X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

VEHICLE DETECTION WITH PLANAR LASER USING DIFFERENT MACHINE LEARNING TECHNIQUES

Francisco de Souza Júnior, Denis Fernando Wolf
Universidade de São Paulo
{fsjunior, denis}@icmc.usp.br

Resumo – A detecção eficiente de veículos em movimento é uma tarefa crucial em comboio de veículos autônomos, também conhecida como o problema do "follow me" (do inglês, siga-me). Neste trabalho são apresentadas análises de três algoritmos de aprendizado de máquina (Redes Neurais Artificiais, Naïve Bayes e Random Forests) para identificação de veículos usando sensores lasers planares. Também é apresentada uma nova técnica de detectar veículos em uma área de leitura laser usando um classificador binário. Foram feitos testes experimentais para selecionar o classificador mais adequado baseado na métrica Area of Receiver Operating Characteristic (ROC) Curve (AUC). Finalmente, em resultados simulados é possível ver que é possível ver que é possível obter uma boa acurácia nos classificadores binários.

Palavras-chave – Aprendizado de Máquina, Naïve Bayes, Random Forests, Redes Neurais Artificiais, MultiLayer Perceptron, Detecção de Veículos.

Abstract – Efficient detection of moving vehicles is a key task for a convoy of autonomous vehicles, also known as the "follow me" problem. We present an analysis of three machine learning algorithms (Artificial Neural Networks, Naïve Bayes and Random Forests) for vehicle identification using a planar laser range data. We present a novel way to detect a vehicle in a laser reading area using one binary classifier. Experimental tests have been performed to select the more adequate classifier based on the Area of Receiver Operating Characteristic (ROC) Curve (AUC) metric. Simulated results show that is possible to obtain a good classification rate for the binary classifier.

Keywords – Machine Learning, Naïve Bayes, Random Forests, Artificial Neural Network, MultiLayer Perceptron, Vehicle Detection.

1. Introduction

A key task in the convoy of autonomous vehicles, also known as the "follow me" problem, is correctly detecting a car in a urban environment. This task needs to be performed with a high reliability to avoid any collision. Various approachs have been tried for this, many of then based on vision systems [1–6]. Unfortunately, a vision system suffers leverage of light, mist and others environmental influences, which can affect the reliability of the system. Lasers range scanners, however, suffers less from these environmental influences.

Several approaches have been proposed to detect a vehicle. Petrovskaya [7] proposed a particle filter and a bayesean network to detect vehicles and their velocity based on the motion of the vehicles. Garcia et al [8] focus in detection of moving obstacles, mainly vehicles, based on the shape of the detected obstacle. Some works present performs fusion with laser readings and vision systems [9]. Keat et al [10] uses a sick range laser raw data with bayesian programming to construct hypothesis about presence and orientation of a vehicle in a car park.

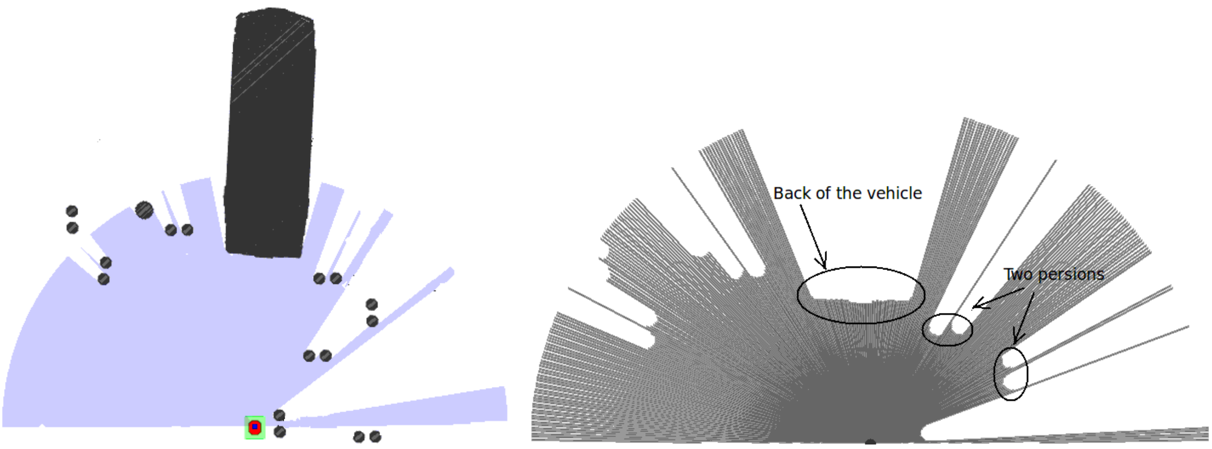
Our approach uses machine learning techniques, i.e. Artificial Neural Networks (ANN) and Random Forests, to correctly identify the presence of a vehicle in a area of a raw planar laser reading.

2 Vehicle Detection

A planar laser provides a horizontal scan of ranges of the obstacles in front of the sensor. In the Sick LMS 200 laser, a possible configuration consists of 361 laser beams in a 180^{o} with maximum range distance of 80 meters for each laser beam. In this way, a horizontal laser in front of a vehicle can detect the shape of another vehicle if it is into the maximum configured range.

In Figure 1 we can see a complete horizontal laser scanner (Figure 1b) with a back of a vehicle in the center and some other obstacles (i.e. persons) around it and the respective map (Figure 1a). The difference between the vehicle and the other obstacles is the shape and the size of them.

To delimit the area of the vehicle (or vehicles) we divide the scan area in nine areas (Figure 2). Therefore, each area can be occupied by a vehicle or not. Our approach uses only one binary classification algorithm to classify each area. This is possible by the use of the entire scan as input of the classifier and rotating the laser range scan in accordance with the block that will be classified. During the rotation, the last area is sent to the begin of the laser scan, and all the areas are shifted by one to the side (right side of the Figure 3)).



(a) Occupancy Map and laser scanning. (b) Only the laser scanning of this scene.

Figure 1: Laser planar reading with various obstacles.

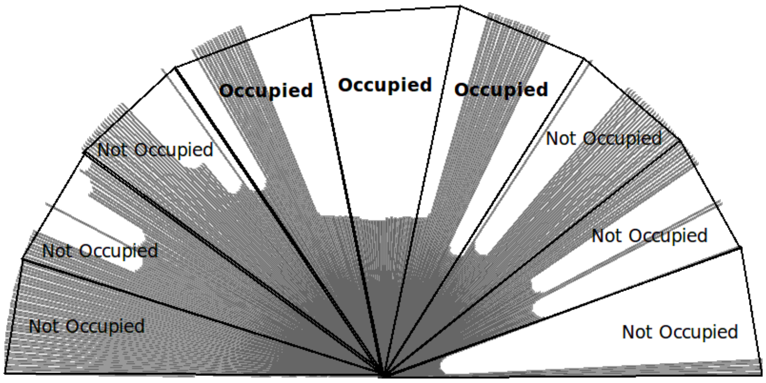


Figure 2: Division of the laser scan for classification in blocks. Occupation, here, means that the area in occupated by a vehicle.

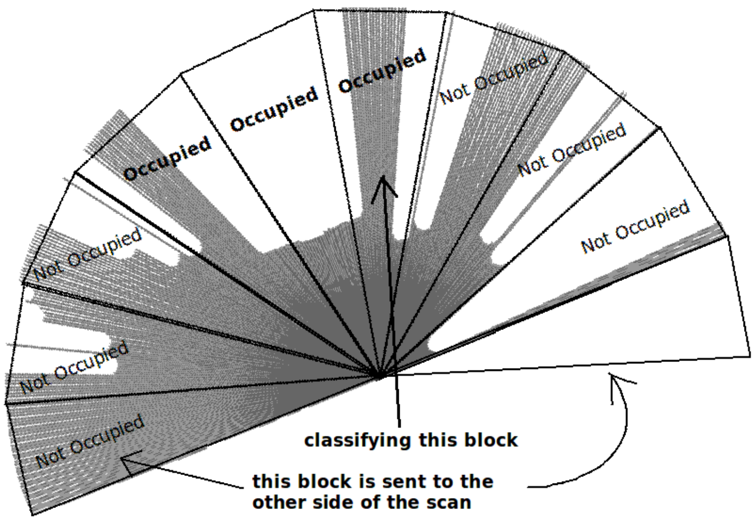


Figure 3: Rotated blocks of laser scan.

3 Methodology

Player/Stage [11] has been used to simulate the environment for data aquisition. Therefore, as we use supervised learning techniques, This data has been manually classified with a user selecting the area that contains a vehicle. Even if the area has only a small part of the vehicle, the block is classified as occupied. Three different maps have been created with eight different scenarios each one. The scenarios include parked vehicles close to others vehicles in the street. Some vehicles turning, vehicles close to trees and persons, two vehicles side by side and others.

Table 1: Datasets Sizes.

Dataset number	Training and validation	Testing
Dataset 1	911	304
Dataset 2	715	239
Dataset 3	1167	390

From these three maps, we created three different datasets, simulating the approach of the vehicle by the robot. We divided the data in a random stratified manner these datasets in 75% for training and validation and 25% for testing the final classificators (Table 1). Thus, the 75% part was used for train and validation to select the more adequate classificator, and the 25% last part was used for test the final selected classificator.

To minimize the number of inputs of the classificators and thus reduce their complexity, we divide the laser readings in sets of five, calculating the mean of each five readings. In this way, we can have 72 inputs for the classificators.

We used the Weka [12] for the classification training, validation and testing. A main problem in robotics is a high throughput of the sensor-predict-act loop. Therefore, there's a limitation of the processing requered to execute the algorithm. Thus, we discard the algorithms with intensive processing for classification, like k-nearest neighbor [13], a "lazy" algorithm. So, we choose the following algorithms to compare: MultiLayer Perceptron (MLP) with Backpropagation with Momentum [14, 15], Naïve Bayes [13] and Random Forests [16]. All these algorithms have a relatively fast execution time in the classification phase. However, as the number of neurons (MLP) or the number of trees (Reandom Forests) increases, the classification task can be slow. To deal with this, we limit the number of neurons and the trees of these algorithms.

The 10-fold cross-validation was used in the the training phase. Every experiment has been run 10 times. We then select the best one classificator based in the ranking of comparison of the CCR of instances and AUC of all the algorithms with a paired *t*-test corrected [13]. The datasets were unbalanced. The class that defines that the are is not occupied by a vehicle dominates the dataset (app. 70%), as the occupied class has only 30%. Thus, the Correctly Classified Rate (CCR) is not the more adequate metric for the classificators comparison. Henceforth, we also used the Area Under ROC Curve (AUC) metric, which is a metric for discrimination of the classifier, not affected by the distribuition of the classes in the dataset. Finally, the more adequate classificator was trained again with full training dataset and then was tested with the respective independent test dataset, to verify if there's a overfit in the model.

4 Experiments and Results

Table 2: MLP experiments parameters and topology.

	Learning rate	Momentum	Hidden l. neurons	Epochs
MLP 1	0.3	0.2	2	200
MLP 2			5	200
MLP 3			10	200
MLP 4			20	200
MLP 5			30	200
MLP 6			2	500
MLP 7			5	500
MLP 8			10	500
MLP 9			20	500
MLP 10			30	500

Table 2 shows all the parameters and topology of the MLPs evalued. The learning rate and momentum was keep the default of the Weka for all experiments, as these parameters affects mostly the the number of epochs of the training time. Only the topology (2, 5, 10, 20, 30 neurons in one hidden layer) and the training epochs (200, 500) was changed. To understand how the change of the parameters and topology affects the results, all the combinations of the parameters and topologies that has been proposed was been combinated and evaluated. In this way, if a more complex MLP has the same performance of a less complex MLP, the less complex can be chosen for the final classificator, as Occam's razor sugests.

Table 3: Naïve Bayes experiments parameters.

	Kernel estimator	Supervised discretization
Naïve Bayes 1	No	No
Naïve Bayes 2	Yes	No
Naïve Bayes 3	No	Yes

The Naïve Bayes classificator has only three parameters combinations (Table 3): using a kernel density estimator, which can be useful if normality assumption is incorrect (used in this work, but not the case of laser range data), and supervised discretization (which takes class into account when discretizing the numeric inputs) [13].

Table 4: Random Forest experiments parameters.

	Depth of trees	Features	Trees generated
Random Forest 1	Unlimited	Unlimited	10
Random Forest 2	Unlimited	Unlimited	20
Random Forest 3	Unlimited	Unlimited	30
Random Forest 4	Unlimited	Unlimited	40

Four parameter combinations of the Random Forest have been evaluated. We do not set any limit for the depth of the trees and for the number of attributes to be randonly selected for the creation of the random tree. The only parameter that has been set was the number of trees generated by the algorithm (Table 4).

Table 5: Results of the classification experiments.

Algorithm	CCR (%)			Area Under ROC Curve		
	Dataset 1	Dataset 2	Dataset 3	Dataset 1	Dataset 2	Dataset 3
MLP 1	72.77	68.76	60.77	0.70	0.74	0.64
MLP 2	73.93	69.68	60.91	0.73	0.73	0.63
MLP 3	79.43	71.49	62.83	0.83	0.76	0.66
MLP 4	82.25	74.33	66.60	0.87	0.79	0.70
MLP 5	83.46	74.53	66.75	0.88	0.79	0.70
MLP 6	74.28	69.22	60.99	0.73	0.72	0.63
MLP 7	72.75	68.90	60.44	0.69	0.74	0.63
MLP 8	79.24	71.18	62.96	0.83	0.76	0.66
MLP 9	82.45	74.25	66.94	0.87	0.79	0.70
MLP 10	83.82	74.65	66.95	0.88	0.80	0.70
Naïve Bayes 1	71.22	67.79	61.41	0.74	0.74	0.65
Naïve Bayes 2	74.76	68.01	61.52	0.74	0.76	0.65
Naïve Bayes 3	73.43	68.51	61.63	0.73	0.74	0.65
Random Forest 1	86.51	81.51	71.90	0.91	0.89	0.79
Random Forest 2	$87.76^\dagger$	83.95	74.71	0.93	0.91	0.82
Random Forest 3	$88.25^\dagger$	$84.85^\dagger$	$75.85^\dagger$	$0.93^\dagger$	$0.92^\dagger$	$0.84^\dagger$
Random Forest 4	$88.35^\dagger$	$84.80^\dagger$	$76.44^\dagger$	$0.94^\dagger$	$0.92^\dagger$	$0.84^\dagger$

Table 5 shows the CCR of instances and the AUCs of the classification algorithms. The best values are marqued with †. It's possible to see in this data that the number of training epochs for a MLP does not affects the result of the classification. However, in MLP, only the topology affect the results. As the number of neurons in the hidden layer increases, the correctly classified instances as well AUC also increases.

The Naïve Bayes classificator does not perform well in this case. It obtained one of the worse performance from all classification algorithms evaluated. The parameters change also does not influenced a variation in the results. The AUC remains almost the same, even with the the kernel density estimator (which, as already said, was expected) and with supervised discretization of the inputs.

The Random Forest, however, showed the best results. As the number of the tress increases, the algorithm increases the classification performance as well as AUC also increases.

Table 6: Ranking of the classification algorithms by AUC and the CCR, ordened by AUC.

Algorithm	CCR			AUC		
Algorithm	Sum	Wins	Loses	Sum	Wins	Loses
Random Forest 4	41	41	0	46	46	0
Random Forest 3	41	41	0	43	44	1
Random Forest 2	39	39	0	37	42	5
Random Forest 1	38	4	0	28	37	9
MLP 10	14	25	11	13	24	11
MLP 9	12	24	12	12	24	12
MLP 5	13	25	12	12	23	11
MLP 4	11	23	12	12	24	12
MLP 8	-11	7	18	-13	7	20
MLP 3	-12	7	19	-13	7	20
Naïve Bayes 2	-25	1	26	-22	0	22
Naïve Bayes 1	-27	0	27	-25	0	25
Naïve Bayes 3	-26	0	26	-26	0	26
MLP 7	-26	0	26	26	0	26
MLP 2	-26	0	26	26	0	26
MLP 6	-26	0	26	26	0	26
MLP 1	-26	0	26	26	0	26

Table 6 shows the final rank of the algorithms evaluated with t-test with 0.05 of significance. The ranking remains almost the same with AUC as a performance metric as well the percent of currently classified instances. As we can see, there's no relevant statistical difference between the algorithm Random Forest 4 (with 40 trees) and the algorithm Random Forest 3 (with 30 trees) for the CCR. Otherwise, with the AUC metric, the Random Forest 4 shows to be better. Thus, we had choice the Random Forest 4 as the better classificator for our problem.

Table 7: Performance of test dataset in the algorithm Random Tree 4.

Dataset number	Area Under ROC Curve
Dataset 1	0.93
Dataset 2	0.90
Dataset 3	0.89

In the testing dataset, the performance remains stable, as we can see in Table 7. This shows that the Random Forest 4 doest not overfit in the training datasets and still can not get the same results with a dataset that was never used to train the algorithm.

The algorithm that bests fit for these datasets is the Random Forest. However, as the classification can get a apparently a good percent of accuracy (88.35% in the very best case), to correctly classify the entire space consisting by nine blocks, this percents goes to 32.79%. That accuracy is very low, and, if the systems relies only in this classification algorithm, the consequences of a autonomous vehicle can be catastrophic.

5 Conclusions and Future Work

Identifying a vehicle in a autonomouns vehicle guidance or driver assistance is a crucial task. The main problem with a horizontal planar laser sensor data is that there's no information about texture or height of the obstacle. A vehicle needs to be detected in the details of their shape. Therefore, only the laser range raw data for a classication method is a limited way to identify a vehicle in a robot. As the individual classificator achieve a apparently good CCR and AUC, multiplying this classificator for the whole scene decreases the the CCR for a critical level.

Thus, future work includes a fusion with a camera sensor to make the car detector more robust, multipling the number of sensors of the system. Another possible approach to improve this task is to use a Kalman Filter in conjunction with a odometer sensor to track the detected vehicles as well the position and velocity of the vehicle is estimated.

6 Acknowledgments

We would like to thank the financial support of INCT/Capes and the Robotics Mobile Labs (LRM). A special thanks for MSc.. Patrick Yuri Shinzato and PhD. Fernando Osório for the crucial discussions, great suggestions and help with the experiments, and MSc. Gustavo Pessin for the great help with statistical analysis and the review of this paper.

References

[1] T. T. Son and S. Mita. "Car detection using multi-feature selection for varying poses". In *2009 IEEE Intelligent Vehicles Symposium*, pp. 507–512. IEEE, June 2009.

[2] J. Hwang, K. Huh and D. Lee. "Vision-based vehicle detection and tracking algorithm design". *International Journal of Optical Engineering*, vol. 48, no. 12, pp. 127201, 2009.

[3] J. Hwang and K. Huh. "Vehicle detection system design based on stereo vision sensors". *International Journal of Automotive Technology*, vol. 10, pp. 373–379, 2009. 10.1007/s12239-009-0043-z.

[4] A. Jazayeri, H. Cai, J. Y. Zheng, M. Tuceryan and H. Blitzer. "An intelligent video system for vehicle localization and tracking in police cars". In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC 09, p. 939–940, Honolulu, Hawaii, 2009. ACM. ACM ID: 1529486.

[5] W. Czajewski and M. Iwanowski. "Vision-Based Vehicle Speed Measurement Method". In *Computer Vision and Graphics*, edited by L. Bolc, R. Tadeusiewicz, L. J. Chmielewski and K. Wojciechowski, volume 6374, pp. 308–315. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[6] Y. Chen, B. Wu and C. Fan. "Real-time vision-based multiple vehicle detection and tracking for nighttime traffic surveillance". In *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009*, pp. 3352–3358. IEEE, October 2009.

[7] A. Petrovskaya and S. Thrun. "Efficient Techniques for Dynamic Vehicle Detection". *Springer Tracts in Advanced Robotics*, vol. 54, pp. 79–91, 2009.

[8] F. Garcia, P. Cerri, A. Broggi, J. M. Armingol and A. Escalera. "Vehicle Detection Based on Laser Radar". In *Computer Aided Systems Theory - EUROCAST 2009*, edited by R. Moreno-Díaz, F. Pichler and A. Quesada-Arencibia, volume 5717, pp. 391–397. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[9] S. Wender and K. Dietmayer. "3D vehicle detection using a laser scanner and a video camera". *IET Intelligent Transport Systems*, vol. 2, no. 2, pp. 105–112, June 2008.

[10] C. T. Keat, C. Pradalier and C. Laugier. "Vehicle detection and car park mapping using laser scanner". In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pp. 2054– 2060. IEEE, August 2005.

[11] B. Gerkey, R. T. Vaughan and A. Howard. "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems". In *Proceedings of the 11th International Conference on Advanced Robotics*, pp. 317–323, Coimbra, Portugal, June 2003.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten. "The WEKA Data Mining Software: An Update". *SIGKDD Explorations*, 2009.

[13] I. H. Witten and E. Frank. *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, third edition, 2011.

[14] F. Rosenblatt. *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Spartan Books, Washington, 1962.

[15] A. E. Bryson and Y.-C. Ho. *Applied optimal control : optimization, estimation, and control*. Halsted Press, revised edition, 1975.

[16] L. Breiman. "Random forests". *Machine Learning*, vol. 45, pp. 5–32, 2001.