Detecting Routines: Applications to Ridesharing CRM

Ryan Dew,[1*] Eva Ascarza,[2] Oded Netzer,[3] Nachum Sicherman[3]

[1] The Wharton School, University of Pennsylvania
[2] Harvard Business School, Harvard University
[3] Columbia Business School, Columbia University

June 28, 2023

Abstract

Routines shape many aspects of day-to-day consumption. While prior work has established the importance of habits in consumer behavior, little work has been done to understand the implications of routines — which we define as repeated behaviors with recurring, temporal structures — for customer management. One reason for this dearth is the difficulty of measuring routines from transaction data, particularly when routines vary substantially across customers. We propose a new approach for doing so, which we apply in the context of ridesharing. We model customer-level routines with Bayesian nonparametric Gaussian processes (GPs), leveraging a novel kernel that allows for flexible yet precise estimation of routines. These GPs are nested in inhomogeneous Poisson processes of usage, allowing us to estimate customers' routines, and decompose their usage into routine and non-routine parts. We show the value of detecting routines for customer relationship management (CRM) in the context of ridesharing, where we find that routines are associated with higher future usage and activity rates, and more resilience to service failures. Moreover, we show how these outcomes vary by the types of routines customers have, and by whether trips are part of the customer's routine, suggesting a role for routines in segmentation and targeting.

Keywords: routines, customer management, customer relationship management, Bayesian nonparametrics, Gaussian processes, machine learning, ridesharing

*Ryan Dew (ryandew@wharton.upenn.edu) is the corresponding author. He is an Assistant Professor of Marketing at the Wharton School, University of Pennsylvania. Eva Ascarza is the Jakurski Family Associate Professor of Business Administration at Harvard Business School. Oded Netzer is the Arthur J. Samberg Professor of Business at Columbia Business School. Nachum Sicherman is the Carson Family Professor of Business, Columbia Business School. The authors thank Via for providing the data, and Saar Golde and Coleman Humphrey in particular for their help and input. The authors thank Soham Mahadik, Jon Wu, Zhangyi (David) Fan, Xuanming (Donny) Gu, and Nikhil Kona for excellent research assistance. Finally, the authors thank seminar participants at Northwestern, ETH Zurich, Chicago, UD, UT Austin, CU Boulder, CityU, Temple, and the CBS "Quant Lab", as well as participants at the Frankfurt AI in Business Workshop, AI/ML 2021, TPM 2021, and Marketing Science 2020 conferences for valuable feedback.

1. Introduction

Routines are an integral feature of consumers' daily lives: for many people, from the time they wake up to the moment they go to sleep, their time is structured around routines. Such routines often involve consumption, like picking up coffee from a favorite coffee chain each morning, checking a weather app before leaving home in the morning and before leaving work in evening, or choosing a mode of transportation to get to and from work. Moreover, consumers are often different in their routines: while one may drink their coffee only in the mornings, seven days per week, another may prefer to have their coffee after lunch, and only on weekdays. Marketers can greatly benefit from understanding consumer routines. Yet, while routines are intuitively important drivers of consumer behavior, prior research has not explored the presence of such routines in consumers' behavior and their implications for customer management. Accordingly, the objectives of this research are first, to build a statistical model that can capture customer routines at the individual-level, and second, to explore the relationship between such routines and behavioral outcomes like transaction frequency and customer activity.

We define a routine as a *behavior with a defined, recurring, temporal structure, such that the same behavior occurs at roughly the same time, period after period*. We focus specifically on the period of a week, as weekly routines capture many common routines, including, for instance, weekday commutes, weekday lunches, weekend brunches, and weekly grocery shopping.[1] Routines are related to habits, which have been studied more extensively in marketing (e.g. Drolet and Wood 2017). It is the emphasis on temporal structure that differentiates routine behavior from habitual (or repeat) behavior. For example, a consumer who always shops at the same store may do so out of habit. A customer who always shops at that same store every Thursday evening exhibits a routine. In this sense, a routine can be viewed as a habit that is embedded in a consumer's day-to-day schedule. We posit that such temporally structured behavior may be an especially important predictor of customer value, and customer behavior more generally.

Little research has been done on capturing routines and understanding their impact on consumer behavior and firm profitability. In this work, we focus on the implications of routines for customer relationship management (CRM). In particular, we hypothesize that, given equal transaction rates, customers who interact with the company as part of their routine may be higher

[1] While weekly routines capture much of the richness of recurring consumption, there are also routines that exist over longer periods, like getting a haircut the first Friday of a month, or biweekly Sunday dinners at one's parents' house, which will not be captured by focusing on weekly routines. Our approach could be easily extended to cyclicalities other than a week, as we describe later.

value customers, in terms of more future purchasing and lower rates of churn, relative to customers with no routine. Routine customers may also be better customers in other ways, including having lower price sensitivities, and higher resilience to service disruptions. We hypothesize that the effect of a routine exists over and above a mere tendency to repurchase the same product, as is already captured in many existing customer relationship management (CRM) frameworks like the recency-frequency-monetary value (RFM) model (e.g., Blattberg, Neslin, and Kim 2008; Neslin, Taylor, Grantham, and McNeil 2013). In other words, given two customers with identical purchasing summary statistics — that is, they both purchased recently, they both made the same number of purchases historically, and they both spent the same amount on each purchase occasion — we predict that the customer whose purchase timings exhibited a higher *routineness* will be a better customer in the future.

To measure customer routineness, we develop a statistical model that allows us to identify the routine of each individual customer, if it exists, and isolate the share of the consumption that can be attributed to that routine. This model enables us to differentiate between, for instance, a customer who has ten routine trips per week (e.g., to and from work every weekday) and another customer who has only two routine trips per week (e.g., going to Yoga class on Tuesday and Thursday afternoons). Specifically, our model is an individual-level, inhomogeneous Poisson process that captures individual-specific patterns in consumption across periods, with a unique Bayesian nonparametric specification of its rate. The individual-specific rate of consumption is decomposed into a component that captures potentially dynamic levels of idiosyncratic or "random" consumption, and a "routine" component that captures individual-level consistencies in consumption timing, which is modeled using a Gaussian process prior with a unique kernel structure. This kernel structure incorporates intuitive aspects of consumption over time — specifically, that certain days exhibit similarities in consumption (e.g., a Tuesday might be more similar to a Thursday than to a Sunday) and that consumption within days exhibits a 24-hour cycle (e.g., 12:05am is similar to 11:55pm) — to precisely estimate individual-specific variation in routine behavior. Using the routine component of the Poisson rate parameter, we construct an individual-specific "routineness" metric that measures to what degree an individual's behavior is structured around a routine. In addition to the routineness metric, the model infers the form or temporal "shape" of the routine for each consumer (e.g., whether a consumer has a Monday through Thursday AM routine, or a Tuesdays evening routine).

We apply our model and routineness metric to data from Via, a leading New York City-based ridesharing company, to estimate consumer routines in requesting rides. Ridesharing is a particularly rich setting for studying routines, as travel is often an integral part of many daily and weekly routines. We identify various patterns in using the ridesharing service across users, including predictable commuting routines, as well as more complex, idiosyncratic routines. More importantly, we show that, as hypothesized, users who are more routine in their behavior are also more valuable to the firm, in terms of both higher future usage and higher rates of remaining active, even after controlling for past usage patterns such as recency, frequency, or clumpiness. Having established the value of routineness in customer value, we then show that routines also play a role in driving and moderating other aspects of the customer-firm relationship, including price sensitivity and customer response to service failures. Our results suggest that firms can benefit from understanding routines, both for better predicting future customer activity, and for developing segmentation and targeting strategies. Notably, routines moderate customer behavior both between customers, insofar as more routine customers behave differently than less routine customers, and within customers, insofar as customers react differently to pricing and service for trips that are part of their routines. The between-customer effects suggest that the firm may benefit from identifying routine users, or even cultivating routines, while the within-customer effects suggest that there is room for firms to optimize the provision of services around customers routines.

The rest of the paper is organized as follows: we start by discussing the prior literature on habits and routines, and the connections between routineness and other extant metrics of transaction timing in CRM. We then present our model for capturing and measuring customer-specific routines. Moving next to our empirical application, we describe the ridesharing data and the results of applying our model: we first apply the model on synthetic data that mimics the real data, validating the model's ability to recover different types of routines. We then apply the model on the ridesharing data, characterizing the types of routines exhibited by riders, and validating the model's fit. Finally, we explore the idea of routineness more deeply, by highlighting the relevance of routines for CRM, exploring how customer-level outcomes vary by the type of routine, and by comparing routineness to other constructs. We conclude with discussion and directions for future research.

2. Conceptual Foundations

While research on routines is relatively scant, the closely related topics of habits and repeat behaviors have been studied extensively, both in marketing and in related disciplines. Early work in marketing used the term *repeat buying habit* to simply indicate repeatedly buying the same product or repeatedly buying from the same company, without considering the more psychological construct of a habit or habit formation (Ehrenberg and Goodhardt 1968). Predicting repeat purchasing has subsequently been the focus of many models in customer base analysis, including popular buy-till-you-die models (e.g., Schmittlein, Morrison, and Colombo 1987) and more general RFM-based specifications (e.g., Dew and Ansari 2018). Repeat buying is also central to other important marketing constructs, including brand and store loyalty and brand inertia (e.g., Guadagni and Little 1983), all of which can also be viewed as forms of habitual behavior. Moving beyond studying simple repeat purchasing, Shah, Kumar, and Kim (2014) generalized the idea of habits to recurring behaviors like returning products, purchasing on promotion, and purchasing low-margin items. They showed that these repeat behaviors are linked to firm profitability, and that firm actions can influence the formation of habitual behaviors.

Habit formation has also been studied in economics, often in the context of consumption and expenditure, where it is typically defined as current expenditures depending on lagged expenditures through a "habit stock." In this literature, habits have been used to explain the smoothness of consumption over time, even in the presence of shocks to income, although evidence for the existence of habit formation in aggregate consumption is mixed (Dynan 2000; Fuhrer 2000).

Much of the theory behind why habits matter, how they develop, and how they can be changed has come from the psychology and consumer behavior literatures. Habits have been studied in psychology since as early as the 19th century (James 1890). In this literature, habits are often defined as tendencies to repeat behaviors, typically automatically or subconsciously (Ouellette and Wood 1998; Wood, Quinn, and Kashy 2002), and sometimes in a goal-directed manner (Aarts and Dijksterhuis 2000), or triggered by contextual cues (Neal, Wood, Labrecque, and Lally 2012). Especially relevant for our empirical application of ridesharing, habits have recently been identified as a primary driver of travel mode choice (e.g., Verplanken, Walker, Davis, and Jurasek 2008), which is of particular interest for developing more sustainable consumer choices (White, Habib, and Hardisty 2019). A noteworthy finding in this literature is the habit discontinuity hypothesis, which states that context changes that disrupt individuals' habits can lead to deliberate choice consideration, and habit breaking (Verplanken et al. 2008). This phenomenon has also been observed in the CRM literature. For example, Ascarza, Iyengar, and Schleicher (2016), show that customers who continue to transact with the firm out of habit may be driven to churn by company retention efforts, even when those retention efforts are intended to save the customer money, simply by means of disrupting their inertia.

2.1. Routines and Habits

In one sense, routines can be viewed as a specific type of habit, where the automaticity of behavior is related to time: if, every day, at a certain time, a consumer carries out an action, then time can be considered the context that triggers that behavior. Thus, many of the predictions made elsewhere in the literature about habitual behavior and customer loyalty (e.g., Ascarza, Neslin, Netzer, Anderson, Fader, Gupta, Hardie, Lemmens, Libai, Neal et al. 2018) carry over to routines: we postulate, for instance, that routines can lead to nearly automatic choices, and will thus be more difficult to break, resulting in stickier long-run behavior, and lower likelihood to react negatively to price increases or service failures. However, we hypothesize that routines are more predictive of customer value than mere habit. The key distinction between habits and routines is that, whereas habits simply imply automatic, repeated behaviors, a behavior is routine only if it additionally has a *recurring, temporal structure*. Intuitively, such behaviors are likely embedded in a consumer's daily life, and thus, may be even more automatic, and indicative of long-run value, than habitual behaviors that lack such a temporal structure. Thus, a customer who is *routinely* consuming a focal product or service may be even more valuable than one who is merely habitually (i.e., repeatedly) consuming the product, but not in a routine manner.

2.2. Clumpiness, Regularity, and Routines

Our work is also related to the growing literature on extending traditional RFM frameworks to incorporate information about usage and purchase timing. RFM-based frameworks, while useful predictive tools, discard much of the richness of a customer's transaction history, and simply summarize a customer's likelihood of repeat purchasing by how recently they purchased, and how often they purchased in the past. Recent work has shown that there is incremental value in moving beyond these simple statistics. A notable contribution in this stream is Zhang, Bradlow, and Small (2015), who defined the *clumpiness* of customer transaction times. Their metric cap-

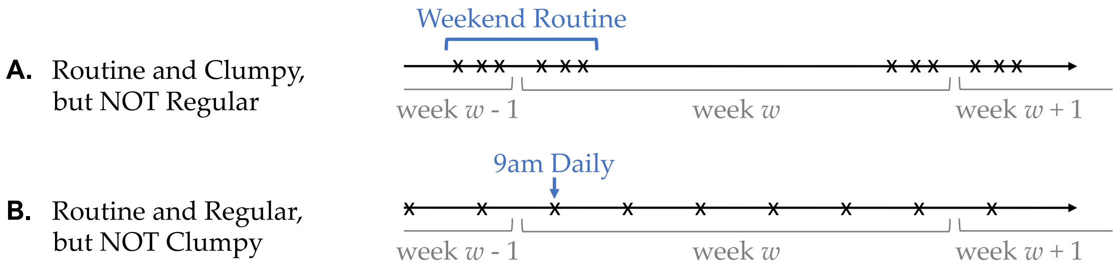


Figure 1: Clumpy and Regular Routines
Illustration of the conceptual links between routineness, regularity, and clumpiness.

tures the customer-level entropy of intervisit times and is higher when a customer exhibits more temporally clustered behavior, or "clumpy," behavior. They show that, in many empirical contexts, especially contexts with bingeable content, this measure of clumpiness is a key (positive) predictor of customer lifetime value. Another key contribution comes from Platzer and Reutterer (2016), who defined the concept of transaction *regularity*. In particular, they introduced a buy-till-you-die model, the Pareto-GGG, where the regularity of transactions is modeled by relaxing the standard exponential-distributed intertransaction time model common to many customer base models, allowing for customer-specific gamma-distributed intertransaction times. They find that incorporating regularity can improve customer-level predictions.

Routineness is conceptually distinct, but related to, these two metrics. In particular, routines can generate clumpy or regular behavior, or even regular clumpy behavior, depending on the type of routine. For example, a customer who takes multiple rides club-hopping every Saturday night exhibits a routine that is clumpy, while a "workaholic" customer who takes a ride to work 7 days a week exhibits very regular behavior. We illustrate these ideas in Figure 1. On the other hand, a customer who commutes only on Mondays and Wednesdays may have a routine that is neither clumpy nor regular. In other words, the measures are distinct: not all routines are clumpy or regular, and not all clumpy or regular behaviors are routine.[2]

That routines can generate clumpy and regular behavior is a key advantage of our framework, for several reasons. First, routines add additional nuance to the possible "types" of clumpiness that can be observed, thereby partly answering a call from Kumar and Srinivasan (2015) to explain the origins of clumpiness in transactions. Relatedly, this link between routines, clumpiness, and regularity also sheds light on when clumpy and regular transaction times, which intuitively seem at odds, may *both* be predictive of higher customer value, insofar as both may be manifestations

[2]We also demonstrate these connections empirically in our "quasi-simulation" analysis, described later.

of routines. More broadly, the interpretability of routines makes routineness a valuable metric for marketers looking to build interpretable yet accurate CRM models, thus addressing an on-going need for new advances in this space (Neslin, Gupta, Kamakura, Lu, and Mason 2006). Finally, our novel approach of identifying and isolating routines using transaction data and relating them to the customer value is consistent with Du, Netzer, Schweidel, and Mitra (2021)'s call to move toward a richer characterization of behavior, and toward relating such behaviors to firm growth through customer value.

3. Model

In this section, we specify a model of usage that yields a natural metric for how routine a customer's behavior is, and what weekly routine the customer exhibits. By "usage," we mean the consumer interacting with the firm in some way, and by "weekly routine," we mean the structure of usage within a given week, which is the main focus of this research. Before describing the model, we first review its methodological underpinnings.

3.1. Methodological Background

The model we propose merges an inhomogeneous Poisson process with a Bayesian nonparametric Gaussian process. While the basis of many customer base analysis models is a homogeneous Poisson process (Schmittlein, Morrison, and Colombo 1987), inhomogeneous Poisson process transaction models have been employed to capture more complex dynamics in usage or transaction behavior (e.g., Ho, Park, and Zhou 2006; Ascarza and Hardie 2013). An inhomogeneous Poisson process is a point process over some space, $\mathcal{S}$, where the rate of observing events, $\lambda(s)$, depends on position in the space, $s \in \mathcal{S}$, such that the number of events in any bounded region of the space, $Y(\mathcal{B})$, $\mathcal{B} \subseteq \mathcal{S}$, is distributed:

$$Y(\mathcal{B}) \sim \text{Poisson}\left(\int_{\mathcal{B}} \lambda(s)\, ds\right).$$

In many marketing applications, the space $\mathcal{S}$ is time (i.e., $\mathcal{S} = \mathbb{R}$), such that the process simply captures that events are not uniformly likely over time. Poisson processes enjoy a number of useful properties, including the *superposition* property, which says, if Y_1 and Y_2 are Poisson processes with rates λ_1 and λ_2, then $Y_1 + Y_2$ is also a Poisson process with rate $\lambda_1 + \lambda_2$ (Kingman 1992). We make use of this property in our model specification to separate routine from non-routine transactions.

In our model, the time-varying rate parameter of the Poisson process is modeled using a Gaussian process, or GP (Williams and Rasmussen 2006). In marketing, GPs have seen increased use in recent years, in both aggregate-level and individual-level CRM and brand choice contexts (Dew and Ansari 2018; Dew, Ansari, and Li 2020; Tian and Feinberg 2021). Regarding our research objective, GPs offer an ideal solution to modeling routines because, unlike other flexible function estimation methods, they enable us to flexibly model customer-level rates of usage, while also allowing us to encode prior knowledge about the structure of time. We elaborate more on this point as we describe our model below. In the broader literature, our model aims to capture time-varying purchasing or usage patterns, and is thus related to a long line of dynamic models in marketing (e.g., Kim, Menzefricke, and Feinberg 2005; Du and Kamakura 2012).

3.2. Model Specification

We propose a model of customer usage of a focal product or service. The key dependent variable, denoted y_{it}, captures how many times customer i interacts with the company during time period t. In later sections, when we apply this model to ridesharing data, the dependent variable will be requesting rides. However, our model is fully general, and can be applied using timing data from any context and at various time intervals, and for myriad customer behaviors of interest (e.g., using a mobile app, making purchases with the firm, visiting the firm's website).

We model a customer's observed usage y_{it} as the amalgamation of two individual-level, inhomogeneous Poisson processes over time (t): a routine process $y_i^{\text{Routine}}(t)$, which captures how often consumption needs arrive as part of the customer's routine, and a non-routine or "random" process, $y_i^{\text{Random}}(t)$, which captures how often consumption needs arrive outside of the customer's routine. Throughout the paper, we will use the words "non-routine" and "random" interchangeably. The idea here is that usage outside of the routine is likely due to random needs arising, from the point of view of the customer, not that this process is totally random (i.e., white noise). Each of these processes has its own customer-specific rate, $\lambda_i^{\text{Routine}}(t)$ and $\lambda_i^{\text{Random}}(t)$. As the analyst, we do not observe whether a given transaction is routine or not; we only observe the collection of all transactions, $y_i(t) \equiv y_i^{\text{Routine}}(t) + y_i^{\text{Random}}(t)$. This additivity assumption implies that usage in a customer's routine can increase or decrease independently of usage outside the routine, and vice versa. By the superposition property described above, we have that $y_i(t)$ is also a Poisson process, with rate given by $\lambda_i^{\text{Routine}}(t) + \lambda_i^{\text{Random}}(t)$. Meaningfully decomposing overall usage into routine and non-routine parts thus requires specifying these rates.

To specify these rates, we first simplify our setting, by assuming the analyst only cares about time on a discrete grid, such that we only consider the number of uses that occur within fixed intervals. Slightly abusing notation, we use t to refer to this fixed time grid. Given our intuition that routines are customer-level behaviors that are consistent in terms of when they occur, week over week, the relevant grid to consider consists of weeks (w), days within weeks (d), and hours within days (h), the collection of which gives us $t = (w, d, h)$. We will assume w indexes weeks since the start of the data, d indexes days of the week starting with $d = 1 =$ Sunday, and $h = 0, \dots, 23$ indexes the 24 hours of a day. To simplify notation even further, we will use the unit of "day-hours," which we denote as $j = 1 + (d-1) \times 24 + h$, such that $j = 1, \dots, 168$, captures all the hours in a week. Under this time structure, the dependent variable $y_{it} = y_{iwj}$ captures the number of interactions customer i has with the firm in week w at day-hour j.

Under this discrete time assumption, our model likelihood can be specified as:

$$y_{it} \sim \text{Poisson}(\lambda_{it}^{\text{Routine}} + \lambda_{it}^{\text{Random}}), \tag{1}$$

where $\lambda_{it}^{\text{Routine}} = \int_{\tau \in t} \lambda_i^{\text{Routine}}(\tau) d\tau$, and likewise for $\lambda_{it}^{\text{Random}}$. We denote the overall usage rate as $\lambda_{it} \equiv \lambda_{it}^{\text{Routine}} + \lambda_{it}^{\text{Random}}$. We then break each of the overall usage rate terms into two dynamic parameters:

$$\lambda_{it} = \exp(\gamma_{iw} + \eta_{ij}) + \exp(\alpha_{iw} + \mu_j). \tag{2}$$

In each term, there is one parameter that varies over weeks (w), and one that varies over day-hours (j). Our substantive focus is primarily on variation over day-hours. Thus, noting that λ_{it} could be rewritten as $\lambda_{it} = \exp(\gamma_{iw}) \exp(\eta_{ij}) + \exp(\alpha_{iw}) \exp(\mu_j)$, we will refer to the terms with j subscripts (i.e., η_{ij} and μ_j) as *day-hour rate* terms, reflecting the rate of usage across day-hours, and refer to the terms with w subscripts (i.e., γ_{iw} and α_{iw}) as *weekly scaling* terms, as these terms scale up or down each of the rate terms over customers and weeks.[3]

We now explain how this structure maps onto the idea of routines as week-over-week consistencies in time of use. In the first term, the routine rate, $\lambda_{it}^{\text{Routine}} = \exp(\gamma_{iw} + \eta_{ij})$, the day-hour rate parameter, η_{ij}, has an i subscript. Thus, η_{ij} captures patterns of day-hour usage that are specific to customer i. If the weekly scaling parameter, γ_{iw}, is large for many subsequent weeks, it suggests that the customer is expected to use the service consistently at the same day-hours each

[3]Note that, even in the case that events occur infrequently, such that y_{it} is effectively binary, the Poisson likelihood is both valid and useful: the Poisson approximates the Binomial, and enjoys computational benefits when paired with sparse data, as we describe subsequently.

week, following the pattern determined by η_{ij}. On the other hand, in the second term, the non-routine or "random" rate, $\lambda_{it}^{\text{Random}} = \exp(\alpha_{iw} + \mu_j)$, the day-hour rate parameter does *not* have an i subscript. Instead, this term captures general patterns of day-hour usage that are common across customers and across weeks. For example, in our empirical application, μ_j captures that, on average, customers tend to take rides during the day, but not in the middle of the night. Said differently, if any given user were to randomly have need of the service, μ_j captures when we might expect that random need to arise, and how the distribution of random needs may deviate from a uniform distribution over day-hours. If this term's weekly scaling parameter, α_{iw}, is high, it suggests that the customer is expected to make many requests that week, but that the day-hour pattern of those requests is not consistent with that customer's week-over-week patterns. It is this tension between consistently using the service at the same (individual-specific) day-hours, versus using the service in a way that is "random" (up to the typical usage patterns in the population), that implicitly defines what our model detects as a routine: if the usage follows a customer-specific day-hour pattern that is consistent over weeks, over and above the general consistency implied by the customer base as a whole, that usage will be captured by the first term, and is what our model defines as routine usage.

Recall that the superposition property implies that our model can be equivalently expressed as the sum of two count processes, $y_{it} = y_{it}^{\text{Routine}} + y_{it}^{\text{Random}}$, such that:

$$y_{it}^{\text{Routine}} \sim \text{Poisson}(\exp(\gamma_{iw} + \eta_{ij})), \tag{3}$$

$$y_{it}^{\text{Random}} \sim \text{Poisson}(\exp(\alpha_{iw} + \mu_j)). \tag{4}$$

This decomposition allows for a natural definition of the levels of random usage and routine usage, through the expectation of Poisson random variables. Specifically, we define two metrics, E_{iw}^{Routine} and E_{iw}^{Random}, which are the expected number of random and routine interactions, respectively, within a single week w, for customer i, such that:

$$E_{iw}^{\text{Routine}} = \sum_j \exp(\gamma_{iw} + \eta_{ij}), \tag{5}$$

$$E_{iw}^{\text{Random}} = \sum_j \exp(\alpha_{iw} + \mu_j). \tag{6}$$

In plain English, these two terms capture how often a user is expected to interact with the firm in a given week, decomposing the total number of interactions into the expected number of interactions happening at random, and the number of interactions stemming from the user's routine.

These metrics allow us to identify how routine customers' behaviors are, and are at the heart of the paper's focus and intended contribution. We call E_{iw}^{Routine} the *routineness* of customer i in week w, and will use this metric and terminology throughout our analysis.

3.3. Specifying the Components of the Usage Rates

Our model captures individual-level, time-varying usage through two count processes, each of which has a rate ($\lambda_{it}^{\text{Routine}}$ and $\lambda_{it}^{\text{Random}}$, respectively) that comprises two parts: scaling terms (α_{iw} and γ_{iw}) and day-hour rates (μ_j, and η_{ij}). To model these parameters, we first recast the problem as estimating latent *functions*, $\alpha_i(w)$, $\mu(j)$, $\gamma_i(w)$, and $\eta_i(j)$, respectively. This switch from subscript notation to functional notation is merely a conceptual pivot: by recasting the problem of estimating rates as a problem of estimating unknown functions, we can capture uncertainty over those rates using Gaussian processes. As we show below, this in turn allows us to encode prior knowledge and assumptions about these parts of the model in a natural way, beyond those that could be incorporated in other specifications (like, for example, state space models).[4]

GPs provide a way of specifying prior distributions over spaces of functions. With this prior, we can encode structural information about the functions in that space, like the smoothness and differentiability of the functions, or other a-priori knowledge about their shape. In this way, GPs allow for flexible estimation of functions, while optimally leveraging both information sharing and a-priori knowledge, to improve the efficiency of those estimates. A GP is a distribution over functions, $f(x) : \mathbb{R}^d \to \mathbb{R}$, defined by two other functions: a mean function, $m(x)$, which captures the a-priori expected function value at inputs x, and a kernel function $k(x, x')$, which captures a-priori how similar we expect the function values $f(x)$ and $f(x')$ to be, for two inputs x and x'. Modeling $f(x)$ using a GP is denoted $f(x) \sim GP(m(x), k(x, x'))$. For a finite, fixed set of inputs, $x = (x_1, \ldots, x_N)$, $f(x) \sim GP(m(x), k(x, x'))$ is equivalent to:

$$f(x_1, \ldots, x_N) \sim N(m(x_1, \ldots, x_N), K), \tag{7}$$

such that element (n_1, n_2) of K is given by $K_{n_1,n_2} = k(x_{n_1}, x_{n_2})$. Mathematically, the matrix K is the kernel $k(x, x')$ evaluated pairwise over all inputs, and is called the kernel matrix. Intuitively, a

[4] We note that there are many links between GPs and state space models, a discussion of which is beyond the scope of this paper (see, e.g., Loper, Blei, Cunningham, and Paninski 2021). There are also links between GP models and time series models. For example, spectral analysis in time series is closely linked to Bochner's theorem for kernel methods (like GPs), which establishes that every kernel function can be expressed equivalently as a spectral density. This connection has been explicitly used in many GP methods to derive new kernels (e.g. Wilson and Adams 2013).

GP specifies a multivariate Gaussian prior over the outputs corresponding to any combination of inputs, by means of its mean function and kernel. Thus, these two objects are the primary source of model specification in GP-based models. In practice, it is common to set the mean function $m(x)$ to be zero or a constant and let the dependencies between the outputs be solely captured by the kernel (Williams and Rasmussen 2006). The primary restriction in specifying the kernel is that the corresponding kernel matrix be positive definite. The mean function and kernel themselves are typically parameterized through an additional set of parameters referred to as hyperparameters.

Returning to our specification, we model:

$$\gamma_i(w) \sim GP(\gamma_{0i}, k_{\text{Exp}}(w, w'; \phi_\gamma)), \tag{8}$$

$$\alpha_i(w) \sim GP(\alpha_{0i}, k_{\text{Exp}}(w, w'; \phi_\alpha)), \tag{9}$$

$$\eta_i(j) \sim GP(0, k_{\text{DH}}(j, j'; \phi_\eta)), \tag{10}$$

$$\mu(j) \sim GP(0, k_{\text{DH}}(j, j'; \phi_\mu)). \tag{11}$$

Here, $k_{\text{Exp}}(\cdot, \cdot)$ is the exponential kernel, and $k_{\text{DH}}(\cdot, \cdot)$ is a novel-day hour kernel, both of which we define subsequently. In $\gamma_i(w)$ and $\alpha_i(w)$, we allow for individual-level constant mean functions, γ_{0i} and α_{0i}, capturing the mean level of routine and non-routine usage for each customer. The ϕs are kernel hyperparameters.

The kernel used for both of the weekly terms is the exponential kernel (e.g. Dew, Ansari, and Li 2020), given by the general form,

$$k_{\text{Exp}}(w, w'; \phi_p = \{\sigma_p, \rho_p\}) = \sigma_p^2 \exp\left[-\frac{|w - w'|}{2\rho_p}\right], \tag{12}$$

where the p subscript corresponds to the hyperparameters of parameter p. For this kernel, p is either γ or α.[5] The exponential kernel is a special form of the more general Matérn kernel, a popular kernel often used to model functions that may exhibit smooth fluctuations over time (Williams and Rasmussen (2006); Dew, Ansari, and Li (2020)).[6] The smoothness of the underlying process is captured by the kernel's lengthscale parameter, ρ. The higher this parameter, the more covariance is expected between function values, given fixed inputs. We illustrate the effect of the lengthscale parameter on smoothness of the function draws in Figure 2. In the context of routines,

[5]Note that the same kernel hyperparameters are used for all customers, for a given parameter. That is, there is no i subscript within p.

[6]Here, we mean smoothness in the lay sense of the word, as illustrated by Figure 2. We do not mean smoothness in the functional analysis sense of the word in terms of differentiability. For additional discussion of these two forms of smoothness, see Dew, Ansari, and Li (2020).

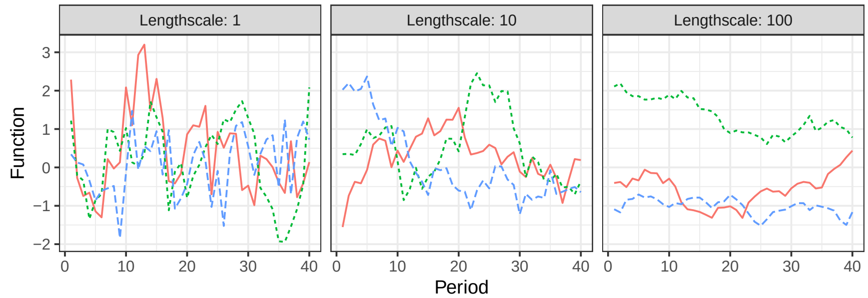


Figure 2: Draws from a GP prior.
The effect of the lengthscale parameter on draws from a GP with the exponential kernel. Each panel represents a different lengthscale (ρ). Each line is an independent draw from a GP prior with an exponential kernel with that lengthscale. In our application, our prior for ρ encourages relatively low smoothness for non-routine usage, and relatively high smoothness for routine usage, consistent with the idea of routines being somewhat (but not absolutely) sticky.

we expect that the routine weekly scaling parameter, $\gamma_i(w)$, will be smoother than the non-routine parameter, $\alpha_i(w)$. This additional smoothness mathematically corresponds to our intuition that a routine should be consistent week over week. We embed this a priori expectation by defining different priors for ρ_γ, the lengthscale of $\gamma_i(w)$, and ρ_α, the lengthscale of $\alpha_i(w)$, as we discuss in the next section.

The other kernel, $k_{\text{DH}}(j, j'; \phi)$, is a novel kernel which we term the *day-hour kernel*. This kernel has a functional form designed to capture the a priori structure we know exists within weeks, specifically that hours follow a 24-hour cycle, and that certain days are more similar to other days (e.g., weekends versus weekdays, or workdays versus days off). To capture these properties, we fuse a periodic kernel (Williams and Rasmussen 2006, Chapter 4) with an unstructured estimate of the correlation between different days of the week. Specifically, we define:

$$k_{\text{DH}}(j, j'; \phi_p = \{\sigma_p, \rho_p, \Omega_p\}) = \sigma_p^2 \times \Omega_p[d, d'] \times \exp\left\{\frac{1}{2\rho_p^2}\sin^2\left(\frac{\pi|h - h'|}{24}\right)\right\}, \tag{13}$$

where, again, p indexes a particular parameter, which in this case, can be either μ or η. The matrix Ω_p is a correlation matrix over days of the week, and the notation $\Omega_p[d, d']$ stands for the (d, d') entry of that matrix (i.e., the correlation between days d and d'). This correlation matrix can be seen as an unstructured kernel that allows the model to detect, in a fully flexible way, the correlation structure that exists between function values across different days. The third term in this product is the periodic kernel with a 24-hour cycle, denoted $k_{\text{Per}}(h, h'; \rho_p)$ for short. It captures the smooth but cyclic variation we expect to see over hours within a day, with smoothness ρ_p. Thus, our day-hour kernel is a specific form of the more general class of multiplicative kernels, formed by specifying kernels separately on input dimensions (in this case, days and hours), then multiplying those kernels together. Crucially, for our day-hour kernel to be valid, the kernel matrix formed by evaluating our kernel pairwise at all input values must always be positive (semi)definite. The kernel matrix (for parameter p) implied by our DH kernel is given by,

$$K_{\mathrm{DH}} = \sigma_p^2 \Omega_p \otimes K_{\mathrm{Per}_p}, \tag{14}$$

where, analogous to previous kernel matrices, K_{Per_p} is the matrix formed by evaluating the periodic kernel $k_{\mathrm{Per}}(h, h'; \rho_p)$ pairwise at all hours. To see that this matrix is positive definite, first, note that we restrict Ω to be a correlation matrix, and thus, Ω is positive definite. K_{Per} is guaranteed to be positive definite, since $k_{\mathrm{Per}}(h, h'; \rho_p)$ is known to be a valid kernel (Williams and Rasmussen 2006). Thus, since the Kronecker product of two positive definite matrices is also positive definite, we see that $k_{\mathrm{DH}}(j, j')$ is a valid kernel.

Intuitively, this day-hour kernel allows us to place a prior over functions that exhibit two natural properties when dealing with weekly usage data: we allow for arbitrary relatedness of days through the unstructured correlation matrix Ω_p, and for a natural 24-hour cycle through $k_{\mathrm{Per}}(h, h'; \rho_p)$, which accounts for the fact that usage at $h = 0$ (12 AM) will be similar to usage at $h = 23$ (11 PM). Finally, through its multiplicative structure, it assumes that these two forces operate together: if day d is similar to day d', as captured by Ω_p, and hour h is similar to hour h', a GP modeled with this kernel is expected to have similar function values at (d, h) and (d', h'). By encoding this natural prior information into our model structure, we facilitate the efficient inference of the mean and individual-level rate functions, $\mu(j)$ and $\eta_i(j)$.

3.4. Hyperparameter Priors

To complete our fully Bayesian specification, we now briefly describe the priors for the hyperparameters of our GP kernels, the most important of which are the length-scale parameters, ρ_α and ρ_γ. Recall that these hyperparameters control the smoothness of the weekly scaling parameters, and that we expect, a priori, that ρ_γ will be larger than ρ_α. To encode that in our model, we draw on the suggestions by Betancourt (2020) and use weakly informative inverse-gamma priors.[7] The inverse-gamma distribution, given by,

$$f(x \mid a, b) = \frac{\beta^a}{\Gamma(a)} x^{-a-1} \exp\left(\frac{-\beta}{x}\right),$$

[7]This prior has two desirable properties: first, it has support over the positive reals, and second, it "avoids" values that are close to zero. Too small lengthscale values can be problematic when the inputs to the function of interest are only observed on a coarse grid, as the smoothness between the observed inputs is unidentified.

has a shape parameter a and a scale parameter β. For both ρ_α and ρ_γ, we set the shape parameter $a = 5$, which implies a 1% quantile of around 1.[8] The value of b is set differently for ρ_α and ρ_γ and may depend on the empirical application. In short, b can be thought of as scaling the inverse-gamma distribution, effectively changing the magnitude of the draws. Thus, higher values will allow for higher lengthscales, corresponding to higher smoothness. In our application, we set $b = 5$ for ρ_α and $b = 11$ for ρ_γ.[9] This setting encourages higher values for ρ_γ, corresponding to higher smoothness of $\gamma_i(w)$ over weeks, which is exactly what our definition of routineness entails. Meanwhile, the relatively lower prior expectation for ρ_α does not enforce any a priori smoothness, which corresponds to allowing random needs to arise at any time.[10]

To estimate the correlation matrix Ω in the day-hour kernel, we use a Lewandowski-Kurowicka-Joe (i.e., LKJ) prior for correlation matrices, such that $\Omega \sim \text{LKJ}(2)$, which puts a weak prior toward the identity matrix (Lewandowski, Kurowicka, and Joe 2009).[11] Intuitively this prior is setting a (weak) expectation that different days will not be related to each other, essentially imposing no prior assumptions about how strongly days will be related to one another, or which days will be related to each other. This flexibility is a key advantage of this approach over a parametric kernel.[12]

Finally, for the individual-level constant mean functions, γ_{0i} and α_{0i}, we specify these in a hierarchical way, such that $\gamma_{i0} \sim N(0, \sigma_{\gamma_0})$, and analogously for α_{i0}. For all other parameters, we use standard, weakly informative normal or half-normal priors.[13]

[8] This value is important because our inputs are defined on a unit grid, so values lower than 1 will not be identified from one another.

[9] These exact values were arrived at following an optimization procedure, similar to that suggested by Betancourt (2020), where the objective was to find an inverse-gamma distribution with a 1% quantile of 1, and a 99% quantile of either 38/4 (for γ) or 38/8 (for α), where 38 is the range of our calibration data. A similar procedure can be used in other empirical settings.

[10] The model is quite robust to these values; for instance, the results are nearly identical if we set $b = 5$ for both. However, we think the $b = 11$ setting for routines is more consistent with our definition of consistent purchasing, week over week.

[11] More generally, for $\Omega \sim \text{LKJ}(\eta)$, the hyperparameter $\eta = 1$ puts a uniform prior over correlation matrices, $\eta > 1$ puts a prior that concentrates toward the identity matrix, and $\eta < 1$ puts a prior that concentrates away from the identity matrix. By setting $\eta = 2$, we bias the model slightly toward finding no relationship between days. This assumption is common in many Bayesian libraries (e.g., it is the suggested weakly informative prior for correlation matrices in Stan, `https://mc-stan.org/docs/stan-users-guide/multivariate-hierarchical-priors.html`).

[12] A parametric kernel, using, e.g., the day index 1-7 as its input, typically implies some smoothness over adjacent days.

[13] Our full implementation can be found at `https://github.com/rtdew1/detecting-routines/`, and in our Web Appendix L.

3.5. Inference

We estimate the model parameters in a fully Bayesian fashion using NUTS, a gradient-based MCMC sampler. To improve the scalability of the framework, we use the NUTS sampler implemented in NumPyro, and code our model in PyMC . This implementation of NUTS can be run on a GPU, which is significantly faster than CPU-based implementations.

In its simplest form, the above model can be computationally difficult: while discretizing the arrival times into hourly buckets makes defining the kernel and estimating the GPs easier (due to the limited number of inputs and natural structure between days, weeks, and hours), it also forces the model to do likelihood computations over many time periods in which nothing happened. That is, customers often interact with the firm sparsely, yet our likelihood function is specified as a count variable over all time periods $t = (w, d, h)$, which forces us to consider all the zeroes. To help facilitate inference in this set up, we draw on a property of Poisson variables described in Gopalan, Hofman, and Blei (2015). Specifically, the log-likelihood of our model for all observations from a single customer i can be decomposed into two terms:

$$\log p(y_i \mid \lambda_i) = \sum_{y_{it} \neq 0} y_{it} \log(\lambda_{it}) - \sum_t \lambda_{it} + C, \tag{15}$$

where C is a constant with respect to λ_{it}. The first term in this expression depends only on the non-zero values of y_{it}, while the second term is a simple sum over all λ_{it}. In this way, the likelihood can operate only on the non-zero values of y_{it}, circumventing the potentially problematic sparsity.

3.6. Parameter Recovery, Model Scalability, and Data Applications

We conducted simulations to test the model's ability to recover the data generating process and the model's performance under a number of different data settings. Specifically, we investigated three questions: (1) For data generated from the model, i.e., with known routines and stochastic transaction process, how well can the model's parameters be recovered? (2) How well does the model perform with varying degrees of data (i.e., number of customers, and number of time periods)? (3) How robust is the model's performance in the presence of customer churn? Here, we briefly summarize the results, and refer interested readers to Web Appendix A for more details.

Even with relatively little data (e.g., 100 customers over 20 weeks), the model can recover the data generating process: correlations between true and estimated (posterior median) parameters ranged from 0.96 for population parameters (like $\mu(j)$) to 0.74 for individual-level parameters (like

$\eta_i(j)$).[14] More importantly, if we look at the estimated number of routine and random requests, the correlation between simulated and estimated values is as high as 0.98. These results hold in the presence of churn, although model performance is best when churn rates are low, in which case churn is relatively rare, or fairly high, in which case most customers churn during the calibration period.

3.7. Model Extensions

The framework introduced above is quite general and only requires the analyst to have access to transactional data. In some cases, we may wish to incorporate covariates in the model specification to understand how other events, like firm interventions, or past service quality, may relate to routine and random usage. Such covariates can be included by expanding the rate specification in Equation 2 to include covariate effects. We give an example of such an extension, and describe the potential complexities that emerge when trying to meaningfully incorporate covariates in our model, in Web Appendix B.

Another potential extension of interest is modeling routines over different periods. The multiplicative structure of the day-hour kernel, combined with the additive structure of the overall routine rate, can be easily adjusted to handle such cases. For example, if the model were aimed at capturing yearly routines, with the main unit of analysis being weeks (i.e., routines in terms of which weeks of the year a person uses the service, year over year), the day-hour kernel introduced above could be changed to a single periodic kernel over weeks (with period 52), and the "weekly" kernel could be specified instead as a yearly kernel, capturing how the strength of the routine changes over years. If, in the same case, daily data were also available, one could decompose usage into days and weeks, specifying a periodic kernel for weeks, and a kernel for days (e.g., the unstructured approach suggested above), multiplying them together in a similar fashion to our own day-hour kernel. Finally, more granular time periods can also be used: if, for example, minute-level routines were of interest, our same model could be estimated with minute-level bins, rather than hourly.[15] We focus on hours because our empirical application lends itself naturally to that unit of analysis, and defer discussion of optimal granularity selection to other work (see, e.g., Kim, Bradlow, and Iyengar 2022). In short, the proposed structure is flexible enough to capture

[14]Note that this lower correlation for these parameters is expected: as we illustrate in the appendix, an individual's routine is only recovered if there are sufficient requests from that routine.

[15]Estimating minute-level routines would require estimating GPs over the 10,080 minutes in a week, which in turn, would require more scalable approaches to estimating GP models (e.g. Snelson and Ghahramani 2005; Loper et al. 2021)

many types of data granularities and routines, over many different time periods.

4. Application: Ridesharing

4.1. Data

We apply our model to data from Via, a popular NYC-based ridesharing company. The data contain detailed records on a sample of 2,000 customers who joined the platform between 2017-2018. Our specific data contain information on their interactions with Via over a 48 week period between January 2018 and November 2018. For each customer, we observe their acquisition date, though churn, if present, is unobserved. We discard the first three weeks of data after acquisition for each customer.[16] Of the 48 weeks, we use the first 38 weeks for calibration, and reserve the final 10 weeks for holdout validation.

Like most ridesharing platforms, Via uses a system for matching riders with rides. Specifically, when a customer uses the company's app to request a ride, their request generates a proposal, assuming a match can be found. The rider can then accept or reject that proposal. Unlike Uber or Lyft, however, Via operates primarily as a ride-*sharing* service, where customers typically share their ride with other customers, and often need to walk short distances from request locations to pick-up locations, and from drop-off locations to requested destinations. Thus, each proposal includes standard information like the cost of the ride, how long the driver will take to get there, and information about how far the user will have to walk to meet the driver.[17] Occasionally, a rider makes a request and then rejects it, possibly multiple times, looking for a better proposal. To handle situations like this, the company uses a unit of analysis called a session, which is a grouping of back-to-back requests. Following the company's lead, the dependent variable we focus on in our analyses is the number of sessions a given user has in a given hour.[18] Summary statistics for our session data are presented in Table 1 and in Figure 3. Most riders have either zero or one session per hour, and most users have less than 10 sessions per week. However, hours with more than one session are also observed.

Importantly, a user can have a session without actually taking a ride if they decline all of the

[16]Some customers' acquisition dates are within our calibration window; for these customers, their activity does not enter the model likelihood until 3 weeks after their acquisition.

[17]Our data also include these covariates, which we describe and use in a later section to understand how routineness moderates likelihood of accepting proposals, and making requests conditional on past trip quality.

[18]Throughout the paper, we will use the terms "request," "transaction" and "session" interchangeably, always referring to sessions.

Table 1: Summary statistics.

Summary statistics for our ridesharing data, summarized over the training data, unless otherwise noted.

Total Customers	2,000
Total Weeks (Training)	38
Total Weeks (Holdout)	10
Number of Sessions	86,952
Sessions / Customer	43.48
Sessions / Customer / Active Week	3.10
Active Weeks in Data / Customer	14.02

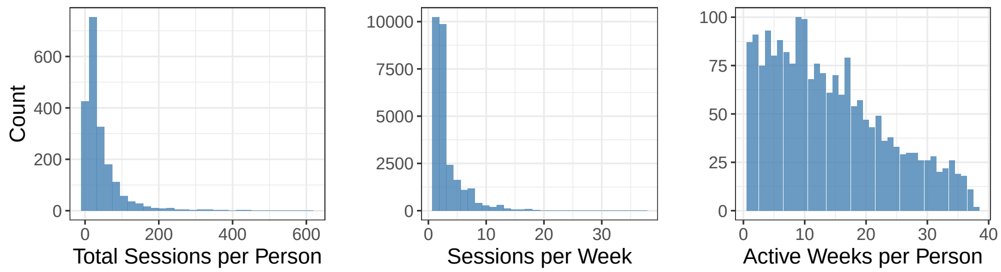


Figure 3: Distributions of summary statistics.

Distribution of three summary statistics in our training data: (1) the total number of sessions per customer; (2) the number of sessions per customer per active week; (3) the number of active weeks per customer.

proposals. We focus on requests and not whether the ride was eventually accepted or completed because it is the most granular level of engagement with the company. A request means the rider was interested in using the service at that time. That said, we further leverage the information about acceptance and rejection of proposals when we subsequently investigate the implications of routines for customer behavior and customer management.

4.2. Quasi-simulation Case Studies

To illustrate in more detail how our model works, we performed what we term a "quasi-simulation," combining real and synthetic customers. The goal of this simulation is to show that the model can recover meaningful patterns of behavior, under realistic data conditions,[19] even when those patterns are not explicitly generated by the model. To that end, we simulated the usage of 32 hypothetical customers, with rates of usage typical of customers in our data, and whose usage follows pre-specified, managerially meaningful patterns. These patterns include different types of routines and different patterns of overall usage, including switching between random and routine usage, and churning from the platform. We merged the data from these 32 synthetic customers with a sample of 500 real customers and estimated the model on this partly synthetic dataset. Be-

[19]By including real transaction data alongside hypothetical cases, we ensure that the model's population parameters and hyperparameters will be estimated at realistic values.

cause the usage patterns of the 32 synthetic customers were not generated from the model itself, they allow us to highlight the the model's ability to meaningfully estimate routine behavior for a range of data patterns, and illustrate how the model parameters capture phenomena not explicitly included in the model, like customer churn. Combining these simulated cases with a much larger set of real customer data ensures that the population-level parameters are consistent with reality. For the sake of brevity, the remainder of this section presents the individual-level model results from two of these simulated customers — one exploring the model's ability to detect routines separately from random usage, and the other illustrating how the model captures churn in the data. The results for the remaining simulated customers, including cases with noisy, clumpy, and regular behavior, are reported in Web Appendix C.

Case 1: Random then Routine In Figure 4, we plot the key model estimates for a simulated individual for whom routine behavior emerges over time. Specifically, this individual was simulated by drawing day-hour request times in two ways: for the first half of the data (i.e., before week 19), each week, we drew five day-hours completely at random, and assumed the individual makes one request at each of these five day-hours. Since the five day-hours are drawn anew each week, there is no pattern to this user's usage, and thus the model should capture this as random activity. Then, at week 19, we simulate this user suddenly adopting a routine. To simulate routine usage, we first drew a set of five random day-hours (e.g., Sunday at 2pm, Tuesday and Wednesday at 8pm, and Thursday at 8am and 6pm), and then assumed the user requests a ride at these same five day-hours each week. Since the user is making requests at the same times, week over week, the model should detect that a routine has emerged around week 19.

There are five panels in Figure 4: at the top left, we plot the posterior median estimates of E_{iw}^{Random} (black/solid) and E_{iw}^{Routine} (red/dashed). To the right of the decomposition, we show the posterior median estimates of the random scale parameter, $\alpha_i(w)$, and the routine scale parameter, $\gamma_i(w)$. Finally, below those, we show the posterior median estimate of the routine rate $\eta_i(j)$, plotted as an intensity over day-hours, and below that, the model's expectations for when this user will request rides during the last week of the training data ($w = 38$).

From Figure 4, we see the model is correctly able to parse this user's behavior: in the Decomposition panel, we see the random component E_{iw}^{Random} is high at the start, capturing around 5 rides per week. We can also see this reflected in the relatively high value of the random scale, $\alpha_i(w)$. Then, in the middle, we see a sudden shift, with E_{iw}^{Random} falling to zero, and E_{iw}^{Routine} ris-

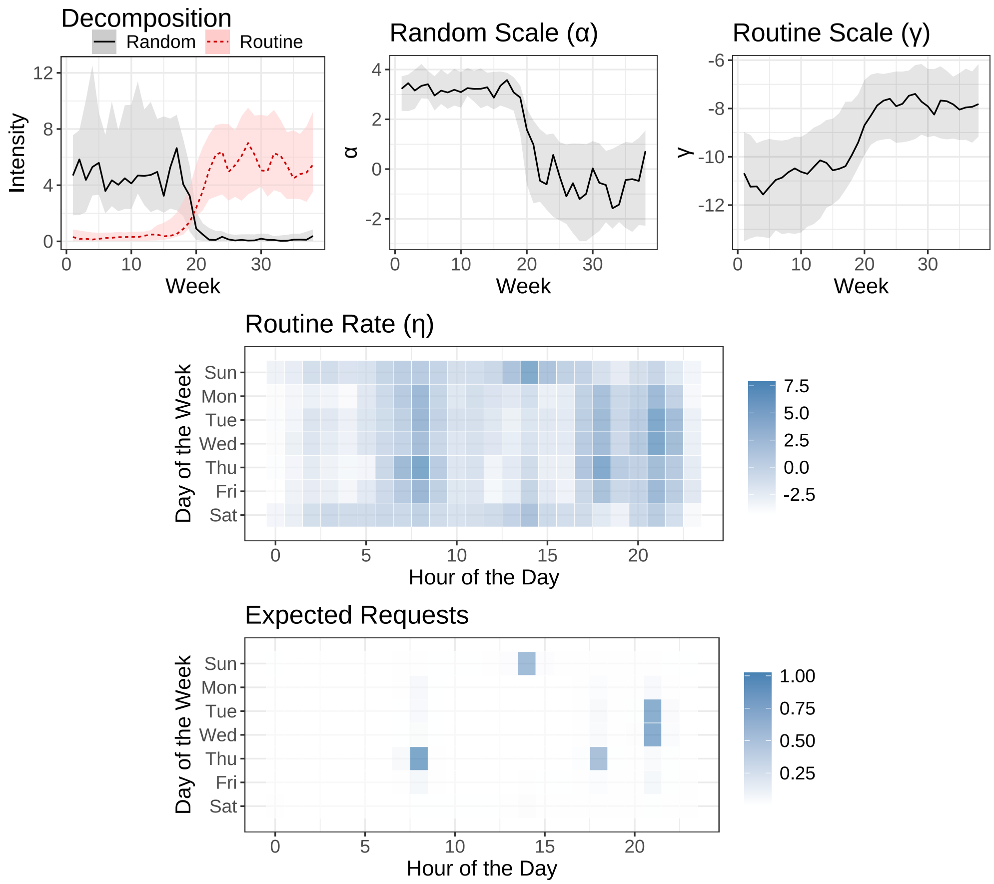


Figure 4: Simulated Case: Random then Routine.
Model estimates for a simulated individual who first uses the service randomly, then adopts a routine. Error bands are 95% credible intervals.

ing to around 5, corroborating the model's ability to detect routines. These changes are driven by changes in the scale parameters, $\alpha_i(w)$ and $\gamma_i(w)$: since the request times became consistent week over week, $\gamma(w)$ increases, meaning the day-hour rate of the model is dominated by $\eta_i(j)$, rather than $\mu(j)$. The relative smoothness of these trajectories, especially $\gamma_i(w)$, is driven by the GP assumption for these terms, which smooths over time, with a degree of smoothing governed by the GP's length-scale parameters, ρ_α and ρ_γ. Finally, the times that the user is expected to request a ride are captured in the user's routine rate, $\eta_i(j)$, for which we can see there are five peaks in usage. These peaks, when combined with the routine scale $\gamma_i(w)$ in week 38, translate exactly to five expected requests at exactly the hours simulated: Sunday at 2pm, Tuesday and Wednesday at 8pm, and Thursday at 8am and 6pm.

Case 2: Routine then Churn In Figure 5, we see the results for a different simulated user, who first exhibited routine behavior (generated analogously to the last 19 weeks of the simulated customer in case study 1), but then stopped using the service altogether. Although our model is not explicitly designed to detect churn, inactivity can be captured in our framework when both scaling terms become very negative, implying zero expected requests. Indeed, we see that this is exactly how the model behaves: we see in the first panel that the decomposition correctly captures, at first, a high level of routine usage, which then dips to zero at the midpoint, when the user churns. Looking at the model components, we see this pattern of routine requests is driven by the routine scale parameter, γ, which starts out relatively high (when the user is active), but then plummets and stays low until the end of the data. Again, the routine rate, η, can recover the correct routine for this user, with five peaks (Fri at 1pm, Sat at 3pm, and so forth). However, as reflected in the bottom figure, when that routine rate is combined with a very negative routine scale, we see that the model predicts essentially no requests for the last week of the data, when the customer becomes inactive.[20]

5. Results

Having established the model's ability to separate routine behavior from random behavior, and its ability to accurately recover routines across different data settings, we now turn to describing the results from the real data, estimated on the full sample of 2,000 customers over the period of 38 weeks used for model calibration.

5.1. Model Estimates

We first describe some of the population-level parameter estimates which characterize usage patterns broadly; for example, what days and times exhibit the highest level of usage across customers, and how often users exhibit random vs. routine behavior. We then describe some individual case studies, exploring the degree of routineness and the specific routine patterns for individual consumers.

[20]One may also notice that there is an apparent pattern in the random scale, where there seems to be a decrease in transactions around week 30: this decrease is an artifact of scale. When combined with the estimate for μ and exponentiated, all these very negative numbers still suggest zero requests. We sometimes observe random fluctuations like this, purely due to this lack of identification between negative values, especially when the random process is zero throughout.

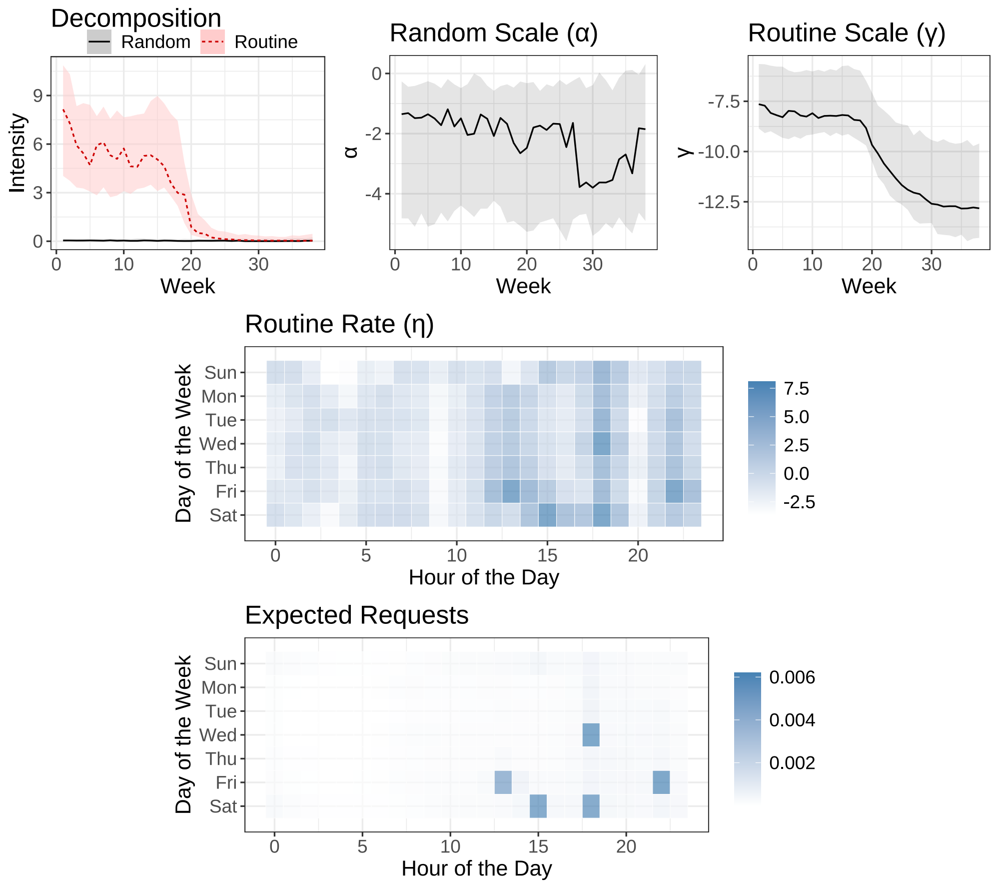


Figure 5: Simulated Case: Routine then Churn
Model estimates for a simulated individual who uses the service in a routine pattern, but then churns. Error bands are 95% credible intervals.

5.1.1. Population Patterns

There are two main population-level parameters of interest: the population-level rate parameter, $\mu(j)$, which governs when users tend to take rides (randomly), and the correlation matrix Ω from the day-hour kernel, which describes how different days are related to one another. We plot the posterior means of these quantities in Figure 6.

Some intuitive patterns emerge: first, from the posterior mean of $\mu(j)$ (at left), we see that random needs tend to arise during all times, except in the middle of the night (i.e., hours 2-5, or 2 AM to 5 AM). This pattern is moderated somewhat on the weekends, when travel times shift a bit later, and when there is a noticeable drop in usage at 4 AM, corresponding to the closing time of many bars in New York City. On weekdays, we also observe a slight increase in usage of the service in the evenings, but the daytime variation is much less stark than the variation between day and night. Similarly, the correlation matrix (at right) captures the expected pattern

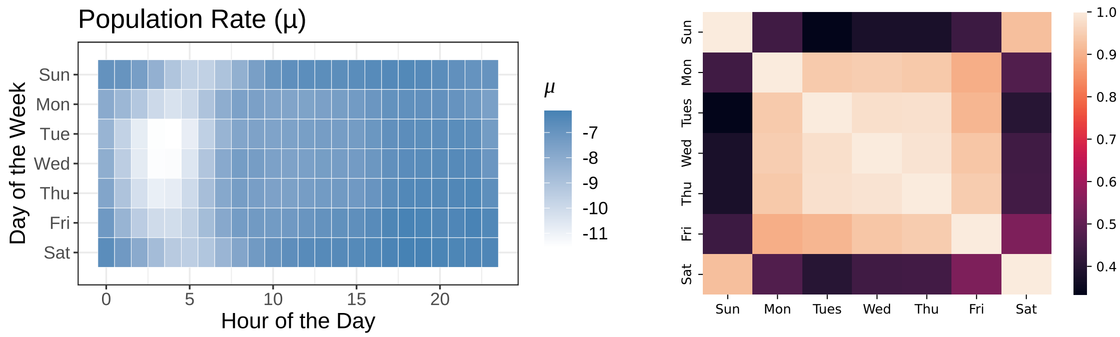


Figure 6: Posterior means of $\mu(j)$ and Ω.
At left, the posterior mean of $\mu(j)$, the common rate of usage across individuals at the day-hour level. At right, a visualization of the posterior mean of Ω, the correlation matrix across days for routines. Lighter colors indicate higher correlation.

that weekdays tend to be more similar to one another than to weekends. Saturday and Sunday are correlated, as are Friday and Saturday.

Another key output of our model is the decomposition of usage into routine and random requests. Figure 7 shows the joint distribution of the two parts E_{iw}^{Routine} and E_{iw}^{Random} in the last week of our data. We find an L-shaped distribution, suggesting that heavy usage customers are either primarily routine or primarily random but rarely both. Most customers fall in the lower left part of the figure, with few requests per week, balanced between random and routine. Although Figure 7 shows the decomposition pattern for the last week of the data, we also find similar weekly decompositions throughout the data period.

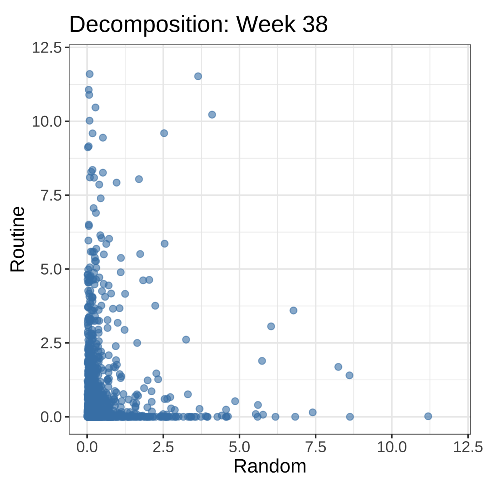


Figure 7: Joint distribution, E_{iW}^{Routine} and E_{iW}^{Random}.
The joint distribution of the posterior medians of E_{iW}^{Routine} and E_{iW}^{Random}, where $W = 38$, the last week of the data.

Finally, the combination of routine and random usage, $E_{iw}^{\text{Routine}} + E_{iw}^{\text{Random}}$, should capture overall usage (i.e., $E(y_{iw})$). Indeed, we find that to be true: in-sample, the correlation coefficient between expected usage and actual usage is $r = 0.947$, $p < 0.001$, reflecting very good fit.

5.1.2. Individual Customers' Routines

We now zero in on the individual-level parameters, to illustrate the insights provided by the model. Relative to the simulated examples, the results on real users are less clean cut in their interpretation, but still offer valuable customer-level insights. In Figures 8 – 9, we show the same posterior estimates and decompositions for two real customers, as we did for the simulated case studies in Figures 4 – 5.

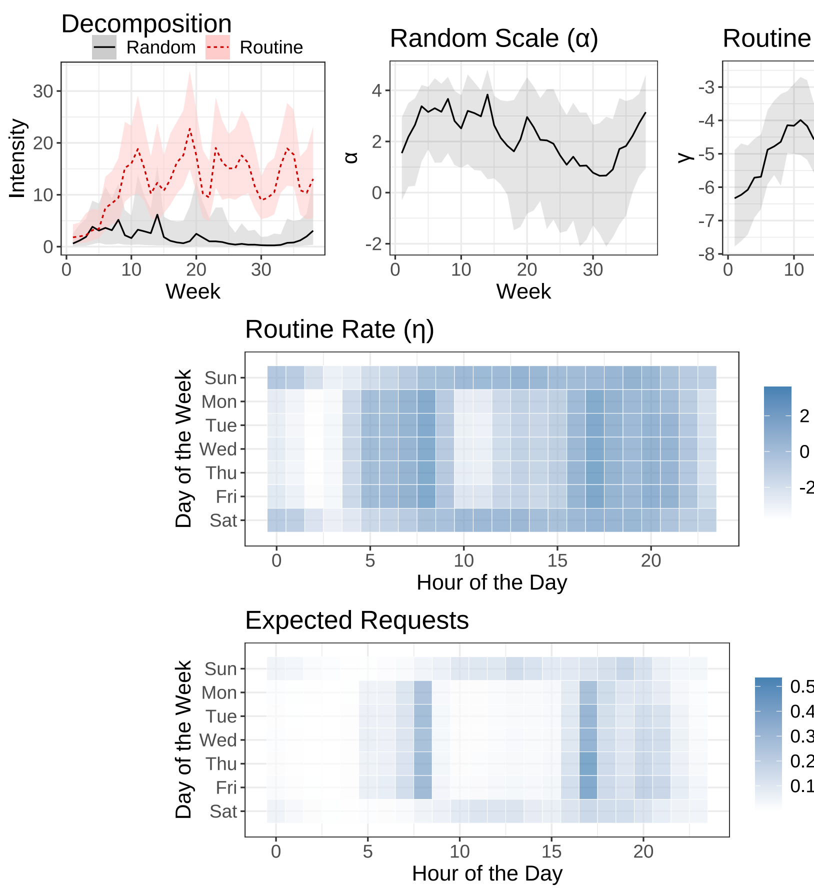


Figure 8: Real User: Commuting Routine.
Model estimates for an individual who is active throughout the 38 week training window, and uses the service in a routine, typically in commuting hours.

In Figure 8, we show an example of a very common type of routine: commuting. As shown in the decomposition, this customer is a fairly heavy user, making roughly 14 ride requests per week, with a high-level of routine usage. This routine usage tends to cluster around commut-

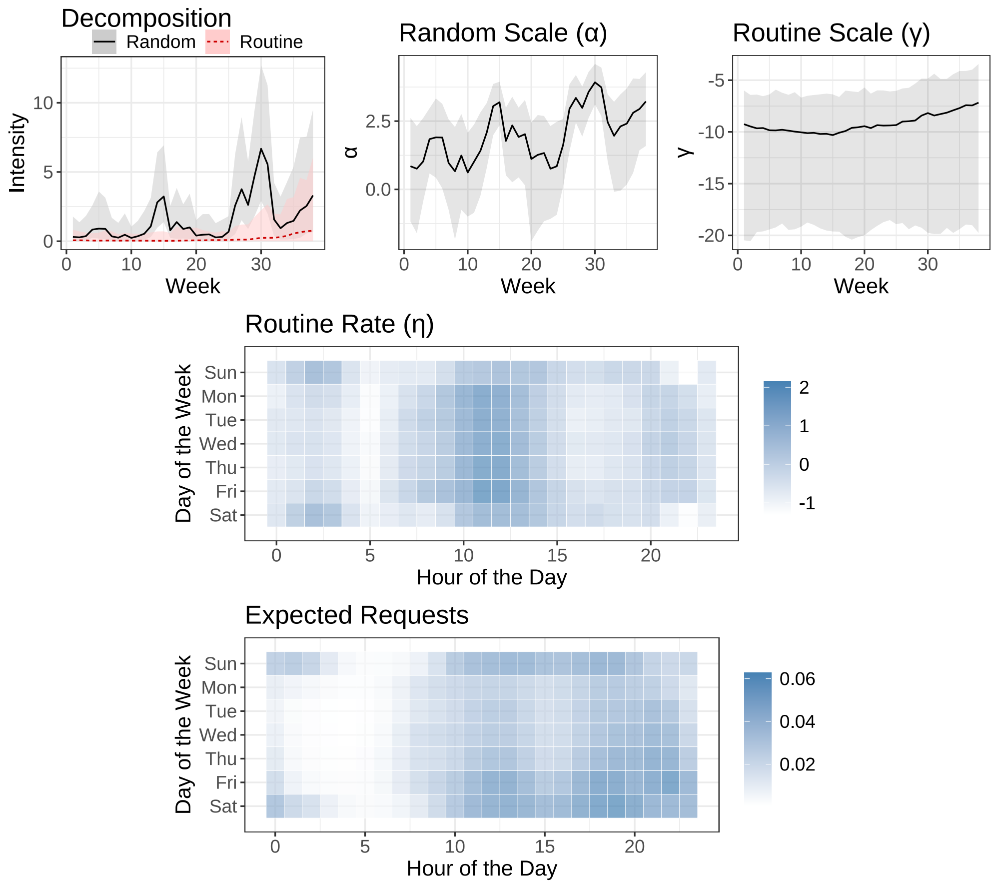


Figure 9: Real User: Random Usage.
Model estimates for an individual who is active throughout the 38 week training window, and uses the service at random.

ing hours, 8am and 5pm, as can be seen both in the routine rate and in the expected numbers of requests. In contrast, in Figure 9, we show a customer with a random pattern of usage. This customer transacts less frequently than the first customer. In the decomposition, we see the random component trending upward toward the end of the calibration window, driven by the increase in this customer's random scale. That the model captures this increase in usage with the random scale suggests the day-hour pattern of those interactions is not consistent, week over week. While the model does still estimate a routine rate, $\eta_i(j)$, when combined with the customer's very low routine scale $\gamma_i(w)$, we see that the customer's expected day-hour pattern of usage is very diffuse, very much resembling the population pattern shown in Figure 6.

5.2. Heterogeneity in Routines: Uncovering Routine Types

While the case study in Figure 8 captures an intuitive and common routine, other users might have different routines. To understand the typical types of routines present in our data, we clustered

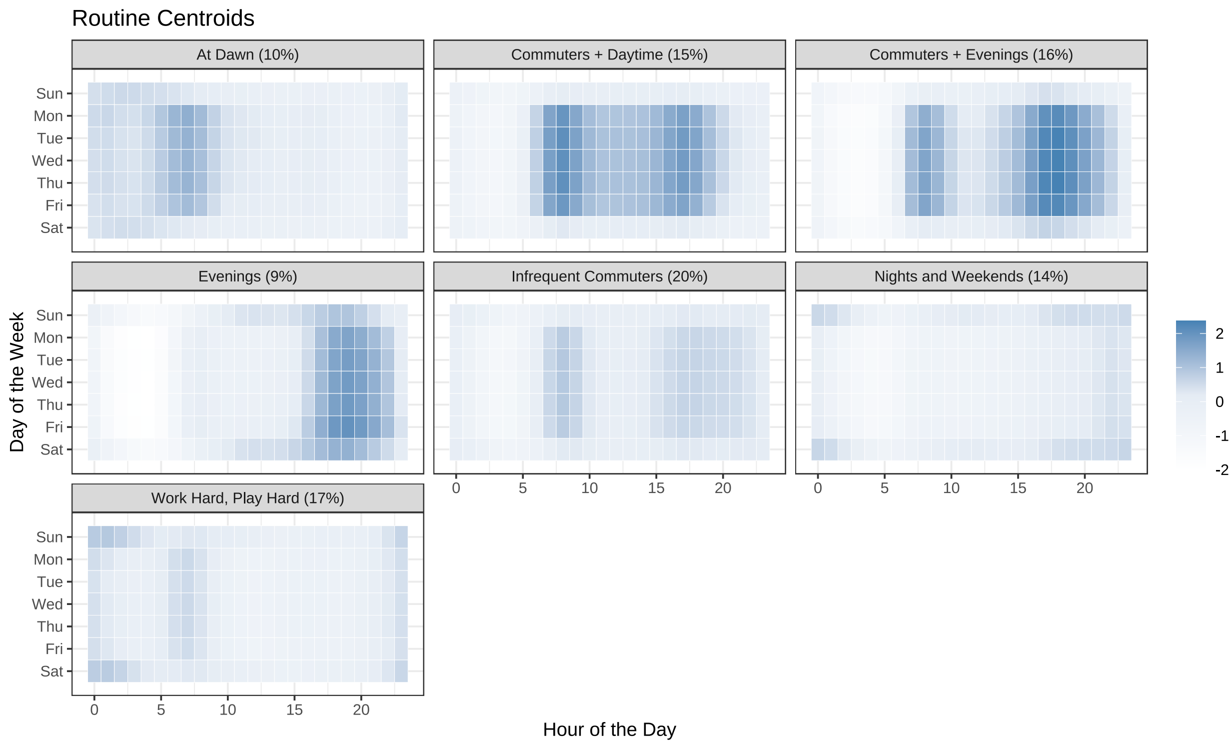


Figure 10: Heterogeneity in Routines: Routine Cluster Centroids
Average values of $\eta_i(j)$ per cluster, with cluster labels in the facet titles, along with cluster sizes as a percentage of customers.

our posterior median estimates of customers' routines, $\eta_i(j)$. Specifically, we focused on those customers who had at least 5 routine rides in total over the calibration period (i.e., $\sum_{w=1}^{38} E_{iw}^{\text{Routine}} \geq 5$). This filtering ensures that the $\eta_i(j)$ parameter captures a meaningful routine,[21] and resulted in 1,042 customers. Then, we performed K-means clustering on the estimated $\eta_i(j)$ parameters and uncovered 7 distinct routine types in ridesharing behavior, which we labeled according to the routine pattern and summarize in Figure 10.[22]

We see that, while commuting is a common routine type, there are also other common routines. We find two types of likely leisure-oriented routines: the "Nights and Weekends" (14% of customers) customers primarily make requests at night, especially on the weekends. The "Work Hard, Play Hard" (17% of customers) cluster exhibits weekend usage that peaks between midnight and 4am (the closing time of most bars in NYC), and weekday morning and nighttime usage.

[21] Per our discussion in the preceding sections, the model always estimates $\eta_i(j)$, but for consumers who have no actual week-over-week consistency, $\gamma_i(w)$ will be very negative, and $\eta_i(j)$ will be meaningless.

[22] More specifically, we used the parametric longitudinal k-means method (lcMethodLMKM in the latrend package in R) designed for clustering trajectories. The number of clusters was selected based on standard K-means metrics, specifically by examining the scree plot of the weighted mean absolute error (WMAE) of the clustering solution. For details, see Web Appendix H.

There are also two clusters that use the service only during one part of the day, either just in the morning ("At Dawn," 10%), or just in the evening ("Evenings," 9%). Understanding this diversity of routines is a key benefit of our individual-level model. For example, in this case, realizing the magnitude of the number of "half commuters," like the "At Dawn" and "Evenings" clusters, spurred our partner company to try to understand how they can capture the full commute for these groups of customers. We will return to these routine types, and explore their differential value to the company, in a later section.

5.3. Model Validation

Before exploring the implications of routineness for customer relationship management, we first validate the model, by examining its ability to predict our holdout data. Mechanically, making predictions from the model is straightforward. Since $\mu(j)$ and $\eta_i(j)$ do not vary over weeks, the only two components of the model that need to be projected forward are $\alpha_i(w)$ and $\gamma_i(w)$. To do so, we utilize the fact that GPs are marginally Gaussian to derive the posterior predictive values, given what we previously observed. Here, we focus on doing this for $\gamma_i(w)$, but the math is analogous for $\alpha_i(w)$. Let w^* indicate a new week that we want to make a prediction for, having observed weeks $\mathbf{w} = (w_1, \ldots, w_W)$, with estimated function values $\gamma_i(\mathbf{w})$. Then $\gamma_i(w^*) \sim N(m^*, s^*)$, where,

$$m^* = \gamma_{0i} + K(w^*, \mathbf{w})K(\mathbf{w}, \mathbf{w})^{-1}[\gamma_i(\mathbf{w}) - \gamma_{0i}], \tag{16}$$

$$s^* = k(w^*, w^*) - K(w^*, \mathbf{w})K(\mathbf{w}, \mathbf{w})^{-1}K(\mathbf{w}, w^*), \tag{17}$$

and where $K(\mathbf{w}, \mathbf{w})$ is the kernel matrix (on the training data), $K(w^*, \mathbf{w})$ is the vector formed by evaluating the kernel $k(w^*, w)$ for all $w \in \mathbf{w}$, and likewise for $K(\mathbf{w}, w^*)$. Examining this equation closely reveals an important feature of GP models with stationary kernels, like our exponential kernel: when forecasting far away from observed data, GPs revert to the mean. This reversion is driven by $K(w^*, \mathbf{w})$ going to zero when the inputs are far apart.

Using this forecasting machinery, we focus on two types of predictions: first, we predict how many sessions someone will have in the future. Second, and more pertinent to our research objective, we predict *when* someone will request rides, in terms of day-hours of a particular future week. We compare the predictive performance of the model to four benchmarks:

- **Non-routine Usage (NR):** A version of our model with the routine term set to zero, equivalent to a Poisson model with rate $\lambda_{it}^{\text{NR}} = \exp[\alpha_i(w) + \mu(j)]$. Comparing the performance of the full model to this benchmark gives a sense of what the routine part of the model captures.[23]

- **No Day-Hour Variation (NDH):** A version of our model with all j terms eliminated, equivalent to a Poisson model with rate $\lambda_{it}^{NDH} = \exp[\alpha_i(w)]$. Comparing the full model to this model gives a sense of what adding day-hour variation captures.
- **Pareto-GGG:** The Pareto-GGG model of Platzer and Reutterer (2016), as implemented in the `BTYDPlus` package.
- **LSTM:** An LSTM deep learning model, similar to Sarkar and De Bruyn (2021)'s LSTM for direct marketing, but with the objective of predicting which times in a week a user will request a ride. For each individual in the data, we train an LSTM with the full 38 week calibration data, then use a sliding window of data to predict, in the subsequent weeks, which day-hours are most likely to have a ride request.[24]

5.3.1. Predicting Volume of Requests

First, we consider the task of predicting how many sessions each customer will have during the 10-week holdout period, ignoring the actual timing of those sessions. A priori, the accuracy of these forecasts for our models will depend on how representative past usage rates are of future usage rates, given the mean reversion property of GPs described previously. We also expect the Pareto-GGG, and BTYD models more generally, to do quite well at this task, as predicting a cumulative number of transactions during a holdout window is often the main use-case for these models. For our LSTM, the loss and training procedure were focused on classifying and ranking request timing (i.e., our next prediction task), not on predicting request volume. It is not a priori obvious how well the LSTM will perform on the distinct (but related) task of forecasting request volume.

In Table 2, we show the mean absolute errors in forecasting the number of requests, where the mean is computed across customers, with their corresponding 95% confidence intervals. We can see that the forecasting results match our intuition: the Pareto-GGG and our proposed model

[23] Note that the Non-Routine model is a separately estimated model. We are not just zeroing out the non-routine term from the full model.

[24] While our model is similar to that of Sarkar and De Bruyn (2021), in the sense that we trained LSTMs, our approach is different in the specifics of our architecture, and in the loss functions used: our goal is to predict, given a customer's history, whether that customer will make a request during each of 168 day-hours; in the parlance of deep learning, it is a classification task, with 168 outcomes. In contrast, Sarkar and De Bruyn (2021) explore predicting donation incidence, given a set of features describing company-customer interactions. For a detailed description of our LSTM benchmark, see Web Appendix D.

Table 2: MAE: Predicting Number of Requests

Mean absolute error across customers for the number of sessions during our 10-week holdout period. Intervals are 95% confidence intervals.

	Proposed Model	NR	NDH	P-GGG	LSTM
MAE	7.73	8.30	8.31	7.51	26.13
	[7.17, 8.28]	[7.64, 8.96]	[7.65, 8.97]	[6.97, 8.05]	[19.45, 32.80]

achieve statistically indistinguishable performance. The other versions of our model also perform well, which makes sense, as we have removed sources of day-hour variation but left the week-over-week variation components intact. The fact that these versions perform marginally worse is suggestive of the additional predictive benefit of including day-hour information. Finally, the LSTM performs quite badly. As we describe in more detail in the appendix, this poor performance is likely the a result of two aspects of how the LSTM was trained: first, its loss is focused on the probability that a request occurs in a given day-hour. Second, the LSTM uses a sliding window to make "one week ahead" forecasts, which, in the holdout period, requires assuming predictions are true to forecast more than one week ahead. To get good performance on the (subsequent) day-hour prediction task, we found it necessary to set the threshold for predicting a request would materialize to be relatively small, which, in this case, leads to overforecasting the actual volume of requests.

5.3.2. Predicting Request Timing

Beyond just predicting how many requests a user will make each week, our model also captures when a request will occur *within* that week, by estimating day-hour rate terms $\mu(j)$ and $\eta_i(j)$. When considering an appropriate metric for validating our model, it is important to consider the continuous nature of time. Imagine a case where a customer takes a ride at 9am, but the model thought the most likely time for such a ride was in the 8am hour. In well-calibrated models that consider the continuity of time, the expected rate of transacting at 8am and 9am should be similar, given 8:59am and 9:00am are a mere minute apart. Yet, standard classification metrics like hit rates fail to take such continuity into account. Thus, a better way to measure the quality of timing predictions is through metrics for rankings, wherein the model produces a ranking of the most likely request times, and success is measured by how highly ranked the actual request times are. In our running example, the well-calibrated model should give similar rankings to the 8am and 9am times, and thus, would score similarly well in terms of predictive ability even if the customer happened to use the service at 9:00am, rather than 8:59am.

We use two ranking metrics to measure how well the model predicts request times: mean average precision (MAP), and conditional precision (CP). For both metrics, higher values represent better rankings. These metrics are both standard in the literature on recommendation systems, where they are used to evaluate the relevance of a ranked list of recommended items.[25] To calculate these statistics for both our model and the nested non-routine version, we look at the ranking implied within each week by the estimated transaction rate (i.e., λ_{it} and $\lambda_{it}^{\mathrm{NR}}$). Because the no day-hour model implies a uniform rate across all day-hours, the ranking statistics are undefined for this model, and we do not include its results in this section. For the Pareto-GGG, we compute the rankings implicitly, using predictions. For each customer, we use the `BTYDPlus` package to draw from the posterior predictive distribution of transacts for each customer. This distribution is cumulative, reflecting the total number transactions we expect to see by time t. Thus, to get an estimated transaction rate for each hour, we evaluate the cumulative predicted transactions for each hour of the holdout period, and compute cumulative differences. We then rank these hours in each week. Finally, for the LSTM, similarly to our proposed model, we form a ranking by ranking the model's day-hour estimated probabilities within each week. The results for the 10-week holdout period are shown in Table 3.

Table 3: Holdout Ride Timing Metrics

MAP and CP statistics for predicting ride timing in the holdout period. Intervals are 95% confidence intervals.

	Proposed Model	**NR**	**P-GGG**	**LSTM**
MAP	0.131	0.069	0.043	0.110
	[0.124, 0.139]	[0.064, 0.074]	[0.041, 0.045]	[0.076, 0.102]
CP	0.072	0.029	0.016	0.089
	[0.066, 0.079]	[0.025, 0.033]	[0.013, 0.018]	[0.076, 0.102]

First, our full model dramatically improves ride time predictions compared to the model with no individual-level routines (NR). This is not surprising: the only part of the NR model that predicts ride times is $\mu(j)$. In that sense, the NR model is assuming the same day-hour ranking across all customers, or just one "population routine," corresponding to the day-hours that customers, in general, are likely to call rides. By comparing the full model with this baseline, we corroborate

[25]See Web Appendix F for details of calculating MAP and CP in our application. For excellent expositions of MAP and recommender systems, see http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html and https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html.

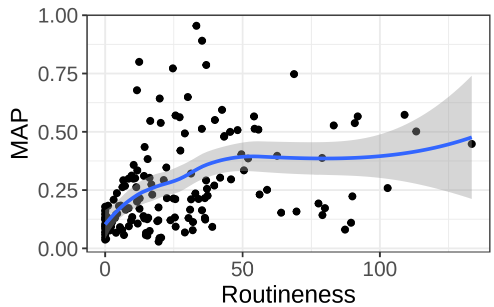


Figure 11: MAP by Number of Routine Rides.
Points on the plot represent customers. On the x-axis, we plot the estimated routineness, summed over the holdout period, per person. On the y-axis, we plot the estimated MAP for that customer. A LOESS smoothing is overlaid, showing the generally increasing pattern.

that there is rich variation in the data in terms of when individual customers request rides, highlighting the predictive validity of the routine component of the model. Our model also improves considerably on the Pareto-GGG, a model that captures regularities in transaction timing, but is not specifically designed to predict day-hour patterns. While, in the extreme, such regularities could theoretically predict transaction timing down to the day-hour, capturing regularity in ride times in our application is not sufficient to predict ride timing accurately. Finally, the LSTM, which was trained specifically for this task, performs on par with our model.

It is noteworthy that the statistics in Table 3 are modest in magnitude. For instance, the CP metric suggests that we are only able to accurately predict roughly 10% of the out-of-sample session times. An important caveat here is that these metrics ignore that some trips are routine, while others are not. By definition, we only expect to be able to predict the *routine* trip times. In Figure 11, we show evidence of this phenomenon: the more routine a person is, the higher their MAP. The plot for CP is nearly identical. These results suggest that our model does, indeed, capture session timing within weeks, when that timing is predictable.

6. Routineness and Customer Management

Having established the validity of the framework, we now return to one of the central questions of the paper: Can routineness help firms better understand and manage their customers? One crucial advantage of being able to distill transaction timing to a single metric — routineness — is that we can subsequently explore how routineness relates to many outcomes of interest, without the need of building new models. In this section, we start by showing that routineness is a key predictor of customer value, over and above other transaction characteristics. We then explore how routineness moderates consumers' reactions to different aspects of the ridesharing service, and how customer value varies by routine type.

6.1. Routineness and Customer Value

Overall, routine customers comprise a significant part of the value in our sample: the 20% of customers who, during the calibration period, had on average one routine request per week (or at least 38 routine requests) make 53% of all sessions and 51% of the total revenue in the holdout period. Alone, these statistics are suggestive, but limited: is it these customers' routineness that explains their high out-of-sample value, or merely their high overall request rates? To understand whether routineness explains value over and above *other* summaries of transaction behavior, we turn to regression analysis. Specifically, we consider the routineness of each user at the end of the calibration period (i.e., E_{i38}^{Routine}, as estimated by the model) and relate it to: (1) `# Requests`, defined as the number of requests a customer makes in the holdout period, and (2) `Active`, defined as whether a customer is active at all in the holdout period (i.e., a measure of 10-week retention). When modeling requests as the dependent variable, we use simple OLS; for modeling activity, we use logistic regression. In each of these regressions, we control for observable characteristics that strongly predict future value, including the number of requests the customer made in the last week of the calibration data, and the commonly used recency and frequency variables that capture how recently a customer last made a request and how many requests the customer has made previously. Moreover, to see if routineness explains behavior over and above extant summaries of transaction timing, we also include the calibration period clumpiness (H_p) and regularity (as estimated by the Pareto-GGG's k_i parameter) as predictors.[26]

Before describing the results, we note two important aspects of these regressions: first, routine requests are part of the total number of requests. As high usage can result from either random needs or routines, this specification allows us to understand whether having a higher routine component is incrementally valuable, over and above controlling for just the level of usage. In other words, by controlling for the number of requests in $w = 38$, we are trying to determine whether the "shape" of within-week usage, in terms of request timing, matters for explaining future customer value. Second, the inclusion of recency and frequency metrics is especially important here:

[26] The connections between these metrics and routineness described previously may raise concerns about multicollinearity in this analysis. However, in our data, we do not find these metrics to be problematically correlated: as expected, clumpiness and regularity are negatively correlated ($r = -0.21, p < 0.0001$), whereas clumpiness and routineness are very weakly negatively correlated ($r = -0.08, p = 0.0005$), and regularity and routineness are modestly positively correlated ($r = 0.42, p < 0.0001$).

a litany of models in customer base analysis have shown that these metrics are key summary statistics for predicting repeat purchasing (e.g., Schmittlein, Morrison, and Colombo 1987; Fader, Hardie, and Lee 2005; Blattberg, Neslin, and Kim 2008). If mere habit (in the "buying habit" sense of the word) were the primary driving force behind customer value, we would expect these two statistics to explain much of the variation in future transactions. Thus, by incorporating these measures in the model, we can establish whether routineness matters, beyond what mere habit would already predict.

Table 4: Relationship between Customer Value and Routineness

Regressions of future activity – either number of future sessions (models 1-3), or a binary measure indicating any activity at all (models 4-6) – on customer-level summary statistics, including regularity, clumpiness, and routineness. The dependent variable is measured either over our entire holdout period (10 weeks), or just in the last 5 weeks of the holdout period. Standard errors are in parentheses.

	Dependent variable:					
	# Requests			Activity		
	OLS			*Logistic*		
	(1)	(2)	(3)	(4)	(5)	(6)
Weeks of Test Data	All 10	All 10	Last 5	All 10	All 10	Last 5
Requests ($w = 38$)	3.838***	1.614***	0.375**	0.611***	0.461***	0.203***
	(0.184)	(0.241)	(0.154)	(0.103)	(0.106)	(0.056)
Recency	−0.238***	−0.284***	−0.157***	−0.140***	−0.134***	−0.123***
	(0.045)	(0.043)	(0.028)	(0.010)	(0.010)	(0.010)
Frequency	0.130***	0.097***	0.052***	0.0003	−0.002	0.002
	(0.008)	(0.008)	(0.005)	(0.002)	(0.002)	(0.002)
Regularity (k)	10.533***	4.511**	2.401*	0.313	0.005	0.407
	(1.986)	(1.952)	(1.244)	(0.463)	(0.484)	(0.395)
Clumpiness (H)	9.269***	8.443***	4.845***	−1.491***	−1.520***	−0.628*
	(1.918)	(1.836)	(1.170)	(0.380)	(0.381)	(0.326)
Routine ($w = 38$)		5.216***	2.685***		1.035***	0.262**
		(0.385)	(0.245)		(0.328)	(0.124)
Observations	2,000	2,000	2,000	2,000	2,000	2,000
R^2	0.540	0.579	0.429			

Note: *p<0.1; **p<0.05; ***p<0.01
Intercept omitted for clarity.

The results of these regressions are shown in Table 4. We estimate each model three times: omitting the routineness metric (columns 1 and 4), measuring the dependent variable using the entire holdout sample of 10 weeks (columns 1-2 and 4-5), and measuring the dependent value using the last month of the holdout data (i.e., the last 5 weeks,columns 3 and 6). The intent behind splitting the DV in this way is to assess how robust routineness is in explaining short- and mid-term customer behavior. We find that higher routineness is positively and significantly associated both with the number of requests a customer makes, and with the customer being active at all. Furthermore, in comparison to the simpler models (columns 1 and 4), routineness not only improves model fit, but is the only metric among all transaction timing metrics considered that positively and significantly explains customer activity levels, even in the mid-term. In sum, even after controlling for the number of requests a customer made at the end of the training data, standard recency and frequency measures from the CRM literature, and clumpiness and regularity, we find that number of routine requests is positively, significantly, and incrementally associated with higher request rates, and a higher tendency to remain active, suggesting customers with routines are more valuable.

6.2. Routines and Other Customer Behaviors

Next, we consider whether understanding customers' routines can be useful for customer management in ways beyond predicting activity levels. In particular, we consider two related questions: first, do highly routine customers interact with the firm's service differently than non-routine customers? And second, do customers behave differently during their routines? We hypothesize that customers whose usage stems primarily from a routine may not only be more likely to engage in activities that are directly valuable to the firm, but may also react differently to various aspects of the firm's service, like pricing and service failures (e.g., pick-up and drop-off delays). There may also be differences between routine and non-routine users in terms of which aspects of the service are more important to them. For instance, users who routinely rely on the service may place higher importance on things like convenience of trips. Such effects may not exist just across customers, but also within customers. For example, if a price change or service failure is associated with a trip that is part of a customer's routine, the customer may react differently than if the price change or service failure was associated with a non-routine trip. The direction of these within-customer effects is not a priori obvious: while the automaticity of routines implies that usage is stickier regardless of variability in service, suggesting that customers may be less sensitive to service quality during their routines, it is equally plausible that the high degree of familiarity customers have with service during their routines may exacerbate their reactions to any deviations from normal service.

To explore these hypotheses, we need to be able to quantify the overall routineness of a given customer, and the degree to which a given trip is *part* of that customer's routine. Both quantities are easily derived from our model. The former is what we previously referred to as (weekly) routineness, measured by $E_{iw}^{Routine}$. The latter we will refer to as *trip routineness*: given an observed trip occurs during week w, day-hour j, the trip routineness is the expected number of routine requests estimated by the model at that time, $E_{iwj}^{\text{Routine}} = \exp[\gamma_i(w) + \eta_i(j)]$.

To understand how these two aspects of routines connect to the company's service, recall that our data include information about the rides that users requested. Some of these variables are *characteristics of the proposal*, including the cost to the user (`Price`), the time until the driver can pick the customer up (`Driver ETA`), how long the customer will have to walk to get the ride (`Pickup Walking Dist.`), the expected time and total distance of the trip (from which we compute `Speed`), and the number of passengers for that request (`# Passengers`). We observe these characteristics for all the requests in the data. Moreover, for rides that were realized — that is, requests that ended up in a trip — we observe variables that capture the quality of the ride. These include whether the driver picked up the rider on time (`Pickup Delay`, which we measure in minutes), whether there were delays in the trip (`Dropoff Delay`), how far the rider had to walk from their drop-off to their final destination (`Dropoff Walking Dist.`), and whether there were other passengers in the car during the trip (`# On-board (Pickup)`, `# On-board (Dropoff)`, and `Max On-board`).[27] Based on this data, we ask two questions: (1) how likely is a customer to accept a proposal, and, particularly, a less favorable proposal, and (2) given a customer accepts a proposal (i.e., takes a ride), how likely is that customer to request a ride again within 7 days, particularly after a service failure? More importantly, we explore how the two types of routineness explain and moderate these dependent variables.

To answer these questions, we run four separate OLS regressions, the results of which are shown in Table 5.[28] The dependent variable in each regression is either accepting a proposal or requesting again within a week. In the first two regressions (models 1-2), the key independent variable of interest is the routineness of each customer at the end of our calibration window, and the unit of analysis is rides requested or taken during our holdout window. This setup allows us to address the "between-customers" question: does a customer's current period routineness explain their subsequent behavior? In these regressions, we include customer-level random effects to control for other unobserved differences across customers. In the second two regressions (models

[27]The full data is summarized in Web Appendix G

[28]We use OLS for these regressions, as opposed to logistic regression, to aid in the interpretability of interaction terms.

3-4), the key independent variable of interest is *trip* routineness, and the analysis is done at the level of requests within our calibration period. We focus here on in-sample rides, to ensure our routineness metric is accurately characterizing the nature of each request. This setup allows us to address the "within-customers" question: do customers behave differently when a trip is part of their routine? Since trip routineness is measured at the request-level, we use customer-level fixed effects to control for potential customer-specific unobservables. In all four regressions, we also include variables describing the proposal. In the "request again" analyses, which condition on a customer actually having completed the trip, we include variables describing the completed trip. Finally, to understand the potential moderating role of routines in explaining these outcomes, we include interactions of all of proposal and trip-related variables with the focal measure of routineness.

Focusing first on the between-customers results, we see that highly routine customers, as measured by week 38 routineness, are indeed different in their holdout behavior. Such customers are more likely to accept proposals in general (OLS coefficient $\beta = 0.077$, $p < 0.01$), and more likely to make requests within a week of any given completed trip ($\beta = 0.052$, $p < 0.01$). In terms of accepting proposals, while longer driver ETA is associated with lower acceptance rates ($\beta = -0.050$, $p < 0.01$), there is a positive interaction with routineness ($\beta = 0.011$, $p < 0.01$), suggesting routine customers are more willing to wait for their rides to come. On the other hand, more routine customers appear to especially prefer faster speeds (interaction $\beta = 0.142$, $p < 0.01$) and shorter walking distances (interaction $\beta = -0.009$, $p < 0.01$). This pattern of moderation is consistent with routine customers caring more about convenience-related variables. In part, this could be driven by self-selection: customers do not adopt a routine if rides are inconvenient for them. In terms of requesting again within a week, we see a prominent effect of week 38 routineness on two very important service variables: price and pickup delay. While higher prices and longer delays are both associated with lower likelihood of returning to the service ($\beta = -0.014$ and -0.007 respectively, both with $p < 0.01$), routineness again moderates these negative effects (interactions $\beta = 0.014, p < 0.01$ and $\beta = 0.005, p < 0.05$ respectively), suggesting that the stickiness of routines makes these customers more resilient to these negative aspects of service. Taken together, these between-customers results suggest that cultivating routine customers may be valuable for the firm in myriad ways.

Turning to trip routineness — that is, the within-customers results — we again find positive and significant effects of routineness ($\beta = 0.051$ for accepting proposals, and $\beta = 0.020$ for re-

Table 5: Modeling Accepting a Proposal and Request Again in 7 days as a function of Routineness

Customer Routineness measured as predicted routineness at the end of the calibration period (week 38). All models include a user-level random effect and control for the properties of the focal ride. Models 1-2 analyze data from the holdout period (weeks 39-48); models 3-4 analyze data from the calibration period (weeks 1-38). The full set of control variables can be found in Table A-4. The variables above the horizontal line are characteristics of a proposal; below the line are variables associated with only realized trips. R^2 values are projected R^2, meaning the variance explained beyond the customer effects. The predictors in both models were standardized to improve readability.

	Customer Routineness (Between-customers)		*Trip Routineness (Within-customers)*	
	Accept Proposal	Request Again	Accept Proposal	Request Again
	(1)	(2)	(3)	(4)
# Requests (Week 38)	-0.019	0.089^{***}		
Routineness	0.077^{***}	0.052^{***}	0.051^{***}	0.020^{***}
Price	-0.031^{***}	-0.014^{***}	-0.071^{***}	-0.022^{***}
Driver ETA	-0.050^{***}	-0.0004	-0.089^{***}	0.001
Speed	0.083^{***}	-0.010	0.048^{***}	-0.006
Pickup Walking Dist.	-0.041^{***}	-0.002	-0.021^{***}	0.001
# Passengers	-0.003	0.006	0.009^{***}	0.012^{***}
Routineness x Price	-0.005	0.014^{***}	0.018^{***}	0.016^{***}
Routineness x Driver ETA	0.011^{***}	0.001	0.003	-0.0001
Routineness x Speed	0.142^{***}	-0.021	0.086^{***}	-0.013
Routineness x Pickup Walking Dist.	-0.009^{***}	0.0003	0.002	-0.001
Routineness x # Passengers	-0.0004	0.003	-0.003	0.004^{*}
Pickup Delay		-0.007^{***}		-0.0002
Dropoff Delay		-0.002		-0.002^{*}
Dropoff Walking Dist.		-0.003		-0.002^{*}
# On-board (Pickup)		-0.004		-0.003^{*}
# On-board (Dropoff)		0.001		0.001
Max On-board		0.003		-0.002
Routineness x Pickup Delay		0.005^{**}		0.001
Routineness x Dropoff Delay		0.001		0.001
Routineness x Dropoff Walking Dist.		0.001		0.001
Routineness x # On-board (Pickup)		0.003		0.002
Routineness x # On-board (Dropoff)		-0.0002		0.00003
Routineness x Max On-board		-0.002		-0.001
Customer Effects	Random	Random	Fixed	Fixed
Other Controls	Yes	Yes	Yes	Yes
Observations	38,166	14,704	113,042	73,630
R^2	0.051	0.093	0.070	0.015

$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

questing again, both with $p < 0.01$). These main effects suggest that, for trips that are part of a customer's routine, the customer is more likely to accept the proposal, and more likely to request again within a week of the completed trip. We also see that the routineness of a trip appears to moderate the negative effect of price: while customers in general are less likely to accept higher-priced proposals ($\beta = -0.071$, $p < 0.01$) and to ride again ($\beta = -0.022$, $p < 0.01$) after taking a higher-priced ride, these effects appear to be dampened when that ride is part of their routine (interactions $\beta = 0.018$ and 0.016 respectively, both $p < 0.01$). There is also a positive interaction between the routineness of a trip and the speed of the proposal: when a customer requests a routine trip, they are more likely to accept the proposal if the speed is high (interaction $\beta = 0.086$, $p < 0.01$). Intuitively, customers are familiar with trips in their routine, and thus more sensitive to the details of those trips. This pattern of effects suggests ways that the firm can explore optimizing service around customers' routines, by, for example, offering faster service at premium prices for customers during their most routine times.

6.3. Segmentation by Routine Types

Recall that a key benefit of our framework is that it not only yields an overall metric of routineness (of a ride, or a customer, as we leveraged in the previous analyses), but also enables us to uncover common routines in the data. In our application, we found there are 7 common routine types, which we summarized in our section on model results. Given these different routine types, we now consider whether customers with different routines differ in other significant ways. We find that not only do these routine types differ in when they typically take rides, but also in many other behaviors and typical ride characteristics. For instance, the more casual routine types (i.e., "Work Hard, Play Hard" and "Nights and Weekends") tend to take longer and more expensive trips (higher ETA). Commuters, on the other hand, tend to take cheaper, solo trips.[29] More interestingly, though, we also find that users with different routines systematically differ in their value to the company. In Figure 12, for each routine type, we show means and standard errors of that type's: (1) proposal acceptance rate; (2) future value, as measured by the total amount spent during our holdout period; (3) proportion of (holdout) rides that came from non-routine usage, as estimated by our model; and (4) number of requests made during the holdout period. We find quite striking differences across the routines. For example, the casual clusters had a much lower probability of accepting rides and generated significantly lower value during our holdout period. In contrast,

[29] In Web Appendix I, we include a figure that shows how six key behaviors differ across our routine types.

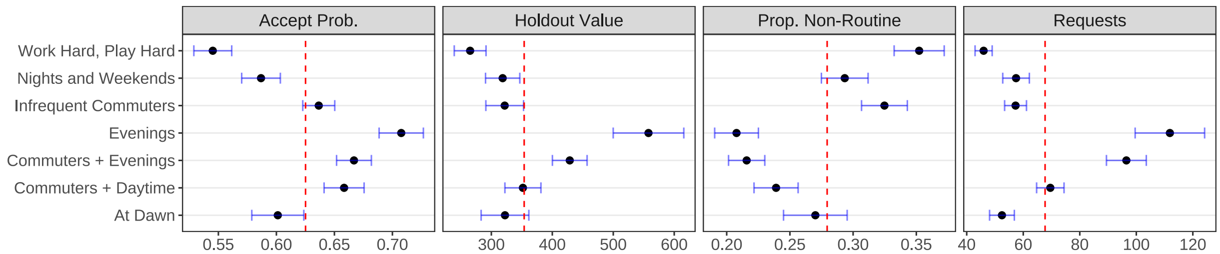


Figure 12: User Behaviors, by Routine Type

The average value of 4 behavior variables, by routine type. Holdout value is the total amount spent over the holdout period. Error bars represent standard errors, and the vertical line represents the overall mean.

the "Evenings" routine appears to be the most valuable: these customers had a higher probability of accepting rides, took more rides, and spent more money during that same period. Interestingly, a higher share of their overall usage was also attributed to their routine. Commuting clusters also appear more valuable, especially those that also incorporate evening usage. In sum, these results suggest that not only is there significant heterogeneity in routine types, but that these routine types are associated with substantially different behaviors, suggesting a role for routines to play in segmenting and targeting customers.

6.4. Additional Analyses: Who and What

Until now, our analyses have proceeded by first identifying routines in terms of when customers interact with the firm, then linking those temporal routines to relevant outcomes. This approach raises two questions: first, can we predict *who* tends to develop routines? And second, beyond timing, is there any value in considering routines in terms of *what* customers do during those interactions? In our context, ridesharing, the "what" of interest is where customers are traveling: customers may always request a ride at the same time, but may go to the same place or different places.

To answer the "who?" question, we examined whether any proposal characteristics, averaged over the calibration period, are predictive of high customer routineness at the end of the training period (week 38). We find several suggestive patterns: first, routine customers tend to have taken lower-priced rides, with fewer other passengers, and shorter walking distances. These results may be indicative of causality, whereby riders develop routines because they are given cheap, convenient trips, or selection, wherein riders with routines happen to take rides during lower-priced, high-supply times. Interestingly, these rides also tended to be longer rides, with higher driver

ETAs, which is consistent with travel during peak times, and the presence of many commuters in our data. These results are described in more detail in Web Appendix J.

To answer the "what?" question, we used trip location data to derive two metrics of location consistency, which capture how often each rider travels between the same locations. We then analyzed the relationship between location consistency, (temporal) routineness, and customer value. We find that location consistency is neither particularly predictive of (timing) routineness, nor predictive of customer value after controlling for routineness. These results suggest that, at least in the context of ridesharing, understanding consistencies in when someone uses the service is more important than understanding consistencies in where they are going. These results are described in more detail in Web Appendix K.

7. Discussion

Summary and Contribution Our work makes two primary contributions: first, from a methodological point of view, to the best of our knowledge, this is the first paper to model customer-level routines. To do so, we leverage a Bayesian nonparametric Gaussian process with a unique kernel structure aimed at estimating temporal routines, nested within a Poisson process. This model can flexibly capture varying routines across customers with high accuracy. Additionally, it yields a customer-level decomposition of usage into a part that is routine and a part that is random, allowing us to quantify the degree of a customer's routineness. Substantively, we apply the model to data from Via, a ridesharing company, and show that we can capture managerially interesting routines. We show that our model-based routineness metric is strongly predictive of customer value, insofar as it is a positive and significant predictor of both future usage and retention. Moreover, this effect is robust, even over longer time horizons, and after controlling for the level of usage, other typical CRM controls, and extant transaction timing metrics. Said differently, this result is noteworthy because it suggests that the temporal shape of usage matters: highly structured usage is more valuable than random usage. While we apply our model in the context of ridesharing, the model we propose is general, and can be applied to usage or purchase data in many business settings.

Beyond our focus on the relationship between routineness and customer value, we also present results that both validate routineness as a construct and establish its more wide-ranging importance in customer management. We show that routine customers are better customers in ways that stretch beyond just lifetime value: they appear to be generally less price sensitive, and more robust to some types of service disruptions. Our results suggest that firms that understand their customers' routines can optimize the provision of services around those routines. Conceptually, we differentiate routineness from constructs like clumpiness and regularity. Finally, we show that routines represent an important source of heterogeneity which can be useful for segmentation and targeting.

Limitations and Future Research We view our work as an initial foray into the topic of modeling and measuring customer routines, and establishing their importance for customer management. As such, there are several limitations of our work, which represent promising directions for future research, to expand both our modeling capabilities, and our understanding regarding when and why routines matter.

From a methodological perspective, scalability of our framework is a limitation. While the runtime of the framework is feasible, estimating our model in a fully Bayesian fashion on a large sample of customers can be costly, as the runtime is superlinear in the sample size (see the simulations in Web Appendix A). In practice, our partner company implements our model on a large sample of data. Our simulation results suggest estimating the model on a sample of consumers is a reasonable solution: the quality of insights from small batches is equivalent to larger data sizes. Thus, running the model in parallel with 10 compute nodes for 1 million customers using 20 weeks of data would take approximately 14 hours, which we argue is quite feasible for most companies with established data science tools. Alternatively, future research may examine new avenues for improving scalability, including variational Bayesian methods for inference (Hoffman, Blei, Wang, and Paisley 2013), or stochastic gradient HMC methods (Dang, Quiroz, Kohn, Minh-Ngoc, and Villani 2019). Our methodology is also limited in that it does not incorporate an explicit latent churn process. While we have provided some evidence that the framework can detect customer inactivity, our model performs best when the purchasing process is relatively stationary.

From a substantive perspective, an area we leave unexplored is the emergence of routines: while we show suggestive results about which customers and trips are more likely to be routine, these patterns are merely suggestive. Our analysis is not causal, and thus cannot establish whether these are indeed drivers of routines. However, given our findings that routine customers are better customers, cultivating routines may be of key interest to companies, and understanding how to do so promises a fruitful area for future research. Likewise, while we show suggestive evidence that routine customers are less price sensitive, price endogeneity is an important issue that we cannot fully resolve with our data. Price experiments should ideally be run to better understand how price sensitivity varies by routineness. We leave a more complete understanding of the relationship between price sensitivity and routines to future studies.

There are several other limitations of our analysis. For instance, our main model captures requests, not actual rides taken. Whether a rider takes a ride may, in part, be driven by supply-side factors like availability of a ride. In turn, whether the service was able to meet a rider's demands in the past may affect how routinely the customer requests rides in the present, leading to perfectly routine behaviors appearing less routine in our model. Another limiting factor in our analysis is our focus on temporal routines. While we provide some evidence that consistency in terms of what customers do is less important, there may still be value in jointly modeling "when" and "what." We leave building such a joint model for future research. Relatedly, because our model focuses just on timing, we cannot differentiate between different subroutines: drawing on an example from the introduction of the paper, if a customer had a routine going to work, and another routine going to Yoga class, our model results would blend these into just a single routine. Finally, our understanding of routines more generally is limited by the fact that we only observe ride-share usage by a single company: more comprehensive panels featuring more alternatives may shed additional light on customer routines, and how they drive consumer choice. We hope these limitations spur additional study of customer routines.

References

Aarts, Henk and Ap Dijksterhuis (2000), "Habits as Knowledge Structures: Automaticity in Goal-directed Behavior," *Journal of Personality and Social Psychology*, 78 (1), 53.

Ascarza, Eva and Bruce G. S. Hardie (2013), "A Joint Model of Usage and Churn in Contractual Settings," *Marketing Science*, 32 (4), 570–590.

Ascarza, Eva, Raghuram Iyengar, and Martin Schleicher (2016), "The Perils of Proactive Churn Prevention Using Plan Recommendations: Evidence from a Field Experiment," *Journal of Marketing Research*, 53 (1), 46–60.

Ascarza, Eva, Scott A Neslin, Oded Netzer, Zachery Anderson, Peter S Fader, Sunil Gupta, Bruce GS Hardie, Aurélie Lemmens, Barak Libai, David Neal et al. (2018), "In Pursuit of Enhanced Customer Retention Management: Review, Key Issues, and Future Directions," *Customer Needs and Solutions*, 5, 65–81.

Betancourt, Michael "Robust Gaussian Process Modeling," (2020) https://betanalpha.github.io/assets/case_studies/gaussian_processes.html.

Blattberg, Robert, Scott Neslin, and Byung Kim (2008), *Database Marketing: Analyzing and Managing Customers* Springer.

Dang, Khue-Dung, Matias Quiroz, Robert Kohn, Tran Minh-Ngoc, and Mattias Villani (2019), "Hamiltonian Monte Carlo with energy conserving subsampling," *Journal of Machine Learning Research*, 20, 1–31.

Dew, Ryan and Asim Ansari (2018), "Bayesian Nonparametric Customer Base Analysis with Model-based Visualizations," *Marketing Science*, 37 (2), 216–235.

Dew, Ryan, Asim Ansari, and Yang Li (2020), "Modeling Dynamic Heterogeneity Using Gaussian Processes," *Journal of Marketing Research*, 57 (1), 55–77.

Drolet, Aimee and Wendy Wood (2017), "Introduction to Special Issue: The Habit-Driven Consumer," *Journal of the Association for Consumer Research*, 2 (3), 275–278.

Du, Rex Yuxing and Wagner A Kamakura (2012), "Quantitative Trendspotting," *Journal of Marketing Research*, 49 (4), 514–536.

Du, Rex Yuxing, Oded Netzer, David A Schweidel, and Debanjan Mitra (2021), "Capturing Marketing Information to Fuel Growth," *Journal of Marketing*, 85 (1), 163–183.

Dynan, Karen E. (2000), "Habit Formation in Consumer Preferences: Evidence from Panel Data," *American Economic Review*, 90 (3), 391–406.

Ehrenberg, Andrew S. C. and Gerald J. Goodhardt (1968), "A Comparison of American and British Repeat-Buying Habits," *Journal of Marketing Research*, 5 (1), 29–33.

Fader, Peter S., Bruce G.S. Hardie, and Ka Lok Lee (2005), "RFM and CLV: Using Iso-value Curves for Customer Base Analysis," *Journal of Marketing Research*, 42 (4), 415–430.

Fuhrer, Jeffrey C. (2000), "Habit Formation in Consumption and its Implications for Monetary-policy Models," *American Economic Review*, 90 (3), 367–390.

Gopalan, Prem, Jake M Hofman, and David M Blei "Scalable Recommendation with Hierarchical Poisson Factorization," "UAI," pages 326–335 (2015).

Guadagni, Peter M and John DC Little (1983), "A Logit Model of Brand Choice Calibrated on Scanner Data," *Marketing Science*, 2 (3), 203–238.

Ho, Teck-Hua, Young-Hoon Park, and Yong-Pin Zhou (2006), "Incorporating Satisfaction into Customer Value Analysis: Optimal investment in lifetime value," *Marketing Science*, 25 (3), 260–277.

Hoffman, Matthew D., David M. Blei, Chong Wang, and John Paisley (2013), "Stochastic variational inference," *Journal of Machine Learning Research*, 14, 1303–1347.

James, William (1890), *The Principles of Psychology* Holt and Macmillan.

Kim, Jin Gyo, Ulrich Menzefricke, and Fred M. Feinberg (2005), "Modeling Parametric Evolution in a Random Utility Framework," *Journal of Business & Economic Statistics*, 23 (3), 282–294.

Kim, Mingyung, Eric T Bradlow, and Raghuram Iyengar (2022), "Selecting Data Granularity and Model Specification Using the Scaled Power Likelihood with Multiple Weights," *Marketing Science*, 41 (4), 848–866.

Kingman, John Frank Charles (1992), *Poisson Processes*, Vol. 3. Clarendon Press.

Kumar, Vineet and Kannan Srinivasan (2015), "Commentary on Predicting Customer Value Using Clumpiness," *Marketing Science*, 34 (2), 209–213.

Lewandowski, Daniel, Dorota Kurowicka, and Harry Joe (2009), "Generating Random Correlation Matrices Based on Vines and Extended Onion Method," *Journal of Multivariate Analysis*, 100 (9), 1989–2001.

Loper, Jackson, David Blei, John P Cunningham, and Liam Paninski (2021), "A General Linear-time Inference Method for Gaussian Processes on One Dimension," *The Journal of Machine Learning Research*, 22 (1), 10580–10615.

Neal, David T., Wendy Wood, Jennifer S. Labrecque, and Phillippa Lally (2012), "How Do Habits Guide Behavior? Perceived and Actual Triggers of Habits in Daily Life," *Journal of Experimental Social Psychology*, 48 (2), 492–498.

Neslin, Scott A, Sunil Gupta, Wagner Kamakura, Junxiang Lu, and Charlotte H Mason (2006), "Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models," *Journal of Marketing Research*, 43 (2), 204–211.

Neslin, Scott A, Gail Ayala Taylor, Kimberly D Grantham, and Kimberly R McNeil (2013), "Overcoming the "Recency Trap" in Customer Relationship Management," *Journal of the Academy of Marketing Science*, 41, 320–337.

Ouellette, Judith A. and Wendy Wood (1998), "Habit and Intention in Everyday Life: The Multiple Processes by Which Past Behavior Predicts Future Behavior," *Psychological Bulletin*, 124 (1), 54.

Platzer, Michael and Thomas Reutterer (2016), "Ticking Away the Moments: Timing Regularity Helps to Better Predict Customer Activity," *Marketing Science*, 35 (5), 779–799.

Sarkar, Mainak and Arnaud De Bruyn (2021), "LSTM Response Models for Direct Marketing Analytics: Replacing Feature Engineering with Deep Learning," *Journal of Interactive Marketing*, 53 (1), 80–95.

Schmittlein, David C., Donald G. Morrison, and Richard Colombo (1987), "Counting Your Customers: Who Are They and What Will They Do Next?," *Management Science*, 33 (1), 1–24.

Shah, Denish, V. Kumar, and Kihyun Hannah Kim (2014), "Managing Customer Profits: The Power of Habits," *Journal of Marketing Research*, 51 (6), 726–741.

Snelson, Edward and Zoubin Ghahramani "Sparse Gaussian Processes Using Pseudo-inputs," Y. Weiss, B. Schölkopf, and J. Platt, editors, "Advances in Neural Information Processing Systems," Vol. 18., pages 1257–1264, MIT Press (2005).

Tian, Longxiu and Fred Feinberg "Broadening the Horizon: Augmenting One-Shot Field Experiments with Longitudinal Customer Data," (2021) Working paper.

Verplanken, Bas, Ian Walker, Adrian Davis, and Michaela Jurasek (2008), "Context Change and Travel Mode Choice: Combining the Habit Discontinuity and Self-activation Hypotheses," *Journal of Environmental Psychology*, 28 (2), 121–127.

White, Katherine, Rishad Habib, and David J. Hardisty (2019), "How to SHIFT Consumer Behaviors to be More Sustainable: A Literature Review and Guiding Framework," *Journal of Marketing*, 83 (3), 22–49.

Williams, Christopher KI and Carl Edward Rasmussen (2006), *Gaussian Processes for Machine Learning*, Vol. 2. MIT press Cambridge, MA.

Wilson, Andrew and Ryan Adams "Gaussian Process Kernels for Pattern Discovery and Extrapolation," "International conference on machine learning," Vol. 28., pages 1067–1075, PMLR (2013).

Wood, Wendy, Jeffrey M. Quinn, and Deborah A. Kashy (2002), "Habits in Everyday Life: Thought, Emotion, and Action," *Journal of Personality and Social Psychology*, 83 (6), 1281.

Zhang, Yao, Eric T. Bradlow, and Dylan S. Small (2015), "Predicting Customer Value Using clumpiness: From RFM to RFMC," *Marketing Science*, 34 (2), 195–208.

Web Appendix

Detecting Routines: Applications to Ridesharing CRM

Ryan Dew, Eva Ascarza, Oded Netzer, Nachum Sicherman

June 28, 2023

These materials have been supplied by the authors to aid in the understanding of their paper. The AMA is sharing these materials at the request of the authors.

Contents

A. Additional Details of Simulations

We conducted a number of simulations to evaluate the performance of the model under different data conditions. In the first set of simulations, we examined parameter recovery and scalability under different data settings, assuming the data was generated by our model. In the second set of simulations, we added a churn process to the data generating process, to examine how the model performed in the presence of customer churn. In both studies, the data (without churn) were generated by the following process:

- $\mu(j)$ was drawn from a GP with a constant mean of 0, and the day-hour kernel with lengthscale 5, amplitude 2, and LKJ scaling parameter 2.
- $\eta_i(j)$ was drawn (hierarchically) from a GP with the same parameters as $\mu(j)$
- $\gamma_i(j)$ was drawn (hierarchically) from a GP with a common mean of -8, and an exponential kernel with amplitude 1.5, and lengthscale 5, implying somewhat less smoothness than that found in our empirical setting.
- $\alpha_i(j)$ was drawn (hierarchically) from a GP with the same parameters as $\gamma_i(j)$, but with a lengthscale of 3.

The key features of this simulation are: (1) the means of -8 for $\gamma_i(j)$ and $\alpha_i(j)$ imply a rate of usage similar to that observed in our true data; and (2) the amplitude values imply a rather high amount of variation across people, such that some customers will be very random and not routine, while others will be routine but not random (and some will be in between).

A.1. Simulation Study 1: Scalability and Parameter Recovery

To assess both parameter recovery and scalability, we varied the number of customers from 100 to 5,000 (assuming 20 weeks of data), and we varied the number of weeks from 10 to 320 (assuming 200 customers). In each case, we then estimated the model, and measured parameter recovery and runtime.

Parameter Recovery Across all the simulations, we found that the quality of parameter recovery was nearly identical, regardless of the number of customers, and the number of weeks. Thus, in this section, we focus on the results from the simulation with the least data: 100 customers, 20 weeks. As reported in the main body of the manuscript, parameter recovery for the focal model parameters is excellent: for population-level parameters like $\mu(j)$, the correlation between true and estimated values is 0.96. For the individual-level parameters, that correlation is slightly lower but still quite high, with a correlation of 0.74 for $\eta_i(j)$, and 0.80 for $\alpha_i(w)$.

While these correlation is quite high, especially for an individual-level parameter, they are imperfect. Much of this imperfection stems from a feature of the model: if most of a customer's transactions were generated from a routine, then estimating the parameters of that person's random process will be difficult to capture, and vice versa. In general, there are many (very negative) values of the parameters that can generate zero transactions. Hence, beyond evaluating the correlation between simulated and estimated values, we explore two additional aspects of parameter recovery: first, for people who actually had routine requests, is the estimated routine correct? And second, is the estimated number of routine requests recovered correctly? Figures A-1 and A-2 show both of these comparisons, respectively. Specifically, in Figure A-1, we show that the correlation between true and estimated values of $\eta_i(j)$ approaches 1 as the (log) number of routine requests grows, becoming very accurate even for a modest number of routine requests (e.g., for more than 20 total routine requests, or, on the log scale, for values above $\log 20 \approx 3$). In Figure A-2, we see that the total number of routine requests over the calibration period is very accurately recovered. Together, these results give strong evidence that our model is statistically identified, even for small amounts of data.

Scalability Figure A-3 shows the runtime (in seconds) of the model as we increase the number of customers and the number of time periods. We see that the scaling of the model is superlinear across both dimensions, though the number of customers is the biggest bottleneck. The superlinear scaling is interesting: it suggests that running minibatches of data is much more efficient than running all the data at once. For example, going from 100 customers to 200 customers increases the computation time by approximately 10 minutes, whereas going from 100 customers to 1000 customers increases the computation time by approximately 3.5 hours ($\sim$200 minutes).

As described above, the model can recover the true data generating process accurately, even with very few customers. Practically, this suggests that, if the goal is to compute routineness and

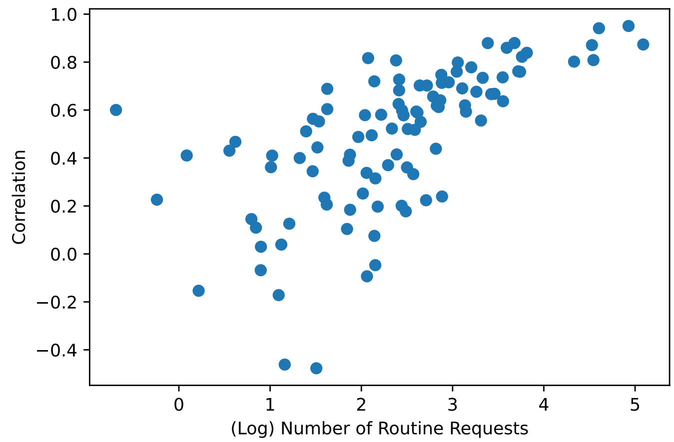


Figure A-1: Correlation Between True and Estimated $\eta_i(j)$
We plot the correlation between the true and estimate values of $\eta_i(j)$, as a function of the logged number of (true) routine requests the person made.

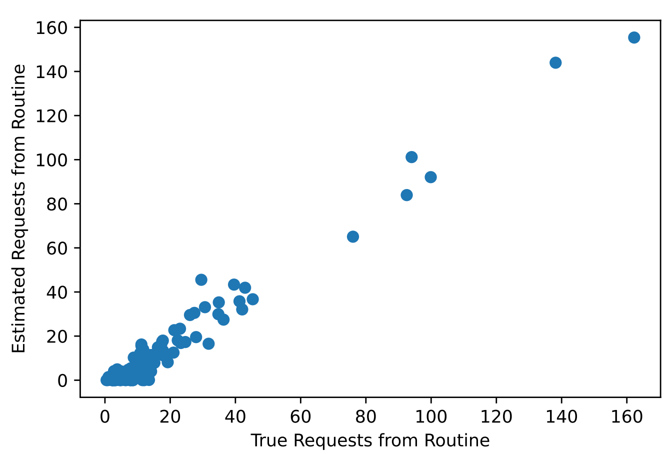


Figure A-2: Correlation Between True and Estimated Total Routineness ($\sum_w E_{iw}^{\text{Routine}}$)
We plot the true versus estimated number of requests coming, in total, from a customer's routine.

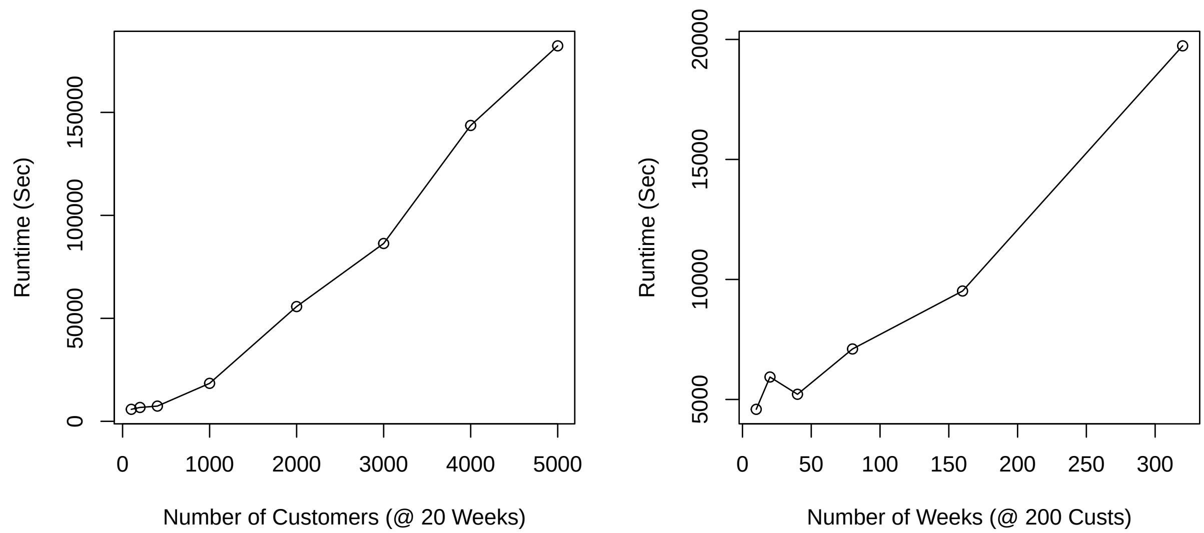


Figure A-3: Scalability Results

Runtime in seconds for estimating the routines model on simulated data.

estimate each customer's routine pattern, leveraging all customers together in one hierarchical model is not essential. Rather, the analyst can split the data into batches that include the full history of different groups of customers, and obtain the desired results in an effective manner. For example, estimating the routines of 1,000,000 customers over 20 weeks of data following this estimation strategy, assuming 10 compute nodes in parallel, would take approximately 14 hours, which we argue is quite feasible for most companies with established data science tools.

Alternatively, relatively recent advances in Bayesian computation, including stochastic gradient HMC (Dang, Quiroz, Kohn, Minh-Ngoc, and Villani 2019) or, in the approximate case, stochastic variational inference (Hoffman, Blei, Wang, and Paisley 2013), could also likely remedy this bottleneck, given those methods can compute computationally demanding parts of the inference algorithm, like the gradient of the log posterior, using only mini-batches of data (in this case, customers). The degree to which those approaches might reduce the total compute time would depend on how much of the total compute time is dominated by calculating the gradient.

A.2. Simulation Study 2: Churn and Forecasting

In our second study we simulated data for 200 customers, for 142 weeks in total. This number of weeks was set by taking the number of calibration weeks in our true data, 38, and assuming we want to forecast 2 years into the future, an additional 104 weeks. Unlike in the previous study, where the data generating process is the same as the model, in this case, we added a churn process that is *not* part of the model, to understand how different rates of churn may affect the reliability of the model and its ability to forecast future spending. We ran three simulations, varying the propensity at which customers churn from the service. Specifically, we assumed that, each week, each customer has a certain probability of churning, which we set to be either 0 (i.e., the same as in the previous study), 0.004 (a moderate rate), or 0.02 (a high rate). If a customer churns, it means their number of transactions is zero from that point forward. We estimate the model using 38 weeks of calibration, and test forecasting over 104 weeks.

Parameter Recovery First, although it was not the focus of this study, we did examine parameter recovery just as in Study 1. As described in the main body of the paper, if customers churn, this can be accommodated in our framework by setting $\alpha_i(w)$ or $\gamma_i(w)$ to low values. Hence, we do not expect these parameters to be "recovered," given our addition of a churn process. However, importantly, the day-hour rate parameters should be relatively unaffected. Indeed, we find this to be true: across all three simulations, the correlation between true and estimated parameters was around 0.95 $\mu(j)$, and (on average) 0.75 for $\eta_i(j)$.

Fit and Forecasting Next, we look at how the rate of churn affects the ability of the model to explain and predict the data. In-sample, across all three simulations, the model performs well. Again, the model can address churn by setting the weekly scaling terms to be negative. However, out-of-sample performance depends on the overall propensity to churn. In the case of no churn, the out-of-sample performance is decent, and moreover, does not deteriorate over time. This is as predicted, given the model assumes stationary transaction processes; the forecasted GPs for the weekly scaling terms revert to their means over time, predicting future spending will occur at the historical average rate. Interestingly, when churn is high, the model also performs well: in this case, many people churn early, leading the weekly scaling terms to be very low throughout

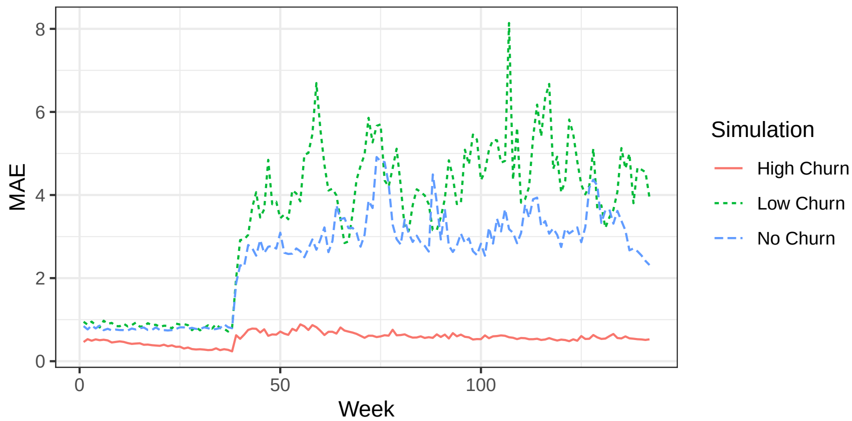


Figure A-4: MAE of Three Simulations with Varying Degrees of Churn
The mean absolute error over the entire 142 weeks (38 calibration, 104 holdout), for the three simulations.

the calibration period, resulting in essentially a forecast of zero usage. In the middle case, with a modest rate of churn, the model's out-of-sample performance somewhat deteriorates: in this case, as people churn out-of-sample, the model's forecast of stationary usage rates cannot match reality, leading to larger errors in forecasting usage rates. We illustrate the week-over-week MAE of the three models together in Figure A-4.

B. Model with Covariates

The model presented in the main manuscript is general as it can be applied to a wide range of contexts and only requires transactional data, which are easily available to essentially any analyst. In some cases, the analyst might also have access to information such as firm interventions or other shocks in the service, and might be interested in exploring to what degree these changes relate to, and possibly drive, routine and non-routine usage. As a general model of usage timing, our framework is fairly flexible, and can be extended to incorporate covariates, in a number of ways.

B.1. Model Specifications

Let us denote the available covariates as x_{it}. These covariates could be at the weekly level (x_{iw}), or at the more granular week-day-hour level (i.e., x_{iwj}), or just at the customer level (i.e., x_i). For example, x_{it} could be an indicator capturing a promotion run by the company during week w, or a lagged variable capturing the quality of service the customer received during their last trip, or has received, on average, until that point.

First, recall the rate of our overall usage request process:

$$\lambda_{it} = \exp(\gamma_{iw} + \eta_{ij}) + \exp(\alpha_{iw} + \mu_j). \tag{A-1}$$

We can incorporate covariates in both terms, linearly or nonlinearly. For now, we focus on the linear case, and on incorporating covariates into the routine term. Intuitively, this specification corresponds to "covariate-driven routines": covariates may affect the degree to which customers adopt routines. To simplify the subsequent discussion, we also assume that the covariates are available at the weekly level, x_{iw}, although the same modeling approach can be taken for any type of covariate. With this assumption, the covariate-driven routines model is given by:

$$\lambda_{it} = \exp(\gamma_{iw} + \beta_R' x_{iw} + \eta_{ij}) + \exp(\alpha_{iw} + \mu_j). \tag{A-2}$$

In this covariates-driven routine model, the degree to which the day-hour rate η_{ij} governs behavior during week w is determined by both the weekly scaling parameter, γ_{iw}, and by the value of the covariates. In this sense, γ_{iw} captures the "residual" routine variation, week-over-week, modulo the covariate effect $\beta_R' x_{iw}$. While this specification is intuitive, it also changes how we think about smoothness and γ_{iw}: in this specification, γ_{iw} may be highly non-smooth, given it can only be interpreted relative to the covariate effect.

B.2. Caveats and Implementation

We now discuss important caveats to using the covariates model as well as details of its implementation, which might be helpful to analysts hoping to use this model in practice. First, as mentioned, the covariates extension of our basic model can be estimated with covariates that vary along any dimension. An alternative formulation could specify that covariates drive the degree to which random needs arise, rather than routines. Given our focus in this paper on understanding routines, we found the covariate-driven routines specification more natural.

Another caveat is that estimating the effects of covariates requires more data per-person: to achieve satisfactory convergence on real data, we had to restrict the data to customers who made at least five requests during the calibration period. While is not particularly onerous, it is more limiting than the no-covariates model presented in the main body of the paper, which typically does not have convergence issues, even in the presence of infrequent users.

Finally, forecasting with the covariates model is non-trivial. All of the estimated parameters are estimated *modulo* to the covariates. Given many covariates of interest will be dynamic, the analyst is thus forced to make forecasts for the value of the covariates, too. Practical strategies for doing so may include using simple lagged or average values, but these assumptions may be far from reality.

B.3. Application: Past Service Quality and Routineness

We estimated the covariate-driven routines model on the same data described in the main body of the paper. For the covariates, we used the lagged, standardized values of several *proposal-level* variables, in particular, the **cost per mile**, **expected wait time**, **trip time per mile**, and **walking distance** of the previous proposal (or, the average of all proposals in the previous session). As noted previously, to achieve satisfactory convergence, we limited the data to customers who had at least five sessions during the calibration period, resulting in 1,841 total customers. Given the use of lagged variables, we also only included trips for which we could observe the previous value

Table A-1: Estimated Covariate Effects

Posterior summary of the estimated covariate effects. The mean and SD refer to the estimated posterior distribution. HDI refers to highest (posterior) density intervals.

Variable	Mean	SD	HDI 2.5%	HDI 97.5%
Cost Per Mile	0.010	0.013	-0.014	0.033
Expected Wait Time	-0.049	0.012	-0.075	-0.028
Trip Time Per Mile	-0.004	0.014	-0.032	0.020
Walking Distance	-0.007	0.010	-0.025	0.012

of the covariates of interest. The benefit of using proposal-level variables is that they are recorded for (nearly) every session.[1]

First, the population-level parameters were estimated to be quite similar to the main model. This finding, by and large, makes sense: covariates should not alter, for instance, that behavior on weekdays tends to be more similar to behavior on other weekdays, and likewise for weekends. Nor does it seem to alter the general day-hour patterns and types of routines recovered by the model.

Recall that the new parameter of interest in this model is β_R, the effect of the covariates on the routineness term. We give the estimated values of β_R in Table A-1, specifically reporting posterior means, standard deviations, and 95% highest (posterior) density intervals (HDI). We see that, while most of the covariates have posterior means in sensible directions, the only covariate whose HDI excludes zero is expected wait time. The negative effect here suggests that higher past wait times are a key predictor of *lower* routineness in the future. This finding is reasonable: the company notes that long wait times are among the biggest factors driving customer satisfaction. Wait time is such a significant variable for Via that they often use wait time as a focal unit of analysis, by, for example, computing “wait time elasticities (of demand)” and thinking of the effects of different potential interventions, like discounts, in terms of “effective wait time.”

[1] Every session, barring occasional technical glitches.

C. Additional Case Studies from the Quasi-Simulation

In this section, we present additional analyses for the 32 synthetic case studies in our quasi-simulation. We present them in three parts: first, we present a set of illustrative cases that highlight the model's ability to meaningfully decompose transactions. Then, we analyze the links between routine, clumpy, and regular behavior. Finally, we analyze data drawn from the Pareto-GGG (Platzer and Reutterer 2016) under our model.

Illustrative Cases Table A-2 describes the set of illustrative cases, in terms of how each simulation was generated, and Figure A-5 shows the model-based decomposition for each case. The figures are interpreted analogously as the "Decomposition" figures in the main manuscript. We corroborate that, in general, the correct insights are well recovered by the model. The only exception occurs in cases where usage is purely random. There, the model may attribute some of that random usage to a routine (e.g., cases 1 and 2). Also of note are cases 6 and 7, where the true data generating process was a mix of two routines: one routine in the first half, and a new routine in the second half. We see that even though the model has no mechanism to learn multiple routines it still classifies these customers as fully routine. However, the routine it learns is a mixture of the two, which is a limitation of our model.

Note that, for all of the simulations in Table A-2, the data generating process was deterministic. These same patterns can be recovered, even in the presence of noise. To illustrate, we included two additional cases, which mimic cases 3 and 4 from Table A-2 (i.e., high and low frequency routines), but where at each routine time, the customer uses the service probabilistically: with probability 0.5, they do not make a request; with probability 0.4, they make one request; with probability 0.095, they make two requests; and with probability 0.005, they make three requests. We plot the full results for these customers in Figures A-6 and A-7. As we can see, the model accurately parses both routines, although there is some fluctuation in the level of routineness, following the stochastic request process. In both cases, the model also accurately recovers the number of request times (5 and 2). For the expected requests, these are now fractional, again following the stochastic request process. In short: even in stochastic settings, the model has no issue meaningfully decomposing behavior.

Table A-2: Illustrative cases.

Descriptions of the simulated customers.

Case	Label	Simulation procedure
1	Random (High)	Customer makes a request at 5 day-hours each week, randomly sampled each week from the empirical distribution
2	Random (Low)	Customer makes a request at 2 day-hours each week, randomly sampled each week from the empirical distribution
3	Routine (High)	Randomly sample 5 day-hours; customer makes a request at these times, every week
4	Routine (Low)	Randomly sample 2 day-hours; customer makes a request at these times, every week
5	Commuter	Customer rides every weekday at 8 AM, and 5 PM
6	Two Routines (High)	For the first 19 weeks, the customer follows a routine, generated as in Case 1; then the customer abruptly shifts to a new routine for the remaining 19 weeks, redrawing the times at which she requests rides
7	Two Routines (Low)	For the first 19 weeks, the customer follows a routine, generated as in Case 2; then the customer abruptly shifts to a new routine for the remaining 19 weeks, redrawing the times at which she requests rides
8	Random then Routine (High)	For the first 19 weeks, the customer follows the Random (High) procedure; for the last 19 weeks, the customer follows the Routine (High) procedure
9	Random then Routine (Low)	For the first 19 weeks, the customer follows the Random (Low) procedure; for the last 19 weeks, the customer follows the Routine (Low) procedure
10	Routine then Random (High)	For the first 19 weeks, the customer follows the Routine (High) procedure; for the last 19 weeks, the customer follows the Random (High) procedure
11	Routine then Random (Low)	For the first 19 weeks, the customer follows the Routine (High) procedure; for the last 19 weeks, the customer follows the Random (High) procedure
12	Random then Dead (High)	For the first 19 weeks, the customer follows the Random (High) procedure, then stops making requests
13	Random then Dead (Low)	For the first 19 weeks, the customer follows the Random (Low) procedure, then stops making requests
14	Routine then Dead (High)	For the first 19 weeks, the customer follows the Routine (High) procedure, then stops making requests
15	Routine then Dead (Low)	For the first 19 weeks, the customer follows the Routine (Low) procedure, then stops making requests

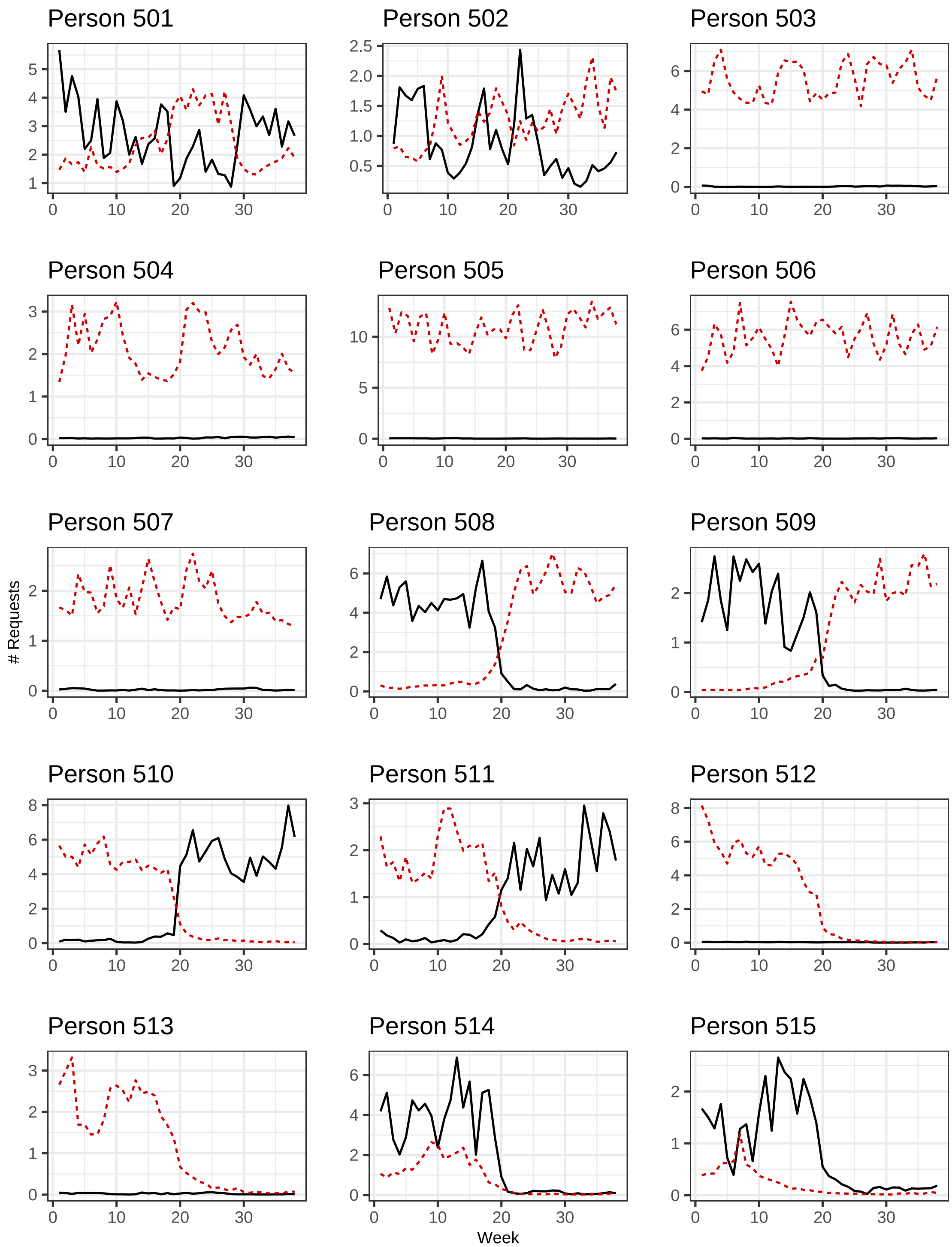


Figure A-5: Full simulation results.
The model-based decomposition for all 15 simulated cases, as described in the main body o f the paper and in Table A-2. The red dashed line is routine usage, while the black solid line is random usage. We see that, by and large, the model can correctly parse the correct data-generating pattern.

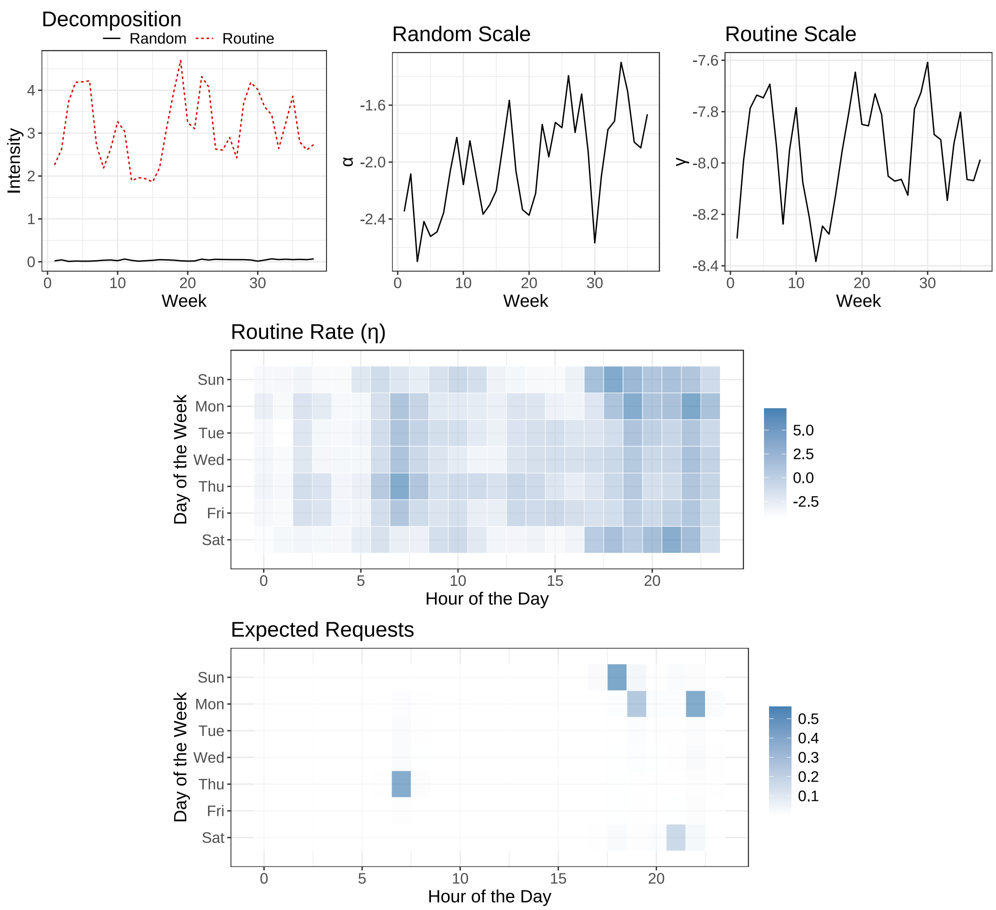


Figure A-6: Noisy version of Case 1

The model-based decomposition and associated parameters for a noisy version of Case 1, i.e., a high frequency routine.

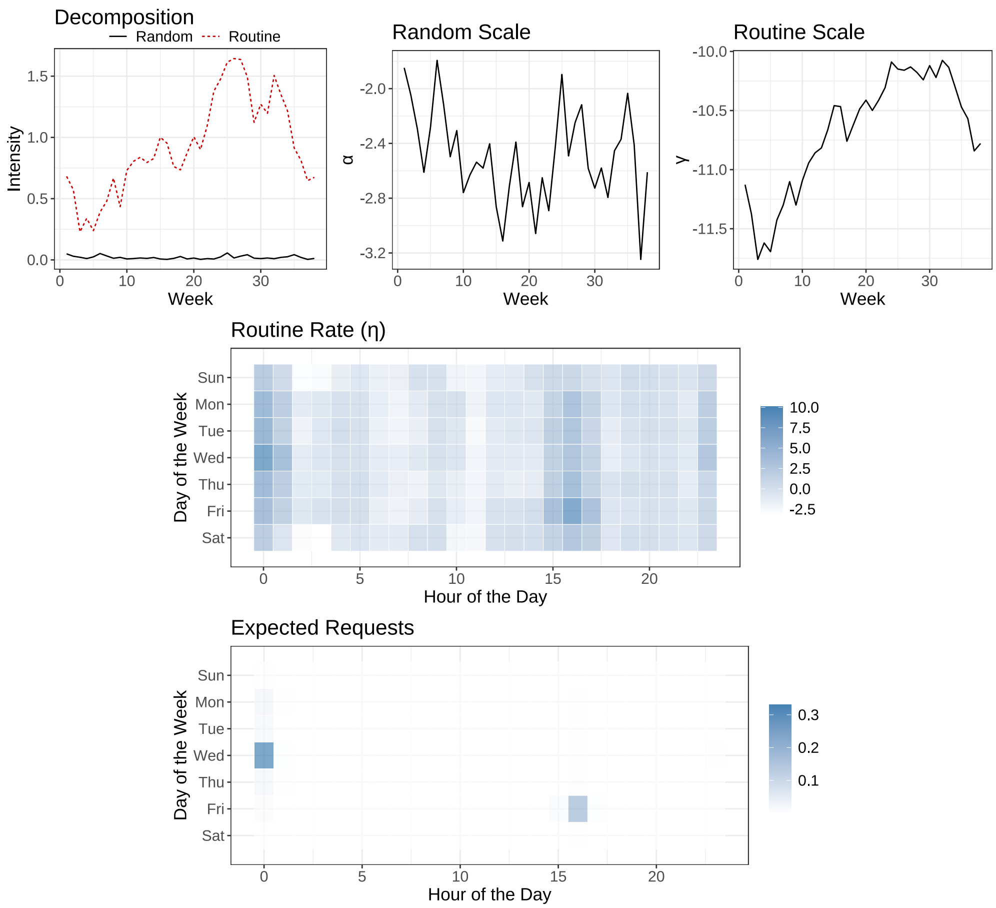


Figure A-7: Noisy version of Case 2

The model-based decomposition and associated parameters for a noisy version of Case 2, i.e., a low frequency routine.

Comparisons to Clumpiness and Regularity In addition to the cases shown previously, we also generated customers with clumpy and regular behavior. We now zero in on several examples of those cases, to illustrate the connections between routines, clumpiness, and regularity. In particular, we first focus on the synthetic cases that inspired Figure **??** (in the main body). As described in the paper, routines can also generate clumpy or regular behavior. In fact, they can also exhibit regular clumps. To illustrate these patterns, we generated several additional customers, featuring clumpy and/or regular behavior. The first is a customer who does not use the service during the week, but always makes requests on the weekend, at 10am, 5pm, and 11pm — this user is high on clumpiness. Then, we generated another user who always transacts at 8am, every day — very regular. To validate that these cases are clumpy and regular, for clumpiness, the first customer has a clumpiness score of 0.17 versus almost zero for the second. In terms of regularity, the first customer always has identical intertransaction times, while the second does not. Figures A-8 and A-9 show our model's estimation of these customers' behavior, respectively. The model identifies that both of these cases have high routineness: the decomposition attributes all of their behavior to a routine. In short, this illustrates that routines can generate both clumpy and regular behavior.[2]

Perhaps more interestingly, we can also generate regularities that are not, exactly routine. Imagine, for instance, a customer who makes a clump of transactions every 40 hours. Now, this is admittedly rather strange transaction behavior, but it is interesting to see how our model decomposes such a pattern. The 168 day-hours in a week are not evenly divisible by 40, so this behavior does not lead to a consistent week-over-week behavior. However, it is, in some sense, equivalent to a probabilistic routine, at all hours that are divisible by 40. We see that is exactly what the model recovers, presented in Figure A-10. Similarly, we can also generate clumpy behavior where the timing of the clumpy behavior is totally random. In this case, the model correctly parses that the behavior is not routine at all, and attributes the majority of usage to the non-routine process.

[2]We have additional case studies that are variations on this theme, including stochastic timing, and lower transaction rates, which have results consistent with these case studies, and which all can be shared upon request.

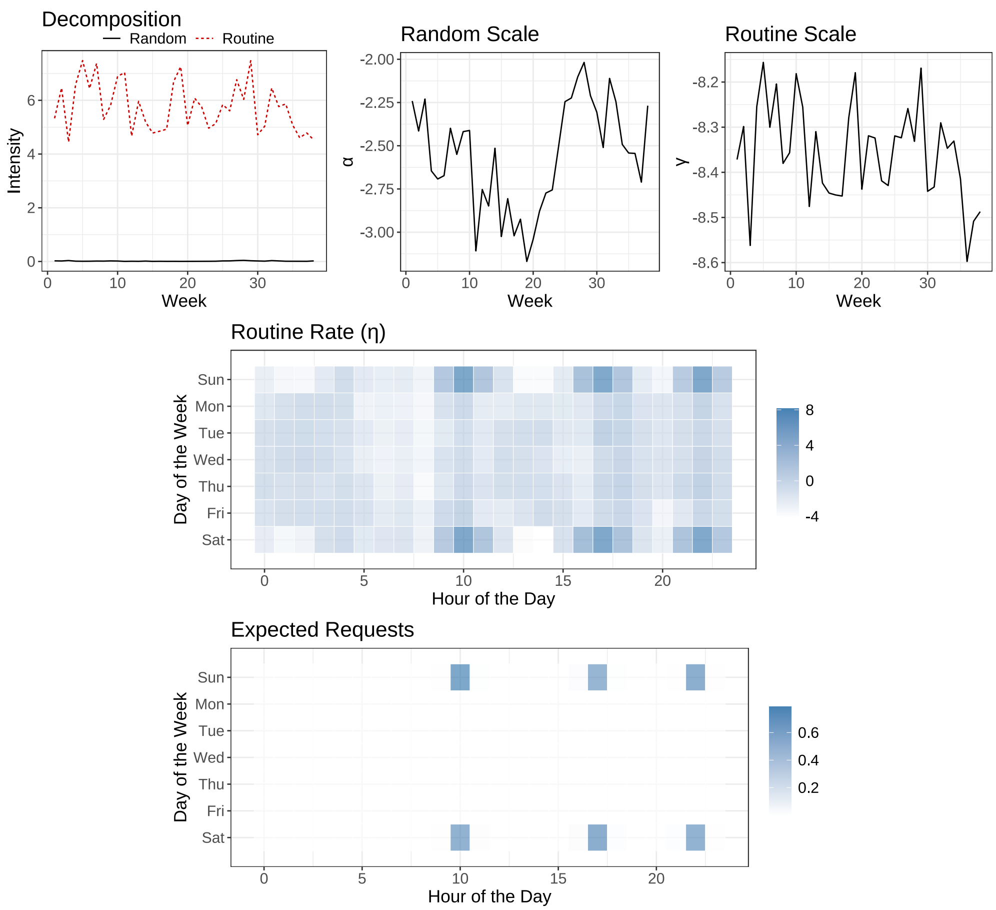


Figure A-8: Routine and Clumpy

The model-based decomposition and associated parameters for a case where behavior is routine and clumpy.

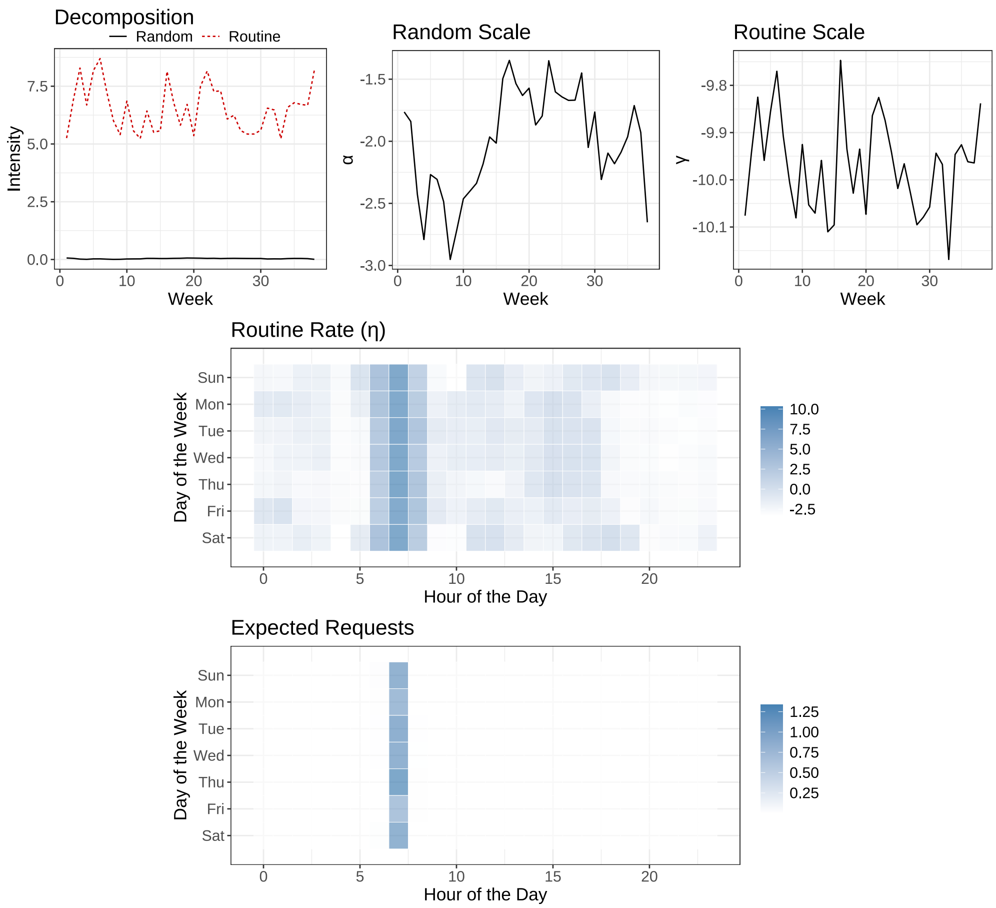


Figure A-9: Routine and Regular

The model-based decomposition and associated parameters for a case where behavior is routine and regular.

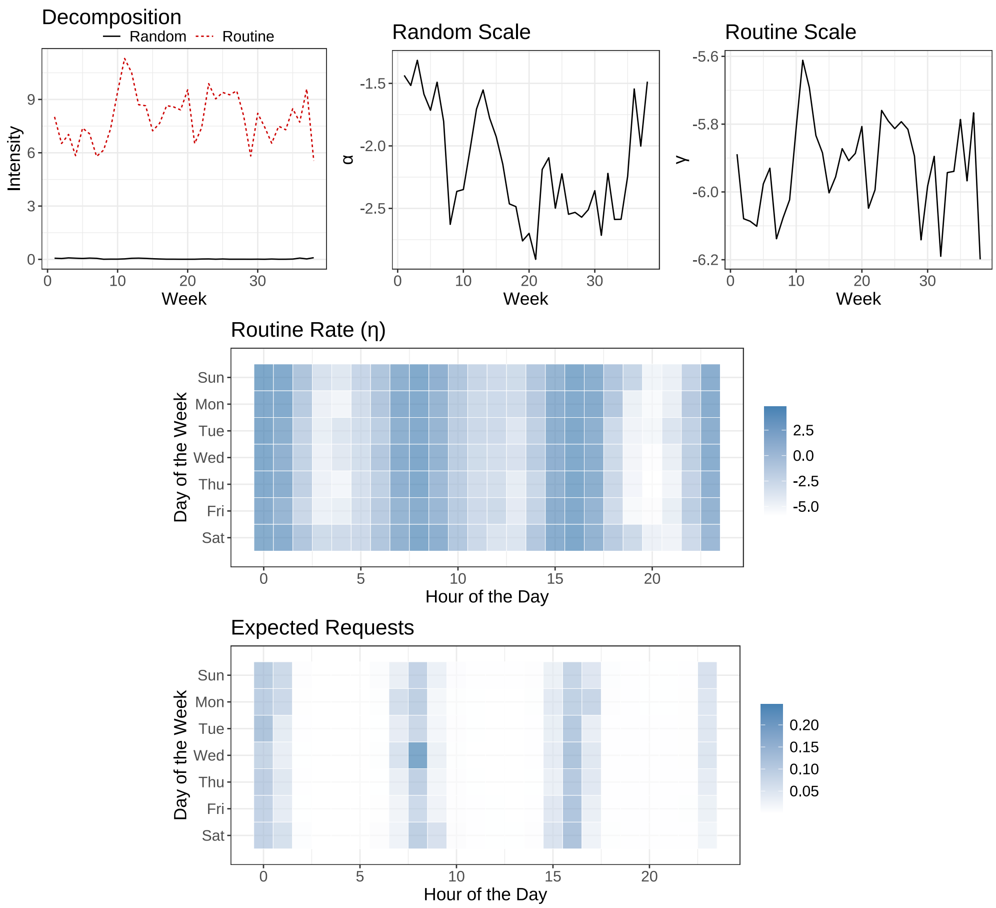


Figure A-10: Clumpy with a 40 hour cycle.
The model-based decomposition and associated parameters for a case where a clump of transactions occurs roughly every 40 hours.

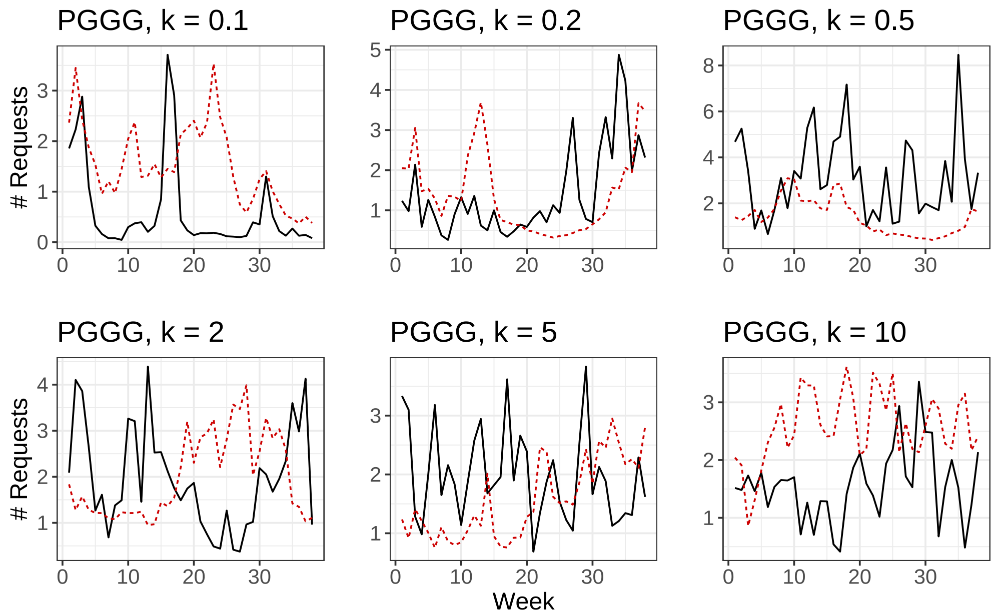


Figure A-11: Six Pareto-GGG Case Studies
The model-based decompositions for six customers generated from the Pareto-GGG model (with no churn). Similar to prior plots, the red dashed line is the routineness, while the black solid line is the non-routine ("random") process.

Pareto-GGG Finally, we generated a set of customers from the Pareto-GGG model with no churn. In essence, this is equivalent to gamma-distributed interarrival times, with individual-specific rates k_i, following the specification given in Platzer and Reutterer (2016). Specifically, we generated six synthetic customers, varying $k_i = 0.1, 0.2, 0.5, 2, 5, 10$, ranging from very clumpy to very regular. Intuitively, we would hypothesize that, for most values of k, the model would classify these consumers as non-routine. Only when k grows very large, and the interarrival times become extremely regular, should we see some routineness emerge. Figure A-11 shows the results of the decomposition, applied to our six synthetic cases. The results are consistent with our hypothesis: as k rises, more usage is attributed to a routine, especially in the case of $k = 10$ which corresponds to very regular transaction patterns, akin to our "every day at 8am" consumer from before. For values lower than $k = 10$, we find that much of the usage is attributed to the random process. Occasionally, we see a bit of routine usage: this can be attributed mostly to stochasticity producing similar weekly transaction times.

D. Details of the LSTM Benchmark

As a benchmark, we trained long-short term memory deep learning models (Goodfellow, Bengio, and Courville 2016; Sarkar and De Bruyn 2021) with the goal of predicting in which day-hours a customer would request a ride. We evaluated two LSTM modeling frameworks: one in which we trained one small LSTM per user, and one in which we trained a single large LSTM to predict across all users. With appropriate tuning of the architectures, we found the small individual-level models performed better, and thus, we focus on that framework below, and include its statistics in the paper. In all cases, the loss function was a binary cross-entropy loss, focused on predicting if there would be a ride request during each of the 168 day-hours in a week. As training data, we used the same 38-week window as the other models. The specific architecture fed a sequence of 3-weeks of data through an LSTM layer with 32 hidden units and tanh activation functions, the output of which was flattened and fed through a dense layer, again with 32 hidden units. The output is a sequence of probabilities over the subsequent week, predicting the likelihood of seeing a ride during each day-hour, for that customer.

To forecast with this architecture, we need to use the same sliding window (three weeks in, one week out). Thus, the forecasting task is slightly different than in the other models, which forecast 10 weeks ahead, given 38 weeks of data. In the LSTM, while the parameters are learned using the full 38 weeks of training data, to actually make a prediction, we use the prior 3 weeks of data to predict what the subsequent week will look like. Using this mechanism to forecast more than one week ahead thus requires us to assume the predictions are true, and feed them back in as part of the subsequent input sequence. In turn, this requires assuming a threshold for the forecasted probabilities, above which we will assume a ride takes place. The threshold we chose is a probability of 0.2. We chose this relatively low threshold for two reasons: first, on the training data, we observed that probability predictions were rarely above the more standard cut-off of 0.5. Moreover, as shown in the paper, the choice of 0.2 gave good performance on our focal task, which again was ranking likely day-hours for rides. Note, though, that the combination of this forecasting mechanism and this threshold choice leads to a high number of forecasted requests: uncertainty is propagated through the sliding window, as prior predictions are treated as truth, and there is a relatively low threshold for assuming a ride occurs. It may be possible to further optimize this choice of threshold to achieve good performance on *both* tasks, or to change the loss in a way that naturally captures both volume and timing. We put a reasonable amount of effort into carefully building this architecture to do a good job at the day-hour ranking task, to ensure we are comparing our model to a properly developed benchmark. We view further optimizations as beyond the scope of this work.

E. Churn and Stationarity in Ridesharing Behavior

As discussed in both the main body of the paper and in Appendix A, while the proposed model for routines can capture changing rates of usage over time through the parameters $\alpha_i(w)$ and $\gamma_i(w)$, it does not explicitly capture latent attrition (i.e., churn). In standard customer base analysis models, like the Pareto-GGG benchmark, the possibility of churn is captured by a latent attrition specification that specifically allows for a customer to move from an active state to a churned or "dead" state, wherein purchases stops forever. In contrast, our model can "zero out" purchasing by setting $\alpha_i(w)$ and $\gamma_i(w)$ to be low values, thus suggesting zero purchasing (i.e., the customers become "inactive"). We model these trajectories using Gaussian processes, which notably, eventually revert to their means when forecasting far from the range of the data. In our model, the means of these functions are individual-level constants reflecting the average level of non-routine and routine behavior (α_{0i} and γ_{0i} respectively). The rate at which the trajectories revert to their means is determined by the GP kernel's lengthscale parameter (ρ_α and ρ_γ respectively). This mean reversion feature suggests that our model will work best when individual-level purchasing (or, in our application, requesting) is relatively stationary; that is, if, in the long run, rates of usage roughly mirror past rates of usage.

In our simulations in Appendix A, we showed forecasting performance of our model is impoverished when there is a moderate level of churn. In this section, we investigate to what degree churn rates, and the lack of an explicit attrition process, may be a problem in our empirical application. To do so, we consider forecasting future number of requests over our 10-week holdout period across different subsets of customers.[3] Specifically, we consider four different subsets of customers:

1. **High vs. low variability:** We look at variance in customer-level in-sample purchasing rates, and perform a median split, dividing customers into low variability and high variability groups.

2. **Zero vs. non-zero holdout purchasing:** Within our 10-week validation window, we look at whether or not customers made any purchases.

[3] Note that our relatively short 10-week holdout period may limit our ability to find a substantial impact of churn: in contrast to typical customer base analysis applications, 10 weeks is a relatively short holdout period. Moreover, the lengthscale of the GP will limit the amount of mean reversion that is seen over short forecasting windows.

3. **Maybe churned, 5-week windows:** We look at whether customers made any requests in the 5 weeks before the end of the training window, and compare that to whether they made any requests in the 5 weeks before that. Intuitively, if a customer made no requests in the most recent 5 weeks, but did request rides in the 5 weeks prior to that, this suggests that they may have churned. Thus, we divide customers in this way, with "maybe churned" customers being those that did not make requests in the most recent 5 weeks, but did make requests before then.

4. **Maybe churned, 10-week windows:** This split is the same as the previous one, but using 10-week windows instead.

Then, we evaluate the mean absolute error in forecasting the number of requests across the two conditions in each split. We compare the performance of our model to the Pareto-GGG.

The full results of the analysis are shown in Table A-3. Across all splits, we find very small performance differences across the two models. Starting with high vs. low variability, the MAE of our model is 4.54 for low variability customers, versus 10.91 for high variability customers. In comparison, for the Pareto-GGG, the MAEs are 4.01 and 11.02, respectively. Turning to zero holdout requests, the MAE of our model is 2.25 for people who made zero holdout requests, versus 1.98 for the Pareto-GGG, suggesting our model is able to identify people who have become inactive. For "maybe churners" computed with the 5-week window, the MAE of our model is 3.69, versus 3.45 for the Pareto-GGG, and for "maybe churners" computed with the 10-week window, the MAE of our model is 2.69, versus 2.46 for the Pareto-GGG. Across these splits, we can see that the Pareto-GGG often improves on the proposed model, but the improvements are marginal. These results suggest that the purchasing processes in our data are relatively stationary, such that the lack of an explicit latent attrition process is not a substantial limitation for our model's forecasting performance.

Splitting Criteria	Our Model	Pareto-GGG
Split 1: Low vs. High Variance in Requests		
Low Variance	4.54	4.01
High Variance	10.91	11.02
Split 2: Made a Request During Holdout		
No	2.25	1.98
Yes	9.60	9.40
Split 3: Maybe Churned (5 Weeks)		
No	8.38	8.17
Yes	3.69	3.45
Split 4: Maybe Churned (10 Weeks)		
No	8.36	8.18
Yes	2.96	2.46

Table A-3: MAE Across Splits and Models
A comparison of model MAE across the four different splitting criterion, and the two models, showing a minimal difference in forecasting performance over our 10-week holdout period.

F. Calculating CP and MAP

In our discussion of our model's ability to predict ride times, we employed the conditional precision (CP) and mean average precision (MAP) statistics. We now describe how to compute those statistics. The basis of both of CP and MAP is the *top-k precision*, denoted $p(k)$, which is computed as follows: Let T denote the set of day-hours that a customer requested rides in a given week, and let R_k denote a ranking of the k most likely day-hours for that customer to request a ride, as predicted by the model. Here, we are ignoring i and w subscripts for notational simplicity: in our application, this ranking would be computed on a per-customer, per-week basis. With this notation, top-k precision is given by $p(k) := |R_k \in T|/k$, which, simply put, captures the fraction of the day-hours in R_k in which the customer actually requested a ride. As a running example, let us consider a person who took four rides, i.e., $y = |T| = 4$. Suppose those rides happened at day-hours $T = \{11, 22, 33, 44\}$. Suppose then that the model's top-6 predicted ride times were $R_6 = \{11, 22, 32, 33, 45, 44\}$. Then $p(1) = 1$ (since the top-ranked ride actually happened), $p(2) = 1$ (since both of the top-2 rides happened), but $p(3) = 2/3$ (since the rank-3 ride did not happen), and so on.[4] Having defined $p(k)$, we can now define our two metrics of interest:

- **Mean Average Precision (MAP):** To define MAP, we first define the average precision (AP) of the day-hour rankings for a given customer (again omitting the i and w subscripts):

$$AP = \frac{1}{y} \sum_{k=1}^{168} p(k)\, \mathbb{I}(\text{Request @ k}). \tag{A-3}$$

  Here, $\mathbb{I}(\text{Request @ k}) = 1$ if the user made a request at the day-hour ranked k and 0 otherwise. The MAP is then the mean AP across all customers. To illustrate the intuition behind MAP, let us return to our example with true ride times $T = \{11, 22, 33, 44\}$, and $R_6 = \{11, 22, 32, 33, 45, 44\}$. The average precision for this user would be $\frac{1}{4}(1 + 1 + 0 + \frac{3}{4} + 0 + \frac{4}{6} + 0 + \dots) = 0.854$. The AP is always between 0 and 1, with higher values indicating that the model is producing better rankings of the day-hours for that customer. The AP will be 1 if all of the user's rides happened during the highest ranked hours.

- **Conditional Precision (CP):** The conditional precision is similar to the classic precision met-

[4] Note that this metric does not separate adjacent mispredictions (e.g., if the true day-hour of a ride were 33, a prediction of day-hour 34 or 14 would both count as mispredictions). However, note that our model is likely to rank similarly adjacent day-hours due to the correlation between days and hours induced by the day-hour kernel.

ric. Suppose that we know a given user rode y times in a given week; the conditional precision captures which of those y day-hours the model predicts correctly. Mathematically, CP is equal to $p(y)$.

Note that, while these metrics all share the word "precision" in their names, they are connected to both precision and recall, in the classic senses of those terms. MAP, in particular, can be viewed as measuring the area under the precision-recall curve. As an example, imagine a customer who made two requests, one of which was expected by the model and ranked first, and another which was unexpected and ranked last. The MAP of this scenario is $\frac{1}{2}(1+\frac{1}{168})$, which takes into account both the successful ranking of the first request, and the very unsuccessful ranking of the second request.

G. Additional Summary Statistics

Table A-4: Additional summary statistics.
Summary statistics for the ride-related covariates. The variables above the horizontal line are variables about the proposal itself; those below the line are about the actual trip that was taken.

Statistic	N	Mean	St. Dev.	Min	Max
Requests Before	38,305	0.9	3.0	0.0	103.0
Ride Distance	38,305	3.3	2.5	0.1	27.9
Airport Ride	38,305	0.03	0.2	0.0	1.0
Solo Trip	38,305	0.9	0.3	0.0	1.0
Had ViaExpress Proposal	38,305	0.4	0.5	0.0	1.0
Had Shared Taxi Proposal	38,305	0.04	0.2	0.0	1.0
Sedan	38,304	0.3	0.4	0.0	1.0
Van	38,304	0.3	0.5	0.0	1.0
Ride Cost (Cents)	38,167	879.1	878.1	0.0	11,936.0
Driver ETA	38,305	7.7	4.0	0.1	70.1
ETA Destination	38,305	31.1	13.5	0.1	168.1
Speed	38,305	0.1	0.7	0.01	131.1
Pickup Walking Dist.	38,305	109.2	83.5	0.0	600.0
# Passengers Request.	38,305	1.2	0.5	1	6
Pickup Delay	16,159	1.1	2.7	−12.2	48.3
Dropoff Delay	16,159	2.6	8.9	−41.4	630.0
Dropoff Walking Dist.	19,374	88.8	70.2	0.0	577.0
# On-board (Pickup)	16,159	0.8	1.0	0.0	5.0
# On-board (Dropoff)	16,159	0.7	1.0	0.0	5.0
Max On-board	16,159	2.5	1.3	1.0	7.0

H. Details of Clustering Routines

To cluster routines into routine types, we used the `latrend` package in the `R` statistical programming language (Den Teuling 2023), which is specifically designed to cluster longitudinal data. In particular, we used the parametric longitudinal k-means method (`lcMethodLMKM` in `latrend`), which has been found to provide superior performance compared to traditional longitudinal K-means methods for trajectory clustering (Den Teuling, Pauws, and van den Heuvel 2023). Regarding the selection of number of clusters, we estimated the model with $K = 1, \ldots, 12$ clusters and examined the WMAE (weighted mean absolute error) as measure of fit (Figure A-12).

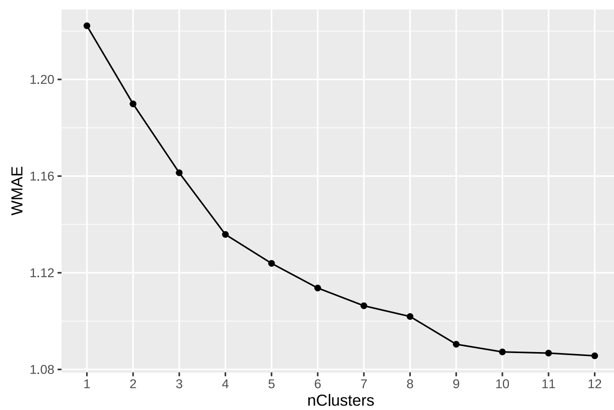
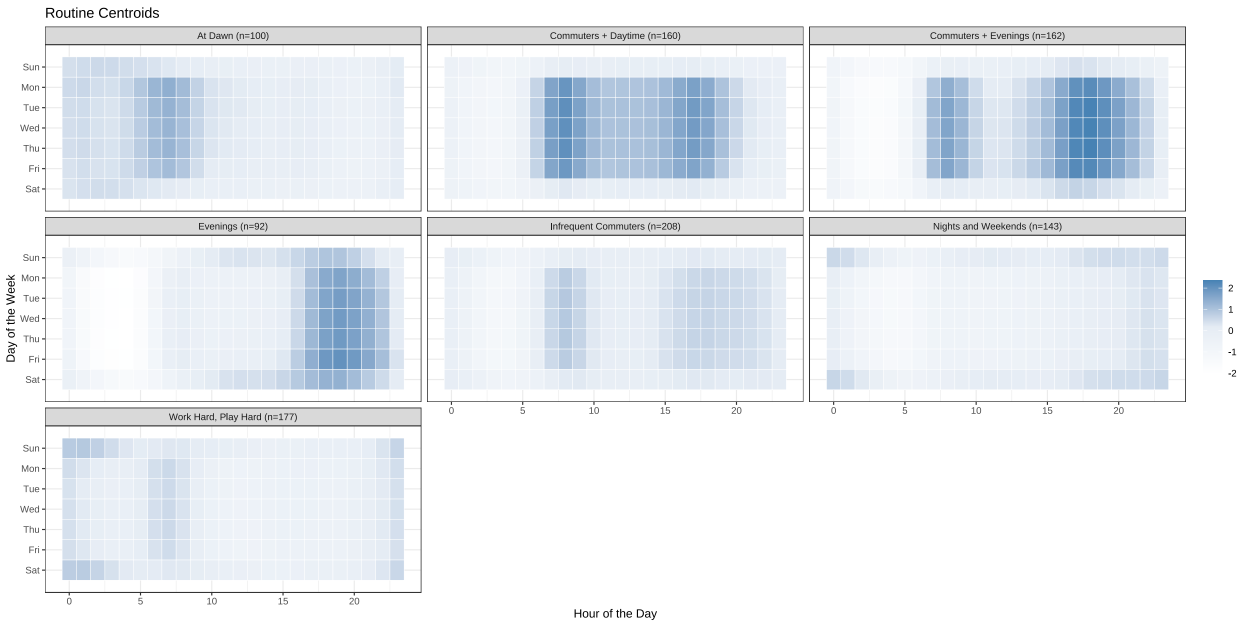
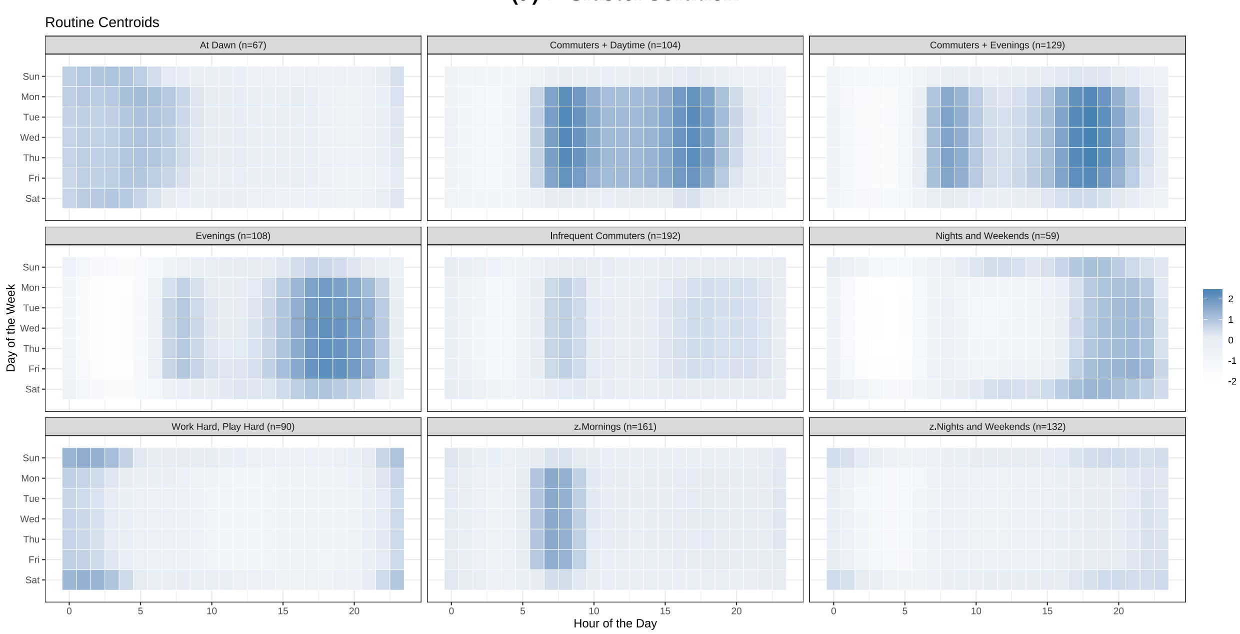


Figure A-12: Clustering Analyses: Selecting the Number of Clusters
WMAE (weighted mean absolute error) of each cluster solution, with clusters varying from $N = 1, \ldots, 12$.

Based on WMAE, there were two solutions that seemed reasonable, which corresponded to the 7-cluster and the 9-cluster solutions. We compared the individual clusters across them (Figures A-13a and A-13b) and corroborated that most clusters (Commuters + Daytime, Evenings, Infrequent Commuters, Commuters + Evenings, At Dawn) remained largely unchanged. Compared with the 7-cluster solution, the 9-cluster solution seemed to split two clusters (Nights and Weekends, Work Hard, Play Hard), providing more nuances to the day-hour patterns of those two clusters. Given the consistency across both solutions, we chose the solution with 7 clusters for simplicity.



(a) 7-Cluster Solution



(b) 9-Cluster Solution

Figure A-13: Comparing 7-cluster and 9-cluster Solutions
For easy comparison, clusters 1 through 7 are in the same order across the figures. The last two clusters in Figure A-13b are the new clusters that emerged from the 9-cluster solution.

I. Additional Behaviors by Routine Type

In Figure A-14, we show how six different behaviors vary across our routine types. The specific behaviors are:

- Airport Rides – whether a request is to the airport or not
- Pickup Walking Dist – how long customers have to walk to get their shared ride
- Proposed ETA – how long the customer is expected to wait until the driver arrives, as part of the ride proposal
- Proposed Speed – calculated by looking at the requested trip length, and the proposed trip time
- Ride Cost (cents) – how much the customer will pay for the trip
- Solo Trips – how often the customer requests a trip just for 1 passenger

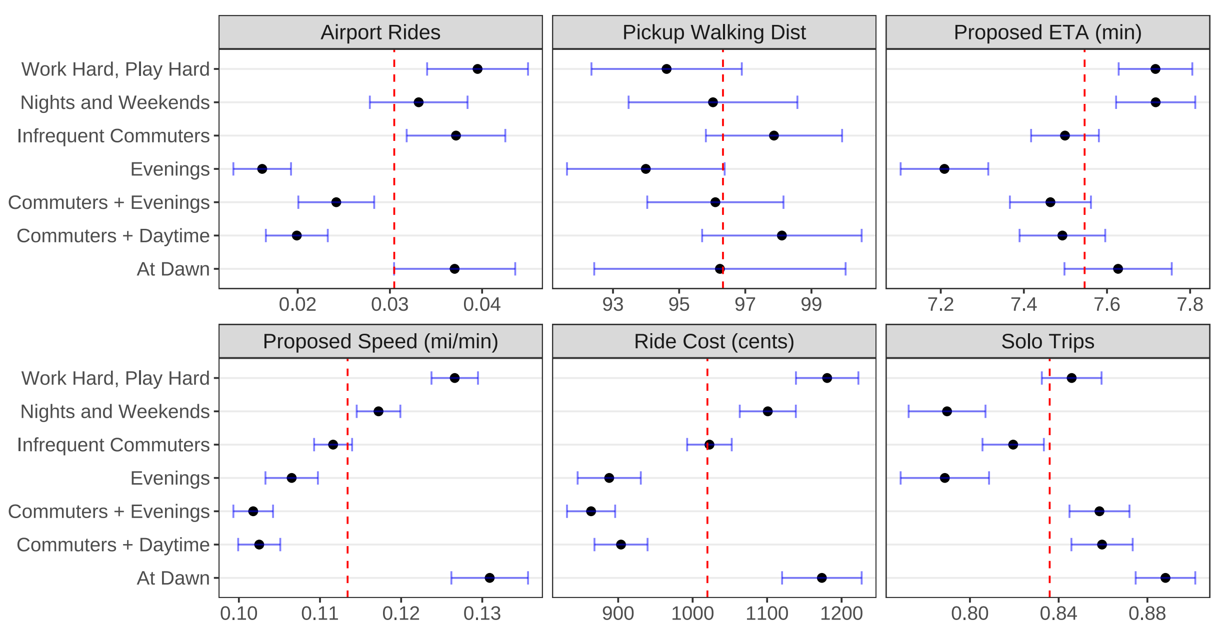


Figure A-14: Request characteristics, by routine type
The average value of 6 proposal-level variables, by routine type. Error bars represent standard errors, and the vertical line represents the overall mean.

J. Which Customers Have Routines

In this appendix, we consider which types of customers, in terms of observable characteristics, tend to develop routines. Understanding what makes for a routine customer can further validate our routineness metric, and suggest ways in which the focal firm might consider acquiring routine customers, or cultivating routines. To explore this phenomenon, we regress each user's week 38 routineness on a number of variables describing that user's activity and trip types averaged over the training period.[5] The results are shown in Table A-5.

Table A-5: Predictors of routineness.
Regression of week 38 routineness (DV) on average trip characteristics (IVs), where the average is taken over the whole training window. (These results are robust if we use the average routineness as the DV instead of its week 38 value.)

	Dependent variable:
	Routineness
Prob. Ride | Request	0.555^{***}
	(0.116)
First Week	-0.0003
	(0.001)
# Requests	0.401^{***}
	(0.008)
Price	-0.001^{***}
	(0.0001)
ETA Driver	0.047^{**}
	(0.019)
ETA Destination	-0.001
	(0.005)
# Passengers Req.	-0.131^{**}
	(0.061)
Walking Distance	-0.001^{*}
	(0.001)
Distance	0.116^{***}
	(0.025)
Airport Ride	-0.033
	(0.269)
Observations	2,000
R^2	0.605
Note:	$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

We find that high-routineness users have much in common: first, echoing our main CRM results, we see that routine customers have a higher probability of accepting a ride, given a request.

[5]The results are very similar when we consider the average routineness over all weeks as a DV. We prefer using week 38 routineness for consistency with the rest of the paper.

We also find that customers who take longer trips, or request a ride with fewer other passengers (i.e., are more likely to request a "solo trip," as opposed to bringing friends along for the ride) are more likely to have higher routineness. Partly, this may be explained by the prevalence of commuting routines, which intuitively may be more likely to be solo trips, and may be from more remote areas of the city to more central ones. A similar self-selection story may explain the positive association of routineness and driver ETA: if routine customers are traveling at peak hours, it may take longer to find a driver. More interestingly, we find that routineness is associated with lower priced trips, and lower walking distance. While our analysis is not causal, these effects are suggestive: customers who are consistently confronted with high prices or high walking distances may stop using the platform, or never form routines.

K. "When" versus "What": Incorporating Location Information

In the main body of the paper, we focused exclusively on temporal routines: that is, *when* someone interacts with the firm, not *what* they do in that interaction. In a retail setting, for instance, a customer may come in at exactly the same time each week, but may buy either the same items each time, or different items. In our focal context, customers may request rides at exactly the same times each week, but may travel to either the same location each time, or different locations. Consider, for example, two work commuters, both of whom work in the same location each day. In the morning, both users may always go between the same locations, home and work. In the evening, however, one of these commuters may always return home, while the other frequently goes out for drinks or dinner. In this sense, both customers have the same "when" routine but different "what" routines. In this section, we explore to what degree "what" routines — that is, location choice — are predictive of "when" routines, and what gains there may be in accounting for "what" routines, in addition to our previously defined routineness measure.

Metrics for Location Dispersion To understand the degree to which there are "what" routines in location choice, we first need a metric of how consistent location choices are. Mathematically, it is more natural to construct measures of how dispersed (that is, how *in*consistent) trip locations are.[6] To understand location dispersion, we look at both pick-up and drop-off locations. In our data, locations are saved as precise latitude/longitude coordinates (or, "lat/long"), measured to five decimals. To discretize our location data to correspond to New York City street blocks, we truncate the decimal to the nearest 300th.[7] With this discretization, we then define two measures of location dispersion:

1. **Shannon Entropy:** For this metric, we consider the empirical distribution of a user's locations (both pick-up and drop-off). For instance, if a user made ten total trips, nine to location 1 and one to location 2, the empirical distribution would be (0.9, 0.1). We then compute the Shannon entropy of that distribution, defined as:

$$\text{Entropy} = -\sum_{\ell=1}^{L} p_k \log p_k. \tag{A-4}$$

[6] Though each of our measures could be easily converted to a consistency measure by inversion.

[7] Specifically, given a raw coordinate x, we compute a truncated coordinate, $x^* = \lfloor 300x + 0.5 \rfloor / 300$.

where p_ℓ is the empirical probability of the ℓth location, and L is the total number of locations. Intuitively, entropy captures how "predictable" a user's locations are. To illustrate, observe that $\text{Entropy}(0.9, 0.1) < \text{Entropy}(0.5, 0.5) < \text{Entropy}(0.1, 0.1, \ldots, 0.1)$.

2. **CRT Dispersion:** A feature of the entropy measure is that it does not take into account the total number of trips or locations for a given user. Hence, a user that takes 1,000 trips to the same 10 locations is just as entropic as a user who takes 10 trips to the same 10 locations. Hence, as an alternative to entropy, we consider a metric inspired by the Chinese Restaurant Table (CRT) process. The CRT process is a stochastic process, derived from the better known Chinese Restaurant Process, commonly used to model assignment to different groups (Zhou and Carin 2013). It captures a "rich get richer" process, whereby new observations are either assigned to existing groups, with probability proportional to the sizes of those groups, or they are assigned to a new group, at a rate proportional to a dispersion parameter. Such a process could be used to model the evolution of trip location choices: for a new trip, that trip may be to an existing location, or it may be to a new location. While estimating the full CRT model is complex and cumbersome, there exists an intuitive and easily computed estimator of the CRT dispersion parameter, θ (Durrett 2008, Chapter 1): given L unique locations in K total trips:

$$\theta \approx \frac{L}{\log K}. \tag{A-5}$$

In plain English, this metric is the number of unique locations divided by the log number of total trips. Contrasting this to entropy, if a user has just two unique locations in ten trips, $\theta = 2/\log(10) \approx 0.87$, regardless of whether those trips were split 5/5 or 9/1. But, unlike entropy, if a user visits 10 locations equally among 1000 trips, $\theta = 10/\log(1000) \approx 1.45$, whereas if the user visits 10 locations equally among 10 trips, $\theta = 10/\log(10) \approx 4.34$.

Location Results Having defined metrics of location dispersion, we now ask: first, to what degree is location dispersion related to temporal routineness? And second, is location choice also an important predictor of customer-level outcomes? To answer the first question, we plot the joint distribution of our routineness metric and the two metrics of location dispersion in Figure A-15.[8]

[8] For consistency, we focus on routineness as measured in the last week of our training data, but all the results are the same if we consider average routineness over the training data instead.

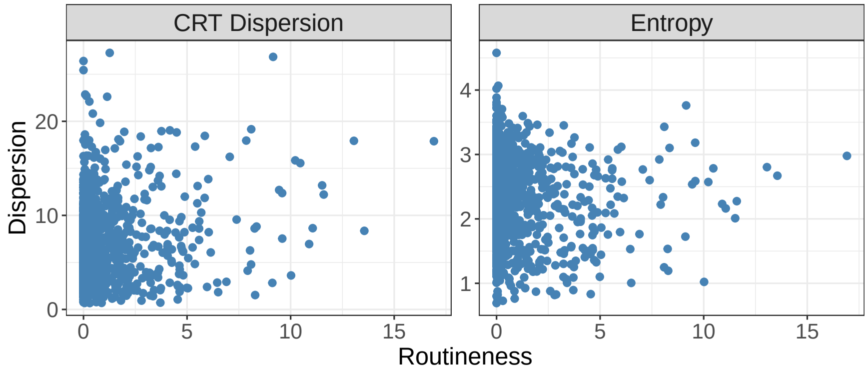


Figure A-15: Joint Distributions of Routineness and the Two Location Dispersion Metrics.
The joint distribution of routineness in week 38 (i.e., E_{i38}^{Routine}), and the two metrics of dispersion in location choice, entropy and CRT dispersion, showing no relationship between the two.

We see that there is no obvious relationship between temporal routineness and location dispersion. This is supported by simple regression analyses: alone, entropy and CRT dispersion explain less than 5% of the variation in routineness.

When we regress routineness on both location metrics, together with other obvious individual-level controls (e.g., number of final week requests, the week the customer was acquired), we find CRT dispersion is a negative and significant predictor of routineness, while entropy is not significant. We report the full results from these regressions in Table A-6. Taken together, these results suggest that location dispersion is minimally predictive of temporal routineness, albeit in an intuitive direction: customers with less dispersed location choices seem to be slightly more routine.

Having established that routineness and location dispersion are distinct, we now ask: does location dispersion have an impact on customer behavior over and above temporal routineness? Specifically, we return to the focal analyses where we explored the role of routineness in explaining the number of future requests and the likelihood of a customer being active in the future, but now also include our location dispersion metrics. We find that, when the location dispersion metrics are included alongside routineness, neither entropy nor CRT dispersion is a significant predictor of either outcome, suggesting that, from a CRM perspective, "when" matters significantly more than "what." We display these results in Table A-7.

Table A-6: Explaining routineness with location dispersion.
Regression of week 38 routineness (DV) on the two measures of location dispersion, along with some standard customer-level controls (IVs).

	Dependent variable:		
	Routineness		
	(1)	(2)	(3)
Entropy	-0.220^{***}		-0.128^{*}
	(0.060)		(0.071)
CRT Disp.		0.081^{***}	-0.036^{***}
		(0.009)	(0.014)
Sessions			0.005^{***}
			(0.0004)
Prob. Ride | Request			0.115
			(0.124)
First Week			0.002^{**}
			(0.001)
# Requests ($w = 38$)			0.453^{***}
			(0.010)
Observations	2,000	2,000	2,000
R^2	0.007	0.036	0.684
Note:			*p<0.1; **p<0.05; ***p<0.01 Intercept omitted for clarity.

Table A-7: Regressions of location dispersion and customer activity.
Regression of future activity (DVs) on individual-level summary statistics and the two measures of location dispersion.

	Dependent variable:	
	# Requests	Activity
	OLS	*logistic*
	(1)	(2)
Requests $(w = 38)$	1.812^{***}	0.398^{***}
	(0.236)	(0.104)
Recency	-0.227^{***}	-0.141^{***}
	(0.042)	(0.010)
Frequency	0.090^{***}	-0.0002
	(0.009)	(0.003)
Routine $(w = 38)$	5.280^{***}	1.069^{***}
	(0.385)	(0.333)
Entropy	-1.848^{*}	0.095
	(1.082)	(0.240)
CRT Disp.	0.108	-0.017
	(0.197)	(0.047)
Observations	2,000	2,000
R^2	0.575	
Note:	*p<0.1; **p<0.05; ***p<0.01	
	Intercept omitted for clarity.	

L. PyMC Implementation

Here, we share an efficient implementation of the model, using `PyMC`, built on the `Aesara` library, which we subsequently compiled to `NumPyro` and `JAX` to run on a GPU. We thank our partner company, Via, for their help on improving this code.

```
# Base imports
import numpy as np

# PyMC-related imports
import pymc as pm
import aesara
import aesara.tensor as at

# Other imports
from numpy import pi as pi

# reference: https://discourse.pymc.io/t/avoiding-looping-when-using-
#   gp-prior-on-latent-variables/9113/9
class FixedMatrixCovariance(pm.gp.cov.Covariance):
    def __init__(self, cov):
        # super().__init__(1, None)
        self.cov = at.as_tensor_variable(cov)
        self.input_dim = 1

    def full(self, X, Xs):
        # covariance matrix known, not explicitly function of X
        return self.cov

    def diag(self, X):
        return at.diag(self.cov)


def create_model(y, nz_mask, include_obs, n_week_fore, args):
    """Creates the usage model for routines.

    Args:
        y: n_cust x n_week_train array of usage counts (note: this code can be adapted to only use
            non-zero counts; this implementation includes zeros in y)
        nz_mask: an array specifying which elements of y are non-zero
        include_obs: an array that captures which columns in y happened after a customer's
            acquisition date + three weeks
        n_week_fore: the number of weeks ahead to forecast

    Returns:
        A PyMC model corresponding to our proposed routines model.
    """

    n_cust, n_week_train, n_dayhour = y.shape
    n_week_total = n_week_train + n_week_fore

    # inputs are always just the range of possible values, used for creating GPs later
    cust_inputs = np.arange(1, n_cust + 1)[:, None]
    week_inputs = np.arange(1, n_week_total + 1)[:, None]
    day_inputs = np.arange(1, 8)[:, None]
```

```python
hour_inputs = np.arange(0, 24)[:, None]

routines_model = pm.Model()
with routines_model:
    # Create aesara objects from data:
    y_at = aesara.shared(y)
    nz_mask_at = aesara.shared(nz_mask)
    include_obs_at = aesara.shared(include_obs)

    # This will be used several times for hierarchical GPs:
    identity_matrix = at.eye(n_cust)
    identity_cov = FixedMatrixCovariance(identity_matrix)

    # GP model for random scaling term, alpha:

    alpha0_scale = pm.HalfNormal("alpha0_scale", sigma=10)
    alpha0 = pm.Normal(
        "alpha0", mu=0, sigma=alpha0_scale, shape=(n_cust)
    )

    ## Hyperparameters
    alpha_amp = pm.HalfNormal("alpha_amp")
    alpha_ls = pm.InverseGamma("alpha_ls", alpha=5, beta=5)

    ## Alpha cov
    cov_alpha = alpha_amp**2 * pm.gp.cov.Exponential(input_dim=1, ls=alpha_ls)

    ## Alpha prior
    alpha_offset_gp = pm.gp.LatentKron(cov_funcs=[identity_cov, cov_alpha])
    alpha_offset = alpha_offset_gp.prior(
        "alpha_offset", Xs=[cust_inputs, week_inputs]
    )
    alpha = pm.Deterministic(
        "alpha",
        alpha0.dimshuffle(0, "x") + alpha_offset.reshape((n_cust, n_week_total)),
    )

    # Common dayhour rate term, mu
    ## Day correlation term:
    mu_omega_chol, mu_omega_corr, mu_omega_scale = pm.LKJCholeskyCov(
        "mu_omega",
        n=7,
        eta=2.0,
        sd_dist=pm.HalfNormal.dist(shape=7),
        compute_corr=True,
    )
    cov_mu_day = FixedMatrixCovariance(mu_omega_corr)

    ## Periodic hour term:
    mu_amp = pm.HalfNormal("mu_amp")
    mu_ls = pm.TruncatedNormal("mu_ls", mu=0.5 * pi, sigma=0.25 * pi, lower=0)
    cov_mu_periodic = mu_amp**2 * pm.gp.cov.Periodic(
        input_dim=1,
        period=24,
        ls=mu_ls / 2, # note: /2 needed to recover original per kernel defn
    )

    ## Combine using Kronecker structure:
    mu_gp = pm.gp.LatentKron(cov_funcs=[cov_mu_day, cov_mu_periodic])
```

```
    mu = mu_gp.prior("mu", Xs=[day_inputs, hour_inputs])

    # GP model for routine scaling term, gamma
    ## Define gamma = gamma0 + gamma_offset
    gamma0_scale = pm.HalfNormal("gamma0_scale", sigma=10)
    gamma0 = pm.Normal(
        "gamma0", mu=0, sigma=gamma0_scale, shape=(n_cust)
    )

    ## Hyperparameters
    gamma_amp = pm.HalfNormal("gamma_amp")
    gamma_ls = pm.InverseGamma("gamma_ls", alpha=5, beta=11)

    ## Gamma Covariance
    cov_gamma = gamma_amp**2 * pm.gp.cov.Exponential(input_dim=1, ls=gamma_ls)

    ## Gamma prior
    gamma_offset_gp = pm.gp.LatentKron(cov_funcs=[identity_cov, cov_gamma])
    gamma_offset = gamma_offset_gp.prior(
        "gamma_offset", Xs=[cust_inputs, week_inputs]
    )
    gamma = pm.Deterministic(
        "gamma",
        gamma0.dimshuffle(0, "x") + gamma_offset.reshape((n_cust, n_week_total)),
    )

    # GP model for routine rate, eta

    ## Day correlation term:
    eta_omega_chol, eta_omega_corr, eta_omega_scale = pm.LKJCholeskyCov(
        "eta_omega",
        n=7,
        eta=2.0,
        sd_dist=pm.HalfNormal.dist(shape=7),
        compute_corr=True,
    )
    cov_eta_day = FixedMatrixCovariance(eta_omega_corr)

    ## Periodic hour term:
    eta_amp = pm.HalfNormal("eta_amp")
    eta_ls = pm.TruncatedNormal("eta_ls", mu=0.5 * pi, sigma=0.25 * pi, lower=0)
    cov_eta_periodic = eta_amp**2 * pm.gp.cov.Periodic(
        input_dim=1,
        period=24,
        ls=eta_ls / 2, # note: /2 needed to recover original per kernel defn
    )

    ## Combine using Kronecker structure:
    eta_gp = pm.gp.LatentKron(
        cov_funcs=[identity_cov, cov_eta_day, cov_eta_periodic]
    )
    eta_unshaped = eta_gp.prior(
        "eta_unshaped", Xs=[cust_inputs, day_inputs, hour_inputs]
    )
    eta = pm.Deterministic("eta", eta_unshaped.reshape((n_cust, n_dayhour)))

    # Compute likelihood
    intensity = at.exp(
        alpha.dimshuffle(0, 1, "x")[:, :n_week_train, :] + mu.dimshuffle("x", "x", 0)
```

```
        ) + at.exp(
            gamma.dimshuffle(0, 1, "x")[:, :n_week_train, :] + eta.dimshuffle(0, "x", 1)
        )
        lp1 = at.sum(y_at[nz_mask_at] * at.log(intensity[nz_mask_at]))
        lp2 = at.sum(intensity[include_obs_at])
        pm.Potential("lp", lp1 - lp2)

    return routines_model
```

References

Dang, Khue-Dung, Matias Quiroz, Robert Kohn, Tran Minh-Ngoc, and Mattias Villani (2019), "Hamiltonian Monte Carlo with energy conserving subsampling," *Journal of Machine Learning Research*, 20, 1–31.

Den Teuling, Niek *latrend: A Framework for Clustering Longitudinal Data* (2023) https://github.com/philips-software/latrend, r package version 1.5.1.

Den Teuling, Niek G. P., Steffen C. Pauws, and Edwin R. van den Heuvel (2023), "A comparison of methods for clustering longitudinal data with slowly changing trends," *Communications in Statistics-Simulation and Computation*, 52 (3), 621–648.

Durrett, Richard (2008), *Probability Models for DNA Sequence Evolution*, Vol. 2. Springer.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016), *Deep Learning* MIT press.

Hoffman, Matthew D., David M. Blei, Chong Wang, and John Paisley (2013), "Stochastic variational inference," *Journal of Machine Learning Research*, 14, 1303–1347.

Platzer, Michael and Thomas Reutterer (2016), "Ticking Away the Moments: Timing Regularity Helps to Better Predict Customer Activity," *Marketing Science*, 35 (5), 779–799.

Sarkar, Mainak and Arnaud De Bruyn (2021), "LSTM Response Models for Direct Marketing Analytics: Replacing Feature Engineering with Deep Learning," *Journal of Interactive Marketing*, 53 (1), 80–95.

Zhou, Mingyuan and Lawrence Carin (2013), "Negative Binomial Process Count and Mixture Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (2), 307–320.