Decision Making in Ambient Assisted Living Environments based on Uncertain and Fuzzy Data

Dominik Brunmeir[1]*, Johannes Kropf[2]

[1]Institute of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria * *dominik.brunmeir@gmail.com*

[2]AIT Austrian Institute of Technology GmbH, Donau-City-Straße 1, 1220 Vienna, Austria

Simulation Notes Europe SNE 25(3-4), 2015, 179-184
DOI: 10.11128/sne.25.tn.10313
Received: September 16, 2015; Revised October 12, 2015;
Accepted: November 20, 2015

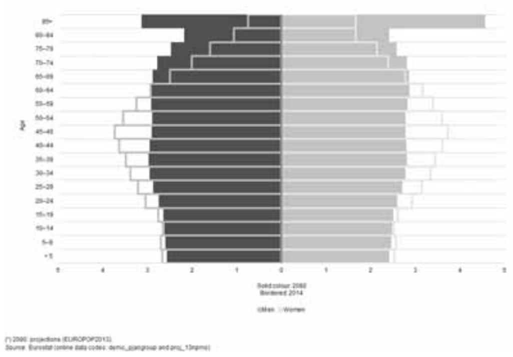
Abstract. A decision making engine for home automation in an assisted living context is presented. A Prototype was implemented, that will be integrated in HOMER, a middleware framework for home event recognition, as a reasoning component of the system. First tests show a reasonable error rate, which can be improved with careful selection of parameters.

Introduction

This article discusses a decision making engine in an *Ambient Assisted Living* (AAL) context. This decision making engine is not only able to learn and categorize human behavior by observing a person, but also replicate this person's actions. It recognizes altering behavioral patterns and adapts its actions to those new or changed situations. Its purpose is to automate certain activities to enhance the quality of life of elderly persons or persons with special needs.

Ambient Assisted Living is a growing field of research, as we are facing a severe demographic shift in Europe (see [1] and [2]) . Better health care leads to higher life expectancy and therefor significantly more people in need of assistance in their later stages of life.

Provided there is a constant number of caregivers, this leads to either more work for them, or less support for those in need of care. To compensate for this, it is a necessity to increase the efficiency of the support or keeping the elderly independent of supporting measures.



(*) 2080: projections (EUROPOP2013)
Source: Eurostat (online data codes: demo_pjangroup and proj_13npms)

Figure 1: Change of population.

AAL tries to make progress on both issues (see [3]). AAL is a general term of different supportive techniques, that can be categorized in the following way [3]:

- Sensing
- Reasoning
- Acting
- Communication
- Interaction

Sensing includes every system that collects data. This includes on-body sensors or sensors in the environment of the person. Reasoning means processing the collected data, while acting is automatic control of actuators or other feedback mechanisms. This is in contrast to communication and interaction, where a reasoning component is included. The presented decision making engine is a reasoning component.

1 Decision Making

Decision making is the process of estimating the consequences of making a decision and picking the more desirable or better choice.

There are two main issues with decision making: First we need to define and quantify the meaning of *better*, then we need set a framework for the decision making process itself. In our case, better means more similar to the desired behavior of an inhabitant, while the framework should imitate general human behavior.

Finite-state machines provide a framework to provide rule based automation, but as the number of sensors increases, they become more and more difficult to maintain in a consistent manner. It's also difficult to represent human behavior with such a strict (crisp) ruleset.

Another approach is the *Adaptive Neuro-Fuzzy Modeling* (ANFIS), which is a fuzzy inference system and better suited in learning and reproducing human behavior ([4]). It's aim is to generate an if-then rulebase with the help of artificial neural networks. Artificial neural networks were motivated by real neural networks and their ability to adapt to many problems. It consists of many connected simple units that work in parallel. The drawback of this approach is computational complexity and the difficulty of online adaptation of the rulebase, as the neural network would have to be rebuilt in case a new rule is discovered. While AOFIS generates easily interpretable membership functions with intuitive linguistic variables, ANFIS is not able to do so. For a performance comparison between AOFIS and ANFIS see [5], where it is shown, that AOFIS performs slightly better than ANFIS while also being able to adapting to user input online.

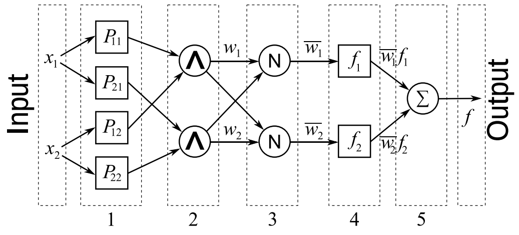


Figure 2: Schematics of ANFIS.

2 AOFIS

The Adaptive Online Fuzzy Inference System (AOFIS) was chosen because of its desirable properties. It shares certain similarities to fuzzy control systems, and thus is able to reproduce human behavior, which by itself is a strictly nonlinear problem (see [6], [7], [8] and [9]). AOFIS is also able to adapt during run time to changing behavior. It is based on a double-clustering technique, which extracts interpretable fuzzy granules. As part of this work, it was implemented in Java for further integration in the HOMe Event Recognition System (HOMER).

AOFIS itself consists of five stages of operation:

- Collection of Data
- Extraction of Membership Functions
- Extraction of Fuzzy Rules
- Application of Fuzzy Rules
- Adaptation of Output

2.1 Collection of data

The data used in the experiments are taken from various sensors and actuates in an sample smart home. The input (sensor data) is stored in a line vector $X = (x_1, x_2, ...) \in \mathbb{R}^n$, as is the output data (the actuators), $Y = (y_1, y_2, ...) \in \mathbb{R}^m$. Further we will consider the combined line vector $R = (x_1, ..., x_n, y_1, ..., y_n) \in \mathbb{R}^{n+m}$.

2.2 Extraction of membership functions

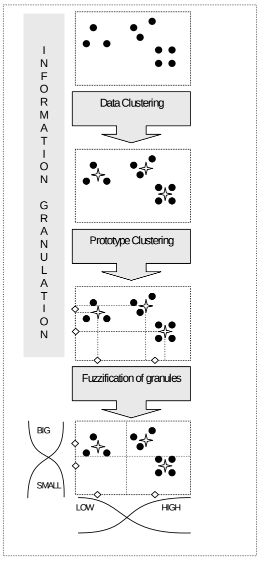


Figure 3: Extraction of Membership Functions schematics.

The Membership functions for AOFIS are generated by a double-clustering technique and extraction of so called fuzzy granules. The first clustering is performed in the $\mathbb{R}^n$ space, while the second one is along each dimension of the input space.

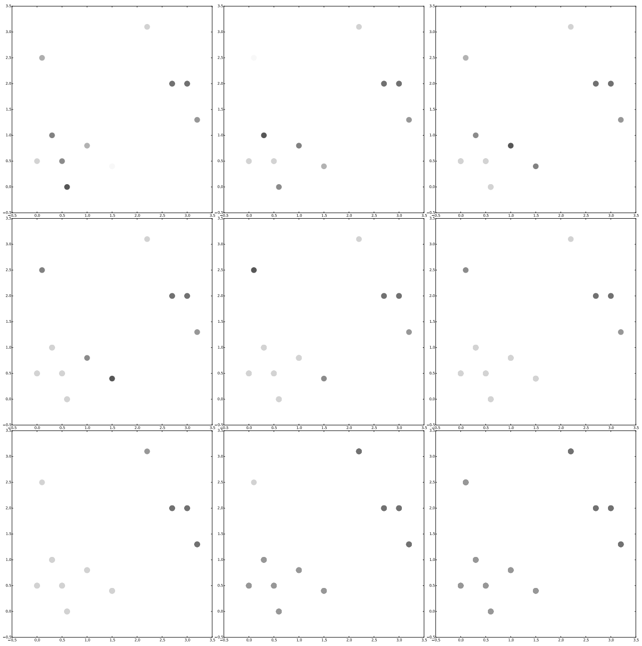


Figure 4: Agglomerative Hierarchical Clustering.

The first clustering algorithm is the Fuzzy c-Means Algorithm (FCM), which was discovered and described by Bezdek (see [10]) . FCM is a well known fuzzy clustering algorithm, where we minimize the following function to obtain cluster centers:

$$J = \sum_{j=1}^{k} \sum_{x_i \in S_j} (u_{ij})^m d(x_i, c_j)^2$$

where c_j is the j-th cluster center, x_i the i-th data vector and d a suitable (inner product-induced) norm. u_{ij} is the membership matrix, where we have the membership of every point x_i to cluster c_j.

The second clustering algorithm is an agglomerative hierarchical one. It starts with all points in a separate cluster, then takes the the two nearest clusters and merges them until the desired number of clusters is attained.

In case of our collected date, the input of the first algorithm is our input data in $\mathbb{R}^n$. The output are j n-dimensional cluster centers. The input of the second algorithm are $j \centerdot n$ one dimensional points, built from coordinates of the cluster centers, with k cluster-centers.

The next step is to fuzzify the information granules that we extracted with the double-clustering. First we partition our input space along the center of two neighboring cluster centers c_i:

$$t_i^j = \begin{cases} 2 \cdot \min(c_i^j \mid c_i^j \in C_j^l) - t_1^j & i = 0 \\ \frac{c_i + c_{i-1}}{2} & 0 < i < l_j \\ 2 \cdot \max(c_i^j \mid c_i^j \in C_j^l) - t_{k-1}^j & i = l_j \end{cases}$$

Now we can describe the input space with fuzzy sets. The chosen membership function is the gaussian function:

$$g(x) = e^{(\frac{-(x-\mu)}{\sigma})^2}$$

where μ and σ are calculated as follows:

$$\begin{aligned} \mu_i^j &= \frac{(t_{i-1}^j + t_i^j)}{2} \\ \sigma_i^j &= \frac{t_i^j - t_{i-1}^j}{2\sqrt{-2\ln\varepsilon}} \end{aligned}$$

Those sets are also easily interpretable with linguistic variables such as *high*, *medium* or *low*.

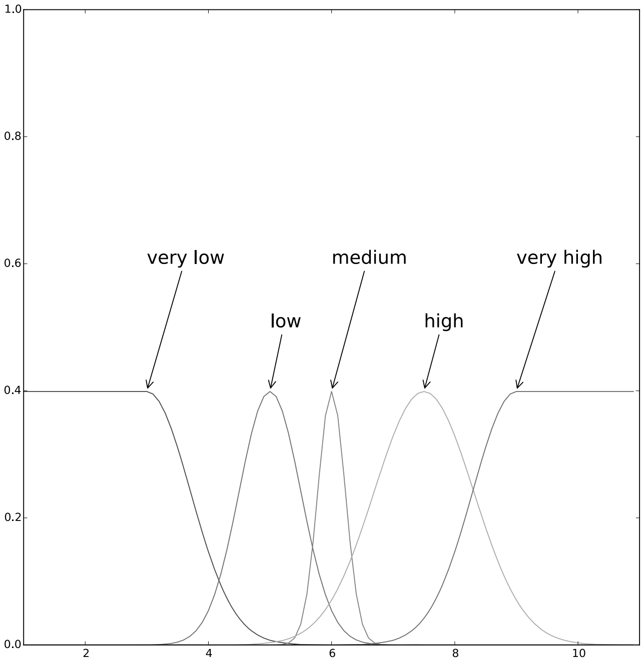


Figure 5: Linguistic Variables.

2.3 Extraction of fuzzy rules

A fuzzy or rule has the following general structure:

$$\begin{aligned} R_i : \textbf{IF } x^1 \text{ is } A_i^1 \textbf{ AND } \ldots \textbf{AND } x^n \text{is } A_i^n \\ \textbf{THEN } y^1 \text{ is } B_i^1 \textbf{ AND } \ldots \textbf{ AND } y^m \text{ is } B_i^m \end{aligned}$$

x^j are the input variables and y^j the output values, while A_i^j and B_i^j are the corresponding fuzzy sets. The set of all fuzzy rules is called a rule base. In our case the input variables is the measured sensor data, while the output values are the actuators. The fuzzy sets are the previously extracted membership functions

If we apply this scheme to the collected data, conflicting rules will be generated. To build a consistent rule base those conflicting rules have to be combined in some way. The method used is a weighted mean of all generated conflicting rules:

$$\tilde{y} = \frac{\sum_{t_k=1}^{k} y_{t_k} w_{t_k}}{\sum_{t_k=1}^{k} y_{t_k}}$$

t_k denominates all conflicting rules with the same antecedent, and w_{t_k} the weight of the t_k-th rule, with w_{t_k} as the product of all memberships of $x_{t_k}^j$ in A^j

$$w_{t_k} = \prod_{j=1}^{n} \mu_{A^j}(x_{t_k}^j)$$

2.4 Application of fuzzy rules

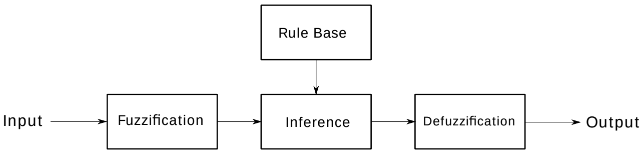


Figure 6: Procedure of the System.

With the help of the now extracted rules, the system may now control the habitat. To generate a crisp output from a crisp input, the following formula is applied:

$$y = \frac{\sum_{l=1}^{M} \bar{y}^l \prod_{j=1}^{n} \mu(x_j)}{\sum_{l=1}^{M} \prod_{j=1}^{n} \mu(x_j)}$$

2.5 Adaptation of output

This system is able to imitate a person's behavior in an habitat, but this behavior may be subject to changes as different needs arise. The person has to be able to override unwished control responses by manually controlling the system. AOFIS is able to register this and adapt its rules or add new ones. With the implementation of a delay, it is possible to filter out one-off actions.

If a manual control happens, the rule adaption routine compares each of the new input variables to those in the rule base. If a rule would have fired, the *then* part is adapted to the one, the user would have preferred, which is the one with the highest output membership value.

In case none of the existing rules would have fired, a new rule with the highest membership values for the input and the output is added.

3 Implementation and Results

3.1 Implementation

AOFIS was implemented in Java with the help of the Apache-commons-math library (see [11]) to ease integration into HOMER (see [12] for more information). The FCM and the agglomerative hierarchical clustering Algorithm had to be implemented, because at the time of writing, these algorithms were not available in the used library. The implementation was tested with various sample data sets [13] to ensure its functionality. One significant difference to the published AOFIS approach was the introduction of a threshold for replacing existing rules. We only replaced rules, that had more than 0.05 weight in the final output rule. This was done because with our sample data, AOFIS tended to also replace rules that didn't contribute much to the final output.

3.2 Results

The system is not yet implemented in HOMER, so only synthetic test could be performed. For the performance of using the system in a real smart home setting we have refer to [5] and [14].

For testing the system, we used real life data from 18 sensors in a smart home. This data set was labeled with different activities of daily life.

From this data set of 9896 input/output pairs, 3000 were taken randomly to train the system and build an initial rule base. The remaining pairs were fed to the system and whenever a mismatch between the output and the fired rule was registered, the system was adapted to simulate a manual override by the user. As parameters we were using 55 initial clusters and 9 double clusters.

As a result we were labeling almost 64% of the activities correctly. Most of the time the NULL label was incorrectly classified, followed by preparing food. Further analysis of the generated date showed, that the majority of these errors were caused by cases, which were

1	NULL
2	Comb Hair
3	Wash Hands
4	Wash Face
5	Brush Teeth
6	Make Bed
7	Change Clothes
8	Put Blinds up/down
9	Prepare Food
10	Eat
11	Open/Close Window

Table 1: Activities of daily life.

1	2	3	4	5	6
638	87	74	73	233	297
7	8	9	10	11	
90	233	589	139	38	

Table 2: Mismatched Labels using 9 double-clusters from 6896 inputs.

not yet covered by the initial rule base. These were classified as errors by the system, but showed, the system was still in the learning process to replicate the user's behavior. This is also the reason for the almost linear increase in accuracy, as those rules were added.

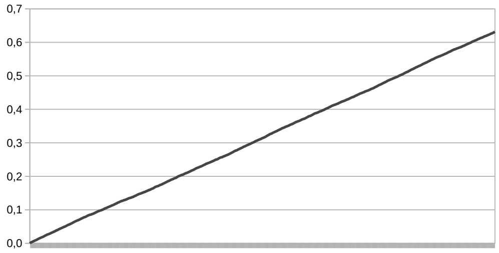


Figure 7: Ratio of mislabeled to correctly labeled data.

Our hypothesis was, that the rule base was too fine-grained, but after testing with different sets of parameters (less clusters), the success rate became worse, with a higher ratio of overrides. This makes it clear, this system needs a relatively fine grained rule system but also a big enough training set to achieve high accuracy in its task. Given enough training data, at least a 90% rate of success is achievable (see [5]).

3.3 Outlook

The aim is to integrate this system into HOMER and test it with real-life data inside a smart home. It may also be interesting to examine the structure of the input data and apply certain techniques of data mining before feeding it to the system. As there are 18^n potential antecedents for the rule base (n is the number of double clusters), maybe some dimension reduction algorithms could be applied to preprocess the data (see [15]).

4 Schematics of AOFIS Program

Algorithm 4.1: AOFIS$(inputFile, outputFile)$

```
procedure GETIO$(inputFile, outputFile)$
  $input \leftarrow generateInputData(inputFile)$
  $output \leftarrow generateOutputData(outputFile)$
  $ioList \leftarrow combineIO(input, output)$
  return $(ioList)$

procedure GETRULES$(ioList)$
  $clusterList \leftarrow DoubleClusterer.cluster(ioList)$
  $msfList \leftarrow ClusterToMSFGaussian.MSF(clusterList)$
  $ruleList \leftarrow RuleGenerator.FinalRules(msfList, ioList)$
  return $(ruleList)$

procedure CHECKRULES$(IO, ruleList)$
  $RightRule \leftarrow RuleGenerator.OutputFromInput(IO)$
  if $RightRule == IO.getOutput()$
    then return $(RightRule)$
    else $RuleGenerator.adaptRules(IO)$
  return $(IO.getOutput())$

main
  $ioList \leftarrow GetIO(inputFile, outputFile)$
  $ruleList \leftarrow GetRules(ioList)$
  if $newIO$
    then $CheckRules(newIO, ruleList)$
```

References

[1] Munstermann M. *Technisch unterstützte Pflege von morgen*. Springer. 2015.

[2] eurostat. Population and population change statistics. http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_and_population_change_statistics.

[3] Van Den Broek G, Cavallo F, Wehrmann C. *AALIANCE Ambient Assisted Living Roadmap*. Amsterdam, The Netherlands, The Netherlands: IOS Press. 2010.

[4] Jang J. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems Man and Cybernetics*;.

[5] Doctor F, Hagras H, Callaghan V. An Intelligent Fuzzy Agent Approach for Realising Ambient Intelligence in Intelligent Inhabited Environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. 2005;35:55 – 65.

[6] Zadeh L. Fuzzy Sets. *Information and Control*. 1965; 8:338–353.

[7] Zadeh L. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics*. 1973; 3(1):28–45.

[8] Zadeh L. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*. 1997;90:111–127.

[9] Cai K, Zhang L. Fuzzy Reasoning as a Control Problem. *IEEE Transactions on Fuzzy Systems*. 2008; 16(3):600–614.

[10] Bezdek J. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer. 1981.

[11] Apache. Commons Math. https://commons.apache.org/proper/commons-math/. 2015.

[12] AIT Austrian Institute of Technology GmbH - Biomedical Systems. HOMER - Home Event Recognition System. http://homer.aaloa.org/. 2011.

[13] UCI. Machine Learning Repositorz. https://archive.ics.uci.edu/ml/datasets.html. 2015.

[14] Holmes A, Duman H, Pounds-Cornish A. The iDorm: Gateway to Heterogeneous Networking Environments. *International ITEA Workshop on Virtual Home Environments*. 2002;.

[15] Maimon O, Rokach L, eds. *Data Mining and Knowledge Discovery Handbook*. Springer. 2009.