2004 Int'l Conference on Robotics and Autonmation
http://www.cds.caltech.edu/~murray/papers/2003v_wdm04-icra.html

# Identification of Decision Rules in a Human-controlled System: Vehicles at a Traffic Intersection

C. Walton, D. Del Vecchio, and R. M. Murray
Division of engineering and applied science
California Institute of Technology
Pasadena, California 91125

***Abstract*— The rules that govern decision making in systems controlled by humans are often simple to describe. However, deriving these rules from the actions of a group can be very difficult, making human behavior hard to predict. We develop an algorithm to determine the rules implemented by drivers at a traffic intersection by observing the trajectories of their cars. We apply such algorithm to a traffic intersection scenario reproduced in the Caltech multi-vehicle lab, with human subjects remotely driving kinematic robots. The results obtained on these data suggest that this kind of human behavior is to some extent predictable on our data set, and different subjects implement similar rules.**

## I. Introduction

In the course of a day, humans are faced with many choices: Do I step on the brakes? Do I invest in this stock? Do I throw the basketball to Barry or Steve? Some of these choices are very complex, making claims of predicting their outcomes sound far-fetched. Others appear, at least on an intuitive level, to be based on little input or to be easily foreseen and classified that prediction algorithms should be simple. There are many potentially beneficial applications of such prediction algorithms. Security systems could be trained to detect abnormal behavior and report it accordingly. The occurrence of traffic accidents and traffic violations could be detected, aiding emergency medical technicians and law enforcement. It would also be advantageous in some strategic games. For example consider Roboflag, a version of capture the flag game being developed at Caltech. Each team has 2 humans and 6 to 10 semi-autonomous robotic vehicles. The humans on the team can choose to be the overall strategists, or to control specific vehicles. The robots have a limited field of vision, so the opposing team must try to find a strategy and guess the opponents' plan by observing the actions of the robots that enter their field. If the rules governing offensive robots and defensive robots could be ascertained, these robots could be more easily identified and the opposition's strategy could be better understood. Ideally, this could also be extended to human sports like soccer.

In this paper we propose a preliminary study and focus our attention on one particular human controlled system: a traffic intersection. There are many rules followed at an intersection. Some are clear-cut, like the laws of traffic and the color of the traffic lights. Others, like when to turn or stop, or when to decelerate or accelerate, are less obvious. A traffic intersection is therefore a good place to test learning algorithms, providing both trivial and nontrivial rules to test an algorithm on.

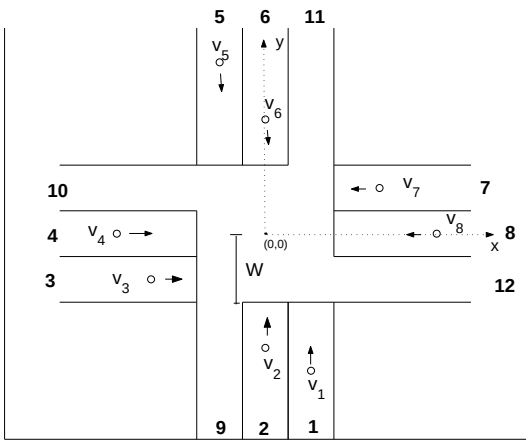


Fig. 1. Example of eight vehicles at s traffic intersection

The problem of learning the decision rules that drive a set of robots has been addressed in [3] for example, where inductive logic programming techniques [5], were used to solve the learning problem. When humans are involved in the decision making process many factors, such as the variability among subjects, give a statistical flavor to the decision rule estimation problem. In this paper we then consider standard statistical pattern recognition methods ([1], [4]) to deal with the problem. We propose an algorithm that takes as input a set of physical quantities, such as the positions of the vehicles at the intersection, and returns the probability of having a certain decision, such as "stop", "turn left", "turn right", etc.

We tested our algorithm on a human-controlled experiment. Using the Multi-Vehicle Lab at Caltech, we ran experiments where human subjects were remotely driving kinematic robots through a three-way traffic intersection. The results obtained on our data set are promising. We obtained $15.81\%$ training error and $16.13\%$ test error when training and testing were performed on different subjects.

## II. Problem Statement

We model the dynamics of a group of cars at a traffic intersection as a set of clauses. Each clause is of the form $g\ : r$, where $g$ is the guard, and $r$ is the rule. When a guard is true the corresponding rule is executed (for more details see [2]). In particular let $(x_i, y_i) \in \mathbb{R}^2$ denote the position in the $xy$ plane of vehicle $v_i$, $i \in \{1, ..., n\}$, let $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$. We model the behavior of each vehicle $i$ as

$$G_{i,j}(x, y, I)\ :\ (x_i', y_i') = r_{i,j}(x_i, y_i, \theta_i), \tag{1}$$

where $j \in \{0, ..., m\}$ with $m$ the number of possible rules that the vehicle can use to update its position. For example, a car in the neighborhood of the intersection will decide its update rule among a set of possibilities such as stop, go straight, turn left, turn right, slow down, etc., on the basis of the guard's value. $I = (I_1, ..., I_n)$ is the intention vector of the drivers. Each $I_i$ is a logic variable with $I_i \in \{1, 2, 3\}$, where $I_i = 1$ indicates the intention to go straight, $I_i = 2$ the one of turning right, and $I_i = 3$ the one of turning left. Typically this variable is known to all of the other vehicles through the signal lights. The guard $G_{i,j}$ takes the right of way, the presence of traffic lights, the configuration of the cars at the intersection as seen by the driver ($(x, y)$), personal evaluation of the situation, and intention of the drivers ($I$) into account. We assume that there are four rules that each vehicle can use to update its position , namely we have $j = 0 \Leftrightarrow\ stop$, $j = 1 \Leftrightarrow\ go\ straight$, $j = 2 \Leftrightarrow\ turn\ right$, $j = 3 \Leftrightarrow\ turn\ left$. It is obvious that if the intention of a driver at the intersection is $I_i = 1$, this will not necessarily cause the car to follow the update rule $j = 1$ since this may cause a conflict with incoming vehicles as taken into account by the $G_{i,j}$.

We address the problem of determining an estimate of the $G_{i,j}$s, by measuring the sequences $\{x_i(k), y_i(k)\}_{k \le K}$ for all $i$, and knowing the intention $\{I_i(k)\}_{k \le K}$ of each vehicle through the light signals. Number $K$ is a finite natural number as only a finite amount of data is available. This problem can be addressed by finding an estimate of the sets of the $(x, y, I)$ values for which a given $G_{i,j}$ is true or false. Depending on the structure of the $G_{i,j}$ this task can be very difficult, and in fact it can be nontrivial even for simple $G_{i,j}$'s structures as the following example shows.

*Example 2.1:* Consider vehicle $v_1$ of Figure 1. We write a possible set of pairs guard-rule to which it obeys.

$$\|(x_1, y_1)\|^2 \ge \gamma\ :\ (x_1', y_1') = (x_1, y_1 + \delta), \tag{2}$$

$$\begin{aligned} &[(I_1 = 3)\ \wedge\ \|(x_1, y_1)\|^2 \le \gamma] \\ &\wedge\ ((x_7 \ge p_{7,1}^{1,3})\ \vee\ (x_7 \le p_{7,2}^{1,3})) \\ &\wedge\ ((x_8 \ge p_{8,1}^{1,3})\ \vee\ (x_8 \le p_{8,2}^{1,3}))\ : \\ &(\theta_1' = \theta_1 + \delta)\ \wedge\ (x_1', y_1') = C + (cos(\theta_1'), sin(\theta_1'))R, \end{aligned} \tag{3}$$

$$\begin{aligned} &[(I_1 = 3))\ \wedge\ (\|(x_1, y_1)\|^2 \le \gamma)] \\ &\wedge\ ((x_7 \le p_{7,1}^{1,3})\ \wedge\ (x_7 \ge p_{7,2}^{1,3})) \\ &\vee\ ((x_8 \le p_{8,1}^{1,3})\ \wedge\ (x_8 \ge p_{8,2}^{1,3}))\ :\ (x_1', y_1') = (x_1, y_1). \end{aligned} \tag{4}$$

Clause (2) formalizes the fact that while the car is far enough from the intersection ($\|(x_1, y_1)\|^2 \ge \gamma$) the vehicle goes straight. The second clause (3) establishes that if the intention of $v_1$ is to turn left ($I_1 = 3$), if it is time to make a decision ($\|(x_1, y_1)\|^2 \le \gamma$), and if the vehicles coming from its right are both far enough ($x_7 \ge p_{7,1}^{1,3}$, $x_8 \ge p_{8,1}^{1,3}$), or if they have already crossed the intersection they are far enough ($x_7 \le p_{7,1}^{1,3}$,$x_8 \le p_{8,1}^{1,3}$) , then it can cross the intersection meeting its intention. As a matter of notation in $p_{7,1}^{1,3},\ p_{8,1}^{1,3},\ p_{7,1}^{1,3},\ p_{8,1}^{1,3}$, the superscripts $1, 3$ indicate that these parameters refer to $v_i = v_1,\ I_i = 3$, while the subscripts $7, 1$ or $7, 2$ indicate a condition on vehicle 7.

In the example we have

$$G_{1,3} = [(I_1 = 3)\ \wedge\ \|(x_1, y_1)\|^2 \le \gamma]\ \wedge g_{1,3}$$

where

$$\begin{aligned} g_{1,3} = &\ ((x_7 \ge p_{7,1}^{1,3})\ \vee\ (x_7 \le p_{7,2}^{1,3})) \\ &\wedge\ ((x_8 \ge p_{8,1}^{1,3})\ \vee\ (x_8 \le p_{8,2}^{1,3})). \end{aligned}$$

Given $I_1 = 3$ (measured) and assuming $\|(x_1, y_1)\|^2 \le \gamma$, the set of $x_7, x_8$ values for which $g_{1,3}$ is true is reported in Figure 2. This set is the union of four disjoint convex sets, while the set for which $g_{1,3}$ is false is a non convex set. In case we have more than two dimensions these sets become even more complicated. As a result, learning methods based on radial basis functions neural networks or support vector machines did not produce fast learning suitable for direct on-line implementation, and the amount of data needed to obtain a good enough estimate became prohibitive as the dimension of the variables space increased. This last phenomenon is usually known as "the curse of dimensionality".

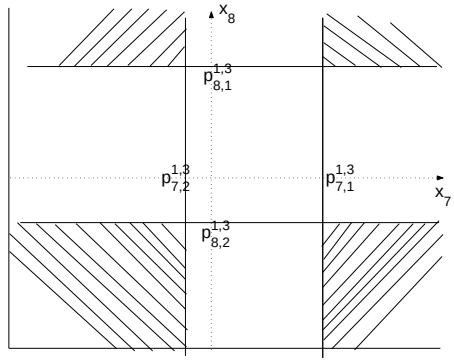


Fig. 2. Dashed regions are the ones where $g_{1,3}$ is true.

We introduce in the following section some assumptions that help to simplify the estimation task.

## III. The Model

Given a group of vehicles at an intersection we denote by $v_i$ the vehicle that approached the intersection from lane $i$ (see Figure 1). We concentrate our attention on the guards that regulate the behavior in the proximity of the intersection since the others are trivial. To model such guards we assign

a variable $d_{l,m} \in \mathbb{R}$ to each lane $l \in \{1, ..., L\}$ representing the distance from the center of the intersection of the vehicle in that lane closest to the center of the intersection. Subscript $m \in \{1, ..., n\}$ indicates what is this vehicle. If in a lane there is no vehicle $m$, then $d_{l,m}$ is set to $\infty$. Note that not all the $(l, m)$ combinations are possible. For example the combination $(9, 2)$ is not possible since vehicle $v_2$ cannot leave the intersection from lane 3. Let $l(v_i)$ denote the lane of vehicle $v_i$, then we assume the following simplified structure for $j = 1, 2, 3$:

$$G_{i,j}(x, y, I_i) = [(I_i = j) \wedge (\|(x_i, y_i)\|^2 \leq \gamma)] \wedge g_{i,j}(x, y, \mathbf{p}) \tag{5}$$

with

$$g_{i,j}(x, y, \mathbf{p}) = \wedge_{l \neq l(v_i), m} \, (d_{l,m} \geq p_{i,j}^{l,m}) \tag{6}$$

and for $j = 0$ we have

$$G_{i,0}(x, y, I_i) = \; (\|(x_i, y_i)\|^2 \leq \gamma) \; \wedge_{j \neq 0} \; (\neg g_{i,j}(x, y, \mathbf{p})), \tag{7}$$

where $p_{i,j}^{l,m}$ are parameters that need to be estimated, $\mathbf{p} = \cup_{l,m,i,j}\{p_{i,j}^{l,m}\}$, the variables $d_{l,m}$ and $I_i$ are measured for all $l, \; m, \; i$, and the value of $\gamma$ is assumed to be known (as we can imagine we have a good estimate of what the region of the intersection is in which the vehicle takes its decision). We denote by $\mathbf{d} \in \mathcal{D} \subset \mathbb{R}^{L \times n}$ the vector obtained by ordering the variables $d_{l,m}$ for all the existing $(l, m)$ combinations. We also denote by $dim(\mathcal{D})$ the dimension of the space $\mathcal{D}$.

According to this model, vehicle $v_i$ will cross the intersection if the vehicles in all other lanes are at a safe distance. Notice also that depending on the lane $l$ the values of the $p_{i,j}^{l,m}$ can be different. For example, an incoming vehicle and a leaving vehicle will be treated differently. In the model in equations (5,6,7) we have not considered the velocity of the cars in each lane, which may be an important variable to take into account in the decision process. However for this work we tried to keep the model as simple as possible to see to what extent we can predict drivers decisions even with a simple-minded model.

The values of the parameters $\mathbf{p}$ define regions in $\mathcal{D}$ in which the corresponding $g_{i,j}$ is true. Then the estimation problem reduces to learn such regions from data, and label them according to the values (true or false) of the $g_{i,j}$s. This is a standard classification problem. We seek to find a simple estimation method suitable for direct on-line implementation and not affected by the curse of dimensionality. It turns out that the particular structure for the $G_{i,j}$s assumed in equations (5,6,7) allows us to reduce the $dim(\mathcal{D})$-dimensional estimation problem into $dim(\mathcal{D})$ one dimensional estimation problems that can be solved in "one shot" using one Gaussian probability distribution function per dimension. This is formally exposed in the following section.

## IV. Estimation method

The model assumed in equations (5,6,7) is a simplified model that does not take into account important factors like the variability among drivers and among different trials for the same driver. This variability gives a statistical flavor to the estimation problem of the $G_{i,j}$, and the parameters $\mathbf{p}$ can be reviewed as average values of a probabilistic distribution. The question we ask is if such an average model is an accurate approximation of the true guards that humans use. In other words, we are asking if it is possible to find a pattern in the behavior of drivers at a traffic intersection, and therefore if their decisions are to some extent predictable. On the basis of these remarks we take a statistical approach to the estimation problem of the $G_{i,j}$. Given the structure of the $G_{i,j}$ reported in equation (5), we concentrate our attention on the estimation of $g_{i,j}$ assuming always that $I_i = j$ and $\|(x_i, y_i)\|^2 \leq \gamma$.

Since we are interested in estimating for each $g_{i,j}$ the values of the variables $\mathbf{d}$ for which $g_{i,j}$ is true and the ones for which it is false, it is enough to be able to have an estimate of one of these sets, since the other will be obtained by taking the complement with respect to $\mathcal{D}$. Thus in the sequel we concentrate our attention on the set of variables values in $\mathbf{d} \in \mathcal{D}$ for which $g_{i,j}$ is true.

From the model in equations (5,6,7) we deduce that the set of $\mathbf{d}$ values for which $g_{i,j}$ is true is a connected convex region in $\mathcal{D}$ in particular it is the intersection of $L$ semi-spaces in $\mathcal{D}$, which can be estimated from data independently from each other by estimating the probability distribution of the one dimensional variable $d_{l,m}$ for any $l, m$ corresponding to true values of the $g_{i,j}$s. We then concentrate on this one dimensional estimation problem.

Let $d_{l,m} \in \mathbb{R}$ be one component of $\mathbf{d}$ and let $\{d_{l,m}^1, ..., d_{l,m}^K\}$ be a set of $d_{l,m}$ values observed when the guard $G_{i,j}$ is true, that is when we see vehicle $v_i$ crossing the intersection. In such a data set we do not consider the values of the variables greater than a given number, as only cars that are sufficiently close to the intersection are taken into account in the decision process. As a parametric model for the probability density function we consider the Gaussian function:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}).$$

If we assume that the points $\{d_{l,m}^1, ..., d_{l,m}^K\}$ are generated independently from an unknown probability density function, and that they are identically distributed, the maximization of the likelihood of the data set

$$\mathcal{L}(\mu, \sigma^2) = \Pi_{k=1}^K p(d_{l,m}^k|\mu, \sigma^2)$$

with respect to $(\mu, \sigma^2)$ gives the well known estimates for the mean and the variance

$$\begin{aligned} \mu_{i,j}^{l,m} &= \frac{1}{K}\sum_{k=1}^{K} d_{l,m}^k \\ \sigma_{i,j}^{l,m} &= \sqrt{\frac{1}{K}\sum_{k=1}^{K}(d_{l,m}^k - \mu_{i,j}^{l,m})^2}. \end{aligned} \tag{8}$$

We consider all values of $d_{l,m}$ within $3\sigma_{i,j}^{l,m}$ from the mean likely to belong to this Gaussian distribution, that is $\mu_{i,j}^{l,m} - 3\sigma_{i,j}^{l,m} \leq d_{l,m} \leq \mu_{i,j}^{l,m} + 3\sigma_{i,j}^{l,m}$. The $3\sigma_{i,j}^{l,m}$ threshold is

empirically chosen in this work, but in principle it could be determined through an optimization process involving the data set. Moreover, given the structure assumed in (6), we will consider the values $d_{l,m} \geq \mu_{i,j}^{l,m} - 3\sigma_{i,j}^{l,m}$ likely to belong to the distribution of values for which $g_{i,j}$ is true.

Thus we finally obtain an estimate for $g_{i,j}$ as

$$\hat{g}_{i,j} = \wedge_{l \neq l(v_i),m} d_{l,m} \geq \mu_{i,j}^{l,m} - 3\sigma_{i,j}^{l,m}. \tag{9}$$

By means of equation (9) we can consider new data, corresponding to a new configuration of vehicles at the intersection, and predict the decision of each one of the vehicles $v_i$.

### A. Algorithm implementation

The algorithm that gives the estimates $\hat{g}_{i,j}$ and takes as input the data sequence $\{x(k), y(k), I(k))\}_{k \in [1,K]}$ can be summarized as follows.

- For each vehicle $v_i$, if $(W - \Delta \leq \|(x_i, y_i)\|^2 \leq W + \Delta)$ do:
  - Get $I_i$, set $j = I_i$;
    - For all existing $(l, m)$ combinations do:
      - Get $d_{l,m}$ for $l \neq l(v_i)$;
      - If $W - \delta \leq \|(x_i(k), y_i(k)\|^2 \leq W + \Delta$ for $k \in [k_1, k_2]$ and $k_2 - k_1 \leq T$ (vehicle $v_i$ did not stop
      - then $GoData_{i,j,l,m} = GoData_{i,j,l,m} \cup d_{l,m}$ for all $l \in [1, L]$, $m \in \{1, ..., n\}$;
- For each vehicle $v_i$ compute $\mu_{i,j}^{l,m}$ and $\sigma_{i,j}^{l,m}$ from $GoData_{i,j,l,m}$.

In the algorithm, $W$ is the width of the intersection (see Figure 1), and $\Delta$ is defining a small region which we refer to as the decision region, which is where we assume the driver makes its decision. The intention $I_i$ of a driver can be obtained by the signal lights of the car. In the experiments explained in the following section signal lights were not available, therefore the intention was reconstructed after having observed the trajectory of the vehicle. To decide if a vehicle stopped or not we considered an amount of time $T$ corresponding to the maximum time a vehicle was supposed to take to cross the decision region if it was not stopping. Here we make only distinctions between "stop" and "go"; a more refined algorithm could also appropriately take the acceleration and deceleration modes into account. Note that this is an off-line implementation of the algorithm, but given the small amount of computation needed to obtain the Gaussian parameters, the algorithm is suitable to be implemented on-line. We leave this to future work.

## V. Experiments

The algorithm described in the previous section was tested on real data obtained from a set of experiments done in the multi-vehicle lab at Caltech.

### A. Experiment set up

We reproduced a traffic intersection scenario in the multi-vehicle lab at Caltech involving a set of human subjects, each remotely driving one kinematic robot, see Figure 3. Each robot of about $30cm$ of diameter was equipped with a PC104 and a wireless card. It was driven remotely from another computer by establishing an internet connection through the local network of the laboratory. On each robot a C program was locally executed that allowed the robot to be driven by using the four arrow keys on the keyboard of the remotely connected computer.

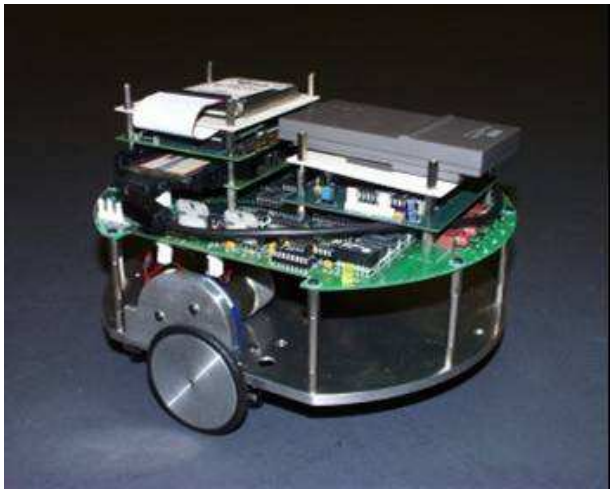

Fig. 3. Robot used as a vehicle in the experiments.

We considered a "T" intersection (see Figure 4) with the following rules:

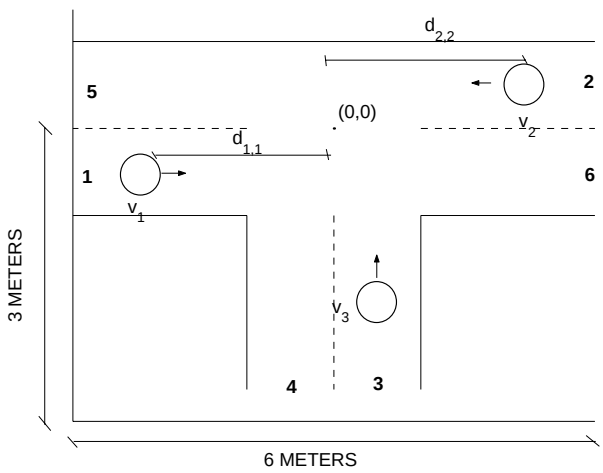


Fig. 4. Intersection simulated in the laboratory experiment.

1. Vehicle $v_1$ always has the right of way;
2. Vehicle $v_2$ yields only to vehicle $v_1$;
3. Vehicle $v_3$ yields to vehicles $v_1$ and $v_2$.

These rules were chosen to have three types of vehicles: one that yields to no other vehicle, one that yields to only one

other vehicle, and one that yields to all other vehicles. This would produce predicates $g_{i,j}$ that are true on increasingly complicated sets going from vehicle $v_1$ to vehicle $v_3$. These were the same rules given to the subjects.

The subjects were allowed to practice for half an hour before we started recording the data. The $xy$ position of each robot was recorded by means of an overhead vision system recording data at 60Hz.

Data was recorded from two sets of two experiment sessions lasting forty minutes each. We had a total of five different subjects driving the vehicles. In each experiment session, once a vehicle had exited the intersection it looped back following a prescribed path outside the intersection and started as a new vehicle in the same lane as before. After the first experiment session some of the drivers were switched and a second session was started. To have data as uniformly distributed as possible and the variables as independent as possible, in the second set of experiments we let all the vehicles loop back to the starting position and we assigned each one of them a random time they had to wait for before starting again.

### B. Experimental results

We consider three vehicles. Each vehicle is numbered according to the lane that it occupies when it approaches the intersection (see Figure 4). For each vehicle $i \in \{1, 2, 3\}$ we have two possible intentions $j \in [1, 2]$. For example for $v_3$ we have $j = 1 \;\Leftrightarrow\; turn\ left$ and $j = 2 \;\Leftrightarrow\; turn\ right$. For each vehicle $i$ the number of decision variables ($dim(\mathcal{D})$) for a given intention $j$ is equal to six, as an example consider vehicle $v_3$ and refer to Figure 4. Assume that $v_3$ intends to turn left, then it should pay attention to the following: $v_1$ in lane 1 ($d_{1,1}$) if $v_1$ is arriving, $v_1$ in lane 4 ($d_{4,1}$) if $v_1$ is leaving from lane 4, $v_1$ in lane 6 ($d_{6,1}$) if $v_1$ is leaving from lane 6, and in an analogous way $v_2$ in lane two ($d_{2,2}$), $v_2$ in lane five ($d_{5,2}$), $v_2$ in lane four ($d_{4,2}$).

We applied the algorithm described in Section IV-A on the data acquired as described in the previous section. We first report some numerical results and show how to interpret them. Consider $v_3$ and assume its intention is to turn left ($j = 2$). The estimate that we obtain for $g_{1,2}$ is given by equation (9) where we have the six decision variables $d_{1,1}, d_{4,1}, d_{6,1}, d_{2,2}, d_{5,2}, d_{4,2}$ (the combinations of $l$ and $m$ like $d_{5,1}$ are not included because they cannot occur.) In particular for the mean and the variances relative to $d_{1,1}$ and $d_{2,2}$ we obtain

$$\mu_{3,2}^{1,1} = 1.53 \quad \sigma_{3,2}^{1,1} = 0.2420$$
$$\mu_{3,2}^{2,2} = 1.40 \quad \sigma_{3,2}^{2,2} = 0.96.$$

The resulting two-dimensional probability density function is shown in Figure 5 (top). In Figure 5 (bottom) we see the boundary of the region where $\hat{g}_{3,2}$ is true, as computed from (9). From these plots we notice that $v_3$ pays a lot of attention to $v_1$ since $v_3$ decides to cross the intersection only if $v_1$ is further than about 0.8 meters from the center of the intersection. On the other hand $v_3$ is statistically not that careful to $v_2$ since the value of $\mu_{3,2}^{2,2} - 3\sigma_{3,2}^{2,2}$ is negative. In fact experimentally

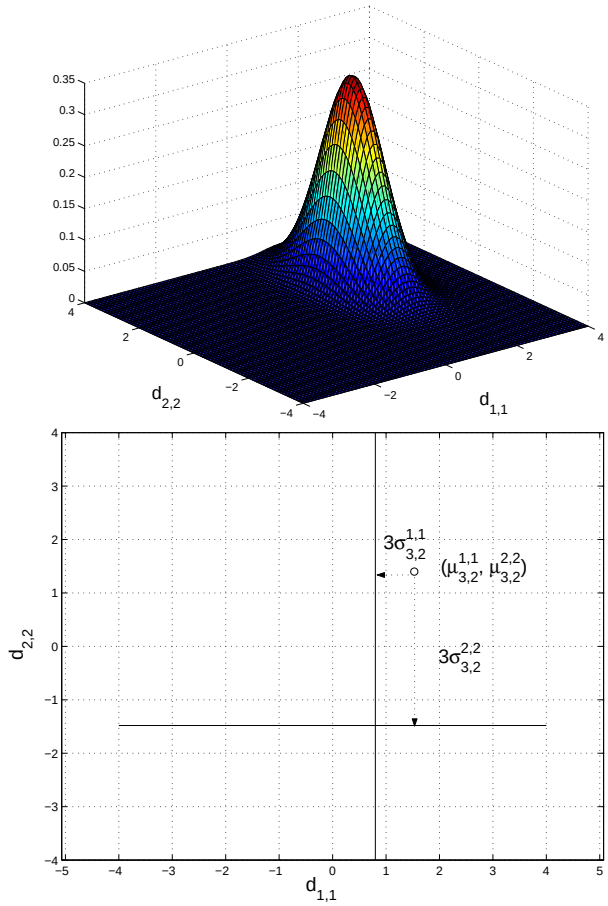


Fig. 5. Probability density for vehicle 3 with intention of turning left as a function of the distances from the intersection of two incoming vehicles $v_1$ and $v_2$ (top). Resulting boundaries of the region of the $d_{1,1}, d_{2,2}$ values for which $\hat{g}_{3,2}$ is true (bottom). The units on the coordinate axis are in meters.

we observed that even if the incoming $v_1$ is close to the center of the intersection, $v_3$ slows down a little bit and crosses the intersection slowly, so that in the time $v_3$ moves from the lower to the upper part of the intersection $v_1$ has already gone far enough. This is probably due to the geometry of the intersection. However using a more sophisticated model that discerns between "go" and "slow down" we may obtain better results.

The overall prediction results are explained in Table 1, Confusion matrix 1 and 2. In Table 1 we refer to as "train error" the error obtained on the same data used for the computation of the Gaussian parameters, while the 'test error' is the error obtained on new data samples. We counted as an error every time our algorithm predicted that vehicle $v_i$ stopped at the intersection, but it did not, and every time the algorithm predicted that vehicle $v_i$ did not stop but it did. The algorithm was tested on data obtained from subjects different from the ones used to train the algorithm.

Given the amount of the training error (15.81%) we can infer that the algorithm found a pattern in the behavior of the

drivers in our experiments, and therefore, in our data set, the behavior of drivers is predictable. Given also that the test error is not much higher than the training error, and that testing and training were performed on different subjects, there is also a good generalization across subjects on our data set. However to make a more general statement we need to acquire more data and consider more subjects, since the data extracted from the experiments for some of the variables is not enough, and the test set (as it appears from Table 1) is small. Finally, as expected, vehicle $v_3$ has the hardest guards to learn, since it has to pay attention to all of the vehicles, while vehicle $v_1$ has the easiest since it does not pay attention to any vehicle.

Another interesting aspect of the algorithm performance is reported in the confusion matrices, where we see in detail the errors on the "go" and on the "stop", that is when the algorithm predicts that the vehicle stops but it actually goes, and when the algorithm predicts that the vehicle goes but it actually stops. From the two confusion matrices we see that the biggest amount of the error is due to the error on the "stop", that is the vehicle was predicted to go, but in fact it stopped. This could be due to the fact that the algorithm does not compare two likelihoods ("go" versus "stop") but just establishes that the prediction is "go" based on the comparison of the "go-likelihood" with a fixed threshold. A clear direction in which to improve the algorithm is to also find an estimate for the "stop" distribution of variables and then make a comparison. However, this is not trivial since the set of variables for which $g_{i,j}$ is false (see equation (6)) is not a convex set, and therefore more sophisticated estimation methods need to be used. Moreover, in this case the estimation problem cannot be directly decomposed into few one dimensional problems with the clear advantage of overcoming the "curse of dimensionality".

Table 1: Algorithm error

| | train error | train samples | test error | test samples |
|---|---|---|---|---|
| $v_1$ | $8.2\%$ | 73 | $8.3\%$ | 24 |
| $v_2$ | $9.7\%$ | 92 | $14\%$ | 50 |
| $v_3$ | $21.5\%$ | 107 | $22\%$ | 50 |
| Total | $15.81\%$ | 272 | $16.13\%$ | 124 |

Confusion matrix 1: training data

| | GO (pred) | STOP (pred) |
|---|---|---|
| GO | $161/164$ | $5/164$ |
| STOP | $23/108$ | $72/108$ |

Confusion matrix 2: test data

| | GO (pred) | STOP (pred) |
|---|---|---|
| GO | $95/101$ | $6/101$ |
| STOP | $14/23$ | $9/23$ |

Note that some of the algorithm's errors are due to mistakes in the training and test sets. In fact the data recorded contained some cases in which vehicles were not respecting the rules and ended up in crashing with each other. This was due to the fact that subjects were not as comfortable driving these vehicles as they usually are when driving real cars.

## VI. Conclusion

We proposed an algorithm for estimating the decision rules that govern the behavior of human drivers at a traffic intersection. We simplified the problem to be able to obtain an algorithm easily capable of on-line implementation, and not suffering from the "curse of dimensionality". We tested our algorithm on data recorded in the multi-vehicle laboratory at Caltech, where subjects were remotely driving small kinematic robots to a three-way intersection. The overall training error was $15.8\%$ and the test error was $16.13\%$ when training and testing were performed on different subjects. This result shows that, at least on our data set, human behavior was to some extent predictable.

We proposed a preliminary study, and in the future we would like to explore how this approach may be generalized to other possibly more complicated situations. This involves performing more laboratory experiments and taking more data to test our algorithm on, as well as trying the algorithm on real traffic intersection data. There are several directions in which the algorithm could be improved. One clear direction is to find an estimate of the variables values for which the $g_{i,j}$ are false, and include in the model more details such as speed, acceleration, deceleration.


## Acknowledgment

The authors would like to thank all the subjects of the experiments.


## References


[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] E. W. Dijkstra. Guarded commands, non-determinacy and a calculus for the derivation of programs. In *Proceedings of the international conference on Reliable software*, pages 2 – 2.13, Los Angeles, California, 1975. http://portal.acm.org.

[3] T. Matsui, K. Kashiwabara, N. Inuzuka, H. Seki, and H. Itoh. Soccer agents learning from past behavior with inductive logic programming. In *Proceedings of the 1st IMC Workshop on Knowledge Mining in the Real-world*, 1999.

[4] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[5] H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*. (Lecture Notes in Computer Science Vol. 1228). Springer-Verlag, 1997.