Cognitive Human Robot Interaction

Forecasting a human's working time based on previous task times

Surya Teja Venteddu, Rishab Manoj

December 26, 2017

1 Problem Statement

We are looking at an industrial setting in which there are multiple humans that are being served by a single robot. Robot's task is to predict the human's task time in future and optimally serve the requests of the humans to increase the total number of jobs and decrease the waiting time. This problem can be subdivided into three parts: **Data Generation**, **Model Estimation**, **Policy Simulation**. Each of these sub problems are addressed in the following sections respectively.

2 Data Generation

To forecast future task times we need the previous task times. We are simulating the data using python.

Assumptions with which the task times data was generated were:

- There is a system (Generation function) that determines the mean of amount of time taken by a person when he starts the task at time t.
- It is assumed that the human's task times will show the same patterns for every shift between the breaks. That's why the Generation function is periodic.
- The data that is going to be used is sampled from the generation function. Ideally, this function needs to be sampled at points where human starts the task time. But, we are sampling it for every second. This is based on the assumption that if we take large amount of data, both sample will depict the same model.

For simplicity, we are assuming the generation function is seasonal with a linear trend in each season for our current simulation.

$$\mu = \mu_0 + c_1 * (t\%\mathcal{T}) \tag{1}$$

where

μ *is mean of the normal distribution that's used for generation of process time at time t*

$\mathcal{T}$ *is the time period of the generation function*

μ_0, c_1 *are constants*

Therefore the task time T_t is generated from this the time varying normal distribution generated by change in μ which varies with time according to the equation 1.

$$T_t = \mathcal{N}(\mu, \sigma^2) \tag{2}$$

where
T_t = Task time generated by time varying normal distribution at time t
σ = The variance of the Distribution

μ_0, σ, c_1 values are different for each person and are dependent on the person's working abilities and characteristics. Suppose a person gets better after each task during a shift, the task time of that person will reduce with respect to time. therefore the mean time decreases with time. This means that c_1 will be negative for that person.

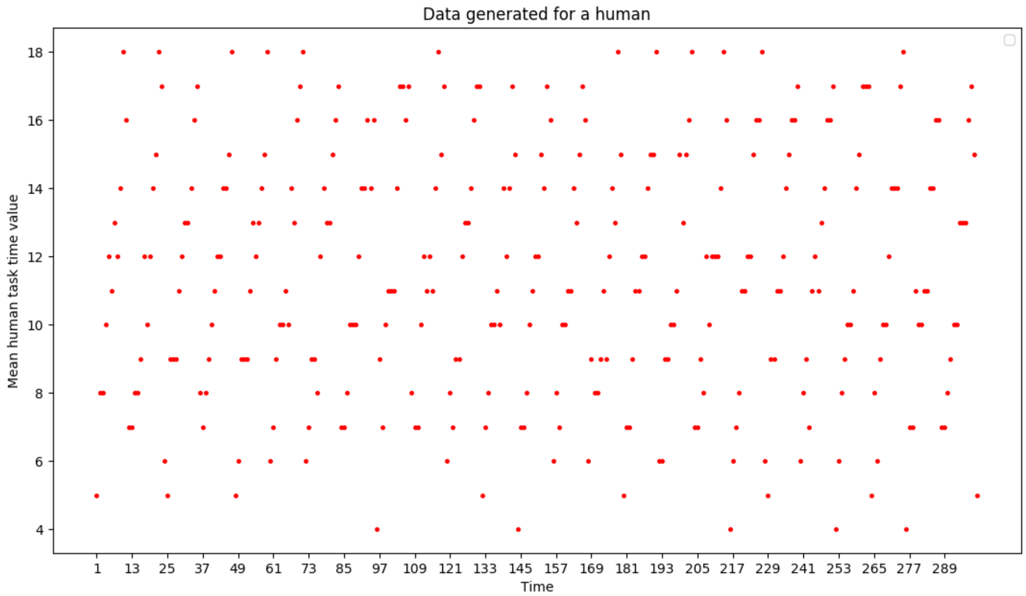

Figure 1: Sample data generated with $\mathcal{T} = 12, \mu_0 = 6, \sigma = 1, c_1 = 1$

Similar to c_1, μ_0 also varies with person;s capability. μ_0 basically decides how good is the person at doing the task. If he is very good at the task, μ_0 will be very low. σ captures how consistent the guy is in doing the task. Since we are fitting a simple linear function. We are implicitly assuming that the person can either get better with time or get worse with time at a constant rate.

3 Model Estimation

From the data generated above, we need to fit a model for forecasting in future. We divided the data into training and test using 7:3 ratio respectively. We have selected 70 percent of data to fit a prediction model and 30 percent of data was forecasted. We tried to fit an ARIMA model onto the data.

Data Exploration and parametre estimation

Auto Correlation Function and Partial Auto Correlation Function plots for the data generated above are as follows. These plots are useful to determine the parametres of the prediction model that is to be fit on the train data.

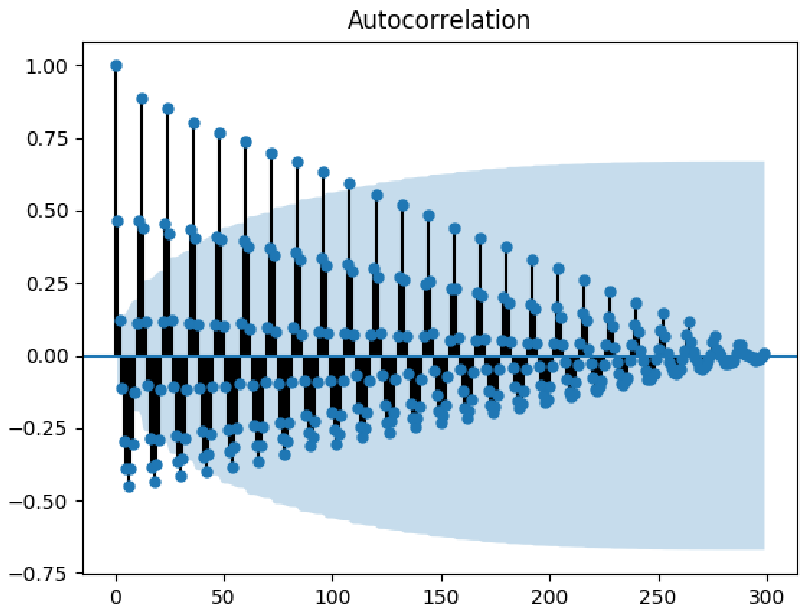

Figure 2: ACF plot of Sample data generated with $\mathcal{T} = 12, \mu_0 = 6, \sigma = 1, c_1 = 1$

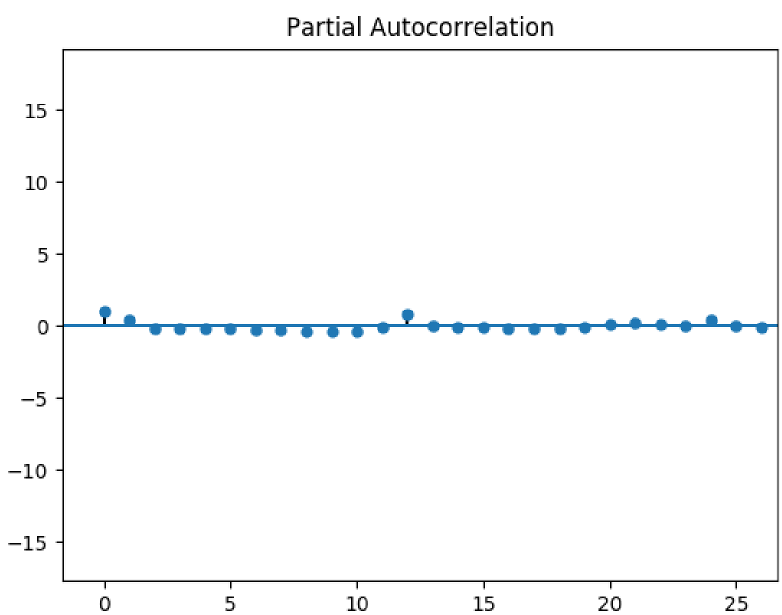

Figure 3: PACF plot of Sample data generated with $\mathcal{T} = 12, \mu_0 = 6, \sigma = 1, c_1 = 1$

Fitting ARIMA(12,1,1) model

There are 3 parametres in the ARIMA(p,d,q) model. p decides the pattern in a season and how many lag

Since the data is time series data and also depends on the previous task times of a person, we tried Auto Regressive Integrated Moving Average (ARIMA) model onto the generated data. In the auto correlation plot we can see that there is an exponential decrease in the tailing off of the autocorrelation. So, we need to decide the parametre p of the model based on the partial auto correlation plot. Since there is a spike in the PACF plot at every 12 intervals and it is decreasing with time, we select p=12 and d=1 since first differencing of linear data is stationary. The MA model is selected based on the variance of data. We selected q = 1.

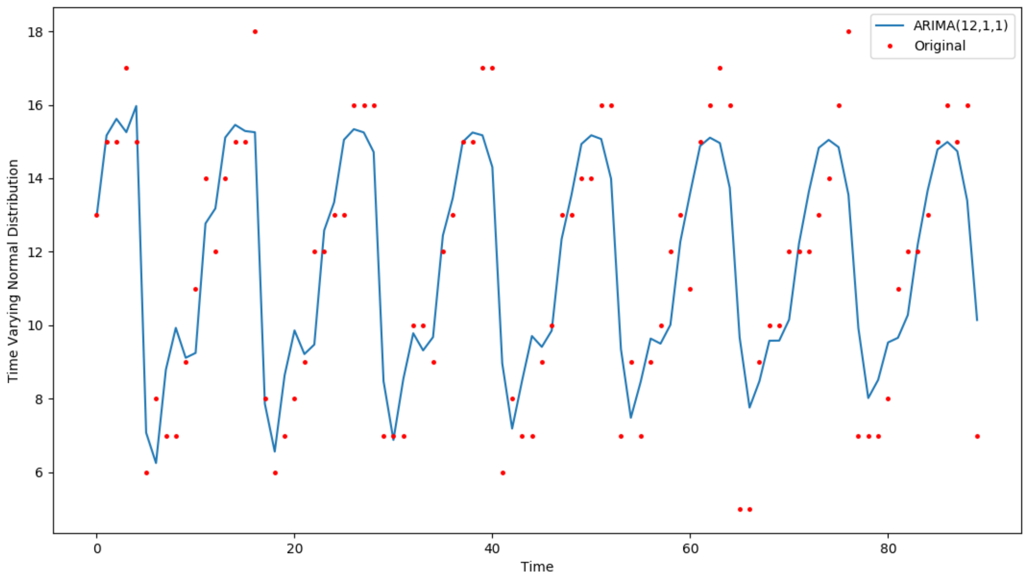

Figure 4: ARIMA fit for Sample data generated with $\mathcal{T} = 12, \mu_0 = 6, \sigma = 1, c_1 = 1$

This is a simple ARIMA model, Fitting this model took a long time(307.644s) since it doesnot ask for seasonality parametre and use it for benefit.

Fitting SARIMAX(1,1,0,1,0,1,12) model

Whereas, seasonal ARIMA model uses the benefit of seasonality of the data. SARIMAX model in python was used for this purpose and it was able to generate the following fit in very less time(5s). New parametres for seasonal ARIMA model were P,D,Q which decide trend for the seasons. Since all the seasons are same D =0, P=1. We use Q=1, to account for the varience in the data.

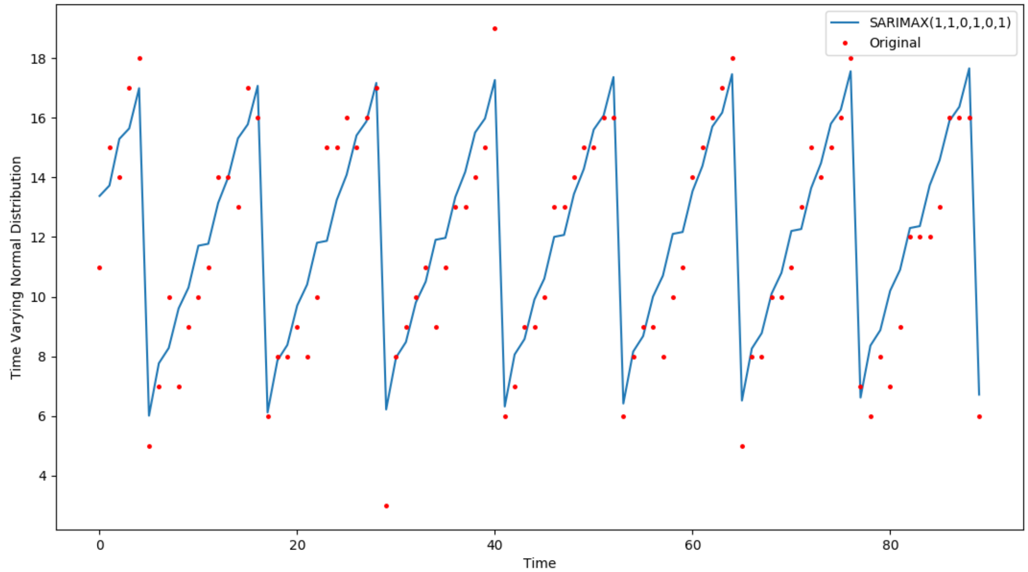

Figure 5: SARIMAX fit for Sample data generated with $\mathcal{T} = 12, \mu_0 = 6, \sigma = 1, c_1 = 1$

4 Policy Simulation

We made a simulation of humans and a robot working together on a task to finally estimate the number of jobs done by each of the human and the waiting time of each of the human. This simulation allows the robot to choose a mechanism for scheduling the humans. We have simulated the jobs done by 5 humans.

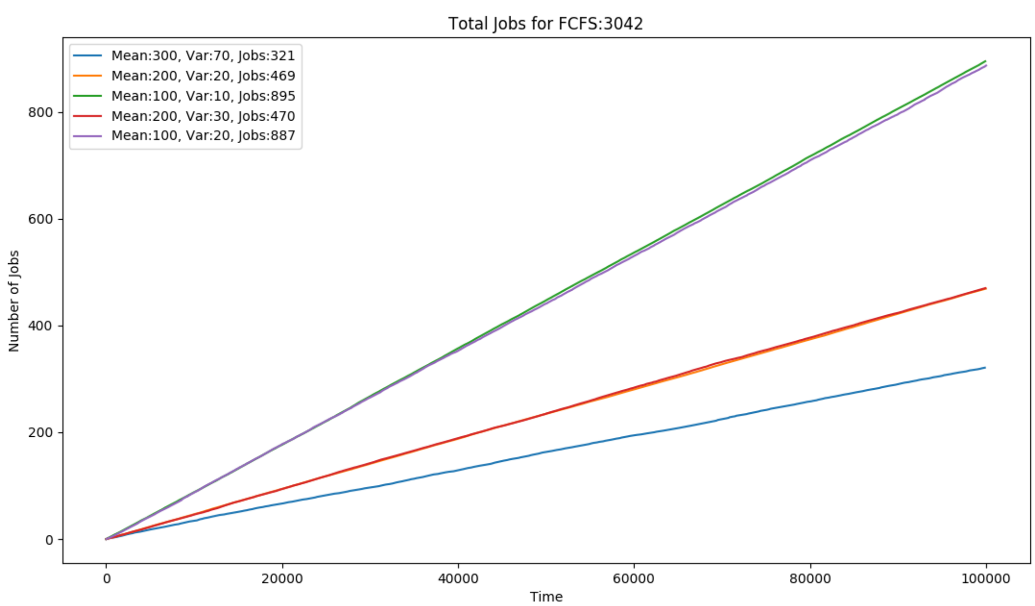

Figure 6: Total number of jobs done for humans over time with FCFS

whereas the humans put requests when a job needs to be done and the robot serves the requests. Here robot uses the First Come First scheduling to serve the humans. FCFS scheduling is not the most optimal scheduling policy, SJF is closer to the optimal policy than FCFS

Shortest Job First Policy

The scheduling policy that can use the advantage of the prediction is Shortest Job First algorithm. In Operating Systems, prediction of a process's CPU burst time is what is necessary for SJF to work. Random predictions are generated in this part of simulation to simulate the SJF policy. The following is the figure showing the jobs done by humans with SJF scheduling policy.

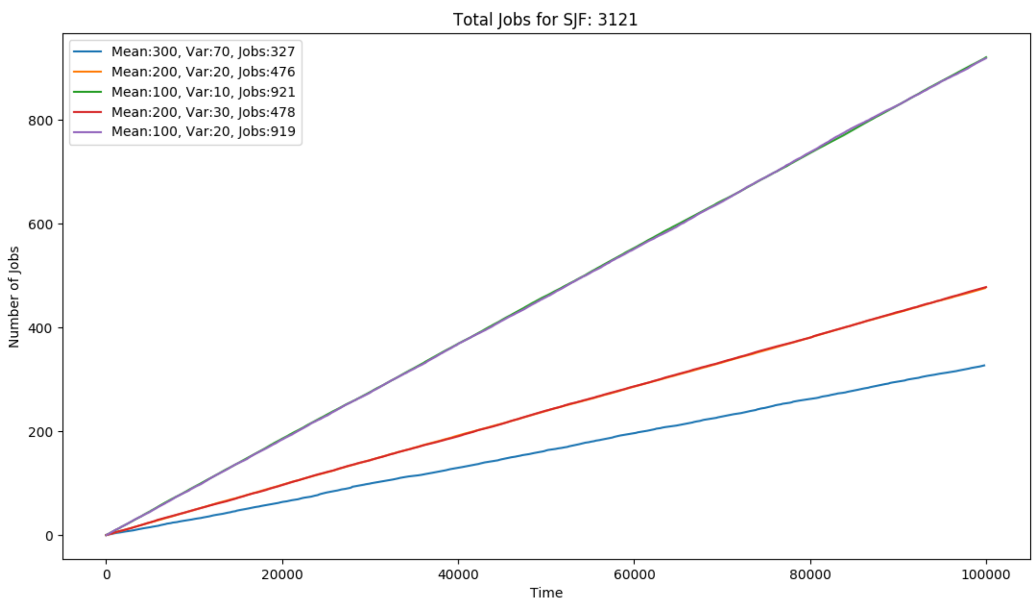

Figure 7: Total number of jobs done for humans over time with SJF

5 Conclusion and Future Work

In the previous part, the predictions for processing time of the human are generated using a random distribution. We are yet to integrate ARIMA model with SJF policy, then predictions will be generated by using the ARIMA model. With these predictions we hope that the results of number of jobs done will increase and the waiting time of the human is decreased.