The AAAI 2017 Spring Symposium on
Learning from Observation of Humans
Technical Report SS-17-06

Learning to Predict Driver Behavior from Observation

Santiago Ontañón
Drexel University
Philadelphia, PA, USA
so367@drexel.edu

Yi-Ching Lee
George Mason University
Fairfax, VA, USA
ylee65@gmu.edu

Sam Snodgrass
Drexel University
Philadelphia, PA, USA
sps74@drexel.edu

Flaura K. Winston
Children's Hospital of Philadelphia (CHOP)
Philadelphia, PA, USA
winston@email.chop.edu

Avelino J. Gonzalez
University of Central Florida
Orlando, FL, USA
gonzalez@ucf.edu

Abstract

This paper focuses on modeling and predicting human driving behavior, with the long term goal of anticipating the behavior of the driver before dangerous situations occur. We formulate this problem as a *Learning from Demonstration* problem, and show how standard supervised learning methods do not perform well in this task. The main contribution of this paper is a new approach we call *indirect prediction*. The key idea of *indirect prediction* is not to predict the behavior directly, but rather to build a model that predicts how certain features of the world state will change over time, and then use those to predict the necessary behavior in order to achieve those changes. We show how this apparently counterintuitive idea directly addresses one of the key reasons for which supervised learning does not perform well for LfD. In addition, we show how using ideas from context-based reasoning can also improve the accuracy of behavior modeling.

Introduction

Motor vehicle crashes are the leading cause of death for U.S. teens, accounting for more than one in three deaths in this age group and claiming the lives of approximately eight teenagers per day, according to the 2010 report by the *Center for Disease Control and Prevention*. In addition, teen drivers have four-times the fatal crash risk of adults (Curry et al. 2011). Previous research has identified speed management as a major contributing factor for crashes involving newly licensed teen drivers (McKnight and McKnight 2003; Curry et al. 2011). By comparison, experienced adult drivers are known to be better at adjusting driving behaviors and speed control when traffic situations change, suggesting that learning and accumulation of driving experience contribute to declining crash risk with age and experience (Foss et al. 2011; McKnight and McKnight 2003). In order to develop and test new training methods and new technology to reduce teen crash risk, techniques are needed to better understand the complex behaviors associated with this increased risk.

This paper focuses on the problem of modeling and predicting human driving behavior, with the long term goal of anticipating the behavior of the driver before dangerous situations occur. We formulate this problem as a *Learning from Demonstration* problem, and show how standard supervised learning methods do not perform well in this task. The main contribution of this paper is a new approach we call *indirect prediction*. The key idea of *indirect prediction* is not to predict the behavior directly, but rather to build a model that predicts how certain features of the world state will change over time, and then use those to predict the necessary behavior in order to achieve those changes. This idea of indirect prediction is used to circumvent the fact that supervised learning algorithms are known to underperform in LfD tasks since LFD violates the *i.i.d.* assumption (training and test data independently and identically distributed). In addition, we also show that using *context-based reasoning* (CxBR) can also significantly improve the performance of the learning algorithms. The key idea behind CxBR is to divide the learning task into a collection of different contexts (e.g., making left turns, overtaking, etc.), and learning a model for each context, thus simplifying the learning task.

The remainder of this paper is organized as follows. We first provide some background of driving behavior modeling. After that, we describe the dataset used in our empirical evaluation. We follow with a description of why supervised learning approaches underperform in LfD, and then present our two improvements (CxBR and indirect prediction). The paper closes with an empirical evaluation, conclusions and directions of future work.

Modeling Human Driving Behavior

Due to its many practical applications and its importance to road safety, driving behavior modeling has been approached from many perspectives. A significant body of work, illustrated by the work of Macadam (Macadam 2003), exists on the manual creation of control models that exhibit specific aspects of human driving behavior (see Markkula et al. (Markkula et al. 2012) for an in-depth review).

A number of approaches employ machine learning methods to automatically acquire models of human driving behavior (Fernlund et al. 2006; Pomerleau 1989). For example, case-based and instance-based learning have been deployed for driving tasks (Ontañón et al. 2014) as well as other LfD tasks (Floyd, Esfandiari, and Lam 2008; Ontañón et al. 2010; Rubin and Watson 2011; Lamontagne, Rugamba, and Mineau 2012). These approaches learn behavior by observing an expert's performance, but except for the work of Rubin and Watson, focus on small-scale datasets, much smaller than the datasets required to analyze driving behavior.

Specific similarity measures have been proposed for the domain of driving behavior. For example, Nechyba and Xu (Nechyba and Xu 1998) developed a similarity measure based on Hidden Markov Models (HMM) (Rabiner and Juang 1986). The results demonstrated the ability of the models to accurately differentiate driving traces generated from one driver from those generated from other drivers however, the measures were based on relatively simple driving data (without other traffic). As elaborated below, our driving data are significantly more complex than that used by Nechyba and Xu (our dataset included other traffic), pointing to the need for new methods.

Finally, previous work has utilized models to automatically predict driver states from driving data. Das et al. (Das, Zhou, and Lee 2012) showed that statistical measures over steering wheel data such as entropy and the Lyapunov exponent can determine whether a given driver is under the effect of alcohol or not. No previous work has attempted to automatically predict degree of driver skill.

Problem Statement

The problem we address in this paper is the following: is it possible to learn a model of a human driver so that we can predict what the driver is going to do in the next few seconds? Solving this problem would have large implications for automotive safety. We envision that real-time or close to real-time feedback via machine learning techniques have the potential to be part of in-vehicle monitoring systems that can track drivers' performance and provide meaningful, adaptive guidance to drivers.

We formulate such a problem as a learning from demonstration (LfD) problem where the training data consists of one or more *demonstrations*, where each demonstration $T = \{(x_0, y_0), ..., (x_m, y_m)\}$ is a sequential record of the behavior of a given driver over time in a given scenario. Demonstrations thus consist of sequences of state-action pairs: (x_t, y_t), where x_t describes the world state at time t, and y_t represents the actions that the driver took at time t. The assumption here is that the world state is represented by a set of variables X and the actions by a set of variables Y. So, x_t and y_t correspond to the multidimensional values that X and Y take at time t. Specifically, in this paper, the world state X is represented by 34 variables (described below), and the action Y contains three variables: *steer*, *throttle*, and *brake*, representing the position of the steering wheel, and the throttle and brake pedals respectively at time t.

Given one or more demonstrations, the task is to predict the behavior of the driver in a different, not previously seen

Figure 1: The driving simulator at the Children's Hospital of Philadelphia, where the data was collected.

test scenario. In the context of this paper, a *scenario* is a particular route that the driver is asked to drive, and is represented as a road layout and a series of instructions (i.e., which turns to make) given to the driver in advance. In order to evaluate the performance of a learning algorithm, we assume that we have a ground truth trace T^{gt}, that captures the actual behavior of the driver in the test scenario.

Dataset

We collected a dataset consisting of data from 32 subjects. Each subject drove 5 times in a driving simulator: one practice drive and four experimental drives. The four experimental drives included representative traffic situations and allowed dynamic interactions with the participant vehicle. *Drive 1* included a variety of traffic situations because this drive is the one used as the training scenario for the machine learning algorithms. *Drive 2* was used as the *test scenario* for our evaluation. Data from Drives 3 and 4 was not used in the experiments reported in this paper.

Apparatus. We used a high-fidelity simulator located within the Center for Injury Research Prevention at CHOP (Figure 1). The driving simulator consists of a Pontiac G6 driver seat, three-channel 46" LCD panels (160 degree field of view), with rearview, left side, and right side mirror images inlayed into the center, left, and right channels, respectively, active pedals and steering system, and a rich audio environment. The simulator is powered by SimVista, a tile-based scenario authoring software, and the real-time simulation and modeling were controlled by SimCreator, a simulation tool, by Realtime Technologies, Inc.®.

Procedures. Based on the most frequent serious crash categories and configurations in NMVCCS[1] for 16-18 year-old teen drivers driving alone or only with a peer passenger we included 1) turning into opposite directions (turning left), 2) right roadside departure, and 3) rear-end events (McDonald et al. 2012). The locations of the scenarios within the drives were pre-determined and each participant experienced the three types of scenarios multiple times with the potential for simulated crashes or run-off-the-road events.

[1]The National Motor Vehicle Crash Causation Survey.

Variables. Raw variables[2] collected by the simulator included those related to participant vehicle's heading, speed, acceleration, deceleration, turning, distance to vehicle in front, location in the simulated route, and instructions provided to the driver (such as instructing her to "take the next left"). A number of derived variables have been computed to inform whether the participant vehicle collided with any other vehicle; if it was in the correct lane; if it came to a complete stop at stop-sign intersections; if pre-determined thresholds for hard braking, hard left turn, and rapid acceleration were met; and if the participant was speeding:

- *3 output variables* (*steer*, *throttle*, *brake*): these variables encode the current position of the steering wheel, and the amount the throttle and brake pedals were pressed, and are used to represent the driver's behavior.
- *3 previous output variables*: these variables encode the value of the output variables 0.1 seconds ago, for knowing wether the output variables are increasing or decreasing with respect to their previous values.
- *31 input variables*: these variables include all the other raw and derived variables described above.

We used data from 32 drivers (ages between 17 - 20). Therefore, for each of those 32 drivers, the dataset had data for the five drives described above. Each drive contained all the 37 variables described above sampled at 60Hz. The 32 traces for each of the two drives used in the experiments reported in this paper had the following lengths:

- *drive 1*: 615.03s to 1232.88s (avg. 824.51s, corresponding to an average of 49470.84 state-action pairs per trace).
- *drive 2*: 331.40s to 700.42s (avg. 477.32s, corresponding to an average of 28639.19 state-action pairs per trace).

Predicting Driving Behavior via Supervised Learning

Given the available training data, it is tempting to formulate learning from demonstration as a supervised learning problem, where each of the state-action pairs in the training set is considered one training instance. In order to see how well this approach works, we evaluated the performance of four separate learning methods:

- *Baseline*: just predict the average of the values seen in the training data.
- *Linear Regression*: assumes a linear dependency between the output variable and the input variables with coefficients learned from data.
- *M5P*: a regression tree learning algorithm (Quinlan 1992).
- *Multi-Layer Perceptron (MLP)*: a standard neural network architecture trained via backpropagation. We used the default configuration set by the MLP implementation in WEKA (Hall et al. 2009), which resulted in a 3-layer network: first layer with the inputs, one hidden layer with 17 sigmoid neurons, and the output layer with a single linear-activation neuron.

[2] We will use the terms *variable* and *feature* interchangeably.

Table 1: Prediction mean square error (MSE).

	steer	*throttle*	*brake*
baseline	3.84 ± 30.5	340.24 ± 1674.9	172.03 ± 444.4
Lin. Regr.	0.03 ± 0.6	6.79 ± 75.3	1.26 ± 13.1
M5P	1.29 ± 20.7	75.46 ± 1040.3	15.78 ± 200.07
MLP	1.90 ± 18.8	203.24 ± 1963.6	23.73 ± 210.63

Table 1 shows the mean square error (MSE) achieved by each of the methods on our dataset. For each of those methods, we trained and tested three separate models per subject to predict *steer*, *throttle* and *brake* respectively. We present the average across all 32 subjects. As expected, the results show that all models can predict *steer*, *throttle* and *brake* much better than the baseline (lower MSE), with linear regression seemingly obtaining very good predictions.

However, if we were to use these models to actually predict what subjects will do over a period of time, we would see that the predictions would be quite poor. To illustrate this, Figure 2 (top), shows the ground-truth *steer* variable plotted over the course of the whole drive 2 for one of the subjects on our dataset, overlaid with the predictions made by the linear regression method. As we can see, predictions accurately match actual data with few exceptions. However, let us define the Δ-extended prediction as a prediction generated in the following way: given a time t, we use the machine learned model to predict *steer*, *throttle* and *brake*, give this predictions to the driving simulator, and get the next time instant $t + 1$, call the model again, and repeat until we have run the predictions generated by the model for an amount of time Δ. We can then see the degree to which the mean square error of the Δ-extended prediction deviates from the ground truth as Δ increases. The MSE reported in Table 1 corresponds to using $\Delta = 0$ (i.e., comparing the immediate predictions to the ground truth). The plots shown in the middle and bottom of Figure 1 correspond to $\Delta = 1.0s$ and $\Delta = 2.0s$, where we see that predictions vary wildly from the ground truth with increasing Δ.

The reason for this is that, as previously shown by Ross and Bagnell (Ross, Gordon, and Bagnell 2011), learning form demonstration is fundamentally different from supervised learning. Namely, LFD violates the *i.i.d.* assumption of supervised learning: while supervised learning algorithms assume that training and test data are drawn from the same distribution, in LfD, the next state a learning agent encounters depends on the policy it has learned, while the training data depends on the policy executed by the demonstrator (which might be very different form the learned one). Because of this key distinction, small prediction errors compound over time, i.e., in the case of our linear regression method, the small amount of error in one time frame makes the car deviate slightly from what the human subject would have done, and bit-by-bit these slight deviations can place the car in positions off the sides of the road for which we do not have data in the training dataset and therefore, would likely result in an increased amount of error in the predictions for the next time step. In fact, Ross and Bagnell showed that error grows quadratically over time (trend confirmed in

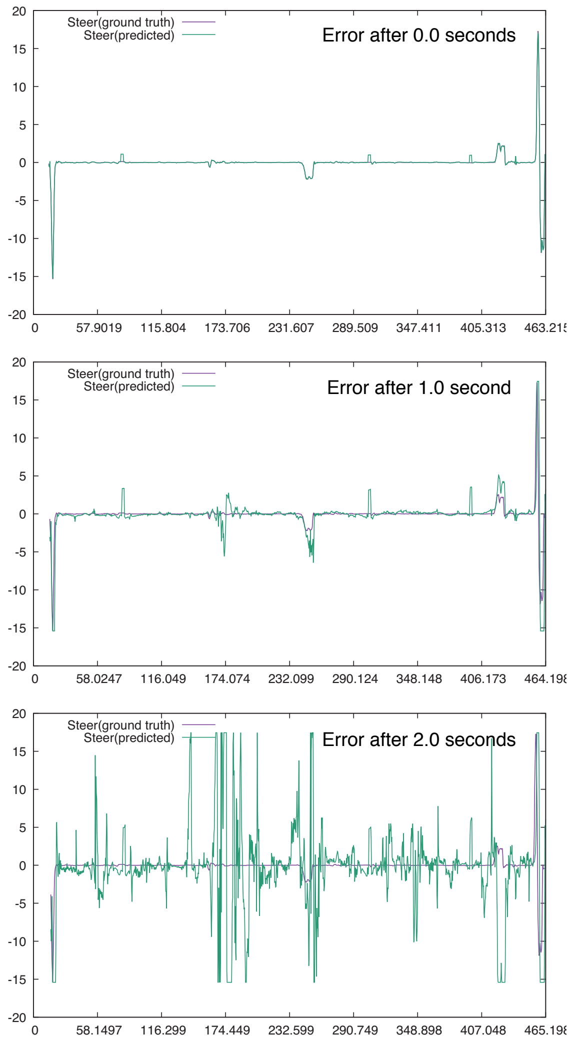

Figure 2: Comparison of actual *Steer* for subject 101 with the Δ-extended predicted steer using *linear regression* for $\Delta = 0$, 1.0 and 2.0 seconds. Horizontal axis is time, and vertical axis is the position of the steering wheel with 0 being driving straight.

our experiments) when supervised learning algorithms are used in LfD tasks, which is why we see the error grow so dramatically in Figure 2 as Δ grows. Our results show that when Δ is larger than 1 or 2 seconds, prediction error becomes as large or larger than our baseline.

LfD algorithms that address this problem exist, such as DAgger (Ross, Gordon, and Bagnell 2011), based on active learning, where the demonstrator is queried repeatedly for new demonstration data which complements the existing training data. The idea is to get data from the areas of the space that are visited by the learning agent, but that were not visited by the demonstrator originally. Such algorithms assume that the learning agent can take control of the simulation, however, and then ask the human demonstrator to take over, which is not the case in our setting. So, in the remainder of this paper, we present two improvements over the base supervised learning algorithms designed to minimize prediction error in the LfD setting, focusing on prediction error when Δ grows to larger values such as 10 seconds (i.e., predicting what the driver would do over the next 10 seconds), and assuming that we cannot request more data from the demonstrator.

Context-based Prediction

Context-based reasoning (CxBR) (Gonzalez, Stensrud, and Barrett 2008) is a reasoning paradigm based on partitioning the problem at hand into a series of *contexts* (e.g., in our application domain these could be "making left turns", "overtaking", "following a car", etc.), which in turn serve to partition the agent's knowledge into what is necessary for each of the different contexts.

In order to apply CxBR to driving behavioral modeling, we manually divided the driving task into 12 different contexts: approaching a stop sign, approaching a traffic light, making a left/right turn, navigating a left/right curve, following a car, etc. We then defined a function that automatically classifies each of the state/action pairs on our dataset into one of the 12 different contexts.

A context-based model in our scenario is then constructed as follows: Given the training data T, slice it into separate training sets, one per context. Then, train a separate machine learning model for each of the contexts. If there is any context for which there is no training data, then a *default* model is trained with the whole training set T for such context.

Similar ideas have been explored in the context of LfD in the past. For example, Sammut et al. (1992) proposed a supervised learning approach to learn how to fly airplanes by dividing the learning task into a collection of maneuvers, and then learning a model for each of them separately.

Indirect Prediction

The idea of *indirect prediction* is to predict variables whose error will not compound, and from which we can derive the control variables (*Steer*, *Throttle*, and *Brake*). We call these variables the *indirect prediction*.

To explain this concept, imagine a supervised learning method that systematically makes a small error when predicting *Steer*, making the car turn a fraction of a degree more to the right than it should. Over time, this small error will make the car drive off of the road. Imagine, however, that instead of predicting *Steer*, the method were predicting the *Road offset* (i.e., the position on the road with respect to the center line) at which the car will be in the next time instant. Now, if the prediction is systematically wrong by a small fraction of a meter, the car would keep driving straight, although just consistently a fraction of a meter away from the position the human demonstrator had demonstrated. This later situation is preferable over the first one, since we see that the error in *Steer* can compound, but the error in *road offset* does not compound.

Specifically, for our driving behavior modeling task, the indirect prediction consists of just two variables: *Velocity*

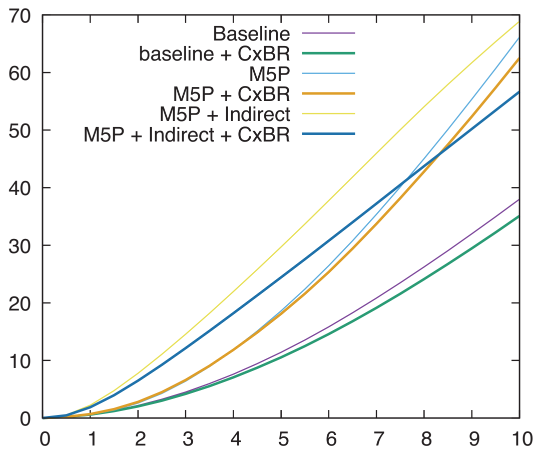

Figure 3: Average distance between the demonstrator car and the predicted position of the car (vertical axis, lower is better), as a function of the amount of time simulated (horizontal axis) for different machine learning approaches .

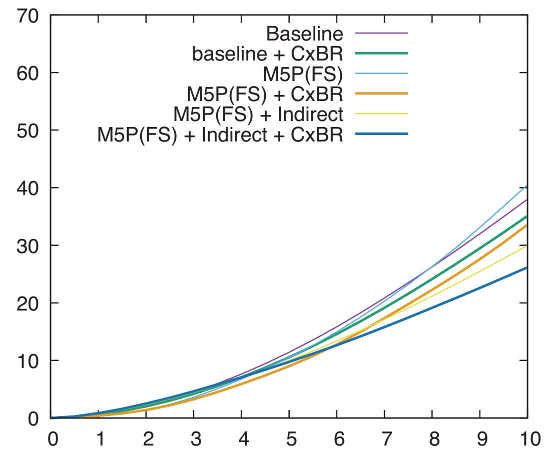

Figure 4: Same experiments as reported in Figure 3, but using feature selection. We kept both baselines for reference (lower is better).

and *Road offset*. Thus, the indirect prediction model works as follows:

1. Given a problem with input variables X and output variables Y, pick a set of *indirect prediction* variables $I \subseteq X$.
2. For each variable x_i, train a model using supervised learning to predict the value of x_i in time step $t + 1$ given the value of the variables in X for time step t.
3. Train a second model to predict Y from the indirect prediction. This second model could be trained via machine learning, or it could be hard-coded. In the experiments presented in this paper, we used a hardcoded model that determined the necessary value of *Steer*, *Throttle*, and *Brake* if we want to have the car be in the desired road offset and at the desired velocity in the subsequent time step. This is calculated using simple physics equations, and then, the values are clamped, to make sure *Steer*, *Throttle*, and *Brake* stay inside of their legal value ranges.

The key idea behind indirect prediction is that if the indirect prediction variables are part of X, and the supervised learning method always predicts values that draw according to the distribution of values seen in the training set, then the indirect prediction will always follow the same distribution as the training set, and thus, we turn LFD into $i.i.d.$, removing the error compounding.

The next section presents an empirical validation of both the CxBR and the indirect prediction ideas.

Empirical Evaluation

In order to evaluate the two proposed improvements (CxBR, and indirect prediction), we performed the following experiment. For each subject on our dataset, we trained a model using *drive 1*, and then asked it to predict the behavior of the given subject in *drive 2*. In order to evaluate the predictions, we used the following procedure:

1. Given the trace $T_{drive2} = \{(x_0, y_0), ..., (x_n, y_n)\}$ for drive 2, of length $n + 1$.
2. We selected points from that trace at regular intervals of $k = 20$ time steps (i.e., (x_0, y_0), (x_{20}, y_{20}), etc.).
3. From each of those points, we started a simulation with a duration of 600 time steps (equivalent to 10 seconds of driving). If the simulation starts at (x_i, y_i), then x_i is given to the learned model to predict y_i, which is then given to the driving simulator, to produce the world state for the next time step, which is in turn given to the learned model, etc. This results on a *predicted trace* of 600 time steps for each of the starting points.
4. We then measured the distance in meters between the position of the car in each of the predicted traces and the actual position of the car in the corresponding time step when the human subject was driving, and calculated the average distance (the prediction error) at each time step.

In this way, we can compute the average distance for predictions of different lengths. Figure 3 shows the average distance for different machine learning approaches as the prediction length increases (horizontal axis). We only report results using the M5P algorithm, which performed the best in our experiments. Performance with linear regression of MLP follows similar trends, but with slightly higher error.

CxBR: we can see that when incorporating CxBR into the baseline predictor (which just predicts the average values in the training data for each context), prediction error decreases slightly. We can see a similar effect when applying CxBR to the M5P method. The difference is not very large, but, as expected, CxBR systematically improves prediction error. The reason is that the machine learning method now only needs to learn to predict the subject's behavior in a specific context, which is an easier task than trying to learn driving behavior in general.

Indirect prediction: results based on the application of indirect prediction need to be interpreted carefully. At a first

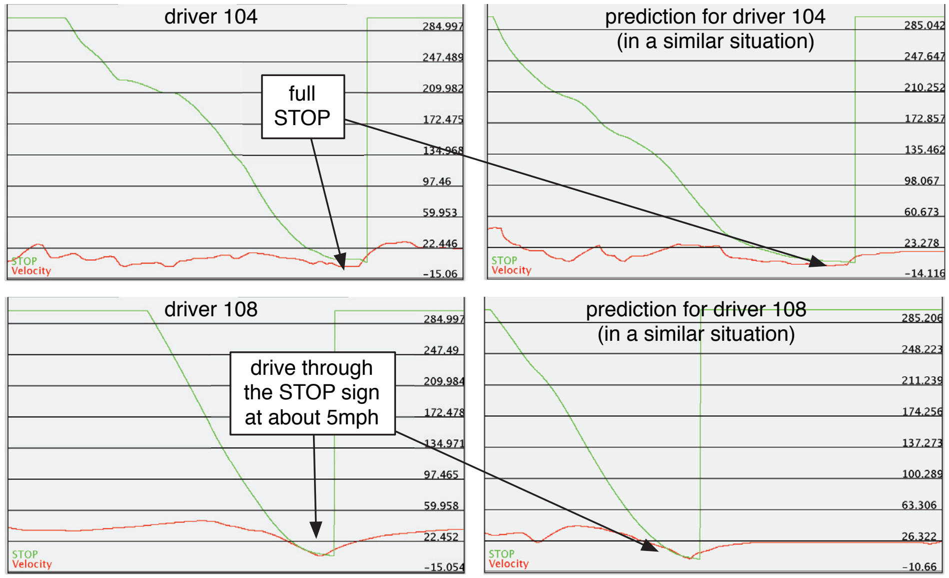

Figure 5: Screenshots showing actual behavior for two subjects (left-hand side), represented as their velocity (red line), and the distance to the next STOP sign (green line), and predictions using the indirect prediction plus CxBR approach (right-hand side).

examination, it might seem that the error of the approach that integrates M5P with indirect prediction is higher than that of just using M5P. However, at a closer examination, we see that the error is higher for shorter predictions, but the error for M5P grows quadratically (as expected), while the growth of the indirect prediction seems to grow at a linear pace, with their errors being almost the same after 10 seconds of prediction. Incorporating CxBR into the M5P + indirect model, significantly reduces the error, outperforming M5P + CxBR for predictions longer than 8 seconds.

However, notice that all of these methods result in prediction errors that are *higher* than that of the baseline predictor which just predicts the average value for each variable! In order to improve the performance of the approaches, we added feature selection to each of them. We used a standard additive wrapper feature selection method (Wettschereck, Aha, and Mohri 1997). To select features, we used the training set of one of the subjects, divided it in two sets, used one to train an M5P model, and the second as the validation set (we used the selected features for the data of this subject for all the experiments, in order to reduce computational cost, since wrapper methods are computationally expensive).

Figure 4 shows the same experiment as Figure 3, but this time using feature selection. As can be seen, the performance of all approaches improves significantly, getting now closer and most of them outperforming that of the baseline. In particular M5P combined with either CxBR or indirect prediction outperform the baselines, and when combining both CxBR and indirect prediction, performance is increased even further, achieving the lowest error for predictions longer than 6 seconds.

Finally, Figure 5 shows a few screenshots from our simulation tool, showing actual behavior for two subjects (left-hand side), represented as their velocity (red line), and the distance to the next STOP sign (green line), and predictions using the indirect prediction plus CxBR approach (right-hand side). The figures show that our model was able to accurately model the driving patterns around STOP signs of the subjects. For example, subject 104 varied speed erratically, and made full stops at STOP signs, while driver 108 changed speed smoothly, but rolled over STOP signs at about 5 miles per hour, rather than making full stops. Both behaviors were successfully captured by our approach.

Conclusions

This paper has focused on modeling and predicting human driving behavior, formulated as a learning from demonstration problem. The key contribution of the paper is a new approach to LfD that we call *indirect prediction*. The key idea is not to predict the output variables directly (which results in error compounding due to LfD violating the i.i.d. assumption), but to predict changes in the world state, from which the output variables can be then inferred.

Our results show that: 1) indirect prediction can prevent the quadratic error compounding of supervised learning in LfD tasks (although we do not provide any theoretical evidence, which is part of our future work, our empirical results show almost linear growth in error), 2) CxBR helps in further reducing the error, and 3) feature selection also helps in reducing prediction error.

As part of our future work, we would like to study further variants of indirect prediction, studying which variables of the world state can be used as the indirect prediction to further improve performance. Additionally, we would like to apply the idea of indirect prediction to other LfD problems, in order to study whether this idea is specific to driving behavior or can be generalized to a larger class of LfD problems. Finally, we are currently working on automatic techniques to infer the contexts directly from data, rather requiring manual manipulation.

Acknowledgements. This project was partially funded by the National Science Foundation (NSF) Center for Child Injury Prevention Studies and by NSF grant SCH-1521943. We would also like to thank the Industry Advisory Board (IAB) members for their support, valuable input and advice, and to Molly Tiedeken, Leif Malm, and Chelsea Ward for their invaluable help in collecting the data used for this work.

References

Curry, A. E.; Hafetz, J.; Kallan, M. J.; Winston, F. K.; and Durbin, D. R. 2011. Prevalence of teen driver errors leading to serious motor vehicle crashes. *Accident Analysis & Prevention* 43(4):1285–1290.

Das, D.; Zhou, S.; and Lee, J. D. 2012. Differentiating alcohol-induced driving behavior using steering wheel signals. *Intelligent Transportation Systems, IEEE Transactions on* 13(3):1355–1368.

Fernlund, H. K. G.; Gonzalez, A. J.; Georgiopoulos, M.; and DeMara, R. F. 2006. Learning tactical human behavior through observation of human performance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 36(1):128–140.

Floyd, M. W.; Esfandiari, B.; and Lam, K. 2008. A case-based reasoning approach to imitating robocup players. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society (FLAIRS)*, 251–256.

Foss, R. D.; Martell, C. A.; Goodwin, A. H.; and O'Brien, N. P. 2011. Measuring changes in teenage driver crash characteristics during the early months of driving. In *AAA Foundation for Traffic Safety*.

Gonzalez, A. J.; Stensrud, B. S.; and Barrett, G. 2008. Formalizing context-based reasoning: A modeling paradigm for representing tactical human behavior. *International Journal of Intelligent Systems* 23(7):822–847.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter* 11(1):10–18.

Lamontagne, L.; Rugamba, F.; and Mineau, G. 2012. Acquisition of cases in sequential games using conditional entropy. In *ICCBR 2012 Workshop on TRUE: Traces for Reusing Users' Experience*.

Macadam, C. C. 2003. Understanding and modeling the human driver. *Vehicle System Dynamics* 40(1-3):101–134.

Markkula, G.; Benderius, O.; Wolff, K.; and Wahde, M. 2012. A review of near-collision driver behavior models. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 54(6):1117–1143.

McDonald, C. C.; Tanenbaum, J. B.; Lee, Y.-C.; Fisher, D. L.; Mayhew, D. R.; and Winston, F. K. 2012. Using crash data to develop simulator scenarios for assessing novice driver performance. *Transportation Research Record: Journal of the Transportation Research Board* 2321(1):73–78.

McKnight, A. J., and McKnight, A. S. 2003. Young novice drivers: careless or clueless? *Accident Analysis & Prevention* 35(6):921–925.

Nechyba, M. C., and Xu, Y. 1998. Stochastic similarity for validating human control strategy models. *Robotics and Automation, IEEE Transactions on* 14(3):437–451.

Ontañón, S.; Mishra, K.; Sugandh, N.; and Ram, A. 2010. On-line case-based planning. *Computational Intelligence Journal* 26(1):84–119.

Ontañón, S.; Lee, Y.-C.; Snodgrass, S.; Bonfiglio, D.; Winston, F. K.; McDonald, C.; and Gonzalez, A. J. 2014. Case-based prediction of teen driver behavior and skill. In *International Conference on Case-Based Reasoning*, 375–389. Springer.

Pomerleau, D. 1989. Alvinn: An autonomous land vehicle in a neural network. In Touretzky, D., ed., *Advances in Neural Information Processing Systems 1*. Morgan Kaufmann.

Quinlan, J. R. 1992. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, 343–348. Singapore.

Rabiner, L., and Juang, B.-H. 1986. An introduction to hidden markov models. *ASSP Magazine, IEEE* 3(1):4–16.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, volume 1, 6.

Rubin, J., and Watson, I. 2011. On combining decisions from multiple expert imitators for performance. In *IJCAI*, 344–349.

Sammut, C.; Hurst, S.; Kedzier, D.; Michie, D.; et al. 1992. Learning to fly. In *Proceedings of the ninth international workshop on Machine learning*, 385–393.

Wettschereck, D.; Aha, D. W.; and Mohri, T. 1997. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review* 11(1-5):273–314.