



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

# Ausarbeitung Anwendungen 1

Wolfram Sokollek

Context-Awareness im Living Place Hamburg

Wolfram Sokollek  
Context-Awareness im Living Place Hamburg

Studiengang Master Informatik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Abgegeben am 28. Februar 2010

## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>4</b>
1.1 Motivation . . . . .	4
1.2 Inhalt dieser Arbeit . . . . .	4
1.3 Begriffsklärung Kontext . . . . .	4
<b>2 Architekturen für Kontexterkenennung</b>	<b>5</b>
2.1 iFlat - Eine dienstorientierte Architektur für intelligente Räume . . . . .	5
2.2 Gaia-Context-Infrastructure . . . . .	6
2.3 Context Aware Learning System . . . . .	7
2.4 Zusammenfassung der Architekturen für Kontexterkenennung . . . . .	8
<b>3 Kontexterkenennung</b>	<b>8</b>
3.1 Kontexterkenennung mit Hilfe von Prädikatenlogik . . . . .	8
3.2 Naive Bayesian Classifier . . . . .	10
3.3 Kontexterkenennung mit Support Vector Machines (SVM) . . . . .	12
3.4 Zusammenfassung Kontexterkenennung . . . . .	13
<b>4 Fazit und Ausblick</b>	<b>13</b>
<b>Literatur</b>	<b>14</b>

# 1 Einführung

## 1.1 Motivation

Im Rahmen des Living Place Hamburg Projektes der HAW-Hamburg wird eine Loftwohnung aufgebaut, die dem Paradigma des Ubiquitous Computing (UbiComp) folgt. Der Living Place soll den Bewohner in vieler Hinsicht unterstützen. Dafür werden viele verschiedene (Context-Aware) Anwendungen erforscht und entwickelt<sup>1</sup>. Die verschiedenen Anwendungen benötigen spezifische Kontexte um zu agieren. Daher wird ein System benötigt, welches die Daten der ubiquitären Sensoren auswertet und den aktuellen Kontext der Wohnung bestimmt. Die Möglichkeit, mich mit einer Agenten basierenden Lösung für diese Anforderung auseinander zu setzen, motiviert mich sehr, da ich mich im Rahmen meiner Bachelorarbeit bereits intensiv mit BDI-Agenten und emotionalen Modellen befasst habe [Sokollek \(2009\)](#). Zu evaluieren, ob Kontext mit Hilfe emotionaler Modelle, wie z.B. dem PSI-Modell [Schneider \(2005\)](#), zu bestimmen ist, wird eine motivierende Aufgabe für die nächsten Semester und die Masterthese.

## 1.2 Inhalt dieser Arbeit

Diese Arbeit beschäftigt sich als erstes mit dem Begriff des Kontextes. Danach werden in Abschnitt 2 verschiedenen Softwarearchitekturen zur Kontextererkennung für UbiComp vorgestellt. Daraufhin werden in Abschnitt 3 ausgewählte Algorithmen für die Kontextererkennung grundlegend beschrieben. Das Ziel dieser Arbeit ist es einen Überblick über bestehende Architekturen und Algorithmen für die Kontextererkennung zu geben.

## 1.3 Begriffsklärung Kontext

Allgemein wird Kontext (lateinisch con-textus »Zusammenhang«, Plural: Kontexte) als ein Zusammenhang oder Umfeld beispielsweise eines Wortes oder einer Handlung bezeichnet, nach [Babylon](#). In Verbindung mit UbiComp wird Kontext nach [Abowd u. a. \(1999\)](#) folgendermaßen beschrieben:

„Context is any information that can be used to characterize the situation of an entity. An entity is a person, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

---

<sup>1</sup>z.B. ein intelligenter Wecker, [Ellenberg \(2010\)](#), eine intelligente Küche, [Zimmermann \(2010\)](#), eine Workflow Engine, [Tennstedt \(2010\)](#)

Aus dieser Definition folgt, dass der Kontext verschiedener Entitäten, z. B. Ort, Temperatur, Beziehung, Emotion oder eine Tätigkeit, auf den relevanten Informationen der Umgebung aufbaut und diese zu einer Situationsbeschreibung vereint, siehe [Bossard und Hert \(2006\)](#). Wenn beispielsweise die Situation der Entität Temperatur beschrieben werden soll, wird der dafür relevante Kontext eines Temperatursensors in Grad Celsius herangezogen. Nicht relevante Informationen wie z. B. die Niederschlagsmenge gehören demnach nicht zum Kontext der Temperatur.

## 2 Architekturen für Kontexterkenennung

In diesem Abschnitt werden ausgewählte Architekturen von Context-Aware Systems vorgestellt.

### 2.1 iFlat - Eine dienstorientierte Architektur für intelligente Räume

Im Rahmen des Ambient Intelligence Labor der HAW-Hamburg wird die iFlat-Software Struktur für die Verwendung im Living Place Hamburg vorgeschlagen. Diese Architektur bietet ein System, das szenarioorientierte Abläufe unterstützt und die Integration von neuen Komponenten gewährleistet. Siehe dazu [Stegemeier u. a. \(2009\)](#).

#### Architektur der iFlat-Servicekomponente

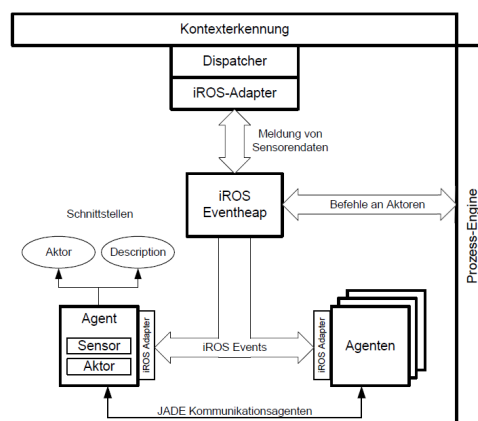


Abbildung 1: Konzeptionelle Sicht der Service-Komponente aus [Stegemeier u. a. \(2009\)](#)

Die Abbildung 1 zeigt die für den Living Place Hamburg vorgeschlagene Architektur der Service Komponente. Zentral ist hierbei der iROS-Eventheap. Er stellt einen Nachrichtenkanal zur Verfügung und verwendet eine Modifikation eines Blackboard Systems der Universität Stanford (iROS) als Nachrichtenmakler und Jade als Anwendungsprotokoll. Die Sensoragenten stellen ihre Sensordaten als iROS-Event dem Rest des Systems zur Verfügung. Nun kann basierend auf diesen Daten die Kontexterkenkung angestoßen werden. Wurde ein Kontext erkannt, kann die Prozessengine die für diesen Kontext nötigen Aktoren (Jadeagenten) anstoßen.

### Architektur iFlat - Kontexterkenkung

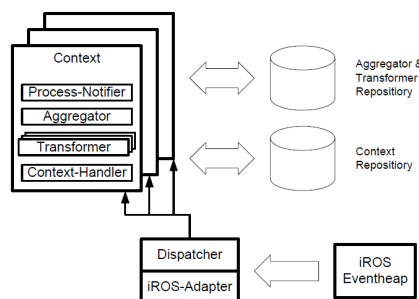


Abbildung 2: Konzeptionelle Sicht der Kontexterkenkung aus [Stegelmeier u. a. \(2009\)](#)

Die Abbildung 2 zeigt den inneren Aufbau der Kontexterkenkung des iFlat-Systems. Folgende Aufgaben lassen sich nach [Stegelmeier u. a. \(2009\)](#) der Kontexterkenkung zuordnen. Zur Entwurfszeit definiert der Entwickler die Zugehörigkeit der relevanten Sensordaten zu einem Kontext. Außerdem werden verschiedene Transformatoren festgelegt, die ein Sensordatum in ein geeignetes Format für die Kontexterkenkung überführen. Der Aggregator hat die Aufgabe, die zur Verfügung stehenden Informationen zu verschmelzen und somit ihre Heterogenität zu verbergen (Sensor Fusion). Beispielsweise könnten mehrere Positionssensoren ihre Daten im System publizieren. Der Aggregator könnte dafür sorgen, dass daraus genau eine 3D-Raumkoordinate berechnet wird, die dem restlichen System zur Verfügung steht. Wurde der Kontext erkannt, informiert der Process-Notifier die Prozess Engine über den Kontext.

## 2.2 Gaia-Context-Infrastructure

Die Gaia-Context-Infrastructure basiert auf einem formalen Modell zur Kontextbeschreibung. Auf Prädikatenlogik basierend werden Schlussfolgerungen getroffen. Einen Überblick über

Kontexterkenkung per Prädikatenlogik zeigt Abschnitt 3.1. Neben den Context Providern, die die Sensoren darstellen, und den Context Consumern, die die Anwendungen darstellen, zeigt die Abbildung 3, wie mit Hilfe sogenannter Context Synthesizer eine Aggregation verschiedener Kontexte realisiert werden kann. Der Context Synthesizer ist hierbei sowohl Context Consumer als auch Context Provider. Der Context Provider Lookup Service(CPLS) ist für die Suche von passenden Context Providern zuständig. Eine Anwendung könnte den CPLS mittels einer prädikatenlogischen Aussage fragen, ob ein dazu passender Context Provider existiert. Der CPLS gibt dann gegebenenfalls eine Referenz auf den Provider zurück.

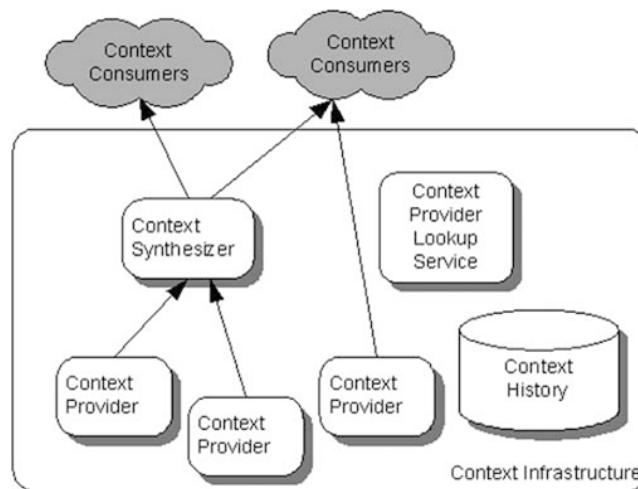


Abbildung 3: Gaia Infrastruktur für Kontexterkenkung aus [Ranganathan und Campbell \(2003\)](#)

### 2.3 Context Aware Learning System

Eine Architektur von [Lee u. a. \(2007\)](#), die das automatische Erlernen von Kontexten direkt unterstützt, wird in Abbildung 4 dargestellt. Das Ziel bei dieser Architektur ist, die Gewohnheiten der Nutzer zu erlernen. Mit diesem Wissen werden dem Nutzer Services bereitgestellt.

#### Kontext erlernen und anbieten

Als erstes verarbeitet die Context Modelling Komponente Sensordaten und stellt Kontextinformationen daraus bereit. Das zweite Teil des Systems ist der Preference Learner. Er besteht aus einer Rule Based Engine und einem Naive Bayesian Classifier. Die Rule Based Engine speichert und verwaltet Regeln. Außerdem erstellt sie neue Regeln basierend auf

dem Nutzerverhalten (Experience). Der Naive Bayesian Classifier hat die Aufgabe abhängig von den Regeln eine Service Vorhersage zu treffen, die dem Nutzer angeboten wird. Die Funktionsweise von Naive Bayesian Classifier wird in Abschnitt 3.2 behandelt.

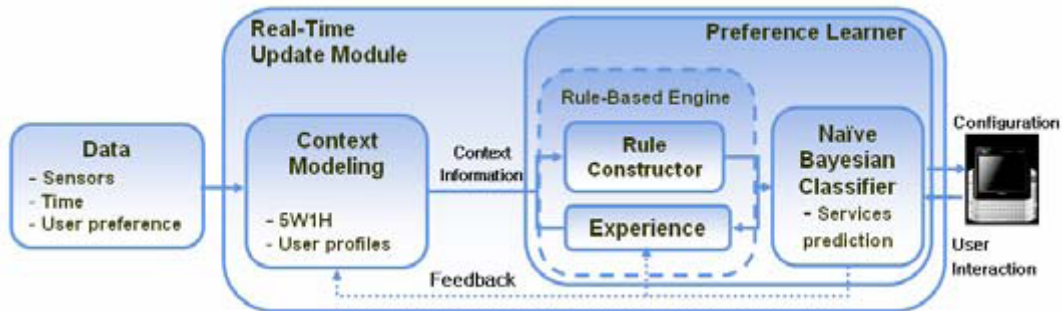


Abbildung 4: Context Aware Learning System aus Lee u. a. (2007)

## 2.4 Zusammenfassung der Architekturen für Kontexterkenkung

Es wurden drei unterschiedliche Architekturen vorgestellt, die alle unterschiedliche Eigenschaften haben. Während bei der iFlat-Architektur 2.1 die Komponierbarkeit des Gesamtsystems im Mittelpunkt steht, konzentriert sich die Gaia Context Infrastructure 2.2 ganz auf eine Aggregation der auf Prädikatenlogik basierenden Kontexte. Das Context Aware Learning System 2.3 setzt den Naive Bayesian Classifier ein um Benutzerverhalten vorherzusagen und lernt Nutzerverhalten mit Hilfe einer Rule Based Engine.

## 3 Kontexterkenkung

In diesem Abschnitt werden verschiedene Verfahren zur Kontexterkenkung vorgestellt und in ihren Grundzügen erläutert.

### 3.1 Kontexterkenkung mit Hilfe von Prädikatenlogik

Ranganathan und Campbell (2003) beschreiben die Struktur der prädikatenbasierten Kontexte. Ein Kontext ist hierbei ein Prädikat erster Stufe (first order logic (FOL)). Der Name des



Prädikats bezeichnet die zu beschreibende Entität<sup>2</sup>. Folgende Beispiele<sup>3</sup> verdeutlichen die Aussage:

- $Location(Chris, Entering, Room3231)$
- $Temperature(Room3231, =, 21^{\circ}C)$

Die Kontextprädikate werden hier über drei Argumente definiert, wobei das mittlere Argument die Beziehung der beiden anderen Objekte zueinander festlegt. [Ranganathan und Campbell \(2003\)](#) schlagen eine Ontologie vor um die gültige Struktur der Kontextprädikate festzulegen.

### Ableiten von Kontext höherer Ordnung

Durch Verknüpfung der einzelnen Kontextprädikate über logische Operatoren kann ein Kontext höherer Ordnung abgeleitet werden. Folgendes Beispiel verdeutlicht das Vorgehen:

$Sound(Room3234, >, 40dB) AND Lighting(Room3234, Stroboscopic) AND \#People(Room3234, >, 6) \Rightarrow SocialActivity(Room3234, Party)$

### Abfrage des Kontextes

Um Kontexte zu erfragen, werden die  $\forall$  und  $\exists$  Quantoren genutzt. Diese Vorgehensweise benötigt eine RuleEngine oder deklarative Programmiersprache. Folgende Beispiele verdeutlichen den Mechanismus der Abfragen:

$\exists_{Location} y Location(Chris, In, y) \Rightarrow Room3234$   
 $\forall_{People} x Location(x, In, Room3234) \Rightarrow Chris, Peter, \dots$

Die Menge der für die Quantoren einsetzbaren Objekte muss hierfür begrenzt sein, da ansonsten der Algorithmus nicht terminieren würde.

### Fazit zur Kontexterkenkung mit Hilfe von Prädikatenlogik

Die Kontexte über Prädikate zu definieren hat die Vorteile, dass sie einfach zu beschreiben sind und logische Operationen auf ihnen anwendbar sind. Außerdem wird eine klare

<sup>2</sup>Wird in [Ranganathan und Campbell \(2003\)](#) auch als Kontext bezeichnet

<sup>3</sup>Alle Beispiele in diesem Abschnitt stammen aus [Ranganathan und Campbell \(2003\)](#)

Schnittstelle für die Abfrage von Kontexten über Quantoren bereitgestellt. Allerdings ist diese Struktur wenig flexibel und benötigt die Erstellung und Pflege der Prädikate (Programmierer). Eine Erweiterung durch Lernverfahren ist nach [Ranganathan und Campbell \(2003\)](#) aber möglich.

### 3.2 Naive Bayesian Classifier

Der Naive Bayesian<sup>4</sup> Classifier ist ein statistisches Verfahren, um die Wahrscheinlichkeit (Whr) für ein Ereignis  $H$ , gegeben Beobachtung  $E$ , anzugeben. Es bietet also die Möglichkeit zur Vorhersage eines bestimmten Ereignisses für eine gegebene Situation. Wie alle statistischen Verfahren benötigt es eine angemessene Anzahl von Trainingsdaten, um diese Wahrscheinlichkeit verlässlich zu berechnen. Es basiert auf den Bayeschen Regeln:

- $Pr[H|E] = \frac{Pr[E|H]*Pr[H]}{Pr[E]}$
- *Apriori Whr von H* :  $Pr[H]$
- *Aposteriori Whr von H* :  $Pr[H|E]$

Die Naive Bayes Annahmen erweitert die Bayeschen Regeln um Klassen und Attribute:

- Whr einer Klasse gegeben Beobachtung  $E$
- Beobachtung wird in Teile (Attribute) zergliedert
- Alle Attribute sind unabhängig voneinander  $Pr[H|E] = \frac{Pr[E_1|H]Pr[E_2|H]...Pr[E_n|H]Pr[H]}{Pr[E]}$

Mit Hilfe dieser Annahmen kann jetzt die Whr für ein Klasse basierend auf einer aus Attributen bestehenden Beobachtung (z. B. Sensordaten) berechnet werden. Die Bedingung, dass die Attribute unabhängig voneinander sind, ist in der Realität nur schwer einzuhalten. Erfahrungen zeigen aber, dass die Ergebnisse auch dann aussagekräftig bleiben, wenn die Attribute eine leichte Abhängigkeit besitzen, siehe [Hinneburg \(2005\)](#).

#### Beispiel für Klassifizierung

Folgendes Beispielszenario<sup>5</sup> soll verdeutlichen, wie die Klassifizierung über Naive Bayesian Classifier funktioniert: Eine Anwendung, die die Musik im Living Place steuert, soll eine Auswahl von Musikstücken passend zur Befindlichkeit des Bewohners abspielen. Um die Befindlichkeit zu bestimmen, werden die Kontextinformationen der Schlafdauer und der Schlafqualität von verschiedenen Sensoren herangezogen. Basierend auf Tabelle 1, Tabelle 2 und Tabelle 3 wird die Whr für gute bzw. schlechte Laune berechnet.

<sup>4</sup>Thomas Bayes, Englischer Mathematiker 1702 - 1761

<sup>5</sup>Das Beispiel ist angelehnt an [Hinneburg \(2005\)](#)

Tabelle 1: Diese Daten sind bereits im System vorhanden

Schlaf	Schlafdauer	Laune
unruhig	lang	gut
ruhig	normal	gut
ruhig	kurz	gut
normal	normal	gut
normal	lang	gut
unruhig	kurz	schlecht
normal	normal	schlecht
ruhig	kurz	schlecht

Tabelle 2: Diese Whr lassen sich für die Daten berechnen

	Schlaf		Schlafdauer			Laune	
	gut	schlecht	kurz	gut	schlecht	gut	schlecht
unruhig	1	1	kurz	1	2	5	3
normal	2	1	normal	2	1		
ruhig	2	1	lang	2	0		
unruhig	1/5	1/3	kurz	1/5	2/3	5/8	3/8
normal	2/5	1/3	normal	2/5	1/3		
ruhig	2/5	1/3	lang	2/5			

Tabelle 3: Dieses Ereignis soll nun klassifiziert werden

Schlaf	Schlafdauer	Laune
unruhig	lange	???

- Wahrscheinlichkeit für gute Laune:  $gut = 1/5 * 2/5 * 5/8 = 0.05$
- Wahrscheinlichkeit für schlechte Laune:  $schlecht = 1/3 * 3/8 = 0.125$
- $Pr(„gut“) = 0.05 / (0.05 + 0.125) = 28.6\%$
- $Pr(„schlecht“) = 0.125 / (0.05 + 0.125) = 71.4\%$

Die Whr für gute Laune beträgt nach diesem Beispiel 28,6%, die für schlechte Laune 71,4%. Die Anwendung wird jetzt Musik für schlechte Laune abspielen. Bei der Berechnung der Whr der schlechten Laune fällt auf, dass die Ausprägung schlechte Laune und langer Schlaf nicht existiert. Eine Möglichkeit mit fehlenden Attributen umzugehen besteht darin diesen Faktor einfach wegzulassen. Das Ergebnis wird dadurch aber erheblich beeinflusst. [Hinneburg \(2005\)](#) bietet verschiedene Lösungsansätze, um den Einfluss fehlender Ausprägungen zu minimieren.

### Fazit zur Naive Bayesian Classifier

Das Verfahren des Naive Bayesian Classifier ist mathematisch einfach und wird häufig als gestütztes Lernverfahren in Verbindung mit Context Awareness genutzt (siehe u. a. Quel-

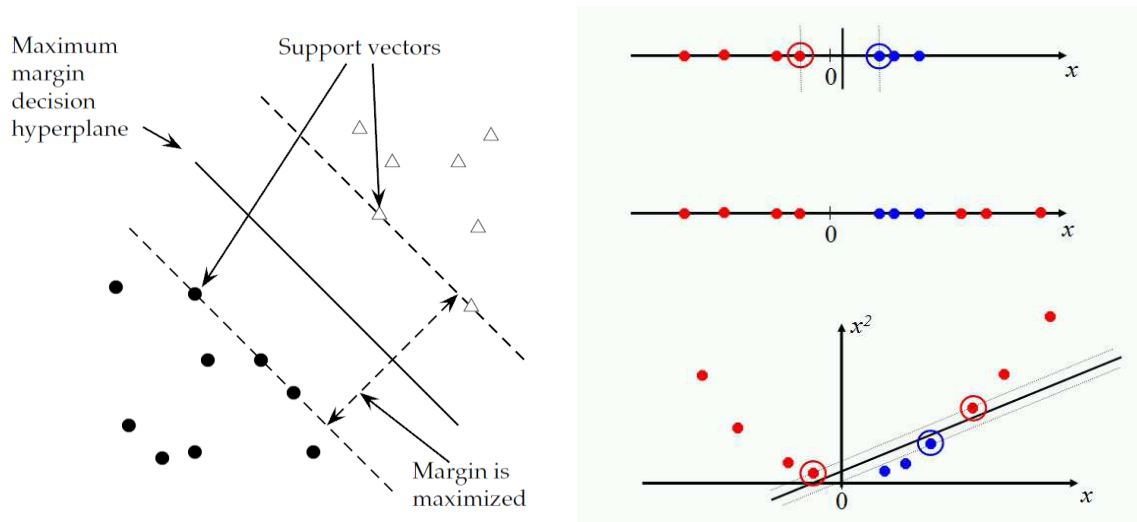


Abbildung 5: SVM - Hyperebene und nicht linear trennbare Daten, aus [Manning u. a. \(2009\)](#)

len [Brdiczka u. a. \(2005\)](#), [Ko und Sim \(2008\)](#), [Lee u. a. \(2007\)](#)). Dennoch muss entschieden werden, ab wann eine Vorhersage als gültig angesehen wird (z. B. Schwellwert >70%). Verbesserungen und Optimierungen, z. B. um kontinuierliche Ereignisräume abzubilden oder Entscheidungsbäume aufzustellen, sind möglich und werden in [Hinneburg \(2005\)](#) beschrieben. Außerdem gibt es viele Tools und APIs die den Naive Bayesian Classifier einsetzen, z. B. [www.norsys.com](http://www.norsys.com).

### 3.3 Kontexterkenkung mit Support Vector Machines (SVM)

Die SVM ist ein mathematisches Verfahren, das klassifizierte Trainingsdaten in einem Hyperraum anordnet und diesen Raum dann durch eine Hyperebene in zwei Klassen unterteilt (sinnvoll sind hier die Klassen  $A$  und  $\neg A$ ). Ist diese Hyperebene einmal aufgestellt, kann sehr schnell berechnet werden, ob eine Datenkonfiguration zu einer Klasse gehört, indem geprüft wird, auf welcher Seite der Hyperebene die Daten angeordnet sind, was die Abbildung 5 (links) verdeutlicht. Wenn keine lineare Trennung der Daten möglich ist, wird wie auf Abbildung 5 (rechts) eine Funktion auf die bestehenden Daten angewendet, um sie in einen Hyperraum höherer Dimension zu transformieren. Dieser Vorgang wird solange wiederholt, bis eine lineare Trennung möglich ist. Der mathematisch schwierige Teil folgt bei der Rücktransformation der Hyperebene auf die ursprüngliche Dimensionalität. Für tiefere Einblicke in die Mathematik der SVM, siehe [Manning u. a. \(2009\)](#).

## Fazit zur SVM

Der Vorteil der SVM besteht darin, dass eine Überanpassung durch die Maximierung der Fläche der Hyperebene und das Verstecken der Daten hinter den Stützvektoren vermieden wird. Während der Lernphase ist die Berechnung der Hyperebene aufwändig. Dafür ist die Klassenbestimmung danach sehr effizient. Diese Eigenschaft könnte gerade bei vielen zu überprüfenden Klassen (Kontexten) ein bedeutender Vorteil sein. Tools und APIs sind verfügbar.

## 3.4 Zusammenfassung Kontexterkenkung

Es wurden verschiedene Verfahren zur Kontextermittlung vorgestellt. Kontextermittlung per Prädikatenlogik bietet sich an, um Kontexte zu definieren und höherwertigen Kontext, durch die Verknüpfung von Kontexten durch logische Operatoren, abzubilden. Für die Zusammenhänge, die nicht logisch eindeutig sind, bieten sich Lernverfahren an. Sowohl Naive Bayesian Classifier als auch SVM benötigen Trainingsdaten. Während Bayes für jede Klassifizierung alle relevanten Trainingsdaten heranzieht, nutzt die SVM nur die Hyperebene zu Bestimmung der Klasse. Weitere Lernverfahren, die hier nicht behandelt wurden, können z. B. mit Neuronalen Netzen oder Fuzzy-Logic umgesetzt werden.

## 4 Fazit und Ausblick

Ziel dieser Ausarbeitung ist einen Überblick über Architekturen und Verfahren zur Kontextermittlung zu geben. Es wurde eine Auswahl verschiedener Architekturen und Verfahren zur Kontextermittlung vorgestellt. Auffällig bei der Recherche war, dass viele Insellösungen für diese Systeme bestehen, jedoch keine Standards für Context-Awareness Systems in Ubi-Comp zu existieren scheinen. Der Artikel [Soylu u. a. \(2009\)](#) macht ebenfalls deutlich, dass ein Bedarf an Zusammenführung der verschiedenen Ansätze besteht.

Für die Modellierung, Erkennung und das Erlernen der Kontexte existieren verschiedene Verfahren. Es bleibt eine spannende Aufgabe für das Sommersemester 2010, Anforderungen für die Kontextverwaltung im Living Place-Hamburg zu erheben und die verschiedenen Verfahren gegen diese Anforderungen zu prüfen und prototypisch zu implementieren. Zur Umsetzung dieser Anforderungen könnte hierbei auf eine agentenbasierte Lösung zurückgegriffen werden. Langfristig soll auch die Vision verfolgt werden, ob mittels emotionaler Systeme die Kontextermittlung erfolgreich stattfinden kann.

## Literatur

- [Abowd u. a. 1999] ABOWD, Gregory D. ; DEY, Anind K. ; BROWN, Peter J. ; DAVIES, Nigel ; SMITH, Mark ; STEGGLES, Pete: Towards a Better Understanding of Context and Context-Awareness. In: *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK : Springer-Verlag, 1999, S. 304–307. – ISBN 3-540-66550-1
- [Bossard und Hert 2006] BOSSARD, Andreas ; HERT, Matthias: Context-Awareness Institut für Informatik, Universität Zürich, Schweiz (Veranst.), 2006
- [Brdiczka u. a. 2005] BRDICZKA, Oliver ; REIGNIER, Patrick ; CROWLEY, James L.: Supervised learning of an abstract context model for an intelligent environment. In: *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*. New York, NY, USA : ACM, 2005, S. 259–264. – ISBN 1-59593-304-2
- [Chen 2005] CHEN, Annie: Context-aware collaborative filtering system: predicting the user's preferences in ubiquitous computing. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 1110–1111. – ISBN 1-59593-002-7
- [Dey und Abowd 2001] DEY, Anind K. ; ABOWD, Gregory D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In: *Cont Aw Comput-HCI J 16:97116* (2001)
- [Ellenberg 2010] ELLENBERG, Jens: Ein Wecker in einem ubicomp Haus / HAW-Hamburg. 2010. – Forschungsbericht
- [Gu u. a. 2009] GU, Y. ; TAN, S.L. ; WONG, K.J. ; HO, M.H.R. ; QU, L.: Using GA-based feature selection for emotion recognition from physiological signals. In: *Intelligent Signal Processing and Communications Systems, 2008. ISPACS 2008. International Symposium on*, Feb. 2009, S. 1–4
- [Hinneburg 2005] HINNEBURG, Alexander: *Vorlesung: Data Mining in Datenbanken*. 2005. – URL [http://users.informatik.uni-halle.de/~hinnebur/Lehre/KDD\\_WS0506\\_web/](http://users.informatik.uni-halle.de/~hinnebur/Lehre/KDD_WS0506_web/)
- [Ko und Sim 2008] KO, Kwang-Eun ; SIM, Kwee-Bo: Development of context aware system based on Bayesian network driven context reasoning method and ontology context modeling. In: *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, Oct. 2008, S. 2309–2313

- [Lee u. a. 2007] LEE, Jieun ; OH, Sanghoun ; JEON, Moongu: A New Context-Aware Learning System for Predicting Services to Users in Ubiquitous Environment. In: *ISUVR '07 5th International Symposium on Ubiquitous Virtual Reality* (2007)
- [Manning u. a. 2009] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *An Introduction to Information Retrieval*. Cambridge University Press Cambridge, England, 2009
- [Oku u. a. 2006] OKU, K. ; NAKAJIMA, S. ; MIYAZAKI, J. ; UEMURA, S.: Context-Aware SVM for Context-Dependent Information Recommendation. In: *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, May 2006, S. 109–109. – ISSN 1551-6245
- [Oyama u. a. 2008] OYAMA, K. ; JAYGARL, H. ; XIA, Jinchun ; CHANG, C.K. ; TAKEUCHI, A. ; FUJIMOTO, H.: A Human-Machine Dimensional Inference Ontology that Weaves Human Intentions and Requirements of Context Awareness Systems. In: *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, 28 2008-Aug. 1 2008, S. 287–294. – ISSN 0730-3157
- [Ranganathan und Campbell 2003] RANGANATHAN, Anand ; CAMPBELL, Roy H.: An infrastructure for context-awareness based on first order logic. In: *Personal Ubiquitous Comput.* 7 (2003), Nr. 6, S. 353–364. – ISSN 1617-4909
- [Schneider 2005] SCHNEIDER, Gordon B.: *Agenten und unsere Emotionen Ein Vergleich von Dörners PSI-Theorie mit der Emotionstheorie von Ortony, Clore und Collins*, Diplomarbeit, 2005
- [Sokollek 2009] SOKOLLEK, Wolfram: *Simulation einfacher Evakuierungsszenarien mit Hilfe intentionaler Agenten*. 2009
- [Soylu u. a. 2009] SOYLU, Ahmet ; CAUSMAECKER, Patrick ; DESMET, Piet: Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering. In: *Journal of Software* 4 (2009), Nr. 9. – URL <http://www.academypublisher.com/ojs/index.php/jsw/article/view/04099921013/1431>
- [Stegelmeier u. a. 2009] STEGELMEIER, Sven ; WENDT, Piotr ; LUCK, Kai von: iFlat - Eine dienstorientierte Architektur für intelligente Räume. In: *Proc. of the VDE 2. Ambient Assisted Living Kongress mit Ausstellung, 27.01. 28.01.2009 in Berlin*, 2009
- [Suganuma u. a. 2008] SUGANUMA, T. ; YAMANAKA, K. ; TOKAIRIN, Y. ; TAKAHASHI, H. ; SHIRATORI, N.: A Ubiquitous Supervisory System Based on Social Context Awareness. In: *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, March 2008, S. 370–377. – ISSN 1550-445X

- 
- [Tennstedt 2010] TENNSTEDT, Sven: Agentenbasierte Workflow Engine für eine intelligente Wohnung / HAW-Hamburg. 2010. – Forschungsbericht
- [Ye u. a. 2008] YE, Juan ; MCKEEVER, Susan ; COYLE, Lorcan ; NEELY, Steve ; DOBSON, Simon: Resolving uncertainty in context integration and abstraction: context integration and abstraction. In: *ICPS '08: Proceedings of the 5th international conference on Pervasive services*. New York, NY, USA : ACM, 2008, S. 131–140. – ISBN 978-1-60558-135-4
- [Zimmermann 2010] ZIMMERMANN, Nico: Intelligent Kitchen / HAW-Hamburg. 2010. – Forschungsbericht