# VEHICLE CLASSIFICATION USING INDUCTIVE LOOPS SENSORS

**Aliaksei Halachkin**
Master Degree Programme (2), FEEC BUT
E-mail: xhalac00@stud.feec.vutbr.cz

Supervised by: Peter Honec
E-mail: honecp@feec.vutbr.cz

**Abstract**: This project is dedicated to the problem of vehicle classification using inductive loop sensors. Developed classifier is based on nearest neighbors and logistic regression models and achieves 94 % accuracy on classification scheme with 9 vehicle classes.

**Keywords**: inductive loops, nearest neighbors, logistic regression, vehicle classification

## 1 INTRODUCTION

Intelligent Transport Systems (ITS) apply information and communication technologies to the field of road transport including traffic safety, operations, maintenance and enable more effective uses of communication, increase traffic safety and reliability, decrease travel times and the impact on the environment. ITS systems measure a bunch of vehicle parameters such as velocity, weight, length, license plate number, number of axles, class of vehicle to control traffic. A various sensing technologies are used with the aim of identifying class of a vehicle. Among them may be video detectors, piezoelectric, ultrasonic and induction loop detectors. While combination of these systems achieve best classification performance, they could be used separately. Further in case of combination of classification systems, improving one of systems affects performance of a full system. This project concentrates on inductive loop detectors that belong to low cost vehicle detectors and shows how machine learning techniques can be applied for vehicle classification using signals from inductive loops.

## 2 TRAFFIC DATA COLLECTION USING INDUCTIVE LOOPS SENSORS

The inductive loop sensor is a continuous run of wire that enters and exits from the same point. The detector powers the loop causing a magnetic field in the loop area. The loop resonates at a constant frequency that the detector monitors. When a large metal object moves over the loop, it induces eddy currents in the wire loops, which decrease their inductance and increase oscillation frequency. Measured frequency changes make a certain signal that differs depending on the passed object. Loop signal is based on metal surface area and a distance between metal surfaces and the loop. Dual-loop (two loops per lane, one after another) detectors installed in the road are used to measure length and speed of the vehicle.

## 3 VEHICLE CLASSIFICATION

### 3.1 DATASET

The traffic data provided by the Camea Ltd. contained around 200 thousand vehicle records (30 GB, about 600 thousand files). A special SW was created using C# and .NET technologies to operate with the data and manually label observations. The dataset created using this SW contains around 10000 vehicles, where each vehicle is represented by its photo, data from loop sensors, data from Weigh-in-motion system and additional attributes extracted by algorithms used in company. Vehicles were dropped into 44 classes. This 44 classes were mapped to the more general 9 classes, such as car, car with trailer, van, cargo vehicle, semitrailer and others.

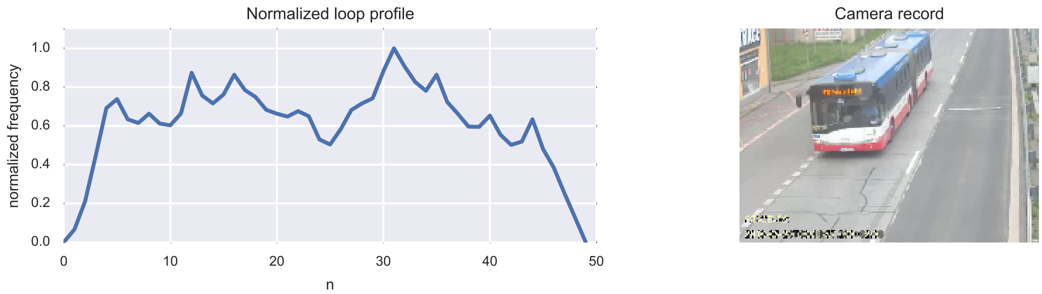


**Figure 1:** Vehicle loop profile example

### 3.2 PREPROCESSING

Most of preprocessing part was already done by the company. It includes cropping and normalizing signal, subsampling it and extracting features. Several features are extracted from the loop profile such as number of peaks, integral of the signal, sum of derivatives. Feature vector used in classifier contains 27 features (length, 20 loop profile data points and 6 extracted features). Min-Max normalization was used nearest neighbors based methods. Min-Max normalization rescales the range of features to the range [0, 1].

### 3.3 CLASSIFIERS

Two opposing model types: nonparametric (nearest-neighbors) and parametric (logistic regression) were applied to our task. Then we combined these models into single model (local logistic regression) that achieves better performance than either model individually. We used macro-average of $F_1$ measure and stratified k-fold cross validation in model assessment and selection phase as a scoring function. We preferred macro-average because of skewed class distribution.

Nearest neighbor model performance highly depends on the metric. We used euclidean distance with feature weighting as a distance function. Several derivative-free methods such as hill-climbing, simulated annealing and coordinate descent with line search along coordinates were tested to find a local optimum of feature weights $\mathbf{w}$ with respect to model performance. We did not notice significant difference between feature-weighting methods with respect to optimal value, but since coordinate-descent is the deterministic method with intuitively understandable parameters, we used it in final implementation.

Second model that had a high classification performance was logistic regression. We used L2-regularized logistic regression solver with one-vs-rest multiclass classification strategy from the LIBLINEAR [1] library. We had a great performance improving by using feature space straightening (polynomial combinations of the features). As a result, trained logistic regression achieves similar results as nearest neighbor with weighted features.

Finally we found that using using logistic regression locally has better performance than two models described above. Algorithm of local logistic regression looks as follows:

1. find k-nearest neighbors of unknown input sample. We have best results neighborhood size equals to 50 neighbors.

2. if all neighbors have same class label, assign to the unknown sample this label, otherwise continue
3. train logistic regression with k-nearest neighbors. We have also weighted samples by the inverse of their distances to the input sample.
4. use trained logistic regression to predict unknown sample class

| score \ model | Nearest Neighbor | Nearest Neighbor with Weighted Features | Logistic Regression | Local Logistic Regression |
|---|---|---|---|---|
| accuracy | 0,9140 | 0,9295 | 0,9328 | 0,9397 |
| macro F1 | 0,9076 | 0,9282 | 0,9284 | 0,9389 |
| micro F1 | 0,9140 | 0,9295 | 0,9328 | 0,9397 |
| macro precision | 0,9132 | 0,9316 | 0,9301 | 0,9354 |
| micro precision | 0,9140 | 0,9295 | 0,9328 | 0,9397 |
| macro recall | 0,9027 | 0,9251 | 0,9269 | 0,9429 |
| micro recall | 0,9140 | 0,9295 | 0,9328 | 0,9397 |

**Table 1:** Classifiers performance comparison on the dataset with 9909 observations and 9 vehicle classes. Single run of 5-fold stratified cross validation was used to estimate classification performance.

| class code | description | precision | recall | F1 | support |
|---|---|---|---|---|---|
| 1 | Motorcycles | 1 | 0,99 | 1 | 118 |
| 2 | Passenger cars | 0,97 | 0,97 | 0,97 | 1766 |
| 3 | Under 3.5 tons with trailer | 0,96 | 0,96 | 0,96 | 624 |
| 4 | Van | 0,9 | 0,88 | 0,89 | 1142 |
| 5 | 2 axle cargo vehicles | 0,9 | 0,93 | 0,91 | 2170 |
| 6 | 3 and more axle vehicles | 0,83 | 0,75 | 0,78 | 645 |
| 7 | Cargo vehicles with trailer | 0,97 | 0,97 | 0,97 | 1072 |
| 8 | Articulated truck | 0,98 | 0,99 | 0,99 | 1950 |
| 9 | Buses | 0,98 | 0,99 | 0,98 | 422 |

**Table 2:** Local logistic regression classification metrics for each of vehicle classes

## 4 CONCLUSION

The goal of our project was developing a vehicle classifier based on signal from inductive loops. Several predictive models were tested and optimized mainly using Python scikit-learn [2] library. Selected model based on combination of logistic regression and nearest neighbor classifiers is able to classify 9 vehicle classes using signal from inductive loop. Currently we are implementing Python-prototype of our classifier in C++.

## REFERENCES


[1] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9(2008), 1871-1874. Software available at `http://www.csie.ntu.edu.tw/~cjlin/liblinear`

[2] PEDREGOSA, Fabian. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*. 2011, **12**, 2825-2830. ISSN 1532-4435. Available: `http://scikit-learn.org/`