Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (IX CBRN)
Ouro Preto 25-28 de Outubro de 2009

DETECTING CAR ACCIDENTS BASED ON TRAFFIC FLOW MEASUREMENTS USING MACHINE LEARNING TECHNIQUES

L. D. Tavares*, D. A. G. Vieira*, R. R. Saldanha*, W. M. Caminhas*

*Department Electrical Engineering
Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG - Brazil

Emails: tavares@cpdee.ufmg.br, douglas@cpdee.ufmg.br, rodney@cpdee.ufmg.br, caminhas@cpdee.ufmg.br

Abstract— This paper deals with the problem of detecting the occurrence of a car accident in an urban environment. For this purpose, machine learning techniques are trained with the traffic flow measurements considering both the normal and the situation in which the accident caused a partial closure of the lanes. Several machine learning techniques results are presented to several car breaking scenarios.

Keywords— Computational Intelligence, Cellular Automata, Traffic Jams Detection, Car accident simulation, Machine Learning.

1 Introduction

The land transportation system is an important resource for the country economy and population well-being, thus, when this system does not work well, several sectors are affected. Considering, for instance, the urban transit system of a Brazilian large city, such as São Paulo or Belo Horizonte, this problem can be even more serious. In these cities the most common problem is related to congestion. Congestion can be generated when the number of vehicles is greater than the capacity of the road or for any momentary interruption (accidents or maintenance of the road). Therefore, it is necessary to develop tools that can detect the moment and place these problems occur. Hence, a corrective action can be taken in order to returns the flow to its normal state. The objective of this paper is to conduct a comparative study of different classifiers in order to detect congestion in an urban traffic. For this study it was built a simulator of Urban Traffic flow using Cellular Automata (CA), called Cellular Automata for Urban Traffic Simulation (CAUTS). This model considers the presence of cars, trucks, traffic lights, buses and bus stops.

CA is, in short, the mathematical model discrete in time, space and states. Its fundamental unit is called *cell*. This kind of model is based on two simple components: local rules and neighborhood. Local rules are responsible for calculating the next state of the cell, based on the influence of its neighborhood. Only with those components CA can reproduce (simulate) dynamic complex systems, ranging from biology to chemical reactions (Wolfram, 1983-1986). CAUTS has resources capable of simulating most of the features of an urban traffic as main roads, secondary roads, traffic lights and bus stop. Moreover, it is possible to generate events that cause traffic jams, such as stopped vehicles and accidents, which is the main focus of this work. The database was test with different methods of classification, so that it can detect which part of the model and at what time an incident occurred. The classifiers used were: (i) Naïve Bayes (NB), (ii) Decision-Tree (DT), (iii) K-Nearest Neighbor (K-NN), (iv) Multi-layer Perceptrons (MLPs), (v) Support Vector Machine (SVM), (vi) Adaptive Neuro-Fuzzy Inference Systems (ANFIS). The paper is organized as follows. Sections 2 and 3 show the basic concepts employed in the construction of CAUTS model. Then, Section 4 contains the results obtained, considering several different scenarios. Finally, the conclusion and future works are in Section 5.

2 Simulator Features

Among several methods to traffic flow simulations, the ones based on the use of Cellular Automata (CA) have received an especial attention of researchers. Some papers from the 1990's presented the bases concerning the use of CA for traffic flow (Blue et al., 1996)(Nagel and Schreckenberg, 1992)(Schadschneider and Schreckenberg, 1993)(Villar and de Souza, 1994)(Nagel, 1996). These results considered the basic acceleration, deceleration, velocity randomization and velocity update rules. A review considering road traffic flow can be found in(Maerivoet and Moor, 2005). It shows that most of the concerns are related to acceleration, deceleration and lane changes for freeways. Makowiec and Miklaszewski (Makowiec and Miklaszewski, 2006) added supplementary rules to the traditional model such a way to increase the mean velocity. It is expected that most of drivers want to travel as close as possible to the maximum allowed speed. The CA is a very useful and efficient method, and can be applied to online simulation of traffic flow, as presented in (Wahle et al., 2001). In

(Boccara and Fuks, 2000) it was derived the critical behavior of a CA traffic flow model by means of an order parameter breaking the symmetry of the jam-free phase. Fuks (Fukś, 1999) considered a deterministic CA model and derived a rigorous flow at arbitrary time. Other important aspect, is the jamming caused by the reduction of the number of lanes. This reduction can be due to repairing, accidents and even because it is part of the road design. Studying the road capacity, Nassab et. al. (Nassab et al., 2006) considered a road partial reduction from two lanes to one lane. The blockage of one lane, caused by an accident car, was recently studied in (Zhu et al., 2009). This paper considers the study of a car accident in an urban environment. By urban environment it is required to consider: (i) multi-lane traffic flow; (ii) crossroads; (iii) traffic lights; (iv) trucks; (v) buses and; (vi) bus stops. The presence of buses and bus stops requires specific rules. These rules are important to the traffic flow in urban areas.

3 Model Definition

The model of urban traffic flow is implemented based on a two-dimensional Stochastic Cellular Automata, called Cellular Automata for Urban Traffic Simulation - CAUTS. CAUTS has resources capable of simulating the features of an urban traffic as main roads, secondary roads, traffic lights and bus stop. Moreover, it is possible to generate events that cause traffic jams, such as stopped vehicles and accidents. The sub sections below will detail the proposed model.

3.1 Maps definitions

The cell of the model can assume one of two states: 0 - empty, 1 - occupied. All cells of the model are square with side equal to 5.5 meters. This measure represents the average sized car in the Brazil, taking into account the distance between cars. The properties of cells are defined as a triple: $c_{i,j} = \{pd, sd, vmax\}$, where: (i) pd is the predominant direction; (ii)sd is the secondary direction; (iii) $vmax$ is the speed limit. For predominant direction, means, the direction in which the vehicle will stay longer; and, by secondary direction the change route or direction, such as lane-changing or street change. The speed limit determines how many cells can be advanced forward, at most, per iteration. Each direction d has a code, and their respective sift in the axis x and y, as can be illustrated in the Figure 1. Moreover, it allows a vehicle to move forward up to 3 cells. To indicate that a cell is not available for transit and the end of road (cell where vehicle is removed from model), two triples, $\{0, 0, 0\}$ and $\{9, 9, 9\}$, are used, respectively.

3.2 Environmental rules

These are the rules that change a set of cells to implement some desired characteristics. One of the most important feature in urban traffic is the presence of traffic lights. Consider the complementary set of traffic lights T_1 and T_2, where the cells affected by these sets are defined as $T_1 = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Similarly, consider T_2, where, for example, T_1 is the set of traffic lights in the main road and T_2 in the secondary road. The complementarity, then, is defined by: $T_1(Green) \Rightarrow T_2(Red)$, $T_1(Yellow) \Rightarrow T_2(Red)$, $T_2(Green) \Rightarrow T_1(Red)$, $T_2(Yellow) \Rightarrow T_1(Red)$. The Equation 1 shows how the traffic lights can be modeled.

$$(RT) : \begin{cases} T(Red) & \Rightarrow vmax_T = 0 \\ T(Yellow) & \Rightarrow vmax_T = 1 \\ T(Green) & \Rightarrow vmax_T = vmax \end{cases} \quad (1)$$

$\forall (x, y) \in T$.

As mentioned, the model contains features that may cause breaks in some regions, is broken by vehicles or accidents. Consider the set $A = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ as the location of the cells where the incident occurs, at time t_0 with k iterations long. Additionally, consider $cod(A, t_{0-1}) = cod(A, t_{0-1})$ as the cell triple code before the incident. Since the cells not available for transit is represented using $\{0, 0, 0\}$, then, the presence of stopped vehicle is modeled as:

$$(RA) : \begin{cases} cod_{(A,t)} = 000, & if\ t \leq t_0 + k \\ cod_{(A,t)} = cod_{(A,t_0-1)}, & otherwise \end{cases} . \quad (2)$$

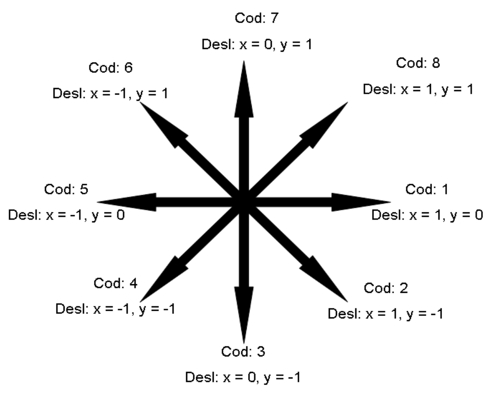

Figure 1: Code and respective dislocation (in x and y axis).

At the beginning and the end of each road there is one sensor. These sensors are responsible for capturing the statistics, such as, number of vehicles (flow) and their speeds.

3.3 Vehicles definitions

The model implemented has three types of vehicles: small vehicles, like cars, and large vehicles, such as buses and trucks. Small vehicles occupy only one cell, while large vehicles occupy three cells in length and the width of one cell. Currently, the model considers that large vehicles can only move in the main roads and can not switch lanes or routes. Buses and trucks differ, themselves, by the fact that buses have to stop at bus stops. The vehicle models have the following structure: (i) kind of vehicle: $1 - car$, $2 - bus$ or $3 - truck$; (ii) vehicle location (x, y); (iii) lane change indicator (t_1); (iv) vehicle current speed (vel_i); (v) time of the vehicle last stopped (t_2); (vi) sensor identifier (sid). The feature (iii) is applied only when the vehicle is a car and indicates how many iterations from the vehicle changed lane for the last time. This serves to prevent the car change its lane by consecutive times. Because of this, the model does not allow a vehicle leaving right lane and go to left lane, whereas there is a central lane, instantly. Consider a vehicle v_i in the set $V = \{v_1, v_2, \ldots v_i, \ldots v_n\}$ at the moment t. The location of the vehicle may be recovered by the expression:

$$loc_i = \begin{cases} [x_1, y_1] = posi(v_i), & if\ v_i = car \\ \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = posi(v_i), & otherwise \end{cases} . \tag{3}$$

Consider the location of all vehicles as $LOC = posi(V)$. The function $dir_i = direc(v_i)$, where $dir_i = [xd_1, yd_1]$ for small cars, indicates the vehicle moving, according to the Figure 1. For instance, a vehicle is moving to the east, the function $direc(.)$ will be $[xd, yd] = [1, 0]$ and $[xd, yd] = [-1, 1]$ for northwest moving. The current speed of the vehicle vel_i is accessed through the function $speed(v_i)$. The maximum speed that a vehicle can achieve depends on its type and its location at time t, as small cars tend to be faster than large vehicles in urban traffic. The speed is computed as cells/iteration, of c/i. The speed limit is calculated by the function $vmax_i = velocmax(v_i, loc_i)$. The Equation 4 defines the rule for local acceleration. This rule represents the intention of the driver to speed up as possible, i.e., the speed limit of the road will be respected.

$$(R1) : vel_{(i,R1)} = min\left(vel_{(i,t)} + 1, vmax_i\right) \tag{4}$$

However, we know that drivers may, so seemingly random, reduce vehicle speed. Consider *alpha* as the probability of a slowing down, then the local rule for this event is given by 5.

$$\begin{array}{l} if\ rand < \alpha_i, \\ (R2) :\ vel_{(i,R2)} = max\left(vel_{(i,R1)} - 1, 0\right) \end{array} . \tag{5}$$

The previous local rule is a representation of a natural factor in the urban transit system and, in some way, can contribute to the rise in congestion. Another condition for the deceleration of the vehicle is the existence of obstacles on the road. The $nfree_i = gap(v_i)$ function is responsible for identifying the maximum number of free cells in which the vehicle can move in a given direction d, according to the Figure 1. The local rule for the downturn by obstacles is given by Equation 6.

$$(R3) : vel_{(i,R3)} = min\left(vel_{(i,R2)}, nfree_i\right) \tag{6}$$

The rule $R3$ simulates, to some extent, the vision of the driver, it means, the maximum that he can move is a combination of factors: the road speed limit, maximum speed that the vehicle can reach and the next obstacle. Furthermore, it is defined in the model rules for local buses to consider the bus stops. Consider $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ the set of cells located in a bus stop. For a bus v_i, consider t_0 the moment where $loc_i \in S$ and k the stop duration, with a probability φ_i, defines de rules RS, as shown in Eq. 7.

$$\begin{array}{l} (RS) : \\ If\ loc_i \in S,\ rand < \varphi_i,\ t < t_0 + k\ and\ v_i = bus \\ \quad vel_{(i,t+1)} = 0, \\ otherwise \\ \quad vel_{(i,t+1)} = vel_{(i,R3)} \end{array} . \tag{7}$$

Finally, the movement of the vehicle v_i given the direction d of displacement dir_i can be calculated by:

$$(R4) : \quad loc_{i,t+1} = loc_i + vel_{i,t+1} * dir_i\ . \tag{8}$$

4 Simulations and Results

4.1 Environment

The Figure 2 illustrates the layout of the implemented map to the simulator. It consists of 1 main (horizontal) and 3 via secondary (vertical) roads.

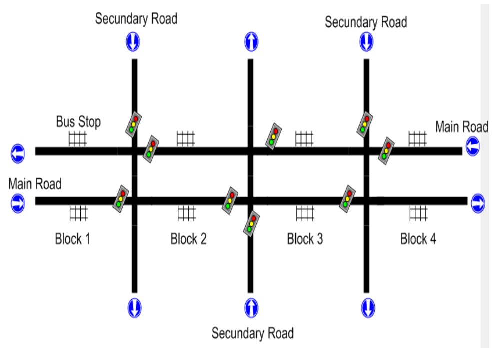

Figure 2: Layout of implemented map of CAUTS

The main routes are composed of 3 lanes and its maximum speed allowed is 60 km/h (or 3 cells per iteration); furthermore, the secondary roads have only 2 lanes and maximum speed allowed is 40 km/h (or 2 cells per iteration). The entry of vehicles in the model is given in the following way:

1. West-east Main roads: Probability of at least 10% of a vehicle entering the model outside the time of greatest movement. This probability increases linearly up to 70% between the hours of 7:00 a.m. to 8:00 a.m.. And, 50% between the hours of 12:00 to 1:00 p.m.

2. East-West Main roads: Probability of at least 10% of a vehicle entering the model outside the time of greatest movement. This probability increases linearly up to 50% between the hours of 12:00 a.m. to 1:00 p.m.. And, 70% between the hours of 4:00 p.m. to 5:00 p.m.

3. Secondary streets: Probability of at least 10% of a vehicle entering the model outside the time of greatest movement. Increasing 30% in the hours between 7:00 a.m. and 8:00 a.m., 12:00 and 1:00 p.m., and, 4:00 p.m. and 5:00 p.m..

For all scenarios are carried out 30% of large vehicles (between bus and trucks), and, all simulated accidents occurred on the central lane of the west-east main road, but in different blocks.

4.2 Parameters and scenarios

The parameters used for the classifier are:

1. DT: was implemented using the C4.5 method, maximum depth=5;

2. K-NN: $k = 17$ and $d =$ Euclidean distance;

3. ANN: MLP neural network with four layers, being [2, 25, 25, 2] the number of neurons in each layer;

4. Fuzzy: Sugeno ANFIS using has 5 membership functions (Gaussian) for each entry;

5. SVM: $\xi = 0.5$;

We simulated 5 different scenarios (08:00 to 09:00) with situations: (i) without incidents; (ii) incident in the first block, (iii) incident in the second block, (iv) incident in the third block and, (v) incident in the fourth block, according to Fig. 2. The techniques were trained with the same parameters for all considered the scenarios. The networks are trained to detect between the situations without and with accidents, therefore, they always consider situation (i) as reference.

4.3 Results

Tables 1-4 present the results in descending order accuracy for all tested topologies, ranging from 99% to 85%. For each scenario there are 400 records, of which 80% were used for training and 20% for validation. Accuracy is the average of 35 runs for each classifier (using validation data). For each simulation the training and validation set are randomly split. The results for all tested topologies presented good accuracy. This is mainly due to the fact that a consistent (big enough) dataset can be arbitrary generated using the CAUTS model.Moreover, it appears that a breakdown in the first blocks is harder to detect than in the last ones. As it is well known, traffic jams propagates backwards, therefore, the information of the first sensor are richer than in the last ones. Indeed, more information is got when the accident takes place in last blocks. This empirical expectation is observed in Tables 1-4.

Classifiers Performance	
Method	Accuracy
MLP	96.50%
SVM	95.17%
DT	92.46%
KNN	92.02 %
NB	86.12%
ANFIS	85.68%

Table 1: Performance of classifiers considering the scenarios (i) x (ii).

Classifiers Performance	
Method	Accuracy
DT	96.66%
SVM	92.21 %
ANFIS	90.95%
MLP	88.78%
NB	88.13%
KNN	87.48%

Table 2: Performance of classifiers considering the scenarios (i) x (iii).

Classifiers Performance	
Method	Accuracy
MLP	99.87%
NB	94.38%
DT	94.37%
KNN	92.64%
ANFIS	90.39%
SVM	89.14 %

Table 3: Performance of classifiers considering the scenarios (i) x (iv).

Classifiers Performance	
Method	Accuracy
ANFIS	99.12%
MLP	97.34%
NB	93.68%
KNN	92.63%
SVM	91.67%
DT	89.15%

Table 4: Performance of classifiers considering scenarios (i) x(v).

5 Final Considerations and Future Works

This paper has studied the use of machine learning techniques to detect car breakdowns in an urban environment. Measurements of traffic flow in several points in the main road are used to train the techniques. These measurements were simulated in our model called CAUTS. Using this simulator it is possible to generate several scenarios with low cost. Combining the tested methods in a voting machine will be explored in a future work. Additionally, this technique, which is based solely in the traffic flow, can be also combined with other ones, as ones based on computer vision. Indeed, detecting the traffic jams is one important aspect in the traffic flow control. Based on this detection, the traffic lights can be adjusted such a way to decrease the harsh caused by the breakdown. This is one of the future aspects to be explored in this work. In fact, it is important to improve both, the CAUTS model and the machine learning techniques.

References

Blue, V., Bonetto, F. and Embrechts, M. (1996). A cellular automata of vehicle self organization and nonlinear speed transitions, *Proceedings of Transportation Reserach Board Annual Meeting*, Washington, DC.

Boccara, N. and Fuks, H. (2000). Critical behaviour of a cellular automaton highway traffic model, *Journal of Physics A: Mathematical and General* **33**(17): 3407–3415.

Fukś, H. (1999). Exact results for deterministic cellular automata traffic models, *Phys. Rev. E* **60**(1): 197–202.

Maerivoet, S. and Moor, B. D. (2005). Cellular automata models of road traffic, *Physics Reports* **419**(1): 1 – 64.

Makowiec, D. and Miklaszewski, W. (2006). Nagel-schreckenberg model of traffic - study of diversity of car rules, *International Conference on Computational Science (3)*, pp. 256–263.

Nagel, K. (1996). Particle hopping models and traffic flow theory, *Physical Review E* (3): 4655–4672.

Nagel, K. and Schreckenberg, M. (1992). Cellular automaton models for freeway traffic, *Physics. I* (2): 2221–2229.

Nassab, K., Schreckenberg, M., Boulmakoul, A. and Ouaskit, S. (2006). Effect of the lane reduction in the cellular automata models applied to the two-lane traffic, *Physica A: Statistical Mechanics and its Applications* **369**(2): 841 – 852.

Schadschneider, A. and Schreckenberg, M. (1993). Cellular automaton models and traffic flow, *Physics A* (26): 679–683.

Villar, L. and de Souza, A. (1994). Cellular automata models for general traffic conditions on a line, *Physica A* (211): 84–92.

Wahle, J., Neubert, L., Esser, J. and Schreckenberg, M. (2001). A cellular automaton traffic flow model for online simulation of traffic, *Parallel Computing* **27**(5): 719 – 735.

Wolfram, S. (1983-1986). Statistical mecachics of cellular automata, *Theory and Applications of Cellular Automata*, World Scientific.

Zhu, H., Lei, L. and Dai, S. (2009). Two-lane traffic simulations with a blockage induced by an accident car, *Physica A: Statistical Mechanics and its Applications* **388**(14): 2903 – 2910.