Bayesian Nonparametric Customer Base Analysis with Model-based Visualizations

Ryan Dew* and Asim Ansari†
Columbia University

June 22, 2017

Abstract

Marketing managers are responsible for understanding and predicting customer purchasing activity, a task that is complicated by a lack of knowledge of all of the calendar time events that influence purchase timing. Yet, isolating calendar time variability from the natural ebb and flow of purchasing is important, both for accurately assessing the influence of calendar time shocks to the spending process, and for uncovering the customer-level patterns of purchasing that robustly predict future spending. A comprehensive understanding of purchasing dynamics therefore requires a model that flexibly integrates both known and unknown calendar time determinants of purchasing with individual-level predictors such as interpurchase time, customer lifetime, and number of past purchases. In this paper, we develop a Bayesian nonparametric framework based on Gaussian process priors, which integrates these two sets of predictors by modeling both through latent functions that jointly determine purchase propensity. The estimates of these latent functions yield a visual representation of purchasing dynamics, which we call the model-based dashboard, that provides a nuanced decomposition of spending patterns. We show the utility of this framework through an application to purchasing in free-to-play mobile video games. Moreover, we show that in forecasting future spending, our model outperforms existing benchmarks.

Keywords: Customer Base Analysis, Dynamics, Analytics Dashboards, Gaussian Process Priors, Bayesian Nonparametrics, Visualization, Mobile Commerce.

*Ryan Dew (email: ryan.dew@columbia.edu) is a doctoral student in Marketing at Columbia Business School, Columbia University.

†Asim Ansari (email: maa48@columbia.edu) is the William T. Dillard Professor of Marketing at Columbia Business School, Columbia University.

1 Introduction

Marketers in multi-product companies face the daunting task of understanding the ebb and flow of aggregate sales within and across many distinct customer bases. Such spending dynamics stem from both the natural stochastic process of purchasing that is characterized by customers' interpurchase times, lifetimes with the firm, and number of past purchases, and from the influence of managerial actions and shocks operating in calendar time. These other shocks are often outside the control of the company, and include events such as holidays, barriers to purchasing like website outages, and competitor actions. While individual-level factors such as the recency of purchasing are often powerful predictors of future spend activity, managers think and act in calendar time. Hence, to successfully execute a customer-centric marketing strategy, managers need to understand how calender time events interact with individual-level effects in generating aggregate sales.

An accurate accounting of the underlying drivers of spending is not possible unless both individual-level and calendar time effects are simultaneously modeled. For example, in models of spending that omit calendar time and rely solely on individual-level effects, momentary disruptions in spending that occur in calendar time may be erroneously conflated with predictable, individual-level purchase propensities. Similarly, a small bump in spending on any given calendar day could represent random noise if many customers are still active on that day, or a significant calendar time event if few customers are still active. Importantly, activity level is unobserved, but can be captured by individual-level variables like interpurchase time. Flexibly including both types of effects in an individual-level model of purchase propensity is thus crucial for dynamic customer base analysis, and the development of such a framework is our primary objective.

In this paper, we describe a flexible and robust Bayesian nonparametric framework for customer base analysis that accomplishes that objective by probabilistically modeling purchase propensities in terms of underlying dynamic components. We demonstrate the utility of our new framework on spending data from mobile video games. Our model uses Gaussian process priors over latent functions to integrate events that occur at multiple time scales and across different levels of aggregation, including both calendar time and individual-level time scales like interpurchase time, time since first purchase (customer lifetime), and number of past purchases. Its nonparametric specification allows for the flexible modeling of different patterns of effects, such that the model can be seamlessly applied across different customer bases and dynamic contexts. The resulting latent function estimates facilitate automatic model-based visualization and prediction of spending dynamics.

Customer base analysis is central to modern marketing analytics. Contributions in this area have focused on the stochastic modeling of individuals in terms of interpurchase time and lifetime, in contractual and non-contractual settings (Fader et al., 2005; Schmittlein et al., 1987; Fader et al., 2010; Schweidel and Knox, 2013). These papers show that customer-level effects can explain much of the variability of spending over time. However, they typically omit, or assume a priori known, calendar time effects. Events in calendar time, including marketing efforts and exogenous events such as competitor actions, holidays, and day-of-the-week effects, can substantially impact spending in many industries.

For digital products, such as those in our application, relevant calendar events include product changes that are launched simultaneously to all customers, and exogenous shocks such as website or e-commerce platform outages and crashes. Moreover, many of these events pose a common problem to marketing analysts: although calendar time events undoubtedly influence spend rates, analysts may be unaware of the form of that influence, or of the very existence of certain events. This problem is exacerbated in larger companies, where the teams responsible for implementing marketing campaigns or managing products may be distinct from the analytics team, and where information may not flow easily across different organizational silos.

To cope both with such information asymmetries and with unpredictable dynamics in spending, sophisticated managers often rely on aggregate data methods, including exploratory data analyses, statistical process control, time series models (Hanssens et al., 2001), and predictive data mining methods (Neslin et al., 2006). These tools can forecast sales, model the impact of calendar time events, and provide metrics and visual depictions of dynamic patterns that are easy to grasp. Unfortunately, these methods typically ignore individual-level predictors of spend, like those captured by customer base analysis models, which precludes their use in characterizing customer-level spend behaviors and in performing CRM-relevant tasks. Furthermore, not including these individual-level effects means these models cannot account for the latent activity level of customers, which may in turn lead to an inaccurate understanding of the true nature of calendar time events.

Building on both the customer base analysis and aggregate data approaches, we use Bayesian nonparametric Gaussian process (GP) priors to fuse together latent functions that operate both over calendar time and over more traditional individual-level inputs, such as interpurchase time, customer lifetime, and purchase number. In this way, we integrate calendar time insights into the customer base analysis framework. We use these latent functions within a discrete hazard specification to dynamically model customer purchase propensities, while controlling for unobserved heterogeneity. We term the resulting model the Gaussian Process Propensity Model (GPPM). While Bayesian nonparametrics have been successfully applied to marketing problems (e.g. Ansari and Mela, 2003; Wedel and Zhang, 2004; Kim et al., 2007; Rossi, 2013; Li and Ansari, 2014), to the best of our knowledge, our paper is the first in marketing to take advantage of the powerful GP methodology. It is important to note that, although our paper applies GPs in the context of customer purchasing, GPs provide a general mechanism for estimating latent functions, and can be employed in many other substantive contexts. We therefore also provide an accessible introduction to GPs in general, to encourage their wider adoption within marketing.

In our application, the GP nonparametric framework means that the shapes of the latent propensity functions that govern purchasing are automatically inferred from the data, thus providing the flexibility to robustly adapt to different settings, and to capture time-varying effects, even when all the information about inputs may not be available. The inferred latent functions allow a visual representation of both calendar time and individual-level patterns that characterize spend dynamics, something that is not possible in standard probability models, where the output is often a set of possibly unintuitive parameters. We refer to the collection of these plots as the model-based dashboard, as it gives a visual summary of the patterns of spending in a particular customer base, and serves as a tool for analyzing the

spending dynamics within and across customer bases. It is important to note that these model-based dashboards are distinct from real-time dashboards that continuously stream various marketing metrics, like those described in Pauwels et al. (2009).

In this paper, we begin by describing what Gaussian process priors are (Section 2.1), and how they can be used to specify latent dynamics in a model for dynamic customer base analysis (Sections 2.2 and 2.3). We then apply our model to spending data from two mobile video games owned by a large American video game publisher. These games are quite distinct, spanning different content genres and target audiences. We show how the parameter estimates and accompanying model-based dashboards generated from our approach can facilitate managerial understanding of the key dynamics within each customer base, both in the aggregate and at the individual level (Sections 3.1 and 3.2). We compare the GPPM to benchmark probability models, including different buy-till-you-die variants such as the BGNBD (Fader et al., 2005) and the Pareto-NBD (Schmittlein et al., 1987), hazard models with and without time-varying covariates (e.g. Gupta, 1991; Seetharaman and Chintagunta, 2003), and variants of the discrete hazard approach, including a sophisticated state-space specification, and show that the GPPM significantly outperforms these existing benchmarks in fit and forecasting tasks (Section 3.4). We conclude by summarizing the benefits of our framework, citing its limitations, and identifying areas of future research.

2 Modeling Framework

In our framework for dynamic customer base analysis, we focus on flexibly modeling individual-level purchase propensity. We model this latent propensity in terms of the natural variability in purchase incidence data along four dimensions: calendar time, interpurchase time (recency), customer lifetime, and number of past purchases. Our focus on modeling purchase incidence is consistent with the majority of the literature on customer base analysis, and also fits nicely with our application area, where we focus on purchasing of a single product, and where there is minimal variability in spend amount.[1] We use a discrete-time hazard framework to specify the purchase propensity, as most customer-level data are available at a discrete level of aggregation. This is also the case in our application, where daily data are available.

The observations in our data consist of a binary indicator y_{ij} that specifies whether customer i made a purchase at observation j, and a corresponding tuple $(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$ containing the calendar time, recency, customer lifetime, and number of past purchases, respectively. Recency here refers to interpurchase time, or the time since the customer's previous purchase, while customer lifetime refers to the time since the customer's first purchase. Depending on the context, a vector $\mathbf{z}_i$ of demographics or other time invariant variables, such as the customer acquisition channel or acquisition date, may also be available. The probability of customer i purchasing is modeled as

$$\Pr(y_{ij} = 1) = \text{logit}^{-1}\left[\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij}) + \mathbf{z}_i'\boldsymbol{\gamma} + \delta_i\right], \tag{1}$$

[1] Throughout the rest of the paper, we use the words purchasing and spending interchangeably to refer specifically to purchase incidence.

where,

$$\text{logit}^{-1}(x) = \frac{1}{1 + \exp(-x)}.$$

We see in Equation 1 that the purchasing rate is driven by a time-varying component $\alpha(.)$ and two time invariant effects, $\mathbf{z}_i'\boldsymbol{\gamma}$ and δ_i, which capture the observed and unobserved sources of heterogeneity in base spending rates, respectively. This setup models spend dynamics via aggregate trajectories—that is, all customers are assumed to follow the same dynamic pattern—while maintaining individual heterogeneity in the spending process via the random effect δ_i and by using other observed individual-specific variables, $\mathbf{z}_i$, when available. In our application, we will focus exclusively on unobserved heterogeneity. It is important to note that while calendar time is an aggregate time scale, the recency, lifetime, and purchase number dimensions are individual-level time scales. That is, customers may, at any given point in calendar time t, be at a different positions in the $(r_{ij}, \ell_{ij}, q_{ij})$ subspace, and therefore the aggregate sales at any given calendar time t are the amalgam of the activities of customers who differ widely in their expected purchase behaviors.

The core of our framework is the specification of the purchase propensity, $\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$. We treat $\alpha(.)$ as a latent function and model it nonparametrically using Gaussian process priors (Rasmussen and Williams, 2006; Roberts et al., 2013). The nonparametric approach models random functions flexibly and allows us to automatically accommodate different patterns of spend dynamics that may underlie a given customer base. These dynamics operate along all four of our dimensions. Furthermore, these dynamics may operate at different time scales within a single dimension, including smooth long-run trends and short-term patterns, as well as cyclic variation, which are inferred from the data. To allow such rich structure, we use an additive combination of unidimensional GPs to specify and estimate the multivariate function $\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$.

2.1 Gaussian Process Priors

We begin by describing GPs and highlight how they can nonparametrically capture rich, dynamic patterns in a Bayesian probability model. A Gaussian process is a stochastic process $\{f(\tau) : \tau \in \mathcal{T}\}$ indexed by input elements τ such that, for any finite set of input values, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_M\}$, the corresponding set of function outputs, $f(\boldsymbol{\tau}) = \{f(\tau_1), f(\tau_2), \ldots, f(\tau_M)\}$, follows a multivariate Gaussian distribution. The characteristics of the stochastic process are defined by a mean function and a covariance function, also called a kernel. For a fixed set of inputs, a Gaussian Process reduces to the familiar multivariate Gaussian distribution, with a mean vector determined by the GP's mean function, and a covariance matrix determined by its kernel. However, unlike a standard multivariate normal distribution that is defined over vectors of fixed length, a Gaussian process defines a distribution over outputs for any possible set of inputs. From a Bayesian perspective, this provides a natural mechanism for probabilistically specifying uncertainty over functions. Since the estimated function values are the parameters of a GP, the number of parameters grows with the number of unique inputs, making the model nonparametric.

While GPs are often defined over multidimensional inputs, for simplicity of exposition, we begin by assuming a unidimensional input, $\tau \in \mathbb{R}$ (e.g., time). To fix notation, suppose

f is a function that depends on that input. Let $\boldsymbol{\tau}$ be a vector of M input points, and let $f(\boldsymbol{\tau})$ be the corresponding vector of output function values. As described above, a GP prior over f is completely specified by a mean function, $m(\tau) = \mathrm{E}[f(\tau)]$, and a kernel, $k(\tau, \tau') = \mathrm{Cov}[f(\tau), f(\tau')]$, that defines a positive semidefinite covariance matrix

$$K(\boldsymbol{\tau}, \boldsymbol{\tau}) = \begin{pmatrix} k(\tau_1, \tau_1) & k(\tau_1, \tau_2) & \dots & k(\tau_1, \tau_M) \\ k(\tau_2, \tau_1) & k(\tau_2, \tau_2) & \dots & k(\tau_2, \tau_M) \\ \vdots & \vdots & \ddots & \vdots \\ k(\tau_M, \tau_1) & k(\tau_M, \tau_2) & \dots & k(\tau_M, \tau_M) \end{pmatrix}, \tag{2}$$

over all the outputs. We discuss specific forms of the mean function and kernel in Sections 2.1.1 and 2.1.2. Generally, these functions are governed by a small set of hyperparameters that embody certain traits of the GP. For instance, the squared exponential kernel, which we discuss in considerable detail in Section 2.2.2, is given by $k_{\mathrm{SE}}(\tau_i, \tau_j) = \eta^2 \exp\{-(\tau_i - \tau_j)^2/(2\rho^2)\}$. This form encodes the idea that nearby inputs should have related outputs through two hyperparameters: an amplitude, η, and a smoothness, ρ. Intuitively, these two hyperparameters determine the traits of the function space being modeled by a GP with this kernel.

Given a fixed vector of inputs $\boldsymbol{\tau}$, letting $f(\boldsymbol{\tau}) \sim \mathcal{GP}(m(\boldsymbol{\tau}), k(\boldsymbol{\tau}, \boldsymbol{\tau}'))$ is equivalent to modeling the vector of function outputs via a marginal multivariate Gaussian $f(\boldsymbol{\tau}) \sim \mathcal{N}(m(\boldsymbol{\tau}), K(\boldsymbol{\tau}, \boldsymbol{\tau}))$. The mean $m(\boldsymbol{\tau})$ and covariance matrix $K(\boldsymbol{\tau}, \boldsymbol{\tau})$ of the above multivariate normal marginal distribution are again parsimoniously determined through the small set of hyperparameters underlying the mean function and kernel of the GP. The fact that the marginal of a GP is a multivariate normal distribution makes it easy to comprehend how function interpolation and extrapolation work in this framework. Conditioned on an estimate for the function values at the observed inputs, and on the mean function and kernel hyperparameters, the output values for the latent function f for some new input points $\boldsymbol{\tau}^*$ can be predicted using the conditional distribution of a multivariate normal. Specifically, the joint distribution of the old and new function values is given by

$$\begin{bmatrix} f(\boldsymbol{\tau}) \\ f(\boldsymbol{\tau}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{\tau}) \\ m(\boldsymbol{\tau}^*) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{\tau}, \boldsymbol{\tau}) & K(\boldsymbol{\tau}, \boldsymbol{\tau}^*) \\ K(\boldsymbol{\tau}^*, \boldsymbol{\tau}) & K(\boldsymbol{\tau}^*, \boldsymbol{\tau}^*) \end{bmatrix} \right), \tag{3}$$

and hence the conditional distribution of the new outputs can be written as

$$\begin{aligned} f(\boldsymbol{\tau}^*) \sim \mathcal{N}(&m(\boldsymbol{\tau}^*) + K(\boldsymbol{\tau}^*, \boldsymbol{\tau}) K(\boldsymbol{\tau}, \boldsymbol{\tau})^{-1} [f(\boldsymbol{\tau}) - m(\boldsymbol{\tau})], \\ &K(\boldsymbol{\tau}^*, \boldsymbol{\tau}^*) - K(\boldsymbol{\tau}^*, \boldsymbol{\tau}) K(\boldsymbol{\tau}, \boldsymbol{\tau})^{-1} K(\boldsymbol{\tau}, \boldsymbol{\tau}^*)). \end{aligned} \tag{4}$$

This equation again makes clear that the kernel and mean functions determine the distribution of the output values both for existing and new inputs. As the mean and covariance of the marginal multivariate normal are parametrized via the mean and kernel functions, the GP remains parsimonious, and can interpolate and extrapolate seamlessly for any set of input values. The choice of mean function allows us to model different a priori expected functional forms, while the kernel determines how much the functions deviate nonparametrically from that mean function.

2.1.1 Mean Functions The mean function captures expected functional behaviors. Within the range of observed inputs, the mean function often has very little influence over the estimated function values; instead, the properties of the estimated function are largely determined by the kernel, as we describe in the next section. Because of this, in many GP applications, the mean function is set to a constant, reflecting no prior assumptions about functional form. However, far from the range of observed inputs, the posterior expected function values revert to the mean function.[2] In some applications, this mean reverting behavior in combination with a constant mean function is problematic, as we may expect the function values to be increasing or decreasing, both in and out of the range of inputs. To capture this expected behavior, we may choose to use a non-constant mean function.

In this paper, we use either a constant mean function, or a parametric monotonic power mean function, given by $m(\tau) = \lambda_1(\tau - 1)^{\lambda_2},\ \lambda_2 > 0$. This specification captures expected monotonic behavior, while also allowing for a decreasing marginal effect over the input.[3] We use $(\tau - 1)$ and restrict $\lambda_2 > 0$, to be consistent with our identification restrictions that we describe later. We emphasize again that the mean function sets an expectation over function values, but does not restrict them significantly. The GP structure allows functions to nonparametrically deviate from the mean function, resulting in function estimates that differ from the mean's parametric form. This is obvious in all panels of Figure 1, where we plot random draws from GPs with different mean functions and kernels. Across the panels of Figure 1, we see shapes that are sometimes dramatically different from the respective constant and power mean functions that generated them. The main role of the mean function is in extrapolating far from the range of the observed inputs, where it determines expected function behavior in the absence of data. While we use only these two mean functions as a simple way of capturing our prior expections, any parametric form could be potentially used as a mean function. Given the capacity of the GP to capture deviations from parametric forms, it is generally considered best practice to use simple mean functions, and let the GP capture any complexities.

2.1.2 Kernels The kernel defines much of the fundamental structure of a GP, and in combination with the mean function, determines the latent function space of a GP prior. As such, kernels are the primary source of model specification when working with GP priors. Any function over two inputs that results in a positive semidefinite gram matrix can be used as a kernel, and many different kernel forms have been explored in the GP literature (Rasmussen and Williams, 2006, Chapter 4). Kernels encode the structure of functions via a small number of hyperparameters, leading to highly flexible yet parsimonious model specification. In this paper, we use two simple kernels that are suitable building blocks for describing functions in our context.

[2]This behavior can be seen through Equation 4, in conjunction with, for example, the squared exponential kernel, briefly mentioned above, which has functional form $k_{\text{SE}}(\tau_i, \tau_j) = \eta^2 \exp\{-(\tau_i - \tau_j)^2/(2\rho^2)\}$. As the distance between the observed inputs and the new input grows, the value of the kernel goes to zero, and we see the mean in Equation 4 will revert to the mean function. This mean reverting property is dependent on the kernel being stationary, meaning that it depends only on the distance between inputs. We refer the interested reader to Rasmussen and Williams (2006), for a comprehensive discussion of these issues.

[3]We note that the properties of this specification are suitable for our specific application, but may not be suitable in other domains and substantive applications.

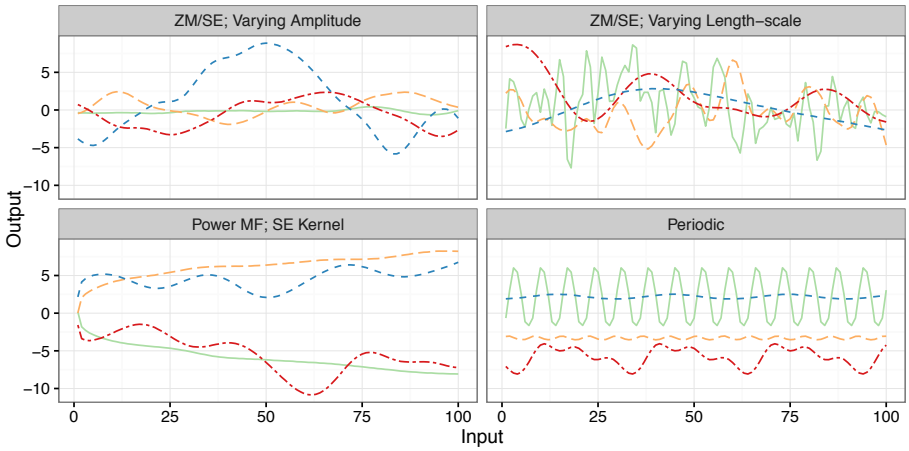

Figure 1: Examples of mean function/kernel combinations. Top-left: zero mean function and SE kernel with $\rho^2 = 50$ and $\eta^2 \in \{0.1, 1, 5, 20\}$; Top-right: zero mean function and SE kernel with $\rho^2 \in \{1, 10, 100, 1000\}$; Bottom-left: power mean function $m(\tau) = \pm 2(\tau - 1)^{0.3}$ and SE kernel with $\rho^2 = 100$ and $\eta^2 \in \{0.1, 5\}$; Bottom-right: periodic kernels with $\eta^2 = 10$, $\rho^2 \in \{2, 100\}$, and $\omega \in \{7, 30\}$.

The first kernel is the squared exponential kernel (SE) defined as

$$k_{\text{SE}}(\tau_j, \tau_k; \eta, \rho) = \eta^2 \exp\left\{-\frac{(\tau_j - \tau_k)^2}{2\rho^2}\right\}, \tag{5}$$

where the hyperparameter $\eta > 0$ is the amplitude, and $\rho > 0$ is the characteristic length-scale or "smoothness." The amplitude can be best explained by considering the case when $\tau_j = \tau_k \equiv \tau$. In this case, $k(\tau, \tau) = \eta^2$, which is the variance of the normal distribution at the fixed input value τ. More generally, η^2 captures variance around the mean function. If $\eta \to 0$, the GP will largely mirror its mean function. We illustrate this using both the constant and power mean functions in the left column of Figure 1, where we randomly draw GPs with a fixed ρ and varying η values. From these two panels, we can see that small values of η, as in the light-colored solid (green) and long-dash (yellow) curves, yield functions that stay closer to their mean functions, relative to the dark-colored dot-dash (red) and short-dash (blue) curves with higher η values. The characteristic length-scale ρ intuitively indicates how far apart two input points need to be for the corresponding outputs to be uncorrelated. Hence, a high value of ρ corresponds to very smooth functions, while a small value of ρ yields jagged, unpredictable functions. We see this illustrated in the top-right panel of Figure 1, where we fix the amplitude η and vary the length-scale ρ. We can see a clear contrast between the highly jagged solid (green) curve with $\rho^2 = 1$, and the increasingly smooth dashed curves, with $\rho^2 \in \{10, 100, 1000\}$.

The second kernel we use is the periodic kernel, defined by

$$k_{\text{Per}}(\tau_j, \tau_k; \omega, \eta, \rho) = \eta^2 \exp\left\{-\frac{\sin^2\left(\pi(\tau_j - \tau_k)^2/\omega\right)}{\rho^2}\right\}. \tag{6}$$

This kernel allows for periodic functions with period ω that are again defined by an amplitude η and a length-scale ρ. Note that this type of variability could also be captured

by the squared exponential kernel; the benefit of using the periodic kernel is that forecasts based on this kernel will always precisely mirror the estimated pattern. Hence, any predictable cyclic variability in the data would be captured both in and out-of-sample. In the bottom-right panel of Figure 1, we plot four draws from different periodic kernels. There, we show different cycle lengths (30 days and 7 days), together with differing smoothness and amplitude parameters.

Other Possible Kernels In addition to the above described kernels, many other types have been proposed in the GP literature. In this paper, we use the simplest kernels that exemplify a given trait (stationary variability with the SE and cyclicality with the periodic). These are by far the most commonly used kernels, the squared exponential especially serving as the workhorse kernel for the bulk of the GP literature. Additional kernels include the rational quadratic, which can be derived as an infinite mixture of squared exponential kernels, and the large class of Matern kernels, which can capture different levels of differentiability in function draws.

2.1.3 Additivity Just as the sum of Gaussian variates is distributed Gaussian, the sum of GPs is also a GP, with a mean function equal to the sum of the mean functions of the component GPs, and its kernel equal to the sum of the constituent kernels. This is called the additivity property of GPs, and can allow us to define a rich structure even along a single dimensional input. Specifically, the additivity property allows us to model the latent function f as a sum of sub-functions on the same input space, $f(\tau) = f_1(\tau) + f_2(\tau) + \ldots + f_J(\tau)$, where each of these sub-functions can have its own mean function, $m_j(\tau)$, and kernel, $k_j(\tau, \tau')$. The mean function and kernel of the function f are then given by $m(\tau) = \sum_{j=1}^{J} m_j(\tau)$ and $k(\tau, \tau') = \sum_{j=1}^{J} k_j(\tau, \tau')$, respectively. This allows us to flexibly represent complex patterns of dynamics even when using simple kernels like the squared exponential. We can, for example, allow the different sub-functions to have different squared exponential kernels that capture variability along different length-scales, or add a periodic kernel to isolate predictable cyclic variability of a given cycle length. It is through this additive mechanism that we represent long-run and short-run variability in a given dimension, for instance, or isolate predictable periodic effects from unpredictable noise, as we discuss in Section 2.2.[4] Until now, we have focused on illustrating GPs in unidimensional contexts. We now show how additivity can be leveraged to construct GPs for multidimensional functions.

2.1.4 Multidimensional GPs In practice, we are often interested in estimating a multidimensional function, such as the $\alpha(.)$ function in Equation 1. Let $h(.)$ be a generic

[4]In general, determining the number of additive components suitable for a given application requires both substantive knowledge and expectations about the nature of the dynamics at work, and data-driven evidence from the estimated hyperparameter values. For instance, depending on the kernel, a small amplitude hyperparameter compared to the output scale could indicate the component is relatively uninfluential in describing the results. Similarly, if the length-scale is estimated to be very large, this can indicate minimal dynamics are being uncovered by that component. Both of these phenomena can indicate redundancy in the specification. Kernel specification is a rich topic in the GP literature, and the interested reader can find considerable discussion in Rasmussen and Williams (2006), Chapter 5.

multidimensional function from $\mathbb{R}^D$ to $\mathbb{R}$. The inputs to such a function are vectors of the form $\boldsymbol{\tau}_m \equiv (\tau_m^{(1)}, \tau_m^{(2)}, \ldots, \tau_m^{(D)}) \in \mathbb{R}^D$, for $m = 1, \ldots, M$, such that the set of all inputs is an $M \times D$ matrix. Just as before, $h(.)$ can also be modeled via a GP prior. While there are many ways in which multi-input functions can be modeled via GPs, a simple yet powerful approach is to consider $h(.)$ as a sum of single input functions, $h_1(.), h_2(.), \ldots, h_D(.)$, and model each of these unidimensional functions as a unidimensional GP with its own mean function and kernel structure (Duvenaud et al., 2013). The additivity property implies that additively combining a set of unidimensional GP's over each dimension of the function is equivalent to using a particular sum kernel GP on the whole, multidimensional function. We use such an additive structure to model $\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$ in the GPPM.

Additively separable GPs offer many benefits: first, they allows us to easily understand patterns along a given dimension, and they facilitate visualization, as the sub-functions are unidimensional. Second, the additivity property implies that the combined stochastic process is also a GP. Finally, the separable structure reduces computational complexity. Estimating a GP involves inverting its kernel matrix. This inversion requires $O(M^3)$ computational time and $O(M^2)$ storage demands for M inputs. In our case, as the inputs $(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$ can only exist on a grid of fixed values, we will have $L < M$ inputs, where L corresponds to all unique observed $(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$ combinations. Despite the reduction, this is a very large number of inputs, and would result in considerable computational complexity, without the separable structure. The additive specification reduces this computational burden to that of inverting multiple (in our case, six) $T \times T$ matrices, where $T \ll M$ is the number of time periods observed in the data.

2.1.5 GPs Versus Other Function Estimation Methods As Gaussian process priors are new to marketing, it is worthwhile to briefly summarize the rationale for using them, instead of other flexible methods for modeling latent functions like simple fixed effects, splines, or state space models. Foremost, GPs allow for a structured decomposition of a single process into several subprocesses via the additivity property. This additive formulation facilitates a rich representation of a dynamic process via a series of kernels that can capture patterns of different forms (e.g., periodic vs. non-periodic) and operate at different time scales. Yet, as the sum of GPs is a GP, the specification remains identified, with a particular mean and covariance kernel. Achieving a similar representation with other methods is either infeasible or more difficult.[5] Moreover, GPs are relatively parsimonious, and when estimated in a Bayesian framework, tend to avoid overfitting. Bayesian estimation of GPs involves estimating the function values and hyperparameters jointly, thus determining both the traits of the function, and the function values themselves. As the flexibility of the latent functions is controlled via a small number of hyperparameters, we retain parsimony. Moreover, the structure of the marginal likelihood of GPs, obtained by integrating out

[5]While we emphasize the relative benefits of GP priors here, we also note that there are many links between these methods, including between GP methods and smoothing splines (Kalyanam and Shively (1998) and Shively et al. (2000)), and between GP methods and state space models. We include a sophisticated state space analog of our model in our benchmarks. Our state space formulation is also closely related to cubic spline specifications (see Durbin and Koopman (2012) for details). As we will describe later, although this method produces fits that are roughly on par with the GP approach, we cannot easily obtain the decompositions that are natural in the GP setting.

the function values, clearly shows how the model makes an implicit fit versus complexity tradeoff whereby function flexibility, as captured by the hyperparameters, is balanced by a penalty that results in the regularization of the fit (for details, see Rasmussen and Williams (2006), Section 5.4.1).

2.2 Full Model Specification

The flexibility afforded by GP priors makes them especially appropriate for modeling our latent, time-varying function, $\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$. Recall that the basic form of the GPPM is:

$$\Pr(y_{ij} = 1) = \text{logit}^{-1}\left[\alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij}) + \mathbf{z}_i'\boldsymbol{\gamma} + \delta_i\right]. \tag{7}$$

For ease of exposition, we will subsequently omit the ij subscripts. For simplicity and to reduce computational complexity, we assume an additive structure,

$$\alpha(t, r, \ell, q) = \alpha_{\text{T}}(t) + \alpha_{\text{R}}(r) + \alpha_{\text{L}}(\ell) + \alpha_{\text{Q}}(q), \tag{8}$$

and model each of these functions using separate GP priors. This structure and the nonlinear nature of the model implies an interaction between the effects: for example, if the recency effect is very negative, calendar time events can do little to alter the spend probability. While additivity is a simplifying assumption, in our application, this compensatory structure seems to explain the data well.

To specify each of these additive components, we return to the mean functions and kernels outlined in Sections 2.1.1 and 2.1.2, and to the additivity property of GPs from Section 2.1.3. Recall that the mean function encodes the expected functional behavior: with the constant mean function, we impose no expectations; with the power mean function, we encode expected monotonicity. The kernel choice endows the GP with additional properties: a single SE kernel allows flexible variation with one characteristic length-scale, while the periodic kernel allows the GP to exhibit predictable cyclic behavior with a given period. Additivity allows us to combine these kernel properties, to achieve variation along more than one length-scale, or to isolate predictable cyclic behavior in a given dimension. We can use these general traits of mean function and kernel combinations to specify our model, based on the expected nature of the variation along a given dimenion. Below, we explain the specification used in our application. The GPPM framework is highly flexible, and throughout the following sections, we also explain how this specification can be modified to handle more general settings.

Calendar Time In calendar time, we expect two effects to operate: long run trends, and short run disturbances. These short run events could include promotions, holidays, or other shocks to the purchasing process. Furthermore, we expect cyclicality such that purchasing could be higher on weekends than on weekdays, or in particular months or seasons. As we describe later, in our application, given the span of our data, we expect only one periodic day of the week (DoW) effect. Together, this description of spend dynamics implies a decomposition of α_{T} into three sub-components,

$$\alpha_{\text{T}}(t) = \alpha_{\text{T}}^{\text{Long}}(t) + \alpha_{\text{T}}^{\text{Short}}(t) + \alpha_{\text{T}}^{\text{DoW}}(t), \tag{9}$$

where we model each component such that,

$$\alpha_{\mathrm{T}}^{\mathrm{Long}}(t) \sim \mathcal{GP}(\mu, k_{\mathrm{SE}}(t, t'; \eta_{\mathrm{TL}}, \rho_{\mathrm{TL}})),$$
$$\alpha_{\mathrm{T}}^{\mathrm{Short}}(t) \sim \mathcal{GP}(0, k_{\mathrm{SE}}(t, t'; \eta_{\mathrm{TS}}, \rho_{\mathrm{TS}})),$$
$$\alpha_{\mathrm{T}}^{\mathrm{DoW}}(t) \sim \mathcal{GP}(0, k_{\mathrm{Per}}(t, t'; \omega = 7, \eta_{\mathrm{TW}}, \rho_{\mathrm{TW}})).$$

Without loss of generality, we impose $\rho_{\mathrm{TL}} > \rho_{\mathrm{TS}}$, to ensure that the long-run component captures smoother variation than the short-run component. We use constant mean functions here because, a priori, we do not wish to impose any assumptions about calendar time behavior. The constant mean μ in the long-run component captures the base spending rate in the model. Far from the range of the data, this specification implies the posterior mean of these effects will revert to this base spending rate, reflecting our lack of a priori knowledge about these effects.

This specification is very general, and has shown good performance in our application, where we illustrate the kinds of trends and disturbances that can be captured across these two components.[6] Furthermore, the modularity of the additive GP specification allows easy modifications to accommodate different settings. Longer spans of data may contain variation in spending along different length-scales, which may require additional SE components. There may also be several periodicities requiring additional periodic components. These can be easily included additively.

Individual-level Effects The remaining effects—recency, lifetime, and purchase number—operate at the customer-level. In most applications, we do not expect short-run shocks along these inputs. We do, however, expect monotonicity. For instance, intuitively, we expect spend probability to be generally decreasing in interpurchase time. Similarly, we expect spend probability to be generally increasing in purchase number,[7] and to be generally decreasing in customer lifetime. Furthermore, while we expect monotonicity, we also expect a decreasing marginal effect. For example, we expect a priori that the difference between having spent 5 versus 10 days ago is quite different than the difference between having spent 95 versus 100 days ago. Together, these expected traits justify using our power mean function:

$$\alpha_{\mathrm{R}}(r) \sim \mathcal{GP}(\lambda_{\mathrm{R1}}(r-1)^{\lambda_{\mathrm{R2}}}, k_{\mathrm{SE}}(r, r'; \eta_{\mathrm{R}}, \rho_{\mathrm{R}})),$$
$$\alpha_{\mathrm{L}}(\ell) \sim \mathcal{GP}(\lambda_{\mathrm{L1}}(r-1)^{\lambda_{\mathrm{L2}}}, k_{\mathrm{SE}}(\ell, \ell'; \eta_{\mathrm{L}}, \rho_{\mathrm{L}})),$$
$$\alpha_{\mathrm{Q}}(q) \sim \mathcal{GP}(\lambda_{\mathrm{Q1}}(r-1)^{\lambda_{\mathrm{Q2}}}, k_{\mathrm{SE}}(r, r'; \eta_{\mathrm{Q}}, \rho_{\mathrm{Q}})).$$

Again, this specification allows for long-run monotonic behavior, even out-of-sample, as captured by the mean function, and for nonparametric deviations from this expected functional form, as captured by the SE kernel. We believe that this specification is very general and widely applicable. In some cases, however, more nuance may be required in specifying these

[6] We also include simulated data examples of these effects in Web Appendix B, where we know the effects true forms, and can show that the GPPM is capable of accurately recovering them.

[7] We may not expect this in our application area, freemium video games, where there can be decreasing returns to repeat purchasing.

effects to accommodate company actions that occur on these time scales. If, for instance, the company offers promotions based on loyalty, these effects will operate along the lifetime dimension. In that case, the lifetime component can be modeled similarly to the calendar time component, with an additive SE component to capture these short-run deviations from the long-run, decreasing trend embodied in the above specification. We include an example of this modification in Web Appendix B.

Heterogeneity, Random Effects, and Priors We accommodate unobserved heterogeneity by assuming that the random effect δ_i comes from a normal population distribution, i.e., $\delta_i \sim \mathcal{N}(0, \sigma^2)$. In our application, we found no significant time-invariant effects $\mathbf{z}_i$, and hence we omit $\mathbf{z}_i'\boldsymbol{\gamma}$ from our model going forward. We estimate the model in a fully Bayesian fashion, and therefore specify priors over all unknowns, including the GP hyperparameters. We use the fact that meaningful variation in the inverse logit function occurs for inputs between -6 and 6, and hence meaningful differences in the inputs to the GPPM will also occur between -6 and 6, to select proper weakly informative Normal and Half-Normal prior distributions that give weight to variation in this range. Thus, we let the population variance $\sigma^2 \sim$ Half-Normal$(0, 2.5)$ and the base spending rate $\mu \sim \mathcal{N}(0, 5)$. For the squared exponential hyperparameters, we specify $\eta^2 \sim$ Half-Normal$(0, 5)$ and $\rho^2 \sim$ Half-Normal$(T/2, T)$. For the mean function, we let $\lambda_1 \sim \mathcal{N}(0, 5)$, and let $\lambda_2 \sim$ Half-Normal$(0, 5)$. Importantly, the fully Bayesian approach, whereby both the GP function values and their associated hyperparameters are estimated from the data, allows us to automatically infer the nature of the latent functions that drive spend propensity.

Identification We need to impose identification restrictions because of the additive structure of our model. Sums of two latent functions, such as $\alpha_1(t) + \alpha_2(t)$, are indistinguishable from $\alpha_1^*(t) + \alpha_2^*(t)$, where $\alpha_1^*(t) = \alpha_1(t) + c$, and $\alpha_2^*(t) = \alpha_2(t) - c$ for some $c \in \mathbb{R}$, as both sums imply the same purchase probabilities. To address this indeterminacy, we set the initial function value (corresponding to input $\tau = 1$) to zero for all of the latent functions, except for $\alpha_{\mathrm{T}}^{\mathrm{Long}}(t)$. In this sense, $\alpha_{\mathrm{T}}^{\mathrm{Long}}(t)$, with its constant mean function μ, captures the base spending rate for new customers, and the other components capture deviations from that, as time progresses. Whenever we implement a sum of squared exponential kernels, as in the calendar time component, we also constrain the length-scale parameters to be ordered to prevent label switching. All of these constraints are easily incorporated in our estimation algorithm, described below.

2.3 Estimation

We use a fully Bayesian approach for inference. For concision, let $\alpha_{ij} \equiv \alpha(t_{ij}, r_{ij}, \ell_{ij}, q_{ij})$, which in our specification, is equivalent to $\alpha_{ij} = \alpha_{\mathrm{T}}^{\mathrm{Long}}(t_{ij}) + \alpha_{\mathrm{T}}^{\mathrm{Short}}(t_{ij}) + \alpha_{\mathrm{T}}^{\mathrm{DoW}}(t_{ij}) + \alpha_{\mathrm{R}}(r_{ij}) + \alpha_{\mathrm{L}}(\ell_{ij}) + \alpha_{\mathrm{Q}}(q_{ij})$. To further simplify notation, we let the independent components of the sum be indexed by k, with generic inputs τ_k, such that this GP sum can be written as $\alpha_{ij} = \sum_{k=1}^{K} \alpha_k(\tau_{k_{ij}})$. Each of these components is governed by a set of hyperparameters, as outlined in the previous section, denoted here as $\boldsymbol{\phi}_k$, with the collection of

all hyperparameters denoted $\boldsymbol{\phi}$. Finally, for each component, we let the vector of function values over all possible inputs along that dimension be denoted as $\boldsymbol{\alpha}_k$. With this simplified notation, the joint density of the data and the model unknowns is:

$$p(\boldsymbol{y}, \{\boldsymbol{\alpha}_k\}, \boldsymbol{\delta}, \boldsymbol{\phi}, \sigma^2) = \left[\prod_{i=1}^{I}\prod_{j=1}^{M_i} p(y_{ij}|\alpha_{ij}, \delta_i)p(\delta_i|\sigma^2)\right]\left[\prod_{k=1}^{K} p(\boldsymbol{\alpha}_k|\boldsymbol{\phi}_k)\right]p(\sigma^2)p(\boldsymbol{\phi}). \tag{10}$$

As the full posterior distribution $p(\{\boldsymbol{\alpha}_k\}, \boldsymbol{\delta}, \boldsymbol{\phi}, \sigma^2|\boldsymbol{y})$ is not available analytically, we use Markov Chain Monte Carlo Methods (MCMC) to draw samples of the unknown function values, random effects, population parameters, and GP hyperparameters from the posterior.

As the function values and the hyperparameters do not have closed-form full conditionals, our setup is non-conjugate, and Gibbs sampling is not an option. Moreover, as the function values and the hyperparameters typically exhibit strong posterior dependence, ordinary Metropolis-Hastings procedures that explore the posterior via a random walk are not efficient. We therefore use the Hamiltonian Monte Carlo (HMC) algorithm that leverages the gradient of the posterior to direct the exploration of the Markov chain to avoid random-walk behavior. HMC methods are ideal for non-conjugate GP settings such as ours, as they can efficiently sample both the latent function values as well as the hyperparameters (Neal, 1998). In particular, we use the No U-Turn Sampling (NUTS) variant of HMC as implemented in the Stan probabilistic programming language (Hoffman and Gelman, 2014; Carpenter et al., 2016). We include an overview of HMC in Web Appendix A.

Stan has recently gained traction as an efficient and easy-to-use probabilistic programming tool for Bayesian modeling. We use Stan as it is an efficient implementation of adaptive HMC. Stan programs are simple to write and modify, and therefore facilitate easy experimentation, without the need for extensive reprogramming. This is important for the wider adoption of this framework in practice.[8] Finally, given the efficiency of HMC and Stan, convergence, as measured by the $\hat{R}$ statistic (Gelman and Rubin, 1992), is achieved in as few as 400 iterations, although in this paper all estimation is done with 4,000 iterations with the first 2,000 used for burn-in.

3 Application

We apply our framework to understand the spending dynamics in two free-to-play mobile games from one of the world's largest video game companies. The data take the form of simple spend incidence logs, with user IDs and time stamps.[9] In free-to-play (or "freemium") settings, users can install and play video games on their mobile devices for free, and are offered opportunities to purchase within the game. These spend opportunities typically involve purchasing in-game currency, like coins, that may subsequently be used to progress more quickly through a game, obtain rare or limited edition items to use with their in-game

[8] We include our Stan code in Web Appendix C.

[9] There is no personally identifiable information in our data; player information is masked such that none of the data we use or the results we report can be traced back to the actual individuals. We also mask the identification of the company as per their request.

characters, or to otherwise gain a competitive edge over non-paying players. Clearly, the nature of these purchases will depend on the game, which is why it is important for a model of spending behavior to be fully flexible in its specification of the regular, underlying drivers of purchasing. We cannot name the games here because of non-disclosure agreements. Instead, we use the general descriptors Life Simulator (LS) and City Builder (CB) to describe the games.

The games and ranges of data used were selected by our data provider, in an effort to understand spend dynamics over specific periods of time. We use a random sample of 10,000 users for each of the two games. Each sample is drawn from users who installed the game within the first 30 days, and spent at least once during the training window. We used 8,000 users for estimation, and 2,000 for cross validation. In the Life Simulator (LS) game, players create an avatar, then live a digital life as that avatar. Purchases in this context can be rare or limited edition items to decorate or improve their avatar or its surroundings. Often times, limited edition items are themed according to holidays such as Christmas or Halloween. Our data come from a 100 day span of time covering the 2014 Christmas and New Year season. In the City Builder (CB) game, players can create (or destroy) a city as they see fit. Customers make purchases to either speed up the building process or to build unique or limited edition additions to their cities. Our data come from an 80 day period of time at the start of 2015, at the tail end of the Christmas and New Year holidays.

The time series of spending for the two games are shown in Figure 2. We have also marked specific time periods of interest to the company, which we will discuss in more detail in our analysis. From these figures, it is difficult to parse out what exactly is driving the aggregate pattern of purchases. The figure includes customers who installed the game any time within the first 30 day window. Typically, customers are most active when they start playing a game, so we expect to see more spending in the first 30-40 days simply because there are likely more people playing in that period, and new players are entering the pool of possible spenders. This rise and subsequent fall is, in essence, the joint impact of the recency, lifetime, and purchase number effects. We see, however, that even the general rise-fall pattern varies across the two games. This could be due to different patterns in these underlying drivers of spending, or it could be due to the influence of calendar time events. In essence, it is unclear what else underlies the aggregate spends.

We also see many peaks and valleys in spending over the entire time horizon, the significance of which cannot be diagnosed without deeper analysis. For example, it is difficult to discern which "bumps" in the plots are meaningful, and which represent random noise. If 5,000 players are active at any given day, then a jump of 50 spends in may represent a random fluctuation. In contrast, if only 1,000 players are active, the same jump of 50 spends may be very meaningful. In other words, the significance of a particular increase in spending depends on how many customers are still actively spending at that time, which in turn depends on the individual-level recency, lifetime, and purchase number effects. An accurate accounting of the impact of calendar-time events cannot be made without considering these individual-level predictors of spending, and it is thus important to develop a model-based understanding of the underlying spend dynamics, which is what we do via the GPPM.

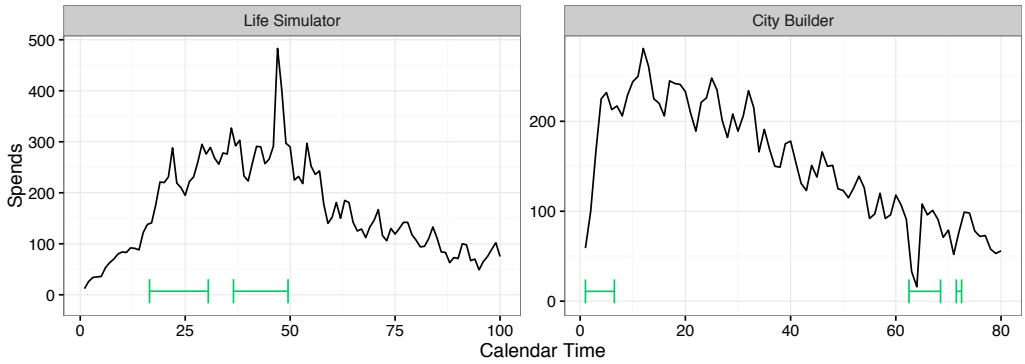

Figure 2: Spend incidence by day (calendar time) in each game. Bars indicate time periods of interest, as specified by the company, and as discussed more in Section 3.2.1.

3.1 Model Output and Fit

The GPPM offers a visual and highly general system for customer base analysis that is driven by nonparametric latent spend propensity functions. These latent curves are the primary parameters of the model, and their posterior estimates are displayed in Figure 3 for LS, and in Figure 4 for CB. We call these figures the GPPM dashboards, as they visually represent latent spend dynamics. As we will see in 3.2, these dashboards can be used to accomplish many of the goals we have discussed throughout the previous sections, including forecasting spending, understanding purchasing at the individual-level, assessing the influence of calendar time events, and comparing spending patterns across products.

These dashboards are underpinned by a set of hyperparameters, and estimated jointly with a random effects distribution capturing unobserved heterogeneity. Posterior medians of these parameters are displayed in Table 1. While the hyperparameters summarize the traits of the estimated dashboard curves, as explained in Section 2.1, we can gain a greater understanding of the dynamics from an analysis of the estimated dashboard curves themselves, as we do in the subsequent sections. The other parameters in Table 1 are the base spending rate, μ, and the population variance of the random effects distribution, σ^2, which reflects the level of heterogeneity in base spend rates estimated in each customer base.

Model Fit First, to validate our model, we look at its fit to the observed daily spending data, both in the calibration sample of 8,000 customers and in the holdout sample of 2,000 customers. A closed-form expression is not available for the expected number of aggregate counts in the GPPM. We therefore simulate spending from the posterior predictive distribution by using the post convergence HMC draws for each parameter, including the latent curves and random effects. The top row of Figure 5 shows the actual spending and the median simulated purchase counts (dashed line) for the two games, along with 95% posterior predictive intervals.

We see that the fit is exceptional, and tracks the actual purchases almost perfectly in both cases. This is not surprising, as we model short-run deviations in the probability of

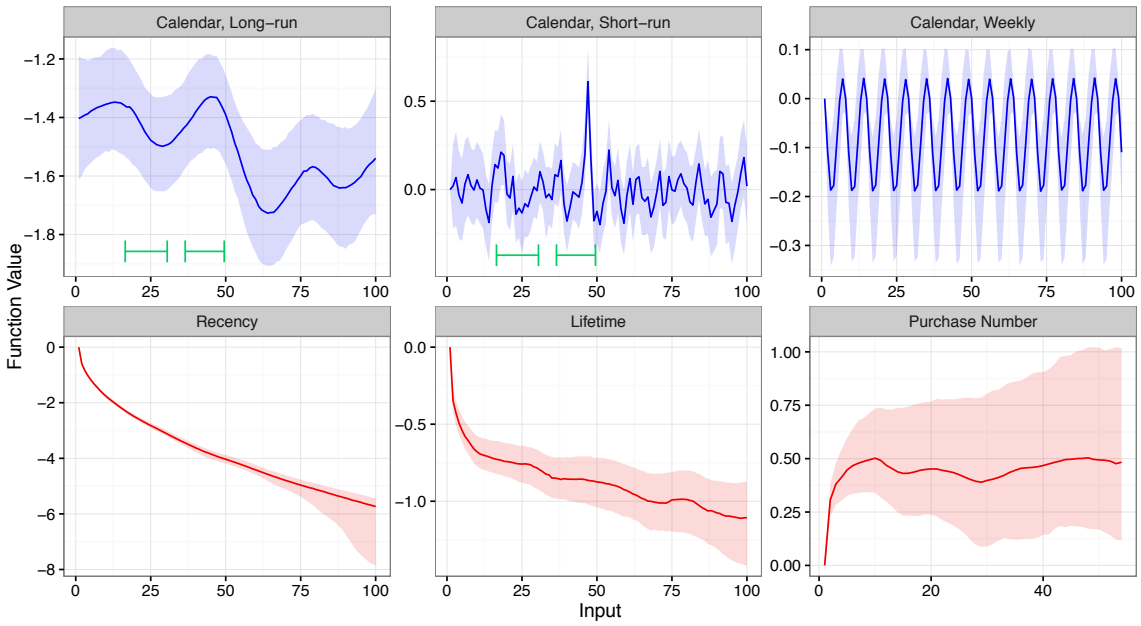

Figure 3: Posterior dashboard for the Life Simulator customer base. Curves are the median posterior estimates for the latent components of $\alpha(t, r, \ell, q)$ with 95% credible intervals. The blue plots (top row) are the calendar time components, while the red (bottom row) are the individual-level effects. The marked time periods (green bars) are areas of interest to the company, as discussed in Section 3.2.1.

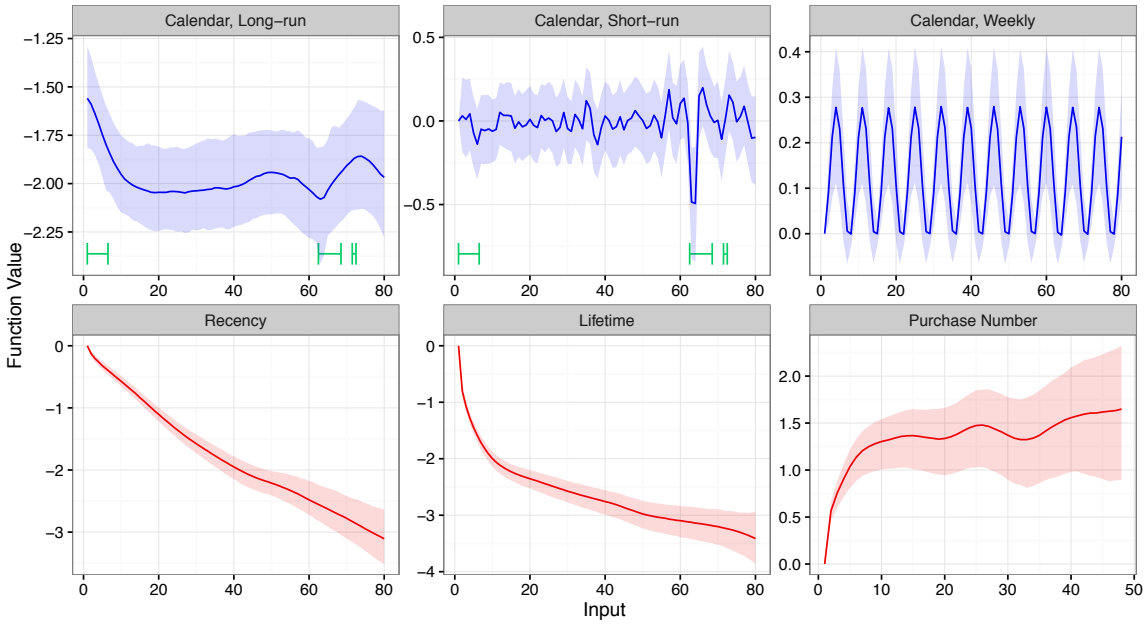

Figure 4: Posterior dashboard for the City Builder customer base. Curves are the median posterior estimates for the latent components of $\alpha(t, r, \ell, q)$ with 95% credible intervals. The blue plots (top row) are the calendar time components, while the red (bottom row) are the individual-level effects. The marked time periods (green bars) are areas of interested to the company, as discuss in Section 3.2.1.

Component		LS	CB	Component		LS	CB
Cal, Long	η_{TL}	0.17	0.22	Lifetime	η_{L}	0.06	0.23
	ρ_{TL}	11.75	10.32		ρ_{L}	9.77	12.25
Cal, Short	η_{TS}	0.15	0.16		λ_{L1}	-0.34	-0.75
	ρ_{TS}	1.11	1.29		λ_{L2}	0.25	0.36
Cal, DoW	η_{TW}	1.08	1.19	Purchase Number	η_{Q}	0.10	0.20
	ρ_{Q}	9.17	9.59		ρ_{Q}	4.93	5.36
Recency	η_{R}	0.04	0.10		λ_{Q1}	0.28	0.52
	ρ_{R}	10.23	11.05		λ_{Q2}	0.15	0.30
	λ_{R1}	-0.59	-0.13	Base Rate	μ	-1.49	-1.92
	λ_{R2}	0.49	0.72	Heterogeneity	σ^2	0.68	0.93

Table 1: *Posterior median parameter estimates for both games.*

spending on a daily basis and therefore essentially capture the residuals from the smoother model components. That is, the short-run calendar time component captures any probability that is "left-over" from the other components of the model, enabling us to fit in-sample data exceptionally well. To test that the model does not overfit the in-sample day-to-day variability, we explore the simulated fit in the validation sample of 2,000 held-out customers. The bottom row of Figure 5 shows that the fit to this sample is still excellent, although not as perfect as in the top row. While the probabilistic residuals from the calibration data are not relevant for the new sample, much of the signal present in the calendar time trends and the individual-level effects continue to matter, thus contributing to the good fit.

Fit Decomposition To better understand how the latent curves in the dashboard contribute to the fits seen in Figure 5, we now break down that fit along our latent dimensions. For that, we focus on the LS game. Our main focus is on assessing how much of the day-to-day spending is explained by the calendar time components of the model versus the typically smoother, individual-level recency, lifetime, and purchase number components. To do that, we examine how the fit changes when different components of the model are muted. We "mute" a component by replacing it with a scalar that is equal to the average of its function values over all its inputs. Note that we do not re-estimate a model when we mute a component; instead, muting allows us to see how much of the overall fit is driven by a given component.

The fit decomposition is shown in Figure 6. Overlaid on the true spending time series, we have three muted fits: in the first, we mute the short-run calendar time component; in the second, we mute both the short and long-run calendar time components; and in the third, we mute all calendar time components. From the continued good fit of the muted models, we can see that the majority of the full model fit is actually driven by the individual-level predictors of spend: recency, lifetime, and purchase number. This finding is largely

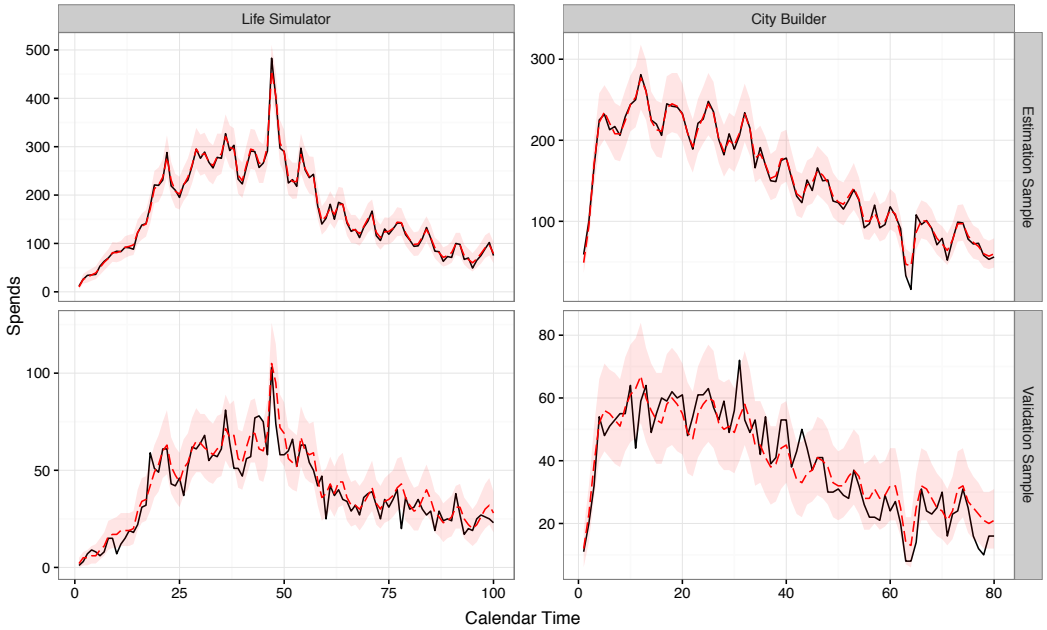

Figure 5: True and simulated spending by day under the GPPM with 95% posterior predictive intervals. The black is the data while the red (dashed) is the median simulated fit. In the top row, we show the fit in the estimation data of 8,000 customers, where the two curves are nearly indistinguishable, while in the bottom row, we show the fit in the validation sample of 2,000 held-out customers.

in keeping with the established literature on customer base analysis, which has robustly shown that models based on these components can do well at fitting and forecasting spend activity. However, we also find that calendar time plays a non-negligible role: while the short-run component generally captures the residuals, as explained before, the long-run component plays an important role in capturing changes in base spending rates over time. Furthermore, the cyclic component, which is a highly predictable yet novel element of our model, plays an important role in explaining day-to-day variability in spending.

3.2 Dashboard Insights

While fit validates the utility of the GPPM, one of the primary motivations of the model is to provide managers with a model-based decision support system that captures effects of interest, and allows for a visual understanding of the drivers of spend behavior. Thus, the key output of our model is the GPPM dashboard (Figures 3 and 4), which portrays the posterior estimates of the latent propensity functions. These latent spend propensity curves are readily interpretable, even by managers with minimal statistical training. We illustrate here the insights that managers can obtain from these model-based visualizations.

3.2.1 Calendar Time Effects Events that happen in calendar time are often of great importance for managers, but their impact is often omitted from customer base analysis models. The GPPM includes these effects nonparametrically through the calendar time

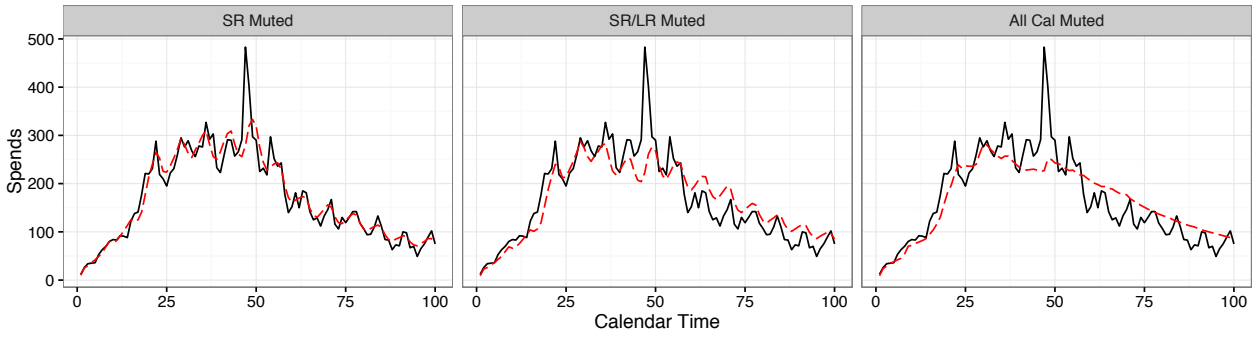

Figure 6: Fit decomposition on the LS spending data. Each panel from left to right represents muting an additional component of the model; the worsening fit shows how much of the full model fit is driven by the muted component.

components of the model, such that impact of calendar time events is captured flexibly and automatically. Calendar time effects are estimated jointly with the individual-level drivers of spending, recency, lifetime, and purchase number. This means the impact of calendar time on propensity to spend is assessed only after controlling for these drivers of respend behavior, which account for the natural ebb and flow of spending, including dynamics in the numbers of active customers.

Importantly, capturing the impact of calendar time events requires no inputs from the marketing analyst, as would be required in a model where time-varying covariates are explicitly specified. This implies that their presence and significance must be evaluated ex post facto. This has many benefits: first, even in the face of information asymmetries or unpredictable shocks, the events will be captured by the GPPM. Second, the shape of the impact of these events is automatically inferred, rather than assumed. Finally, because the impact is captured by changes in the calendar time components of the propensity model, their impact can be assessed visually. We demonstrate the analysis of calendar time events using our two focal games. The top row of plots in each dashboard (colored blue) represents the calendar time effects. From left to right, we have the long-run trends, short-run shocks, and periodic day of the week effects. Beneath these curves, we have placed bars indicating time periods of interest to the company.

Life Simulator Events Two events of note occurred in the span of the data. The first marked time period $t \in [17, 30]$ corresponds to a period in which the company made a game update, introduced a new game theme involving a color change, and also donated all proceeds from the purchases to a charitable organization. The second marked period, around $t \in [37, 49]$, corresponds to another game update that added a Christmas-themed quest to the game, with Christmas itself falling at $t = 48$, right before the end of the holiday quest.

From the dashboard, we learn several things: first, there is a prominent spike in short-run spending the day *before* Christmas. This Christmas Eve effect illustrates that events do not have to be anticipated to be detected in the model, and we illustrate in the subsequent section how the GPPM parses out the impact of short-run events, using this effect as the

example. In the long-run curve, we see a decrease in spending coinciding with the charity update, an increase in spending coinciding with the holiday event, and then a significant drop-off subsequent to the holiday season. Without a longer range of data, it is hard to assess the meaning of these trends. It does appear that the charity event lowered spend rates. The impact of the holidays is more unclear: it could be that the holiday game update elevated spending, and then as time went on, spend levels returned to normal. Alternatively, spend levels could be elevated simply due to the holiday season, with a post-holiday slump that is unrelated to the game updates. Although we cannot conclusively parse out these stories, we can tell that calendar time dynamics are at play, and appear linked to both real world shocks and company actions.

City Builder Events The marked areas of the CB dashboard in Figure 4 again correspond to events of interest. The start of the data window ($t \in [1, 6]$) coincides with the tail end of the holiday season, from December 30 to January 4. Another event begins at $t = 63$, when the company launched a permanent update to the game to encourage repeat spending. We mark five additional days after that update to signify a time period over which significant post-update activity may occur. Finally, at $t = 72$, there was a crash in the app store.

We see, as in the previous game, that the spending level in the holidays ($t \in [1, 6]$) was quite high and fell dramatically subsequently. This lends some credence to a general story of elevated holiday season spending, as there was no game update in CB during this time. Spending over the rest of the time period was relatively stable. The update that was intended to promote repeat spending had an interesting effect: there was an initial drop in spending, most likely caused by reduced playtime on that day because of the need for players to update their game or because of an error in the initial launch of the update. After the update, an uptick in long-run spending is observable, but this was relatively short-lived. Finally, we find no effect for the supposed app store crash, which in theory should have prevented players from purchasing for the duration of the crash. It is plausible that the crash was for a short duration or occurred at a time when players were not playing.

Day of the Week Effects Across both games, we note the significance of the periodic day of the week effect. In both cases, spend propensity varies by day of the week by a magnitude of 0.3. For comparison, the long-run calendar time effect of LS has a range of 0.5, while that of CB has a range of 0.6. The magnitude of the periodic effect serves to re-emphasize a point already made in the fit decomposition: a large amount of the calendar time variability in spending can be attributed to simple predictable cyclic effects, something customer base models have previously ignored, but that can be powerful in forecasting future purchase behavior.

3.2.2 Event Detection Often, calendar time events are unknown a priori, but can significantly affect consumers' spending rates in the short-run. The short-run function is capable of automatically detecting and isolating these disturbances. That is, if something disrupts spending for a day, such as a crash in the payment processing system, or an in-game event, it will be reflected either as a trough or as a spike in the short-run function, as evident for example in the Christmas Eve effect in LS. In this section, we illustrate how

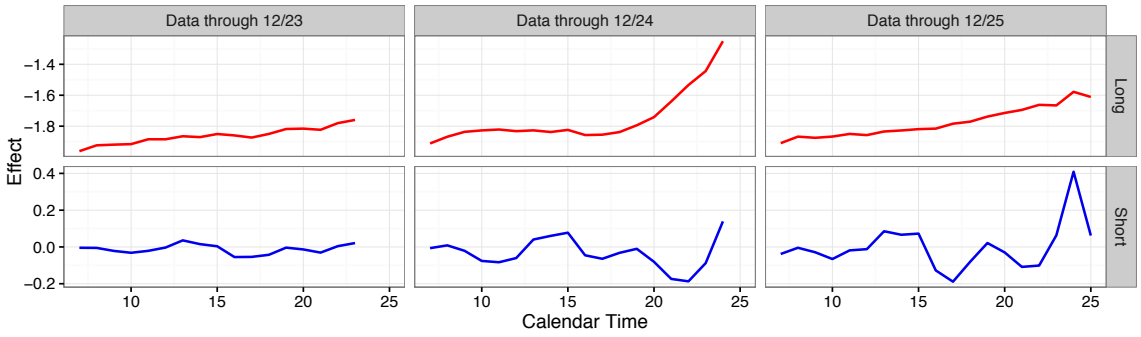

Figure 7: Event detection in the GPPM. From left to right, we add daily data, and see how the impact of Christmas Eve is separated between the long-run (top, red) and short-run (bottom, blue) calendar time curves.

this works in practice.

The GPPM estimation process decomposes the calendar time effect along sub-functions with differing length-scales. As such, when there is a disturbance, the GPPM must learn the relevant time scale for the deviation—here, either short or long-term—and then adjust accordingly. We illustrate this dynamically unfolding adjustment process for the LS Christmas Eve effect in Figure 7 by estimating the model using progressively more data over the range 12/23/2014 to 12/25/2014. The different columns of the figure show how the long-run (top row) and the short-run (bottom row) components vary when data from each successive day is integrated into the analysis. The second column shows the impact of adding the data from Christmas Eve. An uptick in spending is apparent, but the GPPM cannot yet detect whether this uptick will last longer or just fade away. The day after (third column), it becomes clear from looking at the long-run and short-run plots that the effect was only transient, which is reflected clearly in the short-run curve.

This example illustrates that the GPPM can capture effects of interest with no input from the analyst, and that the nature of this effect is visually apparent in the model-based dashboard within days of its occurrence. Note that, importantly, each column of Figure 7 represents a re-estimation of the GPPM, using the past day's data; event detection can only occur at the level of aggregation of the data (in this case, daily), upon re-estimation of the model. Nonetheless, this capability can be immensely valuable to managers in multiproduct firms where information asymmetries abound. For example, in digital contexts, product changes can sometimes be rolled out without the knowledge of the marketing team. Similarly, disruptions in the distribution chain can occur with little information filtering back to marketing managers. The GPPM can capture the impact of such events automatically and quickly, isolate them from the more regular, predictable drivers of spending, and bring them to the attention of managers.

3.2.3 Individual-level Effects While the inclusion of calendar time effects is a key innovation in our model, the primary drivers of respend behavior are the individual-level recency, lifetime, and purchase number effects. We can see this both through the fit de-

composition, where much of the variability in spending is captured even when the calendar time effects are muted, and also by assessing the range of the effects in the dashboard. As mentioned in Section 2.2, the range of relevant inputs in an inverse logit framework is from -6 to 6. For propensity values $\alpha < -6$, the respend probability given by $\text{Logit}^{-1}(\alpha)$ is approximately 0. Similarly, for propensity values $\alpha > 6$, the respend probability is approximately 1. This gives an interpretability to the curves in the dashboard, as their sum determines this propensity, and hence their range determines how much a given component of the model can alter expected respend probability. Relative to the calendar time effects, we can see in the dashboard that the ranges of the individual-level effects are significantly larger, implying that they explain much more of the dynamics in spend propensity than the calendar time components.

Recency and Lifetime In both of our applications, the recency and lifetime effects are smooth and decreasing as expected. For managers, this simply means that the longer someone goes without spending, and the longer someone has been a customer in these games, the less likely that person is to spend. The recency effect is consistent with earlier findings and intuitively indicates that if a customer has not spent in a while, he or she is probably no longer a customer. The lifetime effect is also expected, especially in the present context, as customers are more likely to branch out to other games, with the passage of time. More interesting are the rates at which these decays occur, and how they vary across the games. These processes appear to be fundamentally different in the two games. In LS, the recency effect has a large impact, whereas the lifetime effect assumes a minimal role. In contrast, in CB, both appear equally important. These results may be a result of, for example, the design of the product (game), which encourages a certain pattern of purchasing.

Purchase Number The purchase number effect also appears different across the games. In LS, the effect seems relatively insignificant: although there is initially a slight rise, it quickly evens out, with a large confidence interval. In CB, the effect appears quite significant: it is generally increasing, but again appears to flatten out toward the end. The effect in CB is more consistent with our expectations: significant past purchasing should indicate a loyal customer, and a likely purchaser. A mild or neutral effect, like seen in LS, may indicate decreasing returns to spending in the game, or a limited number of new items that are available for purchase, such that the customer quickly runs out of worthwhile purchase opportunities.

Behavioral Implications The shapes of these curves have implications for player behavior and for designing general CRM strategies. In LS, the recency effect is the primary predictor of churn: if a customer has not spent for a while, she is likely no longer a customer. On the other hand, the lifetime effect seems to operate only in the first few days of being a customer, then levels out. This implies that customers are most likely to spend when they are new to the game, within roughly two weeks of their first purchase. In contrast, in CB, the effects are more equal in magnitude, and more gradual. The customers that are least likely to spend again are those that have been customers the longest, and have gone the longest without spending.

We illustrate these differences here via an individual-level analysis of respend probabil-

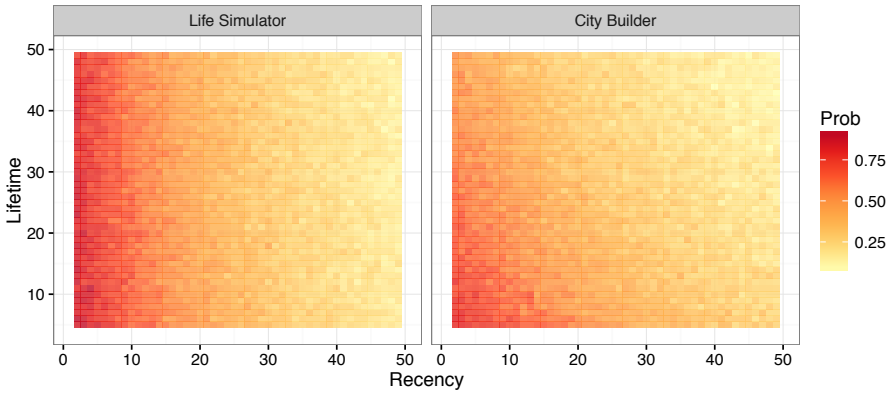

Figure 8: Respend probability heat maps for a customer with $q = 3$ and $\delta_i = 1$. Colors represent the probability of respending in the next 100 days, given the current recency and lifetime values. Note that some pairs of recency and lifetime that are displayed in the plot are not realistic: a customer cannot have recency higher than lifetime.

ity. Specifically, we ask the question, given an individual's recency and lifetime, what is the probability that she spends again in the next 100 days? To carry out this simulation, we fix the calendar time effect to its average value, and assume that the individual has already spent three times. The results of the simulation are displayed in Figure 8, and re-emphasize the point that recency explains much of the respend probability in LS, while lifetime and recency are both relevant in CB. This analysis also emphasizes the idea that, while the dynamic effects in the GPPM are the same for all customers, different positions in the individual-level subspace $(r_{ij}, \ell_{ij}, p_{ij})$ are associated with very different expected future purchasing behavior.

In summary, we have seen that the GPPM weaves together the different model components in a discrete hazard framework, and offers a principled approach for explaining aggregate purchase patterns based on individual-level data. The model-based dashboard generated by the GPPM is not the result of ad hoc data smoothing, but arises from the structural decomposition of spend propensity via the different model components. The GPPM jointly accounts for both the predictable individual-level determinants of respend probability, such as recency, lifetime, and purchase number, and calendar time events along multiple length-scales of variation. It is therefore able to flexibly represent the nature of customer respend probability, as well as accurately portray the existence and importance of calendar time events and trends.

3.3 Predictive Ability and Model Comparison

Apart from interest in understanding past spending dynamics, managers also need to forecast *future* purchasing activity. Although the primary strength of the GPPM is in uncovering latent dynamics, and conveying them in an intuitive fashion through the model-based dashboard, the GPPM also does very well in predicting future spending. Just as in-sample fit was driven by the recency, lifetime, and purchase number components, predictive per-

formance depends primarily on the ability to forecast these components for observations in the holdout data. While forms of recency, lifetime, and purchase number effects are incorporated in most customer base models, the isolation of these effects apart from transient calendar time variability, along with nonparametric characterization of these predictable components, and the inclusion of the cyclic component, allow the GPPM to significantly outperform benchmark customer base analysis models in predictive ability.

In this section, we focus on comparing both model fit and future predictive performance, and therefore reestimate the GPPM by truncating our original calibration data of 8,000 customers along the calendar time dimension. In particular, we set aside the last 30 days of calendar time activity to test predictive validity. Forecasting with the GPPM involves forecasting the latent functions that comprise it. In forecasting these latent functions, we use the predictive mechanisms outlined in Section 2.1 (Equation 4). As the holdout data is constructed by splitting the original dataset along the calendar time dimension, a substantial number of the observations in the holdout data contain recency, lifetime, and purchase number values that are within the observable range of these variables in the calibration dataset. This is especially true for observations belonging to newly acquired customers. However, for the oldest customers, the individual-level curves need to be forecast.

3.3.1 Benchmark Models We compare predictive performance of the GPPM with that of a number of benchmark models. Many individual-level models have been developed to do customer base analysis. At its core, the GPPM is a very general discrete hazard model and as such it can be compared to other hazard models for interpurchase times (Gupta, 1991; Seetharaman and Chintagunta, 2003). Similarly, given its reliance on recency, lifetime, and purchase number dimensions of spending, the GPPM is closely related to traditional customer base analysis models for non-contractual settings of the "buy-till-you-die" (BTYD) vein (Schmittlein et al., 1987; Fader et al., 2005, 2010). Finally, the discrete hazard approach could be modified with a different specification of the spend propensity.

Hazard Models We consider two standard discretized hazard models: the *Log-Logistic* model and the *Log-Logistic Cov* model, which are standard log-logistic hazard models without and with time-varying covariates respectively. We choose the log-logistic hazard as it can flexibly represent both monotonic and non-monotonic hazard functions. In the model with covariates, we use indicator variables over the time time periods of interest indicated at the start of Section 3. In estimating both of these models, we employ the same Bayesian estimation strategy, using Stan, with the same random effect heterogeneity specification as in the GPPM.

BTYD We use the *Pareto-NBD* (Schmittlein et al., 1987) and the *BGNBD* (Fader et al., 2010) as benchmarks in this class. While many variants of BTYD have been developed over the years, the Pareto-NBD has stood the test of time as the gold standard in forecasting power in non-contractual settings, often beating even more recent models (see, e.g., the PDO model in Jerath et al. (2011)). The BGNBD is a more discrete analogue of the Pareto-NBD, where customer death can occur after each purchase, rather than continuously.[10]

[10]We estimate these models using the BTYD package in the R programming language.

Propensity Models In this case, we retain the discrete time hazard inverse logit framework, while altering the specification of the dynamics. In particular, we explore two specifications: the *Linear Propensity Model (LPM)* and the *State Space Propensity Model (SSPM)*. These models have not been explored elsewhere in the literature; we include them here to help understand the benefits of the GP approach to modeling dynamics.

In the LPM, we remove the nonparametric specification altogether, and instead model all effects linearly, as

$$\Pr(y_{ij} = 1) = \text{logit}^{-1}(\mu + \beta_1 t_{ij} + \beta_2 r_{ij} + \beta_3 \ell_{ij} + \beta_4 q_{ij} + \delta_i). \tag{11}$$

This is the simplest discrete hazard model specification that includes all of our time scales and effects.

In the SSPM, we explore an alternate nonparametric specification for the dynamic effects. There are a number of competing nonparametric function estimation techniques, including dynamic linear models and various spline specifications, and there are technical links between many of these modeling approaches. Moreover, within each of class of models, there is a range of specifications that are possible, making the choice of a suitable benchmark difficult. We chose to implement a state space specification that is roughly equivalent to the GP structure in our main model. Specifically, we again decompose the propensity function $\alpha(t, r, \ell, q)$ into additive components along each dimension. For the calendar time dimension, just as in the GPPM, we make no assumptions about its behavior, and hence model it as a random walk:

$$\alpha_{\text{T}}(t) = \alpha_{\text{T}}(t-1) + \epsilon_{\text{T}t}, \;\; \epsilon_{\text{T}t} \sim \mathcal{N}(0, \zeta_{\text{T}}^2). \tag{12}$$

For the other dimensions, we assume as in the GPPM that there will likely be monotonicity, and hence include a trend component. This leads to a local level and trend specification:

$$\alpha_d(\tau) = \alpha_d(\tau-1) + \gamma_d(\tau) + \epsilon_{d\tau}, \;\; \epsilon_{d\tau} \sim \mathcal{N}(0, \zeta_d^2), \tag{13}$$

$$\gamma_d(\tau) = \gamma_d(\tau-1) + \xi_{d\tau}, \;\; \xi_{d\tau} \sim \mathcal{N}(0, \psi_d^2). \tag{14}$$

Interestingly, when used with a Gaussian observation model (meaning the data generating process is $\mathcal{N}(\alpha(\tau), \nu^2)$ instead of our latent propensity formulation), the local level and trend model has links to cubic spline smoothing (Durbin and Koopman, 2012). In addition to the above specified components, we also included a cyclic function of calendar time to mirror the GP periodic kernel component, as well as the random effects.

3.3.2 Forecasting Results The re-estimated in-sample fit and the out-of-sample forecast of the GPPM for both games are displayed in Figure 9. Again, the dashed lines represent medians, while the intervals represent 95% posterior predictive intervals. We see that, again, the GPPM fits very well in-sample, but importantly also fits well in the holdout period. Out-of-sample, we see smooth decreasing trends in both games, together with the predictable day of the week effect. Referring back to Figure 6, we see that the forecast fit is very similar to the fit decomposition with no short and long-run components. This is because, far from the range of the data, components modeled with a stationary kernel will

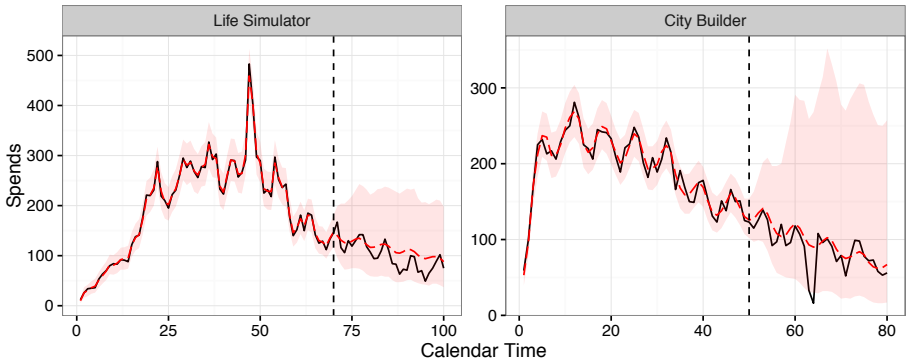

Figure 9: GPPM daily spending forecast. The data is in black with the median simulated GPPM fit in red (dashed) and 95% posterior predictive intervals. The holdout period is the last 30 days of data, demarcated by the dashed line.

revert to their mean function, which for the calendar time effects is constant, effectively muting them far into the holdout period. How long it takes for this reversion to happen depends on the smoothness of the estimated function.

Table 2 shows the predictive performance of the GPPM and all of our benchmark models. The table reports the mean absolute percentage error (MAPE) and the root mean squared error (RMSE) for the calibration and holdout datasets. Several of our benchmark fits are displayed in Figure 10. Crucially, the fit of the GPPM is almost always significantly better than the benchmarks, both in and out-of-sample. We proceed to briefly analyze each of the benchmarks, and give intuition for why the GPPM outperforms them.

The log-logistic hazard models perform particularly poorly. In fact, the fit of the log-logistic models using the full range of the data is worse than forecast fit of the GPPM; thus, we did not re-estimate the log-logistic models in a separate forecasting task. Neither of these models captures the lifetime and purchase number drivers of spending, which are typically highly predictive of spending. Furthermore, the Log-Logistic Covs model includes the covariates as indicator variables. While this is a very common approach for specifying events of interest, as we saw in our analyses of calendar time events, the impacts of these events are unlikely to be constant over time, a fact the GPPM implicitly incorporates in the calendar time effects.

Of primary interest to us is the comparison with the customer base analysis models. We see that the fit statistics of the Pareto-NBD and BGNBD are much better than that of the hazard models. In fact, the fit of the Pareto-NBD in Figure 10 is similar to the calendar time muted fit in Figure 6. This supports our intuition that the GPPM in a sense generalizes these models, by accounting for interpurchase and lifetime effects (in a nonparametric way), while simultaneously allowing for variability in calendar time. Accounting for variability in calendar time is important, as it lets the GPPM isolate predictable individual-level effects from the influence of calendar time events. In models that rely only on recency and frequency data, calendar time events are conflated with base purchasing rates, leading to erroneous predictions in the presence of calendar time dynamics. We show this through a set of simulations in Web Appendix B.

	Life Simulator			City Builder		
	Overall	In-sample	Holdout	Overall	In-sample	Holdout
GPPM	0.09	0.03	0.24	0.15	0.05	0.32
	13.25	5.74	22.54	15.00	9.79	20.97
Log-Logistic	0.42	0.31	0.67	0.41	0.19	0.77
	68.27	71.75	59.35	46.78	46.91	46.55
LL Covs	0.28	0.19	0.48	0.27	0.15	0.48
	62.81	67.22	51.04	36.28	32.78	41.47
Pareto-NBD	0.24	0.20	0.33	0.27	0.16	0.45
	45.10	49.64	32.10	33.54	36.56	27.80
BGNBD	0.23	0.19	0.31	0.34	0.18	0.61
	45.03	50.09	30.04	38.53	39.19	37.41
LPM	0.19	0.16	0.26	0.33	0.18	0.58
	42.78	47.21	30.02	43.14	38.80	49.53
SSPM	0.07	0.03	0.17	0.17	0.05	0.38
	12.57	6.63	20.59	18.25	9.50	27.16

Table 2: *Fit statistics. For each model, we report the mean absolute percentage error (MAPE, first row), and the root mean squared error (RMSE, second row) for both games in the forecasting task. We compute these measures over the entire range of data (Overall), over just the in-sample portion of the data (In-sample), and in just the 30 day holdout period (Holdout). Note that both of the log-logistic models were estimated over the full range of the data; given the poor fit using the full data, we did not estimate them separately using held out data.*

Finally, we see that while a linear specification of the dynamic effects is clearly not sufficiently rich, resulting in the poor fit of the LPM in both settings, a non-GP nonparametric specification like in the SSPM performs similarly to the GPPM. Specifically, we see that the SSPM performs as well as the GPPM in LS, while worse than the GPPM in CB. In some sense, this is not surprising: the SSPM is a complex and novel benchmark, constructed to be equivalent to the GPPM in terms of which effects it represents and how these are modeled. Both models capture the same set of predictable individual-level and periodic calendar time effects. Forecasting spending in the GPPM relies on forecasting these propensity functions, something which the SSPM also appears to do well.[11] Unlike the GPPM, however, the SSPM is more limited in its ability to separate out effects along a given time scale, which constrains its ability to perform the calendar time decompositions that are possible with GPs. This limits the SSPM's ability to provide equivalent dashboard-like representations of spend propensity along a given scale, which is one of the GPPM's core strengths.

[11] In fact, recent research has established deep links between GPs and state space models, such that some GP models can be approximated by state-space specifications (Gilboa et al., 2015). This may also explain their similar performance.

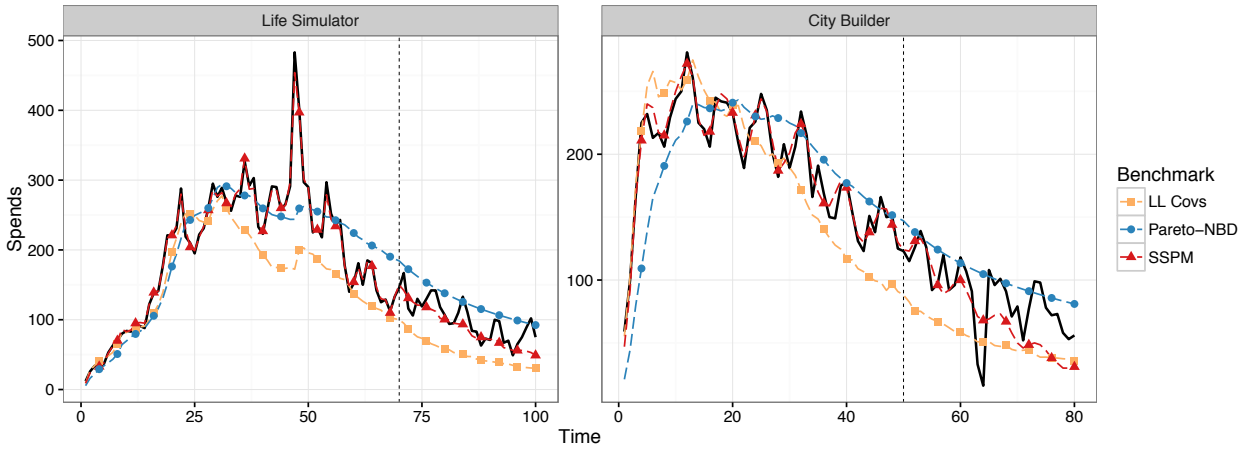

Figure 10: Daily spending forecasts for several of our benchmark models. The data is in black. The holdout period is the last 30 days of data, demarcated by the dashed line. A web app where all benchmark fits can be viewed in isolation and in comparison with the GPPM is available at `https://dr19.shinyapps.io/gppm_benchmarks/`.

4 Conclusion

In this paper, we developed a highly flexible model-based approach for understanding and predicting spending dynamics. Our model, the Gaussian process propensity model, or GPPM, employs Bayesian nonparametric Gaussian process priors to decompose a latent spend propensity into components that vary along calendar time, interpurchase time, customer lifetime, and purchase number dimensions. Our additive structure yields easily interpretable model outputs and fits customer spending data well.

We showed that the GPPM identifies the latent dynamic patterns in the data via a principled probabilistic framework that reliably separates signal from noise. It offers a number of outputs that are of considerable value to managers. First, the GPPM generates a dashboard of latent functions that characterize the spending process. These model-based visualizations are easy to comprehend, even by managers who may lack sophisticated statistical skills. Second, we demonstrated that the GPPM is capable of automatically capturing the effect of events that may be of interest to managers. In situations where certain events may escape the notice of managers, the GPPM is able to detect these events automatically. More importantly, the nonparametric nature of the GPPM allows it to flexibly model the nature and duration of the impact of events (either known or unknown, a priori), without the need to represent these explicitly via covariates. These advantages of the GPPM make it ideal for decision contexts involving multiple products and information asymmetries. The GPPM also flexibly captures the individual-level drivers of spending that reliably explain and predict spending behavior, including recency, lifetime, and purchase number effects. These effects can be used to characterize spending patterns within distinct customer bases, analyze individual customer respend probabilities, and predict future spending activity. Furthermore, since these effects are estimated jointly with the calendar time events, as part of a unified propensity model, the predictable, fundamental individual-level drivers of

spending are determined net of potentially unpredictable calendar time effects. Moreover, calendar time events can be analyzed net of the impact of expected individual-level spend activity, in a way not possible with mere aggregate data analysis.

We demonstrated these benefits of the GPPM on two data sets of purchasing activity within mobile games. We illustrated how the model-based dashboards that are generated from the GPPM yield easily interpretable insights about fundamental patterns in purchasing behavior. We also showed that the GPPM outperforms traditional customer base analysis models in terms of predictive performance, both in-sample and out-of-sample, including hazard models with time-varying covariates and the class of buy-till-you-die models. The predictive superiority of the GPPM stems from the fact that it captures the same predictable effects as traditional customer base analysis models, like recency and lifetime, but does so in a flexible way, net of the influence of calendar time events.

While the paper showcases the many benefits of our framework, it is also important to acknowledge some limitations. First, the framework in its current form is computationally demanding, especially when compared with simpler probability models that can be estimated with maximum likelihood. It is also data intensive. In our application, we used complete individual-level event log data to estimate the model. Some of the benchmark models, in particular, the BGNBD and the Pareto-NBD, use only two sufficient statistics per customer. Both of these limitations can perhaps be addressed in practice by either data subsampling, or by developing faster inference algorithms. Finally, while we believe our model-based dashboard is useful, insofar as it provides a snapshot of the key drivers of spending dynamics, it does not work in real-time, as is the case for many dashboards of marketing metrics. A streaming data version of our model would be an interesting area for future work.

To conclude, we believe the GPPM addresses a fundamental need of modern marketing managers for a flexible system for dynamic customer base analysis. In providing a solution to this problem, this work introduces a new Bayesian nonparametric approach to the marketing literature. While we discuss Gaussian Process priors in the context of dynamic customer base analysis, their potential applicability to other areas of marketing is much broader. GPs provide a general mechanism for flexibly modeling unknown functions, and for doing Bayesian time series analysis. We see many potential applications for GPs in marketing, including in the modeling of the impact of marketing mix variables, such as advertising and promotions, and in the approximation of unknown functions in dynamic programming and other simulation contexts. Our work also makes a contribution to the largely unaddressed field of visual marketing analytics systems, or dashboards. Dashboards and marketing analytics systems are likely to become even more important in the future, given the increasing complexity of modern data-rich environments. As dashboards increase in relevance, we believe that managers will welcome further academic research in this domain.

References

Ansari, A. and Mela, C. F. (2003). E-Customization. *Journal of Marketing Research*, 40(2):131–145.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Li, P., and Riddell, A. (2016). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, VV(Ii).

Durbin, J. and Koopman, S. J. S. (2012). *Time series analysis by state space methods*.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. *Proceedings of the International Conference on Machine Learning (ICML)*, 30:1166–1174.

Fader, P., Hardie, B., and Lee, K. L. (2005). Counting Your Customers the Easy Way: An Alternative to the Pareto/NBD Model. *Marketing Science*, 24(2):275–284.

Fader, P., Hardie, B., and Shang, J. (2010). Customer-Base Analysis in a Discrete-Time Noncontractual Setting. *Marketing Science*, 29(6):1086–1108.

Gelman, A. and Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–511.

Gilboa, E., Saatci, Y., and Cunningham, J. P. (2015). Scaling multidimensional inference for structured gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):424–436.

Gupta, S. (1991). Stochastic models of interpurchase time with time-dependent covariates. *Journal of Marketing Research*, 28:1–15.

Hanssens, D. M., Parsons, L. J., and Schultz, R. L. (2001). *Market Response Models: Econometric and Time Series Analysis*. Kluwer Academic Publishers, 2nd edition.

Hoffman, M. and Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1351–1381.

Jerath, K., Fader, P. S., and Hardie, B. G. (2011). New Perspectives on Customer Death Using a Generalization of the Pareto/NBD Model. *Marketing Science*, 30(5):866–880.

Kalyanam, K. and Shively, T. S. (1998). Estimating Irregular Pricing Effects: A Stochastic Spline Regression Approach Estimating Irregular Pricing Effects: A Stochastic Spline Regression Approach. *Journal of Marketing Research*, 35(1):16–29.

Kim, J. G., Menzefricke, U., and Feinberg, F. M. (2007). Capturing Flexible Heterogeneous Utility Curves: A Bayesian Spline Approach. *Management Science*, 53(2):340–354.

Li, Y. and Ansari, A. (2014). A Bayesian Semiparametric Approach for Endogeneity and Heterogeneity in Choice Models. *Management Science*, 60(5):1161–1179.

Neal, R. M. (1998). Regression and Classification Using Gaussian Process Priors. *Bayesian Statistics*, 6:475–501.

Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., and Mason, C. H. (2006). Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models. *Journal of Marketing Research*, 43(2):204–211.

Pauwels, K., Ambler, T., Clark, B. H., LaPointe, P., Reibstein, D., Skiera, B., Wierenga, B., and Wiesel, T. (2009). Dashboards as a Service: Why, What, How, and What Research Is Needed? *Journal of Service Research*, 12(2):175–189.

Rasmussen, E. and Williams, K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371(1984):20110550.

Rossi, P. E. (2013). Bayesian Semi-parametric and Non-parametric Methods with Applications to Marketing and Micro-econometrics.

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

Schweidel, D. A. and Knox, G. (2013). Incorporating Direct Marketing Activity into Latent Attrition Models. *Marketing Science*, 32(3):471–487.

Seetharaman, P. B. and Chintagunta, P. K. (2003). The Proportional Hazard Model for Purchase Timing: A Comparison of Alternative Specifications. *Journal of Business & Economic Statistics*, 21(3):368–382.

Shively, T. S., Allenby, G. M., and Kohn, R. (2000). A Nonparametric Approach to Identifying Latent Relationships in Hierarchical Models. *Marketing Science*, 19(2):149–162.

Wedel, M. and Zhang, J. (2004). Analyzing Brand Competition Across Subcategories. *Journal of Marketing Research*, 41:448–456.

Web Appendix A: Hamiltonian Monte Carlo

We use a fully Bayesian approach for inference. As outlined in Section 2, the joint density of all unknowns is given by

$$p(\boldsymbol{y}, \{\boldsymbol{\alpha}_k\}, \boldsymbol{\delta}, \boldsymbol{\phi}, \sigma^2) = \left[\prod_{i=1}^{I}\prod_{j=1}^{M_i} p(y_{ij}|\alpha_{ij}, \delta_i)p(\delta_i|\sigma^2)\right]\left[\prod_{k=1}^{K} p(\boldsymbol{\alpha}_k|\boldsymbol{\phi}_k)\right]p(\sigma^2)p(\boldsymbol{\phi}). \quad (15)$$

As the full posterior distribution $p(\{\boldsymbol{\alpha}_k\}, \boldsymbol{\delta}, \boldsymbol{\phi}, \sigma^2|\boldsymbol{y})$ is not available analytically, we use the Hamiltonian Monte Carlo (HMC) algorithm to draw samples of the unknown function values $\boldsymbol{\alpha}_k$, customer-specific random effects $\boldsymbol{\delta}$, population parameters σ^2, and the GP hyperparameters $\boldsymbol{\phi}$, from the posterior. For completeness, we include a brief overview of HMC here, and refer the reader to Neal (2011) for further details.

HMC is a variant of the Metropolis-Hastings algorithm that uses a proposal distribution that is based on the Hamiltonian dynamics of a particle moving in a potential field. Suppose our interest is in sampling a set of parameters $\boldsymbol{\theta} \in R^p$ (i.e., particle positions) from a target posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$. For our model, $\boldsymbol{\theta}$ can contain the entire set of unknown function values and GP hyperparameters. HMC uses a vector of auxilliary momentum variables $\boldsymbol{\zeta} \in R^p$ drawn from a multivariate normal $N(\boldsymbol{\zeta}|0, M)$ where the covariance matrix M is the mass matrix. Both the positions and the momentum variables are jointly sampled from a joint density $p(\boldsymbol{\theta}, \boldsymbol{\zeta}|\boldsymbol{y}) = p(\boldsymbol{\theta}|\boldsymbol{y})p(\boldsymbol{\zeta})$. The values of $\boldsymbol{\theta}$ are retained, where as the samples of $\boldsymbol{\zeta}$ are ignored. Algorithm 1 outlines a single HMC iteration.

Algorithm 1 HMC Iteration (Given stepsize ϵ, number of leapfrog steps, L mass matrix M, and $\boldsymbol{\theta}_{current}$)

Initialize $\boldsymbol{\theta}_{(0)} \leftarrow \boldsymbol{\theta}_{current}$, $\boldsymbol{\zeta}_{(0)} \sim \mathcal{N}(0, M)$

for $l = 0, \ldots, L-1$ **do** ▷ Perform Leapfrog steps
  $\boldsymbol{\zeta}_{(l+1/2)} \leftarrow \boldsymbol{\zeta}_{(l)} + \frac{1}{2}\epsilon\nabla_\theta \log p(\boldsymbol{\theta}_{(l)}|\boldsymbol{y})$
  $\boldsymbol{\theta}_{(l+1)} \leftarrow \boldsymbol{\theta}_{(l)} + \epsilon M^{-1}\boldsymbol{\zeta}_{(l+1/2)}$
  $\boldsymbol{\zeta}_{(l+1)} \leftarrow \boldsymbol{\zeta}_{(l+1/2)} + \frac{1}{2}\epsilon\nabla_\theta \log p(\boldsymbol{\theta}_{(l+1)}|\boldsymbol{y})$
end for

$r = \min\left[1, \frac{p(\boldsymbol{\theta}_{(L)}|\boldsymbol{y})\, p(\boldsymbol{\zeta}_{(L)})}{p(\boldsymbol{\theta}_{(0)}|\boldsymbol{y})p(\boldsymbol{\zeta}_{(0)})}\right]$ ▷ Compute acceptance probability
$u \sim$ Uniform$(0, 1)$ ▷ Uniform draw
if u $< r$, **then** return $\boldsymbol{\theta}_{(L)}$ ▷ Accept or reject proposal
else return $\boldsymbol{\theta}_{(0)}$
end if

As can be seen from Algorithm 1, each iteration of the HMC algorithm involves several leapfrog steps in which $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ evolve according to a discretization of Hamilton's equations. The HMC sampler uses the gradient of the log-posterior to direct the exploration of the posterior. This allows it to avoid the random walk behavior of ordinary Metropolis-Hastings procedures and it therefore traverses the posterior in an efficient fashion. HMC methods

are ideal for non-conjugate GP settings such as ours, as they can efficiently sample both the latent function values as well as the hyperparameters.

In practice, we need to specify values for the step size ϵ, the number of leapfrog steps L and the mass matrix M, and finding the right set of values for these can be sometimes challenging. We therefore use the No U-Turn Sampling (NUTS) variant of HMC as implemented in the Stan probabilistic programming language (Hoffman and Gelman, 2014; Carpenter et al., 2016). Stan uses an adaptive version of the HMC algorithm wherein ϵ, L and M are updated across the MCMC iterations to ensure rapid mixing, while still maintaining detailed balance. Since each iteration of HMC involves multiple leapfrog steps, an HMC iteration is not directly comparable to that of the ordinary Metropolis-Hastings algorithm, and convergence is achieved in much fewer MCMC iterations. Details of NUTS are given in Hoffman et al. (2014).

Web Appendix B: Simulation Studies

In this appendix, we use simulated data to explore various aspects of the GPPM. In the first section, we show how the GPPM can be extended to accommodate different length-scales of variation along different time dimensions, to capture things like "loyalty" promotions, for example, that might occur along the lifetime dimension. In the second section, we explore links between the GPPM and classic buy-till-you-die (BTYD) models for customer base analysis, focusing on the BGNBD model as our example. BTYD models have served as the backbone for many customer base analysis applications, showing a particularly robust ability to forecast future spending and compute customer-centric quantities of interest by modeling just interpurchase times and customer lifetimes. The GPPM extends this framework by also allowing for the consideration of an additional input, calendar time. To explore how the GPPM generalizes these ideas, we simulate data from both models, and show first how the recency and lifetime components of the GPPM are able to capture the equivalent BTYD effects, and second why the inclusion of calendar time effects is important in accurately estimating individual-level spend rates. Since we use simulated data across all of these studies, we can see throughout examples of how the GPPM can capture the shape of events of interest automatically, as we know in these cases exactly the impact a given event.

Extending the GPPM

In Section 2.2 of the paper, we described the modular approach to specifying the GPPM. Recall that each kernel represents a broad type of functions. In the main paper, we used SE kernels to pick up variation along two length-scales, short and long, for the calendar time effects, along with a predictable periodic component. We used a single SE kernel with a monotonic power mean function to isolate variability along the other dimensions. In practice, we may want to extend the model in various ways. One potential deviation from the general model explained in the body of the paper is the need to capture short-run effects of interest that may occur along other dimensions, particularly along the lifetime dimension. These effects could exist, for instance, if the company has loyalty based rewards or promotions, such that the consumer is given a special after a certain number of days after first purchase.

To cope with shocks along the lifetime dimension, we can extend the GPPM quite simply by adding an additional SE component to the lifetime specification. By the additive property of GPs, this specification remains a GP, just with an additive kernel. Hence, we now model:

$$\alpha_{\mathrm{L}}(\ell) = \alpha_{\mathrm{L}}^{\mathrm{Long}}(\ell) + \alpha_{\mathrm{L}}^{\mathrm{Short}}(\ell),$$

where:

$$\begin{gathered}\alpha_{\mathrm{L}}^{\mathrm{Long}}(\ell) \sim \mathcal{GP}(m(\ell), k_{\mathrm{SE}}(\ell, \ell'; \eta_{\mathrm{LL}}, \rho_{\mathrm{LL}})), \\ \alpha_{\mathrm{L}}^{\mathrm{Short}}(\ell) \sim \mathcal{GP}(0, k_{\mathrm{SE}}(\ell, \ell'; \eta_{\mathrm{LS}}, \rho_{\mathrm{LS}})), \\ \rho_{\mathrm{LS}} < \rho_{\mathrm{LL}}\end{gathered}$$

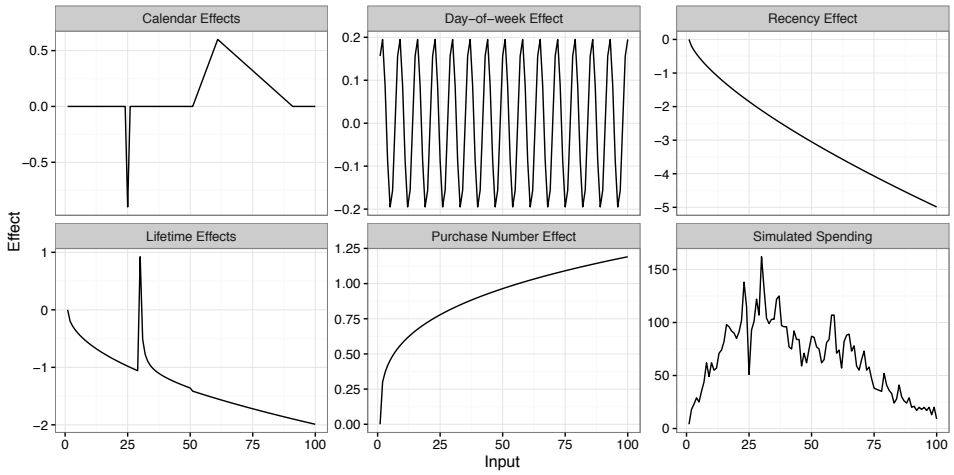

Figure 11: True effects used to generate the simulated data, with the simulated spending time series shown in the bottom right panel.

With this setup, we can capture short-run departures from the smooth trend component along the lifetime dimension, just like we captured both trends and short-run shocks in the calendar time component before. In this case, we include the same mean function as before (power mean) along the long-run curve.[12]

Simulation We simulated data within the GPPM framework, similar to the data from our application. We simulated the spending of 2,000 customers, entering over a period of 30 days, using the effects displayed in Figure 11. The sum of these effects results in the spending time series displayed in the bottom right panel of Figure 11. We then estimated the GPPM on this data, using the extension described above. The resulting extended dashboard is shown in Figure 12. We see that the GPPM recovers all of the effects in the data generating process, without specifying any of them as inputs to the model. More importantly, we see the natural extension of the GPPM to capture the shock to the lifetime dimension. The instantaneous effect of the loyalty reward is captured in the Lifetime, Short panel, with the residual effect slight, but noticeable in the Lifetime, Long panel.

Links between GPPM and BTYD

The GPPM provides a natural generalization of buy-till-you-die customer base analysis models that rely solely on recency and lifetime, such as the BGNBD. While the GPPM does not explicitly account for customer death, it does so asymptotically by allowing the probability of purchase to go to zero via the lifetime and recency effects. To explore this link deeper, we ran a series of simulation studies, testing in which cases the GPPM is able to capture BGNBD data, and vice versa.

[12]By additivity, the results would be equivalent if the mean function were included in the short-run term; however, we find the idea of a trend + shock formulation more intuitive, and hence model it as such.

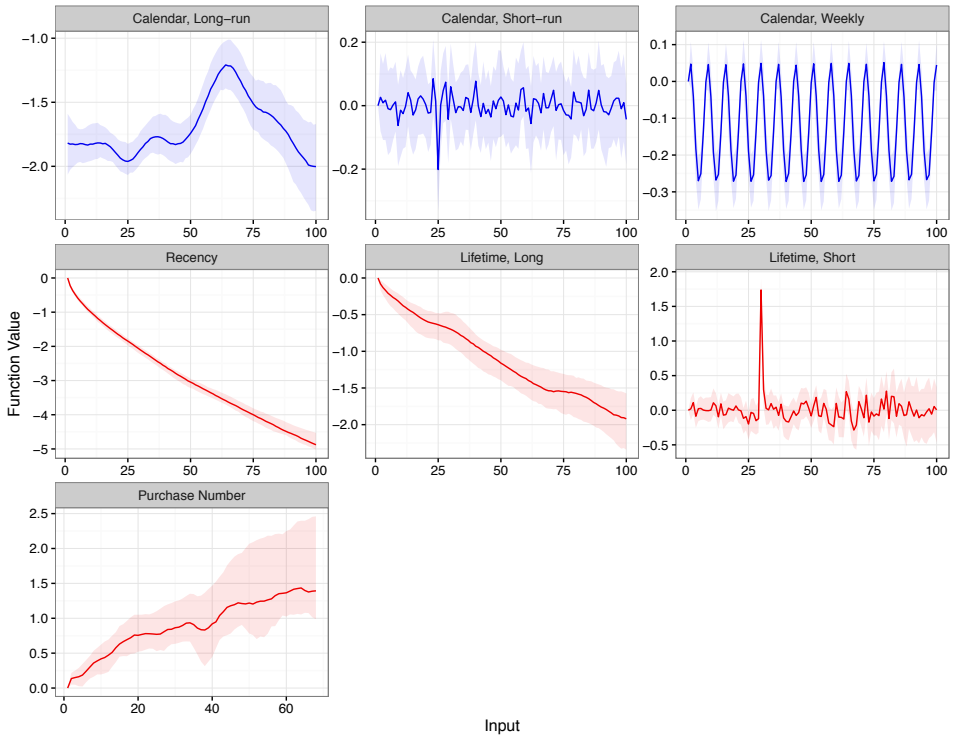

Figure 12: Extended GPPM dashboard for simulated data.

We hypothesize that the dynamic spending patterns that are captured by the BGNBD can also be captured by the GPPM; however, the BGNBD will have a difficult time fitting data generated by the GPPM, depending on the strength of calendar time effects present. This is because the BGNBD and other parametric probability models based on individual-level effects have no way of separating out temporary shifts in spend propensity due to calendar time effects from underlying, predictable individual-level effects. To test these two hypotheses, we first see how the GPPM does at fitting data generated by the BGNBD model. Then we do the reverse and estimate the BGNBD on data from GPPM specifications that vary the strength and nature of the calendar time effects.

BGNBD Data, GPPM Fit If the recency and lifetime components of the GPPM do capture the dynamic patterns inherent in the BGNBD, then the GPPM should be able to do well on data generated from the BGNBD. To see this, we generate data from 8,000 spenders across 30 first spend dates, similar to our real data. We simulate spending over 100 days according to a BGNBD model, and then fit the GPPM on the first 50 days of simulated data, and forecast the activity on days 51 to 100. As our main example, we use the estimated BGNBD parameters ($r = 0.243$, $\alpha = 4.414$, $a = 0.793$, $b = 2.426$) from the original BGNBD paper (Fader, Hardie, and Lee, 2010, subsequently FHL). We also used many combinations of randomly generated parameters to test robustness, with smaller sample sizes of 2,000 customers. The fit statistics for all of the simulations are summarized in Table 3. The good fit offers substantial evidence to our claim that the GPPM nests these

DGP	Model	Overall	Training	Holdout
BGNBD, FHL Parameters	GPPM	0.07	0.05	0.09
BGNBD, Random*	GPPM	0.10	0.06	0.14
GPPM, All*	BGNBD	0.54	0.21	0.87
GPPM, `Nocal` Only*	BGNBD	0.22	0.15	0.29

Table 3: Fit summaries for the simulation studies. The first column contains the data generating process, while the second contains the model used to forecast spending. An asterisk (*) is used to denote the statistics that are the average value across many simulations. The statistics presented are MAPE (mean absolute percentage error). RMSE is not relevant here as each simulation results in spending on a different scale, and hence RMSE is not comparable across simulations.

traditional probability models.

GPPM Data, BGNBD Fit We also study the reverse situation and examine the performance of the BGNBD on data generated from the GPPM. We show that BGNBD is not able to fit such data very well, especially in the presence of calendar time dynamics. Specifically, we use three levels of the day of the week effect — none (`Nocyc`), weak (`Weakcyc`), and strong (`Strongcyc`) — and three kinds of non-cyclic calendar time effects: none (`Nocal`), a long-run peak similar to the general holiday season bump seen in our application (`Peakcal`), and a nonlinear decreasing trend across the whole time period (`NonlinDeccal`). The cyclic effect was set as $\alpha_w(t) = \theta \sin(2\pi t/7)$, where $\theta = 0$, for no cyclic effect, $\theta = 0.15$, for the weak effect, and $\theta = 0.4$, for the strong effect. For the calendar time effects, the non-linear decreasing calendar time trend is given by $\alpha_{\mathrm{T}}(t) = -0.2t^{0.3}$; the peak effect is given by the piecewise function: $\alpha_{\mathrm{T}}(t) = 0$, when $t \leq 20$; $\alpha_{\mathrm{T}}(t) = 0.5(t - 20)$, when $t \in [21, 40]$; $\alpha_{\mathrm{T}}(t) = 0.1(50 - t)$, when $t \in [41, 50]$ and $\alpha_{\mathrm{T}}(t) = 0$, when $t > 50$.

Figure 13 and Table 3 show the results from these simulations. We see that BGNBD fits the mean of the curve in the presence of a cyclic effect. We also see that the BGNBD generally does well in the cases where there is no short or long-run calendar variation, underpredicts in the beginning and then overpredicts in the end when there is a decreasing calendar time effect, and fails significantly at capturing the peak effect. We see in the `Peakcal` case (last row of Figure 13) that the BGNBD attributes the peak to higher rates of spending, and then dramatically overestimates future spending.

The GPPM does not fall prey to this same bias because of its ability to separate out calendar time effects. To emphasize this, we see the GPPM fit to the worst case (`Strongcyc/Peakcal`), together with the estimated calendar time effect, in Figure 14. The excellent fit and near perfect forecast is not surprising: the GPPM is capturing data generated from a GPPM. One thing to point out is that this, again, demonstrates the ability of the GPPM to nonparametrically recover the effects of events, as we see the peak in calendar time is equivalent to the piecewise function described above.

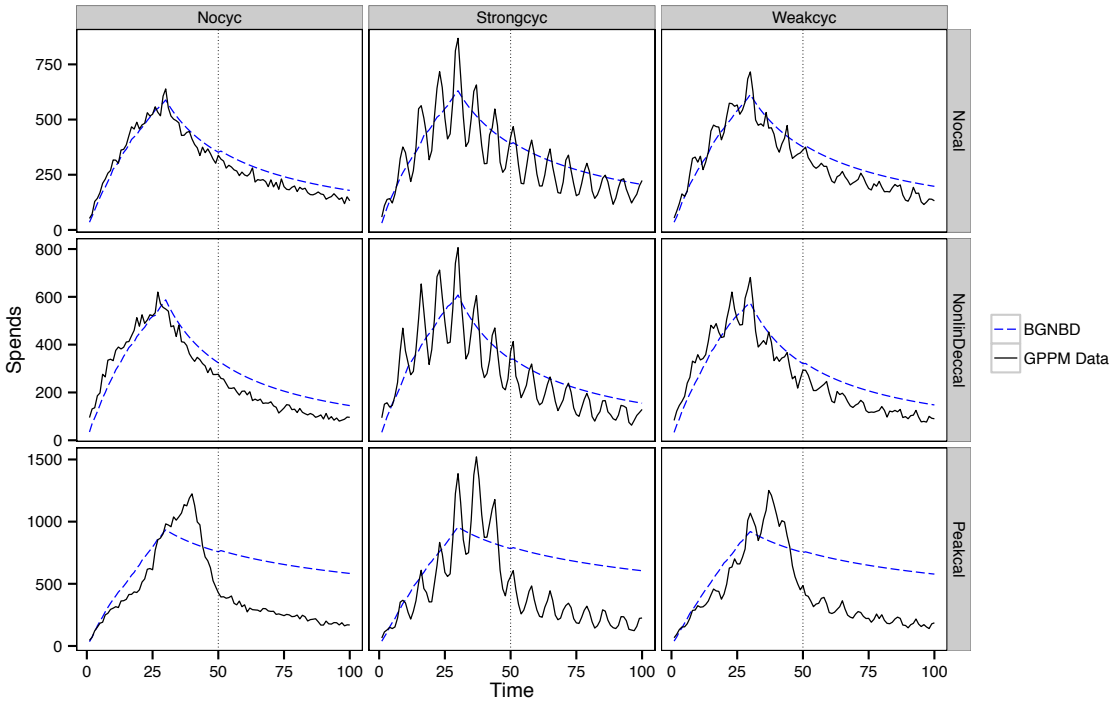

Figure 13: The BGNBD fit on various types of data drawn from the GPPM: `Nocyc`, `Strongcyc`, and `Weakcyc` indicate no, strong, and weak cyclic (day of the week) effects respectively; `Nocal` indicates no calendar time dynamics, `NonlinDeccal` indicates a non-linear decreasing long-run calendar time process, and `Peakcal` indicates a calendar time process that is flat but with a peak during the calibration period.

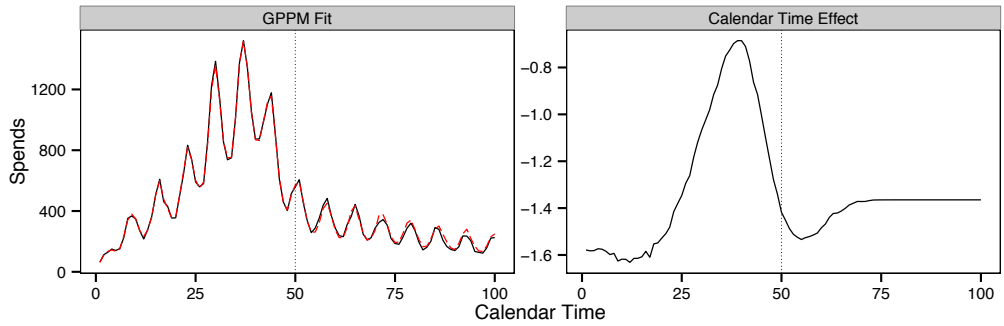

Figure 14: The GPPM fit and forecast on the `Strongcyc/Peakcal` simulated data, together with the estimated calendar time effect. We see that the GPPM captures the pointed piecewise effect, and is therefore able to isolate the predictable, individual-level effects that allow it to accurately forecast future spending.