Generalized Additive Models To Describe Heart-Lung Interactions During Open Thoracic Surgery: Project Documentation

Mads V. Clausen, 201905850 Andreas T. Damgaard, 201906422
Simon Hansen, 201707028 Casper M. Haurum, 201906479

Data Science, Aarhus University

May 31, 2021

Project documentation

What is the overall purpose of this project?
Through monitoring of the central venous pressure (CVP) there could potentially be a more streamlined guidance for the process of injecting fluids. To create some sort of reliable guide, we need to better understand how the CVP dynamically changes throughout a surgery. What effect do the lungs and heart have, and what effect does their interaction have? For surgeries in the thoracic cavity, there is further reason to investigate this – sometimes it is necessary to cut open the sternum to conduct the surgery. Therefore, there is a change in the environment surrounding the heart and lungs, and perhaps this could have an influence on how the CVP behaves.

How is this achieved?
Our contribution to this is to investigate how the effect of the heart-lung interaction on the CVP changes during open thoracic surgery. This is done by creating a visualization tool for researchers to use, where they in an easy manner can inspect their GAM objects, and then compare the effects of the heart and lungs, as well as their interaction, on the CVP before and after thorax has been opened.

Important things to know when using the Shiny application

To use the Shiny application, you are required to give it a GAM object as input. If you are unsure what a GAM is, see the below description of what it is, and why we use it.

Gam object

In order for us to model the central venous pressure (CVP), we use a generalized additive model (GAM), which is an extension of the generalized linear model (GLM) with smoothing functions, in our case cyclic cubic regression splines. Cyclic was chosen because our variables of interest (QRS cycle and inspiration cycle) are both cyclic in nature.

The individual observations in our model is a measure of CVP, the relative position in the heartbeat cycle (measured through relative position since last QRS complex), the relative position in the respiratory cycle (we do not know exact timing of this), and time since first observation for this surgery.

Because we are interested in the heart-lung interaction on CVP, we create a model explaining CVP by the individual effects of the heart and the lungs, the interactive effect of the heart and the lungs, and a time effect. The time effect is included to control for the natural change in CVP over time. The individual effects are added to further distinguish between individual effects and interactive effects. The model also includes an intercept, which ensures that the mean of each effect is equal to 0, avoiding identifiability issues.

We have chosen to use the same parameters in all GAM objects, for all patients, which was done after careful consideration. We specified these exacts values after testing different outcomes on different patients, and found that the parameters that gave the overall best generalization was the ones given below. To further confirm that these parameter values are close to optimal, we have dedicated a good amount of time on model validation, in order for us to be confident in the robustness of the models.

```r
cvp_gam <- bam(cvp ~
                 s(qrs_rel_index, bs = 'cc', k = 50) +
                 s(insp_rel_index, bs = 'cc', k = 30) +
                 ti(qrs_rel_index,
                    insp_rel_index,
                    bs = c('cc', 'cc'),
                    k = c(20, 10)) +
                 s(time_s),
               method = 'REML',
               data = patient_data set,
               rho = 0.95,
               nthreads = 16)
```

Explanation of model code:

- `bam()`
  This is used instead of `gam()` because of its computational efficiency.
- `s()`
  Initialization of smooth class.
- `ti()`
  Initialization of smooth class for tensor product.
- `bs = 'cc'`
  Cyclic cubic regression spline (the chosen smooth class).
- `k`
  Dimension of basis used for smooth term.
- `method = 'REML'`
  Restricted maximum likelihood. Computationally efficient and generally produces the best results.
- `rho = 0.95`
  Accounts for autocorrelation by including a rho-term (the correlation coefficient in an AR(1) model of the residuals). This reduces the models' overconfidence caused by high autocorrelation in the data, effectively widening the confidence intervals.
- `nthreads = 16`
  Number of (virtual) cores. Used for computational gain.

Important functions

The most essential functions used to create the Shiny application are described below.

gen_predictions()

- A function that generates predictions according to the given model.
- The code is a slightly modified version of Johannes'.
- It takes a GAM object as input.

create_patient_preds()

- A function that creates predictions and combines data from closed and open chest into one data frame.

- As input, it takes an RDS file of the open chest GAM and an RDS file of the closed chest GAM.
- It returns predictions for both the open and closed data.

stacked_plot()

- This function takes (inspiration/qrs) and (open/closed chest) and returns the plot for each combination.

stacked_diff_plot()

- This function takes (inspiration/qrs) and (open/closed chest) and returns the plots of the difference in open/closed-chest.

move_contour()

- This function takes the combined data set for open and closed, and the desired index placement of the largest respiration effect for open and closed, respectively, as inputs.
- It returns a new data set where the order has been rearranged according to the desired index placement.

create_contour_diff_data()

- This function takes the combined data set as input.
- It returns the combined data with the difference in open and closed chest as a new column.
- This is done by subtracting the open estimate from the closed estimate.

create_contour_viz()

- This functions creates a contour plot for a data set with x-axis as QRS index and y-axis as respiration index.

create_viz()

- A function that creates a snapshot visualization of CVP as a function of QRS cycle, comparing closed and open predictions at 10 different times in the respiration cycle. Contains both marginal effects, interaction effect and model intercept.

diff_snap_plot()

- This function takes the combined data as input.

- It then takes the difference in open and closed chest and returns a snapshot visualization of this difference.

give_it_center()

- This function takes a data set and the desired index placement of the greatest respiration effect as inputs.
- It calculates whether the inspiration indexes should be shifted to the left or right, in order for the largest respiration effect to be placed at the selected index and returns the shifted data set.

insp_add_sub_val()

- This function takes a data set and the desired index placement of the greatest respiration effect as inputs.
- It returns the value to add (negative for subtraction) from each inspiration index of a data set in order for the largest respiration effect to be placed at the selected index.

Shiny application

In this Shiny application, we have the following tabs:

- **Main page**
  The main page is where the most relevant information is present. This includes an input box in the left side of the page, where you can choose the patient that you want to inspect. Moreover, there is an info box with model details, such as the model intercept and which time interval was used to build the model. Further below, some model scores are presented to allow for model validation.
  In the main section of the page, three plot-sections will then appear:
  Top plots:
  - The top left plot is a comparison of the marginal effects of respiration on CVP when the chest is open and closed, respectively.
  - The top right plot is a comparison of the marginal effects of QRS on CVP when the chest is open and closed, respectively.

  Middle plots:
  - The contour plots show the effect of the interaction between the respiration and the QRS relative index, on the CVP.

– The left contour plot shows the effect when the chest is closed, while the right contour plot shows the effect when the chest is opened.

Bottom plots:

– The snapshot plots are a combination of the marginal effects of the relative position in the two cycles, their interaction, and the intercept. If the lines are on top of each other, there is no marginal effect from the relative position in the respiration cycle. If all the lines are parallel, there appears to be no interactive effect, but if the lines possess different morphology, there appears to be an interactive effect.

– The left snapshot plot shows the effects when the chest is closed, while the right snapshot plot shows the effects when the chest is opened.

Because most interesting answers can be expected to lie in the differences within the plots of closed and open chest, we have made it possible for the user to check-off "Closed/open difference plots". If checked, the user will be presented with three new plot-sections:

Top difference plots:

– The top left plot shows the difference in the marginal effect of respiration on CVP before and after thoracotomy. We have done this by taking the effect during open chest and subtracted the effect during closed chest.

– The top right plot shows the difference in the marginal effect of QRS on CVP before and after thoracotomy. We have done this by taking the effect during open chest and subtracted the effect during closed chest.

Middle difference plots:

– A contour plot showing the difference in the interaction effect between the respiration and the QRS relative index, before and after thoracotomy.

Bottom difference plots:

– This plot shows the difference (calculated as open minus closed) in CVP as a function of the relative QRS index at 10 selected respiration index times.

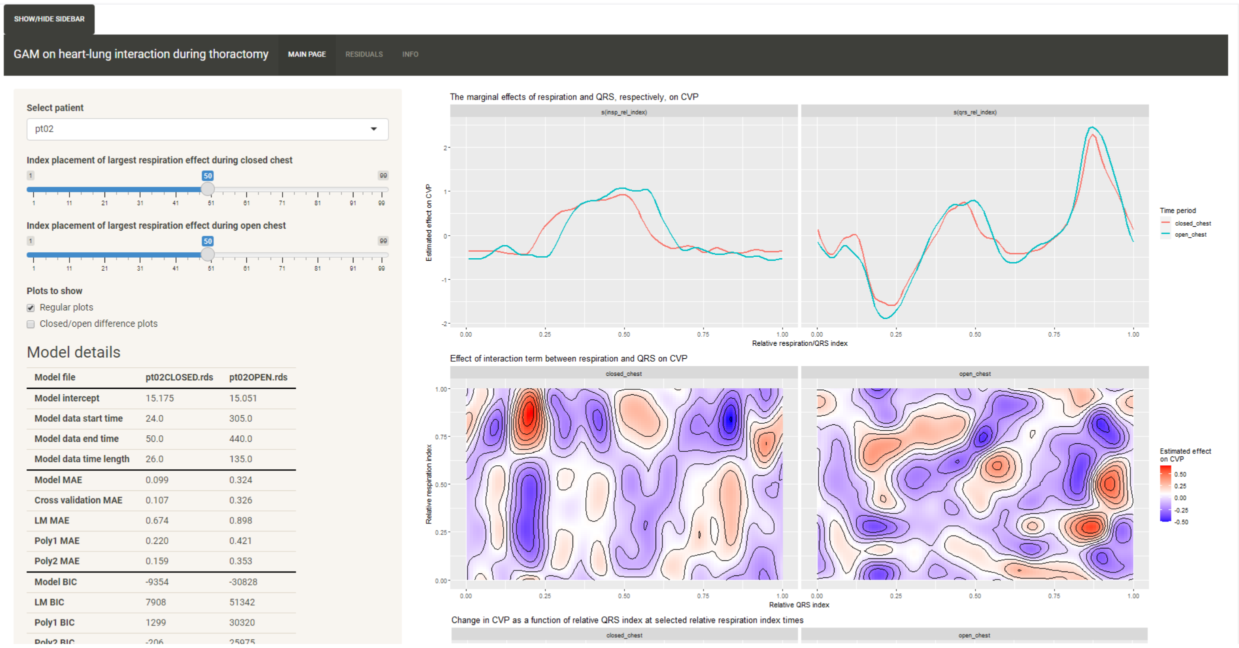
- **Residuals page**
  This page is intended as a way of inspecting the data used to construct the model. Sometimes, the results produced by the models can be confusing or need to be validated in some way. This perplexity could be caused by the presence of noise or a short time

slot. Therefore, we have dedicated a page to enable a user to inspect the observed values, values predicted by the model, and the residuals. The plots are labelled and assigned a legend to ease the interpretation.

- **Info**

  This page can be used as a guideline of the interpretation of the **Main page** as well as the **Residuals page**. There are general descriptions as well as a interpretation of the plots. Furthermore, there is a short description of how to use the application.

Attachments:



Model file	pt02CLOSED.rds	pt02OPEN.rds
Model intercept	15.175	15.051
Model data start time	24.0	305.0
Model data end time	50.0	440.0
Model data time length	26.0	135.0
Model MAE	0.099	0.324
Cross validation MAE	0.107	0.326
LM MAE	0.674	0.898
Poly1 MAE	0.220	0.421
Poly2 MAE	0.159	0.353
Model BIC	-9354	-30828
LM BIC	7908	51342
Poly1 BIC	1299	30320

Fig. 1 Main page in Shiny application.

Fig. 2 Page to inspect residuals in Shiny application.