Machine Learning Engineer Nanodegree

Capstone Proposal

Victor São Paulo Ruela - 01/02/2019

Proposal

VSB Power Line Fault Detection

Domain Background

Partial Discharge (PD) signals are electrical discharges that can occur inside the insulation of high voltage equipments. These signals have a repetitive nature and are confined to small regions, which in the long run can lead to irreparable damage to the equipment. Therefore, it is vital for the energy industry companies to monitor the occcurrance of PDs, in order to prevent accidents and guarantee a realiable energy transmission for its customers.

Measuring these signals in the field is already very challenging task, due to its low intensity and high noise levels from high voltage systems. However, this is just one part of the problem: based on the measurements, how can we predict some of its characteristics, such as faults, the level of damage, local of occurrence and possible causes? These are tasks that can be achieve with signal processing techniques and machine learning algorithms.

I have a special motivation to use this dataset: I have a published paper and also did my undergraduate thesis on PD denoising techniques. Therefore, I believe I can achieve pretty good results due to my domain knowledge on the subject.

Problem Statement

The goal is to train a classification algorithm to predict for PD signal measurements if a power line damaged or not. I will be tackling this problem as a binary classification problem, also applying digital signal processing techiniques to extract the most relevant features from the PD signals.

Datasets and Inputs

The dataset that is going to be used is available from the VSB Power Line Fault Detection Kaggle competition website:

https://www.kaggle.com/c/vsb-power-line-fault-detection

The dataset contains several examples of labeled PD signals. Each signal contains 800,000 measurements of a power line's voltage, taken over 20 milliseconds for each one of the three phases.

The following files are available:

a) metadata_[train/test].csv

- id_measurement: the ID code for a trio of signals recorded at the same time.
- signal_id: the foreign key for the signal data. Each signal ID is unique across both train and test, so the first ID in train is '0' but the first ID in test is '8712'.
- phase: the phase ID code within the signal trio. The phases may or may not all be impacted by a fault on the line.
- target: 0 if the power line is undamaged, 1 if there is a fault.

b) [train/test].parquet - The signal data. Each column contains one signal; 800,000 int8 measurements as exported with pyarrow.parquet version 0.11.

The entire dataset is too big (around 10GB), because of the .parquet files. Therefore, a subset will be randomly chosen, but maitaining the same class distribution from the original one. The time-series from theses files will be processed and new features will be extracted using suitable signal processing techniques.

Solution Statement

The first task will be to apply signal processing techniques in order to remove the background noise from the measurements and obtain a clear representation of the partial discharge signals. This can be done using digital filters (Butterworth, Chebyschev, etc.), the Fourier Transform and the Discrete Wavelet Transform (DWT). After that, new features will be extracted, such as amount of PDs and the frequency-domain content. If necessary, more features can be include based on the thesis from the competition's responsible:

http://dspace.vsb.cz/bitstream/handle/10084/133114/VAN431_FEI_P1807_1801V001_2018.pdf

For the training models, I pretend to compare binary classification models, such as Logistic Regression and Random Forests. I will work with simpler models in order to have more time available for the feature extraction, since this should be the most important taks on this porject. I expect to spend 70% of the time on the signal processing and feature extraction parts and 30% of the time on training models and tweaking parameters.

Benchmark Model

For this problem, the benchmark problem will be a naive predictor that will have a `X%` probability of prediciting a `true` value. `X` will be calculated as the amount of `true (fault)` samples over the total amount.

Evaluation Metrics

The results will be evaluated with the Matthews correlation coefficient (MCC) between the predicted and the observed response:

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. This is a very suitable metric, since the dataset is probably unbalanced because faults are not very frequent.

This is the same metric used in the competition.

Project Design

Before starting training the models and extracting, I will have a quick overview at the data in order to understand level of imbalance in the data and visually observe some differences in normal and fault signals. After that, I will select a smaller subset to be used in this project (for both the train and test .parquet files), and start the signal denoising and feature extraction. Next, I will perform a simple statiscal analysis and visualization of the extracted features in order to decide if they are good enough to proceed for the model training.

For the model training, I plan to evaluate at least 3 classification models. The initial candidates should be Logistic Regression, Naive Bayes and Random Forests. I will use cross-validation for the training and use scikit's grid search library to quickly evaulate several models. The final accuracy will be tested against a subset of the test set provided by Kaggle.