universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

Mathematische Methoden für computergestütztes Wohnen

Verfasserin

Melanie Obernosterer

angestrebter akademischer Grad

Magistra der Naturwissenschaften (Mag.rer.nat)

Wien, April 2013

Matrikelnummer:	0108030
Studienkennzahl lt. Studienblatt:	A 405
Studienrichtung lt. Studienblatt:	Diplomstudium Mathematik
Betreuer:	O.Univ.-Prof. Dr. Arnold Neumaier

Für Arthur und Leandra

Kurzfassung

Mathematische Methoden im Bereich des computergestützten Wohnens (*Ambient Assisted Living*) können eine wertvolle Ergänzung zu bereits existierenden technischen Systemen darstellen. Ein aktuelles Projekt aus diesem Forschungsfeld ist das *eHome*-Assistenzsystem. Dieses wurde am ”Zentrum für Angewandte Assistierende Technologien” zur Unterstützung allein lebender älterer Menschen entwickelt und über mehrere Monate in verschiedenen Wohnungen getestet. In dieser Arbeit wird untersucht, inwieweit mathematische Methoden dazu verwendet werden können direkt aus den Daten, die im Rahmen der Testreihe gesammelten wurden und die in anonymisierten Form zur Analyse zur Verfügung gestellt wurden, Information über das Verhalten der Testpersonen zu generieren – und das in möglichst automatisierter und wohnungsunabhängiger Form.

Die Untersuchung beinhaltet eine statistische Auswertung der Daten und darauf aufbauend eine Vervollständigung der nicht zeitgleich aufgezeichneten Daten zu einem vollständigen numerischen Merkmalsdatensatz, sowie eine automatische Klassifikation der Zeitfenster in Aktionsklassen, welche bereits typische Verhaltensmuster aufzeigen. Des weiteren wird ein Algorithmus zur Bewegungsrekonstruktion untersucht, der auf einem lernfähigen Bayesischen Netzwerk basiert und welcher nach einer anfänglichen Konfiguration für jede Testwohnung direkt angewendet werden kann. Dazu muss lediglich die Topologie und Ausstattung der Wohnung in einem abstrahierten Wohnungsmodell erfasst werden. Mit Hilfe der rekonstruierten Bewegungssequenzen wird versucht relevante Aktivitäten im Tagesablauf der Testpersonen zu erkennen.

Abstract

In the field of home monitoring mathematical methods are completing the already existing technical systems. Actual research in this field is currently being done in the *eHome* Project. The goal was to aid elder people in living on their own and the project was realized at the ”Zentrum für Angewandte Assistierende Technologien” and was tested over several months. This paper is evaluating the possibilities of mathematical methods to generate information about the behaviour of the anonymous test person. To meet practical needs it is tried to generate that information independent of the footprint of the flat and as automated as possible. The examination entails a statistical analysis of the data as well as an automated classification oft the time frames into action classes, which are showing typical behaviour. An algorithm based on a self-learning bayesian network, which allows to reconstruct motion oft the test-subjects is analyzed. Therefore an abstract footprint of flats is developed, so that every new flat can easily be treated by the algorithm. The resulting motion sequences are used to examine whether relevant activities during the course oft the day can be recognized or not.

Danksagung

Mein besonderer Dank gilt Professor Dr. Arnold Neumaier für die sehr intensive und lehrreiche Zeit der Betreuung meiner Diplomarbeit.

Ich möchte mich auch bei Dipl.-Ing. Paul Panek und Professor Dr. Wolfgang Zagler vom ”Zentrum für Angewandte Assistierende Technologien” für die Zusammenarbeit bedanken und für die Daten, die mir für die Erstellung diese Arbeit zur Verfügung gestellt wurden.

Nicht zuletzt danke ich meiner Familie – ganz besonders meinem Lebensgefährten Georg – für die fortwährend Unterstützung.

Inhaltsverzeichnis

1 Einleitung

1.1 Gegenstand und Ziele der Arbeit

Gegenstand der Arbeit ist eine Analyse der durch das *eHome* Assistenzsystem über einen Zeitraum von 18 Monaten in 11 verschiedenen Wohnungen gesammelten Aktivitätsdaten. Neben einer weitgehend wohnungsunabhängigen Analyse steht die Erkennung von Aktionsklassen, die Rekonstruktion von Bewegungsabläufen und das Erkennen von typischen Verhaltensmustern im Vordergrund, sowie das Erkennen von Abweichungen und Unregelmäßigkeiten in diesem Verhalten.

1.2 Forschungsstand

Ambient Assisted Living (*umgebungsunterstütztes Leben*, oft auch *computergestütztes Wohnen*, kurz **AAL**) ist ein aktuelles, zukunftsweisendes Forschungsgebiet. Aufgrund der demographischen und sozialen Entwicklung in den industrialisierten Ländern ist zu erwarten, dass die Zahl allein lebender älterer Menschen stark steigen wird, was dramatische Effekte für das private und öffentliche Gesundheits- und Pflegewesen zur Folge haben kann. Forschung und Entwicklung im Bereich des Ambient Assisted Living kann daher einerseits zu einer großen Entlastung des Gesundheits- und Pflegewesens führen, andererseits Menschen in ihrem persönlichen Wunsch entgegenkommen, in ihrem persönlichen Umfeld zu leben. Viele Forschungsprojekte im Bereich AAL haben daher ein gemeinsames Ziel: die Unterstützung von allein lebenden älteren Menschen und deren Angehörigen, um die Zeit zu verlängern, in der diese Menschen in ihrer gewohnten privaten Umgebung bleiben können, sie sicher zu gestalten, und dadurch die Lebensqualität insgesamt zu verbessern (vgl. z.B. [9]).
Ein aktuelles Forschungsprojekt aus dem Bereich AAL ist das ***eHome* Assistenzsystem** (vgl. Abschnitt 1.4 bzw. [3, 11, 20, 21]), welches bereits erfolgreich über mehrere Monate getestet wurde (vgl. [17]) und in dessen Rahmen diese Diplomarbeit verfasst ist.
Daneben gibt eine Vielzahl unterschiedlicher AAL–Projekte. In der Literatur finden sich eine ganz Reihe von Berichten zu Forschungsergebnissen zu diesen Projekten. Einige Beispiele sind zu finden in [4, 9, 10].
Eine wissenschaftliche Arbeit, die sich mit diesem Thema befasst ist auch die Dissertation von Guo Qing Yin [18]. Er arbeitet zum Teil mit denselben Daten, die auch in der vorliegenden Arbeit verwendet werden. Auch seine Arbeit befasst sich mit der automatischen Erkennung bestimmter Szenarien und umfasst die Erstellung eines Aktivitätsmodells, sowie darauf aufbauend die Erkennung typischer Verhaltensmuster: mittels dem Split-Merge Algorithmus und Hidden Markov

Modellen wird dieses Verhalten beschrieben; mehrere Sensoren werden durch Sensor Fusion miteinander in Verbindung gebracht; eine Fallstudie zeigt, dass erkannt werden kann, ob die Daten eines beliebigen Tages als ungewöhnlich zu klassifizieren sind oder nicht.
Eine etwas allgemeinere Arbeit zu diesem Thema ist [2]. In dieser Dissertation wird die Anwendbarkeit statistischer Methoden im etwas allgemeineren Rahmen von Gebäudeautomationssystemen untersucht, wozu beispielsweise auch die Verkehrsüberwachung in Tunneln gezählt wird.

1.3 Theoretischer Rahmen

Den theoretischen Rahmen bilden mathematische Methoden, die auch im *Data Mining* und *Machine Learning* angewendet werden. Dazu zählen zum Beispiel der *K-Means Algorithmus* zur Quantisierung der Daten, der *EM-Algorithmus für Gaußsche Mixturen* zur Berechnung von Aktionsklassen oder *Bayessche Netze* und Methoden der *Dynamischen Programmierung* zur Bewegungs-Rekonstruktion. Alle Methoden sind so gewählt, dass sie möglichst wohnungsunabhängig und weitgehend automatisch ausgeführt werden können.

1.4 Das *eHome* Assistenzsystem

Die Daten, die in dieser Arbeit verwendet werden, wurden durch Prototypen des ***eHome* Assistenzsystems** gesammelt. Dieses wurde am ***"Zentrum für Angewandte Assistierende Technologien"*** der Technischen Universität Wien (kurz **AAT**, vgl. [19]) entwickelt, mit der Zielsetzung allein lebende ältere Menschen zu unterstützen. Das selbstständige Leben in der eigenen Wohnung sollte sich sicher gestalten und Angehörige sollten insofern entlastet werden, als dass sie im Falle von ungewöhnlichen, eventuell auch gefährlichen Situationen rechtzeitig verständigt werden. Der *eHome* Prototyp wurde über mehrere Monate in 11 verschiedenen Wohnungen getestet. Drei dieser Tests wurden über einen längeren Zeitraum von circa vier Monaten durchgeführt, acht Tests über einen Zeitraum von circa einem Monat.

Der *eHome* Prototyp

Der in den Wohnungen installierte ***eHome* Prototyp** besteht aus einem Netzwerk von Sensoren, einer Zentralen Recheneinheit (HCU) und einer lokalen Benutzerschnittstelle (LUI).
Für das Sensornetzwerk wurden eigens entwickelte Sensormodule (*Multi–Sensor–Boxen*, **MSB**) verwendet. Jede MSB kann folgende Sensoren beinhalten:
- Lichtsensor
- Temperatursensor
- Vibrationssensor (*Accelerometer*)
- Bewegungsmelder (PIR Sensor, *Passive Infra Red*)
- Türkontaktsensor (REED Sensor, Magnet-Kontakt-Schalter)
- Temperatursensor Kochplatte (IR Sensor, *Infra Red*)

Diese MSB wurden an verschiedenen strategischen Orten innerhalb der Testwohnungen platziert. Je nach Ort der Positionierung waren die Boxen unterschiedlich ausgestattet.
Jede Box sammelte nach der Installation selbstständig Daten und übertrug diese an die zentrale Recheneinheit, wo sie aufgezeichnet und teilweise bereits ausgewertet wurden. Da diese Daten sowohl in Echtzeit überprüft werden sollen, als auch in regelmäßigen Abständen auf signifikante Änderungen hin untersucht werden sollen, wurde bereits ein Expertensystem integriert (vgl. [17]). Es besteht aus kontextabhängigen Regeln, die vom AAT zusammen mit ExpertInnen aus dem Pflegebereich, sowie NutzerInnen erstellt wurden. Dieses Expertensystem ermöglicht eine Beurteilung der Alltagssituation und ist in der Lage Alarme auszulösen.
Eine Kommunikation zwischen den NutzerInnen und dem *eHome* System wird über LUI ermöglicht. LUI ist ein Touch–Screen–basiertes Userterminal, welches so gestaltet ist, dass auch Menschen ohne besondere Vorkenntnisse im Umgang mit Computern es nutzen können. Es dient einerseits zum Bestätigen beziehungsweise Entschärfen von Alarmmeldungen, umfasst aber auch verschiedene Funktionen wie Erinnerungen, Freisprech- und Videotelefonie, sowie begrenzte Internetnutzung wie zum Beispiel das Abfragen des aktuellen Wetters oder der Nachrichten. Ausführlichere Beschreibungen des Systems sind unter [3, 11, 20, 21] zu finden.

Die *eHome* Daten

Die Rohdatensätze beinhalten die Aufzeichnungen der übertragenen Sensormessungen. Jede Zeile der insgesamt über 5 Millionen aufgezeichneten Zeilen besteht aus 11 Einträgen wie hier dargestellt:

2010.05.07 10:46:48,1273229208,801,R,0,6,385953,180,1,18.000000,2130706433
2010.05.07 10:46:49,1273229209,197,R,0,6,385972,178,1,25.000000,2130706433
2010.05.07 10:46:49,1273229209,833,R,0,6,385982,182,1,39.000000,2130706433
2010.05.07 10:47:04,1273229224,409,R,0,5,385983,180,1,1.000000,2130706433
2010.05.07 10:47:30,1273229250,125,R,0,5,385988,180,1,1.000000,2130706433
2010.05.07 10:47:34,1273229254,135,R,0,6,385990,183,1,55.000000,2130706433
2010.05.07 10:47:39,1273229259,167,R,0,2,385991,176,1,35.000000,2130706433
2010.05.07 10:47:39,1273229259,171,R,0,2,385992,175,1,64.000000,2130706433

Der erste Eintrag gibt die Zeit und das Datum der Aufzeichnung an und der zweite Eintrag den entsprechenden Unix-Zeitstempel in Sekunden. Der dritte Eintrag – der Millisekundenanteil – wird bei der folgende Analyse nicht berücksichtigt. Auch die Einträge in der vierten und fünften Spalte bleiben unberücksichtigt (genauere Information über die Bedeutung ist nicht gegeben). Die sechste Spalte gibt an, um welchen Sensortyp es sich handelt, die achte Spalte gibt an um welche MSB es sich handelt und die zehnte Spalte enthält den gemessenen Wert. Die Einträge in den Spalten sieben, neun und elf bleiben unberücksichtigt.

Insgesamt wurde das System in 11 verschiedenen Wohnungen (bezeichnet mit TP01 bis TP11) getestet. In Tabelle 1.1 findet sich eine Übersicht über die genaue Anzahl der Testtage und den Testzeitraum der verschiedenen Wohnungen, sowie die Anzahl der Räume, der MSB und der installierten Sensoren.

	Tage	Zeitraum	Räume	MSB	Sensoren
TP01	18	15.05 – 01.06.10	8	13	53
TP02	127	22.04 – 26.08.10	7	14	53
TP03	111	14.05 – 01.09.10	9	15	58
TP04	117	07.05 – 01.09.10	5	14	47
TP05	23	29.04 – 21.05.10	6	14	53
TP06	31	31.05 – 30.06.10	5	12	40
TP07	24	08.06 – 01.07.10	7	13	51
TP08	19	05.07 – 23.07.10	8	14	55
TP09	20	09.07 – 28.07.10	9	13	52
TP10	22	03.08 – 24.08.10	6	13	52
TP11	22	06.08 – 27.08.10	8	13	54

Tabelle 1.1: Statistik über die Testwohnungen

1.5 Vorausblick, Struktur der Arbeit

In Kapitel 2 werden einige Grundlagen und Methoden, die in dieser Arbeit verwendet werden, zusammengefasst. Dazu zählen grundlegende mathematische Begriffe und Methoden; Methoden aus der Datenanalyse zur Vorverarbeitung, Reduktion und Klassifizierung der Daten, sowie eine kurze Beschreibung Bayescher Netze. Daneben wird ein Algorithmus zur Bewegungsrekonstruktion vorgestellt, welcher ausführlich in [14] beschrieben ist.

In Kapitel 3 wird eine exemplarische Analyse am Beispiel der Daten der Testwohnung TP04 durchgeführt. Abschnitt 3.1 beschreibt die deskriptive Analyse: die Rohdaten werden eingelesen und bereinigt; es wird festgestellt, welche Sensoren in der untersuchten Wohnung installiert wurden und über welchen Zeitraum sie in Betrieb waren; es wird festgestellt, ob es zu längeren Ausfällen des Assistenzsystems gekommen ist. Die Zeitintervalle zwischen den Signalen werden untersucht und die daraus resultierenden statistischen Werte werden in einer Übersichts-Tabelle zusammengefasst (vgl. Tabelle 3.1).
Danach wird die Verteilung der Signale der einzelnen Sensoren über einen Zeitraum von 24 Stunden untersucht: mit Hilfe eines Algorithmus wird versucht die theoretische Verteilung der Signale zu schätzen (3.1.5).
Im letzten Schritt werden dann, basierend auf diesen Untersuchungen, die Daten der einzelnen Sensoren mit geeigneten Methoden zu äquidistanten Zeitreihen vervollständigt (3.1.6). Aus diesen äquidistanten Zeitreihen resultiert eine Merkmalsdatenmatrix - jede Sensorzeitreihe stellt ein Merkmal dar - welche für die Anwendung weiterer Methoden geeignet ist.

In Abschnitt 3.2 wird dargestellt, wie diese Merkmalsdatenmatrix ohne relevanten Informationsverlust, mittels Hauptkomponentenanalyse oder QR-Faktorisierung, reduziert werden kann und wie die Ergebnisse aus diesen Analysen interpretiert werden können.
Danach folgt in Abschnitt 3.3 die Berechnung von Aktionsklassen. Dabei wird versucht mit (halb) automatischen Methoden eine Einteilung der Zeitfenster derart zu berechnen, dass die natürlichen Handlungsabläufe wiedergegeben werden. Erst wird ein Verfahren untersucht, bei dem Hierarchisches Clustering in Kombination mit dem K-Means-Algorithmus durchgeführt wird. Danach wird der EM-Algorithmus für Gaußsche Mixturen untersucht und die Ergebnisse dieser Algorithmen werden miteinander verglichen. Es wird zu Bemerken sein, dass diese Einteilungen in Aktionsklassen zwar sehr gute Ergebnisse liefern (welche in weiten Teilen mit den natürlichen Abläufen übereinstimmen), eine Interpretation oder Benennung der Klassen aber schwierig ist, da eine Klasse oftmals nicht einem einzigen Ereignis (zum Beispiel: WC-Gang) zuzuordnen ist, sondern in unterschiedlichen Kontexten auftreten kann. Daher wird in Kapitel 4 eine strukturiertere Herangehensweise entwickelt:

Zuerst wird dargelegt, welche Wünsche es von Seiten der Ingenieure des Assistenzsystems gibt und welche Aktivitäten besonders relevant sind (4.1). Es soll untersucht werden, ob diese Aktivitäten erkannt werden können. Daher wird darauf aufbauend ein Wohnungsmodell entwickelt (4.2), dass diese Aktivitäten berücksichtigt. Danach wird jeder Positionen innerhalb dieser Modelle eine Auswahl von Sensoren zugewiesen (4.3). Mit diesen ausgewählten Sensoren (von denen dann jeder eine konkrete Bedeutung hat) wird zuerst das Clustering der Zeitfenster wiederholt; danach wird in Kapitel 5 versucht die Bewegung der Testperson zu rekonstruieren (exemplarisch am Beispiel der Testwohnung TP07):

Die Rekonstruktion erfolgt durch einen auf Bayesschen Netzen basierenden halbautomatischen Algorithmus, bei dem verschiedene Klassifizierungs- und Optimierungsschritte hintereinander ausgeführt werden. Die Zustände der berechneten Bewegungssequenz beschreiben die verschiedenen, festgelegten Positionen interhalb der untersuchten Testwohnung. Es wird sich zeigen, dass diese Sequenzen in weiten Teilen mit den beobachtbaren Positionen der Testperson übereinstimmen; für eine robuste Ortsbestimmung ist jedoch eine Verfeinerung und Weiterentwicklung des Algorithmus nötig.

Abschließend findet sich in Kapitel 6 eine kurze Zusammenfassung der Arbeit und eine Diskussion der Ergebnisse; ein Ausblick über die nächste Arbeitsschritte wird in Kapitel 7 gegeben.

2 Mathematische Methoden

In diesem Kapitel werden einige der Grundlagen und Methoden, die in dieser Arbeit verwendet werden, kurz zusammengefasst. Es handelt sich dabei um Methoden, die im Data Mining eingesetzt werden. Dabei wird das Ziel verfolgt, aus großen Datenmengen durch (halb) automatische Verfahren bestimmte Regelmäßigkeiten, Gesetzmäßigkeiten oder verborgene Zusammenhänge zu bestimmen. Dazu zählen verschiedene Schritte: von der Vorbereitung und Vorverarbeitung der Daten, über die Anwendung von Verfahren aus Datenanalyse und Machine Learning, bis hin zur Nachbearbeitung und Interpretation der Ergebnisse. Eine sehr anschauliche Einführung in Data Mining wird beispielsweise in [15] gegeben. Viele der hier kurz vorgestellten Methoden finden sich neben anderen fortgeschrittenen Methoden, Algorithmen und Modellen, zum Beispiel in [8] oder [16], sowie auch in den Vorlesungsskripten [13, 14].

2.1 Voraussetzungen

Da die aufgezeichneten Daten reelle Werte besitzen sind, haben alle Vektoren und Matrizen die im folgenden vorkommen ausschließlich reelle Einträge.

2.1.1 Lineare Algebra

Es sei A eine $m \times n$ Matrix und B eine $n \times m$ Matrix. Die Matrix $B = \left(b_{ij}\right)$ ist die **transponierte Matrix** zu $A = \left(a_{ij}\right)$, auch $B = A^T$, falls $b_{ij} = a_{ji}$ gilt. Eine $n \times n$ Matrix A ist **orthogonal**, falls $A^T A = I$, wobei I die $n \times n$ Einheitsmatrix ist. Die **Eigenwerte** $\lambda_1, \ldots, \lambda_n$ von A sind Lösungen der charakteristischen Gleichung $det(A - xI) = 0$ von A. Der **Rang** einer Matrix A ist definiert als die maximale Anzahl linear unabhängiger Spalten.

2.1.1 Definition. (QR-Zerlegung) *Sei X eine $m \times n$ Matrix mit $m \geq n$. Die* **QR-Zerlegung** *von X ist definiert durch die Matrixzerlegung*

$$X = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

wobei Q eine orthogonale $m \times m$ Matrix und R eine rechte obere $n \times n$ Dreiecksmatrix ist.

Die Existenz der QR-Zerlegung kann für jede beliebige Matrix $X \in \mathbb{R}^{m \times n}$ mit $m \geq n$ bewiesen werden. Ein Beweis findet sich zum Beispiel in [7]. Algorithmen zur Berechnung der Zerlegung mit der *Householdertransformation* oder der *Givens–Rotation* finden sich zum Beispiel in [6].

2.1.2 Definition. (Singulärwertezerlegung) *Sei X eine beliebige $m \times n$ Matrix von Rang r. Die* **Singulärwertezerlegung** *(***SVD***, Singular Value Decomposition) ist durch die Matrixzerlegung*

$$X = U\Sigma V^T \tag{2.1}$$

definiert, wobei U eine unitäre $m \times m$ Matrix und V eine unitäre $n \times n$ Matrix ist. Σ ist eine $m \times n$ Diagonalmatrix:

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \boldsymbol{0} \\ & & \ddots & \\ & \boldsymbol{0} & & \sigma_n \end{pmatrix} = \begin{pmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

mit $\Sigma_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ und $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r = \sigma_{r+1} = \ldots = \sigma_n = 0$.

Die Existenz der Singulärwertezerlegung kann für jede beliebige $m \times n$ Matrix X bewiesen werden. Ein Beweis findet sich zum Beispiel in [6] oder in [7].

Die Diagonalelemente von Σ werden als **Singulärwerte** σ_i bezeichnet. Die Menge der Singulärwerte von X wird mit $\sigma(X) = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ beschrieben. $\sigma_i(X)$ bezeichnet den i-ten Singulärwert von X. Für die Eigenwerte λ_i der Kovarianzmatrix $A \coloneqq X^T X$ von X und die Singulärwerte σ_i von X gilt:

$$\lambda_i(A) = \sigma_i^2(X)$$

$\lambda_i(A)$ bezeichnet dabei den i-ten Eigenwert von A.

2.1.2 Statistik

Sei X eine reellwertige Zufallsvariable. Die **kumulierten Verteilungsfunktion** (**CDF**, *Cumulativ Distribution Function*) $F_X : \mathbb{R} \to [0, 1]$ von X ist definiert durch:

$$F_X(x) \coloneqq Pr(X \leq x).$$

Jede Verteilungsfunktion F ist *monoton steigend*, *rechtsseitig stetig* und es gilt: $\lim_{x \to -\infty} F(x) = 0$ und $\lim_{x \to \infty} F(x) = 1$.
Eine Funktion $f : \mathbb{R} \to [0, \infty)$ heißt **Wahrscheinlichkeitsdichtefunktion** (auch *Wahrscheinlichkeitsdichte* oder einfach *Dichte*) der Zufallsvariable X falls für alle $a \leq b \in \mathbb{R}$ gilt:

$$Pr(a \leq X \leq b) = \int_a^b f(x)dx.$$

Es ist $f(x) = F'(x) = \frac{d(F_X)}{dx}$, falls F differenzierbar ist und umgekehrt ist die Verteilungsfunktion F für eine stetige Zufallsvariable X gegeben durch: $F(x) = \int_{-\infty}^{x} f(t)dt$.
Der **Erwartungswert** (auch Mittelwert) μ oder $E(X)$ von X mit Dichtefunktion f ist definiert

durch:

$$\mu = E(X) = \int_{-\infty}^{\infty} t f(t) dt$$

Die **Varianz** $V(X)$ ist definiert als $E((X-\mu)^2)$ und die **Standardabweichung** σ_X als: $\sigma_X = \sqrt{V(X)}$.

Gaußsche Normalverteilung

Für alle $\mu \in \mathbb{R}$ und $\sigma \in \mathbb{R}^+$ ist die Gaußsche Dichtefunktion $f : \mathbb{R} \to \mathbb{R}$ definiert durch:

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

Eine stetige Zufallsvariable X mit dieser Wahrscheinlichkeitsdichte, heißt **normalverteilt** mit den Parametern μ und σ^2, auch geschrieben als $X \sim \mathcal{N}(\mu, \sigma^2)$. Der Erwartungswert von X ist μ und die Varianz ist σ^2. Ist $\mu = 0$ und $\sigma = 1$, so wird X Standard-Normalverteilt genannt.

2.2 Methoden der Datenanalyse und Statistik

2.2.1 Datenvorverarbeitung

Da es sich bei den vorliegenden Daten um Messwerte unterschiedlicher Sensoren und deshalb abhängig vom Sensortyp um kontinuierliche oder diskrete Daten handelt, welche in nicht fest vorgegebenen zeitlichen Abständen gemessen werden, müssen diese Daten vorverarbeitet werden, um mit ihnen arbeiten zu können. Dazu zählt zum Beispiel das Bereinigen der Daten von Ausreißern oder Fehlerwerten oder das Vervollständigen der Daten zu einer Merkmalsdatenmatrix (vgl. Abschnitt 3.1.6), um Methoden wie die QR-Zerlegung oder die SVD anwenden zu können. Oftmals werden auch standardisierte Daten benötigt, denn durch die unterschiedlichen Wertebereiche der verschiedenen Sensortypen können einzelne Sensoren ein zu starkes Gewicht bekommen, was die Ergebnisse verzerren würde:

Standardisierung: Die **μ-σ-Standardisierung** Z einer Zufallsvariable X mit Erwartungswert μ und Standardabweichung σ ist gegeben durch:

$$Z = \frac{X - \mu}{\sigma}.$$

Eine solche Standardisierung wird stark von den größten beobachteten Werten bestimmt und es ist daher wichtig darauf zu achten, dass die Merkmale (in unserem Fall die Zeitreihen der einzelnen Sensoren) näherungsweise Gleich- oder Normalverteilt sind. Gegebenenfalls ist dazu notwendig, manche Merkmale zu transformieren:

Transformation: Mögliche Datentransformationen sind zum Beispiel die Wurzeltransformation oder die logarithmische Transformation:

Wurzeltransformation: $f : (c, \infty) \to \mathbb{R}^+$

$$\begin{aligned} f(x) &= \sqrt{b}x - c \\ f^{-1} &= x^b + c,\ c \in \mathbb{R},\ b > 0 \end{aligned}$$

Logarithmische Transformation: $f : (c, \infty) \to \mathbb{R}$

$$\begin{aligned} f(x) &= \log_b(x - c) \\ f^{-1} &= b^x + c,\ c \in \mathbb{R},\ b > 0 \end{aligned}$$

2.2.2 Theoretische Verteilung der Sensorsignale

Hier wird eine Methode beschrieben, mit der aus der empirischen kumulierten Verteilungsfunktion F_E der Sensorsignale über 24 Stunden, mittels periodischer quadratischer Splineinterpolation (vgl. z.B. [12]), die theoretische Verteilungsfunktion F_T geschätzt werden kann (vgl Abschnitt 3.1.5):

2.2.1 Algorithmus.

1. Schritt: *Scherung (Shear Mapping): Transformation von F_E derart, dass die Werte für 0:00 Uhr und 24:00 Uhr gleich Null sind; alle anderen Werte werden proportional dazu und parallel zur y-Achse verschoben.*

2. Schritt: *Schätzen des natürlichen Anfangs der Periode. Der Zeitraum bis dahin wird an das Ende der Funktion verschoben (es entsteht also eine Periode mit viel Aktivität am Anfang und wenig am Ende). Wir erhalten dadurch eine transformierte empirische Verteilungsfunktion $\widehat{F}_E$;*

3. Schritt: *Das Konfidenzintervall wird mittels Kolmogorov Smirnov Test (vgl. 2.2.2) bestimmt; dazu wird der kritische Wert d_α, abhängig vom gewünschten Signifikanzlevel α, berechnet.*

4. Schritt: *Quadratische periodische Splineinterpolation:*

- *Unsere ersten Stützstellen für die Interpolation sind der Anfangs- und Endpunkt von $\widehat{F}_E$ (beide Null). Die erste geschätzte theoretische Verteilung F_{T_1} ist einfach die Verbindung dieser beiden Punkte (die Annahme ist also Gleichverteilung). Es wird überprüft, ob F_E bereits im Konfidenzintervall von $F_{T_1} - d_\alpha, F_{T_1} + d_\alpha$ liegt (vgl. wieder Kolmogorov Smirnov Test).*
- *Sollte das der Fall sein, so sind die Signale gleichverteilt und wir können abbrechen (das tritt zum Beispiel häufig bei den Temperatursensoren auf).*
- *Wenn nicht, wird in jedem Schritt ein Punkt, nämlich jener mit maximaler Distanz zwischen F_{T_i} und $\widehat{F}_E$ ausgewählt und die neue geschätzte theoretische Verteilungsfunktion $F_{T_{i+1}}$ berechnet.*
- *Sobald $\widehat{F}_E$ im Konfidenzintervall um die berechnete Verteilung liegt, wird abgebrochen und diese berechnete Verteilung, $\widehat{F}_T$, stellt die theoretische Verteilung dar.*

5. Schritt: *Rück-Transformation von $\widehat{F}_T$ zu F_T;*

6. Schritt: *Auf negative Verteilung hin überprüfen:*

- *Wenn es zu keine negativen Verteilung kommt, sind wir fertig;*

- *Kommt es aber doch an irgendeiner Stelle zu einer negativen Verteilung, so kann auf diesem Weg keine theoretische Verteilung bestimmt werden.*

Kolmogorov Smirnov Test

Um zu überprüfen, ob F_E der geschätzten Verteilungsfunktion F_T, mit der Signifikanz α beziehungsweise mit einer Wahrscheinlichkeit von $1 - \alpha$ folgt, wenden wir den **Kolmogorov Smirnov Test** an:
Um festzustellen ob diese Annahme erfüllt ist, wird unabhängig von der Art der Verteilung ein kritischer Wert d_α bestimmt. Dieser Wert ist abhängig vom Signifikanz-Niveau α (typische Werte sind $\alpha = 0.05$ oder $\alpha = 0.01$) und der Anzahl n der zur Verfügung stehenden empirischen Daten.
Für genügend große n ($n > 40$) kann d_α näherungsweise bestimmt werden:

$$d_\alpha = \frac{\sqrt{\log(\frac{2}{\alpha})}}{\sqrt{2n}}$$

Das bedeutet d_α wird kleiner, um so mehr Daten zur Verfügung stehen; d_α wird größer, um so kleiner das vorgegebene Signifikanz-Niveau ist.
Das so genannte **Konfidenzintervall** (auch *Vertrauensbereich* oder *Vertrauensintervall*) $F_T \pm d_\alpha$ ist jener Bereich, in dem die Werte der empirischen Verteilung liegen müssen, wenn die Hypothese F_E folgt F_T nicht abgelehnt werden soll.
Wir berechnen daher in jedem Schritt die maximale Distanz $d_{max} = \max \mid F_T - F_E \mid$, zwischen F_T und F_E, sobald $d_{max} \leq d_\alpha$ gilt können wir abbrechen und die berechnete Funktion F_T ist dann die geschätzte theoretische Verteilung der Signale.

2.2.3 Hauptkomponentenanalyse

Die **Hauptkomponentenanalyse** (**PCA**, Principal Component Analysis) dient dazu, umfangreiche Datensätze zu reduzieren indem die Vielzahl der n Merkmale eines Datensatzes auf eine möglichst geringe Zahl k aussagekräftiger Linearkombinationen, die so genannten *Hauptkomponenten*, reduziert wird. Dazu wird eine Singulärwertezerlegung $X = U\Sigma V^T$ der Datenmatrix X durchgeführt (vgl. 2.1.2). Die Singulärwerte σ_i geben den Anteil der Varianz der i-ten Komponente an der Gesamtvarianz wieder. Aufgrund dieser Anteile wird k festgelegt; als Richtwert werden häufig insgesamt 95% der Gesamtvarianz angestrebt. Der Matrix V, die so genannten **Komponenten**- oder **Ladungsmatrix**, ist die Relevanz der n Merkmale für die Komponenten zu entnehmen (durch Umsortierung wird eine bessere Interpretierbarkeit erreicht (vgl. Darstellung der Ergebnisse). Die Matrix U enthält den neuen, transformierten Datensatz und liefert, eingeschränkt auf die ersten k Spalten, den reduzierten Datensatz, mit dem weiter gearbeitet werden kann.

Darstellung der Ergebnisse

Durch eine Umsortierung der Komponentenmatrix V wird erreicht, dass alle Merkmale gleich gewichtet werden und eine einfachere Interpretation der Ergebnisse möglich ist:

2.2.2 Algorithmus.

Schritt 1: *Dividieren der Spalten der Matrix V durch ihr betragsgrößtes Element; Multiplizieren aller Werte mit 100; Runden.*

Schritt 2: *Sortieren der Zeilen von V nach Relevanz:*
- *Schritt 1: die Variable, die in der 1. Komponente den betragsgrößten Eintrag besitzt, wird in die erste Zeile verschoben.*
- *Schritt k: von den verbleibenden Zeilen wird jene, die bis zur k-ten Komponente den betragsgrößten Eintrag besitzt, an die k-te Stelle verschoben.*

Schritt 3: *Entfernen aller Werte kleiner einer gewünschten Größe (oft 40 oder 60), um bessere Übersicht zu erreichen.*

2.2.4 QR-Zerlegung

Durch eine QR-Zerlegung einer Datenmatrix Xn kann Information über die Relevanz der einzelnen Merkmale und über Abhängigkeiten zwischen den Merkmalen gewonnen werden. Dazu wird erst die vorbearbeitete Datenmatrix X in die Faktoren Q und R, wie in 2.1.1 beschrieben, zerlegt. R enthält dann die gewünschten Informationen und wird zur besseren Interpretation bearbeitet:

Darstellung der Ergebnisse

Wieder wird durch Umsortierung der Ergebnis–Matrix, diesmal der Matrix R, eine bessere Interpretierbarkeit erreicht:

2.2.3 Algorithmus. (Sortieren der Matrix R)

Schritt 1: *Permutation von R so dass der Absolutbetrag der Diagonalelemente* $|\operatorname{diag}(R)|$ *abnimmt (QR-Zerlegung mit Pivotsuche).*

Schritt 2: *Berechnen von* $w := \sum \operatorname{diag} R^2$*;*

Schritt 3: *Ersetzten der Einträge* R_{ij} *von R durch* $\frac{R_{ij}}{w}^2$

Schritt 4: *Auswahl eines Grenzwertes* δ*; alle Werte kleiner* δ *werden, zur besseren Übersicht, nicht dargestellt.*

Die Diagonalelemente der umsortierten Matrix repräsentieren das Gewicht der Merkmale, die restlichen dargestellten Elemente (also jene größer δ) liefern die Abhängigkeitinformation.

2.3 Clustering

2.3.1 Unüberwachtes Clustering

Beim unüberwachten **Clustering** geht es darum, in einer Menge von Daten mehrere Kategorien (Cluster) zu erkennen, wobei die Kategoriebeschriftungen nicht vorgegeben sind.

K-Means Algorithmus

Ein sehr einfaches unüberwachtes Clusteringverfahren stellt der **K-Means Algorithmus** dar, bei dem jedes Cluster durch einen Prototypen repräsentiert wird:

2.3.1 Algorithmus. (K-Means)

1. Schritt: *Es werden zufällig k Prototypen* $v_i, i = 1, \ldots, k$ *ausgewählt.*

2. Schritt: *Zuordnen der Datenpunkte* x_n*, mit* $n = 1, \ldots, N$ *wobei N die Anzahl der Datenpunkte ist, zu den Clustern* $C_i, i = 1, \ldots, k$*, zum Beispiel durch die* Nächste-Nachbar-Regel*:*

$$x_n \in C_i \Longleftrightarrow | \, x_n - v_i \, | = \min_{j=1,\ldots,c} | \, x - y_j \, |$$

3 Schritt: *Berechnen der neuen Clusterzentren:*

$$v_i = \frac{1}{| \, C_i \, |} \sum_{x_n \in C_i} x_n = \frac{\sum_{n=1}^{N} u_{in} x_n}{\sum_{n=1}^{N} u_{in}},$$

dabei sind u_{in}*, die Elemente der* Partitionsmatrix U*, definiert durch:*

$$u_{in} := \begin{cases} 1 & \text{falls } x_n \in C_i \\ 0 & \text{falls } x_n \notin C_i \end{cases}$$

4. Schritt: *Iterieren der Schritte 2 und 3 bis Konvergenz erreicht ist.*

EM-Algorithmus für Gaußsche Mixturen

Ein komplexeres Verfahren ist beispielsweise der **EM-Algorithmus für Gaußsche Mixturen** (*Expectation Maximization Algorithm*), der in vielen Bereichen Anwendung findet:
Es wird davon ausgegangen, dass die Datenpunkte einer Multivariaten Gaußschen Verteilung P entsprechen. Eine solche Verteilung besteht aus k Komponenten, welche jede für sich eine eigene Verteilung darstellt. Die Parameter einer solchen Verteilung sind die Gewichtung $w_i = P(C = i)$, das Mittel μ_i und die Kovarianz Σ_i jeder Komponente $C = i$.
Problemstellung ist nun, sowohl die Parameter des Modells zu schätzten, als auch festzulegen, zu welcher Komponente C die einzelnen Datenpunkte $\mathbf{x_j}$ gehören.

2.3.2 Algorithmus. (EM-Algorithmus für Gaußsche Mixturen) *Zuerst werden die Parameter des Mixtur-Modells zufällig initialisiert. Danach werden die beiden folgenden Schritte bis zur Konvergenz iteriert (vgl. [16]):*

E-Schritt: *Berechnung von:*

$$p_{ij} = P(C = i \mid \mathbf{x}_j),$$

der Wahrscheinlichkeit, dass der Datenpunkt $\mathbf{x}_j$ *von der Komponente i erzeugt wurde. Aus dem*

Satz von Bayes *folgt:*

$$p_{ij} = \alpha P(\mathbf{x}_j \mid C = i)P(C = i).$$

Definiere:

$$n_i = \sum_j p_{ij}$$

als die effektive Anzahl von Datenpunkte, die der Komponente i aktuell zugeordnet sind.

M-Schritt: *Neuberechnung des Mittels, der Kovarianz und der Gewichtungen der Komponenten:*

$$\mu_i \leftarrow \sum_j p_{ij}\mathbf{x}_j/n_i$$
$$\Sigma_i \leftarrow \sum p_{ij}(\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top/n_i$$
$$w_i \leftarrow n_i/N$$

wobei N die Anzahl der Datenpunkte ist.

Die Konvergenz dieses Algorithmus kann bewiesen werden. Ein Beweis findet sich zum Beispiel in [8].

2.3.3 Bemerkung. Oft liefert der EM-Algorithmus sehr gute Ergebnisse – es kann aber auch sein, dass es manchmal schiefgeht. Das kann zum Beispiel an einer schlechten Initialisierung liegen. Eine Wiederholung kann dann zu wesentlich besseren Ergebnissen führen. Es ist auch möglich, die Parameter von vornherein mit sinnvollen Werten zu initialisieren. Sollten mehrere Komponenten in einer vermischt sein, zum Beispiel weil sie eine große Ähnlichkeit aufweisen, so kann diese Komponente mit neuen Zufallsparametern erneut gestartet und in mehrere Komponenten aufgeteilt werden.

Ein anderes Clustering-Verfahren ist zum Beispiel das Hierarchische Clustering. Daneben gibt es auch überwachte Verfahren, die dann verwendet werden können, wenn bereits Kategorien vorgegeben sind. Ein Beispiel dazu ist die Klassifikation mit Entscheidungsbäumen (*Decision–Trees*). Diese und weitere Verfahren finden sich zum Beispiel in [1].

2.4 Bayessche Netzwerke

Ein **Bayessches Netzwerk** ist wie folgt definiert (vgl. [16]):

- Die Knoten des Netzwerks werden durch eine Menge von Zufallsvariablen gebildet. Diese können diskret oder stetig sein.
- Die Knoten sind verbunden durch eine Menge von gerichteten Kanten. X ist ein Elternknoten von Y, wenn es eine Kante von X nach Y gibt.
- Für jeden Knoten X_i ist eine bedingte Wahrscheinlichkeitsverteilung $P(X_i \mid Eltern(X_i))$ definiert.

- Es handelt sich bei dem Netzwerk um einen *azyklischen* Graphen, d.h. es gibt keine gerichteten Zyklen.

Die Topologie des Netzwerks legt die geltenden bedingten Unabhängigkeitsbeziehungen fest. Die Kombination aus der Topologie und den bedingten Verteilungen reicht aus, um die vollständige gemeinsame Verteilung für alle Variablen zu bestimmen:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Eltern(X_i))$$

Jeder Eintrag in der gemeinsamen Wahrscheinlichkeitsverteilung berechnet sich also durch das Produkt der bedingten Wahrscheinlichkeitstabellen des Bayesschen Netzwerks.

2.4.1 Dynamische Bayessche Netzwerke

Ein **dynamisches Bayessches Netzwerk** repräsentiert ein temporales Wahrscheinlichkeitsmodell der folgenden Art:
Es besteht aus eine Menge von Zufallsvariablen, von denen manche beobachtbar sind und manche nicht. $\mathbf{X}_t$ bezeichnet die Menge der nicht beobachtbaren *Zustandsvariablen* zur Zeit t, $\mathbf{E}_t$ die Menge der beobachtbaren *Evidenzvariablen* zur Zeit t.
Zum Beispiel bildet ein Hidden Markov Modell ein einfaches Bayessches Netzwerk. Es besteht aus zwei Zufallsvariablen – einer Zustandsvariable X und einer Evidenzvariable (auch Beobachtungsvariable) Y. Die Wahrscheinlichkeit für X_t zum Zeitpunkt t hängt von X_{t-1} ab (beschreibt also ein Markov Modell 1. Ordnung). Die Wahrscheinlichkeit für Y_t zum Zeitpunkt t hängt von X_t ab (vgl. wieder [16] oder [8].

2.5 Bewegungs–Rekonstruktion

Das folgende Modell zur **Bewegungs–Rekonstruktion** wurde im Zuge der Arbeit von Professor Dr. Arnold Neumaier ausgearbeitet und findet sich im Vorlesungsskriptum [14] zu Methoden der Datenanalyse. Es wird angenommen, dass sich ein Subjekt innerhalb eines Netzwerkes von Knoten und ungerichteten Kanten bewegt. Des weiteren wird angenommen, dass eine unbestimmte Menge von Sensoren an den Knoten und an manchen Kanten Daten sammelt. Wir betrachten dann das Problem, die Bewegung innerhalb eines solchen Netzwerkes zu rekonstruieren.

2.5.1 Problemstellung

Gegeben sei ein Netzwerk $\mathcal{W}$ bestehend aus Knoten $N \in X^{\mathcal{W}}$, der Menge aller Knoten von $\mathcal{W}$, und Kanten AB zwischen den Knoten A und B. Die von den Sensoren gesammelten Daten liegen in Form von äquidistanten Zeitreihen ($t = 1, \ldots, T$) vor.
Jedem Knoten sei eine bestimmte Menge von Sensoren $z_t^N \in Z^N$, der Menge der Sensoren der Knoten, zugeordnet. Abhängig davon wird jedem Knoten ein bestimmter Typ τ zugeordnet, der angibt wieviele und welche Sensoren dieses Knotens zur folgenden Bewegungs–Rekonstruktion herangezogen werden sollen. Dem entsprechen wird eine Abbildung $\zeta^N : Z^N \to Z^\tau$ definiert, so

dass die erhaltenen Zeitreihen $\zeta_t^{\mathcal{N}} := \zeta^{\mathcal{N}}(z_t^{\mathcal{N}})(t = 1 \dots T)$ bei allen Knoten desselben Typs eine vergleichbare Bedeutung haben.
Jeder Kante AB wird eine Zeitreihe $z_t^{AB} \in Z^{AB}$, der Menge der Sensoren der Kanten, $(t = 1 \dots T)$, zugeordnet. Sollten für eine Kante AB keine Daten verfügbar sein, so bleibt z_t^{AB} leer (dasselbe gilt für $\zeta_t^{\mathcal{N}}$, wenn an dem Knoten keine Daten verfügbar sind).
Gesucht ist eine Zeitreihe $x_t^{\mathcal{W}}(t = 1 : T)$ welche die geschätzte Position $x_t^{\mathcal{W}} \in X^{\mathcal{W}}$ des Subjekts zum Zeitpunkt t in $\mathcal{W}$ wiedergibt.

2.5.2 Lösungsschritte

Die Lösung erhalten wir in mehreren Schritten. Der erste Schritt ist unabhängig von $\mathcal{W}$ (d.h. die Knoten mehrerer Wohnungen können gemeinsam verarbeitet werden) und wird *Offline* durchgeführt.

Schritt 1

Für jeden Typ τ wird die Vereinigung aller $\zeta_t^{\mathcal{N}}$ mit $\mathcal{N}$ von Typ τ in c Cluster (mittels *EM-Algorithmus*) aufgeteilt. Daraus resultieren Zeitreihen $\gamma_t^{\mathcal{N}}$ mit Einträgen von 1 bis c. Die Klassennamen (*Labels*) werden so vergeben, dass häufigere Klassen kleinere Labels besitzen.
Jeder Klasse γ eines Knotens $\mathcal{N}$ wird eine geschätzte anfängliche Aufenthaltswahrscheinlichkeit $\widetilde{p}(\gamma)^{\mathcal{N}} \in [0, 1]$ zugewiesen.

Schritt 2

Auch dieser Schritt kann unabhängig von $\mathcal{W}$, also für alle Kanten aller Wohnungen gemeinsam durchgeführt werden. Für jede Kante AB werden drei Wahrscheinlichkeiten berechnet:

$$\widetilde{p}_0^{AB}(\gamma^A, \gamma^B) := (1 - \widetilde{p}(\gamma^A))(1 - \widetilde{p}(\gamma^B)), \tag{2.2}$$

$$\widetilde{p}_A^{AB}(\gamma^A, \gamma^B) := \widetilde{p}(\gamma^A) - q^{AB}(1 - \widetilde{p}(\gamma^A)), \tag{2.3}$$

$$\widetilde{p}_B^{AB}(\gamma^A, \gamma^B) := \widetilde{p}(\gamma^B) - q^{AB}(1 - \widetilde{p}(\gamma^B)). \tag{2.4}$$

wobei

$$q^{AB} = \frac{\widetilde{p}(\gamma^A)\widetilde{p}(\gamma^B)}{2 - \widetilde{p}(\gamma^A) - \widetilde{p}(\gamma^B)}$$

Dabei bezeichnen 2.2 bis 2.4 die Wahrscheinlichkeiten weder in A noch in B zu sein, in A zu sein, in B zu sein. Diese summieren sich für feste A und B zu 1.
Für jede Kante AB werden anfängliche Zustands-Wahrscheinlichkeits-Zeitreihen $p_t^{AB}(x)$ berechnet:

$$p_t^{AB}(x) := \widetilde{p}_x^{AB}(\gamma_t^A, \gamma_t^B),$$

sowie anfängliche Zustandssequenzen x_t^{AB} mit den *Hidden States* A, B und 0:

$$x_t^{AB} := \underset{x}{\operatorname{argmax}}(p_t^{AB}(x)).$$

Dann wird die Vereinigung aller (z_t^A, z_t^B, z_t^{AB}) klassifiziert. Diesmal kann eine überwachte Klassifizierung (*Decision Tree*) in c_1 Klassen durchgeführt werden, so dass die geschätzten Zustandssequenzen x_t^{AB} dabei *sinnvoll* geteilt werden. Es resultieren daraus Zeitreihen $y_t^{AB}(t = 1, \ldots, T)$ mit Einträgen aus $[1, \ldots, c_1]$ wobei wieder häufigere Klassen kleinere Labels tragen.
Dann werden die folgenden bedingten Wahrscheinlichkeiten berechnet:

$$p^{AB}(x \mid y) := \frac{N(x, y)}{N(x)},$$

wobei

$$N(x, y) = 1 + \sum_{y_t = y} p_t^{AB}(x)$$

und

$$N(x) = \sum_y N(x, y).$$

p_x^{AB} wird nun neu definiert:

$$p_x^{AB} := p(x \mid y_t),$$

Wir berechnen das *Potential* (den negativen Logarithmus) der regularisierten relativen Häufigkeiten von $(x_t^{AB}, y_t^{AB}) = (x, y)$.:

$$V^{AB}(x, y) = -\log p(x \mid y).$$

Schritt 3

Dieser Schritt wird für jede Wohnung getrennt durchgeführt und kann so gestaltet werden, dass ein *Online*-Lernen möglich ist:
Für jede Kante wird ein trainiert:
Die versteckten Zustände sind $x_t \in X^{AB}$, die Fronten sind $\{x_t, y_t, x_{t+1}, y_{t+1}\}$, trainiert wird mit Viterbi Training:

$$x_{1:T} = \underset{(x_t, x_{t+1}) \in \mathcal{T}}{\operatorname{argmin}} \sum_{t=1}^{T-1} V(x_t, y_t, X_{t+1}, y_{t+1}). \tag{2.5}$$

Das Potential ist dabei definiert durch:

$$V(x, y, x', y') := -\log \frac{N(x, y, x', y')}{N(x, y)},$$

mit den anfänglichen Werten:

$$N(x, y, x', y') \coloneqq 1 + \sum_{y_t = y, y_{t+1} = y'} p_t(x) p_{t+1}(x'),$$

$$N(x, y) \coloneqq \sum_{x', y'} N(x, y, x', y').$$

Für spätere Iterationen gilt für $p_t(x)$:

$$p_t(x) = \begin{cases} 1 & \text{falls } x_t = x; \\ 0 & \text{sonst.} \end{cases}$$

wobei x_t die Lösung von (2.5) der vorhergehenden Iteration ist.
Nach dem Training wird $V^{AB}(x, y)$ neu berechnet.

Schritt 4

Auch dieser Schritt wird für jedes Netzwerk getrennt durchgeführt:
Für jede Kante AB und für jeden Zeitpunkt t sei $o_t^{AB} = 0$, es sei denn ein Sensor bestätigt mit Gewissheit, dass ein Übergang zwischen A und B zu diesem Zeitpunkt nicht möglich ist (zum Beispiel eine geschlossene Tür), dann ist $o_t^{AB} = 1$. Eine konsistente Zuordnung der Zustände $x_t \in X^{\mathcal{W}}$ wird erreicht durch:

$$x_{1:T} \coloneqq \operatorname*{argmin}_{x_t, x_{t+1} \in \mathcal{T}_t^{\mathcal{W}}, o_t^{AB} = 0} \sum_{t=1}^{T} V_t(x_t)$$

wobei $\mathcal{T}_t^{\mathcal{W}}$ aus allen permutierten Übergängen zwischen den Knoten besteht und

$$V_t(x_t) \coloneqq \sum_{AB} V^{AB}(x_t, y_t^{AB}).$$

Summiert wird dabei über alle Kanten AB der Wohnung $\mathcal{W}$.

Dann wird die Vereinigung der $(y_t^{A_1 B_1}, \ldots, y_t^{A_m, B_m})$ klassifiziert, in s Klassen eingeteilt, die wieder so benannt sind, dass häufigere Klassen kleiner Labels tragen. Daraus resultiert eine Zeitreihe $y_t^{\mathcal{W}}, t = 1, \ldots, T$ mit Einträgen von 1 bis s.

Schritt 5

Für jedes Netzwerk $\mathcal{W}$ wird ein diskretes mit den Fronten $\{x_t, y_t, x_{t+1}, y_{t+1}\}$ trainiert. Die versteckten Zustände nehmen Werte aus $X^{\mathcal{W}}$ an und sind wie oben dadurch beschränkt, dass $(x_t, x_{t+1}) \in \mathcal{T}_t$ und $o_t^{AB} = 0$.
Mittels **Viterbi Training** wird das Netzwerk trainiert:

$$x_{1:T} = \underset{(x_t, x_{t+1}) \in \mathcal{T}}{\operatorname{argmin}} \sum_{t=1}^{T-1} V(x_t, x_{t+1}, y_t, y_{t+1}).$$

Das Potential $V(x_t, x_{t+1}, y_t, y_{t+1})$ ist wieder der negative Logarithmus der regularisierten relativen Häufigkeiten.

3 Analyse

3.1 Deskriptive Analyse

3.1.1 Einlesen der Rohdaten und Feststellen der Sensoren

Die Rohdatensätze wurden einer mathematischen Offline-Analyse unterzogen. Dazu wurde zuerst für jede Wohnung die wesentliche Information aus den Rohdaten extrahiert. Es handelt sich dabei jeweils um die Zeit, den Sensortyp, die MSB und den Wert (vgl. 1.4).
Danach wurden für jede Wohnung die vorhanden Sensoren festgestellt und die Daten nach Sensoren sortiert. Pro Wohnung wurden um die 50 Sensoren installiert. Die genaue Anzahl kann Tabelle 1.1 entnommen werden. Es wurde festgestellt, dass manche Sensoren nur eine sehr kurze Zeit in Betrieb waren – manche nur einen einzigen Tag. Diese Sensoren wurden aus den weiteren Untersuchungen ausgeschlossen. Dazu wurde während dem Einlesen der Daten eine Grenze bestimmt, ab der Sensoren nicht mehr berücksichtigt werden sollten. Abbildung 3.1 zeigt die entsprechenden Abbildungen aus dem Programm: Links ist die Laufzeit aller Sensoren von Testwohnung TP04 zu sehen. Interaktiv wurde an dieser Stelle die Grenze für die Mindestlaufzeit festgelegt. Rechts sind nur jene Sensoren abgebildet, deren Laufzeit die gewählte Grenze nicht unterschreitet und die in weiterer Folge berücksichtigt werden.
Den festgestellten Sensoren wurden daraufhin Namen für die weitere Analyse zugewiesen. Dazu wurden die MSB durchnummeriert (beginnend mit 1) und jedem Sensortyp wurde eine Abkürzung zugewiesen: V = Vibration; M = Move (Bewegung); D = Door (Türkontakt); H = Hot Plate (Herdplatte); T = Temperature (Temperatur); L = Ligth (Licht). Zusammengesetzt ergaben sich daraus die eindeutigen Bezeichnungen wie zum Beispiel V05 für Vibrationssensor der MSB 5.

3.1.2 Feststellen längerer Ausfälle des Assistenzsystems

Um längere Ausfälle des Systems erkennen zu können, wurden die Zeitabstände zwischen allen Signalen berechnet. Wieder konnte mittels interaktiver Auswahl eine Grenze für zu große Zeitintervalle angegeben werden. Dabei kann berücksichtigt werden, dass aufgrund der Konfiguration der Sensoren eigentlich zumindest alle 10 Minuten ein Signal gesendet werden sollte (minimales Sendeintervall der Lichtsensoren). In Abbildung 3.2 sind links alle berechneten Zeitintervalle zwischen den Signalen zu sehen; rechts sind bereits jene Intervalle ausgeschlossen, die als Ausfall eingestuft wurden.

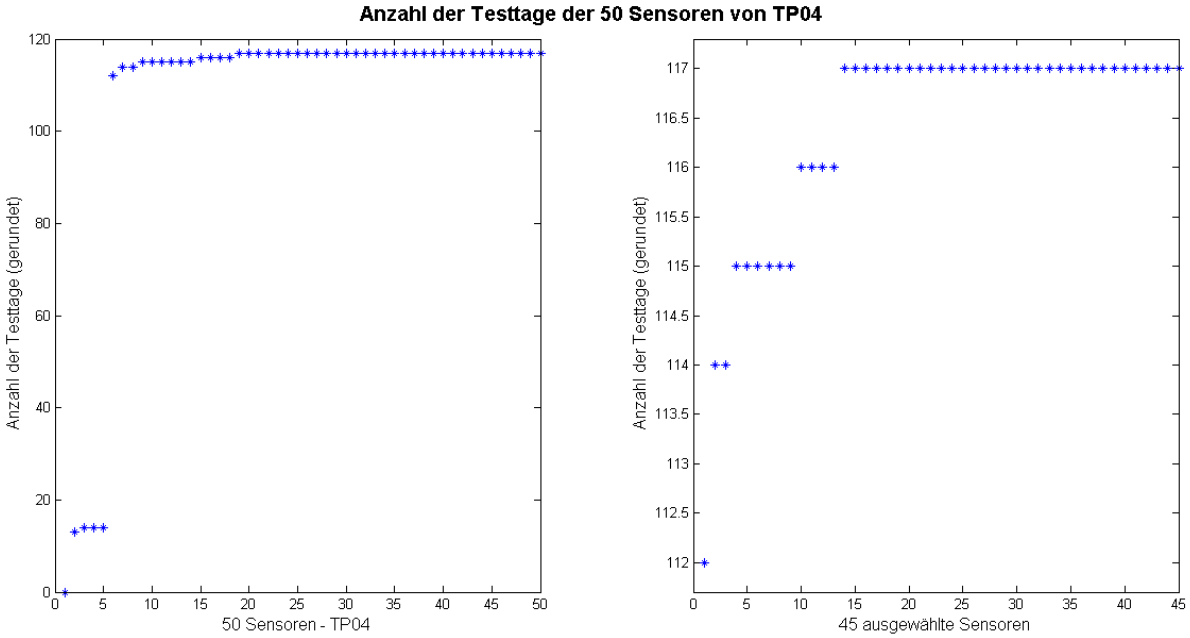


Abbildung 3.1: Anzahl der Testtage der Sensoren von TP04. Auszug aus dem interaktiven Programm. Links: alle Sensoren – interaktiv wurde an dieser Stelle die Grenze für die Mindestlaufzeit festgelegt. Rechts: die ausgewählten Sensoren, die weiter untersucht wurden.

3.1.3 Feststellen von Ausreißern unter den Werten

Auch die gemessenen Werte müssen auf mögliche Ausreißer hin untersucht werden. Es gibt beispielsweise für die Lichtsensoren oder für die Herd-Temperatursensoren Fehlerwerte, die übertragen werden, wenn ein Sensor nicht richtig arbeitet oder wenn er nicht angesteckt ist.
Um solche und andere Ausreißer festzustellen, werden die Werte wieder in einer Abbildung dargestellt (vgl. Abbildung 3.3) und interaktiv werden wieder Grenzwerte für zu große Werte bestimmt. Diese wurden aussortiert und die verbliebenen Werte sind in Abbildung 3.4 zu sehen (im Unterschied zu 3.3 sind sie sortiert).
Es kann beobachtet werden, dass die verschiedenen Sensortypen sehr unterschiedliche Werte senden. Die Türkontaktsensoren senden ausschließlich die Werte 0 und 1. Die Vibrationssensoren dagegen senden kontinuierliche Werte größer Null – nur wenige Werte liegen in einem sehr hohen Bereich, die meisten sind in einem niedrigeren Wertebereich angesiedelt und daher wurden diese Werte für alle weiteren Untersuchungen transformiert (vgl. 2.2.1), wodurch eine bessere Verteilung der Werte erreicht wurde.
Die Werte der Temperatursensoren liegen zwischen 12 und 32 Grad, wobei die meisten im Bereich 20 bis 25 Grad liegen.
Die Bewegungsmelder senden bei Bewegung ausschließlich 1.
Die Lichtsensoren senden kontinuierliche Werte größer, aber auch gleich Null. Wieder gibt es relativ wenige sehr große Werte und viele eher niedrige Werte. Daher wurden auch diese Werte logarithmiert. In Tabelle 3.1 sind die verschiedenen statistischen Werte der einzelnen Sensoren von TP04 zusammengefasst.

Sensor	Signale	Werte					Intervalle				Längere	Intervalle-Klassen				
	Gesamt	Min	Mean	Max	Std	Ausn.	Min	Mean	Max	Std	Ausfälle	Zu Klein	Normal	Zu Groß	Ausn.	Ausn.
			x					$\log_{60}(sec)$				<0s	8s:24h	>24h	0:8s	
D08	3473	0	0.5	1	0.5		0.51	1.50	2.66	0.61	4	0.00	75.70	0.03	24.27	
D10	475	0	0.5	1	0.5		0.51	1.76	2.77	0.72	4	0.00	45.22	3.40	51.38	
D11	1132	0	0.5	1	0.5		0.51	1.83	2.78	0.51	4	0.00	88.65	0.53	10.82	
D12	3286	0	0.5	1	0.5		0.51	1.55	2.66	0.57	4	0.00	78.15	0.00	21.85	
D13	1630	0	0.5	1	0.5		0.51	1.51	2.77	0.66	4	0.00	80.81	0.55	18.63	
			$\log(x)$					$\log_{60}(sec)$				<0s	6s:24h	>24h	0:5s	
V01	12998	2.40	4.78	8.25	0.81		0.44	1.25	2.66	0.55	5	0.00	44.87	0.00	55.13	
V02	2415	2.64	5.17	8.25	1.23		0.44	1.28	2.77	0.77	5	0.00	32.61	1.12	66.27	
V03	5055	2.40	5.01	8.26	1.08		0.44	1.30	2.77	0.64	6	0.00	47.89	0.04	52.07	
V04	2153	2.30	5.13	8.26	1.31		0.44	1.52	2.78	0.71	5	0.00	45.76	0.51	53.72	
V05	3191	2.20	5.03	8.28	1.18		0.44	1.27	2.77	0.71	5	0.00	39.52	0.50	59.98	
V06	4733	2.56	5.07	8.26	1.04		0.44	1.24	2.77	0.67	5	0.00	39.21	0.23	60.55	
V07	4754	2.20	4.78	8.28	0.96		0.44	1.22	2.77	0.64	5	0.00	49.51	0.08	50.41	
V09	125073	2.40	4.97	8.26	0.66		0.44	0.93	2.75	0.30	5	0.00	60.58	0.00	39.42	
V10	13030	2.30	5.94	8.24	1.05		0.44	1.21	2.78	0.65	5	0.00	18.53	0.01	81.46	
			x					$\log_{60}(sec)$				<1m	1m:1h	>24h4m	1h:1h4m	
T01	2866	15.0	19.4	26.0	2.1		1.11	1.69	2.00	0.26	5	0.00	6.33	1.92	91.75	
T02	2844	16.0	19.6	27.5	1.9		1.09	1.67	2.00	0.29	5	0.00	3.06	0.35	96.58	
T03	2829	16.0	19.4	28.0	1.9		1.08	1.62	2.00	0.30	5	0.00	2.37	0.89	96.74	
T04	2774	12.5	19.3	28.5	2.7		1.00	1.60	2.00	0.29	5	0.00	2.67	4.77	92.56	
T05	2840	15.5	19.3	28.0	2.2		1.10	1.62	1.98	0.29	5	0.00	2.26	0.35	97.39	
T06	2775	15.5	19.1	29.5	2.1		1.09	1.53	2.00	0.30	5	0.00	7.40	1.30	91.30	
T07	2856	14.5	19.1	29.5	2.3		1.11	1.63	1.99	0.26	5	0.00	5.54	1.09	93.37	
T08	2910	12.0	19.7	29.5	3.4		1.00	1.77	2.00	0.23	5	0.00	10.46	0.90	88.64	
T09	2562	17.0	20.5	27.5	1.8		1.07	1.68	2.00	0.29	5	0.00	2.74	54.24	43.02	
T10	3053	16.5	21.5	29.0	2.4		1.00	1.70	2.00	0.23	5	0.00	17.78	0.43	81.79	
T11	2892	17.5	23.0	31.0	2.6		1.06	1.84	2.00	0.19	5	0.00	8.80	0.03	91.17	
T12	2836	17.0	21.6	30.0	2.2		1.09	1.74	2.00	0.21	5	0.00	9.68	23.42	66.90	
T13	2915	17.0	21.4	30.0	2.3		1.00	1.73	2.00	0.26	5	0.00	13.68	13.88	72.44	
			x					$\log_{60}(sec)$				<5s	10s:24h	>24h	5:10s	
M04	42608	1	1	1	0		0.59	0.93	2.74	0.43	4	0.10	26.14	0.01	73.76	
M08	8372	1	1	1	0		0.59	1.42	2.67	0.56	5	0.14	40.33	0.00	59.53	
M10	38245	1	1	1	0		0.59	1.15	2.66	0.57	3	0.10	11.24	0.00	88.65	
M12	171825	1	1	1	0		0.59	0.91	2.67	0.36	4	0.05	23.15	0.00	76.80	
M13	69921	1	1	1	0		0.59	0.89	2.76	0.31	4	0.08	43.75	0.00	56.17	
			$\log(x+1)$			65535		$\log_{60}(sec)$				<1m	1:10m	>10m2s	60s	10m:10m2s
L01	58803	0.00	1.19	6.61	0.50		1.00	1.26	1.56	0.14	5	0.00	51.31	1.17	42.41	5.10
L02	22180	0.00	0.65	7.07	0.74		1.00	1.40	1.56	0.17	5	0.00	37.84	2.52	11.41	48.23
L03	29365	0.00	0.70	6.97	0.56		1.00	1.35	1.56	0.15	5	0.00	54.94	2.59	17.04	25.42
L04	22769	0.00	1.60	6.81	1.62		1.00	1.36	1.56	0.18	5	0.01	42.58	6.75	10.00	40.66
L05	25553	0.00	1.61	7.12	1.33		1.00	1.37	1.56	0.17	5	0.01	43.67	1.60	16.41	38.30
L06	19708	1.39	1.77	11.09	0.64	5	1.00	1.41	1.56	0.16	5	0.01	35.54	4.75	9.27	50.44
L07	24814	0.00	1.32	11.09	1.24	2	1.00	1.37	1.56	0.17	5	0.00	40.21	3.57	17.71	38.51
L08	21223	1.39	2.83	11.09	1.20	4	1.00	1.40	1.56	0.17	5	0.00	43.34	6.08	5.89	44.70
L09	15400	1.61	1.69	11.09	0.58	3	1.00	1.51	1.56	0.13	5	0.00	9.18	64.68	0.56	25.58
L10	22356	1.39	2.59	11.09	1.09	1	1.00	1.38	1.56	0.17	5	0.00	41.64	6.02	10.42	41.92
L11	23141	0.00	2.22	11.09	1.65	3	1.00	1.39	1.56	0.16	5	0.01	42.29	3.22	11.38	43.10
L12	20220	1.39	2.35	11.09	0.97	3	1.00	1.38	1.56	0.17	5	0.00	35.38	30.93	5.19	28.49
L13	21142	1.39	2.84	11.09	1.28	1	1.00	1.38	1.56	0.17	5	0.01	42.33	13.76	5.73	38.17

Tabelle 3.1: Statistik über die einzelnen Sensoren; TP04

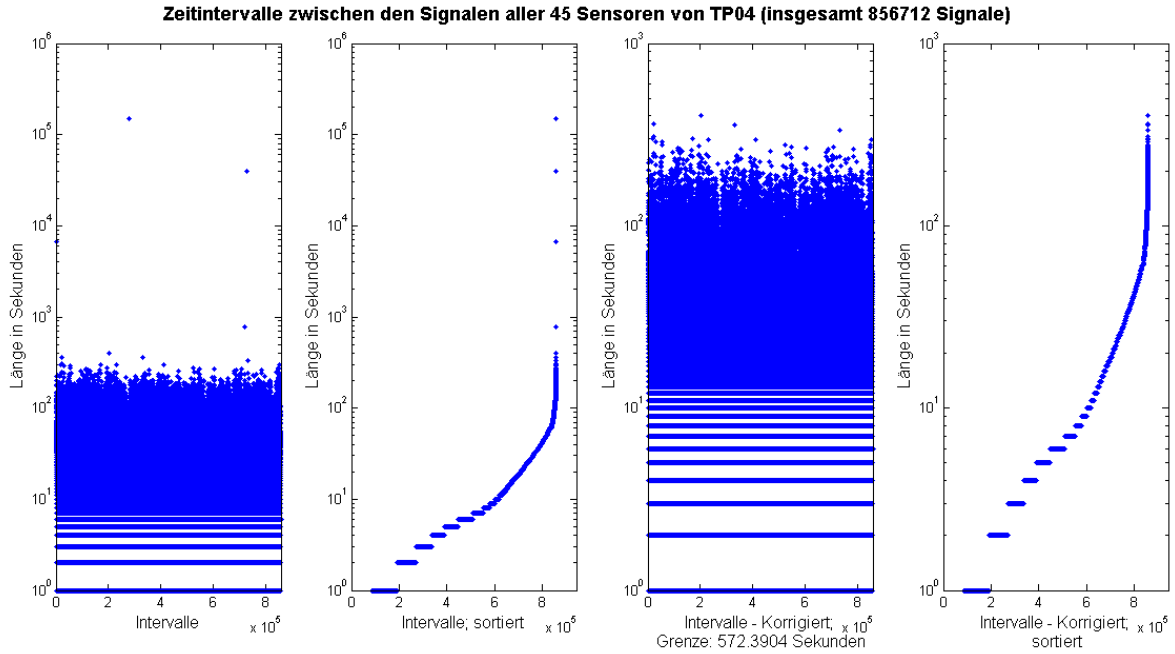


Abbildung 3.2: Zeitintervalle zwischen den Signalen aller Sensoren; Auszug aus dem interaktiven Programm; Die zwei linken Abbildungen zeigen alle Intervalle – links nicht sortiert, rechts sortiert; Die zwei rechten Abbildungen zeigen die akzeptierten Intervalle, alle aussortierten Intervalle gelten als Ausfälle; TP04

3.1.4 Untersuchung der Zeitintervalle

Auch die Zeitintervalle zwischen den Signalen wurden genauer untersucht. Es stellte sich heraus, dass jeder Sensortyp sehr charakteristische Häufungen bei den Zeitintervalllängen ausweißt, welche sich durch die Konfiguration der Sensoren bezüglich minimalem und maximalem Sendeintervall erklären lassen. In Abbildung 3.5 sind die aufgetretenen Zeitintervalle abgebildet.

Die Türkontaktsensoren haben ein minimales Sendeintervall von 100 ms. Daher sind mehrere Signale in einer Sekunde möglich. Meistens werden beim Öffnen und Schließen einer Tür gleich mehrere Signale hintereinander gesendet, dann kann es aber zu signifikant unterschiedlich langen Intervallen bis zur nächsten Türbewegung kommen. Das lässt sich in der Abbildung 3.5, rechts oben, gut beobachten.

Vibrationssensoren haben ein minimales Sendeintervall von 200 Millisekunden. Daher können auch diese mehrmals in der Sekunde Signale senden und auch hier ist zu bemerken, dass kleine Intervalllängen häufig vorkommen.

Das minimale Sendeintervall der Temperatursensoren ist eine Minute. Gesendet wird bei einer Änderung von ±10°C oder maximal nach 60 Minuten. Wie aus der Abbildung 3.5, rechts mittig, ganz klar zu sehen ist, wird am häufigsten nach 60 Minuten gesendet, das heißt die Temperatur ist relativ konstant – häufige und große Temperaturschwankungen sind daher unwahrscheinlich.

Bewegungssensoren senden sofort bei Bewegung, sind dann aber 5 Sekunden lang blockiert. Im Falle einer Bewegung wird daher oft in kurzen Abständen gesendet (festzustellen an den besonders häufigen Intervalllängen um die 6 Sekunden).

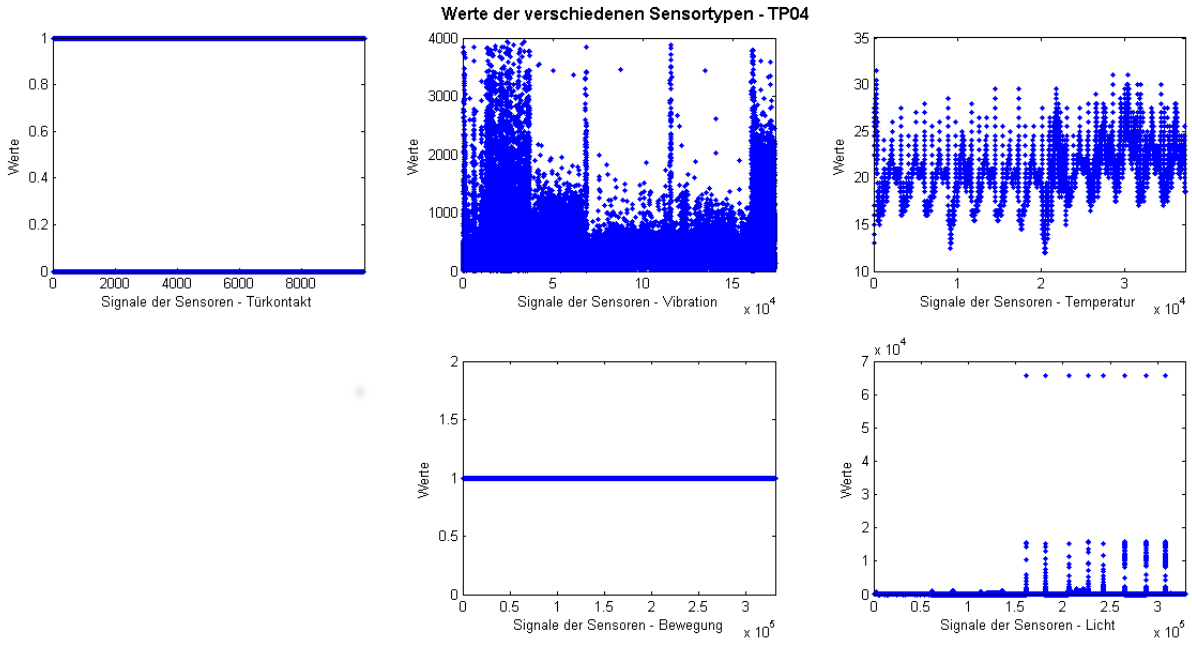


Abbildung 3.3: Werte der Sensortypen; TP04

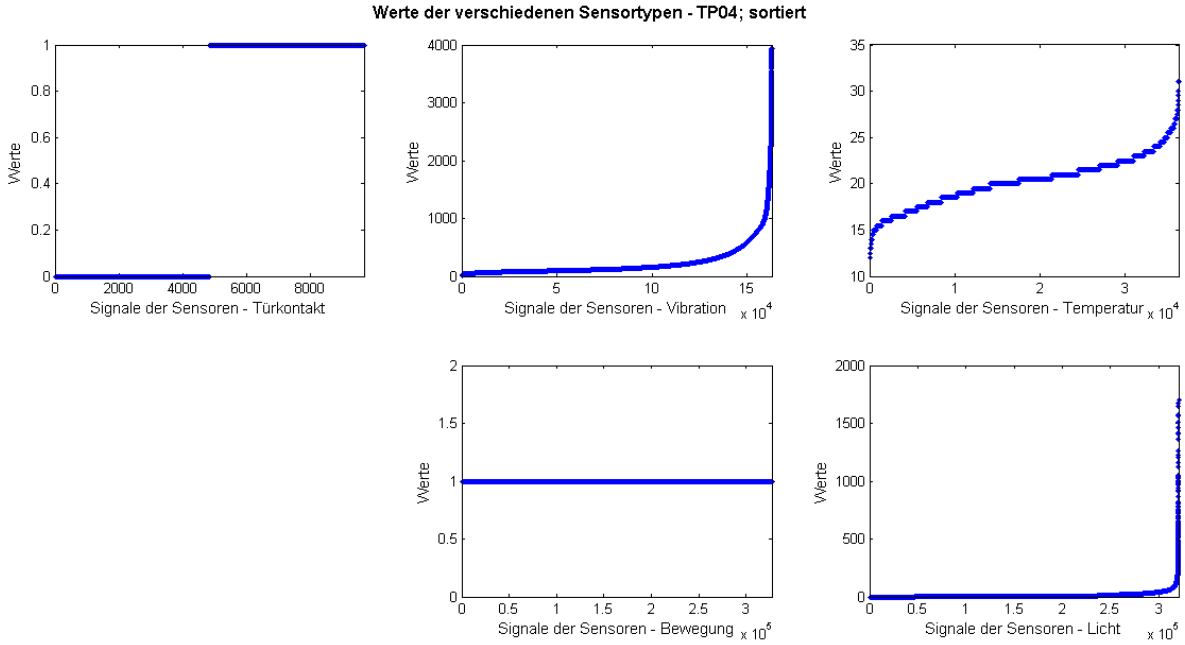


Abbildung 3.4: Werte der Sensortypen, Ausreißer entfernt, sortiert; TP04

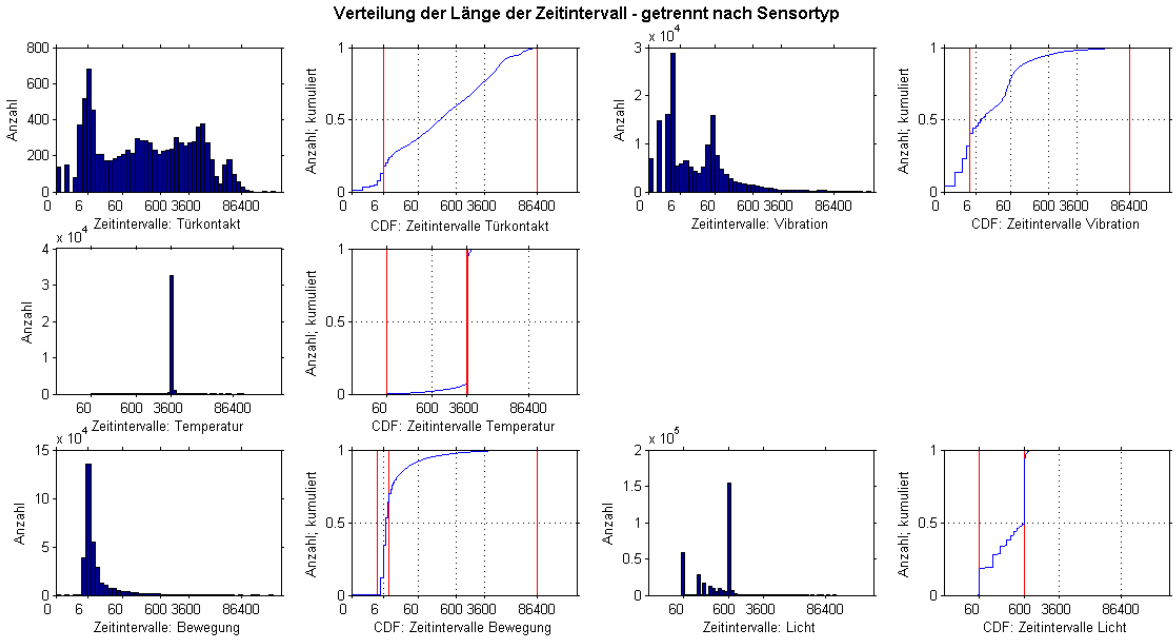


Abbildung 3.5: Verteilung der Sendeintervalle der Sensortypen; TP04

Bei den Lichtsensoren ist das minimale Sendeintervall 1 Minute, gesendet wird bei einer Änderung der Lichtstärke von ± 100 Lux oder ± 20 % oder sonst nach längstens 10 Minuten. Diese Intervalllängen sind deutlich erkennbar in der Abbildung 3.5, rechts unten. Am häufigsten tritt wieder das maximale Sendeintervall von 10 Minuten auf, gefolgt vom minimalen Sendeintervall.

Auch diese Ergebnisse sind in Tabelle 3.1 dargestellt. Zusätzlich ist angegeben, welche Zeitintervalle besonders häufig vorkommen (Ausnahmeklassen), wieviel Prozent der Gesamtzahl aller Intervalle eines Sensors in diese häufigen Zeitbereiche fallen, sowie eine Einteilung der verbleibenden Intervalle - d.h. jener, die nicht in eine dieser Ausnahmeklassen fallen - in ”Zu Kleine” (aufgrund des minimalem Sende-Intervalls), ”Zu Große” und ”Normale”.

3.1.5 Verteilung der Signale über 24 Stunden

Im nächsten Schritt wurde die Verteilung der Signale über einen Zeitraum von 24 Stunden untersucht. In Abbildung 3.6 sind die entsprechenden Histogramme einiger ausgewählter Sensoren dargestellt.
Ausgehend von den empirischen kumulierten Verteilungsfunktionen wurde die theoretische Verteilung der Signale für jeden Sensor berechnet (vgl 2.2.2 – Theoretische Verteilung bestimmen). Nicht in jedem Fall konnte mit dieser Methode ein positives Ergebnis erreicht werden. Gibt es beispielsweise zu große Sprünge in der Verteilung, das heißt in kurzer zeitlicher Abfolge gibt es große Unterschiede in der Anzahl der gesendeten Signale, so kann dies dazu führen, dass die so berechnete theoretische Verteilungsfunktion an einigen Stellen eine negative Steigung aufweist. Es ist daher notwendig das Verfahren an diese Probleme anzupassen.
In Abbildung 3.6 sind aber auch einige Beispiele zu sehen, bei denen mit obiger Methode ein

positives Ergebnis erreicht werden konnte - die entsprechende geschätzte theoretische Verteilung ist in diesem Fall im Histogramm eingezeichnet; mit einem Faktor 2/3 zur besseren Sichtbarkeit.

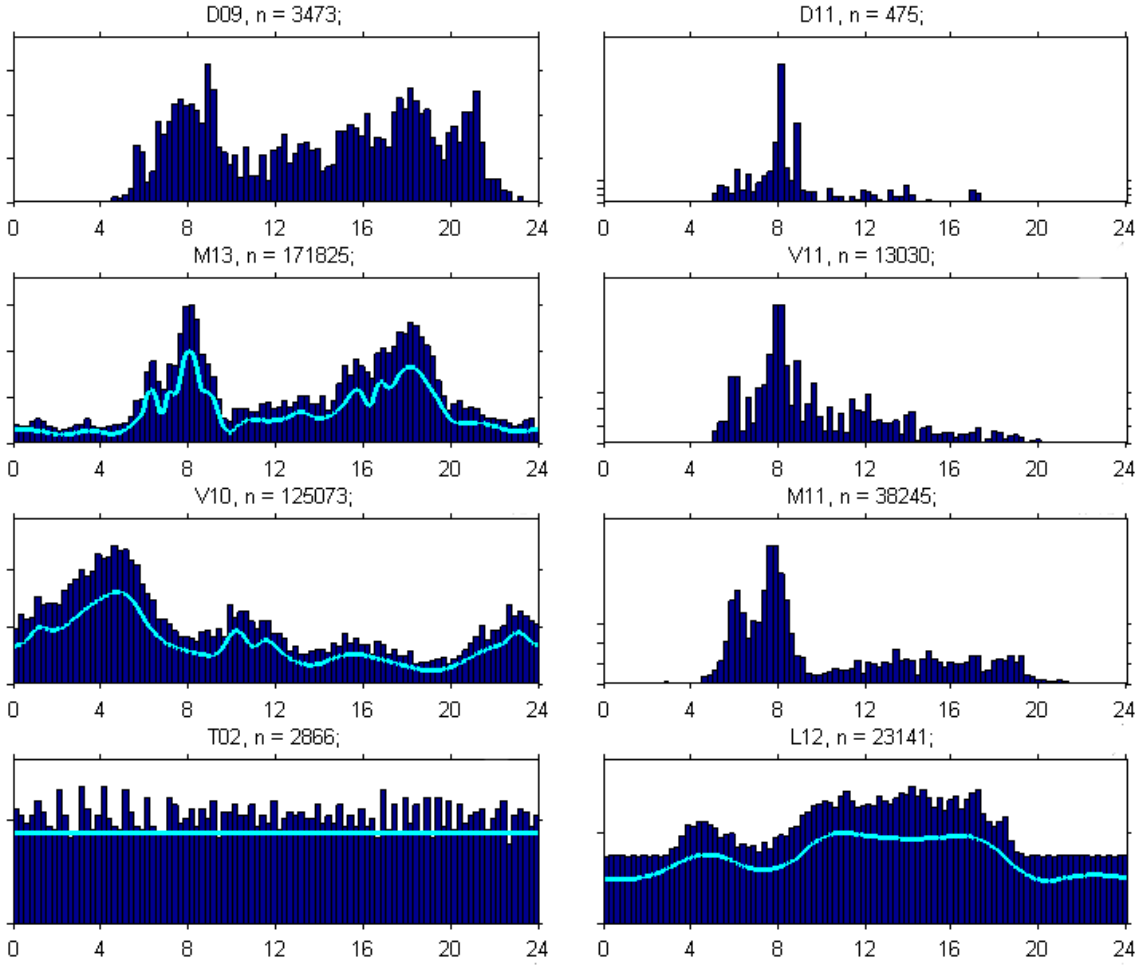


Abbildung 3.6: Verteilung der Signale über 24 Stunden; Abgebildet ist die Verteilung der Signale einiger ausgewählter Sensoren, sowie die berechnete theoretische Verteilung (2/3 des tatsächlichen Wertes zur besseren Sichtbarkeit).

Beobachtungen:

3.1.1 Beobachtungen.

Die Sensoren im Eingangsbereich (D09,D13,V02,M09,M13) zeigen eine sehr ähnliche Verteilung mit Häufungen morgens und am frühen Abend, wohl erklärbar durch die regelmäßigen Besuche der Betreuungsperson (D09 und D13 sind in 3.6 links oben abgebildet).

Der Bett-Vibrationssensor V10 (der zweite von links unten in 3.6) ist rund um die Uhr aktiv. Er weißt Häufungen um 6:00, 11:00, 16:00 und 24:00 Uhr auf. Er besitzt mit Abstand die meisten Signale und laut Beschreibung handelt es sich hier tatsächlich um den bevorzugten Aufenthaltsort der Testperson. Dieser Sensor unterscheidet sich sehr von den anderen Vibrationssensoren, welche

meist nur unter Tags und zu bestimmten Zeiten viele Signale senden (vgl. zum Beispiel V11 in Abbildung 3.6)

Einige Sensoren zeigen deutliche Häufungen am Morgen, bis ca 11:00 Uhr. Den restlichen Tag über gibt es weniger Signale (meist kaum Signale um die Mittagszeit, und leichte Häufungen gegen 16:00). Bei diesen Sensoren handelt es sich hauptsächlich um Vibrationssensoren und zwar jene im hinteren Teil der Wohnung (Gang und Küche) der demnach nur zu bestimmten Zeiten häufig benutzt wird.

Die Mikrowelle wird hauptsächlich morgens gegen 8:00 Uhr benutzt (vgl. D11 in 3.6 rechts oben).

Bei den Lichtsensoren zeigt L02 ein auffälliges Verhalten (auch in der Nacht sehr viele Signale), welches sich wohl durch ein Nachtlicht erklären lässt, das laut Beschreibung installiert ist. Außerdem zeigen einige der Lichtsensoren Häufungen zwischen 4:00 und 8:00 Uhr und zwar im wesentlichen jene, die am Weg vom Bett zum WC liegen (vgl. L12 in 3.6 rechts unten).

Mit einer Ausnahme (T11) folgen alle Temperatursensoren einer gleichmäßigen Verteilung (vgl 3.6; T02 links unten).

Grundsätzlich ist zu bemerken, dass örtlich nahe beieinander liegende Sensoren oft auch eine ähnliche Verteilung der Signale aufweisen (vgl. D11, V11 und M11 in 3.6).

3.1.6 Vervollständigen der Daten zu äquidistante Zeitreihen

Bei vielen numerischen Methoden wird mit **Datenmatrizen** gearbeitet. Jede Spalte einer Datenmatrix stellt ein **Merkmal** (auch Variable) dar, jede Zeile einen **Datenpunkt** (auch Beobachtung oder Merkmalsvektor). Um die gegebenen Daten in Matrixform darstellen zu können, wurden die durch die Sensormessungen gegebenen Zeitreihen im 10 Sekunden Abstand vervollständigt, wobei für die verschiedenen Sensortypen unterschiedlich vorgegangen wurde:

Türkontaktsensoren: Hier wurde im wesentlichen der Vorgängerwert beibehalten. 0 bedeutet, dass die Türe geschlossen ist und 1, dass sie geöffnet ist. Die Zeitreihen wurden begonnen mit dem Wert 0. Solange keine Signale aufgezeichnet waren, wurde der Wert beibehalten. Wenn es innerhalb von 10 Sekunden ein Signal gegeben hat, wurde dieser Wert geändert; es wurde in diesem Fenster der Wert 1 eingetragen. Fortgesetzt wurde dann aber mit dem letzten Wert innerhalb dieses Zeitfensters, der wieder Null sein konnte (innerhalb von 10 Sekunden können mehrmals Signale gesendet werden, es kann die Türe kurz geöffnet, dann aber sofort wieder geschlossen werden, wodurch dann zwar eine Eins im jeweiligen Fenster eingetragen wird, fortgesetzt wird dann aber wieder mit 0).
Bei einem aktuellen Wert von 1 wurde analog vorgegangen.

Vibrationssensoren: Auch diese Sensoren können mehrmals in der Sekunde Signale senden. Verschiedene Möglichkeiten Zeitreihen zu erstellen sind zum Beispiel die Signale pro Zeitfenster zu zählen und auf die Werte zu verzichten. Das ist auch deswegen interessant, da häufige Signale

meist mit sehr hohen Werten einhergehen. Oder aber es wird der Mittelwert aller Signale eines Fensters, oder auch das Maximum, im jeweiligen Zeitfenster eingetragen.
Hier wurde zur Erstellung der Zeitreihen der Mittelwert verwendet.

Temperatursensoren: Hier wurden die vorhandenen Daten durch lineare Interpolation vervollständigt. Dadurch konnte für jeden Zeitpunkt der äquidistanten Zeitreihe ein Wert ermittelt und in das entsprechende Zeitfenster eintragen werden.

Bewegungsmelder: Signale sind immer nach 5 Sekunden, also zwei mal innerhalb des 10 Sekunden Zeitfensters möglich. Zuerst wurde die Anzahl der Signale pro Zeitfenster gezählt und anschließend wurde die Summe über das aktuelle Zeitfenster und die 50 Sekunden vor dem aktuellen Zeitfenster gebildet. Es werden also die Werte über eine Zeitraum von 60 Sekunden berücksichtigt, daher sind Werte zwischen 0 und 12 möglich.

Lichtsensoren: Die Lichtsensoren wurden genauso wie Temperatursensoren durch lineare Interpolation vervollständigt.

Visualisierung der vervollständigten Zeitreihen

Um einen visuellen Eindruck der Daten zu bekommen und um das Geschehen in der Wohnung visuell erfassen zu können, sind die vervollständigten Zeitreihen in einer Gesamtansichten dargestellt. Abbildung 3.7 zeigt zum Beispiel eine Testwoche von TP04. Um so detaillierter diese Zeitreihen betrachtet werden, um so mehr kann daraus abgelesen werden und es kann ein Eindruck davon gewonnen werde, was in diesem Zeitraum in der Wohnung passierte.

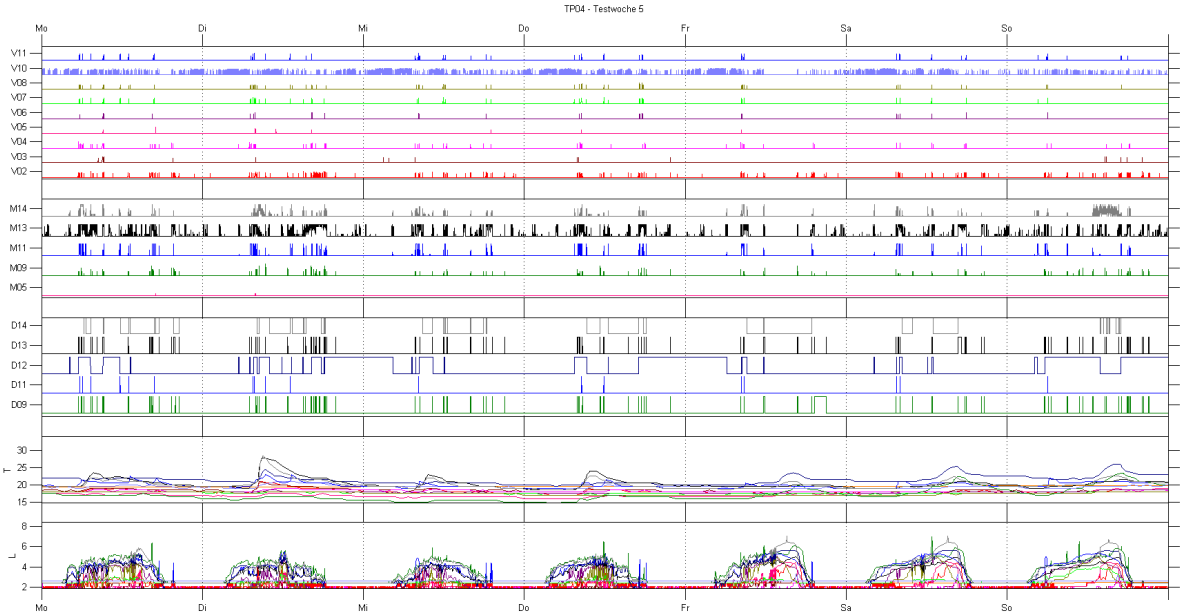


Abbildung 3.7: Äquidistante Zeitreihen, Gesamtansicht; TP04, Testwoche 5

Die resultierenden Datenmatrizen

Bei den Matrizen, die aus den beschriebenen vervollständigten Zeitreihen resultieren, liefert nun jede Spalte die Werte eines Sensors über den gesamten Testzeitraum. Jede Zeile liefert die Werte aller Sensoren zu einem bestimmten Zeitpunkt. Insgesamt wurde dadurch der Datensatz auf ca. 300 Millionen Einträge erweitert (Bei den Langzeit-Tests auf etwa 60 Millionen Einträge; bei den Kurzzeit-Tests auf etwa 15 Millionen Einträge). Um mit diesen Merkmalsdatensätzen effizient arbeiten zu können, war es notwendig, eine Datenreduktion durchzuführen. Eine häufig verwendete Methode dafür ist die zum Beispiel die Hauptkomponentenanalyse (2.2.3), aber auch mit der QR-Zerlegung können wichtige Aspekte hervorgehoben werden.

3.2 Datenreduktion

3.2.1 Hauptkomponentenanalyse

Für die Hauptkomponentenanalyse (vgl. Abschnitt 2.2.3) wurde die Singulärwertezerlegung $X = U\Sigma V^T$ der Datenmatrix X berechnet.
Die Merkmale von X waren die äquidistanten Sensorzeitreihen, die noch vorbearbeitet und ergänzt wurden:

- Bei den Temperatursensoren wurde die mittlere Temperatur aller Sensoren abgezogen (vgl. Abbildung 3.3, rechts oben – der Wochentrend ist sehr auffällig). Diese mittlere Temperatur kann als zusätzliches Merkmal in die Analyse miteinbezogen werden.
- Zusätzlich als zeitliche Komponente wurden zwei Merkmale hinzugefügt:

  $\mathrm{ST} = sin(Zeit * 2\pi/86400sec)$;

  $\mathrm{CT} = cos(Zeit * 2\pi/86400sec)$.

- Die Werte für Licht und Temperatur wurden logarithmiert.
- Alle Merkmale wurden zentriert.

Berechnet wurden die Komponentenmatrix V, die Eigenwerte der Kovarianzmatrix (die Diagonalelemente von Σ) und die Koordinaten bezüglich der neuen Achsen (U). Die Eigenwerte der Kovarianzmatrix sind dabei von großem Interesse, da sie den Anteil der Varianz der Komponenten an der Gesamtvarianz wiedergeben. Sie bilden die Grundlage für die Entscheidung der Anzahl der Hauptkomponenten – insgesamt werden dabei um die 95% der Gesamtvarianz angestrebt.
Tabelle 3.2 ist zu entnehmen, dass bereits mit 25 der insgesamt 48 Komponenten 95% der Gesamtvarianz erreicht sind. Diese 25 Komponenten bilden also die Hauptkomponenten.
In 2.2.2 wurde beschrieben, wie die Hauptkomponentenmatrix (die auf die Hauptkomponenten eingeschränkte Komponentenmatrix) transformiert werden kann, so dass die Merkmale ihrer Relevanz für die Hauptkomponenten nach geordnet sind. Diese transformierte Matrix ist in Tabelle 3.8) dargestellt.
Daraus resultiert dann für jede Hauptkomponente eine Menge von relevanten Sensoren, abhängig vom gewählten Grenzwert (vgl. wieder 2.2.2), welche in Tabelle 3.3) aufgelistet sind.

	% Var.	kum.
Komp. 1	27.64	27.64
Komp. 2	20.22	47.86
Komp. 3	5.98	53.84
Komp. 4	4.17	58.01
Komp. 5	3.46	61.47
Komp. 6	3.33	64.80
Komp. 7	2.88	67.68
Komp. 8	2.65	70.33
Komp. 9	2.29	72.62
Komp. 10	2.18	74.80
Komp. 11	2.04	76.84
Komp. 12	1.98	78.82
Komp. 13	1.76	80.59
Komp. 14	1.73	82.32
Komp. 15	1.70	84.02
Komp. 16	1.56	85.57
Komp. 17	1.52	87.09
Komp. 18	1.40	88.49
Komp. 19	1.34	89.83
Komp. 20	1.28	91.10
Komp. 21	1.25	92.36
Komp. 22	0.97	93.33
Komp. 23	0.86	94.19
Komp. 24	0.85	95.04
Komp. 25	0.79	95.83
Andere	< 0.74	100

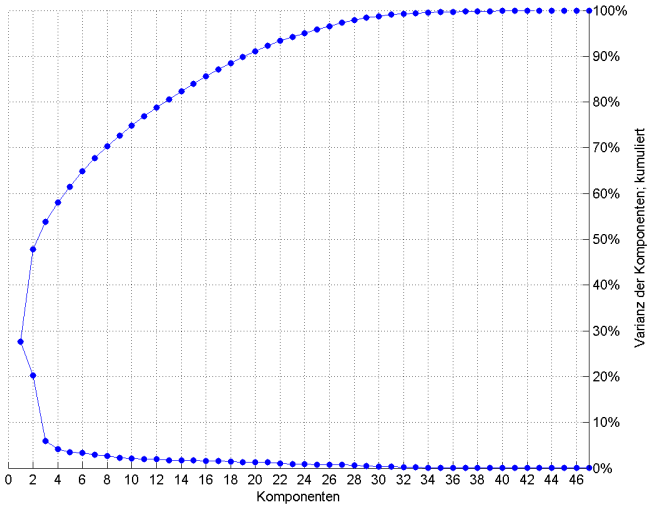


Tabelle 3.2: TP04 - Varianz der Komponenten; Links: Werte der ersten 25 Komponenten; diese decken insgesamt bereits 95 % der Gesamtvarianz ab; Rechts: Varianz aller Komponenten sowie die kumulierte Varianz.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
T10	**100**																								
L12		**100**																							
L13		**100**																							
M04			**100**		76		-62																		
D08				**100**				-69		-64															
M12					**100**										100										
ST						**100**	96																	75	
V02							**100**										-82								
L09					-77			**100**																	
V09							62		**100**			100													
D11										**100**											94				
V04									-75	-97	**100**			-76											
M13					72								**100**	84											
V05			69										-83	**100**						65					
L05		**99**																							
D10																**100**					100				
V03							68		-71								**100**								
L03								70										**100**				-88			
V01					81		72												**100**						
V10			85																	**100**					
L08		**99**																							
L01																						**100**			
T12	89																						**100**	100	
T05	**98**																								
V06			97		-82																				**100**
T07	**98**																								
L10		**98**																							
CT		-90																					-73		
L04	63	70																						-95	
L02		85																							
L07		91																							
D12				95				-63																	
T01	**97**																								
T04	**97**																								
T06	**97**																								
L11		**97**																							
T08	95																						-80		
V07			**97**		-80															-72					-91
D13										**97**						70					-74			-61	
T11	**96**																								
T13	96																							72	
T09	**95**																								
T02	94																								
T03	94																								
M10			**91**				-63									-74									
L06		86																				-66			
M08					65				65										-83						

Abbildung 3.8: Hauptkomponentenmatrix, Spalten durch betragsgrößtes Element dividiert; Werte mal 100; gerundet; sortiert nach Relevanz; Einträge < 60 nicht abgebildet; Abgebildet sind die ersten 25 Komponenten

	Relevante Senoren
Komp. 1	**T10**,**T05**,**T07**,**T01**,**T04**,**T06**,**T11**,T13,T08,**T09**,T02,T03,T12,L04,
Komp. 2	**L12**,**L13**,**L05**,**L08**,**L10**,**L11**,L07,CT,L06,L02,L04,
Komp. 3	**M04**,V06,**V07**,**M10**,V10,V05,
Komp. 4	**D08**,D12,
Komp. 5	**M12**,V06,V01,V07,L09,M04,M13,M08,
Komp. 6	**ST**,
Komp. 7	**V02**,ST,V01,V03,M10,M04,V09,
Komp. 8	**L09**,L03,D08,D12,
Komp. 9	**V09**,V04,V03,M08,
Komp. 10	**D11**,V04,**D13**,D08,
Komp. 11	**V04**,
Komp. 12	V09,
Komp. 13	**M13**,V05,
Komp. 14	**V05**,M13,V04,
Komp. 15	M12,
Komp. 16	**D10**,M10,D13,
Komp. 17	**V03**,V02,
Komp. 18	**L03**,
Komp. 19	**V01**,M08,
Komp. 20	**V10**,V07,V05,
Komp. 21	D10,D11,D13,
Komp. 22	**L01**,L03,L06,
Komp. 23	**T12**,T08,CT,
Komp. 24	T12,L04,ST,T13,D13,
Komp. 25	**V06**,V07,

Tabelle 3.3: Zusammenfassung der relevanten Sensoren der Hauptkomponenten; die fett gedruckten Sensoren sind jene, die bei der Sortierung nach Relevanz (vgl. 2.2.2) ausgewählt wurden.

3.2.1 Beobachtungen.

Für die erst Komponente sind die Temperatursensoren wesentlich (ein Verhalten, das für alle Wohnungen typisches ist).

Für die zweite Komponente dagegen sind die Lichtsensoren von großer Relevanz, sowie das zusätzliche Merkmal CT, welches dem natürlichem Verlauf des Lichts ähnelt (auch das gilt meist für alle Wohnungen).

Die dritte Komponente wird bestimmt durch die Bewegungs- und Vibrationssensoren in der Küche, sowie dem einen Bewegungssensor im Gang am Weg zur Küche.

Für die vierte Komponente sind zwei Türkontaktsensoren relevant – die der Eingangs- und der Vorzimmertür, die beide geöffnet werden, wenn die Wohnung betreten beziehungsweise verlassen wird.

Auf diese Weise ist es möglich den Hauptkomponenten eine Bedeutung zuzuweisen, wie zum Beispiel Aktivität in einem bestimmten Bereich der Wohnung.

Zusätzlich zeigt Tabelle 3.9 eine Möglichkeit, eine Übersicht über die gesamte Komponentenmatrix zu erhalten. Anstatt der Werte geben Symbole Auskunft über die Relevanz der Sensoren. So bedeutet □, dass das entsprechende Merkmal für diese Komponente relevant ist (d.h. größer als die gewählte Grenze). Mit ■ sind jene Einträge gekennzeichnet, die im Sortieralgoritmus (2.2.3) ausgewählt wurden.

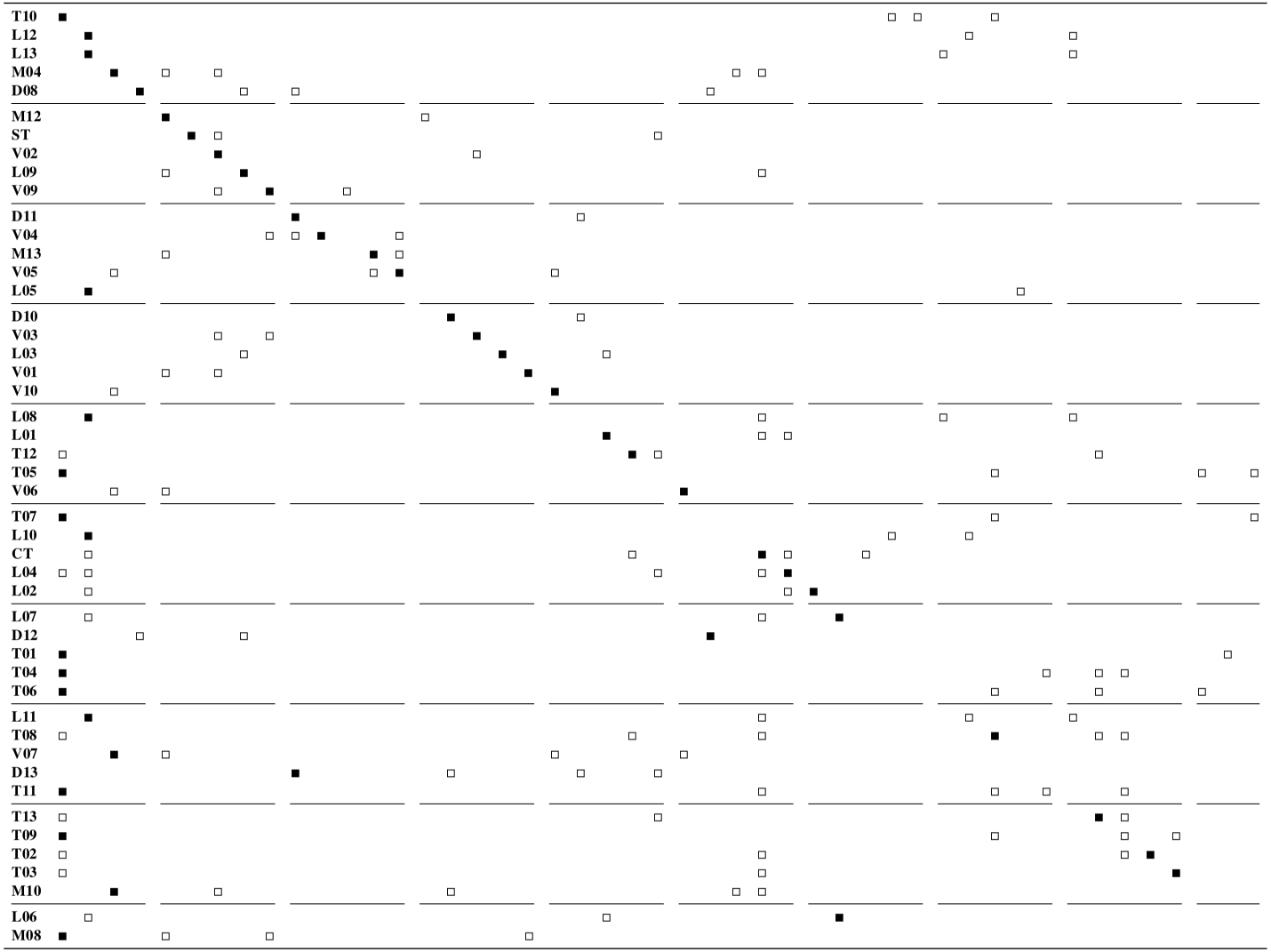
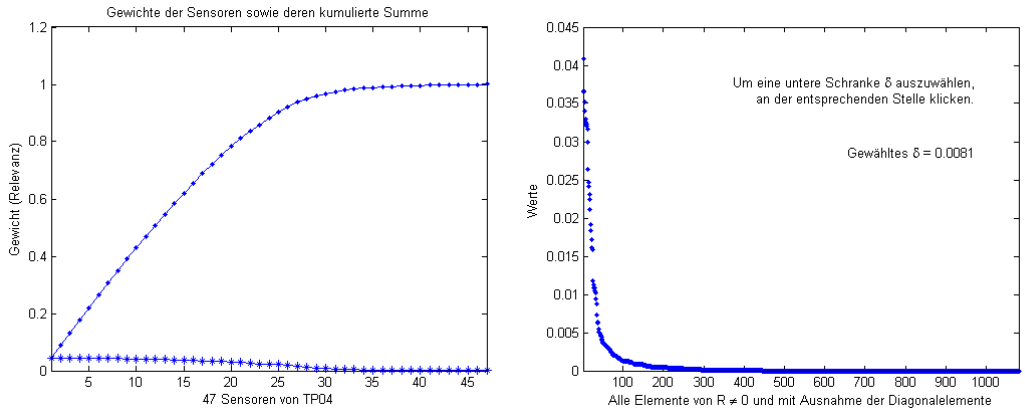


Abbildung 3.9: Hauptkomponentenmatrix wie oben, hier sind alle Werte ≥ 60 durch ein □ markiert; alle Werte, die im Sortieralgoritmus ausgewählt wurden sind durch ■ gekennzeichnet; es sind alle Komponenten abgebildet

3.2.2 QR-Faktorisierung

Eine andere Methode zur Untersuchung der Merkmale stellt die dar (vgl. Abschnitt 2.2.4). Auch mit Hilfe dieser Zerlegung ist es möglich, die Daten zu reduzieren, wobei hier ein anderer Aspekt hervorgehoben wird: Welche Merkmale besitzen hohen Informationsgehalt? Welche Merkmale enthalten redundante Information und können daher ohne größeren Informationsverlust weggelassen werden?
Das Verfahren basiert auf der Zerlegung $X = QR$ der Datenmatrix X. Dabei ist Q eine orthogonale Matrix und R eine rechte obere Dreiecksmatrix. Für die Analyse von Bedeutung ist die obere Dreiecksmatrix R. Die Diagonalelemente von R geben das Gewicht der einzelnen Merkmale wieder, welche aufgrund dieser Gewichte bezüglich ihrer Relevanz bzw. ihres Informationsgehaltes absteigend geordnet werden können.
Nach dem Berechnen der QR-Zerlegung wurde R, wie in 2.2.4 beschrieben, umsortiert und daraufhin wurden die Diagonalelemente untersucht:
In Abbildung 3.10a ist das Gewicht der einzelnen Sensoren, sowie die kumulierte Summe abgebildet. Anhand so einer Darstellung bzw. aufgrund der Gewichte kann entschieden werden, auf wieviele Sensoren die Daten reduziert werden sollen. Wie bei der Hauptkomponentenanalyse kann sich die Entscheidung danach richten, dass insgesamt 95% Gesamtgewicht erreicht werden.
In nächsten Schritt wurden die anderen Elemente von R untersucht:
Diese geben Auskunft über gewisse Abhängigkeiten (redundante Information) zwischen den Sensoren. So kann ein betragsmäßig großer Wert als sehr abhängig und ein betragsmäßig kleiner Wert als wenig abhängig interpretiert werden. In Abbildung 3.10b sind diese Werte sortiert dargestellt.



(a) QR-Zerlegung: Diagonalelemente von R sowie die kumulierte Summe;

(b) QR-Zerlegung: alle Werte von R sortiert (ausgenommen Diagonalelemente);

Nachdem eine gewünschte untere Schranke δ ausgewählt wurde (δ definiert dabei den Richtwert, nach dem zwei Sensoren als voneinander abhängig oder nicht voneinander abhängig eingestuft werden und kann hier nach eigenem Ermessen gewählt werden), können die Ergebnisse der QR-Zerlegung in einer Tabelle zusammengefasst werden – siehe dazu Tabelle 3.4.
Die erste Spalte enthält die Sensoren nach Relevanz beziehungsweise nach Informationsgehalt absteigend sortiert – die jeweiligen Gewichte sind der zweiten Spalte zu entnehmen. In der dritten Spalte sind jene Sensoren angeführt, die aufgrund der Wahl von δ als abhängig eingestuft wurden

	Gewicht %	abhängige Sensoren
V10	4.41	
D12	4.41	
T14	4.41	L01; D11; T09; T04; T11; T13; T12; T05; T08; T10; T03; T07; T02; T06;
V11	4.41	V07;
V02	4.38	
L07	4.35	L04; L12; L02; L10; L05; L08; L03; L11; L09; L06; L14; L13;
M13	4.26	
V05	4.22	
M11	4.15	M05;
V04	3.94	
M14	3.92	
D09	3.91	
D14	3.89	
M09	3.63	
T01	3.60	
V06	3.59	
L01	3.36	
D13	3.33	
V03	3.20	
V08	2.91	
L04	2.85	
D11	2.54	
L12	2.28	L05; L11; L09; L06; L14; L13;
M05	2.26	
V07	2.22	
L02	1.79	
L10	1.65	
L05	1.09	
T09	0.93	
Andere	< 0.7	

Tabelle 3.4: QR-Zerlegung: Relevante Sensoren von insgesamt 51 Sensoren; sortiert nach Relevanz; neben dem Gewicht der einzelnen Sensoren sind zusätzlich jene Sensoren angeführt, die eine große Abhängigkeit zu ihnen aufweisen; TP04

(d.h. je nachdem wie δ gewählt wurde sind hier mehr oder weniger Sensoren eingetragen).

3.2.2 Beobachtungen.

Ganz typisch ist jeweils ein Temperatursensor mit sehr hohem Gewicht; alle anderen Temperatursensoren haben dagegen nur ein sehr kleines Gewicht, zeigen dafür aber eine hohe Abhängigkeit zu diesem einen relevanten Temperatursensor.

Ähnlich verhält es sich mit dem Licht, wenn auch nicht ganz so eindeutig - es gibt zwar immer eine Gruppe von Lichtsensoren, die offenbar sehr ähnliche Information beinhalten, aber immer auch einige, die davon abweichen und selbst ein großes Gewicht besitzen.

Anders verhält es sich mit dem Sensoren für Bewegung, Vibration und Türkontakt. Hier sind im allgemeinen wenige Abhängigkeiten zu finden. Wenn doch, so sind diese meistens durch örtliche Zusammenhänge begründet.

Es ist auffällig, dass all diese Sensoren meist ein sehr ähnliches Gewicht aufweisen, d.h. dass die Reihenfolge in der sie hier auftreten eher zufällig erscheint und sie im Wesentlichen gleich relevant sind.

Es gibt einen wesentlichen Unterschied zwischen den beiden oben beschriebenen Methoden. Durch eine Hauptkomponentenanalyse wird eine Datenreduktion durch eine Neuausrichtung der Koordinatenachsen erreicht. Die reduzierten Daten sind dann jene, die durch Transformation und Reduktion der ursprünglichen gesamten Datenmatrix erhalten werden – stellen also Linearkombinationen der ursprünglichen Daten dar. Das heißt aber auch, dass auf diese ursprünglichen Daten nicht verzichtet werden kann und sie für die Reduktion wesentlich sind.
Dahingegen bestehen die reduzierten Daten, die mittels QR-Zerlegung bestimmt werden, aus einer Auswahl der ursprünglichen Merkmale. Jene Merkmale, die aufgrund ihrer Relevanz ausgemustert werden, werden nicht in die reduzierten Daten miteinbezogen.
Bezüglich der Sensoren bedeutet das also, dass mit Hilfe der QR-Zerlegung auch untersucht werden kann, welche Sensoren des gesamten Netzwerkes besonders hohen Informationsgehalt haben beziehungsweise auf welche eventuell verzichtet werden könnte.

3.3 Berechnung von Aktivitätsklassen

Im nächsten Schritt wurden die Zeitfenster der reduzierten Datensätze untersucht. Ziel war es, die Datenpunkte mittels geeigneter Algorithmen so in unterschiedliche Aktionsklassen zu unterteilen, dass die resultierenden Klassen eine Interpretation über die verschiedenen Aktionen der Testperson zulassen. Die Zuordnung der Datenpunkte zu den verschiedenen Klassen sollte automatisch und alleine aus der Struktur der Merkmalsdatensätzte heraus durchgeführt werden.
Es gibt in der Clusteranalyse unterschiedliche Ansätzte. Die Hierarchische Clusteranalyse beispielsweise bestimmt die Klassenzugehörigkeit aufgrund der Distanzen zwischen den Datenpunkten. Handelt es sich aber wie hier um eine Datenmatrix mit über einer Million Datenpunkten, so

ist die Distanzmatrix, in welcher die Distanzen zwischen allen Datenpunkten eingetragen sind, zu groß, um damit effizient zu arbeiten.
Möglich ist aber beispielsweise eine vorausgehende Quantisierung der Datenpunkte, zum Beispiel durch ein prototypbasiertes Verfahren wie K-Means und darauf folgend die Anwendung eines Hierarchischen Verfahrens auf den Prototypen.

3.3.1 K-Means und Hierarchische Clusteranalyse

Der Merkmalsdatensatz wurde reduziert (vgl. Abschnitt 3.2.1) und anschließend wurde ein Clustering der Datenpunkte durch eine Kombination von K-Means und Hierarchischer Clusteranalyse durchgeführt. In Abbildung 3.10 können verschiedene Schritte visuell nachvollzogen werden:

1. Schritt *Die Anzahl der Beobachtungen mittels K-Means (vgl. 2.3.1) auf wenige Prototypen reduzieren:*

Im gezeigten Beispiel wurden die Datenpunkte auf 100 Prototypen reduziert. In Abbildung 3.10 ist rechts oben zu sehen, wie sich die Datenpunkte auf die verschiedenen Prototypen aufteilen; rot gekennzeichnet sind Prototypen mit einer Anzahl von weniger als 1% der Anzahl aller Datenpunkte.

An diese Stelle könnten diese ”kleinen Klassen” zu einer Restklasse zusammengeführt werden, als neues Zentrum (der Prototyp dieser Klasse) wird dann das Mittel aller Datenpunkte dieser Restklasse berechnet (wurde hier nicht durchgeführt; solche Klassen müssen mit Vorsicht behandelt werden, da sie in der Regel eine deutlich größere Streuung aufweisen als die restlichen Klassen).

2. Schritt *Reduktion der Prototypen auf wenige, vielleicht 20 bis 30 Cluster mittels Hierarchischer Clusteranalyse (vgl. zum Beispiel [1]); Zuordnung der Datenpunkte zu den entsprechenden Clustern:*

Um über die Anzahl der Cluster zu entscheiden wird das linke zweite Bild von oben in Abbildung 3.10 herangezogen. Es zeigt die relativen Kosten einer Einteilung der 100 Prototypen in verschieden viele Klassen (von links nach rechts die Kosten für zwei, drei, vier, usw. Klassen). Ab einer gewissen Anzahl von Klassen sind - etwas vereinfacht gesprochen - die Unterschiede zwischen den Beobachtungen bereits so klein, dass eine weitere Unterteilung als nicht mehr sinnvoll erachtet werden kann.

Hier wurden jene relativen Kosten rot markiert, die kleiner gleich 1 sind, was im gezeigten Beispiel dann der Fall ist, wenn in mehr als 33 Klassen geteilt wird. In der nächsten Abbildung direkt darunter ist dann zu sehen, wie sich die Datenpunkte auf die berechneten 33 Klassen aufteilen.

3.Schritt *Erstellen einer **Farbcodierung** der Clusterzentren und deren Darstellung:*

Dazu wurde für jede der 33 neuen Klassen das jeweilige Zentrum berechnet. Diese wurden dargestellt, indem die ersten drei Koordinaten jeweils auf das Intervall [0, 1] skaliert wurden. Wie sich diese Zentren verteilen kann in den vier Abbildungen rechts in 3.10 beobachtet werden, wo zusätzlich 10 000 zufällig ausgewählte Datenpunkte mit abgebildet sind.

Daneben lieferten die skalierten Koordinaten außerdem direkt eine Farbcodierung der Zen-

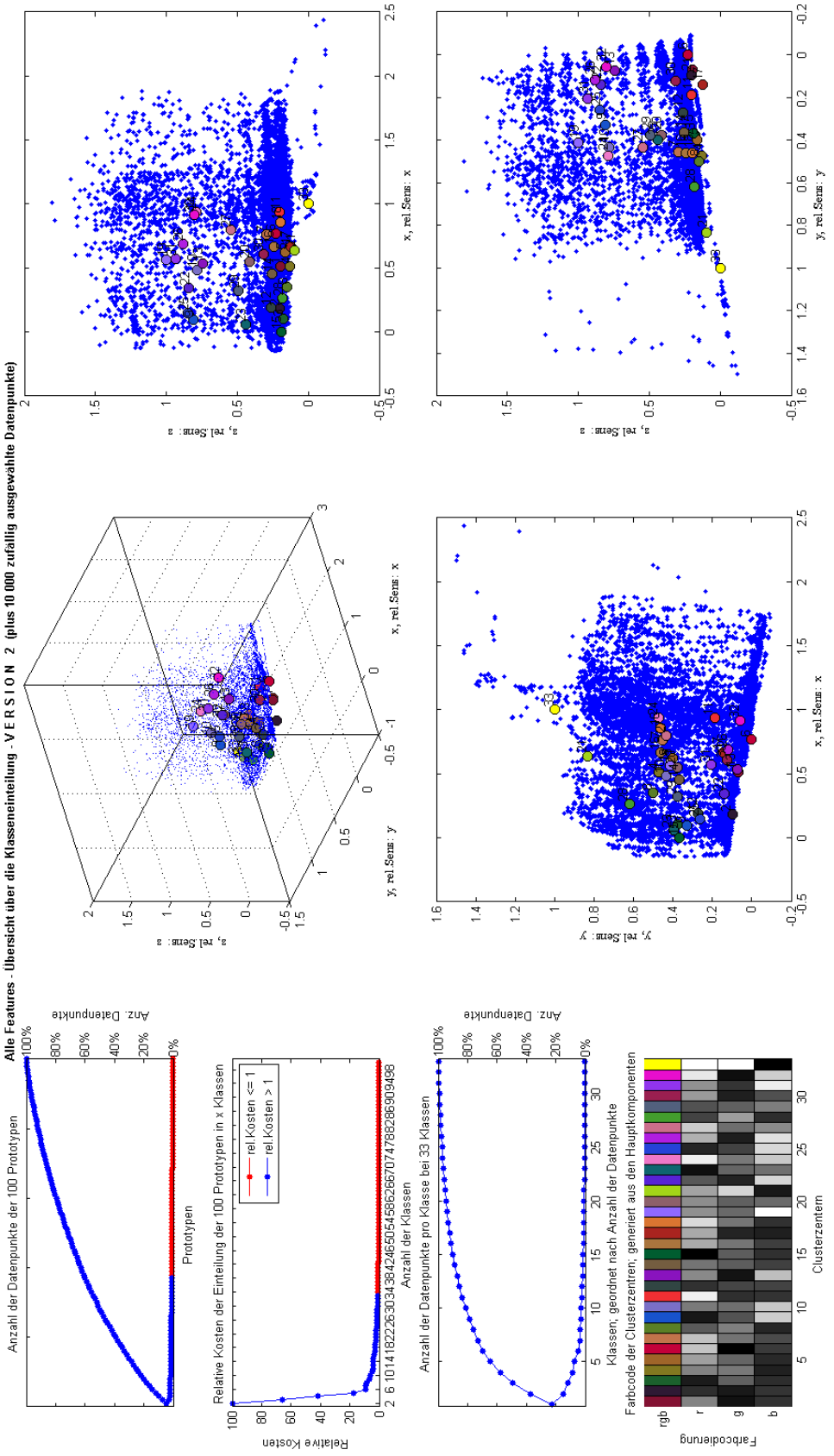


Abbildung 3.10: Übersicht über die Klasseneinteilung der Zeitfenster; TP04

tren dadurch, dass die Koordinaten als Rot-, Grün- und Blauanteil im RGB-Farbraum aufgefasst werdenden. Die resultierende Codierung wurde in den vier Abbildungen rechts verwendet und ist links unten in 3.10 dargestellt.

4.Schritt. *Resultierende Clusterzeitreihen berechnen. Diese visuell darstellen und auf mögliche Interpretationen hin untersuchen:*

Abbildung 3.11 zeigt als Beispiel einen Testtag (TP04; 10 Testwoche; Mittwoch) im Vergleich mit den Daten (vgl. Beobachtungen 3.3.1).

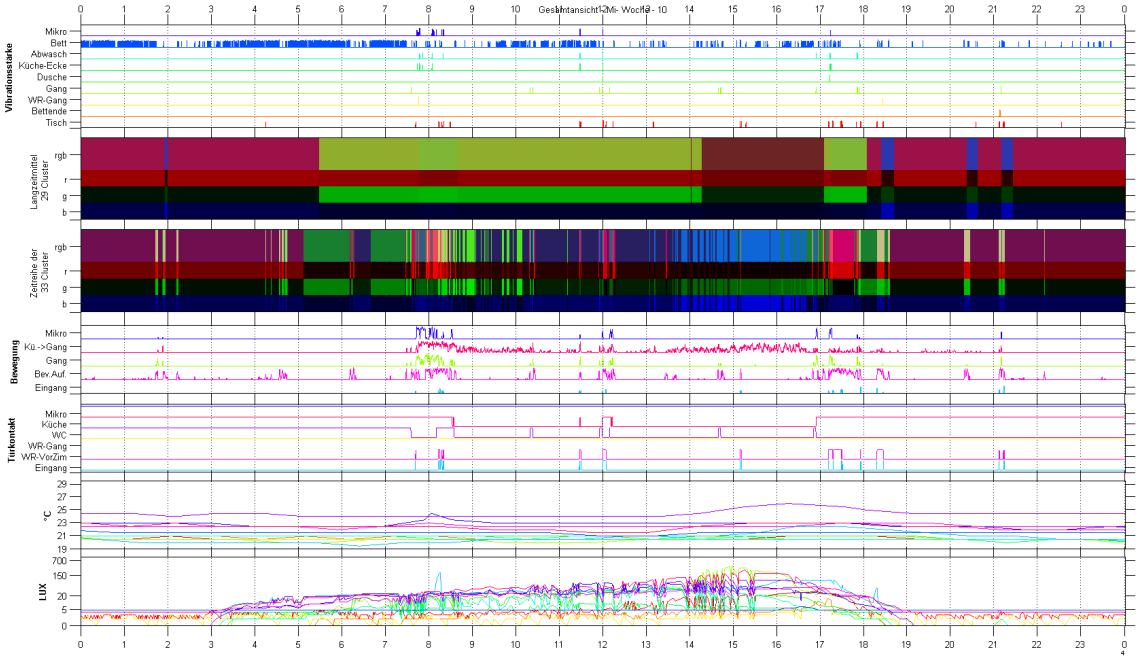


Abbildung 3.11: Vergleich der Daten mit den Klasseneinteilungen der 10 Sekunden-Zeitreihen und den Langzeitmittel-Zeitreihen; Mi, Testwoche 10;

In Abbildung 3.11 ist neben der aus der durchgeführten Einteilung in 33 Klassen resultierenden Cluster-Zeitreihe, eine weitere Cluster-Zeitreihe angegeben, die eine Einteilung der *Langzeitmittel–Zeitreihen* zeigt. Diese Langzeitmittel-Zeitreihen wurden aus den 10-Sekunden Zeitreihen berechnet, indem die Werte über einen längeren Zeitraum (hier 15 Minuten) gemittelt wurden. Danach wurden diese Zeitreihen ebenfalls, wie beschreiben klassifiziert. Wie zu sehen ist, führt das zu einem anderen Ergebnis.

3.3.1 Beobachtungen.

Die Einteilung der 15 Minuten Langzeit Mittel (zweite Zeile von oben in 3.11) führt zu einer viel ruhigeren Cluster-Zeitreihe, als jene über die 10 Sekunden Mittel (dritte Zeile von oben in 3.11).

Auffällige Klassen der Langzeitmitten sind: Violett (relativ ruhiger Schlaf), blau (Unterbrechung des ruhigen Schlafes durch WC-Gang oder starke Bewegungen), olivgrün (relativ wenig Aktivität am Tag), hellgrün (viel Aktivität in der ganzen Wohnung), braun-violett (viel Aktivität im Bett).

Auffällige Klassen der 10 Sekunden Zeitreihen sind: violett (ruhiger Schlaf), beige (nächtliche Aktivität wie WC-Gang oder starke Bewegungen), grün (Dämmerung aber immer noch Schlaf) und

verschiedene Klassen für mehr oder weniger viel Aktivität unter Tags wie zum Beispiel hellblau (viel Aktivität im Bett), dunkelblau (viel Aktivität in der Wohnung).

Es lässt sich also erkennen, dass auf diese Weise durchaus aussagekräftige Klasseneinteilungen berechnet werden können. Allerdings erfordert die Berechnung die Durchführung einer Reihe verschiedener Schritte und einige interaktive Entscheidungen zum Beispiel über die Klassenanzahlen.

3.3.2 Clustering mittels EM-Algorithmus für Gaußsche Mischungen

Ein anderer Algorithmus zur Klassifikation der Zeitfenster ist der EM-Algorithmus (*Expactation Maximization*-Algorithmus) für Gaußsche Mixturen. Angewandt wurde der Algorithmus wieder auf den reduzierten Merkmalsdatensatz aller Sensor-Zeitreihen.
Abbildung 3.12 zeigt wieder eine Beispiel-Testwoche. In der dritten Zeile ist die aus dem EM-Algorithmus resultierende Zeitreihe abgebildet (10 Cluster). In der Zeile direkt darüber kann beobachtet werden, wie sich K-Means verhält, wenn er direkt angewandt wird, das heißt, wenn direkt 10 Cluster berechnet werden und nicht, wie in 3.3.1, eine Kombination aus K-Means und Hierarchischem Clustering durchgeführt wird.

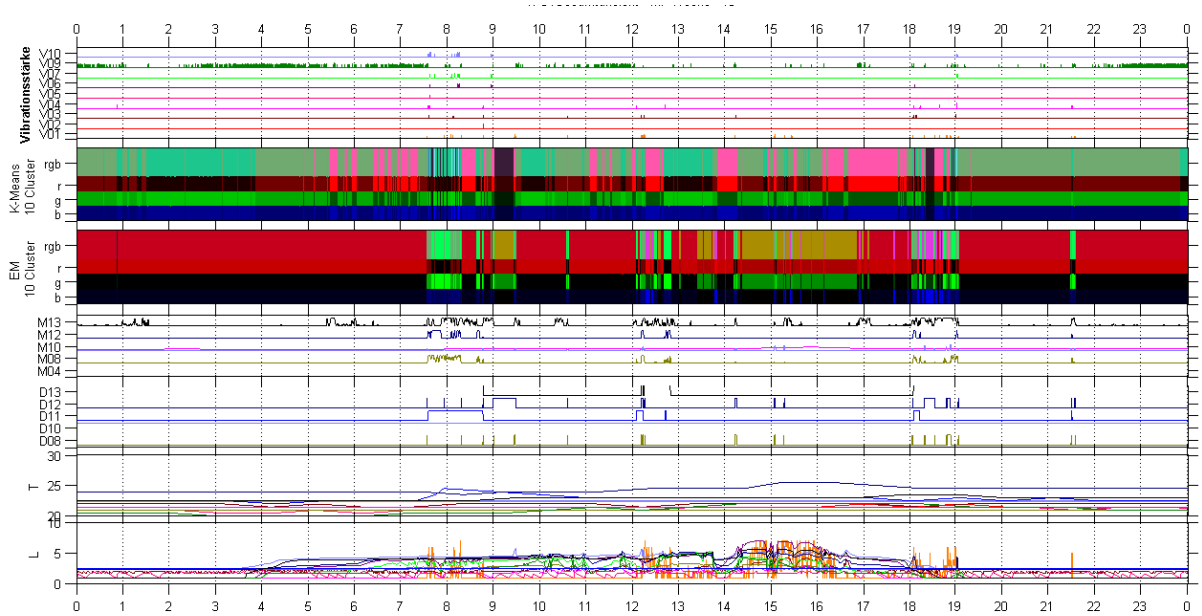


Abbildung 3.12: Vergleich: K-Means – EM-Algorithmus: Oben ist die Klasseneinteilung von K-Means; unten die von EM abgebildet; jeweils wurden die Daten in 10 Cluster aufgeteilt; Samstag, Testwoche 5;

3.3.2 Beobachtungen.

Auffällige Klassen beim EM-Clustering sind zum Beispiel: rot (Schlaf beziehungsweise wenig Aktivität), grün (häufig in Verbindung mit geöffneter WC-Tür), braun (wenig Aktivität aber viel Licht), die restlichen Klassen verteilen sich über die aktive Zeit.

Auffällig beim K-Means-Clustering ist, dass die Klassen sehr schlecht interpretierbar sind: hellgrün/grau (wenig Aktivität; es ist nicht nachvollziehbar wie diese unterschieden werden), rosa (nicht interpretierbar), usw.

Der nächtliche WC-Gang (ca. 21:30 Uhr) wird durch den EM-Algorithmus deutlich erkannt; K-Means identifiziert diesen aber nicht so eindeutig.

Insgesamt sind die Ergebnisse des EM-Algorithmus also doch deutlich besser, vor allem wesentlich leichter zu interpretieren.
Trotzdem ist aber zu bemerken, dass es sehr schwer ist, anhand dieser Klassifizierungen und Darstellungen genauere Aussagen zu treffen – beispielsweise über den genauen Aufenthalt oder eine bestimmte Aktivität. Es ist wohl notwendig, insgesamt etwas strukturierter vorzugehen und eventuell anstatt der gesamten Menge von Sensoren nur eine Auswahl zu verwenden, je nachdem welche Information erhalten werden soll.

4 Wohnungsmodell

4.1 Wünsche der Ingenieure

Nach Erfassung und statistischer Analyse der Rohdaten der einzelnen Testwohnungen wurde herausgearbeitet, welche Fragestellung für das *eHome* Projekt an sich von Interesse sind. Zum einen handelt es sich dabei um technische Fragen, beispielsweise Empfehlungen zur Anzahl der Sensoren oder zur Häufigkeit der Wertemessungen. Zum anderen sind Fragen bezüglich der Erkennung bestimmter Aktivitäten und Phasen im Tagesablauf der Testpersonen interessant.

Fragen aus technischer Sicht:
- Was kann erkannt werden und mit welcher Signifikanz?
- Welche Sensoren erscheinen besonders wichtig?
- Welche und wie viele Sensoren werden an welchen Orten benötigt um Aussagen treffen zu können - in Abhängigkeit von der Anzahl und der Art der interessierenden Objekte/Geräte.
- Welche Sensoren könnten (am ehesten) weggelassen werden – ohne relevanten Informationsverlust!
- Welche Genauigkeit ist nötig?
- Welche Auflösung (Granularität) der Daten ist nötig?
- In welchen zeitlichen Abständen werden die Daten benötigt?
- Was passiert, wenn die Voraussetzungen nicht erfüllt sind?
- Welche Algorithmen werden empfohlen?
- Nötige Standardisierung der Ausstattung?

Relevante Aktivitäten:

1) Aufstehen;
2) Benützung der Toilette;
3) Benützung von Waschmöglichkeit, Dusche oder Bad;
4) Zubereitung Frühstück;
5) Konsumation Frühstück;
6) Zubereitung Mittagessen;
7) Konsumation Mittagessen
8) Zubereitung Abendessen;
9) Konsumation Abendessen;
10) Zu Bett gehen;
11) Tief schlafen;
12) Unruhig schlafen;
13) Wach im Bett sein;
14) Bett in der Nacht verlassen ohne spezielles Ziel;
15) Bett in der Nacht verlassen um die Toilette aufzusuchen;
16) Bett in der Nacht verlassen um etwas essen zu gehen;
17) Zeit außergewöhnlich hoher Aktivität.

18) Verlassen der Wohnung;

19) Rückkehr in die Wohnung;
20) Besuch (d.h. Anwesenheit von mehr als einer Person);
21) Raum lüften;
22) Abwasch;
23) Jause;
24) Medikamenteneinnahme;
25) Ausruhen;
26) Buch lesen;
27) Fernsehen;
28) Telefonieren.

Es wurden von der *eHome* Forschungsgruppe bereits ein Regelwerk entwickelt (dieses bildet das *eHome* Expertensystem), mit dessen Hilfe es möglich ist einige relevante Situationen im Tagesablauf, wie ungewöhnlich lange Toilettenaufenthalte oder der Rückgang der Tagesaktivität (ein Indikator für eine Verschlechterung des gesundheitlichen Zustandes), zu erkennen. Im weiteren Verlauf dieser Arbeit soll untersucht werden, ob diese und die oben angeführten relevante Aktivitäten automatisch und wohnungsunabhängig erkannt werden können.

4.2 Abstraktion der Testwohnungen

Um auf allgemeine Fragen wie: *"Ist das automatische Erkennen bestimmter Aktivitäten möglich?"*, antworten zu können, ist es notwendig, Verfahren auszuwählen beziehungsweise zu entwickeln, die wohnungsunabhängig (ausgenommen geringfügiger Anpassungen für neue Wohnungen), also unabhängig sowohl von Grundriss und Ausstattung, als auch von den Gewohnheiten der Testpersonen, anwendbar sind.
Dazu wurde im ersten Schritt untersucht, auf welche Weise die verschiedenen Testwohnungen vergleichbar sind und welche Gemeinsamkeiten es bezüglich Topologie, Sensorausstattung, usw. gibt.
Aufgrund der oben angeführten Fragestellungen ergibt sich eine besondere Bedeutung bestimmter Räume und Objekte.
Folgende Räume sind von besonderem Interesse:

- Vorzimmer (V)
- Küche (K)
- Wohnzimmer (W)
- Schlafzimmer (S)
- Bad (B)
- Toilette (T)

Zusätzlich zu diesen Räumen sind außerdem vorgekommen:

- Extrazimmer (E1,E2) wie Bügelzimmer, nicht benutztes Kinderzimmer, usw. (maximal sind 2 solcher Zimmer pro Wohnung vorgekommen);
- Gang (G).

Außerdem sind folgende Objekte innerhalb der Wohnungen von besonderem Interesse:

- Herd (h) – Kochen;
- Kühlschrank (k) – Essenszubereitung; Trinken;
- Tisch (t) – Essen;
- Sofa (s) – Ausruhen; Lesen; Fernsehen; evtl. Schlafen;

- Bett (b) – Schlafen; Ausruhen.

Bezüglich Lüften:

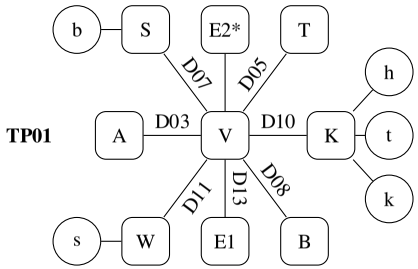
- Für jeden Raum die Temperatursensoren.

Darauf aufbauend wurde ein allgemeines abstrahiertes **Wohnungsmodell** entwickelt. Entsprechend der oben genannten Räume und Objekte wurde für jede Wohnung die Topologie in Form eines Graphen $\mathcal{W}$ erfasst. Die Knoten $\mathcal{N}$ bilden eine abstrakte Repräsentation der Räume (*eckige Knoten*) und der relevanten Objekte (*runde Knoten*) von $\mathcal{W}$. Der Außenbereich wird durch einen eigenen Knoten A repräsentiert (dadurch ist gekennzeichnet wo die Wohnung verlassen werden kann). Die Kanten AB der Graphen $\mathcal{W}$ repräsentieren die Möglichkeiten sich zwischen den Knoten zu bewegen, entweder indem eine Tür passiert wird oder indem zu einem Objekt hin- beziehungsweise von ihm weggegangen wird. Die Graphen aller Testwohnungen sind in den Abbildungen von 4.1 dargestellt.

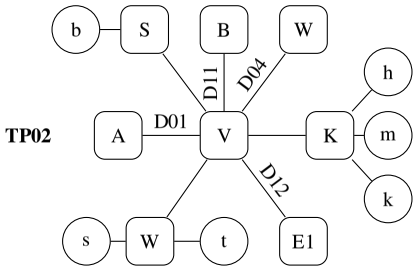
4.3 Zuordnung der Sensoren

Aufbauend auf dem oben beschriebenen Wohnungsmodell und den daraus resultierenden Graphen der Testwohnungen, wurden allen Knoten und Kanten je nach Verfügbarkeit entsprechende Sensoren zugewiesen. Dadurch wurde erreicht, dass zu jedem Zeitpunkt an jedem der Knoten und an jeder Kante Information über die aktuelle Aktivität verfügbar ist.
Grundsätzlich gibt es in den verschiedenen Räumen sehr unterschiedliche Ausstattungen. In manchen Räumen wurde zum Beispiel nur eine MSB installiert, in anderen vier oder fünf. Hier sollte aber darauf geachtet werden, dass den Knoten ähnliche Sensorkombinationen zugewiesen werden: Für jeden Knoten wurde daher maximal ein Sensor pro Sensortyp ausgewählt (es sind also maximal 5 Sensoren pro Knoten möglich). Sollten an einem Knoten mehrere Sensoren zu Auswahl gestanden haben (oft gab es mehrere Temperatursensoren oder mehrere Vibrationssensoren in einem Raum), so wurde jener Sensor ausgewählt, der den Ergebnissen der QR-Zerlegung nach als am relevantesten betrachtet werden konnte. (vgl. Kapitel 3.2.2).
Nicht allen Kanten konnten Sensoren zugewiesen werden, dies war nur bei denen möglich, wo ein Türkontaktsensor installiert wurde. In den Graphen der Testwohnungen sind die vorhandenen Türkontaktsensoren eingetragen (vgl. 4.1).

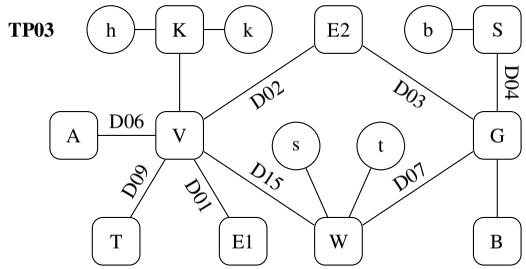
Tabelle 4.1 fasst noch einmal zusammen, welche Räume in den Testwohnung vorgekommen sind und enthält außerdem die Information darüber, welche Sensoren diesen Räumen zugewiesen wurden.
Tabelle 4.2 fasst noch einmal alle relevanten Objekte der verschiedenen Wohnungen zusammen und auch hier ist angegeben, welche Sensoren den Objekten zugewiesen wurden.
In Tabelle 4.3 ist zusammengefasst, welche Verbindungen (Kanten) in den Testwohnung vorgekommen sind und ob diesen Türkontaktsensoren zugewiesen werden konnten.



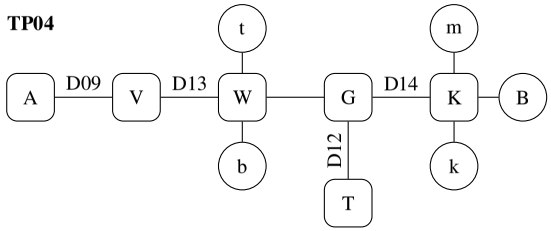
(a) Topologie - Testwohnung TP01;



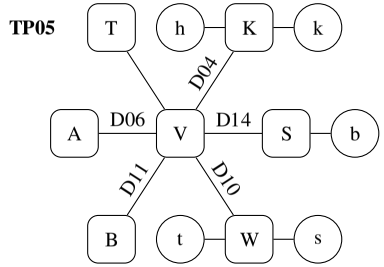
(b) Topologie - Testwohnung TP02;



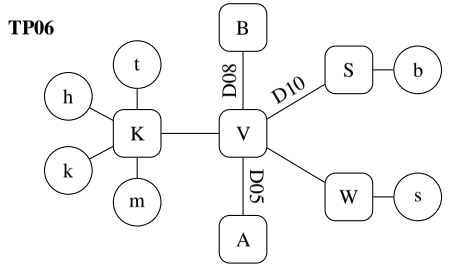
(c) Topologie - Testwohnung TP03;



(d) Topologie - Testwohnung TP04;

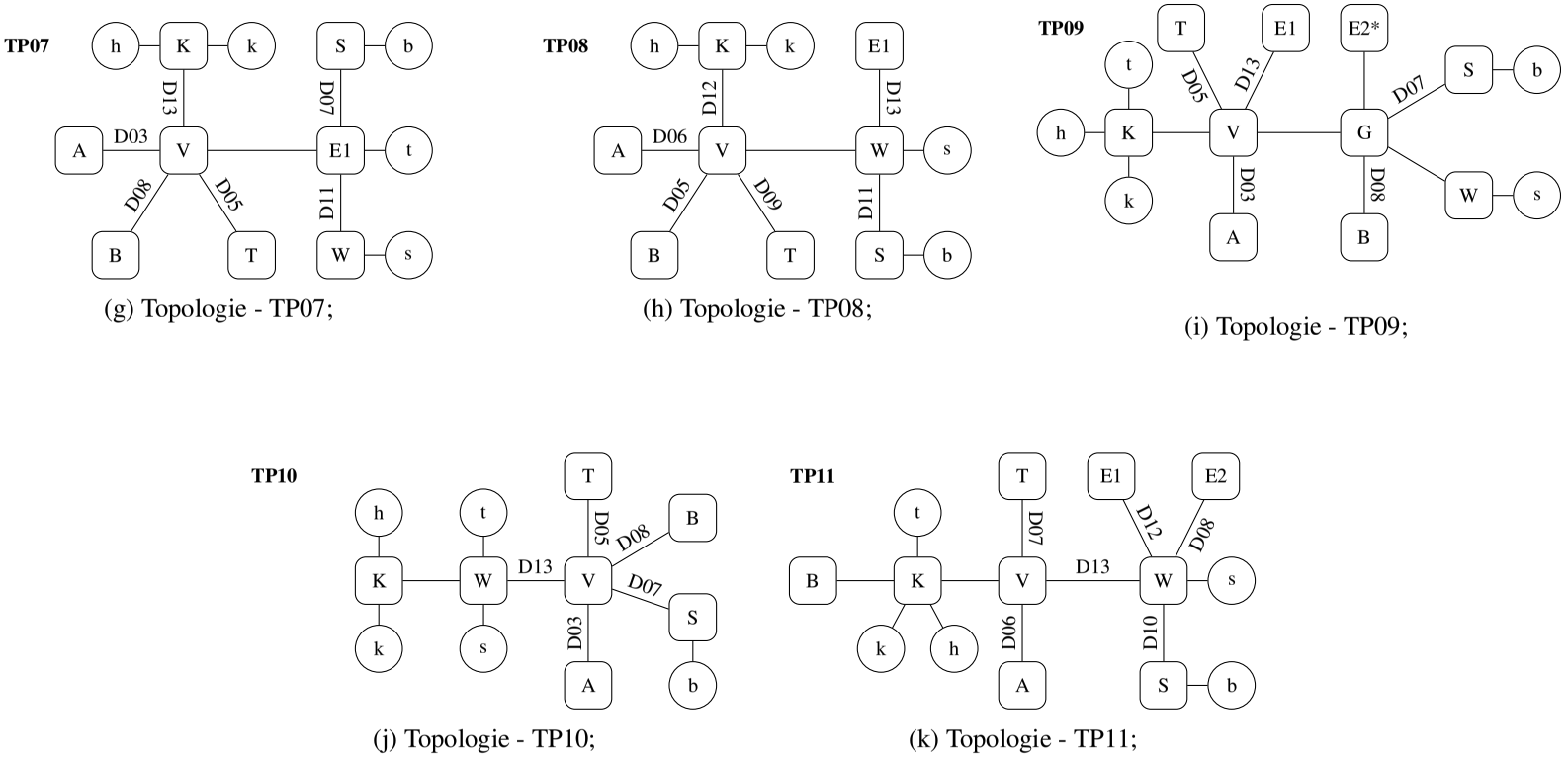


(e) Topologie - Testwohnung TP04;



(f) Topologie - TP04;

Abbildung 4.1: Graphen der Testwohnungen TP01 bis TP06. * Keine MSB installiert, bleibt daher unberücksichtigt. A = Außen; V = Vorzimmer; K = Küche; W = Wohnzimmer; S = Schlafzimmer; B = Badezimmer; T = Toilette; E1 = Extrazimmer; E2 = Extrazimmer; G = Gang; h = Herd; k = Kühlschrank; m = Mikrowelle; t = Tisch; s = Sofa; b = Bett.



(g) Topologie - TP07;

(h) Topologie - TP08;

(i) Topologie - TP09;

(j) Topologie - TP10;

(k) Topologie - TP11;

Abbildung 4.1: Graphen der Testwohnungen TP07 bis TP11. * Keine MSB installiert, bleibt daher unberücksichtigt. A = Außen; V = Vorzimmer; K = Küche; W = Wohnzimmer; S = Schlafzimmer; B = Badezimmer; T = Toilette; E1 = Extrazimmer; E2 = Extrazimmer; G = Gang; h = Herd; k = Kühlschrank; m = Mikrowelle; t = Tisch; s = Sofa; b = Bett;

	TP01	TP02	TP03	TP04	TP05	TP06	TP07	TP08	TP09	TP10	TP11	Anz
Außenbereich	×	×	×	×	×	×	×	×	×	×	×	11
Vorzimmer	×	×	×	×	×	×	×	×	×	×	×	11
V	V03	V13	V06		V06	V05	V03	V06	V03	V03	V06	10
M	M03	M01	M06	M09	M06	M05	M03	M06	M03	M03	M06	11
T	T03	T13	T06	T09	T06	T05	T03	T06	T03	T03	T06	11
L	L03	L13	L06	L09	L06	L05	L03	L06	L03	L03	L06	11
Küche	×	×	×	×	×	×	×	×	×	×	×	11
V	V10		V05	V11	V05	V01	V12	V12	V10	V12	V11	10
T		M14	M05	M11	M05		M13	M12	M10	M12	M11	9
M	T10	T14	T05	T14	T08	T07	T04	T12	T02	T10	T11	11
L	L04	L14	L05	L07	L08	T03	L12	L12	L04	L04	L11	11
Wohnzimmer	×	×	×	×	×	×	×	×	×	×	×	11
V	V11	V05	V11	V04	V10	V09	V11	V04	V06	V13	V04	11
M	M11	M05	M10		M10	M09	M11	M14	M06	M13	M13	10
T	T06	T10	T12	T04	T10	T09	T11	T04	T06	T09	T04	11
L	L06	L09	L13	L04	L10	L09	L11	L02	L06	L06	L01	11
Badezimmer	×	×	×	×	×	×	×	×	×	×	×	11
V	V08	V11	V07	V06	V11	V08	V08	V05	V08	V08	V05	11
M			M07			M08				M08	M05	4
T	T08	T11	T07	T06	T11	T08	T08	T05	T08	T08	T05	11
L	L08	L11	L07	L06	L11	L08	L08	L05	L08	L08	L05	11
WC	×	×	×	×	×		×	×	×	×	×	10
V	V05	V04	V09	V12	V09		V05	V09	V05	V08	V07	10
T	T05	T04	T09	T12	T09		T05	T09	T05	T08	T07	10
L	L05	L04	L09	L12	L09		L05	L09	L05	L08	L07	10
Schlafzimmer	×	×	×		×	×	×	×	×	×	×	10
V	V07	V07	V04		V12	V10	V07	V11	V07	V07	V10	10
M	M01	M07	M04		M14	M10	M07	M11	M07	M07	M10	10
T	T01	T02	T04		T12	T10	T01	T07	T07	T07	T10	10
L	L01	L02	L04		L12	L10	L01	L07	L07	L07	L10	10
Extrazimmer 1	×	×	×				×	×	×		×	7
V	V13						V06	V13	V13		V12	5
M		M12	M01				M06	M13	M13		M12	6
T	T13	T12	T01				T10	T13	T13		T12	7
L	L13	L12	L01				L09	L13	L13		L12	7
Extrazimmer 2	*		×						*		×	4
V											V08	1
M			M03								M08	2
T			T03								L08	2
L			L02								T08	2
Gang			×	×					×			3
V				V05					V11			2
M				M05					M11			2
T				T05					T11			2
L				L05					L11			2
Anzahl Räume	8	7	9	5	6	5	7	8	9	6	8	

Tabelle 4.1: Zuordnung der Sensoren zu den Räumen der Testwohnungen; × – Vorhanden;
* Zwar existiert dieser Raum, er bleibt aber unberücksichtigt, da keine MSB installiert wurde.
V = Vibrationssensor; M = Bewegungsmelder; T = Temperatursensor; L = Lichtsensor.

	TP01	TP02	TP03	TP04	TP05	TP06	TP07	TP08	TP09	TP10	TP11	Anz
Tisch	×	×	×	×	×	×	×	×	×	×	×	11
V	V02	V08	V15	V02	V02	V03	V02		V02	V02		9
Sofa	×	×	×		×	×	×	×	×	×	×	10
V	V06	V09	V13		V01			V01		V09	V01	7
Herd	×	×	×		×	×	×	×	×	×	×	10
T	H04	H03	H08		H08	H07	H04	H08	H04	H04		9
Kühlschrank	×	×	×	×	×	×	×	×	×	×	×	11
D	D04	D14	D05		D05	D04	D12	D10	D12	D12	D09	10
Mikrowelle		×		×								2
D		D03		D11								2
Bett	×	×	×	×	×	×	×	×	×	×	×	11
V	V01	V02		V10	V07	V06	V01	V07		V01		8
Anzahl Objekte	5	6	4	3	5	4	4	4	3	5	2	

Tabelle 4.2: Zuordnung der Sensoren zu den Objekten der Testwohnung; × = Vorhanden;

A	*B*	TP01	TP02	TP03	TP04	TP05	TP06	TP07	TP08	TP09	TP10	TP11	Anz
A	**VZ**	D03	D01	D06	D09	D06	D05	D03	D06	D03	D03	D06	11
VZ	**WC**	D08	D04	D09		×	D08	D05	D09	D05	D05	D07	10
VZ	**KÜ**	D10	×	×		D04	×	D13	D12	×		×	9
KÜ	**h**	×	×	×		×	×	×	×	×	×	×	10
KÜ	**k**	×	×	×		×	×	×	×	×	×	×	10
VZ	**WZ**	D11	×	D15	D13	D10	×		×		D13	D13	9
WZ	**s**	×	×	×		×	×	×	×	×	×	×	10
VZ	**BZ**	D80	D11			D11	D08	D08	D05		×		7
VZ	**SZ**	D07	×			D14	D10				D07		4
SZ	**b**	×	×	×		×	×	×	×	×	×	×	10
VZ	**E1**	D13	D12	D01				×		D13			4
KÜ	**t**	×					×		×	×		×	5
WZ	**t**		×	×	×	×					×		5
E1	**t**							×					1
WZ	**E1**							D11	D13			D12	3
VZ	**E2**	×		D02						D11			3
WZ	**SZ**								D11			D10	2
GA	**SZ**			D04						D07			2
GA	**BZ**			×						D08			2
GA	**WZ**			D07						×			2
KÜ	**BZ**				×							×	2
KÜ	**m**		×				×						2
WZ	**E2**											D08	1
SZ	**E1**							D07					1
BZ	**WC**						×						1
WZ	**KÜ**										×		1
GA	**KÜ**				D14								1
GA	**WC**				D12								1
GA	**VZ**										×		1
WZ	**b**				×								1
	Anz	13	13	14	7	11	13	12	12	12	13	12	13

Tabelle 4.3: Verbindungen zwischen Räumen bzw Objekten der Testwohnungen sowie deren zugeordnete Sensoren; × = Verbindung vorhanden, aber kein Sensor.
A = Außen; V = Vorzimmer; K = Küche; W = Wohnzimmer; S = Schlafzimmer; B = Badezimmer; T = Toilette; E1 = Extrazimmer; E2 = Extrazimmer; G = Gang; h = Herd; k = Kühlschrank; m = Mikrowelle; t = Tisch; s = Sofa; b = Bett;

4.4 Clustering mit ausgewählten Sensoren

Durch die Zuordnung ausgewählter Sensoren zu bestimmten Positionen in den Wohnungen konnte erneut untersucht werden, ob eine automatische Einteilung in Aktionsklassen mittels EM–Algorithmus, wie sie in Abschnitt 3.3.2 für alle Sensoren von TP04 durchgeführt wurde, gut interpretierbar ist. Dazu wurden die ausgewählten Sensoren, diesmal von TP07, genau wie in 3.3.2 geclustert. Abbildung 4.2 zeigt wieder als Beispiel einen Testtag, und zwar den Donnerstag der ersten Testwoche.

4.4.1 Beobachtungen.

Auffällige Klassen sind zum Beispiel:
- blau (abwesend), die Testperson verließ an diesem Tag offenbar zwei mal die Wohnung.
- rot (ruhiger Schlaf) und dunkelblau (unruhiger Schlaf) – treten abwechselnd in der Nacht auf.

Nächtliche WC-Gänge werden erkannt (zwischen 0:00 und 1:00 Uhr, sowie kurz vor 20:00 und kurz vor 13:00 Uhr; Details in Abbildung 4.3 und 4.4).

Die Bedeutung der übrigen Klassen kann in einer verbesserten Auflösung erkannt werden.

In Abbildung 4.3 ist eine detaillierte Ansicht eines nächtliche WC-Gangs dargestellt. Hier lässt sich erkennen, aus welcher Abfolge von Aktionsklassen dieser zusammengesetzt ist: rot (ruhiger Schlaf), dunkelblau (unruhiger Schlaf), hellgrün (Bewegung: Tisch, Vorzimmer; Vibration: Tisch); neongrün (Vibration und Tür: WC), gelb (Licht im WC sowie offene WC-Tür), danach in umgekehrter Reihenfolge wieder zurück. Diese Abfolge entspricht genau dem Weg, der zurückgelegt werden muss - andere WC-Gänge sind gleich oder zumindest sehr ähnlich aufgebaut (vgl. dazu Abbildung 4.4: es wird im Unterschied zum ersten gezeigten WC-Gang auch Aktivität in der Küche aufgezeichnet, vgl. orange Klasse). Daher kann eine Abfolge dieser Art als WC-Gang klassifiziert werden.

4.5 Bewegungs–Rekonstruktion

Die Vorbereitungen der Wohnungen, also das Erfassen der Topologie in einem Graphen und die Zuordnung der Sensoren zu den Knoten und Kanten, wurden bereits im Hinblick darauf durchgeführt, dass die Bewegungen der Testpersonen innerhalb der Wohnungen untersucht werden sollten. Die Graphen $\mathcal{W}_1 \ldots \mathcal{W}_{11}$ der Testwohnungen TP01 bis TP11 stellten dabei die Netzwerke dar, in denen sich die Personen bewegen konnten. Die Zeitreihen der Sensoren lieferten gezielte Information bei den entsprechenden Knoten und Kanten.
Mit dem in Abschnitt 2.5 beschriebenen Bewegungs–Rekonstruktions–Modell wurde daraufhin versucht, die Bewegungen der Personen zu rekonstruieren, indem die einzelnen Lösungsschritte durchgeführt wurden. Diese konnten aufgrund der hier getroffenen Vorbereitungen direkt durchgeführt werden.

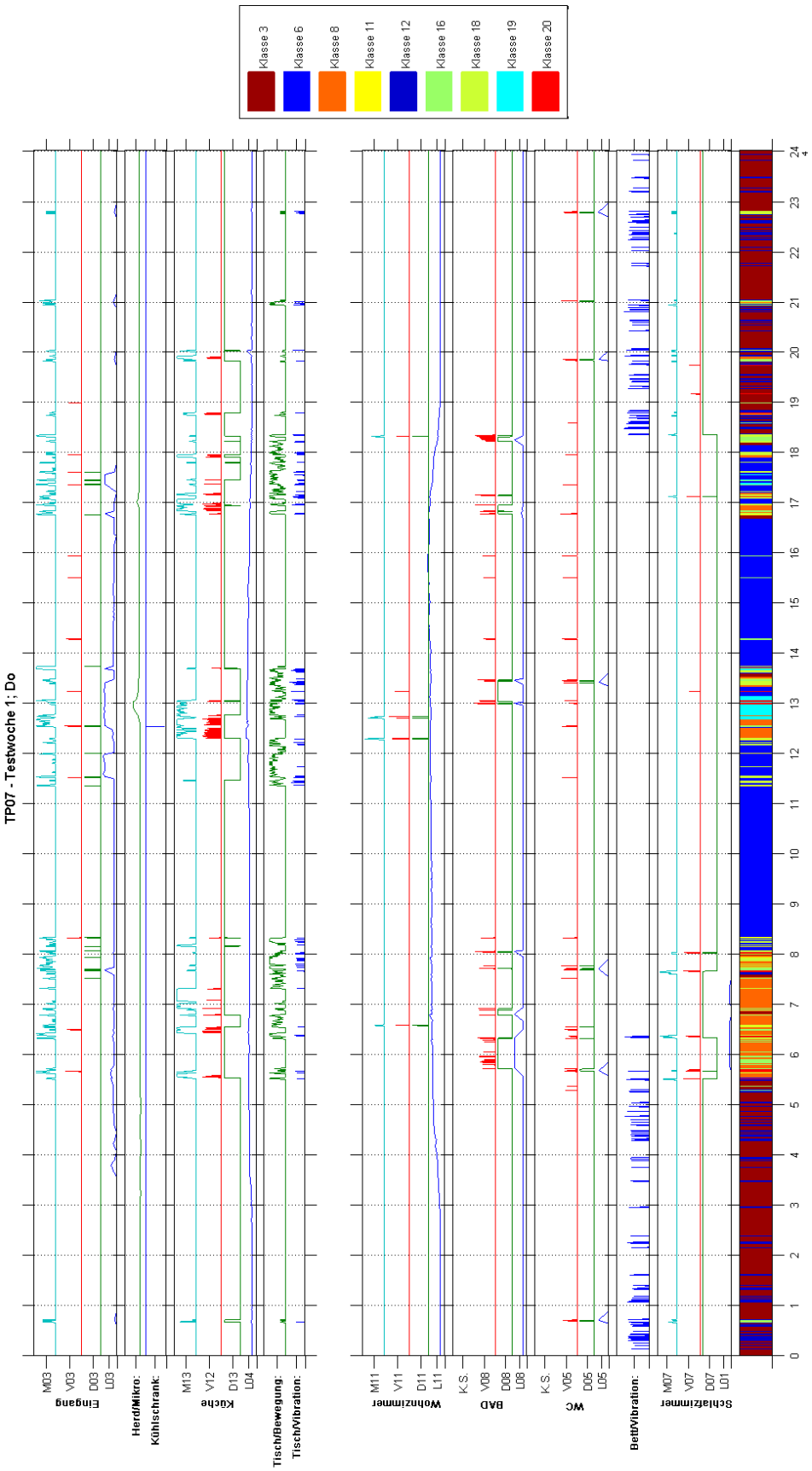


Abbildung 4.2: Von oben nach unten sind die ausgewählten Sensoren für Eingang, Herd, Küche, usw. abgebildet. Die letzte Zeile zeigt die Abfolge der berechneten Aktionsklassen; TP07; Do. Testwoche 1.

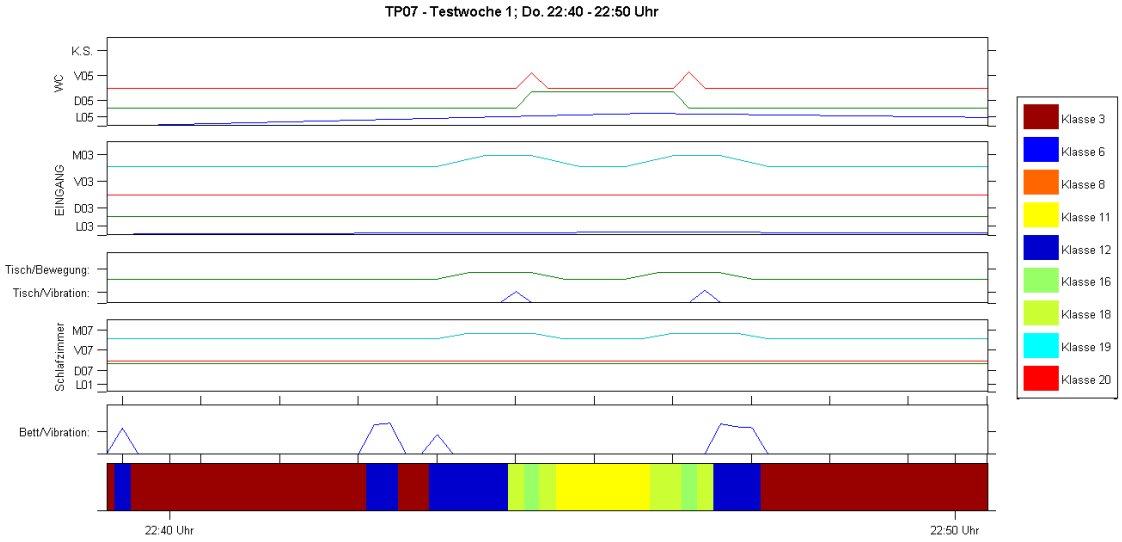


Abbildung 4.3: Ein nächtlicher Gang zur Toilette; Abgebildet sind nur Räume mit Aktivität; Letzte Zeile: Abfolge der Aktionsklassen; TP07; Do. Testwoche 1.

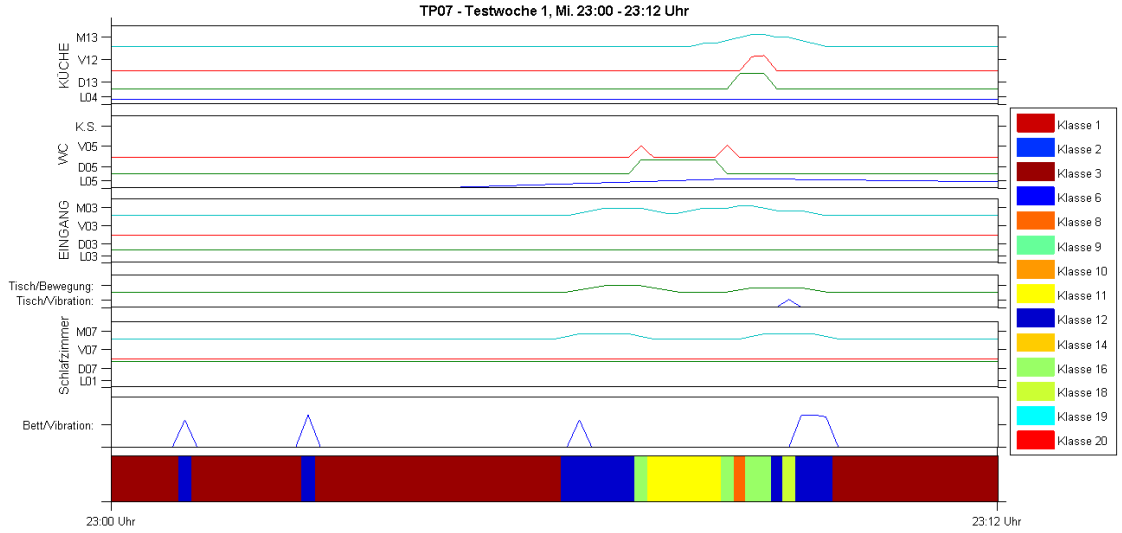


Abbildung 4.4: Ein nächtlicher Gang zur Toilette; Abgebildet sind nur Räume mit Aktivität; Letzte Zeile: Abfolge der Aktionsklassen; TP07; Mi. Testwoche 1.

5 Bewegungs–Rekonstruktion

Durch die Vorbereitungen aus den letzten Kapiteln konnte die Bewegungs–Rekonstruktion wie in 2.5 beschrieben durchgeführt werden. Die Graphen der Testwohnungen TP01 bis TP11, die in Abschnitt 4.2 erstellt wurden, entsprechen den geforderten Netzwerken. Die Subjekte die sich innerhalb der Netzwerke bewegen, sind die Testpersonen. Die Sensorzuordnung zu den Knoten und Kanten wurde in Abschnitt 4.3 durchgeführt (vgl. Tabellen 4.1 und 4.3) und kann hier direkt verwendet werden. Das liefert $\zeta_t^{\mathcal{N}}$ für alle Knoten $\mathcal{N}$ und z_T^{AB} für alle Kanten AB wie in 2.5.1 gefordert.

Durchführung der Lösungsschritte

Wie in Abschnitt 2.5 beschrieben erhalten wir die gesuchte Zeitreihe $x_t^{\mathcal{W}}$, indem die folgenden Schritte ausgeführt werden. Der erste Schritt kann für alle Wohnungen gleichzeitig durchgeführt werden.

Schritt 1 – Aufenthaltswahrscheinlichkeit bei den Knoten

Entsprechen der obigen Beschreibung wurde für jeden Typ τ die Vereinigung aller $\zeta_t^{\mathcal{N}}$ vom Typ τ in c Cluster aufgeteilt. Beobachtet werden konnte:

- Für ein gutes Ergebnis reicht eine kleine Anzahl von Cluster (ca. 3 bis 6).
- Bei Überprüfen der Clusterzeitreihen konnte festgestellt werden, dass manchmal eine Neuberechnung oder Nachbearbeitung (wie das Aufteilen eines Clusters in Sub–Cluster) notwendig ist.

Danach wurde den berechneten Clustern eine anfängliche Aufenthaltswahrscheinlichkeit zugewiesen (vgl. Tabelle 5.1, Angaben in Prozent).

	A	**VZ**	**KÜ**	**WZ**	**BZ**	**WC**	**SZ**	**t**	**s**	**h**	**k**	**b**
Klasse 1	10	0	0	0	5	0	5	10	15	10	0	30
Klasse 2		80	5	90	10	80	80	99	99	90	99	100
Klasse 3		99	90	99	99	99	99					
Klasse 4			99									

Tabelle 5.1: Aufenthaltswahrscheinlichkeit der Klassen der Knoten in Prozent

In Abbildung 5.1 sind die daraus resultierende Zeitreihen für die Knoten der Wohnung TP05 (Testtag 11) zu sehen. Die Farben sind abgestuft so dass Schwarz 100% Wahrscheinlichkeit für den Aufenthalt und weiß 0% Wahrscheinlichkeit für den Aufenthalt bedeutet.

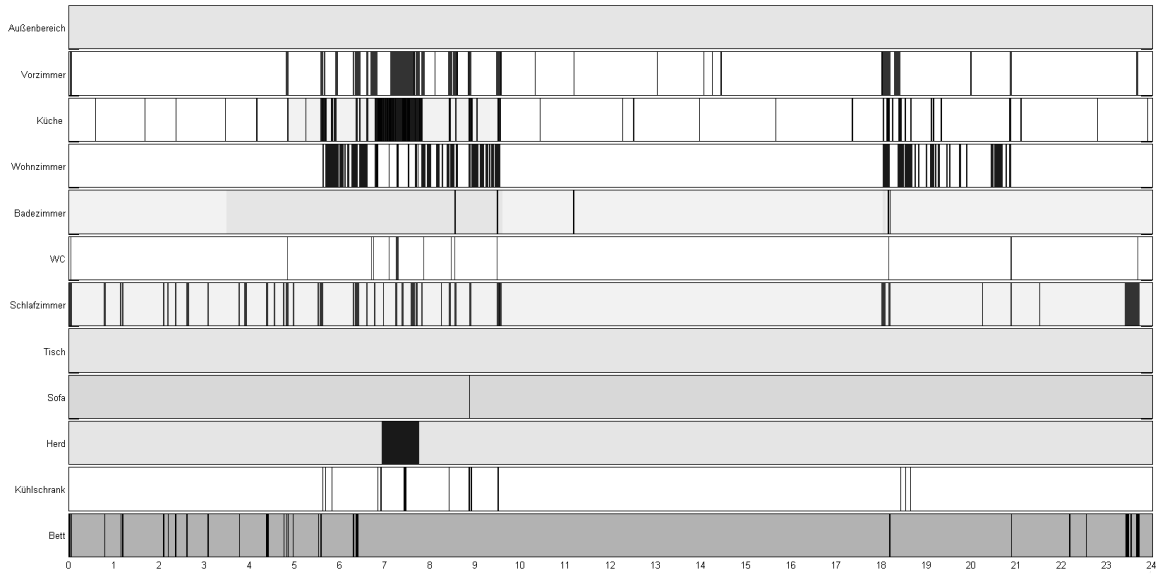


Abbildung 5.1: Zeitreihen der geschätzten Aufenthaltswahrscheinlichkeiten der Klassen; TP05, Testtag 11

Schritt 2 – Aufenthaltswahrscheinlichkeiten entlang der Kanten

Im nächsten Schritt wurden dann wie in 2.5.2 beschieben, die anfänglichen Aufenthaltswahrscheinlichkeiten $p_t^{AB}(x)$ entlang der Kanten berechnet. Diese sind für TP05 und Testtag 11 dargestellt in Abbildung 5.2.
Danach werden die anfänglichen Zustandssequenzen x_t^{AB} berechnet, diese sind wieder für TP05 und Testtag 11 in Abbildung 5.3 dargestellt.
Diese anfänglichen Zustandssequenzen resultieren aber aus den anfänglich per Hand zugewiesenen Wahrscheinlichkeiten. Um diese zu verbessern und um etwaige Fehler auszugleichen wurde zusätzlich Information aus den Daten hinzugenommen: mittels überwachter Klassifizierung wurden die Daten einer Kante AB, also alle Zeitreihen von A vereinigt mit den Zeitreihen von B und, falls vorhanden, mit jener von AB selbst, in Klassen eingeteilt. Die verborgene Sequenz, die zur überwachten Klassifizierung verwendet wurde, war x_t^{AB}. Die resultierenden Zeitreihen y_t^{AB} wurden dann zur Neuberechnung von $p_t^{AB} = p(x \mid y_t)$ verwendet, womit dann das Potential $V^{AB}(x, y)$ berechnet werden konnte, welches in Schritt 4 zur Berechnung der konsistenten Zustandssequenz x_t verwendet wurde.
Optional kann an dieser Stelle Schritt 3 (vgl. 2.5.2) durchgeführt werden. Hier wurde dieser Schritt ausgelassen und direkt mit Schritt 4 (vgl. 2.5.2) fortgesetzt.

Schritt 4 – Konsistente Zuordnung der Zustände

Hier wird mittels dynamischer Programmierung (vgl. z.B. [16]) aus den in Schritt 3 berechneten Potentialen mit bestimmten Beschränkungen die wahrscheinlichste konsistente Zustandssequenz berechnet. Die Zustände bezeichnen bereits die Knoten der Testwohnung und konsistent bedeutet

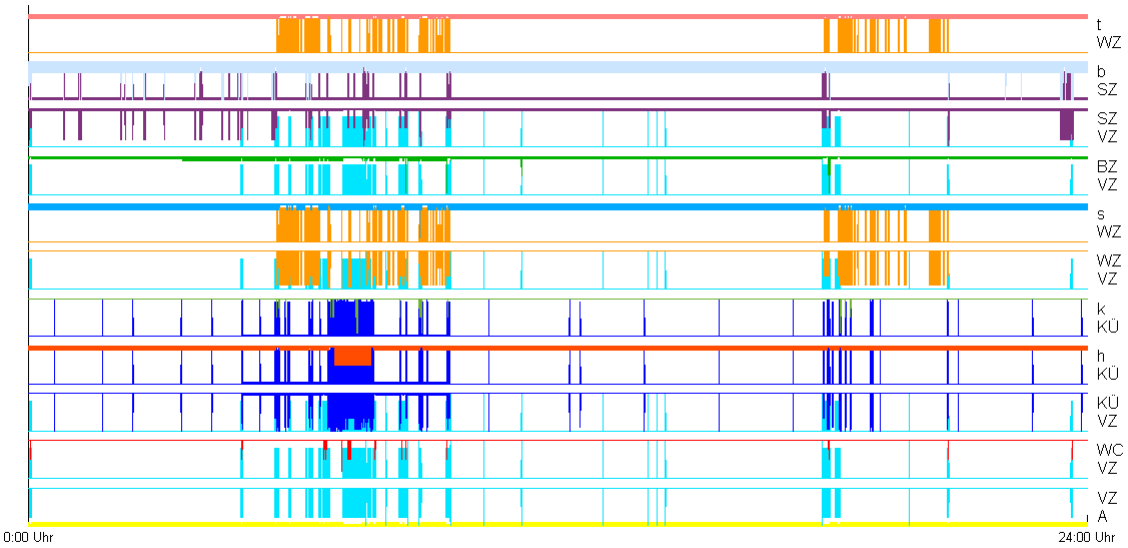


Abbildung 5.2: Anfängliche Wahrscheinlichkeiten für den Aufenthalt entlang der Kanten; TP05, Testtag 11

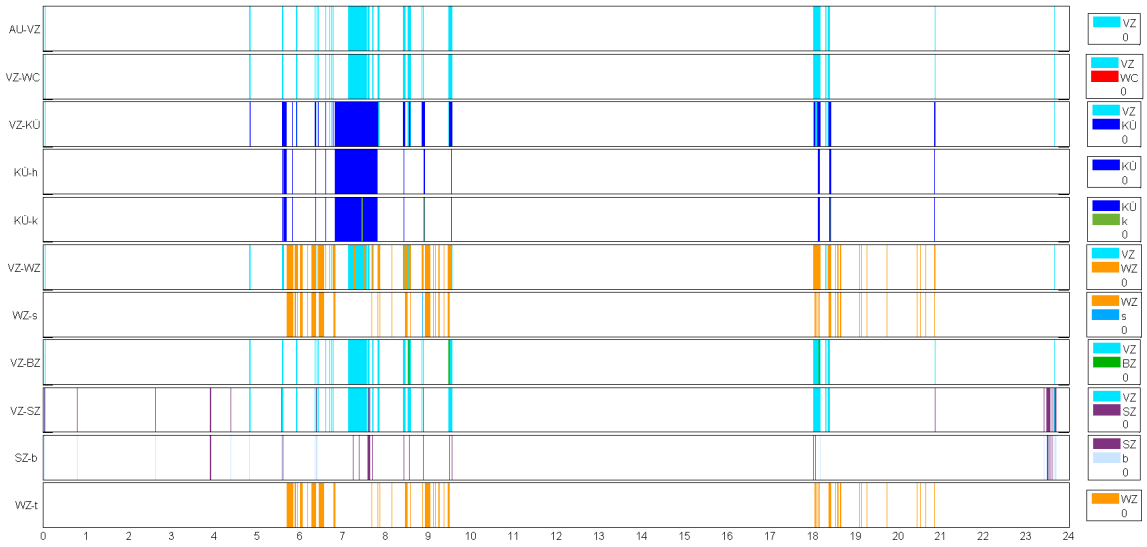


Abbildung 5.3: Anfängliche Zustandssequenzen x_t^{AB}; TP05, Testtag 11

hier, dass ein Zustand (also ein Knoten) nur auf einen anderen folgen kann, wenn es sowohl die Topologie als auch die Türen (d.h. sollte eine Tür vorhanden sein, muss diese auch offen sein, um durchgehen zu können) zulassen.
Es wurde also zunächst bestimmt, wann Türverbindungen definitiv geschlossen waren (vgl. o_t^{AB} in 2.5.2). Mit Einschränkung der geschlossenen Türen, sowie der Information welche Kanten in den Wohnungen existieren, konnte dann die initiale Zustandssequenz x_t berechnet werden.
Abbildung 5.4 zeigt wieder TP05 und Testtag 11. Diesmal sind die Sensordaten aller Kanten und Knoten abgebildet, sowie die initiale Zustandssequenz x_t (letzte Zeile).

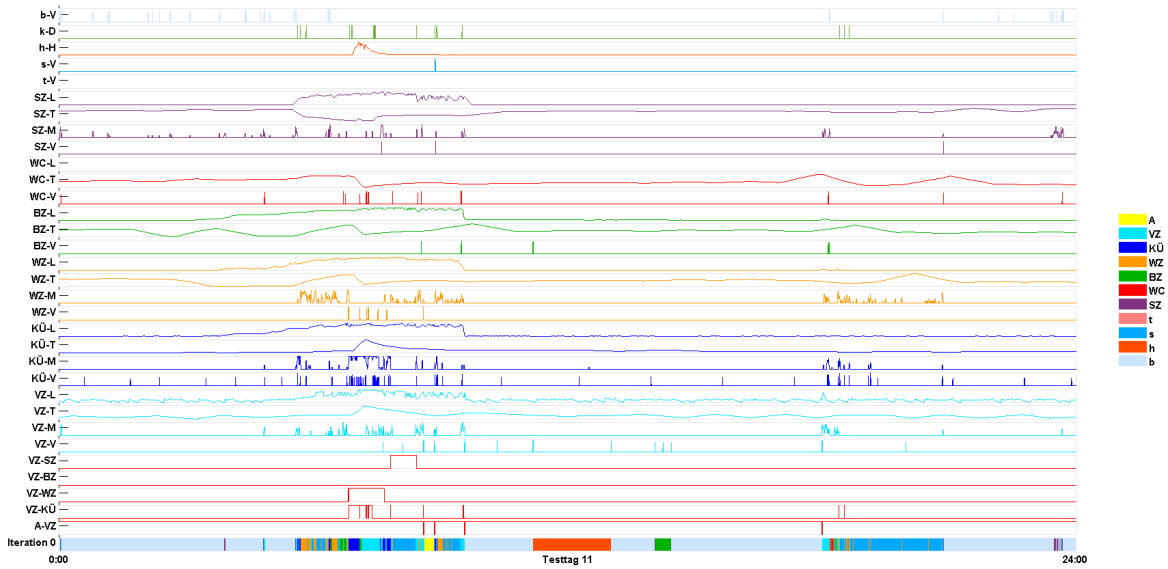


Abbildung 5.4: Zuordnung der Zustände vor dem Training; TP05, Testtag 11.

5.0.1 Beobachtungen.

Es fällt auf, dass die Zuordnung der Zustände zwar konsistent ist, aber noch nicht unbedingt richtig erscheint. Je nachdem wie die Wahrscheinlichkeiten am Anfang definiert wurden, dominierte beispielsweise wie im gezeigten Beispiel der Aufenthalt im Bett.

Aufenthalt im Bett (mint) stimmt in der Nacht.

Unter Tags ist diese Zuordnung aber völlig unzutreffend, da die Person offensichtlich abwesend ist (zwischen ca. 11:00 und 18:00 Uhr), aber vor dem nicht erkannten Verlassen und Betreten der Wohnung ist der geschätzte Aufenthalt das Vorzimmer, was schon in die richtige Richtung weist.

Aufenthalt am Sofa (lila) könnte stimmen, jedenfalls ist Aufenthalt im Wohnzimmer zu diesen Zeitpunkten aufgrund der Daten sehr wahrscheinlich.

Die abwechselnde Zuordnung von Küche (dunkelblau), Wohnzimmer (rot) und Sofa (lila) am Morgen könnte stimmen – obwohl hier auch der Tisch eine Rolle spielen könnte (Frühstück), welcher aber kaum zugewiesen wurde.

Durch ein Training wurde versucht die Unstimmigkeiten auszugleichen.

Schritt 5 – Trainieren der Zustandssequenz

Es wurde wie in Schritt 2.5.2 beschrieben, das diskrete Bayesische Netzwerk mittels Viterbi Training trainiert. In Abbildung 5.5 ist wieder die initiale Zustandssequenz dargestellt (letzte Zeile), sowie die Veränderungen nach mehreren Iterationen. Einerseits lassen sich zwar Verbesserungen feststellen, so wird schon ab der dritten Iteration die Abwesenheit richtig zugeordnet, es lassen sich aber auch Verschlechterung feststellen, wie zum Beispiel in der Nacht, wo die eindeutige Zuordnung zum Bett erst mal verloren geht (zweite Iteration) und erst später wieder zurückkommt.

Insgesamt gab es also Teilsequenzen von x_t, wo der geschätzte Aufenthalt mit dem tatsächlichen Aufenthalt, aufgrund eines Vergleiches mit den Daten, gut übereinstimmte und es gab aber auch Teilsequenzen, wo das nicht der Fall war. Dafür gibt es mehrere Gründe. Einerseits das Trainingsverfahren, welches zwar effizient, aber nicht besonders gut ist – das Baum–Welsch Training (vgl. [8]) kann möglicherweise zu besseren Ergebnissen führen.
Außerdem ist in diesem Modell nicht berücksichtigt, dass sich auch mehrere Personen in diesem Netzwerk aufhalten können. Sollten das der Fall sein, kann es zu Unstimmigkeiten kommen, da es nur auf eine Person ausgerichtet ist.

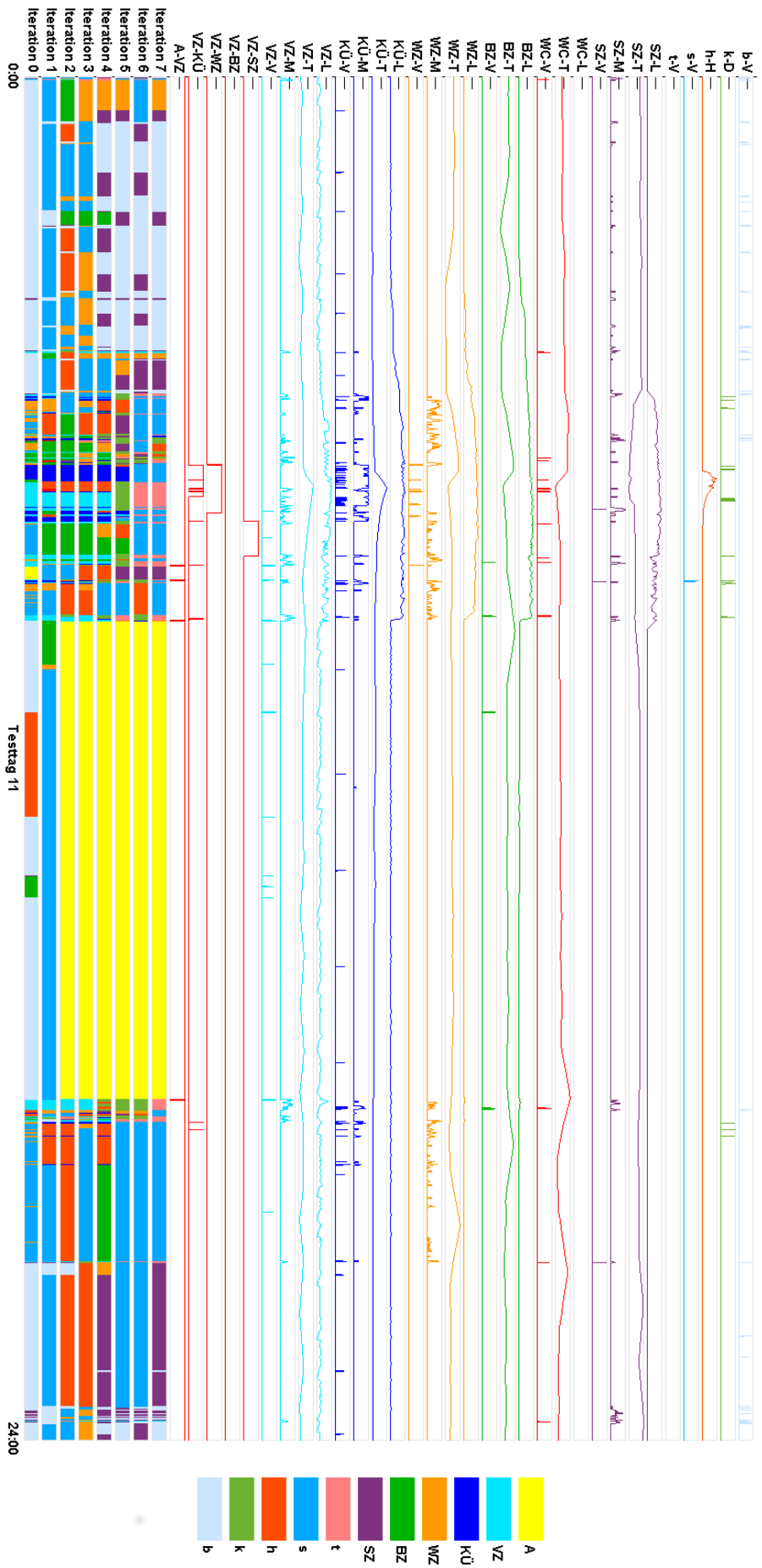


Abbildung 5.5: Zuordnung der Zustände; die ersten sieben Iterationen; TP05, Testtag 11.

6 Zusammenfassung und Diskussion

6.1 Zusammenfassung

Die Daten aus dem eHome Projekt wurde einer Analyse unterzogen. Zuerst wurden eine Bereinigung der Rohdaten vorgenommen und die vorhandenen Sensoren der einzelnen Testwohnungen wurden festgestellt. Es wurde eine deskriptive Analyse durchgeführt. Dazu zählt das Bestimmen verschiedener statistischer Werte der einzelnen Sensoren, beziehungsweise der Sensortypen, sowie die Verteilung der Signale über 24 Stunden und der Zeitintervalle zwischen den Signalen. Darauf aufbauend wurden die Sensordaten in äquidistante Zeitreihen mit Zeitfenstern von 10 Sekunden umgewandelt und eine Klassifikation dieser Zeitfenster wurde durchgeführt. Untersucht wurde dabei der K-Means Algorithmus in Kombination mit Hierarchischer Klassifizierung und der EM-Algorithmus für Gaußsche Mixturen, welcher die besten Ergebnisse lieferte. Mittels Hauptkomponentenanalyse wurden die Daten vor der Anwendung dieser Algorithmen reduziert. Durch ein Abstrahiertes Wohnungsmodell wurde die Topologie jeder Wohnung in Form eines Graphen erfasst und ausgewählte Sensoren wurden bestimmten Positionen innerhalb der Testwohnungen zugewiesen (vgl. Kapitel 4). Dabei wurde berücksichtigt, welche Situationen beziehungsweise Aktivitäten für eine Feststellung von gesundheitlichen Veränderungen relevant sind (vgl. Abschnitt 4.1). Die Klassifikation der Zeitfenster wurde mit den ausgewählten Sensoren wiederholt. In Kapitel 5 wurde dann gezeigt, wie aufbauend auf diesen Wohnungsmodellen eine Rekonstruktion der Bewegung über den gesamten Testzeitraum durchgeführt werden kann. Der Algorithmus zur Rekonstruktion basiert auf Bayesischen Netzen und besteht aus einer Abfolge verschiedener Klassifizierungs- und Optimierungs-Schritte (vgl. [14]).

6.2 Ergebnisse

Es hat sich gezeigt, dass die durch das *eHome* Assistenzsystem aufgezeichneten Daten sehr gute Rückschlüsse auf die Aktivität der Testpersonen zulassen. Die Daten sind für automatisierte Verfahren gut geeignet – es lassen sich mit wenigen interaktiven Schritten eine Reihe aussagekräftiger Ergebnisse berechnen. Das beginnt beim Erfassen der Sensordaten – die verschiedenen Typen senden so unterschiedliche Signale, dass der Typ eindeutig und automatisch erkannt werden kann. Das Ausmustern von Fehlerwerten und Ausreißern erfordert zwar Interaktion, könnte aber auch, durch bereits vorhandene Erfahrungswerte, zukünftig automatisch durchgeführt werden. Äquidistante Zeitreihen zu erstellen ist ohne jegliches Zutun möglich – allerdings gibt es viele Möglichkeiten zur Variation bei der Erstellung. Die vorgeschlagene Variante stellt nur eine Möglichkeit dar, welche weiter verfeinert und verbessert werden kann (zum Beispiel durch mehr

Bezugnahme auf die Vergangenheit oder ein verbessertes Verfahren zu Interpolation der Werte). Die Untersuchung der Daten in 10 Sekunden Abständen liefert an sich gute Ergebnisse, aber auch hier könnten weitere Untersuchungen darüber, wie sich eine Veränderung dieser Intervallänge auswirkt, interessant sein.

Die Einteilungen in Aktionsklassen, wie sie mittels EM-Algorithmus ohne zusätzliche Informationen allein aus den Daten zustande kommen, stimmen in weiten Teilen mit dem natürlichen Verhaltensablauf überein.

Ein interaktives Programm ermöglicht all diese Berechnungen für eine beliebige Testwohnung in relativ kurzer Zeit durchzuführen. Das Aufarbeiten der Daten erfordert ca. 10 Minuten interaktive Aufnahme, sowie ca. 10 bis 15 Minuten interaktives Rechnen. Das Berechnen der Cluster mittels EM-Algorithmus ist in relativ kurzer Zeit möglich (selbst bei den großen Datensätzen nicht mehr als ein paar Minuten).

Durch die Versuche zur Bewegungs–Rekonstruktion konnte festgestellt werden, dass der Aufenthalt streckenweise richtig erkannt werden kann. Allerdings erfordert eine zuverlässige Berechnung der Aufenthaltssequenzen noch einiges an Arbeit. Nicht nur, dass in jedem Schritt die Algorithmen verbessert und die Parameter angepasst werden sollten, auch bei der Auswahl der Sensoren könnten noch Änderungen und Verbesserungen vorgenommen werden. Es lässt sich erwarten, dass eine Weiterentwicklung und Verbesserung der Algorithmen zu gelungenen Rekonstruktionen der Bewegungssequenzen führt. Eine Analyse dieser Sequenzen kann dann Aufschluss über das typische Verhalten der Testpersonen geben. Ein Beispiel wäre die Untersuchungen der Zustandsübergänge. Damit könnte festgestellt werden, wie die Personen sich zwischen den Räumen bewegen; welche Wege sie bevorzugt verwenden; usw. Ein anders Beispiel wäre die Untersuchung der zeitlichen Verteilung der Zustände. Dies könnte Auskunft über die Häufigkeit und die Dauer des Aufenthalts in den einzelnen Räumen liefern. Daneben könnte untersucht werden, ob bestimmte Episoden (d.h. Abfolgen bestimmter Zustände) häufig vorkommen. Solche Episoden könnten katalogisiert werden und darauf basierend können weitere Untersuchungen Auskunft über untypisches Verhalten liefern.

6.3 Empfehlungen bezüglich der vom AAT gestellten Fragen

Bezüglich der Erkennung der in 4.1 genannten relevanten Aktivitäten lässt sich aus den Ergebnissen, die im Laufe der Arbeit beobachtet werden konnten, sagen, dass viele dieser Aktivitäten aus dem zeitlichen Kontext und durch die berechneten Aktionsklassen und Bewegungssequenzen im Nachhinein bestimmt werden können. So kann zum Beispiel eine bestimmte Abfolgen von Zuständen oder Klassen als eine spezielle Aktion klassifiziert werden. Eine Abfolge der Zustände: Bett, Schlafzimmer, Vorzimmer, WC, Vorzimmer, Schlafzimmer, Bett, wäre dann ein nächtlicher WC-Gang oder eine Abfolge: Küche, Herd, Kühlschrank, Herd, Küche, ... Wohnzimmer, Tisch (für einen längeren Zeitraum), wäre Essenszubereitung und Essenskonsumation und aus dem zeit-

lichen Kontext ließe sich dann schließen, um welche Mahlzeit es sich tatsächlich dabei handelt (Frühstück, Mittag- oder Abendessen). Auf diese Weise könnte dann aus den berechneten Bewegungssequenzen und Aktionsklassen durch genaue Klassifizierung von Teilsequenzen eine neue Zeitreihe generiert werden, deren Zustände dann nicht den Aufenthalt beschreiben, sondern konkrete Aktivitäten.

Wenn in einem Raum drei verschieden Positionen unterschieden werden sollen (wie Aufenthalt Wohnzimmer oder speziell Tisch oder Sofa, welche auch beide im Wohnzimmer sind), um verschiedenen Aktivitäten wie Essen oder Ausruhen feststellen zu können, sollte darauf geachtet werden, dass die Sensoren so installiert sind, dass sich zumindest manche Sensoren eindeutig der einen oder anderen Position zuordnen lassen. Außerdem wäre es sehr hilfreich, wenn pro gewünschter Position auch ein Bewegungssensor installiert wäre, denn diese liefern sehr wertvolle Information über den Aufenthalt und das auch bei relativ ruhigen Tätigkeiten wie Lesen oder Essen, bei denen die Vibrationssensoren kaum reagieren.

Trotzdem können meiner Einschätzung nach manche der in 4.1 genannten Aktivitäten nicht erkannt werden. Zum Beispiel ist es wohl nicht möglich festzustellen, ob die Testperson die zum Beispiel am Sofa sitzt nun ein Buch liest oder fernsieht – diese Informationen gehen aus den Daten nicht hervor.

Auch ist es mit Hilfe der vorliegenden Daten meiner Meinung nach nicht möglich festzustellen, ob regelmäßig gelüftet wird. Es gibt keine Fenster-Kontaktsensoren mit welchen das überprüft werden könnte. Es ist zwar davon auszugehen, dass in der kalten Jahreszeit ein plötzliches Abfallen der Temperatur erkannt und dann als Lüften klassifiziert werden könnte. Die vorliegenden Testreihen wurden aber in der warmen Jahreszeit durchgeführt und daher kann darüber keine Aussage getroffen werden.

Die Temperatur- und Lichtsensoren enthalten sehr häufig redundante Informationen – es wäre daher meiner Einschätzung nach möglich, die Anzahl dieser Sensoren zu reduzieren. Auch ist die Anzahl der Signale, die diese Sensoren aufgrund des maximalen Sendeintervalls übertragen, sehr groß. Es könnte überlegt werden dieses Intervall zu vergrößern, eventuell in Kombination mit einer etwas kleineren Änderungsrate. Um Genaueres zur Granularität der Daten oder zur Auflösung sagen zu können, könnten die durchgeführten Berechnungen, mit zuvor manipulierten Daten (entweder weniger Signale oder weniger genaue Werte) wiederholt werden.

7 Ausblick

Wie bereits in der Einleitung erwähnt, gibt es eine Vielzahl verschiedener Berichte und Forschungsergebnisse zu AAL-Systemen. Manche der erprobten Methoden könnten für das eHome Assistenzsystem implementiert und getestet werden. Dazu zählen auch anspruchsvollere Methoden und Routinen, wie sie zum Beispiel in [5] vorgestellt werden.

Bis jetzt wurden die aus dem Projekt anfallenden Daten mit eher einfachen Mitteln verwertet. Es ist damit zu rechnen, dass die Anwendung solcher vorgeschrittenen Methoden zu einer deutlichen Verbesserung der aus den Daten gewonnenen Information führt, so dass eine robuste Erkennung von Routinen und Regelmäßigkeiten möglich wird, wie auch die Erkennung von Abweichungen von diesen Routinen.

Mit erfolgreich getesteten Methoden (zum Beispiel durch Kreuzvalidierung) könnte ein selbstlernendes Programmsystem entwickelt und implementiert werden, das für den Online-Betrieb geeignet ist. Dazu gehört unter anderem der Aufbau einer Datenbank zur Speicherung und Verwaltung der extrahierten Information; das Entwickeln bestimmter Inferenzregeln zur Verarbeitung dieser Information und der Aufbau einer geeigneten Kommunikationsmöglichkeit zwischen dem System und den NutzerInnen, wobei die technischen Voraussetzungen dafür – durch das integrierte Userterminal – gegeben sind.

Schlussendlich könnten diese Weiterentwicklungen dazu führen, dass eine Integration dieses Programmsystems in das bestehende *eHome*-Assistenzsystem eine Bereicherung für das Projekts darstellt.

Abbildungsverzeichnis

Tabellenverzeichnis

Index

Literaturverzeichnis

[1] Bacher, J. ; Pöge, A. ; Wenzig, K.: *Clusteranalyse*. Oldenbourg Verlag, 2010

[2] Bruckner, D.: *Probabilistic Models in Buolding Automation: Recognizing Sezenarios with Statistical Methods*, Technische Universität Wien, Dissertation, 2007

[3] Central European Institute of Technology: *AAL-eHome*. – URL `http://deutsch.ceit.at/ceit-raltec/projekte/aal---ehome`. – [letzter Zugriff: 01.04.2013]

[4] Chen, L. ; Nugent, C.D. ; Biswas, J. ; Hoey, J.: *Activity Recognition in Pervasive Intelligent Environments*. Bd. 4. Atlantis Press, 2011

[5] Georgieva, P. ; Mihaylova, L. ; Jain, L.: *Advances in Intelligent Signal Processing and Data Mining: Theory and Applications (Studies in Computational Intelligence*. Springer Verlag, 2013

[6] Golub, G.H. ; Van Loan, C.F.: *Matrix computations*. 3. Auflage. Johns Hopkins University Press, 1996

[7] Götze, J.: *Orthogonale Matrixtransformationen*. R. Oldenbourg Verlag, 1995

[8] Jaakkola, T.: *Machine Learning, Fall 2006*. Vorlesungsskriptum. 2006. – URL `http://ocw.mit.edu/`. – MIT OpenCourseWare [letzter Zugriff: 01.04.2013]

[9] Kleinberger, T. ; Jedlitschka, A. ; Storf, H. ; Steinbach-Nordmann, S. ; Prueckner, S.: An Approach to and Evaluations of Assisted Living Systems Using Ambient Intelligence for Emergency Monitoring and Prevention. In: Stephanidis, Constantine (Hrsg.): *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments* Bd. 5615. Springer Berlin Heidelberg, 2009, S. 199–208

[10] Marschollek, M. ; Bott, O.J. ; Wolf, K.H. ; Gietzelt, M. ; Plischke, M. ; Madiesh, M. ; Song, B. ; Haux, R.: Home care decision support using an Arden engine–merging smart home and vital signs data. In: *Studies in health technology and informatics* 146 (2009), S. 483

[11] Mayer, P. ; Panek, P.: Assessing Daily Activity of Older Persons in a Real Life AAL System. In: *Telehealth 2012, Innsbruck, Österreich*. 2012, S. 772–775

[12] Neumaier, A.: *Introduction to numerical analysis*. Cambridge University Press, 2001

[13] Neumaier, A.: *Künstliche Intelligenz*. Vorlesungsskriptum. WS 2011. – URL `http://www.mat.univie.ac.at/~neum/`. – [letzter Zugriff: 01.04.2013]

[14] Neumaier, A.: *Methoden der Datenanalyse*. Vorlesungsskriptum. WS 2012. – URL `http://www.mat.univie.ac.at/~neum/`. – [letzter Zugriff: 01.04.2013]

[15] Runkler, T.A.: *Data Mining*. Vieweg+Teubner, 2010

[16] Russell, S. ; Norvig, P.: *Artificial Intelligence*. 3. Auflage. Pearson Education, 2010

[17] Werner, K. ; Werner, F. ; Panek, P. ; Hlauschek, W. ; Diermaier, J.: eHome-Wohnen mit unterstützender Intelligenz. In: *4. Deutscher AAL Kongress, Berlin*, VDE VERLAG GmbH, 2011

[18] Yin, G.Q.: *Automatic Scenario Detection for Ambient Assisted Living of Elderly People*, Technische Universität Wien, Dissertation, 2012

[19] Zentrum für Angewandte Assistierene Technologien. – URL `http://www.aat.tuwien.ac.at/index.html`. – [letzter Zugriff: 01.04.2013]

[20] Zentrum für Angewandte Assistierene Technologien: *eHome Homepage*. – URL `http://www.aat.tuwien.ac.at/ehome/index.html`. – [letzter Zugriff: 01.04.2013]

[21] Zentrum für Angewandte Assistierene Technologien: *eHome Projektergebnisse*. – URL `http://www.aat.tuwien.ac.at/ehome/e-HomeErgebnisse.html`. – [letzter Zugriff: 01.04.2013]

Lebenslauf

Persönliche Daten

Name:	Melanie Obernosterer
Geburtsdatum:	16.05.1981
Familienstand:	In Partnerschaft lebend, zwei Kinder

Ausbildung

seit März 2005	Bachelorstudium Medieninformatik
seit März 2002	Diplomstudium Mathematik
September 2001 - Februar 2002	Diplomstudium Astronomie
Juni 2001	Reife- und Diplomprüfung mit ausgezeichnetem Erfolg
1996 - 2001	Höhere Technische Bundeslehranstalt für Kunst und Design

Wien, 9. April 2013