Learning Driver's Behavior to Improve Adaptive Cruise Control

Avi Rosenfeld[1], Zevi Bareket[2], Claudia V. Goldman[3],
Sarit Kraus[4], David J. LeBlanc[2] and Omer Tsimoni[3]
[1]Department of Industrial Engineering
Jerusalem College of Technology, Jerusalem, Israel 91160
[2]University of Michigan, Transportation Research Institute, USA
[3]General Motors Advanced Technical Center, Israel
[4]Department of Computer Science, Bar-Ilan University, Israel
rosenfa@jct.ac.il, bareket@umich.edu, claudia.goldman@gm.com,
sarit@cs.biu.ac.il, leblanc@umich.edu, omer.tsimhoni@gm.com

ABSTRACT

Adaptive Cruise Control (ACC) is a technology that allows a vehicle to automatically adjust its speed to maintain a preset distance from the vehicle in front of it based on the driver's preferences. Individual drivers have different driving styles and preferences. Current systems do not distinguish among the users. We introduce a method to combine machine learning algorithms with demographic information and expert advice into existing automated assistive systems. This method can reduce the number of interactions between drivers and automated systems by adjusting parameters relevant to the operation of these systems based on their specific drivers and context of drive. We also learn when users tend to engage and disengage the automated system. This method sheds light on the kinds of dynamics that users develop while interacting with automation and can teach us how to improve these systems for the benefit of their users. While accepted packages such as Weka were successful in learning drivers' behavior, we found that improved learning models could be developed by adding information on drivers' demographics and a previously developed model about different driver types. We present the general methodology of our learning procedure and suggest applications of our approach to other domains as well.

1. INTRODUCTION

Cruise control is a known technology that aids drivers by reducing the burden of controlling the car manually. This technology controls the vehicle speed once the user sets a desired speed. Cruise control is not only convenient, but it has the potential to improve the flow of traffic [15], and can be effective in reducing driver fatigue and fuel consumption [1]. In this paper, we focus on a second generation of cruise controls– adaptive cruise control (ACC). ACC is designed as a comfort-enhancing system, which is an extension of conventional cruise control (CC). The ACC system relieves the driver from some of the longitudinal-control tasks by actually controlling speed and headway keeping, but the driver can choose to engage or disengage the ACC at any time. The major difference between ACC and CC is the use of radar technology to maintain a preset distance between the vehicle with the ACC and other vehicles on the road. This distance is controlled by a "gap" parameter which sets the minimum gap (headway distance) to the vehicle in front of it. Figure 1 shows a picture of a steering wheel with the ACC technology. Note the existence of a "gap" switch on the left side of the figure.

While ACC adds more automation to the driving experience, it typically also requires the driver to set and adjust one more parameter, the gap setting. The current approach is to preset the gap setting to a default value which can be adjusted by the driver manually based on his driving preferences. Another approach taken in previous published attempts was to learn this setting focusing on mechanisms such as fuzzy logic [8, 9]. In these previous approaches, rules were learned manually after having interviewed human drivers. Based on these rules the gap setting value was adjusted automatically to the conditions of the drive without considering the particular driver in the vehicle. Individual drivers, however, differ in their driving styles and preferences. Therefore, a personalized learning approach may be valuable.

In this paper, we primarily focus on a method that learns how to quickly and accurately adjust the gap setting based on the specific driver and context of a drive. To accomplish this task, we created general driver profiles based on an extensive database of driving information that had been collected from 96 drivers [5]. We used post-processing of data from that study. Our general method is that once a new driver is identified we classify this driver as being similar to previously known drivers and set the initial gap setting accordingly.

The challenge of this study was to process real world data so as to obtain the most accurate and practical rules from the learning algorithms. We found that the information gleaned from demographics and the driver's type was crucial for creating more accurate learning models. This work focuses on which attributes will help, and a general methodology for adding them. By following this methodology, we found that a better application could be created in this domain, and are confident that better applications can be created in other domains as well.

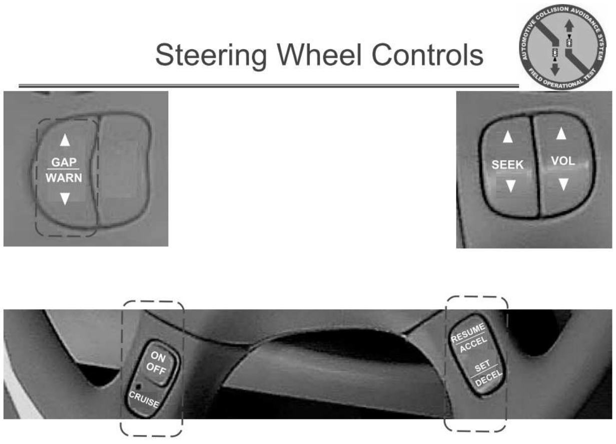

Figure 1: A steering wheel fitted with ACC technology.

This paper is organized as follows. In Section 2 we introduce related work and specify the driver type information that aided in modeling driving behavior. Then, Section 3 describes the theoretical and implementation challenges the ACC learning agent must overcome. Section 4 details the extensive database used to create the ACC agent. Section 5 provides empirical support for the success of effectiveness of this agent. Finally, Section 6 concludes and presents possible directions for future research.

2. RELATED WORK

The concept of using a group of characteristics to learn people's behavior has long been accepted by the user modeling community. Many recommender systems have been built on the premise that a group of similar characteristics, or a stereotype, exists about a certain set of users [12]. Even more similar to our work, Paliouras et. al [10] suggested creating questionnaires, distributing them, and then creating decision trees to automatically define different groups of users. Similarly, our application assumes that some connection exists between users, which can be learned using machine learning techniques. We propose that this approach be applied to customize settings within an application, here ACC, and not within recommender systems.

Previously, Fancher et. al [7], analyzed a group of 36 drivers and their acceptance of adaptive cruise control (ACC). While all drivers enjoyed and accepted the ACC, they found that drivers could be divided into three types with each group demonstrating specific driving tendencies which impact their headway and closing speeds, relative to vehicles ahead. In very general terms, these groups were assumed to be: one that is most aggressive, another that is least aggressive, and a third that is in between. Although it is clear that more detailed grouping may exist, and that a different profiling of the drivers' population can be made, for the purpose of this study the characterization analysis was aimed at identifying the above three grouping types. The three driving styles are: 1. Hunters (aggressive drivers who drive faster than most other traffic and use short headways); 2. Gliders (the least aggressive drivers who drive slower than most traffic or commonly have long headways); and 3. Followers (whose headways are near the median headway and usually match the speed of surrounding traffic). In this scheme of things, Hunters are drivers who tend to drive faster than the surrounding flow and they tend to travel at shorter headway times than those adopted by other drivers. In contrast, at the other end of driver characteristics, Gliders tend to travel slower than the surrounding flow and they tend to travel at longer headway times than those adopted by other drivers. Between the Hunters and Gliders lie the Followers who tend to go with the flow of traffic. They tend to adapt their driving behavior to the situation they are in.

The idea of assisting the driver in the task of longitudinal control has been the focus of research in the last decade [8, 9]. Operation tests have given insight into this task. However, the goal of this project was to attempt to create an intelligent ACC agent that could potentially set this longitudinal value autonomously through adjusting its gap setting per each driver.

In this paper, we use driver characterization into types (hunter, glider or follower) in addition to other demographic information to attempt to build an application that predicts how the ACC should set its gap (headway) given this information and road situation. In general, other research has previously found that we can better predict people's behavior by combining relevant behavior theory, here about people's driving type and demographics, in conjunction with machine learning methods. These studies have included how other behavior theories: Aspiration Adaptation [14] and the Focal Points [18] could be used in conjunction with machine learning algorithms to create an improved classifier. These results also showed some positive correlation between the complexity of the problem domain and the improvement in performance when augmenting the behavior model. Thus, the more complex the learning task, the added gain in the learning model by adding behavior information. This paper explores how the behavior model of a driver's type impacts their gap setting.

3. LEARNING METHOD

Current ACC systems allow the user to choose a value for the gap setting between six possible values (1–6). These values control the distance the ACC autonomously maintains with the vehicle in front of it. Currently, one value is set as default (in our case this value was 6) and the user may change it during his drive as he wishes. In order to study the problem of predicting what gap setting a person would select, we constructed two different types of models. The first type of model was a regression model which attempted to predict the number a given driver would select given the current driving conditions. The second type of model was a decision tree model which treats each number within the system as discrete values representing different categories a driver can choose. Our goal was to use the output of either model to automatically set the gap setting. Towards this goal, the second model is seemingly the better choice as its output directly correlates to a value within the system. In contrast, the regression model outputs a decimal value (e.g. 3.5) that must be first rounded to the closest value within the system to be used. However, the advantage of this model is that a mistake between two close values (e.g. 3.5 being close to 3 and 4) is not as mathematically significant as mistakes between two extreme values (e.g. between 2 to 6). In contrast, the discrete decision tree model weighs all types of errors equally. In practice, the regression model will likely be more useful if the user is willing to accept errors between two similar values.

Additionally, we focus on two secondary goals, when the ACC is first engaged, and when the ACC is disengaged. Here, the goal was not to create an agent to autonomously engage or disengage the ACC. However, by analyzing when people are most comfortable with the ACC, we hope to understand the user acceptance of such systems.

In both of these learning tasks, we are confronted by the known dataset imbalance problem [2]. In many real-world problems, as is the case here, each class is not equally represented. In fact, in the specific case of the ACC engagement task, over 90% of manual driving cases continue their manual driving, and in only a small percentage of cases do people engage the ACC. From a statistical perspective, a classifier could then naively classify all cases as being in the majority case and still have extremely high accuracy. However, because only the "minority" cases are relevant, novel methods are needed to find them. While several algorithms exist, we specifically focused on the MetaCost algorithm [4] because of its flexibility in controlling the bias size given to the minority case.

A second key implementation challenge lie in the algorithms themselves. While we used the popular Weka learning package [16] to implement all learning algorithms, the content experts often believed that the resulting models were extremely overfitted. Unfortunately, many theoretical learning algorithms are prone for overfitting when applied to real-world datasets [17] and the content experts involved with the project found this to be the case in this domain as well. To overcome this challenge we used simplified decision trees. The idea of using simplified decision trees is not new, and a variety of algorithms have been developed for simplifying decision trees [6]. However, these algorithms were developed for **increased** performance. In this application, we intentionally sacrificed a certain level of performance to reduce overfitting. Thus, we applied these algorithms for a different reason than the one they were developed for, but still achieved the desired result – a non-overfitted decision tree.

Specifically, we used the reduced error pruning method developed by Quinlan [11], named REPTree with in the Weka learning package [16]. According to this approach, a decision tree of maximum height T_{max} is reduced to T_{Depth}, with $T_{Depth} \leq T_{max}$, ostensibly to produce improved performance. In our application we chose the maximum value for T_{Depth} that the content experts deemed was not overfitted, as we found that the best performance was achieved when $T_{Depth} = T_{max}$ and our goal was to achieve the best performance within the model without producing an overfitted model. Empirical results detailing specifics of the models used to create the ACC's agent are explained in the next section.

4. EXPERIMENTAL SETUP

Data for our analysis were taken from the Automotive Collision Avoidance System Field Operational Test (ACAS FOT) [5]. In that study, to understand how different drivers use an ACC, each of 96 drivers was presented with a vehicle fitted with the ACC which they used for a period of 4 weeks. During the first week the ACC system was not available. That is, if the driver engaged the cruise control, it simply maintained speed just like the conventional system (CC). During the next three weeks, if the driver chose to engage the cruise control, it functioned as ACC. In general, three different datasets were considered. The first, and most basic, dataset were objective characteristics that can be studied based on the location of the vehicle itself, e.g., headway distance to the lead vehicle, vehicle speed, longitudinal acceleration, road type (country, city, or highway), weather (including day or night) and road density (is there traffic). A second dataset added driver characteristics. These properties focus on driver demographics such as age, sex, income level (high, medium, low), and education level (High School, Undergraduate, and Graduate). The ACAS FOT data consists of a good mixture of these demographics with a 51% male to 49% female split, 31% young (aged 20–30), 31% middle aged (aged 40–50), and 38% older drivers (aged 60–70), and people from a variety of education and socioeconomic levels. The last dataset also logged a previously developed measure used to quantify a driver's behavior [7].

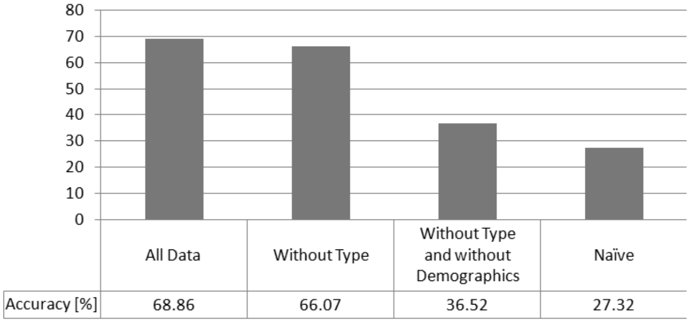

Figure 2: The importance of driver type and demographics in predicting the gap setting within the ACC for a discrete decision tree model.

The experimental design of the ACAS FOT was a mixed-factors design in which the between-subjects variables were driver age and gender, and the within-subject variable was the experimental treatment (i.e. ACAS-disabled and ACAS-enabled). The disabled period was treated as a baseline measure, since the research vehicle operated like a conventional passenger vehicle. The drivers operated the vehicles in an unsupervised manner, simply pursuing their normal trip-taking behavior using the ACAS test vehicle as a substitute for their personal vehicle. Use of the test vehicles by anyone other than the selected individuals was prohibited. The primary emphasis on user selection for the field operation test was to roughly mirror the population of registered drivers, with simple stratification for age and gender. No attempt was made to control for vehicle ownership or household income levels. Thus, although the ACAS FOT participants may not be fully representative of drivers who might purchase such a system, they were selected randomly and represent a wide range of demographic factors.

5. RESULTS

In this section we present results for the three previously defined problems: predicting a driver's gap setting within the ACC using both discrete and regression models, predicting when a driver will engage the ACC, and predicting when a driver will disengage the ACC. In all three problems we present how the driver type and other demographic information helped improve the model's accuracy. Additionally, we analyze which attributes were most significant in this application, how we avoiding overfitting, and how we addressed the dataset imbalance problem within this application.

5.1 Setting the ACC's Gap Setting

Figure 2 presents the accuracy of the decision tree model to learn a driver's preferred gap setting in the discrete model. Clearly, adding the demographic data here is crucial, as the model's accuracy drops from over 66% accuracy with this data to less than 37% accuracy without this. As a baseline, we also include the naive classifier, which is based on the most common gap setting– here the value of 6, which is also the system's default. Note that the naive model had an accuracy of nearly 27%, far less than other models. The user's type did improve accuracy, as adding this information to the type increased accuracy to near 70%. In line with our previous work [13], we hypothesized that adding this behavior model yields less significant increases if it can be learned from other attributes within the data. Here, we believed that adding information about drivers' type is less important, as their type was already evident from information such as the driver's demographics.

To support this hypothesis, we constructed a decision tree (again C4.5) to learn the driver's type. We found that this value could be learned with over 95% accuracy (95.22%) when learned with the full Reptree (T_{max})– which strongly supports our hypothesis. We present a pruned version of this tree (T_{Depth} = 4) within Figure 3. From an application perspective, we were not shocked to find that a driver's age factored heavy in their driving behavior. This characteristic is factored in actuary's insurance tables, and is a known factor in car insurance premiums [3]. Note that this characteristic was the first level below the root of the tree, demonstrating this quality. However, possibly equally interestingly is that we found education, not gender to be the next most important factor as it formed the second level within the decision tree. This factor is typically not considered by insurance companies [3], but may be worth considering. Only in the third level did we find the popular characteristic of gender to factor in, but income also weighed in as an equally important important factor. Overall, we found that young men or women with only a high school degree tended to "hunters" or those with extremely aggressive driving habits, college educated women, and people with higher degrees but lower paying jobs tended to be the less aggressive "gliders". Middle aged men with high school degrees, all middle aged people with college degrees, and people with higher degrees buy lower paying jobs also typically belonged to the middle "gliders" category. But older women with college degrees, people with low or medium paying jobs with only high school degrees, and all older people with higher degrees tend to be of the least aggressive "follower" type. Naturally, exceptions existed, and this simplified tree only is approximately 75% accurate. Nonetheless a general direction is evident from this tree, and was one that the content experts felt was not overfitted.

T_{Depth}	Accuracy [%]
2	47.55%
3	56.41%
4	62.43%
5	65.46%
6	67.51%
7	68.50%

Table 1: Analyzing the tradeoff between the model's accuracy and the height of the tree T_{Depth}.

Similarly, it was important to find a decision tree that models drivers' gap settings that is not overfitted as well. Note that the accuracy of the Figure 2 given all data is nearly 70%. However, while this value is based on the mathematically sound C4.5 algorithm [11], the content experts again felt this decision model was overfitted. We then proceeded to reduce the size of the tree as to generalize the model, thus preventing this phenomenon. However, as Table 5.1, demonstrates reducing the table size does not improve the model's accuracy, as previous theoretical works suggest [6], but did produce trees that were acceptable to the content experts. Note that raising T_{Depth} yields marginal increases in model accuracy with T_{Depth} = 7 being nearly accuracy to the result in Figure 2. In general, we found that the experts were happy with much smaller trees, but those with similar accuracy. For this problem, we display in Figure 4 the resultant tree of T_{Depth} = 4 which is only 6% less accurate the full tree in Figure 2. However, for comparison, the full tree produced with the unpruned C4.5 algorithm has a total size of 1313 leaves and branches, while the pruned tree only has a total size of 50 leaves and branches. Thus, from an application perspective, this tree was strongly favored by the experts, even at the expense of a slightly less accurate model. Note that the rules themselves are still heavily influenced by the driver type and demographic information, with driver type being the first level of the tree and the second and third levels of the tree again being primarily based on demographics such as age, gender, education, and income level.

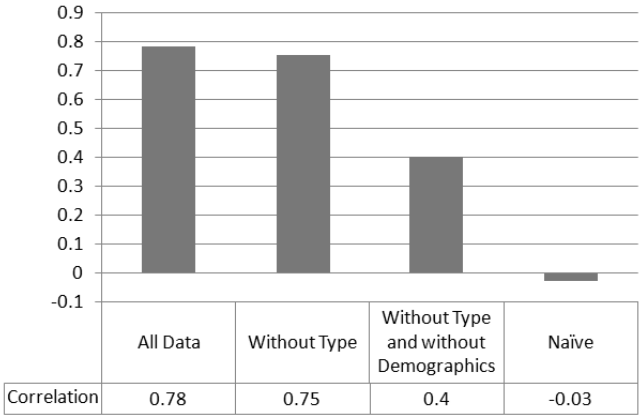

Figure 5: The importance of driver type and demographics in predicting the gap setting within the ACC for a regression model.

Similarly, we were able to create an accurate regression model, the results of which are found in Figure 4. Within these models, correlation values can range from 1.0 (fully positive correlated) to -1.0 (fully negatively correlated) with 0 be with no correlation. We found a model with both demographic and type data yielded a correlation of 0.78, while without this information the accuracy dropped to 0.75. Using only vehicle specific data yielded a model of only 0.4, and the naive model (here using the average gap value of about 3.5) yielded a value of nearly 0. Again, we found that the type only slightly improved the model's accuracy, as much of this information was already subsumed within the drivers' demographics. Here again, the experts opted for a reduced model, despite the sacrifice of slightly less accuracy.

5.2 Predicting when the Driver will Engage and Disengage the ACC

While the focus of the ACC is on the gap setting that differentiates the adaptive cruise control, from the "standard" cruise control, we also considered two additional problems: when people activate the ACC and when they deactivate it. The goal behind the gap value task was to allow an autonomous agent to set, at least initially, this value within the ACC. However, by understanding when people are more likely to use this product we can hopefully increase its

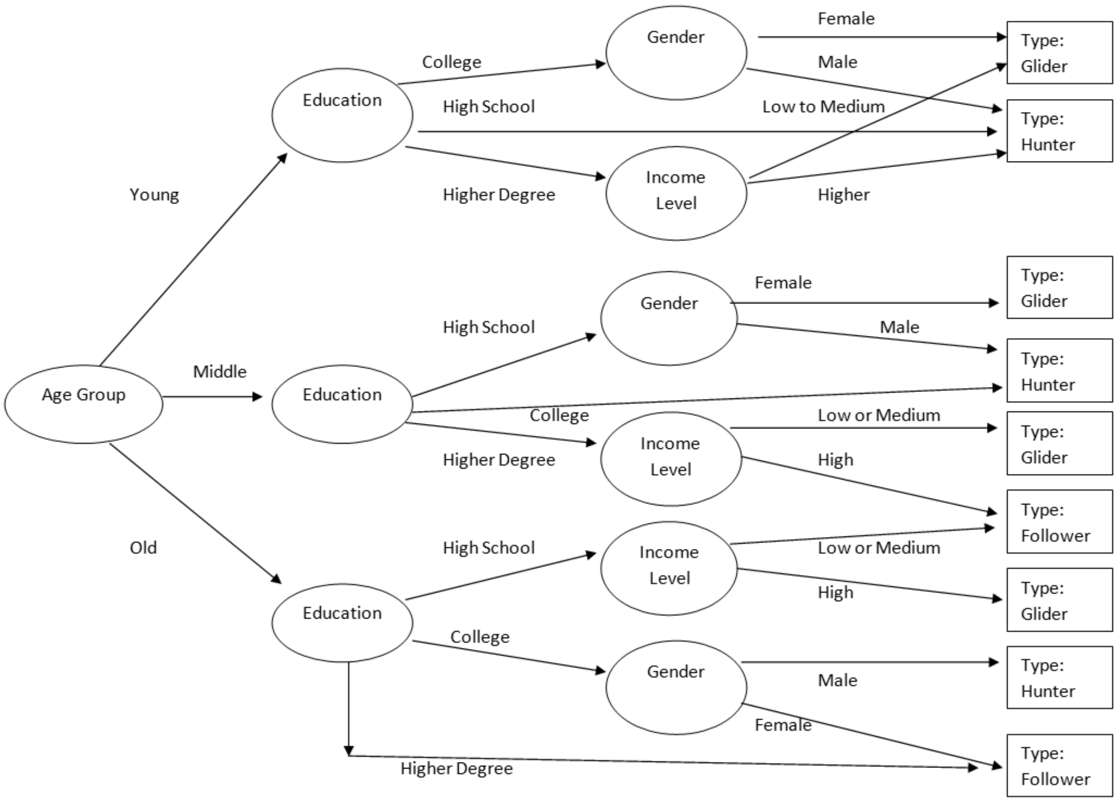

Figure 3: The decision tree for learning a driver's "type".

acceptability and use. Similarly, by understanding when people disengage the ACC we can hopefully create new generations of this technology where people will use it longer and not feel compelled to disengage it.

In both of these learning tasks, we are confronted by the known dataset imbalance problem [2]. In this paper, we constructed two models for these two problems based on the same three types of datasets. The first model is a basic C4.5 without any modification. As was the case in gap setting task, we considered attributes based on the behavior type model, driver demographics and the vehicle's characteristics. In the second model, we again used the same three datasets, but created a learning bias to find the important minority cases. We specifically focused on the MetaCost algorithm [4].

Table 2 displays the complete results demonstrating the tradeoff between a model's accuracy and the success in finding the minority cases in the task of predicting when a driver engages or disengages the ACC. The first four rows represent different models created to predict when a person would activate the ACC. The first model is the standard decision tree algorithm C4.5. In addition, we considered three weight biases within the MetaCost algorithm: 0.5, 0.7 and 0.9. Note that raising these weights allows us to give greater weight to the minority case, thus increasing the recall of cases found, but at a cost to the overall accuracy of the model. For each of these models we trained four different models: one created with all information, one without the type information but with the demographic information, one without the type and without the demographic information, and a naive model that assumes the majority case– that a person continues driving in manual mode. The accuracy of each of these models are found within the first four columns in Table 5.2, and the corresponding recall levels for these models are found in the last four columns of the table. Similarly, we also considered the task of predicting when a person turns off the ACC, and trained models based on the same four algorithms with the same four datasets. The results for the accuracy and the recall of these models are found in the last four rows of Table 2.

Ideally, one would wish for a perfect model: e.g. one with 100% accuracy and recall of all cases. Unfortunately, this is unrealistic, especially in tasks involving people which are prone to variations due to noise. Nonetheless, the overall conclusion is that by adding more information, and specifically about a person's demographics, we were able to achieve higher overall accuracies with better recall.

We would like to present the driver for a recommendation as to when to engage the ACC. Towards this goal, we wish to build a model that is as accurate as possible, but with a minimum threshold. Thus, we wished to set the desired confidence level of the model, as found based on the recall of the minority class, before presenting a recommendation to the user. Figure 6 displays the interplay between the overall model's accuracy and the recall within the minority cases in the task of predicting when a driver engages the ACC. Again, the most desirable result is one in the upper right corner– high accuracy **and** recall. However, as one would expect, and as evident from Table 2, the naive case of continuing without engaging the ACC constitutes over 91% of the cases, but this

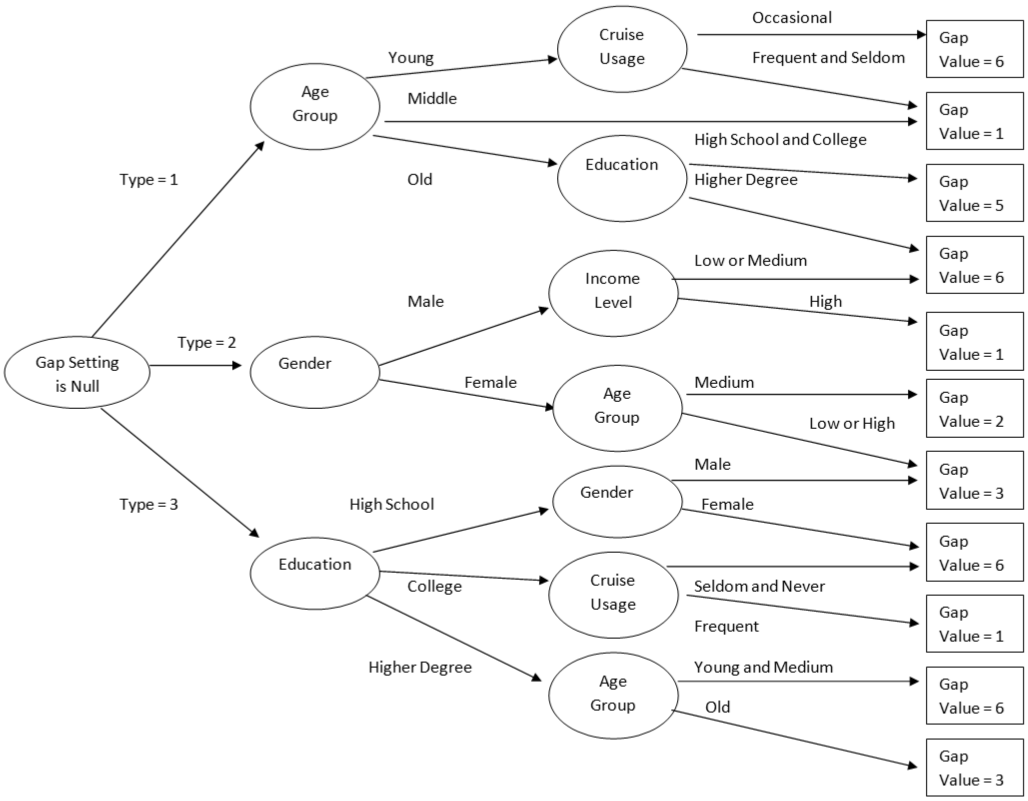

Figure 4: The decision tree for learning the ACC's Gap Setting for T_{Depth} =4.

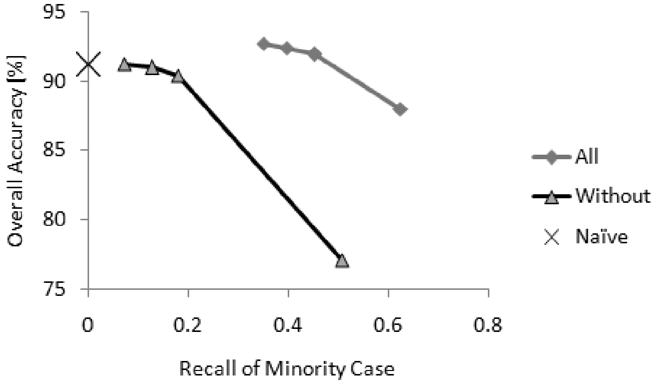

Figure 6: Comparing the overall model accuracy and recall for cases for engaging the ACC

model will have recall of 0 for the minority case. By modifying the weights within the MetaCost algorithm we are able to get progressively higher recall rates over the basic decision tree algorithm. Also note that the model trained with all information achieves significantly better results than one without the type and demographic information.

Similarly, Figure 7 displays the same interplay between the overall model's accuracy and the success in finding the minority cases in the task of predicting when a driver disengages the ACC. In this task, the naive case assumes that the driver will continue with the ACC constitutes over 86% of the cases, but this model will have recall of 0 for the minority case (see the left side of Figure 7). Note that we were again able to raise the recall within the minority case by creating weight biases of (0.5, 0.7 and 0.9), but again at the expense of a lower overall accuracy. However, as opposed to the engage ACC task, we noticed that the gain from the demographic and type information was not very significant. In fact, upon inspection of the output trees, we noticed to our surprise that people's decision to disengage the ACC was more dependent on how quickly the ACC slowed the vehicle down, and not on the overall behavior of the driver. Thus, it should be noticed that simply adding attributes is not a panacea for higher accuracy– it only improves accuracy when relevant to the learning task at hand.

Overall, these results suggest that finding attributes beyond the observed data can be critical for accurately modeling a person's be-

ACC On	All Info	Without Type	Without Demo	Naive	All Info	Without Type	Without Demo	Naive
C4.5	92.67	92.32	91.22	91.27	0.35	0.32	0.07	0
MetaCost 0.5	92.42	91.97	90.97	91.27	0.40	0.36	0.13	0
MetaCost 0.7	91.93	91.38	90.37	91.27	0.45	0.42	0.18	0
MetCost 0.9	87.99	86.60	77.12	91.27	0.63	0.61	0.51	0
ACC Off	All Info	Without Type	Without Demo	Naive	All Info	Without Type	Without Demo	Naive
C4.5	88.71	88.64	88.42	86.37	0.37	0.37	0.35	0
MetaCost 0.5	88.59	88.55	88.14	86.37	0.43	0.42	0.41	0
MetaCost 0.7	87.68	87.49	87.31	86.37	0.49	0.49	0.49	0
MetCost 0.9	82.03	82.23	81.15	86.37	0.66	0.67	0.66	0

Table 2: Analyzing the tradeoff between overall model accuracy (left side of table) and recall of the minority cases (right side) in both the task of when people turn the ACC on (top) and off (bottom).

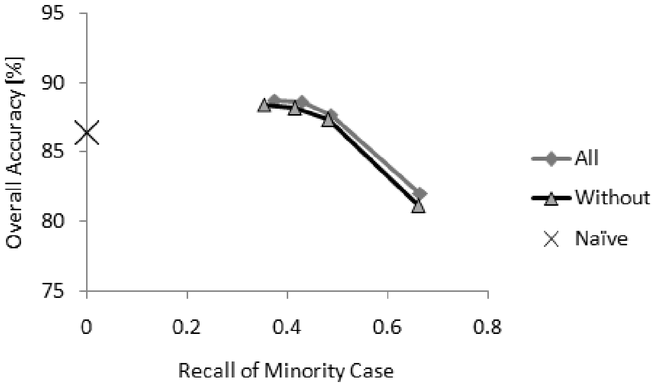

Figure 7: Comparing the overall model accuracy and recall for cases for disengaging the ACC

havior. Similar to previous studies that found that other behaviorial theories can better predict people's actions [14, 18], this work found that a driver's preferred gap setting could be better predicted by using a model of driving behavior [7]. Even if this measure was not readily available, an accurate estimate of this value could be learned based on a driver's demographic data.

Generally, one of the goals of this paper is to encourage people who build applications to consider incorporating data from external measures, such as psychological or behaviorial models. As was true in other domains as well [14, 18], exclusively using behavior models alone, such as the driver type possible in this domain [7], is not sufficient. By combining the driver type with other data, we achieved a prediction accuracy of nearly 70% within the discrete decision tree model (Figure 2) and a correlation of 0.78 within the regression model (Figure 4). However, when we used only the driver type information and removed the demographic information these models dropped to an accuracy of 46% and 0.55 respectively. This suggests that exclusively using behavior models is not as effective as the approach we present. Thus, we advocate for synthesizing data gleaned from behaviorial models in conjunction with observed domain data, something we believe can be effective in many other domains as well.

Practically, we are studying how either or both of these attributes can be used in the company's ACC. The advantage to using the demographic data alone is that ostensibly it can be provided before the driver begins using the car (e.g. in the showroom) and thus can be used to accurately model the driver from the onset. However, people may be reluctant to provide this information due to privacy concerns. Using driver profiling information is relatively difficult to calculate and is based on observed behavior over a period of time [7]. Thus, this value cannot be used to initially set values within the ACC. However, this data can be collected without privacy concerns and can be used to further improve the system's accuracy over time.

6. CONCLUSIONS

This paper presents an in-depth analysis into how learning approaches could be applied to create an intelligent Adaptive Cruise Control (ACC) agent that learns a driver's profile – both in terms of what gap setting she will chose, and when she is likely to engage or disengage their ACC. To create this agent we used real-world data from the past experience of many drivers from the ACAS field test data [5]. Specifically, we created driver models based on two learning approaches: regression and decision trees. Both were able to learn accurately the gap setting of an individual given his demographics characterization and driving type (hunter, glider or follower) with nearly 70% for the decision tree model and with a correlation of 0.78 for the regression model. These experiments emphasized the need for driver information including a behavior model about the driver's type [7] in addition to the information collected on the trips themselves and their demographic information. These results stress the fact that drivers may be very different from each other and previous approaches that set the gap setting similarly for all drivers [8, 9] are less effective. Therefore, driver characterization is essential for adapting automated systems in the vehicle. These differences among humans are made more salient when trying to learn when users engage or disengage from an automated system such as the ACC. Reactions could be very different teaching us also about the tendencies of users towards automation.

We present solutions for two practical challenges in applying learning algorithms to this challenging domain: preventing overfitted models, and creating effective models in cases where a strong majority category existed but the important events were in the minority category. We address the overfitting issue by created simplified decision trees, and we use the MetaCost algorithm [4] to learn from unbalanced data sets. We present extensive results details specifics of this application and how these algorithms were used within this challenging transportation domain.

More generally, adapting automated processes to better serve humans is a challenging task because humans are characterized by inconsistent behaviors, have difficulties in defining their own preferences, are affected by their emotions, and are affected by the complexity of the problems they face within the context of these problems. By understanding the current state of acceptance of automated systems and learning about differences among human users, we can improve the next generations of adaptive automated systems adjusted to their particular human users.

7. REFERENCES

[1] R Bishop. Intelligent vehicle applications worldwide. *IEEE Intelligent Systems*, 15(1):78–81, 2000.

[2] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 2002.

[3] Pierre-Andre Chiappori and Bernard Salanie. Testing for asymmetric information in insurance markets. *Journal of Political Economy*, 108(1):56–78, February 2000.

[4] Pedro Domingos. Metacost: a general method for making classifiers cost-sensitive. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, New York, NY, USA, 1999. ACM Press.

[5] R. Ervin, J. Sayer, D. LeBlanc, S. Bogard, M. Mefford, M. Hagan, Z. Bareket, and C. Winkler. Automotive collision avoidance system (acas) field operational test methodology and results, (us dot hs 809 901). washington, dc: Department of transportation, 2005.

[6] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:476–491, 1997.

[7] Paul Fancher and Zevi Bareket. A comparison of manual versus automatic control of headway as a function of driver characteristics. *3rd Annual World Congress on Intelligent Transport Systems*, 1996.

[8] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro. ACC+Stop&go maneuvers with throttle and brake fuzzy control. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):213–225, 2006.

[9] J. E. Naranjo, C. Gonzalez, J. Reviejo, R. Garcia, and T. de Pedro. Adaptive fuzzy control for inter-vehicle gap keeping. *Intelligent Transportation Systems, IEEE Transactions on*, 4(3):132–142, 2003.

[10] Georgios Paliouras, Vangelis Karkaletsis, Christos Papatheodorou, and Constantine D. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In *UM99 User Modeling: Proceedings of the Seventh International Conference*, pages 169–178, 1999.

[11] J. Ross Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, January 1993.

[12] Elaine Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.

[13] Avi Rosenfeld and Sarit Kraus. Modeling agents through bounded rationality theories. In *IJCAI-09*, pages 264–271, 2009.

[14] Avi Rosenfeld and Sarit Kraus. Modeling agents based on aspiration adaptation theory. *Autonomous Agents and Multi-Agent Systems*, 24(2):221–254, 2012.

[15] Bart van Arem, Cornelie J. G. van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 2006.

[16] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

[17] Zijian Zheng, Ron Kohavi, and Llew Mason. Real world performance of association rule algorithms. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 401–406, 2001.

[18] Inon Zuckerman, Sarit Kraus, and Jeffrey S. Rosenschein. Using focal point learning to improve human-machine tacit coordination. *Autonomous Agents and Multi-Agent Systems*, 22(2):289–316, 2011.