Project Report: Missing Value Prediction for IoT Sensor Data
CP 218: Theory and Applications of Bayesian Learning

By Team: Bayesians
Susmit Das – 21717, Akshay Dixit – 21654, Hrithik Singh – 20857

Abstract

In this project, we aimed to predict missing values of temperature and pressure in a given dataset of IoT sensor data using various machine learning models. We first pre-processed the data from each node and separated them into sessions based on time difference. Then we compiled the data from all nodes to create a combined dataset, which stores the timestamps and the corresponding temperature and pressure in every node. We trained several models on session datasets and the combined dataset, including Linear Regression, Bayesian Ridge Linear Regression, Bayesian Ridge Polynomial Regression, Gaussian Process Regression in a defined neighbourhood of each missing value and studied the individual strengths and Limitations of each model. Finally, we used a probabilistic ensemble learning model with adaptive weights combining the outputs of many of the previous models, which made use of the strengths of each model, while minimizing each other's limitations and provided the highest accuracy and a Root Mean Square Error (RMSE) of 0.83 on public test data.

Introduction

First, to understand the dataset that we had been provided with, we started by exploring it and making observations.
Our primary observations:

1. In the dataset, we have seen that environment monitoring did not happen in a single stretch but in various separate sessions with significant time lapses between each session. In Figure 1 below, we can see the presence of three separate sessions, as seen in the difference in time.

5576	10/12/0014 21:55	22.25	42.46	
5577	10/12/0014 22:06		41.51	
5578	10/12/0014 22:16	22.25	41.98	Session 1
5579	15/12/0014 01:42		41.2	
5580	15/12/0014 01:52	21.29	41.2	
5581	15/12/0014 02:02		41.04	Session 2
5582	16/12/0014 03:32	31.29	9.51	
5583	16/12/0014 03:42	30.96	8.403	
5584	16/12/0014 04:44	30.32	8.72	Session 3

Figure 1

2. The presence of global trends between the features (humidity, temperature and time) is weak, i.e. when using the entire dataset as a whole, we don't find consistent relations between the features as seen in Figure 2(a). However, when we cut out small sections of the dataset, we can obverse strong local trends as shown in Figure 2(b).

Figure 2(a)

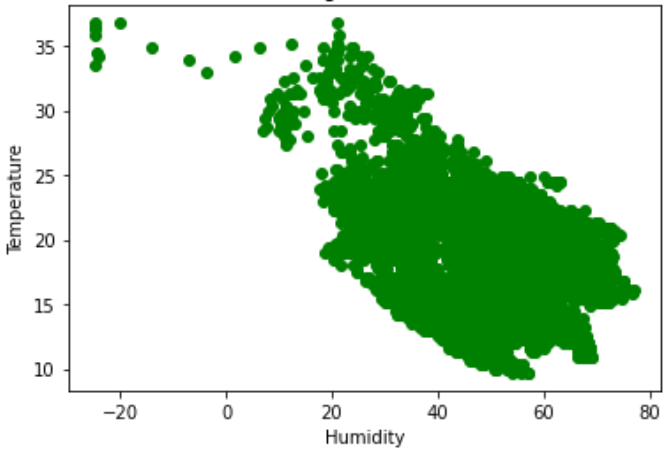
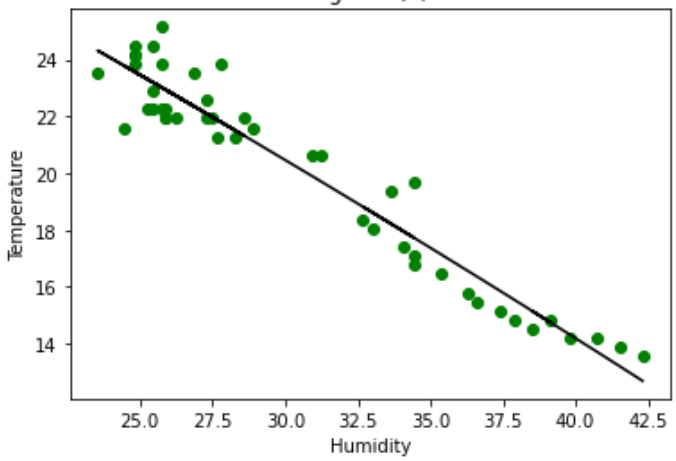


Figure 2(b)



Data Pre-processing

1. First, we converted the timestamps to date-time format and then into integer as minutes past since a fixed time. Then we used these modified timestamps into split the dataset of each nodes into separate session datasets when there was gap of more than 120 minutes between two adjacent readings.
2. We compiled all the timestamps of the nodes into a combined dataset and for each timestamp, we stored the humidity and temperature in every node within a time period of timestamp ±1/2 sampling duration, which is minutes. Now we have a combined dataset, which we can use to predict missing values in one node using the other nodes.
3. We made a copy of the session datasets and the combined dataset, dropped all the rows with the original missing values. Then we randomly removed new values from humidity and temperature, while storing the actual values in a different file to calculate of RMSE of our prediction models. We will use this as test data.

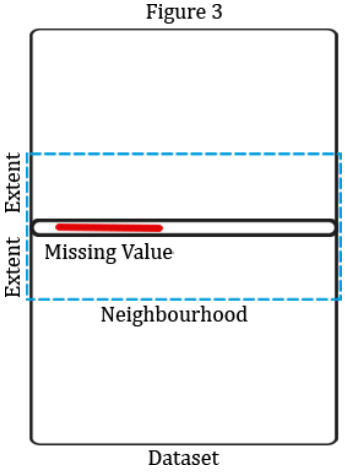
We are now ready to train models to predict the missing values.

Methodology

As seen in observation 2, we have strong local trends and very weak general trends in the data. Hence, for each missing value, we will train models in the **neighbourhood** of each missing value to make predictions instead of the using the entire dataset as shown in Figure 3. The **size of the neighbourhood is defined by its extent, which is a hyper-parameter we are using for each model**. The neighbourhood data we use to train the model will be from row index - extent to row index + index. For example, if there is a missing value in row 50 of a session dataset and we are using of an extent of 10, the neighbourhood would be from row 40 to row 60.

Sometimes, there can be other missing values in the same neighbourhood. So, we drop all rows with missing value in the neighbourhood and use the remaining data to train models to predict the missing value.

Figure 3



Models Tested

1. **Linear Regression** (**Y** – Objective Variable with Concerned Missing Value, **X** – Other Predictor Variable | **For e.g.**, if Temperature is missing, it becomes Y and Humidity becomes X and vice versa) – **Best RMSE: 1.02**
   Strengths observed:
   - Very fast to compute.
   - Worked reasonably well for most missing values when using small extents.

   Limitations observed:
   - Struggled in neighbourhoods without clear linear relationships.
   - Generated absurd predictions when available data in neighbourhood was low after dropping rows with missing values, resulting in increase in RMSE.
   - Susceptible to overfitting, hence not robust.
2. **Bayesian Ridge Linear Regression** (**Y** – Objective Variable with Concerned Missing Value, **X** – Other Predictor Variable) – **Best RMSE: 0.98**
   Strengths observed:
   - Worked very well when using small extents.
   - Worked better than regular linear regression when available data in neighbourhood was low.

   Limitations observed:
   - Gave worse predictions than normal linear regression in neighbourhoods without linear relationships.
   - Significant error when available data in neighbourhood was low, resulting in increase in RMSE.
3. **Bayesian Ridge Polynomial Regression** (**Y** – Objective Variable with Concerned Missing Value, **X** – Other Predictor Variable) – **Best RMSE: 1.04**
   Strengths observed:
   - Was able to fit accurate curves in most neighbourhoods if enough data was available.
   - Was able to capture non-linear relationships, unlike linear models for medium extent values.

   Limitations observed:
   - Generated absurd predictions when available data in neighbourhood was low after dropping rows with missing values, resulting in increase in RMSE.
   - Requires larger extent to use so it can miss out on smaller local trends.

4. **Bayesian Ridge Linear Regression** (**Y** – Objective Variable with Concerned Missing Value, **X** – Timestamp and Value in Other Nodes from Combined Dataset) – **Best RMSE: 0.93**

   Strengths observed:
   - Worked great when used with large extent.
   - Was able to model the relationship of values of the same variable between different nodes very well.
   - Worked well when was not much available data in the neighbourhood using one single node.

   Limitations observed:
   - There was significant error when there are missing values in the same neighbourhood in other nodes as well, resulting in increase in RMSE.
5. **Gaussian Process Regression with Radial Basis Function Kernel** (**Y** – Objective Variable with Concerned Missing Value, **X** – Timestamp and Value in Other Nodes from Combined Dataset) – **Best RMSE: 0.98**

   Strengths observed:
   - Reaches optimum at even larger extent than Bayesian Ridge and hence, with larger neighbourhood, there is more data to work with in case of missing values in other nodes.

   Limitations observed:
   - Generated absurd predictions for certain missing values.
6. **Probabilistic Ensemble Learning Model with Adaptive Weights (Final Model) – Best RMSE: 0.83**

Motivations:

1. We can see that each model has their unique strengths and the strength of one model covers the weakness of another model. For e.g., we can see that the Model 3 and 4 covers both the Limitations of Model 2 and Model 5 partially covers the weakness of Model 4. So, if we can make a new model that combines all of the strengths of these models, we will have a robust model that works for all test data.
2. We want to be able to detect absurd predictions, which is a common weakness of many models and "switch off" the model for such cases and use other models that are not giving absurd values for a particular prediction.
3. We want a model that can have the accuracy of models with low extents and the robustness of models with high extent at once while being able to dynamically change its nature based on the conditions.

This brings to our ensemble model, which uses the weighted mean of multiple models to give the final prediction.

Models Used in Ensemble: Model 2, Model 3, Model 4, Model 5 and an additional Bayesian Ridge Polynomial Regression model with Y – Objective Variable with Concerned Missing Value, X – Timestamp. (Each model has its own extent so we can benefit from the accuracy of low extent and robustness of high extent at once.)

Model Equation: $Predicted\ Y = \frac{\sum \boldsymbol{w_i Y_i}}{\sum \boldsymbol{w_i}}$, where $\boldsymbol{Y_i}$ is the predicted value given by the $\boldsymbol{i}$th individual model and $\boldsymbol{w_i}$ is the weight of the $\boldsymbol{i}$th individual model.

The weight of each model is dynamic. We set initial weights, which are dynamically updated based on optimality conditions, i.e. their individual scores in each neighbourhood as well as percentage of available data to use in each model's neighbourhood. For e.g., in neighbourhoods exhibiting linear trends between features, the Bayesian Ridge Linear Regression model would have the highest weight. If there is not enough data in the neighbourhood, Model 4 and Model 5 which uses data from other nodes to make predictions takes over. This approach is supported by previous research studies that demonstrate the effectiveness of probabilistic ensemble learning models with dynamic weights for similar regression problems [e.g., Adhikari et al.(2015); Liu et al. (2019); Yao et al. (2019)].

Also, we have added an algorithm to detect absurd values in any model, by comparing its deviation of the prediction of each model from the mean of the all outputs to the standard deviations of the outputs. If it crosses a threshold, the weight of the model with absurd predictions is set to zero. Also, we have set an upper and lower limit for predictions, beyond which weights automatically become zero.

Strengths observed:
- Has the combined strength of all its constituent models, so the model can automatically adapt to various different situations, exhibiting highly robust nature as seen in the lowest RMSE of all the models.
- Eliminates the possibility of absurd values.
- Minimizes error by dynamically decreasing weights of models performing badly in a situation.
- Having redundancy built-in into the model increases robustness of the model.

Limitations observed:
- An ensemble model can only be as good as its constituent models, so there can still be situations where none of the constituent models are suitable.
- This ensemble model has 10 hyper-parameters (weights and extents of five models) so tuning is challenging.
- The ensemble model is extremely computationally demanding.