Demand Prediction, Predictive Shipping, and Product Allocation for Large-scale E-commerce

Xiaocheng Li,[a] Yufeng Zheng,[b] Zhenpeng Zhou,[c] Zeyu Zheng[d]*

[a] Department of Management Science and Engineering, Stanford University; [b] Department of Industrial Engineering and Management, Shanghai Jiao Tong University; [c] Department of Chemistry, Stanford University; [d] Department of Industrial Engineering and Operations Research, University of California, Berkeley

Contact: chengli1@stanford.edu, dafengge@sjtu.edu.cn, zhenpeng@stanford.edu, zyzheng@berkeley.edu

In this paper, we study the problems of multiple-product demand prediction, predictive shipping mechanisms, and products allocation across multiple warehouses for large-scale e-commerce. These research problems are triggered by an exploratory data analysis on the transactional level data from Alibaba and its logistics arm Cainiao, including detailed transaction orders, inventory and logistics for 7,013 different products and 130 warehouses. First, we develop a multiple-product demand prediction system and identify unique features presented in the data, which improves prediction accuracy significantly compared to standard machine learning models. A new clustering-based regularization method is developed with the aid of representation learning to capture the product interactions, augment data and prevent over-fitting. Second, we propose and analyze in theory a novel shipping mechanism - Predictive Shipping, which utilizes demand prediction to arrange shipping before orders are placed. We provide analytical bounds on the performance improvement via this new mechanism, which originates from the improvement on demand prediction. Third, we formulate and solve a large-scale products allocation problem across warehouses and show that a change in the current products allocation distribution of Cainiao can potentially be beneficial. Numerical experiments with both real data and synthetic data are conducted to demonstrate our findings.

Key words: Demand prediction, predictive shipping, product allocation, exploratory data analysis, logistics data visualization

1. Introduction

With the advancement of information technology, business data are being amassed at unprecedented levels. Large e-commerce companies such as Alibaba (a Chinese multinational conglomerate specializing in e-commerce, retail and Internet) and its logistics arm Cainiao, while providing an integrated platform of sales, online payment, inventory

* The authors are grateful to the committee of 2018 MSOM Data Driven Research Challenge and Alibaba/Cainiao for providing the data and helpful comments.

management and logistics, collect high-quality customer, market, and operational data. This growth in data availability stimulates *predictive analytics*, provides opportunities to enhance solutions for various operations management problems and allows researchers and practitioners to identify new research opportunities. As Sodhi and Tang (2010) point out, "with the explosion of available data, increased computer power, and advances in modeling methods, every organization needs to find ways to use this data to make better decisions faster."

In this paper, we start with an extensive analysis of the business data (at the level of 100GB) from Alibaba and its logistics arm Cainiao. Driven by the findings in our data analysis, we identify three aspects in which opportunities of improvement naturally arise: demand forecasting, shipping regimes, and product allocation strategies. Our data analysis includes (i) customer orders from January 2017 to July 2017 (in total over 40 million threads); (ii) detailed sell-side and buy-side transaction behavior; (iii) the inventory record of all 7,013 products stored in 130 warehouses owned by Cainiao, including inventory levels, allocation, replenishments, and transfers; (iv) the fulfillment logistics data of each order, including dispatching, shipping and delivery. We summarize our contributions as follows.

(1) We develop a joint demand prediction system for multiple products in Section 4. Both historical demand data and other auxiliary information such as price and reviews are utilized. We show that the demand prediction of one product can be improved by exploiting information from other products. Furthermore, we propose a clustering-based model regularization method to augment data and prevent over-fitting. We compare our new prediction model to other benchmark machine learning models and observe significant accuracy improvement.

(2) We propose a new shipping regime - predictive shipping (PreShip) in Section 5. In this regime, shipping is arranged based on the demand prediction and initiated before customers place orders. The optimal PreShip quantity is determined by a stochastic optimization problem, the inputs of which are distributional predictions of the future demand. The goal of PreShip is to lower the total cost given the same delivery time requirement (or, equivalently, to speed up delivery given the same cost). The effectiveness of PreShip originates from the accuracy improvement in demand prediction and we quantify how the accuracy improvement translates to cost reduction. The feasibility of PreShip, in practice, can be supported by the observation that the 130 warehouses typically have significant vacancy, while most products are only stored in a limited number of warehouses.

(3) We propose an approach for multiple-product allocation in multiple warehouses in Section 6. We adopt a linear programming formulation to optimize the averaged outbound shipping cost, which can be solved efficiently for a large-scale problem. We analytically study the sensitivity of outbound shipping cost relative to a change of scheme in product allocation and quantify the marginal effect of increasing product allocation density across warehouses.

(4) In the data analysis, we design and use data visualization tools that are specialized for large online commerce platforms. In Section 3, we visualize our findings to offer insights and illustrate how the operational problems are identified.

The remainder of this paper is organized as follows: Section 2 reviews related literature; Section 3 presents the exploratory data analysis; Section 4 describes the demand prediction system; Section 5 introduces and develops theory for the new Predictive Shipping model; Section 6 discusses the product allocation problem across warehouses; Section 7 concludes with numerical experiments on both real data and synthetic data.

2. Literature review

Effective demand forecasting in a data-rich environment is crucial to service and operations management, production planning and revenue management. Numerous statistical and machine learning models have been developed for this purpose, including times series models, artificial neural network, support vector machine (see (Carbonneau et al. 2008), (Slimani et al. 2015), (Chen et al. 2004), (Choi et al. 2018), and references therein). Another stream of work explored the web data such as search queries and online reviews to improve prediction accuracy (see (Choi and Varian 2012), (Cui et al. 2015), (Boone et al. 2018) and (Chong et al. 2017)). While the conventional prediction models rely heavily on the historical demand data, these results inspire people to utilize new external information available in the big data era. Some other empirical studies on large e-commerce companies and retailers can be found in (Hortaçsu and Nielsen 2010), (Hwang and Park 2015), (Ahire et al. 2015), and (Cui et al. 2018).

Successful applications of the big data and machine learning techniques go beyond demand prediction. In recent years, researchers revisited various operations management applications from a big data perspective. These applications, to name a few, include the newsvendor (Ban and Rudin 2018), inventory management (Bertsimas et al. 2016)

and (Huang and Van Mieghem 2014), revenue management and pricing (Ferreira et al. 2015, Aral and Walker 2014). Our paper contributes to both demand prediction in operations management and the development of new data-driven decision models. Conventional approaches typically consider the demand prediction in a single-product setting. Our paper investigates the demand prediction problem in a multiple-product context that integrates three aspects: historical demand data across products, covariates information, and new features constructed by clustering and representation learning (Mikolov et al. 2013, Grover and Leskovec 2016). The discovery of these new features, from an operations management perspective, is analogous to facilitating information sharing between different products (see discussions on information sharing in supply chain in (Cachon and Fisher 2000, Lee et al. 2000, Cui et al. 2015)).

3. Exploratory data analysis

In this section we describe key results from the data analaysis.

3.1. The Scope

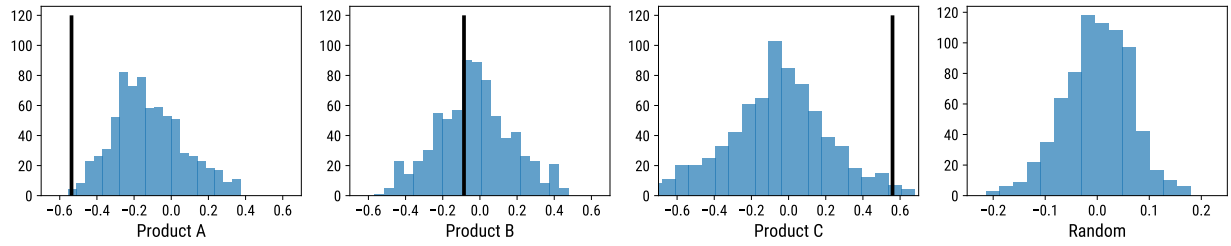
The decision problems we consider naturally arise from the data analysis. In this analysis, we focus on the products whose inventories are managed in Cainiao's warehouses (with a total number of 7,013). The other products managed by individual merchants are not considered, as their inventory management and shipping decisions are not controlled by Alibaba and Cainiao. We devote this section to present the procedure of the data analysis and the associated observations. The exploratory process ranges from order to delivery. We examine the sales and demand data, identify the dependence structure in 3.2 across multiple products and among different features, visualize the current product allocation pattern of Cainiao in 3.3, show the current warehouse replenishment decisions in 3.4, and show the curent delivery performance bottleneck in 3.5. As discussed by Wang and Mersereau (2017), there is a subtle difference between the demand and sales due to the data unobservability issue. Here, we ignore this issue because out-of-stock events are hardly observed for products whose inventories are managed by Alibaba and Cainiao.

3.2. Cross-product Demand-price Relationship

It is known that the price of a product determined by the merchants impacts the demand. In this subsection, we show that in the Alibaba/Cainiao data, the demand for a product can also depend on the price of the other products.

We denote the demand and the price of the i-th product on the t-th day as D_t^i and P_t^i, respectively, where $i = 1, ..., I$ and $t = 1, ..., T$. Demands on special shopping festivals are treated as outliers and removed. Indeed, predicting the surging demands during shopping festivals is itself an intriguing problem, but is not the focus of this paper. From the data, we compute the sample Pearson correlation R_{ij} between i-th demand vector D^i and j-th price vector P^j. For product i, we analyze the correlation between its demand with the price of each product, namely the vector $R_{i\cdot} = (R_{i1}, R_{i2}, ...R_{iI})$. Figure 1 displays the histograms of $R_{i\cdot}$ for three different products and a benchmark histogram as the fourth subfigure. We note that even for a product whose demands by definition do not depend on the price of other products, such a histogram of empirical correlations would also demonstrate a similar Gaussian shape. However, the magnitude of absolute correlations would not be as high. The last subfigure is the histogram of a correlation vector for independently generated prices and demands with the same number of products and the same length of time periods. Note that the magnitude of correlations lies in $[-0.2, 0.2]$ with high probability, which is significantly less than those observed in real data.

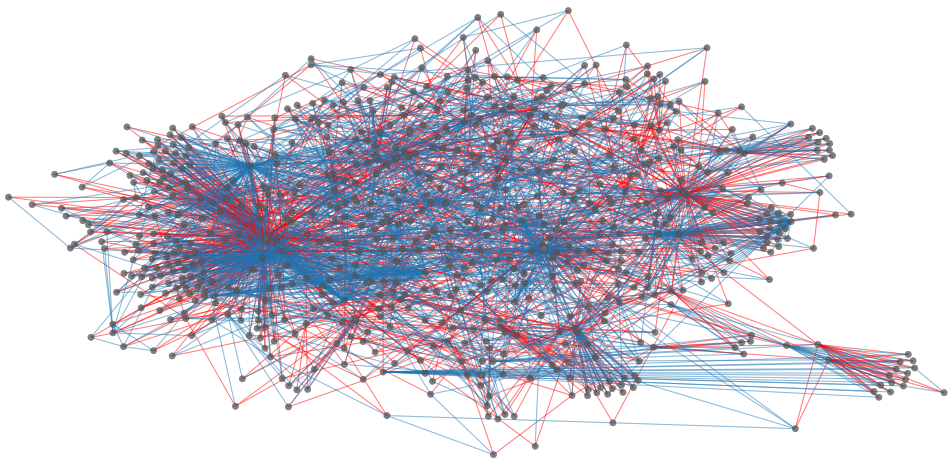
Figure 1 **Histograms of Demand-price Correlations: The vertical black lines represent the correlation between the product's demand and its own price.**



This observation indicates that involving the price of other products as a feature has a potential of significantly improving the demand prediction power. We next propose an approach to identify a list of dependent products to include for the demand prediction of a given product. For each product i, we identify a substitute relationship for the k largest R_{ij} and a complementary relationship for k smallest R_{ij}, say $k = 1$. The intuition is that, if product i's demand has a positive correlation with the product j's price, it indicates that when the j's price goes up, customers will likely purchase more of product i, and therefore

we mark the pair as "substitute". We visualize the substitute and complement network in Figure 2 where blue and red edges stand for complementary and substitute relations respectively. The substitution relation may result from the nature of the products or from the recommendation system. The causality analysis is out of the scope of this paper due to the limitation of data on the recommendation side.

Figure 2 **Visualization of the Substitute and Complement Network: Each node stands for a product; blue and red edges represent complementary and substitute relations, respectively. The visualization plot is done by Fruchterman-Reingold algorithm (Fruchterman and Reingold 1991).**

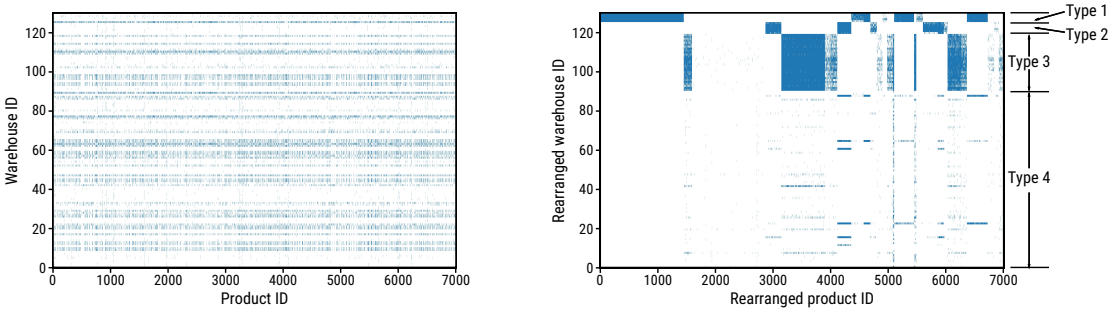


3.3. Multiple-product Multiple-warehouse Allocation

Figure 3a gives a visualization of the current allocation of 7,013 products at 130 Cainiao warehouses, in which the x-axis and y-axis represent the product ID and the warehouse ID respectively. A blue (white) point at the grid (i, j) in the figure indicates the i-th product is (not) allocated at the j-th warehouse. To better demonstrate the allocation pattern, we rearrange the product IDs based on their categories and the warehouse IDs with hierarchy clustering. We refer the implementation details of hierarchy clustering to Chapter 14 of the book (Friedman et al. 2001). After rearranging both the product and warehouse IDs, we plot the allocation in Figure 3b where the dispersed points cluster into blocks.

From Figure 3b, we observe that the current allocation of products is category-based. That is, products from the same category tend to be stored in the same warehouses. The warehouses are grouped into four different clusters (types), as marked in the figure, due to clear identified patterns. Specifically, the type 1 and type 2 cluster of warehouses store 792

Figure 3 **Visualization of Product Allocation at Warehouses. For the left panel, a blue (white) point at location (i, j) in the figure indicates the i-th product is (not) allocated at the j-th warehouse. The right panel rearranges the product and warehouse IDs.**



(a) Product allocation at warehouses

(b) After rearranging IDs

of the top 1000 best selling products, while there are only 11 warehouses in the first two clusters. We refer these two clusters of warehouses as *base warehouse*. In contrast, the type 3 and type 4, with a total number of 119 warehouses, show very different storage patterns. These warehouses are holding fewer number of product items, which are consisted of the less popular ones. Overall, the allocation is quite centralized and sparse: on average, each warehouses holds 7.3% of the products, while each product is stocked at 9 warehouses. This sparsity may help with the in-warehouse operational efficiency. However, the data shows that the logistics network covers in total 353 cities and 177427 facility locations[1]. The centralized allocation and the sparsity may result in a longer outbound shipping distance and slower delivery. The question naturally arises that whether a more distributed and denser product allocation improves the logistics efficiency.

To the best of our knowledge, the optimal allocation of multiple products at multiple warehouses, though important, is not fully studied in the literature. The decision maker needs to trade off between maintaining in-warehouse operational efficiency and shortening outbound shipping distance. The size of the problem (with thousands of products and hundreds of warehouses) makes it even more challenging. In Section 6, we discuss the corresponding optimization problem and analyze its optimal solution and objective value.

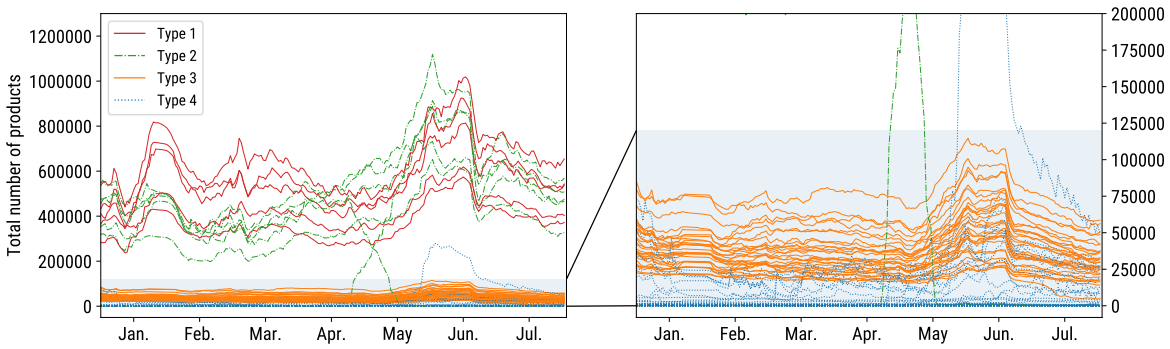
3.4. Inventory Level

We plot the dynamic daily inventory levels (the total number of items of product) throughout 7 months for all the 130 warehouses in Figure 4, where each curve represents a different

[1] Facilities refer to delivery station or logistic transfer center.

warehouse and different colors/line styles denote different warehouse types. The two evident peaks in the inventory level correspond to the lunar new year in January and the shopping festival on June 18th. The major observation is that the averaged daily inventory level is 50.47% of the peak inventory level, averaged through all warehouses.

Figure 4 **Daily Inventory Levels of the Warehouses: The left figure shows the daily inventory level for all the 130 warehouses while the right figure zooms into the lower part of the left figure. Each curve represents a warehouse and the warehouse type is indicated by the curve color.**



The rise of inventory level prior to shopping festivals is inevitable and not unique to Cainiao; for example, the same phenomenon is experienced by the Amazon's fulfillment centers[2]. Figure 4 presents a conservative replenishment rule and indicates a higher-than-necessary inventory level. This motivates the importance of an accurate demand prediction system which enables better inventory management decisions. The released inventory storage space, for example, can be used to facilitate the predictive shipping mechanism (described in Section 5) and a more distributed and denser product allocation plan (described in Section 6).

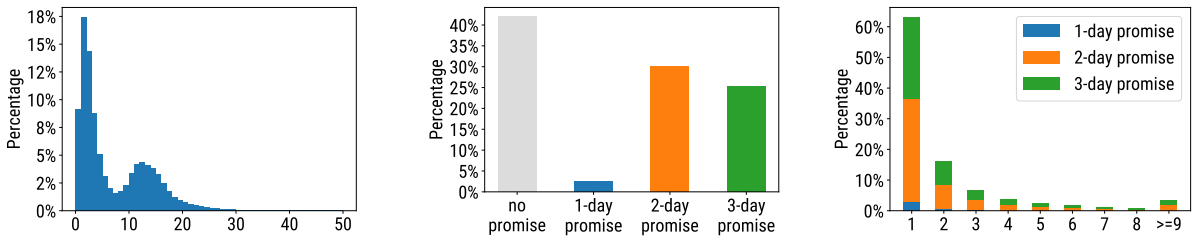
3.5. Delivery Performance

Figure 5 shows three aspects on the product delivery performance. In Figure 5a, we plot the histogram of the time elapsed from the ordering time of an item to its dispatch, i.e., $T_{\text{dispatch}} - T_{\text{ordered}}$. The dispatch time of an item is defined as the time it leaves the warehouse for shipping. The procedure between the ordering and the dispatch involves picking, packaging and labeling. From the data, we find that the average elapse time is 9.686 hours and more than 95.9% of order dispatches are finished within one day from the

[2] https://www.wsj.com/articles/amazon-prods-its-sellers-to-free-up-warehouse-space-1478251802

payment time, which is an indication of efficient in-warehouse operations. Figure 5b is a histogram of promised delivery time. For 42.1% of the orders there are no promise made and for 2.4% of the orders one-day delivery is promised. For orders that fail to meet the promise delivery time, the histogram of delay times are plotted in Figure 5c.

Figure 5 Promise Time and Delays.



(a) $T_{\text{dispatch}} - T_{\text{ordered}}$ (Hours) (b) Promised Delivery Time (c) Delay Time (Days)

In this paper, both the predictive shipping regime (in Section 5) and the product allocation optimization (in Section 6) are aimed to help reducing the delivery cost and improving the delivery speed. These improvements have the potential of enabling the company to make more and better time promises, resulting in an increased customer satisfaction in the end. Figure 5c shows that if one day's shipping time was saved, more than 60% percents of the delayed orders would have met the promise.

4. Demand Prediction

In this section, we introduce our demand prediction model for daily demands. The implementation details and the comparisons with other models will be presented in Section 7. For each product $i \in \mathcal{I} = \{1, ..., I\}$, we observe data $S_i = \{(X_1^i, D_1^i), ..., (X_T^i, D_T^i)\}$, where $D_t^i \in \mathbb{R}$ denotes the demand for product i on day t in the area of interest and $X_t^i \in \mathcal{X}$ denotes the features. Specifically, we assume the following model for the product demand,

$$D_t^i = f_i\left(X_t^i\right) + \epsilon_{it}, \tag{1}$$

where $f_i \in \mathcal{C}$ and $\mathcal{C}$ is some class of functions. The noise ϵ_{it} satisfies $\mathbb{E}\epsilon_{it} = 0$ and $\text{Var}(\epsilon_{it}) = \sigma_i^2$ and is independent of each other. We fit a function $\hat{f}_i$ by minimizing the *empirical risk* on dataset S_i,

$$\hat{f}_i = \arg\min_{f \in \mathcal{C}} \sum_{t=1}^{T} \left(D_t^i - f(X_t^i)\right)^2.$$

This formulation of empirical risk minimization has been widely adopted in statistics and machine learning, and could be potentially adjusted to meet specific decision goals. If we want to predict the demand for the next τ days, we can simply replace D_t^i with the demand over the future τ days. In the following subsections, we elaborate on three aspects for our demand prediction model.

4.1. Features

We consider three types of features for demand prediction:

(i) Past demand: We include the past t_0 days' demands $(D_{t-t_0}^i, ..., D_{t-1}^i)$. In our experiments, we choose $t_0 = 21$ to include the demands of past 3 weeks. Including a longer history of data will inevitably involve non-stationarities such as trend and seasonality. However, a shorter period may not fully capture the historical dependency of product demand. This choice of 3 weeks is made by balancing these two aspects.

(ii) Price: P_t^i denotes the price of product i on t-th day. We define the price change ratio as the ratio of t-th day's price over the average price of past t_0 days, $\frac{p_t^i}{\frac{1}{t_0}\sum_{k=1}^{t_0} p_{t-k}^i}$. Then we compute the Short-Term Price Change Ratios (ST-PCR) for 1-day and 3-day,

$$\text{ST-PCR}_i^t = \left(\frac{p_t^i}{p_{t-1}^i}, \frac{p_t^i}{\frac{1}{3}(p_{t-1}^i + p_{t-2}^i + p_{t-3}^i)} \right).$$

Theses ratios reflect short-term price changes, and are rather informative for demand prediction (see our findings in Section 3.2). Furthermore, we include the ST-PCRs of the top 1 substitutes/complementary products for product i, as defined in Section 3.2.

(iii) Page view: For each product, we compute the total page views from PC and from the app, and the total unique visitors from PC/app. When predicting the demand for the t-th day, we include as features these four quantities of the $t-1$-th day and also the average over the past 3 days.

We note that product interactions should be carefully analyzed and integrated to the prediction model. Furthermore, we introduce a regularization method in the next subsection.

4.2. Clustering-based Model Regularization

In (1), we allow for a different prediction model f_i for each product $i \in \mathcal{I}$. This is a common approach practitioners employ to treat each product separately. However, for the current data set, this approach suffers from two shortcomings: on one hand, it induces a

computational cost which scales linearly with the number of products; on the other hand, it results in a shortage of samples for each product. Therefore, we propose a clustering-based model regularization. The idea is to decompose the product set $\mathcal{I}$ (of size 7,013) into disjoint subsets $\mathcal{I}_1, ..., \mathcal{I}_K$ with $K \leq I = |\mathcal{I}|$ and to set the functions f_i's to be the same for all indices belonging to the same subset. Specifically, we do pooling for the individual datasets S_i's and form $S_{\mathcal{I}_k} = \cup_{i \in \mathcal{I}_k} S_i$. Then, we minimize the empirical risk on the dataset $S_{\mathcal{I}_k}$:

$$\hat{f}_{\mathcal{I}_k} = \arg\min_{f \in \mathcal{C}} \sum_{i \in \mathcal{I}_k} \sum_{t=1}^{T} \left(D_t^i - f(X_t^i) \right)^2,$$

and enforce $\hat{f}_i = \hat{f}_{\mathcal{I}_k}$ for all $i \in \mathcal{I}_k$. In such a way, all the products are grouped into K types and products of the same type share the same demand prediction model. Let I be the size of $\mathcal{I}$. When $K = I$, we indeed do no pooling but treat each product separately as before; while $K = 1$, we fit one single model for all the products. Technically, this approach reduces the number of different functions from I to K and augments the data samples for learning each function. The role can be interpreted as a reduction of "total model complexity."

The question is therefore on how to appropriately cluster the products and choose the number of clusters K. To solve the problem, we propose a new data-driven approach of representing the products and disclosing the relationship between different products, which we call as *product representation learning*.

4.3. Product Representation Learning

Representation learning generally refers to a set of machine learning techniques that automatically discover a feature vector for each individual in a complicated system. The feature vector can be interpreted as a representation of each individual and the procedure of finding such a representation is called representation learning. A recent influential work on is known as Word2Vec (Mikolov et al. 2013), which learns a feature vector for each English word. The feature vector can capture both semantic and syntactic meanings of the word and can be used for tasks of sentiment analysis, article classification, text summarization, machine translation and etc. The paper (Grover and Leskovec 2016) extends the model to learn representation for nodes in a network that represent certain product relations.

In our application, we seek to learn a fixed-dimensional feature vector $V_i \in \mathbb{R}^v$ for product i. Ideally, the closer the two vectors are, the more similar the demand pattern two products share. We first follow the idea of market basket analysis (Berry and Linoff 1997) and

construct a undirected weighted graph $\mathcal{G}$ where each node in the graph represents a product. When there is a customer ordering product i and j together, we forge an edge between node i and node j. Then we perform the node representation learning algorithm for $\mathcal{G}$ and obtain the feature vector V_i's. Basically, when the two products are often ordered together, the connection between the two products will be strong in the graph and hence the feature vectors V_i and V_j will be close to each other.

Then we run the K-means clustering algorithm based on the learned features V_i's and group all the products into K clusters. As mentioned before, we enforce the same prediction model within each cluster. The intuition is that, two products often ordered together are most likely to be in the same group and share similar demand fluctuation patterns. We will demonstrate in the experiment section that the clustering based on representation learning features effectively improve the prediction accuracy and outperforms the clustering based on product category largely.

5. Predictive Shipping (PreShip)

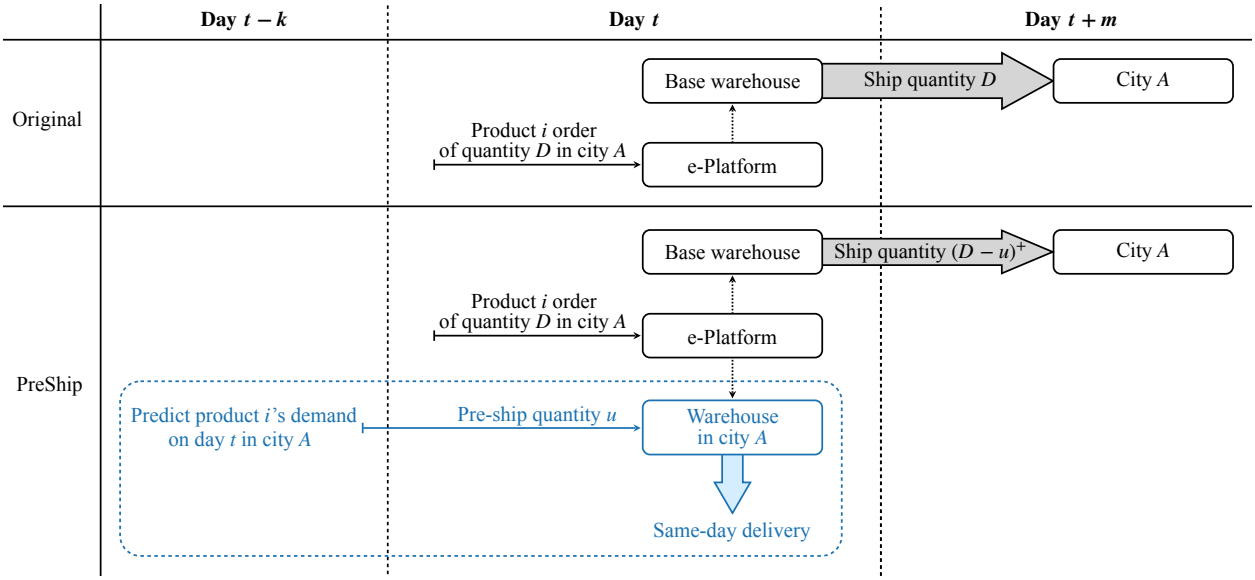
In this section, we first describe and build theory for the PreShip mechanism in a single-shot model. We then extend it to a sequential implementation of this mechanism.

5.1. Single-shot PreShip Optimization

In Figure 6, we give a full description of the new shipping mechanism. Typically, shipment happens after an order is received. Consider on Day t, online orders for item i of quantity D are submitted to the e-platform and the demand happens in some city A. Shipping will then be arranged from the base warehouse (which is most likely not in city A, due to the sparsity of product allocation as shown in the data). Suppose that the required delivery time is m days, incurring a per-item shipping cost c_1. In the PreShip regime, a prediction for the demand quantity D is made k days ahead for $k \geq 1$, and a quantity u is arranged to be shipped on Day $t-k$ to a warehouse in (or close to) city A. Then, on Day t, if it turns out the realized demand D on Day t is larger than u, the pre-shipped quantity u will be directed to the destination in city A, and an additional quantity of $D-u$ will be arranged to ship from the base warehouse to complete the full demand. If otherwise $D < u$, then D out the u pre-shipped quantity will be directed to the destination, while the extra $u-D$ quantity will be held temporarily, say at the warehouse or facility in city A, incurring an inventory cost c_0. We assume that the per-item shipping cost of the pre-shipped item c_2 is smaller than c_1. This holds in general because the pre-shipped quantity

can be shipped in a more economic way. For example, if the delivery requirement is $m = 1$, then without PreShip, the per-item cost c_1 is essentially the same-day delivery cost from the base warehouse to city A. However, with PreShip, the delivery time requirement for the pre-shipped quantity is $1 + k$ days and shall incur a lower cost c_2.

Figure 6 **Diagram of Predictive Shipping (PreShip)**



The optimal PreShip quantity u is determined by an optimization problem. We formulate and study the operational problem here through $\mathrm{OPT_k}$ and $\mathrm{OPT_u}$. For the simplicity of notation, we consider the case of $k = m = 1$. The two optimization formulations are given as

$$\mathrm{OPT_k} \triangleq \min_{u \geq 0} \mathbb{E}\left[c_2 u + c_1 (D-u)^+ + c_0 (u-D)^+\right]$$

$$\mathrm{OPT_u} \triangleq \min_{u \geq 0} \max_{D \in \Xi_{\mu,\sigma^2}} \mathbb{E}\left[c_2 u + c_1 (D-u)^+ + c_0 (u-D)^+\right]$$

where $(\cdot)^+$ is the positive part function and $\Xi_{\mu,\sigma^2} = \{D \in L_2 : \mathbb{E}D = \mu, \mathrm{Var}D = \sigma^2\}$, in which L_2 is the family of all square integrable random variables. The first optimization is the standard treatment of the newsvendor problem which assumes the distribution of the demand as known (denoted by the subscript "k"). The second formulation does not assume the knowledge of the full distribution (denoted by the subscript "u"). Instead, only the mean and variance of the random demand are assumed to be known. A min-max objective

is considered that minimizes the worst-case cost for all distributions of D with the given mean μ and variance σ^2. This second formulation is particularly suitable in our data-driven framework, for the following two reasons: (i) It is unrealistic to assume we know/can fit accurately the demand distributions for large number of items in our application context. Therefore the formulation OPT_{u} is more robust to distribution misspecification. (ii) The second formulation provides a natural upper bound for the first one. Most importantly, if we apply the machine learning models and treat the demand as a random variable (which is conditional on all the available information), the parameters μ and σ^2 can be easily estimated and the second formulation enables a better understanding of the role that machine learning models play in reducing the inventory cost via reducing the variance σ^2. With a little abuse of the notations, we refer to both the optimization problems and their optimal values with $\mathrm{OPT_k}$ and $\mathrm{OPT_u}$.

PROPOSITION 1. *For OPT_k, the optimal PreShip volume is $u^* = F_D^{-1}\left(\frac{c_1-c_2}{c_0+c_1}\right)$, in which $F_D(\cdot)$ is the cumulative distribution function of D. Let $\mu = \mathbb{E}D$. Then we have the following bounds for the optimal expected cost:*

$$c_2\mu \le OPT_k \le c_1\mu.$$

The result in Proposition 1 implies that the company can always benefit from arranging PreShip as long as $c_1 > c_2$. This is true regardless of how much the holding cost c_0 is. Also, the company tends to arrange more PreShip quantity if the difference between c_1 and c_2 becomes larger. The optimal (per item) expected cost lies between the one-day and two-day shipping cost. Of course, the more accurate the future demand predictions are, the better PreShip strategy we could employ. The ideal case will be the future demand can be known for certain, in the sense that $\sigma^2 = 0$. But on the other hand, as long as the distribution future demand is known, we can always achieve a reduction in the per-item cost. Next proposition solves the optimal PreShip problem in scenarios when the full distribution of future demand is not known, but only the mean and variance are known.

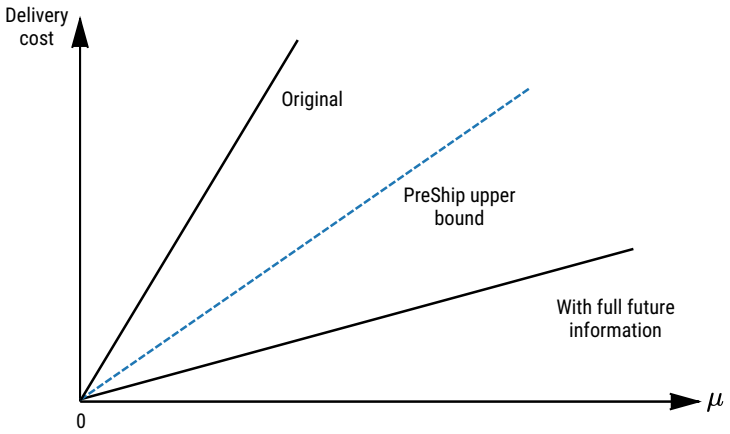
PROPOSITION 2. *Let $h(a) = \frac{1-2a}{2\sqrt{a(1-a)}}$. The optimal two-day shipping volume for OPT_u is given by*

$$u^* = \begin{cases} 0, & \text{if } \frac{c_0+c_2}{c_0+c_1}\left(1+\frac{\sigma^2}{\mu^2}\right) \geq 1, \\ \mu + \sigma h\left(\frac{c_0+c_2}{c_0+c_1}\right), & \text{if } \frac{c_0+c_2}{c_0+c_1}\left(1+\frac{\sigma^2}{\mu^2}\right) < 1. \end{cases} \tag{2}$$

And the optimal expected cost has the following bound:

$$OPT_u \leq \min\left\{c_1\mu, c_2\mu + \sigma\sqrt{(c_0+c_2)(c_1-c_2)}\right\}.$$

Figure 7 **A Schematic Diagram for Delivery Cost: Both the original shipping regime and the shipping regime with full future information have an expected cost that scales linearly with the expected demand μ. The upper bound for PreShip cost lies between these two.**



The optimal volume specified in (2) is a threshold policy, and the threshold value depends on all model parameters. Unlike the result in Proposition 1, the condition $c_1 > c_2$ may not be enough to trigger a PreShip strategy, unless the coefficient of variation σ/μ is relatively small. For the optimal expected cost, it is made clear that the smaller σ is, the more cost reduction can be achieved. An ideal case is, again, $\sigma = 0$, in which case all the demand can be handled ahead of order placements. Consequently, in this case, a one-day shipping cost can be fully reduced to a two-day shipping cost with PreShip. Yet a more practical setting is that there is always some uncertainty σ in future demand. In this case, the machine learning models plays an important role in reducing the uncertainty, by exploiting more information in the transactional data. An interpretation of the optimal cost bound is that we can reduce the shipping cost of roughly $1-\sigma/\mu$ proportion of the full product demand. As demonstrated in our prediction results, we are able to obtain a demand prediction with σ/μ around 0.3, which means around 70% of the products can receive a shipping cost

reduction. In addition to reducing the shipping cost, this strategy may also alleviate the serious delaying situation for certain products.

5.2. Sequential PreShip and Implementation

The $\mathrm{OPT_k}$ and $\mathrm{OPT_u}$ concern the optimal pre-shipped quantity for a single-shot shipping problem. In this subsection, we provide a multi-period PreShip implementation for the sequential setting in practice. Specifically, we repeatedly conduct demand prediction and dynamically solve the PreShip optimization every period (every day, for example). For instance, on day t, the demand prediction system predicts the $t+k$-th day's demand D_{t+k} with its (conditional) expectation μ_{t+k}. The variance σ^2 can be estimated by historical prediction errors in a relatively close time window. With the cost structure (c_0, c_1, c_2) specified in real operations, we can solve the OPT_u and obtain the PreShip quantity u_{t+k} for the $(t+k)$-th day.

Figure 8 **A Schematic Diagram for Sequential PreShip Regime**

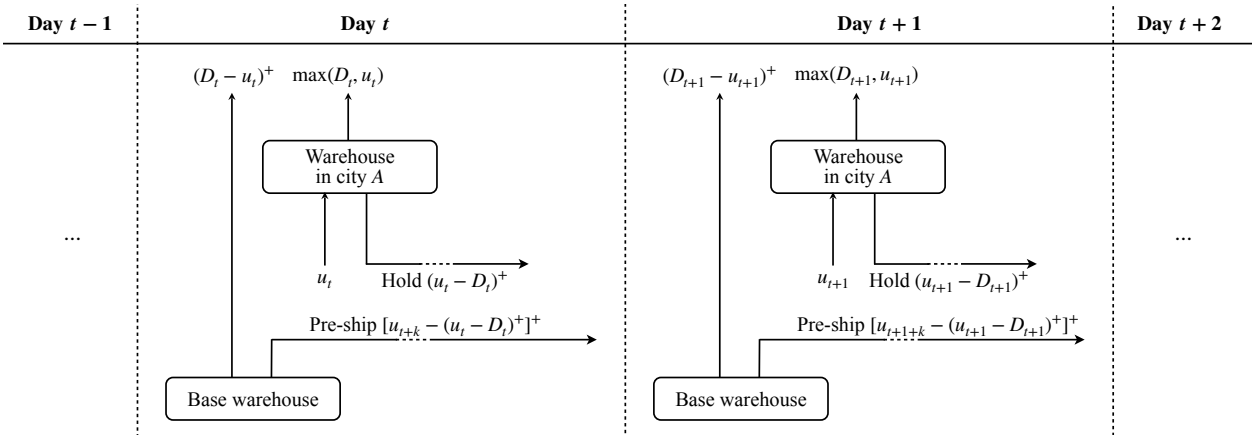


Figure 8 provides a schematic diagram for the implementation. On Day t, u_t quantity of the product that were shipped in Day $t-k$ flow in the warehouse in city A and these items can be delivered immediately to meet the demand D_t in city A. If there is a shortage of product in the warehouse on this day, $(D_t - u_t)^+$ additional items will be shipped from base warehouse directly to the destination. On the opposite, if there are left-overs, the warehouse will hold the quantity $(u_t - D_t)^+$ and then subtract this amount from the PreShip quantity for Day $t+k$ which starts the shipping on Day t.

To fulfill the PreShip, a flexible region can be designed for each warehouse and the space should be reserved only for the PreShip use. The products are held overnight in this region only when the pre-shipped quantity u_t exceeds the demand D_t. A more accurate demand prediction system will therefore keep the flexible region less crowded. In the product dimension, one can select those products whose demands are more predicable for PreShip, while using the conventional regime for the products with less predictable demand. We note that the PreShip regime is not restricted to the warehouse usage. For e-commerce companies like Alibaba, they usually have many facilities, trucks, and pick-up locations, all of which could be utilized as temporary storage for the pre-shipped products.

6. Product Allocation Optimization

Based on the data analysis in Section 3.3, we raised the question of multiple-product multiple-warehouse allocation and visualized Cainiao's current product allocation pattern. Define the *allocation density* as

$$den(A) \triangleq \frac{\sum_{i=1}^{I}\sum_{w=1}^{W} A_{iw}}{IW},$$

where the binary allocation matrix $A = (A_{iw})_{I\times W}$ has the element $A_{iw} = 1$ when the i-th product is allocated at warehouse w and $A_{iw} = 0$ otherwise. We find in the data that the allocation matrix is sparse, with the allocation density around 0.07. A natural operational question to ask is, shall the company increase the allocation density? In this section, we discuss the marginal benefit of increasing the allocation density, measured by the averaged outbound shipping cost reduction. The intuition is that a denser product allocation provides additional shipping options and therefore reduces the outbound shipping cost. We first solve an optimization problem that computes the optimal outbound shipping cost given the allocation density and allocation matrix. Then, we study how the optimal shipping cost reduces as the allocation density increases. Both theoretical analysis and empirical implementation with real data are provided.

6.1. Optimal Outbound Shipping Cost

To compute the optimal averaged outbound shipping cost given the allocation matrix and detailed cost structure, we consider a single-stage shipping problem and formulate a linear program $\mathrm{OPT_s}(A, C, D)$ (where "s" stands for shipping) that will be defined shortly. We first offer an efficient algorithm to solve the optimization problem and then study how

the optimal cost changes with respect to the different choices of allocation matrix A of different density $den(A)$. With a little abuse of notation, we refer to both the optimization problem and its optimal value as $\mathrm{OPT_s}(A, C, D)$.

$$\begin{aligned} \mathrm{OPT_s}(A,C,D) \;\triangleq\; \min_{Z} \;\; & \frac{1}{IL}\sum_{i=1}^{I}\sum_{w=1}^{W}\sum_{l=1}^{L} C_{iwl}Z_{iwl} \\ \text{subject to} \;\; & \sum_{w=1}^{W} Z_{iwl} = D_{il} \text{ for all } i,l \\ & (1-A_{iw})Z_{iwl} = 0 \text{ for all } i,w \\ & Z \geq 0. \end{aligned}$$

Here D denotes the demand matrix where D_{il} represents the demand for product i at location l and the three-dimensional array C specifies the shipping cost structure where C_{iwl} is the unit cost of shipping product i from warehouse w to location l. The demand matrix D is set to reflect the long-run average demand, but it can also be interpreted alternatively as the dynamic one-day demand. The decision variable Z is a three-dimensional array where Z_{iwl} determines the quantity of product i shipped from warehouse w to location l. We temporarily allow Z_{iwl}'s to take non-integer values in the formulation but will show that there exists an optimal solution composed of only integer values. The objective is to minimize the total shipping cost, that is, the element-wise product of C and Z. In terms of the constraints, the first constraint requires the demand to be fully satisfied and the second one specifies that the products can only be shipped out from those warehouses in which the corresponding products are available. We also make the following assumption:

Assumption 1. *(Sufficient Inventory) If $A_{iw} = 1$, then warehouse w has sufficient inventory of item i to satisfy the demand in this single stage.*

The assumption is reasonable as indicated by the data, in which the inventory level is much higher than the daily demand and there are almost no inventory shortage observed.

We use A, C, D as arguments of $\mathrm{OPT_s}(A, C, D)$ to emphasize the dependence of optimal cost on the input parameters. With the number of decision variables being IWL, the linear program can be prohibitively large to solve, particularly for such a large logistic network in presence. To resolve the computational issue, we develop an efficient algorithm by exploiting the underlying structure of $\mathrm{OPT_s}(A, C, D)$. Essentially, the algorithm relies

Algorithm 1: Algorithm for solving $\text{OPT}_\text{s}(A, C, D)$

Input: Demand matrix D, cost array C, allocation matrix A

Output: Optimal solution Z^*

Initialize $Z = \mathbf{0}$

for $i = 1, ..., I$ **do**

  for $l = 1, ..., L$ **do**

    Let $\mathcal{W} = \{w : A_{iw} = 1\}$

    Compute $w_0 = \arg\min_{w \in \mathcal{W}} C_{iwl}$

    Let $Z_{iw_0l} = D_{il}$

  end

end

return $Z^* = Z$

on the following two observations. First, both the objective function and the constraints are separable with respect to the products, such that we only need to solve I linear programs individually, one for each product. Second, there exists an optimal shipping strategy, such that, for each product, the strategy designates one and only one warehouse to conduct the shipping of this product to a specific destination location. We show that there exists an optimal solution Z^*, for every product i and location l, there is one and only one w, such that $Z^*_{iwl} > 0$ (so that $Z^*_{iwl} = D_{il}$).

PROPOSITION 3. *Algorithm 1 solves the problem and its output Z^* is the optimal solution. Its complexity is $O(IWL)$. Furthermore, for two allocation matrices A and A', if we have $A \geq A'$, then $OPT_s(A, C, D) \leq OPT_s(A', C, D)$.*

6.2. Analyzing $\text{OPT}_\text{s}(A, C, D)$ with Random Matrices Inputs

This subsection develops an analytical approach to study how the optimal shipping cost $\text{OPT}_\text{s}(A, C, D)$ relies on the allocation density $den(A)$. We assume that A, C, D are generated from random matrices. (Later on, we will return to this problem with a real-world setting in which these matrices are calibrated by Cainiao's dataset.) Here, imposing the random matrix assumptions allows us to derive closed-form analytical results. These results, as stated in Proposition 4, provide key insights for the relationship between $den(A)$ and the optimal shipping cost. Specifically, the proposition assumes that the entries in the cost array C and the demand matrix D are generated randomly, and for each product,

we randomly allocate it in $\lfloor dW \rfloor$ warehouses (it results in a allocation density around d). In this setting, the shipping cost $\mathrm{OPT_s}(A, C, D)$ can be viewed as a random variable O_d parametrized with the density level $den(A) = d$, and the cost O_d can be approximated by a deterministic function $g_W(d)$ which depends only on the number of warehouses W and the density level d. Additionally, when the cost and demand are generated from uniform/truncated exponential distribution, the function $g_W(d)$ is approximately inversely proportional to the density level d for a large W.

PROPOSITION 4. *Given a density ratio $d \in (0,1)$, we assume the arguments of $OPT_s(A, C, D)$ are generated as follows:*

(i) The cost C_{iwl}'s are i.i.d. random variables with $C_{iwl} \sim X_C$, for $i = 1, ..., I$, $w = 1, ..., W$, and $l = 1, ..., L$. $\mathbb{E}X_C = \mu_C$ and $X_C \in [a_C, b_C]$ almost surely.

(ii) The demand D_{wl}'s are i.i.d. random variables with $D_{wl} \sim X_D$, for $w = 1, ..., W$, and $l = 1, ..., L$. $\mathbb{E}X_D = \mu_D$ and $X_D \in [a_D, b_D]$ almost surely.

(iii) For each i, there are $\lfloor dW \rfloor$ entries of A_{iw}'s being 1 and they are selected randomly. Here $\lfloor \cdot \rfloor$ denotes the floor function which returns the greatest integer less than or equal to the input.

Moreover, we introduce a class of functions $g_W : (0,1) \to \mathbb{R}$ to denote the expectations of extreme value

$$g_W(d) = \mathbb{E} \min \left(X_1, ..., X_{\lfloor dW \rfloor} \right),$$

where X_i's are i.i.d. random variables and $X_i \sim X_C$ for $i = 1, ..., \lfloor dW \rfloor$. Under the above assumptions, we can represent the optimal cost $OPT_s(A, C, D)$ with a family of random variables indexed by d, $\{O_d, d \in (0,1)\}$. Then, for any $d_1, d_2 \in (0,1)$ and any $\epsilon \in (0,1)$, there exists a constant γ depending on (a_C, a_D, b_C, b_D), such that

$$\mathbb{P}\left(\frac{O_{d_1}}{O_{d_2}} \notin \left[\frac{g_W(d_1)}{g_W(d_2)} - \epsilon, \frac{g_W(d_1)}{g_W(d_2)} + \epsilon \right] \right) \leq 4 \exp\left(- \frac{2\gamma I L \epsilon^2}{(b_C b_D - a_C a_D)^2} \right). \tag{3}$$

Furthermore, when X_C follows a uniform distribution or a truncated exponential distribution, for any $d_1, d_2 \in (0,1)$,

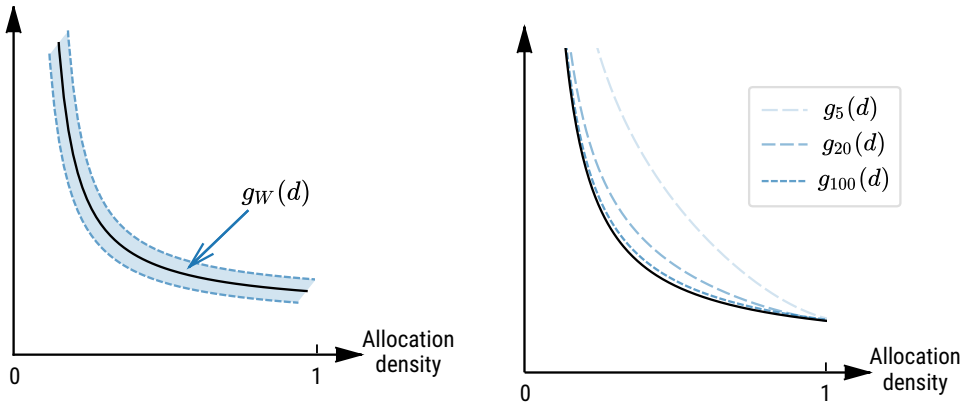
$$\frac{g_W(d_1) - a_C}{g_W(d_2) - a_C} \to \frac{1/d_1}{1/d_2}, \tag{4}$$

as $W \to \infty$.

REMARK 1. The proposition still holds when the boundedness constraints on X_C and X_D are replaced by sub-Gaussianity. Also, the assumption of i.i.d. can be relaxed to independent but not identical. Moreover, similar results can be proved under some weak dependence conditions.

With the i.i.d. assumptions of the entries in C and D, the optimal cost, though random, is concentrated around its expectation. That is, the optimal cost, represented by a random variable O_d parameterized by the density d, can be well approximated by a deterministic function $g_W(d)$. The function $g_W(d)$ describes the expectation of the extreme value, and the form of extreme value comes from the structure of $\mathrm{OPT}_{\mathrm{s}}(A, C, D)$ and Algorithm 1. The extreme value theory provides us the further result 4 on the asymptotic behaviour of $g_W(d)$ when the cost follows uniform distribution or truncated exponential distribution.

Figure 9 **Schematic Diagrams for O_d and $g_W(d)$: O_d stays around $g_W(d)$ with high probability and the curves $g_W(d)$ can be approximated by an inversely proportional function when W is large.**



(a) O_d (b) $g_W(d)$

Figure 9 illustrates the geometry of Proposition 4. The left panel depicts the function $g_W(d)$ and demonstrates that the value $\mathrm{OPT}_{\mathrm{s}}(A, C, D)$ lies in a narrow band around $g_W(d)$ with high probability. The right picture illustrates the last result in the proposition, that is, for large W, $g_W(d)$ can be approximated by an inversely proportional function.

Given that the delivery cost can be approximated by an inverse proportional function, we learn that on one hand, the delivery cost decreases very fast when the allocation density d is small; on the other hand, for a large d, the cost reduction of increasing allocation density is marginal. Consider the current allocation density of Cainiao being 0.07 (as in

Figure 3b), which is in a steep cost-cutting region, the company will enjoy a significant cost reduction with only a small increase of allocation density. We will illustrate the idea with numerical experiments in the next section.

7. Experiments

This section provides the experiments to demonstrate our results with both real data and synthetic data.

7.1. Demand Prediction

We first present the experiment results for our demand prediction model introduced in Section 4. The demand data consists of 731 products, covering from January to July in 2017. Here, we only consider these 731 products that have no missing demand entries. We split the data by time, and use the data from January to April for training, May for validation, and June and July for testing. The validation data are used for tuning the hyper-parameters of the machine learning models. All the prediction accuracies are measured by the performance on the test set. We use relative mean absolute error (Rel. MAE) as the performance metric. Considering that different product can have drastically different demand, we adopt relative error here. Moreover, the relative mean absolute error can be viewed as an estimate of the coefficient of variance $\frac{\sigma}{\mu}$.

The first experiment concerns selecting the right model for the task of predicting demand. In this experiment, we pool together all the data into one cluster, i.e., setting $K = 1$ as in Section 4.2. In total, we include 32 features described in Section 4.1. We consider six different function classes (models) $\mathcal{C}$. The baseline model simply uses the historical mean to predict the future demand. Three linear models are considered: LR (ordinary linear regression), RR (ridge regression), and Lasso (Tibshirani 1996). Two nonlinear models are RF (random forest) (Breiman 2001) and NN (neural networks). The architecture of the neural network is 3-layer and fully-connected; the number of nodes on each layer is (32,16,4,1).

The prediction accuracies are reported in Table 1. The best model achieves a relative MAE of 0.35 and it cuts off 70% of error compared with the baseline model. The inclusion of the price and views features does not improve the performance of linear models but significantly reduces the prediction error for nonlinear models. This means that the product's price and page views most likely have a nonlinear effect on its demand. Also, the additive

Features	Metrics	Baseline	Linear			Nonlinear	
		Mean	LR	RR	Lasso	RF	NN
PD	Rel. MAE	1.20	0.47	0.42	0.43	0.40	0.54
PD+Price+Views	Rel. MAE	1.20	0.97	0.83	0.54	**0.35**	0.42

Table 1 **Prediction Accuracies when the Number of Functions $K = 1$: PD stands for past demands; Price and Views refer to the features of prices and page views, respectively.**

linear models fail to capture the interaction effect. More results on feature interpretation can be found in the Appendices.

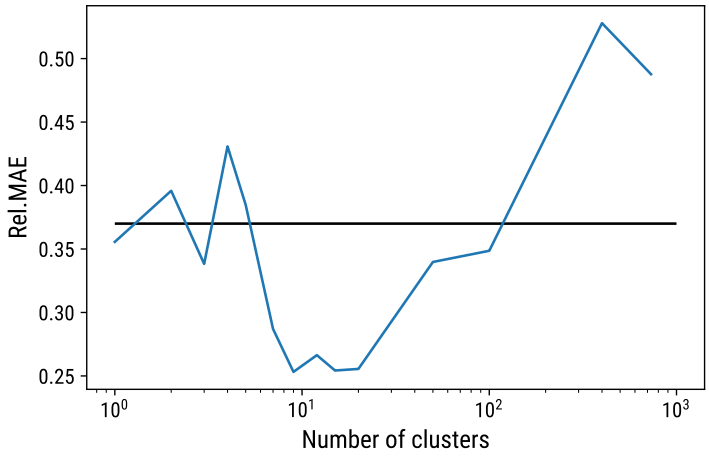


Figure 10 **Clustering Relative MAE: The black horizontal line denotes the Rel. MAE if we do product clustering based on the category.**

In the above experiment, we enforce $K = 1$ and find out the best prediction model is the random forest. The next experiment varies the value of K and plot the results in Figure 10. For each K, we run a K-means clustering for the products based on the feature vectors from representation learning. Then we follow the Section 4.2 and group the products into K clusters. We report and plot Rel. MAEs in Figure 10. We can see that when K takes value of 10 to 20, there is a significant error reduction. Values in this range provide a good trade-off between model complexity (measured by the total number of different f_i's) and the number of samples for training each f_i. The best Rel. MAE achieved is around 0.25; the

PreShip regime can be implemented efficiently with such relative accuracy. Alternatively, if we cluster the products based on category, the Rel. MAE is around 0.37. This shows the efficacy and the meaningfulness of the representation learning features.

Based on the above results, we settle with the random forest model and the number of clusters $K = 15$. Then we proceed with the task of multiple-day demand prediction - predicting the average demand for the next τ days. The looking-forward period is 3-day, 7-day and 14-day. The multiple-day prediction is useful for the task of inventory management with periodic replenishment and with lead time. The prediction accuracies are summarized in Table 2. We can see that the errors will not increase greatly with longer prediction horizon. The time stability of demand prediction enables the company to better optimize the inventory replenishment decisions.

Features	Metrics	1-Day	3-Day	7-Day	14-Day
PD	Rel. MAE	0.334	0.336	0.333	0.334
PD+Price+Views	Rel. MAE	0.267	0.269	0.267	0.270

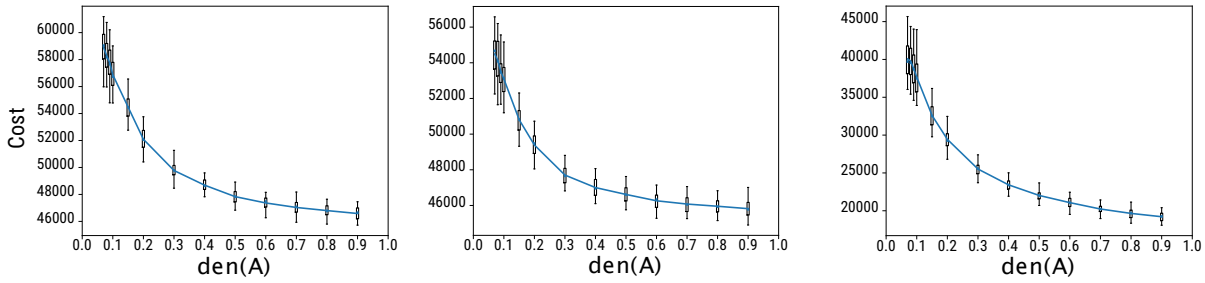
Table 2 Prediction Accuracies for Multiple-period Demand: The model used is random forest and the number of different functions $K = 15$.

7.2. Production Allocation

In Section 6, we study the question of multi-product multi-warehouse allocation and provide analytical results. The results suggest that under certain assumptions, the shipping cost is inversely proportional to the allocation density $den(A)$ when there are a large number of products, locations and warehouses. In this section, we first conduct simulation experiments to verify the results under random matrices assumptions, and then demonstrate with empirical experiments that the results still hold when we calibrate the model to match real data.

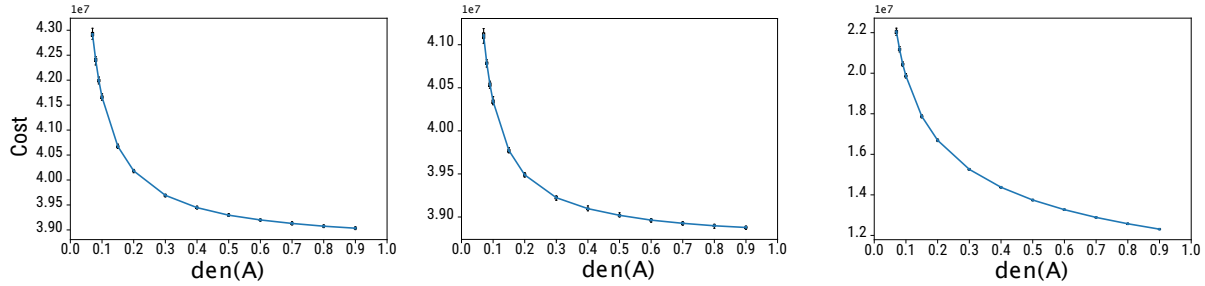
7.2.1. Simulation experiment for $\mathbf{OPT_s}(A, C, D)$. We first run simulation experiments and provide some intuition for Proposition 4. Two groups of experiments are run with different problem size and each group has three experiments with different random variable distributions. The experiment details and results can be found in Figure 11 and 12.

Figure 11 **Simulation Experiment 1 for $\mathbf{OPT_s}(A,C,D)$: The problem size $(I,W,L)=(30,30,10)$. The matrix A is generated as Proposition 4 with respect to different density d. Figure (a): $C_{iwl} \overset{i.i.d.}{\sim}$ Uniform$[10,20]$, $D_{wl} \overset{i.i.d.}{\sim}$ Uniform$[10,20]$. Figure (b): $C_{iwl} \overset{i.i.d.}{\sim} 10+5\cdot$ TruncExp$[0,2]$, $D_{wl} \overset{i.i.d.}{\sim} 10+5\cdot$ TruncExp$[0,2]$. Figure (c): $C_{iwl} \overset{i.i.d.}{\sim}$ Log-normal$(0,1)$, $D_{wl} \overset{i.i.d.}{\sim}$ Log-normal$(0,1)$. Here TruncExp refers to the truncated exponential distribution. The 50% confidence intervals, and 95% confidence interval are indicated by the boxes and whiskers respectively.**



(a) (b) (c)

Figure 12 **Simulation Experiment 2: The problem size $(I,W,L)=(731,130,353)$. The other settings are the same as Figure 11.**



(a) (b) (c)

We set the second group's problem size equal to the size of Cainiao's practical case where $(I,W,L)=(731,130,353)$ and choose a smaller size for the first group where $(I,W,L)=(30,30,10)$. For each density d, we run 100 trials and plot the confidence intervals. Several observations are: (i) Regardless of the choice of random variables, the curve has a shape of inverse proportional function and the cost decreases fast around small density level. (ii) As the problem size goes up, the confidence intervals shrink; this means the cost will approximately be a deterministic function of d.

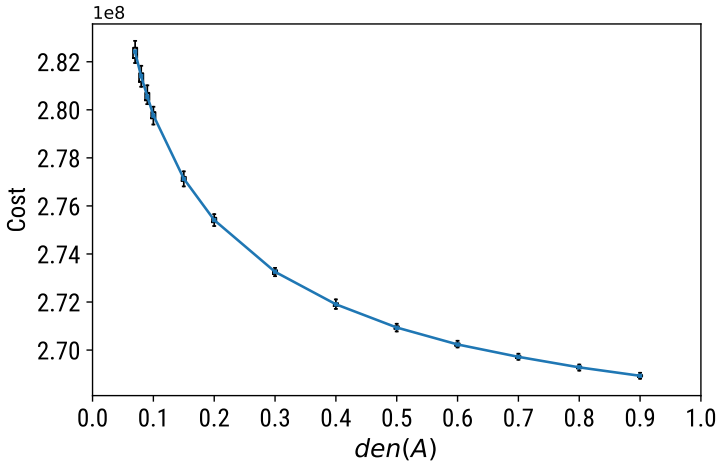
7.2.2. Empirical experiments with real data. We also run the shipping optimization problem $\mathrm{OPT_s}(A,C,D)$ based on Cainiao's data. There are in total of 731 products, 130

warehouses and 353 cities (locations). The demand matrix D is extracted from data, while the demand C is computed by

$$C_{iwl} = (10 + 10\text{Dist}(w, l)) \cdot (1 + 1/q_i)$$

where $\text{Dist}(w, l)$ is the distance between the warehouse w and the location l. We sample $q_i \sim 1 + \text{Exp}(1)$ and use it to account for the cost difference between different products. The original product allocation matrix A for Cainiao has a density of 0.07. Here the 731 products' allocation matrix is a sub-matrix of 7,013 products' allocation matrix in Figure 3b. In this experiment, we randomly replace 0's with 1's in matrix A to achieve a target density level d.

Figure 13 **The Relationship between Shipping Cost and $den(A)$ on Cainiao's Data: The 50% confidence intervals, and 95% confidence intervals are indicated by the boxes and whiskers respectively.**



We plot the results in Figure 13. For each point, we randomly generate 100 allocation matrices and solve the $\text{OPT}_s(A, C, D)$. The 95% confidence intervals are indicated by the whiskers. The plot verifies that the results in Proposition 4 still hold for Cainiao's practical case: the shipping cost decreases fast when the density is small. Another important observation is that the randomly generated allocation matrices result in almost the same shipping cost. We conclude that the allocation density plays a more important role in determining the outbound shipping cost than the detailed allocation scheme.

References

Ahire SL, Malhotra MK, Jensen JB (2015) Carton-mix optimization for walmart. com distribution centers. *Interfaces* 45(4):341–357.

Aral S, Walker D (2014) Tie strength, embeddedness, and social influence: A large-scale networked experiment. *Management Science* 60(6):1352–1370.

Ban GY, Rudin C (2018) The big data newsvendor: Practical insights from machine learning. *Operations Research* .

Berry MJ, Linoff G (1997) *Data mining techniques: for marketing, sales, and customer support* (John Wiley & Sons, Inc.).

Bertsimas D, Kallus N, Hussain A (2016) Inventory management in the era of big data. *Production and Operations Management* 25(12):2006–2009.

Boone T, Ganeshan R, Hicks RL, Sanders NR (2018) Can google trends improve your sales forecast? *Production and Operations Management* 27(10):1770–1774.

Boucheron S, Lugosi G, Massart P (2013) *Concentration inequalities: A nonasymptotic theory of independence* (Oxford university press).

Breiman L (2001) Random forests. *Machine learning* 45(1):5–32.

Cachon GP, Fisher M (2000) Supply chain inventory management and the value of shared information. *Management science* 46(8):1032–1048.

Carbonneau R, Laframboise K, Vahidov R (2008) Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research* 184(3):1140–1154.

Chen BJ, Chang MW, et al. (2004) Load forecasting using support vector machines: A study on eunite competition 2001. *IEEE transactions on power systems* 19(4):1821–1830.

Choi H, Varian H (2012) Predicting the present with google trends. *Economic Record* 88:2–9.

Choi TM, Wallace SW, Wang Y (2018) Big data analytics in operations management. *Production and Operations Management* 27(10):1868–1883.

Chong AYL, Chng E, Liu MJ, Li B (2017) Predicting consumer product demands via big data: the roles of online promotional marketing and online reviews. *International Journal of Production Research* 55(17):5142–5156.

Cui R, Allon G, Bassamboo A, Van Mieghem JA (2015) Information sharing in supply chains: An empirical and theoretical valuation. *Management Science* 61(11):2803–2824.

Cui R, Zhang DJ, Bassamboo A (2018) Learning from inventory availability information: Evidence from field experiments on amazon. *Management Science* .

De Haan L, Ferreira A (2007) *Extreme value theory: an introduction* (Springer Science & Business Media).

Ferreira KJ, Lee BHA, Simchi-Levi D (2015) Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* 18(1):69–88.

Friedman J, Hastie T, Tibshirani R (2001) *The elements of statistical learning*, volume 1 (Springer series in statistics New York).

Fruchterman TM, Reingold EM (1991) Graph drawing by force-directed placement. *Software: Practice and experience* 21(11):1129–1164.

Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864 (ACM).

Hortaçsu A, Nielsen ER (2010) Commentarydo bids equal values on ebay? *Marketing Science* 29(6):994–997.

Huang T, Van Mieghem JA (2014) Clickstream data and inventory management: Model and empirical analysis. *Production and Operations Management* 23(3):333–347.

Hwang M, Park S (2015) The impact of walmart supercenter conversion on consumer shopping behavior. *Management Science* 62(3):817–828.

Lee HL, So KC, Tang CS (2000) The value of information sharing in a two-level supply chain. *Management science* 46(5):626–643.

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Scarf H (1958) A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production* .

Slimani I, El Farissi I, Achchab S (2015) Artificial neural networks for demand forecasting: application using moroccan supermarket data. *Intelligent Systems Design and Applications (ISDA), 2015 15th International Conference on*, 266–271 (IEEE).

Sodhi MS, Tang CS (2010) Conclusion: a long view of research and practice in operations research and management science. *A long view of research and practice in operations research and management science*, 275–297 (Springer).

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Wang Z, Mersereau AJ (2017) Bayesian inventory management with potential change-points in demand. *Production and Operations Management* 26(2):341–359.

Appendices

Proof of Proposition 1

Without loss of generality, we assume the demand D has no point mass, i.e., it has a density function. If this is not true, we can introduce the notion of subgradient and the proof will still hold. Let

$$f(u) = \mathbb{E}\left[c_2 u + c_1(D-u)^+ + c_0(u-D)^+\right].$$

It is easy to see the convexity of $f(u)$. From the optimality condition, we know that the optimal solution u^* satisfies

$$\begin{aligned} 0 &= \nabla f(u^*) \\ &= c_2 + c_1 \mathbb{P}(D > u^*) + c_0 \mathbb{P}(D < u^*). \end{aligned}$$

Therefore, $u^* = F_D^{-1}\left(\frac{c_1 - c_2}{c_0 + c_1}\right)$.

Proof of Proposition 2

The difference between the original and the new cost is

$$\begin{aligned} & c_1\mu - \mathrm{OPT}_u \\ =& c_1 \mathbb{E}D - \mathbb{E}\left[c_2 u + c_1(D-u)^+ + c_0(u-D)^+\right] \\ =& (c_1 - c_2)u - \mathbb{E}\left[c_1(u-D)^+ + c_0(u-D)^+\right] \\ =& (c_1 - c_2)u - (c_0 + c_1)\mathbb{E}\left[(u-D)^+\right] \\ =& (c_0 + c_1)\mathbb{E}\left[\min(D,u)\right] - (c_0 + c_2)u \end{aligned}$$

Then the rest of the proof follows the derivation of the results in paper (Scarf 1958).

Proof of Proposition 3

We denote Z^* as the output of Algorithm 1. If there exists another feasible solution Z' such that

$$\sum_{i=1}^{I}\sum_{w=1}^{W}\sum_{l=1}^{L} C_{iwl} Z'_{iwl} < \sum_{i=1}^{I}\sum_{w=1}^{W}\sum_{l=1}^{L} C_{iwl} Z^*_{iwl},$$

then there must exist a product i and a location l such that

$$\sum_{w=1}^{W} C_{iwl} Z'_{iwl} < \sum_{w=1}^{W} C_{iwl} Z^*_{iwl}.$$

Let $\mathcal{W} = \{w : A_{iw} = 1\}$ and $w_0 = \arg\min_{w\in\mathcal{W}} C_{iwl}$. Then, considering the constraint, the above inequality is equivalent to

$$\sum_{w\in\mathcal{W}} C_{iwl} Z'_{iwl} < \sum_{w\in\mathcal{W}} C_{iwl} Z^*_{iwl}.$$

On the other hand, we have

$$\begin{aligned}\sum_{w\in\mathcal{W}} C_{iwl} Z'_{iwl} &> \sum_{w\in\mathcal{W}} C_{iw_0l} Z'_{iwl} \\ &= C_{iw_0l} \sum_{w\in\mathcal{W}} Z'_{iwl} \\ &\geq C_{iw_0l} D_{wl} \\ &= C_{iw_0l} Z^*_{iwl} = \sum_{w=1}^{W} C_{iwl} Z^*_{iwl},\end{aligned}$$

where the first inequality is from the algorithm and the third one is from the constraint. Here we have a contradiction. Therefore, we prove the optimality of Z^*.

Moreover, from the algorithm, it is obvious that its complexity is $O(IWL)$, it is easy to show that if $A \geq A'$, then $\mathrm{OPT_s}(A, C, D) \leq \mathrm{OPT_s}(A', C, D)$.

Proof of Proposition 4

LEMMA 1 **(Hoeffding's inequality)**. *Assume $X_1, ..., X_n$ are independent random variables and $X_i \in [a_i, b_i]$. Let $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$, then we have*

$$\mathbb{P}(\bar{X}_n \geq \mathbb{E}\bar{X}_n + \epsilon) \leq \exp\left(\frac{-2n^2\epsilon}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

For the proof, we refer to the Chapter 2.3 of the book (Boucheron et al. 2013). With the lemma, we proceed to prove Proposition 4.

We divide the proof into two parts: the probability bound (3) and the convergence result (4). First, we show (3). Consider A is generated randomly according to the assumption with $den(A) = d$. From the algorithm 3, we know that the cost of shipping the i-th product to location l is

$$C'_{il} = \min_{w\in\mathcal{W}_i} C_{iwl}$$

where $\mathcal{W}_i = \{w : A_{iw} = 1\}$. Then the optimal average shipping cost can be written as

$$OPT_s(A, C, D) = \frac{1}{IL}\sum_{i=1}^{I}\sum_{l=1}^{L} C'_{il} D_{il}.$$

By the definition of $g_W(d)$, we know that $\mathbb{E}C'_{il} = g_W(d)$. Consider the independence between C'_{il} and D_{il}, we know that

$$\mathbb{E}C'_{il}D_{il} = g_W(d)\mu_D.$$

Also, $C'_{il}D_{il} \in [a_C a_D, b_C b_D]$. Then, from Hoeffding's inequality, we know that for any $d_1, d_2 \in [0,1]$

$$\mathbb{P}\left(|O_{d_1} - g_W(d_1)\mu_D| \geq \frac{\epsilon}{C_0}\right) \leq 2\exp\left(\frac{-2IL\epsilon}{C_0(b_C b_D - a_C a_D)^2}\right)$$

and

$$\mathbb{P}\left(|O_{d_2} - g_W(d_2)\mu_D| \geq \frac{\epsilon}{C_0}\right) \leq 2\exp\left(\frac{-2IL\epsilon}{C_0(b_C b_D - a_C a_D)^2}\right)$$

with C_0 to be determined. We can choose an appropriate C_0 such that

$$\left\{\frac{O_{d_1}}{O_{d_2}} \in \left[\frac{g_W(d_1)}{g_W(d_2)} - \epsilon, \frac{g_W(d_1)}{g_W(d_2)} + \epsilon\right]\right\} \subseteq \left\{|O_{d_1} - g_W(d_1)\mu_D| \leq \frac{\epsilon}{C_0}\right\} \cap \left\{|O_{d_2} - g_W(d_2)\mu_D| \leq \frac{\epsilon}{C_0}\right\}.$$

Then by letting $\gamma = 1/C_0$, we obtain (3).

For the convergence (4), we give a proof for the case of uniform distribution; the case of truncated exponential distribution can be shown in the same manner. Define $M_n = \min\{X_1, ..., X_n\}$ where $X_i \overset{i.i.d}{\sim} X_C$. We can write out the distribution of M_n and compute

$$\mathbb{E}M_n = \frac{na_C + b_C}{n+1}.$$

Then (4) follows by taking the ratio and letting W goes to infinity. More results on other distributions can be established based on the extreme value theory (see (De Haan and Ferreira 2007)).

Feature Importance in Demand Prediction

We interpret the feature importance in the demand prediction through the model of Lasso and random forest. The top 10 most useful features for both models are presented in Table 3. For the model of Lasso, the features are ranked based on the order that the features come out in the Lasso path and the result is presented in 3a. For the model of random forest, the features are ranked based on the variable importance which is defined as the amount of relative error increase caused by randomly permuting the samples of a certain feature (variable). As we can see, the top important features selected by Lasso are all about the past demand. But random forest is capable of utilizing the price features. Also the demand D_{t-1} turns out to be the most important feature for both models. D_{t-7}, D_{t-14} and D_{t-21} are usually selected because these three account for the day-of-the-week effect.

Feature	Rank
D_{t-1}	1
D_{t-21}	2
D_{t-4}	3
D_{t-5}	4
D_{t-2}	5
D_{t-3}	6
D_{t-14}	6
D_{t-7}	8
D_{t-20}	9
D_{t-6}	10

(a) Lasso path

Feature	Var. Imp
D_{t-1}	0.121
3-day price ratio for itself	0.109
D_{t-5}	0.065
D_{t-14}	0.057
1-day price ratio for itself	0.052
7-day price ratio for itself	0.034
D_{t-3}	0.032
D_{t-15}	0.032
D_{t-2}	0.031
1-day price ratio for its top-1 complimentary	0.030

(b) random forest variable importance

Table 3 Interpretation of Variable Importance