Computational OR Exchange

Haochen Jia, University of Southern California

October 2019

1 Introduction

This research aims at making proper inventory management from Jan 2006 to Dec 2007, based on the information of monthly demand between Jan 1996 and Dec 2005.
All input is the ten-year demand status of the product, which is saved as "Ten-Year-Demand.csv".

2 Methodology

This research implements a two-stage model. The first stage of the model works for predicting the demand for the new month, and the second stage of the model seeks to find the optimal inventory strategy.

2.1 First Stage: Prediction

The first stage of the model seeks to make the prediction of the demand. In this stage, we implemented the Long short-term memory (LSTM) based on the recurrent neural network for predicting. The reason for choosing LSTM is following:

1. Compares with other RNN structure, LSTM could properly keep both sufficient and valuable information from the time series.

2. It has far greater performance compared with time series models ARIMA and Facebooks' prophet, and machine learning models including random forest, decision tree, even the boosting.

From the data, we could easily find out the data have a strong yearly trend, which achieves a peak in Dec. and a valley in August. Information from the past 12 months will be sufficient to make a proper prediction. In the same time, choosing too long time steps may also cause problems, such as long training cost, and losing gradient.

2.1.1 Implementation

In this research, we use the first 8 years data (1996-2003) for training, and the last 2 years for validation, and pick time step as 12 months. We first pick all possible combinations of 12 consecutive months as features and the 13th consecutive month as the label. Now, the problem from a time series prediction problem to a supervised learning problem, with 83 input. Then, we train the LSTM model, with ReLU activation and MSE loss. Finally, we could make the prediction of the demand for the input of 12 month history. The prediction process of the model demonstrates in "prediction.py".

2.2 Second Stage: Finding the optimal inventory level

With the prediction from the first stage, we try to find the fittest inventory level in the second stage:
First, we assume the prediction $\hat{x}_i$ in month i is an unbiased estimator of the real demand x_i. The gap between $\hat{x}_i$ and x_i comes from unknown normally distributed noise. That means $\hat{x}_i - x_i \sim Normal(0, \sigma)$.
Then, with the prediction $\hat{x}_i$, we could generate the distribution of x_i.
Next, with the distribution of x_i, we could find the optimal $\hat{x}_i^*$ to minimize the cost in month i with hill climbing method. Because the cost function is convex, we could always get the optimal.

2.3 Implementation

To estimate the distribution of x_i, we use the Monte Carlo method to simulate 10000 possible $x_i^1 \cdots x_i^{10000}$ normal random variables with mean $\hat{x}_i$, and standard variance 2.06. 2.06 is the average standard error of prediction in the entire training set. For each set of 10000 samples of x_i, we could gain its average cost in month i.

$$cost_i = backorder + restorecost + ecost - esave$$

- backorder is summation of back orders with current inventory $\hat{x}_i^m$ and x_i^n demand, which is $x_i^n - \hat{x}_i^m \forall n \in 1, 10000 if x_i^n - \hat{x}_i^m > 0$.

- Restore cost is plantly the cost of current inventory time 10000 replications, which is max($\hat{x}_i^m * 2 - 90, \hat{x}_i^m$)*10000.

- The ecost is the extra storage cost of backorders hand over in the next month, which could be estimated by total inventory cost of ($\forall n, x_{i+1}^n+$ back orders) -total inventory cost of $\forall n, x_{i+1}^n$.

- The esave is the saved storage cost of backorders hand over in this month, which is the total inventory cost of ($\forall n, x_i^n+$ back orders) -total inventory cost of $\forall n, x_i^n$.

Then we could implement the hill climbing method to find the $\hat{x}_i^*$ with minimum $cost_i$:

1. We make the initial value $\hat{x}_i^1 = \hat{x}_i$, and the step length is $\delta = \frac{5}{k}$ in k-th iteration.
2. for iteration k, we could calculate the cost $\hat{x}_i^{k-1} + \sigma$ and $\hat{x}_i^{k-1} - \sigma$ respectively, and choose $\hat{x}_i^k$ with a smaller $cost_i$
3. With 500 iterations, the step size will be less than 0.01, we could find a heuristic approach of x_i as $\hat{x}_i^{500}$.

The inventory level optimizing process is demonstrated in "inventory.py".

3 Description of how to run the inventory system

With the inventory strategy of the previous method, we could implement the inventory system in the following process: With i between month Jan 2006 to Dec 2007,

1. The beginning inventory of the first month is fixed at 73.
2. For month i, the beginning inventory is $BI_i = \hat{x_i}^* + backorder_{i-1}$, which is optimal inventory strategy plus backorder of last month.
3. For month i, the total demand is $OQ_i = x_i + backorder_{i-1}$ which is the sum of demand for month i plus the backorder of the last month.
4. If $OQ_i > BI_i$, the backorder of the month i is $OQ_i - BI_i$, and the end of month inventory in the month i is 0.
5. If $OQ_i < BI_i$, the end of month inventory in the month i is $BI_i - OQ_i$, and the back order of month i is 0.
6. The holding cost in month i is $max(BI_i * 2 - 90, BI_i)$
7. The back order cost in the month i is 3* backorder of the month i.

This process demonstrates in "output.py"

4 Read me

1. The code is written in Python3, and fully tested on Mac. For making sure of code functioning, please run on Python 3.7 version from Anaconda.

2. Install the dependent library, including Numpy, Pandas, Sklearn, Keras. Also, it needs Tensorflow environment support.

   - We should install in the following order: Numpy, Pandas, Sklearn, Tensorflow, and finally Keras.
   - To install Numpy, Pandas, Sklearn, please run "conda install numpy";"conda install pandas";"conda install sklearn".
   - To install Tensorflow properly, please run following code:
"conda create -n tf tensorflow"
"conda activate tf"
to install Tensorflow CPU version.
   - To install Keras, please run "conda install keras"
   - Now we have finished the preparation.

3. Rename the input file as "Ten-Year-Demand.csv" and put it under the same path as "main.py".

4. Run "main.py".

5. The report will be generated as "report.csv" under the same path. It includes the information of beginning inventory, order quantity, ending inventory, holding cost, and backorder cost.

6. Meanwhile, total and average holding costs, and total and average backorder costs will be printed on the terminal.