I Know What You Bought At Chipotle for $9.81 by Solving A Linear Inverse Problem

MICHAEL FLEDER, Massachusetts Institute of Technology, USA
DEVAVRAT SHAH, Massachusetts Institute of Technology, USA

We consider the question of identifying which set of products are purchased and at what prices in a given transaction by observing only the total amount spent in the transaction, and nothing more. The ability to solve such an inverse problem can lead to refined information about consumer spending by simply observing anonymized credit card transactions data. Indeed, when considered in isolation, it is impossible to identify the products purchased and their prices from a given transaction just based on the transaction total. However, given a large number of transactions, there may be a hope.

As the main contribution of this work, we provide a robust estimation algorithm for decomposing transaction totals into the underlying, individual product(s) purchased by utilizing a large corpus of transactions. Our method recovers a (product prices) vector $p \in \mathbb{R}^N_{>0}$ of unknown dimension (number of products) N as well as matrix $A \in \mathbb{Z}^{M\times N}_{\geq 0}$ simply from M observations (transaction totals) $y \in \mathbb{R}^M_{>0}$ such that $y = Ap + \eta$ with $\eta \in \mathbb{R}^M$ representing noise (taxes, discounts, etc.). We formally establish that our algorithm identifies N, A precisely and p approximately, as long as each product is purchased individually at least once, i.e. $M \geq N$ and A has rank N. Computationally, the algorithm runs in polynomial time (with respect to problem parameters), and thus we provide a computationally efficient and statistically robust method for solving such inverse problems.

We apply the algorithm to a large corpus of anonymized consumer credit card transactions in the period 2016-2019, with data obtained from a commercial data vendor. The transactions are associated with spending at Apple, Chipotle, Netflix, and Spotify. From just transactions data, our algorithm identifies (i) key price points (without access to the listed prices), (ii) products purchased within a transaction, (iii) product launches, and (iv) evidence of a new 'secret' product from Netflix - rumored to be in limited release.

Keywords: Blind Compressed Sensing; Alternative Data; Finance; Consumer Credit Card Transactions

ACM Reference Format:
Michael Fleder and Devavrat Shah. 2020. I Know What You Bought At Chipotle for $9.81 by Solving A Linear Inverse Problem . In *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 4, 3, Article 47 (December 2020). ACM, New York, NY. 17 pages. https://doi.org/10.1145/3428332

1 INTRODUCTION

Tracking granular consumer spending is of great interest to advertisers [2], hedge funds [14, 15] banks [4], and others studying the retail economy. Advertisers like Google purchase transactions data to measure in-store retail sales [2]. Similarly, retailers track competitors through such data [22]. And hedge funds utilize transactions data in tracking public companies [13] and informing investment and risk decisions [14]. The prevalence of transactions, primarily via credit and debit cards, has led to anonymized consumer transactions data becoming widely available from a variety of commercial data vendors [1, 14, 22].

Authors' addresses: Michael Fleder, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA, mfleder@mit.edu; Devavrat Shah, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA, devavrat@mit.edu.

2476-1249/2020/12-ART47 $15.00
https://doi.org/10.1145/3428332

Although transactions data details how much consumers spend in *total* at a given vendor; the data does not reveal *what* consumers are buying. For example, a single transaction total at an Apple store for $182.91 does not provide information on which products were purchased or at what prices.

In this work, we are interested in inferring the product(s) a consumer purchases, given knowledge of only the transaction total (a single number). Inferring the products that constitute transaction totals can lead to a wealth of insights. For example, such inference enables estimation of iPhone unit sales at Apple or guacamole sales at Chipotle. Such unit-sales estimates would enable demand estimation for elements upstream the supply chain, e.g. chipmakers for iPhones or food growers for Chipotle.

Result	A	p	N	Noise	Requirements
Ordinary Least Sq.	Known (Full Rank [18])	Unknown	Known	Yes	$M \gg N$
Compressed Sensing	Known (RIP [6, 10])	Unknown	Known	Yes	$M \gg \|p\|_0$
This Work	Unknown (Signature)	Unknown	Unknown	Yes	$M \gg \|p\|_0$

Table 1. Succinct comparison of ours with relevant prior works. Details in **Related Work**.

1.1 Problem Statement

Formally, we have access to M separate, but not necessarily distinct, transaction totals $y \in \mathbb{R}^M_{>0}$ associated with a given company. We assume the company has a finite number of N products with associated prices $p \in \mathbb{R}^N_{>0}$. We treat both N and p as unknown: that is, an unknown number of products with unknown prices. Each transaction total corresponds to the summation of prices for the products purchased. In addition, we include an additive noise term η to account for price variations due to discounts, promotions, taxes, etc. Therefore, we have

$$y = Ap + \eta. \tag{1}$$

where the unknown matrix $A \in \mathbb{Z}^{M \times N}_{\geq 0}$ represents the product decomposition for each transaction. The goal is to identify the number of products N, their associated prices p, and the decomposition of transactions A; all from observation of y only.

1.2 Contributions

A novel inference algorithm. As the main contribution of this work, we develop a simple, iterative and computationally efficient algorithm for inferring N, A and p from y. The algorithm provably recovers N, A precisely, and p approximately, with approximation error dependent on the ℓ_∞-norm of noise η. Our algorithm succeeds if A satisfies Condition 1 (see Section 3.1), which requires that every product is purchased individually at least once. This requires that $M \geq \|p\|_0 = N$. It also guarantees that A has full column rank as required in traditional linear regression (or ordinary least squares). However, Condition 1 does not necessarily require restricted iso-perimetry-like (RIP) conditions which are common in literature on compressed sensing, cf. [6, 10].

The algorithm recovers prices that are distinct and not multiples of each other. Indeed, no algorithm can distinguish if one or more products are sold at the same price (or as different integral

multiples of the same value) without additional side information. Therefore, we require in Theorem 2 for such a condition to hold.

The algorithm as described in Section 2.1 requires solving M exact subset-sum problems which is known to be NP-hard. However it admits a fully polynomial time (in problem parameters) implementation (see Section 2.3) based on approximate subset-sum that achieves similar performance guarantees as established in Theorem 3.

In summary, we provide a simple, iterative and computationally efficient algorithm for solving the system of linear equations *without* knowledge of A, p or N – unlike any prior works (see Table 1). This might be of interest in its own right.

Company	Number of Transactions	Distinct Trans. Totals	Date Range	Our Findings
Netflix	2.6M	3,094	01/16-02/19	Secret Product Ultra HD ($17.08)
Spotify	387K	527	09/17-09/18	Product Launch Detected ($12.99)
Apple	197K	7,685	08/18-10/18	iPhone XS Sales Volume
Chipotle	133K	2.8K	09/18-02/19	Decomposition Err MAPE < 2%

Table 2. Dataset summary of anonymized transactions. The number of transactions is large compared to the number of products offered: $M \gg \|p\|_0$. In addition to recovering product prices accurately, our method reconstructs bill totals with minimal error.

Empirical validation. We apply the algorithm to anonymized consumer transactions data[1] based on credit and debit card purchases at Netflix, Spotify, Apple and Chipotle. The details of the data in terms of number of transactions, time range and unique totals are shown in Table 2. We measure performance of the algorithm in terms of recovering (a) the number of key products N, (b) their corresponding prices p, and (c) the decomposition (A) of transaction totals into products purchased. In addition, we discuss the implications of accurate inference in terms of detecting product launches and hidden / non-advertised product offerings.

For (a) and (b), it is easy to verify full recovery (or not) of all products for Netflix, since Netflix has few product offerings. As shown in Table 4, we recover all the published offerings by Netflix. In addition, we identify two additional 'hidden' offerings which seem to agree with limited-release products [23]. Table 4 shows a median error in the recovered prices of less than 0.2% (or, less than 4 cents). Accurate recovery of the number of products and their prices implies that that each transaction total is accurately modeled in terms of identifying the product(s) purchased, i.e. performance in terms of (c). For Chipotle, which offers a larger and more complex set of offerings, we reconstruct transaction totals within MAPE of 1.2% using 12 key product prices only!

Our findings enable a plethora of insights into time-varying company product catalogues. For example, for Spotify and Apple, we utilize our methods to automatically detect product launches. Specifically, we automatically detect a new Spotify product offering at a new price - starting in the

[1]We utilized anonymized debit and credit card transactions data provided by alternative data company Second Measure [22] for the purpose of conducting this work. Table 2 provides dataset details.

month of April 2018, which matches the reported fact [20]. Similarly, for Apple our method detects the launch of the iPhone XS Max in September 2018 from anonymized transaction totals only.

Collectively, these experiments verify that our method is able to recover product prices as well as decompose transaction totals accurately across different types of businesses: from Netflix and Spotify with few offerings, to Chipotle and Apple with extremely complex product offerings. Indeed, our method is likely to have more impactful consequences such as that represented by Table 3 which is the estimated sales volume by price range at Chipotle.

Price Range	Estimated Sales	Example Products
\$1.5 - \$4	15.6%	chips, drinks, guacamole
\$9-\$11	79.1%	\$9.74 chicken burrito, steak bowl
\$11-\$20	5%	double steak bowl

Table 3. Inferred sales at Chipotle by product-price range using transaction data. See Section 4.6.

1.3 Related Work

Methodologically, this work is about finding p in the linear system equations (1) by observing y *without* any knowledge of N or A.

Classically, ordinary least squares or linear regression accurately estimates p as $(A^TA)^{-1}A^Ty$ as long as A has full column rank and we *observe* A. For example, [18] is one of the earliest results to provide nearly complete characterization for when such an approach works. This approach requires $M \gg N$ for good recovery of p. Of course, such an approach requires knowledge of A and hence is not applicable in our setting.

A modern take on the question and closer to our setting is popularized as *compressed sensing* cf. [3, 5, 6, 10]. In compressed sensing, p is assumed to be sparse but we do not know which components are non-zero. Let $\|p\|_0 = |\{i : p_i \neq 0,\ i \in [N]\}|$ denote[2] the sparsity of p. Then, as long as A is *designed* carefully (specifically, satisfies conditions like the Restricted Isometry Property (RIP)) and is *known*, then p can be recovered with $M \gg \|p\|_0$. Again, such approaches require carefully designed A as well as knowledge of it, neither of which hold in our setting.

In summary, methodologically, we advance the state-of-the-art for solving "inverse problems" like Eq. (1) when A, p, and N (dimension of p) are all unknown, and p is dense. Our method requires only simple conditions to be satisfied for inversion to succeed. Furthermore, our approach is robust to noisy data.

It is worth remarking that consumer transactions data has emerged as an important "alternative data" source utilized for asset pricing in recent years cf. [9, 17, 24]. Transactions data is now broadly available from commercial data vendors [12, 22]. For example, this work utilized data obtained from [22].

2 ALGORITHM: INFERRING N, p, A

2.1 Description

We start with a narrative description of the algorithm. Subsequently, we present a more detailed, pseudocode version. The algorithm outputs estimators $\hat{p}$ for p, $\hat{N} = \|\hat{p}\|_0$ for N, and $\hat{A}$ for A. The estimates are such that $y \approx \hat{A}\hat{p}$. In addition to y, the algorithm takes as input parameter $\delta > 0$ as proxy for $\|\eta\|_\infty$, and $k \in \mathbb{Z}_{>0}$ that constrains repeats of a product within a single transaction.

[2]We use the notation $[x] = \{1, \ldots, x\}$.

- Input: transaction totals $y \in \mathbb{R}_{>0}^{M}$; error tolerance $\delta > 0$; and $k \in \mathbb{Z}_{>0}$ constraining repeats of a product in a single transaction.
- Reduce y to distinct values; sort from smallest to largest:

$$b = (b_1, \ldots, b_M) \leftarrow \text{sorted}\big(\{y_1, \ldots, y_M\}\big).$$

- Let $\hat{p}$ denote the inferred, ordered set of product prices. Initialize $\hat{p} \leftarrow \{b_1\}$; set $\hat{A}_{1,1} = 1$ and $\hat{A}$ to be a 1×1 matrix. Going forward, we will increase the dimension of $\hat{A}$, but all the entries in the first row other than $\hat{A}_{1,1}$ will be set to 0 in that case.
- Repeat the following for each $b_i \in b$ for $i \geq 2$. Let $\hat{p} = (\hat{p}_\ell)$ be the current estimate of p and $\hat{A} \in \mathbb{Z}_{\geq 0}^{(i-1)\times|\hat{p}|}$ be current estimate of A. Find $J(i) = (a_1, \ldots, a_{|\hat{p}|}) \in [k]^{|\hat{p}|}$ as a solution to

$$\text{minimize} \quad \frac{1}{(1+\sum_\ell a_\ell)} \left| \left(\sum_{\ell \leq |\hat{p}|} a_\ell \hat{p}_\ell \right) - b_i \right| \qquad \text{over} \qquad a_\ell \in \{0, \ldots, k\},\ \ell \leq |\hat{p}| \tag{2}$$

  We consider two cases:
  - $|b_i - J(i)^T \hat{p}| \leq \delta \cdot (|J(i)|_1 + 1)$: add ith row, $\hat{A}_i$, to $\hat{A}$ of length $|\hat{p}|$ with $\hat{A}_i$ equal to $J(i)$ in the $|\hat{p}|$ positions. That is, the ith total can be decomposed amongst the product prices in $\hat{p}$ recovered thus far; and we record the decomposition in $\hat{A}$.
  - Otherwise, b_i is not well approximated by any combination of product prices found thus far. Hence it must be a product price not yet encountered, and needs to be added to $\hat{p}$: augment $\hat{p} \leftarrow \hat{p} \cup \{b_i\}$; increase the number of columns in $\hat{A}$ by adding an all zeros column to the existing $\hat{A}$ of length $i-1$; and then add as the ith row $\hat{A}_i$, a vector with all 0s but a single 1 in the $|\hat{p}|$ position.
- Output: after iteration through all entries in b, output $\hat{p}$, $\hat{A}$ and $\hat{N} = \|\hat{p}\|_0$.

The 'pseudo-code' description algorithm is given below.

Algorithm 1 Estimating N, p and A.

Input: y, δ, k ▷ $y \in \mathbb{R}_{>0}^{M}$, $\delta \in \mathbb{R}_{>0}$, $k \in \mathbb{Z}_{>0}$
$b \leftarrow \text{sorted}\Big(\{y_1, \ldots, y_M\}\Big)$ ▷ sorted, distinct bill totals, from smallest to largest
Let $\sigma : [M] \to [M]$ denote the permutation induced by this sorting, i.e. $b_i = y_{\sigma(i)}$.
$\hat{p} \leftarrow \{b_1\}$ ▷ inferred product prices
$\hat{A}_{\sigma(1)} \leftarrow (1, 0, 0, \ldots)$
for $i = 2$ **to** $|b|$ **do**
  Let $J(i) = (a_1, \ldots, a_{|\hat{p}|}) \in [k]^{|\hat{p}|}$ be a minimizer of the optimization problem

$$\text{minimize} \quad \frac{1}{(1+\sum_\ell a_\ell)} \left| \left(\sum_{\ell \leq |\hat{p}|} a_\ell \hat{p}_\ell \right) - b_i \right| \qquad \text{over} \qquad a_\ell \in \{0, \ldots, k\},\ \ell \leq |\hat{p}|$$

  if $|b_i - J(i)^T \hat{p}| > \delta \cdot (|J(i)|_1 + 1)$ **then**
    $\hat{p} \leftarrow \hat{p} \cup \{b_i\}$, $\hat{A}_{\sigma(i)}$ equals the row vector with all 0s but a single 1 in the $|\hat{p}|$ position.
  else
    $\hat{A}_{\sigma(i)}$ equals $J(i)$ in the first $|\hat{p}|$ positions, and 0s otherwise.
  end if
end for
return $\hat{p}$, $\hat{A}$.

2.2 Parameters k, δ

The parameter δ is a proxy for the noise tolerance in the form of the ℓ_∞ norm of noise vector η. In particular, δ captures price-variations from regional taxes, promotions, etc. The parameter $k \geq 1$ is to bound the number of times the same product can be repeated in a transaction. In effect, k allows for distinct products with prices that are multiple of each other as long as the multiple is larger than k. It also captures the 'prior' information about how often the same product might repeat within the same transaction.

2.3 Polynomial Time Implementation

The algorithm described above requires sorting M elements which takes $O(M \log M)$ time. Then, iterating over the sorted values, the most expensive step per iteration is solving the optimization problem in Eq. (2). The exact problem is the classical subset-sum problem (see below for precise description) which is known to be NP-hard. However, as described in the algorithm, we only need to solve Eq. (2) with an approximation guarantee to make decisions in decomposing a given transaction total. As it turns out, the subset-sum problem has a fully polynomial time approximation algorithm. We shall argue that such an approximation algorithm is sufficient to obtain the decomposition as per the description in Section 2.1.

Subset-sum problem. Consider a set of N positive integers $S = \{x_1, x_2, \ldots, x_N\}$. Given an integer t, the question is whether there exists a subset $S' \subset [N]$ so that $\sum_{i \in S'} x_i = t$. A simple, exhaustive search algorithm is to compute the summation of all possible 2^N subsets of S and verify whether any of them is equal to t or not. Clearly, this is computationally expensive. Indeed, this problem is known to be NP-hard, cf. see [8, 16].

Approximate Subset-sum algorithm. As it turns out, Subset-sum has a simple approximation algorithm (again, see textbooks like [8, 16] for details). To that end, consider approximate error tolerance $\varepsilon > 0$. We wish to answer whether there exists a set $S' \subset [N]$ such that $(1-\varepsilon)t \leq \sum_{i \in S'} x_i \leq t$. There is a deterministic algorithm with computational cost $N^2 \log t/\varepsilon$ that provides the answer to this question.

The algorithm works as follows (also refer to [8, 16] for details). Initially, we start with candidate set $L = \{0\}$. Without loss of generality, let $x_1 \leq x_2 \leq \cdots \leq x_N$. Iteratively, we consider them in the increasing order. In iteration $i = 1, \ldots, N$ do the following.

- (*New Sums*) Set $L = L \cup L + x_i$, where $L + x_i = \{z + x_i : z \in L\}$.
- (*Trim*) Sort elements in L in increasing order: $z_1, \ldots, z_m$. Iteratively, for $j = 2, \ldots, m$ do the following starting with last $= z_1$ and $L' = \{z_1\}$.
  - If $z_j \leq$ last, do nothing.
  - Else, $L' = L' \cup \{z_j\}$ and last $= z_j$.
- (*Set*) $L = \{z : z \in L', z \leq t\}$.

At the end of the above iterations, consider $\tilde{t} = \max\{z \in L\}$. If $\tilde{t} \geq (1 - \varepsilon/N)t$ then we have found a subset (by tracking the corresponding subset of $[N]$) that has a sum within $(1-\varepsilon)t$ and t. Otherwise, no such subset exists.

To see the correctness of the algorithm, we observe that if $\varepsilon = 0$, then we capture *all* possible summations less than t that are achievable by subsets of S. It can be verified inductively that the step of *Trim* across N iterations removes an element from this collection only if there is another element that is at least within $(1 - \varepsilon/N)^N$ present in the final outcome. And $(1 - \varepsilon/N)^N \geq 1 - \varepsilon$. That is, if the set of all possible summations produced by the above does not contain anything within $[(1-\varepsilon)t, t]$, then there is no such subset; on the other hand, if there is a summation within that range, then it must correspond to a such a subset.

Computational cost of approximate algorithm. While the computation cost of the exhaustive algorithm is 2^N, the cost of the approximation algorithm scales as N times the maximum size of set L in any iteration. The elements in L are non-negative and no larger than t. Further, the ratio of any two elements is at least $1-\varepsilon/N$ or $(1-\varepsilon/N)^{-1}$. That is, the maximum number of elements is at most $N \log t/\varepsilon$. Hence, the total computation cost is $O(N^2 \log t/\varepsilon)$. That is, we have a fully polynomial time approximation algorithm.

Using approximate subset-sum for our problem. The key step of the Algorithm in Section 2.1, in iteration i compares whether there exists subset $J(i)$ of prices discovered thus far that is within additive error of $\delta(|J(i)|+1)$ of b_i or not. That is, we are looking for multiplicative error of $\delta(|J(i)|+1)/b_i$. We will utilize $\varepsilon = \delta(|\hat{p}|+1)/b_i$ in iteration i for approximate subset-sum instead of exact subset-sum. As we shall argue in Theorem 3, such an algorithm (with approximate subset-sum) results in the same outcome as the exact subset-sum. Naturally, the computation cost scales inversely with $1/\delta$ and proportionately to $\max_i b_i$.

In summary, the approximate subset-sum method may produce a solution to Eq. (2) as less than $\delta(|J(i)|+1)$ in iteration i if there exists any $J(i) = (a_1, \ldots, a_{|\hat{p}|}) \in [k]^{|\hat{p}|}$ such that $|\sum_\ell a_\ell \hat{p}_\ell - b_i| \leq \delta(|J(i)|+1)+\delta(|\hat{p}|+1)$. And in the Algorithm of Section 2.1, using such an approximate subset-sum routine may result in a decision as if $|\sum_\ell a_\ell \hat{p}_\ell - b_i| \leq \delta(|J(i)|+1)$.

On the other hand, if for any $J(i) = (a_1, \ldots, a_{|\hat{p}|}) \in [k]^{|\hat{p}|}$, we had $|\sum_\ell a_\ell \hat{p}_\ell - b_i| > \delta(|J(i)|+1) + \delta(|\hat{p}|+1)$, then the Algorithm of Section 2.1 using such an approximate subset-sum routine will declare $|\sum_\ell a_\ell \hat{p}_\ell - b_i| > \delta(|J(i)|+1)$ and result in adding a new price element to $\hat{p}$.

Using approximate subset-sum, some fine print. It is worth remarking that the actual numbers are not integers in our setting while the subset-sum described above is for integers. To that end, note that ours has strictly positive, rational numbers. Further, since we are interested in approximation within additive error of δ, we shall assume that each value is integral multiple of δ. That is, in effect, we convert the question to the setting of integers.

Finally, while the approximate subset-sum as stated above attempts to find a subset whose summation is within $[(1-\varepsilon)t, t]$ for a given target t, i.e. only one-sided error, it can be converted to two sided error by setting the target to $(1+\varepsilon)t$ and approximation error to $1-(1-\varepsilon)/(1+\varepsilon) \approx 2\varepsilon$.

3 MAIN RESULT

3.1 Guarantee

Here we state the result regarding correctness and robustness of algorithm. To that end, we assume that the underlying data satisfies the constraint that every product is purchased alone at least once. This is formalized as follows.

Condition 1 (Signature Condition). *Given matrix $A \in \mathbb{R}^{M\times N}$, A satisfies the Signature Condition if for each $i \in [N]$, there exists $j(i) \in [M]$ such that $A_{j(i)i} = 1$ and $A_{j(i)i'} = 0$ for all $i' \neq i$.*

Now we state the main result. We shall assume that $k = 1$ in the algorithm, i.e. each product is repeated at most once. The proof naturally extends for $k \geq 1$ (and with A non-negative and integer-valued).

Theorem 2. *Let $A \in \{0,1\}^{M\times N}$ satisfy Condition 1. Let $p \in \mathbb{R}^N_{>0}$, where for any $S_1, S_2 \subset [N]$, with $S_1 \neq S_2$*

$$\left| \sum_{i\in S_1} p_i - \sum_{i'\in S_2} p_{i'} \right| > 2\delta N \tag{3}$$

for some given $\delta > 0$. Let $y = Ap + \eta$ with $\|\eta\|_\infty < \delta$. Then, with input y, δ and $k = 1$, the algorithm described in Section 2.1 recovers $\hat{p}$ and $\hat{A}$ so that $\|\hat{p} - p\|_\infty \le \delta$ and $\hat{A} = A$.

As discussed earlier, the algorithm described in Section 2.1 requires solving the exact subset-sum problem which is computationally hard. In Section 2.3, we described its variation based on the approximate subset-sum algorithm. In particular, the approximate subset-sum algorithm produces an answer to Eq. (2) within additive error of $\delta(|\hat{p}|+1)$. Below we state the result about the correctness of such an algorithm, similar to Theorem 2 above.

THEOREM 3. *Let $N \ge 4$ and $A \in \{0,1\}^{M \times N}$ satisfy Condition 1. Let $p \in \mathbb{R}^N_{>0}$, where for any $S_1, S_2 \subset [N]$, with $S_1 \ne S_2$*

$$\left| \sum_{i \in S_1} p_i - \sum_{i' \in S_2} p_{i'} \right| > 4\delta(N+1) \tag{4}$$

for some given $\delta > 0$. Let $y = Ap + \eta$ with $\|\eta\|_\infty < \delta$. Then, with input y, δ and $k = 1$, the algorithm using approximate subset-sum described in Section 2.3 recovers $\hat{p}$ and $\hat{A}$ so that $\|\hat{p} - p\|_\infty \le \delta$ and $\hat{A} = A$.

3.2 Need for conditions

The Signature condition 1 effectively states that each product is purchased alone, at least once. This is natural, and guarantees the full column rank of A that is needed to solve the linear equation.

To understand the condition in Eq. (3), suppose $\delta = 0$, i.e. there is no noise in the data. In that case, Eq. (3) would indicate a difference in the summation of any two distinct subsets of product prices – indeed, if that were not the case, *no algorithm could recover A* uniquely. For example, distinguishing between products is impossible if one or more products are sold at the same price (or as different integral multiples of the same value) without additional side information. In that sense, our algorithm recovers A and p in a robust manner under the 'quantitative' version of the necessary recovery condition as described in Eq. (3).

We note that our method may still yield insights even if Condition (3) is violated. For example, if multiple products are identically priced, our algorithm simply groups those products together to be treated as a single item.

3.3 Proof of Theorem 2

Without loss of generality, let the elements of p be ordered such that $p_1 < \cdots < p_N$. From Eq. (4), we obtain the following implications: (i) For choice of $S_1 = \{1\}$ and $S_2 = \emptyset$, it follows that $p_1 > 2\delta$; and (ii) if $N \ge 2$, then for $1 \le i < N$, $p_{i+1} > p_i + 4\delta$ for choice of $S_1 = \{i\}$ and $S_2 = \{i+1\}$. As a consequence, we have that for $1 \le i \le N$,

$$p_i > (i-1)4\delta + 2\delta. \tag{5}$$

Next, we shall argue inductively that it is feasible to find $\hat{p}_i$, $1 \le i \le N$, so that $|\hat{p}_i - p_i| \le \delta$. Now since $A \in \{0,1\}^{M \times N}$, for each $j \in [M]$, $y_j = \sum_{i \in S_j} p_i + \eta_j$, with $S_j \subseteq [N]$ and $|\eta_j| \le \delta$. From Eq. (5), $p_i > 2\delta$; and therefore if $S_j \ne \emptyset$ then $y_j > \delta$. We define

$$J^1 := \{j \in [M] : y_j > \delta\}, \quad j_1 := \arg\min_{j \in J^1} y_j, \text{ and}$$

$$J(1) := \{j \in J^1 : y_j = p_1 + \eta_j\}.$$

We set $\hat{p}_1 := y_{j_1}$, and $A_{j_1,1} := 1$. To justify this, we argue that $y_{j_1} = p_1 + \eta_{j_1}$, and hence $|\hat{p}_1 - p_1| < \delta$. By Condition 1, for each $i \in [N]$, there exists some index $j(i) \in [M]$ such that $|y_{j(i)} - p_i| \le \delta$; and hence $j(i) \in J^1$ since $y_{j(i)} \ge p_i - \delta > \delta$. Clearly $J(1) \ne \emptyset$. Effectively, we want to argue that

$j_1 \in J(1)$. To that end, suppose otherwise. Then, there exists $S \subset [N]$ such that $S \neq \emptyset, S \neq \{1\}$ and $y_{j_1} = \sum_{i \in S} p_i + \eta_{j_1}$. Then

$$\begin{aligned} y_{j_1} &> \sum_{i \in S} p_i - \delta, && \text{since } |\eta_{j_1}| < \delta, \\ &> p_1 + 2N\delta - \delta, && \text{by Eq. (4)}, \\ &\geq p_1 + \delta, && \text{since } N \geq 1, \\ &> y_j, && \text{for any } j \in J(1) \subset J^1. \end{aligned}$$

But this is a contradiction, since $y_{j_1} \leq y_j$ for all $j \in J^1$. Thus, $j_1 \in J(1)$, $S = 1$, and $y_{j_1} = p_1 + \eta_{j_1}$. Thus, we have found $\hat{p}_1 = y_{j_1}$ such that $|\hat{p}_1 - p_1| < \delta$.

If $N \geq 2$, then for any $j \in J^1$ with $y_j = \sum_{i \in S} p_i + \eta_j$ such that $S \cap \{2, \ldots, N\} \neq \emptyset$, with notation $p(S) = \sum_{i \in S} p_i$,

$$\begin{aligned} |\hat{p}_1 - y_j| &= |p_1 - y_j + \hat{p}_1 - p_1|, \\ &= |p_1 - p(S) - \eta_j + \hat{p}_1 - p_1|, \\ &\geq |p_1 - p(S)| - |\eta_j| - |\hat{p}_1 - p_1|, \\ &> 2N\delta - \delta - \delta, \\ &> 2\delta, && (N \geq 2). \end{aligned}$$

And if $S = \{1\}$, then $|\hat{p}_1 - y_j| < 2\delta$. Therefore, we set

$$J^2 \leftarrow J^1 - \{j \in J^1 : |y_j - \hat{p}_1| < 2\delta\}$$

Note, the above implies:

$$A_{j,1} = 1 \text{ if } |\hat{p}_1 - y_j| < 2\delta, \text{ for } j \in J^1.$$

Clearly

$$\begin{aligned} j \in J^2 \iff\ & j \in [M], y_j = p(S) + \eta_j, \\ & \text{such that } S \cap \{2, \ldots, N\} \neq 0. \end{aligned}$$

Now suppose, inductively, we have found $\hat{p}_1, \ldots, \hat{p}_n$, $1 \leq n < N$ so that $|\hat{p}_i - p_i| < \delta$ for $1 \leq i \leq n$ and

$$\begin{aligned} j \in J^{n+1} \iff\ & j \in [M], y_j = p(S) + \eta_j, \\ & \text{such that } S \cap \{n+1, \ldots, N\} \neq \emptyset. \end{aligned}$$

To establish the inductive step, (with $\hat{p}(S) = \sum_{i \in S} \hat{p}_i$)

$$\begin{aligned} \hat{p}_{n+1} &\coloneqq y_{j_{n+1}} \quad \text{where} \quad j_{n+1} \in \arg\min_{j \in J^{n+1}} y_j, \\ J^{n+2} &\leftarrow J^{n+1} - \{j \in J^{n+1} : |y_j - \hat{p}(S)| < (|S| + 1)\delta, \\ &\qquad \text{for some } S \subset [n+1]\}. \end{aligned}$$

We shall argue that $|\hat{p}_{n+1} - p_{n+1}| < \delta$ by showing that $y_{j_{n+1}} = p_{n+1} + \eta_{j_{n+1}}$ and establishing

$$\begin{aligned} j \in J^{n+2} \iff\ & j \in [M], y_j = p(S) + \eta_j, \\ & \text{such that } S \cap \{n+2, \ldots, N\} \neq \emptyset. \end{aligned}$$

To that end, let

$$\begin{aligned} &y_{j_{n+1}} = p(S) + \eta_{j_{n+1}},\ S \subset [N], \\ &\quad \text{with } S \cap \{n+1, \ldots, N\} \neq \emptyset, \text{(Inductive Hypothesis)}, \\ &J(n+1) \coloneqq \{j \in J^{n+1} : y_j = p_{n+1} + \eta_j\}. \end{aligned}$$

We want to argue that $j_{n+1} \in J(n+1)$. Suppose otherwise: that $S \neq \{n+1\}$, then

$$\begin{aligned} y_{j_{n+1}} &> p(S) - \delta, \\ &= (p(S) - p_{n+1}) + p_{n+1} - \delta, \\ &> 2\delta + p_{n+1} - \delta, \qquad (6) \\ &= \delta + p_{n+1}, \\ &> y_j \quad \text{for any } j \in J(n+1). \end{aligned}$$

Where Eq. (6) follows from the fact that since $S \cap \{n+1, \ldots, N\} \neq \emptyset$ and $S \neq \{n+1\}$, it must be that $p(S) \geq \min\{p_1 + p_{n+1}, p_{n+2}\}$. In either case, using Eq. (4) and that $N \geq 1$, it follows that $p(S) - p_{n+1} > 2\delta$. We note that due to the Condition 1 and inductive hypothesis about J^{n+1}, it follows that $J(n+1) \neq \emptyset$. But $y_{j_{n+1}}$ is the minimal value of y_j for $j \in J^{n+1}$. This is a contradiction. Therefore $S = \{n+1\}$. That is, $\hat{p}_{n+1} = y_{j_{n+1}}$, satisfies $|\hat{p}_{n+1} - p_{n+1}| < \delta$.

Now consider any set $S \subset \{1, \ldots, n+1\}$ and any $j \in J^{n+1}$ such that $y_j = \sum_{i \in S'} p_i + \eta_j$ with $S' \cap \{n+2, \ldots, N\} \neq \emptyset$. Using notation $\hat{p}(S) = \sum_{i \in S} \hat{p}_i$ we have

$$\begin{aligned} |\hat{p}(S) - y_j| &= |p(S) - y_j + \hat{p}(S) - p(S)|, \\ &= |p(S) - p(S') - \eta_j + \hat{p}(S) - p(S)|, \\ &\geq |p(S) - p(S')| - |\eta_j| - |\hat{p}(S) - p(S)|, \\ &> 2N\delta - (1 + |S|)\delta, \\ &\geq (|S| + 1)\delta, \end{aligned}$$

where we have used the fact that $|S| + 1 \leq N$. Therefore if

$$J^{n+2} \leftarrow J^{n+1} - \Big\{ j \in J^{n+1} : |y_j - \hat{p}(S)| \leq (|S| + 1)\delta, \text{ for some } S \subset [n+1] \Big\},$$

then it follows that

$$\begin{aligned} j \in J^{n+2} \iff &j \in [M], y_j = p(S) + \eta_j \\ &\text{such that } S \cap \{n+2, \ldots, N\} \neq \emptyset. \end{aligned}$$

Also, $A_{j,i} = 1$ for all $j \in J^{n+1}$ and $i \in S \subset [n+1]$ such that $|y_j - \hat{p}(S)| \leq (|S| + 1)\delta$.

This complete the induction step and establishes the desired result that we can recover $\hat{p}$ so that $\|\hat{p} - p\|_\infty \leq \delta$ and recovery of A.

3.4 Proof of Theorem 3

The proof follows by arguing that the additional approximation error of $\delta(|\hat{p}| + 1)$ in the solution to Eq. (2) does not change the decisions taken by the algorithm. We shall argue by induction over the iteration of the algorithm for $i = 1, \ldots, M$.

Initially, for $i = 1$, trivially the algorithm sets $\hat{p}_1 = b_1$ and as argued in the proof of Theorem 2, we have $|\hat{p}_1 - p_1| \leq \delta$. Inductively, suppose we have discovered $n \geq 1$ prices by processing up to $i \geq 1$ transaction totals such that the resulting estimate has $\|\hat{p} - p\|_\infty \leq \delta$.

Consider the $i+1$st transaction total, b_{i+1}. There are two possibilities: with $|\eta_{i+1}| \leq \delta$, either (a) $b_{i+1} = p_{n+1} + \eta_{i+1}$, or (b) $b_{i+1} = \sum_{j \in S_{i+1}} p_j + \eta_{i+1}$ for some $S_{i+1} \subset [n]$.

In case (a), for any subset $S \subset [n]$,

$$\begin{aligned} |\hat{p}(S) - b_{i+1}| &= |\hat{p}(S) - p_{n+1} - \eta_{i+1}| \\ &\geq |\hat{p}(S) - p(S) + p(S) - p_{n+1}| - |\eta_{i+1}| \\ &\geq |p(S) - p_{n+1}| - |\hat{p}(S) - p(S)| - \delta \\ &\overset{(i)}{\geq} 4\delta N - |S|\delta - \delta \\ &= (4N + 4 - 1 - |S|)\delta, \end{aligned} \tag{7}$$

where (i) follows from the condition of Theorem 3. That is, the approximate subset-sum, with additive error of at most $\delta(|\hat{p}| + 1) \leq N\delta$ will declare any such S to have $|\hat{p}(S) - b_{i+1}| \geq (4N + 3 - |S|)\delta - N\delta$ which is at least $2N\delta$ since $|S| + 1 \leq n + 1 \leq N$. That is, the Algorithm in Section 2.1, even with approximate subset-sum makes the correct decision in such scenarios in terms of identifying the new element and setting $\hat{p}_{n+1} = b_{i+1}$.

In case (b), by definition $|b_{i+1} - \hat{p}(S_{i+1})| \leq \delta(1 + |S_{i+1}|)$. Therefore, the approximate subset-sum will find that there exists at least some subset $S \subset [n]$ so that $|b_{i+1} - \hat{p}(S)| \leq \delta(1+|S|)\delta + \delta(|\hat{p}|+1) \leq (2N + 2)\delta$. We want to argue that $S = S_{i+1}$. Using an argument identical to Eq. (7), we conclude that $|\hat{p}(S) - b_{i+1}| > (4N + 3 - |S|)\delta$. That is, the approximate subset-sum will declare such an S to have $|\hat{p}(S) - b_{i+1}| > (4N + 3 - |S|)\delta - \delta(|\hat{p}| + 1) \geq (2N + 2)\delta$. Thus, in case (b), the Algorithm in Section 2.1 with approximate subset-sum will identify S_{i+1} as the correct 'decomposition' for b_{i+1} as desired.

This completes the induction step and proof of the Theorem 3.

4 EXPERIMENTS

4.1 Dataset

We utilize a dataset of consumer transactions provided by alternative data vendor Second Measure [22]. The data consists of roughly 13.4 million anonymized consumer debit and credit card transactions in the period from January 2016 through February 2019 (see Table 2) for four companies: Netflix, Spotify, Chipotle, and Apple. Each data point contains only (i) the transaction total (ii) the company name (iii) daily-resolution timestamp (iv) city in which the purchase was made. Table 2 suggests that the number of transactions is large compared to the number of products offered at each company: $M \gg \|p\|_0$. We discuss the results on a per-company basis.

4.2 Pre-processing

We apply a simple pre-processing step to remove anomalies from the transaction data. Specifically, after sorting all transactions for a given company, we remove a small percentile of the top and bottom transactions as a robustness step. For example, with Netflix we see transaction totals ranging from $.01 to $182.5; however, more than 94% of transactions lie within an $11 range. And we retain this 'majority' range.

4.3 Netflix: Inferring Products and Prices

Data. The transactions data for Netflix spans the 38-month period from January 2016 through February 2019 for Boston, Chicago, Los Angeles, New York, Philadelphia, Phoenix, San Francisco, and Washington D.C. The data consists of 2.6 million separate transactions, translating to $29.5

million in observed sales. Amongst them, there are 3094 distinct bill totals ranging from single-penny transactions to $182.5. After the pre-processing described above, we are left with 1046 unique totals.

Task. We wish to recover product prices, the number of products, and the transaction decompositions. We emphasize that the number of products N is treated as unknown; and knowledge of N is used only in evaluating our method's performance. We utilize the algorithm with $k = 1$ and δ as 1% of the transaction total.

Findings. Table 4 shows the inferred product prices. Specifically, we find 11 candidate products and corresponding prices. Next, by hand, we match the first 9 of 11 inferred prices to actual Netflix product prices. The results are shown in Table 4. To our knowledge, this is complete coverage of Netflix's advertised product prices in that period. The median error in inferred prices is less than 0.2% (or less than 4 cents!). Since product prices have changed over time, several of the products appear at multiple price points.

For almost all inferred products we can associate clear, low-error matches in Netflix's product catalogue. However, we also find two additional products (and associated prices) that do not correspond to publicly disclosed Netflix products (at least not in the time period of the data). The inferred product with price $17.08 matches the rumored Ultra HD product of Netflix available at $16.99 [23]. It seems that there might be another such product being offered at a higher price around $18.45.

In summary, we recover the entire Netflix product price catalog accurately - from just transaction totals. In addition, we recover hidden or unadvertised products - one rumored and another completely unknown. This establishes the efficacy of our algorithm.

Actual Price	Inferred Price	Abs Error	Product(s)
$7.99	$7.98	0.13%	SB; DS
$8.99	$8.98	0.11%	SB
$9.99	$9.97	0.20%	SS; RS
$10.99	$10.93	0.55%	SS
$11.99	$11.97	0.17%	SP; DP
$12.99	$12.95	0.31%	SS
$13.99	$14.00	0.07%	SP
$14.99	$14.96	0.20%	RP
$15.99	$15.99	0%	SP
$16.99	$17.08	0.53%	Rumored new product
-	$18.45	-	-

Table 4. Products and their prices found by our algorithm from transaction totals for Netflix. We hand match inferred prices to Netflix's published catalog / prices from [11, 19, 21, 23]. The key products are streaming basic (SB), standard (SS), and premium (SP); DS and DP refer to DVD Standard (DS) and Premier (DP) subscriptions; RS and RP refer to Netflix's HD Blu-Ray Standard (RS) and Premier (RP) subscriptions. In addition, we find two unmatched products: one at $17.08 which might be the rumored Ultra HD [23], another at $18.45 is likely to be another such unknown product.

4.4 Spotify: Product Launch Detection

Data. The transaction data for Spotify spans the period from September 2017 through September 2018. As shown in Table 2, the data includes 387,000 transactions, of which 527 are unique totals. This corresponds to $4.1M in observed revenue. The transactions range from 24 cents to $250. After pre-processing, we are left with transactions in the range from $4.98 to $16.19.

Task. Given the success of the algorithm in identifying Netflix's price catalogue, we next look to utilize our method in detecting product launches. By repeatedly running our inference method over short, rolling time windows, we hope to detect product launches and the associated prices by examining changes in the inferred prices over time. In particular, on April 11th, 2018, a new product was launched by Spotify at a price of $12.99 [20]. Again, we emphasize that the number of products N is always treated as unknown. Knowledge of N is used only in evaluating our method's performance.

Findings. We run our algorithm on transactions associated with three-month sliding windows. Table 5 details the results. We see a clear shift in inferred prices following the release of the $12.99 product, during the month of launch. Indeed, the new price is detected with small error (<8%) as expected due to noise like variations in taxes, discounts, etc.

Window	Top-Two Inferred	Product Matches	Error	New Product
Sept'17-Nov'17	$10.64, $16.0	$9.99, $14.99	6.5%, 6.7%	No
Oct'17-Dec'17	$10.73, $16.0	$9.99, $14.99	7.4%, 6.7%	No
Nov'17-Jan'18	$10.73, $16.0	$9.99, $14.99	7.4%, 6.7%	No
Dec'17-Feb'18	$10.61, $16.0	$9.99, $14.99	6.2%, 6.7%	No
Jan'18-Mar'18	$10.65, $15.89	$9.99, $14.99	6.6%, 6%	No
Feb'18-Apr'18	$13.99, $15.89	$12.99, $14.99	7.7%, 6%	Yes
Mar'18-May'18	$14.03, $15.93	$12.99, $14.99	8.0%, 6.3%	Yes
Apr'18-Jun'18	$14.11, $15.93	$12.99, $14.99	8.6%, 6.3%	Yes
May'18-July'18	$13.61, $15.93	$12.99, $14.99	4.8%, 6.3%	Yes
June'18-Aug'18	$13.93, $15.93	$12.99, $14.99	7.2%, 6.3%	Yes
July'18-Sept'18	$13.97, $15.93	$12.99, $14.99	7.5%, 6.3%	Yes

Table 5. The top two (most expensive) product prices inferred from transaction totals for the corresponding three-month window are reported. Prices are as published by Spotify during the corresponding period. Our method clearly detects the newly launched product on April 11th, 2018 priced at $12.99 [20].

4.5 Apple: iPhone Launch and Sales

Data. We examine data for Apple sales in New York, NY for the two month period from August 21st, 2018 through October 21st, 2018. The data covers the month before and after the release of new iPhone products on September 21st, 2018. It consists of 197,000 transactions, with 7,685 unique transaction totals and corresponds to $6.2M in observed revenue. The transaction totals range from $.01 to $18,000, and after pre-processing, we are left with transaction totals in the range of $100-$2083.87 with 2364 unique transaction totals.

Task. Similar to the work with detecting product launches at Spotify, we wish to detect the launch of new iPhones in September 2018. The launch included: (i) release of a new iPhone XS at $999, and (ii) the release of the iPhone XS Max priced at $1449. Pre-launch, Apple offered products priced

	Pre-Launch	Post-Launch
Actual Product Range	$974 ± $25	$999
Inferred Price	$948	$959
Percent Error	< 5.4%	4%
Estimated Sales Increase	-na-	114.5%

Table 6. Inferring product prices and associated sales volume before and after the iPhone XS product launch on September, 21st 2018. Pre-launch, a collection of products were in the range $974 ± $25 which matches an inferred price of $948 with error < 5.4%. Post-launch, the price of the new (launched) iPhone XS is $999 which then matches an inferred price of $959 with error < 4%. Using these inferred prices, we detect an increase in sales volume for products priced around $999 by 114.5% indicating the impact of the launch of iPhone XS.

around $999. Therefore, we would expect to see an increase in estimated sales volume around that $999 price post-launch. On the other hand, no iPhone products were priced around $1449. Hence, we expect to also detect a new product price.

Findings. As described in Table 6, we estimate a 114.5% increase in inferred sales for products priced around $999. As noted in Table 7, a new product priced within 1% (error) of the iPhone XS Max appears in the inferred product prices. Thus, our algorithm, manages to detect product launches even within Apple's extremely complex product catalog.

	Pre-Launch	Post-Launch
Max Inferred Price	$1208	$1434
Closest Product Price	$1149	$1449
Error	5.1%	1.0%

Table 7. Preceding the launch, there are no iPhone products priced near $1449. Following the launch, the top-inferred price changes to $1434 which is within 1% error of the price of the launched iPhone XS Max.

4.6 Chipotle: Revenue attribution and reconstruction error

Data. The transactions data for Chipotle in New York, NY covers the 6-month period from September 2018 through February 2019. The data consists of 133, 000 transactions, with 2800 unique transaction totals and a corresponding $1.7M in revenue. The transaction totals range from $.03 to $552.54. After pre-processing, we retain transactions in the range of $1.5-$20 and 1230 unique transaction totals.

Task. Chipotle's menu is too complex to analyze fully: even for bill totals under $20, there are a large number of possible order combinations; and in addition, the menu contains multiple items at the same price point. Nevertheless, we hope to gain insight into Chipotle sales with two goals. First, we examine how well we are able to model the thousands of unique bill totals assuming just a small number of product price-points. Second, we would like to determine if a relatively small price range for products accounts for a large percentage of Chipotle revenue. We use $k = 2$ and δ of 0.1% of each transaction total.

Findings. The inference algorithm identifies 12 price points within the Chipotle menu, from $1.5 to $18.65. Using these 12 products and associated prices, we examine how well each of the 1230

distinct bill totals can be decomposed into combinations of these prices. We find that with mean-absolute-percent error (MAPE) of 1.2%, all the transaction totals can be decomposed using just these 12 products / prices.

Next, we decompose each transaction total into its inferred, constituent products. Given the inferred decompositions, we examine the concentration of sales (in USD) by product price. Table 3 shows that a large majority of estimated sales concentrate in a narrow price range of $9-$11: suggesting that a large fraction of Chipotle's customers order items like the "chicken burrito" at $9.74 [7].

To benchmark the performance of our algorithm, we compare it with a simple, randomized algorithm. We consider the approach of randomly selecting X transaction totals as individual item prices; and then using these selected prices, we attempt to decompose all transaction totals into sums of these prices (with the previously mentioned constraints). We repeat this random selection-then-decomposition process 30 times and average the results. The resulting MAPE using the randomized algorithm is 7.5%; and if we weight the results by frequency, we see a weighted MAPE of 22.7%. That is, this randomized benchmark has a MAPE of 5-6x worse than our approach. To understand if this is statistically significant, we construct a hypothesis test: after each randomized trial, we compare the per-trial MAPE to our method. Let ψ be the probability that our method outperforms the randomized benchmark on a given trial. The null hypothesis is $\psi \leq 0.5$. The alternative hypothesis is $\psi > 0.5$. We will reject the null hypothesis if we observe a p-value of less than 10^{-2}. On 30/30 trials (with both unweighted and weighted errors) our method outperforms the randomized approach. We obtain a p-value of $(\frac{1}{2})^{30} < 10^{-9}$. Thus we reject the null-hypothesis with overwhelming statistical confidence. Similarly, the hypothesis test results are identical if we use a train-test split of 4 months/2 months: (i) first infer a price menu from the train set (ii) approximate totals in the test set and then (iii) run the randomized comparison on the test set - which, in fact, provides advantage to the randomized method.

5 CONCLUSION

In this work we perform seemingly counterintuitive inference: given an anonymous consumer's transaction total (a single number), we estimate the number of products purchased along with the associated prices. This boiled down to solving a system of linear equations with observing only an aggregates vectors on one-side of the equation, and nothing on the other side. Specifically, we estimate the unknown number of products N, their associated prices $p \in \mathbb{R}^N_{>0}$ and the transaction decomposition given by matrix $A \in \mathbb{Z}^{M\times N}_{\geq 0}$ all from observing only transaction totals $y \in \mathbb{R}^M_{>0}$ such that $y = Ap + \eta$ where η represents the noise (e.g. taxes).

As the main contribution of this work we provide an iterative algorithm to recover N, p, and A with $M \geq N$. We prove correctness of our algorithm under mild conditions. We apply our algorithm to anonymized credit card transactions associated with consumer spending at Apple, Chipotle, Netflix, and Spotify in the years 2016-2019. Using transactions data only, our algorithm recovers the number of products and product prices accurately, decomposes transaction totals with small error (MAPE < 2%), identifies product launches, and discovers a rumored 'secret' product of Netflix in the market.

The success of our method suggests a future direction in understanding the limits of anonymization. For example, under what conditions are consumer product choices irrecoverably obfuscated? Retailers wishing to obfuscate units sales, limited trials, and other financial details, might consider how simple bill totals shed light on their financials.

Acknowledgments

We would like to thank alternative data company Second Measure for providing the anonymized debit and credit card transactions used here. Second Measure's datasets were wonderful to use, and we appreciate the time they spent preparing the data and talking with us. This work was supported in parts by NSF projects CMMI-1462158, CMMI-1634259, CNS-1523546, TRIPODS Phase 1; a joint project with KAIST (South Korea); and a project funded by KACST.

REFERENCES

[1] AlternativeData.org. Alternativedata.org database point of sale data. https://alternativedata.org/data-providers/category,point-of-sale. Accessed: 2019-05-19.
[2] Mark Bergen and Jennifer Surane. Google and mastercard cut a secret ad deal to track retail sales. https://www.bloomberg.com/news/articles/2018-08-30/google-and-mastercard-cut-a-secret-ad-deal-to-track-retail-sales, August 2018. Accessed: 2019-05-19.
[3] Radu Berinde, Anna C Gilbert, Piotr Indyk, Howard Karloff, and Martin J Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 798–805. IEEE, 2008.
[4] Florentin Butaru, QingQing Chen, Brian Clark, Sanmay Das, Andrew W Lo, and Akhtar Siddique. Risk and risk management in the credit card industry. Working Paper 21305, National Bureau of Economic Research, June 2015.
[5] Emmanuel Candes and Terence Tao. Near optimal signal recovery from random projections: Universal encoding strategies. *arXiv preprint math/0410542*, 2004.
[6] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.
[7] Chipotle. Chipotle online ordering. https://order.chipotle.com/Meal/Index/1597?showloc=1, 2019. Accessed: 2019-05-01.
[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.
[9] Ryan Dezember. Your smartphone's location data is worth big money to wall street. https://www.wsj.com/articles/your-smartphones-location-data-is-worth-big-money-to-wall-street-1541131260, November 2018. Accessed: 2018-11-04.
[10] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.
[11] DVD.COM. Dvd.com choose a plan. https://dvd.netflix.com/Plans?dsrc=DVDWEB_NMHOME_NMHEADER_PLANS. Accessed: 2019-05-27.
[12] Amir Efrati. U.S. slowdown at Uber and Lyft. https://www.theinformation.com/articles/u-s-slowdown-at-uber-and-lyft, September 2018. Accessed: 2018-10-25.
[13] Michael Fleder and Devavrat Shah. Forecasting with alternative data. In *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '20, page 23–24, New York, NY, USA, 2020. Association for Computing Machinery.
[14] Bradley Hope. Provider of personal finance tools tracks bank cards sells data to investors. https://www.wsj.com/articles/provider-of-personal-finance-tools-tracks-bank-cards-sells-data-to-investors-1438914620, April 2015. Accessed: 2018-05-10.
[15] IO&C. The big trends in data reshaping financial industry. https://ioandc.com/the-big-trends-in-data-reshaping-financial-industry, April 2019. Accessed: 2019-04-07.
[16] Jon Kleinberg and Eva Tardos. *Algorithm design.* Pearson Education India, 2006.
[17] S.P Kothari. Capital markets research in accounting. *Journal of Accounting and Economics*, 31(1):105 – 231, 2001.
[18] Tze Leung Lai, Ching Zong Wei, et al. Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics*, 10(1):154–166, 1982.
[19] Netflix. Netflix pick your price. https://www.netflix.com. Accessed: 2019-05-27.
[20] Sarah Perez. Spotify and Hulu launch a discounted entertainment bundle for $12.99. https://techcrunch.com/2018/04/11/spotify-and-hulu-launch-a-discounted-entertainment-bundle-for-12-99-per-month, April 2018. Accessed: 2019-06-11.
[21] Ashley Rodriguez. A history of netflix us price hikes, charted. https://qz.com/1524449/netflix-just-raised-prices-in-the-us-a-history-of-hikes-charted. Accessed: 2019-05-27.
[22] Second Measure. Data points. https://secondmeasure.com/datapoints. Accessed: 2019-05-19.
[23] Todd Spangler. Netflix testing out pricier new "Ultra" plan at $16.99 per month. https://variety.com/2018/digital/news/netflix-ultra-plan-hdr-ultrahd-test-1202865305, July 2018. Accessed: 2019-05-27.

[24] Robin Wigglesworth. Asset management's fight for alternative data analysts heats up. https://www.ft.com/content/2f454550-02c8-11e8-9650-9c0ad2d7c5b5, January 2018. Accessed: 2018-05-07.

Received August 2020; revised September 2020; accepted October 2020