2014-9

NudgeAlong: A Case-Based Approach to Changing User Behaviour

Eoghan O'Shea
Technological University Dublin

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

Rob Lane
Climote

See next page for additional authors

Recommended Citation

Eoghan O'Shea, Sarah Jane Delany, Rob Lane &Brian Mac Namee, (2014) NudgeAlong: A Case Based Approach to Changing User Behaviour. *International Conference on Case-Base Reasoning 2014,* Cork, Ireland. LNCS 8765, 345-359 doi:10.1007/978-3-319-11209-1_25

Funder: CeADAR

Authors
Eoghan O'Shea, Sarah Jane Delany, Rob Lane, and Brian Mac Namee

NudgeAlong: A Case Based Approach To Changing User Behaviour

Eoghan O'Shea[1], Sarah Jane Delany[1], Rob Lane[2], and Brian Mac Namee[1]

[1] Applied Intelligence Research Centre,
Dublin Institute of Technology (DIT), Ireland
[2] Climote, Ireland
{eoghan.oshea,brian.macnamee,sarahjane.delany}@dit.ie,rlane@climote.ie

Abstract. Companies want to change the way that users interact with their services. One of the main ways to do this is through messaging. It is well known that different users are likely to respond to different types of messages. Targeting the right message type at the right user is key to achieving successful behaviour change. This paper frames this as a case based reasoning problem. The case representation captures a summary of a user's interactions with a company's services over time. The case solution represents a message type that resulted in a desired change in the user's behaviour. This paper describes this framework, how it has been tested using simulation and a short description of a test deployment.

Key words: Case Based Reasoning, Recommender Systems, Simulation

1 Introduction

In almost all industries businesses make attempts to change the behaviours of their customers. In some cases this is driven by a selfish motivation on behalf of the business - for example, encouraging customers to use a product or service in a particular way so as to earn maximum profit for the business - but it can also be driven by more socially aware motivations - for example, utility companies encouraging customers to use resources more efficiently. Fogg [1] proposes that in order for someone to be successfully convinced to change their behaviour to a new target behaviour that person must (1) have the ability to perform the target behaviour, (2) be sufficiently motivated, and (3) be triggered to perform the new behaviour. Fogg [2] uses the term *captology* to refer to the use of technology to persuade people to change their behaviours.

Many digital consumer scenarios (such as home utilities and the use of online services) are particularly attractive for the application of captology for two reasons. Firstly, at almost all times users have the ability to perform a target behaviour - for example a user can almost always visit a website or interact with their home utility service. Secondly, the current behaviour of users can be constantly and accurately monitored. This monitoring can offer insights into users' motivations and how open they are to behavioural change triggers. Furthermore, constantly monitoring user behaviour allows a system to determine when behaviour has changed.

To actually effect changes in behaviour, particularly in digital consumer scenarios in which large numbers of users are involved and personal communication is not possible, mass communication with users via SMS, email or other channels is frequently used. This introduces a tension between motivating users and providing them with the right triggers for behaviour change. Different types of communication are likely to motivate and trigger different users. For example, if the goal is to encourage users of a home heating control system to reduce the amount of heating that they use, some users might respond best to a message that focuses on the environmental impact of using excessive home heating, while others are likely to respond better to a message describing the financial benefits of using less home heating energy. This suggests that a degree of personalisation is required so that each user can be sent personalised messages that are most likely to motivate and trigger them to change their behaviour to a desired target.

In this paper we describe the *NudgeAlong* system which uses techniques from *case based reasoning* (CBR) to automatically send users personalised messages to motivate them to use a product or service in a particular way. To do this the system first builds a profile of each user based on information about their usage of the relevant service or product, as well as any available personal details (for example age or gender). When the system determines that a message should be sent to a user this profile is used to retrieve a set of most similar users from a case base containing details of user profiles and previous messaging attempts. The message type that was most effective at encouraging behaviour change for these retrieved users is then used for the current user. Monitoring the user's subsequent behaviour allows the system to determine whether or not the message effectively encouraged behavioural change and this result can be added to the case base. This is a typical CBR cycle and makes CBR an attractive solution to this problem.

Evaluating systems like NudgeAlong is notoriously difficult as actual user interactions are required to determine whether or not behaviour has changed. To address this issue we use a simulation approach to evaluation. Users of a service are simulated to be more or less responsive to different types of motivating messages, and we measure how effectively NudgeAlong can personalise messages so as to take this into account. While an evaluation based on this type of simulation is not as compelling as a real deployment it does provide useful results that can help in preparing for a real deployment. We also describe an initial pilot deployment that has been undertaken with a partner company in the smart-connnected-home space, Climote (`www.climote.com`).

The remainder of this paper proceeds as follows: in Section 2 we give a general overview of the approach taken in NudgeAlong; in Section 3 we present an evaluation of NudgeAlong using simulated data and briefly describe a test deployment; in Section 4 we discuss some related work; and, finally, in Section 5 we present our conclusions.

2 The NudgeAlong System

Figure 1 shows the key components of the NudgeAlong system. User data (coming, for example, from a company's monitoring of a user's interaction with their product) is collected to build a profile of each user and these profiles are fed into a Recommendation Engine. The Recommendation Engine, which is built on a case based reasoning system, uses these profiles and past history about the success or failure of attempts to change users' behaviours, to recommend which type of message will be most likely to encourage a particular user to change their behaviour. Message types are broad categories of messages such as financial, environmental, gamification or motivational.

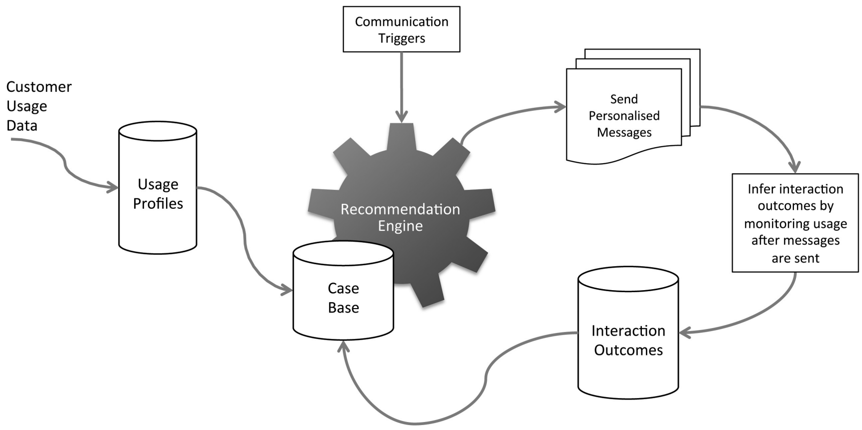

Fig. 1. The key components of the NudgeAlong system.

Each case in NudgeAlong represents an instance of a message, of a particular message type, being sent to a specific user and successfully leading to behaviour change. In the current version only details of messages that successfully changed a user's behaviour are stored in the case base. The case representation in NudgeAlong consists of two components: a user profile, containing information on a user's interaction with the system in question and any available personal information, and the type of message sent. The user profile includes features that describe how a user has been interacting with a particular service in the time leading up to the message being sent and varies based on application. Typically the user profile will include features such as how many times a user has interacted with a system in the recent past and how many of the different interaction types that a system allows that a user has recently performed.

For case retrieval the similarity measure we use is a simple Euclidean distance. We choose cases to reuse from the case base by selecting the k nearest neighbours, where, following [3], k is chosen as being the square root of the number of complete cases. Within the k nearest neighbours we use majority voting to choose the most commonly occurring message type. Before being reused, however, we perform a test to see if this message type has recently failed to convince a user to change their behaviour. If this is the case an alternative message type is randomly chosen instead. An alternative would be to use a different message type from the k most similar cases. However, the use of random messages ensures that the system is more dynamic, ensuring users are more likely to receive a variety of message types, particularly in the early stages, so leading more quickly to a situation where users are receiving the most appropriate message type.

The case based system just described determines what type of message to send to a user when the NudgeAlong system has determined that a message should be sent. The triggers shown in Figure 1, are responsible for determining when a message should be sent to a user. These triggers are simple rules that fire if a user's behaviour has diverged sufficiently from expected target behaviour. For example, a message might be sent to a user if they are not using a service a sufficient number of times per week, or if the distribution of a user's use of the different services offered by a company is not optimal. The triggers also include a restriction that imposes an upper limit on the number of messages that users can be sent per week so as to avoid users being bombarded with messages. The trigger rules are evaluated on a regular basis and users who should be sent a message are selected.

Once a user has been deemed to require a message the Recommendation Engine is used to select the message type most likely to encourage that user's behaviour to change. After this message has been sent, the user's subsequent behaviour is monitored over a period of time to determine whether their behaviour has changed or not. The definition of message success or failure is dependent on the targeted behavioural change and is therefore application dependent. Typically, however, a successful behavioural change is signified by a user changing the frequency of their interactions with a particular aspect of a service within a given time window.

3 Evaluation

Evaluating systems like NudgeAlong is notoriously difficult as real user interactions are required to determine whether or not behaviour has changed and, if so, to what extent. To address this issue we primarily use a simulation approach to evaluation. In this section we first outline the simulation framework that we have designed to perform these evaluations and then present our evaluation results. Finally, we provide a short description of a small, live deployment experiment that we have performed with Climote.

3.1 Simulation Framework

Our simulation framework generates daily simulated interactions with a service for a group of simulated users. We assume that any service that NudgeAlong will work with will offer a number of different interaction types that users can potentially perform each day. We simulate a number of different user segments, each of which is characterised by varying likelihoods of performing each of the available interactions each day. These likelihoods are defined by normal distributions of different means and standard deviations. To simulate a user's behaviour we draw a random number of interactions from the appropriate distribution each day for each user. These interactions are then summarised to populate the user profiles, that in turn populate the case base. The number of user segments simulated and the likelihood of users in each segment performing each of the possible interactions each day are set as part of the design of the simulation and vary for different application scenarios.

For each user segment we also predefine the likelihood that a particular message type will result in a user changing their behaviour. Each message type defined in NudgeAlong is assigned a different probability of producing a behavioural change for users in each user segment. When a user is sent a message in the simulation we randomly determine whether or not that message will result in their behaviour changing based on these probabilities. To simulate that a user's behaviour has changed we simply adjust the distributions that determine the likelihoods of a user performing different interactions.

In order to make the simulation more realistic, we also simulate a degree of *satiation*, an important issue in captology research. Satiation refers to the tendency of some users to revert back to their original behaviours after their behaviour has been successfully changed to a new target. In our simulation, we examine two different scenarios: (1) a situation where *every* user whose behaviour has been successfully changed is eventually assumed to revert back to their pre-existing behaviour; and (2) a situation where each user is allocated a satiation probability and only those users with a probability greater or equal to 50% revert back to their pre-existing behaviour. To simulate satiation, a user that is assumed to have gone back to their old behaviour has their interaction likelihoods reset to their original distributions after a pre-set period of time.

In the next section we will illustrate how this simulation framework has been used to evaluate the NudgeAlong system.

3.2 Simulation Evaluation

In order to test the viability of NudgeAlong, we evaluated it using the simulation framework described in the previous section. The simulation was modelled using data collected over a one year period from the remote-controlled home-heating system developed by Climote[3].

Climote allows a user to control their home heating system at a distance, either online, through a smartphone app, or by SMS messaging. In our simulation we allow users 5 separate interactions with the Climote system: *Program* (programming the heating to come on at a particular time), *Boost* (a short-term ad-hoc request for heating, e.g. for 30 mins, 1 hour at a time), *Schedule*

[3] http://www.climote.com/

(a more comprehensive, overarching heating schedule covering larger periods such as Spring, Summer, Autumn or Winter), *Temp* (an adjustment of the thermostat) and *Hold* (a suspension of heating for a set period). This is a slight simplification of the interactions that Climote actually offers users.

Corresponding to their use of these 5 interactions, each user has a usage profile built for them that forms the basis of the NudgeAlong case base. This profile stores the number of times a user has used each of the 5 different interactions over the last 7 days as well as the average number of times per week that the user has used each of these interactions since the simulation began. We experimented with a number of other metrics in the profile, e.g., average weekly values, weekend averages, etc. but it was found that this small set of features was sufficient in order to differentiate users in different segments.

The behaviour change sought in this example is to increase users' overall usage of the Climote system and so the total number of interactions performed by each user per week is used as the basis for the message trigger in the simulation. This, relatively simple, behaviour change was of interest to Climote as user engagement with the Climote system is one of their key metrics. If the total number of interactions that a user performs decreases or remains static, then a message is sent out to *nudge* a user back to the desired behaviour. In this simulation, we define a sent message as being successful if it is followed by an increase in a user's total number of interactions with the system that is maintained for a period of time.

In this evaluation we simulated 100 users. Each of the 100 simulated users were assigned to one of 5 different pre-defined user segments (20 users were assigned to each segment). To define these segments we performed a k-means clustering of user profiles generated from real Climote usage data that covered a period of one year. This segmentation revealed 5 distinct segments: *Power Boosters* (PB), *Power Programmers* (PP), *Casual Boosters* (CB), *Casual Programmers* (CP) and *Others*. Power Boosters are considered to use a larger number of Boosts compared to Program interactions, while Power Programmers are considered to use a greater number of Program interactions compared to Boost interactions. In both cases, the user's use of either Program or Boost far outstrips their use of any other command type and overall system usage is high. Casual Boosters and Casual Programmers are considered to use more Boosts and Programs, respectively, than other interactions, but the difference is not as stark as for the Power Boosters/Programmers and overall usage of the system is lower. The Others segment is defined as those users that do not fall neatly into the other 4 segments, e.g. those that use more Schedule and Hold interactions compared to the other types.

Depending on which of the 5 segments a user is in, values, corresponding to command use in our simulated system, were randomly sampled from normal distributions with means corresponding to that segment. For example, for the Power Programmer segment, values were sampled from 5 normal distributions, each with a different mean value, representing Boost, Program, Schedule, Temp and Hold command usage, respectively. For the Power Programmer segment, we gave a higher value mean to the normal distribution representing the values for the Program command, indicating the preference for users in this segment for this command type. In the same vein, for the Power Booster segment, we gave a higher value mean to the normal distribution representing the values for the Boost command. The distributions for the other segments are determined in a similar way.

For our simulation we chose four different message types: Email1, Email2, Text1 and Text2, corresponding to two different types of email messages and two different types of SMS messages. For example, the Email1 message type could be financial messages, giving details of offers, savings, price reductions, etc., while the Email2 message type could be motivational messages such as “Use Climote more!”, etc. We note that in this simulation, by using the names Email1, Email2, etc., we are simulating not only the message-type, i.e., financial-type, etc. but also the effectiveness of using a different medium to deliver that message, i.e., email or SMS messaging. This is intentional.

In the simulation, each message type has a probability of success allocated to it for each user segment. These probabilities were not based on the actual Climote data, but were chosen to ensure that a higher value (above 50%) was allocated to a preferred message type for each segment. For example, we choose that for Power Programmers the preferred message type Email2 will have a 57.5% probability of causing a change in behaviour, with Email1, Text1 and Text2 having lesser probabilities of causing a change at 25%, 25% and 10%, respectively. Table 1 shows the probabilities chosen for each segment.

Table 1. Probabilities that a message type will succeed in producing a behavioural change

Segment	Email1	Email2	Text1	Text2
PP	25%	57.5%	25%	10%
PB	25%	15%	20%	60%
CP	70%	10%	12.5%	25%
CB	67%	22%	25%	20%
Other	22%	6%	60%	20%

As mentioned in Section 2, once a message is sent out, the system waits for a specified amount of time before determining whether that message was a success or a failure; in this case has the total number of Climote interactions increased. In this simulation we expect behaviour change to happen at most within 7 days of a message being sent. Within the simulation we also ensure that users will at most receive a message once every 7 days to avoid overloading users. As described in Section 3.1, the simulation is initially set up so that every simulated user experiences satiation (scenario 1). Each user is randomly assigned a satiation value in the range of 7–14 days (following a uniform distribution).

In order to test the simulation, it was set to run through 6 months' worth of simulated user interactions with the Climote home-heating system, one day at a time. The first 7 days of the simulation are considered a bootstrapping phase during which the user profiles are populated and no messages are actually sent out. On the 7^{th} day, a random message was sent to all users in order to initially *seed* the simulation. These initial message types are chosen randomly in the expectation that some of the messages will cause a positive behaviour change. If a message type produces a successful behaviour change, a case is generated for it combining the user profile at the time the message was sent and the successful message type. The case is then added to the case base. Subsequently, as the simulation moves on in time, and users are found to be not behaving as desired, the most appropriate message type for each user in each segment can then be selected using the case base. Eventually the system will begin to choose the *correct* or optimum message types for users in each segment. From Table 1, the optimum message types for each segment are those with the highest probabilities.

As well as simulating the behaviour of users who received messages from NudgeAlong, we also included a control group of users who received randomly selected messages. Whenever the trigger fired, indicating that a user required a message, they were sent a message randomly selected from the 4 message types (following a uniform distribution).

In the top panel of Figure 2, we show the overall result of the simulation, in terms of the average number of interactions all users have with the system over time. There are three results shown here. The solid line shows the interactions performed by users receiving recommended messages from NudgeAlong, the dashed line shows the interactions performed by users receiving random messages (the control group), while the dotted line shows the results for a situation where no user received a message over the course of the simulation.

If we consider the recommended messages first, it can be seen in this plot that for the first 7 days the average number of interactions of all users is about 12. Following the messages that are sent to every user on the 7^{th} day, however, this rises dramatically to approximately 19 (as all users are spurred to act) before decreasing again as satiation takes hold. The system eventually settles down to a steady state that varies between about 16 and 17.5. If we compare this with the initial value of approximately 12, this is a noticeable increase in use of the system. It is also noticeable that there is an increasing trend with time in the average number of interactions. That is, as NudgeAlong is running, it is becoming more efficient at nudging users to a higher usage, as expected (as it is more consistently picking the correct message type for them).

If we compare these results to the control condition (dashed line), we can see some similarities and some important differences. For the control condition, we again see a large increase following the initial sending of messages to users, followed by a decrease. However, here the system settles down to a steady state at a significantly lower level than for the recommended message example, occupying a range of between about 14 and 16. Also no increasing trend over time is evident. We find that, between the 1^{st} November 2013 and the end of the simulation, the improvement in the average number of interactions as a result of using recommended messages as opposed to random messages is of the order of 10%.

If we look at the situation where no user receives a message (the dotted line), we can see that this benchmark level remains fixed at ∼12 for the duration of the simulation, as expected.

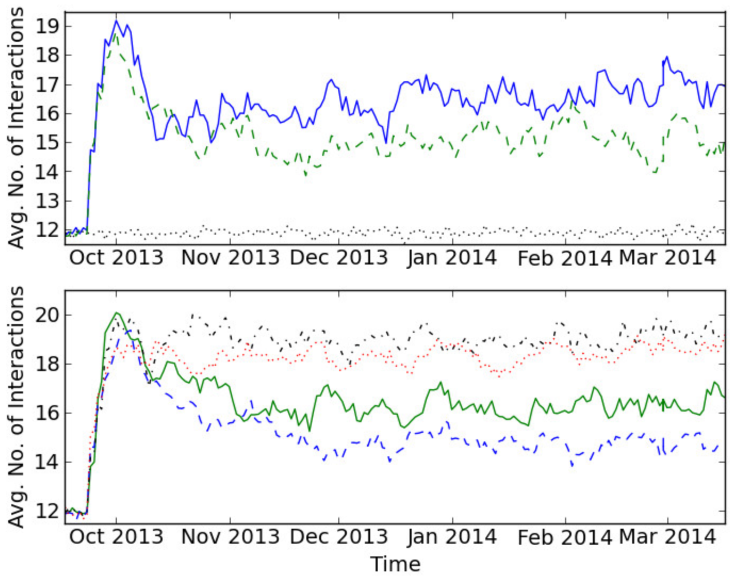

Fig. 2. (upper panel) Variation of the average number of interactions users have with the simulated home heating system. Results from recommended messages are shown as the solid line; results from random messages are shown as the dashed line; results from the situation where no user received a message are shown as the dotted line. (lower panel) The same results for a situation where only users with a satiation probability of greater than or equal to 50% experience satiation (see text for details).

As mentioned Section 3.1, we examined two satiation scenarios in the simulation. In the previous example, we examined satiation scenario 1. For comparison, we will now examine scenario 2, i.e., a situation where only some percentage of users, those that equal or exceed a probability threshold, experience satiation within the 7–14 day range. To achieve this, each user is randomly assigned a probability (from a uniform distribution) of between 1 and 100%. This modification of the simulation is set up so that only those users who equal or exceed a probability of 50% will experience satiation. To ensure a situation where users with low initial probabilities also eventually experience satiation, the probability of satiation is increased by a value of 0.5 each day for all users with an initial satiation probability <50% (once satiation occurs, the 50% threshold passed, these probabilities are reset to their default values). This seems realistic as, as more and more time goes by, it is probable that users will forget about the recommended message they received some time (up to 14 weeks) previously and revert back to a pre-existing behaviour. We show the results of this modification in Figure 2 (lower panel). From this figure, the solid line shows the results for recommended messages from NudgeAlong, while the dashed line shows the results for random messages (the control). Again, we find that, between the 1^{st} November 2013 and the end of the simulation, the improvement in the average number of interactions, as a result of using recommended messages as opposed to random messages, is of the order of 10%.

Alternatively, if we look at a situation where strictly *only* those users who equal or exceed a satiation probability of 50% experience satiation, we get the results shown in Figure 2 (lower panel) by the dot-dash and dotted lines for recommended and random messages, respectively. Because approximately half of users do not now ever experience satiation the average number of interactions has increased. As all message types have a non-zero probability of changing user behaviour the case base can initially, following the initial "seeding", contain a majority of non-optimum message types for each segment. As users who do not experience satiation will not receive another message after they change behaviour (they will consistently behave thereafter), they will not contribute further to the development of the case base, and non-optimum messages can, as a result, remain a majority in the case base for each segment. The result of this is that message types recommended for a user may not be the optimum message type for any given segment. This potentially means that the difference in the level of interactions between using random messages and recommended messages is reduced, i.e., recommended messages may be no different or even worse than random messages in producing a positive behaviour change (a higher level of interactions). This is, in fact, what we observe, with the improvement between the results from the recommended and random messages being reduced to $\sim 3\%$ in this scenario. This result is perhaps not too surprising as, if users do not interact with the system, it cannot "learn" (expand the case base) and so improve its recommendations to users (select the optimum message type).

In Figure 3, we show the continuous variation over time of message types allocated to users in each of the 5 user segments. We note that the results in this figure are from our original set-up, scenario 1, i.e., a situation where *all* users are assumed to experience satiation after a period of between 7–14 days. As an example, we will concentrate on the first segment at the top of this figure, which shows the results for the Casual Programmers (CP) segment. In the first 5 weeks of the simulation, it can be seen that the system is preferentially allocating the Text1 message type to users and not the Email1 message type, which according to Table 1 has the highest probability of success. After 5 weeks, however, the system determines which message type is most appropriate for each user, i.e., Email1, and from then on this is the message type that is preferentially allocated to each user in this segment.

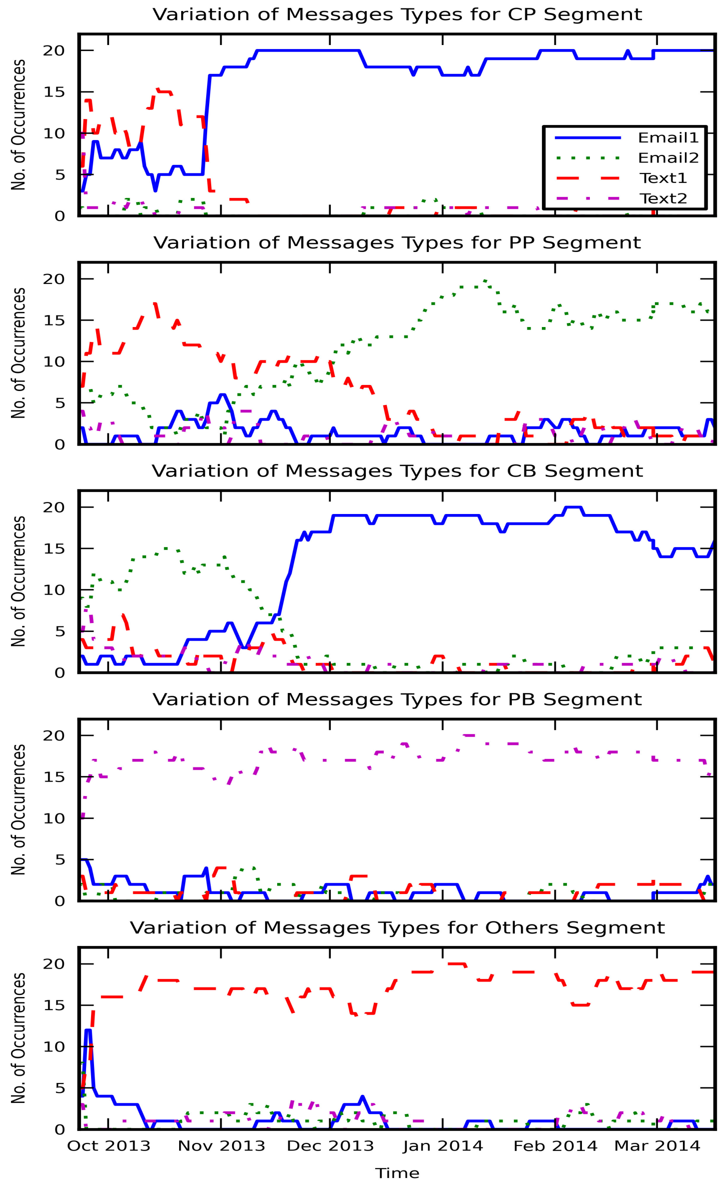

Fig. 3. Variation of the message types allocated for the different segments as a function of time. The different segments are, from top to bottom, Casual Programmer (CP), Power Programmer (PP), Casual Booster (CB), Power Booster (PB) and Others.

The relative success of the Text1 message type at the beginning of the simulation can be explained. At the beginning, it can be seen that 6 Text1 message types (out of a possible 20) are randomly allocated to users, compared to 3 Email1 message types. After waiting the specified number of days, the system checks the results and finds that 2 out of the 6 Text1 message types are successful in causing an increase in the use of interactions in our simulated home heating systems, while only 1 out of the 3 Email1 message types is successful. In comparison, all 10 Text2 message types allocated at the beginning are found to fail. The case base, as a result, begins with a majority of cases corresponding to the Text1 message type (2 Text1 cases compared to 1 Email1 case) and this is consequently the message type that is initially preferentially allocated to users of this segment when their behaviour triggers a message to be sent. Over time, and by the 5th week, the number of failures from the Text1 message type has reached a point where the number of Email1 cases in the case base surpasses it. At this point, due to the higher probability of success of the Email1 message type, it soon dominates and, indeed, soon, at the end of April and the start of May, accounts for all possible message types allocated.

Similar results are found for the Power Programmer (PP) and Casual Booster (CB) segments. In the case of the Power Booster (PB) and Others segments, the "correct" message type is chosen essentially from the start of the simulation. By the start of May it can be seen that all segments are being allocated message types consistent with the "preferred" message type for their segment, i.e., the message types with the highest probabilities of success from Table 1. For example, Text2 for the PB segment, Email1 for the CB segment, etc. This remains the case for the remainder of the simulation, up to August. While users in the different segments preferentially and more frequently receive the particular message type that has the highest probability of success, they do not exclusively receive only those message types. As mentioned in Sect 2, a test is performed to see if a particular message type has recently failed to convince a user to change their behaviour. The test we use is that if a particular message type fails three times, out of the last five messages sent, then a new random message type is sent instead. As no message type has a 100% probability of success, users in the different segments, therefore, also receive message types other than the preferred one as the simulation progresses. This accounts for the small number of alternative message types, e.g. for the Power Booster segment these are Email1, Email2, Text1, present at all times up to the end of the simulation in August. It is clear, however, that the system is correctly finding the "correct" message types for each segment, as it was designed to do.

3.3 Real Deployment

Following the positive results of this simulation of the NudgeAlong system, we carried out an initial trial deployment of the system using actual users from Climote. The trial, using 9 users of the system, was carried out between the 15^{th} February and the 28^{th} March 2014. Messages to users were sent out using SMS, with 4 different message types (motivational, gamification (comparing user's usage to encourage competition), financial and environmental). With the proviso that this was a very small trial, and was not carried out as a blind experiment, it was found that within the first four weeks (a period within which messages were being sent out at least every 4 days) total system usage increased from on average approximately 7 daily interactions to approximately 17 interactions. While this was a very small and limited trial, carried out primarily to test the system "in the field", the initial results are promising and further trials are planned.

4 Related Work

The NudgeAlong system is at heart a recommender system. Recommender systems have been used successfully for many years in e-commerce applications to encourage users to buy more - more books, movies, trips, etc. Content-based recommenders [4] use the similarity between the items purchased or to be purchased whereas collaborative filtering recommenders [5] make recommendations based on similarities across users' opinions and preferences. Hybrid systems adopt elements of both approaches. Trust-based recommendations [6] use social network structures of their users to enhance and improve the quality of the recommendations made.

The NudgeAlong system differs from the more traditional types of recommenders as it uses user interaction data to drive the recommendations. The more typical use of user interactions in recommender systems has been in deriving ratings from implicit feedback for collaborative filtering systems, where the user preferences are modelled based on user interactions [7, 8]. However there has been some work where user interactions are used directly to derive recommendations.

Trace-based reasoning [9] uses user traces, a sequence of events that occur as part of a user activity, as knowledge sources. Trace-based recommendation based on a trace-based reasoning approach has been used for contextual recommendation [10] in video recommendation [11] and in the acceleration of annotation and minimisation of user error in the user annotation process of digitised cultural heritage documents [12].

Esslimani et al. [13] proposed a variation to the standard collaborative filtering approach to recommendation which used usage traces, in this case users' interactions on a website including the pages they accessed and the times they were accessed. Their Behaviour Network based Collaborative Filtering approach to recommendation used these usage traces to determine behavioural similarities between users in order to recommend resources and other webpages on the site.

Related work by Liao et al. [14, 15] predicts users' preferences by exploring their usage traces of apps on mobile devices. The usage trace in this work records the number of uses of a particular app.

The aim of the NudgeAlong system is to provide recommendations which will encourage change in user behaviour. There is little evidence of other recommender systems which attempt to do this but work by Nepal et al. [16] also focuses on recommendations to change user behaviour. They built a social recommender for an online community to deliver government services to citizens. Certain individuals in this community were moving from one payment to another which would potentially result in receiving less money and required them to go back into the workplace. The objective of the recommender system was to recommend people and activities to encourage behaviour change, i.e., to help those in the midst of such transitions. The recommender system captured the social behaviour of the individuals through their interactions with the online community, including viewing and commenting on forum posts and generating, accepting and declining friend invitations.

5 Conclusions

In this paper we presented the NudgeAlong system for encouraging behaviour change in users of digital consumer applications. The heart of this system is a case-based recommendation engine which uses similarities between user interaction behaviour to identify the appropriate types of messages to be sent to users to encourage behaviour change. The system incorporates a simulation framework to allow evaluation and we have presented a simulation evaluation of this system which is modelled closely on real user data from the Climote remote heating control system. Running a simulation, based on 100 users, and using four different message types (Email1, Email2, Text1, Text2), we show that the system can successfully find the most appropriate message type for each segment after a period of 1-2 months. We note that this system has also undergone preliminary trials using real Climote users and the initial results of this field testing have been found to be promising.

In the future we plan to perform significant evaluation deployments of the system with Climote and with other partner companies in different industries. There are significant improvements that could be made to the system. Firstly, the current case structure is relatively limited and we will work on integrating more extensive user profiles. This relates to an overall question regarding NudgeAlong: is there sufficient information in the user profiles to actually determine the likely success or failure of different message types? Another option would be to expand the case representation to include things like the desired behaviour change required for each user. The concept of a single message type being most successful for a particular user may be overly simplistic. Future work would involve an investigation into whether a mix of message types would be more successful. We also plan to modify the reuse aspect of our system to further take into account message failures, to build on the relatively simple test mentioned in Section 3, and fully exploit information regarding messages that did not result in behaviour change. Finally, we plan to extend the use of case-based approaches to the triggers as well as the selection of the message type.

References

1. Fogg, B.: A behavior model for persuasive design. In: Proceedings of the 4th international conference on persuasive technology, ACM (2009) 40

2. Fogg, B.: Captology: The study of computers as persuasive technologies. In: CHI '97 Extended Abstracts on Human Factors in Computing Systems. CHI EA '97, New York, NY, USA, ACM (1997) 129–129
3. Jonsson, P., Wohlin, C.: An evaluation of k-nearest neighbour imputation using likert data. In: Proceedings of the Software Metrics, 10th International Symposium. METRICS '04, Washington, DC, USA, IEEE Computer Society (2004) 108–118
4. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web. Volume 4321 of Lecture Notes in Computer Science., Springer (2007) 325–341
5. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artificial Intellegence **2009** (2009)
6. Victor, P., Cornelis, C., Cock, M.D., Teredesai, A.: Trust- and distrust-based recommendations for controversial reviews. IEEE Intelligent Systems **26**(1) (2011) 48–55
7. Jawaheer, G., Szomszor, M., Kostkova, P.: Characterisation of explicit feedback in an online music recommendation service. In Amatriain, X., Torrens, M., Resnick, P., Zanker, M., eds.: RecSys, ACM (2010) 317–320
8. Palanivel, K., Sivakumar, R.: A study on collaborative recommender system using fuzzy-multicriteria approaches. IJBIS **7**(4) (2011) 419–439
9. Cordier, A., Lefevre, M., Champin, P.A., Georgeon, O.L., Mille, A.: Trace-based reasoning - modeling interaction traces for reasoning on experiences. In Boonthum-Denecke, C., Youngblood, G.M., eds.: FLAIRS Conference, AAAI Press (2013)
10. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. AI Magazine **32**(3) (2011) 67–80
11. Zarka, R., Cordier, A., Egyed-Zsigmond, E., Mille, A.: Contextual trace-based video recommendations. In Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S., eds.: WWW (Companion Volume), ACM (2012) 751–754
12. Doumat, R., Egyed-Zsigmond, E., Pinon, J.M.: User trace-based recommendation system for a digital archive. In Bichindaritz, I., Montani, S., eds.: ICCBR. Volume 6176 of Lecture Notes in Computer Science., Springer (2010) 360–374
13. Esslimani, I., Brun, A., Boyer, A.: From social networks to behavioral networks in recommender systems. In Memon, N., Alhajj, R., eds.: ASONAM, IEEE Computer Society (2009) 143–148
14. Liao, Z.X., Pan, Y.C., Peng, W.C., Lei, P.R.: On mining mobile apps usage behavior for predicting apps usage in smartphones. In He, Q., Iyengar, A., Nejdl, W., Pei, J., Rastogi, R., eds.: CIKM, ACM (2013) 609–618
15. Liao, Z.X., Peng, W.C., Yu, P.S.: Mining usage traces of mobile apps for dynamic preference prediction. In Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G., eds.: PAKDD (1). Volume 7818 of Lecture Notes in Computer Science., Springer (2013) 339–353
16. Nepal, S., Paris, C., Bista, S.K.: Srec: a social behaviour based recommender for online communities. In Herder, E., Yacef, K., Chen, L., Weibelzahl, S., eds.: UMAP Workshops. Volume 872 of CEUR Workshop Proceedings., CEUR-WS.org (2012)