

train_predict

November 12, 2021

```
[1]: import os
import pickle

from forecasting.forecast import PatientForecast
from forecasting.utils import (
    FORECAST_HOURS, # 24
    HISTORIC_HOURS, # 168
    HOURS_IN_WEEK, # 168
    START_FORECAST, # 21/06/2021 00:00
)
```

1 Time series model

In order to inform the bed allocation process we developed a demand predictor to forecast the number of patients due to be admitted into the hospital. We adopted a Bayesian modelling approach using numpryo and jax. The fundamental idea behind this approach is that the model itself is considered to be a statistical object. In particular, the model parameters that define the model are interpreted as being drawn from a distribution defined using Bayes' Theorem. The output to the model is then a full posterior distribution for the predicted number of admissions rather than a single number. This allows us to draw samples for the predicted number of admissions and generate confidence intervals over our estimates.

The admissions forecast model consisted of four components which account for: - The long term trend: to account for variations such as the current rise in admissions after the downturn during COVID-19 lockdowns; - The day of the week: to account for variations connected with specific days in the week, e.g. admissions are typically lower on the weekends; - The hour of the day: to account for variations connected with the time of day, e.g. fewer patients are admitted at night than during the day; - Whether it is a bank holiday: to account for variations that are due to these specific dates, e.g. fewer patients are admitted on bank holidays.

We captured the long term trend in the historic admissions data using a Gaussian Process (GP), inspired by previous work of Vehtari et al., as summarised in this [blog post](#). A GP does not constrain the model to take on any particular form. Instead, it returns a distribution over the functions which are consistent with the observed data, in this case, the historic admissions timeseries. In practice, exact GPs can be inefficient to calculate, we therefore utilised the Hilbert Space approximation based on [this numpyro tutorial](#) to ensure tractable runtimes.

```
[2]: # Initialise class
forecast_model = PatientForecast()
```

2 Train model

We trained the model on previous admissions data, aggregated to show the total number of admissions (medical and surgical) every hour for the past 120 days. We chose this as our training period because it allows enough time for both the short and long term trends to show up within the data, whilst reducing the effect of the abnormalities in admission caused by COVID-19. Once the model has been trained, we can then use it to predict the number of patients that will be admitted each hour. Rather than a single number, the demand predictor produces a posterior distribution, meaning we can draw samples for the predicted number of patient admissions. We can also use the posterior to generate confidence intervals for our model. Here we use the 95% confidence interval, which means that 95 times out of 100 we expect the patient admission numbers to fall within the range provided.

```
[3]: # Train model, with training period ending at START_FORECAST
forecast_model.train_model(START_FORECAST, training_hours=2880)
```

```
0%|          | 0/4000 [00:00<?, ?it/s]
0%|          | 0/4000 [00:00<?, ?it/s]
0%|          | 0/4000 [00:00<?, ?it/s]
0%|          | 0/4000 [00:00<?, ?it/s]
```

3 Generate predictions

```
[4]: # Get prediction for 1 week before START_FORECAST and 24hrs after a week after
      ↪ START_FORECAST
results = forecast_model.call_forecast(
    START_FORECAST,
    historic_hours=HISTORIC_HOURS,
    forecast_hours=HOURS_IN_WEEK + FORECAST_HOURS,
)
```

4 Save results

```
[5]: # results is then saved as forecast_results.pkl
      #pickle.dump(results, open("../data/forecast_results.pkl", "wb"))
```