

G1.4: Data science and machine learning in geodesy

Application of Machine Learning for Evaluation of GNSS Processing Protocols

Severin Rhyner¹, Denis Jordan¹, Rolf Dach²

¹Institute of Geomatics, University of Applied Sciences Northwestern Switzerland, Muttenz, Switzerland,

²Astronomical Institute of the University of Bern, Bern, Switzerland

Introduction

The Center for Orbit Determination in Europe (CODE) hosts one of the global analysis centers of the International GNSS Service (IGS). Each day, the data of about 250 GNSS stations are processed in a highly automated manner. In the context of a reprocessing campaign, 25 years of GNSS measurements are analyzed within a short timeframe and therefore a huge number of processing protocols are generated. A manual inspection of all these protocols is highly time consuming. Machine learning (ML) represents a promising approach to provide a data driven and objective evaluation of these protocols.

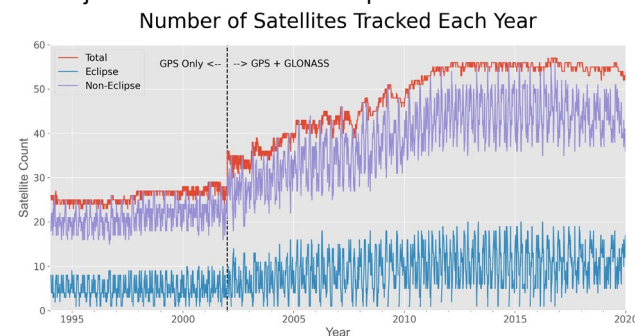


Figure 1: Cumulative number of satellites being tracked over the span of 25 years.

Problem Definition

The data used was provided by the Astronomical Institute of the University of Bern (AIUB), encompassing processing protocols between 1994 and 2019 of the continuously tracked satellites (see figure 1). The protocols contain quality parameters related to station coordinates, satellite orbits, and satellite/receiver clock corrections. We focused on one aspect of the processing protocols, namely the orbit misclosures at midnight (comparing the end of one orbital arc to the beginning of a consecutive orbital arc, see figure 2). It is known that the orbit modelling of GNSS satellites is more difficult during eclipse seasons. In order to assess the capabilities of different ML algorithms for our purpose, we have evaluated the magnitude of the orbit misclosures and have tried to classify whether the satellite was passing the Earth shadow (Umbra) or not.

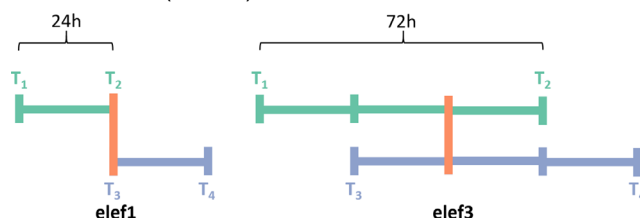


Figure 2: Computation of orbit misclosures from consecutive orbit arcs. At $T = (T_2 + T_3)/2$, marked by the vertical red lines, the differences in osculating elements, position difference, and absolute value of the 3D position and velocity difference vector are determined. 'elef1' and 'elef3' processing protocols compare orbit arcs spanning over 24h and 72h respectively.

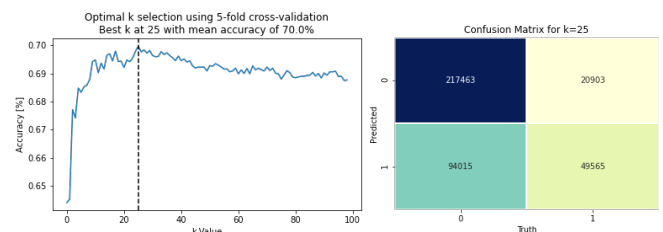
Methods

A Jupyter Notebook running in a python 3.6 environment, containing the scikit-learn, numpy and pandas libraries was used to inspect, preprocess, and apply the ML algorithms to the data. This approach was selected to allow for rapid prototyping and visual inspection of progress.

ML represents a promising approach to provide a data driven and objective evaluation of these protocols. The main objective of a ML framework is to analyze a big number of independent quality parameters in order to automatically detect individual days with problems in the data analysis.

The task at hand is a classification problem. Furthermore, only supervised learning algorithms would be used in order to learn how to handle the data. The obtained classification accuracy of the following algorithms were to be compared: logistic regression, k-nearest-neighbors (kNN), decision tree, random forest (RF), and Support Vector Machine (SVM).

The initial training of the models achieved a classification accuracy of 80% for all selected algorithms. However, due to the imbalance of the class sizes, these results were misleading. This could be observed when calculating the precision for the classification of the 'eclipse' class: the most precise model only achieving 30%. To combat this, two methods can be applied separately: upsampling or downsampling. Both methods achieve equal class balance in the training data.



Figures 3 & 4: Selection of k value for kNN and resulting confusion matrix when using optimal k .

Results

Ensuring balanced classes in the training data significantly improved the precision of the 'eclipse' class. The logistic regression model did not produce a satisfying result, as the data was not linearly separable. Furthermore, SVM struggled to cope with the dimensions of the data. Best values were yielded by kNN and RF. Optimal model parameters such as k -value (see figure 3) or node depth were determined by grid-search. The consequent accuracy and class precision were calculated from the confusion matrix (e.g., figure 4).

Model	Accuracy [%]	Precision [%]
kNN	71	68
RF	79	70

Table 1: Results obtained with balanced training classes

Conclusion & Outlook

The results obtained in our research show that ML is a powerful tool to extract information from the data at hand. The challenge to further improve the classification accuracy also speak for the quality of the processing protocols. For future research, the viability of unsupervised classification algorithms as well as neural nets in the context of deep learning need to be investigated. These methods could offer an increase in accuracy as well as the ability to detect unexpected systematics in the solutions in order to improve the GNSS analysis strategy.