Enhancing Reputation via Price Discounts in E-Commerce Systems: A Data-Driven Approach

HONG XIE and RICHARD T. B. MA, National University of Singapore
JOHN C. S. LUI, The Chinese University of Hong Kong

Reputation systems have become an indispensable component of modern E-commerce systems, as they help buyers make informed decisions in choosing trustworthy sellers. To attract buyers and increase the transaction volume, sellers need to earn reasonably high reputation scores. This process usually takes a substantial amount of time. To accelerate this process, sellers can provide price discounts to attract users, but the underlying difficulty is that sellers have no prior knowledge on buyers' preferences over price discounts. In this article, we develop an online algorithm to infer the optimal discount rate from data. We first formulate an optimization framework to select the optimal discount rate given buyers' discount preferences, which is a tradeoff between the *short-term profit* and the *ramp-up time* (for reputation). We then derive the closed-form optimal discount rate, which gives us key insights in applying a *stochastic bandits framework* to infer the optimal discount rate from the transaction data with regret upper bounds. We show that the computational complexity of evaluating the performance metrics is infeasibly high, and therefore, we develop efficient randomized algorithms with guaranteed performance to approximate them. Finally, we conduct experiments on a dataset crawled from eBay. Experimental results show that our framework can trade 60% of the short-term profit for reducing the ramp-up time by 40%. This reduction in the ramp-up time can increase the long-term profit of a seller by at least 20%.

CCS Concepts: • **Information systems → Data mining**; **Electronic commerce**;

Additional Key Words and Phrases: E-commerce, reputation systems, price discounts, stochastic bandits, randomized algorithms

ACM Reference format:
Hong Xie, Richard T. B. Ma, and John C. S. Lui. 2018. Enhancing Reputation via Price Discounts in E-Commerce Systems: A Data Driven Approach. *ACM Trans. Knowl. Discov. Data.* 12, 3, Article 26 (January 2018), 29 pages.
https://doi.org/10.1145/3154417

1 INTRODUCTION

E-commerce services are integral parts of our modern society. In general, E-commerce systems serve as online shopping markets, where buyers can purchase products from online stores, each of which is operated by a seller. Large scale E-commerce systems include Alibaba [1], Amazon [2], eBay [8], and Taobao [26], which have generated tremendous economic values for the society.

The work of John C.S. Lui was supported in part by the GRF 14630815.
Authors' addresses: H. Xie and R. T. B. Ma, School of Computing, National University of Singapore, Computing 1, 13 Computing Drive, Singapore 117417, Republic of Singapore; emails: hxie@cse.cuhk.edu.hk, tbma@comp.nus.edu.sg; J. C. S. Lui, Ho Sin-Hang Engineering Building, The Chinese University of Hong Kong, Shatin N.T. Hong Kong, The People's Republic of China; email: cslui@cse.cuhk.edu.hk.

https://doi.org/10.1145/3154417

According to the latest Fortune 500 ranking in the total revenue, Amazon and eBay were ranked 29th and 172nd, respectively [10], and the sales of Alibaba for 2014 were estimated at $420 billion [20].

Reputation system has become an indispensable component of modern E-commerce systems, as it helps buyers make informed decisions in choosing trustworthy sellers [23]. It is critical for buyers to know whether a seller is trustworthy or not because it is quite common for buyers to purchase products from sellers whom they have never transacted with before in an E-commerce system [23]. In fact, reputation systems are needed not only to reflect the *trustworthiness* of sellers, but also to incentivize sellers to be honest [8, 26]. Consequently, the reputation system differentiates sellers in a way that a reputable seller attracts more buyers while a poor-reputation seller discourages buyers [23].

One *inefficiency* of real-world reputation systems is that new sellers may need to spend a substantial amount of time to attain a trustworthy reputation [27], and we call this the "*ramp-up time.*" More concretely, it was found in [27] that sellers in eBay need to spend around 800 days (on average) to get ramped up. One major reason for this inefficiency is that new sellers are initialized with a low reputation score, i.e., zero, and buyers are usually not willing to purchase products from sellers with low reputation scores. This forms a negative loop, and as a consequence, it is difficult for new sellers to increase their reputation scores.

Reducing the ramp-up time is highly desirable for sellers in real-world E-commerce systems. For example, more than 11,000 sellers in Taobao were found to increase their reputation scores via illegitimate methods, i.e., fake transactions [32]. Another evidence is the emergence of professional fake-transaction services, such as Lantian, Shuake, Kusha, and so on [32]. These companies create a large volume of fake transactions, e.g., the number of fake transactions created one of such companies is estimated to be 6,700,000 [32]. Since fake transactions may cause a reputation system losing the functionality of maintaining the trustworthy of sellers, they are prohibited in E-commerce systems. It is also highly risky for sellers to misbehave via creating fake transactions because once detected the seller will be heavily penalized.

In this work, we propose a legitimate way for sellers to accelerate the reputation accumulating process through price discounts. The idea is that by providing price discounts, a new seller can attract more buyers even when this seller has a low reputation score. The challenge is to determine the appropriate price discount. The larger the discounts, the more buyers a new seller can attract. However, it also reduces his profits, which is critical for a new seller to survive in a competitive E-commerce market. Therefore, the first challenge we need to address is *quantifying the tradeoffs in selecting the appropriate price discount.* Furthermore, sellers usually have no prior knowledge on buyers' preferences over price discounts, i.e., for some buyers, a small discount is sufficient to attract them to purchase a product, while other buyers may need to have a higher price discount to lure them for a transaction. The second challenge we want to address is *learning (or inferring) the buyers' preferences from the transaction data and set the appropriate price discount in an online manner.* Our contributions are

— We formulate an optimization framework to select the optimal discount rate given buyers' discount preferences, which trades the *short-term profit* against the *ramp-up time* (for reputation). We also derive the closed-form optimal discount rate. Through this we gain key insights to infer the optimal discount rate from data when sellers have no prior knowledge on buyers' discount preferences.
— Based on these insights, we first extract the features from a seller's historical transaction data and then we apply a *stochastic bandits framework* to design an online discount algorithm, which infers the optimal discount rate with regret upper bounds.

— We show that the computational complexity of evaluating the performance metrics (e.g., ramp-up time, regret upper bounds) is infeasibly high. We develop efficient randomized algorithms to approximate them and use the Chebyshev and the Hoeffding inequality to provide the guaranteed performance.
— We conduct extensive experiments on an eBay's dataset to evaluate the efficiency and effectiveness of our online discount algorithm. Experimental results show that our algorihtm can trade 60% of the short-term profit for reducing the ramp-up time by 40%. This reduction in ramp-up time increases sellers' long-term profits by at least 20%.

This article organizes as follows. Section 2 presents the system model. Section 3 presents the design objective. Section 4 presents the baseline online discount algorithm. Section 5 extends it to allow multiple reputation levels. Section 6 extends it to capture human factors like biases in assigning ratings. Section 7 presents the experimental results on a dataset from eBay. Section 8 presents related work, and we conclude in Section 9.

2 SYSTEM MODEL

A typical E-commerce system is composed of five elements: *sellers*, *buyers*, *products*, *online stores,* and a *reputation system*. Sellers post products in their online stores. The price of a product is denoted by $p \in \mathbb{R}_+$. In this article, we assume that the standard price p is known to all sellers. Each product is associated with a cost (i.e., manufacturing cost, transaction fee charged by the e-commerce operator) of $c \in \mathbb{R}_+$. A buyer pays the seller p to purchase a product and then the seller ships the product to the buyer. Let $u \triangleq p - c$ be the profit to a seller by selling one product. Sellers can advertise their products (i.e., claiming that their products are of high quality) in their online stores. In fact, many such E-commerce systems like eBay [8] or Taobao [26] provide this advertisement feature. Not all sellers are honest in their advertisements. Thus, E-commerce systems need to have some mechanisms to reflect the reputation of sellers. Table 1 summarizes the key notations.

2.1 Reputation System Model

We consider an eBay-like reputation system, which is composed of a "*feedback rating system*" and a "*rating aggregation policy.*" After receiving a product, a buyer assigns a rating to indicate the quality of the transaction, i.e., the product quality and the shopping experience. To be consistent with the eBay-like reputation system [8], we consider a three-level feedback rating system, i.e., $\{1\,(\text{positive}), 0\,(\text{neutral}), -1\,(\text{negative})\}$. A positive rating indicates that a buyer is satisfied with the transaction, while a negative rating indicates that a buyer is unsatisfied with the transaction. For the ease of presentation, we first focus on that buyers are unbiased and provide accurate ratings. In Section 6, we capture personal biases in assigning ratings.

A rating aggregation policy is applied to quantify sellers' trustworthiness from their past feedback ratings. Formally, a seller's reputation is quantified by the total sum of all his feedback ratings. Let $r \in \mathbb{Z}$ denote the reputation score of a seller. A new seller who joins an E-commerce system is initialized with $r = 0$. The reputation score r will be increased (or decreased) by one if a seller earns a positive (or negative) feedback rating, and will remain unchanged if he earns a neutral feedback rating. We use a continuous time system to characterize the reputation update dynamics. Let $r(t)$ denote a seller's reputation score at time $t \in [0, \infty)$. When a seller joins an E-commerce system (at time $t = 0$), his reputation profile is initialized as $r(0) = 0$. The reputation score is updated in a real-time manner, i.e., update the reputation score once a feedback rating is assigned.

E-commerce systems usually classify sellers into different types based on their reputation scores. For example in eBay, sellers are classified into thirteen types [9], i.e., no star ($r \leq 9$), one star ($r \in [10, 49]$), two stars ($r \in [50, 99]$), . . . , twelve stars ($r \geq 1{,}000{,}000$). The star ratings are public

Table 1. Main Notations

p, c, u	The price, cost and unit profit of a product
r, s	The reputation score and the number of stars of a seller
t	The time horizon
$\mathcal{S}$	The reputation score map
S	The maximum number of stars
n_i	The classification threshold
M	The number of positive price discount levels
λ_s	The buyers' arrival rate
P_m	The probability of buying a product under the level m discount
$T(s)$	The ramp-up time for attaining s stars
$\mathcal{M}$	A mechanism to set the price discount level
$T(s, \mathcal{M})$	The ramp-up time under the mechanism $\mathcal{M}$
$\mathcal{R}(s, \mathcal{M})$	The ramp-up time reduction
$\mathcal{L}(s, \mathcal{M})$	The short-term profit loss
$z(s, \mathcal{M})$	The objective function of the optimal mechanism design problem
α	The factor to balance the ramp-up time and shot-term profit
d_m	The price discount rate corresponds to the level m discount
$\mathcal{D}^*, \mathcal{D}^*_{of}$	The optimal price discount level set and optimal fixed discount level set
m^*, m^*_{of}	An optimal discount level and an optimal fixed discount level
$\mathcal{O}(t)$	The observed transaction data up to time t
t_i, w_i, r_i	The arrival time, waiting time and reputation score associated with the ith transaction
$\mathcal{M}^*, \widehat{\mathcal{M}}$	The optimal mechanism and online discount algorithm
$\hat{R}_i$	The reward in the ith round for the multi-armed bandit interpretation
N_m	The number of times that arm m has been pulled
P^+, P^0, P^-	The probability of receiving a positive, neutral and negative rating
$N(s)$	The number of products sold in the ramp-up process
δ, G^∞	The discounting factor and long-term profits
ϵ, ξ	The approximation error and fail probability of a randomized algorithm
K, I	The number of simulation round and truncation threshold
R_{err}, L_{err}	The ramp-up time reduction error and the short-term profit loss gap

accessible by all buyers and sellers, and they serve as a first hand reputation indicator to buyers. Formally, our model classifies sellers into $S + 1$ types, i.e., {0 star, 1 star, ..., S stars}, where $S \in \mathbb{N}_+$. Let

$$\mathcal{S} : \mathbb{Z} \rightarrow \{0, 1, \ldots, S\}$$

denote a map, which prescribes a number of stars for each reputation score, i.e., $\mathcal{S}(r) = 0$, if $r < n_1$ and $\mathcal{S}(r) = S$ if $r \geq n_S$. For $i = 1, \ldots, S-1$, $\mathcal{S}(r) = i$ if $n_i \leq r < n_{i+1}$. Here $n_1, \ldots, n_S$ denote *classification thresholds*, which satisfy $n_1 < \cdots < n_S$. For example, for the eBay system we have $S = 12$, $n_1 = 10, n_2 = 50, \ldots, n_{12} = 1{,}000{,}000$.

2.2 Buyers' Arrival Model

Reputation is crucial to a seller. On the one hand, buyers tend to trust sellers who have sufficiently large number of stars. If the number of stars is small, buyers may not even click into the store at all because buyers do not trust such sellers and do not care what this store is selling. One the other hand, a seller who has a large number of stars would be more likely to be ranked higher in either product searching or recommendation. As a result, this store will be visible to more potential buyers. To model the impact of reputation, we use a Poisson process to characterize the buyers' arrival process. One can vary the arrival rate of the Poisson process to differentiate sellers in terms of their reputation scores. More concretely, the larger the number of stars, the higher the arrival rate. Via empirical analysis of a historical transition data of eBay, Xie et al. [28] found that the arrival of sellers follows a Poisson process. Let λ_s denote the buyers' arrival rate to a seller who has s stars, where $s \in \{0, 1, \ldots, S\}$. In general, $\lambda_i \leq \lambda_j$ for all $0 \leq i \leq j \leq S$. For example, for the eBay system, we have $S = 12$ and 13 types arrival rates: $\lambda_0, \lambda_1, \ldots, \lambda_{12}$.

A seller can set one of $M + 1$ potential price discount levels $\{0, 1, \ldots, M\}$, where $M \in \mathbb{N}_+$. Here, level 0 corresponds to that a seller does *not* provide any discounts. The higher the discount level, the larger the discount rate on the price. For example, consider $M = 2$, then we have three discount levels. One possible choice for the price discount rate can be level $0 = 0\%$, level $1 = 10\%$, and level $2 = 20\%$. The price discount attracts buyers to purchase products, which in turn increase the transactions' arrival rate.

Definition 2.1. Let $P_m, m = 0, 1, \ldots, M$, denote the probability that a buyer who clicks into an online store eventually purchases a product if the seller sets a level m price discount.

We assume $P_0 < P_1 < \cdots < P_M$ to model that the higher the price discount level, the higher the probability that a buyer purchases a product. We refer to P_m as buyers' preferences over discount levels. One technical challenge is that sellers *do not know buyers' preferences over discount levels*. In other words, the values of $P_m, \forall m = 0, 1, \ldots, M$, are unknown to sellers. Note that the transactions' arrival process is still a Poisson process (via the Poisson thinning argument) and the transactions' arrival rate is $\lambda_s P_m$ when a seller has s stars and sets a level m discount.

2.3 Ramp-Up Time

Reputation significantly influences buyers' arrival rate to an online store. Therefore, it is vital for new sellers to ramp up their reputation as quickly as possible. Recall that new sellers are initialized with 0 star, with which it is quite difficult to attract buyers. Hence, one critical metric for sellers is the minimum time to earn a high enough reputation score.

Definition 2.2. Let

$$T(s) \triangleq \inf\{t | \mathcal{S}(r(t)) \geq s\}$$

denote the minimum time to earn a number of $s \in \{0, 1, \ldots, S\}$ stars. We call $T(s)$ the ramp-up time to attain an s-star label.

Note that sellers can set a target on the number of stars to be earned. The value of s is determined by sellers' self-assessment and investment budgets. If a seller has a large budget and aims for a high reputation score, he sets a large s. If a seller has a limited budget and aims for a medium reputation score, he could choose a medium value of s.

Our objective is to reduce the ramp-up time (via the price discount) for new sellers in the practical scenario that sellers have no prior knowledge on buyers' preferences over discount levels, i.e., $P_0, P_1, \ldots, P_M$ are unknown. We focus on that sellers are honest in advertising the product quality. We achieve our objective in two steps: (1) formulate an optimization framework to select the optimal discount level given full preferences (i.e., $P_0, P_1, \ldots, P_M$); (2) when these preferences are unknown, we develop online algorithms to infer the optimal discount level based on sellers' historical transaction data.

3 DESIGN OBJECTIVE

We define two metrics, namely the *ramp-up time reduction* and *short-term profit loss*, to quantify the tradeoffs in selecting discount levels given full preferences (i.e., $P_0, P_1, \ldots, P_M$). We formulate an optimization framework to select the optimal discount level subject to different tradeoffs.

3.1 Metrics

Our objective is to reduce the ramp-up time by providing price discounts in the ramp-up process. We refer to the ramp-up process as the process that a new seller earns s stars for the first time. For the simplicity of presentation, we define a mechanism $\mathcal{M}$ to represent one selection of discount levels for each product sold in the ramp-up process.

Definition 3.1. Let $\mathcal{M}$ denote a mechanism, which prescribes a price discount level for each product sold in the ramp-up process.

Thus, selecting the optimal discount levels in the ramp-up process corresponds to selecting the optimal mechanism $\mathcal{M}$. Before showing how to select the optimal $\mathcal{M}$, let us define some metrics to quantify the tradeoffs.

One benefit of the mechanism $\mathcal{M}$ is in reducing the ramp-up time, since setting a price discount can attract more buyers. To quantify this benefit, we define a metric that we call the *ramp-up time reduction*. When a seller sets no discounts at all, the expected ramp-up time is simply denoted as $\mathbb{E}[T(s)]$.

Definition 3.2. Let $T(s, \mathcal{M})$ denote the ramp-up time under the mechanism $\mathcal{M}$. We define

$$\mathcal{R}(s, \mathcal{M}) \triangleq \frac{\mathbb{E}[T(s)] - \mathbb{E}[T(s, \mathcal{M})]}{\mathbb{E}[T(s)]}$$

as the ramp-up time reduction achieved by the mechanism $\mathcal{M}$.

Notice that we consider a *normalized* ramp-up time reduction, i.e., $\mathcal{R}(s, \mathcal{M}) \in [0, 1]$. The larger the reduction $\mathcal{R}(s, \mathcal{M})$, the shorter the ramp-up time. To illustrate, assume a seller aims for one star $s = 1$, with a classifying threshold $n_1 = 3$ and a unit profit $u = 0.2p$. Consider a simple mechanism which sets the highest discount level M (assume level $M = 20\%$) for each product. Suppose $P_0 = 0.1$ and $P_M = 0.5$. Then without price discounts the ramp-up time is $\mathbb{E}[T(1)] = \frac{n_1}{\lambda_0 P_0} = \frac{3}{0.1\lambda_0}$. With the mechanism $\mathcal{M}$, the ramp-up time becomes $\mathbb{E}[T(1, \mathcal{M})] = \frac{n_1}{\lambda_0 P_M} = \frac{3}{0.5\lambda_0}$. Hence, the ramp-up time reduction is $\mathcal{R}(1, \mathcal{M}) = 0.8$. This is achieved at the price of losing an amount of $0.2p \times 3 = 0.6p$ in profits.

Notice that the mechanism $\mathcal{M}$ achieves the ramp-up time reduction at the "cost" of losing some short-term profits. Let $G(s)$ and $G(s, \mathcal{M})$ denote the short-term profits earned in the ramp-up process when a seller sets no discounts at all and uses the discount mechanism $\mathcal{M}$, respectively. This short-term profits are critical to the survivability of a new seller. Sellers open online stores with certain investment budgets, a small short-term profit may force the seller to drop out of the E-commerce system, or they may get discouraged and change to another E-commerce system. We define a metric called the *short-term profit loss* to quantify the "cost" of the mechanism $\mathcal{M}$.

Definition 3.3. We define

$$\mathcal{L}(s, \mathcal{M}) \triangleq \frac{\mathbb{E}[G(s)] - \mathbb{E}[G(s, \mathcal{M})]}{\mathbb{E}[G(s)]}$$

as the short-term profit loss due to the mechanism $\mathcal{M}$.

Again, here we consider a *normalized* profit loss. It quantities the amount of profits that a mechanism trades to quickly attain a desirable reputation (or reduce the ramp-up time). Consider the above example that illustrates the ramp-up time reduction. Note that the unit profit is $u = 0.2p$. Then the short-term profits without price discounts is $\mathbb{E}[G(1)] = n_1 u = 0.6p$. The short-term profits with mechanism $\mathcal{M}$ is $\mathbb{E}[G(1, \mathcal{M})] = n_1(u - 0.2p) = 0$. Hence, $\mathcal{L}(1, \mathcal{M}) = 1$. This means that a seller does not earn profit in the ramp-up process with $\mathcal{M}$.

3.2 Design Tradeoffs

Our objective is to design a mechanism $\mathcal{M}$, which trades price discounts for reputation subject to different tradeoffs between the *ramp-up time reduction* and the *short-term profit loss*. The optimization formation is

Problem 3.1. *Selecting the optimal discount level for each product in the ramp-up process, given buyers' discount preferences.*

$$\begin{aligned} \max_{\mathcal{M}} \quad & z(s, \mathcal{M}) \triangleq \alpha \mathcal{R}(s, \mathcal{M}) - (1 - \alpha) \mathcal{L}(s, \mathcal{M}) \\ s.\,t. \quad & \mathcal{M} \textit{ selects discount levels for products sold in the ramp-up process,} \\ & \textit{Given } P_0, P_1, \ldots, P_M, \end{aligned} \tag{1}$$

where $\alpha \in [0, 1]$ denotes a balance factor, which can be controlled by a seller. The value of α reflects the aggressiveness of a seller in reducing the ramp-up time. For example, $\alpha = 1$ implies that a seller is extremely aggressive in reducing the ramp-up time. This case occurs when a seller has a large investment budget and he does not care about the short-term profit loss. While $\alpha = 0$ implies that a seller is very keen about the short-term profits. This occurs when a seller has a small investment budget and wants to accumulate profits as soon as possible so that his online store can survive.

We next solve problem 3.1 to obtain the closed-form optimal discount level given buyers' discount preferences. The optimal discount level gives us key insights in developing online discount algorithms to infer them from sellers' transaction data when buyers' discount preferences are unknown to sellers. There are several challenges in designing and analyzing online discount algorithms. Thus, in the following of this article, we start from a simplest case. Then, we generalize our results step by step, and in each step we show how new challenges arise and how we address them.

4 RAMPING UP ONE STAR

Let us start with a simple case that sellers aim to ramp up one star. We will extent the results to ramp up multiple stars in next section. We first derive the closed-form optimal discount level given buyers' preferences over discounts, i.e., $P_0, P_1, \ldots, P_M$. The optimal discount level gives us insights to extract key features of the transaction data and apply a *stochastic bandit* framework to infer the optimal discount level online with regret upper bounds when these preferences are unknown.

4.1 Optimal Mechanism With Full Information

We consider the full information scenario, i.e., $P_0, P_1, \ldots, P_M$ are known to a seller. Our objective is to derive the closed-form optimal discount level. Throughout this section, we focus on that sellers aim to ramp up one star $s = 1$, i.e., they aim to improve their reputation from 0 star to one star. This case simplifies the problem in that buyers' arrival rate to a seller who has zero star (i.e., before gets ramped up) are homogeneous, which is equal to λ_0. Note that sellers are honest and each product earns one positive rating (In Section 6, we address errors in assigning ratings.). Hence, a seller needs to collect n_1 ratings to get ramped up. Let $d_m \in [0, 1]$ denote the price discount rate (in terms of percentage) corresponds to level m discount. A tight upper bound for $z(1, \mathcal{M})$ can be derived as

$$z(1, \mathcal{M}) \le \max_m \{\alpha + Z(m)\}. \tag{2}$$

where $Z(m)$ denotes

$$Z(m) = -\alpha \frac{1}{\lambda_0 P_m} \frac{1}{\mathbb{E}[T(1)]} - (1-\alpha) \frac{p d_m}{\mathbb{E}[G(1)]}, \tag{3}$$

$$\mathbb{E}[T(1)] = \frac{n_1}{\lambda_0 P_0}, \quad \mathbb{E}[G(1)] = n_1 u. \tag{4}$$

Inequality (2) is obtained by the linearity of expectation and that the arrival rate of buyers to the seller in the ramp-up process is homogeneous, i.e., equals λ_0. More concretely, these two properties imply that the maximum value of the objective function can be attained by setting the same discount level for each product. The right side of Inequality (2) enumerates all the possible discount levels to locate the maximum objective function value. Observe that $1/(\lambda_0 P_m)$ and $d_m p$ correspond to the expected waiting time and profit loss of a transaction with a level m price discount. This means that if we focus on one product, the optimal discount level aims to attain a balance between the waiting time and profit loss. The optimal discount set is the same for each product (or invariant of the reputation score). We can then express the closed-form optimal discount level.

Lemma 4.1. *Let $\mathcal{D}^*$ denote a set of optimal discount levels. We derive it as*

$$\mathcal{D}^* \triangleq \arg\max_m Z(m). \tag{5}$$

There can be multiple optimal discount levels, namely $|\mathcal{D}^*| > 1$. An optimal mechanism selects one of the optimal discount levels for a transaction. Let $m^* \in \mathcal{D}^*$ be an optimal discount instance and let $\mathcal{M}^*$ denote an optimal mechanism. We outline $\mathcal{M}^*$ in Algorithm 1.

ALGORITHM 1: Optimal Mechanism $\mathcal{M}^*$ for Ramping Up One Star

```
for i = 1 to n_1 do
    Compute one optimal discount levels, i.e., m* ∈ D*
    Set level m* discount for the ith transaction
end for
```

So far, we have assumed the full information scenario. However, the values of $P_0, P_1, \ldots, P_M$ are unknown in fact, which means that the optimal discount set $\mathcal{D}^*$ cannot be determined in advance. We next design an online discount algorithm to infer the optimal discount level (instances of $\mathcal{D}^*$) from historical transaction data. This involves two steps: (1) extracting features of the transaction data; (2) designing online algorithms based on these features.

4.2 Extracting Features of the Transaction Data

We present a model to extract features of the transaction data and specify the appropriate time to update the price discount.

The data model. Since a seller has access to his own historical transaction data, he can use these data to infer buyers' preferences to discounts and estimate the optimal discount level. Each transaction item corresponds to one product. More precisely, the ith transaction item carries the following meta-information: (1) the waiting time (or interarrival time) denoted by w_i, (2) the price discount level denoted by m_i, and (3) the reputation score of the seller who conducts the transaction denoted by r_i. Then, $w_i = t_i - t_{i-1}$, where $t_i \in [0, \infty)$ denotes the arrival time of the ith transaction and $t_0 = 0$. For example, consider that a new seller does not set any price discounts at all and sells two products at time $t_1 = 1$ and $t_2 = 3.5$, respectively. Then, for the first transaction item we have $w_1 = 1, m_1 = 0, r_1 = 0$, and for the second transaction item we have $w_2 = 2.5, m_2 = 0, r_2 = 1$.

A seller at time t can only observe the the data of all transactions up to time t only. Let $\tilde{N}(t)$ be the cumulative number of products that a seller has sold up to time t. We formulate a seller's observed transaction data as follows.

Definition 4.2. Let

$$O(t) \triangleq \{(t_i, m_i, r_i) | i = 1, \ldots, \tilde{N}(t)\}$$

denote the observed transaction data of a seller up to time t.

Each seller has full access to its own transaction data $O(t)$, as it contains the meta-information of each transaction only.

The discount updating model. A mechanism $\mathcal{M}$ updates the price discount level when we observe some new transaction data. This article focuses on deterministic mechanisms $\mathcal{M}$ which prescribes a unique price discount level for each observed data $O(t)$, namely

$$O(t) = O(t') \Rightarrow \mathcal{M}(O(t)) = \mathcal{M}(O(t')). \quad (6)$$

Note that the observed data $O(t)$ is updated when a transaction is completed. Namely, a mechanism updates the discount when a transaction is completed, i.e., at time $t_1, t_2, \ldots, \infty$. Hence, we have $m_i = \mathcal{M}(O(t_{i-1}))$.

4.3 Online Algorithms To Infer the Optimal Discount Level

We apply a *stochastic bandit* framework to infer optimal discount level online from a seller's transaction data. A stochastic bandit problem can be characterized by four elements: *arms, rewards*, *a forecaster,* and the *number of rounds* to be played. Typically, the number of arms is finite, and each arm is associated with a reward, which is a random variable with an unknown parameter (i.e., mean) to the forecaster. The forecaster realizes a reward after pulling an arm. The forecaster plays a finite number of rounds, and in each round the forecaster can pull only one arm and realizes a reward from that arm. The objective is to develop an arm pulling policy to maximize the expected total rewards. The key idea is that the forecaster must strike a balance between the exploitation (i.e., arms that realized large rewards in the past) and the exploration (i.e., arms that might realize large rewards in the future). One representative algorithm is the upper confidence bound (UCB) algorithm [3]. The basic version of the UCB algorithm can be expressed by the following index policy:

$$I_N \in \arg\max_m \left(\bar{R}_m + \sqrt{\frac{2\ln(N-1)}{N_m}} \right), \quad (7)$$

where I_N denotes the index of the arm pulled in round N, $\bar{R}_m$ denotes the average reward realized by arm m, N_m denotes the number of times that arm m has been pulled so far. In this policy, the term $\bar{R}_m$ corresponds to the exploitation and the term $\sqrt{\frac{2\ln(N-1)}{N_m}}$ corresponds to the exploration. The UCB algorithm has sound theoretical performance guarantees, and we refer readers to [3] for details.

The bandit interpretation. Based on the optimal discount levels expressed in (5), we now derive an equivalent formulation (i.e., stochastic bandit formulation) for our optimization problem expressed in (1), which connects the observed data $O(t)$. With some basic algebraic arguments, we have $z(1, \mathcal{M}) = \alpha + \frac{1}{n_1}\mathbb{E}[\sum_{i=1}^{n_1} \hat{R}_i]$, where

$$\hat{R}_i = -\alpha\lambda_0 P_0 w_i - (1-\alpha)\frac{pd_{m_i}}{u}, \quad (8)$$

and the optimal discount level maximizes the expectation of $\hat{R}_i$, i.e., $\arg\max_{m_i} \mathbb{E}[\hat{R}_i] = \mathcal{D}^*, \forall i = 1, \ldots, n_1$. This implies that the optimal mechanism stated in Algorithm 1 is also an optimal solution for the following optimization problem:

$$\begin{aligned} \max_{\mathcal{M}} \quad & \mathbb{E}\left[\sum_{i=1}^{n_1} \hat{R}_i\right] \\ s.t. \quad & \hat{R}_i \text{ satisfies (8)}, \\ & w_i \sim Exp(\lambda_0 P_{m_i}), \\ & \mathcal{M} \text{ satisfies (6)}. \end{aligned} \tag{9}$$

This optimization problem can be interpreted as the following stochastic bandit problem. All the potential discount levels $\{0, 1, \ldots, M\}$ correspond to $M + 1$ arms. The seller corresponds to the forecaster and he will play n_1 rounds. An algorithm updates a discount level can be viewed as pulling an arm. A reward $\hat{R}_i$ (expressed in (8)) will be realized by pulling arm m_i (or setting an m_i level discount) in ith round. As a seller, she sets her own aggressive parameter α and observes the waiting time w_i and discount rate d_{m_i} from her transaction data. One challenge here is that in the reward $\hat{R}_i$ the values of λ_0 and P_0 are unknown to sellers. The multi-armed bandit (MAB) framework requires that the reward realized by pulling an arm must be known to the forecaster. We can address this via data mining techniques because the value of $\lambda_0 P_0$ is the transactions' arrival rate in the ramp-up process when a seller does not provide any discounts. We propose to infer it from real-world E-commerce system datasets (in Section 7).

An UCB based online discount algorithm. We now apply the UCB algorithm to infer the optimal discount level online. Let us first outline our online discount algorithm in Algorithm 2. We use $\widehat{\mathcal{M}}$ to denote it. In Algorithm 2, step one to step five correspond to initialization, where the seller tries each discount level once. Step seven is the key step, which sets the appropriate discount level based on the historical transaction data. It is obtained by applying Equation (7) to our setting with some modifications. We state the regret bound in Theorem 4.3. The regret bound is a measure of performance as compare with the optimal mechanism $\mathcal{M}^*$.

ALGORITHM 2: Online Discount Algorithm $\widehat{\mathcal{M}}$ for Ramping Up One Star

1: **for** $i = 1$ to $M + 1$ **do**
2: $\quad m_i \leftarrow i - 1$
3: $\quad$ Realize reward $\hat{R}_i$ of the ith transaction via Equation (8)
4: $\quad R_{m_i} \leftarrow \hat{R}_i, N_{m_i} \leftarrow 1$
5: **end for**
6: **for** $i = M + 2$ to n_1 **do**
7: $\quad m_i \leftarrow \arg\max_m \left\{ \frac{R_m}{N_m} + \max\left\{ \frac{4\ln(i-1)}{N_m}, \sqrt{\frac{4\ln(i-1)}{N_m}} \right\} \right\}$
8: $\quad$ Realize reward $\hat{R}_i$ of the ith transaction via Equation (8).
9: $\quad R_{m_i} \leftarrow R_{m_i} + \hat{R}_i, N_{m_i} \leftarrow N_{m_i} + 1$
10: **end for**

THEOREM 4.3. *Consider $s = 1$. Algorithm 2 has the following regret upper bound:*

$$|z(1, \widehat{\mathcal{M}}) - z(1, \mathcal{M}^*)| \leq \frac{6}{n_1} \sum_{m \notin \mathcal{D}^*} (\mu_{m^*} - \mu_m) + \frac{\ln n_1}{n_1} \sum_{m \notin D^*} \max\left\{ \frac{8}{\alpha}, \frac{16}{\alpha^2} \frac{1}{\mu_{m^*} - \mu_m} \right\},$$

where $\mu_m \triangleq \mathbb{E}[\hat{R}_i | m_i = m] = -\alpha \frac{P_0}{P_m} - (1-\alpha)\frac{pd_m}{u}$ *denotes the expected reward by pulling arm m (or level m discount).*

Remark. The regret bound is asymptotically equal to $\Theta(\frac{\ln n_1}{n_1} + \frac{1}{n_1}) = \Theta(\frac{\ln n_1}{n_1})$. Namely, $|z(1, \widehat{\mathcal{M}}) - z(1, \mathcal{M}^*)| \leq \Theta(\frac{\ln n_1}{n_1})$. It implies that as n_1 goes to infinity, $|z(1, \widehat{\mathcal{M}}) - z(1, \mathcal{M}^*)|$ converges to zero. In other words, the online discount algorithm $\widehat{\mathcal{M}}$ asymptotically converges to the optimal mechanism $\mathcal{M}^*$.

5 RAMPING UP MULTIPLE STARS

We extend our model to ramp up multiple stars. We show that the optimal discount level evolves dynamically with the number of stars. As a result, naively repeating the online discount algorithm (Algorithm 2) to ramp up multiple stars may have low accuracy, especially when the number of stars increases fast with the reputation score. To address this problem, we derive and optimal fixed discount level and bound performance gap as compared to the optimal one. We extend the online discount algorithm developed in last section to infer the optimal fixed discount level and derive the regret upper bound.

5.1 The Optimal Fixed Discount Level

Generalizing our model to ramp up multiple stars, buyers' arrival rate varies with the number of stars in the ramp-up process instead of being homogeneous. This makes the optimal discount set $\mathcal{D}^*$ varies with the number of stars. Let $\mathcal{D}_i^*$ denote the optimal discount set for the ith product. We generalize the closed-form optimal discount level expressed in (5) as

$$\mathcal{D}_i^* = \arg\max_m Z_i(m), \tag{10}$$

where $Z_i(m)$ denotes

$$Z_i(m) = -\alpha \frac{1}{\lambda_{\mathcal{S}(r_i)} P_m} \frac{1}{\mathbb{E}[T(s)]} - (1-\alpha)\frac{pd_m}{\mathbb{E}[G(s)]} \tag{11}$$

$$\mathbb{E}[T(s)] = \sum_{j=1}^{n_s} \frac{1}{\lambda_{\mathcal{S}(r_j)} P_0}, \qquad \mathbb{E}[G(s)] = n_s u, \tag{12}$$

and $1/(\lambda_{\mathcal{S}(r_i)} P_m)$ is the expected waiting time of the ith transaction. Now, the optimal discount level varies with the reputation score. We still use $\mathcal{M}^*$ to denote the optimal mechanism that selects a discount level from $\mathcal{D}_i^*$ for the ith transaction.

One possible approach to infer the optimal discount level $m_i^* \in \mathcal{D}_i^*$ is by repeating the online discount algorithm outlined in Algorithm 2. More precisely, we can divide the whole ramp-up process into several sub-processes, where each sub-process increases a seller's star by one. Notice that the optimal discount level m_i^* varies with a seller's star level. In each sub-process, we run Algorithm 2 separately. However, this approach works only when the number of stars increases flatly with the reputation score, i.e., to earn one more star a seller needs to accumulate sufficiently large number of ratings (as indicated by Theorem 4.3). However, this may not hold in real-world systems, especially in the early stage when a seller joins an E-commerce system. For example, in eBay, with 10 positive ratings, a new seller earns one star, then 40 ratings can earn another star, and 50 ratings can earn another star. Formally, when the number of stars increases fast with the reputation scores, the transaction data in each sub-process will not be sufficient for the learning algorithm to produce an accurate estimation. This motivates us to develop a general methodology to set the appropriate discount instead of repeating Algorithm 2 naively.

To address the above challenge, we propose to infer an optimal fixed discount level, i.e., the best choice in the scenario that a seller sticks to one discount level in the ramp-up process. The underlying tradeoff is that the optimal fixed discount level may not be globally optimal, however, as we will show later that the seller can use the transaction data in the whole ramp-up process (instead of a sub-process) to infer it and thus have a more accurate estimation. Formally, we proceed to show the following: (1) the optimal fixed discount level is good enough in practice; (2) we can infer this discount level with performance guarantees (Section 5.2). Let $\mathcal{D}^*_{of}$ denote the set of optimal fixed discount levels. With a similar analysis as Section 4.1, we have

$$\mathcal{D}^*_{of} = \arg\max_{m} \left(\alpha \left(1 - \frac{P_0}{P_m} \right) - (1-\alpha)\frac{pd_m}{u} \right). \tag{13}$$

In fact, $1 - \frac{P_0}{P_m}$ and $\frac{d_m p}{u}$ correspond to the ramp-up time reduction and the short-term profit loss, respectively, for a given level m discount. Let $\mathcal{M}^*_{of}$ denote an optimal fixed discount mechanism, which selects a discount level from $\mathcal{D}^*_{of}$ for each product. We next state the gap between $\mathcal{M}^*_{of}$ and the optimal mechanism.

Theorem 5.1. *The performance gap between the optimal fixed mechanism $\mathcal{M}^*_{of}$ and the optimal mechanism $\mathcal{M}^*$ is*

$$|z(s, \mathcal{M}^*_{of}) - z(s, \mathcal{M}^*)| = \frac{\alpha}{\mathbb{E}[T(s)]}\Delta T(n_s) + \frac{1-\alpha}{\mathbb{E}[G(s)]}\Delta G(n_s),$$

where $\Delta T(j)$ and $\Delta G(j)$ are defined as

$$\Delta T(j) = \sum_{i=1}^{j} \left(\frac{1}{\lambda_{\mathcal{S}(r_i)} P_{m^*_{of}}} - \frac{1}{\lambda_{\mathcal{S}(r_i)} P_{m^*_i}} \right), \qquad \Delta G(j) = p\sum_{i=1}^{j} \left(d_{m^*_{of}} - d_{m^*_i} \right),$$

and $m^*_{of} \in \mathcal{D}^*_{of}$ and $m^*_i \in \mathcal{D}^*_i$.

The gap is zero when sellers aim to ramp up one star. This gap reflects the dynamics of the optimal discount level. As we will show in Section 7.3, this gap is in fact very small in practice. Note that we can derive the closed-form $z(s, \mathcal{M}^*_{of})$ and $z(s, \mathcal{M}^*)$. The performance gap is obtained by plugging these closed-form expressions into $|z(s, \mathcal{M}^*_{of}) - z(s, \mathcal{M}^*)|$ and by some basic algebra operations.

5.2 Inferring the Optimal Fixed Discount Level

We now extend Algorithm 2 to infer an optimal fixed discount level $m^*_{of} \in \mathcal{M}^*_{of}$. First, we connect m^*_{of} with the transaction data. Note that $\mathbb{E}[\hat{R}_i] = -\alpha\frac{P_0}{P_{m_i}} - (1-\alpha)\frac{pd_{m_i}}{u} = \alpha(1 - \frac{P_0}{P_{m_i}}) - (1 - \alpha)\frac{pd_{m_i}}{u} - \alpha$, where $\hat{R}_i$ is derived as

$$\hat{R}_i = -\alpha\lambda_{\mathcal{S}(r_i)} P_0 w_i - (1-\alpha)\frac{p}{u} d_{m_i}. \tag{14}$$

This implies that $D^*_{of} = \arg\max_{m_i} \mathbb{E}[\hat{R}_i]$, where $\hat{R}_i$ satisfies (14). Hence, inferring the optimal fixed discount level is equivalent to the following stochastic bandit optimization problem:

$$\begin{aligned} \max_{\mathcal{M}} \quad & \mathbb{E}\left[\sum_{i=1}^{n_s} \hat{R}_i\right] \\ s.t. \quad & \hat{R}_i \text{ satisfies (14)}, w_i \sim Exp(\lambda_{\mathcal{S}(r_i)} P_{m_i}), \mathcal{M} \text{ satisfies (6)}. \end{aligned} \tag{15}$$

One can observe that we can use the transaction data in the whole ramp-up process to infer the optimal fixed discount level. This optimization generalizes the transactions' arrival rate in the optimization problem stated in (9) from $\lambda_0 P_0$ to $\lambda_{\mathcal{S}(r_i)} P_0$. Namely, from a homogeneous transactions' arrival rate to a heterogeneous transactions' arrival rate, which is sensitive to the reputation score. One technical note is that the value of $\lambda_{\mathcal{S}(r_i)} P_0$ is unknown in general. Again, we propose to infer it from real-world E-commerce system datasets (Section 7).

We extend Algorithm 2 to infer the optimal fixed discount level. We outline it in Algorithm 3. We still call it the online discount algorithm and use $\widehat{\mathcal{M}}$ to denote it. The key difference lies in that we generalize the transactions' arrival rate (without discounts) in Algorithm 2 from $\lambda_0 P_0$ to $\lambda_{\mathcal{S}(r_i)} P_0$ and generalize the number of rounds from n_1 to n_s. We can also derive a regret bound, i.e., $|z(s, \widehat{\mathcal{M}}) - z(s, \mathcal{M}^*_{of})|$, which is similar with Theorem 4.3.

ALGORITHM 3: Online Discount Algorithm $\widehat{\mathcal{M}}$ for Ramping Up s Stars

for $i = 1$ to $M + 1$ **do**
    $m_i \leftarrow i - 1$.
    Realize reward $\hat{R}_i$ of the ith transaction via (14)
    $R_{m_i} \leftarrow \hat{R}_i$, $N_{m_i} \leftarrow 1$
end for
for $i = M + 2$ to n_s **do**
    $m_i \leftarrow \arg\max_m \left\{ \frac{R_m}{N_m} + \max\left\{ \frac{4\ln(i-1)}{N_m}, \sqrt{\frac{4\ln(i-1)}{N_m}} \right\} \right\}$
    Realize reward $\hat{R}_i$ of the ith transaction via (14)
    $R_{m_i} \leftarrow R_{m_i} + \hat{R}_i$, $N_{m_i} \leftarrow N_{m_i} + 1$
end for

Our results thus far assume that buyers are unbiased and provide accurate ratings to sellers. In practice, ratings are subject to personal biases or preferences. Such biases lead to that a buyer assigns neutral or even negative ratings to a honest seller. We next extend our results to capture such personal biases.

6 HUMAN FACTORS

We now extend our online discount algorithms to incorporate human factors like personal preferences or biases in assigning ratings. We show that this leads to computational infeasibility in evaluating the optimal discount level and various performance metrics (e.g., ramp-up time, regret upper bounds). We design computationally efficient randomized algorithms (with theoretical performance guarantees) to approximate them.

6.1 Online Discount Algorithm Under Human Factors

Human Factor Model. Ideally, a buyer should assign a positive rating if a seller is honest. However, due to human factors such as personal biases (or preferences) a buyer may assign a neutral or negative rating. We use a probabilistic model to capture the collective biases of the whole buyer population. Let P^+, P^0, P^- denote the probability that a seller receives a positive, neutral, and negative rating respectively, where $P^+ + P^0 + P^- = 1$. One can vary the values of P^+, P^0, P^- to reflect different levels of personal biases. For example, $P^+ = 1$ implies there are no personal biases, and the smaller the P^+, the higher the level of personal biases. We like to point out that P^+, P^0, and P^- are independent of a seller's status, as they capture buyers' personal biases in perceiving the product quality. They model the overall rating bias of the whole buyer population.

We now generalize the optimal discount set $\mathcal{D}_i^*$ to accommodate human factors. In fact, the human factors lead to dynamics of reputation score, i.e., the reputation score may increase or decrease instead of always increasing. We express the optimal discount set as $\mathcal{D}_i^* = \arg\max_m Z_i(m)$. This expression looks the same as Equation (10). However, it generalizes Equation (10) to capture human factors. More concretely, it captures human factors via generalizing $\mathbb{E}[T(s)]$ and $\mathbb{E}[G(s)]$ nested in Equation (10) to Lemmas 6.3 and 6.4 (will be shown later). The transactions' arrival rate $\lambda_{\mathcal{S}(r_j)}P_0$ encodes the human factors in the reputation score. As we will show later, it is computationally expensive to compute the optimal discount level, due to high complexity in evaluating $\mathbb{E}[T(s)]$ and $\mathbb{E}[G(s)]$ (Lemmas 6.3 and 6.4). Again, we use $\mathcal{M}^*$ to denote the optimal mechanism.

We extend the optimal fixed discount mechanism in Section 5 to incorporate human factors. We express the optimal fixed discount set as $\mathcal{D}_{of}^* = \arg\max_m(\alpha(1-\frac{P_0}{P_m}) - (1-\alpha)\frac{d_m p}{u})$. Compared with Equation (13), it is important to observe that the optimal fixed discount set is invariant of human factors. Again we use M_{of}^* to denote the optimal fixed discount mechanism. Let $N(s)$ denote the number of products sold in the ramp-up process. When we do not consider human factors, $N(s)$ is equal to n_s. When human factors are considered, $N(s)$ is a random variable with a probability mass function derived in Equation (22). We next extend Theorem 5.1 to incorporate human factors into the performance gap and prove that it is computationally expensive to evaluate it.

Theorem 6.1. *Consider human factors in assigning ratings, the performance gap between M_{of}^* and $\mathcal{M}^*$ is*

$$|z(s, M_{of}^*) - z(s, \mathcal{M}^*)| = \sum_{j=n_s}^{\infty} \mathbb{P}[N(s)=j]\left(\frac{\alpha}{\mathbb{E}[T(s)]}\mathbb{E}[\Delta T(j)|N(s)=j] + \frac{1-\alpha}{\mathbb{E}[G(s)]}\mathbb{E}[\Delta G(j)|N(s)=j]\right),$$

where $\mathbb{P}[N(s)=j]$ is stated in (22). Evaluating the exact value for this gap is with a complexity of $\Omega(\sum_{j=n_s}^{\infty} j)$.

To infer the optimal fixed discount level $m_{of}^* \in \mathcal{D}_{of}^*$, we first generalize the optimization framework stated in (15) to incorporate human factors:

$$\begin{aligned} \max_{\mathcal{M}} \quad & \mathbb{E}\left[\sum_{i=1}^{N(s)} \hat{R}_i\right] \\ s.t. \quad & \hat{R}_i \text{ satisfies (14)}, w_i \sim Exp(\lambda_{\mathcal{S}(r_i)}P_{m_i}), \\ & N(s) \text{ has a pmf derived in (22), get ramped up after collecting } N(s) \text{ ratings,} \end{aligned} \tag{16}$$

It generalizes (15) in two aspects: (1) from collecting n_s ratings to $N(s)$ ratings, which is a random variable; (2) generalizing the increase of the reputation score by one after each transaction to the case that the reputation score can increase or decrease, and it satisfies the ramp-up condition for the first time when $N(s)$ ratings are collected. Note that the second generalization influences the transactions' arrival rate, i.e., $\lambda_{\mathcal{S}(r_i)}P_0$. With these observations, we next utilize Algorithm 3 as follows to incorporate human factors. First, we extend the stopping rule from collecting n_s ratings to the case that the reputation score hits n_s for the first time. Note that the transactions' arrival rate $\lambda_{\mathcal{S}(r_i)}P_0$ in Algorithm 3 is general enough to reflect the update dynamics of the reputation score. We still use the $\widehat{\mathcal{M}}$ to denote this online discount algorithm. We next extend Theorem 4.3 to derive the regret upper bound of this algorithm and prove that it is computationally expensive to evaluate the regret upper bound.

Theorem 6.2. *In the presence of human factors, we have*

$$|z(s,\widehat{\mathcal{M}}) - z(s,M^*_{of})| \leq \sum_{j=n_s}^{\infty} \mathbb{P}[N(s)=j] \left(\frac{\ln j}{j} \sum_{m \notin D^*_{of}} \max\left\{ \frac{8}{\alpha}, \frac{16}{\alpha^2} \frac{1}{\Delta_m} \right\} + \frac{6}{j} \sum_{m \notin \mathcal{D}^*_{of}} \Delta_m \right), \tag{17}$$

where $\Delta_m = \mu_{m^*_{of}} - \mu_m$, μ_m is defined in Theorem 4.3 and $\mathbb{P}[N(s) = j]$ is derived in (22). The complexity of evaluating the exact value of this regret bound is $\Omega(\sum_{j=n_s}^{\infty} j)$.

6.2 Addressing Computational Challenges

Complexity analysis. To illustrate the computational complexity in evaluating the ramp-up time, we consider ramping up one star, i.e., $s=1$, and a seller sets no price discounts.

Lemma 6.3. *Suppose $s=1$ and a seller sets no discounts. We express the ramp-up time as $\mathbb{E}[T(1)] = \sum_{i=n_s}^{\infty} \mathbb{P}[N(s)=i] \frac{i}{\lambda_0 P_0}$, where $\mathbb{P}[N(s) = i]$ is derived in Equation (22). The complexity of evaluating the exact value of this expression is $\Omega(\sum_{i=n_s}^{\infty} i)$.*

It is computationally expensive to evaluate the ramp-up time even for ramping up one star. This is because in the presence of human factors, the sample space for the ramp-up process is very large and we need to enumerate all of them. As one can imagine, when sellers aim to ramp up multiple stars, or they set some discounts, the computational complexity would be higher. Similarly, in Lemma 6.4 we use a simplified case (a seller sets no discounts at all) to illustrate that it is computationally expensive to evaluate the short-term profit.

Lemma 6.4. *Suppose a seller sets no discounts. We express the short-term profit as $\mathbb{E}[G(s)] = \sum_{i=n_s}^{\infty} \mathbb{P}[N(s) = i]iu$, where $\mathbb{P}[N(s) = i]$ is derived in Equation (22). The complexity of evaluating the exact value of this expression is $\Omega(\sum_{i=n_s}^{\infty} i)$.*

Reducing the ramp-up time is at the "cost" of losing some short-term profits. It is interesting to ask: *For a survived seller, how will this mechanism influence its profits in the long run?* To quantify the long-term impact, we define a long-term discounted profit metric where the profit earned from the ith transaction (arrival time is t_i) is discounted by a factor of δ^{t_i}, and $\delta \in (0, 1)$ is the discounting factor. Let

$$G^{\infty} \triangleq \mathbb{E}\left[\sum_{i=1}^{\infty} u_i \delta^{t_i} \right] \tag{18}$$

denote the long-term discounted profit, where u_i denotes the profit for ith product. We also call G^{∞} the long-term profit for short. We next derive its closed-form expression.

Lemma 6.5. *Consider a seller sets no discounts:*

$$G^{\infty} = \sum_{i=1}^{\infty} u \sum_{\{(r_1,\ldots,r_i)\}} \prod_{j=1}^{i} \frac{\lambda_{\mathcal{S}(r_j)} P_0}{\lambda_{\mathcal{S}(r_j)} P_0 - \ln \delta} \prod_{\ell=2}^{i} (P^{-} \mathbf{I}_{\{r_\ell = r_{\ell-1} - 1\}} + P^{0} \mathbf{I}_{\{r_\ell = r_{\ell-1}\}} + P^{+} \mathbf{I}_{\{r_\ell = r_{\ell-1} + 1\}}) \tag{19}$$

The computation complexity of evaluating exact value of G^{∞} is $\Omega(\sum_{i=1}^{\infty} i)$.

When sellers do not set any price discounts, it is computationally expensive to evaluate G^{∞}. The complexity would be higher if a seller sets price discounts due to variation in μ.

Randomized Algorithms. We develop efficient randomized algorithms with theoretical guarantees to approximate the ramp-up time, profits (both short-term and long-term), performance gap (Theorem 6.4), regret bounds (Theorem 17), and so on.

One can approximate the ramp-up time via the stochastic Monte Carlo framework [22]. The key idea is that we simulate the reputation update process for $K \in \mathbb{N}_+$ rounds. We obtain one sample of the ramp-up time $T(s)$ in each round via simulating the reputation accumulation process until a seller finally get ramped up. We use the average of these K samples, which we denote as $\widehat{\mathbb{E}}[T(s)]$, to estimate $\mathbb{E}[T(s)]$. We present the stochastic Monte Carlo algorithm in Algorithm 4.

ALGORITHM 4: Randomized Algorithm for $\mathbb{E}[T(s)]$

Require: Ramping star $s, \lambda_0, \lambda_1, \ldots, \lambda_S, P_0, P_1, \ldots, P_M, \mathcal{M}$.
1: **for** $i = 1$ to K **do**
2: $\quad r \leftarrow 0, T_i \leftarrow 0, O(0) = \emptyset$
3: $\quad$ **while** $r < n_s$ **do**
4: $\qquad$ If $\mathcal{M} = \emptyset$, $m \leftarrow 0$, else $m \leftarrow \mathcal{M}(O(T_i))$
5: $\qquad w \sim Exponential(\lambda_{S(r)} P_m)$
6: $\qquad O(T_i + w) = O(T_i) \cup \{(T_i + w, m, r)\}, T_i \leftarrow T_i + w$
7: $\qquad r \leftarrow r + Multinomial(P^-, P^0, P^+) - 2$
8: $\quad$ **end while**
9: **end for**
10: **return** $\widehat{\mathbb{E}}[T(s)] \leftarrow \sum_{i=1}^{K} T_i / K$

In Algorithm 4, step 2 corresponds to the initialization. In step 4, we apply a discount mechanism to generate a discount level, where the a level 0 discount is generated if a seller does not set any discounts, i.e., $\mathcal{M} = \emptyset$. Step 5 generates a waiting time of a transaction. Step 6 updates the observed data and the ramp-up time when a transaction arrives. Step 7 generates a rating for this transaction and uses this rating to update the reputation score. Step 10 estimates the ramp-up time using the sample average. We next analyze the computational complexity of Algorithm 4, and apply Hoeffding's inequality [22] to derive the appropriate simulation rounds K to guarantee an accurate estimation.

Definition 6.6. Let $\epsilon \in \mathbb{R}_+$ denote the approximation error and let $\xi \in [0, 1]$ denote the fail probability of a randomized algorithm.

THEOREM 6.7. *The expected computational complexity for Algorithm 4 is*

$$O\left(\frac{K}{(P^+ - P^-)^2} + \frac{K n_s}{P^+ - P^-}\right).$$

If the number of simulation rounds satisfies

$$K \geq \frac{1}{\epsilon^2} \frac{\lambda_{s-1} P_M}{\lambda_0 P_0} \frac{1}{n_s},$$

then Algorithm 4 guarantees $|\widehat{\mathbb{E}}[T(s)] - \mathbb{E}[T(s)]| \leq \epsilon \mathbb{E}[T(s)]$ *with probability at least* $1 - \xi$.

We can now approximate the short-term profit $\mathbb{E}[G(s)]$. We can extend Algorithm 4 to estimate $\mathbb{E}[G(s)]$ with some minor modifications. We can also derive similar theoretical performance guarantees as Theorem 6.7. Furthermore, this idea also applies to compute the estimate for the performance gap derived in Theorem 6.1 and the regret bound derived in Theorem 6.2. For brevity, we omit the details.

We next develop a Monte Carlo algorithm to evaluate the long-term profit derived in Equation (19). It is computationally expensive to obtain one sample of G^∞, because we need to simulate infinity number of transactions. To address this challenge, we obtain some approximate samples via truncation. That is, simulate until we obtain $I \in \mathbb{N}_+$ ratings. We outline this idea in Algorithm 5.

ALGORITHM 5: Randomized Algorithm for G^∞

Require: Ramping star $s, \lambda_0, \lambda_1, \ldots, \lambda_S, P_0, P_1, \ldots, P_M, \mathcal{M}$.
1: **for** $i = 1$ to K **do**
2: $r \leftarrow 0, G_i \leftarrow 0, t_i = 0, O(0) = \emptyset$
3: **for** $j = 1$ to I **do**
4: If has not got ramped up $m \leftarrow \mathcal{M}(O(t_i))$, else $m \leftarrow 0$
5: $w \sim \mathit{Exponential}(\lambda_{S(r)} P_m)$
6: $O(t_i + w) = O(t_i) \cup \{(t_i + w, m, r)\}$, $t_i \leftarrow t_i + w$
7: $G_i \leftarrow G_i + (u - d_m p)\delta^{t_i}$
8: $r \leftarrow r + \mathit{Multinomial}(P^+, P^0, P^-) - 2$
9: **end for**
10: **end for**
11: **return** $\widehat{G}^\infty \leftarrow \sum_{i=1}^{K} G_j / K$

In Algorithm 5, step 2 corresponds to the initialization. Step 4 generates a discount level if a seller is in the ramp-up process, otherwise generates a level zero discount. Step 5 generates the waiting time of a transaction. Step 6 updates the observed transaction data. Step 7 adds the profit of the new transaction to the long profit. Step 8 generates a rating for this transaction and uses this rating to update the reputation score. Step 11 estimates the long-term profit using the sample average. We next analyze the complexity of Algorithm 5, and apply Chebyshev's Inequality [22] to derive the number of simulation rounds M and truncating threshold I to guarantee an accurate estimation.

Theorem 6.8. *The complexity for Algorithm 5 is $O(KI)$. If the simulation rounds satisfies*

$$K = \Theta\left(\frac{1}{\epsilon^2 \xi}\left(\frac{\max\{\lambda_{s-1} P_M, \lambda_S\}}{(\lambda_0 P_0)}\right)^2\right)$$

and truncating threshold satisfies

$$I \geq \left(\ln\frac{\epsilon}{2} - \ln u - \ln\left(1 + \frac{\max\{\lambda_s P_M, \lambda_S\}}{\ln \delta^{-1}}\right)\right) \Big/ \ln\left(1 + \frac{\ln \delta^{-1}}{\max\{\lambda_s P_M, \lambda_S\}}\right),$$

then $|\widehat{G}^\infty - G^\infty| \leq \epsilon$ with probability of at least $1 - \xi$.

7 EXPERIMENTS ON EBAY DATA

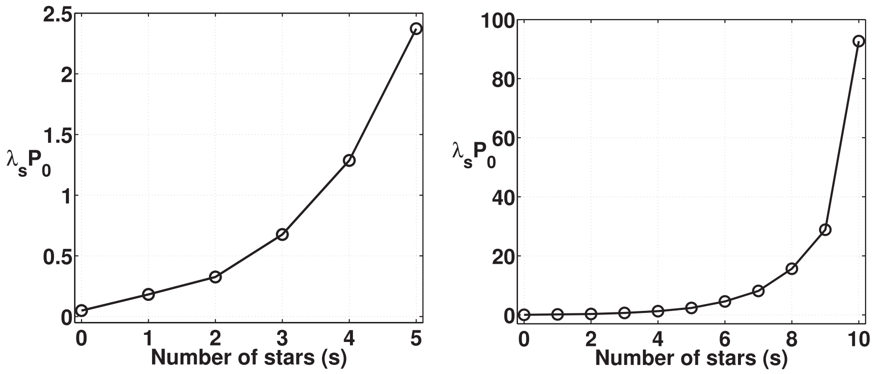
We present experimental results on a dataset from eBay. In summary, our results show that a seller needs to spend 203 days to earn one star, 426 days to earn two stars, and so on. Our optimal fixed discount mechanism achieves at least 90% of the performance of the optimal mechanism. Furthermore, the performance gap (in terms of the ramp-up time reduction or the short-term profit loss) between our online discount algorithm and the optimal mechanism is at most 20%. Our online discount algorithm can trade 60% of the short-term profit for reducing the ramp-up time by 40%. More importantly, it also increases the long-term profit by at least 20% for survived online stores.

7.1 Dataset and Parameter Inference

We use a dataset from eBay, which was published by [29]. Table 2 presents its overall statistics. It contains historical ratings of 4,362 sellers received from the first day that a seller joined the eBay

Table 2. Overall Statistics for an eBay Dataset

total # of sellers	4,362
total # of ratings	18,533,913
maximum/minimum on the # of ratings per seller	117,100/1
mean/median on the # of ratings per seller	4,190/1,437



(a) Star 0 to star 5 (b) Star 0 to star 10

Fig. 1. Transactions' arrival rate across the number of stars.

till April 2013. In eBay, buyers transact with sellers online. A buyer assigns a rating to indicate the quality of a transaction. A rating can be one of three levels, i.e., {−1 (negative), 0 (neutral), 1 (positive)}. Each seller is tagged with a reputation score. A new seller has a reputation score of 0. It is increased (or decreased) by one when a positive (or negative) rating is assigned. It remains unchanged if a neutral rating is assigned. Based on the reputation score, sellers are classified into 13 types, i.e., {no star, 1 star, 2 stars, ..., 12 stars} [9]. More specifically, no star ($r \leq 9$), one star ($[r \in 10, 49]$), and so on [9].

Let us first infer the transactions' arrival rate for the case that sellers do not set any discounts. Consider sellers having s stars, we compute their transactions' arrival rate, i.e., $\lambda_s P_0$, as their total number of completed transactions divided by the total amount of time to accumulate these transactions:

$$\lambda_s P_0 = \frac{\#[\text{transations by sellers having } s \text{ stars}]}{\text{total time to accumulate these transactions}}. \tag{20}$$

Figure 1 presents the transactions' arrival rate inferred from the dataset. One can observe that the transactions' arrival rate increases with the number of stars. The transactions' arrival rate corresponds to zero star is very close to zero. The transactions' arrival rate to three-star sellers is around 0.5 per day. These results indicate that new online stores have difficulty to attract buyers.

We now infer the variation in assigning ratings, i.e., P^+, P^0, P^-. We infer P^+ as the fraction of positive ratings across all sellers:

$$P^+ = \frac{\#[\text{positive ratings across all sellers}]}{\#[\text{ratings across all sellers}]}.$$

We have $P^+ = 0.9943$. Similarly, $P^0 = 0.0034$, $P^- = 0.0023$.

Table 3. Ramp-Up Time Across Number of Stars

# of stars to ramp	1	2	3	4	5	6
Transactions' arrival rate	0.18	0.33	0.68	1.29	2.37	4.59
$\mathbb{E}[T(s)]$	202.90	425.92	578.96	1,171.41	1,561.54	3,262.58

7.2 The Ramp-Up Time Without Discounts

Let us study the ramp-up time without discounts. We input the above inferred parameters into our model and apply Algorithm 4 to compute the expected ramp-up time. Using Theorem 6.7, we set $K = 10^6$ for Algorithm 4. We present the ramp-up across the number of stars s in Table 3. One can observe that on average a seller needs to spend around 202.90 days to earn one star, 425.92 days to earn two stars, and 578.96 days to earn three stars. The transactions' arrival rate corresponds to three stars is 0.68 per day, which is quite small. Namely, getting ramped up to three stars may not be sufficient, even though it take a long time. One can observe that ramping up to five stars may be good enough, since the transactions' arrival rate is 2.37. However, the corresponding ramp-up time is 1,561.54 days, a very long duration. This is a clue that new sellers are difficulty to survive and are discouraged to join. It also uncovers a reason why in practice some sellers conduct fake transactions to increase their reputation especially in early stages [32].

7.3 Evaluating the Optimal Fixed Discount Mechanism

We show that the optimal fixed discount mechanism is a good approximation of the optimal mechanism. We consider

$$\text{Gap ratio} \triangleq |z(\mathcal{M}^*_{of}) - z(\mathcal{M}^*)|/z(\mathcal{M}^*). \tag{21}$$

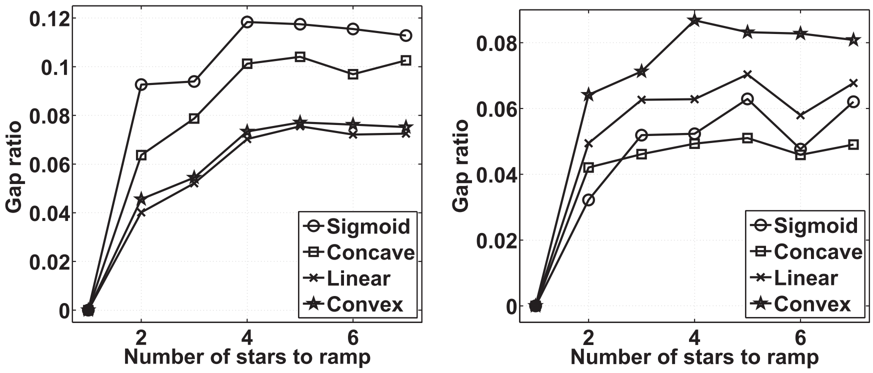
The gap ratio quantifies the accuracy of the optimal fixed discount mechanism in approximating the optimal mechanism. The smaller the gap ratio, the higher the accuracy.

Now, we introduce the evaluation settings. We consider six levels of discounts (or $M + 1 = 6$) and the discount rate corresponds to level m is $d_m = m \times 5\%$. The value of $P_0, P_1, \ldots, P_5$ are not known. We synthesize them to reflect the real-world scenario as accurate as possible. Without loss of generality, we consider four representative types of buyers' preference to discounts:

$$\begin{aligned}
&(1)\ \text{Sigmoid: } P_m = 0.5/(1 + e^{-(m-3)}), && m = 0, 1, \ldots, 5,\\
&(2)\ \text{Concave: } P_m = 0.0237(m+1)^{0.5}, && m = 0, 1, \ldots, 5,\\
&(3)\ \text{Linear: } P_m = 0.0237(m+1), && m = 0, 1, \ldots, 5,\\
&(4)\ \text{Convex: } P_m = 0.0237(m+1)^{1.5}, && m = 0, 1, \ldots, 5,
\end{aligned}$$

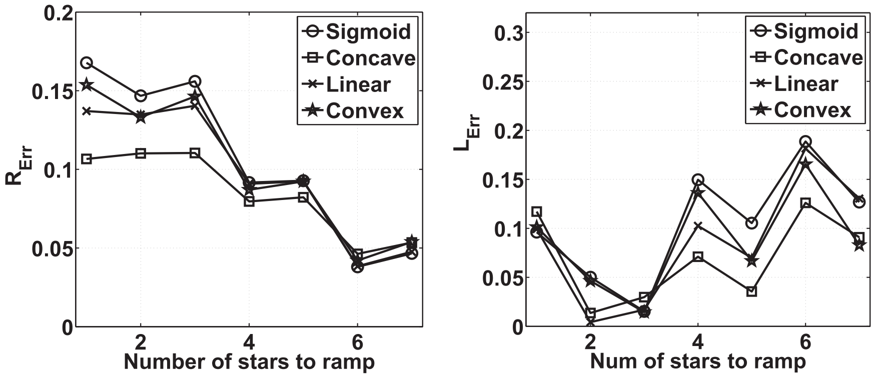
where the parameters 0.0237 and 0.5 are carefully selected to guarantee that the value of P_0 (probability of purchasing without discounts) is the same for these four preference models. Each of these four preference models increases in the discount level meaning that the higher the discount level the higher the adopting probability. Furthermore, they represent four typical types of increasing behaviors, i.e., sigmoid, concave, linear and convex, which capture buyers' sensitivity to discount.

Figure 2 presents the experimental results on the gap ratio (defined in Equation (21)), where Figure 2(a) corresponds to $\alpha = 0.5$ and Figure 2(b) corresponds to $\alpha = 0.8$. We vary the number of stars that a seller aims to earn in the ramp-up process from one star to seven stars. We also call it the ramp-up star. One can observe that the gap ratio is roughly at most 0.1. This means that the optimal fixed mechanism can achieve at least 90% of the optimal mechanism. Namely our optimal fixed mechanism is close to the optimal mechanism. When the ramping-up star is one, i.e., sellers aims for one star. The gap ratio is zero. This is because the buyers' arrival rate to a seller



(a) $\alpha = 0.5$ (b) $\alpha = 0.8$

Fig. 2. Performance gap ratio between $\mathcal{M}^*_{of}$ and $\mathcal{M}^*$.



(a) Ramp-up time reduction (b) Short-term profit loss

Fig. 3. Performance gap ratio between $\widehat{\mathcal{M}}$ and $\mathcal{M}^*$.

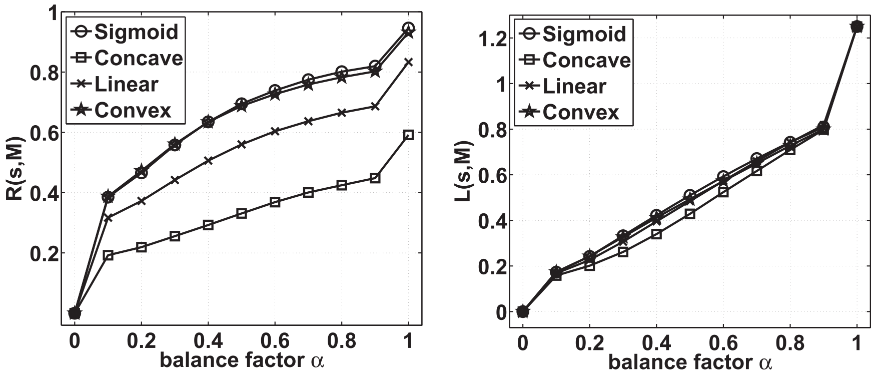
in the ramp-up process is homogeneous, i.e., λ_0. As we increase the ramp-up star, the gap ratio increases roughly. The reason is that increasing the ramp-up star increases the heterogeneity in the buyers' arrival rate in the ramp-up process. As we increase the ramping up star beyond four, the gap ratio curve becomes flat. This implies that the optimal fixed discount mechanism is robust in approximation.

7.4 Evaluating the Online Discount Algorithm

Now we evaluate the online discount algorithm in terms of the ramp-up time reduction and the short-term profit loss. We compare our online discount algorithm $\widehat{\mathcal{M}}$ with the optimal mechanism $\mathcal{M}^*$. We study the gap in the ramp-up time reduction and the short-term profit loss. Formally,

$$R_{err} \triangleq |\mathcal{R}(s, \widehat{\mathcal{M}}) - \mathcal{R}(s, \mathcal{M}^*)|, \qquad L_{err} \triangleq |\mathcal{L}(s, \widehat{\mathcal{M}}) - \mathcal{L}(s, \mathcal{M}^*)|.$$

We adopt these two metrics because sellers are most interested in the ramp-up time reduction and the profit loss. We adopt the experimental setting used in Section 7.3 and use the inferred parameters in our experiments. Figure 3(a) and (b) presents R_{err} and L_{err}, respectively. Examining Figure 3(a), one can observe that both R_{err} and L_{err} are less than 0.2. As we increase the ramp-up



(a) Ramp-up time reduction (b) Short-term profits loss

Fig. 4. Ramp-up time and short-term profits.

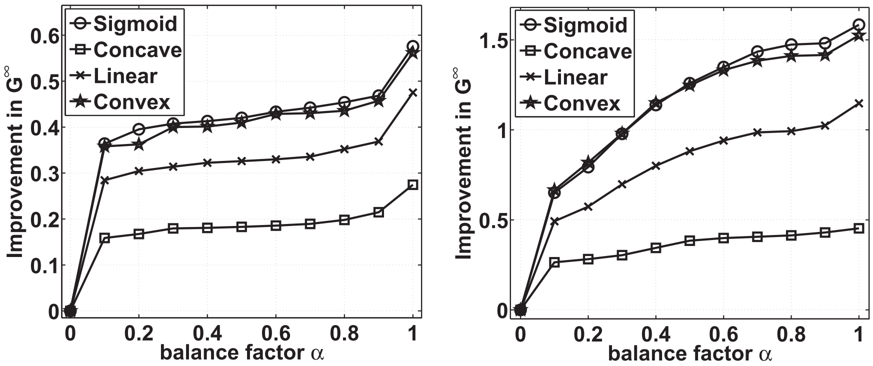
star, R_{err} decreases. When the ramp-up star is more than three, R_{err} is less than 0.1. This shows that in terms of the ramp-up time reduction and short-term profit loss, our online discount algorithm produces an accurate approximation.

7.5 Ramp-Up Time and Short-Term Profits

Now, we study the ramp-up time and short-term profit under our online discount algorithm. We adopt the experimental setting used in Section 7.3 and use the inferred parameters in our experiments. Without loss of generality, sellers aim to earn five stars in our experiments. Figure 4 presents the experimental results on the ramp-up time reduction and the short-term profit loss, respectively. Figure 4(a) shows that as the α increases, the ramp-up time reduction increases. In other words, as a seller becomes more aggressive in reducing the ramp-up time, our online discount algorithm can significantly reduce the ramp-up time. However, as shown in Figure 4(b) that this is achieved at a "cost" of losing more short-term profits. If a seller has a large investing budget and does not care about the short-term profit (i.e., $\alpha = 1$), our algorithm can reduce the ramp-up time by at least 60% trading around 1.2 times the short-term profit (i.e., a seller losing $0.2G(s)$). If a seller has a moderate investing budget (i.e., $\alpha = 0.6$) our algorithm can reduce the ramp-up time by at least 40% by trading 60% of the short-term profit. If a seller has a small investing budget (i.e., $\alpha = 0.1$), we can reduce the ramp-up time by at least 20% by trading 20% of the short-term profit.

7.6 Long-Term Profits

Now, we study the impact our online discount algorithm on the long-term profit. We consider the long-term profit improvement ratio of our mechanism over the case without discounts. We adopt the experimental setting used in Section 7.3 and a discounting factor $\delta = 0.999$. We consider two cases of the ramp-up star, i.e., earning five stars and earning three stars. We input the inferred model parameters into our model and apply Algorithm 5 to evaluate the long-term profit setting $K = 10^6$ and $I = 500{,}000$. Figure 5 presents the long-term profit. Figure 5(a) shows that if a seller aims for three stars in the ramp-up process and survives the ramp-up process, our algorithm can improve the long-term profit by at least 15%. As α increases, the long-term profit improvement ratio increases. This means that if a seller is more aggressive in reducing the ramp-up time, the more long-term profits he can earn. However, this also requires a seller to have a larger investing budget to survive the ramp-up process, since he sacrifice more short-term profits. Consider $\alpha = 0.6$, our algorithm can improve the long-term profit by at least 20%. In practice, sellers are more likely



(a) Three stars (b) Five stars

Fig. 5. Improvement in long-term profits.

to set a medium value for α because sellers usually have medium investing budgets and the short-term profit is important to sustain their online store business.

8 RELATED WORK

Reputation systems [23] serve as a core building block for E-commerce systems. Three fundamental aspects have been extensively studied: (1) *reputation management*, (2) *attacks and defenses*, and (3) *efficiency*. On widely practiced reputation management approach is the feedback-based model [16, 25, 31, 34] that consists of a feedback system [16, 25] and a rating aggregation policy. The transitive trust model [13, 18, 24, 33] captures the propagation of trust via graphs. Graph algorithms were proposed to compute the reputation score [18, 24, 33, 35]. In [14], the authors proposed to use intertemporal discounting factors to measure the trustworthiness of users. Dishonest feedbacks are a typical attack to reputation systems and peer-prediction mechanisms were designed [21] to address it. Reputation inflation (or fake feedbacks) is another possible attack to reputation systems. A number of algorithms were proposed to resolve this attack [7, 15, 33]. Several works investigated efficiency issues in E-commerce systems. In [6], the authors studied how the leniency or criticality of buyers in assigning ratings may influence sellers' advertising behavior. Our work investigates the "*ramp-up problem*" in eBay like reputations systems, which was first studied in [27]. It uncovered that the ramp-up time of the eBay reputation system is around 800 days. Our work is different in that we formulate a data-driven problem on how to infer the optimal discount to quickly ramp up the reputation. We like to mention that the two pages extended abstract [30] contains some preliminary results of this article.

MAB framework addresses the tradeoff between exploration and exploitation in sequential decision makings. Three fundamental frameworks are *stochastic bandits, adversarial bandits,* and *Markovian bandits*. Stochastic bandits have four elements: *arms, rewards, a forecaster,* and the *number of rounds* to be played. When one pulls an arm, a reward is generated via a probability distribution with unknown parameters (e.g., mean, variance). The forecaster plays a finite number of rounds (one arm per round). The objective is to maximize the expected total rewards. One representative algorithm for the stochastic bandit is the UCB algorithm [3]. In adversarial bandits, an adversary sets a reward for each arm. Exponential weights for Exploration and Exploitation (Exp3) algorithms [5] are a family of playing strategies for adversarial bandits. In Markovian bandits, arms are associated with states and state transition matrices [11]. Gittins Indices [11, 12] are a family of playing strategies for Markovian bandits. This article applies a stochastic bandit framework to E-commerce systems, i.e., develop mechanisms to trade price discounts for reputation.

A variety of work studied the reputation effect on the product price. Landon et al. [19] investigated the impact of product reputation on the price using data from the wine market. Ba et al. [4] conducted online experiments and empirical studies on the data from an online auction market to show that reputation can generate price premiums for sellers. Jin et al. [17] studied how the reputation of a seller influences the product price in internet auctions.

9 CONCLUSION

This article develops an online algorithmic framework to enhance seller reputation (in E-commerce systems) via inferring optimal price discounts from the transaction data. We formulate an optimization framework to select the optimal discount rate which explores the tradeoff between the ramp-up time and the short-term profit. We derive the closed-form optimal discount rate and apply a stochastic bandit framework to infer it from a seller's historical transaction data with regret upper bounds. We show that the computational complexity of evaluating the performance metrics like the ramp-up time are infeasibly high. We develop efficient randomized algorithms to approximate them with guaranteed performance. Finally, we conduct experiments on an eBay's dataset. Experimental results show that our online discount algorithm can trade 60% of the short-term profit for reducing the ramp-up time by 40%. Furthermore, this reduction in the ramp-up time can increase a seller's long-term profits by at least 20%.

APPENDIX

With some basic probability and combinatorial arguments we can derive the pmf of $N(s)$ as follows.

Lemma 1. *For all $j > n_s$, we express the pmf for $N(s)$ as*

$$\mathbb{P}[N(s) = j] = \sum_{N^-=0}^{\lfloor (j-n_s)/2 \rfloor} \sum_{\ell_1=1}^{n_s} \sum_{\ell_1+1}^{n_s+2} \cdots \sum_{\ell_{k-1}+1}^{n_s+2k-2} \cdots \sum_{\ell_{N^- -1}+1}^{n_s+2N^- -2} \binom{j-1}{j-n_s-2N^- -1} (P^+)^{n_s+N^-} (P^0)^{j-n_s-2N^-} (P^-)^{N^-}, \quad (22)$$

Proof. Given the number of positive ratings N^+, neutral ratings N^0 and negative ratings N^-, the probability that these ratings take these signs is $(P^+)^{N^+}(P^0)^{N^0}(P^-)^{N^-}$. A configuration is valid if the ramp-up condition is satisfied by the $(N^+ + N^0 + N^-)$-th rating. Now, we compute the number of valid configurations. Let ℓ_i, where $i \leq N^-$, denote the position of the first negative rating. This means that the ith negative rating is the ℓ_i-th rating. Consider the first negative rating. It must occurred before the n_s-th positive rating, otherwise the configuration will be invalid. Hence, its position ranges from the first rating to the n_s-th rating. Consider the ith negative rating. It must occurred before the $(n_s + i - 1)$-th positive rating. Hence, its position ranges from the $(\ell_{i-1} + 1)$-th rating to the $(n_s + 2i - 2)$-th rating. Therefore, the total number of valid configurations can be expressed as $\sum_{\ell_1=1}^{n_s} \sum_{\ell_1+1}^{n_s+2} \cdots \sum_{\ell_{i-1}+1}^{n_s+2i-2} \cdots$
$\sum_{\ell_{N^- -1}+1}^{n_s+2N^- -2} 1$. Now, we determine the position of neutral ratings. We have a total number N^0 neutral ratings to be placed among a sequence of $N^+ + N^-$ ratings. Note that the position of the neutral ratings does not change the validity of a configuration. For each valid configuration, there

are $\binom{N^++N^0+N^--1}{N^0-1}$ ways to place the neutral ratings. Hence, the total probability is

$$\mathbb{P}[N(s)=i] = \sum_{N^-=0}^{\lfloor (j-n_s)/2\rfloor} \sum_{\ell_1=1}^{n_s} \sum_{\ell_1+1}^{n_s+2} \cdots \sum_{\ell_{j-1}+1}^{n_s+2j-2} \cdots \sum_{\ell_{N^--1}+1}^{n_s+2N^--2} \binom{N^++N^0+N^--1}{N^0-1} (P^+)^{N^+}(P^0)^{N^0}(P^-)^{N^-}.$$

We conclude this lemma by observing that $N^+ + N^0 + N^- = i$ and $N^+ = n_s + N^-$, yielding $N^0 = i - n_s - 2N^-$. □

Proof of Theorem 4.3. Recall that in the bandit interpretation, the reward corresponds to level m discount is $-\alpha\lambda_0 P_0 w - (1-\alpha)d_m p/u$, where w follows an exponential distribution with parameter $\lambda_0 P_m$. Hence, $\alpha\lambda_0 P_0 w$ follows an exponential distribution with parameter $\alpha P_0/P_m$. We employ large deviation theory to complete this proof.

Lemma 2. *Let $X_1,\ldots,X_n$ denote n independent and identically distributed (IID) exponential random variables with parameter λ. Let $\overline{X}_n = \sum_{i=1}^n X_i/n$. Then: 1) $\forall \gamma_1 > \frac{1}{\lambda}, \mathbb{P}[\overline{X}_n \geq \gamma_1] \approx \exp\{n(\lambda\gamma_1 - 1 - \ln(\lambda\gamma_1))\}$; 2) $\forall \gamma_2 < \frac{1}{\lambda}, \mathbb{P}[\overline{X}_n \leq \gamma_2] \approx \exp\{n(\lambda\gamma_2 - 1 - \ln(\lambda\gamma_2))\}$.*

Also note that for any $x \geq 0$, the following inequality holds $x - \ln(1+x) \geq \frac{3x^2}{6+4x}$, and for all $x \in [0,1]$ the following inequality holds $-x - \ln(1-x) \geq \frac{3x^2}{6-4x}$. With these observations and applying the proving framework developed in [3], we complete this proof. □

Proof of Theorem 5.1. The expected ramp-up time can be expressed as $\mathbb{E}[T(s)] = \sum_{j=1}^{n_s} 1/(\lambda_{\mathcal{S}(r_j)}P_0)$. The expected short-term profit can be expressed as $\mathbb{E}[G(s)] = n_s u$. With some basic algebraic arguments, we have

$$z(s,\mathcal{M}^*) = \sum_{i=1}^{n_s}\left(-\alpha \frac{1}{\sum_{j=1}^{n_s} 1/(\lambda_{\mathcal{S}(r_i)}P_0)} \frac{1}{\lambda_{\mathcal{S}(r_i)}P_{m_i^*}} - (1-\alpha)\frac{p}{n_s u} d_{m_i^*}\right).$$

Similarly,

$$z(s,\mathcal{M}^*_{of}) = \sum_{i=1}^{n_s}\left(-\alpha \frac{1}{\sum_{j=1}^{n_s} 1/(\lambda_{\mathcal{S}(r_j)}P_0)} \frac{1}{\lambda_{\mathcal{S}(r_i)}P_{m^*_{sub}}} - (1-\alpha)\frac{p}{n_s u} d_{m^*_{sub}}\right).$$

Computing the difference we complete this proof. □

Proof of Theorem 6.1. First, one can easily obtain that

$$|z(s,M^*_{of}) - z(s,\mathcal{M}^*)| = \sum_{j=n_s}^{\infty} \mathbb{P}[N(s)=j]\mathbb{E}[|z(s,M^*_{of}) - z(s,\mathcal{M}^*)||N(s)=j].$$

Note that

$$\mathbb{E}[|z(s,M^*_{of}) - z(s,\mathcal{M}^*)||N(s)=j] = \sum_{i=1}^{j} \mathbb{E}\left[\frac{1-\alpha}{\mathbb{E}[G(s)]} p(d_{m^*_{sub}} - d_{m_i^*}) + \frac{\alpha}{\mathbb{E}[T(s)]}\left(\frac{1}{\lambda_{\mathcal{S}(r_i)}P_{m^*_{sub}}} - \frac{1}{\lambda_{\mathcal{S}(r_i)}P_{m_i^*}}\right) |N(s)=j\right].$$

By the linearity of expectation, we conclude. □

Proof of Theorem 6.2. Observe that

$$|z(s, \widehat{\mathcal{M}}) - z(s, M^*_{of})| = \sum_{j=n_s}^{\infty} \mathbb{P}[N(s) = j]\mathbb{E}[|z(s, \widehat{\mathcal{M}}) - z(s, M^*_{of})||N(s) = j].$$

We complete this prof by deriving $\mathbb{E}[|z(s, \widehat{\mathcal{M}}) - z(s, M^*_{of})||N(s) = j]$ with a similar approach as Theorem 4.3. □

Proof of Lemma 6.5. Note that each product earns a profit of u. We then have $G^\infty = \mathbb{E}[\sum_{i=1}^{\infty} u\delta^{t_i}] = \sum_{i=1}^{\infty} u \sum_{(r_1,\ldots,r_i)} \mathbb{P}[r_1, \ldots, r_i]\mathbb{E}[\delta^{t_i}|r_1, \ldots, r_i]$. Observe

$$\mathbb{E}[\delta^{t_i}|r_1, \ldots, r_i] = \prod_{j=1}^{i} \mathbb{E}[\delta^{w_i}|r_1, \ldots, r_i] = \prod_{j=1}^{i} \frac{\lambda_{\mathcal{S}(r_j)}P_0}{\lambda_{\mathcal{S}(r_j)}P_0 - \ln \delta}.$$

Also observe that $\mathbb{P}[r_1, \ldots, r_i] = \prod_{j=2}^{i}(P^- \mathbf{I}_{\{r_j = r_{j-1}-1\}} + P^0 \mathbf{I}_{\{r_j = r_{j-1}\}} + P^+ \mathbf{I}_{\{r_j = r_{j-1}+1\}})$. This completes the proof. □

Proof of Theorem 6.7. The expected computational complexity for Algorithm 4 is $O(K\mathbb{E}[N(s)])$. Let R_i denote the ith rating. Observe that

$$\mathbb{E}[N(s)] = \sum_{i=1}^{\infty} \mathbb{P}[N(s) \geq i] \leq n_s - 1 + \sum_{i=n_s}^{\infty} \mathbb{P}\left[\sum_{j=1}^{i} R_j < n_s\right].$$

Now consider $\ell \leq \frac{2n_s}{P^+ - P^-}$. We have $\sum_{i=n_s}^{\ell} \mathbb{P}[\sum_{j=1}^{N(s)} R_j < n_s] \leq \ell - n_s + 1$. Consider $\ell > \frac{2n_s}{P^+ - P^-}$, applying Hoeffding's inequality [22] we have

$$\begin{aligned}\mathbb{P}\left[\sum_{j=1}^{\ell} R_j < n_s\right] &= \mathbb{P}\left[\sum_{j=1}^{\ell} R_j < \ell(P^+ - P^-) - \frac{\ell}{2}(P^+ - P^-)\right] \\ &\leq \exp\left(-2\frac{\ell^2}{4}\frac{(P^+ - P^-)^2}{4\ell}\right) = \exp\left(-\frac{\ell}{8}(P^+ - P^-)^2\right).\end{aligned}$$

Hence, $\sum_{i=\ell}^{\infty} \mathbb{P}\left[\sum_{j=1}^{i} R_j < n_s\right] \leq O((P^+ - P^-)^{-2})$. So, $\mathbb{E}[N(s)] = O((P^+ - P^-)^{-2} + \frac{n_s}{P^+ - P^-})$.

We now derive the minimum simulation rounds needed to guarantee an accurate estimation. Observe that $\mathbb{E}[T_i] = \mathbb{E}[T(s)]$, and $\mathrm{Var}[T_i] = \mathrm{Var}[T(s)]$. Note that T_i are IID random variables. Applying Chebyshev's inequality [22], we have

$$\begin{aligned}&\mathbb{P}[|\widehat{\mathbb{E}}[T(s)] - \mathbb{E}[T(s)]| \geq \epsilon\mathbb{E}[T(s)]] \\ &\quad \leq \frac{\mathrm{Var}[\sum_{i=1}^{K} T_i]}{\epsilon^2 K^2 \mathbb{E}[T(s)]^2} = \frac{\mathrm{Var}[T(s)]}{\epsilon^2 K \mathbb{E}[T(s)]^2} = \frac{1}{\epsilon^2 K}\frac{\mathrm{Var}[T(s)]}{\mathbb{E}[T(s)]^2}.\end{aligned}$$

We derive an upper bound for $\mathrm{Var}[T(s)]$ as

$$\begin{aligned}\mathrm{Var}[T(s)] &= \sum_{i=n_s}^{\infty} \mathbb{P}[N(s) = i]\mathbb{E}[\mathrm{Var}[T(s)]|N(s) = i] \\ &\leq \frac{1}{\lambda_0 P_0} \sum_{i=n_s}^{\infty} \mathbb{P}[N(s) = i] \sum_{j=1}^{i} \frac{1}{\lambda_{\mathcal{S}(r_j)} P_{m_j}},\end{aligned}$$

where the last step follows $\frac{1}{\lambda^2_{S(r_j)} P^2_{m_j}} \leq \frac{1}{\lambda_0 P_0} \frac{1}{\lambda_{S(r_j)} P_{m_j}}$. We express $\mathbb{E}[T(s)]$ as $\mathbb{E}[T(s)] = \sum_{i=n_s}^{\infty} \mathbb{P}[N(s) = i] \sum_{j=1}^{i} \frac{1}{\lambda_{S(r_j)} P_{m_j}}$. Then, it follows that

$$\frac{\mathrm{Var}[T(s)]}{\mathbb{E}[T(s)]^2} \leq \frac{1}{\lambda_0 P_0} / \left(\sum_{i=n_s}^{\infty} \mathbb{P}[N(s) = i] \sum_{j=1}^{i} \frac{1}{\lambda_{S(r_j)} P_{m_j}} \right)$$
$$\leq \frac{\lambda_{s-1} P_M}{\lambda_0 P_0} \frac{1}{\mathbb{E}[N(s)]} \leq \frac{\lambda_{s-1} P_M}{\lambda_0 P_0} \frac{1}{n_s}.$$

To make $\frac{1}{\epsilon^2 K} \frac{\mathrm{Var}[T(s)]}{\mathbb{E}[T(s)]^2} \leq \xi$, we only need $K \geq \frac{1}{\epsilon^2} \frac{\lambda_{s-1} P_M}{\lambda_0 P_0} \frac{1}{n_s}$. This completes the proof. □

Proof of Theorem 6.8. Let $G_I = \sum_{i=1}^{I} u_i \delta^{t_i}$. Then the approximating error $|\widehat{G}^\infty - G^\infty| \leq |\widehat{G}^\infty - \mathbb{E}[G_I]| + |\mathbb{E}[G_I] - G^\infty|$. We next derive the minimum K to guarantee $|\widehat{G}^\infty - \mathbb{E}[G_I]| \leq \epsilon/2$ and the minimum I to guarantee $|\mathbb{E}[G_I] - G^\infty| \leq \epsilon/2$.

Let us derive the minimum K to guarantee $|\widehat{G}^\infty - \mathbb{E}[G_I]| \leq \epsilon/2$ first. Observe that G_i are IID random variables having mean $\mathbb{E}[G_i] = \mathbb{E}[G_I]$ and variance $\mathrm{Var}[G_i] = \mathrm{Var}[G_I]$. Applying Chebyshev's inequality [22], we have

$$\mathbb{P}\left[|\widehat{G} - G_I| \geq \frac{1}{2}\epsilon\right] = \mathbb{P}\left[|\sum_{i=1}^{K} G_i - KG_I| \geq \frac{1}{2}\epsilon K\right]$$
$$\leq \frac{4\mathrm{Var}[\sum_{i=1}^{K} G_i]}{\epsilon^2 K^2} = \frac{4\mathrm{Var}[G_I]}{\epsilon^2 K}.$$

We next derive an upper bound for $\mathrm{Var}[G_I]$. To guarantee $\frac{4\mathrm{Var}[G_I]}{\epsilon^2 K} \leq \xi$, we only need $K \geq \frac{4\mathrm{Var}[G_I]}{\epsilon^2 \xi}$. We next derive an upper bound for $\mathrm{Var}[G_I]$. First applying the Cauchy–Schwarz inequality, we bound $\mathrm{Var}[G_I]$ as

$$\mathrm{Var}\,[G_I] = \mathbb{E}\left[\left(\sum_{i=1}^{M} u_i \delta^{t_i} - \sum_{i=1}^{M} u_i \mathbb{E}[\delta^{t_i}]\right)^2\right]$$
$$= \sum_{i=0}^{M} \sum_{j=0}^{M} \mathbb{E}\left[(u_i \delta^{t_i} - \mathbb{E}[u_i \delta^{t_i}])(u_j \delta^{t_j} - \mathbb{E}[u_j \delta^{t_i}])\right]$$
$$\leq \sum_{i=0}^{M} \sum_{j=0}^{M} \sqrt{\mathrm{Var}[u_i \delta^{t_i}] \mathrm{Var}[u_j \delta^{t_j}]}.$$

Now, let us derive an upper bound for the variance $\mathrm{Var}[u_i \delta^{t_i}]$. With some basic probability arguments, we have

$$\mathrm{Var}[u_i \delta^{t_i}] = \mathrm{Var}[u_i \delta^{t_{i-1} + w_i}] = \mathbb{E}[\mathrm{Var}[u_i \delta^{t_{i-1} + w_i}] | t_{i-1}]$$
$$= \mathbb{E}[\delta^{2t_{i-1}} \mathrm{Var}[u_i \delta^{w_i}]].$$

For the ease of presentation, denote $y_i = \frac{\ln \delta^{-1}}{\lambda_{S(r_i)} P_{m_i}}$. We can then bound $\mathrm{Var}[u_i \delta^{w_i}]$ as

$$\mathrm{Var}[u_i \delta^{w_i}] \leq u^2 \mathbb{E}[\delta^{2w_i}] - u^2 (\mathbb{E}[\delta^{w_i}])^2 = \frac{u^2}{1 + 2y_i} - \frac{u^2}{1 + 2y_i + y_i^2}$$
$$= \frac{u^2 y_i^2}{(1 + 2y_i)(1 + 2y_i + y_i^2)} \leq u^2 y_i^2 \leq u^2 \left(\frac{\ln \delta^{-1}}{\lambda_0 P_0}\right)^2.$$

Furthermore, we bound $\mathbb{E}[\delta^{2t_{i-1}}]$ as

$$\mathbb{E}[\delta^{2t_{i-1}}] \le \mathbb{E}[\delta^{2(w_1+\ldots+w_{i-1})}] = \prod_{j=1}^{i-1} \mathbb{E}[\delta^{2w_j}]$$
$$= \prod_{j=1}^{i-1} \left(1 + \frac{2\ln\delta^{-1}}{\lambda_{\mathcal{S}(r_j)} P_{m_j}}\right)^{-1}$$
$$\le \left(1 + \frac{2\ln\delta^{-1}}{\lambda_{\max}}\right)^{-(i-1)},$$

where $\lambda_{\max} = \max\{\lambda_s P_M, \lambda_S\}$. We then have

$$\mathrm{Var}[u_i \delta^{t_i}] \le u^2 \left(1 + \frac{2\ln\delta^{-1}}{\lambda_{\max}}\right)^{-(i-1)} \left(\frac{\ln\delta^{-1}}{\lambda_0 P_0}\right)^2.$$

For the ease of presentation, denote $x = 1 + \frac{2\ln\delta^{-1}}{\lambda_{\max}}$. We can then bound $\mathrm{Var}[G_I]$ as follows:

$$\mathrm{Var}[G_I] \le u^2 \left(\frac{\ln\delta^{-1}}{\lambda_0 P_0}\right)^2 \sum_{i=1}^{M}\sum_{j=1}^{M} \sqrt{x^{-(i+j-2)}}$$
$$\le u^2 \left(\frac{\ln\delta^{-1}}{\lambda_0 P_0}\right)^2 \frac{(1-(x^{-0.5})^M)^2}{(1-x^{-0.5})^2}$$
$$\le u^2 \left(\frac{\ln\delta^{-1}}{\lambda_0 P_0}\right)^2 \frac{1}{(1-x^{-0.5})^2}$$
$$= u^2 \left(\frac{\ln\delta^{-1}}{\lambda_0 P_0}\right)^2 x(\sqrt{x}+1)^2 \left(\frac{\lambda_{\max}}{2\ln\delta^{-1}}\right)^2$$
$$\le u^2 \left(1 + \frac{2\ln\delta^{-1}}{\lambda_{\max}}\right)\left(2 + \frac{\ln\delta^{-1}}{\lambda_{\max}}\right)^2 \left(\frac{\lambda_{\max}}{2\lambda_0 P_0}\right)^2,$$

where the last step follows $x < (1 + \frac{\ln\delta^{-1}}{\lambda_{\max}})^2$. We can then conclude the low bound of K. In fact, $\frac{\ln\delta^{-1}}{\lambda_{\max}}$ can be treated as a constant. Hence, $K \ge \Theta(\frac{1}{\epsilon^2\xi}(\frac{\max\{\lambda_s P_M, \lambda_S\}}{\lambda_0 P_0})^2)$.

Now, we derive the minimum I to guarantee $|\mathbb{E}[G_I] - G^\infty| \le \epsilon/2$. Observe that

$$|\mathbb{E}[G_I] - G^\infty| = \left|\mathbb{E}\left[\sum_{i=I+1}^{\infty} u_i \delta^{t_i}\right]\right| \le \mathbb{E}\left[\sum_{i=I+1}^{\infty} u\delta^{t_i}\right]$$
$$= \sum_{i=I+1}^{\infty} u\mathbb{E}[\delta^{w_1+\ldots+w_i}] = \sum_{i=I+1}^{\infty} u \prod_{j=1}^{i} \mathbb{E}[\delta^{w_i}]$$
$$= \sum_{i=I+1}^{\infty} u \prod_{j=1}^{i} \left(1 + \frac{\ln\delta^{-1}}{\lambda_{\mathcal{S}(r_j)} P_{m_j}}\right)^{-1}$$
$$\le \sum_{i=I+1}^{\infty} u \left(1 + \frac{\ln\delta^{-1}}{\lambda_{\max}}\right)^{-i}$$
$$\le u\left(1 + \frac{\ln\delta^{-1}}{\lambda_{\max}}\right)^{-I-1}\left(1 + \frac{\lambda_{\max}}{\ln\delta^{-1}}\right).$$

Hence, we need M satisfies

$$u\left(1+\frac{\ln\delta^{-1}}{\lambda_{\max}}\right)^{-I-1}\left(1+\frac{\lambda_{\max}}{\ln\delta^{-1}}\right)\le\frac{\epsilon}{2},$$

which yields

$$I>\frac{\ln\frac{\epsilon}{2}-\ln u-\ln\left(1+\lambda_{\max}/\ln\delta^{-1}\right)}{\ln(1+\ln\delta^{-1}/\lambda_{\max})}-1.$$

This completes the proof. □

REFERENCES

[1] Alibaba. 1999. Online Shopping Website. Retrieved from http://www.alibaba.com/.

[2] Amazon. 1994. Online Shopping Website. Retrieved from http://www.amazon.com/.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47, 2–3 (2002), 235–256.

[4] Sulin Ba and Paul A. Pavlou. 2002. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS Quarterly* 26, 3 (2002), 243–268.

[5] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning* 5, 1 (2012), 1–122.

[6] Chrysanthos Dellarocas. 2001. Analyzing the economic efficiency of eBay-like online reputation reporting mechanisms. In *Proc. of ACM EC.*

[7] Prashant Dewan and Partha Dasgupta. 2010. P2P reputation management using distributed identities and decentralized recommendation chains. *IEEE TKDE* 22, 7 (2010), 1000–1013.

[8] eBay. 1995. Homepage. Retrieved from http://www.ebay.com/.

[9] eBay. 1995. eBay Classifies Sellers into Twelve Stars. Retrieved from http://pages.ebay.com/help/feedback/scores-reputation.html.

[10] Fortune 500. 2015. Fortune Ranking. Retrieved from http://fortune.com/fortune500/.

[11] John Gittins, Kevin Glazebrook, and Richard Weber. 2011. *Multi-armed Bandit Allocation Indices.* John Wiley & Sons.

[12] John C. Gittins. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B* (1979), 148–177.

[13] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. 2004. Propagation of trust and distrust. In *Proc. of WWW.* 403–412.

[14] Christopher J. Hazard and Munindar P. Singh. 2011. Intertemporal discount factors as a measure of trustworthiness in electronic commerce. *IEEE TKDE* 23, 5 (2011), 699–712.

[15] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. 2009. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys* 42, 1 (December 2009), Article no. 1, 31 pages.

[16] Daniel Houser and John Wooders. 2006. Reputation in auctions: Theory, and evidence from eBay. *Journal of Economics & Management Strategy* 15, 2 (2006), 353–369.

[17] Ginger Zhe Jin and Andrew Kato. 2006. Price, quality, and reputation: Evidence from an online field experiment. *The RAND Journal of Economics* 37, 4 (2006), 983–1005.

[18] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. 2003. The eigentrust algorithm for reputation management in P2P networks. In *Proc. of WWW.*

[19] Stuart Landon and Constance E. Smith. 1998. Quality expectations, reputation, and price. *Southern Economic Journal* 64, 3 (1998), 628–647.

[20] Walter Loeb. 2014. 10 Reasons Why Alibaba Blows Away Amazon and eBay. Retrieved from http://www.forbes.com/sites/walterloeb/2014/04/11/10-reasons-why-alibaba-is-a-worldwide-leader-in-e-commerce/.

[21] Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51, 9 (September 2005), 1359–1373.

[22] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing.* Cambridge University Press.

[23] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. 2000. Reputation systems. *Communications of the ACM* 43, 12 (December 2000), 45–48.

[24] Paul Resnick and Rahul Sami. 2009. Sybilproof transitive trust protocols. In *Proc. of ACM EC.*

[25] Aameek Singh and Ling Liu. 2003. TrustMe: Anonymous management of trust relationships in decentralized P2P systems. In *Proc. of P2P.*

[26] Taobao. 2003. Online Shopping Website. Retrieved from http://www.taobao.com/.

[27] Hong Xie and John C. S. Lui. 2015. A data driven approach to uncover deficiencies in online reputation systems. In *Proc. of IEEE ICDM*.
[28] Hong Xie and John C. S. Lui. 2017. Mining deficiencies of online reputation systems: Methodologies, experiments and implications. *IEEE Transactions on Services Computing* (2017). DOI: http://dx.doi.org/10.1109/TSC.2017.2730206
[29] Hong Xie and John C. S. Lui. 2015. Modeling eBay-like reputation systems: Analysis, characterization and insurance mechanism design. *Performance Evaluation* 91 (2015), 132–149.
[30] Hong Xie, Richard T. B. Ma, and John C. S. Lui. 2016. Trading discount for reputation?: On the design and analysis of E-commerce discount mechanisms. In *Proc. of ACM SIGMETRICS*.
[31] Li Xiong and Ling Liu. 2004. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE TKDE* 16, 7 (2004), 843–857.
[32] Haitao Xu, Daiping Liu, Haining Wang, and Angelos Stavrou. 2015. E-commerce reputation manipulation: The emergence of reputation-escalation-as-a-service. In *Proc. of WWW*.
[33] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. 2006. SybilGuard: Defending against sybil attacks via social networks. In *Proc. of ACM SIGCOMM*.
[34] Xiuzhen Zhang, Lishan Cui, and Yan Wang. 2014. Commtrust: Computing multi-dimensional trust by mining e-commerce feedback comments. *IEEE TKDE* 26, 7 (2014), 1631–1643.
[35] Runfang Zhou, Kai Hwang, and Min Cai. 2008. Gossiptrust for fast reputation aggregation in peer-to-peer networks. *IEEE TKDE* 20, 9 (2008), 1282–1295.

Received March 2017; revised September 2017; accepted September 2017