ICT for Health Laboratory
Regression using Gaussian processes – Lab #2

Monica Visintin

Politecnico di Torino

2023/24

Table of Contents

Table of Contents

Data preparation [1]

- As in Lab #1, load the Parkinson's disease dataset, remove the unwanted features, shuffle the data, get the training, and test datasets. Use the following features as regressors: 'sex', 'test_time', 'motor_UPDRS', 'Jitter(%)', 'Jitter(Abs)', 'Jitter:RAP', 'Jitter:PPQ5', 'Jitter:DDP', 'Shimmer', 'Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'Shimmer:APQ11', 'Shimmer:DDA', 'NHR', 'HNR', 'RPDE', 'DFA', 'PPE' (they are 19), and use 'total_UPDRS' as regressand.
- As a difference with respect to Lab # 1, the data must be divided into training, **validation** and test datasets, for example according to the percentages: 50%, 25%, 25%. Note that with such a division, you technically have 75% of training data and 25% of test data; the training data are further divided into validation data (33%) and true training data (66%).
- As in Lab # 1 find mean and standard deviation of each feature in the true training dataset (as if you did not know validation data and test data), and normalize the entire dataset using these means and standard deviations.
- As in Lab # 1 extract the regressand, i.e. total UPDRS, for the training, validation and test datasets.

Data preparation [2]

- The order of these 3 initial operations can be changed as you like, but the final result must be that you have a matrix `X_train_norm`, a vector `y_train_norm`, a matrix `X_test_norm`, a vector `y_test_norm`, a matrix `X_val_norm` a vector `y_val_norm`, all normalized, with obvious meanings. For simplicity, use Ndarrays, not Pandas dataframes.

Table of Contents

Goal [1]

We want to regress total UPDRS from the regressors (initially only motor UPDRS, age, PPE) using Gaussian processes:

$$Y = g(\mathbf{X}) + \nu$$

where $g(\mathbf{X})$ is the Gaussian process and ν the Gaussian measurement error.
The relevant formulas are the following:

- $N \times N$ covariance matrix $\mathbf{R}_{Y,N}$ in position n, k has value:

$$\mathbf{R}_{Y,N}(n,k) = \theta \exp\left(-\frac{\|\mathbf{x}_n - \mathbf{x}_k\|^2}{2r^2}\right) + \sigma_\nu^2 \delta_{n,k}, \quad n,k \in [1,N] \tag{1}$$

- $N \times N$ covariance matrix $\mathbf{R}_{Y,N}$ is written as

$$\mathbf{R}_{Y,N} = \begin{bmatrix} \mathbf{R}_{Y,N-1} & \mathbf{k} \\ \mathbf{k}^T & d \end{bmatrix}$$

Goal [2]

- The pdf of Y_N for know values $\mathbf{x}_N$, given the measured values $\mathbf{y} = [y_1, \ldots, y_{N-1}]$ and corresponding regressors $\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}$ is

$$f_{Y_N|\mathbf{y}}(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

$$\mu = \hat{y}_N = \mathbf{k}^T \mathbf{R}_{Y,N-1}^{-1} \mathbf{y} \tag{2}$$

$$\sigma^2 = d - \mathbf{k}^T \mathbf{R}_{Y,N-1}^{-1} \mathbf{k} \tag{3}$$

- In the above equations, couples $(\mathbf{x}_n, y_n)$ for $n = 1, \ldots, N-1$ belong to the training dataset, couple $(\mathbf{x}_N, Y_N)$ belongs to the test or validation datasets. Note that Y_N is the random variable whose pdf we want to estimate, y_N is its true value, $\hat{y}_N$ is its estimated value.
- **Hyperparameters θ, r, σ_ν^2 must be found using the validation dataset, i.e. using $(\mathbf{x}_N, Y_N)$ from the validation dataset.**

Initial procedure to check that the system works [1]

In the model, initially set the autocorrelation hyperparameters $r^2 = 100$ (call it `r2` in the Python script), $\theta = 1$ and $\sigma_\nu^2 = 0.001$ (call it `s2` in the Python script). Moreover, set in Python `N=10` (`N` has the same meaning as N in the slides). You will change the value of `N` later on, do not "hardcode" 10 in your script, use the identifier `N`.

1. Pick all the rows of `X_val` and perform the following steps for each of the rows in the validation set. Let `x` be the k-th row of `X_val_norm` with normalized UPDRS equal to `y_val_norm[k]`

```
x=X_val_norm[k,:]
```

2. Find the $N-1$ points in the training dataset that are closer to `x`. Search the web on how to do this, or ask ChatGPT, or do it on your own.
3. Generate a matrix $\mathbf{X_r}$ with the $N-1$ closer points in rows from 0 to $N-2$ and `x` in the last row $N-1$.
4. Build the covariance matrix $\mathbf{R}_N$ using Eqn. (1), where $\mathbf{x}_n$ and $\mathbf{x}_k$ are the n-th and k-th row of $\mathbf{X_r}$.

Initial procedure to check that the system works [2]

5. Find the estimated normalized value of total UPDRS for `x` using GP regression (Eqn. (2) for the value and Eqn. (3) for the standard deviation).
6. Generate the (normalized) UPDRS value corresponding to `x`, i.e. the estimate of `y_val_norm[k]`.
7. Once you have processed all the validation points, verify that you get reasonable results by plotting, as usual, the estimate of `y_val_norm` versus `y_val_norm` (regression line) or the estimate of `y_val` versus `y_val` (using denormalization, as done in the previous lab).

Table of Contents

Procedure to set the hyperparameters [1]

- Note that, since the dataset was normalized, the autocorrelation $R_Y(\mathbf{0})$ (i.e. the values on the main diagonal of the covariance matrix) are equal to 1 and it is not necessary to optimize this value (i.e. $\theta = 1$ is correct).
- Parameters to be set are: `r2` (r^2) and `s2` (σ_ν^2). They must be set so that **the mean square value of `y_val-yhat_val` (i.e. the validation mean square error) is minimized**. Equivalently you can minimize the mean square value of `y_val_norm-yhat_val_norm`. Use a set of values for `r2`, a set of values for `s2`, measure the validation mean square error, find the minimum (this procedure is called **grid search**).
- You are responsible of finding a reasonable grid. Don't ask the instructor about the range that you must consider for `r2` and `s2`, use common sense, find a solution, you are a master student.

Procedure to set the hyperparameters [2]

- In the overall you must find reasonable values of the hyperparameters

  `r2, s2.`

  keeping $N = 10$ and the suggested regressors.
- See what happens if `N` is changed: we cannot use `N` equal to the number of datapoints in the training set (about 500, the covariance matrix would be too large), but `N=10` might be too small.

Table of Contents

Test

Once `r2, s2` have been optimized for the validation dataset, measure the following for the **test dataset** and for the **validation dataset**:

1. the mean, the standard deviation, the mean square value of the estimation error, the correlation coefficient for the un-normalized total UPDRS
2. the histogram of the un-normalized estimation error
3. the plot of the estimated total UPDRS versus the true one with the errorbars (use 2 times the value of `sigma` generated by your code, but remember to un-normalize this value)
4. the value of R^2

so that you can compare the results obtained with the Gaussian process regression with the results obtained with linear regression LLS in Lab #1.
Note that each time you run your code you might get different results, if shuffling of the data is random. You can set the seed to be able to exactly reproduce the script and find an error in your script, if it helps, but your Python code should work (reasonably) for any seed.
In the oral exam you might be asked to run (on your PC) and comment the results of this laboratory.