Who Ate the Lunch? A Belief Propagation Algorithm for Identifying Refund Hunters*

Chenkai Yu† Arpit Agarwal‡ Hongyao Ma§

February 10, 2026

Abstract

For online platforms such as UberEats and DoorDash, a central challenge in customer service is the lack of ground truth: when a customer reports that an order was never received, it is difficult for support agents to determine if the driver kept the food, a passer-by took the delivery, or the customer is falsely claiming a missing order. This fundamental uncertainty often results in platforms shouldering refunds and appeasement costs without holding either side of the market accountable. In this work, we propose a belief propagation (BP) algorithm for identifying strategic customers and drivers, considering both their frequency of platform use and the trustworthiness of the participants they had interacted with. In large markets, we show that our proposed BP algorithm approximately recovers the correct types on the driver side, and achieves the highest statistical power on the customer side. Additionally, there is an exponentially large gap (with respect to the number of orders per participant) in the ratio of false positive rates between our algorithm and a naive benchmark that only relies only on the average past refund rate of each participant. Extensive experiments on both synthetic data and data from our industry collaborator—a major Southeast Asian platform—demonstrate that the BP algorithm provides substantial and robust accuracy improvements.

1 Introduction

From ordering food and grocery deliveries to booking rides and accommodations, online marketplaces have become an integral part of modern life. This convenience, however, sometimes leads to issues like missing orders or spoiled groceries. A central challenge for customer support on

*The authors would like to thank Itai Ashlagi, Omar Besbes, Frank Chang, Yeon-Koo Che, Nikhil Garg, Sergey Gitlin, Yash Kanoria, Cinar Kilcioglu, Jake Marcinek, Hamid Nazerzadeh, David C. Parkes, Sophie Wang, Jens Witkowski, Helin Zhu, and participants at various seminars and conferences for helpful comments and discussions. Yu was partially supported by the Deming fellowship. Ma gratefully acknowledges the support of the Columbia Digital Future Initiative research grant.

†Decision, Risk, and Operations Division, Columbia Business School, cyu26@gsb.columbia.edu.

‡Department of Computer Science and Engineering, IIT Bombay, aarpit@iitb.ac.in.

§Decision, Risk, and Operations Division, Columbia Business School, hongyao.ma@columbia.edu.

these platforms is the *lack of ground truth*. For example, when an UberEats or DoorDash customer claims that an order marked as "delivered" by the driver cannot be found, the customer support agents have limited means to determine *who ate the lunch*: Did the driver keep the food? Was the delivery taken by a passer-by? Or did the customer, in fact, receive the order but is falsely reporting it in order to get a free meal?

Similar problems arise when a Lyft rider reports that the driver drove through a red light, or when a FreshDirect (an online grocery company) customer claims that the pack of strawberries they just received was moldy. Partially due to this inherent uncertainty, platforms often end up issuing refunds and appeasements to customers to sustain their satisfaction and loyalty, without being able to recoup these costs from the other side of the market. This relatively lenient approach opens avenues for "refund hunters" to game the system, resulting not only in substantial financial losses but also in considerable challenges for quality management, which in the absence of in-person interactions relies almost solely on customer feedback.

Over the years, numerous guardrails have been implemented to deter blatant abuses of the customer support systems. For instance, UberEats denies refunds for customers who reported issues with a large fraction of their recent orders.[1] Similarly, DoorDash tracks each driver's number of missing orders in the past 100 deliveries and may deactivate those who have lost multiple orders.[2] Despite these measures, ensuring fair and effective resolutions for all parties remains challenging for several reasons:

- *Limited data on individual histories.* When the average customer places no more than two dozen orders per year and the overall defect rate is only a few percentage points, relying solely on each individual's past outcomes to detect fraud inevitably leads to high *false negatives* and *false positives*. Both errors carry significant costs. False positives, such as denying legitimate claims, foster frustration and can lead to long-term consequences like customer churn. Conversely, false negatives on the customer side imply that honest drivers also "lose" orders occasionally. This increases the *noise* and false positives on the driver side, where unjust deactivations are seen as profoundly unfair.[3]
- *Cumbersome mitigation strategies.* Many existing guardrails effectively require customers to "jump through hoops", which is only marginally effective and imposes costs on both sides of the market. DoorDash, for instance, launched a feature in 2024 that requires customers to provide a 4-digit PIN to the driver at delivery.[4] Waiting to hand over the order in person and receive the PIN costs drivers valuable time. Moreover, when customers are not at home or are otherwise engaged, they might feel compelled to share the PIN through calls or messages, in

[1] https://x.com/JoshDenny/status/1555809077522882560?mx=2, accessed February 3, 2025.

[2] See https://help.doordash.com/dashers/s/article/customer-reported-order-never-arrived-faq? and https://www.reddit.com/r/doordash/comments/mj22he/customer_grabbed_their_food_and_said_it_never/, February 3, 2025.

[3] Uber, for example, acknowledges the distress unfair account deactivation imposes on drivers, and promises additional pathways to review deactivation decision as of November 2023. See CEO announcement https://www.uber.com/newsroom/onlyonuber23/ and details https://www.uber.com/blog/deactivations-principles/, accessed February 3, 2025.

[4] See https://about.doordash.com/en-us/news/doordash-announces-changes-for-account-deactivations for a detailed explanation on the PIN mechanism as well as discussions on unfair deactivation, accessed February 3, 2025.

which case a strategic driver can keep the order without further accountability once the PIN is confirmed.

1.1 Our Results

In this work, we develop a belief propagation (BP) algorithm for identifying strategic customers and drivers in the context of customer support on online platforms. The algorithm is approximately optimal, and achieves substantial and robust accuracy gains on both synthetic data and data provided by our industry collaborator, a major Southeast Asian platform.

When an order is reported, assessing trustworthiness solely by the customer's past refund rate and the driver's average defect rate—what we call the "naïve score"—is suboptimal:

- A customer with 10 orders in total, 5 reported defective, is more likely to be strategic than a customer who requested a refund once out of 2 total orders.
- A customer is more likely to be credible if many other customers also experienced issues with drivers the customer reported in the past. Conversely, the customer is more likely to be strategic if the customer reports drivers who appear highly reliable while serving other customers.

To address these limitations of the naive score, which is simple and widely used in practice, our BP algorithm leverages the entire history of interactions among all customers and drivers, producing coherent trust scores for both sides of the market that also takes into consideration the frequency of platform use of each agent.

Model. We consider a delivery platform with a number of customers and drivers, where each participant is either *strategic* or *straightforward*. When a customer places an order, a driver is assigned to deliver it. A straightforward driver delivers the order, while a strategic driver may, with some probability, keep the order for themselves but mark it as delivered. Similarly, a straightforward customer reports missing orders truthfully, whereas a strategic customer may falsely claim the order as missing with some probability. External factors, such as a neighbor or a passerby intercepting the order, also occur with some probability.

The platform observes the *order history*, including which customer placed each order, which driver was assigned to deliver it, and whether the order was reported as missing by the customer. However, the platform does not observe the actual outcome of each order, i.e., whether the order was received by the customer, retained by the driver, or intercepted by a neighbor. The goal of the platform is to identify strategic customers and drivers based on the observed history.

A naïve approach widely adopted in practice is to compute the average past outcomes, i.e., the fraction of orders reported missing for each user. However, this approach leaves much information on the table, as it overlooks participants' frequency of platform use and the interactions between the two sides of the market. A full Bayesian approach assumes a prior probability for agents on each side of the market to be strategic, and considers the full history of interactions among all agents. While this works well on small economies, it doesn't scale to realistic settings, where a single city may have millions of customers and hundreds of thousands of drivers.

Main Results. We propose a belief propagation (BP) algorithm for estimating the strategic or straightforward nature of each agent, based on the observed order history. The algorithm

alternates between updating driver scores and customer scores. For each side, the algorithm updates the beliefs about each agent's type based on the ratio (between assuming the agent is strategic and straightforward, respectively) of the expected likelihoods of observing the realized outcomes of each agent's orders. The expectation is taken over the type of the agents on the other side of the market, given the algorithm's current beliefs. The estimated beliefs approximate the posterior marginal probabilities of each agent being strategic, conditional on the observed data. It follows from the classical results that the approximation is exact if the underlying bipartite graph is a tree, but in general it is not exact and the algorithm is not guaranteed to converge.

Our first theoretical result (Theorem 1) is a finite-sample error bound on the estimated beliefs after one iteration of the algorithm. After initializing the customer scores, the algorithm updates the driver scores. We show that at this point, the difference between the estimated driver scores and the true driver types is exponentially small in the number of orders per driver, with high probability. The algorithm then updates the customer scores based on the estimated driver scores. At this point, a similar error bound can be established for the customer scores, which is exponentially small in the number of orders per customer. However, since in practice the number of orders per customer is often small, we instead establish a different optimality result for the customer scores. Specifically, we show that the customer score at this stage is close to an "oracle" score based on the true driver types, which achieves the optimal tradeoff between true positive rates and false positive rates for all customers, and the error is exponentially small in the number of orders per *driver*.

Our second theoretical result (Theorem 2) compares the decay rate of the false positive rates under the naïve approach, the "oracle" approach which uses true types on opposing side, and the belief propagation algorithm. We show that as the number of orders per agent increases, the false positive rates under all three approaches decay exponentially to zero. However, the decay rate under the belief propagation approach is the same as that under the "oracle" approach, and is strictly faster than that under the naïve approach. As a result, the gap between naïve and BP, in terms of the ratio of false positive rates, grows exponentially in the number of orders per agent.

Extensive simulations on both synthetic and real data demonstrate that the algorithm achieves substantially higher accuracy in comparison to all benchmarks, including the naïve approach that only considers the average past outcomes of each user, the maximum likelihood estimation, and the message passing algorithm of Karger et al. [2011] adapted to our setting. The algorithm is also robust to parameter mis-specification. On real data provided by our industry collaborator, we split the data into training and testing sets and evaluate the algorithm's performance in the absence of ground truth.

1.2 Related Work

To the best of our knowledge, ours is the first work focusing on fraud detection and quality management in the context of two-sided platforms.

Crowdsourcing. In a typical crowdsourcing setting the 'workers' and 'tasks' constitute two sides of the market. Workers provide reports on matched tasks and the goal is to aggregate the reports of multiple workers in order to arrive at a more 'accurate' aggregated report. Addition-

ally, one also wants to understand the 'accuracy' or 'reliability' of the workers. The Dawid-Skene model is a popular model for crowdsourcing which assumes that worker i has a single 'accuracy' parameter p_i that is independent of the matched task and the worker reports accurately on a task with probability p_i. Dawid and Skene [1979] propose an EM-based algorithm for this problem which iterates between estimating task labels and worker accuracy. Karger et al. [2011] propose an iterative message passing algorithm and provide theoretical bounds for the error in the recovered worker parameters and aggregated reports. Gao and Zhou [2013] provide theoretical guarantees for the original EM algorithm but their guarantees only hold when all $n \times m$ entries are observed. Zhang et al. [2016] also provide theoretical guarantees for the EM algorithm given an initialization using a spectral method. The key difference between our work and the crowdsourcing literature is that our setting allows both sides of the market to be unreliable and the final defect rate is a combination of this two-sided unreliability. This is not usually the case in the crowdsourcing literature, barring a few exceptions [Khetan and Oh, 2016]. Another difference is that our setting is much more large scale than typical crowdsourcing settings where the number of tasks is often limited. Moreover, the crowdsourcing setting typically allows more explicit control on the worker and task matching. This leads to more balanced and dense bipartite graphs, which is a property that is often exploited by the proposed algorithms.

Many other settings of crowdsourcing have also been studied, for example, peer grading where the goal is grade a student's homework using the 'reports' provided by their peers. The closest to our setting is Zarkoob et al. [2023] which considers Bayesian inference for the problem of peer grading.

Item-response. The literature on *item-response* theory also considers two sides: 'students' and 'questions'. The students are associated with a 'competency' level, and the questions are associated with a 'difficulty' level. Whether the student answers a question correctly depends on a combination of these 'competency' and 'difficulty' factors. The widely-studied Rasch model uses a sigmoid function for mapping the 'competency' and 'difficulty' parameters to the probability of correct answer [Rasch, 1960]. Given responses from the students, the goal is to estimate the difficulty of the questions as well as the competency of the students.

Several algorithms have been proposed for this problem. This includes algorithms based on maximum likelihood estimation Rasch [1960], Fischer [1981] as well as on the spectral method Nguyen and Zhang [2022]. The closest to our work is perhaps the variational Bayes approach of Wu et al. [2020] under the Rasch model. However, these approaches are not directly applicable to our setting because of two key differences. Firstly, our model is motivated by our two-sided fraud detection problem and is different than the models considered in the item-response setting such as the Rasch model. Secondly, the number of questions and students in the item-response setting is rather small, and the algorithms typically rely on observing a fully connected bipartite graph. On the other hand, we are interested in a large economy setting where the number of customers and drivers is large while the bipartite graph is sparsely connected. Moreover, there are very limited theoretical recovery guarantees known for the item-response problem, barring a few exceptions, but even these rely on every student answering every question Nguyen and Zhang [2022].

Matrix completion. Another related problem is matrix completion where the goal is to infer missing entries in a matrix. This problem appears most popularly in the context of movie recommendation where the goal is to complete the sparsely populated user-movie rating matrix. A common assumption in this problem is that the matrix is low-rank, and the observed entries are sufficient to infer the unobserved entries. Our problem can be cast into a low-rank matrix completion problem by observing that the probability of each successfully delivered order is a product of two terms that depend on the customer and the driver, respectively (see Section 2 for more details). However, using standard off-the-shelf method for matrix completion did not yield good results for our problem during our simulation study. The most related to our work is perhaps Gopalan et al. [2015] who consider a Bayesian approach to matrix completion and propose a variational algorithm for inference. However, their approach is based on a generative hierarchical Poisson model which is very different from our model of strategic or straightforward behavior.

Belief propagation. Belief propagation (BP) is a message-passing algorithm for performing inference on graphical models [Wainwright and Jordan, 2008]. While it is exact on tree-structured graphs, it is widely used as an approximate inference method on graphs with cycles, known as loopy belief propagation. BP has been successfully applied to various problems such as decoding error-correcting codes [Richardson and Urbanke, 2008] and community detection in stochastic block models [Decelle et al., 2011]. Our algorithm adapts BP to the problem of two-sided fraud detection. Unlike most prior work which relies on asymptotic analysis in the limit of large graphs, we provide finite-sample guarantees for our algorithm that explicitly characterize the error decay. While our theoretical analysis assumes uniform order counts, our extensive simulations demonstrate that the algorithm is robust to heterogeneity in the number of orders.

2 Preliminaries

In this section, we introduce a model for customer support on two-sided platforms, and formulate the problem of identifying strategic customers and drivers. We also describe a naïve approach that fails to make effective use of available information, and a Bayesian approach which performs well on small economies but does not scale to practical settings.

Model. Consider a platform with $n \geq 1$ customers and $m \geq 1$ drivers. Each agent is either *strategic* or *straightforward*. Straightforward drivers always deliver their assigned orders, while strategic drivers may keep some orders for themselves. Similarly, strategic customers may fraudulently report missing order even when they have received the orders, while straightforward customers always report truthfully. Furthermore, orders are subject to theft by external parties (e.g., neighbors or passersby).

Formally, we use $\mathcal{N} = [n]$ and $\mathcal{M} = [m]$ to denote the set of customers and drivers, respectively. Let $\boldsymbol{u} \in \{0,1\}^n$ be the vector of customer types, where $u_i = 1$ if customer $i \in \mathcal{N}$ is strategic and $u_i = 0$ if the customer is straightforward. Similarly, $\boldsymbol{v} \in \{0,1\}^m$ denotes the vector of driver types. Additionally, $\alpha^* \in (0,1)$ is the probability that a strategic customer falsely reports a missing order, $\beta^* \in (0,1)$ is the probability that a strategic driver retains the order, and $\gamma^* \in (0,1)$ denotes the probability that an (otherwise successful) order is taken by a neighbor.

The timeline of each order is as follows.

1. A customer $i \in \mathcal{N}$ places an order, and the platform assigns the order to a driver $j \in \mathcal{M}$.
2. Driver j keeps the order with probability $\beta^* v_j$, and delivers it with probability $1 - \beta^* v_j$.
3. With probability γ^*, a delivered order is taken by a neighbor or passerby.
4. If the order is missing, the customer reports it to the platform. Otherwise, the customer still reports the order as not delivered with probability $\alpha^* u_i$.

Assuming that all events are independent, the probability that an order delivered by driver j to customer i is reported missing is

$$1 - (1 - \alpha^* u_i)(1 - \beta^* v_j)(1 - \gamma^*).$$

Let $\mathcal{H}$ to denote the *order history*. For each order, the platform observes the customer and the driver associated with this order, and whether the order is reported missing by the customer. Let $\mathcal{H}_{i,j}$ denote the subset of orders involving customer $i \in \mathcal{N}$ and driver $j \in \mathcal{M}$. From this, we derive $T_{i,j} \triangleq |\mathcal{H}_{i,j}|$, the total number of orders between customer i and driver j, and $R_{i,j}$, the number of *reports* (i.e., orders that are reported) between the two. The number of successful (unreported) orders is $S_{i,j} \triangleq T_{i,j} - R_{i,j}$. Without loss of generality, we assume that every agent has at least one order, i.e., $\sum_{j=1}^{m} T_{i,j} \geq 1$ for all $i \in \mathcal{N}$, and $\sum_{i=1}^{n} T_{i,j} \geq 1$ for all $j \in \mathcal{M}$.

The platform does not observe the customer and driver types, or who is responsible for each order that is reported missing. The goal of the detection problem is to identify strategic agents, i.e., to infer the customer and driver types from the order history $\mathcal{H}$.

Naïve Approach. A simple and transparent approach, widely used in practice, is to score each agent by the fraction of their orders that are reported missing. If this average defect rate is high, the agent is considered more likely to be strategic. Formally, the *naïve scores* are defined as

$$p_i^{\text{naïve}} \triangleq \frac{\sum_{j=1}^{m} R_{i,j}}{\sum_{j=1}^{m} T_{i,j}}, \ \forall i \in \mathcal{N}, \quad q_j^{\text{naïve}} \triangleq \frac{\sum_{i=1}^{n} R_{i,j}}{\sum_{i=1}^{n} T_{i,j}}, \ \forall j \in \mathcal{M}. \tag{1}$$

The platform may set a threshold for each side of the market, and *flag* agents with naive scores above the thresholds as strategic. This, however, fails to take two factors into consideration:

- Number of observations: A customer with 10 orders in total, 5 reported missing, is intuitively more likely to be strategic than a customer who reported 1 of her 2 orders as missing.
- Attribution based on interaction history: A customer is more likely to be "credible" if many other customers also experience missing orders with the drivers who were reported by this customer in the past.

Bayesian Approach. To address the limitations of the naïve approach, we may consider a Bayesian approach that assumes some prior distribution on customer and driver types, and updates the beliefs based on the entire order history $\mathcal{H}$. This results in a posterior *joint* distribution on the customer and driver types, and the posterior marginal probabilities provide an estimation of trustworthiness for each customer and each driver.

Specifically, we assume independent and identical Bernoulli distributions for the types of each side of the market, and use $\phi \in (0, 1)$ and $\psi \in (0, 1)$ to denote the prior probabilities for

each customer and each driver to be strategic. We also assume that the defect probabilities for customers, drivers, and other factors (such as neighbors) are $\alpha \in (0,1)$, $\beta \in (0,1)$, and $\gamma \in (0,1)$, respectively, which can be different from the true defect probabilities α^*, β^* and γ^*.

Denote the posterior distribution as $\pi : \{0,1\}^{n+m} \to [0,1]$, we know from Bayes' rule that for all $\boldsymbol{x} \in \{0,1\}^n$ and $\boldsymbol{y} \in \{0,1\}^m$, the posterior probability given order history $\mathcal{H}$ is

$$\pi(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{P}(\boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{y} \,|\, \mathcal{H}) = \frac{\mathbb{P}(\boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{y})\, \mathbb{P}(\mathcal{H} \,|\, \boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{y})}{\mathbb{P}(\mathcal{H})}. \tag{2}$$

Here, the denominator is a normalizing constant given the order history $\mathcal{H}$ and the prior is

$$\mathbb{P}(\boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{y}) = \prod_{i=1}^{n} \phi^{x_i} (1-\phi)^{1-x_i} \prod_{j=1}^{m} \psi^{y_j} (1-\psi)^{1-y_j}, \tag{3}$$

The likelihood of observing the order history $\mathcal{H}$ is given by

$$\mathbb{P}(\mathcal{H} \,|\, \boldsymbol{u} = \boldsymbol{x}, \boldsymbol{v} = \boldsymbol{y}) = \prod_{i=1}^{n} \prod_{j=1}^{m} \mathbb{P}\big(\mathcal{H}_{i,j} \,\big|\, u_i = x_i, v_j = y_j\big), \tag{4}$$

where the likelihood of observing orders $\mathcal{H}_{i,j}$ between customer i and driver j is

$$\mathbb{P}\big(\mathcal{H}_{i,j} \,\big|\, u_i = x_i, v_j = y_j\big) = \big(1 - (1-\alpha x_i)(1-\beta y_j)(1-\gamma)\big)^{R_{i,j}} \big((1-\alpha x_i)(1-\beta y_j)(1-\gamma)\big)^{S_{i,j}}.^{5} \tag{5}$$

The marginal posterior probability of a customer $i \in \mathcal{N}$ being strategic can be obtained by summing over the possible types of the rest of the customers and all the drivers

$$\mathbb{P}(u_i = 1 \,|\, \mathcal{H}) = \sum_{\boldsymbol{x}_{-i} \in \{0,1\}^{n-1}} \sum_{\boldsymbol{y} \in \{0,1\}^m} \mathbb{P}(u_i = 1, \boldsymbol{u}_{-i} = \boldsymbol{x}_{-i}, \boldsymbol{v} = \boldsymbol{y} \,|\, \mathcal{H}), \tag{6}$$

and the marginal posterior probability for each driver $j \in \mathcal{M}$ to be strategic $\mathbb{P}\big(\mathrm{v}_j = 1 \,\big|\, \mathcal{H}\big)$ can be computed in the same way. The platform may then flag agents with high marginal posterior probabilities as strategic.

The Bayesian approach is appealing, as it addresses the weaknesses of the naïve approach—the introduction of the prior means that the algorithm will not be too confident when an agent with a high naive score has few orders, and the likelihood term allows the algorithm to take into account the trustworthiness of everyone the agent has interacted with.

However, this full Bayesian approach is computationally intractable: the posterior π is a joint distribution of $n+m$ binary variables, which assigns a probability to each one of the 2^{n+m} possible combinations of customer and driver types; computing the marginal probabilities (6) requires summing over an exponential number of terms. Although it works well for small instances, it does not scale to markets with millions of customers and hundreds of thousands of drivers.

[5] Assuming that the order history $\mathcal{H}_{i,j}$ contains information on the sequence of outcomes of all orders between customer i and driver j, the expression (5) does not have a $\binom{T_{i,j}}{R_{i,j}}$ term. This is without loss of generality because the same term would also appear in the denominator of (2), thereby canceling out each other.

The following Example illustrates that despite independent priors, the posterior beliefs over agents' types are no longer independent after conditioning on observed data.

Example 1. Consider an economy with one customer and one driver. Assume that the prior probabilities for the customer and the driver to be strategic are $\phi = 0.1$ and $\psi = 0.1$, respectively. Suppose $\alpha = 0.03$, $\beta = 0.03$, and $\gamma = 0.002$, and that the order history $\mathcal{H}$ contains a total of 100 orders, 3 of which are reported missing. The posterior joint distribution π for the customer and the driver being strategic are shown in Table 1.

Table 1: Posterior joint distribution π for the customer and driver types in Example 1.

	$v = 0$	$v = 1$
$u = 0$	0.0204	0.4801
$u = 1$	0.4801	0.0193

The posterior joint probabilities take higher values when exactly one of the agents is strategic. The posterior distribution is not independent, even though the prior distribution over agent types is independent. Technically, this is because the likelihood (5) in the Bayes rule contains non-trivial product terms between the customer and the driver types, which are then taken to the power of the number of reported and successful orders. In this example, γ is small, while the fraction of missing orders is precisely $3/100 = \alpha = \beta$. Such an observation is strong enough to deduce that it is likely that the driver or the customer is strategic, but not both or neither.

3 A Belief Propagation Algorithm

In this section, we introduce a belief propagation (BP) algorithm for identifying strategic customers and drivers. The algorithm iteratively updates the estimates on each side of the market, given current beliefs on the other side of the market. The estimation for each agent approximates the posterior marginal probability that the agent is strategic, given the observed order history.

Formally, the BP algorithm (Algorithm 1) outputs a score $p_i \in [0, 1]$ for each customer $i \in \mathcal{N}$ and a score $q_j \in [0, 1]$ for each driver $j \in \mathcal{M}$, representing the extent to which they are considered strategic. It takes as input the order history $\mathcal{H}$, the priors $\phi \in (0, 1)$ and $\psi \in (0, 1)$ on agent types, the defect probabilities $\alpha, \beta, \gamma \in (0, 1)$ for customers, drivers, and other factors, respectively.

The algorithm relies on the following notation. Let N_i denote the set of drivers with whom customer i has interacted, and N_j be set of customers with whom driver j has interacted. Additionally, for each order of this pair of agents $\ell \in \{1, 2, \ldots, T_{i,j}\}$, $h_{i,j,\ell} \in \{0, 1\}$ denotes the outcome of the order, where $h_{i,j,\ell} = 1$ indicates that the order is reported missing by the customer. Denote the *likelihood* (based on the parameters assumed by the algorithm) of observing the outcome $h_{i,j,\ell}$ given customer type $x \in [0, 1]$ and driver type $y \in [0, 1]$ as

$$L_{i,j,\ell}^{(x,y)}(h_{i,j,\ell}) \triangleq \mathbb{P}\big(h_{i,j,\ell} \,\big|\, u_i = x, v_j = y\big) = \begin{cases} 1 - (1 - \alpha x)(1 - \beta y)(1 - \gamma), & \text{if } h_{i,j,\ell} = 1, \\ (1 - \alpha x)(1 - \beta y)(1 - \gamma), & \text{if } h_{i,j,\ell} = 0. \end{cases} \tag{7}$$

When $h_{i,j,\ell}$ is clear from the context, we drop the argument and write $L_{i,j,\ell}^{(x,y)}$ for simplicity.

For each customer $i \in \mathcal{N}$, each driver $j \in \mathcal{M}$, and each of their orders $\ell \in \{1, 2, \ldots, T_{i,j}\}$, we denote the *messages* from customer i to driver j (and vice versa) corresponding to order ℓ as

$$Z_{j\leftarrow i,\ell}^{(t)} \triangleq \log \frac{\mathbb{E}_{x_i \sim p_{i,-j}^{(t-1)}} L_{i,j,\ell}^{(x_i,1)}}{\mathbb{E}_{x_i \sim p_{i,-j}^{(t-1)}} L_{i,j,\ell}^{(x_i,0)}}, \quad Z_{i\leftarrow j,\ell}^{(t)} \triangleq \log \frac{\mathbb{E}_{y_j \sim q_{j,-i}^{(t)}} L_{i,j,\ell}^{(1,y_j)}}{\mathbb{E}_{y_j \sim q_{j,-i}^{(t)}} L_{i,j,\ell}^{(0,y_j)}}. \tag{8}$$

Intuitively, each message corresponds to the log of the ratio of the expected likelihoods of observing the realized outcome of the order $h_{i,j,\ell}$, where the expectation is taken over the type of the message sender based on the algorithm's current belief of the sender's type.

Algorithm 1 A Belief Propagation (BP) Algorithm

Input: $\mathcal{H}$, ϕ, ψ, α, β, γ, $t_{\max}$ and $\varepsilon_{\text{term}}$

Output: $(p_i)_{i\in\mathcal{N}}$, $(q_j)_{j\in\mathcal{M}}$

1: Initialize $p_{i,-j}^{(0)} = \phi$ for all $i \in \mathcal{N}$ and $j \in N_i$

2: **for** $t = 1, 2, \ldots, t_{\max}$ **do**

3: For all drivers $j \in \mathcal{M}$, update the driver scores:

$$q_j^{(t)} = \text{logit}^{-1}\left(\text{logit}\,\psi + \sum_{i\in N_j} \sum_{\ell=1}^{T_{i,j}} Z_{j\leftarrow i,\ell}^{(t)}\right)$$

$$q_{j,-k}^{(t)} = \text{logit}^{-1}\left(\text{logit}\,\psi + \sum_{i\in N_j\setminus k} \sum_{\ell=1}^{T_{i,j}} Z_{j\leftarrow i,\ell}^{(t)}\right) \text{ for all } k \in N_j$$

4: For all customers $i \in \mathcal{N}$, update the customer scores:

$$p_i^{(t)} = \text{logit}^{-1}\left(\text{logit}\,\phi + \sum_{j\in N_i} \sum_{\ell=1}^{T_{i,j}} Z_{i\leftarrow j,\ell}^{(t)}\right)$$

$$p_{i,-k}^{(t)} = \text{logit}^{-1}\left(\text{logit}\,\phi + \sum_{j\in N_i\setminus k} \sum_{\ell=1}^{T_{i,j}} Z_{i\leftarrow j,\ell}^{(t)}\right) \text{ for all } k \in N_i$$

5: Break if $t \geq 2$ and $\frac{1}{n}\sum_i (p_i^{(t)} - p_i^{(t-1)})^2 < \varepsilon_{\text{term}}$ and $\frac{1}{m}\sum_j (q_j^{(t)} - q_j^{(t-1)})^2 < \varepsilon_{\text{term}}$

6: **end for**

The algorithm initializes the customer scores $\boldsymbol{p}^{(0)}$ at the prior, and alternates between updating the driver scores with current customer scores, and updating the customer scores with current driver scores. The updates are based on the expected likelihood ratio of the observed data given the agent types, where the likelihood term in lines 3 and 4 of Algorithm 1 is defined in (5).

Specifically, line 3 calculates $q_{j,-k}^{(t)}$, the score of driver j excluding the information from interaction with customer k. It depends on the prior ψ that the driver is strategic, as well as, for each customer $i \in N_j \setminus k$, the ratio between the expected likelihood assuming the customer is strategic and the expected likelihood assuming the customer is straightforward. The larger the ratio, the more likely the driver is strategic. The expectation is taken over the customer type x_i,

which is assumed to follow a Bernoulli distribution $\mathrm{Ber}(p_{i,-j}^{(t-1)})$ parameterized by the current customer scores calculated in the previous iteration (excluding the information from driver j). From an information perspective, the score of j gets information from the edge (i, j) and the score of i calculated without information from (i, j) or j itself. This "tell your neighbor what everyone else told you" rule prevents double counting and yields exact dynamic programming on trees; on loopy graphs it's the same idea applied approximately. The customer score update in line 4 is similar, but with the roles of customers and drivers reversed. The algorithm terminates when the mean squared change in the customer scores and driver scores falls below a threshold $\varepsilon_{\text{term}}$, or after a fixed number of iterations $t_{\max}$.

Conceptually, belief propagation is a fixed point iteration algorithm whose fixed points correspond to stationary points of an objective function called the Bethe free energy [Yedidia et al., 2003]. This objective function approximates the Kullback-Leibler (KL) divergence between an approximated distribution and the true posterior distribution over agent types given the observed data. The approximation is exact when the underlying interaction graph is a tree, and the algorithm converges to the true posterior marginals in that case. When the interaction graph has cycles, the algorithm is not guaranteed to converge, and even if it does, it may not converge to the true posterior marginals. Nevertheless, it often works well in practice, and in our numerical experiments, it always converges quickly to a solution that performs well in identifying strategic agents.

3.1 Optimality of the Algorithm

In practice, each driver typically has many more orders than each customer. In this subsection, we prove that with a large number of orders per driver, Algorithm 1 correctly recovers the driver types and achieves the highest statistical power on the customer side.

Oracle Customer Scores. For each customer $i \in \mathcal{N}$, we denote $p_i^\star$ as a sigmoid transformation of the log-likelihood ratio given the set of orders between customer i and each driver $j \in \mathcal{M}$ and the *true* driver types $(v_j)_{j\in\mathcal{M}}$:

$$p_i^\star \triangleq \operatorname{logit}^{-1}\left(\operatorname{logit}\phi + \sum_{j\in N_i}\sum_{\ell=1}^{T_{i,j}} \log \frac{L_{i,j,\ell}^{(1,v_j)}}{L_{i,j,\ell}^{(0,v_j)}}\right). \tag{9}$$

By Neyman-Pearson lemma [Lehmann and Romano, 2005], any (strictly) monotonic transformation of the likelihood ratio is a uniformly most powerful statistic, which maximizes the true positive rate given any false positive rate. In other words, customer scores $(p_i^\star)_{i\in\mathcal{N}}$ maximizes the area under the ROC curve (AUC). Compared to $p_i^\star$ in (9), the customer scores computed in line 4 of Algorithm 1 replace the true driver types with a distribution over driver types based on current beliefs $\boldsymbol{q}^{(t)}$ and then take expectation, since the true driver types are unknown.

Assumptions. Our theoretical results rely on the following assumptions. Note that Algorithm 1 is well specified and can be applied without any of these assumptions.
(A1) Matching between customers and drivers is independent of agent types.

(A2) Each pair of customer and driver has at most one order.[6]
(A3) Algorithm 1 is specified with correct defect probabilities α, β and γ, and priors ϕ, ψ.

Theorem 1 (Optimality). *Assume (A1)–(A3).*[7] *For each driver $j \in \mathcal{M}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the first-iteration driver score is close to the true type:*

$$|q_j^{(1)} - v_j| \le \exp(-C_1 T_{\cdot,j} + C_0\sqrt{T_{\cdot,j}\log(1/\delta)} + C_2),$$

where $T_{\cdot,j}$ is the number of orders for driver j, and C_0, C_1, and C_2 are positive constants. For each customer $i \in \mathcal{N}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the first-iteration customer score is close to the oracle score in (9):

$$|p_i^{(1)} - p_i^\star| \le \exp(-C_1(d-1) + C_0\sqrt{(d-1)\log(T_{i,\cdot}/\delta)} + \log T_{i,\cdot} + C_3),$$

where $d = \min_{j\in N_i} T_{\cdot,j}$, $T_{i,\cdot}$ is the number of orders for customer i, and C_3 is another constant.

Note that the customer side error bound diminishes exponentially with d, the number of orders per *driver*. This reflects the information flow in Belief Propagation: accurate identification of driver types allows the algorithm to re-evaluate customer interactions with high confidence, effectively increasing the signal-to-noise ratio for the customer. This makes the bound meaningful in practice, since each driver typically has many more orders than each customer. If the number of orders per customer $T_{i,\cdot}$ is large, then both $p_i^{(1)}$ and $p_i^\star$ would be close to the true type of customer i, with an error depending on $T_{i,\cdot}$. See Section 4.1.2 for justification of the asymmetric bounds via experimental results.

We provide the full proof of this theorem in Appendix A.2. Briefly, we first prove a high probability error bound for the driver scores in the first step of Algorithm 1 using Hoeffding's inequality. Then, we pass this error to the second step to bound the error of the customer scores.

3.2 Theoretical False Positive Rate

To show the superiority of our algorithm over the naïve approach, we analyze the false positive rate (FPR) of both algorithms, with a large number of orders per agent for both customers and drivers.

In our algorithm, each iteration consists of two steps, updating driver and customer side scores respectively. In the following statement, we compare the performance of three algorithms: *naïve*, which uses only the historical report rates of each agent to score them; *oracle*, which uses the true types of agents on the other side to score agents on one side (given in (9) and denoted by a star $\star$); and *BP(1)*, which is our algorithm after one iteration (i.e., one step of driver score update

[6] This approximately holds in practice when there are millions of customers and hundreds of thousands of drivers in a market. In the data from our industry collaborator discussed in more detail in Section 4.1, over the course of half a year, less than 3% of all orders occur between customer and driver pairs who have more than 1 order with each other.

[7] This assumption can be relaxed to a condition on both the consumed parameters $\alpha, \beta, \gamma, \phi$ and the true parameters $\alpha^*, \beta^*, \gamma^*, \phi^*$. Specifically, we need $\mu_1(\phi^*, \phi) > 0$ and $\mu_0(\phi^*, \phi) < 0$, where $\mu_y(\phi^*, p)$ is defined in (18), (11), and (12). Our proof allows the weaker assumption, but for ease of exposition we present the stronger assumption here.

and one step of customer score update). We show that as the number of interactions per agent grows, BP(1) approaches oracle, and both outperform naive, in terms of the log false positive rate under any true positive rate in $(0, 1)$.

Since, we are assuming that BP is specified with true parameters (A3), the probability that an order between a customer of type x and a driver of type y gets reported is $\rho^{(x,y)} = 1-(1-\alpha x)(1-\beta y)(1-\gamma)$. For $p, q \in [0, 1]$, since $\rho^{(\cdot,\cdot)}$ is bilinear, we have $\rho^{(p,q)} = \mathbb{E}_{x\sim p,\, y\sim q}\, \rho^{(x,y)}$. We denote the Kullback–Leibler (KL) divergence between two Bernoulli distributions with parameters p_1 and p_2 as $D_{\mathrm{KL}}(p_1 \,\|\, p_2)$.

Theorem 2. *Assume (A1)–(A3), and that each agent interacts with $d > 0$ other agents.*[8] *The customer scores of the three algorithms satisfy the following property. For any fixed target true positive rate (TPR) in $(0, 1)$,*

$$\lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{naive} = c_-, \quad \lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{\star} = c_+, \quad \lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{BP(1)} \geq c_+,$$

where $c_- = D_{\mathrm{KL}}\left(\rho^{(1,\psi)} \,\middle\|\, \rho^{(0,\psi)}\right)$, $c_+ = \mathbb{E}_{y\sim\psi}\, D_{\mathrm{KL}}\left(\rho^{(1,y)} \,\middle\|\, \rho^{(0,y)}\right)$, *and* $c_+ > c_-$.

A direct consequence of this theorem is that the gap between naïve and BP(1) is exponentially large:

$$\lim_{d\to\infty} \frac{1}{d}\log \frac{\mathrm{FPR}^{\mathrm{naive}}}{\mathrm{FPR}^{\mathrm{BP(1)}}} \geq c_+ - c_- > 0. \tag{10}$$

We provide the proof in Section A.3. Briefly, for each algorithm, we characterize the relationship between TPR and the classification threshold via quantile convergence, and then use large deviation theory to analyze the FPR at the threshold.

Since the assumptions of Theorem 2 are symmetric with respect to customers and drivers, the same results hold for driver scores as well. To be precise, after one step of customer score update and one step of driver score update, the driver scores will achieve similar limit performance, with constants c_- and c_+ defined symmetrically.

In Section 4.1.1, we provide simulation results that verify the theoretical findings. The TPRs and FPRs in the theorem are population quantities. Empirical TPRs and FPRs, calculated on a finite number of agents, will have additional sampling noise. Nonetheless, with a large number of agents, the empirical FPRs will be close to the population FPRs, and we observe an exponential gap between the empirical FPRs of BP and naïve in the simulations in Figure 4.

In Section 2, we have pointed out two weaknesses of the naïve approach: it does not properly handle agents with different number of interactions, and it does not utilize the information from agents on the other side. Theorem 2 shows that even if we ignore the first weakness (by assuming each agent interacts with the same number of agents), the second weakness alone causes an exponentially large gap in FPR between naïve and BP(1). The consequence of the first weakness depends on the distribution of the number of interactions per agent, and is more difficult to analyze theoretically. The simulation results in Section 4.1.1 suggest that the first weakness causes

[8] For simplicity, we consider a symmetric setting where $n = m$ and the graph is d-regular on both sides. The proof can be extended to a graph that is d_{c}-regular on the customer side and d_{d}-regular on the driver side, with minor modifications.

the FPR of the naïve approach to decrease slower than exponentially, while BP(1) still enjoys an exponential decrease in FPR, thus further enlarging the gap between the two algorithms.

4 Experimental Results

In this section, we show via extensive experiments that our proposed BP algorithm achieves substantial improvements in accuracy on both synthetic data and data from our industry collaborator. Additionally, the BP algorithm is fast to converge and robust to parameter mis-specification.

4.1 Synthetic Simulations

We first present the results on synthetic data. The benefit of using synthetic data is that with access to the ground truth agent types and the data generation process, we are able to evaluate the accuracy of the identification of strategic customers and drivers, and demonstrate the robustness of our algorithm.

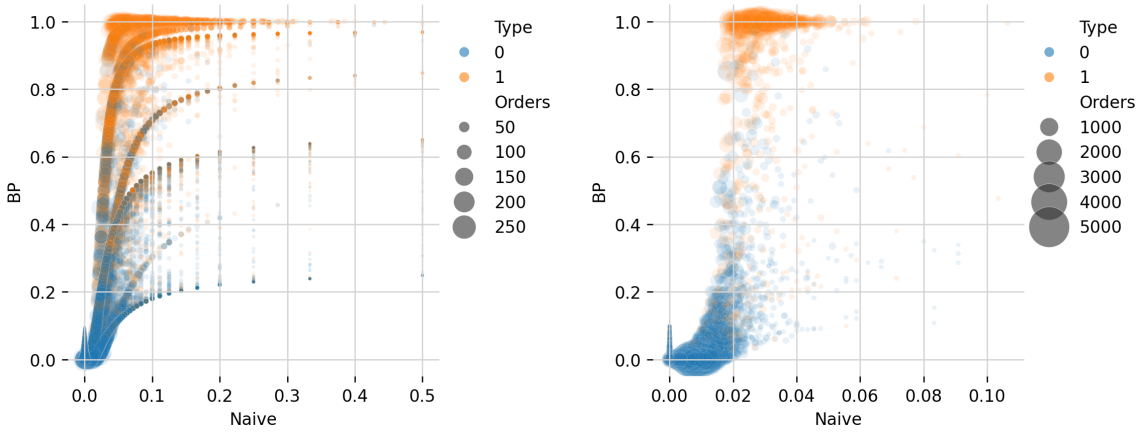
The Economy. We consider an economy with a total of $n = 300\,000$ customers and $m = 10\,000$ drivers. The total number of orders is 5 million, meaning that on average each customer has 16.7 orders and each driver has 500 orders. The numbers of orders per customer and driver, respectively, follow i.i.d. geometric distributions (Figure 9 in Appendix B.1 illustrates the distributions of order counts on each side of the market). Customer orders are matched to driver orders uniformly at random, as detailed in Appendix B.1, where we describe how we generate the matching.

We assume that overall 10% of all customers and drivers are strategic. The customer and driver types (i.e., whether an agent is strategic) are drawn from i.i.d. Bernoulli distributions with parameter $\phi = \psi = 0.1$. The defect probabilities for strategic customers, strategic drivers, and restaurants/neighbors are set to $\alpha = 0.05$, $\beta = 0.02$, and $\gamma = 0.003$, respectively. Figure 10 in Appendix B.1 provides the distribution of the realized refund request rates (i.e., the naive scores) for each side of the market.

Algorithms and Benchmarks. In this section, we compare BP and the naive approach.

- *BP:* The BP algorithm (Algorithm 1) with the true hyperparameters on the distribution of agent types and the defect probabilities on each side of the market.
- *BP-Mis:* The BP algorithm (Algorithm 1) with mis-specified but sensible hyperparameters derived from the data. Specifically, let ξ be the fraction of all orders that are reported. We set $\alpha = \beta = \phi = \psi = \sqrt{\xi/3}$ and $\gamma = \xi/3$. Intuitively, this "educated guess" assumes that the defects are equally attributed to the three sides of the market, and the fraction of strategic agents is equal to the defect rate of strategic agents. Given the simulation setting described above, the overall report rate ξ is around 1%, meaning that the BP-Mis algorithm takes $\alpha = \beta = \phi = \psi = 0.057$ and $\gamma = 0.0033$.
- *Naive:* The naïve approach defined in (1), evaluating each customer and each driver using the fraction of reports out of their past orders.

For both BP algorithms, we initialize the scores of each agent using their naïve scores from the data, and set the termination slack to be $\varepsilon_{\text{term}} = 10^{-5}$.



(a) Customer.

(b) Driver.

Figure 1: Comparison of agent-level naïve and BP scores. Each circle represents an agent. The color represents the true type of the agent and the size is proportional to the number of orders of the agent.

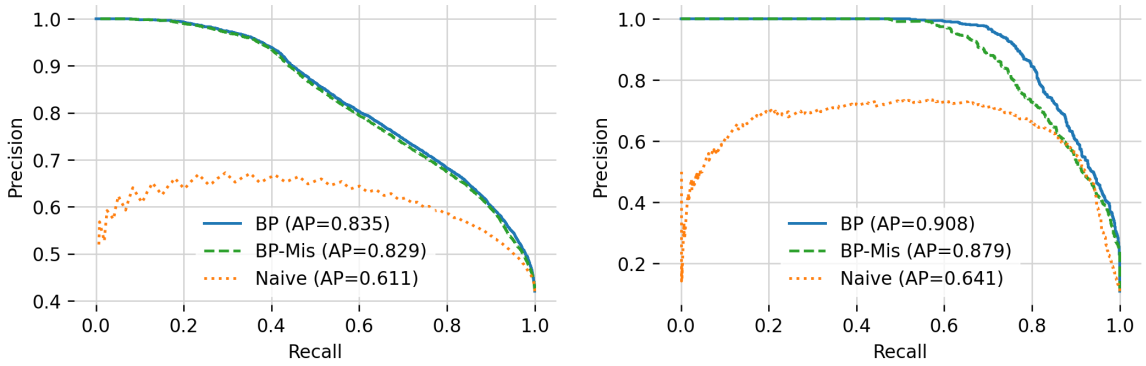
Note that in Appendix B.2, we also compare with additional benchmarks, including an algorithm analogous to the message passing algorithm analyzed in Karger et al. [2011], and a maximum likelihood estimation (MLE) with continuous type relaxation, intuitively estimating a defect probability of each agent. Neither substantially outperforms the naive benchmark.

Comparison of Individual Scores. Figure 1 compares agent-level scores under the Naive approach and the correctly specified BP algorithm. Each point corresponds to an agent, with the vertical axis and horizontal axis indicating the agent's BP and naive scores, respectively. Blue and orange dots represent straightforward and strategic agents, respectively, and the size of each dot is proportional to the order count of that agent. First note that among customers who share the same naive score (e.g., around 0.1), there is often a large spread in their BP scores. This arises from the number of orders each customer has, and the (estimated) trustworthiness of the drivers they interacted with. While the naive approach does not distinguish these customers, those with higher BP scores are indeed more likely to be strategic.

We also observe that many strategic customers with large number of orders cluster in the top-left corner of Figure 1a. These customers are not the first ones to be identified by the naive approach. Customers with few orders may end up with substantially higher naive scores due to the inherent randomness of the system, but the BP algorithm still flags them with confidence. Finally, since drivers tend to have more orders on average, the naive approach separates driver types better than it does for the customer-side. Nevertheless, we observe a clearer vertical separation compared to the horizontal separation.

Precision and Recall. Two metrics commonly used for evaluating binary classifiers are precision and recall.[9] In our context, given a set of scores and a particular threshold for each side of

[9]The precision-recall curves are more informative than ROC curves when (i) the number of positive samples is small and (ii) false positives are a significant concern, especially for the driver side.



(a) Customer.

(b) Driver.

Figure 2: Precision-recall curves for the different methods on agents with at least one request.

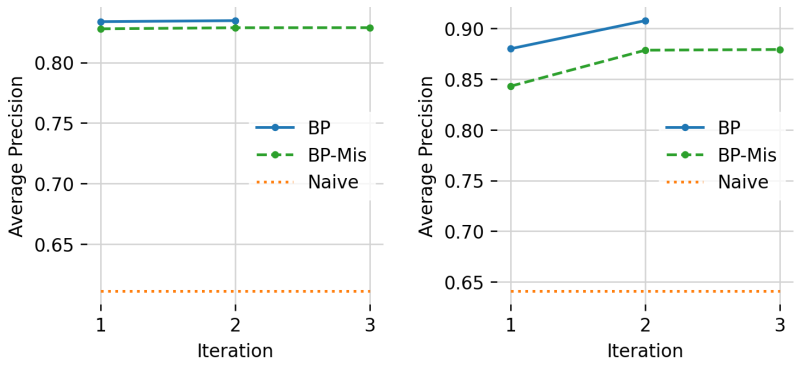
the market, *precision* is the fraction of flagged agents who are in fact strategic, while *recall* is the fraction of strategic agents in the population that are successfully flagged. As we vary the threshold, we obtain different precision-recall pairs, and the precision-recall curve shows the trade-off between precision and recall achieved at different thresholds. The average precision (AP), which is the precision integrated over all recall levels, is a summary statistic that quantifies the overall performance of a classifier.

Figure 2 plots the precision-recall curves achieved by each method. We include here only agents with at least one report, since in practice no platform will flag a customer or driver as strategic before any of their orders is reported missing/defective.[10] Overall, the BP algorithms achieve substantially higher precision than the naive approach. Moreover, BP-Mis is close to BP at most recall levels, suggesting that the BP algorithm is not sensitive to parameter misspecification. Since drivers have more orders in general, the precision on the driver side is higher under all methods in comparison to the customer side.

When recall is low (i.e., when the threshold is high, flagging only agents who seem the most suspicious), precision is close to 1 under the BP algorithms, meaning that almost all flagged agents are indeed strategic. In particular, the BP algorithms can successfully identify around 80% of all strategic drivers, without mis-labeling any straightforward drivers as strategic. Meanwhile, the precision under the naïve method never exceeds 70% on the customer side and 80% on the driver side. This implies that under the naive approach, even if the platform is only interested in identifying a small group of "most strategic" agents, at least 3 out of 10 flagged customers and 2 out of 10 flagged drivers are in fact straightforward.

It is also worth noting that as recall increases, the naïve method's precision first increases and then decreases, suggesting that drivers with the highest naïve scores are not the most strategic. This is because straightforward agents with few orders may end up with high report rates due to inherent randomness and defects on the other sides of the market, but agents with very large order counts are unlikely to have a very large fraction of their orders reported. Figure 12 in Appendix B.1 plots the number of orders per flagged agent across different recall levels. We see that naïve starts with agents with few orders, while those who are flagged first under BP have

[10]Figure 11 in Appendix B provides precision and recall curves that include all agents.



(a) Customer's average precision. (b) Driver's average precision.

Figure 3: Performance over iterations. The BP algorithms with true and mis-specified parameters terminated after 2 and 3 iterations, respectively.

very high order counts on average.

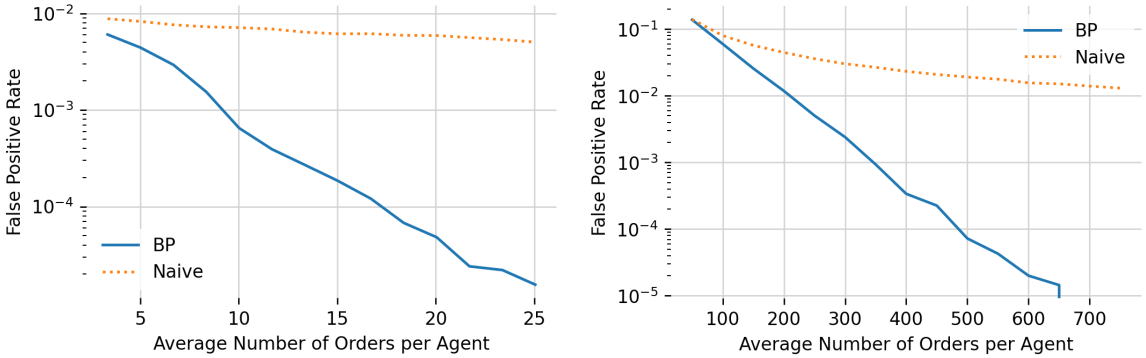
Performance over Iterations. Figure 3 shows the performance over iterations of the belief propagation algorithm. After just one iteration, the average precision is already much higher than that of the naïve approach. The remaining one or two iterations further improve the performance, but the marginal improvement is small.

4.1.1 Validation of Asymptotic False Positive Rate

Theorem 2 suggests that the false positive rate (FPR) decreases exponentially with the number of orders per agent, at a fixed true positive rate (TPR), for both the BP and naive algorithms. While Theorem 2 assumes that each customer/driver has the same number of orders, here we evaluate it on economies where the number of orders per customer/driver follows a geometric distribution, as in Section 4.1. This setting introduces realistic degree heterogeneity that makes the problem more challenging. We show that although the naive algorithm's FPR degrades from exponential to sub-exponential decay, the BP algorithm still enjoys exponential decay in FPR as the average number of orders per agent increases.

Setup. We simulate an economy with the same hyperparameters as in Section 4.1, while fixing the total number of orders to be 1×10^8 and the ratio of customers to drivers to be 30. We vary the number of customers and drivers simultaneously to achieve different average order counts per agent on both sides of the market.

Results. Figure 4 plots the FPR at a fixed TPR against the average number of orders per agent. The BP (blue) algorithm exhibits a linear relationship between log FPR and number of orders per agent, as predicted by Theorem 2. The naive (orange dotted) algorithm exhibits a sub-linear relationship especially on the driver side, which is worse than the theoretical prediction in Theorem 2. The variation in order counts makes the gap between the two algorithms even larger than what Theorem 2 predicts. Intuitively, the naive algorithm does not take into account the number of orders of each agent, and this information should help improve performance.



(a) Customer FPR at TPR = 0.1.

(b) Driver FPR at TPR = 0.5.

Figure 4: False positive rate versus number of orders per agent, at a fixed true positive rate.

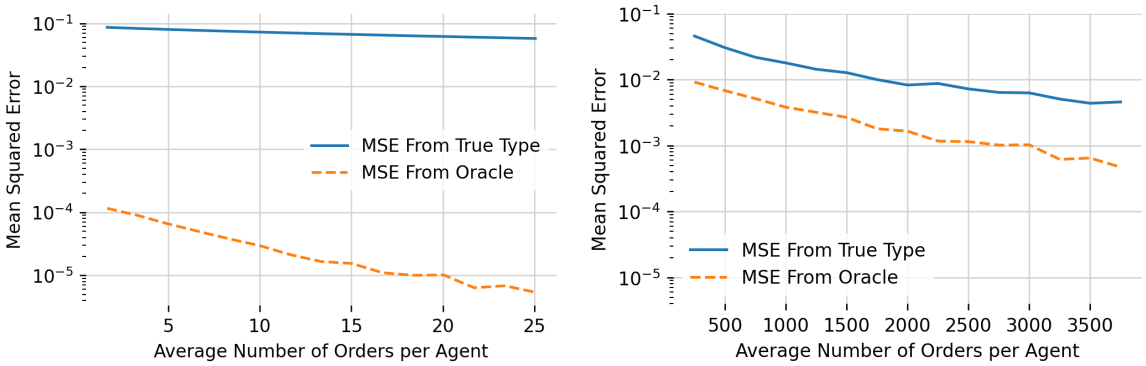
4.1.2 Justification for Asymmetric Bounds

A key feature of Theorem 1 is the different treatment of customers and drivers, bounding the error from the true type for drivers but the error from an oracle estimator for customers. While our mathematical model in Section 2 is structurally symmetric between customers and drivers, in practice, since customers typically have far fewer orders than drivers, the performance characteristics of the two sides of the market differ significantly.

Setup. We simulate an economy with the same hyperparameters as in Section 4.1, while fixing the total number of orders to be 5×10^7 and the ratio of customers to drivers to be 150. We vary the number of customers and drivers simultaneously to achieve different average order counts per agent on both sides of the market.

Figure 5 plots the mean squared error (MSE) of the BP algorithm against the average number of orders per agent. The blue curves represent the MSE from the true type, while the orange dashed curves represent the MSE from the oracle estimator. Specifically, on the customer side, we plot $\frac{1}{n}\sum_{i=1}^{n}(p_i - u_i)^2$ and $\frac{1}{n}\sum_{i=1}^{n}(p_i - p_i^\star)^2$, where p_i is the BP score, u_i is the true type, and $p_i^\star$ is the oracle estimator (9) (the optimal estimator given knowledge of the drivers' types) for customer i. On the driver side, we plot $\frac{1}{m}\sum_{j=1}^{m}(q_j - v_j)^2$ and $\frac{1}{m}\sum_{j=1}^{m}(q_j - q_j^\star)^2$, where q_j is the BP score, v_j is the true type, and $q_j^\star$ is the oracle estimator for driver j, defined analogously to (9).

Observations. As shown in Figure 5, the two curves exhibit different behaviors on the two sides of the market. On the driver side, the MSE from the true type (blue) and the MSE from the oracle estimator (orange dashed) are close to each other across different average order counts, and both decrease quickly as the average number of orders per driver increases. However, on the customer side, while the MSE from the oracle estimator (orange dashed) decreases quickly as the average number of orders per customer increases, the MSE from the true type (blue) remains relatively high. This empirical gap validates our theoretical focus. A bound on the MSE from the true type for customers would require unrealistically large average order counts per customer.



(a) Customer estimation error.

(b) Driver estimation error.

Figure 5: Mean squared error (MSE) of the BP algorithm from the true type and the oracle estimator, against the average number of orders per agent.

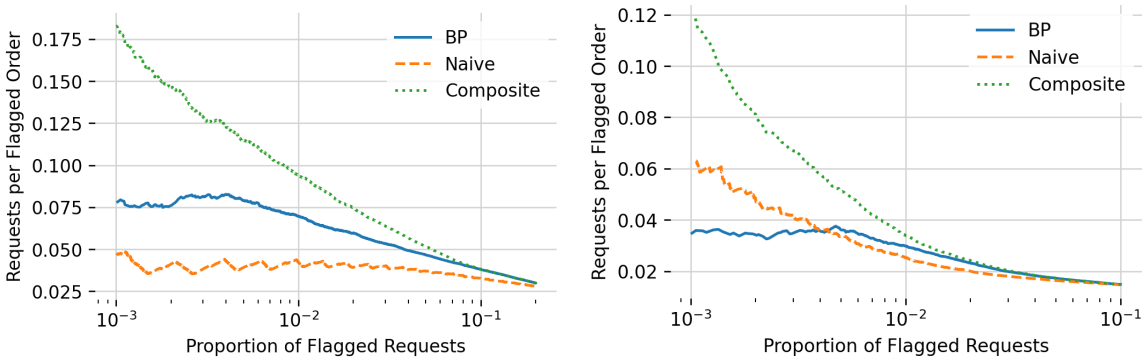
4.2 Results on Real Data

We now evaluate the performance of our algorithm on data from our industry collaborator, which is one of the largest platforms in Southeast Asia providing services including ridesharing and food delivery. Most results presented in this section are based on order data from a major city in the area (but we also include in Section 4.2.1 results on a second city with very different market dynamics, and the observations are very similar). From July 2021 to June 2023, there are a total of 164 million orders, involving 4.3 million customers and 146 thousand drivers. Figure 17 illustrates the order volume at different hours of the week. We can see clear peaks during lunch and dinner periods.

Overall, roughly 1.01% of all orders are reported by the customers. Customers may request refunds for many reasons, including missing entire orders, missing certain items in the orders, spillage and broken packaging, etc. We consider an order "reported" if the customer requested refund for any reason, and construct the order history $\mathcal{H}$ using the associated hashed customer ID, hashed driver ID, and hashed merchant ID of the orders. Figures 19 and 20 in Section C illustrate the distributions of order counts per customer and driver, as well as the distribution of naive scores.

Algorithms. We compare the performance of the BP algorithm, the naïve method, and a composite method that combines the two via a weighted average.

- *BP.* The BP algorithm (Algorithm 1) using the following hyperparameters: $\alpha = 0.05$, $\beta = 0.017$, $\phi = 0.066$, and $\psi = 0.2$. In addition, we allow the remaining defect probability γ to depend on the merchant. Specifically, for each merchant k, we set γ_k to be 1/3 of the merchant's naïve score.
- *Naive.* The naïve approach defined in (1).
- *Composite.* A hybrid method that takes a weighted average of the BP scores and the naïve scores, with weights 0.9 and 0.1, respectively.



(a) Customer.

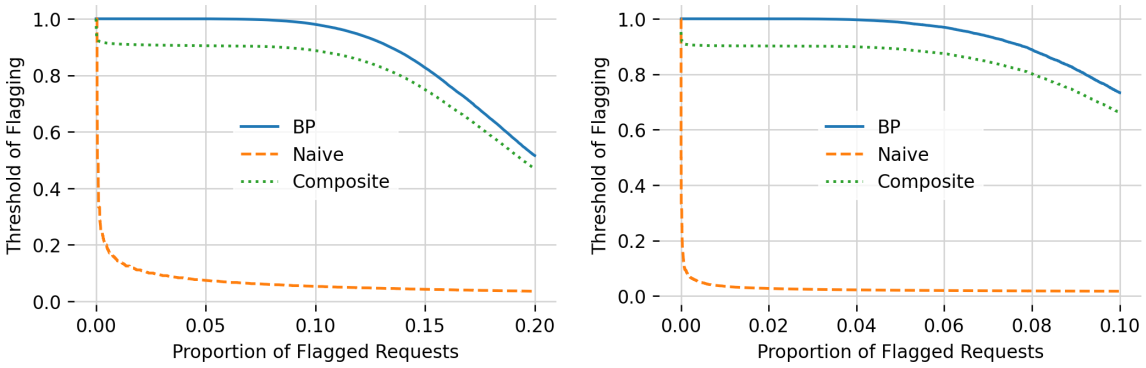
(b) Driver.

Figure 6: Surrogate precision-recall curves. A curve closer to the upper right corner is better. The horizontal axis represents the proportion of flagged requests (similar to recall), and the vertical axis represents the number of requests per flagged order (similar to precision).

Performance Evaluation Without Ground Truth. In practice, neither the ground truth labels (i.e., whether an agent is truly strategic) nor the "data generation model" is available. Therefore, to evaluate the performance of our method, we randomly split the data into a *training set* and a *testing set*, run our algorithms on the training set, and examine how well their outputs predict the observed data on the test set. Since whether each order is reported is observable in our data, our goal is to predict whether an agent's future (test set) orders will result in refund requests. Based on the predictions made on the training data, we flag certain agents as strategic. This flagged/unflagged status is then used to predict if an agent's order in the test set will result in a request. With this prediction goal in mind, we define two metrics, which can be interpreted as the recall and precision on the *order* level:

- *Proportion of Flagged Requests (Recall).* This metric is computed as the number of requests in the test set that are associated with agents flagged as strategic, divided by the total number of requests in the test set. In our new classification framework, every request in the test set represents a positive event. Thus, this metric measures the fraction of all positive events that we successfully predict by flagging the corresponding agents—exactly how recall is defined.
- *Requests per Flagged Order (Precision).* This metric is computed as the number of requests in the test set that are associated with flagged agents, divided by the number of orders in the test set associated with flagged agents. Here, each order from a flagged agent is treated as a positive prediction. The metric indicates the likelihood that a flagged agent's order actually results in a request. This is analogous to precision in standard classification, where precision is the fraction of predicted positives (flagged orders) that are true positives (orders that indeed generate a request).

In summary, rather than assessing performance using unobservable agents' types, we focus on predicting whether an agent's order will be associated with a request in the future. Under this alternative classification goal, the *Proportion of Flagged Requests* corresponds to recall (capturing the rate of true positives among all actual positives), and the *Requests per Flagged Order* corresponds to precision (measuring the accuracy of our flagged predictions).



(a) Customer.

(b) Driver.

Figure 7: Threshold of flagging vs. the proportion of flagged requests for customers and drivers.

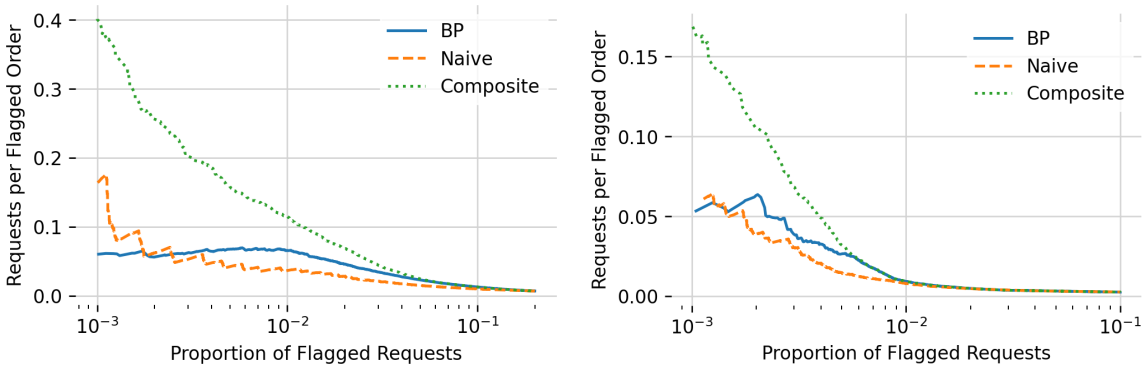
Results. Similar to precision-recall curves discussed in the previous subsection, different the classification thresholds will result in different pairs of these two metrics, and we plot the parametric curve to show the trade-off between the two metrics.

As shown in Figure 6, the BP algorithm outperforms the naïve method for customers and drivers over a wide range of thresholds. Moreover, the composite method, which combines the BP and naïve scores, substantially outperforms both the BP and naïve methods. This can be explained by the discrepancy between the assumed data generation model and the unknown, real data generation model. The current framework assumes a binary model for the user types, which can be continuous in practice. The BP algorithm only decides whether a user's defect probability is more likely to be the high support (i.e., α or β) or the low support (i.e., 0). Once there is overwhelming evidence ($> \alpha$), the score converges to 1, losing the ability to rank "super-strategic" agents with defect rates far exceeding α. As shown in Figure 7, the BP scores are 1 for a wide range of thresholds. In this case, the naïve score can provide additional information to break the ties. The composite method, which combines the BP and naïve scores, can leverage the strengths of both methods and outperform either method alone.

4.2.1 Results on a second city.

In addition to the results on the first city presented above, we include in this section results for the same algorithms on data from another major city in Southeast Asia. The two cities are from two different counties and have different market conditions in general. For example, the overall order report rate in this city is at 0.14%, much lower than the 1.01% defect rate observed for the first city. For this second city, we set the hyperparameters of the BP algorithm simply as $\alpha = \beta = \phi = \psi = \sqrt{\xi}$, where $\xi = 0.0014$ is the overall order report rate. Additionally, we set each merchant's defect probability γ_k to be 1/3 of the merchant's naïve score.

Figure 8 shows the order-level precision-recall curves for the second city on testing data, as apply varying thresholds to scores derived from the training data. The overall trends are similar to that in Figure 6. However, the customers with highest composite scores in the training data may in fact report as much as 40% of their orders in the testing data, much higher than the peak of around 17.5% for the first city. Intuitively, this is likely a result of the customer support agents



(a) Customer.

(b) Driver.

Figure 8: Precision-recall curves for identifying requested orders in the test set, using customer and driver scores, respectively, using data from a different city.

being more "lenient" with customers in the second city—with a substantially lower overall defect rate, support agents who are allocated similar refund and appeasement budget per order do not have to reject refund requests from customers regardless of their past behavior.

5 Concluding Remarks

In this work, we study the problem of identifying strategic customers and delivery drivers in the context of customer support, when the platform has no ground-truth information on which party is responsible for each reported missing or defective order. We propose a belief propagation (BP) algorithm for identifying strategic customers and drivers, which iteratively updates the belief on each agent's trustworthiness based on the realized order outcomes and the algorithms' current beliefs on the types of agents on the other side of the market. In large markets, we BP approximately recovers the correct types on the driver side, and achieves the highest statistical power on the customer side. Additionally, there is an exponentially large gap (with respect to the number of orders per participant) in the ratio of false positive rates between our algorithm and a naive benchmark that only relies only on the average past refund rate of each participant. Extensive experiments on both synthetic data and data from our industry collaborator—a major Southeast Asian platform—demonstrate that the BP algorithm provides substantial and robust accuracy improvements in comparison to naive benchmarks widely used in the industry.

References

R. R. Bahadur and R. Ranga Rao. On deviations of the sample mean. *The Annals of Mathematical Statistics*, 31(4):1015–1027, 1960. ISSN 00034851. URL http://www.jstor.org/stable/2237800.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6): 065701, 2011.

Gerhard H Fischer. On the existence and uniqueness of maximum-likelihood estimates in the rasch model. *Psychometrika*, 46:59–77, 1981.

Chao Gao and Dengyong Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *arXiv preprint arXiv:1310.5764*, 2013.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335, 2015.

David Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems*, 24, 2011.

Ashish Khetan and Sewoong Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4844–4852, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Erich Leo Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer, 2005.

Duc Nguyen and Anderson Ye Zhang. A spectral approach to item response theory. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Georg Rasch. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.

Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

M Wu, RL Davis, BW Domingue, C Piech, and ND Goodman. Variational item response theory: Fast, accurate, and expressive. international educational data mining society. 2020.

Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8(236–239):0018–9448, 2003.

Hedayat Zarkoob, Greg d'Eon, Lena Podina, and Kevin Leyton-Brown. Better peer grading through bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6137–6144, 2023.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *J. Mach. Learn. Res.*, 17:102:1–102:44, 2016. URL https://jmlr.org/papers/v17/14-511.html.

Appendices

We include in Appendix A proofs omitted from the body of the paper. Appendix B provides additional simulation results on synthetic data, including comparisons with two additional benchmarks algorithms designed for related settings. Appendix C provides descriptive statistics on market dynamics for cities studied in Section 4.2, as well as additional simulation results.

A Proofs

Throughout the theoretical analyses in this section, we assume at most one order per customer-driver pair. Hence, we will drop the order subscript ℓ from the message notation as it will be clear from context. Thus, the order history between each customer-driver pair (i, j) who has an order can be summarized by a binary variable $h_{i,j} \in \{0, 1\}$. Let $T_{i,\cdot}$ denote the total number of orders placed by each customer $i \in \mathcal{N}$, and let $T_{\cdot,j}$ to denote the number of orders accepted by each driver $j \in \mathcal{M}$.

For any set of defect probabilities $\alpha^*, \beta^*, \gamma^* \in (0, 1)$, and any $x, y \in [0, 1]$, denote $\rho_*^{(x,y)}$ as

$$\rho_*^{(x,y)} \triangleq 1 - (1 - \alpha^* x)(1 - \beta^* y)(1 - \gamma^*). \tag{11}$$

When the customer type is $x \in \{0, 1\}$ and the driver type is $y \in \{0, 1\}$, $\rho_*^{(x,y)}$ corresponds to the probability for an order between this pair of agents to be reported, but note that $\rho_*^{(x,y)}$ is well defined for any $x, y \in [0, 1]$. Similarly, given any parameters $\alpha, \beta, \gamma \in (0, 1)$ adopted by a BP algorithm, we define for any $x, y \in [0, 1]$

$$\rho^{(x,y)} \triangleq 1 - (1 - \alpha x)(1 - \beta y)(1 - \gamma). \tag{12}$$

A.1 Properties of the BP Messages

We first provide notation and establish a number of properties that we use in the proofs of our main theorems. The first result provides a convenient expression for the expectation of $\rho^{(x,y)}$ since it is bilinear in x and y.

Claim 1. *Given any $\alpha, \beta, \gamma \in (0, 1)$, and any $p, q \in [0, 1]$,*

$$\mathop{\mathbb{E}}_{x\sim p} \rho^{(x,y)} = \rho^{(p,y)},$$
$$\mathop{\mathbb{E}}_{y\sim q} \rho^{(x,y)} = \rho^{(x,q)}.$$

Proof. By definition of ρ (12),

$$\begin{aligned} 1 - \mathop{\mathbb{E}}_{x\sim p} \rho^{(x,y)} &= p(1 - \alpha)(1 - \beta y)(1 - \gamma) + (1 - p)(1 - \beta y)(1 - \gamma) \\ &= (1 - \beta y)(1 - \gamma)(p(1 - \alpha) + (1 - p)) \end{aligned}$$

$$= (1-\beta y)(1-\gamma)(1-\alpha p)$$
$$= 1 - \rho^{(p,y)}$$

The other equation can be proven in a symmetric way. □

Note that the same expressions also hold for $\rho_*^{(x,y)}$ as defined in (11). Intuitively, $\rho_*^{(p,v)}$ corresponds the probability that an order between a customer with current belief $p \in [0,1]$ of being strategic, and a driver with type $v \in \{0,1\}$, gets reported.

With Claim 1, we can simplify the expression of the message sent from a customer with current score p, when an order has a realized outcome $h \in \{0,1\}$.

Claim 2. *For an order with realized outcome $h \in \{0,1\}$, the message sent by a customer with score $p \in [0,\ 1]$ to a driver is of the form*

$$\log \frac{\mathbb{E}_{x\sim p}\,\mathbb{P}(h \mid u=x, v=1)}{\mathbb{E}_{x\sim p}\,\mathbb{P}(h \mid u=x, v=0)} = h \log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} + (1-h)\log \frac{1-\rho^{(p,1)}}{1-\rho^{(p,0)}}. \tag{13}$$

Similarly, the message sent by a driver with score $q \in [0,\ 1]$ is:

$$\log \frac{\mathbb{E}_{y\sim q}\,\mathbb{P}(h \mid u=1, v=y)}{\mathbb{E}_{y\sim q}\,\mathbb{P}(h \mid u=0, v=y)} = h \log \frac{\rho^{(1,q)}}{\rho^{(0,q)}} + (1-h)\log \frac{1-\rho^{(1,q)}}{1-\rho^{(0,q)}}. \tag{14}$$

Proof. Plugging in the expression of $\mathbb{P}(h \mid u, v)$, the likelihood of observing outcome h given customer and driver types u, v, we have:

$$\begin{aligned}
\log \frac{\mathbb{E}_{x\sim p}\,\mathbb{P}(h \mid u=x, v=1)}{\mathbb{E}_{x\sim p}\,\mathbb{P}(h \mid u=x, v=0)} &= \log \frac{\mathbb{E}_{x\sim p}(\rho^{(x,1)})^h(1-\rho^{(x,1)})^{(1-h)}}{\mathbb{E}_{x\sim p}(\rho^{(x,0)})^h(1-\rho^{(x,0)})^{(1-h)}} \\
&= \begin{cases} \log \dfrac{\mathbb{E}_{x\sim p}\,\rho^{(x,1)}}{\mathbb{E}_{x\sim p}\,\rho^{(x,0)}} & \text{for } h=1, \\ \log \dfrac{\mathbb{E}_{x\sim p}(1-\rho^{(x,1)})}{\mathbb{E}_{x\sim p}(1-\rho^{(x,0)})} & \text{for } h=0 \end{cases} \\
&= h \log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} + (1-h)\log \frac{1-\rho^{(p,1)}}{1-\rho^{(p,0)}}.
\end{aligned}$$

The last equality combines the two cases, and applies Claim 1. The other equation can be proven in a symmetric way. □

With Claim 2, we can establish the following bounds on messages in the BP algorithm.

Claim 3. *For any realized order outcome $h \in \{0,1\}$, any score $p, q \in [0,1]$ and any defect parameters $\alpha, \beta, \gamma \in (0,1)$, the message sent by a customer (13) is bounded by*

$$\left| h \log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} + (1-h)\log \frac{1-\rho^{(p,1)}}{1-\rho^{(p,0)}} \right| \le \max\left\{ \left| \log \frac{1-(1-\beta)(1-\gamma)}{\gamma} \right|, |\log(1-\beta)| \right\},$$

and the message sent by a driver (14) *is bounded by*

$$\left|h \log \frac{\rho^{(1,q)}}{\rho^{(0,q)}} + (1-h) \log \frac{1-\rho^{(1,q)}}{1-\rho^{(0,q)}}\right| \leq \max\left\{\left|\log \frac{1-(1-\alpha)(1-\gamma)}{\gamma}\right|, |\log(1-\alpha)|\right\}.$$

Proof. When $h = 1$, (13) is of the form

$$\log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} = \log \frac{1-(1-\alpha p)(1-\beta)(1-\gamma)}{1-(1-\alpha p)(1-\gamma)}.$$

Let $\xi = 1-\alpha p$. The term inside the logarithm is of the form $g(\xi) = \frac{1-A\xi}{1-B\xi}$ where $A = (1-\beta)(1-\gamma)$ and $B = 1-\gamma$. Since $\beta \in (0,1)$, we have $A < B$. The ratio is always positive, thus the maximum absolute value is achieved when $g(\xi)$ is maximized. The derivative $g'(\xi) = \frac{B-A}{(1-B\xi)^2}$ is positive, so the ratio is increasing in ξ, and maximized at $\xi = 1$ (in which case $p = 0$):

$$\log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} \leq \log \frac{1-(1-\beta)(1-\gamma)}{1-(1-\gamma)} = \log \frac{1-(1-\beta)(1-\gamma)}{\gamma}.$$

When $h = 0$, the message is

$$\log \frac{1-\rho^{(p,1)}}{1-\rho^{(p,0)}} = \log \frac{(1-\alpha p)(1-\beta)(1-\gamma)}{(1-\alpha p)(1-\gamma)} = \log(1-\beta),$$

which yields the second term in the maximum. The bound on the message sent by a driver can be proved in a symmetric way. □

For convenience of notation, we denote the upper bound on all messages as

$$C \triangleq \max\left\{\left|\log \frac{1-(1-\alpha)(1-\gamma)}{\gamma}\right|,\ |\log(1-\alpha)|,\ \left|\log \frac{1-(1-\beta)(1-\gamma)}{\gamma}\right|,\ |\log(1-\beta)|\right\}. \tag{15}$$

Recall that $\phi^* \in (0,1)$ and $\psi^* \in (0,1)$ corresponds to the prior probability for a customer and a driver to be strategic, respectively.

Claim 4. *Under (A1), for a driver with type $v \in \{0,1\}$, the expectation of the realized outcome h for any order of the driver is*

$$\mathop{\mathbb{E}}_{u\sim\phi^*} \rho_*^{(u,v)} = \rho_*^{(\phi^*,v)}. \tag{16}$$

Similarly, the expected outcome of an order for a customer of type $u \in \{0,1\}$ is given by:

$$\mathop{\mathbb{E}}_{v\sim\psi^*} \rho_*^{(u,v)} = \rho_*^{(u,\psi^*)}. \tag{17}$$

For any driver type $v \in \{0,1\}$, posterior belief on customer's type $p \in [0,1]$, and customer

side prior $\phi^* \in [0,1]$, denote

$$\mu_v(\phi^*, p) \triangleq \rho_*^{(\phi^*,v)} \log \frac{\rho^{(p,1)}}{\rho^{(p,0)}} + (1 - \rho_*^{(\phi^*,v)}) \log \frac{1-\rho^{(p,1)}}{1-\rho^{(p,0)}}. \tag{18}$$

We know from Claim 2 and Claim 4 that $\mu_v(\phi^*, p)$ represents the expectation of the message sent by a customer with current score p to a driver of type $v \in \{0,1\}$, when there is only one order between the customer and the driver. Note that $\rho_*^{(\phi^*,v)}$ depends on the true defect probabilities $\alpha^*, \beta^*, \gamma^*$, while $\rho^{(p,1)}$ and $\rho^{(p,0)}$ depend on the parameters α, β, and γ adopted by the BP algorithm.

The following result proves that when the BP algorithm is probably specified, in the first iteration where customer-side scores are initialized as $p = \phi$, the average message received from a customer by a driver of type 0 (or type 1) is negative (or positive), when there is only one order between this pair of driver and customer.

Claim 5. *For any $\alpha^*, \beta^*, \gamma^*, \phi^* \in (0,1)$, and when $\alpha = \alpha^*$, $\beta = \beta^*$, $\gamma = \gamma^*$ and $p = \phi = \phi^*$, we have $\mu_0(\phi^*, p) < 0$ and $\mu_1(\phi^*, p) > 0$.*

Proof. When $p = \phi^*$, we have $\rho_*^{(\phi^*,y)} = \rho_*^{(p,y)} = \rho^{(p,y)}$. By definition of the KL divergence between two Bernoulli random variables, it is straightforward to verify that

$$\mu_1(\phi^*, p) = D_{\mathrm{KL}}\left(\rho^{(p,1)} \middle\| \rho^{(p,0)}\right) > 0,$$
$$\mu_0(\phi^*, p) = -D_{\mathrm{KL}}\left(\rho^{(p,0)} \middle\| \rho^{(p,1)}\right) < 0.$$

The KL divergence is strictly positive when the parameters are different, and we know from (12) that $\rho^{(p,1)} \neq \rho^{(p,0)}$ when $\alpha, \beta, \gamma \in (0,1)$. □

A.2 Proof for Theorem 1

Theorem 1 (Optimality). *Assume (A1)–(A3).*[11] *For each driver $j \in \mathcal{M}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the first-iteration driver score is close to the true type:*

$$|q_j^{(1)} - v_j| \leq \exp(-C_1 T_{\cdot,j} + C_0 \sqrt{T_{\cdot,j} \log(1/\delta)} + C_2),$$

where $T_{\cdot,j}$ is the number of orders for driver j, and C_0, C_1, and C_2 are positive constants. For each customer $i \in \mathcal{N}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the first-iteration customer score is close to the oracle score in (9):

$$|p_i^{(1)} - p_i^\star| \leq \exp(-C_1(d-1) + C_0 \sqrt{(d-1)\log(T_{i,\cdot}/\delta)} + \log T_{i,\cdot} + C_3),$$

where $d = \min_{j \in N_i} T_{\cdot,j}$, $T_{i,\cdot}$ is the number of orders for customer i, and C_3 is another constant.

[11] This assumption can be relaxed to a condition on both the consumed parameters $\alpha, \beta, \gamma, \phi$ and the true parameters $\alpha^*, \beta^*, \gamma^*, \phi^*$. Specifically, we need $\mu_1(\phi^*, \phi) > 0$ and $\mu_0(\phi^*, \phi) < 0$, where $\mu_y(\phi^*, p)$ is defined in (18), (11), and (12). Our proof allows the weaker assumption, but for ease of exposition we present the stronger assumption here.

In order to prove this theorem, we first prove a bound on the first iteration driver scores, then bound the first iteration customer scores, and finally arrange these bounds together to get the final result.

A.2.1 Driver Scores

Recall that in Algorithm 1, the iteration 0 customer side score $p^{(0)}_{i,-j}$ is initialized as ϕ for all $i \in \mathcal{N}$ and all $j \in \mathcal{M}$. Define

$$c_1 \triangleq \min\{-\mu_0(\phi^*, \phi),\ \mu_1(\phi^*, \phi)\},$$

we know from Claim 5 that $c_1 > 0$ when $\phi = \phi^*$. We will assume $c_1 > 0$ for the parameters used by the algorithm.

Claim 6. *Assume that there is at most one order per pair of driver and customer, and that $c_1 > 0$. For all $\epsilon \in (0, c_1)$, with probability at least $1 - \exp(-c_0(T_{\cdot,j} - 1)\epsilon^2)$, we have*

$$\left|q^{(1)}_{j,-k} - v_j\right| \le \exp\big(-(T_{\cdot,j} - 1)(c_1 - \epsilon) + \operatorname{logit}\psi\big), \tag{19}$$

and we always have $\left|q^{(1)}_{j,-k} - v_j\right| \le 1$. Similarly, we have, with probability at least $1 - \exp(-c_0 T_{\cdot,j}\epsilon^2)$, we have

$$\left|q^{(1)}_{j} - v_j\right| \le \exp\big(-T_{\cdot,j}(c_1 - \epsilon) + \operatorname{logit}\psi\big), \tag{20}$$

and we always have $\left|q^{(1)}_{j} - v_j\right| \le 1$.

Proof. Recall that we initialize $p^{(0)}_{i,-j}$ with the prior ϕ for all i, j. In the first iteration of the *BP* algorithm, the message sent by customer i to driver j is of the form (recall that we only have 1 order per pair)

$$Z^{(1)}_{j\leftarrow i} = \log \frac{\mathbb{E}_{x_i\sim\phi}\, L^{(x_i,1)}_{i,j,\ell}}{\mathbb{E}_{x_i\sim\phi}\, L^{(x_i,0)}_{i,j,\ell}} = \log \frac{\mathbb{E}_{x_i\sim\phi}\, \mathbb{P}\big(h_{i,j} \,\big|\, u_i = x_i, v_j = 1\big)}{\mathbb{E}_{x_i\sim\phi}\, \mathbb{P}\big(h_{i,j} \,\big|\, u_i = x_i, v_j = 0\big)}.$$

Applying Claim 2, we know

$$Z^{(1)}_{j\leftarrow i} = \left(h_{i,j} \log \frac{\rho^{(\phi,1)}}{\rho^{(\phi,0)}} + (1 - h_{i,j}) \log \frac{1 - \rho^{(\phi,1)}}{1 - \rho^{(\phi,0)}}\right).$$

Taking the expectation of the messages with respect to the randomness in customer types, matching, and the realization of order reports, using Claim 4 we get

$$\begin{aligned}\mathbb{E}\left[Z^{(1)}_{j\leftarrow i} \,\middle|\, v_j\right] &= \mathbb{E}\big[h_{i,j}\big] \log \frac{\rho^{(\phi,1)}}{\rho^{(\phi,0)}} + (1 - \mathbb{E}\big[h_{i,j}\big]) \log \frac{1 - \rho^{(\phi,1)}}{1 - \rho^{(\phi,0)}} \\ &= \mu_{v_j}(\phi^*, \phi).\end{aligned}$$

where $\mu_{v_j}(\phi^*, \phi)$ is defined in Equation (18).

By Hoeffding's inequality, for $\epsilon > 0$,

$$\mathbb{P}\left(\sum_{i \in N_j \backslash k} \left(Z^{(1)}_{j \leftarrow i} - \mu_{v_j}(\phi^*, \phi) \right) > (T_{\cdot,j} - 1)\epsilon \,\middle|\, v_j \right) \leq \exp\left(-c_0 (T_{\cdot,j} - 1)\epsilon^2\right) \tag{21}$$

for some constant $c_0 > 0$. With probability at least $1 - \exp(-c_0(T_{\cdot,j} - 1)\epsilon^2)$, we have

$$\sum_{i \in N_j \backslash k} \left(Z^{(1)}_{j \leftarrow i} - \mu_{v_j}(\phi^*, \phi) \right) \leq (T_{\cdot,j} - 1)\epsilon.$$

For $v_j = 0$,

$$\log q^{(1)}_{j,-k} \leq \operatorname{logit} q^{(1)}_{j,-k} \leq \operatorname{logit} \psi + (T_{\cdot,j} - 1)\mu_0(\phi, p^{(0)}_{i,-j}) + (T_{\cdot,j} - 1)\epsilon,$$

where the first inequality follows because $q^{(1)}_{j,-k} \in [0, 1]$ and the second inequality follows from the definition of the BP updates. This implies

$$q^{(1)}_{j,-k} \leq \exp\left(\operatorname{logit} \psi + (T_{\cdot,j} - 1)\mu_0(\phi, p^{(0)}_{i,-j}) + (T_{\cdot,j} - 1)\epsilon \right).$$

Similarly, for $v_j = 1$,

$$-\log(1 - q^{(1)}_{j,-k}) \geq \operatorname{logit} q^{(1)}_{j,-k} \geq \operatorname{logit} \psi + (T_{\cdot,j} - 1)\mu_1(\phi, p^{(0)}_{i,-j}) - (T_{\cdot,j} - 1)\epsilon,$$

which implies

$$1 - q^{(1)}_{j,-k} \leq \exp\left(-\operatorname{logit} \psi - (T_{\cdot,j} - 1)\mu_1(\phi, p^{(0)}_{i,-j}) + (T_{\cdot,j} - 1)\epsilon \right).$$

Recall that $c_1 = \min\left\{ -\mu_0(\phi^*, \phi), \mu_1(\phi^*, \phi) \right\} > 0$. With probability at least $1 - \exp(-c_0(T_{\cdot,j} - 1)\epsilon^2)$, we have

$$\left| q^{(1)}_{j,-k} - v_j \right| \leq \exp\left(-(T_{\cdot,j} - 1)(c_1 - \epsilon) + \operatorname{logit} \psi \right), \tag{22}$$

and we always have $\left| q^{(1)}_{j,-k} - v_j \right| \leq 1$.

The proof of the bound $\left| q^{(1)}_{j} - v_j \right|$ follows the same argument. This completes the proof of this claim. □

We also have the following corollary of the above claim:

Corollary 1. *Under the setting of Claim 6, for some constants $c_2, c_3 > 0$,*

$$\mathbb{E}\left| q^{(1)}_{j,-k} - v_j \right| \leq c_3 \exp(-c_2 T_{\cdot,j}). \tag{23}$$

Proof. Taking expectation over the randomness in the messages (including randomness in the

types of customers, the matching between the driver and the customers, and the outcomes of orders) in Claim 6, and using the fact that $\left|q^{(1)}_{j,-k} - v_j\right| \le 1$ we get

$$\mathbb{E}\left|q^{(1)}_{j,-k} - v_j\right| \le \exp\big(-(T_{\cdot,j}-1)(c_1-\epsilon) + \operatorname{logit}\psi\big) + \exp(-c_0(T_{\cdot,j}-1)\epsilon^2).$$

Choosing $\epsilon = c_1/2$, and appropriate constants c_2 and c_3 gives the final result. □

A.2.2 Customer Scores

For each $i \in \mathcal{N}$, $j \in \mathcal{M}$, denote the "oracle" message $Z^{\star}_{i\leftarrow j}$ as the message sent from j to i in the hypothetical situation when the BP algorithm has access to the true driver types:

$$Z^{\star}_{i\leftarrow j} \triangleq h_{i,j} \log \frac{\rho^{(1,v_j)}}{\rho^{(0,v_j)}} + (1-h_{i,j}) \log \frac{1-\rho^{(1,v_j)}}{1-\rho^{(0,v_j)}}. \tag{24}$$

Claim 7. *Assuming that $\gamma > 0$, we can bound the difference between BP and oracle:*

$$\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right| \le c_4 \left|q^{(1)}_{j,-i} - v_j\right|$$

for some constant $c_4 > 0$.

Proof. Using the definition of $Z^{(1)}_{i\leftarrow j}$ and Claim 2, we have

$$Z^{(1)}_{i\leftarrow j} = h_{i,j} \log \frac{\rho^{(1,q^{(1)}_{j,-i})}}{\rho^{(0,q^{(1)}_{j,-i})}} + (1-h_{i,j}) \log \frac{1-\rho^{(1,q^{(1)}_{j,-i})}}{1-\rho^{(0,q^{(1)}_{j,-i})}}.$$

Taking the absolute difference between $Z^{(1)}_{i\leftarrow j}$ and $Z^{\star}_{i\leftarrow j}$ we get

$$\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right| = h_{i,j}\left|\log \frac{\rho^{(1,q^{(1)}_{j,-i})}}{\rho^{(0,q^{(1)}_{j,-i})}} - \log \frac{\rho^{(1,v_j)}}{\rho^{(0,v_j)}}\right| + (1-h_{i,j})\left|\log \frac{1-\rho^{(1,q^{(1)}_{j,-i})}}{1-\rho^{(0,q^{(1)}_{j,-i})}} - \log \frac{1-\rho^{(1,v_j)}}{1-\rho^{(0,v_j)}}\right|.$$

Using Claim 13 to bound the difference in likelihood ratios in terms of the absolute difference between $q^{(1)}_{j,-i}$ and v_j, we get

$$\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right| \le h_{i,j} c_4 \left|q^{(1)}_{j,-i} - v_j\right| + 0 \le c_4 \left|q^{(1)}_{j,-i} - v_j\right|.$$

□

A.2.3 Putting it together

Proof of Theorem 1. We first prove the result on the driver side.

Driver Side Error Bound. We have derived a high probability error bound for $q^{(1)}_{j,-k}$ in (22). Rewrite it by setting $\epsilon = \sqrt{\frac{1}{c_0(T_{i,\cdot}-1)}\log\frac{1}{\delta}}$. Then, with probability at least $1-\delta$, we have

$$\left|q^{(1)}_{j,-k} - v_j\right| \le \exp\left(-c_1(T_{\cdot,j}-1) + \sqrt{\frac{T_{\cdot,j}-1}{c_0}\log\frac{1}{\delta}} + \operatorname{logit}\psi\right). \tag{25}$$

Following the same procedure, with probability at least $1-\delta$, we have

$$\left|q^{(1)}_{j} - v_j\right| \le \exp\left(-c_1 T_{\cdot,j} + \sqrt{\frac{T_{\cdot,j}}{c_0}\log\frac{1}{\delta}} + \operatorname{logit}\psi\right). \tag{26}$$

Customer Side Error Bound. Recall that

$$\operatorname{logit} p^{(1)}_i = \operatorname{logit}\phi + \sum_{j\in N_i} Z^{(1)}_{i\leftarrow j}$$

Using Claim 11 about Lipschitzness of the $\operatorname{logit}^{-1}$ function, and Claim 7 that bounds $\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right|$, we have

$$\left|p^{(1)}_i - p^{\star}_i\right| \le \frac{1}{4}\left|\operatorname{logit} p^{(1)}_i - \operatorname{logit} p^{\star}_i\right| \le \frac{1}{4}\sum_{j\in N_i}\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right| \le \frac{1}{4}T_{i,\cdot}c_4 \max_{j\in N_i}\left|q^{(1)}_{j,-i} - v_j\right|. \tag{27}$$

Apply union bound to (25) over all $j \in N_i$ with δ substituted by $\delta/T_{i,\cdot}$. Let $d = \min_{j\in N_i} T_{\cdot,j}$. Then, with probability at least $1-\delta$,

$$\max_{j\in N_i}\left|q^{(1)}_{j,-i} - v_j\right| \le \exp\left(-c_1(d-1) + \sqrt{\frac{d-1}{c_0}\log\frac{T_{i,\cdot}}{\delta}} + \operatorname{logit}\psi\right). \tag{28}$$

Combine (27) and (28). With probability at least $1-\delta$,

$$\left|p^{(1)}_i - p^{\star}_i\right| \le \exp\left(-c_1(d-1) + \sqrt{\frac{d-1}{c_0}\log\frac{T_{i,\cdot}}{\delta}} + \log(c_4 T_{i,\cdot}/4) + \operatorname{logit}\psi\right). \tag{29}$$

This completes the proof of this theorem. □

A.3 Proof of Theorem 2

Theorem 2. *Assume (A1)–(A3), and that each agent interacts with $d > 0$ other agents.*[12] *The customer scores of the three algorithms satisfy the following property. For any fixed target true*

[12]For simplicity, we consider a symmetric setting where $n = m$ and the graph is d-regular on both sides. The proof can be extended to a graph that is d_c-regular on the customer side and d_d-regular on the driver side, with minor modifications.

positive rate (TPR) in $(0,1)$,

$$\lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{naive} = c_-, \quad \lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{\star} = c_+, \quad \lim_{d\to\infty} -\frac{1}{d}\log \mathrm{FPR}^{BP(1)} \geq c_+,$$

where $c_- = D_{\mathrm{KL}}\left(\rho^{(1,\psi)} \,\middle\|\, \rho^{(0,\psi)}\right)$, $c_+ = \mathbb{E}_{y\sim\psi} D_{\mathrm{KL}}\left(\rho^{(1,y)} \,\middle\|\, \rho^{(0,y)}\right)$, *and* $c_+ > c_-$.

Proof. The proof for each algorithm consists of two parts: connecting the TPR with the threshold, and connecting the threshold with FPR. Given any TPR $r \in (0,1)$, we can find the threshold τ such that

$$\mathbb{P}(\text{score} > \tau \,|\, \text{true type} = 1) = r.$$

Given the threshold τ, the FPR is

$$\mathrm{FPR}(\tau) = \mathbb{P}(\text{score} > \tau \,|\, \text{true type} = 0).$$

In the following analyses, we will study the customer scores p_i^{naive}, $p_i^{\star}$, and $p_i^{\text{BP(1)}}$ for customer i.

Recall that the probability of observing $h_{i,j}$ given types u_i and v_j is $\mathbb{P}\left(h_{i,j} = 1 \,\middle|\, u_i, v_j\right) = \rho^{(u_i,v_j)}$ and $\mathbb{P}\left(h_{i,j} = 0 \,\middle|\, u_i, v_j\right) = 1 - \rho^{(u_i,v_j)}$.

A.3.1 Naive

Under the naive approach, a customer with type any $u_i \in \{0,1\}$ has score

$$p_i^{\text{naive}} = \frac{1}{d}\sum_{j\in N_i} h_{i,j},$$

One can equivalently think of the above equation as an empirical estimate of Bernoulli samples where each sample $h_{i,j} \sim \mathrm{Ber}(\rho^{(u_i,\psi)})$. This equivalence holds due to Claim 4, combined with the fact that a mixture of Bernoulli's is also a Bernoulli random variable.

Let I^{naive} be the large deviation rate function for the h for straightforward customers with type $u_i = 0$. By Lemma 3, for any $\tau \in (0,1)$,

$$I^{\text{naive}}(\tau) = D_{\mathrm{KL}}\left(\mathrm{Ber}(\tau) \,\middle\|\, \mathrm{Ber}(\rho^{(0,\psi)})\right).$$

Lemma 2 allows us to bound the false positive rate (FPR) in terms of the rate function I^{naive} evaluated at the threshold τ set for positive classification:

$$\begin{aligned} -\frac{1}{d}\log \mathrm{FPR}^{\text{naive}}(\tau) &= -\frac{1}{d}\log \mathbb{P}\left(p_i^{\text{naive}} > \tau \,\middle|\, u_i = 0\right) \\ &= I^{\text{naive}}(\tau) + O(d^{-1}\log d). \end{aligned}$$

The next step is to evaluate the threshold τ that is required to achieve a certain TPR r. Let Q_X be the quantile function (inverse CDF) of any random variable X. By Lemma 1, the threshold to

achieve a true positive rate r is of the form

$$\begin{aligned}\tau^{\text{naive}}(r) &= Q_{p_i^{\text{naive}}}(1-r)\\ &= \mathbb{E}\left[h_{i,j}\,\middle|\,u_i=1\right] + \frac{1}{\sqrt{d}}\Phi^{-1}(1-r)\sqrt{\operatorname{Var}\left(h_{i,j}\,\middle|\,u_i=1\right)} + O(d^{-1})\\ &= \rho^{(1,\psi)} + \frac{1}{\sqrt{d}}\Phi^{-1}(1-r)\sqrt{\rho^{(1,\psi)}(1-\rho^{(1,\psi)})} + O(d^{-1}).\end{aligned}$$

Intuitively, the mean of the messages is going to concentrate around $\rho^{(1,\psi)}$. When $r = 1/2$, the $\frac{1}{\sqrt{d}}$ term disappears, and when $r \neq 1/2$, the threshold can deviate from $\rho^{(1,\psi)}$ by roughly $O(d^{-1/2})$.

Plugging in $\tau^{\text{naive}}(r)$ into the FPR expression derived above, we get

$$\begin{aligned}-\log \text{FPR}^{\text{naive}}(\tau^{\text{naive}}(r)) &= d \cdot I^{\text{naive}}(\tau^{\text{naive}}(r)) + O(\log d)\\ &= d \cdot I^{\text{naive}}(\rho^{(1,\psi)} + O(d^{-1/2})) + O(\log d)\\ &= d \cdot D_{\text{KL}}\left(\rho^{(1,\psi)} + O(d^{-1/2}) \,\middle\|\, \rho^{(0,\psi)}\right) \qquad (30)\\ &= d \cdot D_{\text{KL}}\left(\rho^{(1,\psi)} \,\middle\|\, \rho^{(0,\psi)}\right) + O(d^{1/2}), \qquad (31)\end{aligned}$$

where (30) holds because of Lemma 3, and (31) holds because of Claim 12.

A.3.2 Oracle

Recall that the oracle scores are of the form:

$$\begin{aligned}\operatorname{logit} p_i^\star &= \operatorname{logit}\phi + \sum_{j\in N_i} \log \frac{\mathbb{P}\left(h_{i,j}\,\middle|\,1, v_j\right)}{\mathbb{P}\left(h_{i,j}\,\middle|\,0, v_j\right)}\\ &= \operatorname{logit}\phi + \sum_{j\in N_i}\left(h_{i,j}\log\frac{\rho^{(1,v_j)}}{\rho^{(0,v_j)}} + (1-h_{i,j})\log\frac{1-\rho^{(1,v_j)}}{1-\rho^{(0,v_j)}}\right)\\ &\triangleq \operatorname{logit}\phi + \sum_{j\in N_i} Z^\star_{i\leftarrow j},\end{aligned}$$

We first evaluate the distribution of messages $Z^\star_{i\leftarrow j}$'s. Considering the randomness in v_j and $h_{i,j}$, the distribution of $Z^\star_{i\leftarrow j}$ is of the form

$$Z^\star_{i\leftarrow j} = \begin{cases}\log\frac{\rho^{(1,1)}}{\rho^{(0,1)}} & \text{w.p. } \psi\rho^{(u_i,1)},\\ \log\frac{\rho^{(1,0)}}{\rho^{(0,0)}} & \text{w.p. } (1-\psi)\rho^{(u_i,0)},\\ \log(1-\alpha) & \text{w.p. } \psi(1-\rho^{(u_i,1)}) + (1-\psi)(1-\rho^{(u_i,0)}).\end{cases} \qquad (32)$$

We can evaluate the expectation of the message from each neighbor j as:

$$\mathbb{E}\left[Z^\star_{i\leftarrow j}\,\middle|\,u_i=1\right] = \mathop{\mathbb{E}}_{v_j\sim\psi}\mathbb{E}\left[Z^\star_{i\leftarrow j}\,\middle|\,u_i=1, v_j\right] = \mathop{\mathbb{E}}_{v_j\sim\psi} D_{\text{KL}}\left(\rho^{(1,v_j)}\,\middle\|\,\rho^{(0,v_j)}\right). \qquad (33)$$

Let $I^\star$ be the rate function for the message $Z^\star_{i\leftarrow j}$ (note that these messages are not Bernoulli and this rate function needs to be evaluated later). Similar to the case of Naive, Lemma 2 allows us to bound the FPR in terms of the rate function $I^\star$ evaluated at the threshold τ set for positive classification:

$$\begin{aligned}
-\log \mathrm{FPR}^\star(\tau) &= -\log \mathbb{P}\big(p_i^\star > \tau \,\big|\, u_i = 0\big) \\
&= -\log \mathbb{P}\left(\frac{1}{d}\operatorname{logit} p_i^\star > \frac{1}{d}\operatorname{logit}\tau \,\middle|\, u_i = 0\right) \\
&= -\log \mathbb{P}\left(\frac{1}{d}\sum_{j\in N_i} Z^\star_{i\leftarrow j} > \frac{1}{d}(\operatorname{logit}\tau - \operatorname{logit}\phi) \,\middle|\, u_i = 0\right) \\
&= d\cdot I^\star\left(\frac{1}{d}(\operatorname{logit}\tau - \operatorname{logit}\phi)\right) + O(\log d). \qquad (34)
\end{aligned}$$

Now, given a TPR $r \in (0,1)$, we need to find the corresponding cutoff threshold and then use the above equation to establish a FPR at this threshold.

Apply Lemma 1 to get the quantile function for $\sum_{j\in N_i} Z^\star_{i\leftarrow j}$, the cutoff threshold for the oracle score to achieve a TPR of $r \in (0,1)$ is

$$\begin{aligned}
\operatorname{logit}\tau^\star(r) - \operatorname{logit}\phi &= d\,\mathbb{E}\big[Z^\star_{i\leftarrow j} \,\big|\, u_i = 1\big] + d^{1/2}\Phi^{-1}(1-r)\sqrt{\mathrm{Var}\left(Z^\star_{i\leftarrow j} \,\middle|\, u_i = 1\right)} + O(1) \\
&= d\,\mathbb{E}\big[Z^\star_{i\leftarrow j} \,\big|\, u_i = 1\big] + O(d^{1/2}), \qquad (35)
\end{aligned}$$

where we have used the fact that $\mathrm{Var}\left(Z^\star_{i\leftarrow j} \,\middle|\, u_i = 1\right)$ is bounded since $Z^\star_{i\leftarrow j}$ is a discrete and bounded random variable. The above equation shows that the threshold is close to the mean of the messages for large d irrespective of the rate r.

What is left to calculate is the large deviation rate function $I^\star$ for the messages received by a type 0 customer, evaluated at $(\operatorname{logit}\tau^\star(r) - \operatorname{logit}\phi)/d = \mathbb{E}\left[Z^\star_{i\leftarrow j} \,\middle|\, u_i = 1\right] + O(d^{-1/2})$.

To study the large deviation rate function of $Z^\star_{i\leftarrow j}$ when $u_i = 0$, we study its moment generating function $M(\theta)$ and cumulant generating function $\Lambda(\theta)$ defined as:

$$M(\theta) = \mathbb{E}\big[\exp(\theta Z^\star_{i\leftarrow j}) \,\big|\, u_i = 0\big] \qquad (36)$$

$$\Lambda(\theta) = \log M(\theta). \qquad (37)$$

In Claim 9, we prove that

$$M(1) = 1, \qquad M'(1) = \mathbb{E}\big[Z^\star_{i\leftarrow j} \,\big|\, u_i = 1\big], \qquad (38)$$

$$\Lambda(1) = 0, \qquad \Lambda'(1) = M'(1). \qquad (39)$$

Denote the rate function as

$$I^\star(x) \triangleq \sup_\theta (x\theta - \Lambda(\theta)). \qquad (40)$$

We know that at $x = M'(1) = \mathbb{E}\left[Z^{\star}_{i\leftarrow j} \,\middle|\, u_i = 1\right]$, $\theta = 1$ satisfies the first order condition of $x\theta - \Lambda(\theta)$:

$$\frac{\partial}{\partial\theta}\big(x\theta - \Lambda(\theta)\big)\Big|_{x=\mathbb{E}\left[Z^{\star}_{i\leftarrow j}\,\middle|\,u_i=1\right]} = \mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big] - \mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big] = 0.$$

Since $Z^{\star}_{i\leftarrow j}$ is not a constant, its cumulant generating function $\Lambda(\theta)$ is strictly convex, and $\Lambda'(\theta)$ is strictly increasing. As a result, $x\theta - \Lambda(\theta)$ is strictly concave, and the supremum in (40) is achieved at $\theta = 1$, meaning that

$$I^{\star}(\mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big]) = \mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big] \cdot 1 - \Lambda(1) = \mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big]. \tag{41}$$

The above equation allows us to evaluate the rate function at the threshold of interest, namely $\mathbb{E}\left[Z^{\star}_{i\leftarrow j} \,\middle|\, u_i = 1\right]$. Finally combining, (34), (35), and (33), and using Claim 14 we have

$$\begin{aligned}
-\log \mathrm{FPR}^{\star}(\tau^{\star}(r)) &= d \cdot I^{\star}\left(\frac{1}{d}(\operatorname{logit}\tau^{\star}(r) - \operatorname{logit}\phi)\right) + O(\log d) \\
&= d \cdot I^{\star}\left(\mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big] + O(d^{-1/2})\right) + O(\log d) \\
&= d \cdot I^{\star}\left(\mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big]\right) + O(d^{1/2}) \\
&= d \cdot \mathbb{E}\big[Z^{\star}_{i\leftarrow j} \,\big|\, u_i = 1\big] + O(d^{1/2}) \\
&= d \cdot \mathop{\mathbb{E}}_{y\sim\psi} D_{\mathrm{KL}}\left(\rho^{(1,y)} \,\middle\|\, \rho^{(0,y)}\right) + O(d^{1/2}).
\end{aligned}$$

This completes the proof for the Oracle scores.

A.3.3 First Iteration BP Score: BP(1)

Recall that the BP scores are defined in terms of the messages $Z^{(1)}_{i\leftarrow j}$'s sent from the drivers to customers. Let $I^{\mathrm{BP}(1)}$ be the large deviation rate function for the $Z^{(1)}_{i\leftarrow j}$ at $u_i = 0$ (again this rate function needs to be evaluated later). Similar to the case of oracle, Lemma 2 allows us to bound the FPR in terms of the rate function evaluated at the threshold τ set for positive classification:

$$-\log \mathrm{FPR}^{\mathrm{BP}(1)}(\tau) = d \cdot I^{\mathrm{BP}(1)}\left(\frac{1}{d}(\operatorname{logit}\tau - \operatorname{logit}\phi)\right) + O(\log d). \tag{42}$$

Apply Lemma 1 to get the quantile function for $\sum_{j\in N_i} Z^{(1)}_{i\leftarrow j}$:

$$\begin{aligned}
\operatorname{logit}\tau^{\mathrm{BP}(1)}(r) - \operatorname{logit}\phi &= d\,\mathbb{E}\left[Z^{(1)}_{i\leftarrow j} \,\middle|\, u_i = 1\right] + d^{1/2}\Phi^{-1}(1-r)\sqrt{\mathrm{Var}\left(Z^{(1)}_{i\leftarrow j} \,\middle|\, u_i = 1\right)} + O(1) && (43) \\
&= d\,\mathbb{E}\left[Z^{(1)}_{i\leftarrow j} \,\middle|\, u_i = 1\right] + O(d^{1/2}). && (44)
\end{aligned}$$

Combining the above two equations, we need to evaluate the rate function $I^{\mathrm{BP}(1)}$ at $\mathbb{E}\left[Z^{(1)}_{i\leftarrow j} \,\middle|\, u_i = 1\right]+$

$O(d^{-1/2})$. To study the large deviation rate function of $Z^{(1)}_{i\leftarrow j}$ when $u_i = 0$, we study its moment generating function $M(\theta)$ and cumulant generating function $\Lambda(\theta)$.

$$M(\theta) = \mathbb{E}\Big[\exp(\theta Z^{(1)}_{i\leftarrow j}) \,\Big|\, u_i = 0\Big], \tag{45}$$

$$\Lambda(\theta) = \log M(\theta). \tag{46}$$

Then

$$I^{\mathrm{BP}(1)}(x) \triangleq \sup_{\theta}(x\theta - \Lambda(\theta)). \tag{47}$$

However, we no longer get a clean expression for $M(1)$ and $\Lambda(1)$ similar to one we got for the oracle. Hence, we need to rely on the fact that oracle messages are close to the BP(1) messages.

In Appendix A.1, we have shown in Claim 7 that the BP driver score is close to the true type. Moreover, Corollary 1 shows that when the driver score is close to the true type, the 1st iteration messages received customer under BP $Z^{(1)}_{i\leftarrow j}$ is close to $Z^{\star}_{i\leftarrow j}$. Combining the two results, we have

$$\left|\mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big] - \mathbb{E}\big[Z^{\star}_{i\leftarrow j}\,\big|\,u_i = 1\big]\right| \le \mathbb{E}\left[c_4\left|q^{(1)}_{j,-i} - v_j\right|\,\middle|\,u_i = 1\right] \le c_4 c_3 \exp(-c_2 d). \tag{48}$$

Then, using this fact along with Claim 10, we get

$$M(1) = 1 + O(\exp(-c_2 d)), \qquad \Lambda(1) = O(\exp(-c_2 d)). \tag{49}$$

Using this, we establish

$$\begin{aligned}
I^{\mathrm{BP}(1)}(\mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big]) &= \sup_{\theta}(\mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big]\theta - \Lambda(\theta)) \\
&\ge \mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big] - \Lambda(1) \\
&\ge \mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big] - O(\exp(-c_2 d)) \\
&\ge \mathbb{E}\big[Z^{\star}_{i\leftarrow j}\,\big|\,u_i = 1\big] - O(\exp(-c_2 d))
\end{aligned}$$

Using the above bound on the rate function, Claim 14, and equation (41)

$$\begin{aligned}
-\log \mathrm{FPR}^{\mathrm{BP}(1)}(\tau^{\mathrm{BP}(1)}(r)) &= d \cdot I^{\mathrm{BP}(1)}\left(\frac{1}{d}(\operatorname{logit}\tau^{\mathrm{BP}(1)}(r) - \operatorname{logit}\phi)\right) + O(\log d) \\
&= d \cdot I^{\mathrm{BP}(1)}\left(\mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big] + O(d^{-1/2})\right) + O(\log d) \\
&= d \cdot I^{\mathrm{BP}(1)}\left(\mathbb{E}\Big[Z^{(1)}_{i\leftarrow j}\,\Big|\,u_i = 1\Big]\right) + O(d^{1/2}) \\
&\ge d\left(\mathbb{E}\big[Z^{\star}_{i\leftarrow j}\,\big|\,u_i = 1\big] - O(\exp(-c_2 d))\right) + O(d^{1/2}) \\
&\ge d \mathop{\mathbb{E}}_{y\sim\psi} D_{\mathrm{KL}}\left(\rho^{(1,y)} \,\middle\|\, \rho^{(0,y)}\right) - O(\exp(-c_2 d)d) + O(d^{1/2}).
\end{aligned}$$

This concludes the analysis of the BP score, and completes the proof of Theorem 2. □

A.4 Technical Lemmas

We provide in this sections additional notations and formal statements and proofs of a number of technical results used throughout the proof.

A.4.1 Large Deviations

Let Q_X be the quantile function (inverse CDF) of any random variable X. The first result on the quantile function of the mean of a number of random variables is used to bound the cutoff thresholds under each algorithm in the proof of Theorem 2 in Appendix A.3.

Lemma 1 (Quantile convergence)**.** *Let $X_1, X_2, \ldots, X_k$ be i.i.d. random variables with mean μ and variance σ^2 and finite third moments $\mathbb{E}[|X_i|^3] < \infty$. Define the sum $S_k = \sum_{i=1}^k X_i$ and the mean $\bar{X}_k = \frac{1}{k}\sum_{i=1}^k X_i$. Then, for all $\xi \in (0,1)$, as $k \to \infty$,*

$$Q_{S_k}(\xi) = k\mu + \sqrt{k}\sigma\Phi^{-1}(\xi) + O(1), \tag{50}$$

$$Q_{\bar{X}_k}(\xi) = \mu + \frac{\sigma}{\sqrt{k}}\Phi^{-1}(\xi) + O(k^{-1}), \tag{51}$$

where Φ is the standard normal CDF, and Q_{S_k} and $Q_{\bar{X}_k}$ be the quantile functions (inverse CDFs) of S_k and $\bar{X}_k$, respectively.

Proof. Consider the standardized sum defined as $Z_k = \frac{S_k - k\mu}{\sigma\sqrt{k}}$. Let $F_{Z_k}(z)$ denote the cumulative distribution function (CDF) of Z_k. Since the third moment is finite, the Berry-Esseen theorem guarantees that F_{Z_k} converges to the standard normal CDF Φ uniformly with a rate of $O(k^{-1/2})$:

$$\sup_{z\in\mathbb{R}} |F_{Z_k}(z) - \Phi(z)| = O(k^{-1/2}).$$

We seek the quantile $Q_{S_k}(\xi)$, which corresponds to the quantile $q_k(\xi)$ of the standardized variable Z_k such that $F_{Z_k}(q_k(\xi)) \approx \xi$. Substituting the Berry-Esseen bound at this quantile, we have:

$$|\xi - \Phi(q_k(\xi))| = O(k^{-1/2}).$$

Let $z_\xi = \Phi^{-1}(\xi)$. Since Φ^{-1} has a bounded derivative for any fixed $\xi \in (0,1)$, we can apply a first-order Taylor expansion (or the Mean Value Theorem) to approximate $q_k(\xi)$:

$$q_k(\xi) = \Phi^{-1}(\xi + O(k^{-1/2})) = z_\xi + O(k^{-1/2}).$$

Finally, we map this back to the original variables. For the sum $S_k = k\mu + \sigma\sqrt{k}Z_k$:

$$\begin{aligned} Q_{S_k}(\xi) &= k\mu + \sigma\sqrt{k}\, q_k(\xi) \\ &= k\mu + \sigma\sqrt{k}\Big(z_\xi + O(k^{-1/2})\Big) \end{aligned}$$

$$= k\mu + \sigma\sqrt{k}z_\xi + O(1).$$

For the sample mean $\bar{X}_k = S_k/k$, we divide the result by k:

$$Q_{\bar{X}_k}(\xi) = \frac{1}{k}Q_{S_k}(\xi) = \mu + \frac{\sigma}{\sqrt{k}}z_\xi + O(k^{-1}).$$

This completes the proof of this lemma. □

Lemma 2 is a classical result by Bahadur and Rao [1960].

Lemma 2 (Large deviations). *Let $X_1, X_2, \ldots, X_k$ be i.i.d. random variables on $\mathbb{R}$. Define the mean $\bar{X}_k = \frac{1}{k}\sum_{i=1}^k X_i$ and the rate function*

$$I(x) = \sup_\theta(\theta x - \log \mathbb{E}\exp(\theta X_1)).$$

Then, for all $x > \mathbb{E}X_1$, as $k \to \infty$,

$$-\log \mathbb{P}\big(\bar{X}_k > x\big) = kI(x) + \frac{1}{2}\log k + O(1).$$

The following classic result is used to derive the false positive rates for the naive approach in the proof of Theorem 2.

Lemma 3 (Large deviation rate function for Bernoulli distribution). *The large deviation rate function for the Bernoulli distribution with parameter $p \in (0,1)$ is*

$$I(x) = x\log\frac{x}{p} + (1-x)\log\frac{1-x}{1-p} = D_{\mathrm{KL}}(\mathrm{Ber}(x) \,\|\, \mathrm{Ber}(p)).$$

This is a classic result in the literature and we do not provide a proof (see, for example Lattimore and Szepesvári [2020]).

Claim 8. *For any $\alpha, \beta, \gamma, \phi, \psi \in (0,1)$, the following holds*

$$c_+ = \mathop{\mathbb{E}}_{y\sim\psi} D_{\mathrm{KL}}\Big(\rho^{(1,y)} \Big\| \rho^{(0,y)}\Big) > c_- = D_{\mathrm{KL}}\Big(\rho^{(1,\psi)} \Big\| \rho^{(0,\psi)}\Big), \tag{52}$$

Proof. We will first show that $D_{\mathrm{KL}}\Big(\rho^{(1,y)} \Big\| \rho^{(0,y)}\Big)$ is strictly convex in y Taking the second derivative, we get

$$\frac{\mathrm{d}^2}{\mathrm{d}y^2} D_{\mathrm{KL}}\Big(\rho^{(1,y)} \Big\| \rho^{(0,y)}\Big) = \frac{(\alpha^*)^2(1-\beta^*)^2(1-\gamma^*)^2}{(1-(1-\gamma^*)(1-\beta^* y))^2(1-(1-\alpha^*)(1-\gamma^*)(1-\beta^* y))} > 0. \tag{53}$$

Every term in the fraction is strictly positive for $y \in [0,1]$, hence the strict convexity. The result then follows due to Jensen's inequality. □

A.4.2 MGF and CGF of Messages

Claim 9. *Let $M(\theta)$ and $\Lambda(\theta)$ be the MGF and CGF of the oracle message $Z^{\star}_{i\leftarrow j}$ for straightforward customers with $u_1 = 0$ (as defined in (36) and (37)). For any $\alpha, \beta, \gamma, \phi, \psi \in (0,1)$, at $\theta = 1$,*

$$M(1) = 1, \qquad M'(1) = \mathbb{E}\left[Z^{\star}_{i\leftarrow j} \,\middle|\, u_i = 1\right],$$
$$\Lambda(1) = 0, \qquad \Lambda'(1) = M'(1).$$

where $\mathbb{E}\left[Z^{\star}_{i\leftarrow j} \,\middle|\, u_i = 1\right]$ is the expectation of the messages received by customers with type $u_i = 1$.

Proof. Applying (32), we get

$$M(\theta) = \psi\rho^{(0,1)} \exp\left(\theta \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}}\right) + (1-\psi)\rho^{(0,0)} \exp\left(\theta \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}}\right) + (\psi(1-\rho^{(0,1)}) + (1-\psi)(1-\rho^{(0,0)})) \exp(\theta \log(1-\alpha)).$$

Take the derivative w.r.t. θ, we get

$$M'(\theta) = \psi\rho^{(0,1)} \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}} \exp\left(\theta \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}}\right) + (1-\psi)\rho^{(0,0)} \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}} \exp\left(\theta \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}}\right) + (\psi(1-\rho^{(0,1)}) + (1-\psi)(1-\rho^{(0,0)})) \log(1-\alpha) \exp(\theta \log(1-\alpha)).$$

To evaluate $M(\theta)$ and its derivative at $\theta = 1$, we utilize the identity

$$(1-\rho^{(0,y)})(1-\alpha) = 1 - \rho^{(1,y)} \tag{54}$$

which follows from the definition of $\rho^{(x,y)}$. Substitute $\theta = 1$ into $M(\theta)$ and applying (54), we get:

$$\begin{aligned} M(1) &= \psi\rho^{(0,1)} \frac{\rho^{(1,1)}}{\rho^{(0,1)}} + (1-\psi)\rho^{(0,0)} \frac{\rho^{(1,0)}}{\rho^{(0,0)}} + (\psi(1-\rho^{(0,1)}) + (1-\psi)(1-\rho^{(0,0)}))(1-\alpha) \\ &= \psi\rho^{(1,1)} + (1-\psi)\rho^{(1,0)} + \psi(1-\rho^{(1,1)}) + (1-\psi)(1-\rho^{(1,0)}) \\ &= \psi(\rho^{(1,1)} + 1 - \rho^{(1,1)}) + (1-\psi)(\rho^{(1,0)} + 1 - \rho^{(1,0)}) \\ &= \psi + 1 - \psi = 1. \end{aligned}$$

Substitute $\theta = 1$ into $M'(\theta)$ and again apply (54):

$$\begin{aligned} M'(1) &= \psi\rho^{(0,1)} \frac{\rho^{(1,1)}}{\rho^{(0,1)}} \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}} + (1-\psi)\rho^{(0,0)} \frac{\rho^{(1,0)}}{\rho^{(0,0)}} \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}} \\ &\quad + (\psi(1-\rho^{(0,1)}) + (1-\psi)(1-\rho^{(0,0)}))(1-\alpha)\log(1-\alpha) \\ &= \psi\rho^{(1,1)} \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}} + (1-\psi)\rho^{(1,0)} \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}} \\ &\quad + (\psi(1-\rho^{(1,1)}) + (1-\psi)(1-\rho^{(1,0)}))\log(1-\alpha) \end{aligned}$$

$$= \psi\rho^{(1,1)} \log \frac{\rho^{(1,1)}}{\rho^{(0,1)}} + (1-\psi)\rho^{(1,0)} \log \frac{\rho^{(1,0)}}{\rho^{(0,0)}}$$
$$+ \psi(1-\rho^{(1,1)}) \log \frac{1-\rho^{(1,1)}}{1-\rho^{(0,1)}} + (1-\psi)(1-\rho^{(1,0)}) \log \frac{1-\rho^{(1,0)}}{1-\rho^{(0,0)}}.$$

The coefficients in the expression above correspond exactly to the probabilities of the score outcomes conditioned on $u_i = 1$. Thus,

$$M'(1) = \mathbb{E}\left[Z^{\star}_{i\leftarrow j} \mid u_i = 1\right].$$

Finally, by the definition $\Lambda(\theta) = \log M(\theta)$, we have:

$$\Lambda(1) = \log M(1) = 0,$$
$$\Lambda'(1) = \frac{M'(1)}{M(1)} = M'(1).$$

This completes the proof of this claim. □

Claim 10. *Let $M(\theta)$ and $\Lambda(\theta)$ be the MGF and CGF of the BP message $Z^{(1)}_{i\leftarrow j}$ for straightforward customers with $u_1 = 0$ (as defined in equations (45) and (46)). For all $\alpha, \beta, \gamma, \phi, \psi \in (0,1)$,*

$$M(1) = 1 + O(\exp(-c_2 d)),$$
$$\Lambda(1) = O(\exp(-c_2 d)).$$

Proof. In order to prove this claim we utilize the following inequality:

$$|\exp(a) - \exp(b)| \le c \cdot |a - b|$$

for an appropriately chosen constant c. We then have

$$\left|\mathbb{E}\left[\exp(Z^{(1)}_{i\leftarrow j}) \,\middle|\, u_i = 1\right] - \mathbb{E}\left[\exp(Z^{\star}_{i\leftarrow j}) \,\middle|\, u_i = 1\right]\right| \le \mathbb{E}\left[\left|\exp(Z^{(1)}_{i\leftarrow j}) - \exp(Z^{\star}_{i\leftarrow j})\right| \,\middle|\, u_i = 1\right]$$
$$\le c \cdot \mathbb{E}\left[\left|Z^{(1)}_{i\leftarrow j} - Z^{\star}_{i\leftarrow j}\right| \,\middle|\, u_i = 1\right]$$

Combining Corollary 1 and Claim 7, we get

$$\left|\mathbb{E}\left[\exp(Z^{(1)}_{i\leftarrow j}) \,\middle|\, u_i = 1\right] - \mathbb{E}\left[\exp(Z^{\star}_{i\leftarrow j}) \,\middle|\, u_i = 1\right]\right| \le c \cdot c_4 \cdot c_3 \exp(-c_2 d) = O(\exp(-c_2 d)).$$

First, we analyze $M(1)$. Decomposing the expectation:

$$M(1) = \mathbb{E}\left[\exp(Z^{(1)}_{i\leftarrow j}) \mid u_i = 1\right]$$
$$= \mathbb{E}\left[\exp(Z^{\star}_{i\leftarrow j}) \mid u_i = 1\right] + O(\exp(-c_2 d)) \tag{55}$$
$$= 1 + O(\exp(-c_2 d)), \tag{56}$$

where the last equality follows from Claim 9.

Finally, for the CGF:

$$\Lambda(1) = \log M(1) = \log(1 + O(\exp(-c_2 d))) = O(\exp(-c_2 d)). \tag{57}$$

where we use $\log(1 + x) = x$ for a small enough x. □

A.4.3 Lipschitz Continuity

Claim 11. *For any $a, b \in [0, 1]$, we have that*

$$|a - b| \leq \frac{1}{4}|\text{logit}\, a - \text{logit}\, b|$$

Proof. This claim follows easily using the fact that the sigmoid function is the inverse of the logit function, and it is a Lipschitz function with constant $L = 1/4$. Hence,

$$|a - b| = |\text{sigmoid}(\text{logit}\, a) - \text{sigmoid}(\text{logit}\, b)| \leq \frac{1}{4}|\text{logit}\, a - \text{logit}\, b|$$

□

Claim 12. *For any $x, p \in (0, 1)$ and any small $\delta > 0$ such that $x + \delta \in (0, 1)$, the KL divergence between two Bernoulli distributions satisfies*

$$|D_{\text{KL}}(x + \delta \,\|\, p) - D_{\text{KL}}(x \,\|\, p)| \leq c_5 \delta \tag{58}$$

where $c_5 = \max_{z \in \{x, x+\delta\}} \left| \ln \frac{z(1-p)}{p(1-z)} \right|$. [13]

Proof. Let $f(t) = D_{\text{KL}}(t \,\|\, p)$. By definition, for $t \in (0, 1)$:

$$f(t) = t \ln \frac{t}{p} + (1 - t) \ln \frac{1 - t}{1 - p}.$$

We compute the derivative of $f(t)$ with respect to t:

$$\begin{aligned} f'(t) &= \frac{d}{dt}(t \ln t - t \ln p + (1 - t) \ln(1 - t) - (1 - t) \ln(1 - p)) \\ &= (\ln t + 1) - \ln p + (-\ln(1 - t) - 1) + \ln(1 - p) \\ &= \ln \frac{t}{1 - t} - \ln \frac{p}{1 - p} \\ &= \ln\left(\frac{t(1 - p)}{p(1 - t)}\right). \end{aligned}$$

[13] Note that c_5 might depend on δ, but it becomes a constant as δ goes to 0.

Since $f(t)$ is continuously differentiable on $(0,1)$, by the Mean Value Theorem applied to the interval $[x, x+\delta]$, there exists a $\xi \in (x, x+\delta)$ such that:

$$f(x+\delta) - f(x) = f'(\xi) \cdot \delta.$$

Taking the absolute value of both sides yields:

$$|f(x+\delta) - f(x)| = |f'(\xi)|\delta.$$

We observe that the second derivative $f''(t) = \frac{1}{t(1-t)}$ is strictly positive for all $t \in (0,1)$, implying that the first derivative $f'(t)$ is strictly increasing. Consequently, the function $|f'(t)|$ attains its maximum on the closed interval $[x, x+\delta]$ at one of the endpoints. We define the constant c_5 as this maximum value:

$$c_5 \triangleq \max(|f'(x)|, |f'(x+\delta)|).$$

Substituting this bound into the Mean Value Theorem result, we obtain the desired inequality:

$$|D_{\mathrm{KL}}(x+\delta \,\|\, p) - D_{\mathrm{KL}}(x \,\|\, p)| \le c_5\delta.$$

□

Claim 13. *For any $q \in [0,1]$ and $v \in \{0,1\}$, we have*

$$\left|\log \frac{\rho^{(1,q)}}{\rho^{(0,q)}} - \log \frac{\rho^{(1,v)}}{\rho^{(0,v)}}\right| \le c_4|q-v|\,,$$

where $c_4 = \frac{\alpha\beta(1-\gamma)}{\gamma(\alpha+\gamma-\alpha\gamma)}$. and

$$\left|\log \frac{1-\rho^{(1,q)}}{1-\rho^{(0,q)}} - \log \frac{1-\rho^{(1,v)}}{1-\rho^{(0,v)}}\right| = 0\,,$$

Proof. The latter bound holds by simple algebra. We now focus on proving the former bound. Let us define the function $f(q)$ for $q \in [0,1]$ as the log-likelihood ratio:

$$f(q) \triangleq \log \frac{\rho^{(1,q)}}{\rho^{(0,q)}} = \log \rho^{(1,q)} - \log \rho^{(0,q)}\,.$$

By the Mean Value Theorem, for any $q, v \in [0,1]$, there exists a ξ strictly between q and v such that:

$$|f(q) - f(v)| = |f'(\xi)| \cdot |q-v|\,.$$

To find the proper constant c_4, we determine the maximum of $|f'(\xi)|$ over the interval $[0,1]$.

Recall the definition of $\rho^{(x,y)}$ given in Eq. (14):

$$\rho^{(x,y)} = 1 - (1 - \alpha x)(1 - \beta y)(1 - \gamma)\,.$$

Let $C = 1 - \gamma$. Differentiating $\rho^{(x,q)}$ with respect to q:

$$\frac{\partial}{\partial q}\rho^{(x,q)} = \beta C(1 - \alpha x)\,.$$

Now, we differentiate $f(q)$:

$$\begin{aligned} f'(q) &= \frac{\frac{\partial}{\partial q}\rho^{(1,q)}}{\rho^{(1,q)}} - \frac{\frac{\partial}{\partial q}\rho^{(0,q)}}{\rho^{(0,q)}} \\ &= \frac{\beta C(1-\alpha)}{\rho^{(1,q)}} - \frac{\beta C}{\rho^{(0,q)}} \\ &= \beta C\left(\frac{(1-\alpha)\rho^{(0,q)} - \rho^{(1,q)}}{\rho^{(1,q)}\rho^{(0,q)}}\right). \end{aligned}$$

We simplify the numerator term $(1 - \alpha)\rho^{(0,q)} - \rho^{(1,q)}$:

$$\begin{aligned} (1-\alpha)\rho^{(0,q)} - \rho^{(1,q)} &= (1-\alpha)[1 - C(1-\beta q)] - [1 - (1-\alpha)C(1-\beta q)] \\ &= (1-\alpha) - (1-\alpha)C(1-\beta q) - 1 + (1-\alpha)C(1-\beta q) \\ &= (1-\alpha) - 1 = -\alpha\,. \end{aligned}$$

Substituting this back into the expression for the derivative:

$$f'(q) = \frac{-\alpha\beta C}{\rho^{(1,q)}\rho^{(0,q)}} = \frac{-\alpha\beta(1-\gamma)}{\rho^{(1,q)}\rho^{(0,q)}}\,.$$

The magnitude is $|f'(q)| = \frac{\alpha\beta(1-\gamma)}{\rho^{(1,q)}\rho^{(0,q)}}$. To maximize this magnitude, we minimize the denominator. Since $\beta, \gamma \in (0, 1)$, the term $(1 - \beta q)$ is decreasing in q, which implies $\rho^{(x,q)}$ is strictly increasing in q. Therefore, the product $\rho^{(1,q)}\rho^{(0,q)}$ is minimized at $q = 0$.

Evaluating the terms at $q = 0$:

$$\begin{aligned} \rho^{(0,0)} &= 1 - (1)(1)(1-\gamma) = \gamma\,, \\ \rho^{(1,0)} &= 1 - (1-\alpha)(1)(1-\gamma) = \alpha + \gamma - \alpha\gamma\,. \end{aligned}$$

Thus, we obtain the constant:

$$c_4 = \frac{\alpha\beta(1-\gamma)}{\gamma(\alpha + \gamma - \alpha\gamma)}\,.$$

□

Claim 14. *Given a small $\delta > 0$ and a rate function I defined as*

$$I(y) = \sup_{\theta} y\theta - \Lambda(\theta),$$

if $\Lambda(\theta + O(\delta)) = \Lambda(\theta) + O(\delta)$ for all θ, and $I(x) = x$ for some value bounded x, then

$$I(x + \delta) = x + O(\delta).$$

Proof. We will use the property that I is a strictly concave function. Let $\theta_1 \triangleq \arg\sup_\theta x\theta - \Lambda(\theta)$ and $\theta_2 \triangleq \arg\sup_\theta (x + \delta)\theta - \Lambda(\theta)$. Also, let $f_1(\theta) = x\theta - \Lambda(\theta)$ and $f_2(\theta) = (x + \delta)\theta - \Lambda(\theta)$. We will now argue that θ_1 is close to θ_2.

Using Taylor's theorem on $\nabla f_2(\theta)$ we get

$$\nabla f_2(\theta_2) = \nabla f_2(\theta_1) + \nabla^2 f_2(\theta') \cdot (\theta_1 - \theta_2)$$

where θ' lies between θ_1 and θ_2. Since, $\nabla f_2(\theta_2) = 0$, we get

$$(\theta_1 - \theta_2) = -\frac{\nabla f_2(\theta_1)}{\nabla^2 f_2(\theta')}.$$

We now evaluate $\nabla f_2(\theta_1)$ as:

$$\nabla f_2(\theta_1) = (x + \delta) - \nabla\Lambda(\theta_1) = \delta,$$

which follows due to the fact that $\nabla f_1(\theta_1) = x - \nabla\Lambda(\theta_1) = 0$. Furthermore, since the function f_2 is strictly concave we know that $-\nabla^2 f_2(\theta') > C$ for some constant C. This implies $\theta_2 = \theta_1 + O(\delta)$. Using this

$$\begin{aligned} I(x + \delta) &= \theta_2(x + \delta) + \Lambda(\theta_2) \\ &= (\theta_1 + O(\delta))(x + \delta) + \Lambda(\theta_1 + O(\delta)) \\ &= \theta_1 x + O(\delta) + \Lambda(\theta_1) + O(\delta) \\ &= x + O(\delta). \end{aligned}$$

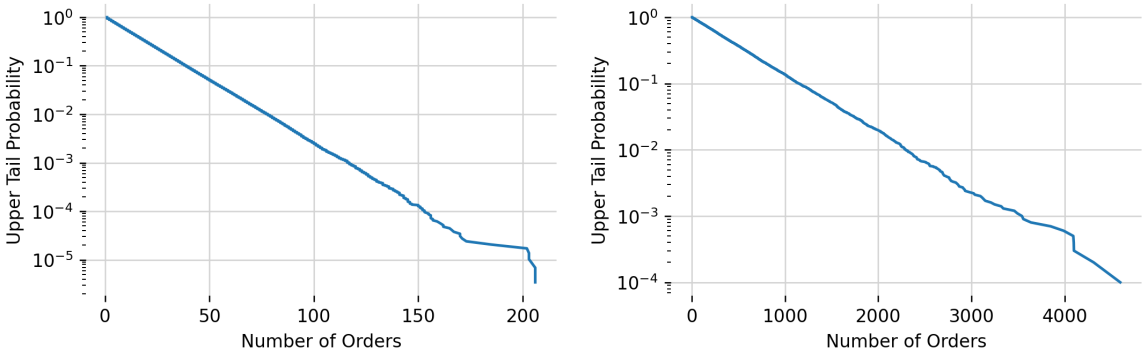
This completes the proof of the claim. □

B Simulation Details and Additional Results

B.1 Synthetic Data

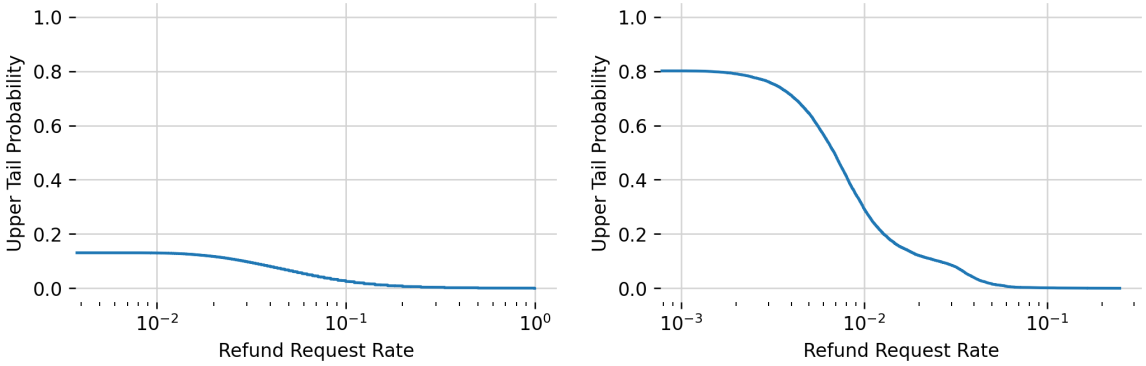
We first provide additional details for the synthetic simulations discussed in Section 4.1.

Data Generation. The number of orders per customer and driver, respectively, follows i.i.d. geometric distributions conditioned on the total number of orders. Figure 9 shows the empirical distribution of orders among customers and drivers. Once the number of orders of each customer



(a) Customer. (b) Driver.

Figure 9: Distribution of the number of orders per agent in the synthetic data. The curve represents the CCDF (complementary cumulative distribution function), the probability that an agent has more than a certain number of orders.



(a) Customer. (b) Driver.

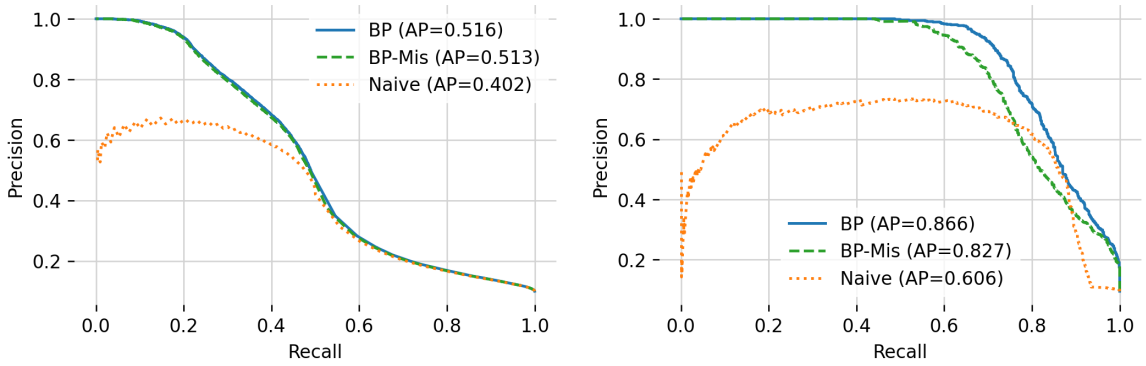
Figure 10: Distribution of the request rate of orders associated with each agent in the synthetic data.

and each driver is determined, we randomly match customer orders with driver orders. The matching is done uniformly at random (over all permutations of orders).

In the generated data, over 80% of customers have no refund requests, and around 20% of drivers have never delivered an order that has been requested for a refund. The distribution of the request rate of orders associated with each user is shown in Figure 10.

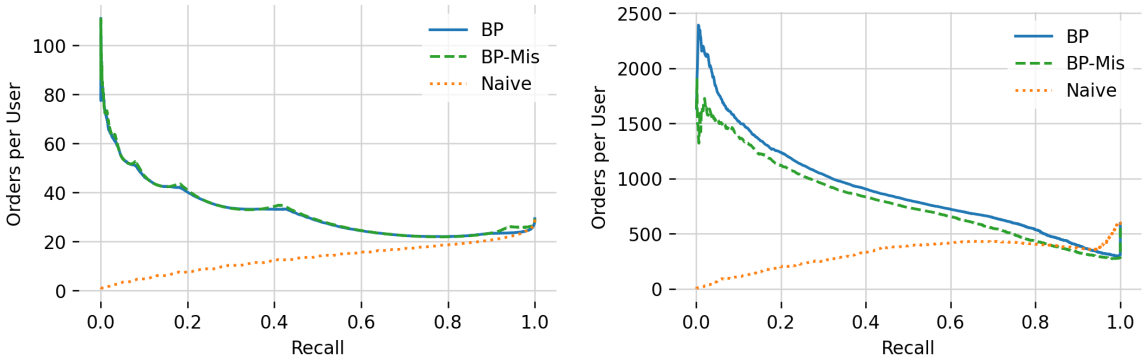
Precision and Recall. Figure 11 shows the precision-recall curves for the different methods on *all agents*.

Receiver Operating Characteristic (ROC) Curve. Another common way to evaluate the performance of a binary classifier is the receiver operating characteristic (ROC) curve, which shows the true positive rate (TPR) against the false positive rate (FPR) over different thresholds. The area under the ROC curve (AUROC) is a summary statistic that quantifies the overall performance of a classifier.



(a) Customer. (b) Driver.

Figure 11: Precision-recall curves for the different methods on all agents.



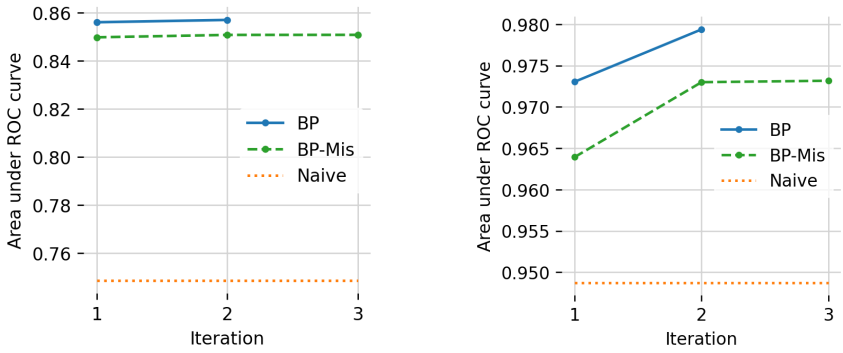
(a) Customer. (b) Driver.

Figure 12: Number of orders per flagged agent against recall.

Given the estimated scores and a threshold in the range $[0, 1]$, we can classify agents as strategic or not based on whether their estimated score is above the threshold. Ideally, we want to minimize both the errors of classifying non-strategic agents as strategic (false positive) and the errors of classifying strategic agents as non-strategic (false negative). In the setting of detecting strategic agents in the platform, false positives are more costly than false negatives, as we want to avoid penalizing agents who are not strategic. This corresponds to higher thresholds and the lower left corner of the ROC curve being more important.

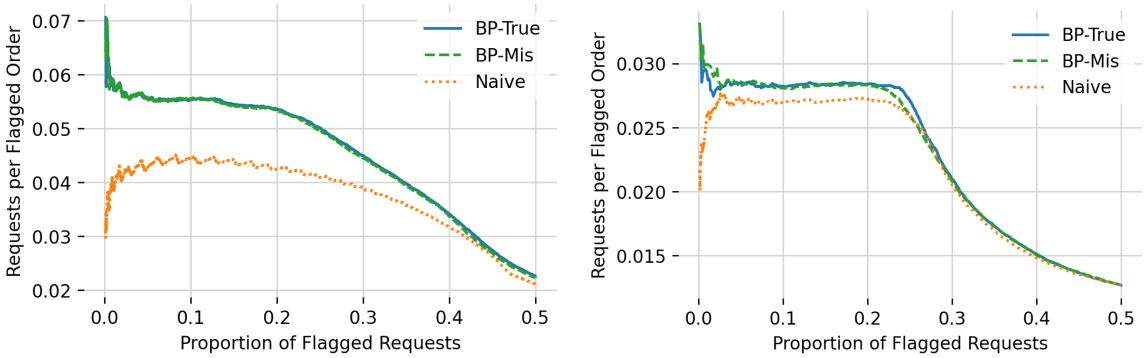
Figure 13 illustrates the AUC achieved by the BP algorithms (in comparison with the AUC under naive) over iterations. The BP algorithms with true and mis-specified parameters terminated after 2 and 3 iterations, respectively, with termination slack set to $\varepsilon_{\text{term}} = 10^{-5}$.

Evaluation Without Ground Truth. In practice, the ground truth is not available, and we have proposed different metrics to evaluate the performance without the ground truth in Section 4.2, and presented the results for the real data in Figure 6. As a comparison, Figure 14 shows the same metrics for the synthetic data.



(a) Customer. (b) Driver.

Figure 13: AUC (area under the ROC curve) over iterations, for agents with at least one refund request.



(a) Customer. (b) Driver.

Figure 14: Precision-recall curves for identifying requested orders in the test set, using customer and driver scores, respectively.

B.2 Additional Benchmarks

In addition to the naïve and the proposed BP algorithm, we also implemented and evaluated two non-Bayesian message passing (MP) algorithms, as well as a maximum likelihood estimation (MLE) algorithm. We present the results in Figure 15 and Figure 16.

MP1. The message passing algorithm in Karger et al. [2011] is designed for crowdsourcing applications, where each task is assigned to multiple workers, and each worker completes multiple tasks. The update rules can be adapted to our setting as follows:

$$x_{i\to j} \leftarrow \sum_{j'\in N_i\setminus j} \sum_{\ell=1}^{T_{i,j}} (-1)^{h_{i,j',\ell}} y_{j'\to i},$$

$$y_{j\to i} \leftarrow \sum_{i'\in N_j\setminus i} \sum_{\ell=1}^{T_{i',j}} (-1)^{h_{i',j,\ell}} x_{i'\to j},$$

where $x_{i\to j}$ and $y_{j\to i}$ take values in $\mathbb{R}$. Intuitively, when orders are good ($h = 0$) and neighboring agents are positive ($x > 0$ or $y > 0$), the messages reinforce each other. This is wrong in our setting, since one should not infer that an agent is strategic just because they have many good interactions with other strategic agents.

MP2. An alternative non-Bayesian message passing algorithm can be defined as follows:

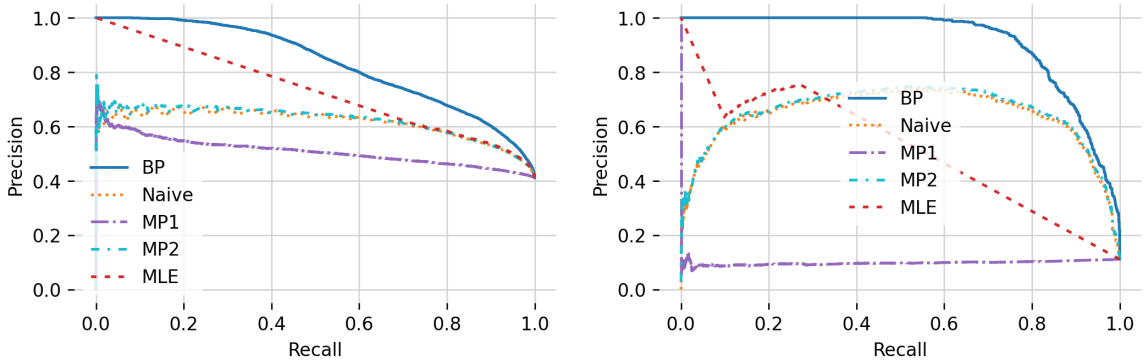
$$p_{i,-j} \leftarrow \frac{1}{T_{i,\cdot}} \sum_{j' \in N_i \setminus j} R_{i,j'}(1 - q_{j',-i}),$$
$$q_{j,-i} \leftarrow \frac{1}{T_{\cdot,j}} \sum_{i' \in N_j \setminus i} R_{i',j}(1 - p_{i',-j}),$$

where $p_{i,-j}$ and $q_{j,-i}$ take values in $[0, 1]$. Under this algorithm, the score of an agent is the average request rate of their orders weighted by the scores of the agents on the other side. An agent gets a high score if they have many bad interactions with agents with low scores. This is an improvement over the naïve algorithm, by weighting interactions with less strategic agents more heavily.

MLE. Maximum Likelihood Estimation (MLE) can be used to estimate the types of all agents by maximizing the likelihood of the observed data. For discrete types, this corresponds to finding the most likely type configuration out of exponentially many possibilities, which is computationally intractable. For tractability, we relax the types of customers and drivers to be continuous in $[0, 1]$. The algorithm aims to maximize over $\boldsymbol{x} \in [0, 1]^n$ and $\boldsymbol{y} \in [0, 1]^m$ the log likelihood of the observed data, which is given by

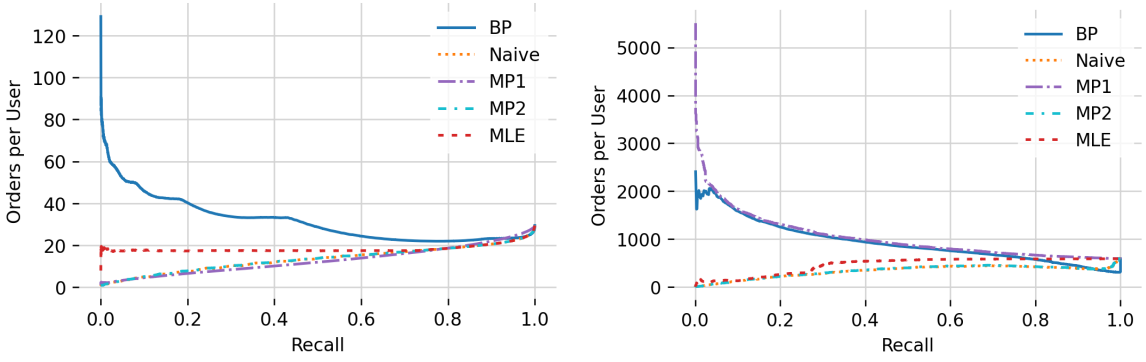
$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\ell=1}^{T_{i,j}} \Big(h_{i,j,\ell} \log \rho^{(x_i, y_j)} + (1 - h_{i,j,\ell}) \log(1 - \rho^{(x_i, y_j)}) \Big).$$

The objective is non-convex, and we use the L-BFGS-B method (in `scipy.optimize`) for optimization. In the solution, many agents have scores at the boundaries (0 or 1), resulting in straight lines in the precision-recall curves in Figure 15.



(a) Customer. (b) Driver.

Figure 15: Precision-recall curves for the additional algorithms on agents with at least one request.



(a) Customer. (b) Driver.

Figure 16: Number of orders per flagged agent against recall for the additional algorithms.

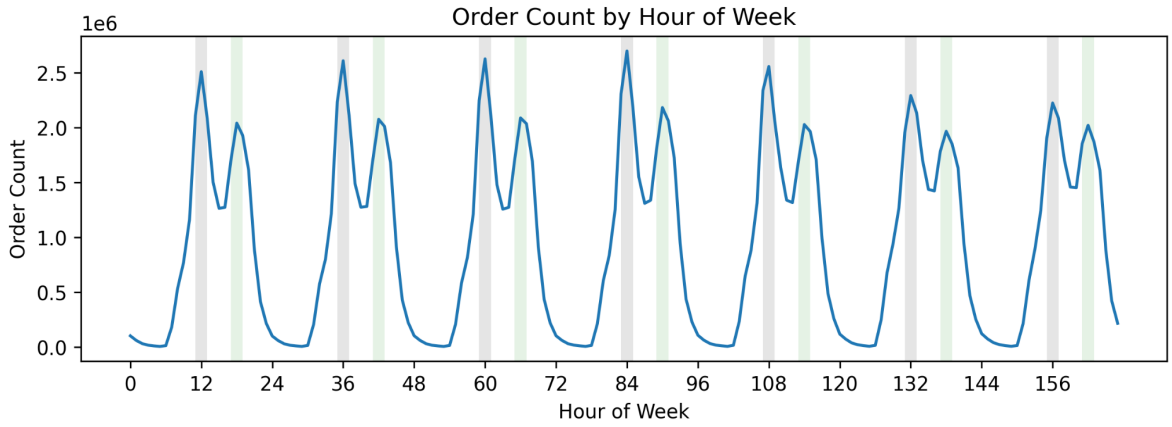


Figure 17: Number of orders by hour of the week. The horizontal axis represents the hour of the week, with 0 being Monday 00:00. The vertical axis represents the number of orders. The gray area represents 11 AM – 1 PM lunchtime, and the green area represents 5 PM – 7 PM dinner time.

C Real Data

Pattern over Time. First, Figure 17 shows the number of orders by hour of the week. As we can see, the number of orders has a clear daily pattern, with peaks around lunchtime and dinner time. The peaks at lunchtime are higher than the peaks at dinner time. The lunchtime peaks are more pronounced during weekdays than weekends, while the peaks at dinner time are similar across days. Moreover, there are much fewer orders during the early morning hours (1 AM – 6 AM) than during the rest of the day.

Figure 18 shows the request rate of the orders created in each hour of the week. The request rate is the proportion of orders that have a refund request. The request rate is lower during afternoon hours and higher during the early morning hours. The pattern is consistent across days.

Orders and Requests per agent. Customers have an average of 36 orders, with a median of 6 orders. Drivers have an average of 1053 orders, with a median of 398 orders.

Figure 19 shows the distribution of the number of orders per agent. Figure 20 shows the distribution of the request rate of orders associated with each customer and driver. As we can see from Figure 20a, around 85% of customers have zero request rate. Figure 20b shows that 20% of drivers have never delivered an order that has been requested for a refund, and the median request rate is 1%.

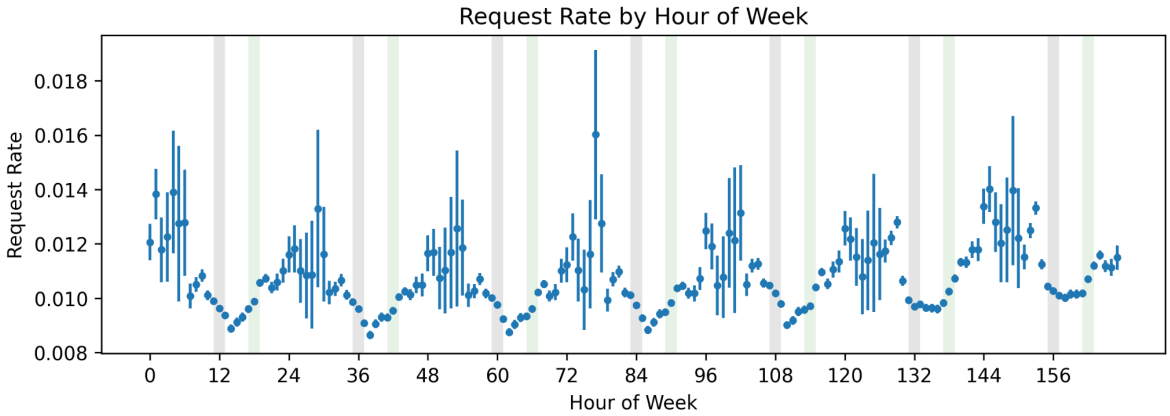
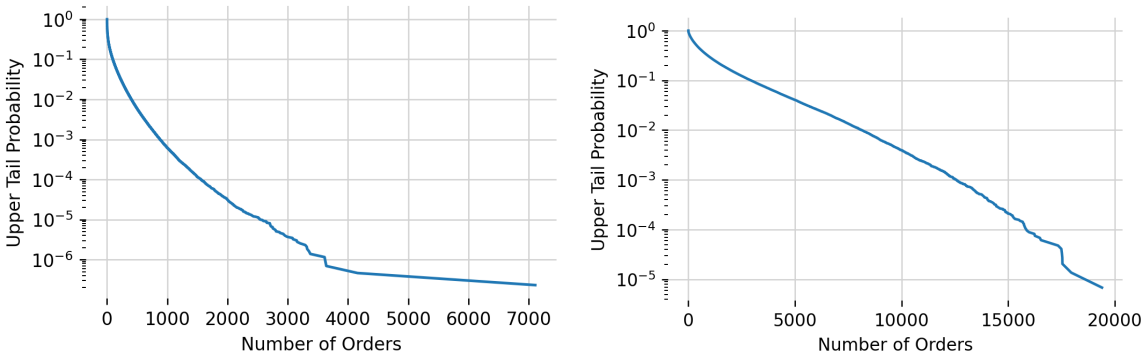


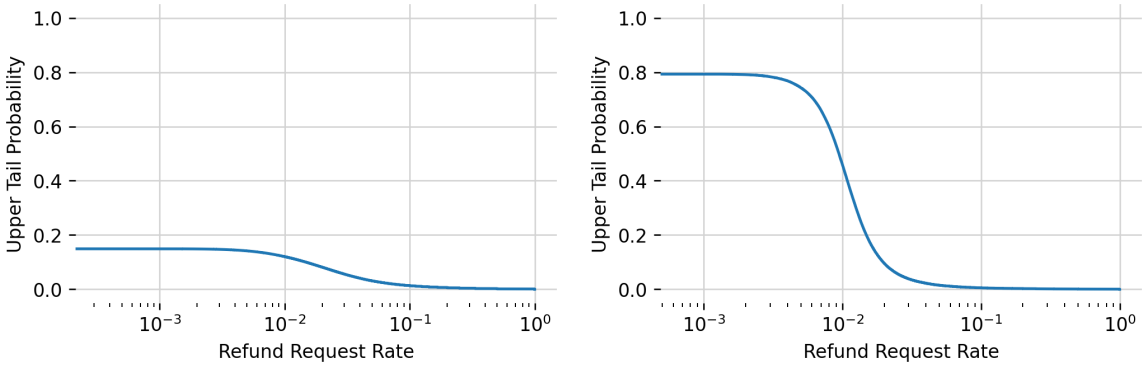
Figure 18: Request rate by hour of the week of the orders. The horizontal axis and colors are the same as in Figure 17.



(a) Customer.

(b) Driver.

Figure 19: Distribution of the number of orders per agent in the real data. The curve represents the CCDF (complementary cumulative distribution function), the probability that an agent has more than a certain number of orders.



(a) Distribution of request rate over customers.

(b) Distribution of request rate over drivers.

Figure 20: Distribution of the request rate of orders associated with each agent in the real data.