VI. Clickstream Big Data and "Delivery before Order Making" Mode for Online Retailers

Yeming (Yale) Gong
EMLYON Business School

Haoxuan Xu*, Jinlong Zhang
School of Management, Huazhong University of Science &Technology

Abstract

Our research is inspired by a leading online retailer using clickstream big data to estimate customer demand and then ship items to customers or hubs near customers by a mode of "delivery before order making" (DBOM) mode. Using clickstream data to obtain advance demand information in order quantities, we integrate the forecasting with a single-item uncapacitated dynamic lot sizing problem in a rolling-horizon environment. Using the simulated clickstream data, we evaluate the performance of DBOM mode.

1 Introduction and Literature

A leading online retailer, with 10 billion USD turnovers in China, uses big data of online clicks and data mining algorithms to estimate the expected order quantity in different locations including collection locations, locker locations and hubs, then ships items to the locations by a mode of "delivery before order making" operational mode. Amazon, another leading online retailer in USA, has lately announce a new distribution method AS ("anticipatory shipping", see [1]), specifying a method to start shipping packages before customers really buy products. Amazon AS method can predict the customer demand to obtain the geographical destination area information by analyzing different variables, including historical ordering behavior, wish-lists, clicking data. The packages are in transit or waiting at a hub until an order arrives, and then shipped to the specific location quickly.

Inspired by these new logistics modes using big data, this paper addresses an operational problem concerning the use of a kind of big data—clickstream data—in a

specific online retailing environment. Lee et.al [2] define the clickstream data of online stores to be the paths information from visitors. Many researchers have studied the marketing benefits of using clickstream data, or rather clickstream tracking, in e-commerce settings. For a detailed review, see [3] and references therein. Different from this stream of research, we investigate the benefits of clickstream data from an operational perspective. Huang and Van Mieghem [4] specify the cost-saving effects on inventory management for a specific company by using clickstream tracking data on its "non-transactional" website. To our knowledge, existing literature has not studied the operational benefits of clickstream data in an environment of online retailing.

A natural question arises whether or not online retailers can use such clickstream data to mitigate the demand uncertainty and improve the inventory management process. Agatz et al [5] indicate that delivery and after-sales service are becoming key competitive factors in today's e-commerce. Hence, the match of supply and customers' demands is essential for online retailers to assure fast delivery and good service. Nonetheless, a trade-off exists between the highly guaranteed stock and demands uncertainty. Unlike demand forecasts in traditional offline transaction setting, which is usually based on historical data, online retailers can better predict the future demands by further using clicking data before customers placing orders (see [6]).

In an online retailing environment of selling perishable or customized products (e.g., produce, assembled computers and jewels), retailers may expect a timely supply or fast production, and face time-varying demands. Based on the historical trading data, inventory managers can forecast future demands and treat them as deterministic data (e.g., the mean value), then model the inventory replenishment processes as dynamic lot sizing (DLS) problems. Based on this application setting, this paper specifies the use of clickstream data in an integrated dynamic inventory control policy. We first develop an adaptive forecasting method by the use of clickstream data to better predict the future demand pattern. Then, we embed such advance demand information into a DLS model in a rolling-horizon environment. Given that the clickstream data evolves dynamically, we update the demand information accordingly.

Gallego and Özer [7] classify advance demand information (ADI) into observed and unobserved parts. The observed part of ADI is easy to obtain for online retailers when customers place orders online, since they are usually satisfied several periods later. This is also the case in some traditional retailing and production settings when the requirements of some products or components are released in advance. As for the unobserved part of ADI, of which traditional retailers has no information, online retailers can use clickstream data to get access. Although researchers extensively investigate the value of ADI in operational management, few of them explore to obtain the ADI for online retailers by using the clickstream data. Using the simulated clickstream data according to real online retailing environment, we examine the cost saving effect and fast delivery effect of our inventory model.

2 Formulation

2.1 Problem Description

We consider an online retailer maintaining its own stock for a certain commodity. The manager needs to develop a good inventory control policy to minimize the related production/purchasing cost and inventory cost. Before applying a certain inventory model, it is necessary to identify the demand pattern. As usual, one can make use of the historical demand data to predict the future demands. However, it is not that accurate since many unseen factors exist.

A typical feature of online purchasing is that customers generate a large amount of clickstream data which could be tracked by online retailers. Our problem is to explore the use of such "big data" in predicting a more accurate future demand pattern. This task can be handled by a suitable algorithm in machine learning theory. Specifically speaking, we apply an on-line algorithm into the forecasting process. That is, after the learning model predicts, the true result will be revealed and act as feedback to update the algorithm accordingly. The algorithm then adaptively makes a proper prediction.

Without any assumption on the demand distribution, we use the machine learning algorithm described above to mine the clickstream data for predicting future demand. Then we incorporate it into a dynamic lot-sizing model to solve the replenishment problems in a rolling-horizon environment for this online retailer. The dynamic lot-sizing models are widely used by online retailers since they widely use ERP systems containing a MRP modular to make replenishment plans (See [9]).

2.2 A Clickstream-based Adjusted Rolling DLS

We develop an operational decision framework to improve the inventory control policy for online retailers. We first develop an adaptive demand forecasting method, which includes two minor steps. At the first step, we incorporate the historical demand data of a certain commodity into a traditional forecasting algorithm to generate an initial demand pattern. At the second step, we apply an on-line machine learning algorithm, winnow algorithm (see [8]), to adaptively forecast the demand of the nearest future period by using the latest clickstream data. Thereby, the predicted demand pattern is updated. This process is dynamically evolved as time goes on. After the predicted demand data is obtained, we incorporate it into a lot-sizing model in a rolling horizon environment to dynamically make the replenishment plans with an objective of minimizing the total inventory related cost. The overall decision framework is shown in figure 1.

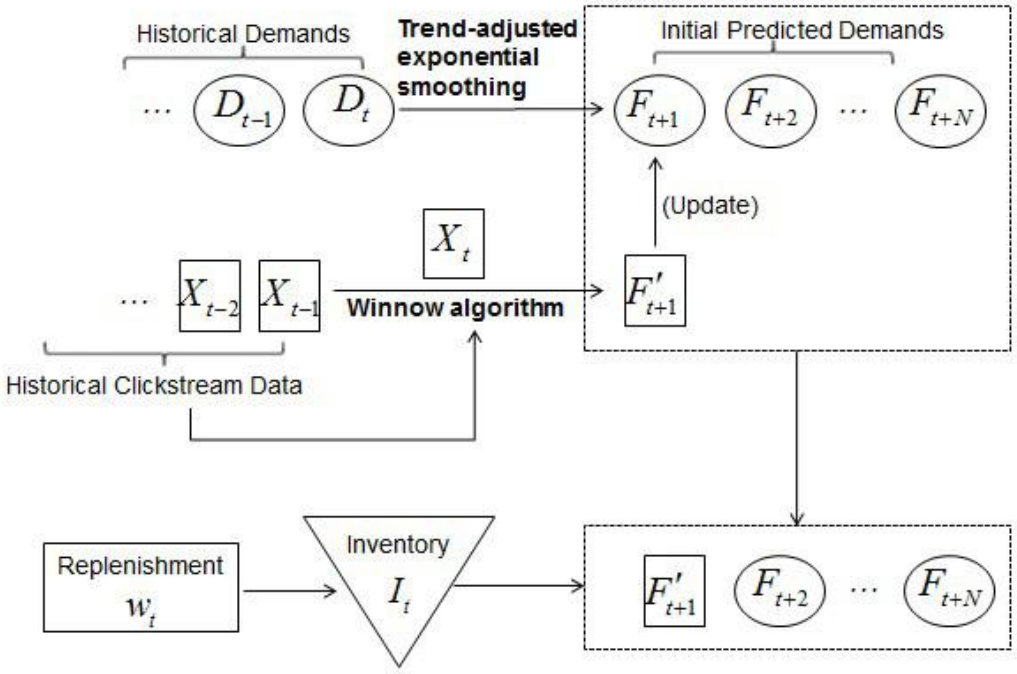

Figure 1: Clickstream-based Adjusted Rolling DLS

2.2.1 Using Clickstream Data in Demand Forecasting

Managers estimate the demand pattern before making any specific replenishment decisions. Traditional forecasting relies heavily on a demand history. Different from the mode in conventional retailing, online retailers can not only observe a demand history, but also obtain a clickstream history. Using only the historical demand data to estimate the future demand may lead to deviations, since a lot of fluctuations exist. As a result, we design a dynamic procedure to adaptively forecast a more accurate demand. Based on the historical demand data, we first apply a trend-adjusted exponential smoothing (TAES) algorithm to generate an initial predicted demand. Then we use the clickstream data with an on-line machine learning algorithm to dynamically update the forecasting.

2.2.1.1 A trend-adjusted exponential smoothing algorithm

Demand history provides valuable information for online retailers to predict the future demand. In this paper, we adopt a TAES algorithm to generate the initial predicted demand pattern. The algorithm uses two parameters, α and β, as coefficients for the average demand and its trend, respectively. The following equations are the forecasting

algorithm:

$$A_t = \alpha D_t + (1-\alpha)(A_{t-1} + T_{t-1}) \quad (1)$$

$$T_t = \beta(A_t - A_{t-1}) + (1-\beta)T_{t-1} \quad (2)$$

$$F_{t+1} = A_t + T_t \quad (3)$$

D_t : Demand in period *t*;

A_t : Exponentially smoothed average of the series in *t*;

T_t : Exponentially smoothed average of the trend in period *t*;

F_{t+1} : Predicted demand in period *t*;

α : Smoothing parameter for the average $(0 < \alpha < 1)$;

β : Smoothing parameter for the trend $(0 < \beta < 1)$;

Using this forecasting method, we predict the demands of the following planning horizon from t to $t+N$. We denote them by a demand vector $F = (F_{t+1}, F_{t+2}, ..., F_{t+N})$. In practice, many conventional retailers, even online retailers, just finish the forecasting process here, while this is just the initial predicted demand in our forecasting method. In the following, we use clickstream data to update and improve the forecast.

2.2.1.2 A winnow algorithm

Winnow is a typical on-line machine learning algorithm, which is firstly developed by Littlestone [8]. Based on the variables of the clicking examples, winnow keeps learning the weights of each variable and makes a binary prediction of whether a visit/click leads to a purchase. In an on-line setting, once the algorithm makes a prediction, the real value is revealed then and gives feedback to the algorithm. A simple version of winnow algorithm is as follows:

- Step 1. Initialize each weight ω_i of variable x_i to 1;
- Step 2. Given a clicking example $x = \{x_1, x_2, ..., x_n\}$:

  $$\begin{cases} \sum_{i=1}^{n} \omega_i x_i \geq n, & output\ 1 \\ \sum_{i=1}^{n} \omega_i x_i < 0, & output\ 0 \end{cases}$$

- Step 3. The weights of the variables are updated when the algorithm makes a mistake:

  a). If the algorithm predicts 1 and the true value is 0, then $\omega_i = p\omega_i, 0 < p < 1$;

  b). If the algorithm predicts 0 and the true value is 1, then $\omega_i = q\omega_i, q > 1$;
- Step 4. Go to 2.

Using the historical clickstream data as the training set, we can obtain an updated vector of the weight of each variable. Applying the weight vector to the latest clickstream data as the test set, the winnow algorithm can make a good prediction of those clicks in period *t* leading to purchasing in period *t+1*. Thereby, we can use this information to update the demand of the nearest future period, i.e. F_{t+1}. As time goes on, the predicted demand vector F can be dynamically updated by combining these two algorithms.

2.2.2 A rolling-horizon lot-sizing model

Replenishment or production planning problems in online retailing are usually solved in a dynamic, rolling-horizon pattern. At first, say in period *t*, a decision problem is solved to optimality in a planning horizon of given length *T*. The manager then will implement the first-period decision for *k* periods in the resulting solution. Afterwards, the system evolves to period *t+k*. Obtaining the updated demand information, the manager has to make the next decision. This process is repeated under such rolling framework (see [11]).

At the second step of our operational decision framework, given the updated demand information of a new forecast horizon obtained at step one, we apply a single-item uncapacitated DLS model to formulate the inventory replenishment problem. In a rolling-horizon environment, although we dynamically obtain a new demand vector F for the next forecasting horizon, we can regard any review period *t* as the beginning of a new forecast horizon when making decisions. Using the result at step one, we get the predicted integer demands of *T* periods, i.e. $F = \{F_{t+1}^{'}, F_{t+2}, ..., F_{t+T}\}$. At any period *t* when we need to make decisions, we reset *t=1*, and have the following lot-sizing problem:

$$Min \sum_{t=1}^{T} (k_t y_t + p_t w_t + h_t I_t) \tag{4}$$

$$\text{S.T.} \quad I_0 + w_1 = I_1 + F_1(x_1, x_2, ..., x_n) \tag{5}$$

$$I_{t-1} + w_t = I_t + F_t(\alpha, \beta), \quad t = 2, ..., T \tag{6}$$

$$0 \le x_t \le M y_t, \quad t = 1, ..., T \tag{7}$$

$$w_t, I_t \ge 0 \tag{8}$$

$$y_t \in \{0,1\} \tag{9}$$

In the above model, *T* is the forecast horizon, k_t is the fixed ordering cost in period *t*, p_t and h_t denote unit purchasing cost and unit holding cost alternatively in period *t*. I_t is the inventory at the end of period *t*, y_t is a binary decision variable indicating whether to replenishment in period *t*. w_t denotes how much to replenishment in period *t*. *M* is a very large number. $F_1(x_1, x_2, ..., x_n)$ is the demand of the first period, which is decided by the winnow algorithm using the clickstream $x_1, x_2, ..., x_n$. $F_t(\alpha, \beta)$ is the demand beyond the

first period, which is decided by the trend-adjusted exponential algorithm with parameter α and β.

3 Analysis

In this section, we analyze how to apply our clickstream-based adjusted rolling DLS decision framework in an online retailing environment through a simulated example. Since the TAES algorithm is a typical time-series forecasting technique and easy to be executed in EXCEL. We can directly use it to obtain the initial predicted demands based on historical demands.

The key function of winnow algorithm is to disjunct the most important variables and to make a good prediction. The online purchasing behaviour may be correlated with thousands of input factors. Using a specific variable selecting technique, Van den Poel and Buckinx [10] identify nine key variables out of 92 possible measures in predicting whether a visitor will purchase during her next visit. While Van den Poel and Buckinx [10] focus on the visitor level, we focus on the product level, i.e., whether a visit to a certain product will lead to a purchase of this product. Based on [10], we use variables shown in Table 1 for the winnow algorithm.

Table 1: Variables of winnow algorithm

Variables	Definition	Description
x_1	the visitor is a registered member or not	1 yes and 0 not
x_2	the customer visited during last period or not	1 yes and 0 not
x_3	the customer visited before last period or not	1 yes and 0 not
x_4	the visitor clicks the personal pages or not	1 yes and 0 not
x_5	the visitor clicks only this product or not	1 yes and 0 not
x_6	the visitor supplies personal information or not	1 yes and 0 not
x_7	whether the customer purchase this product before	1 yes and 0 not
x_8	whether the average time per click is higher than the average	1 yes and 0 not

We then build a basic winnow classifier in MATLAB to judge whether a click leads to purchase. The classifier works by the following steps:

Step 1: Initialize each weight $\omega_i = 1 \, (i = 1, ..., m; m = 8)$;

Step 2: Apply the training set to adjust the weight ω_i:

- If $\sum_{i=1}^{m} \omega_i x_i \geq \theta$, and the click does not lead to a purchase, then reduce the weight ω_i of those $x_i \neq 0$ to $\omega_i = p\omega_i$ $(0 < p < 1)$, till $\sum_{i=1}^{m} \omega_i x_i < \theta$;
- If $\sum_{i=1}^{m} \omega_i x_i \leq \theta$, and the click does leads to a purchase, then increase the weight ω_i of those $x_i \neq 0$ to $\omega_i = q\omega_i$ $(q > 1)$, till $\sum_{i=1}^{m} \omega_i x_i > \theta$;

Step 3: Apply the updated weight ω_i obtained in the training set to the test set, calculate $\sum_{i=1}^{9} \omega_i x_i$ and compare it to the threshold θ, then predict if a click will lead to a purchase.

We use MATLAB to generate a 50-period demand vector based on a normal distribution N (20, 5). Then we randomly generate 500 clicks for each period, each click with a feature vector $(x_1, x_2, ..., x_8)$ and a "purchase or not" indicator (1 stands for purchase and 0 not). The sum of the indicators in each period is equal to the true demand of that period. In our example, the average rate of conversion from click to purchase is 3.58%, which is reasonable in e-commerce setting according to [6].

We divide the 50-period data into two sets, the former 25 periods as the training set and the latter 25 periods as the test set. By setting $p = 0.9$, $q = 2$ and the threshold $\theta = 0.5$, we first use the winnow classifier in the training set to obtain an updated weight vector, and then use this vector to predict whether a click in the test set will lead to a purchase. Figure 2 shows a comparison between the performance of the clickstream-based winnow algorithm and the TAES algorithm ($\alpha = 0.8, \beta = 0.7$) in predicting the demands in the test set. We find that the clickstream-based algorithm is much better than the TAES algorithm.

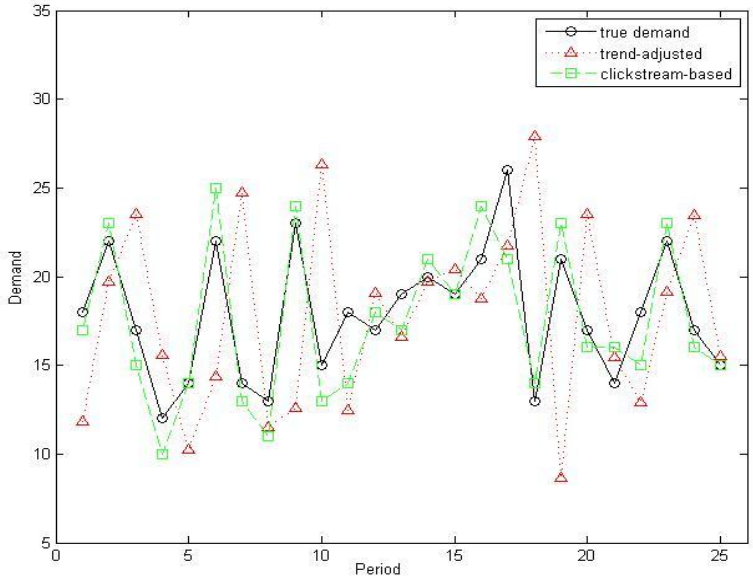

Figure 2: Comparison between TAES and clickstream-based winnow

We then solve a rolling-horizon dynamic lot sizing problem for the 25-period test set. We assume the fixed cost $k_t = 40$, purchasing cost $p_t = 0$ and holding cost $h_t = 2$ for all $t = 1,...,25$. We use the rolling schedule described in [11] to solve our problem. The forecast horizon *T* is chosen to be 2, 3, 4, 5, 6, 7, 8, separately. The demand of the first period in the forecast horizon is predicted by the winnow algorithm and the rest demands are predicted by TAES algorithm. Only the first decision in the optimal solution of the forecast horizon is implemented, then the process rolls to the next decision period to solve another DLS problem with a planning horizon of *T*. The schedule ends when reaching the 25th period. Table 2 shows the percentage deviation of the cost from the optimality, which is obtained by solving the entire 25-period DLS problem with the true demand. We find that the cost of using the clickstream-based demand is closer to optimality than using demand obtained by TAES only.

Table 2: Percentage deviation from optimality

Forecast horizon	TAES demand only	Clickstream-based demand
2	3.5%	1.3%
3	5.3%	2.9%
4	5.7%	3.3%
5	5.7%	3.3%
6	5.7%	3.3%

7	5.7%	3.3%
8	5.7%	3.3%

4 Concluding Remarks

This paper presents an integrated clickstream-based operational decision framework for online retailers. It explores the use of an on-line machine learning algorithm, winnow algorithm, to mine clickstream big data and improve the demand management process, initially based on traditional forecasting method. Applying the updated demand information in a rolling-horizon dynamic lot sizing problem, we analyze its cost advantage over traditional forecasting method. Our current clickstream mining algorithm can only predict whether a click will lead to purchase or not, but cannot predict the quantity a purchase contains. It is interesting to explore other algorithms to solve the problem.

Acknowledgements

This research is supported by Collaborative Innovation Center for Modern Logistics and Business of Hubei (Cultivation), Modern Information Management Research Center (MIMRC) of HUST and NSFC (No.70901028; 71271095).

References

[1] Spiegel, J., McKenna, M., Lakshman, G. and Nordstrom, P., "Method and system for anticipatory package shipping", *US Patent*, 8, 615, 473 (2013).

[2] Lee, J., Podlaseck, M., Schonberg, E. and Hoch, R., "Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising," In *Applications of Data Mining to Electronic Commerce*, 59–84. Springer, US (2001).

[3] Hui, S. K., Fader, P. S. and Bradlow, E. T., "Path Data in Marketing: An Integrative Framework and Prospectus for Model Building," *Marketing Science*, 28, 2, 320-335 (2009).

[4] Huang, T. and Van Mieghem, J. A., "Clickstream Data and Inventory Management: Model and Empirical Analysis," *Production and Operations Management*, 23, 3, 333-347 (2014).

[5] Agatz, N. A., Fleischmann, M. and Van Nunen, J. A., "E-fulfillment and Multi-channel Distribution - A Review," *European Journal of Operational Research*, 187, 2, 339-356 (2008).

[6] Moe, W. W. and Fader, P. S., "Dynamic Conversion Behavior at E-commerce Sites," *Management Science*, 50, 3, 326-335 (2004).

[7] Gallego, G. and Özer, Ö., "Integrating Replenishment Decisions with Advance Demand Information," *Management Science*, 47, 10, 1344-1360 (2001).

[8] Littlestone, N., "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine learning*, 2, 4, 285-318 (1988).

[9] Gunasekaran, A., Marri, H. B., McGaughey, R. E. and Nebhwani, M. D., "E-commerce and its impact on operations management," *International Journal of Production Economics* 75, 1, 185-197 (2002).

[10] Van den Poel, D. and Buckinx, W., "Predicting online-purchasing behavior," *European Journal of Operational Research*, 166, 2, 557-575 (2005).

[11] Baker, Kenneth R., "An experimental study of the effectiveness of rolling schedules in production planning," *Decision Sciences*, 8, 1, 19-27 (1977).