

Scientific Computing Tools - Homework 01

Davide Carrara, Lorenzo Galvan, Luca Sosta

March 2022

1 Problem Request

Anti-tachycardia pacing (ATP) delivers pacing pulses to interrupt a tachyarrhythmia episode and restore normal sinus rhythm. Devices such as the implantable cardioverter-defibrillator (ICD) employ this technique before delivering high-voltage shocks to reduce the patient's sensation of pain and extend the life of the device's battery.

To ensure that ATP is effective, these devices must be programmed by selecting the proper pacing pulse width (fixed in the simulations), duration and timing. Note that it is preferred to keep the impulse limited, to reduce battery consumption. The goal of this homework is to identify an optimal ATP strategy based on a single impulse.

The dataset comprises a single bipolar trace (first row), the time discretization (second row), and an impulse shape for three different patients (third row). This process can be modeled using the monodomain model coupled with a system of ODE given by the Rogers-McCulloch model.

The numerical simulation comprises two sinus activation followed by an extra-stimulus which triggers a sustained re-entrant circuits. All the physical coefficients of the model are well defined, outside of ν_2 , that can vary between 0.01160 and 0.01240.

As solver parameters, it is recommended to use the following configurations:

- **Coarse Grid:** $N=128$, $M=64$, $\Delta t=0.01$
- **Refine Grid:** $N=256$, $M=128$, $\Delta t=0.005$

For the mathematical formulation, three different time windows can be considered:

- **measurement windows:** $(0, 450)$ ms used by the device for selecting the impulse characteristics;
- **ATP windows:** $(450, 525)$ ms where the impulse (maximum duration 10ms) is delivered;
- **tracking window:** $(600, 800)$ ms to verify the effectiveness of the strategy.

2 Problem conceptualization and mathematical formulation

2.1 Goal

Our main goal is to idealize and simulate a pacemaker that can interrupt tachyarrhythmia problems starting on a dataset of three different and unique patients.

We have at our disposal several parameters for each patient, most notably ν_2 that we need to identify in the first part of our research. After this first step, we can start simulating our pacemaker on the tachycardia signal in order to look for the best time of application and shortest time duration of the impulse we give, so that we can stop the cyclic signal.

Our complete signal is made of two healthy heartbeats and a shock that triggers the arrhythmia (analyzed in the *measurement window*) and our pacemaker impulse (given in the *ATP window*). Finally, we can check the effectiveness of our pacing protocol in the *tracking window*.

It may be useful to note that in our simulations we do not consider heartbeats after the first two, but that is just for simplicity since the heart can restore its normal beating afterwards. Still, this is not relevant in our research.

2.2 Step 0: definition of the model

We consider the Rogers-McCulloch model, that depends on certain parameters. So we can identify our data with:

$RM(T, \Delta t; \nu_{2,i})$ is the simulated ECG with the Rogers-McCulloch model,

T is the time of application of our stimulus,

Δt is the duration of the applied stimulus,

$\nu_{2,i}$ is the characteristic parameter ν_2 of the patient i .

We will identify with ν what is the actual parameter ν_2 for the sake of simplicity.

Our data is a noisy output of $RM(800, 0; \nu_{2,i,ex})$, where these $T, \Delta t$ mean we did not apply our pacemaker impulse, and the $\nu_{2,i,ex}$ are unknowns of the problem.

2.3 Step 1: estimating ν_2

Our first part of the problem is to find the characteristic parameter ν_2 of our three patients.

In order to do so, we need to minimize the following distance:

$$\nu_{2,i}^* = \underset{\nu}{\operatorname{argmin}} \|RM(800, 0; \nu_2) - RM(800, 0; \nu_{2,i,ex})\|_{L^2} \quad \text{in } (0, 450) \text{ ms}$$

in the measurement window, to be sure we have a correct representation of the original data.

2.4 Step 2: estimating time application and duration

Then, once we obtained reasonable estimates of ν_2 , we can start looking for the main objective of the problem, that is time application and duration.

In order to do so, we need to minimize:

$$(T, \Delta t) = \underset{T, \Delta t}{\operatorname{argmin}} \|RM(T, \Delta t, \nu_{2,i}^*)\|_{L^2} \quad \text{in } (600, 800) \text{ ms}$$

where we compute the norm of the curve complete with our shock that should restore the normal heartbeat. We look for our norm to be zero, since we do not consider any new heartbeats (in practice, with our simulations we want to achieve the patient's death).

Moreover, between the multiple pairs $(T, \Delta t)$ we will choose the one with smallest Δt . This choice is related to the pacemaker efficiency, since with shorter impulse we can save battery power with every stimulus.

3 Design and Implementation

3.1 Gaussian Processes Regression

The main component of our research consists of Gaussian Processes, which are extensively used in our regression models.

3.1.1 Gaussian Processes Overview

Gaussian processes can be thought as a generalization of a Gaussian random vectors. Let $I \in \mathbb{R}^n$ and $X_t, t \in I$, a collection a random field. A Gaussian process is a stochastic process where all the finite dimensional distributions are Gaussian, i.e. $X = (X_{t_1}, \dots, X_{t_n})$ is a random Gaussian vector.

A Gaussian process is fully defined by:

- **Mean Function:**

$$m_X : I \rightarrow \mathbb{R} : m_X(t) = \mathbb{E}[X_t] \quad \forall t \in I$$

- **Covariance Function:**

$$C_X : I \times I \rightarrow \mathbb{R}, C_X(t, s) = \mathbb{E}[(X_t - m_X(t))(X_s - m_X(s))] \quad \forall s, t \in I$$

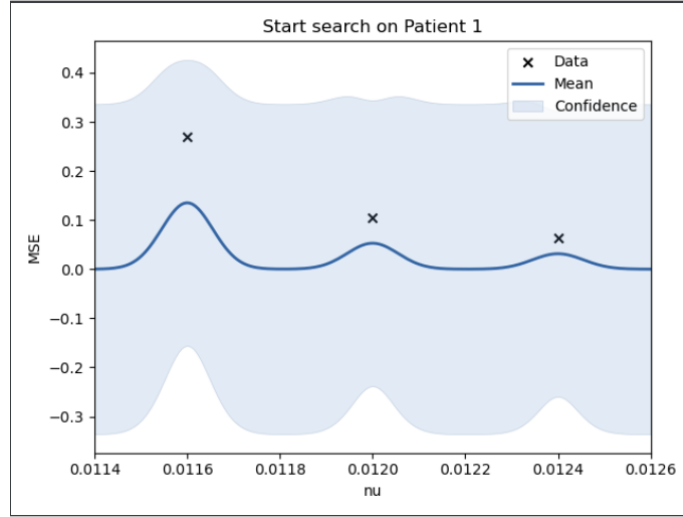


Figure 1: GP with a few data

3.1.2 Kernel Function

The kernel function or covariance function is the one describing the covariance between two random variables in the model.

In our case, we used a Gaussian-form Radial Basis Function (RBF) kernel, defined in 1 dimension as:

$$\kappa_{RBF}(\|x - y\|) = \sigma^2 \exp\left(-\frac{\|x - y\|^2}{2l^2}\right)$$

where σ^2 is the variance and l is the lengthscale.

In the following Figure 1 we can observe the behaviour of our gaussian process with few data.

3.1.3 Acquisition Function

The acquisition function is the technique that guides the search on the parameter space by using both the predicted mean and the predicted variance generated by the model. It balances the effects of *exploitation*, which makes the model focus on an area close to current results, and *exploration*, which pushes the model towards region of the domain without evaluations. In this way, the acquisition function makes the model explore both points with minimum mean and areas not visited yet.

In our specific case, we introduced a multi-step acquisition function, able to adapt to the different levels of exploration of the parameters space.

When solving the problem of estimating ν^2 , our acquisition function was:

$$M = -\mu + \beta\sqrt{\sigma^2}$$

where

μ is the vector containing the mean in each point of the domain,

β is a coefficient for the variance (set at 2),

σ^2 is the vector containing the variance in each point.

and in every iteration of the Gaussian Process we take the point with highest value as our estimate. When solving the 2D problem of estimating the best pacing protocol, we used the aforementioned loss for the first 30 iterations of the Gaussian process. The second part of the exploration was conducted using:

- After predicting the loss for all the parameters in our grid, choose the couple with the smallest duration among those belonging to the lowest 3% quantile. This is done for iteration between 30 and 50.

- Choose the couple with smallest duration among those in lowest 1% quantile, for any iteration higher than 50.

3.1.4 Offline Training

At every iteration of our algorithm we have a new estimate of our target values and we need to run the Rogers-McCulloch model with different parameters.

This is obviously computational demanding and while we cannot avoid it, we can speed up the initialization of the Gaussian Processes through a first pilot run with a large number of iteration and store all the evaluation of the model in a database.

From now on we were able to use those data to initialize the GP in an accurate way with very low effort, relying only on already computed points, and run the model only in the last part of the optimization process, using just a few iterations.

3.2 Second Level Optimization / Hyperparameter Tuning

The algorithm presented before was able to reach good results, which will be presented in the following section. However, our offline training procedure allowed us to try different approaches to improve our results without having to refit a complete training.

In particular, we tried to understand how we could change the loss function, until now corresponding to the MSE. Introducing a term correlated to the duration would of course lead the Optimizer to a different proposal, even if starting from the same dataset already computed. However, choosing the right weight for this term is fundamental, since an excessive value would lead to very small duration of the impulse, likely ineffective in stopping the re-entry circle.

$$\text{New Loss} = \|RM(T, \Delta t, \nu_{2,i}^*)\|_{L^2} + \alpha * (\Delta t/10)^2$$

This problem can be led back to a single parameter optimization, analogous to the estimation of ν_2 . The technique used is thus the same, relying on Gaussian Processes.

In particular, we use two Gaussian Processes, one in 1D, only to estimate alpha, and one in 2D, to build the pacing protocol. When the first GP proposes one value for alpha, the second one loads the dataset, and analyzes it using the new loss instead of simple MSE; a pacing procedure is then proposed, and the resulting curve is modeled. If the re-entry cycle is stopped, the loss returned to the first process is equal to the proposed duration, otherwise it corresponds to an arbitrary value, that we chose to set at the result of the previous point.

4 Results and Testing Phase

At first we tried to estimate ν_2 using the coarse grid proposed in the text. Even if useful in order to develop our algorithms with less computationally intensive models, the obtained results were significantly worse: we will thus report only the ones obtained with the more refined grid.

4.1 Search of ν_2

We evaluated 46 values of nu, obtaining the following results:

$$\text{Patient1} : \nu_2 = 0.012342 \quad (1)$$

$$\text{Patient2} : \nu_2 = 0.011912 \quad (2)$$

$$\text{Patient3} : \nu_2 = 0.011711 \quad (3)$$

These values have been found using MSE as loss: other approaches, such as MAE and MASE were tried, with comparable but slightly worse results. Moreover, the choice of using MSE is coherent with the noise applied to the datum, which is gaussian.

The curve of the gaussian processes for the three patients are reported in Figure 2.

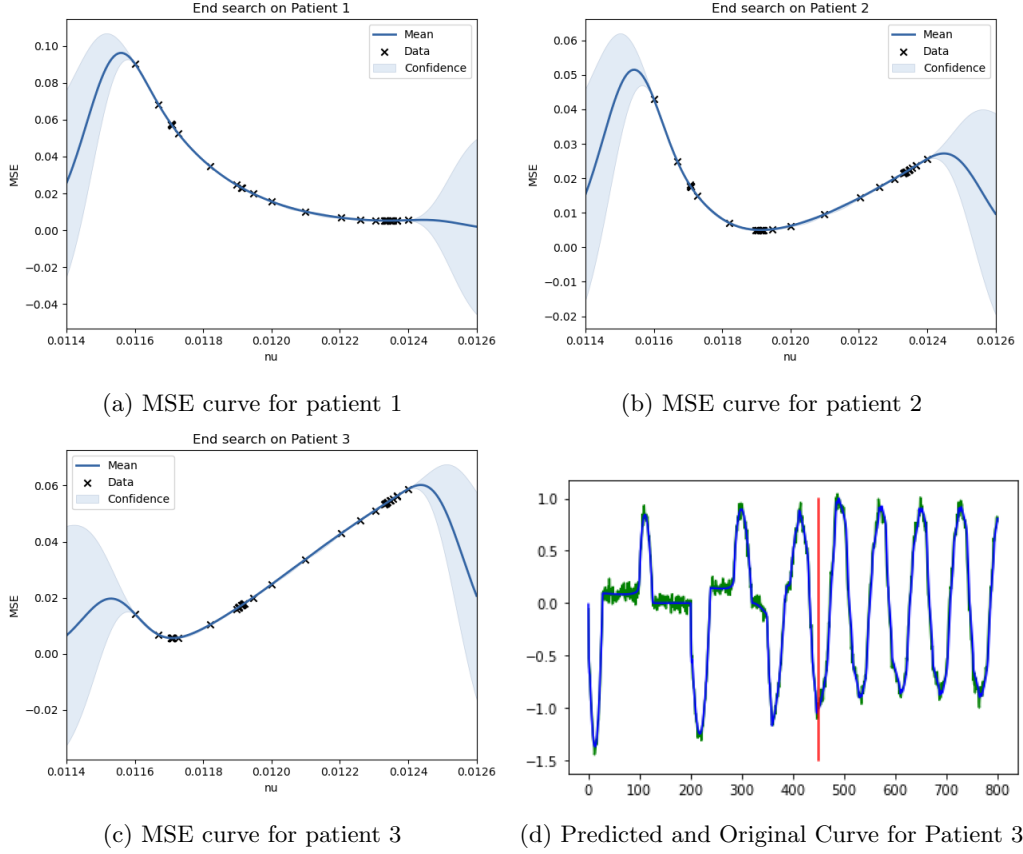


Figure 2: Estimated MSE depending on different values of ν for the three patients

As we can see, the confidence bands for all the values belonging to the parameter space are negligible: this is a sign that we could likely obtain comparable results with less iterations. Moreover, our results are not affected by noise, since we are using the complete Rogers-McCulloch model to generate the curve. Gaussian Processes are, however, more powerful instruments, and they would be able to handle also a certain quantity of noise in the loss evaluation.

However, our offline training procedure is particularly useful in this situation. Given an eventual fourth patient it would be possible to load the curves already computed and calculate the MSE with the new one. The entire procedure would be almost immediate, and the estimation of ν_2 extremely accurate.

In Figure 2d, we see the curve generated for patient 3 using our estimate ν_2 , compared with the given one.

4.2 Pacing Protocol

In order to evaluate the pacing protocol, we compute 58 samples for the first patient and 62 for the second and third. The computational time is reduced by generating the first 450 instants only once (since the input will be given with $T \geq 450$) and by storing all the computed curves and reloading them.

Using a bidimensional gaussian process we obtain the following results, with a total duration of 4.835 ms:

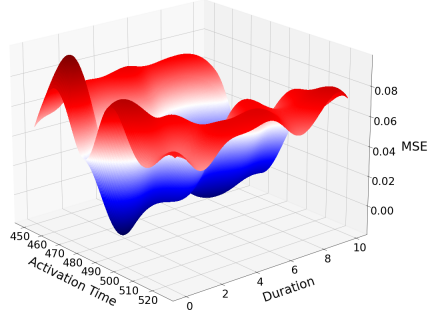
$$\text{Patient1} : T = 480.97 \quad \Delta t = 1.150 \quad (4)$$

$$\text{Patient2} : T = 493.05 \quad \Delta t = 1.285 \quad (5)$$

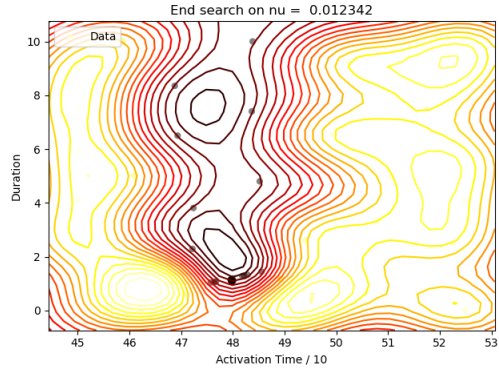
$$\text{Patient3} : T = 504.10 \quad \Delta t = 2.400 \quad (6)$$

All of the provided pacing protocols were able to stop the re-entry cycle, reaching the main goal of the project. However, the duration could still be reduced, as will be shown in the next sections. The following figures provide, for each patient, a visualization of the integral of the heartbeat on the parameter space.

MSE evaluated for $\nu = 0.012342$



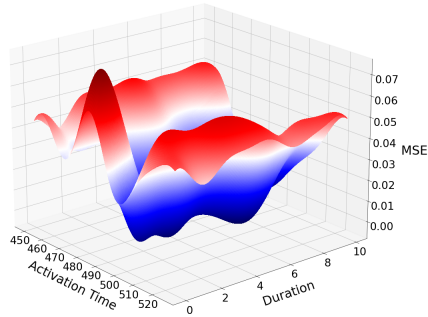
(a) 3D Plot of the loss



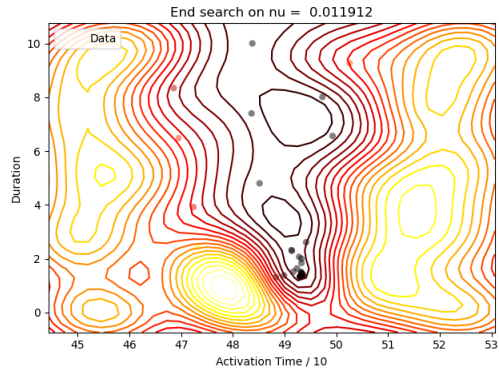
(b) Level curves for the loss

Figure 3: Patient 1

MSE evaluated for $\nu = 0.011912$



(a) 3D Plot of the loss

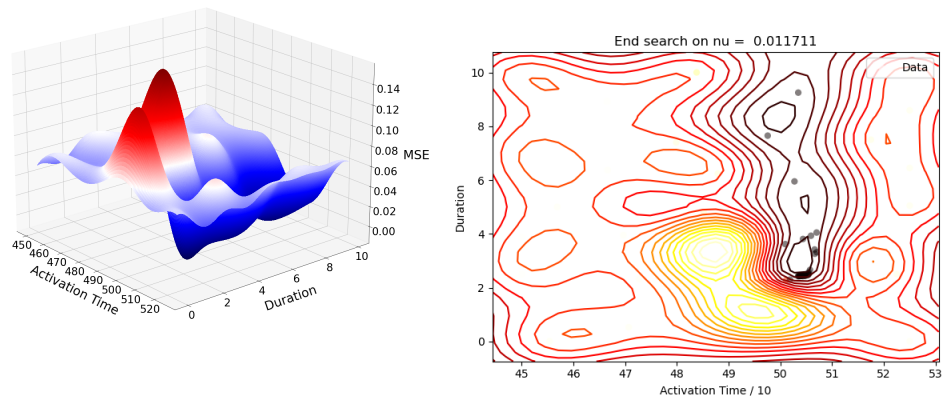


(b) Level curves for the loss

Figure 4: Patient 2

As we can see from the following Figure 6, representing the three heartbeats, Patient 3 provides a small noisy behaviour before completely stopping the re-entry. This is, however, limited in duration and thus considered acceptable.

MSE evaluated for $\nu = 0.011711$



(a) 3D Plot of the loss

(b) Level curves for the loss

Figure 5: Patient 3

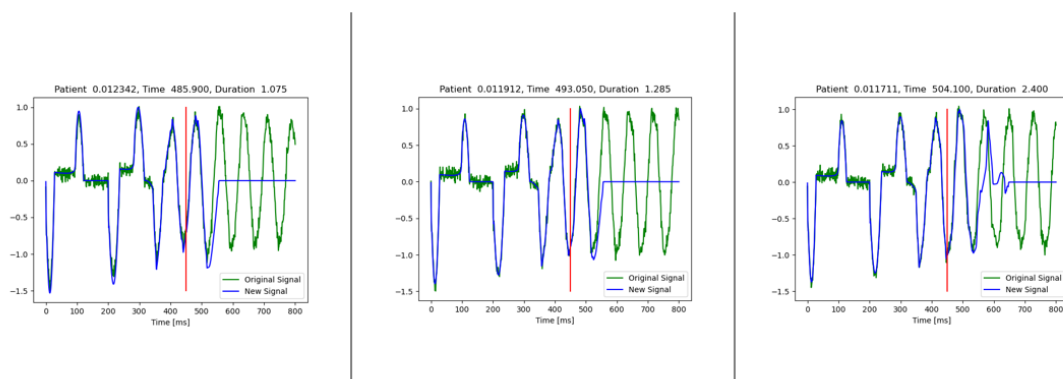


Figure 6: Curves for P1, P2, P3

4.3 Bayesian Hyper-Parameter Tuning

After performing Gaussian Process hyper-parameter tuning for alpha, we obtain the following values, which are patient specific.

$$\text{Patient1} : \alpha = 9.65970 \quad (7)$$

$$\text{Patient2} : \alpha = 10.1601 \quad (8)$$

$$\text{Patient3} : \alpha = 23.3734 \quad (9)$$

Considering this result, multiple approaches may be possible. The first one, which we chose, is to consider this new loss as an enhancements of our model, which is able to identify a better minimum starting from the same dataset. A different approach would be to train any new patient using the combined loss, considering the three patients provided simply as a training dataset. Lastly, it would also be interesting to understand if a common value of alpha can be found, in order to have a more generalized model.

The new values for time and duration show a significant improvement from the point of view of battery saving, with a total duration of 3.495 ms.

$$\text{Patient1} : T = 478.945 \quad \Delta t = 0.765 \quad (10)$$

$$\text{Patient2} : T = 493.000 \quad \Delta t = 0.805 \quad (11)$$

$$\text{Patient3} : T = 505.835 \quad \Delta t = 1.925 \quad (12)$$

In the following figures, we will provide for each patient the level curves of the new loss and the estimated function linking alpha and duration.

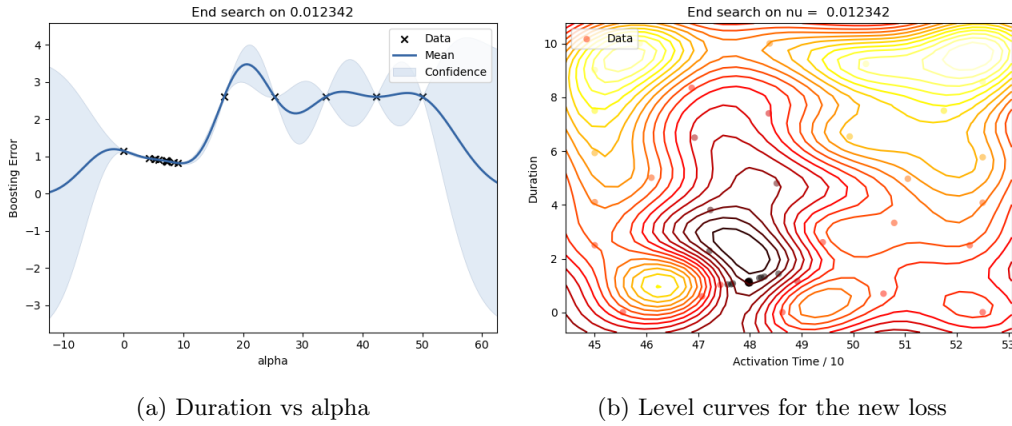
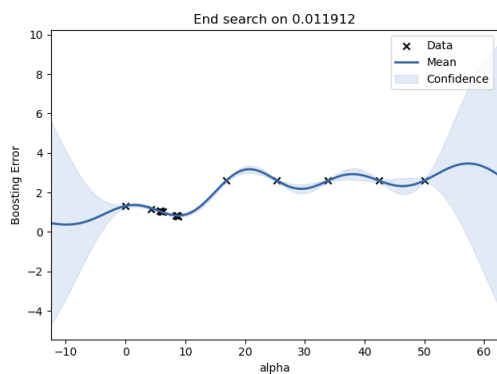
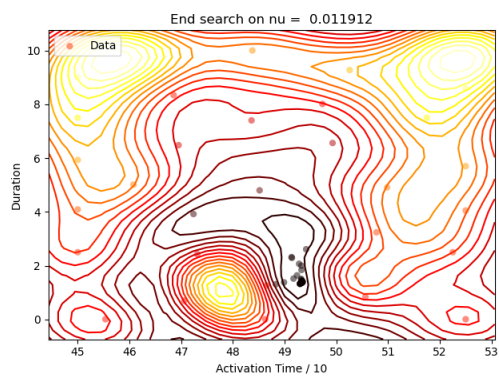


Figure 7: Patient 1

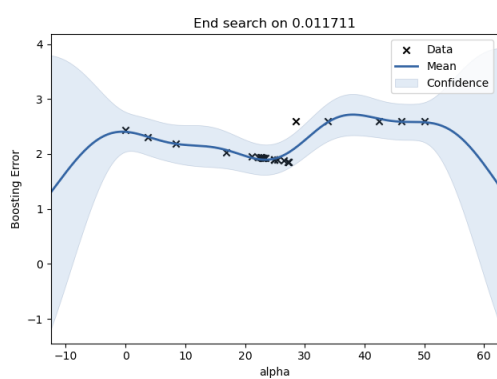


(a) Duration vs alpha

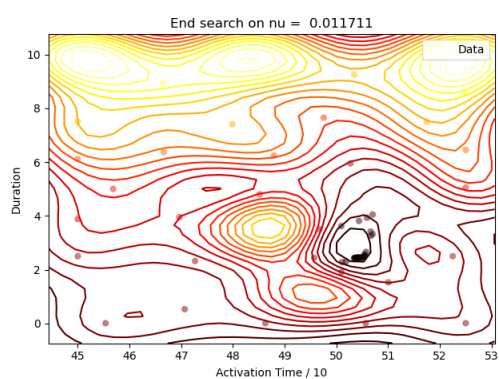


(b) Level curves for the new loss

Figure 8: Patient 2



(a) Duration vs alpha



(b) Level curves for the new loss

Figure 9: Patient 3

5 References

- Our code repository: [Repository](#)
- GPy package website: [Documentation](#)
- A nice post on Toward Data Science that expands our idea of hyperparameter tuning: [TDS](#)