Model Selection for Machine Learning based Slip Estimation using Proprioceptive Sensors.

Sarkar, Sambuddha
Proto Innovations- Fall 2017

***Abstract*—Slip estimation on planetary rovers is an active field of research and the jury is still not out on which is the best machine learning algorithm to classify slip without visual odometry and just pro-prioceptive sensors. This report tries to answer that question by testing out four major machine learning models 1) Multi-Layer Perceptron 2)Decision Trees 3) Support Vector Machine and 4) Random Forest Trees. Through a set of procedural testing it is shown that Decision Trees perform better than the Support Vector Machines contrary to [1]'s results and observations. The software being used here are Weka and LightSide.**

I. Introduction

Calculating and correcting wheel slip in planetary rovers is crucial to any unmanned mission. Wheel slip can cause pre-planned trajectories to be violated during critical mission procedures. Slip estimation using proprioception is a new concept where-by relatively cheaper inertial sensors and existing sensors related to drive mechanisms can be utilized to infer indirectly slip instead of expensive instrumentation clusters added onto the wheel to measure slip directly. One way to do it is to use machine learning to predict slip. The primary aim of this project is to effectively estimate and classify **wheel slip** of a vehicle in a particular terrain. The wheel slip can be classified into three classes namely: **Low**, **Moderate** and **High**. It is important to classify slip instead of just predicting the slip value because the traction controller is designed according to the slip categories. The goal is to perform Supervised Classification and Detection of rover slip in different soil types and terrains.

Though in [1] they test supervised and unsupervised models, there is no proper basis in the paper substantiating that the model they selected was best and also they don't show and/or tune the models to obtain optimum performance. This report attempts to establish a way to selecting an appropriate machine learning model which can effectively classify slip and some results pertaining to that. The procedure of model selection is not an exact science and is heavily dependent on the engineer's understanding on the machine learning models and also the intuition gained from observing the data.

II. Related Work

Simple algorithms[2] are currently used on Martian Rovers to detect embedding in soft soil. The logic behind it is that they average the motor currents from the wheels over a 20 second window and then checks it against a safe threshold which is predetermined during lab test. It is currently used on the Curiosity Rover on the Martian surface and has successfully stopped the rover from embedding in soft soil when crossing the sandy ripples on surface of the planet[3]. The major limitation of this approach is that it fails when climbing steep inclines or slopes as the wheel current increases and maybe exceed the manually set threshold without an embedding event actually occurring, hence a lot of work has been done in the field of using proprioception sensor data to classify slip events on planetary rovers. Some used vibration sensor data to train classifiers[3][5], some Inertial Measurement Units (IMU) like in [8], some accelerometers to indirectly measure disturbances in the rover chassis [6][7] and some even micro phones mounted on the rover [9]. In this report the focus is on the [1] whereby they the approach based on those previous studies, where in particular, the features considered (i.e. IMU linear acceleration, IMU vertical acceleration, IMU pitch rate, and motor torque) are used for identifying various degrees of slip and are not used for classifying the terrain itself.

III. Approach

[1] has tested both supervised and unsupervised methods, but for the scope of this report the focus is on supervised classifiers used to detect slip in different terrains and not only one particular soil type. The algorithms used in [1] are namely Support Vector Machines(SVM) and Multi-Layer Perceptron (MLP) and concluded that Support Vector Machines perform the best. This report tries to reproduce their results and also introduce new algorithms such as Decision Trees and Random Forests to see how they fare against the original algorithms used in [1]. It is important to note that from now onwards SVM will be referred to as SMO (using the Weka library where it uses an SVM with John Platt's sequential minimal optimization algorithm) and Decision Trees as J48 (C4 algorithm).

The approach to tackle this problem of selecting the best algorithm is as follows. Firstly the data analysis stage is performed. It has four parts: data collection, data cleaning, data set-up and data exploration. Then once the data has been analyzed and prepared, many experiments are performed to determine the baseline model, optimize the best performing model out of the baseline model line up. The optimization procedure includes feature selection (filter/statistical methods and wrapper methods), once the good features are determined parameter tuning is performed to fine tune the model to aug-ment performance and then finally error analysis to see what are problematic features are. These processes are discussed in detail below.

IV. Data Analysis

A. Data Collection

Various physical experiments[1] were conducted using a single-wheel testbed developed by the Robotic Mobility Group (RMG) at MIT. The system limited the wheels movement primarily to its longitudinal direction.The wheel and carriage are driven at different rates to impose variable slip ratios. The wheel in use for the experimentation was a Mars Science Laboratory (MSL) flight spare wheel. The sensing system of the testbed consists of: an IMU (MicroStrain, 3DM-GX2), a torque sensor (Futek, FSH03207), and a displacement sensor (Micro-epsilon, MK88). Data was recorded at 100 [Hz] in an external computer. The soil used during testing was a Mars regolith simulant developed at MIT to replicate conditions being experienced by the MSL rover on Mars. Numerous experiments were carried out inducing wheel slip under various operation conditions (i.e., ripple geometries, wheel and pulley velocity rates) and loading conditions of the carriage pulley. These conditions included small soil ripples in the path of the wheel to create soil compaction resistance in a manner similar to what is currently being experienced on Mars by MSL. A video with this experiment is available online at: http://web.mit.edu/mobility/videos/embeddingMITPI.mp4. For ground-truth purposes, slip was estimated measuring the angular velocity of the wheel and the angular velocity of the carriage pulley.

B. Data Cleaning

The outliers were removed,from the data collection stage above, for example when slip had values out of the range [0, 100]. From the data collected four features were selected. The first feature is the absolute value of the wheel torque:

$$q_{i,1} = abs(T_i)$$

where T_i is the $i-th$ instance of motor torque. During normal outdoor driving, terrain unevenness leads to variations in wheel torque. This value is increased when the robot is experiencing moderate or high slip. The rest of features collected by the IMU sensor. These features were chosen as the variance of the N_w element groupings i of the linear acceleration (x-axis), x_{i,N_w}. The degree of pitch (y-axis), $_{i,N_w}$ and the vertical acceleration (z-axis), z_{i,N_w} like in [11]. The sliding variance is explained filter used for the IMU derived features are explained in [1].

C. Data Set-Up

The data cleaning and data collection was all done in [1]. This MIT Dataset needs to be setup. The data was split up into three distinct sets with respective percentages of original dataset:

Development Set: 20%
Cross Validation Set: 60%
Test Set: 20%.

Before splitting the dataset all the instances were randomized. The qualitative analysis is performed on the development dataset and iterative development on the cross validation dataset. The error analysis is also performed on the development data set and final testing is done by training on the cross validation set and testing on test set as unseen data. The development set is utilized to gain intuition about the data and features in a better manner. Parameters are tuned and also the error analysis performed on the development set which is later validated/tested on the cross validation set.

D. Data Exploration

It is tempting to use powerful machine learning and statistical models to find a solution to a problem, but before applying any learning techniques to solve a problem statement it is very important to understand and summarize a dataset without making any explicit assumptions about its contents. It is a crucial step to take before diving into machine learning or statistical modeling because it provides the context needed to develop an appropriate model for the problem at hand and to correctly interpret its results. To aid our understanding about the dataset it is important to use quantitative and visual inspection methods vis-a-vis do Exploratory Data Analysis (EDA). We need this stage to understand the data we are handling and what type of models will be appropriate for this type of classification problem. In our EDA we have used histograms to better see the distribution and nature of the data.

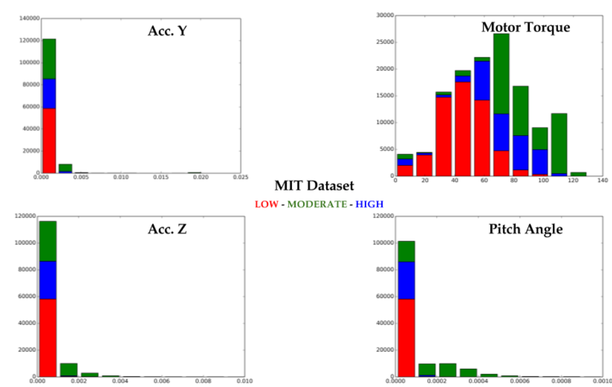

Fig. 1. Histograms of MIT dataset.

In Figure 1. we can see that except Motor Torque all the other features in the MIT dataset has a decaying exponential distribution, where-as Motor Torque has Gaussian distribution. This strongly tells us that a Gaussian based learner wouldn't be able to effectively learn the representations in the dataset.The data representations are not at all Gaussian and we can also observe major class imbalances, that is the majority of slip cases are Low Slip events. This leads us to two conclusions.

1) Gaussian based classifier wouldn't be effective in our application.
2) This might give us false sense of good performance, hence when we test the models and compare per-

formances we utilize Cohen's Kappa Statistic paired with Accuracy and Classification Accuracy in terms of percentage of correctly classified to determine a true prediction rate of the classifiers. Cohen's Kappa statistic takes into account how much agreement can be expected by chances. We use this instead of ROC or F1 score as ROC is only good when we have to check probability that a random positive example will be ranked above a random negative example and F1 score is more suited for test-mining related classifiers where we have high-dimensional data.

V. Experiments

A. Baseline Performance

An initial intuition was obtained by running six machine learning models with their default settings in Weka. The models are:

1) ZeroR
2) OneR
3) MLP
4) J48
5) Random Forest
6) SMO

The ZeroR model or the Majority predictor and OneR model were utilized to see if this problem was actually learnable and if it is actually worth to go through all the process of selecting a machine learning model. The results of the baseline performance run tabulated in Table I show that the problem is learnable and worth solving as the performance of the ZeroR model is the least i.e. just predicting the majority class is worse than then decision rule learned by other models. This how it should be ZeroR's performance should be the worst if the problem is worth solving. There is another observation that can be made from the baseline performances which reinforces the mathematical relationship between wheel torque and slip. When the OneR model is run, the minimum-error feature used is Motor Torque which is in line with our finding in Data Analysis section.

The base-line performance for the four algorithms are shown in Table I.

TABLE I
LEARNING MODEL SELECTION

Model	*Kappa Statistic Score*	*% Correct Classified*
ZeroR	0	45.5
OneR	0.6081	75
MLP	0.7144	82
J48	**0.7267**	**82.5**
Random Forest	0.6914	80.5
SMO	0.598	75

From Table I it is evident that MLP and J48 are better than SMO(in disagreement with [1] which is interesting and needs further analysis to why such a huge difference as compared to 99% classification accuracy in [1], we only get 75% even with same configuration of the model.) and Random Forest. Out of MLP and J48, J48 performs the best and hence is chosen as the selected model and will be optimized further for better performance. The Kappa Statistic all the models except for ZeroR and OneR are > 0.61 and hence substantial according the Kappa Statistic Scoring scheme (0.61 to 0.80 is substantial).

B. Optimization

In the optimization step the J48 is carried over from the results obtained in the baseline performance test and tuned. The first step is to select appropriate features for the problem at hand, so the feature selection is performed.

1) Feature Selection:

- **Filter Method**
  First, a statistical model independent approach is taken to select the best features or remove the insignificant ones. Namely two statistical scoring metrics are used: **Pearson's Scoring** and **OneR Classifier** which uses information gain inherently. The Pearson's Scoring measures the correlation between the ground truth class labels and the individual features. The OneR classifier utilizes information gain to rank the features according their influence in decision making. The attribute selection in Weka is utilized to obtain the scoring and ranking of the features. The results are shown in Table II & III.

  Pearson's Scoring (Correlation): Referring to Table II, with a threshold of 0.2, all the four features appear to be relevant to the problem at hand i.e. the features have correlation to the classes for slip prediction. It is important to note that IMU acceleration reading along the Y-axis, i.e. the sideways motion of the wheel is the least relevant whereas Motor Torque is the most relevant with the IMU pitch angle coming in a close second. This is in agreement with the mechanics of wheel slip occurrence, i.e. when wheel slip occurs, the torque profile of the motor will change and so will the pitch angle of the wheel.

TABLE II
FEATURE SELECTION USING PEARSON'S SCORING

Ranked Score	*Feature*	*Feature Label*
0.49	1	Motor Torque
0.484	3	IMU-ang-X
0.456	2	IMU-acc-Z
0.32	4	IMU-acc-Y

OneR Classifier: Same result as the Pearsons correlation coefficient , which reinforces our understanding of the problem at hand.

TABLE III
FEATURE SELECTION USING ONER CLASSIFIER

Ranked Score	*Feature*	*Feature Label*
70.5	1	Motor Torque
60	3	IMU-ang-X
56.499	2	IMU-acc-Z
56.0001	4	IMU-acc-Y

The rankings obtained in both filter methods results clearly show that Motor Torque is the best feature and the results also highlight the fact that all the features are important for decision making and none of them should be eliminated from a statistical point of view.

- **Wrapper Method**
  Second, a wrapper method is utilized where by different combinations of the features are trained and tested with the particular model, here J48, and the results are obtained as to which feature combination gives the best performance. This was done using Weka. The results show that all the four original features are well suited for this problem and none of them are problematic.

To summarize the feature selection stage, all the four features are desirable and none of them are problematic. The importance of these features is analyzed further in the error analysis stage and the possibility of new features discussed.

2) Parameter Tuning: In this stage parameter tuning is performed on the model and features that are selected from the baseline test and feature selection. J48 is the model with a Kappa Statistic of 0.726 and classification accuracy of 82.5 % with all four features is tuned. The parameter being tuned was chosen to be the minimum number of samples per leaf. Other parameters like tree depth and random splits per node were tested but none of them showed any variation in accuracy.
The default in Weka is 2 minimum samples per leaf, hence a parameter search was performed ranging from 1 minimum sample per leaf upto 10 minimum samples per leaf. The results of the tuning are shown in Fig 2 and Fig 3 where the Classification Accuracy and Kappa Statistic is plotted as a function of minimum samples per leaf. The best parameter settings seem to be with **minimum samples per leaf** set to **3** with an Classification Accuracy of 81.7% and Kappa Statistic of 0.71 after being validated using the cross-validation set and tested on testing set.

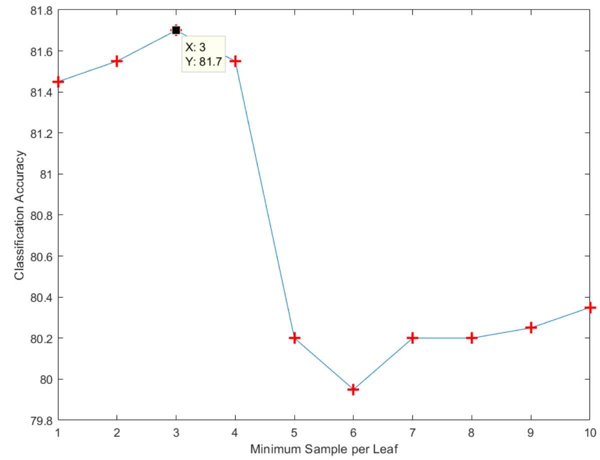

Fig. 2. Percent Correctly Classified of J48 by varying minimum samples per leaf.

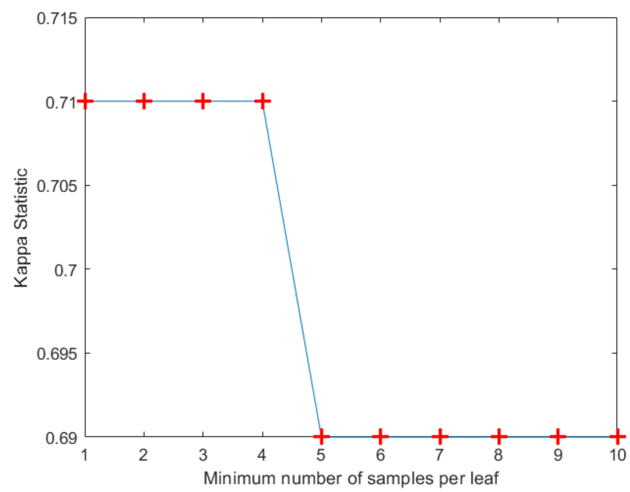

Fig. 3. Kappa Statistic of J48 by varying minimum samples per leaf.

3) Error Analysis: Error analysis is utilized to find out fundamental faults with the features so that they can be rectified. Error analysis in conjunction with feature selection provide an deterministic way to know for sure that the features are suitable for the problem at hand. The scope of representation of the data at hand is studied. Using LightSide, the dataset was loaded and ran using J48 algorithm via the Weka plugin. A study of the most misclassified data points was done using the metrics in LightSide namely:

- Average Cell Value
- Horizontal Absolute Difference: dissimilarity between the feature
- Vertical Absolute Difference: similarity between the features

Identifying the features with a low horizontal-absolute-difference and a high absolute vertical difference so that they can be eliminated or rectified. By visually inspecting

the confusion matrices and selecting the quadrants that had got misclassified, it was apparent that most mis-classification occurs with extremely low slip and extremely high slip. It can be inferred that this occurs because the model is trying to classify them as embedding of the wheel. Embedding of the wheel occurs when the wheel has displaced enough soil so that it has surrounded it now and cannot move any more. It is important to understand what happens during embedding to actually interpret the error analysis correctly. Just before embedding it is high slip or near 100% slip ratio i.e. the wheel keeps slipping without the rover moving and then once embedding occurs the slip value goes to very low slip or 0 as the rover is completely immobilized and dug itself into the sand. Now this is actually a very good thing from the perspective of learning capability of the model as it has the inherent ability to detect embedding of the rover wheel.
The results of the error analysis reinforce the fact that all the four features originally in the dataset are well suited for this problem. The mis-classification of very high slip and very low slip is an artifact of the model not because of the features but because of the class labels as none of the class labels account for embedding and the cases of embedding are removed by [1] during the data cleaning process. Thus this misclassification can be corrected by introducing another class for embedding and including the data points from embedding into the data set instead of deleting those instances.

VI. Results & Conclusion

It can be decisively said that Decision Trees or J48 in this case is the better than SMO, MLP and Random Forest contradicting the findings from [1]. All the features are retained and none of them are identified as problematic through feature selection and error analysis. The error analysis stage gives insight into the misclassification issue leading to a conclusion that introducing a new class for embedding and not omitting embedding events from the data sets is a way increase the model's performance instead of removing features.
The final model chosen is the baseline model with minimum samples per leaf as 2 instead of minimum samples per leaf as 3, even though that's the result parameter tuning gave us. This is because when significance test was run comparing the baseline J48 with the optimally tuned J48 the result was insignificant. Hence the baseline model of J48 with default settings in Weka is chosen as the best model for this problem. The performance of the selected J48 model on unseen data has a classification accuracy of 84% and kappa statistic of 0.73.

VII. Future Work

Some investigations related to the slip class boundaries, i.e. how the different classes of slip are split and what would be the optimum boundaries for differentiating different slip zones from a data perspective is a worthy endeavor, specially including the cases for embedding of the wheel. Also the use of online learning techniques like Reinforcement Learning and Inverse Reinforcement Learning instead of Supervised Learning based Classifiers is a desirable research direction in the field of traction control as these online models can learn new terrains efficiently in absence of prior supervised and label training data and can be integrated directly with the vehicle architecture without a high-level supervisory traction controller.

References

[1] Gonzalez, R., Byttner, S., & Iagnemma, K. (2016, September 12-14). Comparison of Machine Learning Approaches for Soil Embedding Detection of Planetary Exploratory Rovers. In 8th ISTVS Amercias Conference. Detroit, MI, USA.

[2] Iagnemma, K. (2008).[*Navigation and Hazard Avoidance for High-Speed Unmanned Ground Vehicles in Rough Terrain. doi:10.21236/ada498562*]

[3] Lakdawalla, E. (2014).[*Curiosity Update. Retrieved November 1, 2016, from www.planetary.org/blogs/emily-lakdawalla/2014/07241401-curiosity-update-sols-671-696.html*]

[4] Weiss, C., Fechner, N., Stark, M., & Zell, A. (2007, September 19-21). Comparison of Different Approaches to Vibration-based Terrain Classification. In European Conference on Mobile Robots. Freiburg, Germany.

[5] Weiss, C., Frohlich, H., & Zell, A. (2006). Vibration-based Terrain Classification using Support Vector Machines. In IEEE Int. Conf. on Intelligent Robots and Systems (IROS) (pp. 4429 4434). IEEE.

[6] Giguere, P., & Dudek, G. (2009, March). Clustering Sensor Data for Autonomous Terrain Identification using Time-Dependency. Autonomous Robots, 26 (171).

[7] Mahadhir, K., Tan, S., Low, C., Dumitrescu, R., Amin, A., & Jaffar, A. (2014). Terrain Classification for Track-driven Agricultural Robots. Procedia Technology, 15 , 775 782.

[8] Reinstein, M., Kubelka, V., & Zimmermann, K. (2013, May). Terrain Adaptive Odometry for Mobile Skid-steer Robots. In IEEE Int. Conf. on Robotics and Automation (ICRA) (pp. 4706 4711).

[9] Brooks, C., & Iagnemma, K. (2005). Vibration-based Terrain Classification for Planetary Exploration Rovers. IEEE Transactions on Robotics, 21 (6), 11851191.

[10] Libby, J., & Stentz, A. (2012, May). Using Sound to Classify Vehicle-Terrain Interactions in Outdoor Environments. In IEEE Int. Conf. on Robotics and Automation (ICRA) (pp. 3559 3566). Saint Paul, Minnesota, USA: IEEE.

[11] Iagnemma, K., & Ward, C. C. (2009). Classificationbased Wheel Slip Detection and Detector Fusion for Mobile Robots on Outdoor Terrain. Autonomous Robots, 26 (1)