STAT542 Project 2 Report

XXX (netID)

Fall, 2021

1 Introduction

In this project, we are going to predict the future weekly sales of some Walmart stores on department levels. In particular, we are going to train a seasonal linear regression model and a ARMA(2,1) model (if some conditions are satisfied) for each store-department combination. This report, written by the only member of the team (XXX; netID), documents and discuss our major findings.

2 Pre-processing

We mainly follow the advice in *Project 2: A more e icient implementation* on Campuswire to pre-process the training data. On top of that, points 3 to 5 are our remarks:

1. We extract the store-department combinations with at least 1 observation.
2. For these combos, we extract the year, weekly sales and one-hot encode the week number.
3. We try to include `IsHoliday` as an additional binary predictor. However, it seems that the one-hot encoded week number better captures the holiday effect in the seasonal model. Therefore, we exclude `IsHoliday` in the training data.
4. We also try to include lagged weekly sales as numerical predictors in the seasonal model. The predictions worsen a lot due to issues such as non-stationarity (weekly sales may not be a stationary time series) and jumping week numbers (weekly sales may be observed at weeks 5 and 7 instead of at weeks 5 to 7).
5. We think of treating everything as a panel dataset as we have week numbers (time) and stores/departments (cross-section). We give up this approach at the end as we do not have specialized libraries to handle aforementioned issues in the time domain.

3 Trainning

3.1 Seasonal linear regression model

After pre-processing, we first train the seasonal model described in *Project 2: What we have tried (III)*. In simple words, it uses the historical average weekly sales to predict future weekly sales of the same week number. This model has reasonable performance as shown on Campuswire, and can be understood as capturing the long-run aggregate consumer behavior in the dataset. We do make some minor adjustment in the final prediction, which will be explained in Section 3.3.

3.2 ARMA(2,1) model

In contrast to *Project 2: What we have tried (IV)*, we do not consider dimension reduction here. The reason is that we believe the seasonal model have not yet captured the short-run behavior. To do so, we use `arima` in the `stat` package:

1. Due potential non-stationarity, we fit the ARMA model on the first-differenced weekly sales. The model fitting is also put in a `tryCatch` block as the first-differencing does not guarantee stationarity for some store-department combos.
2. We choose `order=c(2, 0, 1)`, i.e., a ARMA(2,1) model:

$$\Delta Y_t = \mu + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \theta_1 \epsilon_{t-1} + \epsilon_t,$$

   where $\Delta Y_t = Y_t - Y_{t-1}$ is the first-differenced weekly sales, μ is its mean, ϕ_1 and ϕ_2 are the AR parameters, θ_1 is the MA parameter, and $\epsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$; see, e.g., Brockwell and Davis (1991).
3. It is possible to perform model selection for different store-department combos. However, this will increase implementation/run-time, and more importantly, we find that a fixed `order` gives a satisfactory improvement already.
4. Since we performed first differencing, we append the last observation in the training data to the prediction and sum up cumulatively. This restores the level prediction, e.g.,

$$\widehat{\Delta Y_{T+2}} + \widehat{\Delta Y_{T+1}} + Y_T = \widehat{Y}_{T+2} - \widehat{Y}_{T+1} + \widehat{Y}_{T+1} - Y_T + Y_T = \widehat{Y}_{T+2}.$$

3.3 Final prediction

For each store-department combo, the final prediction is formed as follows:

1. We always train the seasonal linear regression model.
2. If the training data has more than 50 observations, we try to train the ARMA(2,1) model. This is done to mitigate the issue of jumping week numbers and better understand the short-run behavior.
3. If we fail to train the ARMA(2,1) model (either due to non-stationarity or less than 50 observations), we assign weights `c(1,0)` to predictions of the seasonal and ARMA(2,1) model, respectively. Otherwise, we assign weights `c(0.8,0.2)`.
4. Before doing the weighted sum of predictions, we check whether some of the predictions is negative and change them to zero. This is intuitive as (gross) sales cannot be negative.
5. The final prediction is the weighted sum of predictions from the seasonal and ARMA(2,1) model.

4 Discussion

We report the WMAE of our predictions and total runtime in Table 1. The experiment is conducted in R 4.1.1 on Windows 10 with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz. From these tables and the previous sections, there are several takeaways:

1. It is useful to incorporate domain knowledge in statistical modeling. In this project, we consider the short-run and long-run behavior to be captured by ARMA(2,1) and seasonal model separately. The improvement in WMAE is greater than the dimension reduction discussed in *Project 2: What we have tried (IV)*.
2. It is also helpful to consider model ensembling. Due to non-stationarity, ARMA models probably have poor prediction performance on its own. However, combining ARMA with seasonal models achieves better predictions.

Table 1: WMAE and runtime (seconds) of our final predictions

	Result
fold1	1819.4890
fold2	1394.5746
fold3	1373.5149
fold4	1488.4123
fold5	2290.0648
fold6	1699.6922
fold7	1656.4894
fold8	1351.3598
fold9	1406.6077
fold10	1340.9912
average	1582.1196
runtime	284.8161

3. We can further improve the final predictions in many ways. Apart from ARMA order selection, we can try to optimize the weights between different models, or train the ARMA model with the residuals from the seasonal model (as pointed out by Feng after I discussed my time series approach with her). Better predictions may be achieved, e.g., due to shrinkage effect.
4. Interestingly, a simple inflation rate adjustment will lead to slightly higher WMAE if we multiply the 2012 US-wide inflation to the predictions. A plausible explanation is that we may need to adjust for region difference (those stores may locate in different states) and product difference (different departments may have different inflation rates). Since we do not have those information, we give up the inflation rate adjustment.

Reference

Brockwell, Peter J, and Richard A Davis. 1991. *Time Series: Theory and Methods*. Second. New York, NY: Springer. https://doi.org/10.1007/978-1-4419-0320-4.