**An integrated forecasting and optimization system for the inventory planning problem**

**Richard Alaimo a, Heena Jain a**

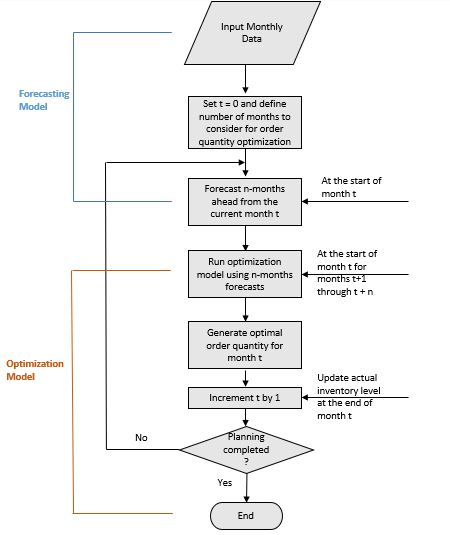
*a Department of Systems Engineering and Engineering Management, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC*

**1. Introduction**

For this report, we propose an inventory planning system to address the inventory planning problem using the open source demand data provided from Amazon Web Services. Our approach is a combination of *predictive and prescriptive data analysis*, specifically *forecasting and mathematical programming*. The overall system is represented as an integrated model, where demand is forecasted for future time periods (or months) and order quantity sizes are optimized for each month given the uncertain future demand. The outputs of the integrated model include the following metrics for each month: beginning and ending inventory level, order quantity size, and backorder and holding costs. To account for the uncertainty that exists for future months and their potential impacts on the current month, an aggregate planning strategy is used where future demand, up to -months ahead, is considered for deciding how many units to order (i.e. order quantity level) for the upcoming month. This approach will be referenced as the -month ahead strategy for the remainder of this report. We test different values of that provide the best results for optimization of order quantity and make a recommendation on the value of at the end of this report that may be used to generate efficient real-time planning parameters.

At the start of month , the sales for month are observed and the order quantity for month is placed. The inventory stock is then replenished at the beginning of month since a lead time of one month is required. A *seasonal auto-regressive integrated moving average* (SARIMA) forecasting model generates the -month ahead forecasts for months through . Following this, a *linear programming* (LP) optimization model uses these forecasts and their respective standard error estimates as inputs to randomly generate demand scenarios in order to determine the order quantity level for month . The optimized order quantity size is then presented to the user so that an order can be placed in month to satisfy the forecasted demand in month .

The flow chart in Figure 1 explains the process flow of the integrated model. Because the model is validated on past data, a case of ex-post forecasting, the length of planning horizon is equal to the number of months that are planned in advance. In a more realistic scenario of ex-ante forecasting, the planning horizon will be one month where the integrated model is executed at the beginning of each month in order to plan for the upcoming month. The remainder of this report includes: an analysis of the data and a discussion on the insights that were found (Section 2), the formulation of the forecasting and optimization models (Section 3 and 4, respectively), validation of the proposed system (Section 5), and a conclusion to summarize the report (Section 6). In the end, there is a “Read Me” section that provides the instructions to run the model (Section 7).



**Figure 1: Process flow of integrated model**

**2. Data Analysis**

The data was analyzed to identify the presence of trend and seasonality. Stationarity was observed using auto correlation function (ACF) and partial auto correlation function (PACF) plots. It was observed that the monthly demand from 1996 to 2005 is non-stationary. Non-stationary data is an implication of trend and seasonality in the data that should be identified to build an efficient forecasting model. To study the trend that exists in the data, the annual average demand for each year was calculated to see if the mean remains constant or almost constant. The graph in Figure 2 suggests a clear indication of an increasing trend in monthly demand. There are two outliers for the trend line in the data for the years 1999 and 2000. It advocates a hike in demand in those years that could be because of market conditions, temporary change in customer preferences, etc. Sudden peaks and falls affect the accuracy of forecasts and need to be studied separately. Due to the small size of the dataset and limited information available, the outliers were not studied separately for this report. To analyze the seasonality, we plotted the seasonality graph (Figure 3) that shows the monthly demand for each year in one graph. The yearly plots follow the same seasonal patterns throughout the span of 10 years suggesting a seasonal period of one year.

|  |  |
| --- | --- |
|  |  |
| **Figure 2: Trend line of yearly average demands** | **Figure 3: Seasonal plot of monthly demands by year** |

**3. Forecasting Model**

Instead of using an auto-regressive integrated moving average (ARIMA) model that captures only trend, we created a SARIMA model to generate -step ahead forecasts for the optimization model since the data has both trend and seasonality. The SARIMA model contains seven variables: three for trend, three for seasonality and one variable to capture the index at which seasonality happens in the data. The trend variables are (), seasonal variables are (), and is the number of time periods for a season to repeat.

Firstly, it can be noticed from Figure 3 that the duration of a seasonal period is one year for the given data. From this, can be set to 12 for yearly seasonality. The trend variable refers to the order of auto regressive term, refers to the moving average term in the model and is the degree of differencing required for a non-stationary series. Figure 4 shows the monthly demand and Figure 5 shows the lag-one differenced demand for the given data. It can be observed that the differenced series appears stationary, suggesting the value of as 1. The ACF and PACF plots were used to identify the degree and order of the auto-regressive and moving average terms as 2 and 0, respectively. Therefore, values of the variables for trend part of the SARIMA model, corresponding to , are .

|  |  |
| --- | --- |
|  |  |
| **Figure 4: Monthly demand** | **Figure 5: Lag-one differenced demand** |

In a similar fashion, the values for the seasonal variables, were identified as . Finally, the SARIMA model for the inventory planning problem can be written in the form SARIMA. To train the model, the dataset was divided into training and test data with the ratio 9:1. In other words, data from January 1996 to December 2004 was used as the training set and the remaining data as the test set (January 2005 to December 2005). The rolling forecast approach with window has been used in the model to forecast -months ahead from the current month. These forecasts along with the standard error estimate of each prediction is passed to the optimization model. The mean absolute percentage error (MAPE) of the test forecasts obtained was 1.61% with mean squared error (MSE) of 4.42. The Durbin Watson test statistic was at 1.66 suggesting that the errors may be positively autocorrelated at lag-one. Because the value of the test statistic is close to 2, the error auto correlations are considered insignificant.

**4. Optimization Model**

Before presenting the optimization model, the following notations are defined for denoting problem parameters and variables.

|  |  |
| --- | --- |
| number of months that are being considered for the -month ahead planning strategy. | continuous variable representing the end of month inventory level of month for scenario . |
| number of randomly generated demand scenarios for each month. | continuous variable representing the positive inventory level of month for scenario . |
| indexed set for each month within the planning horizon. | continuous variable representing the inventory level beyond 90 units during month for scenario . |
| indexed set for each scenario that was randomly generated. | continuous variable representing the negative inventory level of month for scenario . |
| demand of month for scenario . | continuous variable representing the order quantity size for month . |

Using the notation from above, the proposed optimization model is as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |
|  |  |  |
|  |  | (2) |
|  |  | (3) |
|  |  | (4) |
|  |  | (5) |
|  |  | (6) |
|  |  |  |

Objective function (1) minimizes the expected handling and backordering costs for months across all scenarios. Constraint (2) measures the end of month inventory level of scenario during month . The beginning inventory level of month consists of the order quantity from month , as well as the ending inventory level of month . Since is unrestricted, it can be used to measure the positive or negative ending inventory level and , respectively. Constraint (3) evaluates the number of units that were backordered (i.e. negative inventory) during month for scenario . Constraint (4) evaluates the number of units that were not sold (i.e. positive inventory) during month for scenario . Constraint (5) determines the amount by which the end of month inventory level is beyond 90 units during month for scenario . Constraint (6) is used for enforcing non-negativity conditions for the respective variables.

It is worth noting that is evaluated across all scenarios to minimize the expected holding and backorder costs for months. Since the demand for future months is uncertain and there are few constraints for this problem, generating random demand scenarios using the error from the forecasting model and determining the order quantity size that minimizes the expected cost for all scenarios was deemed to be an appropriate approach. After the optimization is completed for an initial month , the order quantity level and end of month inventory level (same value for all since is observed at the beginning of the month) are stored and used for determining the order quantity size for the upcoming month. For example, suppose and month is represented as December 2004. This suggests that month are represented as January 2005, February 2005, and March 2005, respectively. After finding an optimal solution over this planning horizon, and are referenced for determining the beginning inventory level of January 2005, which leads to another optimization problem to be solved. This process is repeated until order quantity sizes are determined for all months of interest, excluding. Once these values are determined for all months, then inventory levels for each month can be measured by differencing the beginning inventory level for each month and the observed demand, allowing for the costs to be calculated as well.

**5. Validation of the Proposed System**

Since an -month ahead strategy is incorporated into the proposed system, it is necessary to determine the best value of where holding and backorder costs are minimized for the months of interest. From the problem description, this is not simple to accomplish since there aren’t any constraints for maximum order quantity sizes and inventory levels for each month. To determine , it was decided to create a testing framework where different values of are evaluated using randomly generated demand scenarios (based on output from the forecasting module) for multiple iterations. Executing the system for multiple iterations allows for a wide range of randomized demand scenarios to be considered where it is likely that actual demand for future months (not yet observed) is within the range of the simulated data.

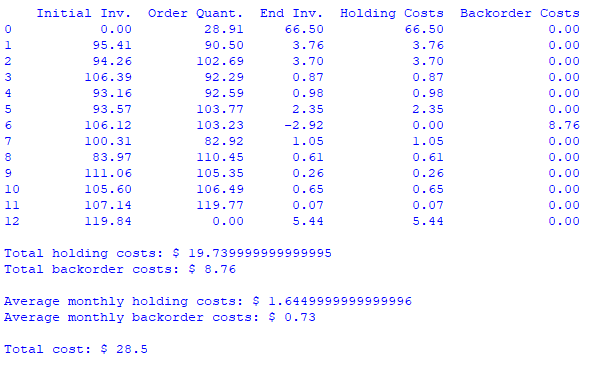
For implementing the integrated system, Python programming language was used due to the abundance of packages that are available. *statsmodel.api* package was used for creating the SARIMA model and *gurobipy* was used to interface with Gurobi advanced optimization solver within the Python environment. The different values of that were considered include 1, 2, 3, and 4 and the number of scenarios was set to 300. These values of were individually evaluated for 10 iterations where randomly generated scenarios were produced during each iteration. This implies that the system was executed 40 times (10 times per value of ).

Once an iteration is completed, the value of that resulted in the lowest total cost was recorded. After all iterations were finished, the number of times that each value of resulted in the lowest cost were counted and the one with the highest frequency was selected as the recommended value of . Since the optimization model starts in December 2004, the initial inventory level was assumed to be equal to 65.5 (average of the given ending inventory levels of December 1995 (60) and December 2005 (73)). The output from this test is displayed in Table 1.

**Table 1: Results from validation test**

|  |  |  |
| --- | --- | --- |
| Iteration |  | Total Cost |
| 1 | 1 | $ 28.50 |
| 2 | 1 | $ 28.34 |
| 3 | 4 | $ 28.71 |
| 4 | 3 | $ 28.61 |
| 5 | 1 | $ 29.06 |
| 6 | 2 | $ 28.39 |
| 7 | 2 | $ 28.76 |
| 8 | 4 | $ 28.45 |
| 9 | 1 | $ 29.32 |
| 10 | 2 | $ 29.33 |

Looking at Table 1, it can be noticed that had the most cases where total cost was minimized with respect to other values of . Using the criteria specified earlier, this suggests that the recommended value of for the integrated system is 1. An example of the output that was generated from the system (iteration 1) is displayed in Figure 6. Since the initial inventory level of December 2004 () is unknown, it is set equal to 0. The end of month inventory level for December 2004 was estimated to be 65.5, as mentioned earlier. During , it is of interest to determine the order quantity level to satisfy the forecasted demand during (January 2005) , and similarly for the other months. It can be noticed that there was a negative inventory level one time (, or June 2005) and the maximum positive inventory level was 5.44 (, or December 2005). During (November 2005), is optimized for satisfying the forecasted demand for (December 2005). Since December 2005 is the final month for our validation, it is not of interest to measure to satisfy the forecasted demand for (January 2006). For this reason, is set equal to 0, as shown in Figure 6. Using this data, ending inventory levels and costs can be calculated.



**Figure 6: Example of generated output from the system**

**6. Conclusion**

An integrated forecasting and optimization system was proposed in this report. Forecasting is utilized for generating accurate demand forecasts for upcoming months using SARIMA and optimization determines order quantity levels such that expected holding and backorder costs are minimized across the planning horizon with respect to the -step ahead strategy. It was decided from the validation test that the system performs best when is equal to 1, suggesting that order quantity levels for the current month are decided upon only from the forecasted demand of the upcoming month . Randomly generated scenarios are produced during the optimization phase to consider multiple forecasted values of demand for the upcoming month, hence the objective of minimizing expected costs. After determining the optimal order quantity sizes, inventory levels for each month are determined from beginning inventory levels and observed demand, ending with the calculation of the total cost and monthly average costs.

**7. Read Me**

Running this code requires Python IDE (any 3.7 version) and a license for Gurobi 8.1.0. A number of packages have been used to design this model and it is important to install each of these packages before running the code. The list of packages needed are:

1. calendar

2. pandas

3. statsmodels.api

4. numpy

5. gurobipy

6. math

7. random

Please follow these instructions to run the code:

1. Place the file named “TenYearDemand.csv” provided during submission in the working directory without changing the name. This data contains the monthly demand values from January 1996 to December 2005 and is used for forecasting the unknown demand values from January 2006 to December 2007.

2. In the same directory, place the input file that will be used for validation (demand data from January 2006 to December 2007). It is assumed that this demand data will be structured exactly the same as the data that was provided for the competition, as specified in the problem description.

3. Change the file name that is in line 19 to the name of the file that will be used for the testing (currently set as “Ten-Year-Demand.csv”).

4. Run the program.

5. Output will be displayed in the format shown in the report in Section 6 for the 24 months.