Project Report for Introduction to Data Mining

Time-series Data Trend Prediction in E-Commerce

College of Information Science & Electronic Engineering,
Zhejiang University
Liu Jie
3160102821
jayloong@zju.edu.cn
2019/6/13

Abstract

This model is used to predict the overall sale quantity of 100 key products in future based on past data. The report started from the data preprocessing following the normal data preprocessing process, and then try two methods, including ARIMA and LSTM, to predict the overall sale quantity for all the 100 key products for each day between the 118th to 146th day.

Keywords: Quantity Prediction, LSTM

I. Data Preprocess

Data Format Transfer and Data Integration(The first block in jupyter)

The datasets, including buyer information and product information, are given in 7 tables in txt format, which is inconvenient to retrieve the specific data. So the 7 tables of data are transferred to 2 python-based dictionaries data structure in json format. The first dictionary record the buyer information with the key of "basic_info", "quantity", "money" and "average_money". The second dictionary record the product information with the key of "basic_info" and "trade".

Data Redundancy detection and Data Reduction(The second block in jupyter)

With the manual detection, the maximum buyer_id in "buyer_basic_info.txt" is 31490, while the maximum buy_id in "trade_info_training.txt" is 91491. After search, the total number of buyer who make contribution to the quantity of production is 34927. But the half of them have no data. If we take the relationship of buyer and product into consideration, we don't know the actual value of these missing data, which is not helpful for the result. **So the datasets related to buyer are dropped.**

For the product data, we only know the trading information about the 100 key products. With the task of predicting the overall quantity of 100 key products, we could only focus the data about key products. Then, I sum the quantity of 100 key products respectively each day and compare with the data in "product_distribution_training_set.txt". **They are consistent.** To my surprise, the data in "trade_info_training.txt" conclude the sale quantity in 118th day.

After printing the attribute2 for key products, we can find it clearly that all the attribute2 of key products is 0, which means we can drop the attribute2.

Then, I roughly analysed the correlation between attribute1, price and sale quantity in 118 days by scatter plot. But it didn't help.

Conclude

After data preprocessing, all the data used to predict the overall sale quantity of key products are attribute1, original price and the quantity of each key products from 0th day to 117th day.

II. Model Building

ARIMA(The third block in jupyter)

Because the data was non-seasonable and non-stationary, it is challenging to use ARIMA model to converge the data after trying my best. So I don't talk more about it here. The related progress was showed in jupyter. (PS: THIS PART WAS COMPLETED WITH REFERENCE IN THE INTERNET)

LSTM(The fourth block in jupyter)

LSTM is one of the Recurrent neural networks, having excellent power to process the time series. The details of LSTM will not be discussed here.

I build a single layer LSTM with 32 hidden neurons with consideration of a small amount of data. Each time input,9 dimensional vectors, consist past 7 days

quantity, attribute1 and the original price of product. All of them have been normalized. The output is the sale quantity in 8th day.

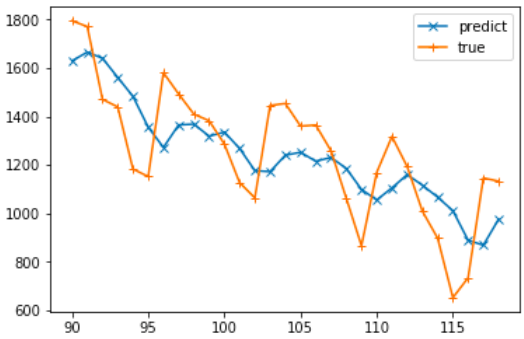

Figure1. The prediction data vs. True data

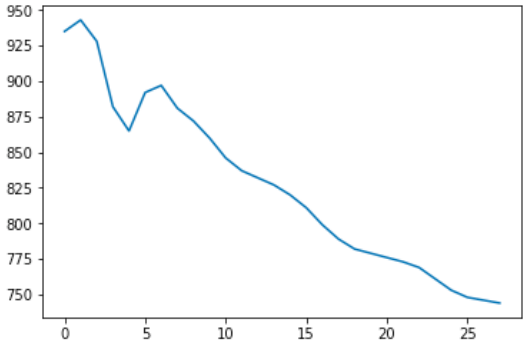

Figure2. The result

I separate the data in two parts. The first part, 0th day to 90th day, is used to train the LSTM model. The rest data is used to validate the performance of the model. After 1847 iterations, the model seems to converge because the error in training datasets and validation datasets become smallest. So I stop training the model and compare with the prediction data with true data. The I use the last 7 days data in 118 days to predict the future quantity of each product recurrently. Then I sum up the quantity and get the overall sale quantity.

At last, I just stored the most important file in the directory. If you are interested in the process, you can run the code in “data_mining_prj.ipynb” step by step. Some important comments has been written besides the code.