Markov Models and their Application to Support Police Chase

Fernando Martínez[1], Víctor Hernández[1], Luis González[1], and Gabriel González[2]

[1]Universidad Autónoma de Chihuahua, Facultad de Ingeniería, 31125, Chihuahua, MX
{fmartine, vhernandez, lcgonzalez}@uach.mx
[2]Centro Nacional de Investigación y Desarrollo Tecnológico, Palmira, 62490, Cuernavaca, MX
gabriel@cenidet.edu.mx

Abstract. For any country the safety of citizens is of high priority. Social security should be provided opportunely and efficiently without exposing the lives of people. Developed countries have a large scale of technology to monitor criminal activities in urban spaces. This technology helps central station to coordinate police patrols when on the chase of a car. Developing countries on the other side often have technological limitations that constraint the police possibilities to chase law offenders. This work presents a vehicular network platform to help police departments particularly with situations of chasing a car on the run. We highlight the work done with the utilization of Markov chains to predict future location of the offender's car. At the current stage the algorithm can predict the escape route with 40% of success. SUMO simulator is being used to test the route predictor.

Keywords: vehicular networks, police chasing, urban computing, security and police assistance.

1 Introduction

One of the main concerns for a government is to offer a secure urban ambient for citizens, a task that can be challenging. The European Forum for Urban Security (Efus) issued the "Manifesto of Aubervilliers and Saint-Denis" in which member cities are called for the establishment of policies for crime prevention. For instance, surveillance cameras are employed in developed countries to monitor criminal activities. This information enable police department to offer quick responses. Other technologies such as sensor networks complement video monitoring and allows for tracking criminal on the move. On the other hand, the lack of technology in developing countries compromises the secure environment offered to citizens. Mobile technology together with the cloud services help explore solutions to support and assist the police department activities in the streets, An example of this situation could be, the logistic for the creation of a virtual barrier in order to constraint the space of navigation for a car being driven by a criminal. In order to clarify our application scenario we offer the following application context:

Immediately after committing a crime the criminal jumps into his vehicle and starts the escape. A police officer observes the offender and starts a persecution. By means of the available technology in the car, including 3G communication capabilities to a web service, the officer signals his vehicle as the leader of the persecution. Using geo-location data the system running in the car continuously calculates possible future locations for the car being chased. This information is shared via web services with other police's vehicles that are around in order to signal and bring them together to the virtual barrier to be created to restrict escape's roads.

With this scenario in mind we explore two settings for applications. First, the development of a technology platform that integrates a vehicular network with the cloud hence automating the creation of the virtual barrier. Second, a statistical model based on Markov chains for the prediction of navigation routes to support police chasing. The rest of this work is structured as follows. In section two we review the application of VANETs-like technology as well as the application of artificial intelligence techniques to support intelligent transportation systems. Section three describes the implementation of the two components of the system, the route predictor and the cloud-based communication services. Section four discusses results from experiments, and section five presents the conclusions and future work.

2 Motivation

Patterson et al. [1] presents a probabilistic framework that seeks to infer and predict the mode of transportation a person uses, bus, foot or car. Position and velocity information, obtained from GPS readings, were processed by means of Bayesian filters in order to track the location and transportation mode of a person. In addition to the identification of a mode of transportation, Liao et al. [2] work aims to infer whether the location of a person matches with his/her daily goals (home/workplace), if there was an opportunistic event that modifies the person's routines (why the person missed the bus), and what would the next user movement be (if the person goes to the workplace s/he will turn left on the next corner); Rao-Blackwellised particle filters process GPS data and to build a predictive model that could help guidance systems for cognitively-impaired individuals. Simmons et al. [3] considers the context of driving routines and the information can be draw to predict route trajectories and destination intent. By training Markov models with past performance of drivers it was possible to make predictions of his/her destination with 98% of accuracy. Mao et al. in [4] equipped taxi vehicles with GPS sensors to collect information about the car's trajectories and with this data build a map of the city with points of interest (POI's). By means of the POI's map the weighted Markov model offers interesting results for driving direction prediction, for instance, to identify roads with the most flow traffic.

By looking up onto discrete road's segments the car has just been driven its near-term future path can be predicted [5], including the anticipation of what direction the driver should turn in order to prepare for the next intersection [6]. The prediction Markov model was trained with the driver's past history and on this basis offered different destinations options, for instance, when there is the situation of slow traffic or when the car needs to fill gas. The implementation of accurate travel route prediction systems is challenging because its nature dependency with, for instance, the urban architecture and its associated traffic conditions [7].

As observed there is a great interest for implementing services that track and predicts individuals' mobility. Most of the proposed works seek to build systems that could assist persons with special needs but no one, as far as our understanding, has considered a police chase context, an urgent need for support in developing countries. An extra word in this regard is that we have talked to a police department and have found together the opportunity to contribute with a system that can offer support to the efforts they do to offer social security.

3 Supporting Police Activities

As mentioned before the proposed system aims at exploring the integration of a vehicular network with the cloud technology in contexts that supports police chasing, figure 1. In this work, however, we focus our attention to the route prediction Markov model.

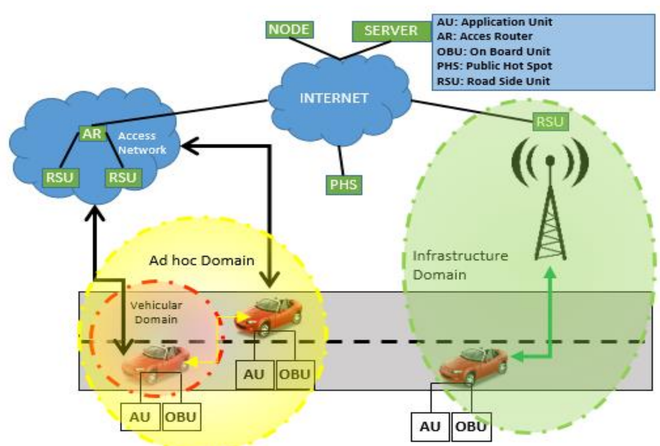


Fig. 1. General architecture for a system that support police chasing.

3.1 Route prediction

Markov chains are commonly referenced in the literature as a mathematical model that fit well for the prediction of future location for mobile targets. For the context of police chasing we are applying Markov chains of first and second order. The car being hunted moves through different adjacent intersections (as in a "chain"). The intersection that will be taken next depends only on the current state of the system, i.e. what intersection the car is coming from. In our police's chasing context the changes of intersections the car performs are called transitions. The probabilities associated with various state changes are called transition probabilities, represented in a transition matrix. Generally speaking, the transition matrix offers the conditional probability distribution space from which it is possible to predict a chain for different states given an initial state.

The probability distribution $P[X(n+1)]$ for the next intersection is independent of all, except $X(n)$, the current intersection. Then:

$$P[X(n+1)|X(n), X(n-1), \ldots, X(0)] = P[X(n+1)|X(n)]$$

Thus, on the first hand the first order Markov model allows the prediction of what intersection the car being hunted would direct to in the very near future if the current intersection the car is in or has passed through is known. The probability distribution for such a case is given by $[X(n+1)|X(n)]$. That is, the probability distribution is defined considering all of the next immediate intersections available with respect to the direction the car is being driven to. The calculation of the first order Markov's transition probabilities considered that any of the periphery nodes are available is done by means of a heuristic like the one shown in figure 2.

```
Start (calculation of probabilities first order)
i, j are indices of the nodes in the transition matrix
For each i
       For each j
              p_ij = V(i,j) / N_i
       EndFor
EndFor
End
```

Fig. 2. Pseudo-code for the calculation of the transition matrix for a first order Markov model.

Where p_{ij} is the transition probability from node i to node j, $V(i,j)$ is the number of outgoing paths from node i to node j and N_i is the number is the total number of outgoing lanes of node $\boldsymbol{i}$**.** Verifying that $\sum_{j \in E} p_{ij} = 1 \quad \forall\, i \in E$, i.e., sums per row is 1.

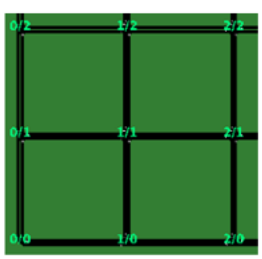
```
Start (Calculation of probabilities second order)
i, j, k are indices of the nodes in the transition matrix
For each i
      For each j
            For each k
                  If i ≠ k Then
                        p{k|j,i} = V_jk / (S_j − Q_ij)
                  EndIf
            EndFor
      EndFor
EndFor
End
```

Fig. 3. Pseudo-code for the calculation of the transition matrix for a second order Markov model.

On the second hand, the second order Markov model considers the last but one intersection the car was seen, $P[X(n+1)|X(n), X(n-1)]$. Using the second order allows the consideration of a couple of nodes, hence a street segment, during the calculation of the transitions matrix. Figure 3 presents the heuristic can help for the calculation of the transition probabilities. There we must consider that $p_{\{k|j,\,i\}}$ is the conditional probability of going to the node k given nodes j, i, V_{jk} is the number of outbound lanes from node j to node k, S_j is the total outgoing lanes of node j, Q_{ij} is the number of lanes connecting node i to node j. Verifying that $\sum_{k\in E|k\neq i} p_{\{k|i,j\}} = 1 \ \ \forall\, i, j \in E$, i.e., sums per row is 1.

Considering an urban area of nine intersections for which the transition matrix is the one given by figure 4, we can traverse through the following intersections tracing the following chain. Starting from intersection 0/0 the next probable transition can be either intersection 0/1 or intersection 1/0, because both intersections have the same weight. If it moves to the intersection 0/1 the car can go to either the intersection 0/2 or the intersection 1/1, or even return to intersection 0/0. In general, if the current state (intersection) for a mobile object is known then we can seek up into the transition matrix to know the potentially near future transition for that object.

map segment



transition matrix

	0/0	0/1	0/2	1/0	1/1	1/2	2/0	2/1	2/2
0/0	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0/1	0.33	0.00	0.33	0.00	0.33	0.00	0.00	0.00	0.00
0/2	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00
1/0	0.33	0.00	0.00	0.00	0.33	0.00	0.33	0.00	0.00
1/1	0.00	0.25	0.00	0.25	0.00	0.25	0.00	0.25	0.00
1/2	0.00	0.00	0.33	0.00	0.33	0.00	0.00	0.00	0.33
2/0	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00
2/1	0.00	0.00	0.00	0.00	0.33	0.00	0.33	0.00	0.33
2/2	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.00

Fig. 4. The segment of a map with nine nodes and the correspondant Markov model transition matrix.

Two important elements worthy to note from the transition matrix in figure 4, are that the transitions sets to zero, and the transitions are equally weighted. Transitions with a value of zero imply that it is not possible to create an immediate preceding chain between these transitions and therefore these can be ignored by the system. For instance, a car at the intersection 0/1 cannot move to the intersection 2/1 if it has not been at intersection 1/1 first, therefore, we would say that the transition from 0/1 to 2/1 is impossible. On the other hand, intersections with same weight implies that in this context a car can transit on any of the probably roads at the same speed and without any trouble. Thus, in order for the Markov model to predict escape routes it has to seek up into the transition matrix and based on intersection weights identify the probable routes of trajectories.

The previous scenario considers a city with roads that may have not traffic congestion, without accidents or where road maintenance is not needed. If on the contrary we live in a city with such kinds of events there is necessary to count on a route predictor that listens for these changes, update the transition matrix accordingly, and to count on efficient search algorithms. Once opportunistic events are detected the predictor must first reassign the adequate weights to the lanes ahead. To do that, the first task is the identification of the available roads that are allowed for vehicle traffic, which is done with the heuristic presented in figure 5:

```
Start (change event)
i,j  Are indices of the nodes in the transition matrix
      get s_node and d_node from loc
      For each i
            For each j
                  If s_node == i and d_node == j Then
                        S_i = S_i - R_i
                  EndIf
            EndFor
      EndFor
      Remove transition corresponding  s_node and d_node
End
```

Fig. 5. Pseudo-code that helps with the calculation of the new roads' weights given the fact that an accident is blocking any road.

For the pseudo-code presented in figure 5, consider that loc is the is the location where an accident occurred event, s_{node} and d_{node} correspond to the source node and the destination node on the location respectively, S_i is the total outgoing lanes of node i and R_i is the number of affected lanes node i to d_{node} .

Once the update of roads' weights is done, the re-calculation of the probabilities space can be done with either the pseudo-code presented in figure 2, first order Markov model, or the one presented in figure 3, second order Markov model.

4 Results

As described in previous sections the main component of the proposed system to support police chasing is the route predictor algorithm. In order to test the first and second order Markov chains implemented for route prediction (RP) four simulations were conducted: two RP_L (route prediction with location data) and two RP_LRO (route prediction with location data and random obstacles). Before dealing with the simulation results we first describe the setting up for these experiments.

4.1 Experiment set up

The first component required for testing the route predictor is a geo-referenced map. A map of a city in the United States was downloaded from www.openstreetmap.org. The map was manually configured, for instance, to not allow for U-turns and to remove traffic lights. These considerations are based on the run on escape context in which, we assume, a theft would not respect traffic lights and would never drive in the opposite direction over the same lane is being hunted. In addition, street categories were also properly configured, for instance, to differentiate the number of lanes that are part of a road and the maximum speed allowed for that road. For example, a road with category 1 has 4 lanes and an allowed speed of up to 110 km/hr. whereas a road with category 3 has one lane and an allowed speed of up to 50 km/hr. The lowest the category the highest the priority assigned to a car being driven on a road.

The second component that is part of the test experiment and that need some configurations is the simulation of the urban mobility platform, SUMO [8]. This tool can be configured with the number of cars that could travel around in the map, their direction, and their navigation speed among others. The car being hunted in our experiment, for instance, has an acceleration value of 2.9 m/s^2 and a deceleration value of 7.5 m/s^2, and the maximum speed it can reach is 250 km/hr. These parameters are important because these have a great impact with the decisions made by the simulator to create navigation trajectories for the car on the run. Another important configuration for the simulator is the random generation of obstacles on roads. These obstacles consist on the sudden presence of broken cars that block or alter the mobility on roads, which force the simulator to look up for alternate escape trajectories. Once the setup is done the next step is the evaluation of the route predictor system, see figure 6.

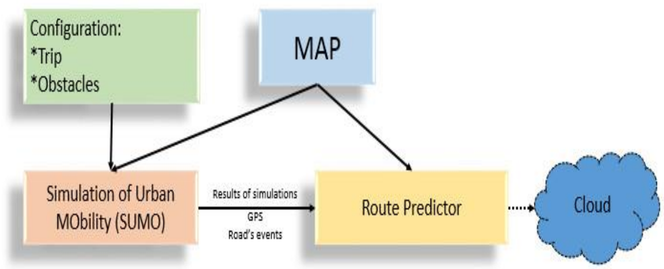


Fig. 6. Events synchronization between the simulation tool and the route predictor.

4.2 Route prediction evaluation

Four experiments were carried out in order to test the performance of the service that predicts potentially escapes routes. For the sake of space we discuss only the fourth of the experiments. For evaluation purposes only five square kilometers of the city of Denver in the US are considered, area that comprises 281 intersections and around 930 different roads, see figure 7.

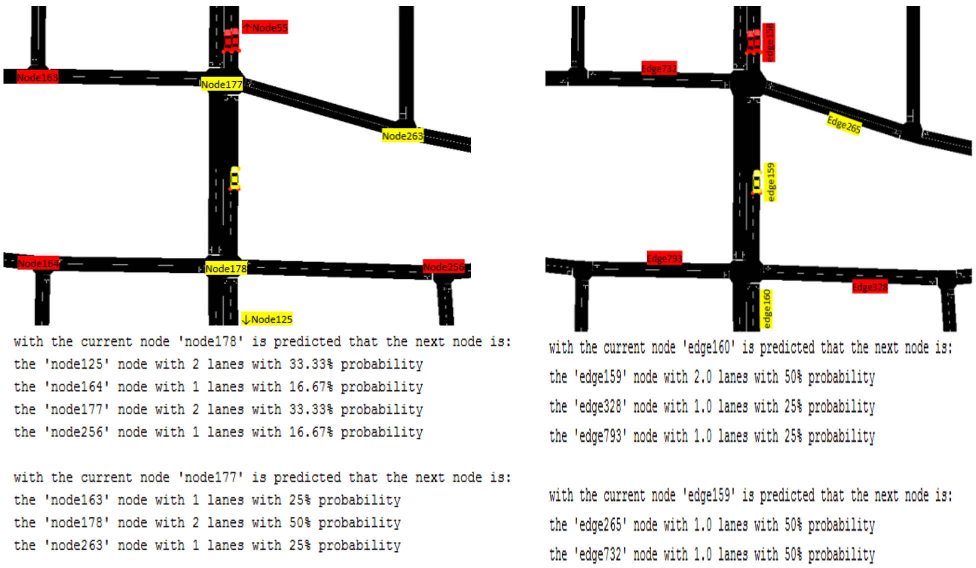
a) RP_L: Calculates the probabilities for a car to take available roads based solely on a geo-referenced map. When running the RP_L experiment with a first order Markov chain the prediction route takes into account the GPS location of the car and the roads' priorities to seek up into the transition matrix and to predict the next road a car will probably take. Basically, the probability weight for every intersection at the vicinity of the lane used by the mobile node is 33.33%.

For the second experiment we considered the same road and city map configurations but applying a second order Markov model. The first thing noteworthy is the fact that for a second order Markov the state of previous transition is important, it is not possible for a car to return to the same road on which it is escaping, as it is the number of lanes offered by the routes ahead of the mobile node. The predictor offers three nodes for continuing escaping. The road with two lanes was assigned 50% of probability whereas the other two roads got 25% of the probability space each. The prediction hence is that the car would use the two lanes road for escaping.

Fig. 7. The geo-referenced map used for the route prediction model corresponds to five square kilometers of a US city.

b) RP_LRO: With a first order Markov the model calculates the probabilities for a car to take available escape routes based on a geo-referenced map and random presence of accidents on roads. The simulation tool is in charge of inserting the obstacles into the city map. An obstacle on the road means that it is not possible for a vehicle to get through it. Remember that the simulation tool reports the location of the car and any event occurring on a particular road. Based on that information the predictor must update the transition probabilities for the city map. From the figure 8 a), it is possible to identify that when the car reaches the node 178 there are four probable routes but the ones with the higher probability are the way out to node 125 (33%) or to node 177 (33%). However when the car is moving towards intersection 177 the predictor realizes there is a car's accident on node 55, therefore, indicating that there are three probable routes to continue the travel being the node 178 (50%) the best road for escaping.

The fourth experiment consisted on repeating the third experiment (RO_LRO) but using a second order Markov model. If there is not an obstacle the predictor offers three probable roads for continue driving. When the car travels on road 160 (labeled as edge 160), for instance, the predictor outcomes three probable routes, road 159, road 328 and road 793. However, when the car is being driven on road 159 (labeled as edge 160) the predictor offer only two probable routes because it is taken into account that the road 158 is blocked. Observe that because in this case both routes (road 265 and road 732) have the same number of lanes and the allowed travel speed is also the same, the two of routes got the same probability of 50%, see figure 8 b).



(a) (b)

Fig. 8. First (a) and second (b) order Markov route prediction with the random presence of obstacles. The Markov simulation performance is graphically shown whereas the prediction outcomes are in text format.

5 Conclusions and Future Work

In this paper we have presented the application of Markov models for the context of police chase. The aim is to predict the probable escape routes that a criminal driving a car could use. If the leader patrol shares its location and prediction data with other patrol vehicles it might be possible to constraint the escape options by means of the creation of a virtual barrier, i.e. the strategic and coordinated colocation of the police vehicles at different roads and streets to limit or block flow traffic. A particular focus was given to demonstrate how the route prediction is done using first and second order Markov models. Two applications scenarios have been tested: streets with free traffic flow and streets with obstacles (cars' accidents). In any case it has been shown that the performance of the second order Markov model outperforms that of one of first order. A Markov model of second order takes into account previous locations, which provide a car's direction and help improve the predictor performance. Although we are already considering opportunistic events that can occur on the streets, e.g. cars' accidents, there are others that need to be included in the route prediction model, e.g. traffic lights and road maintenance, in order to build a robust system that offers support to the police activities. For future development we have considered to increase the order of the Markov model and to integrate other probabilistic techniques such as Bayesian networks to improve the updating of weights for the Markov probability space.

Acknowledgment
The authors wish to express their gratitude to reviewers for their invaluable feedback. We would also like to thank the PROMEP for funding "Platform for the Experimentation with Mobile Technologies" (PEMT) project, at the Autonomous University of Chihuahua.

References

1. D. J. Patterson, L. Liao, D. Fox and H. Kautz, "Inferring High-Level Behavior from Low-Level Sensors," in UbiComp 2003: Ubiquitous Computing, 2003.
2. L. Liao, D. J. Patterson, D. Fox and H. Kautz, "Learning and inferring transportation routines," Artificial Intelligence, vol. 171, no. 5, pp. 311-331, 2007.
3. R. Simmons, B. Browning, Y. Zhang and V. Sadekar, "Learning to Predict Driver Route and Destination Intent," in Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, 2006.
4. B. Mao, J. Cao, Z. Wu, G. Huang and J. Li, "Predicting driving direction with weighted Markov model," Springer Berlin Heidelberg, vol. 7713, pp. 407-418, 2012.
5. J. Krumm, "A Markov model for driver turn prediction," in SAE World Congress & Exhibition, Detroit, MI USA, 2008.
6. J. Krumm, "Where Will They Turn: Predicting Turn Proportions At Intersections," Personal and Ubiquitous Computing, vol. 14, no. 7, pp. 591-599, 2010.
7. U. Nagaraj and N. Kadam, "Vehicular Route Prediction In City Environment Based On Statistical Models," Journal of Engineering Research and Applications, vol. 3, no. 5, pp. 701-704, 2013.
8. Simulation of Urban MObility (SUMO). Homepage. http://www.sumo.dlr.de