To Be or Not To Be: Analyzing & Modeling Social Recommendation in Online Social Networks

Technical Report

Abstract—**Firms are now considering to offer rewards to customers who recommend the firms' products/services in online social networks (OSN). However, the pros and cons of such social recommendation scheme are still unclear. Thus, it is difficult for firms to design rewarding schemes, and for OSN platforms to design regulating policies. Via empirical analysis of data, we first identify key factors that affect the spreading of a firm's product in OSNs. These findings enable us to develop an accurate (i.e., with a high validation accuracy) mathematical model on social recommendations. In particular, our model captures how users decide whether to recommend an item, which is a key factor but often ignored by previous social recommendation models such as the "Independent Cascade model". We also design algorithms to infer model parameters. Using these parameters in our model, we uncover conditions when social recommendation improves a firm's profit and users' utilities, as well as when it cannot improve the profit or hurts users' utilities. These conditions help the design of both rewarding schemes and regulating policies. Finally, we extend our model to an online setting and design reinforcement learning algorithms for a firm to dynamically optimize its rewarding schemes to improve its profit.**

I. Introduction

With the prevalence of online social networks (OSNs) such as Facebook and Twitter, products can reach a large number of customers via friend-to-friend recommendations (called social recommendations). To encourage social recommendations, a number of firms are providing non-monetary or even monetary reward. For example, Dropbox gives extra storage space to users who recommend friends to use Dropbox [11]. Some businesses in Yelp, e.g., restaurants, provide gifts to customers who do *"check-in"* in the social network [26]. DiDi gives discount to users who refer friends to use its app [23].

We consider an incentivized social recommendation scheme, in which a user gets rewards from firms by recommending firms' products/services to friends. To illustrate the tradeoff for firms and the platform, consider the following three examples:

Example 1 (Baseline). *Consider the baseline case where the firm gaves no reward to recommenders, i.e., scenario 1 in Fig. 1. Three users form a social network as a line-graph. The production cost of a product is* \$3 *and the firm sets a price of* \$7*. The firm used a traditional advertisement (Adv.) in which only user 1 was informed. She purchased the product. But she would not recommend it to her friends in the OSN because there was no reward. User 2 and 3 would not be informed of the product. The firm's total profit was* (\$7−\$3)+\$0+\$0=\$4.

Example 2 (Benefits of social recommendation). *Consider social recommendation with rewards, i.e., scenario 2 in Fig. 1. The firm offered a reward of* \$4 *to users who recommend the product to their friends. The firm increased the price from* \$7 *to* \$9 *to compensate the cost of reward. With such reward, user 1 would recommend the product to user 2, so user 2 knew the product from* "friends' recommendations" *(Rec.). User 2 purchased it but decided not to recommend it, and user 3 was not informed of the product. The firm's total profit was* (\$9−\$3−\$4)+(\$9−\$3)+\$0=\$8*, which was higher than the profit of* \$4 *in scenario 1 without rewards to recommenders.*

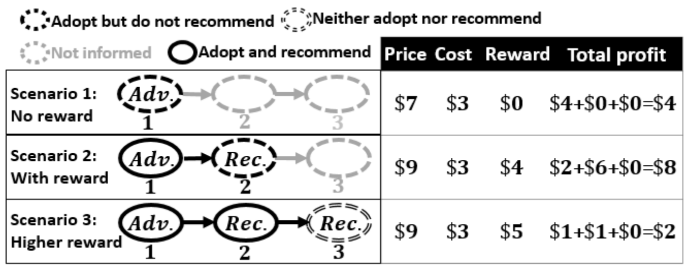


Fig. 1. Profitability and tradeoffs of social recommendation

Example 2 highlights that the firm can *improve* the profit by rewarding social recommendations. However, such scheme has certain risks as illustrated in the following example.

Example 3 (Risks of social recommendation). *Consider the scenario 3 in Fig. 1, where the firm now increases the reward to* \$5*. Then, both user 1 and user 2 are willing to recommend the product. User 3 thinks the price is too high and decides not to buy it. Therefore, the firms's total profit is* \$2*, which is smaller than the profit, i.e.,* \$4*, of not rewarding social recommendation. Worse yet, user 3 could be annoyed as she receives an uninteresting recommendation.*

Example 3 highlights that social recommendation may lead to profit loss to the firm. In addition, some users may receive uninteresting recommendations, threatening OSNs' eco-systems.

The above examples motivate us to answer the following important questions: *Should a firm use such scheme? For social recommendation in OSNs, what are the key factors that affect firms' profit gains and users' utilities? How much reward/price should a firm set so to increase profit?* It is challenging to answer these questions. First, real-world social network consists of millions of users and the topology is more complicated than the line graph in Fig. 1. Second, when the price and reward changes, firms do not know consumers' behaviors and the potential profit gains. Third, the OSN platform does not know users' utilities on the recommended items. We address these questions and our contributions are:

- **Novel model with data:** With large datasets of WeChat and Yelp, we conduct an in-depth empirical analysis of the spreading pattern of social recommendations, and we uncover number of key factors, e.g. users' recommendation rate. These findings enable us to develop an accurate mathematical model on social recommendations. In particular, our model captures how users decide whether to recommend an item, which is often ignored by previous social recommendation models such as the "Independent Cascade model" [14]. Compared with Independent Cascade model, we reduce the error to predict the number of recommenders for WeChat's items by more than 80%.

- **Novel findings:** By both theoretical analysis and trace-driven simulations, we reveal the pros and cons of social recommendations: (1) a firm can improve profits from social recommendations when its item is not well-known, or users' recommending rate is near a *"critical value"*, etc.; (2) although social recommendation improves the utilities of recommenders, it often hurts non-recommenders' utilities especially for OSNs whose users have weak ties with friends (e.g. Twitter), etc.
- **Algorithm to improve firms' profits:** We design an efficient algorithm to estimate a firm's profit, and it is more than 6000 times faster than agent-based simulations [9], making it easier to search for a firm's optimal strategy. We also extend our model to an online setting and design reinforcement learning algorithms for a firm to dynamically optimize rewarding schemes. Compared with influence maximization algorithms [24] that are not explicitly aware of firms' profits, our algorithm improves a firm's profit by as high as 71%.

II. Implications from Real-World Social Recommendation Data

In this section, we analyze real-world datasets from WeChat and Yelp. Through this we find that the users' recommending rate is critical to the spreading of a firm's product in OSN. We also observe various factors, e.g., reward, information source, product price, etc., which significantly impact users' decisions on whether to recommend an item. These observations help us to build an accurate mathematical model in Section III.

A. *Implications From WeChat Data*

The WeChat data. WeChat is an OSN in China with more than one billion monthly active users [3]. Our WeChat dataset is anonymized and contains aggregated statistics about many social recommendation campaigns within one week in Sep. 2018. There are three types of rewards for the recommenders: reward the customers (1) who *"shares"* an item to his friends for a certain number of times; or (2) whose friends *"click"* the item shared by him; or (3) who successfully invites a *"new user"* to adopt an item. In addition, a user can get *"discounts"* when he buys an item via a friend's referral link. Also, an item can have no reward and no discount, abbreviated as *"no-reward"*. Therefore, an item has one of the five marketing activities: *share, click, new-user, discount* and *no-reward*. We collected statistics in the social recommendation campaigns to quantify rewards and discounts, which are normalized to $[0, 1]$.

Table I presents an example of data for a firm with multiple products (items). For item 1, a recommender can get a reward

Table I
Statistics of social recommendation campaigns (in WeChat)

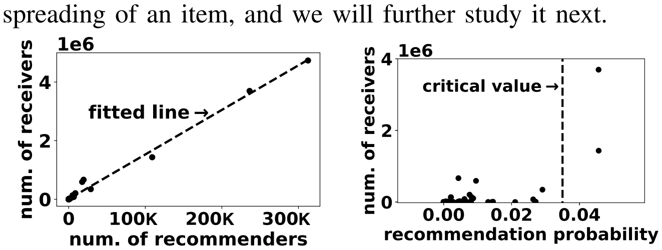
Item ID	marketing activity	re-ward	disc-ount	num. reco-mmenders	recomm-end rate	click rate
1	*share*	**1**	0	**236,191**	**4.56%**	**2.1%**
2	*share*	1	0	109,023	4.56%	1.3%
3	*click*	0.05	0	6,185	2.64%	1.2%
4	*new-user*	1	0	17,599	0.95%	4.0%
5	*new-user*	0.5	0	6,828	0.82%	2.3%
6	*discount*	0	0.1	1,675	0.19%	0.9%
7	*discount*	0	1	28,848	2.91%	0.8%
8	*no-reward*	0	0	8,677	0.76%	6.4%
9	*no-reward*	0	0	317	0.54%	4.5%

of 1, and 236,191 users recommend this item. Among the users who obtain *"at least one"* recommendations of item 1, 4.56% of them further recommend the item to their friends. Moreover, 2.1% of the users who receive a recommendation of item 1 would click the item. Note that a user can receive a recommendation of item 1 for multiple times. This is why we see 4.56%>2.1%. For each item, our dataset contains the set of recommenders (i.e., anonymized users who recommended the item) and the set of receivers (i.e., anonymized users who received friends' recommendations of the item). These ***25.14 million anonymous users*** form an undirected social network. Due to privacy restrictions, we only use aggregated statistics such as the degree distribution to create the social network.

• **Data analysis 1: uncover the spreading pattern of an item.** We study the spreading pattern of an item by investigating the number of receivers that obtain friends' recommendations about the item. Fig. 2(a) shows the impact of the number of recommenders on the number of receivers, where each point corresponds to one item. One can observe that the number of receivers increases *nearly linearly* in the number of recommenders with a slop of 15.6. Fig. 2(b) further shows how the number of receivers is imapcted by the users' recommending probability (or rate), which is defined as

$$\textit{recommendation probability} \triangleq \frac{\text{num. of receivers who recommend it}}{\text{num. of receivers of an item}}.$$

One can observe that there is a threshold (we call it the *"critical value"*) on the recommendation probability (or rate), below which the number of receivers is small and above which the number of receivers is large. This implies that the *recommendation probability* is critical for the information spreading of an item, and we will further study it next.



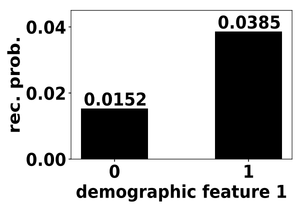
(a) (b)

Fig. 2. Impact of *(a) the number of recommenders, (b) recommendation probability* on the number of receivers

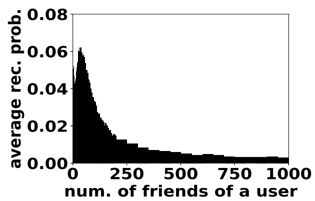
• **Data analysis 2: uncover key factors that influence recommendation probability.** Fig. 3(a) shows the recommendation probability varies significantly for a demographic feature. Fig. 3(b) shows that as the degree (num. of friends) of a user increases, the recommendation probability first increases and then decreases. In particular, users with 35 friends have the highest recommendation probability. To see how the information from friends' recommendations improves recommendation probability, we note that a user can know an item from other sources (say, conventional media). We define *improvement ratio of recommendation probability*

$$\triangleq \frac{\mathbb{P}[\text{recommend the item}|\text{receive friends' recommendation}]}{\mathbb{P}[\text{recommend the item}|\text{do not receive recommendation}]} - 1.$$

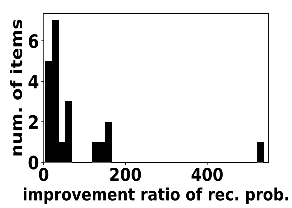
Fig. 3(c) is the histogram of items' improvement ratios. We see that the information of friends' recommendations always improves users' recommendation probability. The improvement ratio can reach two orders of magnitude. Table II shows that on average, offering reward can increase users' recommendation probability by 250% compared with that without reward.



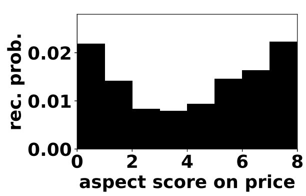
(a)



(b)



(c)



(d)

Fig. 3. Impacts of *(a) user's demographic feature; (b) degree of nodes; (c) information source, (d) item's price* on the recommendation probability.

Table II
IMPACT OF REWARD ON RECOMMENDATION PROBABILITY.

	With reward (average)	Without reward (maximum)
Rec. prob.	0.0271 ($> 3.5 \times 0.0076$)	0.0076

• **Data analysis 3: users' satisfiability for social recommendation.** We use click rate to measure users' satisfiability. In Fig. 4, we use the item with the highest number of recommenders as the representative of each firm. We observe that users' click rates become lower for items that are associated with rewards, compared to those with zero reward. According to the regression line (where the rewards for "zero", "low" "medium", "high" are set to 0, 0.2, 0.5, 0.9), we see that users' click rates and rewards are negatively correlated.

Lessons learned. The recommendation probability for an item needs to be above a critical value so that the item's information can widely spread in an OSN. A number of factors, e.g., users' characteristics, reward, information source, influence the recommendation probability significantly. We observe that users' click rates become lower when firms offer reward for recommendations. Its justification is given in Section VI-A.

B. Implications From the Yelp Data

The Yelp Data. The dataset is from the 11th round of Yelp's dataset challenge [27]. This dataset contains an unweighted

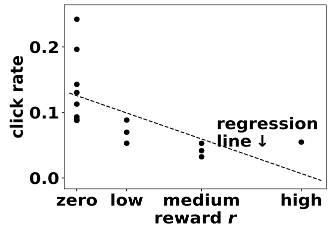


Fig. 4. Rewards and click rates (one point for one firm's representing item)

Fig. 5. Perceivability on improvement ratio of recommendation probability

undirected social network of $1,326,101$ users and $85,474$ businesses. We treat check-ins and reviews as recommendations since friends are notified. Businesses can provide rewards for customers who do check-in, i.e. "check-in offer" [26].

• **Data analysis 4: more factors that influence recommendation probability.** Based on whether the quality of an item is perceivable before purchasing, we classify items into experience goods (one cannot perceive the quality before purchasing, e.g. medical services) and search goods (one can perceive the quality before purchasing, like houses). Fig. 5 shows that the improvement ratios of recommendation probability for experience goods are much higher than that for search goods. In Fig. 3(d), a lower aspect scores on price [25] means the product is less worthy of its price. Users are likely to bad mouth a product when the product is not worthy of the price (low price score), or recommend it positively when it is worthy of the price (high price score). This is because users care about their friends' utilities.

Lessons learned. When an item's quality is hard to perceive before purchasing, friends' recommendations greatly increase a user's recommendation probability because she trusts her friends' recommendations. Moreover, users tend to recommend a product that is worthy its price.

III. SYSTEM MODELS

In this section, we present a model that captures all the observations in Section II. First, we present a reward scheme for social recommendations. Second, we present users' decision model, capturing factors on recommendation probability. Third, we formulate a firm's problem to maximize its profit.

A. The Rewarding Scheme and Firm's Decision Space

We consider a market, which consists of $N \in \mathbb{N}_+$ users denoted by $\mathcal{N} \triangleq \{1, \ldots, N\}$ and $I \in \mathbb{N}_+$ items (or products) denoted by $\mathcal{I} \triangleq \{1, \ldots, I\}$. An item could be the service of Dropbox, the newest iPhone, etc. Each item i is provided by one firm and we call it "firm i". In the case that a firm sells multiple items, we treat it as multiple "virtual" firms. The OSN is modeled as a weighted directed graph $\mathcal{G} \triangleq (\mathcal{N}, \mathbf{W})$ among users, where $\mathbf{W} \triangleq [w_{mn}: m, n \in \mathcal{N}] \in [0,1]^{N \times N}$. A directed link from user m to n captures that user m can recommend items to user n. The weight w_{mn} quantifies the influence strength of user m on user n. A larger w_{mn} indicates a stronger influence.

The firm i can post a reward scheme [11] to incentivize users to recommend items to their friends, which consists of: (1) a recommendation task associated with item i; (2) the reward $r_i \in \mathbb{R}_+$ for completing the task. For example, the task for

Dropbox is to invite a certain number of new users, and the reward is a maximum volume of 22GB cloud storage space. Only the users who adopt (or use) the item are allowed to accept the task. Each user can adopt at most one unit of item and accept at most one task. The firm i sets a price of $p_i \in \mathbb{R}_+$ for item i, which has a per unit marginal cost of $c_i \in \mathbb{R}_+$. The decision for the firm i is to jointly select the price and reward (p_i, r_i) for each item i, where $p_i \geq 0, r_i \geq 0$.

B. The Users' Decision Model

We consider two types of information sources that influence users' decision, i.e., recommendation and other information sources (e.g. traditional advertisements). We track user n's information status on item i using a state variable $S_{ni} \in \{\emptyset, \mathcal{O}, \mathcal{R}, \tilde{\mathcal{O}}, \tilde{\mathcal{R}}\}$, where $S_{ni} = \emptyset, \mathcal{O}$ or $\mathcal{R}$ means that user n does not know item i, is informed of it from conventional information sources (e.g., TV or newspaper advertisement), or is informed of it from friends' recommendations respectively. To describe the diffusion of recommendations, we have two additional states $\tilde{\mathcal{O}}$ and $\tilde{\mathcal{R}}$ for a user. The state $S_{ni} = \tilde{\mathcal{O}}$ (or $\tilde{\mathcal{R}}$) means that user n will be informed of item i from other information sources (or friends' recommendations), but are not currently informed. For example, a user is in $\tilde{\mathcal{R}}$ when he receives a recommendation but does not click it. Fig. 6 depicts the state transition, and these states have a total order $\emptyset \prec \tilde{\mathcal{O}} \prec \mathcal{O} \prec \tilde{\mathcal{R}} \prec \mathcal{R}$, representing the strength of information can only evolves from weak to strong. A user's behaviors depend on her states. This captures the impacts of information sources on users' behaviors (Fig. 3(c)).

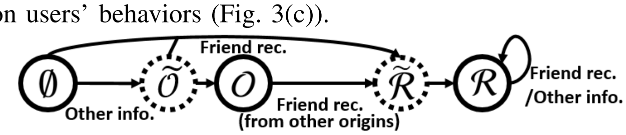


Fig. 6. Transitions of a user's information state

When a user is in state $\mathcal{O}$ or $\mathcal{R}$, he makes decisions. Let $v_{ni} \in \mathbb{R}_+$ denote the valuation of item i by user n, which is the highest price that user n is willing to pay for item i. The intrinsic utility for user n to buy item i or u_{ni}, is

$$u_{ni} \triangleq v_{ni} - p_i, \qquad \forall n \in \mathcal{N}, i \in \mathcal{I}. \tag{1}$$

The u_{ni} represents the personalized quality of item i for user n. Before adopting an item i, user n can only *"estimate"* u_{ni} partially based on the true value u_{ni} and partially based on her information source (represented by S_{ni}). We denote the estimated utility as $\hat{u}_{ni}(S_{ni})$ and model it as

$$\hat{u}_{ni}(S_{ni}) \triangleq \gamma_i u_{ni} + (1 - \gamma_i) Q_{ni}(S_{ni}), \tag{2}$$

where $\gamma_i \in [0, 1]$ models the perceivability of item i, and $Q_{ni}(S_i) \in \mathbb{R}$ denote the personalized quality summarized from a user's information sources. A larger γ_i models that an item is more perceivable in its value (or quality). For example, the haircut service has a small perceivability γ_i while a house has a large γ_i. Let $\tau_n, \tilde{\tau}_n \in \mathbb{R}$ denote user n's trust on recommendations and other information sources respectively. The trust reflects how a user thinks an item known from some information source is worthy of its price. We model Q_{ni} as

$$Q_{ni}(S_{ni}) \triangleq \begin{cases} \tau_n p_i, & \text{if } S_{ni} = \mathcal{R}, \\ \tilde{\tau}_n p_i, & \text{if } S_{ni} = \mathcal{O}. \end{cases} \tag{3}$$

We require $\tau_n \geq \tilde{\tau}_n$ to capture that users often have higher trust on their friends' recommendations. Our models of item's perceivability and users' trust capture previous observation in Fig. 5, i.e., friends' recommendations have different effects on users' behaviors for items with different perceivability.

We have modeled a user's utility to adopt an item. Now we define a user n's utility gain to recommend an item i as:

$$g_{ni} \triangleq r_i + \tilde{v}_{ni} - \tilde{c}_n, \qquad \forall n \in \mathcal{N}, i \in \mathcal{I}, \tag{4}$$

where $\tilde{v}_{ni} \in \mathbb{R}$ denotes the social value and $\tilde{c}_n \in \mathbb{R}_+$ denotes the cost to take up a recommendation task. Recommending good items enhances friendship, leading to a large social value, while recommending a poor item leads to a small social value.

A user n's decision is to choose whether to adopt and recommend item i denoted by $A_{ni} \in \{0, 1\}$ and $R_{ni} \in \{0, 1\}$ respectively, where $A_{ni}=1$ (0) means user n adopts (or not) and $R_{ni}=1$ (0) means user n recommends (or not). Recall that a user can take up a recommendation task if and only if she adopts an item. Thus, the decision space for user n with respect to item i is $(A_{ni}, R_{ni}) \in \{(0,0), (1,0), (1,1)\}$. We define the estimated utility for decision (A_{ni}, R_{ni}) as

$$U_{ni}(A_{ni}, R_{ni}|S_{ni}) \triangleq \begin{cases} 0, & \text{if } (A_{ni}, R_{ni}) = (0, 0), \\ \hat{u}_{ni}(S_{ni}), & \text{if } (A_{ni}, R_{ni}) = (1, 0), \\ \hat{u}_{ni}(S_{ni}) + g_{ni}, & \text{if } (A_{ni}, R_{ni}) = (1, 1). \end{cases}$$

A user decides whether to adopt or recommend an item, when she is informed of the item (her state changes to $\mathcal{O}$ or $\mathcal{R}$). We assume her decision only depends on her estimated utility $U_{ni}(A_{ni}, R_{ni}|S_{ni})$. This *"utility-driven"* assumption is not restrictive, since the utility captures factors such as the item's price (Fig. 3(d)), item's reward (Table II), the item's category (Fig. 5), user's features (Fig. 3(a) and Fig. 3(b)), the user's trust to her friends, etc. For example, when user n maximizes her estimated utility, the optimal decision is

$$(A^*_{ni}, R^*_{ni}) \in \arg \max_{(A_{ni}, R_{ni}) \in \{(0,0),(1,0),(1,1)\}} U_{ni}(A_{ni}, R_{ni}|S_{ni}).$$

Recall that once a user has adopted or recommended an item, the adoption or recommendation can not be revoked. Also, a user only adopts or recommends an item for at most one time. After adopting an item, whether a user will recommend an item is independent of the information status, so a user n's action R^*_{ni} should be same in state $\mathcal{O}$ and $\mathcal{R}$. Thus, users can be categorized into five classes based on their actions under different states, as shown in Fig. 7. For example, both class-2 users and a class-3 users neither adopt nor recommend in state $\mathcal{O}$, and adopt in state $\mathcal{R}$. But class-2 users recommend in state $\mathcal{R}$, while class-3 users do not.

C. A Firm's Decision Model

We divide time into slots starting from t=1. Let us focus on item i for descriptions. At the beginning of time slot 1, each user is in state $\tilde{\mathcal{O}}$ with probability δ_i. Namely, every user can know item i from other information with probability δ_i afterwards. When a user gets informed of an item (from any sources), we say the user *"arrives"* at the firm. At the beginning of time slot t>0, one of the users in states $\tilde{\mathcal{O}}$ or $\tilde{\mathcal{R}}$ arrives. Then, the arrived user makes decisions as described

	$S_{ni}=\mathcal{O}$	$S_{ni}=\mathcal{R}$
Class 1	(1,1)	(1,1)
Class 2	(0,0)	(1,1)
Class 3	(0,0)	(1,0)
Class 4	(1,0)	(1,0)
Class 5	(0,0)	(0,0)

Fig. 7. User's decision (A^*_{ni}, R^*_{ni}) in different states: 5 classes

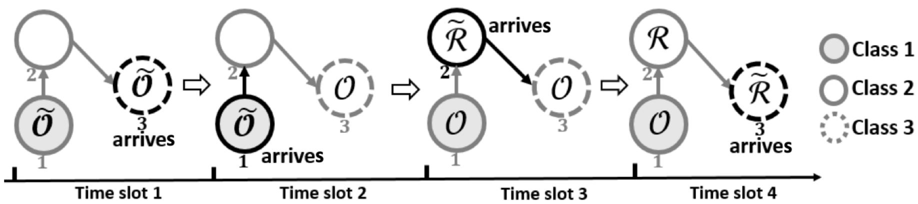


Fig. 8. Diffusion of an item's information. (User n has class n in Fig. 7, n=1, 2, 3)

in Section III-B. If user m decides to recommend, then with probability w_{mn}, a user n will be informed of the item and transit to state $\tilde{\mathcal{R}}$. Recall that w_{mn} is the influence strength of user m on user n. It captures that a user may receive recommendations from (and be influenced by) close friends.

To illustrate, consider the example in Fig. 8. At the beginning of time slot 1, user 3 arrives because he is in state $\tilde{\mathcal{O}}$. Then, user 3, as a class-3 user, neither adopts nor recommends. At the beginning of time slot 2, user 1 arrives and recommends to user 2. Then in time slot 3, user 2 arrives and recommends to user 3. Finally in time slot 4, user 3 re-visit the item because of the recommendation. After that the diffusion terminates as no users will adopt or recommend the item anymore.

Remark. Note that our model is a generalization of the *"Independent Cascade model"* used in the influence maximization problem [14], where users always recommend an item once they are informed of it. In this case, whether a user can receive information (or "be influenced") of item i only depends on the weights of the edges $[w_{mn}{:}m, n{\in}\mathcal{N}]$, given the set of users who initially receive information from other sources.

To characterize a firm's uncertainty about the market, we model a user n's valuation v_{ni}, the social value $\tilde{v}_{ni}$ and recommendation cost $\tilde{c}_n$ as random variables. We assume $(v_{ni}, \tilde{v}_{ni}{-}\tilde{c}_n)$ are independent across different n and i. Moreover, the parameters $(v_{ni}, \tilde{v}_{ni}{-}\tilde{c}_n)$ follow a probability distribution $\mathcal{D}_{ni}$. From the perspective of a firm, a user n in state S_{ni} will recommend (or adopt) an item i with certain probability. We define a user n's probability to adopt and recommend the item i to be $a_{ni}(S_{ni})$ and $q_{ni}(S_{ni})$ respectively, where $S_{ni}{\in}\{\mathcal{O}, \mathcal{R}\}$. Note that $a_{ni}(S_{ni})$ are $q_{ni}(S_{ni})$ are personalized and item-specific. For example, when users maximize their estimated utilities, the probabilities are

$$a_{ni}(S_{ni}){=}\mathbb{P}_{(v_{ni}, \tilde{v}_{ni}-\tilde{c}_n)\sim\mathcal{D}_{ni}}[\hat{u}_{ni}(S_{ni}){+}g_{ni}{>}0 \text{ or } \hat{u}_{ni}(S_{ni}){>}0],$$

$$q_{ni}(S_{ni}){=}\mathbb{P}_{(v_{ni}, \tilde{v}_{ni}-\tilde{c}_n)\sim\mathcal{D}_{ni}}[\hat{u}_{ni}(S_{ni}){+}g_{ni} > 0 \text{ and } g_{ni} > 0].$$

As shown in Fig. 7, whether a user n (in any class) eventually adopts or recommends an item i is determined only by her state S_{ni} when the diffusion terminates. Recall that we consider the case where the firm has the same price p_i and reward r_i for all users. We can then derive the demand (i.e., expected number of adoptions) of item i as

$$D_i(p_i, r_i) {\triangleq} \sum\nolimits_{n\in\mathcal{N}} \left(\sum\nolimits_{S\in\{\mathcal{O},\mathcal{R}\}} \mathbb{P}\left[S_{ni}^{\text{Ter}}{=}S\right] a_{ni}(S) \right).$$

where S_{ni}^{Ter} denotes the final state of a user when the diffusion terminates. Similarly, we derive the expected number of users who make recommendations during the lifetime of item i as:

$$R_i(p_i, r_i) {\triangleq} \sum\nolimits_{n\in\mathcal{N}} \left(\sum\nolimits_{S\in\{\mathcal{O},\mathcal{R}\}} \mathbb{P}\left[S_{ni}^{\text{Ter}}{=}S\right] q_{ni}(S) \right). \quad (5)$$

Then item i's *"net profit"* is $D_i(p_i{,}r_i){\cdot}(p_i{-}c_i){-}R_i(p_i{,}r_i){\cdot}r_i$.

Optimal rewarding scheme problem. It is formulated as:

$$\underset{p_i\geq 0, r_i\geq 0}{maximize}\, P_i(p_i, r_i) {\triangleq} \{D_i(p_i, r_i)\cdot(p_i{-}c_i) - R_i(p_i, r_i)\cdot r_i\}.$$

This problem is challenging to solve. Increasing the reward r_i improves the demand, but also the firm will incur higher expenses. Decreasing the price p_i improves the demand, but also hurts the firm's profit gain. The problem becomes more difficult when the price and reward jointly affect the profit.

IV. Analysis

In this section, we first characterize the demand. Then we characterize the optimal reward and the optimal price.

Characterizing the demand. To facilitate the analysis, we consider the case in which $\mathcal{D}_{ni}, \tau_n, \tilde{\tau}_n$ are identical across different users n, and $\tau_n = \tilde{\tau}_n$. In other words, all users' utilities have identical distributions. We can abbreviate the $a_{ni}(S_{ni})$ and $q_{ni}(S_{ni})$ as a_i and q_i respectively. In the following lemma, we characterize the demand when δ_i is small.

Lemma 1. *Suppose $\mathcal{G}{=}(\mathcal{N}, \mathbf{W})$ degenerates to an unweighted and undirected graph, i.e., $w_{mn} \in \{0, 1\}$ and $w_{mn} = w_{nm}$ for $\forall m, n \in \mathcal{N}$. If the degree of nodes are bounded, then there is a critical value q_i^c such that*

1) $q_i{<}q_i^c \Rightarrow \lim_{|\mathcal{N}|\to\infty, \delta_i\to 0} \frac{D_i}{|\mathcal{N}|}{=}0$,
2) $q_i{>}q_i^c \Rightarrow \lim_{|\mathcal{N}|\to\infty, \delta_i\to 0} \frac{D_i}{|\mathcal{N}|}{>}0$.

Furthermore, if $\mathcal{G}$ is a random graph [4] with degree distribution $\{\rho_k\}_{k=0}^{+\infty}$, then $q_i^c{=}\left(\sum_{k=1}^{+\infty} k\rho_k\right) / \left(\sum_{k=2}^{+\infty} k(k-1)\rho_k\right)$.

Proof sketch. We map the diffusion of recommendations to a percolation process [4] [5]. In this technical report, we put the detailed proofs to the appendix. ■

Lemma 1 states that only if the recommendation probability is higher than a critical value q_i^c, then an infinitesimal fraction ($\delta_i{\to}0$) of users who are informed from other sources (e.g. traditional advertisements) can boost a positive fraction of user to adopt the item. This implies that a firm can offer reward r_i so that the recommendation probability q_i will exceed q_i^c. Simulations on Yelp's graph (Fig. 9) show that the demand is small when the recommendation probability is smaller than a critical value near 0.02. Also, the demand increases slowly when recommendation probability is high enough.

Characterizing optimal reward and price. The following theorem states the optimal reward r_i^* and optimal price p_i^*.

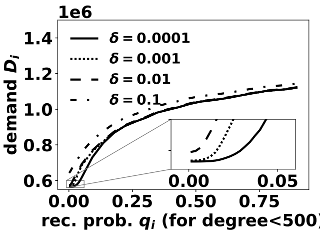


Fig. 9. Recommendation probability q_i and demand D_i (Yelp's graph)

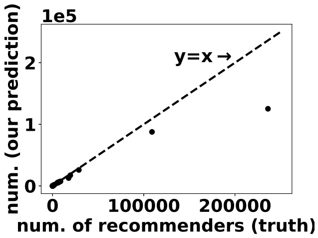


Fig. 10. (WeChat) Predicting the num. of recommenders ($|\mathcal{T}| = 4$)

Theorem 1. *(1) If the parameters satisfy the following condition:* $\left[\frac{dq_i}{dr_i}\frac{q_i dD_i}{D_i dq_i}(p_i - c_i) - \frac{q_i^2}{a_i}\right]\Big|_{r_i=0} > 0$, *then* $r_i^* > 0$.
(2) Assume users maximize their expected utilities. Let $\bar{p}_i$ *be the optimal price when the reward* $r_i = 0$, *then* $p_i^* > \bar{p}_i$ *when* $(V_{ni}, \tilde{V}_{ni} - \tilde{C}_n)$ *follows a 2-dimensional uniform distribution.*

Proof Sketch. We express derivatives of the profit $P_i(p_i, r_i)$ w.r.t. p_i and r_i, then we characterize the optimal p_i^* and r_i^*. ∎

Theorem 1 states (1) a sufficient condition under which a firm should adopt the rewarding scheme; (2) a firm will increase its item's price, when it adopts the scheme to reward the recommenders. This theorem uncovers key factors that determine a firm's profitability: the recommendation probability q_i, the adoption probability a_i, the demand D_i etc. This guides us to study the impacts of these factors in Section VI.

V. Parameter Inference & Model Validation

In this section, we infer model parameters from social recommendation data of a set of items. We also evaluate the accuracy of our inferred model on the WeChat's dataset.

Data model. Consider a social recommendation outcome dataset, which is item-centric, i.e., for each item $i \in \mathcal{I}$, the data contains: (1) the price p_i (or discount $-\Delta p_i$) and reward r_i; (2) whether user n recommends it, i.e., the value of $R_{ni} \in \{0, 1\}$; (3) whether user n receives recommendations of item i, indicated by $\tilde{R}_{ni} \in \{0, 1\}$ ($\tilde{R}_{ni} = 1$ means user n receives and $\tilde{R}_{ni} = 0$ means not). Furthermore, we know the social network graph except for the weights $\mathbf{W}$ on edges.

A. *Predicting an item's diffusion in OSN*

Inferring edge weights. Recall that the weight w_{mn} represents the probability that user n can receive a recommendation from user m. When user m has at least one recommendation (i.e., $\sum_i R_{mi} \geq 1$), we infer w_{mn} as its empirical mean:

$$w_{mn} = \frac{(\text{num. of recommendations } n \text{ receives from } m)}{\left(\sum_{i \in \mathcal{I}} R_{mi}\right)}.$$

When edge (m, n) exists, but user n makes no recommendations in the data, we set w_{mn} to be the average of all edges:

$$\overline{w} \triangleq \sum_{m,n \in \mathcal{N}} w_{mn} \mathbf{1}_{\{\sum_{i \in \mathcal{I}} R_{ni} \geq 1\}} \Big/ \sum_{m,n \in \mathcal{N}} \mathbf{1}_{\{\sum_{i \in \mathcal{I}} R_{ni} \geq 1\}} \quad (6)$$

Parameters for an item's diffusion. In our model, the parameters $\delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R})$ and $\{q_{ni}(\mathcal{R})\}_{n \in \mathcal{N}}$ fully describe a diffusion process, which is shown by Theorem 4 in the appendix. One can derive that $\delta_i \frac{q_{ni}(\mathcal{O})}{q_{ni}(\mathcal{R})} = \frac{\mathbb{P}[R_{ni}=1|\tilde{R}_{ni}=0]}{\mathbb{P}[R_{ni}=1|\tilde{R}_{ni}=1]}$, so we use the empirical value of $\frac{\mathbb{P}[R_{ni}=1|\tilde{R}_{ni}=0]}{\mathbb{P}[R_{ni}=1|\tilde{R}_{ni}=1]}$ to infer $\delta_i \frac{q_{ni}(\mathcal{O})}{q_{ni}(\mathcal{R})}$:

Table III

MAPE for predicting the number of recommenders (WeChat, more than 50 items). *Our model can accurately predict the outcomes of diffusion processes even though the diffusion processes are random. Our model (with 4 types) can reduce the error of IC model by more than 80%.*

	IC Model (Item-independent)	Our model (One type)	**Our model (four types)**	Our model (4K+ types)
MAPE	68.8% (at best)	13.9%	**11.7 %**	11.6%

$$\delta_i \frac{q_{ni}(\mathcal{O})}{q_{ni}(\mathcal{R})} = \frac{\sum_{n \in \mathcal{N}} R_{ni} \times (1 - \tilde{R}_{ni}) / \sum_{n \in \mathcal{N}} (1 - \tilde{R}_{ni})}{\sum_{n \in \mathcal{N}} R_{ni} \times \tilde{R}_{ni} / \sum_{n \in \mathcal{N}} \tilde{R}_{ni}}$$

To infer the recommendation probabilities $\{q_{ni}(\mathcal{R})\}_{n \in \mathcal{N}}$ of each users, we categorize users into types, and estimate $\{q_{ni}(\mathcal{R})\}_{n \in \mathcal{N}}$ for each type. For example, we can categorize the users by their degree, or demographic features, etc. Let us denote the type of user n as $T_n \in \mathcal{T}$ where $\mathcal{T}$ is the set of all types. Then, the probability for a type-T user to recommend an item i in state $\mathcal{R}$ is inferred as

$$q_{T,i} = \left(\sum\nolimits_{n, T_n = T} R_{ni} \times \tilde{R}_{ni}\right) \Big/ \left(\sum\nolimits_{n, T_n = T} \tilde{R}_{ni}\right).$$

Evaluation on WeChat's Data. The number of recommenders reflects how an item's information diffuses in OSN. The number of recommenders is also *nearly linear* in the number of recommendation receivers (Fig. 2(a)). To evaluate the models, we compare the ground truth on the number of recommenders and the models' predictions. In the evaluation, we use a random graph [4] with the same degree distribution as WeChat's social network. Let $N_i^{(R)}$ be the true number of recommenders, and $\widehat{N}_i^{(R)}$ be a model's prediction. We use the mean absolute percentage error (MAPE) as our evaluation metric:

$$MAPE \triangleq \left(\sum\nolimits_{i \in \mathcal{I}} |N_i^{(R)} - \widehat{N}_i^{(R)}| / N_i^{(R)}\right) / |\mathcal{I}|.$$

Table III compares the MAPE's of the independent cascade (IC) model and our model (with different number of types). For our model, we classify the type of nodes according to their degrees because of the observation in Fig. 3(b) First, we treat all nodes as a single type. Second, we group nodes whose degrees are in regions [0,20), [20,60), [60,200), [200,∞) into four types. Third, we group nodes with the same degree into one type, and we get more than 4,000 types. For all these predictions, we use a uniform weight $\overline{w}$ on each edge which is defined in (6). Since we do not train the model with labeled data, we do not split the dataset for training and testing.

The "Independent Cascade model" [14] [12] serves as the baseline. For the IC model, users' recommendation probability q=1, and the number of recommenders only depends on the initial users who know the item from other information sources. In the evaluation, we allow the independent cascade model to tune the uniform weight w on edges, so we get the lowest MAPE that the IC model can achieve, i.e. 68.8%. As pointed out by authors of the paper [12], IC model ignores users' different reactions for different items, hence it cannot capture different items' diffusion outcomes. That is the reason why our model significantly outperforms the IC model.

Fig. 10 illustrates that the number of recommenders predicted by our model matches with the ground truth. In Fig. 2(b), we plot the critical value q_i^c=0.035 predicted by our model (dotted line), which matches our previous observation.

B. *Counterfactual prediction on users' behavior.*

Inferring users' behaviors when firms have different rewarding schemes is counterfactual, since the data is about firms' current reward scheme. As stated in our model, various factors affect a user n's recommending probability on an item i. These factors include the price p_i, the reward r_i, user type T_n, and item type which are denoted by $X_i \in \mathcal{X}$. Since we are interested in the impact of price and reward, users' type T_n and item's type X_i are *"confounding variables"* (a term in causal inference [20]). In particular, we assume a linear form:

$$q_{T,i} \simeq C_T(\theta_0^{X_i} + \theta_1^{X_i} p_i + \theta_2^{X_i} r_i) \quad (7)$$

where C_T is a constant for type T and $\boldsymbol{\theta}^{X_i}$ is for type X_i.

First, $\widehat{q}_i \triangleq \left(\sum_{n\in\mathcal{N}} R_{ni} \times \tilde{R}_{ni}\right) / \left(\sum_{n\in\mathcal{N}} \tilde{R}_{ni}\right)$ is the frequency for a user to recommend item i after receiving recommendations. Second, for each item's type X, we fit a linear function $L_q^{\boldsymbol{\theta}^X}(p,r) = \theta_0^X + \theta_1^X p + \theta_2^X r$ via ordinary least squares where $\widehat{\boldsymbol{\theta}}^X = \arg\min_{\boldsymbol{\theta}} \sum_{i\in\mathcal{I}, X_i = X} (\widehat{q}_i - L_q^{\boldsymbol{\theta}}(p_i, r_i))^2$. The inferred function $L_q^{\widehat{\boldsymbol{\theta}}^X}(p,r)$ reflects the overall impact of price and reward on users' recommendation probability for the items with type X. Third, we infer different types of users' recommendation probability. Suppose a type-T user's recommendation probability is $q'_{T,i}(p'_i, r'_i)$ when the price and reward becomes p'_i and r'_i. Since the recommendation probability of a user in type T is proportional to C_T in (7), we have $q'_{T,i}(p'_i, r'_i)/q_{T,i} = L_q^{\widehat{\boldsymbol{\theta}}^{X_i}}(p', r')/L_q^{\widehat{\boldsymbol{\theta}}^{X_i}}(p, r)$. Therefore, we infer

$$q'_{T,i}(p'_i, r'_i) = q_{T,i} \times L_q^{\widehat{\boldsymbol{\theta}}^{X_i}}(p'_i, r'_i)/L_q^{\widehat{\boldsymbol{\theta}}^{X_i}}(p_i, r_i). \quad (8)$$

Similarly, we infer users' adoption probabilities. Since users' adoption behaviors are mainly affected by an item's price, we use a linear function $L_a^{\widehat{\boldsymbol{\beta}}^X}(p)$ to fit $\{\widehat{q}_i\}_{X_i = X}$. Under price p'_i, the probability for a type-T user to adopt item i is:

$$a'_{T,i}(p'_i) = a_{T,i} \times L_a^{\widehat{\boldsymbol{\beta}}^{X_i}}(p'_i)/L_a^{\widehat{\boldsymbol{\beta}}^{X_i}}(p_i) \quad (9)$$

Note that if one knows a firm's discount instead of price in the data, one can replace p_i with Δp_i in our inference method.

Evaluation on WeChat's Data. The nine items in Table I are from the same firm and have similar characteristics. Hence these nine items have the same item's type. We focus on the impact of price and reward for this particular type of items. We fit a linear function $L_q(\Delta p_i, r_i)$ for these nine items:

$$L_q(\Delta p_i, r_i) = 0.00803 + 0.0239 \times r_i - 0.0202 \times \Delta p_i. \quad (10)$$

The mean squared error $\sqrt{\left(\sum_{1\le i\le 9} (L_q(\Delta p_i, r_i) - \widehat{q}_i)^2\right)/9}$ is 1.54×10^{-4}, which means the error is at most 8.1% of all $\widehat{q}_i$'s. We point out one can hardly validate counterfactual predictions using offline data, so a firm needs to learn users' behaviors in an online setting (Section VII). But one can see that a linear model can approximately reflect how price and reward affect users' recommending probabilities. The linear formula (10) also allows us to simulate users' behaviors in Section VI.

VI. Trace Driven Simulations

With our model and parameters inferred from WeChat and Yelp datasets, we simulate social recommendations. Then, we uncover important insights on when social recommendations can (or cannot) improve firms' profit and users' utilities. Here, the cost c_i is set to 0 so that the parameter p_i stands for $p_i - c_i$.

Table IV
Running time to estimate a firm's profit for different parameters (Monte-Carlo Simulation 2000 samples).

Method	Average time	(Min, Max) time
Agent-based	2970.6s	(2561.8s, 3365.7s)
Pre-computed RR set (ours)	**0.083s**	**(0.016s, 0.55s)**

Simulation algorithms. Note that it is NP-hard to compute the expected number of receivers of information that is diffused in general social networks [6]. Therefore, we design an efficient Monte-Carlo algorithm, in order to quickly search for a firm's optimal strategy. We pre-compute Reverse Reachable(RR) sets [24] of each user, and then estimate the expected number of recommendation receivers by importance sampling using these pre-compuated RR sets. Table IV shows that our algorithm is more than 6000 times faster than agent-based simulation [9] (which treats each user as an agent who behaves according to our model). Details of our algorithm are in the appendix.

A. *Simulating WeChat's Social Recommendations*

We use the parameters inferred from WeChat dataset as input of our model and do simulations, where one firm has one representative item. Fig. 17 shows that social recommendations can improve firms' profits for 9 out of 15 items, by 7%∼71%. In Fig. 18, the items are ranked by their click rates, where the right-most items have the lowest click rates. We observe that when an item's optimal price with the current reward is higher than the optimal price without reward, the item has a low click rate. In other words, our model explains the observation in Fig. 4 that "users' low click rates are associated with rewards" as follows: if a firm offers rewards, then the firm will increase the price which hurts users' utilities, hence users have low click rates. Fig. 15 shows that when firms do not increase prices (and users' utilities are not hurt), social recommendations can improve 4 firms' profits by 7%∼21%.

Lessons learned. Firms will increase the price after providing reward. This explains the association between items' reward and users' low click rates. When firms do not increase price, social recommendations can improve both firms' profit and users' utilities. This suggests the OSN platform to regulate firms so that they will not increase the prices of products.

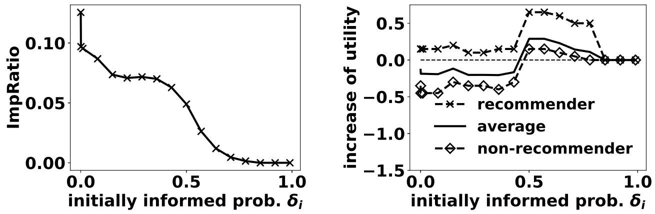
B. *Simulating Yelp's Social Recommendations*

We input the Yelp's graph into our model to do simulations, and further uncover the conditions when social recommendation can (or cannot) improve firms' profits and users' utilities.

1) **Firms' profitability***:* To quantify the improvement of a firm's profit by rewarding recommenders, we compare the profits with/without reward, and define improvement ratio as:

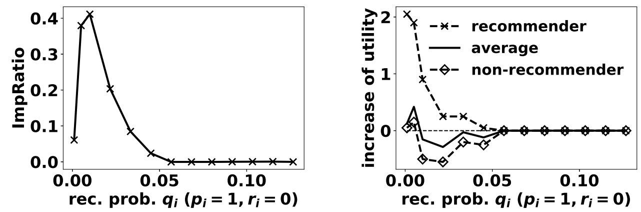
$$\text{ImpRatio} \triangleq \frac{\max_{p_i, r_i \in \{0.05m\}_{m=0}^{40}} P_i(p_i, r_i)}{\max_{p_i \in \{0.05m\}_{m=0}^{40}} P_i(p_i, 0)} - 1.$$

• **Information from other source δ_i.** Fig. 11(a) shows that the improvement ratio decreases as δ_i increases. This implies that incentivized recommendations are benefitial only for firms whose items are not well-known by the public.



(a) **Firm's profit** (b) **Users' utility**

Fig. 11. Impact of prob. to be informed from other sources



(a) **Firm's profit** (b) **Users' utility**

Fig. 12. Impact of users' recommending probability

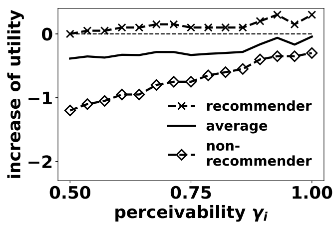


Fig. 13. Perceivability γ_i and users' utilities (τ=0,$\tilde{\tau}$=−0.5)

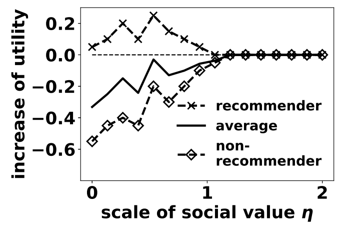


Fig. 14. Social value's scale and users' utilities, $\tilde{v}_{ni}$=$\eta(1{-}p_i)$

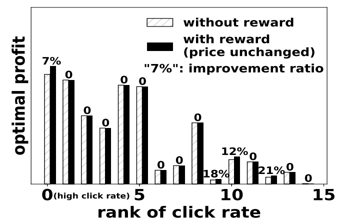


Fig. 15. Profit improvement with items' prices unchanged

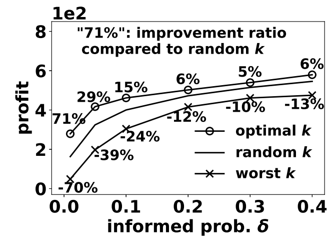


Fig. 16. Choose k for influence maximization, $k{\in}\{100{+}300m\}_{m=0}^{9}$

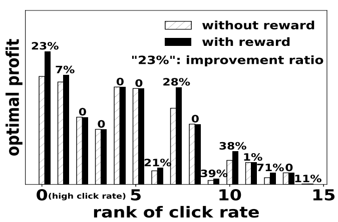


Fig. 17. Profit improvement for items in WeChat

Fig. 18. Reward and the optimal price for items in WeChat

• **Recommendation probability** q_i. In Fig. 12(a), as the recommendation probability increases, we observe that the improvement ratio first increases and then decreases. Moreover, when the initial recommendation probability is close to the critical value $q_i^c = 0.02$ (which matches the critical value in Fig. 9), the improvement ratio reaches the highest value. This is because when users' recommending probability is near the critical value, a small increament of recommending probability driven by rewards can cause a significant improvement of demand. But when users' recommendation probability is originally high (>0.05) without reward, a firm should *not* offer rewards for social recommendations, as the ImpRatio is 0.

Lessons learned for the firms. Rewarding recommenders can improve a firm's profit when the item is not well-known, and users' recommending probability is near a critical value for outbreak of diffusion. Otherwise, firms will not be profitable by using social recommendations.

2) **Users' utility***:* Since a firm's rewarding scheme does not change its item's quality, the item's price is the only factor that affects users' utilities. The *"baseline price"* is a firm's optimal price without rewards, i.e., $\bar{p}_i \triangleq \arg\max_{p_i} P_i(p_i, r_i{=}0)$. Under the rewarding scheme, the firm sets price to p_i^* and sets reward to r_i^*. Then, a recommender needs to pay $p_i^*-r_i^*$. Compared to the baseline price, his *"increase of utility"* is $\bar{p}_i{-}(p_i^*{-}r_i^*)$. Similarly, a non-recommeder has an increase of utility $\bar{p}_i - p_i^*$. We also consider the *"average increase of utility"* of both recommenders and non-recommenders.

• **Information from other sources** δ_i**.** Fig. 11(b) shows that when the proportion $\delta{>}0.5$ of users know the item from other sources, social recommendations improve customers' average utility. This is because a firm will not increase price to lose the large portion of users who know the item from other sources. Also, firms' rewards improve the recommenders' utilities.

• **Recommendation probability** q_i**.** We observe in Fig. 12(b) that when users' recommending probability is very small ($q_i{<}0.01$) without rewards, social recommendation improves users' average utility. This is because only by both decreasing the price and offering reward, a firm can let the recommendation probability exceed its critical value. However, as users' recommendation probability becomes higher ($q_i{\in}[0.01, 0.08]$), social recommendation hurts the non-recommenders' utilities.

• **Perceivability** γ_i**.** In Fig. 13, we observe that social recommendations hurt the utilities of non-recommenders when the perceivability is low. The reason is that when an item has lower perceivability, users rely more on their friends to make decisions. Hence, firms choose to offer high rewards to recommenders and increase prices.

• **Social value.** We set social value as $\tilde{v}_{ni}{=}\eta(1-p_i), \eta{\geq}0$. A larger η indicates that users care more about their friends' utilities. In OSNs whose users have weak-ties with friends (e.g. Twitter), η is small. From Fig. 14, we see the non-recommenders' utilities are heavily hurt when η is small. This is because when users do not care about friends' utilities, firms will offer high reward to recommenders and increase prices.

Lessons learned for the OSN platform. Social recommendation benefits both firms and users when an item is known by a large portion of users from other sources, or when an item has low recommending probability without rewards. OSN platform can select these items for social recommendations. However, the platform should prohibit social recommendation for items whose quality cannot be perceived before purchasing. Moreover, social recommendation will heavily hurt non-recommenders' utilities in "weak-tie" OSNs (e.g. Twitter).

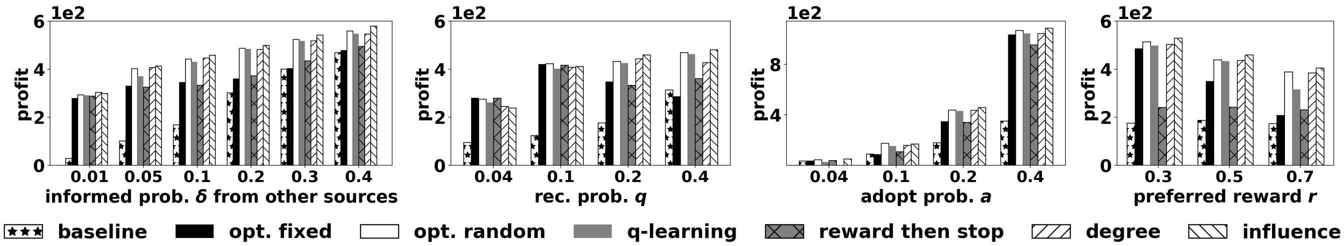


Fig. 19. Firm's total Profits for different algorithms (Facebook graph, average of 200 runs)

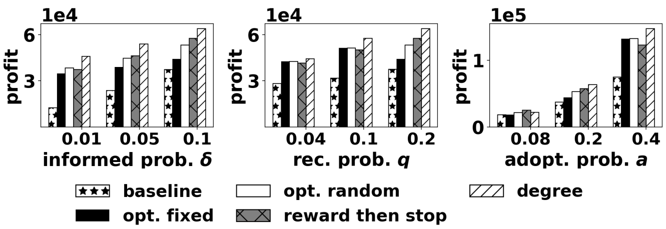


Fig. 20. Firm's total profits (Yelp's graph, average of 200 runs)

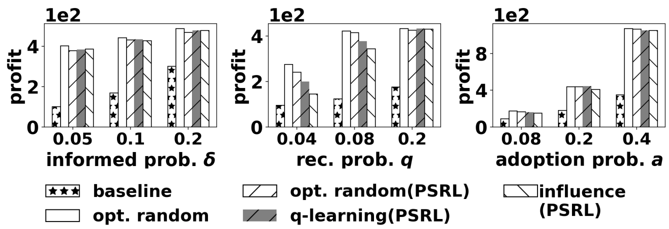


Fig. 21. Profits with known/unknown parameters (Facebook's graph)

Table V
THREE KINDS OF HEURISTIC ALGORITHMS.

Stable strategy	State-dependent	User-dependent
Fixed action	Reward and then stop	High degree
Random action	Q-learning	High influence

VII. Learning to optimize the profit

Now, we consider firms' *"dynamic prices and rewards"*, when model parameters are known or unknown to a firm.

Settings and Markov Decision Process. Recall that we have a discrete time system with $t \in \{1, 2, \dots, \infty\}$. Let $n^{(t)}$ denote the user who arrives in time slot t, whose state $S_{ni}^{(t)}$ is $\tilde{\mathcal{R}}$ or $\tilde{\mathcal{O}}$. Each user in states $\tilde{\mathcal{R}}$ (or $\tilde{\mathcal{O}}$) arrives independently in a Poisson process with an arrival rate $\lambda_{\tilde{\mathcal{R}}} \geq 0$ (or $\lambda_{\tilde{\mathcal{O}}} \geq 0$). Then, a user n in state $S \in \{\tilde{\mathcal{R}}, \tilde{\mathcal{O}}\}$ becomes the next arriver with probability $\lambda_S / (\lambda_{\tilde{\mathcal{R}}} \times |\{n | S_{ni}^{(t)} = \tilde{\mathcal{R}}\}| + \lambda_{\tilde{\mathcal{O}}} \times |\{n | S_{ni}^{(t)} = \tilde{\mathcal{O}}\}|)$. If there are no users in state $\tilde{\mathcal{O}}$ or $\tilde{\mathcal{R}}$, then the system transits to the "Terminal" state. When user $n^{(t)}$ arrives, the firm needs to decide the reward $r_{n^{(t)}i}$ and price $p_{n^{(t)}i}$ for this user. Suppose under $p_{n^{(t)}i}$ and $r_{n^{(t)}i}$, the user's recommending and adopting probability are $q_{n^{(t)}i}$ and $a_{n^{(t)}i}$ respectively. Then the firm i's expected profit in time slot t is $R^{(t)} \triangleq (p_{n^{(t)}i} - c_i) a_{n^{(t)}i} - r_{n^{(t)}i} q_{n^{(t)}i}$.

In summary, a firm's decision process is a Markov Decision Process (MDP). The state of the system at the beginning of time slot t is represented by the state of each user in the OSN and the arriving user, i.e. $(\{S_{ni}^{(t)}\}_{n \in \mathcal{N}}, n^{(t)})$. If no user arrives, then the state transits to "Terminal". A firm's action space is $\mathcal{A} \subseteq \{(p_i, r_i) | p_i \geq 0, 0 \leq r_i \leq p_i\}$ which we assume to be a finite set due to real-world constraints. A firm's policy is a function $\pi(\{S_{ni}\}_{n \in \mathcal{N}}, n) \in \mathcal{A}$ that outputs a price and reward for the arriver n when users' states are $\{S_{ni}\}_{n \in \mathcal{N}}$. The firm's problem is to find a policy π that maximizes the total expected profit $\sum_{t=1}^{+\infty} \mathbb{E}[R^{(t)}]$ (0 profit in "Terminal" state). Note that $S_{ni}^{(t)}$ has 5 possible values, so the system has more than $5^{|\mathcal{N}|}$ possible states, which introduces challenges to optimize a firm's policy.

Algorithms. To optimize a firm's policy, we consider the algorithms shown in Table V, when model parameters are given. A firm's baseline profit is its optimal total profit without reward. The "*fixed action*" algorithm gives all users the same price and reward, and "*random action*" algorithm picks the reward and price with the same probability distribution in each time slot. The "*reward and then stop*" algorithm gives reward only to early arrivers, and "*Q-learning*" sets price and reward using the Q-learning algorithm [13]. The "*high degree*" or "*high influence*" algorithm gives reward to users with "high degree" or "high influence" [14], [24]. We do not consider firms' strategies that depend on users' features (e.g. demographic features) and we leave it as our future work.

Evaluating dynamic reward algorithms. We evaluate these algorithms using Yelp's social network [27] (Section II-B) and a Facebook's social network with 4,038 users [17]. We consider that the firm has a preferred amount of rewards (e.g. 22GB storage space for Dropbox). The detailed settings and the algorithms are in the appendix.

• **Dynamic rewarding with model parameters.** We run each algorithm in Table V for 200 times to estimate a firm's expected profit. Fig. 19 shows the profit on Facebook's graph. First, we observe that randomizing the reward and price can further improve the firm's profit over the fixed action, especially when the preferred reward r_i is large. This is because this randomization can reduce a firm's expenses on rewards by giving rewards to only a fraction of users. Second, the state-dependent strategies "reward and then stop" or "Q-learning" cannot further improve a firm's profits compared to "random action". The indicates that the optimal strategies do not change a lot when users' states change. Third, the user-dependent strategies such as "high degree" or "high influence" can further improve a firm's profit compared to "random action". We have similar observations on Yelp's graph as shown in Fig. 20.

• **Choosing the "k" for influence maximization.** Influence maximization (IM) problem [14] is well studied that aims to select k "seed users" to give rewards, to maximize the influence of a product. However, the IM problem does not consider how to choose the proper value of k. Our *"high influence"* algorithm applies the IM algorithms to select k users to give rewards, and tune k to optimize the firm's profit. Fig. 16 shows that by tuning k, a firm improves its profit by

5%$\sim$71% compared with the strategy to uniformly choose k.

• **Dynamic rewarding with unknown parameters.** Fig. 21 shows the firm's average total profit by running the PSRL version of the algorithms for 25 times. We see that when parameters are unknown, the PSRL version of "random action" or "Q-learning" yields a profit that is comparable to the profit by the optimal random strategy with known parameters.

Lessons learned. Randomizing rewarding strategies can improve a firm's profit. Selecting some users to give rewards can further improves a firm's profit. The PSRL framework can optimize a firm's profit, when model parameters are unknown.

VIII. Related Works

The spread of word-of-mouth in online social network has been studied extensively. Domingos and Richardson [10] [21] studied the marketing problem of rewarding the customers in the social network. Many works studied influence maximization problem [14] [24] [6] that aims to select a set of seed users (to give rewards), to diffuse the product's information via word-of-mouth. However, these works did not address the problem of how to choose the number of seed nodes k because they did not model the firm's profit. Our model can be used to decide the optimal k, as well as the optimal rewards to offer to the recommenders. Our experiments (Fig. 16) show that the choice of the number of seed nodes k has a great impact on firm's profit. Various works proposed models to predict and forecast information cascades in networks [16] [28] [12]. Our model extends the Independent Cascade (IC) model [14] [12] for information's diffusion in OSNs. In particular, our model considers users' decisions on whether to adopt or recommend an item, which significantly reduces the error to predict the number of recommenders compared to the IC model.

Social recommendation scheme was studied from an economic perspective (a.k.a. referral reward programs). Authors in paper [15] studied firms' optimal strategies. Campbell [5] studied impacts of word-of-mouth communication and social network's structure on firms' profits. Our work differs from these works, because our model includes important factors observed from data and we validate our model using data.

There is a rich literature about online decision making and reinforcement learning. Auer et al. in 2002 [2] did finite-time analysis on the upper confidence bound (UCB) algorithm for the MAB problem. Chu et al. [8] consider the contextual MAB problem where the decision maker have extra contextual information. Posterior sampling (a.k.a. Thompson sampling) has provable performance for general reinforcement learning problem [19]. Chen et al. [7] studied the combinatorial MAB problem that can deal with the online influence maximization problem. To the best of our knowledge, we are the first to apply the posterior sampling reinforcement learning framework to decide firms' rewards and prices for social recommendation.

IX. Conclusions

This paper develops a data analytical framework to study social recommendations. We build mathematical models that are inspired by observations from data, and are calibrated with parameters inferred from data. Our model shows that rewarding recommenders can improve a firm's profit when the item is not well-known, or users' recommending probability is near a critical value for outbreak of diffusion. We also uncover conditions under which social recommendation may improve or hurt users' utilities. Moreover, for an online setting, we design reinforcement algorithms for a firm to optimize its strategies for social recommendations.

A firm can use our model to check whether the social recommendation scheme is profitable, and use our online algorithms to optimize its strategies. The OSN platform can use our model to evaluate the impact of social recommendations on its ecosystem, and design regulating policies so that both firms and users can benefit from social recommendation.

References

[1] Technical report, open source code and open data: https://1drv.ms/u/s!AuhX-fJM-sJvgnHt5aEUeGp64j3c.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[3] BBC. Wechat hits one billion monthly users - are you one of them? https://www.bbc.com/news/business-43283690.

[4] D. S. Callaway, M. E. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

[5] A. Campbell. Word-of-mouth communication and percolation in social networks. *American Economic Review*, 103(6):2466–98, 2013.

[6] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. KDD' 10.

[7] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[8] W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. AISTATS' 2011, 2011.

[9] P. Davidsson. Agent based social simulation: A computer science view. *Journal of artificial societies and social simulation*, 5(1), 2002.

[10] P. Domingos and M. Richardson. Mining the network value of customers. KDD' 01, pages 57–66. ACM, 2001.

[11] Dropbox. How to earn more space by inviting friends: https://www.dropbox.com/help/space/earn-space-referring-friends, 2018.

[12] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. WSDM' 10. ACM, 2010.

[13] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? NeuralIPS' 18, pages 4868–4878, 2018.

[14] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. KDD' 03. ACM, 2003.

[15] L. J. Kornish and Q. Li. Optimal referral bonuses with asymmetric information: Firm-offered and interpersonal incentives. *Marketing Science*, 29(1):108–121, 2010.

[16] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[17] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. NeurIPS' 12, 2012.

[18] M. Newman. *Networks*. Oxford university press, 2018.

[19] I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning? ICML' 17, 2017.

[20] J. Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[21] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. KDD' 02, pages 61–70. ACM, 2002.

[22] P. R. Rosenbaum and D. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[23] I. Tam. Didi hk gives $100 discount and launches in-app payment for taxis: https://www.marketing-interactive.com/didi-hk-gives-100-discount-and-launches-in-app-payment-for-taxis, 2018.

[24] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. SIGMOD' 15. ACM, 2015.

[25] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. KDD' 10. ACM, 2010.

[26] Yelp. Check-in offer: biz.yelp.com/support/check_in_offers, 2018.
[27] Yelp. Yelp dataset challenge: www.yelp.com/dataset/challenge, 2018.
[28] L. Yu, P. Cui, F. Wang, C. Song, and S. Yang. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. ICDM' 15, pages 559–568. IEEE, 2015.

Appendix

The appendix is divided into three parts. First, we describe our algorithm for simulation that uses Pre-computed RRsets, and the simulation settings in Section VI. Second, we describe the algorithms used in Section VII to optimize a firm's profit, and the experiments settings. Third, we give the detailed proofs in Section IV.

A. Simulation algorithms and settings in Section VI

We first describe the simulate algorithm, and then describe detailed simulation settings. For more implementation details, we refer to our open source code.

Algorithms for simulation. Recall that computing the expected number of receivers of information diffused in OSN is NP-hard for general graphs [6]. Therefore, we first design efficient Monte-Carlo simulation algorithm.

Algorithm 1 describes how we sequentially generate the samples to know the terminating state of a node with type T. The idea to sequentially generate samples come from Newman's book [18]. First, we uniformly randomly select a node v with type T. Then, we repeatedly select a new node u to be willing to recommend after receiving information, i.e., $\mathcal{S}_R{=}\mathcal{S}_R{\cup}\{u\}$. For a set of users $\mathcal{S}_R$ who are willing to recommend, the reverse reachable set $RRSet(v,\mathcal{S}_R)$ [24] is defined as the set of users, where if any user in the set $\mathcal{S}_R$ is informed by other information, the user v will be informed from recommendation. In each step n, we collect a sample s_n which records the number of nodes in the $RRSet$, and the number of selected users in each type $|\mathcal{S}_R{\cap}\mathcal{N}_T|$, $\forall T{\in}\mathcal{T}$, where $\mathcal{N}_T$ is the set of users of type T.

Algorithm 1: Generating samples for a node type T

```
Function get_sequence_samples():
    v ← drawn uniformly randomly from all type-T nodes
    S_R ← ∅
    for n = 1 to |N| do
        u ← a node selected randomly in N − S_R
        S_R = S_R ∪ {u}
        // {RRSet(v, S_R) is Reverse Reachable Set
        s_n ← (|RRSet(v, S_R)|, {|S_R ∩ N_T|}_{T∈T})
    return {s_n}_{n=1}^{|N|}
// generating the samples
for k = 1 to K do
    {s_{k,n}}_{n=1}^{|N|} ← get_sequence_samples()
```

Algorithm 2 shows how we estimate the expected probability for a user to be informed of an item, using importance sampling. The probability to get a sample $(|RRSet(v,\mathcal{S}_R)|,\{|\mathcal{S}_R{\cap}\mathcal{N}_T|\}_{T\in\mathcal{T}})$ is
$$w(s){\triangleq}\Pi_{T\in\mathcal{T}}\left(\binom{|\mathcal{N}_T|}{|\mathcal{S}_R{\cap}\mathcal{N}_T|}q_{T,i}(\mathcal{R})^{|\mathcal{S}_R\cap\mathcal{N}_T|}(1{-}q_{T,i}(\mathcal{R}))^{|\mathcal{N}_T|-|\mathcal{S}_R\cap\mathcal{N}_T|}\right).$$

Given a size of reverse reachable set $|RRSet(v,\mathcal{S}_R)|$, the probability for the randomly selected node v to be informed from friends' recommendation is
$$v(s) = 1-(1-\delta)^{|RRSet(v,\mathcal{S}_R)|}$$
because each users in the reverse reachable set is independently informed from other information with probability δ.

Algorithm 2: Estimate the probability for a type-T node to be in state $\mathcal{R}$ when the diffusion terminates

```
Function estimate_FriRec_prob(δ_i, {q_{T,i}(R)}_{T∈T}):
    P̂_{T,i} ← Σ_{k≤K,n≤|N|} w(s_{k,n}) v(s_{k,n}) / Σ_{k≤K,n≤|N|} w(s_{k,n})
    return P̂_{T,i}
```

Theorem 2. $\mathbb{E}[\hat{P}_{T,i}]{=}\mathbb{P}[S^{Ter}_{ni}{=}\mathcal{R}|T_n{=}T]$. *Namely, $\hat{P}_{T,i}$ is an unbiased estimator of the probability that a user is in state $\mathcal{R}$.*

Proof. This is importance sampling with inverse propensity score. In particular, the propensity score is accurate. By importance sampling theory [22], our estimator is unbiased. ∎

Simulation settings. We do simulations on the publicly available Yelp's graph.

• **Types of nodes.** We have 3 types for Yelp's graph. The nodes with degrees in ranges $(\{0\},[1,500],[500,+\infty])$ have types (zero, small, large) respectively, denoted as $\mathcal{T}=\{Z,S,L\}$.

• **Basic parameters.** We set the values of the following 4 groups of parameters, which determine the profit of a firm. (1) The default value of δ is set to 0.1, which means 10% of the users in the social network know the item from other information source (e.g. traditional advertisement). (2) The default value of the recommending probability is set to $(q_{Z,i},q_{S,i},q_{L,i})=(0,0.033,0.0198)$, which means that the users with degree less than 500 will recommend an item with probability 0.033, and the users with degree higher than 500 will recommend an item with probability 0.0198. These value are the average recommendation probability for WeChat's users with type "small" and "large". (3) The default value of the adopting probability is set to $(a_{Z,i},a_{S,i},a_{L,i})=(0.2,0.2,0.2)$. (4) We set the unit cost $c_i=0$, as the items in our dataset are mostly Internet products/services which has a low marginal cost to have one more customer. Moreover, the recommending probability and the adopting probability under different prices and rewards are set according to our inference method (8) and (9). In the simulations of Section VI, we vary one parameter around its default values.

• **Other parameters.** Because we assume that users' decisions are *"utility driven"*, users' behaviors are the same when their utilities are the same. The change of trust τ (or $\tilde{\tau}$) and the perceivability γ_i could be seen as the adjustment of price. Regarding to a user's estimated utility, suppose changing the value of perceivability from 1 to γ_i is equivalent to changing the price from p_i to $\tilde{p}_i$. Then we have $\gamma_i(v_{ni}-p_i)+(1-\gamma_i)\tau_n p_i = v_{ni}-\tilde{p}_i$, and we get $\tilde{p}_i = p_i+(1-\gamma_i)(v_{ni}-p_i-\tau_n p_i)$. Since in our model the adoption (recommending) probability is inferred as a linear function of price, Setting the price to $\mathbb{E}[\tilde{p}_i] = p_i+(1-\gamma_i)(\mathbb{E}[v_{ni}]-p_i-\tau_n p_i)$ can get the (expected) probability for a random user to adopt (or recommend).

We thus call $\mathbb{E}[\tilde{p}_i]$ "the effective price" of item i. Similarly, we define the sum of reward and social value $r_i + \tilde{v}$. One can see that an item with reward $r_i + \tilde{v}$ and social value 0 gives a user the same utility of recommendation as the item with reward r_i and social value $\tilde{v}$. Using the effective price and effective reward, we can get the adopting and recommending probability to be $a'_{T,i}(p'_i = \mathbb{E}[\tilde{p}_i])$ and $q'_{T,i}(p'_i = \mathbb{E}[\tilde{p}_i], r'_i = \tilde{r}_i)$ by equations (8) (9) under different parameters of perceivability γ_i, trust τ and social value. Recall that in simulation, we set the social value as $\tilde{v} = \eta \times (1-p_i)$. One can see that for a larger η, a decrease of price corresponds to larger increment of the effective reward. Now, the effective reward is set as $\tilde{r}_i \triangleq r_i + \eta \times (1-p_i)$.

We set the default perceivability γ_i=1 (which means the value of an item is perceivable). The trust for friends' recommendation is set to τ=0, and the default trust for information from other sources is set to $\tilde{\tau}$=−0.5. The expected valuation $\mathbb{E}[v_{ni}]$=1. The default scalar for the "social value" η is 0.5.

• **Comparing the running time for simulation algorithms.** We compare the running time for the agent-based simulation [9] and our algorithm. We use Yelp's graph. We vary the parameters $\delta_i \in \{0.001, 0.01, 0.1, 0.4\}$, and vary the parameters $(q_{Z,i}, q_{S,i}, q_{L,i}) \in \{(0, 0.033, 0.0198)\cdot 0.2, (0, 0.033, 0.0198)\cdot 0.5, (0, 0.033, 0.0198)\cdot 1, (0, 0.033, 0.0198)\cdot 1.5\}$. For each parameter settings, we use 2,000 samples for each of these two algorithms. In particular, we run the agent-based simulation for 2000 times, and we pre-compute 2,000 samples in the estimation of our algorithm.

B. Algorithms and experiment settings in Section VII

Algorithm details for optimization (offline version).

• **(Stable) Randomized action.** A mixed strategy is represented as a vector $\boldsymbol{s}=(s_1,\ldots,s_{|\mathcal{A}|})$ where the sum $\sum_{i=1}^{|\mathcal{A}|} s_i$=1. With probability s_i, the firm takes the i^{th} action in $\mathcal{A}$ ($1 \le i \le |\mathcal{A}|$), for all users.

• **"High-degree".** Select the users whose degree is higher than a threshold, and the firm tunes the threshold to optimize profit.

• **"High-influence".** Use the algorithm in [24] to select k seed users to maximize the "influence" (defined [24]), and tune the k. Note that the Independent Cascade model does not consider a dynamic edge weight, but we can still apply the influence maximization algorithm by setting all the edge weights to users' recommendation probability q_i for item i.

• **"Reward and then stop".** Only give rewards to the first-k arriving users, and the firm tunes the parameter k.

• **"Q-learning".** The details of the Q-learning algorithm are described in the following Algorithm 3.

Algorithm 3: Q-learning algorithm to optimize policy

Function optimal_policy($\{q_T(p,r)\}_{(p,r)\in\mathcal{A},T\in\mathcal{T}}$, $\{a_T(p,r)\}_{(p,r)\in\mathcal{A},T\in\mathcal{T}}$) **:**
  `// This is a Q-learning version`
  Note: users' adopting and recommending probabilities $q_T(p,r), a_T(p,r)$ defines our MDP
  Input: the learning rate α
  for $i \leftarrow 1$ *to* `N_iteration` **do**
    $\boldsymbol{\mathcal{S}}_1 \leftarrow$ initial state, $t \leftarrow 1$
    while *the propogation is not terminated* **do**
      $\tilde{\boldsymbol{\mathcal{S}}}_t \leftarrow$ `get_aggregated_state(`$\boldsymbol{\mathcal{S}_t}$`)`
      $a_t \leftarrow \begin{cases} \text{a random action} & \text{with prob. } \epsilon_i \\ \arg\max_a Q(\tilde{\boldsymbol{\mathcal{S}}}, a) & \text{with prob. } 1-\epsilon_i \end{cases}$
      $\boldsymbol{\mathcal{S}}_{t+1} \leftarrow$ the next state after the action a_t
      $v \leftarrow$ the immediate profit $R^{(t)}$, $t \leftarrow t+1$
    $T \leftarrow t$
    for $t \leftarrow T$ *downto* 1 **do**
      $Q^i(\tilde{\boldsymbol{\mathcal{S}}}_t, a_t) \leftarrow$
        $(1-\alpha)Q^{i-1}(\tilde{\boldsymbol{\mathcal{S}}}_t, a_t) + \alpha(v + \max_a Q^i(\tilde{\boldsymbol{\mathcal{S}}}_{t+1}, a))$

Function get_aggregated_state($\boldsymbol{\mathcal{S}}$**) :**
  return
    $(\lfloor|\{n|S_{ni}=\mathcal{R} \text{ or } \mathcal{O}\}|/G_1\rfloor, \lfloor|\{n|S_{ni}=\tilde{\mathcal{O}} \text{ or } \tilde{\mathcal{R}}\}|/G_2\rfloor)$

To reduce the number of states, we aggregate similar states into a big state. In particular, we represent a state using (1) the number of users who have been informed from either sources $|\{n|S_{ni}=\mathcal{R} \text{ or } \mathcal{O}\}|$, (2) the number of users who will be informed later $|\{n|S_{ni}=\tilde{\mathcal{O}} \text{ or } \tilde{\mathcal{R}}\}|$. We further reduce the number of states by setting the constants G_1=200 and G_2=5.

We use ϵ-decreasing exploration and set ϵ_t to be $\sqrt{1/t}$ for the t^{th} arriver. The default action for a new state is set to be the optimal randomized action. Also, the parameter `N_iteration` in Algorithm 3 is set to 2000.

Extending the above algorithm to an online version. Algorithm 4 outlines the PSRL algorithm. For us, the unknown parameters are different types of users' adopting and recommending probabilities ($q_T(p,r)$ and $a_T(p,r)$) for different price p and reward r. Since whether a user adopts (or recommends) is a Bernoulli event, we record the number of successes and failures. For example, we use $N_q(p,r;T)$ (or $\overline{N}_q(p,r;T)$) to count the number of arriving type-T users who recommends (or does not recommend) under price p and reward r. One can replace the function `optimal_policy` with any algorithms in Table V.

We evaluate the algorithms under different parameters. We consider that the firm already has a preferred amount of rewards, if he decides to offer rewards. Then, the firm has two possible actions: either to give the preferred amount of rewards or to give no rewards. The price of the firm is set to $p_i = 1$. In each experiment, we vary one of the following four parameters, and set the others to default values: (1) the preferred amount of reward r_i, with a default value 0.5; (2) the probability δ_i that a user is initially informed of the item, with

Algorithm 4: Posterior sampling reinforcement learning

$N_q(p,r;T) \leftarrow 1, \overline{N}_q(p,r;T) \leftarrow 1$
$N_a(p,r;T) \leftarrow 1, \overline{N}_a(p,r;T) \leftarrow 1$
while *The state* `Terminal` *is not reach* **do**
  for $\forall T \in \mathcal{T}$ *and* $\forall (p,r) \in \mathcal{A}$ **do**
    Sample $q_T(p,r) \sim Beta\left(N_q(p,r;T), \overline{N}_q(p,r;T)\right)$
    Sample $a_T(p,r) \sim Beta\left(N_a(p,r;T), \overline{N}_a(p,r;T)\right)$
  $\hat{\pi} \leftarrow$ `optimal_policy`$(\{q_T(p,r)\}_{T,p,r}, \{a_T(p,r)\}_{T,p,r})$
  Use the policy $\hat{\pi}$ during the next episode
  for $\forall T \in \mathcal{T}$ *and* $\forall (p,r) \in \mathcal{A}$ **do**
    Update $N_q(p,r;T), \overline{N}_q(p,r;T), N_a(p,r;T), \overline{N}_a(p,r;T)$
    /* E.g. $N_q(p,r;T)$+= num. of type-T recommenders under price p and reward r in this episode */

a default value 0.1; (3) every user's recommendation probability q_i, where the default value is 0.2. A user's recommendation probability without reward is set as $0.1q_i$; (4) every user's adopting probability a_i, with a default value 0.2.

C. Detailed proofs

The main technique we use in the proofs is the "Percolation theory" by network science. To show how the diffusion process in our model can be mapped to a percolation process, we first show that a user's behaviors are independent of the order for him to receive information, formally via the following lemma. Suppose a user n's current action on item i is $(\bar{A}_{ni}, \bar{R}_{ni})$, then her eligible actions are $\mathcal{A}_{ni} \triangleq \{(A_{ni}, R_{ni}) | A_{ni} \geq \bar{A}_{ni}, R_{ni} \geq \bar{R}_{ni}\}$ since she cannot revoke previous actions.

Lemma 2. *A user's decision is independent of her eligible decisions,*

$$\begin{aligned} &\arg\max_{(A_{ni},R_{ni})\in\mathcal{A}_{ni}} \quad U_{ni}(A_{ni},R_{ni}|S_{ni}) \\ = &\arg\max_{(A_{ni},R_{ni})\in\{(0,0),(1,0),(1,1)\}} \quad U_{ni}(A_{ni},R_{ni}|S_{ni}). \end{aligned}$$

Proof. In Table 10, we observe that for any class (class 1 to 5) of users, the optimal decision under the state $\mathcal{R}$ is greater than the optimal decision under the state $\mathcal{O}$. Also, a user's informatino state can only transfer from $\mathcal{O}$ to $\mathcal{R}$. Then, the optimal decision is always eligible when a user's information state transits. ∎

Lemma 2 implies that to calculate the number of adoptions, we only need to focus on the final state (when the diffusion terminates) of users, instead of tracking the historic states.

Proof of Lemma 1. First we map the diffusion of recommendation to a percolation process.

Mapping the diffusion to a percolation process. The diffusion might be more understandable if we describe it as a random process in the graph (i.e., the percolation process). To simulate the diffusion process, we have three steps, as we illustrate in Fig. 22.

In the first step, each node n is selected with probability $q_{ni}(\mathcal{R})$ to make recommendations after being informed in state $\mathcal{R}$. Also, an edge (m,n) is selected with probability w_{mn}. These selected nodes and edges form several connected subgraphs that *"conducts"* recommendations.

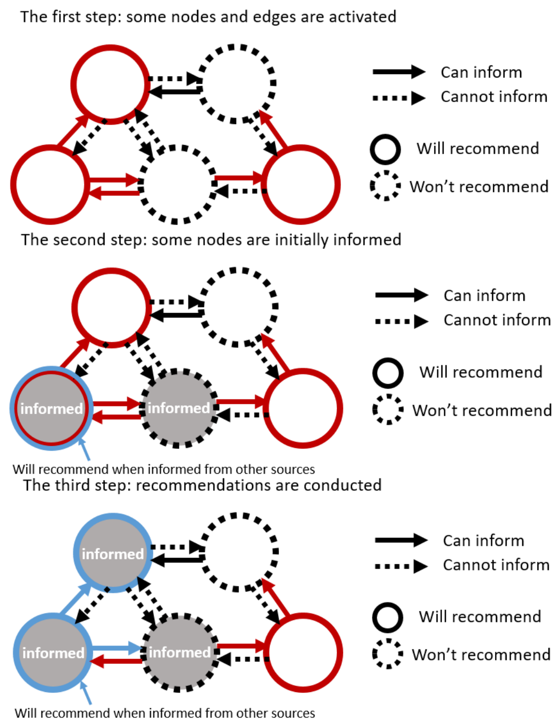


Fig. 22. Three steps of the diffusion process

In the second step, a node n is selected with probability δ_i to be initially informed from other information sources, and the selected nodes transit to state $\mathcal{O}$. Recall that a user has higher trust for recommendations than other information sources, i.e. $\tau_n > \tilde{\tau}_n$. Hence, the recommendation $q_{ni}(\mathcal{O}) < q_{ni}(\mathcal{R})$. Moreover, if a node will not recommend in state $\mathcal{R}$, then he will not recommend in state $\mathcal{O}$. In this step, only the nodes selected in the first step will recommend with probability $q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R})$ which is in $[0,1]$.

In the third step, these initial recommendations get through those *"conductive"* nodes and edges in the social network which are selected in step 1, just like electricity can flow through conductive materials. All the nodes that are got through in this step will receive recommendation and transfer to state $\mathcal{R}$.

In the above process, $q_{ni}(\mathcal{R})$ is called the "occupation probability" on node, and w_{mn} is called the "occupation probability" on edge.

After this mapping, the proof follows the percolation theory [4] [5]. The diffusion of recommendations maps to a percolation process, where the recommendation probability maps to *"occupation probability"*. In the percolation process, when the occupation probability is smaller than a certain threshold, no giant cluster exists so the demand approaches 0 when $\delta \to 0$. ∎

Theorem 3. *Consider diffusion processes for two sets of parameters*

- $\langle \delta_i, \{q_{ni}(\mathcal{O})\}_{n\in\mathcal{N}}, \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}\rangle$
- $\langle \delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R}), \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}, \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}\rangle$

The probability $\mathbb{P}[R_{ni} = 1]$ is the same under the two sets of parameters, provided that $q_{ni}(\mathcal{R})/q_{ni}(\mathcal{O})$ is a constant for every user n.

Proof. In these two processes, the probability that a user who is initially in state $\tilde{\mathcal{O}}$ to make recommendations are the same: $\delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R})$. Also, for a user who is in state $\mathcal{R}$, the recommendation probability are the same, i.e., $q_{ni}(\mathcal{R})$.

If we map the diffusion process in our model as a percolation process illustrated in Fig. 22, then one can see the diffusion process under these two sets of parameters are exactly the same. In fact, the "occupation probability" in step 1 is the same ($q_{ni}(\mathcal{R})$ for user n). Also, in step 3, the probability for a user to be in state $\mathcal{O}$ and make recommendation is the same ($\delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R})$ for user n). ∎

Proof of Theorem 1. We prove the two claims one-by-one.

(1) The first part is a direct result by taking derivative with respect to the price p_i. Let's consider the optimal reward r^*. When the adoption probability $a_{ni}(S) = a$ and the recommendation probability $q_{ni}(S) = q$ for any $n \in \mathcal{N}$ and $S \in \{\mathcal{O}, \mathcal{R}\}$, $R_i(p_i, r_i) = \frac{q}{a} D_i(p_i, r_i)$. Since we fix the price, total profit of the firm is a function of the reward $P(r) = D(p, r)\cdot(p - c) - \frac{q}{a}D(p, r)\cdot r$. Take the derivative with respect to r, we have

$$P'(r) = \frac{D}{q}\left[\left(\frac{q}{D}\frac{dD}{dq}\right)\left(\frac{dq}{dr}\right)(p - c - \frac{q}{a}r) - \frac{q^2}{a}\right].$$

The no-reward strategy corresponds to $r = 0$, so a sufficient condition for the firm to offer reward is $P'(r) > 0$ when $r = 0$.

(2) The second part is about the special case when users' valuations $(V_{ni}, \tilde{V}_{ni} - \tilde{C}_n)$ follow a 2-dimensional uniform distribution.

Recall that firm i's optimal price is p_i^* when rewards are allowed. Recall that firm i's optimal price is $\bar{p}_i$ when reward $r_i = 0$. Let us denote the optimal price as $p_i^\dagger$ when there is no recommendation (i.e. a firm only has users who are informed from other information sources). We prove a stronger result via the following lemma.

Lemma 3. *For an item i with γ_i=1,*

1) $p_i^\dagger > \bar{p}_i$.
2) *If users' parameters $(V_{ni}, \tilde{V}_{ni} - \tilde{C}_n)$ follow two-dimensional uniform distribution, then $p_i^* \geq p_i^\dagger$.*

Proof. 1) First, we show that when there is social network but there is no reward for recommenders, a firm's optimal price will decrease, i.e., $p_i^\dagger > \bar{p}_i$. This is because with the existance of social network (and social recommendations), decreasing the price can not only increase users' adopting probability, but can also increase the number of users who receive friends' recommendations. Formally, we write down a firm's optimization problem ($c_i = 0$):

$$\text{maximize}_{p_i} \qquad P_i \triangleq p_i D_i a_i.$$

Here, D_i denotes the number of users who are informed of the item, and a_i is the probability for a user to adopt the item i, P_i is the firm's profit. Taking derivative w.r.t. p_i, we get $\frac{dP_i}{dp_i} = D_i a_i + p_i \frac{dD_i}{dp_i} a_i + p_i D_i \frac{da_i}{dp_i}$. The price p_i' that maximizes the profit satisfies $\frac{dP_i}{dp_i}|_{p_i=p_i'} = 0$. Furthermore, when the demand D_i is not zero, which is satisfied when $\delta_i \neq 0$, the optimal price should satisfy

$$p' = \frac{a_i}{-\frac{da_i}{dp_i}|_{p_i=p_i'} - \frac{dD_i}{D_i dp_i}|_{p_i=p_i'} a_i}$$

If there are no recommendations of the users, then $\frac{dD_i}{dp_i}$ will always be 0. But when there is social network and users can make recommendations, $\frac{dD_i}{dp_i}$ is negative. One can see that p_i' becomes smaller when there is social network. It means that a firm's optimal price $\bar{p}_i$ (with OSN) is smaller than $p_i^\dagger$ (the optimal price without OSN).

2) Second, for the case where users' valuation and costs are uniformly distributed. We consider the profit gain by increasing the price.

In Fig. 23, we show the fraction of users who will recommend q_i ("Adopt and recommend") and the fraction of users who will adopt a_i ("Adopt and recommend"+"only adopt"), with a price p_i and reward r_i, Let users' recommendation probability q_i be fixed. Then, when price p_i changes, the reward r_i also changes.

As we can see in Fig 23, increasing the price (from p_2 to p_1) will result in a higher reward (from r_2 to r_1), for a firm to reach the same recommendation probability.

Let us consider a price-reward pair (p, r) for the firm (we do not have subscript here). Suppose V is the maximum valuation of a user in the users' population. Then, it is easy to see the optimal price without social recommendation is $p_i^\dagger = V/2$, in this uniform valuation case.

Let $q_i = Q^*$ be the optimal recommendation probability (which is fixed), then we have (from Fig 23's illustration):

$$(V - p)(r + r_0) + \frac{(r + r_0)^2}{2} = Q^*.$$

It implies that

$$p = V - \frac{Q^*}{r + r_0} + \frac{r + r_0}{2}. \tag{11}$$

From (11), we see the price p increases in r.

Suppose C is the maximum value of $c_i - \tilde{v}_i$. The profit for the firm is $P \triangleq (C - r)(V - p)\cdot p + Q^*(p - r)$, where the recommendation probability Q^* is a constant. This is a function of price. Taking derivative of profit with respect to price p_i, we have

$$\frac{dP}{dp} = (C - r)(V - 2p) + Q^*(1 - \frac{dr}{dp}).$$

According to (11), we have $\frac{dp}{dr} = \frac{Q^*}{(r+r_0)^2} + \frac{1}{2}$. Because $V > p$, we have $Q^* > \frac{1}{2}(r + r_0)^2$, so $\frac{dp}{dr} > 1$. It implies that the reward increases slower than compared to the increment of price. Moreover, $\frac{dp}{dr}$ decreases in r, and decreases in p. One could see at $p = V/2$, $\frac{dP}{dp} = Q^*(1 - \frac{dr}{dp}) > 0$. Furthermore, one could verify that $P''(p) < 0$. Therefore, there is a unique optimal price $p_i^* \geq V/2 = p_i^\dagger$. ∎

∎

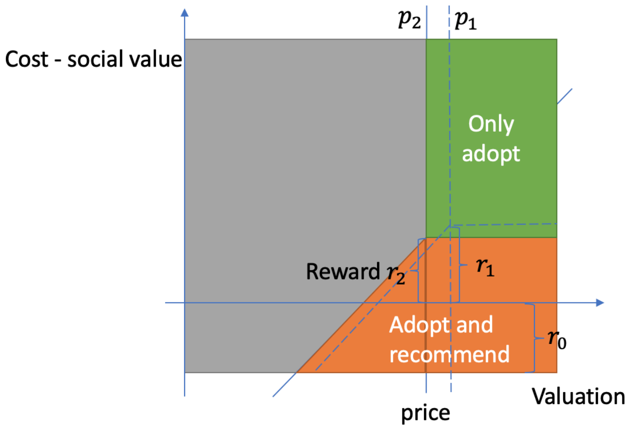


Fig. 23. Population of users with different actions

The following result reduces the number of parameters to describe the diffusion process, which is used in Section V.

Theorem 4. *Consider diffusion processes for two sets of parameters*

- $\langle \delta_i, \{q_{ni}(\mathcal{O})\}_{n\in\mathcal{N}}, \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}\rangle$
- $\langle \delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R}), \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}, \{q_{ni}(\mathcal{R})\}_{n\in\mathcal{N}}\rangle$

The probability $\mathbb{P}[R_{ni} = 1]$ is the same under the two sets of parameters, provided that $q_{ni}(\mathcal{R})/q_{ni}(\mathcal{O})$ is a constant for every user n.

Proof. In these two processes, the probability that a user who is initially in state $\tilde{\mathcal{O}}$ to make recommendations are the same: $\delta_i q_{ni}(\mathcal{O})/q_{ni}(\mathcal{R})$. Also, for a user who is in state $\mathcal{R}$, the recommendation probability are the same, i.e., $q_{ni}(\mathcal{R})$. ∎