BAMIS, Athanasios ; FANG, Jia ; SAVVIDES, Andreas: Discovering routine events in sensor streams for macroscopic sensing composition. In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. New York, NY, USA : ACM, 2010 (IPSN '10), S. 408–409. – URL <http://doi.acm.org/10.1145/1791212.1791278>. – ISBN 978-1-60558-988-6
- [Basener 2011] BASENER, Andreas: Ausarbeitung AW1 Drahtlose Sensornetzwerke im Kontext Ambient Assisted Living / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-aw1/basener/bericht.pdf>, Februar 2011. – Forschungsbericht
- [Beierle und Kern-Isberner 2006] BEIERLE, Christoph ; KERN-ISBERNER, Gabriele: *Methoden wissensbasierter Systeme*. Vieweg+Teubner, 2006. – ISBN 978-3-8348-0504-1
- [Cook und Holder 2011] COOK, Diane J. ; HOLDER, Lawrence B.: Sensor selection to support practical use of health-monitoring smart environments. In: *ata Mining and Knowledge Discovery* (2011). – URL <http://eecs.wsu.edu/~cook/pubs/dmkd11.pdf>
- [Ellenberg 2009] ELLENBERG, Jens: *Event Stream Processing mit ESPER unter Einsatz von Datamining Verfahren*. Hamburg, Hochschule für Angewandte Wissenschaften, Diplomarbeit, 2009. – URL <http://opus.haw-hamburg.de/volltexte/2009/819/>
- [Fang u. a. 2009] FANG, Jia ; BAMIS, Athanasios ; SAVVIDES, Andreas: Discovering Routine Events and their Periods in Sensor Time Series Data / Yale University. URL [http://www.eng.yale.edu/enalab/publications/RTSS\\_Periodicity.pdf](http://www.eng.yale.edu/enalab/publications/RTSS_Periodicity.pdf), 2009. – Forschungsbericht
- [Fink 2003] FINK, Gernot A.: *Mustererkennung mit Markov-Modellen - Theorie - Praxis - Anwendungsgebiete*. Vieweg+Teubner, 2003. – ISBN 9783519004530
- [Gesellschaft für Informatik E.V. 2011] GESELLSCHAFT FÜR INFORMATIK E.V.: *Complex Event Processing (CEP)*. online. Juni 2011. – URL <http://www.gi.de/service/informatiklexikon/informatiklexikon-detailansicht/meldung/complex-event-processing-cep-221.html>
- [Hamid 2008] HAMID, Muhammad R.: *A Computational Framework For Unsupervised Analysis of Everyday Human Activities*, Georgia Institute of Technologie, Dissertation, 2008. – URL <http://etd.gatech.edu/theses/available/etd-06232008-101404/>

- [Heierman und Cook 2003] HEIERMAN, Edwin O. ; COOK, Diane J.: Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns. In: *Proceedings of the Third IEEE International Conference on Data Mining*. Washington, DC, USA : IEEE Computer Society, 2003 (ICDM '03), S. 537–. – URL <http://portal.acm.org/citation.cfm?id=951949.952078>. – ISBN 0-7695-1978-4
- [von Luck u.a. 2010] LUCK, Prof. Dr. K. von ; KLEMKE, Prof. Dr. G. ; GREGOR, Sebastian ; RAHIMI, Mohammad A. ; VOGT, Matthias: Living Place Hamburg – A place for concepts of IT based modern living / Hamburg University of Applied Sciences. URL [http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg\\_en.pdf](http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf), Mai 2010. – Forschungsbericht
- [LyMBERopoulos u.a. 2008] LYMBEROPOULOS, Dimitrios ; BAMIS, Athanasios ; SAVVIDES, Andreas: Extracting spatiotemporal human activity patterns in assisted living using a home sensor network. In: *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. New York, NY, USA : ACM, 2008 (PETRA '08), S. 29:1–29:8. – URL <http://doi.acm.org/10.1145/1389586.1389621>. – ISBN 978-1-60558-067-8
- [Merkl und Waack 2005] MERKL, Rainer ; WAACK, Stephan: *Bioinformatik interaktiv: Algorithmen und Praxis*. Weinheim : Vch Verlagsgesellschaft MbH, 2005. – ISBN 978-3527306626
- [Rashidi u.a. 2011] RASHIDI, Parisa ; COOK, Diane J. ; HOLDER, Lawrence B. ; SCHMITTER-EDGEcombe, Maureen: Discovering Activities to Recognize and Track in a Smart Environment. In: *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* (2011). – URL <http://eecs.wsu.edu/~cook/pubs/tkde10.pdf>