Homework 4: missing value imputation (MVI) in IoT monitoring networks.

Jose M. Barcelo Ordinas

Universidad Politècnica de Catalunya (UPC-BarcelonaTECH),
Computer Architecture Dept.
jose.maria.barcelo@upc.edu

May 21, 2025

Missing data are one of the biggest problems for data preprocessing in an IoT architecture, and it is crucial that missing values are recovered to improve the reliability of monitoring applications. In the literature, missing values are usually classified depending on the underlying mechanism that has generated them. In this sense, we can classify the missing value pattern as:

- **Missing completely at random (MCAR)**: This is the most common type of missing pattern, where the probability of having a missing in one variable (or sensor) does not depend on any other variable, therefore, the underlying missing mechanism is completely random. Most of the state-of-the-art techniques focus on this type of missing, e.g., using matrix completion techniques, autoencoders, and multiple imputation among others. This type of missing is ”easy” to impute.
- **Missing at random (MAR)**: in this case, the probability of having a missing value depends on the values of the other variables of sensors. For instance, imagine when certain levels of Active Power are reached the multiplexor stops sending temperature data (hypothetical case). This type of missing data is not that ”easy” to impute since the variables that affect the presence of missing may not provide information about the specific values of the missed data.
- **Missing not at random (MNAR)**: in this case, the probability of missing depends directly on the unobserved value of the variable. These are the most complicated types of missings.

In this scenario, we will face MCAR since no governing rule seems to generate the missing values. There are two main methods of missing value imputation: univariate variable missing value imputation methods are able to impute values in the i-th feature dimension using only non-missing values in that feature dimension. The most commonly used technique is to predict missing values from a window of past values. There are several univariate methods:

- **Last observed carried forward (LOCF)**: replaces the missing value with the last measured value.
- **Polynomial interpolation**: is a technique for interpolating a data set or function by a polynomial.

- **Autoregressive integrated moving average (ARIMA)**: is a generalization of an autoregressive moving average (ARMA) model, which provides a description of a (weakly) stationary stochastic process in terms of two polynomials, one for the autoregression (AR) and the second for the moving average (MA).
- **Recurrent neural networks (RNN)**: the use of an LSTM with a window time T allows the signal to be reconstructed at the missing values.

Multivariate imputation methods use the entire set of available feature dimensions to estimate missing values. The missing values in the i-th feature dimension can be derived from the values of related (statistically correlated or geographically closed) sensors. Examples of methods are: **K-nearest neighbors (KNN)**, **Multiple Imputation by Chain Equation (MICE)**, **Autoencoders (AE)** or **Variational Autoencoders (VAE)**.

The objective of this homework is to analyse the performance of various missing value imputation models on an ozone sensor data set. The objective of this homework is to analyse the performance of various missing value imputation models on an ozone sensor data set. The data set consists of ozone values from 8 ozone sensors. In principle, there are no missing values, so we will produce random missing values in bursts of B missing values per sensor, as will be explained below. Once the missing values are created, several models will be compared. We give you a code called **missing-generator.py** that produces a specific amount of missing values for specific bursts of missing values on a sensor. For example, you can create a 10% of missing values with bursts of 5 missing values in the sensor. The input is a pandas time series with the sensor values, the percentage of missing values you want to create, and the length of the bursts. The function returns a variable with the index of missing values and a pandas time series with NaN in the positions indicated in the index variable.

Data set: The data set is the same as in Homework 1: you have a CSV file called *data_matrix.csv* that contains the data. The first column contains a timestamp with hourly measurements (2258 samples, from 1st of June 2017 to 06 of October 2017). Then, you have 8 columns (one per node) with O_3 measurements. If you are interested in the locations, remember that you had a Node-Location.csv file in homework 2 with the location of each node.

Project: the goal of the project is to evaluate the performance of several MVI methods. For that purpose, we will evaluate univariate and multivariate models.

- **Univariate models**: test two univariate models (polynomial interpolation of degree 3 and a LSTM with a window of T values). Take some of the single sensors, and produce missing values (with several percentage of missing values and with certain bursts). Since, you have the true values (you know the indices of the missing values produced), you can compare the imputations with the true value using metrics such as R^2 and the RMSE.
- **Multivariate models**: Multiple Imputation by Chain Equation (MICE, with MLR and with KNN) and AE. The idea is to substitute the NaN (missing values) by a constant, for example, the mean value of the values after removing missing values.

  The **multivariate MICE model** works as a matrix completion model: each column (sensor S_1, $\dots$, S_N) acts as a regressor for the other columns in k iterations. The process is the following:

  1. mean value imputation (a single imputation process) is first carried out for every missing value on the dataset (this mean imputation could be regarded as a place holder).
  2. the place holder created in step 1 for one of the variables (i.e. output from one of the sensors (S_j) is set back to missing.

3. the observed values from S_j in step 2 are applied in a linear regression (or KNN) with the other variables. S_j being the dependent variable and the other variables being the independent variables on the regression model.
4. The missing values in S_j are then replaced by predictions from the regression model.
5. Step 2-4 is repeated for each variable with missing values.

A complete cycle for each variable constitutes one iteration. At the end of a cycle, all missing data are replaced by predicted values from the regression. The entire process of iteration through all variables is repeated until convergence. At the end, the final imputations are retained, this final set of imputed values and the observed values result in a complete data set.

The **AE model** also works as a matrix completion model. Here, the AE has as many inputs as sensors. We build a AE with a small latent space (with 8 sensors you can try a latent space of dimension 2 or 3). Now, the process can be:

1. mean value imputation (a single imputation process) is first carried out for every missing value on the dataset (this mean imputation could be regarded as a place holder).
2. now, iterate doing the following; reconstruct the signal using the AE. Substitute ONLY the gaps with the reconstructed values, and put again the signal as input until it converges. It is to say, each epoch is an iteration in which the gaps are substituted with the reconstructed signal of the previous epoch..

Compare the different univariate and multivariate models for various situations, e.g. the size of the missing burst, the percentage of missing values and some of the parameters of the approaches, such as the window size in the LSTM model, or the number of sensors involved in the multivariate model, or the dimension of the latent space, etc.