INTERNSHIP REPORT
Gaussian Processes for the prediction of length of stay in ICU

Author:
Thomas Maitre

Supervisor:
Aleksei Tiulpin

2021/2022

Contents

1 Introduction

As it was shown during the COVID crisis, an efficient management of the intensive care unit (ICU) service is crucial not only for the sake of a greater number of people but also in order to reduce the costs from the hospitals perspective. That being said, this paper focus on predicting information that would help such a management: the length of stay and the mortality. The prediction of these information are respectively a regression and a classification task. This study aims to use Gaussian Processes (GP) to achieve these tasks. GP are equivalent to many models used to make predictions such as Bayesian linear models, spline models or large neural networks (under suitable conditions) [1] but their interpretation is easier.

The problem at hand that needs to be solved is to, given a training data set, find a function which enables the prediction for every possible input values. In order to do this, in the GP framework one can give a prior probability to find these functions, with more likeliness for smoother functions for instance. Then one can get a posterior by adding the information given by the prior and the data set. The benefit of this is that uncertainties are reduced near the data points, so the larger the data set and the more accurate the prediction is.

This work was done as a three-month internship within the Intelligent Medical Systems (IMEDS) team, which is part of the research units of medical imaging, physics and technology of University of Oulu's faculty of medicine. This research unit develops new AI methods for medical applications such as the early detection of osteoarthritis. In the future, the team also hopes to build new AI-tailored hospital processes. [2]

The internship first task was to learn about linear regression (which in fact can be extended to polynomial regression thanks to the use of a feature space [3]), Bayesian logistic regression, GP regression and GP classification using either the Laplace or Expectation Propagation (EP) method to approximate the posterior. At the same time, I was asked to implement these methods on Python. Once this was done, I worked on the original problem with real medical data.

This paper is essentially divided in two main sections. In the first one, the general theory [1] [3] is discussed and some essentials results are given. Considering that this report is size-limited, only a fraction of the studied theory is reviewed here. In the second part, the theory is applied to the specific case of predicting the length of stay of patients in ICU using the eICU [4] data set.

2 Theory

We can define a GP as a collection of random variables with any finite number of them having a joint Gaussian distribution. As such, a GP is completely defined by a mean and a covariance function:

$$\begin{cases} m(\mathbf{x}) = \mathbb{E}(f(\mathbf{x})) \\ k(\mathbf{x}, \mathbf{x}') = \mathbb{E}((f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))) \end{cases} \tag{1}$$

A GP allows to find some kind of interpolation over the input data we feed it, but in the real world there is almost always noise to consider. We represent this noise with the variable $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, and with y being the target values we have the following equation:

$$y = f(\mathbf{x}) + \varepsilon \tag{2}$$

Let $\mathbf{X}$ be the aggregation matrix of input vectors and $\mathbf{y}$ the target vector. What we need to get from this point on is the posterior $p(f|\mathbf{X}, \mathbf{y})$. This information is crucial, once this probability is known one can make predictions on new points $\mathbf{X}_*$ thanks to the relation:

$$p(f_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{X}_*, f)p(f|\mathbf{X}, \mathbf{y})df \tag{3}$$

To have knowledge over the posterior, one have to use the training data points. Given such points $\mathbf{X}$, the Bayes formula gives:

$$\begin{aligned} p(f \mid \mathbf{X}, \mathbf{y}) &= \frac{p(f, \mathbf{X}, \mathbf{y})}{p(\mathbf{X}, \mathbf{y})} = \frac{p(\mathbf{y} \mid f, \mathbf{X})p(f, \mathbf{X})}{p(\mathbf{y} \mid \mathbf{X})p(\mathbf{X})} \\ &= \frac{p(\mathbf{y} \mid f, \mathbf{X})p(f)}{p(\mathbf{y} \mid \mathbf{X})} \\ &= \frac{\text{likelihood . prior}}{\text{marginal likelihood}} \end{aligned} \tag{4}$$

because f and $\mathbf{X}$ are independent random variables.

Having a certain knowledge over the data, one must choose adequate likelihoods [5] and prior. Moreover in GP, the likelihood should be chosen with regard to three aspects. First, since for many distributions the parameters are constrained meanwhile the GP function isn't, there is a need of using link functions (logistic function, cdf, ...) in order to adapt the gaussian f to the constraints. Then the likelihood should be log-concave to ensure no convergence related issues. Finally the likelihood should verify $p(\mathbf{y}|f) = \Pi_i p(y_i|f_i)$.

Equation (4) eventually becomes:

$$p(f|\mathbf{X}, \mathbf{y}) = \frac{p(f)\Pi_i p(y_i|f_i)}{\int p(f')\Pi_i p(y_i|f_i')df'} \tag{5}$$

We can see that for a Gaussian likelihood, the posterior remains Gaussian. However a non-Gaussian likelihood will lead to a non-Gaussian posterior, which is intractable because of the computation of the marginal likelihood for high dimensions. Then we need to approximate this term. Several methods exists, but here we're going to approximate this with a Gaussian distribution. We're going to see in particular the Laplace and Expectation Propagation (EP) algorithms. These algorithms allow the approximation $p(\mathbf{y}|f) = q(f) = \mathcal{N}(\mu, \Sigma)$ thanks to the determination of the mean and variance parameters. These parameters match the data points mean and variance for the Laplace method, meanwhile the EP focus on minimizing the divergence. In the following, we're going to illustrate both the case of a Gaussian and a non-Gaussian likelihood respectively in the GP regression section and the GP classification section.

2.1 Bayesian Linear Regression

A typical regression problem that one would want to solve is the standard linear regression model:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} \tag{6}$$

where the prior $w \sim \mathcal{N}(0, \Sigma_p)$ and $\phi : \mathbb{R}^D \to \mathbb{R}^K$ is a non-linear function of the inputs. $\phi(x)$ is called the feature vector and is used to fit more than straight lines to data, such as polynomial lines for instance if $\phi(x) = (1, x, x^2, ..., x^{K-1})^T$ [3].

We want information about the posterior. We saw with equation (4) that this information is given by the likelihood and the prior. Since we assumed i.i.d (independent identically distributed) Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, the likelihood can be expressed as:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \prod_i \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma_n^2}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{|\mathbf{y} - \Phi^T\mathbf{w}|^2}{2\sigma_n^2}) \\ &= \mathcal{N}(\Phi^T\mathbf{w}, \sigma_n^2 I) \end{aligned} \tag{7}$$

where Φ is the matrix aggregation of columns $\phi(\mathbf{x})$. Using the "completion of the square" method (see [3] for more details), we have:

$$\begin{aligned} \log p(\mathbf{w}|\mathbf{y}, \mathbf{X}) &= \log p(\mathbf{y}|\mathbf{w}, \mathbf{X}) + \log p(\mathbf{w}) + C \\ &= -\frac{|\mathbf{y} - \Phi^T\mathbf{w}|^2}{2\sigma_n^2} - \frac{1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w} + C \\ &= -\frac{1}{2}(\mathbf{w} - \bar{\mathbf{w}})^T(\sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1})(\mathbf{w} - \bar{\mathbf{w}}) + C \end{aligned} \tag{8}$$

where C is a constant expression w.r.t w that may differ from one line to another and $\bar{\mathbf{w}} = \sigma_n^{-2}(\sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1})^{-1}\Phi y$. We recognize a Gaussian distribution for the posterior with mean $\bar{\mathbf{w}}$ and covariance matrix $A^{-1} = (\sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1})^{-1}$. The mean of the Gaussian distribution is also its mode and is called the MAP (maximum *a posteriori*).

Now that we determined the posterior for the linear regression, we can make predictions using equation (3) such that:

$$p(f_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\Phi_*^T\bar{\mathbf{w}}, \Phi_*^T A^{-1}\Phi_*) \tag{9}$$

A pseudo-code version of the linear regression is given at Algorithm 1.

Algorithm 1: Mean and variance predictions for LR

```
Data: $\mathbf{X}$(inputs), $\mathbf{y}$(targets), $\mathbf{X}_*$(test inputs), $\phi$, $\Sigma_p^{-1}$(prior's covariance)
Result: $\bar{f}_*$(mean), $V[f_*]$(covariance)
$A = \sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1}$
$L = \text{cholesky}(A)$
$\bar{w} = L^T \backslash (L \backslash \sigma_n^{-2}\Phi y)$
$\bar{f}_* = \Phi_*^T \bar{w}$
$V[f_*] = \Phi_*^T A^{-1}\Phi_*$        ($n_*$ calls to the triangular solver
                                         using Cholesky on each column of $\Phi_*$)
```

We can note that this problem is a GP with mean and variance given by the definition (1) in equation (10):

$$\begin{cases} \mathbb{E}(f(\mathbf{x})) = \phi(\mathbf{x})^T \mathbb{E}(\mathbf{w}) = 0 \\ \mathbb{E}(f(\mathbf{x})f(\mathbf{x}')) = \phi(\mathbf{x})^T \mathbb{E}(\mathbf{w}\mathbf{w}^T)\phi(\mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}') \end{cases} \tag{10}$$

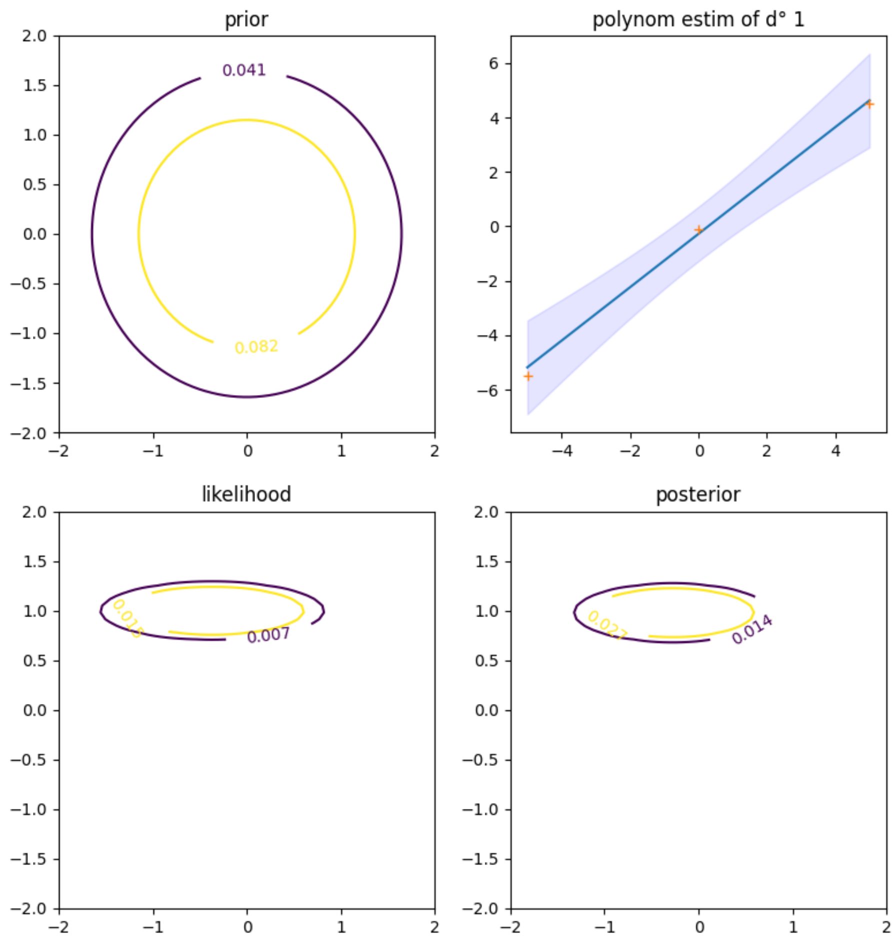

Figure 1: Bayesian Linear Regression in dimension 2 with $\mathbb{R}[X]$ as feature space

Figure 1 shows an example of Linear Regression in dimension 2. At the top right hand corner, contours (1 and 2 standard deviation equi-probability) of the Gaussian prior $\mathcal{N}(0, I)$ are represented. In the lower right hand corner, the same contours of the likelihood given by equation (7) are also drawn. The same contours for the posterior (given by Bayes rule) are represented on the lower left hand corner. On the top left hand corner, the predictive mean with 3 training points (+) are represented. The blue shaded area represents the mean plus or minus two standard deviation.

In Figure 2, we represented the predictive mean as well as the 95% confidence interval (shaded in blue) for the same inputs as in Figure 1 for different feature spaces. It seems that the higher the dimension of the latter, the higher the uncertainties around the predictive mean are.

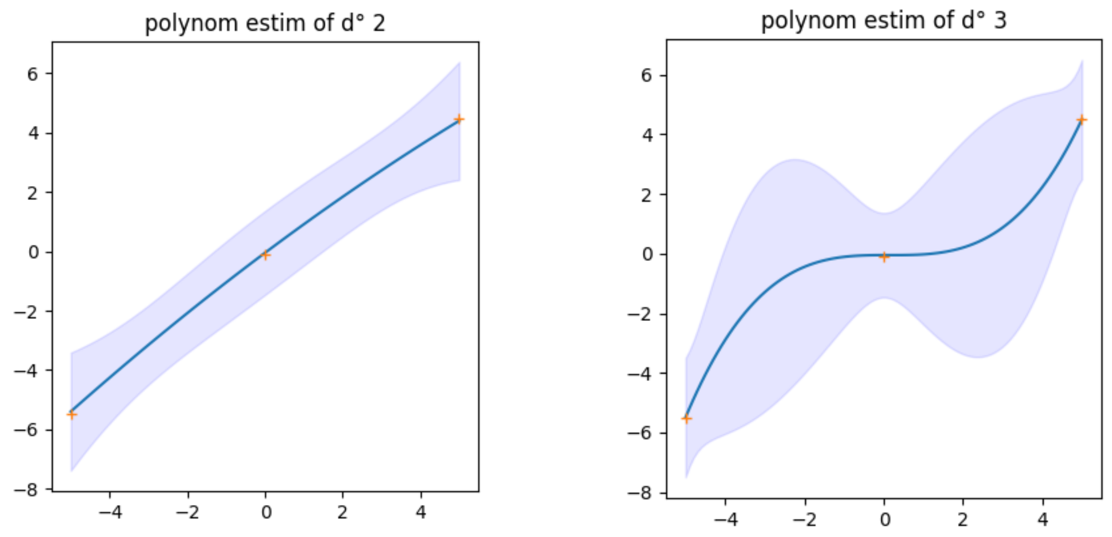

Figure 2: Bayesian Linear Regression with feature space being $\mathbb{R}_2[X]$ on the left and $\mathbb{R}_3[X]$ on the right

2.2 Gaussian Process Regression

As previously defined in (1), a GP is given by its mean and covariance function. Usually, a simple zero-mean is chosen. On the other hand, the choice of the covariance function is much broader and depends of its properties. There are some mandatory properties needed for a function to be a covariance function. Such properties are listed here, as well as some other more optional.

First, the covariance function must be symmetric. It also has to be definite semi-positive i.e that $\int k(\mathbf{x},\mathbf{x}')f(\mathbf{x})f(\mathbf{x}')d\mu(\mathbf{x})d\mu(\mathbf{x}') \geq 0,\ \forall f \in L^2$. An interesting property to examine is the invariance of the covariance function. A function of $\mathbf{x}-\mathbf{x}'$ is said to be stationary and is invariant to translations. A special case of stationary covariance functions are the isotropic functions which depends of $|\mathbf{x}-\mathbf{x}'|$. On the other hand, functions of $\mathbf{x}\mathbf{x}'$ are called dot products and are invariant to rotations. We write $K(\mathbf{X},\mathbf{X})$ the covariance matrix where $K(\mathbf{X},\mathbf{X})_{i,j} = k(\mathbf{x}_i,\mathbf{x}_j)$.

A common choice of covariance function is the squared exponential covariance function $k(\mathbf{x}_p,\mathbf{x}_q) = \exp(-\frac{1}{2}|\mathbf{x}_p-\mathbf{x}_q|^2)$. This choice is made such that the covariance function decomposition from Mercer's theorem is actually a linear combination of Gaussian-shaped basis functions (see [1] for more information).

The specification of the covariance function implies a distribution over functions. We consider the simplest case of a Gaussian likelihood. Let's choose some inputs points $\mathbf{X}_*$, we can generate a random Gaussian vector $f_* \sim \mathcal{N}(0, K(\mathbf{X}_*,\mathbf{X}_*))$ which is the prior. Then we want information on the posterior i.e the distribution with knowledge provided by the training data f. We have the joint distribution:

$$\begin{pmatrix} f \\ f_* \end{pmatrix} = \mathcal{N}\left(0, \begin{pmatrix} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{pmatrix}\right) \tag{11}$$

By conditioning this joint distribution on the observations, we get:

$$\begin{aligned} f_*|\mathbf{X}_*,\mathbf{X},f \sim \mathcal{N}(&K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}f, \\ &K(\mathbf{X}_*,\mathbf{X}_*) - K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}K(\mathbf{X},\mathbf{X}_*)) \end{aligned} \tag{12}$$

(see Annex 6.2 for more details). As previously mentioned, noise can be added to the observations thanks to the relation $y = f(\mathbf{x}) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. To do that, one simply needs to replace occurrences of $K(\mathbf{X}, \mathbf{X})$ with $K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I$ and any occurrences of f by $\mathbf{y}$. An algorithm of Gaussian Process Regression (GPR) is given (see algorithm 2) where $K(\mathbf{X}, \mathbf{X})$ is written as K for lighter notations.

Algorithm 2: Mean and variance predictions for GPR, as well as the log marginal likelihood

```
  Data: $X(\text{inputs}), y(\text{targets}), X_*(\text{test inputs})$
  Result: $\bar{f}_*(mean), \mathbb{V}[f_*], \log p(y|X)$
  $k_* = K(X, X_*)$
  $L = \text{cholesky}(K + \sigma_n^2 I)$
  $\alpha = L^T \backslash (L \backslash y)$
  $\bar{f}_* = k_*^T \alpha$
  $v = L \backslash k_*$
  $\mathbb{V}[f_*] = K(X_*, X_*) - v^T v$
  $\log p(y|X) = -\frac{1}{2} y^T \alpha - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$
```

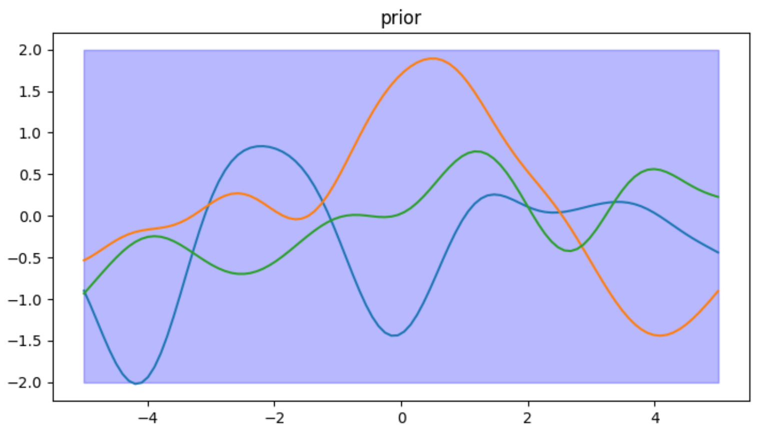

Figure 3: GP Regression in 1D - three samples from the prior $\mathcal{N}(0, 1)$ are drawn

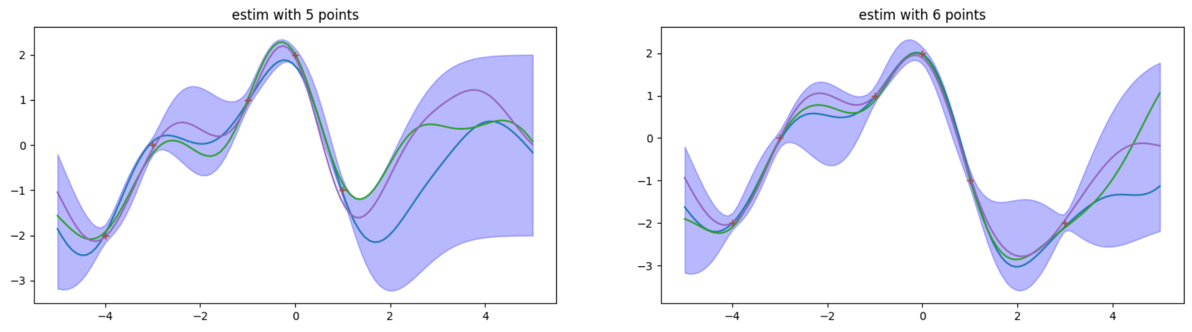

Figure 4: GP Regression in 1D - three samples from the predictive posterior given 5 inputs points in the left and 6 in the right

To illustrate the GP Regression with a concrete example, we draw three samples from the prior (represented in Figure 3). Then considering the inputs points $(X, Y) = [(-4,-2), (-3,0), (-1,1), (0,2), (1,-1)]$, we get predictive mean and variance for the predictive posterior (see Algorithm 2) which are used to generate three samples in the left of Figure 4. The right part of Figure 4 is obtained when adding another observation $(x, y) = (3, -2)$ to the inputs points. Represented in shaded blue still is the 95% confidence interval, which is the point-wise mean plus or minus two times the standard deviation. Figure 4 really shows that uncertainty around the inputs points are small and that adding new inputs points allows for these uncertainties to drastically decrease.

2.3 Gaussian Process Classification

In this section we discuss GP binary classification, that is to said we are interested in the probability of the target belonging whether to class $+1$ or -1. In order to have knowledge on that, as it was done for GPR we first need to determine the distribution over $f_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*$. Because we are now considering a non-Gaussian likelihood, this step is more difficult than in GPR and requires an approximation method. Our focus in this section is on the Laplace approximation, but it is worth noting that this method is quite poor compared to more sophisticated ones like the Expectation Propagation (EP) method [1].

2.3.1 *Laplace approximation*

The goal of the Laplace method is to approximate the posterior with a normal distribution. The probability density function of a Gaussian with mean μ and variance Σ can be written as $p(f) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp(-\frac{1}{2}(f-\mu)\Sigma^{-1}(f-\mu))$. This expression is such that its log probability is quadratic and can be written as $\log p(f) = -\frac{1}{2}(f-\mu)^T\Sigma^{-1}(f-\mu) + C$ where C is a constant expression w.r.t f. Let's transform the posterior expression to get a similar expression.

According to Bayes rule,

$$\log p(f|\mathbf{X}, \mathbf{y}) = \log \frac{p(f)p(\mathbf{y}|f, \mathbf{X})}{Z} = h(f) + C \tag{13}$$

where Z is the marginal likelihood, $h(f) = \log p(f)p(\mathbf{y}|f, \mathbf{X})$ and C refers to a constant expression. We make a second order Taylor expansion of h around the maximum of the posterior $\hat{f}$:

$$h(f) = h(\hat{f}) + (f-\hat{f})^T\nabla h(\hat{f}) + (f-\hat{f})^T\nabla\nabla h(\hat{f})(f-\hat{f}) \tag{14}$$

Since $\nabla h(\hat{f}) = 0$, we can see that h is proportional ($\implies$ the posterior is proportional) to a Gaussian law with mean $\hat{f}$ and variance $\nabla\nabla \log p(\hat{f}|\mathbf{X}, \mathbf{y})$. We consider that the prior $f \sim \mathcal{N}(0, K)$ with $K = K(\mathbf{X}, \mathbf{X})$, and choose the likelihood as explained previously (common choices are the logistic function and the cumulative Gaussian function). We get the following by including any constant expressions that comes up to the already existing C constant:

$$\log p(f|\mathbf{y}, \mathbf{X}) = \log p(\mathbf{y}|f, \mathbf{X}) - \frac{1}{2}f^TK^{-1}f + C \tag{15}$$

Thanks to that equation, one can get the derivatives of the log-posterior:

$$\nabla \log p(f|\mathbf{X}, \mathbf{y}) = \nabla \log p(\mathbf{y}|f, \mathbf{X}) - K^{-1}f \tag{16}$$

$$\nabla\nabla \log p(f|\mathbf{X}, \mathbf{y}) = \nabla\nabla \log p(\mathbf{y}|f, \mathbf{X}) - K^{-1} = -W - K^{-1} \tag{17}$$

where $W = \nabla\nabla \log p(\mathbf{y}|f, \mathbf{X})$. What remains to be done now is to find the value of the mode $\hat{f}$. To achieve this result one can use Newton's method for instance, with the iteration:

$$f_{k+1} = f_k - (\nabla\nabla h)^{-1}\nabla h = (K^{-1} + W)^{-1}(Wf_k + \nabla \log p(\mathbf{y}|f_k, \mathbf{X})) \tag{18}$$

Once the mode has been found, we can finally explicitly specify the posterior approximation as the Gaussian distribution $q(f|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\hat{f}, (\nabla\nabla p(\hat{f}|\mathbf{X}, \mathbf{y}))^{-1} = (K^{-1} + W)^{-1})$.

2.3.2 Predictions for GP classification

Predictions can then be made in the same way as in section 2.2 (see relation (12)). The mean of the predictive Gaussian can be determined by the relation:

$$\bar{f}_* = \mathbb{E}_q(f_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*) = K(\mathbf{X}_*, \mathbf{X})K^{-1}\hat{f} = K(\mathbf{X}_*, \mathbf{X})\nabla p(\mathbf{y}|\hat{f}, \mathbf{X}) \tag{19}$$

The variance of the predictive Gaussian is a little harder to obtain, as it is now the sum of two terms. The first is the variance given by the relation (12), the second appears because $\mathbb{E}[f_*|X, f, X_*] = K(X_*, X)K^{-1}f$ depends of f. Consequently, the equation (20) is found:

$$\begin{aligned}
\mathbb{V}_q[f_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*] &= \mathbb{E}_{p(f_*|\mathbf{X},\mathbf{y},\mathbf{X}_*)}[(f_* - \mathbb{E}[f_*|\mathbf{X}, f, \mathbf{X}_*])^2] \\
&+ \mathbb{E}_{q(f|\mathbf{X},\mathbf{y})}[(\mathbb{E}[f_*|\mathbf{X}, f, \mathbf{X}_*] - \mathbb{E}[f_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*])^2] \\
&= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})K^{-1}K(\mathbf{X}, \mathbf{X}_*) \\
&+ K(\mathbf{X}_*, \mathbf{X})K^{-1}(K^{-1} + W)^{-1}K^{-1}K(\mathbf{X}, \mathbf{X}_*) \\
&= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})(K + W^{-1})^{-1}K(\mathbf{X}, \mathbf{X}_*)
\end{aligned} \tag{20}$$

where the last line is found using the matrix inversion lemma (see Annex 6.1).

Now that the mean and variance of f_* are known, the predictions are made using $\pi_* = p(y = +1|\mathbf{X}_*) = \sigma(f_*)$ with the computation of the averaged prediction:

$$\bar{\pi}_* = p(y_* = +1|\mathbf{X}, \mathbf{y}, \mathbf{X}_*) = \mathbb{E}[\pi_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*] = \int \sigma(f_*)q(f_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*)df_* \tag{21}$$

where σ is a sigmoid function (e.g logistic, cumulative Gaussian, ...).

This prediction, often intractable because of the non-linearity of σ, is not to be mistaken with the MAP prediction:

$$\hat{\pi}_* = \sigma(\mathbb{E}_q[f_*|y]) \tag{22}$$

The algorithm for the prediction of the class probability is given in Algorithm 4.

In the event that the sigmoid function σ in equation (21) is the cumulative Gaussian (or alternatively the logistic function that can be approximated by the cumulative Gaussian), one can analytically compute this integral. In this case, according to Annex 6.4 we have:

$$\tilde{\pi}_* = \Phi(\frac{\bar{f}_*}{\sqrt{1 + \mathbb{V}[f_*]}}) \tag{23}$$

Algorithms 3 and 4 were made in order to sum-up each steps of the process. For the Cholesky decomposition to be safe, we noticed that $(K^{-1} + W)^{-1} = K - KW^{1/2}B^{-1}W^{1/2}K$ using the matrix inversion lemma with $B = I + W^{1/2}KW^{1/2}$ (refer to Annex 6.1).

These algorithms were computed on two test cases defined below with σ being the logistic function, and subsequently Figures 5 and 6 were produced. In the first test case, we chose a random dataset $\mathbf{X}$ made from a mixture of three Gaussians with a standard deviation of 0.8 and means of respectively $6, 0, 2$. 20 points were drawn from the first Gaussian, 30 from the second and 10 for the third. A weight of 1 was given to the points of the second Gaussian while -1 was given to the first and second Gaussians. The predictive probability is then plotted in function of the inputs in Figure 5. We can see both the MAP and probit approximations to compute the intractable integral in equation (21). Additionally, the predictive probability of a target belonging to class

Algorithm 3: Mode determination through Newton's method

```
Data: $y$(targets), $K$(covariance matrix)
Result: $\hat{f}$(mode)
$f = 0$, $f_{old} = 1$, $\epsilon = 1e-6$
while $|f - f_{old}| > \epsilon$ do
    $W = -\nabla\nabla \log p(y|f)$
    $B = I + W^{1/2} K W^{1/2}$
    $L = \text{cholesky}(B)$
    $b = Wf + \nabla \log p(f|y)$
    $a = b - W^{1/2} L^T \backslash (L \backslash W^{1/2} K b)$
    $f_{old} = f$
    $f = Ka$
end
$\hat{f} = f$
```

Algorithm 4: Predictions with Laplace approximation for GP Classification

```
Data: $X$(inputs), $y$(targets), $X_*$(test inputs), $\hat{f}$(mode)
Result: $\bar{f}_*$(mean), $V[f_*]$(covariance), $\bar{\pi}_*$(predictive class probability)
$k_* = K(X, X_*)$
$\bar{f}_* = k_*^T \nabla p(y|\hat{f})$
$W = -\nabla\nabla \log p(y|\hat{f})$
$L = \text{cholesky}(I + W^{1/2} K W^{1/2})$
$v = L \backslash (W^{1/2} k_*)$
$\mathbb{V}[f_*] = K(X_*, X_*) - v^T v$
$\bar{\pi}_* = \int \sigma(z) \mathcal{N}(z|\bar{f}_*, \mathbb{V}[f_*]) dz$
```

one is also computed with the EP method. The hyper-parameters l (length scale) and σ_f (signal variance) in the squared-exponential covariance function $k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp(-\frac{1}{2l^2}|\mathbf{x}_p - \mathbf{x}_q|^2)$ were set to $l = 2.6$ and $\sigma_f = 7.0$ in order to maximize the approximate marginal likelihood for Laplace's method (refer to [1] for more information).

Overall, these predictions have similar shapes and are indicative of the inputs repartition. We can also notice that in this example, the probit approximation seems to be a bit more shy than the others (for instance, it does not ascertain with a probability of 1 that a input belongs to class "+1"). In the same experience, we represented the possibles latent functions $f(x)$ in the left part of Figure 5 with the mean in lines and the 95% confidence interval represented with shaded areas. The results relative to the EP method were plotted in red while the results of the Laplace method were plotted in blue. Several results from this test case are explained in [1], but one that seems important to me is the effect of uncertainties of the latent function over the predictive probability plotted in right part of Figure 5. This effect can be seen in the region delimited by $x = 2$ and $x = 4$ for instance: one can notice that while the latent mean increase, the predictive probability decrease because of the augmentation of the uncertainties in this same region.

The second test case is bi-dimensional. Given some random points represented by a mixture of two Gaussians with means of $(-2, -1)$ and $(2, 3)$ and standard deviation of 1. In Figure 6, the data points labelled "+1" are represented with dots while the other are represented with crosses. Several contours are plotted, including the decision boundary with a predictive value of 0.5. The

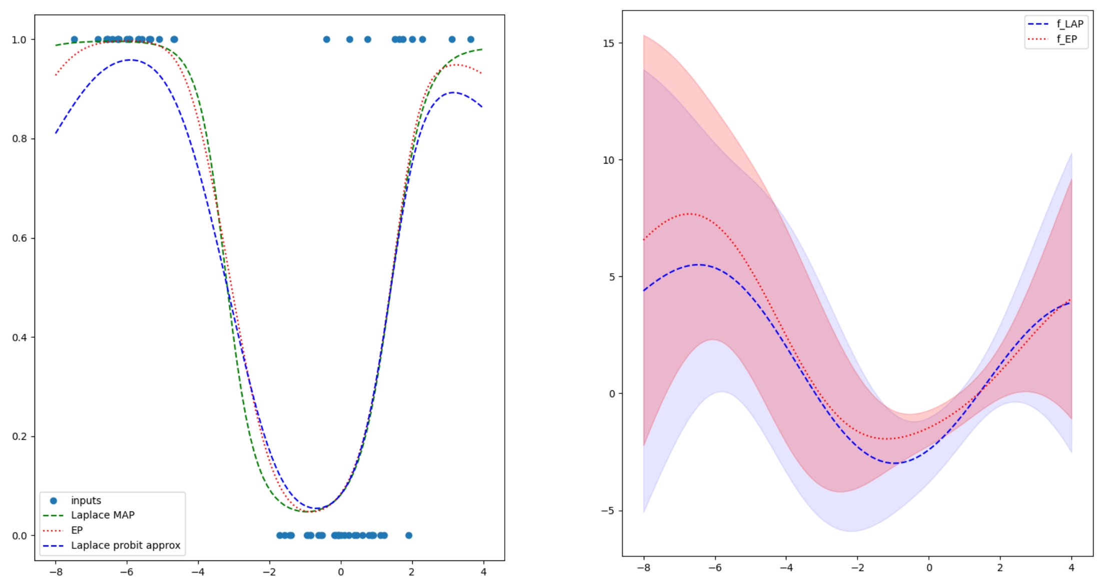

Figure 5: Predictive probability in function of the inputs for binary classification (right figure) and latent function f (left figure) in 1D for the first test case

two classes of points are well separated with this method. Moreover there is no assumptions as to the class value of the part of the input space in which there should be no information about the predictive class value. This isn't the case when performing Bayesian Logistic Regression [6] with Laplace method as in Figure 7.

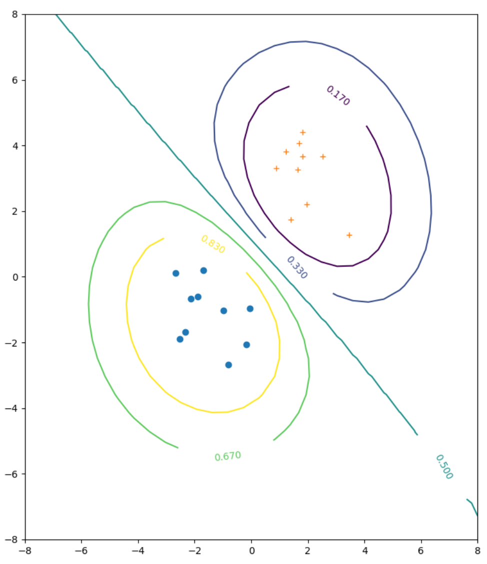

Figure 6: Contours of the predictive probability in function of the inputs for binary classification in 2D with Laplace's method

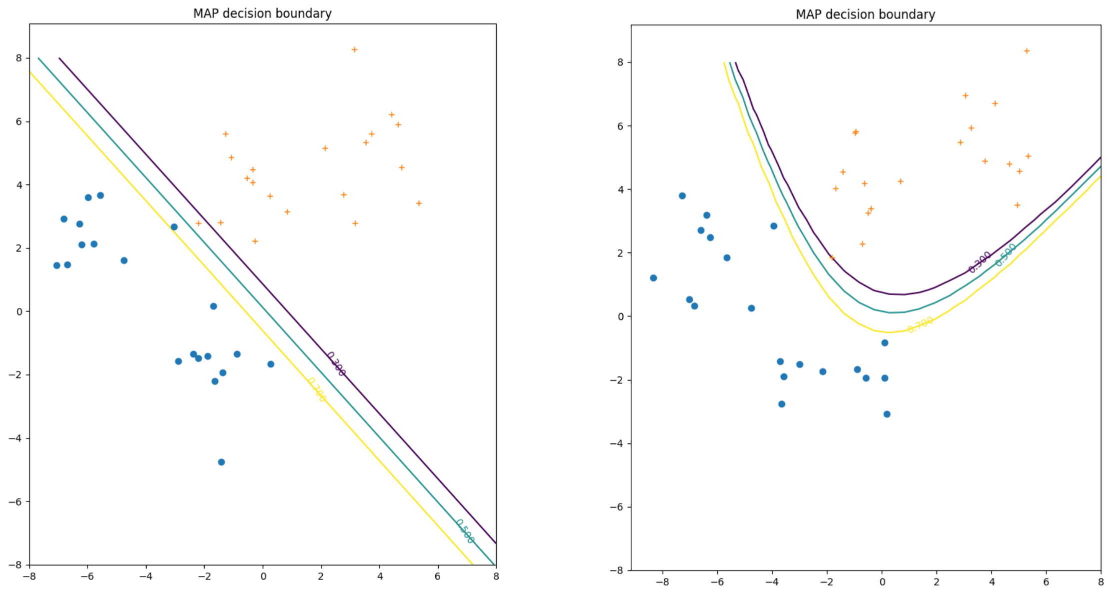

Figure 7: Bi-dimensional Bayesian Logistic Regression with feature space being $\mathbb{R}[X]$ in the left and $\mathbb{R}_2[X]$ in the right

2.4 Conclusion

To conclude this theoretical section, we first defined what was a GP and reminded important results like predicting by averaging over the posterior and Bayes rule. Then after presenting the linear regression model both the GP regression and the GP classification frameworks were studied, the former with and the later without a Gaussian likelihood as an attempt to illustrate different mechanisms. Keep in mind that even though it was not shown here, GP regression could also be done with a non-Gaussian likelihood and GP classification with a Gaussian likelihood. Additionally, among the different techniques to determine the posterior for a non-Gaussian likelihood, the Laplace approximation was also explained in this section.

3 Application to the eICU data set

3.1 Presentation of the data set

In order to make use of the theoretical results described in the previous section, real data from the eICU data set available through PhysioNet [7] was used. The eICU Collaborative Research Database [4] is a multi-center database comprising deidentified health data associated with 200,859 stays of 139,367 distincts patients to ICUs across the United States between 2014 and 2015. The database includes vital sign measurements, care plan documentation, severity of illness measures, diagnosis information, and treatment information.

3.2 Preprocessing

Scattered across many tables, eICU's data was found to be highly "sparse" in that measurements of the different features were not always done at the same time. If gathered in matrix, the resulting bi-dimensional array with axes being time and features would have a huge amount of missing (NaN) values. Because of that, there was a need for preprocessing in order to have exploitable data. The first part of the method described below is highly inspired by [8]. The second part essentially consists of adapting the output to be used for pythons GP libraries such as [9] or [10]. Data is selected mainly from three tables: *patient* contains general information on the different patients and stays while *lab* and *nurseCharting* provides specific information such as laboratory or vital measurements.

Each stay can be seen as a time series where features are measured at different time steps. The collection of features recorded for a specific time step is called a record. In a record the features measured can be either numerical or categorical, in which case encoding is necessary. In this paper, 13 numerical features and 7 categorical features have been considered. Table 1 gives the list of these features as well as their "normal" values [11].

Feature	Type	Normal value
Heart rate	Numerical	86 bpm
Mean arterial pressure	Numerical	77 mmHg
Diastolic blood pressure	Numerical	56 mmHg
Systolic blood pressure	Numerical	118 mmHg
O2 saturation	Numerical	98
Respiratory rate	Numerical	19 BPM
Temperature	Numerical	36 °C
Glucose	Numerical	128 mg/dL
FiO2	Numerical	0.21
pH	Numerical	7.4
Weight	Numerical	81 kg
Height	Numerical	170 cm
Age	Numerical	
Admission diagnosis	Categorical	
Ethnicity	Categorical	
Gender	Categorical	
Glasgow Coma Score Total	Categorical	15
Glasgow Coma Score Eyes	Categorical	4
Glasgow Coma Score Motor	Categorical	6
Glasgow Coma Score Verbal	Categorical	5

Table 1: List of features with their type (numerical or categorical) and normal value

The categorical features will be seen to be essential to solve the problem of the LoS prediction. However, they do require extra work in order to be used in a code. First we convert these features to an arbitrary integer value: for instance the gender 'Male' is represented as 1, 'Female' as 2 and if there is no information the value 0 is taken. Then each categorical feature is built one on top of another, that is to say that the final integer value of a certain variable is the sum of its arbitrary value and the max of the value of the categorical feature it is built on. At this step of the process, each categorical feature have a different integer value. However, such a pre processing can give more importance to certain values (for instance, a specific algorithm could favor high integer values over the smaller). Two methods have been considered here in order to avoid that: one hot encoding (OHE) which converts the categorical features into independent vectors, and embedding [12] which uses neural networks to learn the value of the features of interest.

To continue on the data preprocessing, only patients that are over 18 years old and for which we have at least 15 records are taken into account. We also take care to discard patients with more than one ICU stay so that the prediction of the length of stay is relevant. All of the exclusion criteria used during the preprocessing can be visualized in Annex 6.5 on Figure 8 if needed. For the selected 73,718 patients we group these records on 1 hour window. Then we impute the missing values based on the mean of that window and we take the last valid record. If there isn't any value of a certain feature in this one hour window, imputation is made using the normal values from Table 1.

In [8] data is stored in a three dimensional array of shape (n_patients, n_records_max, n_features), meanwhile in GP libraries data inputs need to be in specified as a two-dimensional array of shape (n_samples, n_features). What was therefore done was to iterate on the patients to extract the (n_records, n_features) matrix X_s containing the different features values for each record. Then all of these arrays were concatenated to form X_{full}. This operation was done for both the training and the testing data set.

3.3 Evaluation metrics

There are many metrics that allows oneself to know whether a model performs well or not. We will be using in this paper the coefficient of determination R^2 and mean absolute error (MAE) for the regression task. We remind here how these scores are defined:

$$\begin{cases} R^2 &= 1 - \dfrac{\sum(y_{true} - y_{pred})^2}{\sum(y_{true} - y_{mean})^2} \\ MAE &= \dfrac{1}{n} \sum_1^n |y_{true} - y_{pred}| \end{cases} \tag{24}$$

where y_{true} is the value of the observed target, y_{pred} is the predicted value of the target and y_{mean} is the mean value of the observed targets. The model gets better and better when the R^2 score tends to 1 and the MAE score tends to 0.

A good practice in machine learning is to make sure our results are repeatable and extendable to unseen data. To achieve this, the k-fold cross validation is a commonly used method. Its first step consists of dividing our data set into k folds (hence the name). Then k-1 folds are used to train the model while the remaining one is used to test the model. After the evaluation of the model obtained that way, another fold is used for the testing data and the ones that are left are used for training. The same operation is repeated until all of the k folds have been used as a testing data set.

Five folds are used in this paper. For each of these folds, the R^2 and MAE scores are computed. Ultimately, we take the mean and standard deviation of these values to assess the performance of the model considered.

3.4 Length of stay prediction

The training and predictions are made based on the records from a sliding 12-hour window. First the model is trained over the time interval $[w_{beg}, w_{end}]$ with all of the training patients, the target of the regression being the remaining length of stay (RLoS) at $t = w_{end}$. To be more specific, the median value of each feature in this time window was taken to create a new record that would characterize the patient. Then we predict the RLoS of the test patients for a median record (created in the same way as for the training) in the same window. After that, we add 6 hours to w_{beg} and w_{end} and the same process is repeated until $w_{end} > w_{lim} = 200$ (according to our preprocessing). In the event of a stay ending at a time which is not a multiple of 12, special care has been taken to reduce the last window for it to end at the same time the stay ends.

Two models were implemented in this study. The first one is the famous linear regression (LR) and serves as a baseline. The second one is Gaussian Process regression (GPR). As it was mentioned in section 2, a likelihood and a kernel must be specified in order to perform GPR. Concerning the former, choosing a Poisson likelihood was the first idea as we are dealing with time dependent "sparse" data. The Gamma likelihood [5] could perhaps be considered. A perhaps less classical choice would have been the Weibull likelihood [5] which is commonly used for life data analysis. On the other hand, the Matern 3/2 kernel (which basically is a generalization of the squared exponential kernel) was the main object of our considerations in this study. This choice is questionable as there may be more suitable choices, and researching the best choice of kernel for our problem could be the object of a future work.

The performance of these model are given in Table 2. The 'Cat' column indicates if categorical data was used, in which case the value is set to 1. The 'Rep' column indicates how the categorical features were represented ('Int' is for the arbitrary integer representation, 'OHE' for one hot encoding and 'Emb' for embedding). The results shown for GPR are obtained with a Poisson likelihood and for a hundredth of the training data, the reason behind this being that the data set was too large for the dense methods of [10] to handle. This problem was taken notice of but because there was already issues with the baseline model (see next paragraph for more information), it was decided that it did not required immediate attention.

Model	Cat	Rep	R2 $\pm$ std	MAE $\pm$ std
LR	0	Int	0.016 $\pm$ 0.001	1.222 $\pm$ 0.009
LR	1	Int	0.014 $\pm$ 0.001	1.226 $\pm$ 0.009
LR	1	OHE	0.003 $\pm$ 0.002	1.232 $\pm$ 0.009
GPR	1	Int	-0.024 $\pm$ 0.006	1.221 $\pm$ 0.014
GPR	1	OHE	-0.029 $\pm$ 0.009	1.243 $\pm$ 0.017

Table 2: Prediction of RLoS in ICU - R2 and MAE scores

Let's first take a look at the performance of the LR model from table 2. Overall, the scores (especially the low R^2 score) are showing that the model performs very poorly on this data set, meaning that the RLoS can not be explained linearly. Such an outcome was expected (see [8] for example). However, one can see that the R^2 score is plummeting down as we add categorical features to the model and change their representation to OHE. This is totally unexpected and the root of this problem could not be find during this internship. Because the base of the code is the same for GPR, there is a high probability that it is for the same reason that incoherent negative R^2 scores are found. On the other hand, all of the MAE scores seems fine and in accordance with previous studies [8].

3.5 Conclusion

To conclude this section, we first presented the eICU [4] data set and how we preprocessed and extracted relevant information from this database. Then an attempt to predict the RLoS in ICU was made using both LR and GPR models. Unfortunately this attempt was unsuccessful and more work is needed in order to conclude on the efficiency of GP in this medical context.

4 Conclusion

First the reader was introduced to the GP framework. In particular, results on GP regression and GP classification were given and the Laplace approximation to the posterior was explained in detail. Then this knowledge was applied on real medical data from eICU [4] data set. An important part of the work consisted of preprocessing the data for it to be exploitable. After that preprocessing was completed, LR and GPR models were trained and tested on a sliding 12-hour window in order to make predictions on the RLoS. The results from this last part of the internship were not really convincing and it would take more time and work to resolve the issues encountered.

In future work, identifying and correcting the problem on the R^2 scores is what takes priority. Afterward, it would be nice to implement the embedding representation [12] of categorical features. As mentioned before, the choice of the likelihood and kernel for GPR is an interesting object of study. The choice of the approximation method to the posterior is also something that would require more thoughts to be put into.

This internship in Finland was a great occasion for me to learn more about a culture that I did not know much about. I also had the opportunity to be part of an international team which allowed me to broaden my horizons even more. On that note, I would like to express my thanks to my supervisor Aleksei Tiulpin as well as to all of my coworkers, be it for the help they may have provided me or the interesting conversations we had all over these three months.

5 References

[1] C. E. Rasmussen & C. K. I. Williams, "Gaussian Processes for Machine Learning", the MIT Press, 2006, ISBN 026218253X, © 2006 Massachusetts Institute of Technology, www.GaussianProcess.org/gpml

[2] Aleksei Tiulpin, University of Oulu's website, page titled "Intelligent Medical Systems", https://www.oulu.fi/en/research-groups/intelligent-medical-systems

[3] Marc Peter Deisenroth & A. Aldo Faisal & Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press, 2020

[4] Pollard TJ & Johnson AEW & Raffa JD & Celi LA & Mark RG & Badawi O, "The eICU Collaborative Research Database, a freely available multi-center database for critical care research", Scientific Data, 2018, DOI: http://dx.doi.org/10.1038/sdata.2018.178

[5] Johnathan Mun, "Advanced Analytical Models: Over 800 Models and 300 Applications from the Basel II Accord to Wall Street and Beyond" (Appendix C), 2008, https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119197096.app03

[6] Bishop Christopher M., "Pattern Recognition and Machine Learning", Springer, 2006

[7] Goldberger A. et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.", 2000

[8] Seyedmostafa Sheikhalishahi & Vevake Balaraman & Venet Osmani, Benchmarking Machine Learning Models on eICU Critical Care Dataset, 2019, arXiv:1910.00964v1

[9] Pedregosa et al., "Scikit-learn: Machine Learning in Python", JMLR 12, pp. 2825-2830, 2011

[10] W.J. Wilkinson & S. Särkkä & A. Solin, "Bayes-Newton Methods for Approximate Bayesian Inference with PSD Guarantees", 2021, arXiv preprint arXiv:2111.01721

[11] Harutyunyan H & Khachatrian H & Kale DC & Ver Steeg G & Galstyan A, "Multitask learning and benchmarking with clinical time series data", Scientific data, 2019

[12] Guo C & Berkhahn F, "Entity embeddings of categorical variables", 2016, arXiv preprint arXiv:1604.06737

6 Annex

6.1 Matrix Inversion Lemma

The matrix inversion lemma in its general form can be written as:

$$(Z + UWV)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + VZ^{-1}U)^{-1}VZ^{-1} \tag{25}$$

It is also known as the Woodbury matrix identity.

6.2 Conditional distribution

For x and y two jointly Gaussian vectors such that $\begin{pmatrix} y \\ f \end{pmatrix} = \mathcal{N}(\begin{pmatrix} \mu_y \\ \mu_f \end{pmatrix}, \begin{pmatrix} A & C^T \\ C & B \end{pmatrix})$, the conditional distribution of $y|x$ is also Gaussian so that

$$y|x \sim \mathcal{N}(\mu_y + CA^{-1}(f - \mu_f), B - CA^{-1}C^T) \tag{26}$$

Refer to [1] if further details are necessary.

6.3 Product of Gaussian

The product of two Gaussians distributions $x \sim \mathcal{N}(a, A)$ and $y \sim \mathcal{N}(b, B)$ gives the unnormalized Gaussian $z \sim Z^{-1}\mathcal{N}(c, C)$ where:

$$\begin{cases} c = C(A^{-1}a + B^{-1}b) \\ C = (A^{-1} + B^{-1})^{-1} \\ Z^{-1} = (2\pi)^{-n/2}|A + B|^{-1/2} \exp(-\frac{1}{2}(a - b)^T(A + B)^{-1}(a - b)) \end{cases} \tag{27}$$

6.4 Evaluation of the integral of the cdf times pdf of a Gaussian law

Let X and Y be two independent random variables such that $X \sim \mathcal{N}(\mu, \sigma^2)$ and $Y \sim \mathcal{N}(0, 1)$. We write ϕ the pdf and Φ the cdf of the standard normal distribution. We have

$$\begin{aligned} P(X \leq Y) &= \int_{-\infty}^{\infty} P(X \leq Y \mid Y = w)\phi(w)dw \\ &= \int_{-\infty}^{\infty} P(X \leq w)\phi(w)dw \\ &= \int_{-\infty}^{\infty} \Phi(\frac{w - \mu}{\sigma})\phi(w)dw \end{aligned}$$

By noticing that $X - Y \sim \mathcal{N}(\mu, \sigma^2 + 1)$, we have

$$P(X - Y \leq 0) = \Phi(-\frac{\mu}{\sqrt{1 + \sigma^2}})$$

Therefore, we have the relation:

$$\int_{-\infty}^{\infty} \Phi(\frac{w - \mu}{\sigma})\phi(w)dw = \Phi(-\frac{\mu}{\sqrt{1 + \sigma^2}}) \tag{28}$$

6.5 Preprocessing

Figure 8, taken from [8], is given here to better visualize the exclusion criteria used during the preprocessing of the data.

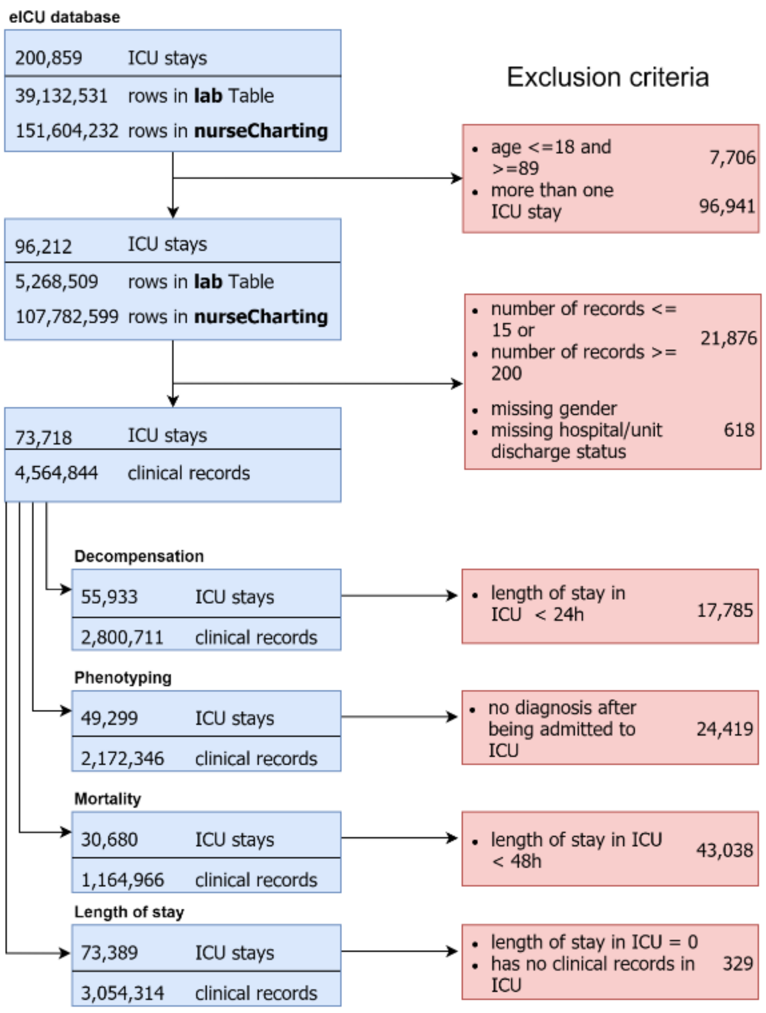
eICU database
200,859 ICU stays
39,132,531 rows in **lab** Table
151,604,232 rows in **nurseCharting**
Exclusion criteria
age <=18 and >=89 7,706
more than one ICU stay 96,941
96,212 ICU stays
5,268,509 rows in **lab** Table
107,782,599 rows in **nurseCharting**
number of records <= 15 or
number of records >= 200 21,876
missing gender
missing hospital/unit discharge status 618
73,718 ICU stays
4,564,844 clinical records
Decompensation
55,933 ICU stays
2,800,711 clinical records
length of stay in ICU < 24h 17,785
Phenotyping
49,299 ICU stays
2,172,346 clinical records
no diagnosis after being admitted to ICU 24,419
Mortality
30,680 ICU stays
1,164,966 clinical records
length of stay in ICU < 48h 43,038
Length of stay
73,389 ICU stays
3,054,314 clinical records
length of stay in ICU = 0
has no clinical records in ICU 329

Figure 8: Exclusion criteria used during the preprocessing