Kapil Bastola

4/23/2017

Assignment 11

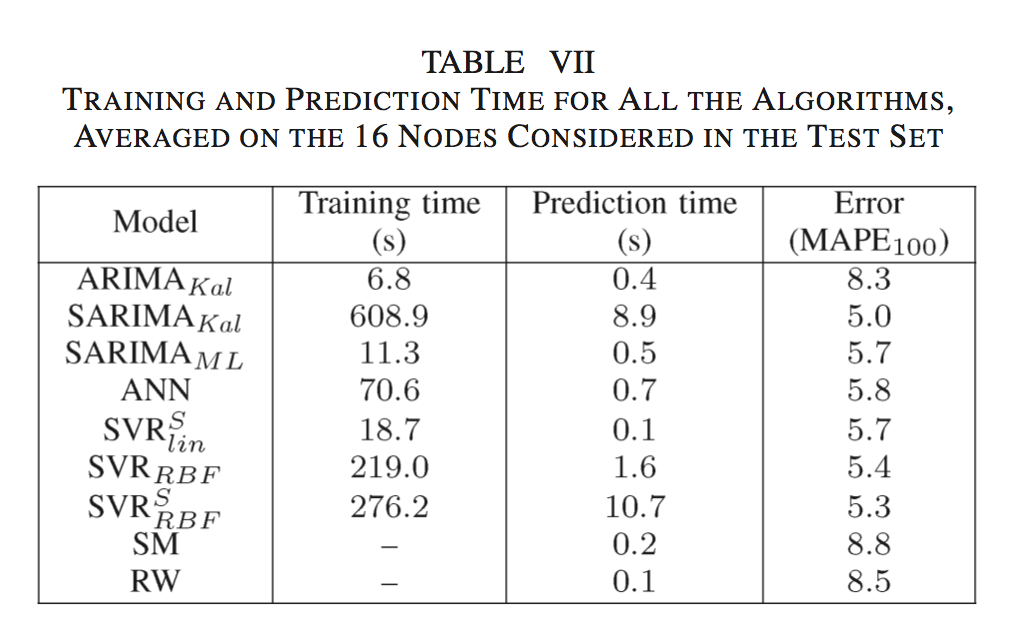
**Discussions**

The competitor article I chose to compare against is the article titled “Short-Term Trafﬁc Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning” by Lippi, M., Bertini, M., & Frasconi, P published in *IEEE Transactions on Intelligent Transportation Systems.* In this article the authors first reviewed existing approaches to traffic flow forecasting the common view of probabilistic graphical models, presenting an extensive experimental comparison, which proposes a common baseline for their performance analysis and provides the infrastructure to operate on a publicly available data set. Then the authors provide two new support vector regression models, which are speciﬁcally devised to beneﬁt from typical trafﬁc ﬂow seasonality and are shown to represent an interesting compromise between prediction accuracy and computational efﬁciency.

For the result the competitor did eight different models, which were either supervised learning or time series analysis. The set of tested competitors consists of the following algorithms:

* + RW, which is a simple baseline that predicts trafﬁc in the future as equal to current conditions
  + SM, which predicts for a given time of the day the average in the training set;
  + ARIMA model with Kalman ﬁlter
  + SARIMA model with maximum-likelihood ﬁtting
  + SARIMA model with Kalman ﬁlter
  + ANNs
  + Support Vector Regressor (SVR) with RBF kernel
  + SVR with RBF kernel multiplied by a seasonal kernel
  + SVR with linear seasonal kernel

The authors found that the Seasonal ARIMA (SARIMA) model coupled with a Kalman ﬁlter is the most accurate model. However, when forecasting during the most congested periods the seasonal support vector regressor (SVR) was found to be highly competitive method as well.

I did three different models with one supervised learning model with the decision tree and two time-series analysis model with one seasonal ARIMA and one non-seasonal ARIMA. For the time series analysis and the supervised learning algorithms, the competitor use Mean Absolute Percentage Error (MAPE) to evaluate the accuracy of their models while computing the time taken to run the model. The time taken to run the model between the competitor and our dataset cannot be compared since we are not sure how may data points there were in the competitor’s dataset but we can look at MAPE to evaluate how our model stands with respect to the competitor’s model. The competitor, in their article, presented a table with the MAPE for each model as shown in the figure below:

The table from the competitor article shows that the lowest mean absolute percentage error they got was for a Seasonal ARIMA or SARIMA model with MAPE of 5.0. Both of our time series models performed better than the best Seasonal ARIMA from the competitor.

First we let R select the best order for the auto regressive integrated moving average (ARIMA) model and it picked an order of (4, 1, 4) with no seasonality. We got a Mean Absolute Percentage Error of 2.52. We also predicted the last 12 values of the time series. When comparing the actual versus the predicted and the variation in the actual speed is not as prevalent in the predicted speed. Also looking at the Ljung-Box test for significance, we see that there are first few lags that are not significant. We wanted to pick our own orders for AR and MA by looking at the ACF and PACF. Looking at the ACF and PACF we see that there is some seasonality in the data and decided to add a seasonality component to the ARIMA and performed an ARIMA order of (0, 2, 7) with a seasonality of (0, 0, 18). Using these orders, we reran the ARIMA fit and predicted the last 12 hours of average speed of the time series again. For this model we got an MAPE of 2.19, which is an improvement from the previous time series model. Both MAPE of models that we created were better than the best MAPE of the competitor’s model.

We also performed a supervised learning model with a decision tree to predict average speed based on two variables hour of the day and vehicle count. A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node represents a class label, which is the decision taken after computing all attributes.

First node of the decision tree resulted is whether the vehicle count is higher than 0.5 or not. If it is higher than 0.5, average speed is 47 kph. If the vehicle count is lower than 0.5 we encounter another node, which examines whether time of day is greater than 4 am or not. If the time of day is less than 4 am, average speed is 46. If the time of day is greater than 5 am we encounter another node that examines whether time of day is after 6 pm or not. If time of day is after 6 pm the predicted average speed is 42 kph and if it is before 6 pm and after 5 am, the predicted average speed is 33 kph. We looked at the error of this model and saw that the average MSE is 478. This is a high error and hence we decided to focus mainly on the time-series analysis, as we do not have many variables we can use.

**Conclusion**

As the competitor we also found that Seasonal Auto Regressive Integrated Moving Average (SARIMA) model is best model to predict traffic flow as we observed the least amount of mean absolute percentage error using this model. Using supervised learning models like the decision trees was not the best way to approach the dataset we had because of the lack of variables and we observed a high error in the decision tree’s result. The seasonal ARIMA and ARIMA model that we built had a much lower MAPE compared to the competitor. Since the dataset the competitor’s article used and I used are different with different variables and attributes, it may be difficult to compare the two results but only looking at how the model we built predicted the results compared to the actual values, we can say that seasonal auto regressive integrated moving average (SARIMA) is the best predictor of traffic flow.