IJCAT - International Journal of Computing and Technology, Volume 3, Issue 7, July 2016
ISSN : 2348 - 6090
www.IJCAT.org

Using Machine Learning to Improve Signal Timeouts

Sushant Mathur

Department of Computer Engineering, PICT,
Savitribai Phule Pune University, Company
Pune, Maharashtra 411043 , India

Abstract - A major problem with Indian traffic signals is the fact they are hard coded with the timeouts. i.e. The signal has the same timeout throughout the day regardless of the traffic condition. I have proposed two different approaches to tackle the same issue with machine learning. In the first case we predict the timeout of the signal based on previous training data, in the second we classify the amount of traffic at the time using previous training data. This would help improve traffic flow and hence increase efficiency of our roads.

Keywords **– Machine Learning.**

1. Introduction

Machine learning is the ability of computers to learn and adapt using previous experiences. The idea of the paper is to use different machine learning algorithms to improve the efficiency of the road signals which are being used today. A study has shown that India loses around Rs 60,000 crore annually due to traffic congestion.[1] This is an astronomical amount and should be reduced.

The two approaches I am taking are as follows, First is to predict what the signal timeout should be based on the traffic condition. The system will be trained on the previous data. The second approach is to classify the amount of traffic at a particular place by using various features such as time of the day and the type of area. i.e. urban or outskirts.

The rest of the paper is organized as follows, in section 2 I discuss about the various machine learning approaches taken to either predict the timeout or classify the amount of traffic. In section 3 I have shared the results of all the methods being used. In section 4 I have written about the future scope of the paper and finally have concluded in section 5.

2. Machine Learning Techniques Used

2.1 K Nearest Neighbors Classifier

In K nearest neighbors, the output is a class membership. An object is classified with the vote of its nearest neighbors, with the object being assigned to the class most common amongst its K nearest neighbors. If k=1, it is simply assigned to the class of that neighbor.[2]

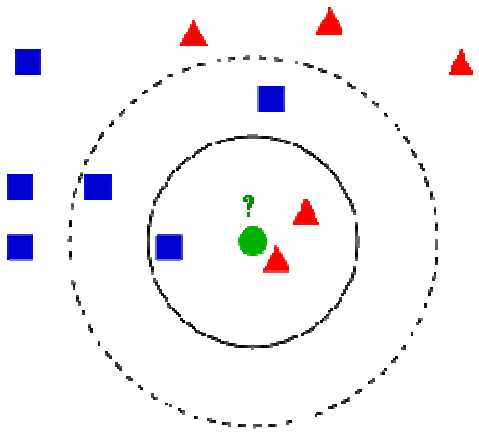

Fig1. Example of K nearest neighbors algorithm.[3]

In the above case, if we take the 3 nearest neighbors, indicated by the solid circle, we see that 2 neighbors are of type triangles and 1 is a square, hence it will be assigned to the triangle class. In case we take 5 nearest neighbors, we see that 3 are of type square and 2 are of triangle. Hence we will assign it the class square.

2.2 Decision Tree Classifier

The decision tree classifier uses decision trees as a predictive model that maps observations about an item to conclusions about the item's target value. In these tree structures leaves represent class labels and branches

represent conjunctions of features that that lead to those class labels.[4]

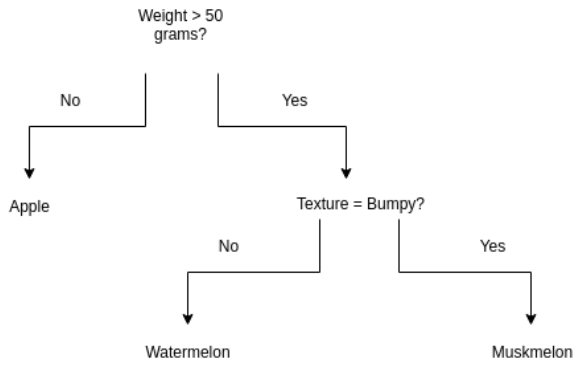

Fig2. Example of decision tree

The above decision tree is used to classify a fruit considering only 3 possible types of inputs, an apple, a watermelon and muskmelon. In the example given above, we first check the weight of the fruit, if it weighs less than 50 grams then we conclude that it is an apple. If it is more than 50 grams we check the texture of the fruit. If it is bumpy then we conclude that it is a muskmelon and if it is not then we conclude that it is a watermelon.

2.3 K Neighbors Regression

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. [5] In both cases, the input consists of the k closest training examples in the feature space. In k-NN regression the output is the average value of it's k nearest neighbors.

2.4 Decision Tree Regression

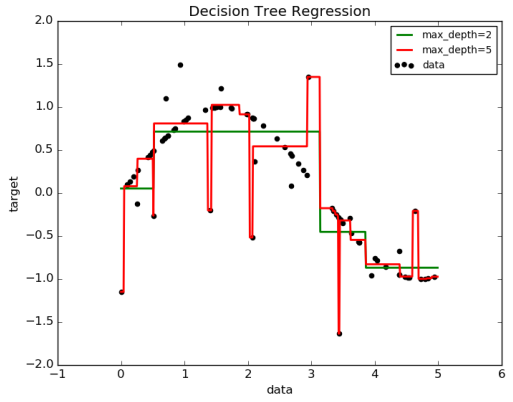

Fig3. Decision Tree Regression with different max depths.[7]

Decision Trees are a non parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value target by learning simple decision rules inferred from the data features.[6]

For instance in the example above, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and fitter the model.

3. Results

The tests were done in python using scikit learn[7] library. The features passed to the classifiers were the time of the day and the type of area. The type of area had 2 possible inputs, 0 for rural and 1 for urban. The outputs obtained could be 4. Negligible traffic(0 timeout), Light Traffic(20-30 second timeout), Medium Traffic(31-60 seconds timeout) and lastly heavy traffic(above 60 seconds timeout). The systems were trained on hypothetical data.

In case of the regressors, the input just consisted of the time of the day and the output returned would be the timeout.

3.1 Classifiers

Table1. Classifiers results

Time	Area	K-NN	Decision Tree
5:40	Urban	Negligible	Negligible
7:30	Urban	Light	Medium
20:25	Rural	Light	Light
9:00	Rural	Medium	Medium
10:30	Urban	Heavy	Heavy

3.2 Regressors

Table2. Regressors results

Time	K-NN	Decision Tree(MD=5)
10:45	61 seconds	66 seconds
18:30	74 seconds	80 seconds
22:45	51 seconds	0 seconds
6:45	19 seconds	20 seconds
9:50	55 seconds	60 seconds

MD = Max Depth

Thus we see that different methods have different results. In the case of 22:45 in regressor output, we see there is a huge difference in output as K-NN predicts 51 seconds and the Decision Tree is predicting a timeout of 0 seconds. When compared to the training data the output of 0 seconds would seem more appropriate in the situation as there is barely any traffic at 22:45. As the training data was limited and hypothetic we can increase the accuracy a lot more by obtaining more training data.

5. Conclusion

Thus I have demonstrated how we can use machine learning to learn the patterns in traffic and predict the best possible timeout for a signal. This helps in improving traffic flow and also predicting the amount of traffic at a signal at a given time of day and type of area.

4. Future Scope

The future scope of this concept is immense. We can improve overall traffic flow by adjusting the signal timeouts everywhere. We can have many more parameters or in this case features that could define the timeout of one signal. We can for example consider how wide the lane is, what is the majority of the traffic on that lane etc.

Also, we can create a relation amongst all the signals in a city as one signal affects the others as well. We can thus be more aware when adjusting the signal timings as we will know what affect it shall have on the neighboring and other signals in the area.

References

[1] http://www.academia.edu/23089988/The_Econonomic_Cost_of_Traffic_Congestion_in_India

[2] L. Kozma, k nearest neighbor algorithm, Helsinki University of Technology. Available : www.lkozma.net/knn2.pdf, 2008

[3] Antti Ajanki AnAj - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2170282

[4] Rokach, Lior; Maimon, O. (2008). *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc. ISBN 978-9812771711.

[5] Altman, N.S.(1992) “An introduction to kernel and nearest neighbor nonparametric regression”. The American Statistician 46(3):175-185. DOI: 10.1080/00031305.1992.1047579

[6] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth