Division
Department

TECHNICAL NOTE

Title	Learning hidden constraints using a Stepwise Uncertainty Reduction strategy based on Gaussian Process Classifiers – additional note		
Author(s)	MENZ Morgane		
Division visa *Scientific authentication*	*LAST NAME First name*	**Project visa** *Specification compliance*	*LAST NAME First name*
Partners	*Partners*	**Contract**	*Contract*
Recipient(s)	*Division 1 : LAST NAME First name, … Division 2: LAST NAME First name, Organization : LAST NAME First name*		
Copy to			
Enclosure(s)			

Keywords	

Summary
GAEL Summary (text only)
Rich text

This note comes in addition to the preprint "Learning hidden constraints using a Stepwise Uncertainty Reduction strategy based on Gaussian Process Classifiers" made available at https://hal.archives-ouvertes.fr/hal-03848238.

We provide more detailed computational times of the criterion proposed in the preprint and also a further view of the difference between the two Gaussian Process Classifier models: the classical GPC model and the GPC with sign.

The mean computational times for 20 repetitions are assessed for the following criteria:

- ARCHISSUR with the old formulation
- ARCHISSUR
- ARCHISSUR BATCH with two points
- SMOCU

and for different integration points numbers.

The mean times obtained on two and ten-dimensional examples are given respectively in Table 1 and Table 2.

Table 1 – Mean computational time (in s) on a two-dimensional example

	ARCHISSUR with old formulation(1 point)	ARCHISSUR (1 point)	ARCHISSUR BATCH (2 points)	SMOCU (Without IS)
2000	CPU-User : 0.221 CPU-system : 0.09 Elapsed: 2.927	CPU-User : 0.0785 CPU-system : 0.003 Elapsed : 0.08	CPU-User : 0.246 CPU-system : 0.1655 Elapsed: 5.163	CPU- user + system : 0.849
4000	CPU-User : 0.355 CPU-system : 0.1125 Elapsed: 3.377	CPU-User : 0.128 CPU-system : 0.004 Elapsed : 0.1335	CPU-User : 0. 5245 CPU-system : 0. 3225 Elapsed: 6.0890	
6000	CPU-User : 0.5 CPU-system : 0.1395 Elapsed: 3.8255	CPU-User : 0.1695 CPU-system : 0.005 Elapsed : 0.1755	CPU-User: 1.0095 CPU-system : 0.5845 Elapsed: 7.47	

Table 2 - Mean computational time (in s) on a ten-dimensional example

	ARCHISSUR with old formulation (1 point)	ARCHISSUR (1 point)	ARCHISSUR BATCH(2 points)	SMOCU (Without IS)
2000	CPU-User : 0.213 CPU-system : 0.086 Elapsed: 2.917	CPU-User : 0.0990 CPU-system : 0.0015 Elapsed : 0.0995	CPU-User : 0.2855 CPU-system : 0.1915 Elapsed: 5.3785	

6000	CPU-User : 0.4585 CPU-system : 0.127 Elapsed: 3.8605	CPU-User : 0.225 CPU-system : 0.006 Elapsed : 0.2285	CPU-User: 1.221 CPU-system : 0.708 Elapsed: 8.526	
10000	CPU-User : 0. 6850 CPU-system : 0. 985 Elapsed: 5.1510	CPU-User : 0.3530 CPU-system : 0. 01 Elapsed : 0.3635	CPU-User: 2.8785 CPU-system : 1.6505 Elapsed: 12.4205	CPU- user + system 1.67

We can observe that for the same number of integration points, the computational times between dimensions 2 and 10 do not increase much. However, the increase of integration points is necessary for higher dimensions. A variance reduction technique might be used to reduce the number of samples but it was observed that it does not perform well in this context in the regression setting. Moreover, the numerical cost of the algorithm increases also due to the optimization cost in higher dimensions.

We also compared the different models: classical GPC and GPC with signs with different levels of noise on the wind turbine case presented in the preprint. Let us consider the learning points represented in Figure 1, with a visible outlier at the top (x=17.57, y=13.42, z=20)

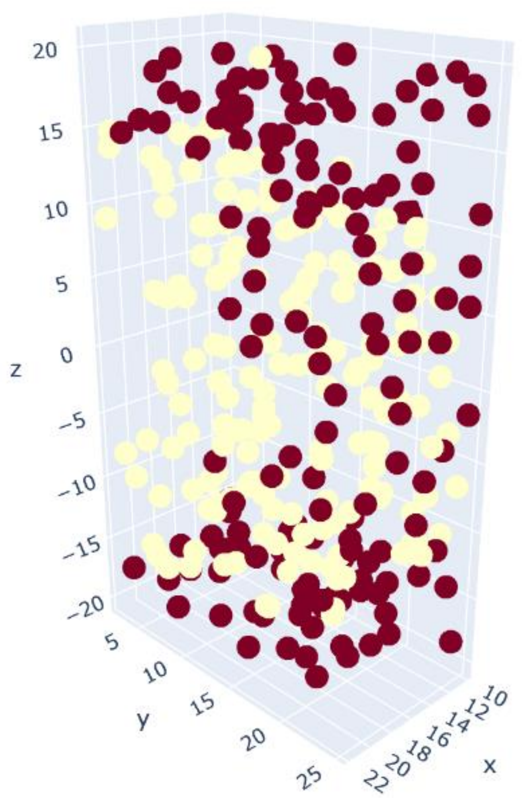

Figure 1 - Learning points (yellow dots are feasible and red dots unfeasible)

Figure 2 shows the predictions obtained with the GPy implementation of the classical GPC model with an expectation propagation approximation. It can be observed that this model completely smooths the prediction, in particular, the prediction is not impacted by the value of the outliers in his neighborhood.
This model has a sigmoid likelihood whose steepness and consequently the model sharpness is modeled by the latent function scale. In fact, label noise is modeled by this variable (small latent function scale) which is treated as an hyperparameter to be optimized in the model.

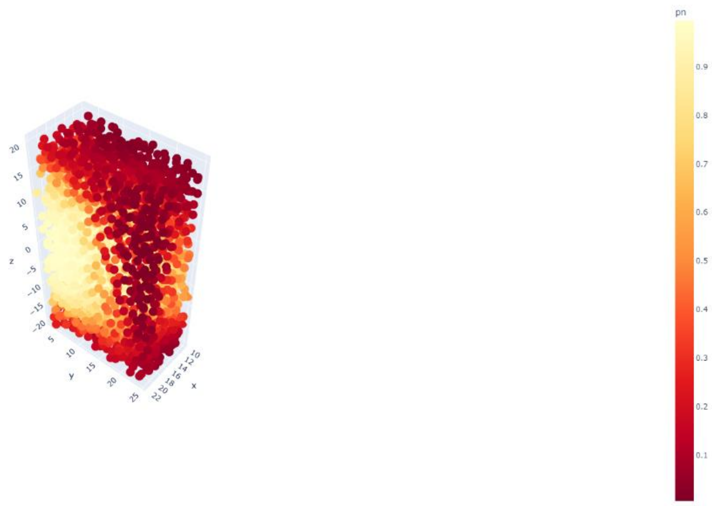

Figure 2 - Prediction with classical GPC model

Figures 3-4 represent the predictions obtained with the GPC with signs for different levels of noise variance. We can note that the impact of the outlier on the model can also be smoothened by raising the noise on the observations.

Figure 3 - GPC with signs with $\sigma_{noise}^2 = 5 \times 10^{-5}$

Figure 4 - GPC with signs with $\sigma_{noise}^2 = 10^{-6}$