*DRAFT*

Master thesis report for the MSc Embedded Systems

TU Delft – Interactive Intelligence

User valued smart reminders: Finding Appropriate Moments for Support in Socially Adaptive Electronic Partners

Remy Kabel

4132165

Supervised by:  
Dr. M. Birna van Riemsdijk  
Dr. Myrthe Tielman

01-11-2018

# Abstract

This project will focus on finding what defines an appropriate moment in regards to providing support through a Social Adaptive Electronic Partner (SAEP). It paves the way to ultimately answering the question “Given a user’s daily activity, what is considered an appropriate time for support feed-back, taking into consideration the user’s norms and values, to achieve a certain goal?”. **TODO**

# Table of common terms

|  |  |
| --- | --- |
| **Term** | **Description** |
| ADL | Activities of daily living |
| SAEP | Socially Adaptive Electronic Partner |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Table of Contents

[Synopsis 2](#_Toc528515045)

[Table of Contents 3](#_Toc528515046)

[1 Project fundamentals 4](#_Toc528515047)

[1.1 Introduction 4](#_Toc528515048)

[1.2 Problem description 5](#_Toc528515049)

[1.3 Initial research 6](#_Toc528515050)

[1.3.1 Existing implementations 6](#_Toc528515051)

[1.3.2 Prior research 6](#_Toc528515052)

[1.4 Research description 7](#_Toc528515053)

[2 Research 9](#_Toc528515054)

[2.1 Existing implementations 9](#_Toc528515055)

[2.1.1 Comparison 13](#_Toc528515056)

[2.2 Discussion 14](#_Toc528515057)

[2.3 Feasibility 15](#_Toc528515058)

[3 Concept design 17](#_Toc528515059)

[3.1 Expectation Maximization 17](#_Toc528515060)

[3.2 Apriori algorithm 18](#_Toc528515061)

[4 References 19](#_Toc528515062)

# Introduction

The use of technology to support the daily lives of people is an ever-prevalent topic. Through applications in smart homes, wearables, virtual coaches and many others, we can improve our health, efficiency and be more connected. Conversely, the abundance of apps and notifications causes us to grow immune to the constant stream of information that is presented to us in a daily basis [1]. Especially the elderly or people with a mental impairment could benefit from an effective support agent [2]–[7]. In order to create a truly effective support agent, it is crucial to not only generate feedback in relation to the user’s actions but to provide this feedback at an appropriate time.

But what actually is an appropriate time? The appropriate time for feedback is inherently linked to the nature of the user’s action. To illustrate this, consider the following example throughout this report.

An elderly gentleman, Peter, often forgets to close the garden doors before leaving the house or going to sleep.

In this example, timely notification is of the essence. Preferably, notification just before sleeping or leaving the house is desired. Generally, these are quite predictable activities. In the current technological landscape, a simple scheduled notification would be the likely solution. Possibly a geofence[[1]](#footnote-1) may be used to trigger a notification when leaving the house, but this will be post factum.

Identifying such an appropriate time for support feedback for a specific scenario is not difficult. The difficulty of this lies in the generalization. While the above examples can be implemented relatively easy at design time, diversions from normal behavior are not handled. Existing technologies are often made by hardwiring norms and as such are very rigid and unable to adapt to evolving norms [8]. Furthermore, dealing with different problems, such as remembering to turn on the alarm system before leaving work, would require a completely different implementation. Nonetheless, generalization requires analysis of goals and the values underlying the user’s daily activities.

## Approach

The problem of finding boils down to a few steps; each worth further analysis in their own right. Working our way back, the first question that arises is what defines the goal. The goal is defined by the users and can be anything such as: *“I want my garden doors to be closed when I go to sleep or leave the house”.* Assuming we know the user’s activities of daily living (ADL), and optionally the status of the garden doors at any moment, the first step is analyzing which prerequisites there are to attaining that goal. Usually, a goal is not an independent action taken, but rather the consequence of a series of actions. As such, knowledge is required on how a goal be deconstructed into a number of distinct prerequisites.

In order to analyze arriving at this goal, some sort of model needs to be created from the user’s ADL. Once this model has been created, we can use it to analyze the limits of the possible moments for support. More directly, the prerequisites will indicate a number of actions that will have to have been completed, but also some actions may not have been completed. For example, a user will first have to arrive home, but should have received the support feedback before leaving once again, when the user will need their keys. However, Finding the most suitable moment for support is dependent on more than just this.

Finding the most “appropriate” time for the support feedback boils down to finding a moment which is both maximally effective and minimally invasive. Depending on the chosen solution, a number of other values are negatively affected. For example, sounding an alarm in the middle of a person’s sleep may be very effective, but it sure is annoying. The problem is, however, that it’s difficult to quantify invasiveness.

Summarizing, the required steps are:

* Definition of the goal and its prerequisites
* Analysis and modelling of the user’s ADL
* Analysis of effectivity
* Analysis of invasiveness

*(This should, however, be analyzed with respect to the consequences of not remembering.) In case Peter forgets before sleeping, he will either wake up with a sense of insecurity, or if he wakes up at night, he will have to get out of bed and properly interrupt his sleep. If he forgets and leaves the house, the only solutions may be to return home, to ask a friend, or to leave it be. In all cases, his value of security will be diminished, let alone if a break-in were to actually happen.* ***Dit moet nog ergens***

# Background and related work

There are plentiful existing implementations, related papers and interesting concepts. This chapter revolves around those existing and past works, in service of finding an approach to the aforementioned problems.

## Background

**Is dit echt nog nodig?**

## Existing implementations

More and more apps are taking advantage of the increased use of smart devices and services in order to get a more accurate picture of the user’s ADL. The following examples are finished

Olisto/IFTTT [9], [10] Can combine date, location and smart device information to, for example, give reminders when leaving home and a specific power consumption is still high (i.e. the TV is still on) and subsequently turn it off.

Maps/Waze [11]–[13] Combines real-time traffic information and address in calendar events to provide timely departure reminders.

Timeful [14] Combines user activity, calendar and to-do items to estimate duration of to-do items, plan them in and generate reminders at off-peak times.

While very promising implementations, most apps predominantly rely on design time logic. Exceptions to this usually create a predictive model and verify this with the user in order to strengthen the model [14], [15].

## Prior research

There have been various approaches as to how and when to provide feedback to the user. Generally, the preferred method of feedback is “smart reminders” [16]. Similar to the implementations, papers frequently focus on finding novel ways of combining information from smart devices into producing reminders, following norms provided at design time. Examples include combinations of location and time [17]–[19], events based on smart devices [3], [20], [21], or a combination of numerous sources of information [22]–[24].

The more innovative ideas add an extra logic layer on top of the data of the user’s ADL. Analyzing the user’s values is an intrinsic part of establishing a model. The concept of a Socially Adaptive Electronic Partner (SAEP) has been previously introduced by van Riemsdijk [8]. It follows the ideology that technology should adapt to the user and not vice versa. As such, its logic incorporates the norms and values of the social context. Subsequent work has been done expanding on this, including temporal logic and analyzing actions and habits. [25]–[27]. A simple but tedious approach is to ask for user feedback whenever values are needed. Instead, Zhou et al. [28] use a fuzzy linguistic approach to determine value levels.

Rather than specifying norms at design time, they are constructed based on the ADL. Several approaches are proposed. Chaminda et al. [29] suggest coupling complex activities that have a strong relationship among initiation and conclusion, such as closing the tap after opening it. Other papers [2], [30] support this analysis of temporal relationships between activities, in order to generate a set of norms for the support agent. Other context-aware approaches vary greatly. For example, Vurgun et al. [31] apply a dynamic Bayesian statistical approach. Giorgini et al. [32] use label propagation algorithms to break down goals and identify all prior actions necessary to achieve the goal.

Another approach for this makes use of Behavior Change Support Systems (BCSS) [33] by applying principles of Human Computer Interaction (HCI) [34]. This practice is used increasingly in health focused applications to make sense of the abundance of data. Examples of applications [35], [36] share large similarities with the analysis of the user’s norms and values.

# Research approach

As previously mentioned, there are several steps in finding an appropriate time for supportive feedback. However, time is limited and several aspects have already been researched plenty. As such, let us limit the focus of the thesis research.

The first two steps, goal definition and ADL analysis are all linked to activity recognition and analysis to provide smart reminders. As discussed in the previous paragraph, several models covering exactly this already exist. Each having their own properties, advantages and disadvantages.

The effectiveness and invasiveness are both quite difficult to quantify. However, they can be combined into user values. These, in return, are more quantifiable. To illustrate this, let us revisit the example of the elderly man, Peter. The goal, taking his medicine in time, drastically promotes his value of health. The moment the supportive feedback is provided, however, may demote that value or another. For example, if it causes him to wake up from his sleep, it will demote his value of health, or if it interrupts him during a phone call it may demote his value of social contact.

## Research questions

Combining the previous matters and these realizations, the focus of this thesis will be combining the concepts of a SAEP and expanding on the existing research as discussed before. The overall research question is:

How can existing smart reminder systems be extended to incorporate user values to provide appropriately timed supportive feedback and thereby increase the user values.

The expected outcome of this question is a model which provides timed feedback based on the user’s ADL and value input.

Subsequently this leads to a number of sub-questions that need to be answered before this:

R1: What are the requirements for the smart reminder system model?

R2: Which existing models and systems exist for smart reminder systems and how do they compare.

These two questions should provide a good overview on the abilities of the existing systems and the amount of work required to extend them to incorporate user values. Of course, for this we need to be able to actually find out about the user values.

R3: What are possible ways of analyzing and quantifying the values of the user?

R4: How can the model be extended to incorporate user values?

Ultimately, all knowledge can be combined into a model which can be used to approximate the most “appropriate time” for support feedback. This model can subsequently be implemented in a piece of software in order for the model to be dynamically generated depending on new input regarding the ADL, goals, norms and values. Once such an implementation has been made, the model can be tweaked according to findings and should be tested. This brings us to the final sub-question:

R5: Does the use of the extended model improve support for user values?

This will require prior planning of possible testing methods and clearly defined testing scenarios.

## Roadmap

#### Literature study

An extension of the preliminary research, focusing on papers related to answering the research questions. Specifically, this revolves around analyzing and comparing past papers and reports to see possible ways of doing activity prediction, analyzing user goals and values, and ultimately combining them. All concepts should be compared on a number of key points, to quickly establish the most valuable papers.

Subsequently, all sufficiently valuable and interesting concept should be checked for feasibility and, if necessary, reproducibility.

#### Concept design

Based on the findings of the feasibility analysis, an initial concept can be designed.

# Literature study

asjdhaksjhdkajhs

# Concept design

The initial design is based on combining the ideas of two papers, [30], [41]. Consecutively this model is combined with knowledge of user values to extend the model and allow for statistical analysis in order to predict the most ideal moment for notification. First, the individual concepts are explained and consecutively the combined design.

## Activity prediction

Activity prediction is done based on the TEREDA paper by Nazerfet et al. [30]. It focuses on two concepts to create a model for activity prediction; the Expectation Maximization [48] and Apriori [49], [50] algorithms.

### Expectation Maximization

Expectation Maximization (EM) is a clustering algorithm which works iteratively to find maximum likelihood parameters of a statistical model. It is used when such parameters cannot be solved through equations directly. The reason for this may be missing data points, latent variables, or further, still unobserved, data points are to be assumed.

Within clustering there is a division between two types: hard and soft (or fuzzy) clustering. In hard clustering, an element either belongs to a cluster or it does not. In soft clustering, on the other hand, elements can belong to multiple clusters but with different degrees of belief, or confidence. In order to statistically analyze soft clustering, mixture models can be used.

Mixture models are a probabilistically sound way of analyzing soft clustering cases. With this method, each cluster is described as a generative model[[2]](#footnote-2), such as a Gaussian or multinomial. However, the parameters of the model are unknown (for example the mean and covariance in the case of a Gaussian model).

If the source cluster of each observation is known, the estimation of these parameters is trivially done through a simple calculation. However, even when not knowing the source, as is the case in a clustering problem, the EM-algorithm will guess the cluster each point likely belongs to. This is done by using the Baysal formulae, those of conditional probability. However, in order to use these formulae, the parameters of the models need to be known. This leads to a “chicken and egg” problem. The algorithm works on any n-dimensional dataset by first performing a random estimate (expectation) to the initial parameters and iteratively improving (maximizing) them.

### Apriori algorithm

The Apriori algorithm is a machine learning algorithm used to find patterns in large data sets. Specifically, the patterns of frequent item sets. At its core it attempt to identify frequent item sets in order to generate association rules used to describe general trends in the data. The algorithm finds its roots in analyzing and predicting store transactions to find products frequently bought together.

Every transaction, or customer purchase if looking at the example of a store, is logged in a database. Consequently, a breadth-first search is done to find all items having been purchased at least a percentage of times; the threshold or support. These individual items are extended to larger and larger item sets, given those item sets appear sufficiently often in the database. Using these frequent item data sets, association rules can be generated. The association rules can be described using numerous measures. Among others, there are confidence, lift and conviction [51].

Firstly, the confidence of an association rule indicating X leads to Y, or , is the indication of how often the rule has found to be true. The previously defined support, the indication of how often an item set appears in the data set, can be described as:

Where is a transaction within the database of all transactions . As a result, the confidence of the rule is the proportion of transactions that contain set X, that also contain set Y:

Where is the union of the items in the two sets. Rewritten in probabilities, the support can be seen as simply the probability of an event , where is a transaction containing item set X. However, since regards the items in a set, it can rather be written as . Linking to Bayesian formulae, the confidence can be seen as an estimate of the conditional probability . The drawback of the confidence measure is that it only takes the popularity of itemset X into account.

The lift measure takes both item sets into consideration and compares their dependence to each other to that expected if they were independent of each other. It is defined as:

A lift of 1 would indicate that occurrences of X and Y are independent of each other and thus no rule can be drawn. The higher the value is above 1, the larger the degree in which the occurrence of Y is dependent on that of X and as such is potentially more useful for prediction. Note that a lift below 1 actually indicates that X and Y have a negative impact on each other.

Lastly, the conviction of a rule is an indication of the frequency of an incorrect prediction. It is defined as:

For example, a conviction value of 1.2 indicates that an incorrect prediction occurs 20% more often than if the association was simply by random chance.

The process of the Apriori focuses on first finding all possible datasets which have a minimum support and then creating rules based on the confidence. Depending on the implementation, either just the confidence can be used as a baseline for the rule generation, or a number of measures more. Note that there are more measures of interestingness than just those described above, including, but not limited to, collective strength [52] and leverage [53]. In [30], however, none of the measures other than the confidence are used, which will as such be the starting point for this concept.

The main drawback of the Apriori algorithm is that given the bottom up approach, a large number of subsets are required to be generated. As such, the number of database accesses are very high requiring it to be loaded into memory entirely. Furthermore, the time complexity is obviously very high. Consequently, numerous improved algorithms have been suggested. However, its simplicity makes it much easier to implement on any sort of database. This is interesting because whereas the algorithm is initially only interesting for true transactional databases such as those resulting from stores, the Apriori algorithm can be used to find patterns in any sort of data set.

In the case of [30], the Apriori algorithm is used to analyze following activities given the cluster of the current activity, as previously found using the EM algorithm.

## Value based design

WIP

## Concept description

adsdasd

# Implementation

The implementation is a major aspect in this report. In order to test the proposed concept, a number of things have to be done. First, a suitable platform has to be chosen. This platform should not only allow for all desired datasets to be supported, but preferably also allow for connection to a real-life application for field testing. Secondly, the algorithms of the conceptual design have to be implemented in code and linked to one another and to the data sources. Lastly, the implementation should provide some sort of reporting mechanism which allows analysis of the results.

## Platform

What platform to choose isn’t just dependent on what algorithm is chosen, or what libraries are available. More important is to see how the data is obtained. Keeping an open mind as to where data can come from, and not just restricting oneself to using premade datasets, allowing streaming data is important. Why? Because of the rapid rise in Internet of Things devices.

### Internet of Things

The field of activity recognition is a rapidly evolving one. This is mainly due to the exponential rise in Internet of Things (IoT) devices. Currently, there are over 17 billion connected devices in the world. Of these, there are over 7 billion IoT devices (so excluding smartphones, computers and similar) with over 6.5 million new devices being connected every day [54]. This is expected to grow to between 20 and 200 billion within the next five to ten years. The promise of IoT doesn’t end at just connecting the devices to the internet. It is just the first step.

Advances in RF technology and low power computing will bring Internet-connectivity everywhere. Advances in Big Data and machine learning will unlock new business opportunities and models. The possibilities are nearly endless, but they all still lie quite out of reach from the direct consumer. However, specifically for activity recognition, suddenly a lot more data is available than there was 10 years ago. Consequently, more and more papers and implementations such as **<fill in references>** are analyzing activity based on random sensor data.

Whether the activity data or the sensor data is available, in any case a prediction can be made on past events. As long as the event corresponding to the deadline is known before which the notification should have been presented, any form of data should fit within the design. As such, a server based solution, preferably in the cloud, seems most logical.

### Programming language

When it comes to implementing machine learning algorithms, there are several go to languages. The five most used languages [55], in order, are:

* Python
* C/C++
* Java
* R
* JavaScript

While there are many other options, they fall below a 5% mark of prioritization in the field of machine learning. Python takes the clear lead in this field. This is due to the large number of readily available libraries. This dramatically decreases the time required to implement machine learning algorithms in applications. However, regardless of popularity it is shown that professional background is key to choosing a language.

For now ignoring the fact of whether the programmer has any existing proficiencies, it is important to note that there is no best language to use for machine learning and it is important to take the goal into consideration. In this case the goal is to create a server based cloud platform. Whereas the algorithms can still be run on any language, the web part and a possible API[[3]](#footnote-3) interface are likely to be implemented in JavaScript.

### Set-up

Taking the above choices into consideration and looking at the current professional landscape, there is a single, simple way forward.

Node.js

# System architecture

# References

[1] T. Okoshi, H. Nozaki, J. Nakazawa, H. Tokuda, J. Ramos, and A. K. Dey, “Towards attention-aware adaptive notification on smart phones,” *Pervasive Mob. Comput.*, vol. 26, pp. 17–34, Feb. 2016.

[2] L. S. Shafti, P. A. Haya, M. García-Herranz, and X. Alamán, “Personal Ambient Intelligent Reminder for People with Cognitive Disabilities,” in *Ambient Assisted Living and Home Care*, 2012, pp. 383–390.

[3] J. K. Zao, M. Y. Wang, P. Tsai, and J. W. S. Liu, “Smart phone based medicine in-take scheduler, reminder and monitor,” in *The 12th IEEE International Conference on e-Health Networking, Applications and Services*, 2010, pp. 162–168.

[4] A. Arcelus, M. H. Jones, R. Goubran, and F. Knoefel, “Integration of Smart Home Technologies in a Health Monitoring System for the Elderly,” in *21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW ’07*, 2007, vol. 2, pp. 820–825.

[5] W. Jih, J. Y. Hsu, and T.-M. Tsai, “Context-Aware Service Integration for Elderly Care in A Smart Environment,” 2006.

[6] N. Mitabe and N. Shinomiya, “Support system for elderly care with ambient sensors in indoor environment,” in *2017 Eleventh International Conference on Sensing Technology (ICST)*, 2017, pp. 1–4.

[7] M. Neerincx, M. Tielman, C. Horsch, W.-P. Brinkman, K. Bosch, and R. J. Beun, “Virtual Health Agents,” 2015.

[8] M. B. van Riemsdijk, C. M. Jonker, and V. Lesser, “Creating Socially Adaptive Electronic Partners: Interaction, Reasoning and Ethical Challenges,” in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, Richland, SC, 2015, pp. 1201–1206.

[9] “Olisto makes smart thing smarter, according to your rules.,” *Olisto*. [Online]. Available: https://olisto.com/. [Accessed: 19-Apr-2018].

[10] IFTTT, “IFTTT helps your apps and devices work together.” [Online]. Available: https://ifttt.com. [Accessed: 19-Apr-2018].

[11] “Maps - Navigation & Transit - Apps on Google Play.” [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=en. [Accessed: 19-Apr-2018].

[12] “Free Community-based GPS, Maps & Traffic Navigation App | Waze.” [Online]. Available: https://www.waze.com/en. [Accessed: 19-Apr-2018].

[13] Peter G. Chin, “Smart reminders,” 25-Mar-2005.

[14] “Timeful,” *Internet Archive*, 02-Mar-2015. [Online]. Available: https://web.archive.org/web/20150302091124/http://www.timeful.com/. [Accessed: 19-Apr-2018].

[15] N. Clarkson, “Spotify to predict the music you want to listen to,” *Virgin*, 19-Nov-2014. [Online]. Available: https://www.virgin.com/music/spotify-to-predict-the-music-you-want-to-listen-to. [Accessed: 19-Apr-2018].

[16] F. Kargl, B. Dong, T. Illmann, and M. Weber, *Smart Reminder - Personal Assistance in a Mobile Computing Environment*. 2002.

[17] A. Robertson, “Location/time-based reminder for personal electronic devices,” 06-Dec-2000.

[18] Jason F. Hunzinger, “Location specific reminders for wireless mobiles,” 15-Nov-2001.

[19] Michael Sean McGee, Michael S. McIntyre, and James Randall Walker, “Generating an alarm based on location and time,” 17-Apr-2003.

[20] S. W. Kim, M. C. Kim, S. H. Park, Y. K. Jin, and W. S. Choi, “Gate Reminder: A Design Case of a Smart Reminder,” in *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, New York, NY, USA, 2004, pp. 81–90.

[21] S. Helal, C. Giraldo, Y. Kaddoura, C. Lee, H. El Zabadani, and W. Mann, “Smart Phone Based Cognitive Assistant,” Apr. 2018.

[22] D. Zhang, M. Hariz, and M. Mokhtari, “Assisting Elders with Mild Dementia Staying at Home,” in *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2008, pp. 692–697.

[23] M. Philipose *et al.*, “Inferring activities from interactions with objects,” *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 50–57, Oct. 2004.

[24] A. Hristova, A. M. Bernardos, and J. R. Casar, “Context-aware services for ambient assisted living: A case-study,” in *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies*, 2008, pp. 1–5.

[25] M. S. Kließ and M. B. van Riemsdijk, “Requirements for a Temporal Logic of Daily Activities for Supportive Technology.”

[26] P. Pasotti, M. B. van Riemsdijk, and C. M. Jonker, “Representing human habits: towards a habit support agent,” in *Proceedings of the 10th International workshop on Normative Multiagent Systems (NorMAS’16)*, 2016.

[27] P. Pasotti, C. M. Jonker, and M. B. van Riemsdijk, “Towards a formalisation of Action Identiﬁcation Hierarchies∗.”

[28] S. Zhou, C.-H. Chu, Z. Yu, and J. Kim, “A context-aware reminder system for elders based on fuzzy linguistic approach,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9411–9419, Aug. 2012.

[29] H. T. Chaminda, V. Klyuev, and K. Naruse, “A smart reminder system for complex human activities,” in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, 2012, pp. 235–240.

[30] E. Nazerfard, P. Rashidi, and D. J. Cook, “Using Association Rule Mining to Discover Temporal Relations of Daily Activities,” in *Toward Useful Services for Elderly and People with Disabilities*, 2011, pp. 49–56.

[31] S. Vurgun, M. Philipose, and M. Pavel, “A Statistical Reasoning System for Medication Prompting,” in *UbiComp 2007: Ubiquitous Computing*, 2007, pp. 1–18.

[32] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, “Reasoning with Goal Models,” in *Conceptual Modeling — ER 2002*, 2002, pp. 167–181.

[33] H. Oinas-Kukkonen, “A foundation for the study of behavior change support systems,” *Pers. Ubiquitous Comput.*, vol. 17, no. 6, pp. 1223–1235, Aug. 2013.

[34] R. Klaassen, “HCI Perspectives on Behavior Change Support Systems,” Feb. 2015.

[35] A. Fritzen, N. Leipold, N. Terzimehic, M. Böhm, and H. Krcmar, “HeadacheCoach: Towards Headache Prevention by Sensing and Making Sense of Personal Lifestyle Data,” 2017.

[36] E. S. Poole, “HCI and mobile health interventions,” *Transl. Behav. Med.*, vol. 3, no. 4, pp. 402–405, Dec. 2013.

1. A virtual geographic boundary, defined by GPS or RFID technology, that enables software to trigger a response when a mobile device enters or leaves a particular area. [↑](#footnote-ref-1)
2. In machine learning (and other forms of statistical classification) there are two main approaches: generative and discriminative. Given a target Y and an observation X, the generative model is a statistical model of the joint probability distribution. Whereas the discriminative model looks at conditional probability of Y given X=x. [↑](#footnote-ref-2)
3. See section **<fill in later>** [↑](#footnote-ref-3)