Inrobin
Gareth W. Peters
Dorota Toczydłowska
Marta Campi

EMD and Gaussian Processes Project

Experiments summary

We discussed and scheduled the followings experiments in the case of synthetic data:

- **Experiment 1 - Calibration of the Hyperparameters to assess performances of the KTA**
  Select a kernel family (stationary and non-stationary). Take k kernels of that class by differentiating the hyperparameters widely enough so that we can separate different type of GP realisations. For each region, compute the correspondent Gram Matrix (Kernel Matrix). The resulting kernel matrices will be stochastically ordered, meaning that the moments of the underlying processes are ordered according to the set of selected hyperparameters. Choose one GP with the given hyperparameters and perform the KTA between the covariance matrix of that GP and the computed gram matrices. Select the Kernel with the highest alignment and consider that region for the hyperparameters to be selected for the true GP. We apply this procedure for several kernels by simulating 1000 time-series to check if the behaviour is consistent across different regions of the hyperparameters.

- **Experiment 2 - Unknown parameters - Frequency bands location**
  Select one kernel family and take k elements from it corresponding to different hyperparameters sets (generated in Experiment 1). Take each set, aggregate the k elements together. Repeat this experiment for 1000 time-series. The procedure goes for each kernel considered in Experiment 1. That aggregation provides a synthetic observation set. Mathematically, it will be equivalent to simulate Gaussian Processes with mixture kernel. Once that 1000 time series are simulated, for each of them, we perform the EMD. For each IMF, take the IF and score the number of times it lies inside the partitions implied by the hyperparameters sets. Such frequency bands correspond to each kernel partition (histogram to do it). Do this for every replicate. Alternatively, we could do clustering on the frequency.

- **Experiment 3 - Model misspecification or Model Risk**
  Choose a family of kernel type A and construct the data from that family (as explained in experiment 2). Perform the decomposition of the IMFs and fit a kernel of type B to the IMFs. The goal is assessing the relative performance of that, i.e. how model misspecification impacts on that kernel choice. It is known that if a nested kernel would be

a good starting point because it will not have too much influence. For example, if the kernel used to generate the data was a polynomial one, then employing a sinusoid as the kernel will not be a too big deal because a sinusoid can approximate a polynomial (a polynomial is a subset of the sinusoid). However, if you choose a sinusoid to generate the data, and then a linear kernel is employed, then you will probably have more misspecification. We want to assess how long that can go because, in practice, the true kernel is unknown.

- **Experiment 4 - Sensitivity**
  Sensitivity corresponds to having the true set of parameters, doing the IMF decomposition, and then slightly changing the hyperparameters, and re-do the decomposition, and, afterwards, compare the two sets of IMFs. If they vary significantly, they are too sensible. We have to do this on average, not by path (or we could do it as follows: take the same sequence of input random variables to generate the multivariate-normal, change the kernel and its parameters, so the only stochasticity comes from the input process and not from the re-simulation with different parameters, and then we compare them).

- **Experiment 5 - Use long time series and sliding windows**
  Fix the kernel and the hyperparameters. Generate a very long time series (10 times the length of the original ones). Then slide a window over the time series and, for each window, repeat the above experiments (1 and 2). We then extract the hyperparameters on the sliding windows and plot the time series of the hyperparameters and check if they are constant.

- **Experiment 6 - Robust Assessment to different type of perturbation**
  Take the data of any trials simulated by the Gaussian Process and, at random times, add noise from different distributions (Cauchy for example) and redo the estimation and fitting etc.. This is a robust assessment. It is essential because, with sensor data, even with pre-process, some noise may still be there and affects our estimation process.

- **Experiment 7 - Computational cost - Table**
  We need to assess the computational cost associated with each task. We need to check how much data we can process through the packages that we want to use (the feasible limit for it). We build a table that outlines the amount of RAM and CP requirements versus length of time series and complexity of the problems (estimation, the average number of iteration) so that we can know if coding in python instead of R will be more expensive.

- **Criteria to assess the performances**
  There are multiple criteria to assess the performances of the selected models. We can determine (1)the ability to get the hyperparameters estimates correct. (2) The ability to choose the right kernel (cause you could make model selections on the kernel). (3) The capacity to accurately in-sample estimate the Gaussian Process by leaving up some samples and then using a squared error criterion. We can access this on a mean or quantile and assess forecasting in a time-series environment. Another critical measure to consider will be the integrated energy since if you integrate a GP, the integrated process will still be a GP, equivalent to the cumulation of all the samples of the Gaussian Process before that.

Then, you can summarise the discrepancy between the population law of that integrated process and the integrated sample process obtained by cumulating the samples up to that point and through a k-s distance and a Cramer-van-distance or goodness of fit test, assessing whether the sample came from that law or not or an inference procedure of this type