Adaptive Advice in Automobile Climate Control Systems*

Ariel Rosenfeld
Department of Computer
Science, Bar-Ilan University
Ramat Gan, Israel
rosenfa5@cs.biu.ac.il

Amos Azaria
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, USA
amos.azaria@gmail.com

Sarit Kraus
Department of Computer
Science, Bar-Ilan University
Ramat Gan, Israel
sarit@cs.biu.ac.il

Claudia V. Goldman
General Motors Advanced
Technical Center
Herzliya, Israel
claudia.goldman@gm.com

Omer Tsimhoni
General Motors Advanced
Technical Center
Herzliya, Israel
omer.tsimhoni@gm.com

ABSTRACT

Reducing an automobile's energy consumption will lower its dependency on fossil fuel and extend the travel range of electric vehicles. Automobile Climate Control Systems (CCS) are known to be heavy energy consumers. To help reduce CCS energy consumption, this paper presents an adaptive automated agent, MDP Agent for Climate control Systems (MACS), which provides drivers with advice as to how to set their CCS. First, we present a model which has 78% accuracy in predicting drivers' reactions to different advice in different situations. Using the prediction model, we designed a Markov Decision Process which solution provided the advising policy for MACS. Through empirical evaluation using an electric car, with 83 human subjects, we show that MACS successfully reduced the energy consumption of the subjects by 33% compared to subjects who were not equipped with MACS. MACS also outperformed the state-of-the-art Social agent for Advice Provision (SAP).

Categories and Subject Descriptors

I.2.m [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Miscellaneous*

General Terms

Human Factors, Experimentation

Keywords

Energy Aware Systems; Advice Provision; Human-Agent Interaction

1. INTRODUCTION

According to Lee and Lovellette [18], by 2030 a rising middle class in China and India will cause a high demand for passenger cars which will result in approximately 1.1

---

*This work was supported in part by ERC grant #267523.

billion cars on the road worldwide, compared to 750 million in 2010. Such a fleet would probably not be able to rely on fossil fuels alone for long. The growing interest in electrical cars seems to offer a great solution to the escalating number of cars on the road [18]. Between 2012 and mid-2014, there was an increase of 170% in electrical cars worldwide [26, 21]. Yet, one of the most reported reasons for refraining from switching to electric cars is the relatively limited travel range it has (w.r.t petrol cars). Auto experts have reported that the car's climate control system (CCS) reduces the cars power efficiency by up to 10 percent [23, 12]. Thus, reducing energy consumption of the CCS is important in order to reduce the human ecological footprint and extend the travel distance of electric vehicles.

There is a broad consensus that modifying a driver's behavior may allow a reduction in vehicle fuel consumption [9]. Inspired by these results, in this paper, we propose an automated agent that advises a driver how to set the car's CCS, in a way that would reduce energy consumption while keeping the driver comfortable. The agent operates in a repeated interaction environment, in which it offers *adaptive* advice to the driver. While the agent is mainly concerned with the car's energy consumption, the driver is usually more interested in her own comfort level—an interest which changes from one driver to another. This partial conflict, which may occur, is a challenge in the advice provision process—the agent must consider the driver's comfort in order to provide reasonable advice. Furthermore, the agent should consider the trust the driver has in it and take into account the long term effect of each piece of advice.

We present a methodology which combines a *driver model*, a *CCS model* and an *environment model* in a Markov Decision Process (MDP) setting [27] in order to generate adaptive advice for the driver on how to set her CCS. In this work we mainly focus on the *driver model*, which considers the driver's preferences, the dynamically changing environment and the repeated interaction process between the driver and the agent. Modeling the driver in a personalized manner is very challenging. First, different drivers may have different preferences when it comes to CCS settings. Their preferences vary according to the environment changes and can even change during the ride itself. Second, the interaction between the driver and the agent also affects the driver's preferences and her future reactions to new advice suggested

by the agent.

In order to learn drivers' preferences and behavior while they interact with an agent in repeated interactions, we used 2 simple repeatedly interacting agents that were tested with 38 human subjects using a GM Chevrolet Volt electrical car. These agents use different advising procedures and examine different advice as well as different interaction sequences from which we can learn about the aforementioned factors. Generalizing findings from only 38 subjects is extremely difficult in repeated interaction settings, especially because we cannot simulate all of the different advice and interactions with our subjects.

Using machine learning techniques and careful selection of predicative features we were able to overcome the above challenges and construct a personalized driver model for predicting drivers' reactions to advice depending on its content, the context in which it was given and the previous agent-driver interactions. Predicting drivers' reactions necessitates the identification of predicative features which describe the driver's preferences, the environmental factors that influence her decision-making and a representation of the trust between the driver and the agent which affect her decision-making as well. We were able to identify a set of features with which we can provide a probability assessment for each specific driver of whether she will accept the given advice, based on the details of the advice, the current setting inside and outside the car and interactions before the advice was given. In a similar fashion, an environment model was constructed. A CCS model was obtained from previous work on a single interaction advice provision in CCS settings [5].

We used an MDP modeling, which utilizes the aforementioned models to account for the dynamic and changing environment in which the agent operates, and solved it to calculate an advising policy. The resulting policy, which is used by our proposed agent, MDP Agent for Climate control Systems — MACS, was tested and compared to the state-of-the-art Social agent for Advice Provision (SAP) modeling [4] and a non-advising agent (Silent). Our experiments, with 83 human drivers, show that subjects equipped with our agent significantly saved energy compared to both competing agents. Subjects equipped with MACS reduced their CCS energy consumption by 33% (on average) compared to subjects using a non-advising agent (Silent). These results were unexpected compared to the limited savings reported in [5] for a single interaction advising agent.

MACS can be deployed in petrol and electric cars alike and hold both long-term and short-term benefits for the driver and the environment.

Our proposed strategic advice provision approach can also be implemented in other human interacting systems where there is tension between the users' and system's goals.

2. RELATED WORK

Persuasive technologies have been used in different domains that target behaviors; from a healthy/safe lifestyle and the ecological behavior of an individual (see [13] for an excellent review) to changing a group's decision in voting [11]. These technologies use different techniques to persuade humans to change how they think or what they do [6] and differ in their theoretical background [20]. Among these persuasive technologies, one can find many agents for the improvement of energy efficiency. In many of these works, the agent is not required to provide *advice* as to how to achieve

the goal, but persuade the user into doing so by providing socio-feedback or eco-feedback [10, 8]. For example, in [7] a mobile application is presented that senses and reveals information about transportation behavior, in an attempt to persuade people to increase their use of green transportation. In [2] the authors investigate the design and evaluation of an intelligent agent that helps to persuade family members to conserve energy in their home. Other agents negotiate with humans in order to conserve energy, for example in the work-place [17, 1]. In the context of automobiles, some agents use drivers' observed behavior to automatically elicit the drivers' intention and goals [29, 25, 19, 16]. Yet, to the best of our knowledge, the only work that deals directly with advice provision concerning CCSs is [5].The authors propose CARE—a method to persuade a driver to reduce the energy consumption of the climate control system of her electric car. CARE is designed for a *single round interaction* with a driver, and is not *adaptive* to the driver's actions. That is, the driver receives advice which does not change throughout the ride. Furthermore, CARE requires the driver to report her "initial comfort level" (how comfortable the driver feels at the beginning of the ride) in order to personalize the advice such that different drivers receive different advice depending on their initial comfort level. Without obtaining this initial comfort level, CARE is *not personalized*, and all drivers would receive advice depending solely on the external and internal temperatures. In real world implementation we would like the agent to be able to personalize and adapt the advice to the drivers' actions and refrain from explicitly requesting data from the driver, as requesting such information may be annoying and even dangerously distracting [28]. In this work, the agent elicits the drivers' preferences implicitly from their observed actions and interactions with the system and does not require any additional information from the driver, prior to or during the ride.

Agents who consider their own utility while offering advice are common in the literature. While most of these works assume one-shot interactions, some do consider the long term effect of the advice in repeated interactions [24, 4]. The two state-of-the-art approaches for the latter used MDP modeling and maximized a weighted social utility function in a Social agent for Advice Provision model (SAP). The MDP and SAP models were compared in [3] in a *simulated* CCS environment, yet their results are inapplicable in our setting as the authors assumed explicitly given utility functions for both the agent and the user. In our work, we confront a much greater challenge: first, we assume no prior knowledge of the driver's preferences and utility. Second, the utility which the driver gains from the different climate control settings is not homogeneous among the drivers. Thus, an extensive context and preference-related analysis of observed actions was conducted in this work to provide a probabilistic estimation of the driver's reactions to possible advice. This estimation was in turn combined into an advice provision model in which the human driver could be persuaded to save the energy consumed by her electric car's climate control system.

3. THE CHEVROLET VOLT CLIMATE CONTROL SYSTEM

Our study is based on the GM Chevrolet Volt climate control system. In this system the drivers have control over
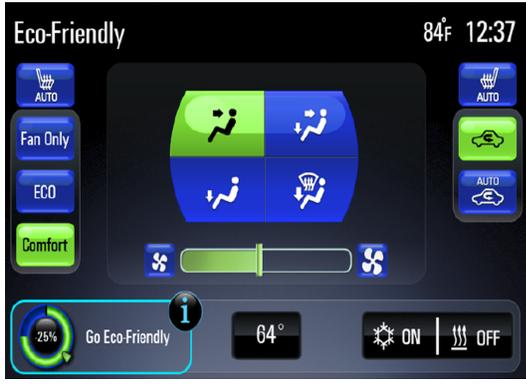
Figure 1: Main screen.

the setting S which is a tuple (T, F, D, M) consisting of the following variables:

- Temperature (T): this variable can receive values between 16 and 35, and is associated with a temperature in Celsius.

- Fan strength (F): this variable can receive values between 1 and 6 and is associated with the fan blower power.

- Air delivery (D): in the Volt climate control system, the air delivery may be set to face (panel), face and feet, only feet and windshield and feet. In our study we limited the air delivery to either the face (in which D is set to 0) or the face and feet (in which D is set to 1).

- Mode (M): This variable may either be set to 'eco' (when M is set to 0) or to 'comfort' (when M is set to 1).

The Volt climate control system also allows the following additional variables to be set, which we did not include in this study. These variables are heating of the driver and passenger seats, recirculation to either manual or automatic, and a 'fan only' mode which does not use the climate control's compressor. We set the air source to recirculation in all of the experiments. Figure 1 presents the climate control system panel.

Additional important parameters are E, which is the external temperature as displayed in the central stack, and I, which is the internal temperature as we measured with a manual thermometer located between the 2 front seats. We denote these two parameters together as *world state* $v \in V$, where V is the set of all possible world states.

Given a setting s, we use subscript s_T to refer to the temperature in that setting, s_F to refer to the fan strength, s_D for the air delivery and s_M for the mode of the setting.

3.1 Energy Consumption Model

We followed the model of the CCS energy consumption as described in [5]. This model was compared by the authors to others and yielded the greatest fit for the data they collected.

$$e(T, F, D, M, E, I) = (w_1 \cdot (-T) + w_2 \cdot F + w_3 \cdot D + w_4 \cdot E + w_5 \cdot I) \cdot (1 + w_6 \cdot M) \quad (1)$$

We use the same $w_1, w_2, ..., w_6$ calculated by the authors.

4. THE REPEATED ADVICE PROVISION PROBLEM

In our setting, there is a repeated interaction between a receiver (the driver) and a sender (the agent). The interaction is composed of *rounds* (intuitively, different rides in the car). In round t, after i interactions in that round, the sender observes the state of the world $v = (E, I) \in V$, the current climate control settings $s = (T, F, D, M) \in S$ and the history of interactions with the agent, $h^{t,i}$, and suggests that the receiver change its climate control setting to some $d \in S$. The agent is also allowed to "keep silent" and not offer any suggestion. After observing advice d from the sender (if given) the receiver chooses one of the following reactions $\{accept, reject\}$. Regardless of d and the timing, the receiver can choose any $s' \in S$ at her discretion.

The prediction model, which we will soon describe, is intended to assess $P(accept \mid d, t, v, s, h^{t,i})$, the probability that the receiver will choose to *accept* the sender's suggestion d at the time and context in which d was provided.

For a given world state v, settings s and history $h^{t,i-1}$, we define the sender's expected cost for advice d as follows:

$$EC_S^{t,i}(v, s, h^{t,i-1}, d) = \sum_{a \in \{accept, reject\}} P(a \mid d, t, v, s, h^{t,i-1})$$
$$\left(R(a, v, s, d) + \gamma(t, i) \sum_{v' \in V, s' \in S} \right.$$
$$\left. p(v', s' \mid a, v, s, d)(\min_{d'} EC_s^{\alpha(t,i), \beta(t,i)}(v', s', h^{t,i}, d')) \right) \quad (2)$$

$R(a, v, s, d)$ is the immediate reward/cost function, which in our domain is the expected energy consumption given the current world state v, setting s, advice d and the driver's reaction a. $\gamma(t, i)$ is the discount factor, which represents the difference in the significance between future rewards and present rewards. $p(v', s' \mid a, v, s, d)$ is the probability of reaching world state v' and setting s' from state v and setting s, when the receiver takes action a and the advice was d, which in our case corresponds to the probability that the world state will change from v to v' and the setting will change from s to s' between consecutive interactions. Note that not only may the settings s change according to the receiver's action, but also the world state v. For example, acceptance ($a = accept$) of suggestion d while in state v may change the internal temperature.

Intuitively, $EC_s^{\alpha(t,i), \beta(t,i)}$ sums up the expected utility in the current time period and in the future. $\alpha(t, i) = t + \left\lfloor \frac{t_l - mod(i/t_l)}{t_l} \right\rfloor$ and $\beta(t, i) = \begin{cases} i+1, & \text{if } i < t_l \\ 1, & \text{otherwise} \end{cases}$ where t_l is the limit on the number of interactions per round. $EC_s^{\alpha(t,i), \beta(t,i)}$ considers the possible responses and multiplies the probability that the receiver will choose this action $(P(a \mid d, t, v, s, h^{t,i}))$ with the sum of two elements: The first specifies the immediate reward ($R(a, v, s, d)$) and the second specifies the expected future reward. The future expected reward first depends on whether the interactions will continue at all, which will happen with a probability depending on the number of rounds and interactions $\gamma(t, i)$. Then, it depends heavily on the future world state v' and setting s'. For any possible future world state v' and settings s' that will occur with a probability of $p(v', s' \mid a, v, s, d)$ the agent will choose the best advice d' and will obtain, recursively, the expected utility from giving this advice in the next interaction $i + 1$ with the updated history $h^{t,i}$.

The advice that minimizes the sender's cost is

$$\pi^*(v, s, h^{t,i-1}) = \text{argmin}_d EC_S^{\alpha(t,i),\beta(t,i)}(v, s, h^{t,i-1}, d) \quad (3)$$

where π^* is the advice function.

Hitherto, we have dealt with all variables in the optimization problem except $P(a \mid d, t, v, s, h^{t,i})$ and $p(v', s' \mid a, v, s, d)$ which require estimations. The next subsection is dedicated to the data collection and analysis which in turn resulted in prediction models for the above.

4.1 Data Collection

In order to estimate $P(a \mid d, t, v, s, h^{t,i})$ we recruited 38 subjects, of whom 20 were males and 18 were females[1] ranging in age between 23 and 67, with a mean of 35.

Each subject was given 10 minutes to be in the car and he or she was free to tell the experimenter what settings to set in the climate control system once the experiment began. The experiment screen (see Figure 1) was displayed on a laptop while the central stack screen was covered. The system presented advice (if available) as a small "Go Eco-Friendly" button on the bottom left side of the experiment screen. The subject could choose to click on that button at any given moment, and thereby change the climate control setting to the advice suggested by the agent (see Figure 2). Notice that the advice itself is not presented to the driver *before* "Go Eco-Friendly" is applied. At that point, the system merely presents the energy saving that the advice would yield if applied. During the next 10 seconds the drivers could choose whether to accept the advice (keep the new settings) or reject it (return to her previous setting). If for the next 10 seconds the driver does not react, the new setting is automatically accepted[2]. Regardless of the advice, the subject could change the climate control settings in whichever way she saw fit – the experimenter updated the climate control of the car as many times as requested by the driver. While in the car the subject was given a cell phone with a driving simulator "Bus Simulator 3D"[3] to be played while the experiment was conducted. The motivation was to set the conditions similar to regular driving conditions and give the subjects something to do. After 10 minutes, the subject had to exit the car and wait until the inside of the car was warm again to simulate initial conditions. This phase took about 10 minutes while the car was switched off and the car doors and trunk were left open. Then, the subject reentered the car to simulate a new trip. This process was then repeated again. That is, each subject spent 30 minutes in the car in three 10-minute sessions and an additional 20 minutes waiting between the different rounds. Including the collection of demographic data and payment (the subjects were paid 100 NIS each, the price of a fancy lunch in Israel), each subject spent about 60 minutes in our garage.

In order to have a variety of advice and drivers' responses, we had to develop algorithms for repeated advice provision. The goal of these algorithms was to simulate varying interactions between the agent and different subjects. Simulating different interactions is challenging. The time with each driver is limited, thus testing all possible advice in every possible order is impossible. Consequently, we needed to collect

---

[1] All experiments with human subjects were approved by the corresponding IRB.

[2] We designed several GUI's to provide advice and carried out an experiment to identify the most preferred GUI.

[3] Available free at Google Play store.
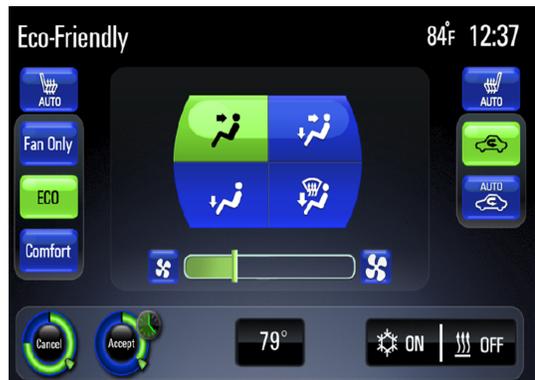
Figure 2: Advised screen.

data on "reasonable" interactions, that is, interactions which we believed people might consider accepting as the agent's advice.

We used the advice that would have been generated by CARE [5] as a starting point. We removed the initial comfort level parameter from CARE's comfort model and retrained the model using the data that the authors originally collected. We emphasize that the goal of the experiment was not to evaluate these algorithms but rather to collect data on drivers' behavior as the basis for the development of a methodology for advice provision in repeated interactions. During the experiment we logged all of the actions performed by the agent, all of the actions performed by the subjects and the energy consumption of the car.

We implemented two algorithms; The *Pusher* algorithm, which advises the settings that maximize the comfort level but consume no more than the average between the current consumption and the "ideal consumption" – that is, CARE's advice. The intuition behind this algorithm is that the system will always try to persuade the subject to choose the "ideal" settings. The *Lenient* algorithm, which advises the settings that maximize the comfort level but consume no more than the "ideal consumption" (CARE's advice) plus the delta between the user's maximal consumption and the current consumption. The intuition behind this algorithm is that if the driver is willing to compromise and consume less energy (than her maximal consumption) the system is also willing to compromise and offer more "generous" advice (in terms of energy consumption).

Nineteen subjects participated in the experiments with the Pusher algorithm. The average number of times in which new advice was suggested to a subject per session was 3.44. In 74% of the cases the subject chose to consider the advice and clicked the "Go Eco-Friendly" button. Furthermore, in 59% of these cases she chose to accept the advice. Similarly, 19 subjects participated in the experiments with Lenient. The average number of times that new advice was given to a subject per session was only 1.18. Moreover, in only 59% of the cases in which new advice was proposed, the subject chose to consider the advice and chose the "Go Eco-Friendly" button. Furthermore, in 59% of these cases she chose to accept the advice. There was no significant change between the sessions.

We compared the energy consumption per condition. While in the first condition the consumption was slightly higher (0.1605KWH vs 0.1525KWH), the difference was not statistically significant. It is also interesting to observe that the

Table 1: List of Features

notation	meaning
$save$	d's energy saving percentage of the advice.
Δ_T	d's temperature difference from current s_T.
Δ_F	d's fan difference from current s_F.
Δ_I	d's temperature difference from current I.
$d\%$	the advice acceptance rate thus far.
$d\%^t$	the advice acceptance rate in round t.
$acc\Delta_T$	the average Δ_T of accepted advice so far.
$acc\Delta_F$	the average Δ_F of accepted advice so far.
$rej\Delta_T$	the average Δ_T of rejected advice so far.
$rej\Delta_F$	the average Δ_F of rejected advice so far.
$acc\Delta_T^t$	the average Δ_T of accepted advice in round t.
$acc\Delta_F^t$	the average Δ_F of accepted advice in round t.
$rej\Delta_T^t$	the average Δ_T of rejected advice in round t.
$rej\Delta_F^t$	the average Δ_F of rejected advice in round t.

energy saving average "Go Eco-Friendly" suggested to the subject was -16.86% with Pusher and -36.91% with Lenient.

As for the rounds: in the first round 0.76 of the times that new advice appeared, the subjects clicked on it, however, only 0.69 of the times that the subjects clicked on new advice in rounds 2 and 3. In the first round the subjects rejected 0.37 of the advice received, in the second round 0.46 and in the third 0.42. The number of pieces of advice received per subject dropped from 2.8 in round 1 to 2.6 in round 2 and 2.2 in round 3. The average saving of the advice offered dropped from -22.5% in round 1 to -20.8% in round 2 and -18.1% in round 3. The average consumption was reduced from 0.162 in round 1 to 0.155 in rounds 2 and 3. Combining all of this data, it seems that although the subjects received less advice in the latter rounds, they apparently consumed less energy. This is because the subjects were probably influenced by the first round and consumed less energy to begin with and therefore required less advice. However, the advice still seemed to have a large impact on the subjects also in the later rounds.

We also separated all rounds according to the algorithms used (Pusher, Lenient). With the Pusher algorithm the acceptance rate of new advice (percent of advice in which the subjects also clicked on "Go Eco-Friendly" and also accepted the advice) did not vary much in the rounds (0.45 in the first round, 0.41 in the second and 0.49 in the third). Lenient demonstrated a major drop from the first round's acceptance rate, from 0.57 to 0.1 in the last round's acceptance rate.

4.2 Prediction Model

Recall that in order to calculate Equation 2 we need to estimate $P(a \mid d, t, v, s, h^{t,i-1})$. Given the data obtained in the Data Collection phase, denoted Λ, we extracted features, as described in Table 1, which help predict the driver's response given the advice d, the round t, the world state v, and the interaction history $h^{t,i-1}$. For each $d, t, v, s, h^{t,i-1}$ we encountered in the data we extracted the features and labeled the instances *accepted* or *rejected* according to their status. This process merely translated the interaction (which was in a textual format) to a set of vectors, whereby each is a set of features and a label – the subject's response. This vector represents a single interaction in the experiment and was used in the prediction model.

The resulting labeled set ψ was divided in a one-left-out

fashion, where for every subject j we created ψ_j as a test set and ψ_{-j} as a training set. Then, a prediction model using K-Nearest-Neighbors algorithm (KNN) was created. The Algorithm was trained on ψ_{-j} and tested over ψ_j for every j. KNN is a classical non-parametric lazy classifying algorithm in which an object is classified according to its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. The classes in our settings were *accepted* and *rejected*, and k was set to 21 as it provided the highest prediction accuracy on ψ.

Unfortunately, ψ turned out to be very unbalanced; out of the 436 instances in ψ, 324 where *accepted* and 112 were *rejected* instances. This was caused due to the use of the Pusher and Lenient agents that only advised settings which we a priori believed the driver would accept (predicted comfort of at least 7). We used 2 methods to overcome this problem: 1) We oversampled the minority group (*rejected*). 2) We synthetically injected instances in the following manner – for every *rejected* labeled instance in ψ we added rejected instances for any warmer[4] advice than the one rejected. For example, if a subject rejected a setting of $T = 23, F = 1$ she would most likely reject $T = 24, F = 1$ and $T = 25, F = 1$ as well. We were very careful with this injection and ensured that the suggestion was rejected because it was too hot and not because it was too cold. We only synthetically created instances for actual rejected instances with temperatures higher than 22 and fan lower than 3.

Of the 436 suggestions, we correctly classified 337 instances (78%), where we had greater success with *accepted* instances (85%) than with the *rejected* group (59%). Moreover, we identified which features were good predicative features and which were not. Namely, $save, \Delta_T, \Delta_F, \Delta_I, \Delta_F, d\%$ features were found to be influential whereas the rest had little (if any) effect on the accuracy rate (features appear in Table 1).

We note that the use of different machine learning algorithms such as SVM and Decision Trees provided very similar yet lower prediction rates (72%, 63%, respectively).

The use of the KNN algorithm also provided a probability measurement of $P(a \mid d, t, v, s, h^{t,i-1})$, which was needed to design the agent. We estimated $P(a \mid d, t, v, s, h^{t,i-1})$ as the number of *accepted* labeled instances among the 21 nearest neighbors divided by 21. That is, if 7 of the 21 nearest neighbors are labeled *accept* we would assume that the probability of the current advice d to be accepted, in the context in which it is given, would be 1/3. This method achieved a Mean Absolute Error of 0.37 and a Root Mean Squared Error of 0.42.

4.3 Agents

Based on both the prediction model and the energy consumption model we constructed two agents, *SAP agent* and *MACS*. MACS tries to solve the optimization problem given in Equation 3 whereas the SAP agent does not.

4.3.1 MACS

An MDP [27] is a tuple $(\mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ where

1. $\mathcal{O}$ is the set of possible states of the system (S);

2. $\mathcal{A}$ is the set of possible advice the agent can suggest; in our case it is $S \cup \{\text{"silent"}\}$;

---

[4]Data collection and experiments were conducted during the summer period.

3. $\mathcal{T}$ represents the interaction dynamics – the driver's reactions to advice and the environmental changes;

4. $\mathcal{R}$ is the energy consumption function for each $s \in S$.

Acting in an MDP results in a sequence of states and actions $o_0, a_0, o_1, a_1, o_2, \ldots$

A policy π is a sequence of mappings $(\mu_0, \mu_1, \mu_2, \ldots, \mu_{t_l})$, where, at time t, the mapping $\mu_t(\cdot)$ determines the action $a_t = \mu_t(o_t)$ to take when in state o_t. t_l denotes the limit on the amount of advice the agent can suggest.

The objective is to find an optimal policy that minimizes the expected cost accumulated over time. In particular, a policy π is good if its *Expected Cost* is low. Expected Cost is defined as:

$$ExpectedCost(\pi) = \mathbb{E}[\sum_{t=1}^{t_l} \gamma^t \mathcal{R}(o_t) | \pi] \qquad (4)$$

γ is the probability of continuing to the next round. In our environment the decision maker is the *advisor* (the agent), which has to decide which, if any, advice to provide. The driver controls the actual dynamics of the system as she controls the actual climate control settings.

We model this environment using an MDP. Each state $o \in \mathcal{O}$ consists of a world state v, the current climate control setting s, interaction number i, round number t and the advice acceptance rate thus far, d%. These states cover all possible scenarios in which the agent is required to provide advice or to keep silent. There is an infinite number of states, which require discretization. We discretized the environment by rounding degrees to the closest integer and the acceptance rate was restricted to 2 digits after the floating point. This process resulted in about 100,000 states. The current state is completely observable by the agent from the car's data and the previous interaction with the driver. As such, no uncertainty is induced as to the state in which the agent works.

The agent can only suggest climate control settings to the driver or it can keep silent, i.e. $\mathcal{A} = T \times F \times D \times M \cup \{silent\}$, where $T \times F \times D \times M$ is the space of all possible climate control settings. Notice that all of these actions are possible at each state.

$\mathcal{T}(o, d, o')$ provides the probability of reaching MDP state o' given advice d in MDP state o. In our environment this transition is controlled mainly by the driver. The driver is free to change the climate control settings whenever she chooses to, and as shown in [5], drivers are also influenced by the agent's advice, even when it is rejected. Because the drivers are free to change the climate control settings regardless of advice, it is hard to model these changes. Furthermore, the external and internal temperatures may vary during the experiment. This makes it extremely hard to accurately assess $\mathcal{T}$. For example, after advice is rejected the climate control settings are automatically switched back to the previous ones. Before the next advice appears, a time we configured to 60 seconds (as proposed by our engineers) to avoid distracting the driver, the driver can change the CCS setting in whichever way she wants and the internal temperatures tend to decline. We could not detect any clear behavior of the driver's setting changes except for when a driver was considering whether to accept or reject advice. Thus, we assumed that the climate control settings change only by explicitly accepting them and/or at the beginning of each round. This assumption is restricted only for modeling — in actual implementation the

driver receives advice according to her actual state, regardless of how she reached that state — namely by advice or by manual changes. Due to the short time intervals in which the experiment is conducted (10-minute episodes), the external temperature is unlikely to change and therefore was assumed to remain constant. The internal temperature on the other hand tended to change, mostly decreasing during the experiment. We estimated these changes using $p_c(v' \mid a, v, s, d)$ which was estimated using another KNN model (with k again set to 21). Each instance in ψ was translated into a feature vector containing a, v, s, d and was labeled with the change of the I during the 60 second intervals between the different suggestions. For example, given $I = 35, E = 36$, $T = 19, F = 2, D = 0, M = 0$, the advice of $T = 22, F = 1, D = 0, M = 0$ which was *accepted*, we estimated a 9.5% chance of I decreasing by 1 degree Celsius and a 90.5% chance of I not changing. This was due to the fact that 2 of the 21 nearest neighbors experienced a decline of 1 degree Celsius and 19 did not. In addition, the driver can choose to ignore the advice by avoiding clicking on the "Go Eco-Friendly" button. This factor was also learned statistically by a Maximum Likelihood Estimator, denoted $p_s(stop \mid t, i)$. That is, given t, i, we estimated the probability of an interaction to stop at t, i (not to continue to $i+1$) by dividing the number of instances in round t and interaction i which did not continue to $i + 1$ by the number of instances of round t and interaction i in ψ.

$\mathcal{T}$ is derived from the prediction model described above, $p_c(\cdot)$ and $p_s(\cdot)$. The prediction model provides us with the probability of suggestion d being accepted in state o, which leads us to a new state – o'. Yet, regardless of the estimation, $p_c(\cdot)$ and $p_s(\cdot)$ are applied and therefore there are multiple possible outcomes: "accept" with I declining by 1 degree Celsius, "accept" with no change to I, "ignore" etc. . . If we predict that the driver will stop the interaction we again assume she will keep the current setting of the climate control system.

The $\mathcal{R}$ function is naturally derived from the previously described Energy Consumption Model. At each phase the agent faces a cost, which is the energy consumed by the CCS. The accumulation of all costs encountered during the interaction is the payoff to the agent, which it tries to minimize.

Recall that t_l is the horizon of the interaction, which in our experimental setting is set to 9 (3 rounds, 3 pieces of advice per round). t_l was chosen according to GM's experience in designing driver-interaction systems. γ is 1 if $t \le 9$, and otherwise 0. This is due to the fact that we do not distinguish between the different rounds in terms of energy consumption. t in Equation 4 indicates the current number of the interaction ($t \le 9$).

We solved the MDP problem using Dynamic Programming [22] and received a policy π^*, which determines which advice d (if any) to provide in every state o. We note again that π^*, an optimal policy for the given MDP, does not take into account the manual changes the driver can perform. Nevertheless, if the driver reconfigures the climate control setting manually (by changing her climate control setting) the suggestion will be according to her new state. We anticipate that π^* will still provide solid recommendations.

To deploy π^* in the GM Chevrolet Volt car, we enumerated all states and advice as $< o, \Pi^*(o) >$ pairs and saved them in a table. The agent loads the table at the begin-

ning of the interaction and simply identifies each state (when needed) and provides the specified advice.

4.3.2 SAP agent

According to social preference theory, people consider the outcomes of others as well as their own when making strategic decisions. SAP modeling explicitly reasons about the trade-offs between the costs to both participants in the selection process based on a social weight [3]. SAP modeling provides advice that maximizes a social utility function which is a weighted sum of the agent and human's utilities. SAP uses simulation runs of repeated human-agent interaction to identify the weights that maximize the agent's utility over time.

In climate control settings the agent's utility is based on the energy consumption and the driver's utility is based mainly on her comfort level. Unfortunately, the driver's comfort level (even if the driver is able to quantify it) is not available to the agent in a real car deployment. In [3], the authors asked the participants explicitly to quantify their comfort level on a scale of 1 to 10 where 7 means —"I'm comfortable; I would like to drive under these conditions". However, preferences over final outcomes vary according to the reference point from which those outcomes are judged [15]. That is, the comfort level setting s provided depends heavily on the last setting the driver experienced. Therefore, the proposed comfort level function is *not adaptive* to the reference point and the projected comfort level from setting s relies solely on s's components – s_T, s_F, s_D and s_M. Nevertheless, we wanted to test the predicative ability of the comfort level function (as calculated in [3]). An analysis of Λ (see Section Data Collection) reveals that although suggesting only settings in which the comfort level suggested will result in a comfort level of 7 (at least), we encountered 41% misclassifications – that is, 41% of the suggestions were rejected despite their expected comfort level being 7 or more.

In order to work around this problem we propose to estimate the driver's comfort using the prediction model described above. We assume that there is a strong correlation between the acceptance probability of a suggestion and the utility that the suggestion provides to the driver – the higher the probability of acceptance, the higher the utility of the driver and vice versa.

The SAP agent provides advice d which maximizes the following social utility function u, given a defined weight w.

$$u(d) = -w \cdot e(d) + (1-w) \cdot P(accept \mid d,t,v,s,h^{t,i-1}) \quad (5)$$

That is, given w (which is defined offline) and $t,v,s,h^{t,i-1}$ the SAP agent searches for advice d which maximizes u. This advice will be presented to the driver.

In order to compute the optimal weight w^* of the social utility function we performed an (offline) exhaustive search over all possible weights in the interval $[0,1]$ in steps of 0.01, each simulated with 100,000 games of 9 interactions. At each interaction in the simulation, similar to MACS, the SAP agent provides advice while maximizing Esquation 5. If the advice is considered by the simulated driver (using $p_s(\cdot)$), $P(accept \mid d,t,v,s,h^{t,i-1})$ was applied to simulate the driver's response. Immediately thereafter, the change in I using $p_c(\cdot)$ is simulated.

In order to avoid online calculations we generated all states offline and created a table of $< s,d >$.

4.4 Empirical Evaluation

The aforementioned agents, MACS and SAP agent, alongside a *Silent* agent (that does not provide any advice), were tested in the exact same fashion as described in the Data Collection section above.

We recruited 45 subjects in order to evaluate the 3 agents. Of the 45 subjects, 23 were males and 22 were females ranging in age from 23 to 72 with a mean of 41.

Each subject spent about 60 minutes in our garage. Due to that fact, we could not implement a within-subject experimental setting (it would take too long). In our between-subjects experimental setting the testing of policies in extremely similar world-states is important for the integrity of the results. All subjects who took part in this stage of the research participated in the experiments between mid-August 2014 and mid-September 2014. In these 4 weeks, between $11:00-15:00$ (the time of the experiment), temperatures in our garage were almost constant. All participants experienced temperatures between 35-37 degrees Celsius, *without any exceptions.*

4.4.1 Results

The MACS and SAP agent were limited to 3 pieces of advice per round (9 overall), yet neither of them reached the 9 suggestion limitation. MACS offered 6.07 pieces of advice to each subject whereas the SAP agent offered 6.13 suggestions (on average). Of the 91 suggestions MACS offered its 15 subjects, 84% of the time (76) "Go Eco-Friendly" was clicked, and 58% of such clicks resulted in the suggestion being accepted (44). On the other hand, SAP offered 92 pieces of advice to its 15 subjects, yet only 64% of the time "Go Eco-Friendly" was clicked (59), and 54% of such clicks resulted in the suggestion being accepted (32).

When we break down the data according to rounds, we can see that the low click rate of the SAP agent starts from the first round and keeps declining. In the first round 67% of the time new advice was presented, the subjects clicked on "Go Eco-Friendly". However, in the second and third rounds a decline was recorded – 65% and 59%, respectively. The amount of advice given per round also declined from 2.2 in the first round to 2.1 in the second and 1.8 in the third. However, the average savings from the advice offered increased from 32.5% in the first round to 33.3% and 35.8% in the second and third rounds, respectively. The average energy consumption varied across the rounds, where in round 1 an average consumption of 0.249KWH was recorded and only 0.216KWH was recorded in the second round. In the third round the consumption increased to 0.247KWH.

MACS, on the other hand, kept a relatively constant click rate (the percentage of times new advice was considered via the "Go Eco-Friendly" button) across the different rounds — 85%,82%,83%, respectively. Similarly to the SAP agent, in the first round 2.2 pieces of advice were presented to the subject (on average). Yet in the second round it offered only 1.8 pieces of advice and 2 in the third round. As for the energy consumption, in the first round 0.191KWH on average was recorded whereas in the second and third rounds 0.158KWH and 0.173KWH were recorded, respectively.

4.4.2 Analysis

A total of 45 participants took part in the experiment. We used a between-subject design, with three levels of conditions (MACS, SAP agent, and Silent). Each participant
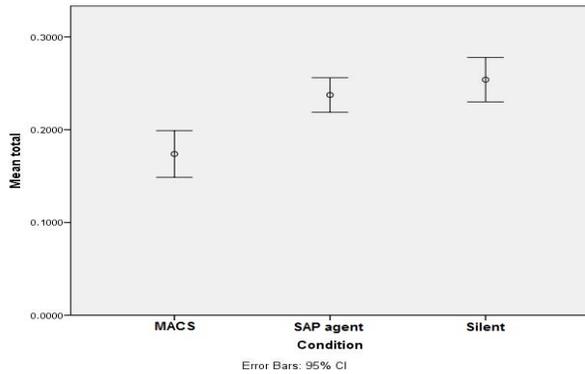
Figure 3: Agents' energy consumption per round (on average, in KWH).

performed the task under one of the three conditions and performed three consecutive runs. The data was analyzed with a univariate ANOVA, with condition and repetition as the independent factors and the energy consumption as the dependent variable. Post-hoc comparisons among the three conditions were also run.

The energy consumption was significantly affected by the condition, $F(2,126)=14.0$, $p < 0.001$, $\eta^2 = 0.18$. Figure 3 shows the mean energy consumption (per round) and 95% confidence intervals under the three experimental conditions. Post-hoc comparisons among the conditions (using Bonferroni correction for multiple comparisons) showed that MACS was significantly different from both SAP and Silent agents, $p < 0.001$. On average, a subject using MACS consumed 33% less energy than the benchmark group of Silent. Subjects who used the SAP agent consumed 5% less energy than subjects equipped with Silent, though this difference was not found to be statistically significant using the post-hoc comparison.

4.5 Discussion

A decline in the clicking rate (the percentage of times new advice was considered by clicking on "Go Eco-Friendly") is natural as the subjects learn different settings and perhaps find a setting in which they are unwilling to compromise. Therefore, there is a small decline in the pieces of advice provided by the agents as the experiment proceeds. Both MACS and the SAP agent offered 33 pieces of advice in the first round opposed to 30 and 27 in the last round (respectively).

Our results show that MACS outperforms the SAP agent in our repeated interaction CCS settings. This result contradicts previous results presented in [4], which have shown that the authors SAP based agent outperformed their MDP based agent. However, recall that in our implementation of SAP we used the probability estimation of the driver to accept a suggestion rather than the driver's actual utility function which we could not articulate. Learning the driver's precise utility function, if one even exists, may be possible, but would certainly require a lot more data on each driver.

Our findings show that the SAP agent was much more *aggressive* in its suggestions than MACS. On average, the SAP agent offered to reduce 33.7% of the energy consumption in its advice, whereas MACS offered to reduce the energy by 23.1%. It seems that this aggressiveness was the reason why some subjects stopped trying the advice. Those who did

choose to continue to click on the "Go Eco-Friendly" button demonstrated a relatively good acceptance rate (54%).

In the implementation of both MACS and SAP agents, we used a KNN prediction model to assess $P(a \mid d, t, v, s, h^{t,i-1})$. We reevaluated the model to examine whether its predictions of drivers' reactions to advice were accurate w.r.t the observed interaction with MACS and SAP agents. When we tested the model on the data collected by the Pusher and Lenient algorithms (using the 1-left-out methodology) we revealed a prediction accuracy of 78%, a Mean Absolute Error of 0.37 and a Root Mean Squared Error of 0.42. We retested the prediction model using the interactions with MACS and SAP agents as a *test-set*. The model's prediction accuracy was found to be 76% while its Mean Absolute Error and Root Mean Squared Error were 0.39 and 0.41, respectively. These findings suggest that our prediction model is able to generalize across different drivers and agents.

It has been shown in literature that women are more likely to "express thermal dissatisfaction" and are more sensitive to cooler conditions in *indoor* experimentation than men [14]. Our data also shows a significant difference between men and women in automotive CCS settings. On average, a woman equipped with one of the three agents (MACS, SAP agent and Silent) set her CCS to 22 degrees and a fan speed of 2, whereas a man, set his CCS to 20 degrees temperature and a fan speed of between 3 and 4 on average. Consequently, men consumed 0.72KWH and women consumed 0.59KWH (on average, per subject) in 3 rounds total. This difference was statistically significant using post-hoc testing ($p = 0.02$). Although women accepted more advice than men on average (59% vs. 51%) the difference was not statistically significant and does not provide an explanation for the difference in energy consumption.

5. CONCLUSIONS

In this work we present a methodology for the development of a repeated interacting agent for automobile climate control systems, MACS. MACS offers adaptive advice which considers the drivers' reactions and the long term effect of each piece of advice. The use of machine learning techniques enabled us to satisfactorily predict different drivers' reactions to different advice, which in turn was used for the modeling of MACS. This prediction and modeling resulted in an adaptive policy which was found to be beneficiary – drivers equipped with MACS consumed 33% less energy (on average) than those who were not.

We can conclude that an MDP modeling for the long-term effect of advice is good practice for advising agents operating in environments with repeated settings. The use of machine learning techniques helped us bridge over the lack of prior knowledge of the human's preferences and desires which is required to correctly model interactions. Despite the very few examples, which required simulating different repeated interactions in a reasonable fashion, and having limited prediction accuracy over the minority group we were able to generalize our findings to provide solid prediction accuracy and beneficiary adaptive advising policies.

The methodology presented herein can be used in different domains and settings which require strategic advice provision by self-interested agents and is not restricted to automotive climate control systems.

Our agent and methodology are being considered for implementation in future GM cars.

REFERENCES

[1] S. Abras, S. Ploix, S. Pesty, and M. Jacomino. A multi-agent home automation system for power management. In *Informatics in Control Automation and Robotics*, pages 59–68. Springer, 2008.

[2] A. Al Mahmud, P. Dadlani, O. Mubin, S. Shahid, C. Midden, and O. Moran. iparrot: towards designing a persuasive agent for energy conservation. In *Persuasive Technology*, pages 64–67. Springer, 2007.

[3] A. Azaria, Z. Rabinovich, S. Kraus, and C. V. Goldman. Strategic information disclosure to people with multiple alternatives. In *AAAI*, 2011.

[4] A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and Y. Gal. Strategic advice provision in repeated human-agent interactions. In *AAAI*, 2012.

[5] A. Azaria, A. Rosenfeld, S. Kraus, C. V. Goldman, and O. Tsimhoni. Advice provision for energy saving in automobile climate control systems. In *AI-magazine*, 2015.

[6] B. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies)*. Morgan Kaufmann, 2002.

[7] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1043–1052. ACM, 2009.

[8] J. Froehlich, L. Findlater, and J. Landay. The design of eco-feedback technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2010.

[9] J. Gonder, M. Earleywine, and W. Sparks. Analyzing vehicle fuel saving opportunities through intelligent driver feedback. Technical report, SAE Technical Paper, 2012.

[10] J. Ham and C. Midden. A robot that says bad!: using negative and positive social feedback from a robotic agent to save energy. In *Proc. of the 4th ACM/IEEE international conference on HRI*, pages 265–266. ACM, 2009.

[11] N. Hazon, R. Lin, and S. Kraus. How to change a group's collective decision? In *Proc. of the Twenty-Third international joint conference on Artificial Intelligence*, pages 198–205, 2013.

[12] W. Hill, D. Lebut, G. Major, and F. Schenkel. Affect of windows down on vehicle fuel economy as compared to AC load. Technical report, IEA, 2004.

[13] F. Jazizadeha, G. Kavulyaa, J.-Y. Kwakb, B. Becerik-Gerberc, M. Tambe, and W. Woode. Human-building interaction for energy conservation in office buildings. In *Proc. of the Construction Research Congress*, pages 1830–1839, 2012.

[14] S. Karjalainen. Thermal comfort and gender: a literature review. *Indoor air*, 22(2):96–109, 2012.

[15] B. Kőszegi and M. Rabin. A model of reference-dependent preferences. *The Quarterly Journal of Economics*, 121(4):1133–1165, 2006.

[16] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *UbiComp 2006: Ubiquitous Computing*, pages 243–260. Springer, 2006.

[17] J.-y. Kwak, P. Varakantham, R. Maheswaran, M. Tambe, F. Jazizadeh, G. Kavulya, L. Klein, B. Becerik-Gerber, T. Hayes, and W. Wood. Saves: A sustainable multiagent application to conserve building energy considering occupants. In *AAMAS*, pages 21–28. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[18] H. Lee and G. Lovellette. *Will electric cars transform the US vehicle market?: An analysis of the key determinants.* Harvard Kennedy School, Belfer Center for Science and International Affairs, 2011.

[19] J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (TRIP): Incorporating personalization into route planning. In *Proc. of the IAAI*, volume 21, pages 1795–1800, 2006.

[20] H. Oinas-Kukkonen. A foundation for the study of behavior change support systems. *Personal and ubiquitous computing*, 17(6):1223–1235, 2013.

[21] J. Pontes. World Top 10, May 2014. Technical report, EVSales.com, 2014.

[22] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*, volume 414. John Wiley & Sons, 2009.

[23] P. Reed and B. Romans. We test the tips part ii, save gas with smart driving and slick aerodynamics. Technical report, edmunds.com, 2009.

[24] G. Shani, R. I. Brafman, and D. Heckerman. An MDP-based recommender system. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 453–460. Morgan Kaufmann Publishers Inc., 2002.

[25] K. Torkkola, K. Zhang, H. Li, H. Zhang, C. Schreiner, and M. Gardner. Traffic advisories based on route prediction. In *Proceedings of Workshop on Mobile Interaction with the Real World*, pages 33–36, 2007.

[26] T. Trigg and P. Telleen. Global ev outlook 2013 - understanding the electric vehicle landscape to 2020. Technical report, International Energy Agency, 2013.

[27] C. C. White III and D. J. White. Markov decision processes. *European Journal of Operational Research*, 39(1):1–16, 1989.

[28] K. Young, M. Regan, and M. Hammer. Driver distraction: A review of the literature. *Distracted driving. Sydney, NSW: Australasian College of Road Safety*, pages 379–405, 2007.

[29] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 322–331. ACM, 2008.