ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Real Time Drunkenness Analysis in a Realistic Car Simulation

ROBINEL Audrey & PUZENAT Didier
{arobinel, dpuzenat}@univ-ag.fr *

Laboratoire LAMIA, Université Antilles Guyane
Campus de Fouillole - Guadeloupe (France)

Abstract. This paper describes a blood alcohol content estimation method for car driver, based on a comportment analysis performed within a realistic simulation. An artificial neural network learns how to estimate the subject's blood alcohol content. Low-level recording of user actions on the steering wheel and pedals are used to feed a multilayer perceptron, and a breathalyzer is used to build the learning examples set (desired output). Results are compared with a successful previous work based on a simple video game and demonstrate the "complexity scalability" of the approach.

1 Introduction

Driving under influence of alcohol is dangerous and a major cause of accidents. The most common approach to measure the Blood Alcohol Concent (BAC) of a driver is to use a "breathalyser". Such a device measures the amount of alcohol contained in the exhausted air blown by the subject by using a chemical reaction, and then computes the BAC from this measure. Law enforcement class devices are very reliable but expensive. Most of the low cost devices are unreliable, provide erroneous measures, and require regular calibration or parts replacement. Furthermore, the driver must not forget to test himself before driving which is very likely to happen for a drunk person. But what if the car could estimate whether or not the driver is in condition for driving, without any time consuming or even volunteer human intervention, and even reassess the driver each kilometer? Our strategy is to estimate driver's BAC through his performance [1]. In a first prototype [2], the "Tux-prototype", we have been monitoring the steering wheel used while playing a simple instrumented video game (Fig. 1, left). The Tux-prototype was a success and opened new research paths (serious gaming, teaching, emotion aware interfaces, etc.). However, the simplicity of the Tux-prototype environment does not demonstrate that our approach could be embedded to assess the driver of a real car. Therefore, the present article presents a new prototype within a definitively more complex and realistic environment (Fig. 1, center) using more realistic commands (Fig. 1, right). This new "realistic-prototype" is presented in section 2. Section 3 introduces the artificial neural networks used, and section 4 presents results for a real-time driver evaluation. The last section (5) concludes on the capabilities of our realistic-prototype regarding Tux-prototype, and presents some perspectives.

*This work has been funded by *ApportMédia* (www.apportmedia.fr), *la Région Guadeloupe* (www.cr-guadeloupe.fr), and *the European Social Fund* (ec.europa.eu/esf).

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Fig. 1: The old "Tux-prototype" (based on an instrumented version of the "super-tuxkart" game) and its basic steering wheel (left), and the new "realistic-prototype" (center) and its "force feedback" steering wheel with pedals (right).

2 The realistic-prototype

We used a driving simulation developed by ApportMédia to teach driving [4]. This simulator is extremely realistic including regarding the dynamic of the vehicle. The simulator is controlled with a Logitech G27 steering wheel and pedal board (Fig. 1, right), this steering wheel has "feedback force" capability and the simulator takes advantage of it. During an experiment, subjects are asked to enter an highway, to drive the highway (Fig. 2, left), to take the first exit, and to drive on a countryside road until it ends (Fig. 2, right). From now on we will call such a performance a "run". Furthermore, subjects are asked to drive has if it was a normal car and to follow regulation. The duration of a run depends on the subject driving speed and of the traffic, with an average of about 3 minutes. Other vehicles follow a scenario and partially adapt to our driver actions. The experiments have been conducted during a party, and subject's BAC has been measured with a breathalyzer before each run.

Fig. 2: Highway (left) and countryside (right) roads from our experiment.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

3 The Artificial neural network

The artificial neural network (ANN) used is a multilayer perceptron (MLP) with back-propagation learning, from the FANN library [5]. In order to compare results with previous ones, we kept the ANN's inputs of the Tux-prototype for the realistic-prototype; motivations for using such measures in the first place can be found in [2]. These inputs are: (i) the average duration of the user steering wheel actions; (ii) the number of steering wheel actions; (iii) the number of direction changes; and (iv) a measure of how often the driver drove on or beyond the road's limit lines. We define a steering wheel action as the succession of actions of the user on the steering wheel from the time when it left the neutral (center) value to the time when it returns to this value. A direction change occurs when the user changes the steering wheel direction from left to right, or right to left. The "out of limit lines" measure (iv) is not available on car's computer, thus such a feature would require a new sensor (e.g. to detect line crossing) or an interaction with the car's GPS. Unfortunately, we had to use measure (iv) since too few features were measurable in the Tux-prototype due to its extreme simplicity, and we wanted to keep the same features for the realistic-prototype. However, without anymore need to compare with the Tux-prototype, a future realistic prototype will easily replace measure (iv) with one or several more suitable alternatives – which does not require to add new sensors – since numerous features are available within such a rich environment.

3.1 Evaluation of the neural network responses

For each example, we know the subject's BAC since all subjects are assessed with a breathalyser before "driving". When the network returns an estimation, we compute the error for this example (noted E) and an Average Error (noted AE) that is the average E computed over all the examples tested in generalization.

$$E = |(measured\ BAC) - (ANN's\ BAC\ estimation)|$$

We will also compute a Success Rate (SR) in generalization by dividing the amount of correct estimations by the total number of tests, using an ε threshold to determine the correctness of an estimation (i.e. if $E \leq \varepsilon$). In order to obtain more accurate results, we will also compute an Average of the Success Rates (noted ASR) for values of ε ranging from 0.1 to 0.15, by increments of 0.01 :

$$SR_\varepsilon = \frac{number\ of\ succes\ for\ an\ \varepsilon}{number\ of\ tests} \qquad ASR = \frac{\sum_{\varepsilon=0.1}^{0.15} SR_\varepsilon}{6}$$

3.2 Methodology for testing the performances of the network

We used the "Leave-One-Out" cross-validation algorithm [3] to test the generalization success rate of our neural network: the ANN is trained on a learning base containing all examples but one, then tested on the excluded example. The operation is repeated until the generalization capabilities of our ANN has been tested for each example. This algorithm maximize the number of examples used for learning, without ever testing the ANN with an example used for learning.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

4 General results and real time analysis

For a first experiment, we trained our ANN using full runs, that is with data collected during the full duration of the run. The network achieved an *ASR* of 79%, and an *AE* of 0.085 g/l. With the Tux-prototype, we obtained an *ASR* of 77% in the best configuration [2] (and 69% in general) and the best *AE* was 0.0836 g/l (0.0997 in general), which means that our realistic-prototype is on par or better than its simple ancestor in this setup.

4.1 Real time analysis using "time windows"

We will now analyse our realistic-prototype performances when using subparts of a run, to perform real-time estimations. For this purpose, we use "windows" that is subparts of the whole run defined by a starting time and a length (in seconds). As an example, the window $[t_0, D]$ leads to examples constructed using all the measures from $t = t_0$ to $t = t_0 + D$ of a run. For the best windows size we found (starting at $t_0 = 15$s and $D = 150$s), we obtain an *ASR* of 97% and an *AE* of 0.045 g/l, which is a huge improvement over the best results of Tux-prototype.

4.2 Impact of the window's size on the ANN's average success rate

We will now see how the *ASR* of the prototype is impacted by the window's length. For that purpose, we trained the ANN with various configurations of the windows, with a starting point set to $t_0 = 30s$ for all windows. However, the length of windows ranged from 20s to 150s, as illustrated in figure 3. All other parameters remained constant. We obtained the best results with the largest windows (see Fig. 4 for details), but we note that 50s windows provide interesting results (*ASR* of 73%). The Tux-prototype performed better with small windows sizes (less than 50s), but this was expected since we still use only 4 features while the realistic-prototype is quite more complex: variability of the situations (type of roads, traffic on the highway, etc.), the use of the pedal to control the car speed while in the Tux-prototype speed was constant, etc. At last, the realistic-prototype has a lower *AE* (0.08 g/l compared to the 0.12 g/l for the Tux-prototype [2]) when considering 50s windows, and results improve with larger windows.

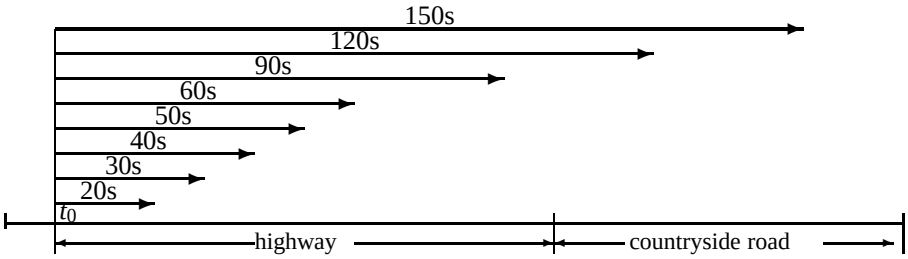

Fig. 3: Several sizes of windows, with $t_0 = 30s$, and the corresponding type of road (highway or countryside road). The exact localization (in time) of the highway exit and of the end of the simulation depend on the speed of the car and on traffic.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

D	ASR
20	44
30	66
40	65
50	73
60	78
80	76
100	74
120	82
150	83

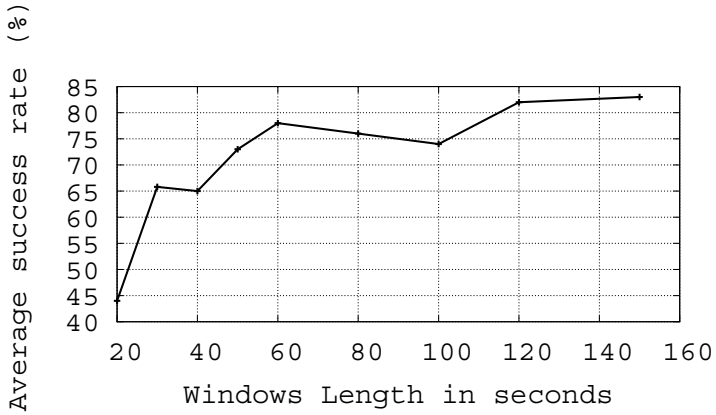

Fig. 4: Average Success Rate (ASR) in percent, for various sizes of windows (D).

4.3 Results of the ANN when using two non-overlapping windows

We will now generate two non-overlapping windows per run, thus generating two examples per run. One of the window cover highway driving while the other cover countryside driving. By this mean we will be able to analyze how the ANN performs when using examples from different road sections. We obtain an *AE* of 0.0841 g/l and an *ASR* of 75% in this configuration. Compared to a similar experiment with the Tux-prototype (with two different racing circuits), we obtain a much lower average error since the *AE* was of 0.123 to 0.156 g/l for Tux-prototype in similar configurations [2]. The increased complexity did not hinder the ANN's capability here. This demonstrates that the realistic-prototype is capable of efficient real time BAC estimation, even if limited to Tux-prototype features (that is Tux-prototype's ANN inputs) to allow significant comparisons.

5 Conclusions and perspectives

The Tux-prototype was intended to be very simple, and quite few valuable features were usable (i.e. possible measures). On the contrary, the realistic-prototype is very complex, with a high variability between the runs, but it offers hundreds of potential features, and consequently many possible and valuable configurations of features. However, the 4 features we used in the article where sufficiently discriminant to be able to achieve results equivalent to those of Tux-prototype or even better in terms of both average success rate (*ASR*) and mean error (*AE*), except for the smaller windows sizes of less than 50 seconds. Such results demonstrate the "complexity scalability" of our approach.

For short-term future works, we plan to study the best possible configurations of features for our realistic-prototype, and we expect to be able to further improve the results, especially regarding the real time assessment of the driver. Those future works will introduce new challenges, e.g. the need of a largest set of examples to deal with the increase of the number of features. Therefore, we are developing new methods and tools to automatically explore all possible feature configurations and to tune the ANN. Regarding windows lengths (for real-time assessment), and beyond the present

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

academic work, we plan to use a window length based on the nature of the physiological phenomenon estimated, that is about 20 minutes for BAC. Finally, we plan to use our work to deal with other problematics such as detecting driver fatigue, or to assess a subject aptitude to drive (for example after an accident).

For middle-term and long-term future works, we hope to have the opportunity to experiment with real cars, embedding our artificial neural network in the car and exchanging with the car's computer to collect the ANN's inputs and eventually to send back outputs of the ANN to the car's dashboard. This long-term research will be proceeded by the use of a Barracuda simulator from ApportMédia (Fig. 5). We are already using the software of the Barracuda for the realistic-prototype, but the Barracuda controls are the ones of a real car.

Fig. 5: The "Barracuda 2" simulator that ApportMédia should soon make available for our research, with the same software as for our realistic-prototype, but with a gear shift, handlebar controls, a parking break, and a familiar dashboard.

References

[1] D. Puzenat and I. Verlut, Behavior analysis through games using artificial neural networks. In *proceedings of the Third International Conferences on Advances in Computer-Human Interactions* (ACHI 2010), pages 134-138, Sint Maarten (Netherlands).

[2] A. Robinel and D. Puzenat, Real time drunkenness analysis through games using artificial neural networks. In *proceedings of the Fourth International Conferences on Advances in Computer-Human Interactions* (ACHI 2011), pages 206-211, Gosier (France).

[3] S. Arlot and A. Celisse, A survey of cross-validation procedures for model selection. In Statistics Surveys, 4, pages 40-79, American Statistical Association, 2010.

[4] ApportMédia, "Stars AF 2011" realistic car simulator product on-line documentation `http://www.simucar.com/logiciels/STARS_AF.html`.

[5] S. Nissen, Implementation of a Fast Artificial Neural Network library (FANN). Technical report, Department of Computer Science University of Copenhagen, October 2003.

[6] C. M. Bishop, Neural Networks for Pattern Recognition. Clarendon Press, Oxford, 1995.

[7] E. Bogen, Drunkenness, a quantitative study of acute alcoholic intoxication. California and Western Medecine, XXVI(6):778-783, 1927.