fORged by Machines

Kartik Kulkarni, Yingqiu Zhang

Virginia Tech

October 13, 2019

Due to the rise in the incorporation of data analytics, inventory management has been one of the primary areas of focus in the supply chain industry. In this competition, we present two methodologies to tackle the inventory planning problem with uncertain demand.

1 Methodology 1.0

Our aim is to optimize the order quantities across the planning horizon, such that the overall holding and backordering costs are minimized. The main challenge of the problem is that the demand is stochastic. Had the demand been deterministic, the problem would be reduced to a special case of the Uncapacitated lot-sizing problem which was first studied by Wagner and Whitin [2], but unfortunately that is not the case. The deterministic version of the problem with T periods in the planning horizon and where $d_1, d_2, \ldots, d_T$ are the known demands, can be formulated as follows:

$$\text{Minimize} \sum_{t=1}^{T} \Big(p_t x_t + h_t s_t + b_t r_t \Big) \tag{1a}$$

$$\text{s.t. } s_{t-1} + x_{t-1} - r_{t-1} = d_t + s_t + r_t, \; t \in \mathcal{T}, \tag{1b}$$

$$x_t, s_t, r_t \geq 0, \; t \in \mathcal{T}, \tag{1c}$$

where x_t is the amount ordered in period $t \in \{1, \ldots, T\}$, s_t is the inventory at the end of period $t \in \{1, \ldots, T\}$, and r_t is the backorder quantity in period t. p_t, h_t and b_t are per unit production costs, holding costs, backorder costs, respectively. Notice that the costs in our problem are non-speculative. Therefore, there is not motive behind ordering the item before it is required. This problem is relatively easier than the case when demand is stochastic.

However, we know the monthly demand over a period of previous ten years. We use this data to forecast the demands for the years 2006 and 2007. Let us first take a look at the demand history.

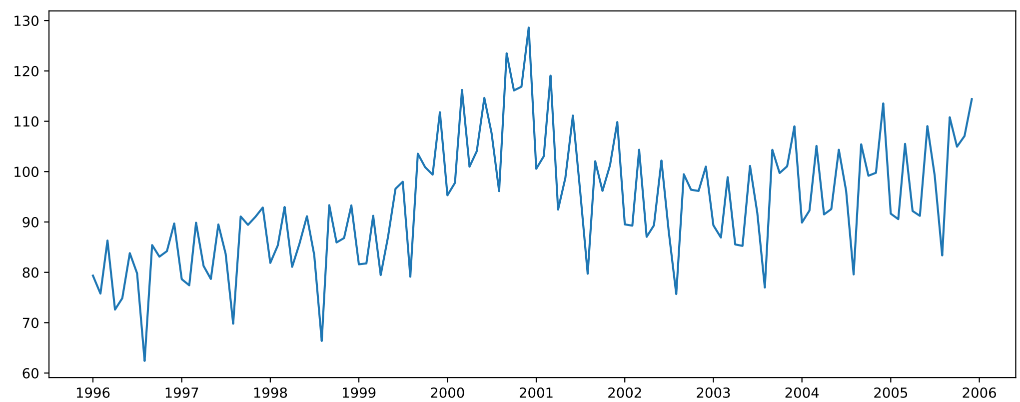

Figure 1: *Plot of the demand data*

It can be observed (figure 1) that there is an increasing trend in demand over the years. Another interesting observation is the seasonality i.e. the demand seems to follow a similar pattern every year. To examine this we take a closer look at the demand history by decomposing the data into its trend and seasonal components. We use `statsmodels` package in Python 3.7 to perform a decomposition of this time series. The decomposition of time series is a statistical task that deconstructs a time series into several components, each representing one of the underlying categories of patterns. With `statsmodels` we will be able to see the trend, seasonal, and residual components of our data.

Based on the charts in figure (2), it looks like the trend is always increasing at a steady rate except during the 5^{th} and 6^{th} years, where the trend shows a sudden rise and fall. From the plot below, we can also clearly see the presence of seasonality in the data.

Now that we have analyzed the data, let us forecast the demand values for the first year.

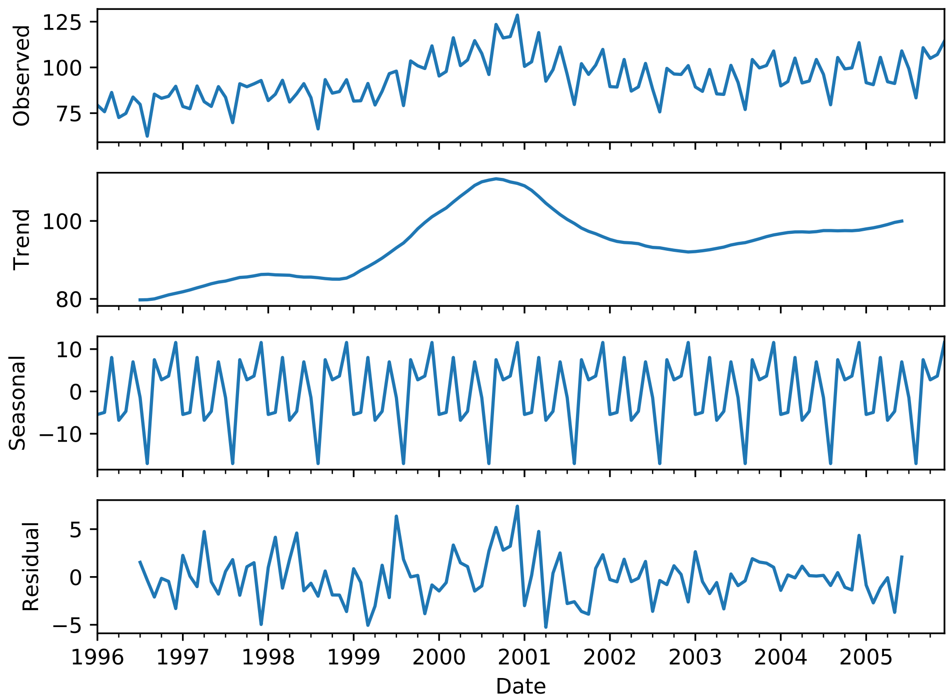

Figure 2: *Seasonal Decomposition of the demand history*

One of the widely used models for such type of data is the seasonal `ARIMA` model. In order to do this we will need to choose p,d,q values for the `ARIMA`, and P,D,Q values for the Seasonal component. There are many ways to choose these values statistically, such as looking at auto-correlation plots, correlation plots, domain experience, etc. One simple approach is to perform a grid search over multiple values of p,d,q,P,D,and Q using some sort of performance criteria. The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. The AIC value will allow us to compare how well a model fits the data and takes into account the complexity of a model, so models that have a better fit while using fewer features will receive a better (lower) AIC score than similar models that utilize more features. The `pmdarima` library for Python allows us to quickly perform this grid search and even creates a model object that you can fit to the training data. This library contains an `auto_arima` function that allows us to set a range of p,d,q,P,D,and Q values and then fit models for all the possible combinations.

Then the model will keep the combination that reported back the best AIC value.

Once, we find the optimal `SARIMA` parameters, we predict the demand for the first year i.e. 2006. After performing the predictive analysis, we find the order quantities, inventory and backorders for the first year. After observing the actual demand for the year 2006, we refit the `SARIMA` model again and predict the demands for the next whole year. One of the alternate ways to obtain a forecast would be to predict the demand for one time period and refit the model immediately after the demand is realized. However, because of the risk of overfitting, we reevaluate the model after one whole year.

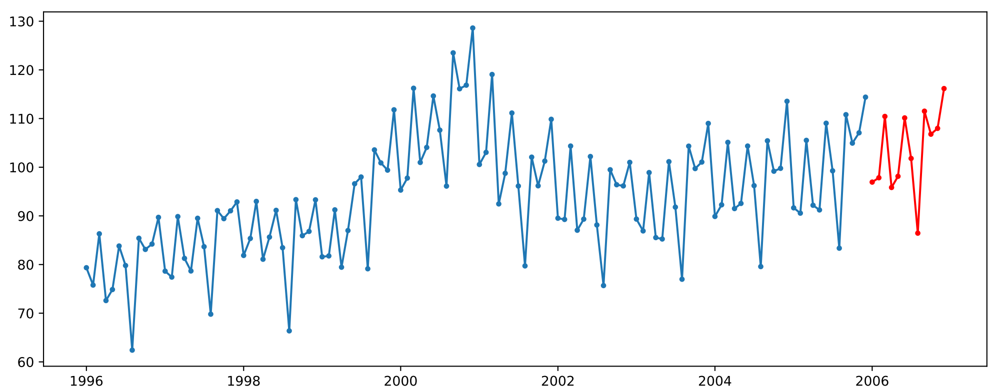

Figure 3: *Demand Forecast for year 2006*

1.1 Instructions for the Code - Methodology 1.0

We implement the above methodology using Python 3.7. Our python code requires the following libraries (or packages):

- `pandas`
- `numpy`
- `pmdarima`
- `statsmodels`

The program also needs two separate input files: `Ten-Year-Demand.csv` and `Two-Year-Demand-New.csv` and the format should be same as the files uploaded on `Github` and in the same folder as the python file. Essentially, copy and paste the demand values in the second column of the `Two-Year-Demand-New.csv` file. We use Pycharm as the IDE and it is advisable to use the same IDE. However, the code should run using any IDE, provided the packages are installed. After loading the code on to the IDE, just hit the `Run` button and the program will display the output in 2-4 minutes depending on the processor.

2 Methodology 2.0

Let the number of inventory left at the end of month t be s_t, the amount of backorder at the end of month t be r_t, the inventory cost at month t be z_t, and the amount of order at month t be x_t. At the beginning of month t, we have information about the amount of backorder/inventory from month $t-1$. The order placed at the beginning of month $t-1$ also arrives. The demand is the only uncertain parameter in this problem, which is defined by random vector ξ with sample space Ω. We assume that the demand follows a normal distribution with mean and variance obtained from our forecast. We apply sample average approximation (SAA) [1] by randomly drawing $|\Omega|$ scenarios from our forecast of demand with probability $\frac{1}{|\Omega|}$. In order to handle the uncertainty in the forecasted demand, we consider following stochastic model for month t:

$$\mathbf{P} : \min \ \mathbb{E}_{\xi}[(Q_t(x_{t-1}, \omega_t)] \tag{2}$$

$$\text{s.t. } \ x_{t-1} \geq 0 \tag{3}$$

where for given x_{t-1} and $\omega \in \Omega$, the second-stage recourse function is defined as

$$
\begin{aligned}
Q_t(x_{t-1}, \omega_t) = \min \;\; & z_t^\omega + 3r_t^\omega && (4)\\
\text{s.t.} \;\; & z_t^\omega \geq s_t^\omega && (5)\\
& z_t^\omega \geq 2s_t^\omega - 90 && (6)\\
& s_{t-1} - r_{t-1} + x_{t-1} - d_t^\omega = s_t^\omega - r_t^\omega && (7)\\
& z_t^\omega, s_t^\omega, r_t^\omega \geq 0 && (8)
\end{aligned}
$$

In the first-stage problem, the only non-negative decision variable is the amount of order at the beginning of month $t-1$, i.e., x_{t-1}. In the second-stage problem, the objective function (4) is to minimize the total inventory and backorder cost of month t. There are three non-negative continuous decision variables, z_t, s_t, r_t. Constraints (5) and (6) are related to inventory cost, which is piecewise linear function of the number of inventory s_t. Constraint (7) is the inventory balance constraint, where s_{t-1} and r_{t-1} are known fixed numbers obtained from realized demand. By utilizing stochastic programming procedure, we incorporate the uncertainty in our forecast demand, thereby giving us more robust decisions.

2.1 Instructions for the Code - Methodology 2.0

We again use Python 3.7 to implement the aforementioned methodology. The softwares needed to run this code are:

- `Gurobi`
- `ATOM IDE for Python`

Packages needed in Python 3.7:

- `pandas`
- `numpy`
- `gurobipy`
- `math`

- `sys`
- `statsmodels`

Following steps are used to run the code:

- Input the validate data into `Two-Year-Demand-New.csv` file, and keep the format as data given in `Ten-Year-Demand.csv`.
- Run the python code in `ATOM IDE for Python`.
- Results will be printed out in Python Console.

References

[1] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, 2006.

[2] Harvey M Wagner and Thomson M Whitin. Dynamic version of the economic lot size model. *Management science*, 5(1):89–96, 1958.