Kernel Based Learning Methods - Mid Term Project

Authors:
Alexander Parunov,
Pau Rodríguez Esmerats

Date:
December 11, 2018

1 Introduction

In this project we are going to apply Kernel methods to a data analysis problem. Our work will be based on a research paper that builds a prediction model for the bit error rate(BER) in optical fiber communications in the event of a physical disturbance of a single fiber. We aim to build a new prediction model, that uses kernel methods to predict the value of the error rate.

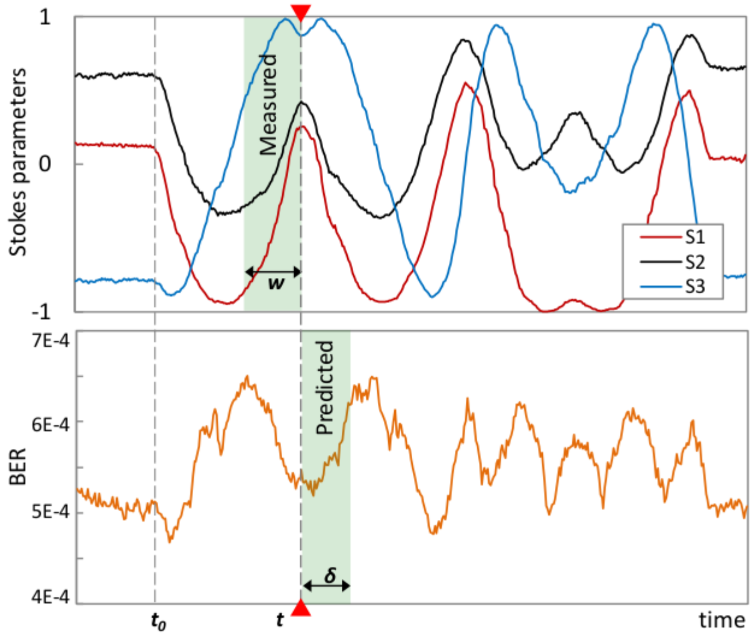

Figure 1: State of polarization (SOP) and Bit Error Rate (BER) with recording and detection windows (w and d)

The data set consists of the measurement of 3 variables of an optical fiber communication channel which are called State of Polarization (SOP), and the bit error rate measurement (BER). Those variables are gathered during periods of 4 seconds, during which a robotic arm that holds the optic fiber performs a physical action like rotation and or translation. The values are stored at a rate of 4000 values per second (1 value every 0.25ms) and there are a total of 360 experiments that last 4 seconds each. The SOP variables are constrained to form a vector that lives in the unit sphere.

2 Previous Work

In the previous work, the researchers built a prediction model of the error rate based on 6 variables gathered in time windows of 10ms. Those variables consisted in :

- the final value in the time window (at time t) of the 3 SOP variables
- the trends of each SOP within the time window [t-w,t]

The prediction goal was the BER value at δ ms after the last instance of the time windows. The values of the time window and the δ tested where between 10 and 300ms. The variables and the BER where then discretized into p and q segments of equal length respectively. Then a Naive Bayes model is trained. Different hyper parameters of the model where tested. Accuracies obtained ranged from 3 to 12 percent.

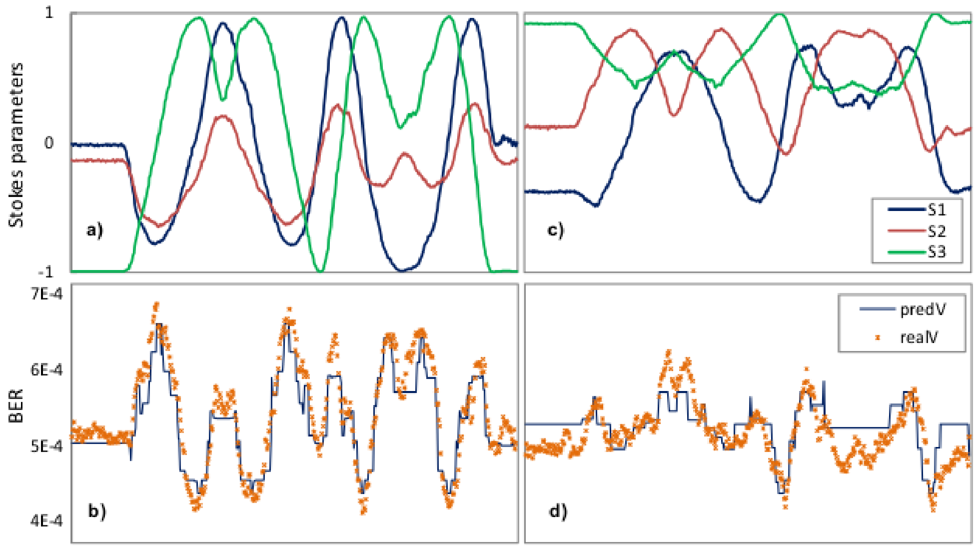

Figure 2: Prediction of the BER with a Naive Bayes algorithm

3 Predictive Model

Different alternatives are proposed for this project:

- Classification model: a similar discretization could be performed on the value of the BER. For the variables, we could then use all the values of each SOP during a specific time window, with a specific granularity (for example: one value at each ms during a time window of 10 ms). In that case, an SVM model could be trained. One of the interesting points of this approach is to choose a powerful kernel that yields good results. We could also use a BER value from $t + \delta$ for a time window [t-w,t].
- Regression model: we could avoid doing a discretization of the values and train a regression model using SVM for regression. We could also use a BER value from $t + \delta$ for a time window [t-w,t].
- Time series prediction: we will base our work on a paper by Müller, Vapnik et. al, which explains how to use support vector machines for time series prediction.

4 Preprocessing

The preprocessing will consist in the following steps:

1. Import data from multiple csv files into one data frame, selecting the columns related to the SOP and the error (BER)
2. Downsample the data from 1 value each 0.25ms to some reasonable value like 1 value each 1ms, 5ms or 10ms. We will consider also applying a moving average smoothing procedure to reduce noise, specially in the BER value.
3. Transform the dataset from 3 variables (SOP values) and one target (BER) to a dataset that contains w values from time t-w to time t for each SOP variables and for the BER, plus a value for the BER at time t+d. We will select the values of w and d according to previous work, applicability to the real case scenario where this model could be implemented and also suitability of the predictive models.

4.1 Importing data from csv files

Each csv file corresponds to a 4 second experiment. The data has been imported into a data frame, by selection the values of the x1,x2 and x3 SOP variables and the BER. The dataframe contains also an indication of the origin file. This will used when computing the lagged values of each SOP variables, since we won't be merging values from one experiment (one csv file) with values from another one.

4.2 Downsampling and Moving average

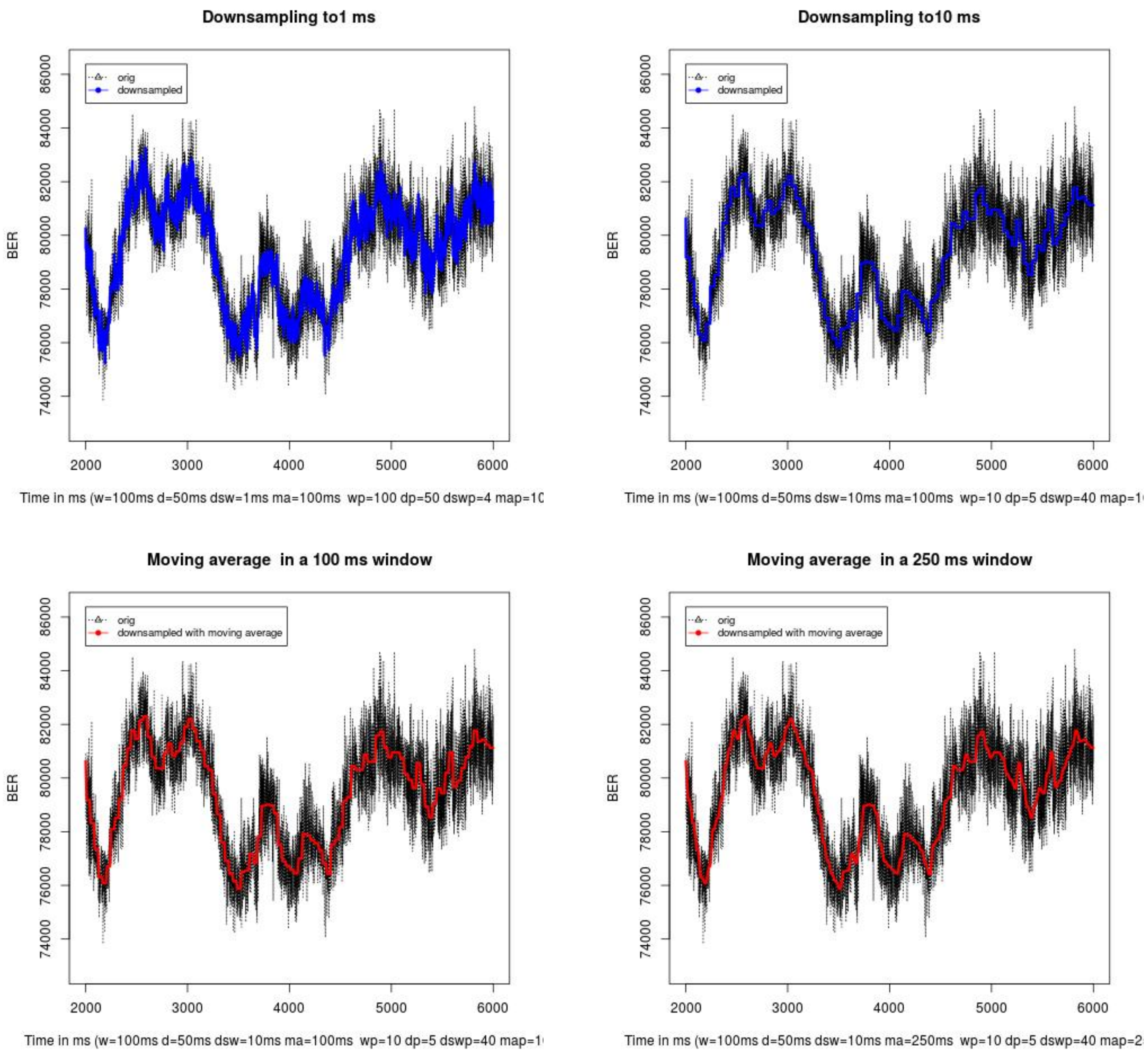

Figure 3: Ploting different downsampling rates and moving averages combined

We see that the downsampling to 10ms rate removes the a good quantity of noise in the target variable but even if it maintains the major trend and eliminates small noise it also erases some important fluctuation we want to capture.

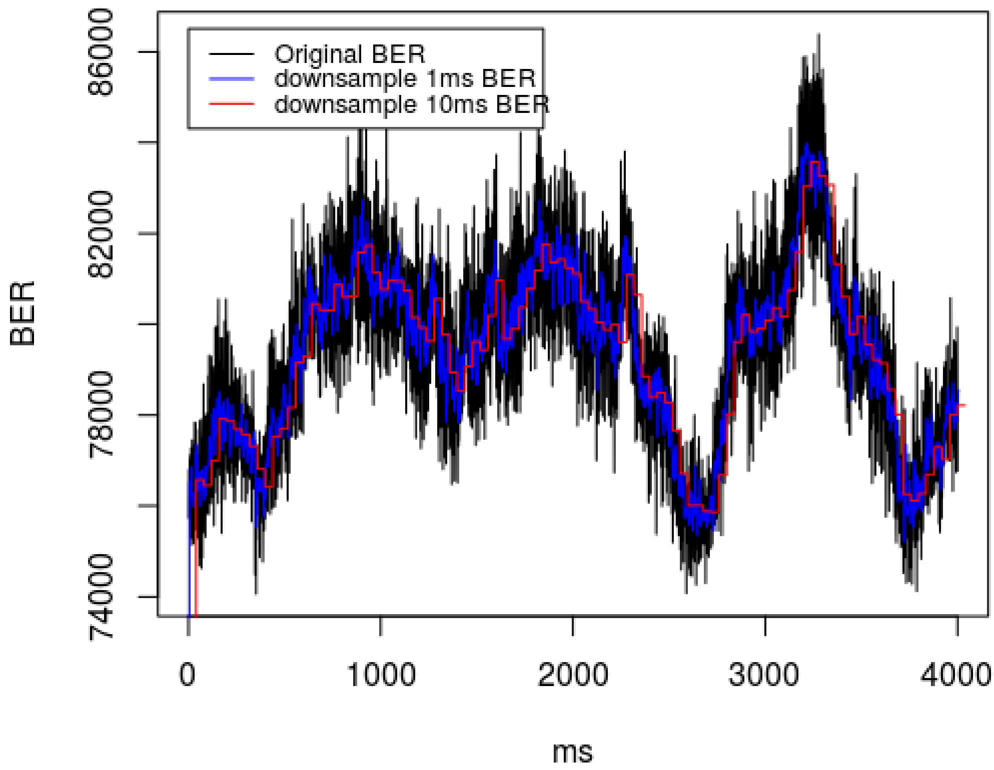

Figure 4: Different downsampling applied to the BER signal

A second plot with the values of the SOP variables shows that a downsampling to 1ms rate is the best choice.

4.3 Transforming the dataset

We will transform the data set into a sequence of time series values for each SOP variable and the BER value in a time window from t-w to t. We will also add the future BER value in an instance t+d. The selected values that we will test the models on will be:

- w=50,100,200,500 and 1000ms
- d=50,100,200ms

5 Predictive Models

5.1 Model selection

We will train different models with different hyperparameters for different parameters of the prediction (w,d). We will perform model selection by cross-validation based on the Root Mean Squared Error (RMSE) averaged over the different folds. Once the model has been selected, it is trained again with all the training data. The comparisons of different models will be performed agains this averaged RMSE. We will also try to compared the new models to the model used in the previous work, for which we will have to transform our predicted output to the discretized classes they used in this previous study.

5.2 Types of models

We will filt SVM and RVM for regression models to our data set to predict the value of the BER at a future time t+d. For such task we will train those types of models with several different kernels (Gaussian RBF, Polynomial, Laplace, String) with different values of their hyperparameters.

5.3 Training procedure

We sample the dataset into a training and testing subset, and then we will divide the training set into k subsets to be able to apply a cross-validation procedure. We limit the complexity of the model to adjust it to our computation resources (training models for more than a couple of hours is prohibitive). Since the after applying the kernel transformation, the complexity of the model becomes a function of the number of samples in the original space, we limit the number of samples that we feed to each model according to the number of features (we need n < d, with n number of samples and d number of features).

6 Results

	Algorithm	Kernel	e	C	sigma	w (ms)	d (ms)	downsampling (ms)	moving avg (ms)	RMSE
1	SVM	RBF	0.1	100	1e-07	1000	100	10	250	230.89
2	SVM	RBF	0.1	1000	1e-06	500	100	1	10	144.57
3	RVM	RBF			1e-11	1000	100	10	250	240.59
4	SVM	RBF	0.1	1000	1e-05	500	100	1	10	3674.66

Table 1: Model selection results

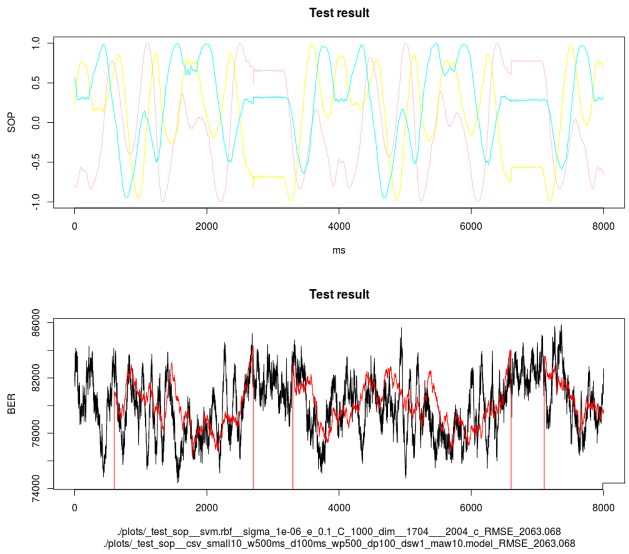

Figure 5: Model 2 prediction over the test set

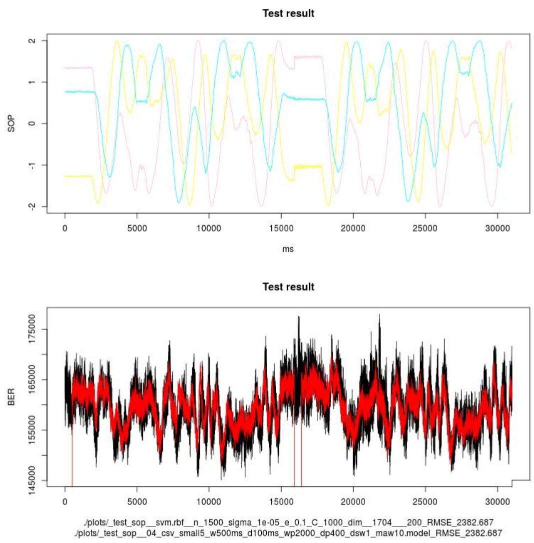

Figure 6: Model 4 prediction over the validation set

7 Conclusion

The focus of this project has been twofold, apply a kernelized machine learning algorithm to a previous work, and study the performance of different kernels for the task of time series prediction.

8 References

[1] Autonomic Transmission Through pre-FEC BER Degradation Prediction Based on SOP Monitoring, Benham Shariati et. al