Universidad de La Sabana Felix Mohr
Facultad de Ingenería 2024.1

Time Series Machine Learning

- Sheet 1-

Exercise 1 (6 points)

1. Write a function `flatten_relative_and_full_ts_df(df)` that takes a dataframe, in which rows are subjects/instances, columns are features, and each cell is a sorted iterable (list, numpy array, Series). The output is a dataframe with a column for the subject (this can be just some integer, e.g, what previously used to be the index), one for the timestamp, and one other for each sensor.

   This function assumes (and indicates in its name) that the time series objects are complete in the sense that even if a piece of information is missing, then this is indicated through a `np.nan`. Since the lists describing the time series have no timestamp yet, this timestamp is inferred implicitly from the observations' index in the list.

2. Write a function

   `insert_times(df, timestamps, sub_col="x", time_col="timestamp")`

   that generates, for each subject, all timestamps according to the time scheme defined in `timestamps`, which is a list or Series, and inserts respective rows into (a copy of) `df` with `np.nan` values for all datapoints.

   This function assumes that `df` is in the flattened format. `sub_col` and `time_col` simply indicate the names of the columns that contain the subjects and the timestamps, respectively.

3. Write a function

   `impute(df, technique, reach=2, lookahead=False, sub_col="x", time_col="timestamp")`

   that creates a copy of `df` in which all missing values have been imputed. `technique` is a string with the following possible values: `average`, `linear`, `cubic`, `barycentric`.

   `reach` defines how many values are included for the computation, and `lookahead` whether only past data or also future values should be used. If `lookahead` is true, then split the values in `reach` so that half of the points are prior to the imputed and half are posterior to the imputed times (1 more point from the past if `reach` is odd). You can simulate forward fill via `reach = 1` and `technique=average`.

   Hint 1: Operate by subject and sensor. For each such combination, first find all indices with missing values and group them into blocks (use the helper function). Based on the indices of the blocks, you can find the indices of usable data. Observe that all imputation functions are based on x- and y-values of data that you *do* have. So it could be a good idea to create too lists (or a dataframe) with these pieces of information, independently of which imputation technique is being used.

   Hint 2: If `lookahead` is false and `technique` ∈ {`linear`, `cubic`, `barycentric`}, then you must do an extrapolation (instead of interpolation), and this cannot be done with `np.interp`. You might want to check `np.polyfit` to solve this problem for the linear case. Generally, for the other methods check `scipy.interpolate` package, which has objects that allow for both interpolation and extrapolation.

4. Write a function

   ```
   prepare_data(
       df, frequency, start_time=0, end_time=None,
       sub_col="x", time_col="timestamp",
       imp_technique="linear", imp_reach=2, imp_lookahead=False
   )
   ```

   that does the following:

   a) it automatically detects whether the dataframe has already been flattened. Ohterwise, it does so (making sure that the subject and time columns have the given names).

   b) then it creates a series of all timestamps required by the time scheme.

      Hint: You may want to check `pandas.date_range` to do this and to allow for *very* convenient values in the time scheme definition.

   c) Perform upsampling by combining `insert_times` and `impute`, taking into account the given parameters.

   d) Then perform downsampling, simply eliminating all rows with timestamps that are not in the time scheme.

   Return the dataframe created in this way.

Exercise 2 (4 Points) We now study two time series dataset, which are contained in the files `SEMG_DB1.rar` and `factory.zip`. Both datasets are from real world data.

1. SEMG Dataset.

   a) Read in the data data and create a non-flattened dataframe of 22 rows (11 abnormal and 11 normal subjects), each of which is one subject, with 5 columns (we only look at the data where subjects were standing), and list entries for each cell.

      Observe that the time series are of different lengths here.

   b) Create a reasonable time scheme and prepare the dataset using your above function. Is imputation being used here? Why or why not?

      Visualize the full dataset by creating five figures, one for each feature, in each of which you show the 22 curves of the subjects.

2. Factory Dataset.

   a) This dataset consists of data for 49 sensors. How many subjects does this dataset have?

   b) Prepare a dataset (is it necessary to flatten this dataset?) in pandas.

   c) Prepare the dataset using your function with different imputation techniques, and try to show the differences through visualizations of sensors where these differences become visible.