Olga Dmitriyeva

Description:
"Shuttle valve data has been made publicly available: http://cs.fit.edu/~pkc/nasa/data/ The goal is to design a system to classify a given time series as either normal or anomalous. The solution will be tested on data similar but not identical to that provided in the provided set". Data consists of current waveforms recorded for various forced failures. There are 12 waveforms: normal TEK00000.CSV through TK00003 (4 files); anomalous TEK00010 through TEK00017 (8 files).

Approach:
Ideally the algorithm should be able to classify the signals based on the entire shape of the waveform provided the error bars. An example of this approach is described in Ref (Listgarten, Jennifer, et al. "Bayesian detection of infrequent differences in sets of time series with shared structure." Advances in neural information processing systems. 2006). However, the signal-aligning procedure can potentially cause classification to fail if the duration of the waveform itself is a classification parameters. Thus, different approach was implemented. Each waveform supplies time-dependent and time-invariant features upon which the classification may be run. Smaller, sharper peaks represent valve opening and closing events. A broader peak represents opened valve state. Failure to the normal behavior will change either shape or the number of the peaks. Difference between normal and anomalous waveforms (variance) can be extracted by applying principal component analysis. The most variance associated with the difference between signals is linked to the peaks formation/decay behavior. Thus, PCA should be a valid approach to extract the vectors of the highest eigenvalues (and potentially speed up the analysis based on reduced number of features). The feature space then can be "reinforced" by adding the additional information on number of peaks (implemented in this study), amplitude and duration of the peaks, which are indicative of the either normal or abnormal behavior.

Implementation
Python 2.6 or higher/NumPy/SciPy/scikit-learn needs to be installed to run the script.
Two classes were implemented: Signal() to process the waveforms and ML() to analyze the data using machine learning techniques.
Original data is loaded as a 12x10000 array. Signal processing included noise filtering and differentiation to extract the information on number of peaks.
6 forst principal components were chosen to extract the most variance from the waveform data. As a result the original feature space was reduced from 1000 to 6. The number of peaks and PCA transformed data were then concatenated into one feature space resulting in 12x7 array. Data was scaled to zero mean.

Machine Learning techniques were applied in both supervised and unsupervised manner. For supervised binary classification list called TARGET was created. The labels: 1 – normal, 0 – anomalous. Data was split into two parts 75% for training and 25% for testing. Several algorithms were applied (SVM- rbf and linear kernel, kNN, DecisionTree, LDA). Ensemble methods showed the best performance (Gradient Boosting and Random Forest). Another anticipating benefit from using the ensemble method is the overfitting suppression. The accuracy of prediction by Gradient Boosting on 3 unseen waveforms (25% reserved for testing) is 100%. Based on a limited amount of data I ran Leave-One-Out cross-validation technique using Gradient Boosting algorithm as a classifier. The mean accuracy over 12 evaluated waveforms was 0.67+/-0.142, which may be interpreted as a low limit of the expected performance. Unsupervised clustering using affinity propagation method confirmed 2 clusters.

Execution
Run script: waveform_classification.py. Data files for training the classifier (12 waveforms provided) should be stored in CSV format in a separate folder. The path to this folder needs to be assigned to the global variable MYPATH_TRAIN (default files_train). The files that contain the waveforms to evaluate the performance of the classifyer need to be stored inside yet another folder and the path assigned to MYPATH_PREDICT (default files_predict). The script will generate the text file Predicted_results.txt. The file will contain the list of the file names and the labels assigned by the classifier (1-normal, 0-anomalous). By default the classification and prediction is run on the same initially provided 12 waveforms.