Predicting Feedback Compliance in a Teletreatment Application

Harm op den Akker
Roessingh Research and Development
Enschede, the Netherlands
Email: h.opdenakker@rrd.nl

Val Jones
Telemedicine Group, University of Twente
Enschede, the Netherlands
Email: v.m.jones@ewi.utwente.nl

Hermie Hermens
Roessingh Research and Development
Telemedicine Group, University of Twente
Enschede, the Netherlands
Email: h.hermens@rrd.nl

***Abstract*—Health care provision is facing resourcing challenges which will further increase in the 21^{st} century. Health care mediated by technology is widely seen as one important element in the struggle to maintain existing standards of care. Personal health monitoring and treatment systems with a high degree of autonomic operation will be required to support self-care. Such systems must provide many services and in most cases must incorporate feedback to patients to advise them how to manage the daily details of their treatment and lifestyle changes. As in many other areas of healthcare, patient compliance is however an issue. In this experiment we apply machine learning techniques to three corpora containing data from trials of body worn systems for activity monitoring and feedback. The overall objective is to investigate how to improve feedback compliance in patients using personal monitoring and treatment systems, by taking into account various contextual features associated with the feedback instances. In this article we describe our first machine learning experiments. The goal of the experiments is twofold: to determine a suitable classification algorithm and to find an optimal set of contextual features to improve the performance of the classifier. The optimal feature set was constructed using genetic algorithms. We report initial results which demonstrate the viability of this approach.**

***Index Terms*—Mobile healthcare, activity monitoring, feedback compliance, machine learning, genetic algorithms.**

I. Introduction

An ambulant system has been developed designed to guide the patient to reach a healthy distribution of activity over the day. The system consists of a 3D-accelerometer to assess the patient's daily activity pattern in counts per minute, combined with a PDA for providing feedback. By comparing his activity to some predetermined reference activity pattern the patient is provided with feedback messages at regular intervals advising them to be more or less active or that they are performing well. This system was used in three different patient groups: chronic low back pain (CLBP) patients [1], chronic fatigue syndrome (CFS) patients [2], [3] and people suffering from obesity ($BMI > 30$). In the case of CLBP and CFS patient populations, the goal of the feedback is to spread activity over the day, while for obesity patients, the goal is to encourage them to increase activity over all.

In this research we are looking at the responses to the individual feedback message with a view to developing a method of generating messages in a smarter, more efficient and personalized manner. Related work in the field is reported in [4], [5] where the aim is to cluster diabetic patients based on the type of messages that seem to provoke a positive response in the patients. They report preliminary, but promising results in a dynamic clustering system that learns the preferences of users over time. Our overall objective is to investigate how to improve feedback compliance in patients by taking into account various contextual features associated with the feedback instances.

II. Datasets

For this research we used retrospective data consisting of three datasets (or corpora): the CLBP corpus, the CFS corpus and the Obesity corpus. The patients from our three populations had carried the monitoring system on average 13.6 days ($\sigma = 11.7$). In all three studies, the protocol included an experimental group who received feedback on some days, and on other (control) days did not; as well as a control group who never received feedback. A total of 45 patients received feedback from the system. For these patients, feedback was given on average on 13 days ($\sigma = 8.8$). Patients were asked to wear the system during waking hours (approximately from 8 am, until 10 pm). In all three studies the measurement system consisted of a PDA connected wirelessly to a 3D-accelerometer measuring the patient's physical activity levels throughout the day. Based on measured values, a variety of feedback messages are given to the patient via the PDA screen. The system logs acceleration as an integrated value, summed up over the three axis of movement per 60 second interval [6] as well as the timing and content of the given feedback messages.

Table I gives an overview of some relevant statistics on these three corpora. Due to gaps in the sensor data, or erroneous timing of the feedback messages (sometimes two messages would be generated too close to each other, rendering one of them useless) we could not use all the data for our data analysis. The last row in Table I shows the total number of data instances that were used for our machine learning experiments following exclusion of the problem data.

III. Method

To see how people respond to the feedback messages we use a *compliance measure*. In this article we report on a compliance measure which compares the amount of activity performed in the 30 minute interval before the feedback

	CLBP	CFS	Obesity	Total
Subjects	17	38	40	95
Measured days	322	675	308	1305
Feedback subjects	17	11	17	45
Feedback days	210	269	109	588
Feedback given	1772	1300	1006	4078
Usable feedback	621	455	536	1612

TABLE I
CORPUS STATISTICS.

event (Δ_1), with the amount of activity performed in the 30 minute interval after the feedback event (Δ_2). By comparing these values we can see if a subject was more active after an encouraging feedback message (F_{enc}) or less after a discouraging message (F_{dis}). We refer to messages that suggest increasing activity as *encouraging*, and those that suggest decreasing activity as *discouraging*. An example of an encouraging messages is "you should go for a walk" and an example of a discouraging message is "read the newspaper". In addition, *neutral* messages are generated to reinforce when the patient is doing the right thing. If no more than three data points (3 minutes of measurement) are missing in Δ_1 and Δ_2, we can reliably determine the compliance and differentiate between the following cases:

1) Message Type 'F_{enc}' and $\Delta_1 < \Delta_2$.
2) Message Type 'F_{enc}' and $\Delta_1 >= \Delta_2$.
3) Message Type 'F_{dis}' and $\Delta_1 <= \Delta_2$.
4) Message Type 'F_{dis}' and $\Delta_1 > \Delta_2$.

In cases (1) and (4), we determine that the subject *complied with* the message, while in cases (2) and (3) we say that the subject *did not comply with* the message. The reason that we choose to make a hard split between *compliant* and *non-compliant* instead of using a numerical *compliance value* (the numerical compliance would be 0.0 in cases (2) and (3), in case (1) it would be $\frac{\Delta_2}{\Delta_1}$ and in case (4) $\frac{\Delta_1}{\Delta_2}$) is that in an application that uses our compliance prediction method, the final decision will be to either generate a feedback message or not. Instead of using a compliance value now, and applying thresholding later, it is sufficient for the purpose of this experiment to apply the thresholding immediately.

We want to find out why patients sometimes comply with feedback and sometimes not. To do this we try to predict the compliance of a feedback messages by looking at its context. After calculating the compliance for every feedback message in our datasets, the next step is to define this context in terms of features (see Section IV). After enriching our datasets with features, we use a statistical machine learning approach to find the relationships between context features and the compliance to the feedback message in Section V.

IV. CONTEXTUAL FEATURES

The context of each feedback message instance is captured in a set of features related to that specific feedback message instance. These features primarily should contain information that might be relevant for the patient's (unconscious) decision to either ignore or follow the given advice. Also, these features should be available to the system quickly and automatically. There are many conceivable reasons for a patient not to follow the advice that was given. For example "*I don't feel like walking now*", "*I am tired*" or "*I am in too much pain right now*" are all perfectly valid reasons, but they are difficult (if not impossible) to measure automatically. Not only would these factors be difficult to measure automatically, in this case they were not recorded in the corpora used here. That limits us to the data that was gathered during the feedback experiments and data that can be added retrospectively.

The features that we defined fall roughly into five categories: **(A)** time related, **(B)** message related, **(C)** weather related, **(D)** history related and **(E)** activity related. We shall discuss these in detail now.

A. Time related features

The time related features that are calculated for each feedback message instance concern the time at which the message was generated.

- **[dayOfWeek]** which day of the week it is.
- **[weekDay]** whether this is a weekday or not.
- **[dayPart]** whether the message was given in the morning (<12:00), afternoon (12:00-18:00) or evening (>18:00).
- **[hourOfDay]** the hour of the day on which the message was given (rounded off).

The rationale behind these features is that people have both a weekly and a daily rhythm. This means that on some days (e.g. sundays) people might want to relax and hence may be less motivated to be active. In case of daily rhythm, people might be bound to a sedentary job during certain fixed hours of the day. Adding these features can, given sufficient data, help to detect individual's patterns and enable adaptive behaviour of the system concerting timing of feedback.

B. Message related features

The following message related features contain information about the message itself.

- **[feedbackType]** whether this is an *encouraging* or *discouraging* message.
- **[feedbackMessage]** the exact textual content of the feedback message.
- **[messageGoOutside]** whether the feedback message advises the patient to go outside.
- **[messageIsAQuestion]** whether the feedback message was phrased as a question.
- **[messageSuggestIdle]** whether the feedback message suggests that the patient sits idle for a while.

The first two of these message related features are self-explanatory, but the last three might require some background. The reason for including the **[messageGoOutside]** feature is that if the system advises a patient to go out for a walk when the weather is bad, the patient might be inclined to ignore the message. This feature is thus related to the weather

features (Section IV-C). Whether or not a message is phrased as a question (**[messageIsAQuestion]**) might influence the patient's willingness to react. Some people might prefer a system that is more strict and issues its feedback as "commands", while others might be more stubborn and dislike a commanding tone, thus preferring a more suggestive style of message. In the case of the **[messageSuggestIdle]** feature, some patients might prefer a more detailed suggestion (e.g. read the newspaper), while others might react more favorably to a message such as "take it easy".

C. Weather related features

The following features contain information about the weather on the day the feedback message was generated. They are taken from the Royal Netherlands Meteorological Institute[1] and were added to the corpora retrospectively.

- **[meanTemperature]**
- **[minimumTemperature]**
- **[maximumTemperature]**
- **[cloudScale]**
- **[precipitationSum]**
- **[precipitationDuration]**

The reason for using weather data as features is that the weather conditions might influence the patient's willingness to respond, especially when the message tells them to go outside.

D. History related features

The following features are related to the history of usage of the feedback system.

- **[dayOfUsage]** a count of how many days the subject has been receiving feedback from the system.
- **[totalMSGSToday]** the total number of messages received so far this day.
- **[encouragingMSGSToday]** the total number of encouraging messages received so far today.
- **[discouragingMSGSToday]** the total number of discouraging messages received so far today.
- **[neutralMSGSToday]** the total number of neutral messages received so far today.
- **[averageCompliance]** the average numerical compliance of all previous feedback messages this day.
- **[sameMessageTodayCount]** the number of times the exact same message (as this message) was received today.
- **[sameMessageOverallCount]** the number of times the exact same message was received in the total history.
- **[sameTypeMessageTodayCount]** the number of times the same type of message (F_{enc} or F_{dis}) was received today.
- **[sameTypeMessageOverallCount]** the number of times the same type of message was received in the total history of usage.

These 10 features are designed to capture the state of previous interactions with the system. For example, receiving the same message several times on the same day might cause a habituation effect or even irritation leading to non-compliance. The same kind of reasoning lies behind the rest of the history-related features.

E. Activity related features

The last two features are calculated from the measured activity level of the subject.

- **[distanceFromReference]** the distance from the reference line at the time of the feedback message instance.
- **[approachingReference]** whether or not the patient is approaching the reference line, calculated as the difference between the time of the feedback message instance T and T - 30 minutes.

A patient whose current activity level is much lower than his reference line (**[distanceFromReference]**) is possibly harder to motivate than someone who is closer to his optimal activity level.

V. Experiments

The goal of the machine learning experiments is twofold: to determine a suitable classification algorithm and to find the set of features that result in the best performance. Performance in this case is measured by accuracy of the classifier, defined as the number of correctly classified instances divided by the total number of instances in the dataset. Because the goal is to have a patient specific classification method, we choose to perform the experiments on the datasets of individual patients. Some patients in our datasets were given so few feedback messages that there was not enough data to perform the machine learning experiments on them, so they were excluded, leaving a total of 38 patients: 12 CLBP, 11 CFS and 15 Obese patients.

A. Baseline

In order to judge the accuracy of a certain machine learner outcome, a baseline is required. This was calculated for every patient by using a ZeroR, or most occurring class, classifier. The ZeroR classification scheme calculates the relative occurrence of classes in the training set, then, in the test phase it assigns to each unseen instance the most occurring class. In our two-class classification problem, if e.g. 60% of the instances are in the 'yes' class and 40% in the 'no' class, the ZeroR classifier will assign to each instance the 'yes' class and will achieve 60% accuracy. The average baseline performance over the 38 patients was 60.74% ($\sigma = 7.01$).

B. Genetic Algorithm

For all further experiments we use a genetic algorithm (GA) to search for good combinations of features.

"*Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm [...]. In every generation, a new set of artificial creatures (strings) is generated using bits and pieces of the fittest of the old [...]*" [7].

[1] http://www.knmi.nl/klimatologie/daggegevens/index.cgi

In our case, we use chromosomes (or binary strings) that represent subsets of the complete set of features: each position in the string maps to a specific feature. If the string contains a '1' at a certain posititon, the feature to which that position is mapped is selected for the feature set, otherwise it is not. The fitness of chromosomes is calculated by filtering out all the features that are mapped to a '0' in the bitstring, and performing Leave-One-Out (LOO) classification with the remaining set using a certain machine learning scheme. The population size (numbers of chromosomes in each generation) is set to 100. In the selection step we use tournament selection (see e.g. [8]) with a tournament size of 2. In the case that the two chromosomes selected in the tournament have the exact same fitness, the chromosomes with the lowest number of 1's is selected for reproduction. This favouring of smaller feature sets has been shown not to negatively influence the results of the search procedure, and it makes practical sense to do so since it results in classifiers that are faster to train and faster to test. The crossover- and mutation rates were fixed to 0.8 and 0.001 respectively. These values are close to the ones often cited in literature and have been experimentally determined to provide decent convergence rates. To reduce the probability of finding local optima from a GA run, all experiments were repeated 200 times, storing the global optimal results along the way. With these settings, in the feature selection experiments (see Section V-D) on average 5036 feature sets were tested ($\sigma = 270$) per patient, which is 0.0038% of the total search space in an average time of 85 minutes ($\sigma = 100$) per patient.

C. Classifier Selection

The first part of the experiments deal with the selection of a suitable classification method. We applied a set of 35 different machine learning schemes from the WEKA Machine Learning toolkit [9]. In the first step we selected the patient with the highest instances count and ran the genetic machine learner for all classifiers. This resulted in an initial selection of the 10 best scoring classifiers: Ridor, part, ADTree, JRip, J48graft, J48, REPTree, NBTree, RandomForest and BFTree. In the next step we repeated the experiments with a larger number of patients. Because of the time constraint associated with running the genetic machine learning algorithm for all patients, we chose a set of 12 patients: 6 with relatively high instance counts ($\mu = 74$, $\sigma = 19$) and 6 with relatively low instance counts ($\mu = 26$, $\sigma = 5$). To determine the overall best performing classifier from the set of 10, we chose to implement a voting system, whereby for each run, the best performing classifier receives 3 points, the second-best receives 2 points and the third-best 1 point. Overall the Ridor classifier received the highest number of points: 27 out of a possible 36 (runner-ups had 22 points or less) and thus was selected as the most suitable classifier. The Ridor, or RIpple-DOwn Rule learner, is a simple rule learning scheme whereby first a most general rule is derived from the data and subsequently exception rules are generated in a cascading manner [10].

D. Feature Selection

For the feature selection experiments we used the Ridor classification scheme with the aim of determining for all patients how the different features influence the classification of compliance performance. The genetic algorithm settings that were used are described above in Section V-B and will not be repeated here. Table II shows the average baseline, performance and relative improvement over the baseline per corpus as well as the overall average values. Numbers in brackets indicate the standard deviations. These results represent the best recorded scores during the genetic search over the feature space.

Corpus	Baseline	Score	Improvement
CLBP	61.94 (5.91)	86.16 (3.49)	+63.64%
CFS	63.70 (9.05)	87.24 (3.55)	+64.85%
Obesity	57.59 (5.00)	85.05 (4.83)	+64.75%
Total	60.74 (7.01)	86.03 (4.09)	+64.42%

TABLE II
CLASSIFICATION RESULTS PER CORPUS. THE IMPROVEMENT IS CALCULATED BY PLACING THE SCORE ON A SCALE FROM BASELINE TO 100; THE THEORETICAL PERFORMANCE LIMIT.

All scores are significant improvements over the baseline with p-values less than 0.0001, computed using a paired t-test over individual patient results. For each patient, we look at the feature sets that achieved the highest score using the least number of features. Figure 1 shows the list of features ranked by number of occurrences in the top-scoring, minimal feature set solutions. If e.g. for one patient there were 3 unique solutions and a feature was selected in 2 out of the 3 solutions, that feature's weight is increased by $\frac{2}{3}$.

On average each feature was selected 6.28 times ($\sigma = 2.73$). This is observable from Figure 1 where only the feature '**approachingReference**' really stands out. This means that for each patient, the Ridor classifiers that were trained and showed to have the highest performance each rely on very variable feature sets. This diversity in feature selection among the different subjects is most likely caused by the small datasets used for training the classifiers. Figure 2 shows the average improvement over baseline over all subjects plotted against the number of instances used for training. When ignoring the datapoint for 10 instances ($\mu = 32.5$, $\sigma = 37.1$), a trend of rising improvement over baseline can not be seen. Usually when plotting instances versus performance a rise in performance is expected when using more training data, which is not the case here. This could mean that either all results are random (which they have proven not to be) or overall performance will only start increasing with much more data (more likely).

VI. DISCUSSION

The work presented in this paper demonstrates the possibility of predicting a user's reaction to a feedback message based on a simple set of contextual features. This findings contribute to the development of a *decision component* that

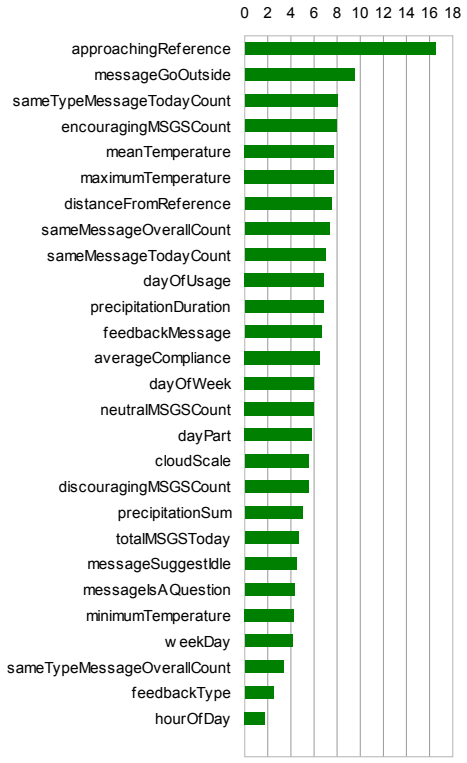

Fig. 1. Features weighed by their occurrence in the best-scoring, minimal feature sets.

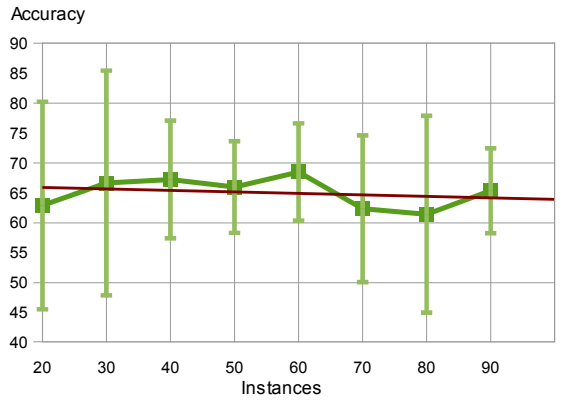

Fig. 2. Average improvement over baseline for all subjects when using increasing numbers of instances for training. In red is the linear trend line: $-0.25 * x + 66.08$.

will be integrated with our current activity feedback software which runs on a PDA. The decision component will optimise the selection of timing and content of feedback messages to the patient. Instead of generating feedback messages at fixed time intervals, we can regularly poll the decision component after supplying it the contextual situation as a feature vector, and let it return a decision (yes or no) about whether or not it is a good time to give feedback. As the message texts are present in the feature vector, these can be varied and presented to the decision component to find out if one message text is more effective than an alternative text conveying the same intention. If the decision is made to send a particular feedback message to the patient, the compliance to that message is automatically calculated from the activity data after a certain time interval (currently 30 minutes). This information can then be used to re-train the decision component with the added data in order to improve prediction performance over time.

It should be noted that the results reported in this paper of 64.42% improvement over the baseline are based on historical data. This means that the sort of contextual data that could be used was limited to that which was recorded at the time plus any that could be added retrospectively. In future research on activity feedback we have the possibility of recording much more context data in order to create a richer dataset. In order to be able to show the learning capabilities of the system, more longitudinal measurements of patients are also preferred.

Although it has been shown that it is theoretically possible to predict compliance to feedback messages, the question remains how such a real-time implemented system will behave and how its behaviour will be perceived by its users. The next step of implementing and testing with real users is therefore crucial and will be conducted within the European AAL-funded IS-Active project, which aims to improve the physical condition of COPD patients.

ACKNOWLEDGEMENTS

Part of the research was funded by the European AAL-funded IS-Active project. The authors thank M. van Weering and R. Evering for being able to use their data.

REFERENCES

[1] M. van Weering, M. M. R. Vollenbroek-Hutten, T. Tönis, and H. J. Hermens, "Daily physical activities in chronic lower back pain patients assessed with accelerometry," *European Journal of Pain*, vol. 13, no. 6, pp. 649–654, August 2008.

[2] M. van Weering, M. M. R. Vollenbroek-Hutten, E. M. Kotte, and H. J. Hermens, "Daily physical activities of patients with chronic pain or fatigue versus asymptomatic controls : a systematic review," *Clinical Rehabilitation*, vol. 21, no. 11, pp. 1007–1023, November 2007.

[3] R. Evering, M. van Weering, K. Groothuis-Oudshoorn, and M. Vollenbroek-Hutten, "Daily physical activity of patients with the chronic fatigue syndrome: A systematic review," *Clinical Rehabilitation (Accepted)*.

[4] F. Cortellese, M. Nalin, A. Morandi, A. Sanna, and F. Grasso, "Static and dynamic population clustering in personality diagnosis for personalized ehealth services," in *Proceedings of the 3rd International Workshop on Personalisation for e-Health, AIME2009*, July 2009, pp. 1–9.

[5] ——, "Personality diagnosis for personalized ehealth services," in *Proceedings of the Second International ICST Conference on Electronic Healthcare for the 21st century*, September 2009, pp. 173–180.

[6] C. Bouten, W. V. van de Venne, K. Westerterp, M. Verduin, and J. Janssen, "Daily physical activity assessment: comparison between movement registration and doubly labeled water," *Journal of Applied Physiology*, vol. 81, no. 2, pp. 1019–1026, 1996.

[7] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[8] B. L. Miller and D. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[9] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, 1st ed. Morgan Kaufmann, October 1999.

[10] B. R. Gaines and P. Compton, "Induction of ripple-down rules applied to modeling large databases," *Journal of Intelligent Information Systems*, vol. 5, no. 3, pp. 211–228, 1995.