Chapter 4: Data & Experimental Design

Draft version

This chapter describes the case study that was done to test the performance of DBAFS. It is structured as follows. First, the characteristics of the bike sharing system that served as a data source for the experiment, are presented. The data retrieval process is described in section two. Finally, the third section explains the methodology of the experiment itself.

4.1 Data source

`*NOTE: Describe JUMP Bikes, show system area in San Francisco, mention that it are electric bikes, and give main characteristics of the system (how much is it used, by who, for what purposes, etc)*`

`*NOTE: Also discuss the following plot, adapted from the San Francisco Municipal Transportation Agency, which shows the usage of the dockless JUMP system in San Francisco compared to that of the station-based GoBike system. It can be seen that the pattern of JUMP is less consistent, which can have a large influence on the forecasts*`

4.2 Data retrieval

JUMP Bikes provided access to their historical database containing the geographical locations of their available bikes in San Francisco. The database fulfilled all DBAFS requirements described in section 3.4. Data collection started at September 9th, 2018, 15:41:08, Pacific Daylight Saving Time (PDT). The data had a temporal resolution of one minute, meaning that every minute, the location of each available bike in the system was recorded. Timestamps were stored with six-digit precision. Because of that, the time of recording was not exactly the same for each available bike, but could vary up to a few seconds. Therefore, before using the data in DBAFS, all timestamps were truncated to minute precision.

When calculating distances, only the historical data at every quarter of an hour were used in the experiment. Hence, i_s was set equal to fifteen minutes. There were several reasons for this choice. Firstly, it is not expected that the data change drastically from minute to minute. That is, using data with a temporal resolution of one minute will probably not contain a lot more information than using data with a temporal resolution of fifteen minutes. Consequently, if a forecast for a specific timestamp is in practice a forecast for a few minutes earlier, this will not be problematic. On the other hand, using only data every fifteen minutes will decrease the size of the data with a factor fifteen, and speed up computations considerably. Furthermore, a lower order ARIMA(p, d, q) model can be used to capture the same patterns in the data. This is important, since lower order models will result in lower errors arising from parameter estimation (Brockwell and Davis 2002).

`*NOTE: Talk about pre-processing of usage data*`

4.3 Experimental design

The complete data period of 100 days was divided into a training and a test period. The training period spanned from the start of the period at September 9th up to and including December 2nd. The last two weeks of data, starting from December 3rd, formed the test period.

4.3.1 Training period

The approximately 12 weeks of data in the training period were used for both the cluster loop and the model loop. In fact, a real-world situation was simulated in which consecutive passes through the cluster loop and the model loop took place at the end of December 2nd. After laying a square grid with 500 m^2 cells over the study area, distance data were calculated from the training dataset with a fifteen minute temporal resolution, for each grid cell centroid. Additionally, usage data were calculated from the original training dataset with a one minute temporal resolution. Hence, $m_c \approx 12$. With these data, the locations of the model points were defined, and passed on to the model loop. There, for each model point, distance data were calculated from the complete original training dataset, and these data were used to build the models on. Hence, $m_m \approx 12$. Both clustering and model building was only done once, so n_c and n_m, the number of weeks between respectively passes through the cluster loop and passes through the model loop, were not defined. What had to be chosen was the set of integers K, containing all values that would be considered as the number of desired clusters k in the spatially constrained hierarchical clustering procedure. Taking into account the size and shape of the system area, along with some exploratory research on the characteristics of the city, all integers $4 \leq k \leq 10$ were included in K.

4.3.2 Test period

In the two week test period, forecast requests were simulated, and send to the forecast loop. To do this in a realistic way, the following requirements should be fulfilled.

- More forecast requests should occur at locations where the usage intensity of the system is higher.
- More forecast requests should occur at times when the usage intensity of the system is higher.
- The times when the usage intensity is higher, can vary per location, and vice versa.

Taking this into account, the following approach was developed. Usage data were calculated from the original test dataset. Then, 1000 pick-ups were randomly sampled from all pick-ups in these usage data, and the location-time combinations belonging to them were retrieved. Pick-ups reflect the usage of the bike sharing system. That is, the sample will contain more locations from areas where the usage intensity is high, and more timestamps from time periods in which the usage intensity is high. Furthermore, the location and time come as a combination, rather than as separate entities. In this way, the approach fulfills all three requirements mentioned above. The location-time combinations in the sample will from now on be referred to as the *test points*.

Not only one single forecast was made per test point. Instead, starting from the timestamp of the test point, all time lags up to one day ahead, i.e. 96 time lags in total, were forecasted, and compared to the corresponding observations of those time lags. Then, for each test point, the RMSE and MAE were calculated. For each test point, 4 weeks of historical distance data were used for decomposition. Hence, $m_f = 4$. Obviously, these time periods partially overlapped with the training period. Having 1000 test points, the total number of individual forecasts was 96000.

Obviously, by using the approach described above, the reported overall forecast errors will be dominated by those made during peak hours, and in crowded areas, when and where obtaining good forecasts is generally harder. However, this is intended, because reporting a large amount of forecast errors made during off-peak hours, and in non-crowded areas, or, alternatively, using adjusted error metrics such as the RMSLE, may give results that look nicer, but do not reflect the real usefulness of the forecasting system.

To compare the performance of DBAFS in a relative manner, all test points were also forecasted 96 time lags ahead with a simple baseline method. The naïve method, as described in section 2.4.3, was chosen for this task. The RMSE and MAE of those forecasts were calculated, and compared to those of DBAFS.

4.3.3 Benchmark

NOTE: Describe the process of comparing the computing times between DBAFS and the Naïve method. Also include a third method in the experimental design: forecasts where models are fitted to the data for each forecast individually, and the whole system with model points is not used.

The complete methodology of the experimental design as described above is summarized in Figure 4.x.

NOTE: Insert figure to summarize experimental design

References

Brockwell, Peter J., and Richard A. Davis. 2002. *An Introduction to Time Series and Forecasting.* Vol. 39. doi:10.1007/978-1-4757-2526-1.