Toward Adapting Cars to Their Drivers

Avi Rosenfeld, Zevi Bareket, Claudia V. Goldman, Sarit Kraus, David J. LeBlanc, Omer Tsimhoni

Traditionally, vehicles have been considered as machines that are controlled by humans for the purpose of transportation. A more modern view is to envision drivers and passengers as actively interacting with a complex automated system. Such interactive activity leads us to consider intelligent and advanced ways of interaction leading to cars that can adapt to their drivers.

In this article, we focus on the adaptive cruise control (ACC) 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. Although individual drivers have different driving styles and preferences, current systems do not distinguish among 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 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 generic 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.

With the advance of automation and infotainment systems in vehicles, driving can be perceived as a comprehensive interactive activity occurring among human drivers and their vehicles. Classically, these users were considered the supervisors of the automated systems running in the cars. In the last decade, we have seen progress toward more autonomous driving automation, including features such as automatic lane centering, lane keeping, cruise control, and adaptive cruise control.

In this article, we concentrate on the adaptive cruise control function and look into the future by attempting to learn automatically how such a system can adapt its settings to its user and context. Cruise control is a known technology that aids drivers by reducing the burden of longitudinal control of 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 (van Arem, van Driel, and Visser 2006) and can be effective in reducing driver fatigue and fuel consumption (Bishop 2000). In this article, we focus on a second generation of cruise controls — adaptive cruise control (ACC) as it was used in the Automotive Collision Avoidance System Field Operational Test (ACAS FOT) (Ervin et al. 2005). ACC is designed as a comfort-enhancing system, which is an extension of conventional cruise control (CCC). The ACC system relieves the driver from some of the longitudi-

  ISSN 0738-4602

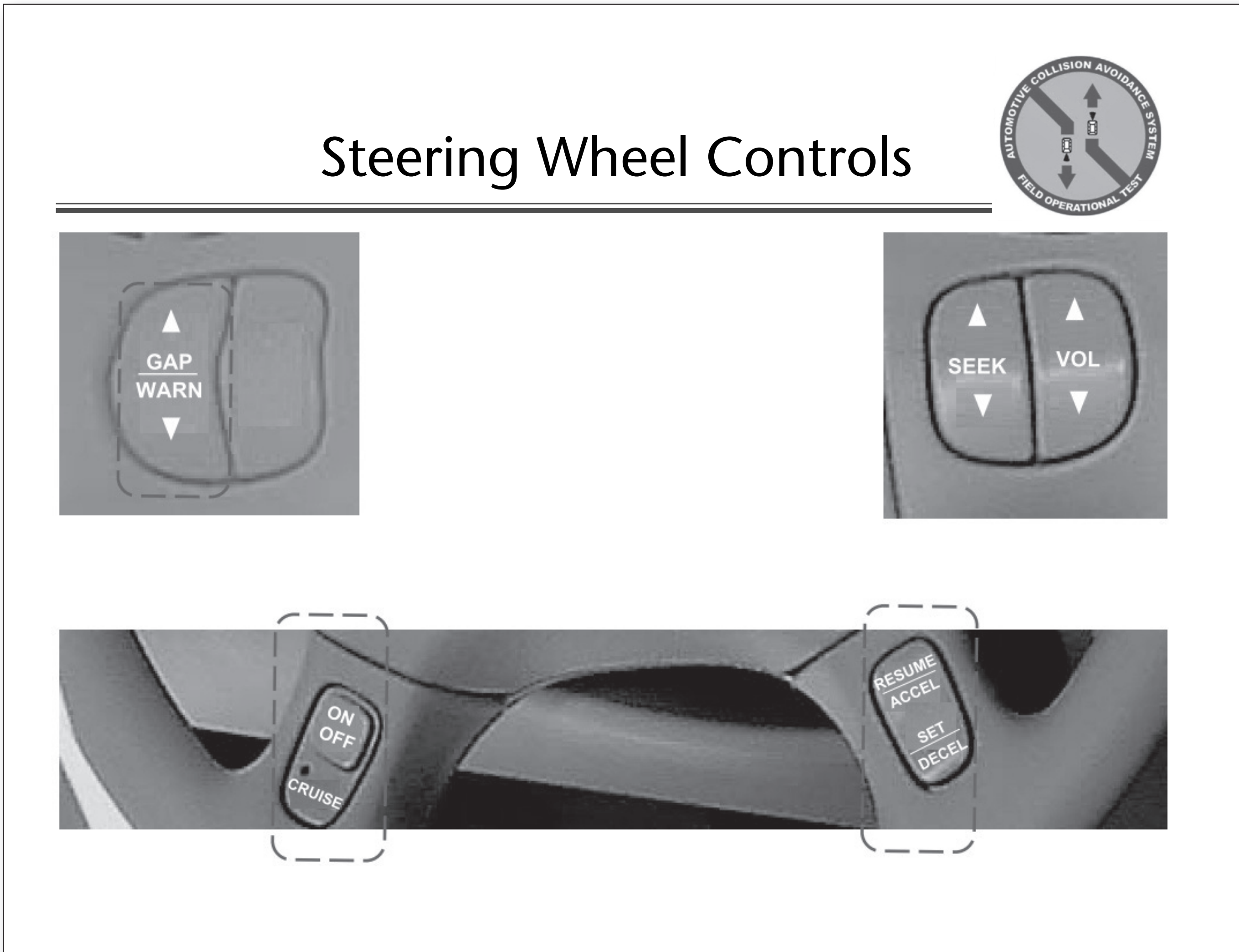

Figure 1. A Steering Wheel Fitted with ACC Technology.

nal-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 CCC 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.

Currently, commercial ACC systems preset the gap value to a default value, which can be adjusted by the driver manually based on his or her driving preferences. We envision that as the number of automated features grows, and as the number of user-preferred settings consequently grows, there will be an unavoidable need to add intelligence to set the default setting selection through context-dependent adaptation and user modeling to improve the driving experience.

In this article, we primarily focus on a method that learns how to quickly and accurately adjust the gap value 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 (Ervin et al. 2005). We used postprocessing of data from that study. Our general methodology is that once a new driver is identified we classify this driver as being similar to previously known drivers and set the initial gap value accordingly. In the next section we further detail the general methodology used within this work, including how this methodology was applied here.

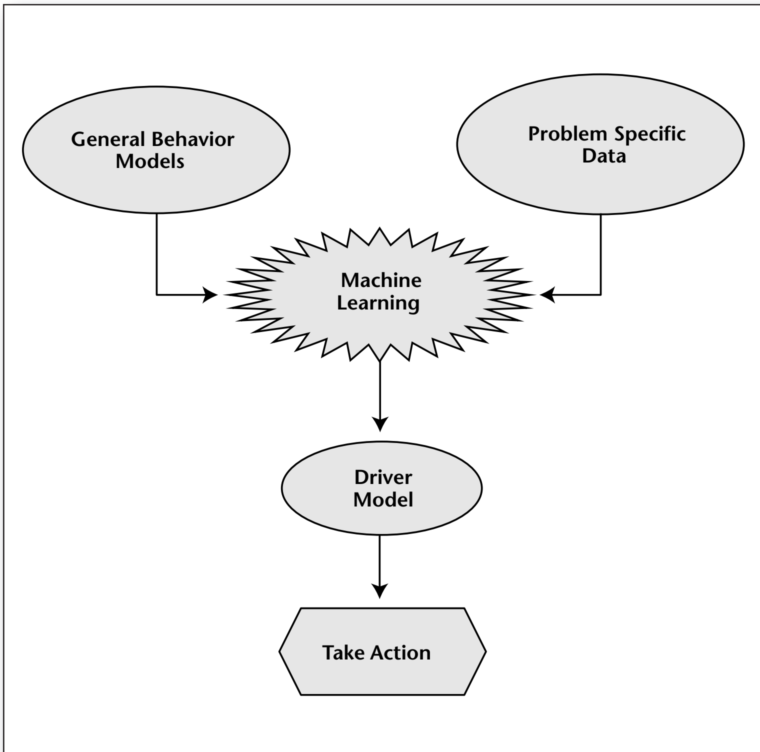

Figure 2. A High-Level Overview of the Methodology of Combining Cognitive Models with Machine Learning.

General Methodology and Related Work

The main challenge of this study was to process real-world data so as to obtain the most accurate and practical rules from the learning algorithms. 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 (Rich 1979). Even more similar to our work, Paliouras et. al (1999) 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.

Previous works within the last decade did study how to assist drivers in the task of longitudinal control (Naranjo et al. 2003, 2006). Within these approaches, rules were learned manually after interviews with human drivers. Based on these rules the gap value was adjusted automatically to the conditions of the drive without considering the particular driver in the vehicle. However, we found that individual drivers differ in their driving styles and preferences. Therefore, 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 value per each driver.

To accomplish this goal, we present a methodology in figure 2. We have previously explored an approach where machine learning is used in conjunction with generalized cognitive and behaviorial models from other disciplines, including experimental economics and psychology. In general, we previously found that merging this synthesis yields one of two general results. In some cases, there is ample data about people's interaction within the system allowing us to form an accurate model of people's behavior using machine learning alone. While in these types of cases machine learning does not provide a more accurate model than the best psychological model, machine learning can be used to confirm the effectiveness of a given model in predicting people's behavior. This can be particularly useful if multiple cognitive models were possible, allowing us to judge which model is best without human bias. Additionally, knowing which cognitive model is applicable to a given problem allows us to quickly form an accurate model of people's behavior, even with limited or noisy data (Rosenfeld and Kraus 2009, Rosenfeld and Kraus 2012). In contrast, in more complex environments a lack of data makes it unfeasible to elicit an accurate model with machine learning alone. In these types of problems we found that using attributes from cognitive models allowed for significantly more accurate models than models created from machine learning or the cognitive models alone (Rosenfeld and Kraus 2009; Rosenfeld and Kraus 2012; Zuckerman, Kraus, and Rosenschein 2011).

In this work, we considered how machine learning could be used in conjunction with a behavior driving model previously developed by Fancher and Bareket (1996). Their work analyzed a group of 36 drivers and their acceptance of adaptive cruise control. They found that while all drivers enjoyed and accepted ACC, their behavior could be divided into three types. Each group demonstrated specific driving tendencies that affected their headway and closing speeds relative to vehicles ahead during manual driving (that is, without cruise control engaged). 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

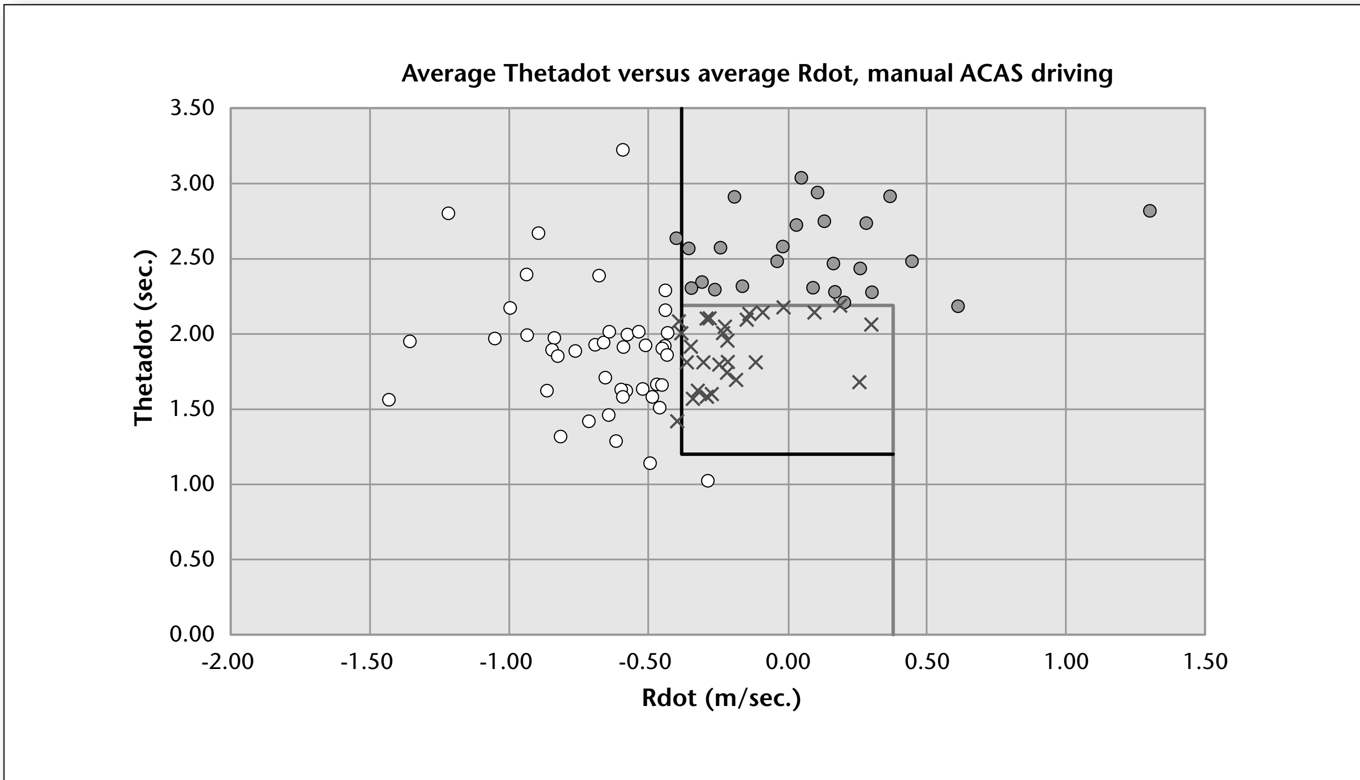

Figure 3. Classification of Drivers Based on Their Average Headway Times Versus Their Average Speeds.

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).

Figure 3 depicts the averages Rdot and ThetaDot of all the drivers on a scatter plot. Rdot is a measure of change within the range rate (the mathematical derivative), such that if a driver tends to travel faster than the neighboring and preceding vehicles then Rdot will be less than 0, and conversely if that driver drivers slower than the neighboring vehicles then Rdot will be greater than 0. ThetaDot is the change (derivative) of the gap divided by speed and provides a measure of gap in units of time.

The dotted and solid lines in figure 3 are the boundary lines that divide the ACAS drivers into three classes: (1) To the left of the black line are the hunters, depicted by circles; (2) to the right and top of the red line are the gliders, depicted by dots; and (3) those within the rectangle bounded by the red and black lines are the followers, depicted by 'x.'

In numerical terms, hunters have headway times less than about 1.2 seconds and they tend to travel by at least 1 mph faster than neighboring vehicles in the traffic stream. Gliders have headway times greater than 2.2 seconds and they tend to travel by at least 1 mph slower than neighboring vehicles in the traffic stream. The followers lie within these bounds between hunters and gliders.

In this article, we use these driver classes as attribute types (hunter, glider, or follower), and they were used in addition to other demographic information to attempt to build an application that predicts how the ACC should set its gap given this information and road situation. As we now detail, we found that Fancher and Bareket's behavior model was crucial for accurately predicting a driver's ACC preferences. However, in many cases sufficient data existed from a driver's demographic background to predict his or her driving type. This can be significant as the above calculations required to find a driver's type require an extensive learning period.

Learning Method

Current ACC systems allow the user to set the gap value between a finite number of possible values (1 to 6 in our case). These values control the gap the ACC autonomously maintains from the vehicle in front of it. Currently, one value is set as the default (in our case this value was 6) and the user may change it during driving as he or she wishes.

In order to study the problem of predicting what gap value a person would select, we constructed two different types of models. The first type of model was a regression model. Regression models operate by statistically predicting the value of a continuous dependant variable from a number of independent variables. In this problem, the goal was to predict the gap value a given driver would select based on the independent variables of the current driving conditions. The second type of model was a decision tree model (C4.5). Decision trees predict the value of a discrete dependant variable from a number of independent variables. Specifically, here we learn which of the discrete gap values a driver will likely choose given all possible values given current driving conditions. Note that while discrete regression functions and continuous decision algorithms also exist, we focused on these two types of models to differentiate between these categories of models.

Our goal was to use the output of either model to automatically set the gap value. Toward 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 (for example, 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 (for example, 3.5 being close to 3 and 4) is not as mathematically significant as mistakes between two extreme values (for example, 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, namely, learning (1) when the ACC is first engaged and (2) 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 data-set imbalance problem (Chawla et al. 2002). 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, more than 90 percent 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. MetaCost is a general algorithm for making any type of classification learning algorithm cost sensitive, allowing us to stress certain categories more than others. MetaCost has the advantage of working well with any classification algorithm, as it operates by wrapping a cost-minimization procedure around any classifier (Domingos 1999). We opted to use this algorithm because of its flexibility and the ease within this algorithm of controlling the bias size given to the minority case. Empirical results for learning the gap value and classifying engagement and disengagements of and from the ACC are explained in the next section.

Experimental Setup

Data for our analysis were taken from the automotive collision avoidance system field operational test (ACAS FOT) (Ervin et al. 2005) (see figure 4). In that study, to understand how different drivers use an ACC, each of 96 drivers was presented with an ACC-fitted vehicle. which they used for a period of 4 weeks. For the first week, the ACC system was not available, allowing drivers to acclimate to their vehicles. If the driver engaged the cruise control during this period, it simply maintained speed just like the conventional system (CCC). For the next three weeks, if the driver chose to engage the cruise control, it functioned as ACC. In general, three different data sets were considered. The first, and most basic, data set consisted of objective characteristics that can be studied based on the location of the vehicle itself, for example, gap to the lead vehicle, vehicle speed, longitudinal acceleration, road type (country, city, or highway), weather (if it was raining and if it was dark outside) and road density (is there traffic). A second data set 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 consists of a good mixture of these demographics with a 51 percent male to 49 percent female split, 31 percent young (aged 20–30), 31 percent middle aged (aged 40–50), and 38 percent older drivers (aged 60–70), and people from a variety of education and socioeconomic levels. The last data set also used the drivers' observed behavior from the first week to label drivers as hunters, gliders, or followers as per Fancher and Bareket's previous work (1996).

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 (that is, ACAS-disabled and ACAS-enabled).

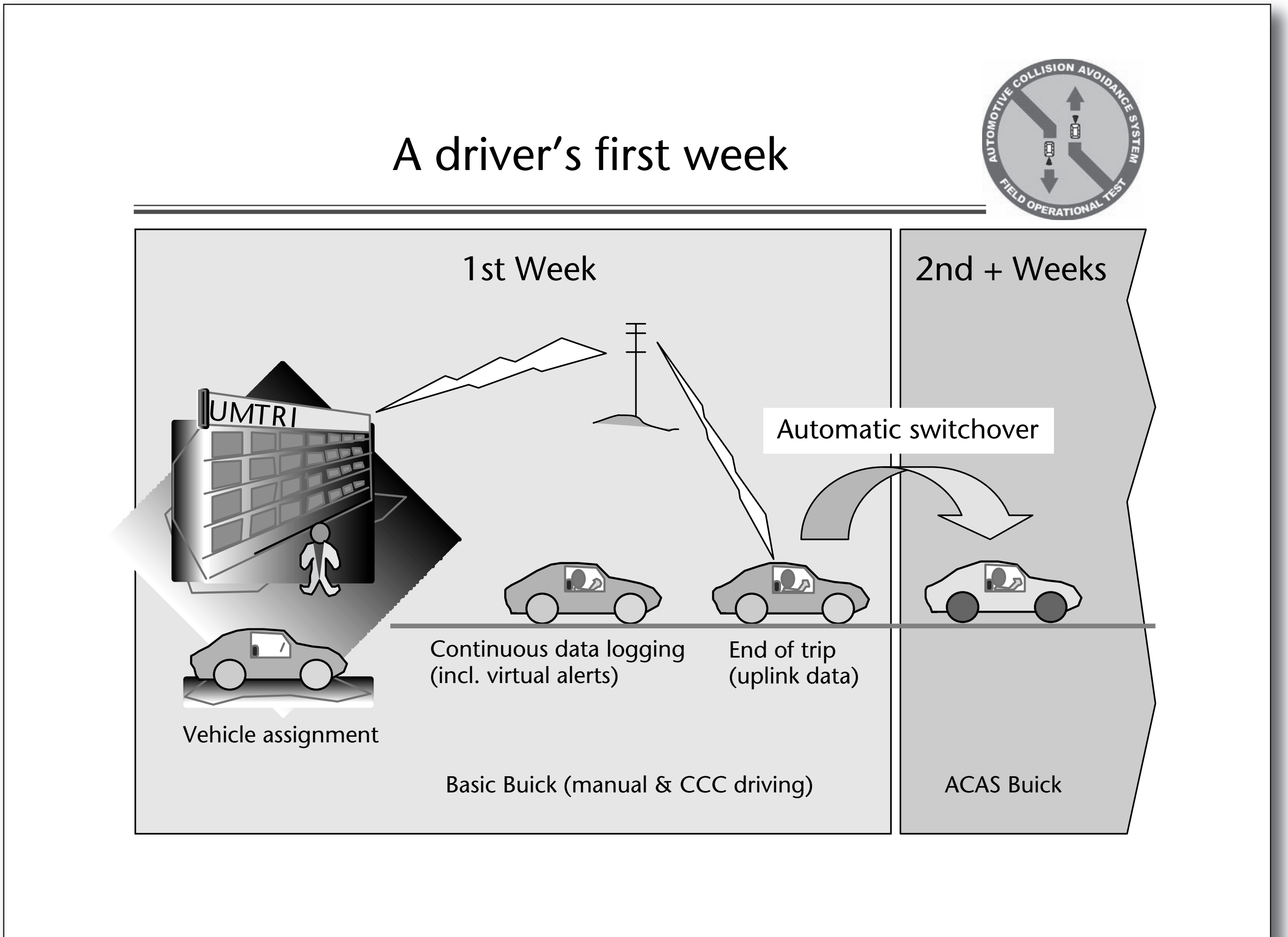

Figure 4. An Overview of the Data Collection Process Within ACAS FOT Study.

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.

Results

In this section we present results for the three previously defined problems: predicting a driver's gap value 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 prominent in this application, how we avoided overfitting, and how we addressed the data-set imbalance problem within this application.

Setting the ACC's Gap Value

Figure 5 presents the accuracy of the decision tree model to learn a driver's preferred gap value in the discrete model. Clearly, adding the demographic data here is crucial, as the model's accuracy drops from over 66 percent accuracy with this data to less than 37 percent accuracy without this. As a baseline, we also include the naive classifier, which is based on the most common gap value — here the value of 6, which is also the system's default. Note that the naive model had an accuracy of nearly 27

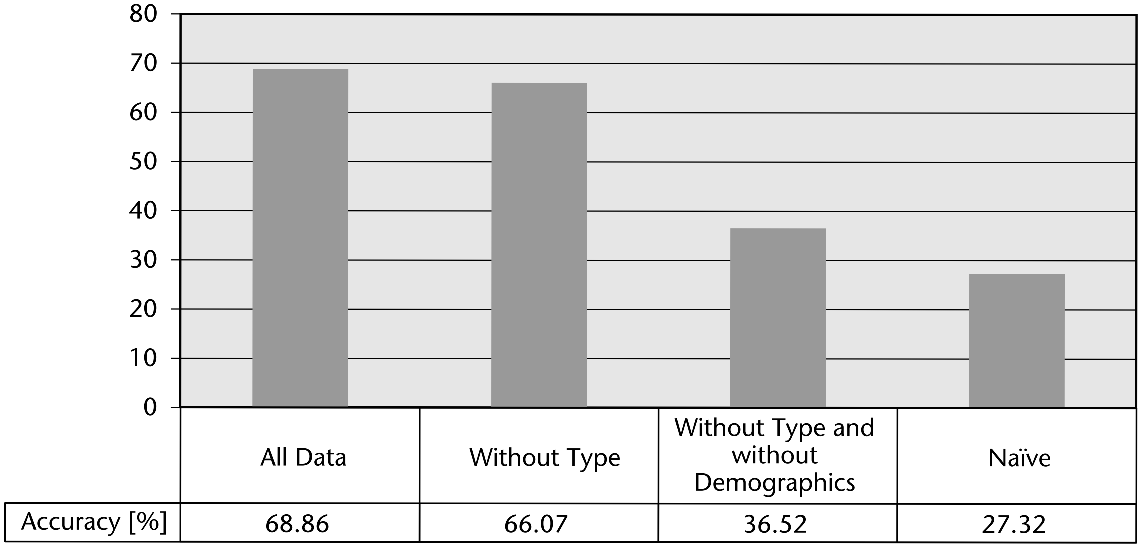

Figure 5. The Importance of Driver Type and Demographics in Predicting the Gap.

percent, far less than other models. The user's type did improve accuracy, as adding this information to the type increased accuracy to near 70 percent. In line with our previous work (Rosenfeld and Kraus 2009), we hypothesized that adding this behavior model yields smaller 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.

Value Within the ACC for a Discrete Decision Tree Model.

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 percent accuracy (95.22 percent) 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 6. From an application perspective, we were not surprised to find that a driver's age factored heavy in his or her driving behavior. This characteristic is factored in actuarial insurance tables, and is a known factor in car insurance premiums (Chiappori and Salanie 2000). Note that this characteristic was the first level below the root of the tree, demonstrating this quality. However, possibly equally interesting 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 often not considered by insurance companies (Chiappori and Salanie 2000) due to several concerns, including privacy concerns; however, this issue may be worth revisiting. 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 factor. Overall, we found that young men or women with only a high school degree tended to be 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 but 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 is only approximately 75 percent accurate. Nonetheless a general direction is evident from this tree, and was one that the content experts felt was not overfitted.

Similarly, it was important to find a decision tree that models drivers' gap value that is not overfitted. Note from table 1 that the model accuracy given all data is nearly 70 percent. However, while this value is based on the mathematically sound C4.5 algorithm (Quinlan 1993), the content experts again felt this decision model was overfitted. We then proceeded to reduce the size of the tree to generalize the model, thus preventing this

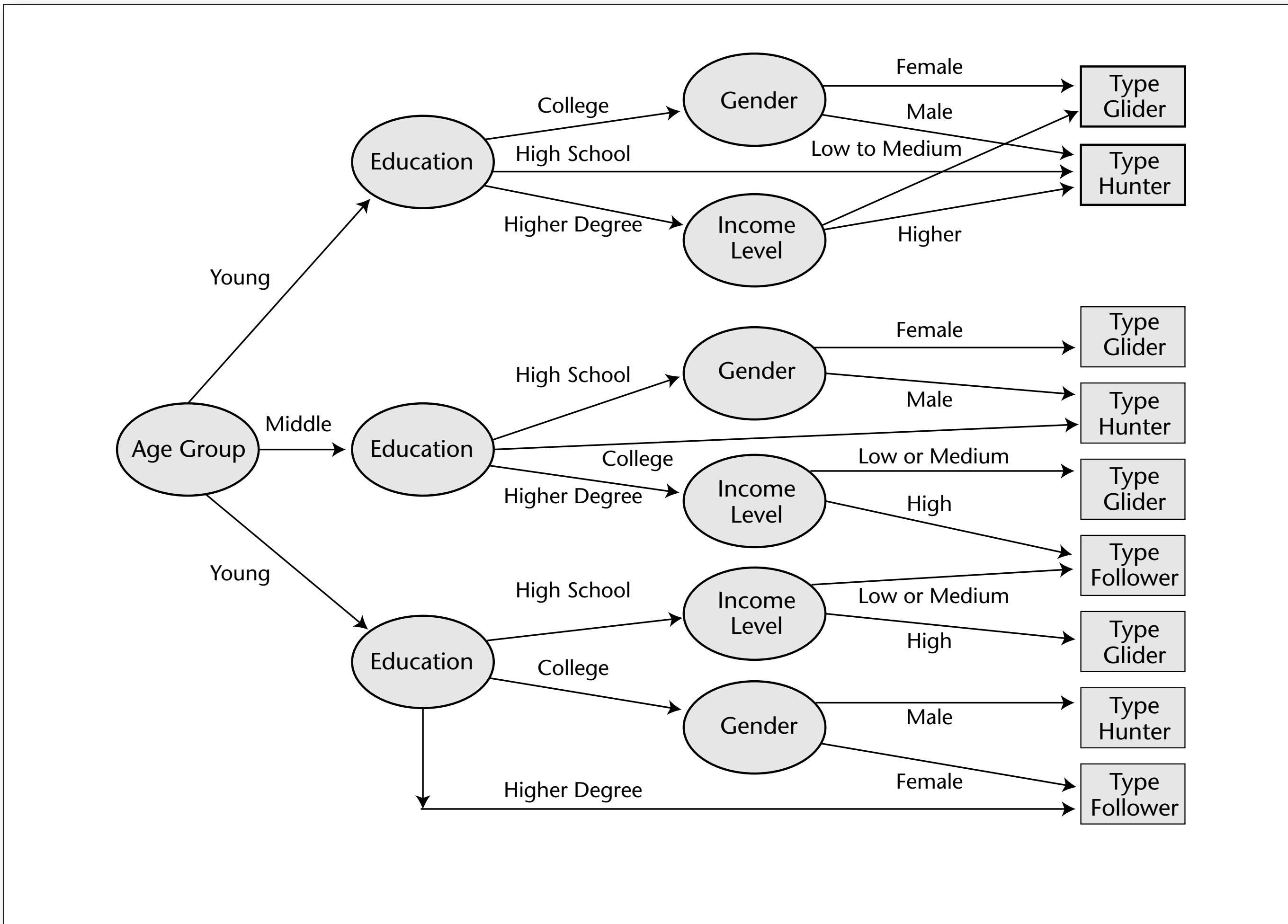

Figure 6. The Decision Tree for Learning a Driver's Type.

phenomenon. However, as table 1 demonstrates, reducing the tree size does not improve the model's accuracy, as previous theoretical works suggest (Esposito, Malerba, and Semeraro 1997) but did produce trees that were acceptable to the content experts. Note from table 1 that raising T_{Depth} yields only marginal increases in the model's accuracy after $T_{Depth} = 4$. 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 7 the resultant tree of $T_{Depth} = 4$, which is only 6 percent less accurate than the full tree in table 1. 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 has a total size of only 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.

Similarly, we were able to create an accurate regression model, the results of which are found in figure 8. Within these models, correlation values can range from 1.0 (fully positive correlated) to –1.0 (fully negatively correlated), with 0 meaning 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. The experts again opted for a reduced model, despite the sacrifice of slightly less accuracy.

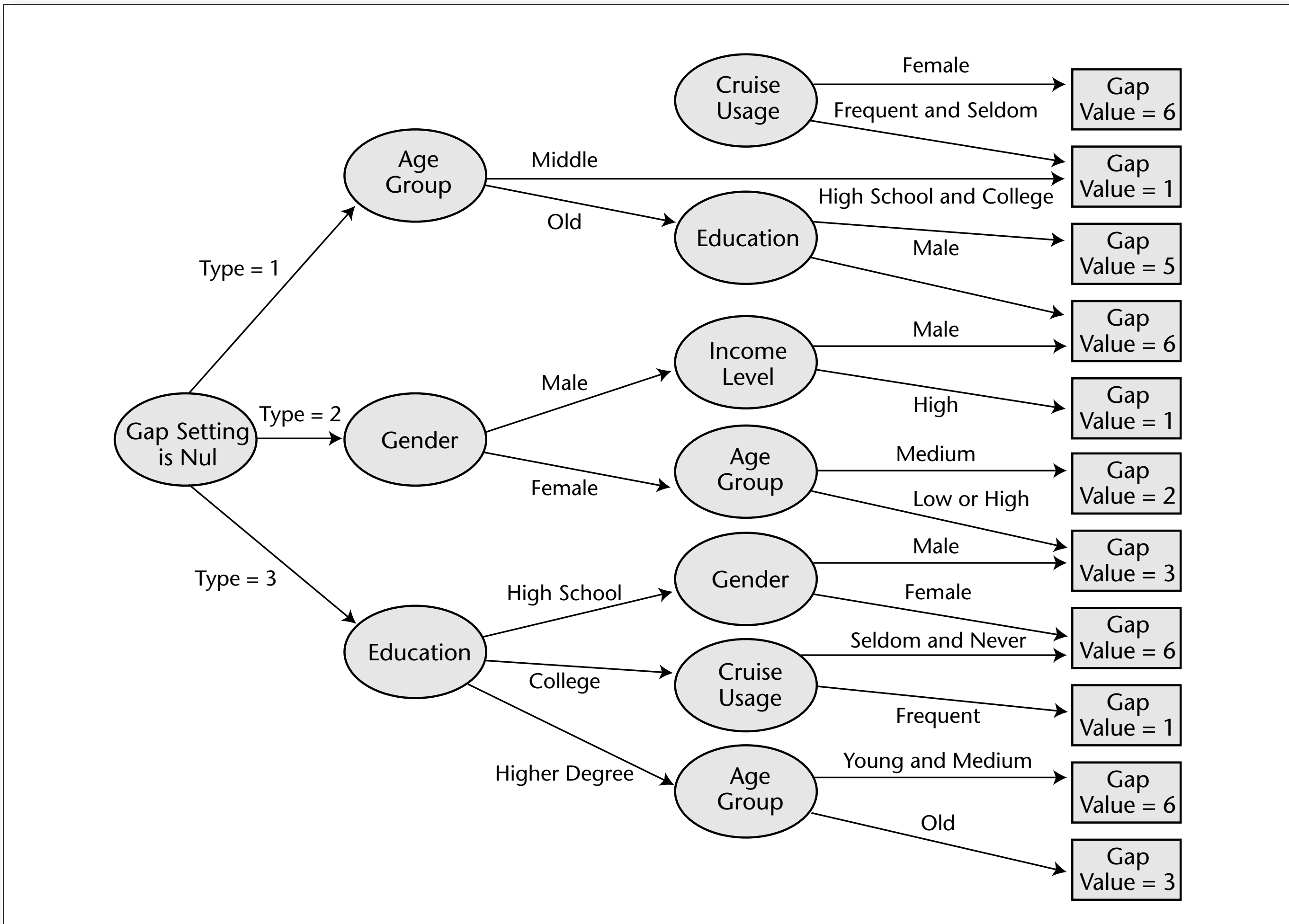

Figure 7. The Decision Tree for Learning the ACC's Gap Value for $T_{Depth} = 4$.

Predicting When the Driver Will Engage and Disengage the ACC

While the focus of our work was on the gap value 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 hope we can increase its acceptability and use. Similarly, by understanding when people disengage the ACC we hope to 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 data-set imbalance problem (Chawla et al. 2002). In this article, we constructed two models for these two problems based on the same three types of data sets. The first model is a basic C4.5 without any modification. As was the case in the task of setting the gap value, we considered attributes based on the behavior type model, driver demographics (for example, sex, age, and income level), and the vehicle's characteristics (for example, gap to the lead vehicle, vehicle speed, and road type). In the second model, we again used the same three data sets, but created a learning bias to find the important

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 Trade-off Between the Model's Accuracy and the Height of the Tree T_{Depth}.

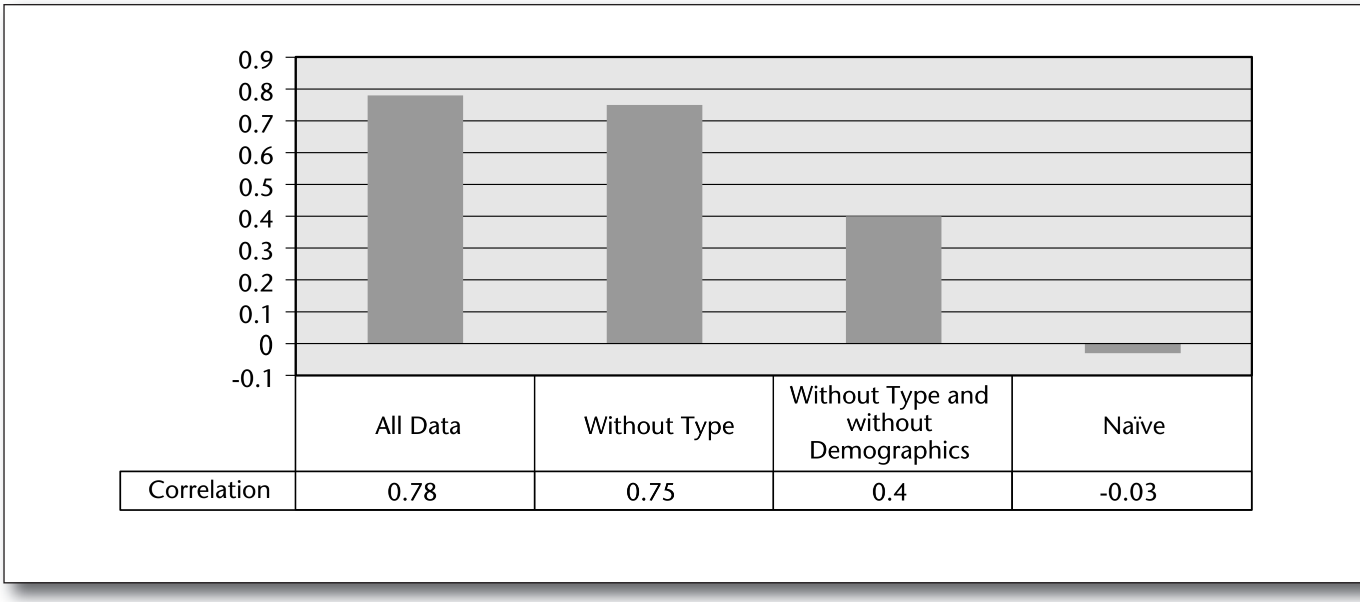

Figure 8. The Importance of Driver Type and Demographics in Predicting the Gap Value Within the ACC for a Regression Model.

minority cases. We specifically focused on the MetaCost algorithm (Domingos 1999).

Table 2 displays the complete results demonstrating the trade-off 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 is found in the first four columns in table 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 data sets. 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: for example, one with 100 percent accuracy and recall of all cases. Unfortunately, this is unrealistic, especially in tasks that are prone to variations due to noise. In this domain, the noise comes from two factors: Noise from people's lack of consistency, and noise from unmeasured factors from within the environment. For example, this study did not consider traffic density in lanes to the right or left of the driver — something that clearly may affect a driver's decision. 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 a recommendation as to when to engage the ACC. Toward this goal, 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 9 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 percent of the cases, but this 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 better results than one without the type and demographic information.

Similarly, figure 10 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

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
MetaCost 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
MetaCost 0.9	82.03	82.23	81.15	86.37	0.66	0.67	0.66	0

Table 2. Analyzing the Trade-off Between Overall Model Accuracy (Left) and Recall of the Minority Cases (Right) in Both the Task of When People Turn the ACC on (Top) and Off (Bottom).

task, the naive case assumes that the driver will continue with the ACC constitutes over 86 percent of the cases, but this model will have recall of 0 for the minority case (see the left side of figure 10). 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 large. 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 behavior. Similar to previous studies that found that other behaviorial theories can better predict people's actions (Rosenfeld and Kraus 2012; Zuckerman, Kraus, and Rosenschein 2011), this work found that a driver's preferred gap value could be better predicted by using a model of driving behavior (Fancher and Bareket 1996). Even if this measure was not readily available, an accurate estimate of this value could be learned based on a driver's demographic data.

Practically, we are studying how either or both of these attributes can be used. The advantage to using the demographic data alone is that ostensibly it can be provided before the driver begins using the car (for example, 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 (Fancher and Bareket 1996). 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.

Conclusions

Adapting automated processes to better serve humans is a challenging task because it is difficult to predict setting values. Humans are characterized by inconsistent behaviors (or at least seemingly inconsistent), may have difficulties in defining their own preferences, are affected by their emotions, and are affected by the complexity of the problems they face together with the context of these problems. In particular, human drivers also need to react fast enough to road conditions and changes in traffic. This task was particularly challenging as incomplete information was not only inherent about drivers' preferences, but also from the domain itself. For example, this study did not have complete knowledge about traffic in the area of the driver, and even issues such as traffic patterns or weaving behavior from surrounding drivers were not explicitly measured. Nonetheless, we report on the success of how we quickly and accurately learned the ACC gap value given historical data of the many drivers from the ACAS field test data (Ervin et al. 2005). Successful learning of the gap value in this task should serve as a necessary but not sufficient step toward prediction of settings in driving assistance systems and in other systems and situations. As with other user adaptation systems, the trade-off between the added value and potential inaccuracy of the prediction system will determine its usefulness and acceptance.

We empirically studied two learning approaches: regression and decision trees. By combining the driver type with other data, we achieved a prediction accuracy of nearly 70 percent within the decision tree model (figure 5) and a correlation of 0.78 within the regression model (figure 8). However, when we used only the driver type information and removed the demographic information these models dropped to an accuracy of 46 percent and 0.55 respectively. These experiments emphasize the need to construct models that not only consider driving data such as the car's speed, road condition, and weather, but also include driver demographic information and a behavior model about the driver's type (Fancher and Bareket 1996). These results stress the fact that drivers may be very different from each other and previous approaches that set the gap value similarly for all drivers (Naranjo et al. 2003, 2006) 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 ACC. Reactions could be very different, teaching us also about the tendencies of users toward 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 creating simplified decision trees, and we use the MetaCost algorithm (Domingos 1999) to learn from unbalanced data sets. We present extensive results details of this application and how these algorithms were used within this challenging transportation domain.

One way of building machines that could interact successfully with humans is by adapting their automated processes to their users. In order to do so, these machines need to consider characteristics of human behaviors, including, for example: inconsistent behaviors, having difficulties in defining the user's own preferences, emotional influences, and problem complexity and context effects. Therefore, adapting automated processes is a challenging task. 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.

One of the larger goals of this article 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 we studied (Rosenfeld and Kraus 2012; Zuckerman, Kraus, and Rosen-

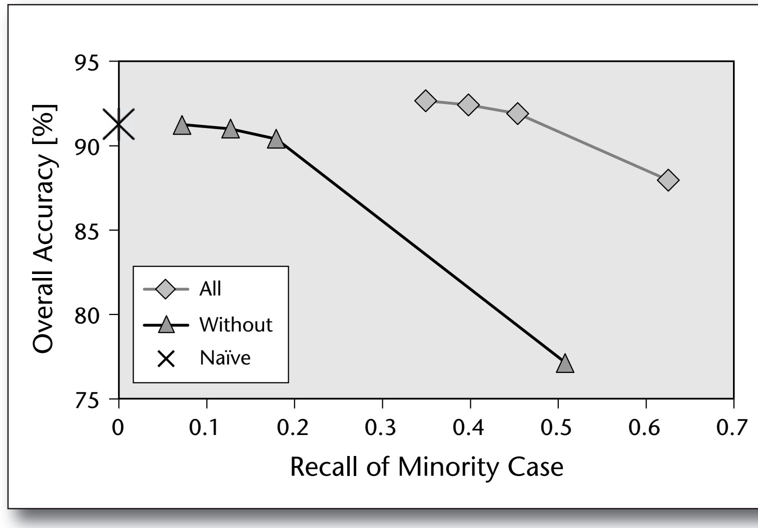

Figure 9. Comparing the Overall Model Accuracy and Recall for Cases for Engaging the ACC.

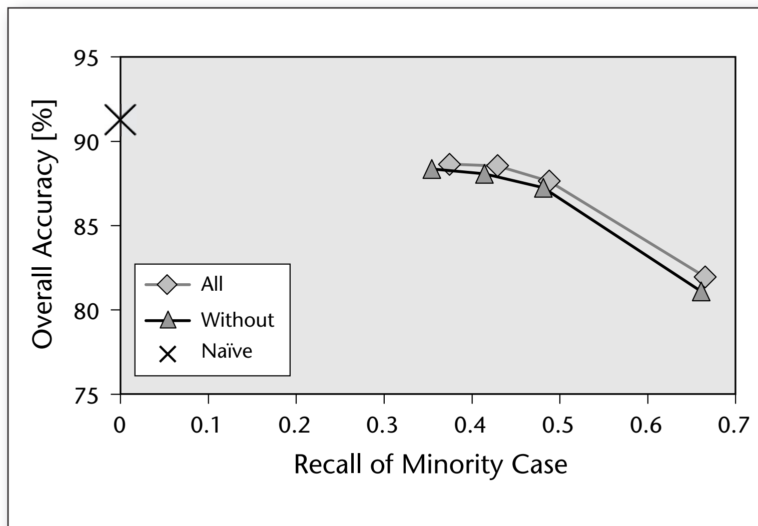

Figure 10. Comparing the Overall Model Accuracy and Recall for Cases for Disengaging the ACC.

schein 2011), modeling user behavior based on cognitive models alone, such as the driver type possible in this domain (Fancher and Bareket 1996), is not sufficient. Instead, we advocate for synthesizing data gleaned from behaviorial models in conjunction with observed domain data, something that we believe can be effective in many other domains as well.

Acknowledgements

This work was supported in part by ERC grant #267523, MURI grant number W911NF-08-1-0144, ARO grants W911NF0910206 and W911NF11103 44, and JCT grant 963.

References

Bishop, R. 2000. Intelligent Vehicle Applications Worldwide. *IEEE Intelligent Systems* 15(1): 78–81.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. Smote: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* (JAIR) 16: 321–357.

Chiappori, P.-A., and Salanie, B. 2000. Testing for Asymmetric Information in Insurance Markets. *Journal of Political Economy* 108(1): 56–78.

Domingos, P. 1999. 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*, 155–164. New York: Association for Computing Machinery.

Ervin, R.; Sayer, J.; LeBlanc, D.; Bogard, S.; Mefford, M.; Hagan, M.; Bareket, Z.; and Winkler, C. 2005. *Automotive Collision Avoidance System (ACAS) Field Operational Test Methodology and Results* (US DOT HS 809 901). Washington, DC: Department of Transportation.

Esposito, F.; Malerba, D.; and Semeraro, G. 1997. A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19:476–491.

Fancher, P., and Bareket, Z. 1996. A Comparison of Manual Versus Automatic Control of Headway as a Function of Driver Characteristics. Paper presented at the 3rd Annual World Congress on Intelligent Transport Systems. Orlando, FL, 14–18 October.

Naranjo, J. E.; Gonzalez, C.; Garcia, R.; and de Pedro, T. 2006. ACC+Stop&Go Maneuvers with Throttle and Brake Fuzzy Control. *IEEE Transactions on Intelligent Transportation Systems* 7(2): 213–225.

Naranjo, J. E.; Gonzalez, C.; Reviejo, J.; Garcia, R.; and de Pedro, T. 2003. Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping. *IEEE Transactions on Intelligent Transportation Systems* 4(3): 132–142.

Paliouras, G.; Karkaletsis, V.; Papatheodorou, C.; and Spyropoulos, C. D. 1999. Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities. In *UM99 User Modeling: Proceedings of the Seventh International Conference,* 169–178. Berlin: Springer.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning* (Morgan Kaufmann Series in Machine Learning). San Francisco: Morgan Kaufmann.

Rich, E. 1979. User modeling Via Stereotypes. *Cognitive Science* 3(4): 329–354.

Rosenfeld, A., and Kraus, S. 2009. Modeling Agents Through Bounded Rationality Theories. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence,* 264–271. Menlo Park, CA: AAAI Press.

Rosenfeld, A., and Kraus, S. 2012. Modeling Agents Based on Aspiration Adaptation Theory. *Autonomous Agents and Multi-Agent Systems* 24(2): 221–254.

van Arem, B.; van Driel, C. J. G.; and Visser, R. 2006. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. *IEEE Transactions on Intelligent Transportation Systems* 7(4): 429–436.

Zuckerman, I.; Kraus, S.; and Rosenschein, J. S. 2011. Using Focal Point Learning to Improve Human-Machine Tacit Coordination. *Autonomous Agents and Multi-Agent Systems* 22(2): 289–316.

Avi Rosenfeld earned in Ph.D. in applied AI from Bar Ilan University. He works at the Jerusalem College of Technology.

Zevi Bareket is a senior research associate at the University of Michigan Transportation Research Institute. He has an MS in mechanical engineering.

Claudia V. Goldman is a senior researcher at General Motors Advanced Technical Center in Israel. She earned her Ph.D. in computer science at the Hebrew University.

Sarit Kraus is a professor of computer sciencee at Bar-Ilan University and an adjunct professor at the University of Maryland. She earned her Ph.D. in computer science from Hebrew University.

David J. LeBlanc is an assistant research scientist at the University of Michigan Transportation Research Institute. He earned a Ph.D. in aerospace engineering from the University of Michigan.

Omer Tsimhoni is a group manager in human machine interface research at GM Advanced Technical Center, Israel. He earned a Ph.D. in industrial and operations engineering, human factors, from the University of Michigan.