Human Age Prediction Based on Real and Simulated RR Intervals using Temporal Convolutional Neural Networks and Gaussian Processes

Maximilian Pfundstein (maxpf364)

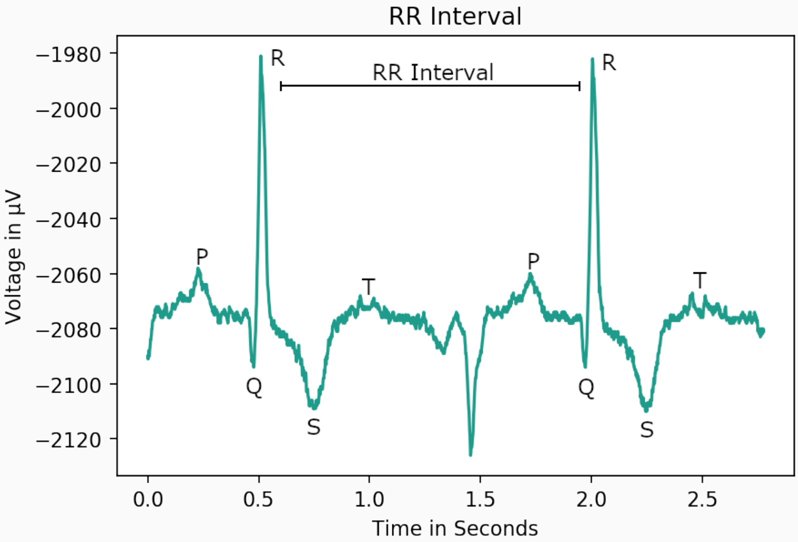
April 29, 2020

Linköpings University

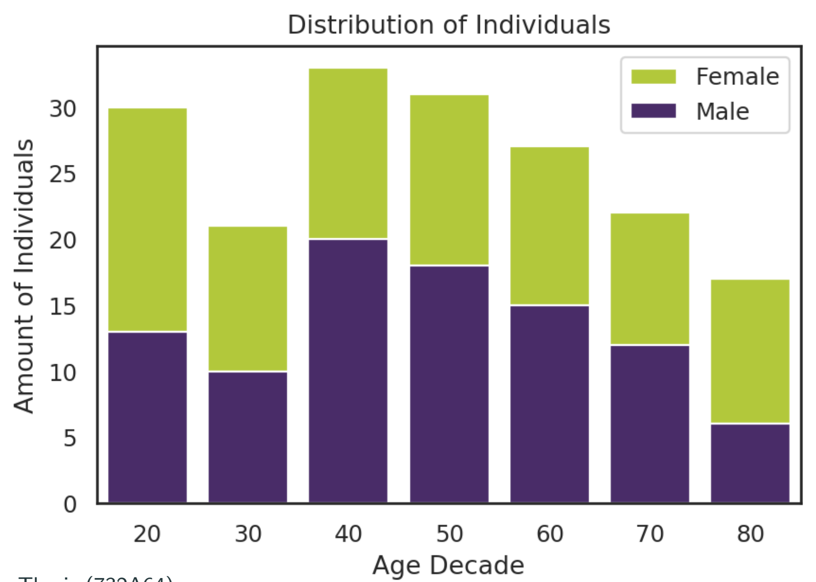
Table of contents

Recap

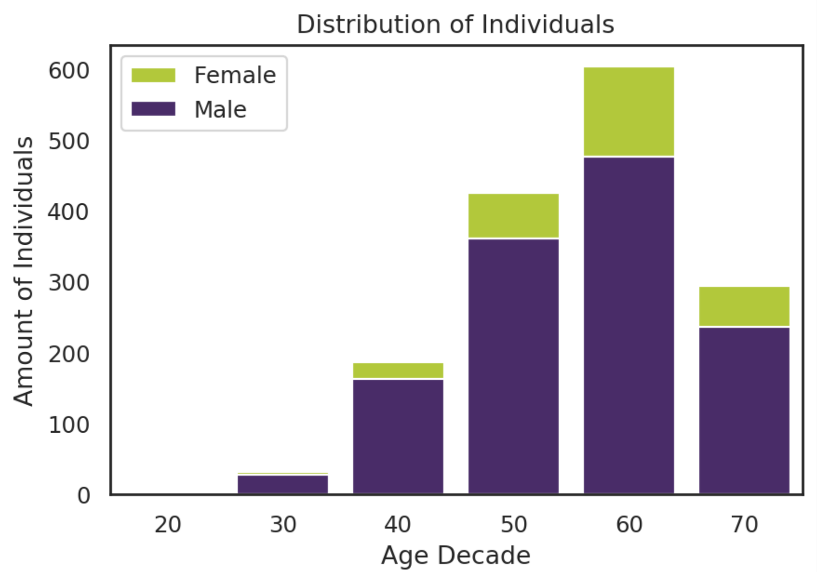
Recap — Age Prediction



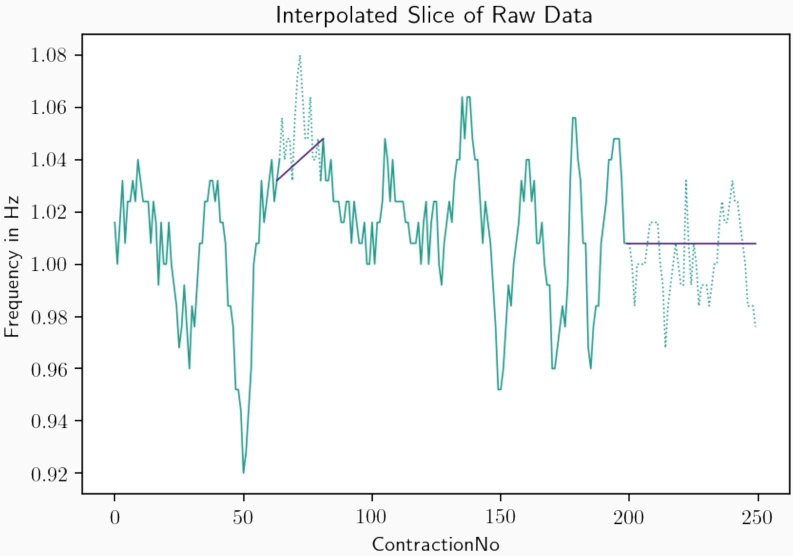
Recap — Data Sets — Gdańsk



Recap — Data Sets — PhysioNet



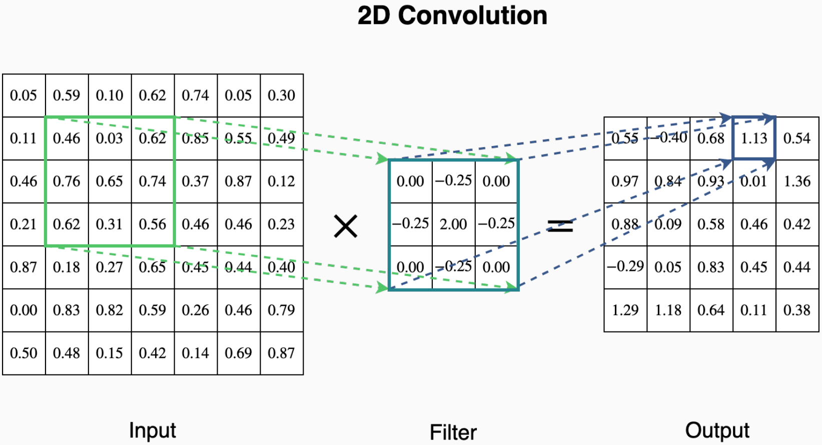
Recap — Impurity and Padding



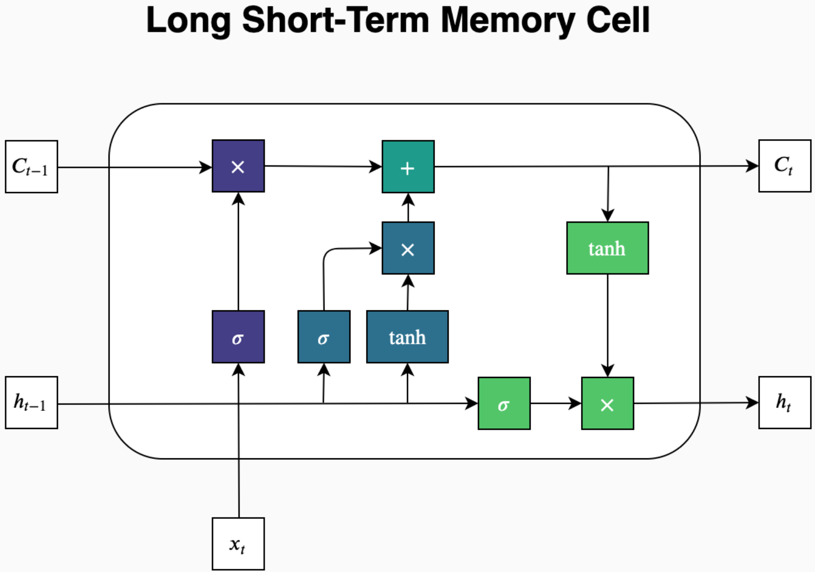
Methods and Models

- *complete* and *constant*
  - 48 slices of around 5 minutes
- *classification* and *regression*
  - classes have no order
- 33 features as used in a previous paper (`hrvanalysis`)

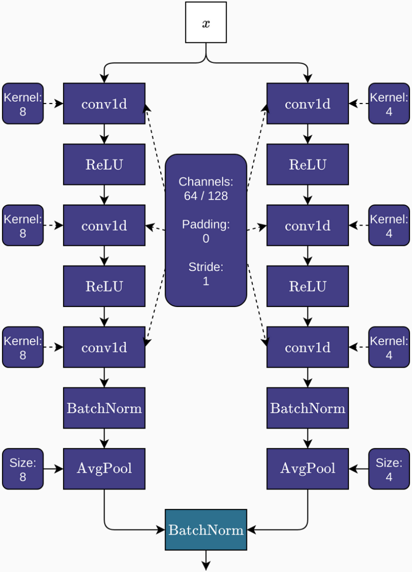
- Naive Bayes
- Support Vector Machines
  - Hyperparameters
    - Classification: γ_C (kernel length scale), C_C (regularisation)
    - Regression: γ_R (kernel length scale), C_R (regularisation), ϵ_R (slack)
  - 5 fold cross-validation
- Random Forest
- XGBoost



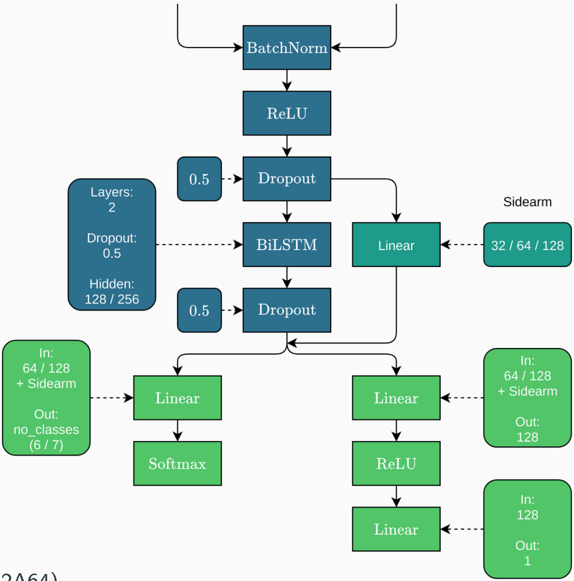
2D Convolution
0.05 0.59 0.10 0.62 0.74 0.05 0.30
0.11 0.46 0.03 0.62 0.85 0.55 0.49
0.46 0.76 0.65 0.74 0.37 0.87 0.12
0.21 0.62 0.31 0.56 0.46 0.46 0.23
0.87 0.18 0.27 0.65 0.45 0.44 0.40
0.00 0.83 0.82 0.59 0.26 0.46 0.79
0.50 0.48 0.15 0.42 0.14 0.69 0.87
×
0.00 −0.25 0.00
−0.25 2.00 −0.25
0.00 −0.25 0.00
=
0.55 −0.40 0.68 1.13 0.54
0.97 0.84 0.93 0.01 1.36
0.88 0.09 0.58 0.46 0.42
−0.29 0.05 0.83 0.45 0.44
1.29 1.18 0.64 0.11 0.38
Input
Filter
Output


Long Short-Term Memory Cell
C_{t-1}
×
+
C_t
tanh
×
σ
σ
tanh
h_{t-1}
σ
×
h_t
x_t

Methods and Models — DeepSleep



Methods and Models — DeepSleep



Results

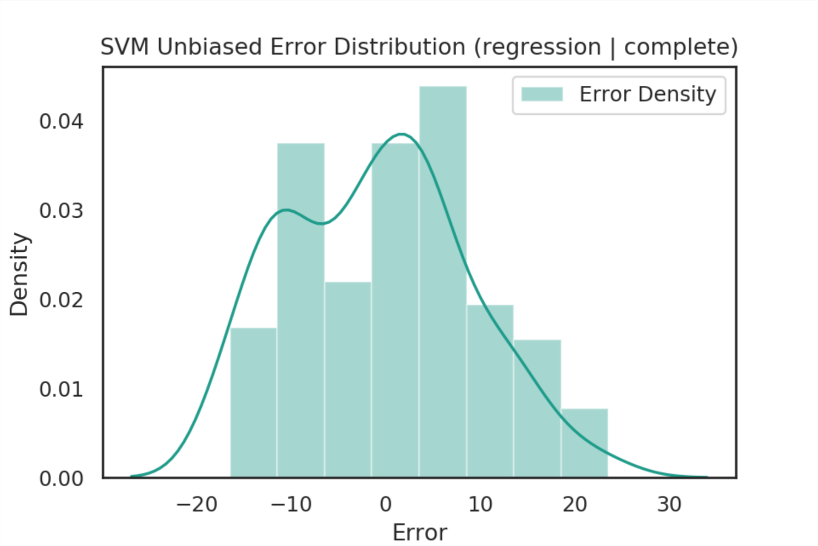
Results — Gdańsk

Gdańsk	Naive Bayes	Random Forest	SVM	XGBoost	DeepSleep
Regression / Complete					
Training		57.64%	13.19%	32.64%	18.52%
Validation					19.44%
Testing		27.03%	**32.43%**	29.73%	24.32%
Regression / Constant					
Training		71.35%	13.19%	24.35%	15.74%
Validation					11.11%
Testing		25.84%	**32.43%**	23.25%	29.73%
Classification / Complete					
Training	32.64%	100.00%	20.14%	100.00%	22.22%
Validation					13.89%
Testing	29.73%	**32.43%**	10.81%	27.03%	10.81%
Classification / Constant					
Training	21.53%	100.00%	18.75%	99.31%	22.22%
Validation					13.89%
Testing	27.03%	**32.43%**	5.41%	27.03%	10.81%

Results — Physionet

Physionet	Naive Bayes	Random Forest	SVM	XGBoost	DeepSleep
Regression / Complete					
Training		76.44%	40.20%	27.09%	25.36%
Validation					29.87%
Testing		35.48%	29.68%	**38.71%**	30.32%
Regression / Constant					
Training		78.64%	11.74%	27.62%	39.79%
Validation					42.20%
Testing		30.04%	29.03%	29.56%	32.90%
Classification / Complete					
Training	15.35%	99.78%	39.91%	99.78%	39.55%
Validation					37.66%
Testing	12.90%	29.03%	25.81%	27.10%	**38.10%**
Classification / Constant					
Training	13.62%	99.78%	27.55%	51.22%	39.38%
Validation					33.77%
Testing	12.65%	33.68%	28.13%	33.94%	**42.58%**

Results — Physionet



Data Simulation

Data Simulation — Gaussian Processes

A Gaussian Process is defined as

$$f(x) \sim \mathcal{GP}\big(m(x), k(\chi, \chi)\big) \tag{1}$$

The whole statistical model is then defined as

$$y = \mathcal{N}\big(f(x), \sigma_{\text{noise}}\big) \tag{2}$$

Data Simulation — Gaussian Processes

The joint of observations and points to predict is defined as

$$\begin{bmatrix} y \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(m(x), \begin{bmatrix} k(\chi, \chi) + \sigma^2_{\text{noise}} I & k(\chi, \chi^*) \\ k(\chi^*, \chi) & k(\chi^*, \chi^*) \end{bmatrix} \right) \tag{3}$$

Data Simulation — Gaussian Processes

Then we condition on the observations to obtain the predictive distribution

$$f(x^*)|\chi^*, \chi, f(x) \sim \mathcal{N}\big(\bar{f}_*, \mathrm{cov}(f_*)\big) \tag{4}$$

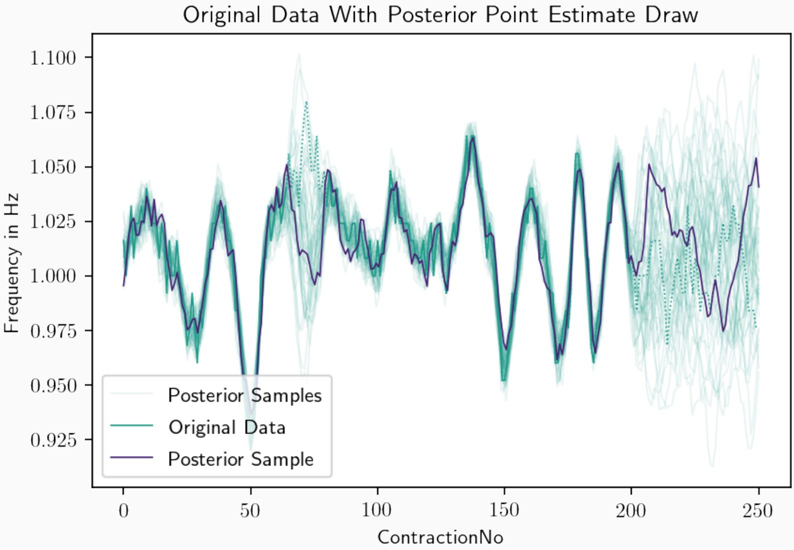
$$\bar{f}_* = k(\chi^*, \chi)\big(k(\chi, \chi) + \sigma^2_{\text{noise}} I\big)^{-1} y \tag{5}$$

$$\mathrm{cov}(f_*) = k(\chi^*, \chi^*) - k(\chi^*, \chi)\big(k(\chi, \chi) + \sigma^2_{\text{noise}} I\big)^{-1} k(\chi, \chi^*) \tag{6}$$

Optimise the log likelihood proportion, with respect to each parameter in θ

$$\mathcal{L}(y|X,\theta) \propto -\frac{1}{2}y^T\left(K_\theta + \sigma^2_{\text{noise}}I\right)^{-1}y - \frac{1}{2}\log|K_\theta + \sigma^2_{\text{noise}}I| \quad (7)$$

Data Simulation — GP — Point Estimates



Data Simulation — GP — Posterior Predictive Distribution

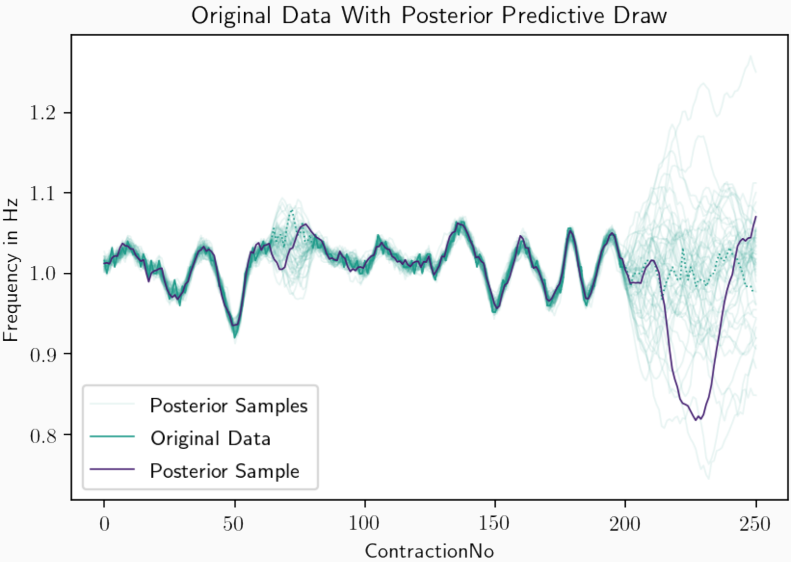
The posterior predictive distribution is given as

$$p(\tilde{y}|y) = \int_{\theta} p(\tilde{y}|\theta, y) p(\theta|y) d\theta \tag{8}$$

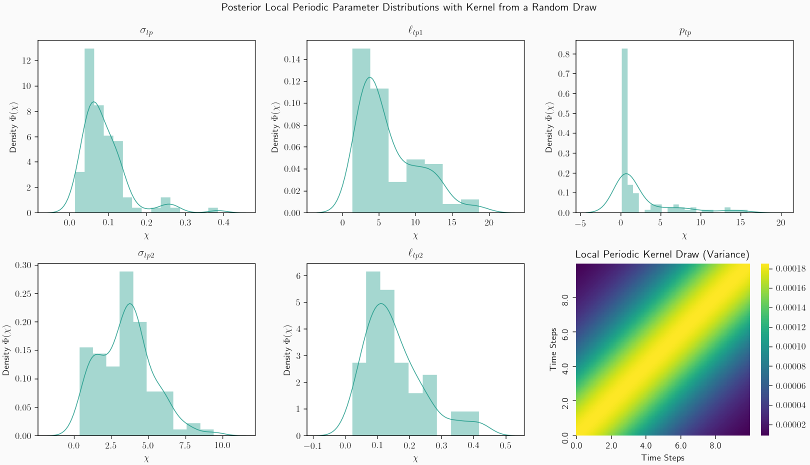
Update parameters for each slice, as

$$p(\theta|\chi_a, \chi_b) \propto p(\chi_b|\theta) p(\theta|\chi_a) \tag{9}$$

Data Simulation — GP — Posterior Predictive Distribution



Data Simulation — GP — Posterior Predictive Distribution



Status and Outlook

Status and Outlook

- Bayesian simulation is currently running
- For comparison: point estimates
    - Numerical solution already implemented, but takes a lot of time
    - Maybe solving analytically
- Evaluate models on simulated data
- Analyse results

Appendix

References i