# Identifying Refund Hunters with Peer Networks

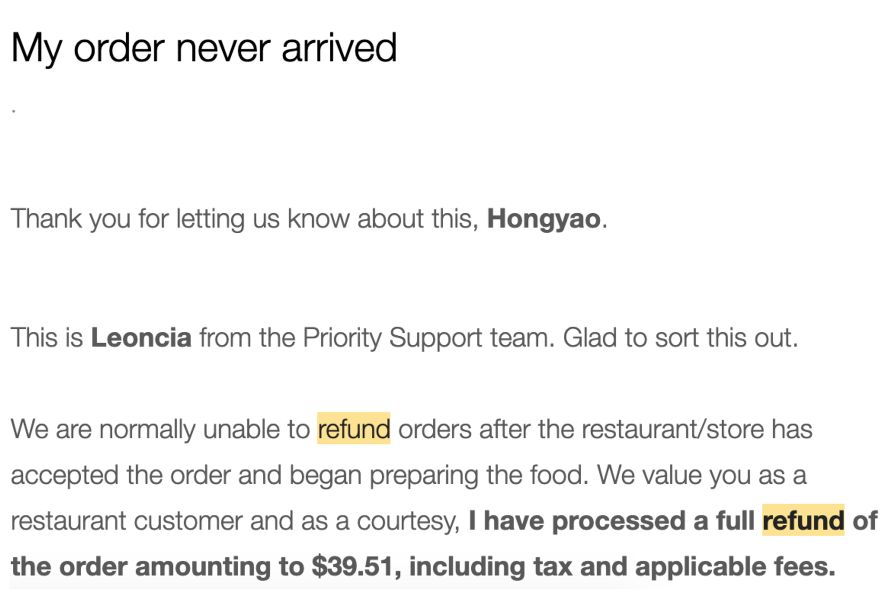
Chenkai Yu[1] Arpit Agarwal[2] Hongyao Ma[1]

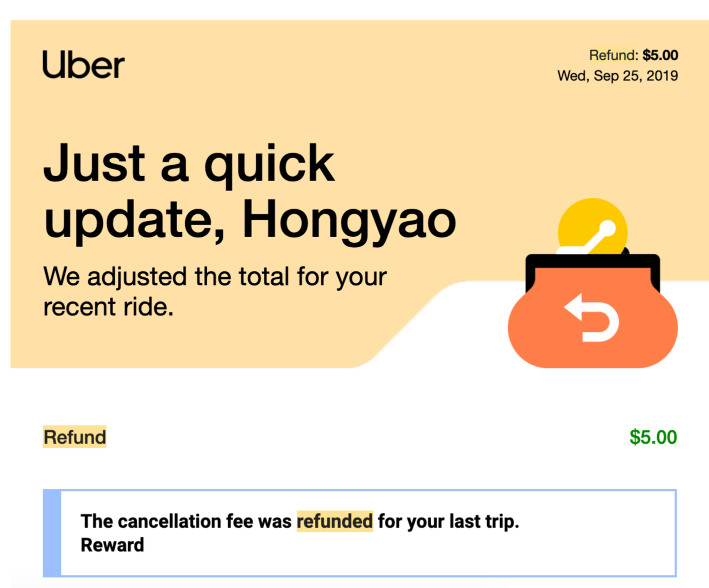
[1]Columbia University [2]IIT Bombay

## Abstract

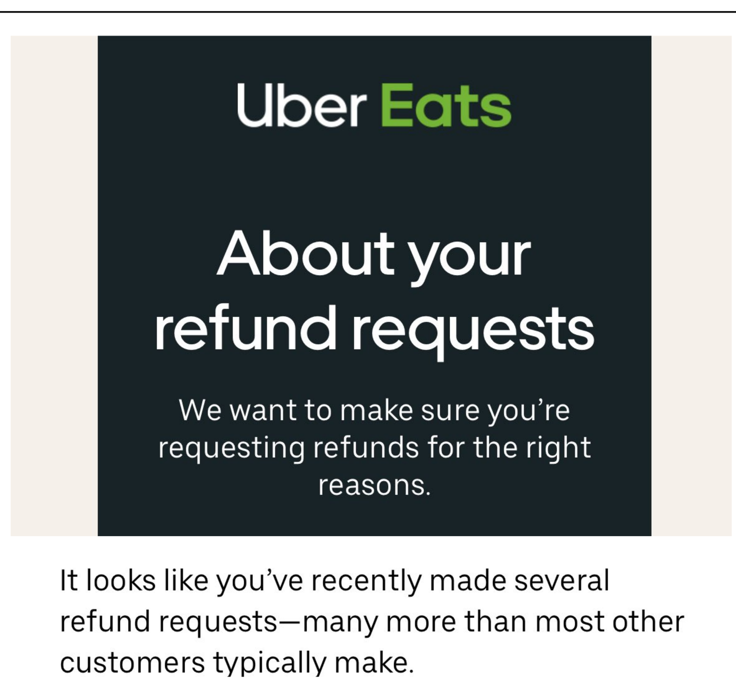
For online platforms such as UberEats and DoorDash, a central challenge in customer service is the lack of ground truth — when a customer reports that an order was never received, it is difficult for support agents to determine if the driver kept the food, a passer-by took the delivery, or the customer 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 variational Bayesian (VB) algorithm for identifying strategic customers and drivers**, considering both their frequency of platform use and the trustworthiness of the participants they had interacted with. When there is a large number of customers each with at least a few orders, we prove that the VB scores (i) recover the correct types on the driver side, and (ii) achieve the highest statistical power on the customer side (i.e., maximizing the true positive rate at any given false positive rate). Extensive experiments on both synthetic data and data from our industry collaborator — a major Southeast Asian platform — demonstrate that the proposed algorithm provides substantial and robust accuracy improvements over a number of benchmarks.

## Customer Support Without Ground Truth

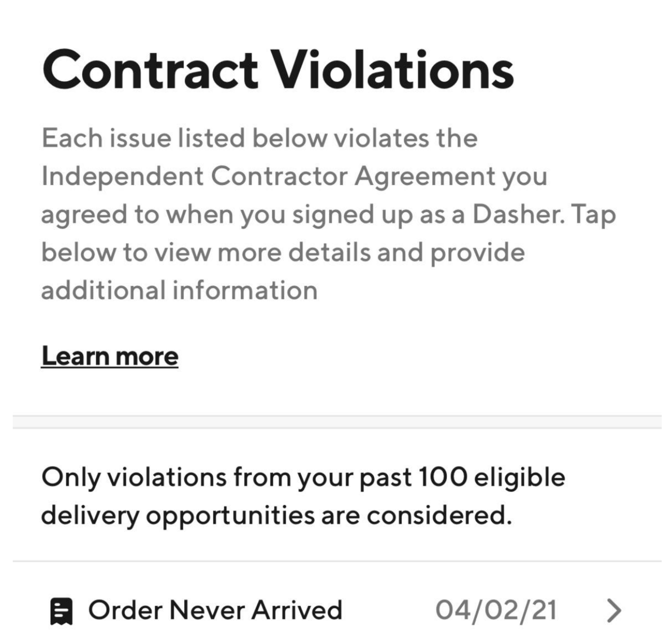



## Existing Guardrails Based on Naïve Scores




(a) UberEats declining a refund request from a customer.




(b) A driver with 1 missing order out of the past 100 deliveries.

### A naïve algorithm

- For each participant, compute the fraction of orders reported missing
- If above some threshold, the participant is flagged as strategic

Weakness — failing to take two important factors into consideration

- *Attribution based on interaction history:* a report from a customer is more likely to be "credible" if many other customers (i.e., the customer's *peers*) also report the same driver for missing orders
- *Number of observations:* a customer with 10 orders in total, all reported missing, is more likely to be strategic than a customer with a total of 1 order which is reported missing

## Model

$n$ customers with types $u_i \in \{0,1\}$; $m$ drivers with $v_j \in \{0,1\}$; $1 =$ strategic.

Each order involves a customer $i$ and a driver $j$:

- The driver $j$ delivers the order with probability $1-\beta\, v_j$, $\beta \in (0,1)$
- A delivered order is taken by a neighbor with probability $\gamma \in (0,1)$
- If there is no order, the customer reports a missing order. Otherwise, the customer still reports with probability $\alpha\, u_i$, $\alpha \in (0,1)$

The platform observes the order history $\mathcal{H}$: for each order, the corresponding customer $i$, driver $j$, and whether it is reported missing.

## Bayesian Inference on the Network

Full Bayesian inference:

- Assume some prior distributions for the types $\boldsymbol{u}$ and $\boldsymbol{v}$
- $\mathbb{P}(\boldsymbol{u},\boldsymbol{v} \,|\, \mathcal{H})$: posterior **joint** distribution of $(\boldsymbol{u},\boldsymbol{v})$ given $\mathcal{H}$

The marginal posterior probability $\mathbb{P}(u_i = 1 \,|\, \mathcal{H})$ (or $\mathbb{P}(v_j = 1 \,|\, \mathcal{H})$) measures the "strategicness" of customer $i$ (or driver $j$).

This works well for small economies, but is not scalable, since the posterior joint distribution is supported on $2^{n+m}$ points.

## Our Algorithm

The algorithm computes a product-form approximation $\mathrm{Ber}(\boldsymbol{p},\boldsymbol{q})$ of the posterior joint distribution $\pi(\cdot,\cdot) = \mathbb{P}(\cdot,\cdot \,|\, \mathcal{H})$ by alternating minimization of the KL divergence:

$$\hat{\boldsymbol{p}}^{(t)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{p}\in[0,1]^n} D_{\mathrm{KL}}\Big(\mathrm{Ber}(\boldsymbol{p}, \hat{\boldsymbol{q}}^{(t-1)}) \,\Big\|\, \pi\Big),$$
$$\hat{\boldsymbol{q}}^{(t)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{q}\in[0,1]^m} D_{\mathrm{KL}}\Big(\mathrm{Ber}(\hat{\boldsymbol{p}}^{(t)}, \boldsymbol{q}) \,\Big\|\, \pi\Big).$$

Local convergence; improvement in every iteration.

**Optimality:** In a large market limit, with high probability, first-iteration driver-side scores are close to the true driver types $\boldsymbol{v}$, and the second-iteration customer-side scores are close to the optimal scores $\boldsymbol{p}^\star$.

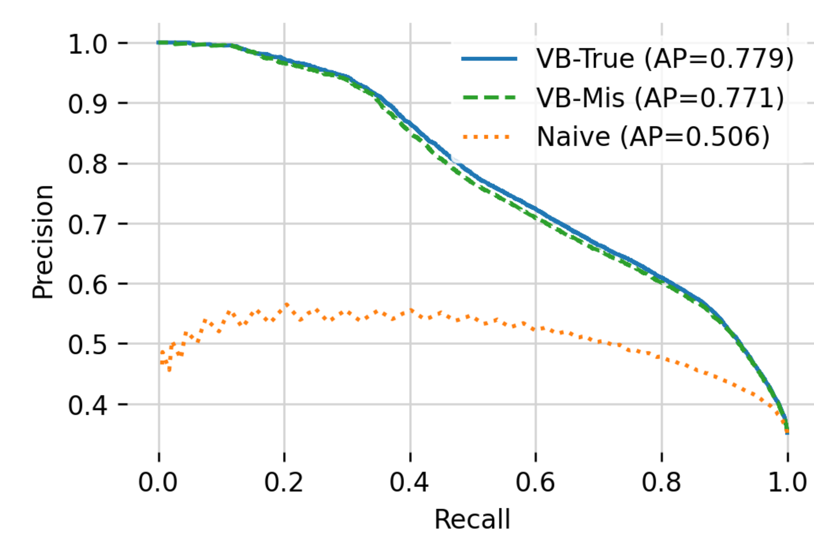
## Simulation

$n = 300\,000$ customers, $m = 10\,000$ drivers, and $5\,000\,000$ orders.

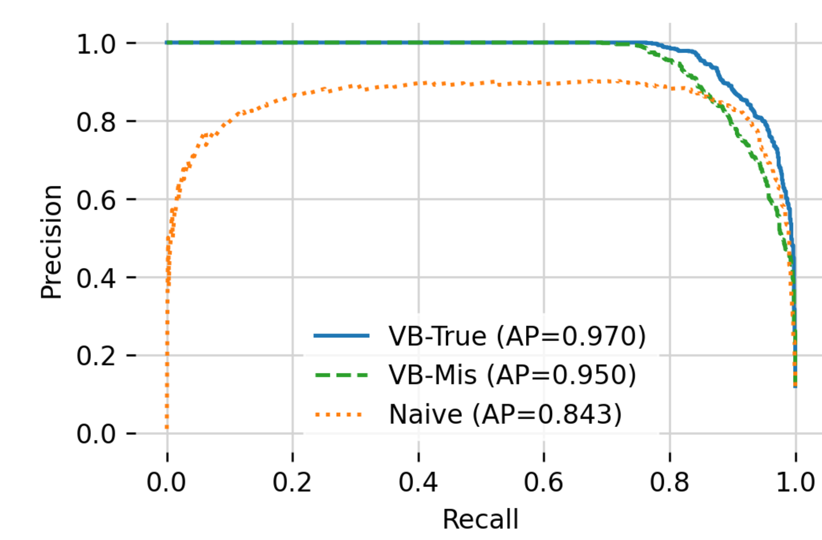
10% strategic participants on each side; strategic customer "misbehaves" w.p. $0.04$; strategic driver $0.03$; restaurants/neighbors $0.003$.

The number of orders per participant is geometrically distributed.

Customers and drivers are matched uniformly at random.

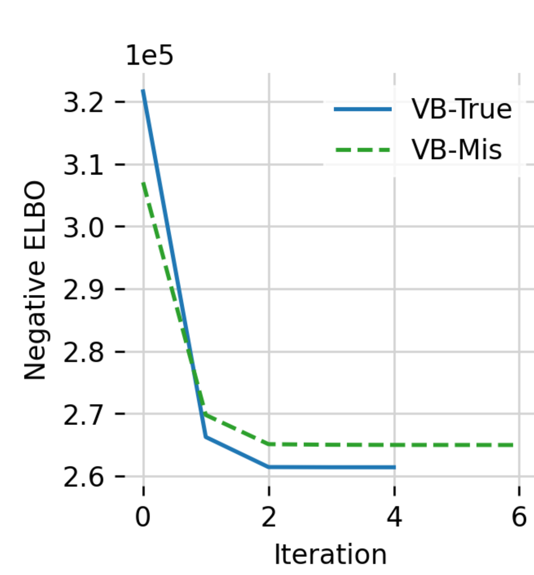



(a) Customer.

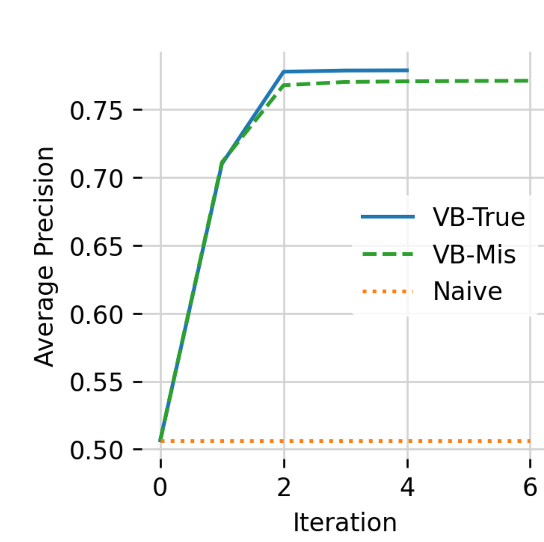



(b) Driver.

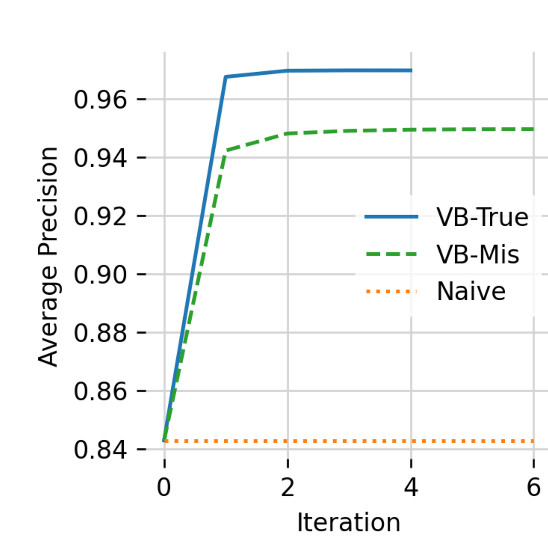
Figure 1. Precision-recall curves on agents with at least one request.




(a) Evidence lower bound.




(b) Customer AP.

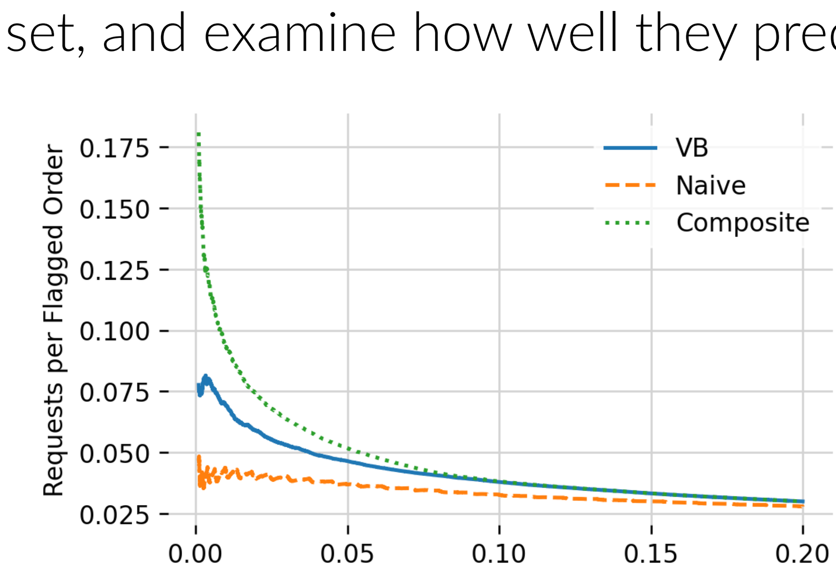



(c) Driver AP.

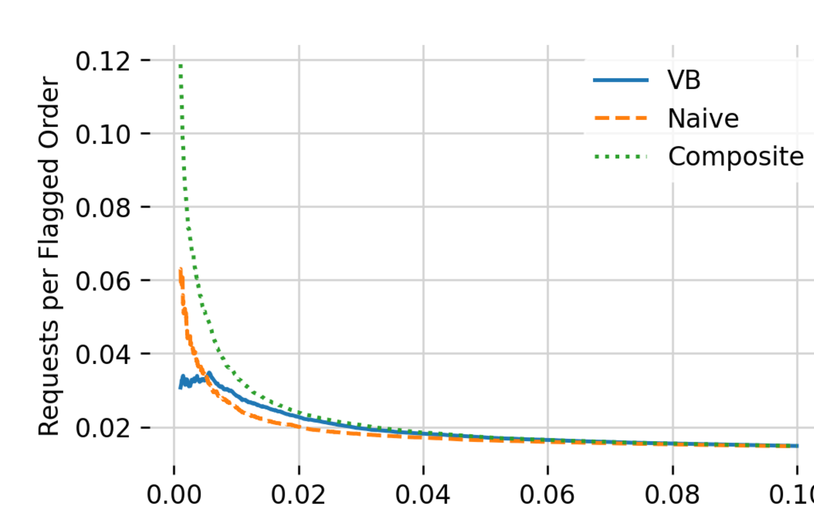
Figure 2. Performance over iterations.

**Real Data.** 164 million orders, 4.3 million customers, 146 thousand drivers.

Split the data into training and testing sets, run algorithms on the training set, and examine how well they predict the test set.




(a) Customer.




(b) Driver.

Figure 3. Surrogate precision-recall curves. A curve near the upper right corner is better.