DA 2180: Probabilistic Machine Learning: Theory and Applications

Jan-April Semester 2023, Project Spec

Due: Competition closes 12th March 2023 11:59PM IST, Report due next day (Monday) 11:59PM IST

Competition Link: `https://www.kaggle.com/t/25631b67358c40de899616c9a2bd2e5c`

Weight: 25% of final mark

Introduction

Internet of Things (IoT) enables the seamless integration of sensors, actuators, and communication devices for real-time applications. IoT systems require good quality of sensor data for making real-time decisions. However, we often encounter missing values from the collected sensor data due to faulty sensor, loss of data in communication, interference and measurement errors. The spatiotemporal nature of IoT data and the uncertainty of the data collected by sensors makes missing value estimation a challenging task.

Your task:

In this project, you are given measurements of five sensor nodes from an IoT deployment for environment monitoring where each sensor node is measuring humidity and temperature values. However, there are some missing values in collected measurements. The goal of this project is to predict the missing values in sensor measurements so that the imputed values are as close as possible to the true values. More specifically, you will be developing/implementing a probabilistic machine learning model to estimate missing values in sensor measurements. You may use any probabilistic/Bayesian model or any new/innovative probabilistic technique, as long as it is properly justified or acknowledged. There will be 1 bonus mark for using such innovative approach (which was not taught in the classes and workshops in this course, and its use is properly justified).

To make the project fun, we will run it as a Kaggle in-class competition. Kaggle is one of the most popular online platforms for predictive modelling and analytics tasks. You will be competing with other students in the class. The following sections give more details on data format, the use of Kaggle, and marking scheme. Your assessment will be based on your final ranking in the competition, the absolute score that you achieve, and your report. The marking scheme is designed so that you will pass if you put in effort. So fear not and embrace the power of probabilistic machine learning.

Dataset Descriptions

The data for this project is uploaded in a ZIP file (Project1_Data.zip) in Class Team –> Project 1 Channel – > Files Section. In this zip file, you will have access to three types of data. The first type is the sensor measurements from five sensor nodes where each sensor node is measuring humidity and temperature values. In this type, you are given five CSV files, one for each sensor node namely Node_501.csv, Node_502.csv, Node_505.csv, Node_507.csv, and Node_508.csv. Each file contains sensor measurements collected between 09/12/2014 and 09/01/2015 at 10 seconds sampling rate. Below are the column details in each Node_XXX.csv file

Column(s)	Name	Meaning
1	ID	An integer identifier which is unique for each measurement row.
2	Timestamp	Timestamp for sensor each measurement (humidity and temperature)
3	Temperature	Temperature measurement by a sensor node at given timestamp
4	Humidity	Humidity measurement by a sensor node at given timestamp

There are some missing values in each sensor node CSV file, indicated by NaN in temperature and humidity columns. Each sensor node have 100 missing values - 50 in temperature measurements and 50 in humidity measurements. Therefore, there are total 500 missing values (100 in each sensor node file) in given data. At any timestamp (or in any row of a measurement CSV file), a sensor node has missing values only in one measurement variable (either in temperature or in humidity column). Your goal in this project is to learn a probabilistic model from non-missing measurements (you can consider it as training data) and estimate the 500 missing (NaN) values (you can consider it as test data).

The second type of file, Node_locations.csv contains the locations (latitude and longitude) for each sensor node. Although, you can get reasonable estimates for missing values without using these location information, you may achieve better estimates by incorporating sensor locations in your model effectively.

Finally, the third type of the file, sample_submission.csv is an example submission file on Kaggle competition website. This sample_submission.csv comprises the IDs of missing measurements in their first column in the ascending order. These IDs are same as the IDs which have missing values in either humidity or measurement column in five measurement files described above. The second column in this file will be your predictions. For example, say ID 1 has missing value in temperature and ID 2 has missing value in humidity, and your predictions for ID 1 and 2 missing values are 24 and 45, respectively, then the structure of this file is given as follows

```
ID,Prediction
1, 24
2, 45
...
```

Currently, in the provided sample file sample_submission.csv to you, all the 500 values in the second column are filled with random values between 20 and 50. The actual values are not provided here as this is what you need to predict in this project.

Once you have estimated missing value for each ID, you should create a submission_file.csv with above-mentioned structure i.e., the first row should be a header, exactly as shown in sample_submission.csv. There should be 500 (excluding header row) rows in total, each with a unique ID. The IDs of predictions should match the IDs of missing values in the ascending order (same as IDs mentioned in sample_submission.csv) file. The values in the second column (with header "Prediction") should be your predictions for these 500 IDs.

Note that you **should not** manually enter the predictions in submission file by just inspecting the data, as this is cheating and compromises the point of the project (though please inspect your submissions to ensure your files are in the right format, of the right size, etc.)

As we provide no explicit test set, you may want to reserve part of the given dataset (without missing values) as training partition for this purpose during model development. Your job is to develop an algorithm that can automatically capture the nuances of the problem, in order to generalise well to predict missing values (considered here as test data).

Kaggle In-class Competition

Kaggle Competition Link: `https://www.kaggle.com/t/25631b67358c40de899616c9a2bd2e5c`

Team Registration Google Form Link: `https://forms.gle/ZJhoQPBjQrLQMLLD8`

This is a group-based project, so you need to form a team of 3-4 (Ideally 3, max 4) students. Please do the following **by the end of the first week** after receiving this project:

- Setup an account on Kaggle with username and email being your IISc student email ID (preferable) or your personal email ID.
- Form your team of student peers; Ideally, your team should have 3 members, but you can also form a team of 4 members (provided there is no team in the project with < 3 students).
- Connect with your team mates on Kaggle as a Kaggle team, using a team name. You can choose any team name you wish, such as Superman) which can hide your team's identity from others in the class.
- Only submit via the team;
- Register your team using the Google forms link above, so that we know your team and your team members for evaluation purpose.

The actual values for missing data (measurements) in the data are hidden from you, but were made available to Kaggle. Each time a submission is made, half of the predictions (50% of the missing values) will be used to compute your public score and determine your rank in public leaderboard. This information will become available from the competition page almost immediately. At the same time, the other half of predictions is used to compute a private score and rank in private leaderboard, and this information will be hidden from you. At the end of the competition, only private scores and private ranks will be used for assessment, and will be revealed publicly. This type of scoring is a common practice and was introduced to discourage overfitting to public leaderboard. A good model should generalize and work well on new data, which in this case is represented by the portion of data with the hidden score.

The evaluation score used in this competition is the Root Mean Squared Error (RMSE) over all the predictions, defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{1}$$

where n (=500) is the total number of missing values, $\bar{y}_i$ is estimated missing value, and y_i is corresponding ground truth for samples $i = 1, \ldots, n$.

Each participant can do maximum 5 submissions everyday. Before the end of the competition, each of you will need to choose your 3 best submissions for final scoring. These do not have to be the latest submissions. Kaggle will compute a private score for the chosen submissions only. The best out of the 3 will then be automatically selected and this private score and the corresponding private leaderboard ranking will be used for marking. If you don't choose any submission, Kaggle will by default consider your best submission performance on public leaderboard for computing the private score.

Report

Each team will submit a report with the description, analysis, and comparative assessment (where applicable) of the method or methods used. There is no fixed template for the report, but it should start with a very brief introduction of the problem and notation used. Then the report should describe the approaches that you have attempted along with the motivation for trying them. Reflect on why the method(s) performed or didn't perform well. If you tried different models or different hyperparameters, compare the methods to each other in the context of this competition. Description of at least two methods you tried in your project. Your reasoning can be in the form of empirical evaluation, but it must be to support your reasoning (examples like "method A with X learning algorithm and Y value of parameter got RMSE 1.20 and method B, got RMSE 0.50, hence we use method B", with no further explanation, will be marked down).

Your description of the models should be clear and concise. You should write it at a level that a postgraduate student can read and understand without difficulty. If you use any existing algorithms, you do not have to rewrite the complete description, but must provide a summary that shows your understanding and references to the relevant literature. In the report, we will be very interested in seeing evidence of your thought processes and reasoning for choosing one algorithm over another. Dedicate space to briefly describing the methods tried, validation, sampling techniques (if used), choice of priors and parameters, any interesting details about software setup or your experimental pipeline, and any problems you encountered and what you learned, conclusion what was learned from the data analysis, self-reflection of what the group learned while making the project. In many cases these issues are at least as important as the learning algorithm, if not more important.

The report should be submitted as a PDF, and be no more than three A4 pages of content, including all plots, tables and references[1] (single column, font size of 11 or more and margins at least 1 cm, much like this document). You do not need to include a cover page. If a report is longer than three pages in length, we will only read and assess the report up to page three and ignore further pages.

Submission and Assessment

In summary, each team (**only one student per team**) is required to make the following submissions for this project:

- One or more submission files (submission_file.csv) with predictions for missing data (at kaggle). This submission must be of the expected format as described above, and produce a place somewhere on the leaderboard. Invalid submissions do not attract marks for the competition portion of grading;
- Report in PDF format (Upload via "Assignment" Section for this project in Class Teams);
- Source code used in this project as a single ZIP archive (Upload via "Assignment" Section in Class Teams). Your code can be in any of the following languages C, C++, Python, Jupyter Notebook, R or Matlab. If there is another language you like to use, please contact us first. If the language requires compiling, a makefile or script must be provide to build the executables. We may or may not run your code, but we will definitely read it. You should not include the training or test data file in the ZIP file.

The project will be marked out of 25. No late submission of Kaggle portion will be accepted; late submissions of reports will incur a deduction of 5 marks per day, or part thereof. Based on our experimentation with the project task and the design of the marking scheme below, we expect that all reasonable efforts at the project will achieve a passing grade or higher. So relax and have fun!

Marking Scheme

Kaggle competition (15 marks) This mark takes into account both achieved score (RMSE in this project), as well as your standing in the class. Assuming *N* is the number of students, and *R* is your rank in the class, the mark you

[1] Plots can be useful for model selection, assessing convergence, displaying results and model interpretation, among other things. For instance, plotting the parameters of your model with respect to the objective function can often give insights into what the model has learned.

get for the competition part is

$$12 \times \frac{\min\{8.00 - \max(RMSE, 0.50), 0\}}{7.50} + 3 \times \frac{N - R}{N - 1}$$

The first term constitutes up to 12 marks, and rewards lowest RMSE systems with a maximum score for excellent systems with < 0.50 RMSE, and zero score to those with scores ≥ 8 RMSE which are barely better than random guessing. The second term, worth 3 marks, is based on your rank and is designed to encourage competition and innovation. Ties are handled so that you are not penalised by the tie. All who are tied will get the same marks for score, but ranking will be decided based on total number of submission entries. The score with less entries will be ranked higher among tied ones.

External teams of unenrolled students (auditing the subject) may participate, but their entries will be removed before computing the final rankings and the above expression, and will not affect registered students' grades. Note that invalid submissions will come last and will attract a mark of 0 for the score, so please ensure your output conforms to the specified requirements, and have at least some kind of valid submission early on!

Report (10 marks) The report will be marked using the rubric in Table 1.

Bonus Mark (1 mark) you will get 1 bonus mark if you have used any probabilistic/Bayesian learning model which was not taught in the classes/workshops, or if you have use any innovative techniques in probabilistic learning that improves your model performance on test data (missing values). You need to provide this information in your report with proper justification, to get this 1 bonus mark.

Plagiarism policy

You are reminded that all submitted project work in this subject is to be your own individual work. Automated similarity checking software will be used to compare submissions. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student(s) concerned. For more details, please see the policy at `https://iisc.ac.in/about/student-corner/academic-integrity/`.

Critical Analysis (7 marks)	Report Clarity and Structure (3 marks)
7 marks Final approach is well motivated and its advantages/disadvantages clearly discussed; thorough and insightful analysis of why the final approach works/not work for provided training data; insightful discussion and analysis of other approaches and why they were not used	*3 marks* Very clear and accessible description of all that has been done, a postgraduate student can pick up the report and read with no difficulty
5–6 marks Final approach is reasonably motivated and its advantages/disadvantages somewhat discussed; good analysis of why the final approach works/not work for provided training data; some discussion and analysis of other approaches and why they were not used	*2.5 marks* Clear description for the most part, with some minor deficiencies/loose ends (e.g., there are no- table gaps and/or unclear sections)
3–4 marks Advantages/disadvantages discussed; limited analysis of why the final approach works/not work for provided training data; limited discussion and analysis of other approaches and why they were not used	*2 mark* Generally clear description, but there are notable gaps and/or unclear sections.
1–2 marks Final approach is barely or not motivated and its advantages/disadvantages are not discussed; no analysis of why the final approach works/not work for provided training data; little or no discussion and analysis of other approaches and why they were not used	*1 mark* The report is unclear on the whole, omits all key reference,and the reader can barely discern what has been done

Table 1: Report marking rubric.