Forecasting Product Life Cycle Curves: Practical Approach and Empirical Analysis

Kejia Hu[1], Jason Acimovic[2], Francisco Erize[3], Doug Thomas[2], Jan A. Van Mieghem[1]

[1]Kellogg School of Management, Northwestern University

[2]Smeal College of Business, Penn State University

[3] Supply Chain, Dell

May 1, 2017

We present an approach to forecast customer orders of ready-to-launch new products that are similar to past products. The approach fits product life cycle (PLC) curves to historical customer order data, clusters the curves of similar products, and uses the representative curve of the new product's cluster to generate its forecast.

We propose three families of curves to fit the PLC: Bass diffusion curves, polynomial curves and simple piecewise-linear curves (triangles and trapezoids). Using a large data set of customer orders for 4,037,826 units of 170 Dell computer products sold over three years, we compare goodness-of-fit and complexity for these families of curves. Fourth-order polynomial curves provide the best fit with piecewise-linear curves a close second. Using a trapezoidal fit, we find that the PLCs in our data have very short maturity stages; more than 20% have no maturity stage and are best fit by a triangle.

The fitted PLC curves of similar products are clustered either by known product characteristics or by data-driven clustering. Our key empirical finding is that data-driven clustering of simple triangles and trapezoids, which are simple-to-estimate and explain, performs best for forecasting. Our conservative out-of-sample forecast evaluation, using data-driven clustering of triangles and trapezoids, results in mean absolute errors approximately 3-4% below Dell's forecasts. We also apply our method to a second data set of a smaller company and find consistent results.

Many companies seek to innovate and bring new products and services to market, and growth and product innovation remain top priorities for executives. CEOs have indicated their commitment to new product development growing over time (PWC 2017) and at least one survey reports new product development as their top investment priority (KPMG 2016). One common metric used to evaluate the success of a firm's innovation efforts is the percentage of revenue derived from new products. Based on a cross-industry survey, Cooper and Edgett (2012) report that an average of 27% of a firm's revenue comes from new products. (This percentage varied dramatically across respondents; 27% would be quite high for a food or consumer goods manufacturer and quite low for a technology firm.) The same survey also reports that the percentage of *profit* coming from new products lags the percentage of revenue coming from new products. This suggests that while new products are essential to growth, they are expensive to support.

A key challenge in managing new product introductions is creating sales forecasts. Here it is important to distinguish how firms approach forecasting in general, and how the approach may differ for new product forecasting. Several studies indicate that statistical methods play a major role for sales forecasts of existing products. For example, based on a survey of 144 forecasting practitioners, Fildes and Goodwin (2007) report that 75% of all forecasts are generated or at least influenced by a statistical forecast. Contrast this with new product forecasting where market-research based methods and executive opinion dominate (Kahn 2002). While these approaches may be best, or even the only viable approach, for completely new market entries, most new products are not *un*like anything we have ever seen before. Focusing just on new product forecasting, Kahn (2002) separates new products according to their "newness," ranging from incremental cost or product-attribute improvements to "new-to-the-world" market entries. Perhaps surprisingly, the survey results in Kahn (2002) indicate that the most popular three techniques used in practice—market research, executive opinion and sales force input—are the same across the range of product newness. That is, even when a firm has historical data on a similar product, market research, executive opinion and sales input are still the most commonly used approaches.

Data-driven, statistical forecasting for new, but not earth-redefining new products, is the focus of this paper and where we seek to make a contribution. Our objective is to develop an approach that can be effectively applied to generate forecasts for new products that are similar to previous products. Our main industrial partner is Dell and our motivating setting is the personal computer industry, which is characterized by very high reliance on new product revenue and short product life cycles. We describe the business context for Dell in greater detail below. The difference in product life cycles for computers was observed quite early, with Goldman (1982) pointing out that computer sales were "...characterized by a short life on the market, a steep decline stage and the lack of a maturity stage." Figure 1 shows a typical product life cycle curve with introduction, growth, maturity and decline stages next to actual customer orders for one of the products in our data set. The simple triangular and trapezoid PLC curves shown in Figure 1 fit historical orders quite well. As we will see, the pattern in Figure 1 is representative as many products in our data set have a very short maturity stage (50% have maturity stages of less than 12% of life cycle) and are well fit by simple, piecewise-linear PLC curves. Triangular and trapezoid PLC curves are attractive in practice because they are easy to explain and quantify. In addition to the large data set from Dell, we also analyze a smaller data set from a manufacturer of computer accessories with consistent findings.

While new product forecasting is of critical importance for personal computers, and that industry is an important one, our approach for product life cycle forecasting is general and can be applied in other settings. The central idea behind our approach is to (1) use the historical product life cycle (PLC) customer order information of previous similar products to fit a PLC curve and to

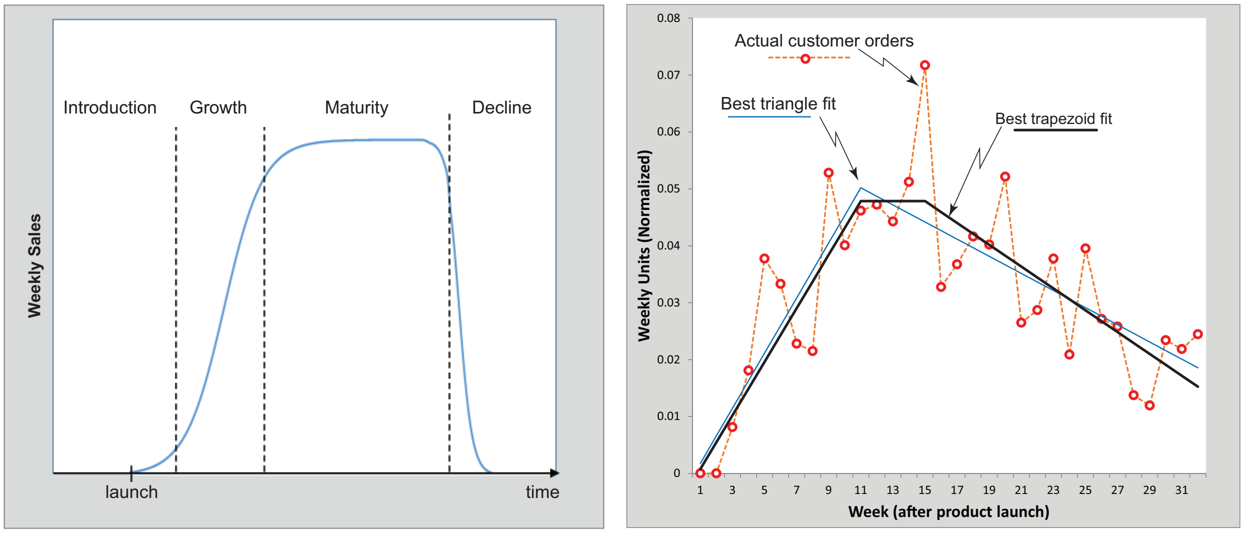

Figure 1 **A typical PLC curve (left) has four stages. The actual orders of the majority of short life cycle technology products at our partner company are best described by a triangle or by a trapezoid with a very short maturity stage (right). (Right-hand figure corresponds to SKU150.)**

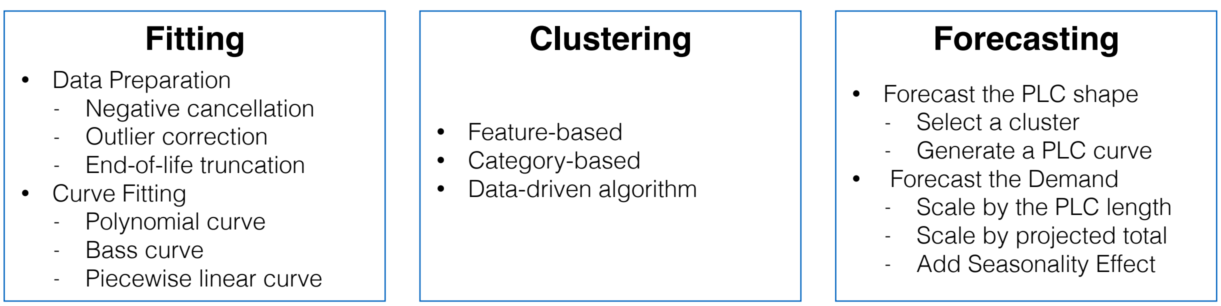

Figure 2 **Forecasting using PLC curves requires three steps: data preparation, curve fitting, and forecast generation.**

(2) use the PLC curve to forecast the entire customer order evolution of ready-to-launch new products that are similar to past products. Our three-step approach, summarized in Figure 2, involves preparing and normalizing the data, fitting several families of curves to normalized PLC data and then identifying representative PLC curves and using these curves to generate forecasts.

Four elements in our approach are important. First, to accommodate product heterogeneity, we use normalized product life cycle data with clustering to operationalize "similarity" between products. We normalize the total volume and the length of the PLC to 1. This normalization allows us to look for similar patterns across items that may have different volumes and life cycles. Clustering could be provided exogenously (e.g., by the company's product hierarchy or features) and/or could be automated or refined using a data-driven clustering algorithm. We implement and evaluate both of these approaches. Second, we focus on static forecasting of the entire PLC just before the product launch. The key reason for, and advantage of, forecasting the entire PLC curve is that in many settings, including the setting at Dell that we analyze below, the product life cycles are short relative to production-transportation leadtimes. Characterizing the PLC curve ahead-of-launch helps not only with internal planning to meet customer demand but also with coordinating external sourcing and capacity decisions. Third, forecast accuracy is our measure of effectiveness. It is well known that forecast accuracy directly drives safety inventory and capacity requirements, but it also significantly impacts sourcing commitments and transportation decisions, which leads us to the final element in our approach. Fourth, we apply our approach on a large set of actual customer order data covering 4,037,825 Dell computers over 170 products sold over 3.4 years representing well over a billion dollar in revenues. We find that simple piecewise-linear curves (trapezoids, which include triangles) have lower forecast errors than smooth curves (polynomial and Bass (1969)) with comparable numbers of parameters. Figure 3 summarizes the quantitative results of the extensive evaluation study that we explain in detail below. That study will show that our approach reduces absolute forecast errors by almost 4% relative to Dell's forecasts in the best case, and more than 2% on average under robustness tests. We also apply our method to a second data set of another smaller company and find consistent results.

According to an internal study conducted at Dell during the time period we study, an improvement in forecast accuracy of the magnitude we report would result in transportation and inventory expense savings of \$2-\$4 per unit on annual volumes in the millions. Furthermore, we anticipate that demand planners will be able to adjust our initial PLC forecasts using additional information such as sales events or possible cannibalization to achieve forecast accuracy improvements beyond the 3-4% we report for or initial PLC forecasts.

1. Literature review

Half a century ago, Levitt (1965) wrote that "most alert and thoughtful senior marketing executives are by now familiar with the concept of the product life cycle." His critical review of strengths and weaknesses, including the importance of forecasting its shape to put the concept to work, remains surprisingly relevant. According to Rink and Swan (1979), the idea of the PLC was introduced in 1950 yet 30 years later there remained a paucity of empirical evidence. Golder and Tellis (2004) and Stark (2015) provide contemporary overviews of the vast field of PLC theory and management. The remainder of this section focuses on how our work contributes to selective relevant strands of literature on product life cycle forecasting.

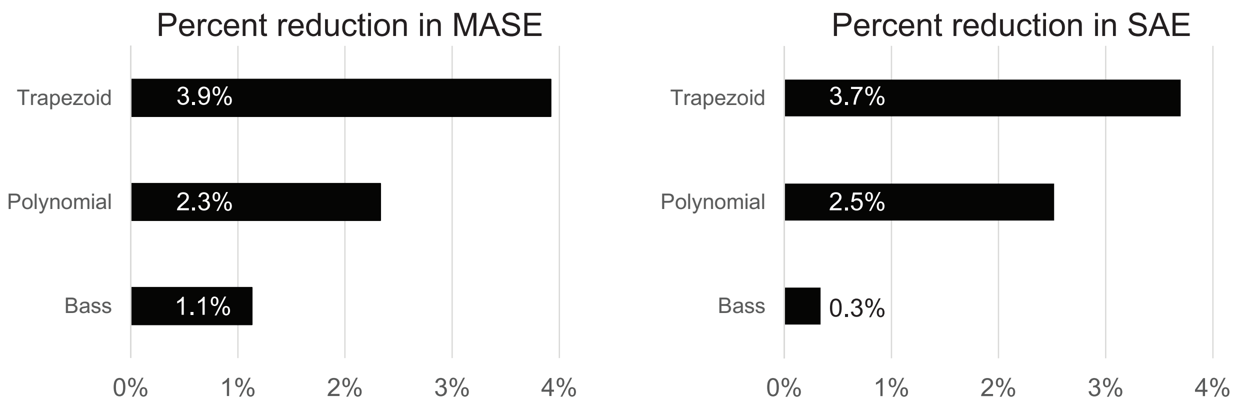

Figure 3 **Overall summary of reduction in forecast errors derived from our approach relative to Dell's internal forecasts over the initial months of a product's life cycle, segmented by family of curves used in fitting individual SKUs' life cycles. MASE (mean absolute scaled error) is a normalized measure of forecast accuracy that can be compared across SKUs, while SAE (sum of absolute errors) is the sum of the unnormalized mean absolute errors across a portfolio of SKUs.**

1.1. PLC curves: theory

The diffusion model introduced by Bass (1969) remains a cornerstone in PLC theory. Its ensuing differential equation can be solved explicitly (Lemma 1 in Kumar and Swaminathan (2003) and reproduced later); we will refer to its particular bell-shape as the Bass curve. It should be noted that the classic Bass model has been extended or modified in several dimensions. In terms of market and company structure, Tigert and Farivar (1981) account for whether or not there is a monopoly and whether or not the company is public. From an operational perspective, for example, Ho et al. (2002) and Kumar and Swaminathan (2003) include supply availability and constraints. Niu (2006) proposes a stochastic Bass model and allows its parameters to vary over time periods.

1.2. PLC curves: empirical studies

In addition to theory-inspired Bass curves, authors have fitted other curves to empirical data. One popular family of curves are polynomials, which have been validated with demand data for recreation programs (Crompton 1979), municipal library services (Crompton and Bonk 1978) and grocery products (Headen 1966). Another popular family are piecewise-linear curves, as suggested by demand data for ethical drugs (Cox 1967), food (Buzzell and Nourse 1967) and chemicals (Frederixon 1969). Our empirical analysis will include Bass curves, as well as polynomial and piecewise-linear curves.

1.3. New Product and PLC Forecasting

Goldman (1982) states that many high-tech companies frequently face a situation of a long lead time and a short PLC. Such a situation is most demanding managerially. It is noteworthy to point out that classical time series forecasting models require the knowledge of some realized demands or sales to generate their forecasts, which are unavailable in this situation. To help that challenge, our paper analyzes how to forecast entire PLC curves using historical data of similar previous products. Earlier, Fisher and Raman (1996) demonstrated the importance of an initial forecast (not necessarily of the entire PLC curve) quality and of forecast updating and responsive fulfillment. Gallien et al. (2015) analyze flexibility in replenishment for new product launches where initial forecasts are updated (for Zara). Another approach to address the challenge is a product portfolio approach where some product demands may serve as leading indicators for a group of products (Wu et al. 2006). Ban et al. (2017) also study how to optimize the procurement of a new short life cycle product. Although their main contributions lie in the optimization of the multi-stage demand learning and sourcing planning, they similarly use past similar products and product characteristics. Diermann and Huchzermeier (2016) show how a new demand forecasting model significantly improved planning at the internet retailer Canyon Bicycles. The method adopts a bottom-up approach based on teams' estimates of individual SKUs. Chen et al. (2013) analyze data from the social media company Twitter; they utilize a latent source model to classify time series and then predict which topics will be popular in the future.

1.4. Integrated PLC Forecasting and Operational Planning and Execution

Hayes and Wheelwright (1979) advocate how to link the manufacturing process and PLC, and a large literature has coupled forecasting with operational planning and execution. For example, Fisher and Raman (1996), Kurawarwala and Matsuo (1996), Zhu and Thonemann (2004) and Gallien et al. (2015) analyze joint forecasting and inventory decisions. As do we, Kurawarwala and Matsuo (1996) consider computers and forecast PLC curves, but they focus on, and in-sample test, only four make-to-order products using only Bass curves. As we describe further below, our focus is on out-of-sample testing of our method's PLC curve forecasts of 170 make-to-stock products at Dell, where higher forecast accuracy would reduce expedited air transportation in favor of ocean transportation. For this situation, integrated forecasting and dual sourcing models are desired; Boute and Van Mieghem (2014) may provide some initial ideas by coupling exponential smoothing with dual sourcing. Forecasting over the PLC implies non-stationary demand, and Graves (1999) addresses (single-sourcing) inventory management for non-stationary demand. Our paper focuses on PLC forecasting; future work will integrate with operations planning execution.

2. Context and Business Environment at Dell

Our industrial partner, Dell, is the third largest producer of personal computers globally. Historically, Dell has fulfilled personal computer demand using a configure-to-order (CTO) approach. With such an approach, forecasts and inventory decisions must be made at the component level. Kapuscinski et al. (2004) provide an overview of Dell's CTO operations and discuss the challenges of forecasting and managing component inventory to support CTO fulfillment. In recent years, Dell made the strategic decision to shift to fulfilling significant volume with a make-to-stock (MTS) model. Products selected to be managed as MTS span multiple product categories (e.g., laptop, fixed workstation) and multiple target markets (e.g., business and consumer). The intent is to select products to be managed as MTS where customers may value a simplified ordering process and fast delivery over the ability to customize their product. Some MTS products are available on Dell's website under a program called Smart Selection with the stated aim to provide "a simplified ordering process for our best value, prebuilt systems custom-designed based on customer feedback."[1]

Our data set, described in further detail below, is for North America only although Dell uses this MTS approach globally. For North America, the most cost effective product flow for Dell is to have their contract manufacturing partner in China produce and ship products via ocean—with an 8 week lead time—into fulfillment centers in the United States. For laptops, air freight from China can be used for faster delivery, but at substantially higher expense. Desktops can be delivered to U.S. fulfillment centers more quickly by having them produced in Mexico rather than China, but this results in substantially higher manufacturing cost. In both these cases, the additional transportation or manufacturing cost associated with faster delivery may make adopting the MTS approach financially unattractive; thus, generating accurate forecasts is critical.

The forecasting process for a new MTS product starts with a product team providing a projected life cycle length, a launch date, and a weekly baseline demand estimate for the maturity stage. Using these inputs, on a quarterly basis, a demand planner creates a weekly forecast over a fixed horizon that is in general less than the full life cycle length. The planner may then include adjustments for known seasonal effects, planned promotions or sales initiatives, ramp up (introduction or launch) and ramp down (decline or end of life) stages, and the potential impact of the introduction of other new products that may cannibalize demand. Planners may also examine orders for similar products from the past in creating the weekly forecast. We expect promotional effects to be somewhat limited in our data set for two reasons. First, the majority of the products in our data set are aimed

[1] `http://www.dell.com/learn/us/en/04/campaigns/smart-select-consumer?c=us&l=en&s=dhs` accessed on April 28, 2017.

at business customers where promotions are limited. Second, for products targeted at consumers, promotions can occur, but these plans may not be known at the time of product launch in which case they would not affect the *initial* forecasts made, and it is these initial forecast we use for comparison. Since we do not have information regarding promotions and cannibalization, we do not incorporate these effects into our PLC fitting and forecasting process.

3. Data

Our data set includes weekly North American customer orders and forecast data for 170 complete product life cycles from February 2013 until November 2016. The orders cover 4,037,825 Dell computers and several billion dollars in revenues. While our initial raw data set included more SKUs, we removed SKUs whose full life cycle extended beyond the data set (that is, the life cycle started before February 2013 or ended after November 2016), whose life cycle was fewer than 13 weeks, or whose mean weekly volume was fewer than 20 units. Our approach is more tailored towards forecasting medium to high volume SKUs and whose life cycles are significantly longer than the 8-week lead time required to ship products via ocean from China. These 170 SKUs represent 80% of the original count of SKUs and 94% of the original total order volume.

We first provide a summary of this data, discuss its limitations, and then describe how we clean and prepare it for analysis. Cleaning and preparation of the data is necessary so that the PLC curves are normalized: this allows PLC curves to be compared to each other regardless of total volume or length of the PLC.

In order to provide the research community access to actual industry PLCs, we make our scaled order data available at `www.vanmieghem.org`. This URL also contains data on which SKUs had outlier adjustments, negative order corrections, and end-of-life adjustments.

3.1. Overview

These 170 products belong to one of four product categories—laptops, desktops, mobile workstations, and fixed workstations—and are all managed with a MTS model. For nine products we do not have information regarding their categories; we group these together into a fifth category called 'unknown.' Dell holds all these 170 items in its fulfillment centers to serve individual consumers, institutions, and retailers. There is one exception: this data set also includes some large orders requested by large organizations. Often, these large orders are not filled from stock at fulfillment centers but rather added to the production schedule at a long lead time to the customer in a build-to-plan workstream. We try to filter these large orders out (see discussion below in Section 3.3) as our intent is to forecast MTS customer orders satisfied from Dell's fulfillment centers. We share summaries of the data's volume and launch distribution in Tables 1 and 2 respectively. (Note that these tables show information on the data after preparation and cleaning; see Section 3.3 below for preparation details.)

SKU Set	SKU Count	Weekly net customer orders (percentile)			Number of weeks of PLC (percentile)		
		25th	50th	75th	25th	50th	75th
All	170	170	297	576	30	40	49
Fixed Workstation	7	43	52	61	30	41	54
Laptop	22	289	401	522	31	38	40
Mobile Workstation	5	85	101	114	27	32	34
Optiplex Desktop	127	196	357	728	32	44	51
Unknown	9	77	163	219	10	10	11

Table 1 Summary statistics of the data after preparation (see Section 3.3 below for preparation details). (For 'weekly' net customer orders, the 25th, 50th, and 75th percentiles are over the observations for that product across the periods in its own life cycle.)

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Number launched	10	47	3	2	2	26	21	2	17	6	30	4

Table 2 Distribution of number of new products' launches across different calendar months. Launches are distributed across different months, which implies that the life cycle curves we observe are not due solely to seasonal patterns.

We have varying levels of information for each of the 170 items, summarized in Figure 4. In addition to the product category, for 45 of the 170 products we have Dell's *initial* forecasts. By *initial* we mean the first pre-launch forecast created for a SKU, which is updated on a quarterly basis. This forecast is generally for about 8 months into the future (34 weeks). However, the actual number of weeks for which we have an initial forecast may range from 12 to 45 weeks due to launches not occurring at the start of the quarter or other forecasting anomalies. The $\{min, 1stQuartile, median, 3rdQuartile, max\}$ of the number of weeks for which we have an initial forecast across these 45 SKUs are (respectively) $\{12, 34, 35, 35, 45\}$. Thus, at week 0 (time of launch) we have the point customer order forecasts for week $1, 2, \dots$ until week $w_{Initial}$ where $w_{Initial} \in \{12, \dots, 45\}$. At week 13 (a quarter later), Dell updates its forecast and thus we have forecasts for the next 8 months, starting in week 14. Dell updates these initial horizon quarterly forecasts until the end of each product's life cycle. We denote the set of the initial forecast together with these forecasts that follow as *updated*.

For 86 products, we also know key features, such as the processor, RAM, hard drive size, and optical drive type. Later in this paper, we compare clustering products by feature (processor, RAM, etc.), category (fixed workstation, laptop, etc.), as well as by data-driven (using time-series) clustering in order to generate forecasts. For the products for which we also have Dell's forecasts, we can compare our forecasts to Dell's: both the *initial* forecast as well as the *updated* full life cycle forecast. For those products without Dell's forecasts, we compare different methods of clustering (feature, category, and data-driven) and those methods' impact on forecast quality.

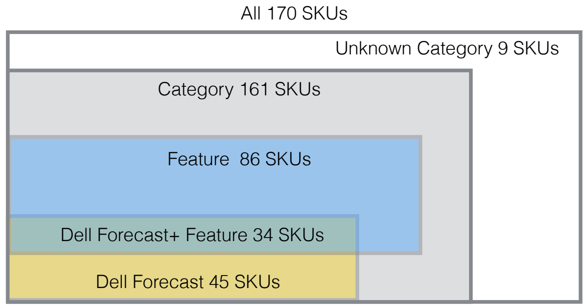

Figure 4 **Summary of levels of information we have for the products in our data set.**

Since our approach works with normalized PLC curves, we require estimates of PLC length and volume in order to generate forecasts from a PLC curve. At Dell, and presumably with many other technology firms, the launch and end-of-life dates for a product are planned and often coincide with a technology roadmap (that may align with a roadmap from a key supplier such as Intel). Thus, PLC lengths are known with a relatively high degree of certainty. In practice, lifetime volumes can be estimated in multiple ways, including using volume information from predecessor products, estimates from marketing or product planning teams or estimates from a forecasting team. Below, we use initial forecasts from Dell as the basis for the estimate of lifetime volume. In Section 7 we generate forecasts for a computer accessories firm. In this setting, we do not have initial forecasts, so we use PLC length and volume from the known predecessor product to generate forecasts.

3.2. Limitations

The data we have is the same data available to Dell's demand planners. We note here some limitations.

1. Net customer orders only: For each week, we observe the sum of total orders placed minus returns and cancellations in that week. We do not know the true total customer orders in a given week, nor do we know the breakdown (whether it was one large order or several small orders). Additionally, if a large order is placed in one week and returned or cancelled the next week, it may lead to negative net customer orders observed in the following week. We discuss how we treat this below in Section 3.3.

2. Censored demand: We observe only customer orders and do not have access to inventory information. From the customer point of view, if an item is not in a fulfillment center she will not see the item as explicitly out of stock. Rather the lead time will be longer for items which are not in the fulfillment center. The customer may or may not decide to continue placing her order. Due to data limitations, we ignore potential censored demand.

3.3. Preparation

For each product, we have access to the total net customer orders for each week of its life cycle. Thus, before forecasting the customer orders, it is necessary to prepare the historical raw data to address the phenomena of negative orders, large orders, and managed end-of-life behavior. Figure 2 in the introduction above summarizes our overall approach including treatment of the raw data. In developing these steps of cleaning and treating the data, we met with a demand planner at Dell who informed us as to the root cause of the phenomena and helped guide us as to the proper treatment of these phenomena.

Promotions and seasonality. As noted above, since we do not have promotional information, we do not include that step in data preparation for this Dell data set. Additionally, we do not adjust for seasonality. The limitations of our data set make incorporating seasonality difficult. Products in different categories for different markets have different seasonality patterns. For instance, different industrial customers and governmental organizations have different fiscal years; thus the phenomenon of 'end-of-fiscal-year-buying' will occur in different (but perhaps fixed) months throughout the year. Individual customer orders may be driven by holiday gift-giving or 'back-to-school' shopping. Industrial and governmental organizations may purchase more workstations and fewer entertainment-focused laptops as compared to individual customers.

We do not attempt to adjust for seasonality for two reasons. First, we do not know which products are aimed at which of these markets; thus, we do not know which subsets of products to group together to try to estimate seasonal effects. Second, we have no individual product life cycle that covers two complete years, so we cannot estimate any seasonal effects at the product level. As a robustness check, we applied a multiplicative seasonal effects model whose results we report in our robustness section. However, due to the different seasonal patterns of different products, the forecast quality of the seasonal model we implemented was worse than the same approach ignoring seasonality.

Although we do not adjust this Dell data for promotions or seasonality, we retain these steps in Figure 2 as they may be important in applying our approach to other firms or settings. For instance, in our case study of a computer accessories manufacturer in Section 7, we show how seasonality factors provided by the firm can be incorporated into our forecasting approach and indeed—because seasonality at this accessories firm plays such a large role—is necessary to generate good forecasts. We include the details of the seasonality step in Appendix A.

Detecting cancellations. In the raw data, we observe several negative net customer orders for various products. Let D_t^i be the raw observed net customer order value in week t for product i. T_i is the length of the life cycle (in weeks) of product i, and $t \in \{1, \ldots, T_i\}$ is the product-specific week relative to that product's launch week. The total observed customer orders for a product in week t

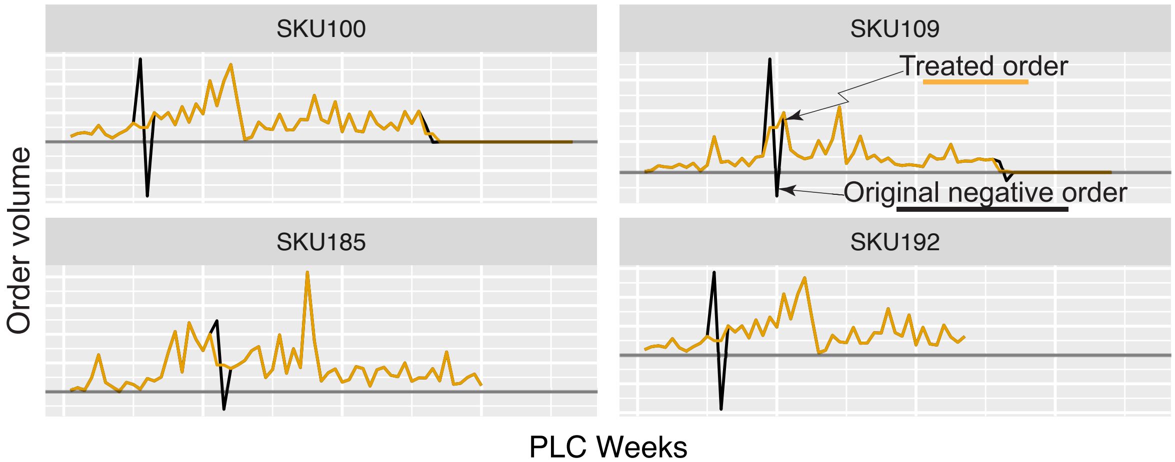

Figure 5 Illustration of our negative order correction method.

is the sum of all the actual customer orders minus all the order cancellations and returns for that week. Returns are relatively rare, and the negative data points suggest that an order which was placed in an earlier week is being cancelled. We 'correct' these cancellations by implementing the following algorithm which smooths out negative orders while keeping constant the total lifetime order amount. First, we move from the last period T to the first period 1 time-step by time-step. If we come across a negative order at time t, then we set this time period's order and the previous period's order equal to the simple average of their old orders: $D_t^{Back[1],i} = D_{t-1}^{Back[1],i} = \frac{D_t^i + D_{t-1}^i}{2}$, where $Back[1]$ denotes we are moving backwards in time on the first iteration. After this first pass, negative orders may still exist: for instance, if a large negative cancellation was really tied to an order placed more than one week ago. If this is the case, then move forward in time time-step by time-step. If a negative order is encountered on this pass at time t, set $D_t^{Front[1],i} = D_{t-1}^{Front[1],i} = \frac{D_t^{Back[1],i} + D_{t-1}^{Back[1],i}}{2}$, where $Front[1]$ indicates this is the first forward pass. If necessary, repeat m times until no negative orders are found (for this data set, we never had to perform more than one backward and one forward pass). Then, set $D_t^{C[1],i} = D_t^{Front[m],i}$, where $C[1]$ denotes cleaning step number 1. Figure 5 shows four SKUs on which we apply negative order treatment.

Excluding build-to-order customer orders (Outlier correction). We focus on forecasting PLCs for MTS products whose orders are filled from on-hand inventory when there is not a stockout. However, if an individual customer order is large enough, it will be moved from the MTS workstream to the build-to-order workstream, thereby incurring a longer lead time which includes production and transportation. Because these orders do not draw on on-hand inventory, we want to remove them. We do not know exactly which portion of a week's customer orders was due to these very large orders. As a proxy, we identify weeks with very large customer order totals using outlier detection, assuming that this outlier is actually mostly made up of a very large order with a different

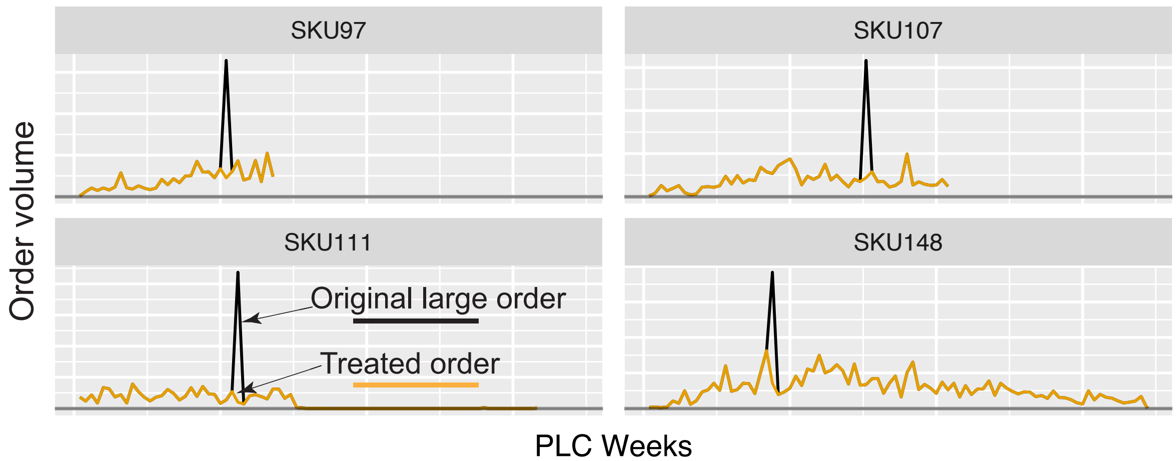

Figure 6 **Illustration of our large-order treatment method on four SKUs. The outliers denoted by the black lines (one outlier in each subfigure) are detected and replaced by the value as defined by Equation (1).**

workstream. We replace these outliers with ‘reasonable’ values (defined below) because once the build-to-order workstream units are removed, there are still likely underlying MTS customer orders. We identify outliers by the time series outlier detection method described in Chen and Liu (1993). In essence, the method outlined in that paper fits a time series model to the data and then identifies outliers significantly deviating from this time series model. Thirteen orders are identified as outliers across 13 SKUs. We replace the detected outliers for product i at week t with their weighted moving averages, as outlined in Roberts (2000). Figure 6 shows examples of outlier correction for four SKUs. Recalling that T_i denotes PLC length of product i, we set:

$$D_t^{C[2],i} = \frac{D_1^{C[1],i} + 2D_2^{C[1],i} + \cdots + (t-1)D_{t-1}^{C[1],i} + (T_i - t)D_{t+1}^{C[1],i} + \cdots + D_{T_i}^{C[1],i}}{(1 + \cdots + t - 1) + (1 + \cdots + (T_i - t))}. \qquad (1)$$

End-of-life truncation. Customer orders near the end of the life cycle can be strongly influenced by managerial decisions such as promotions or timing of the introduction of a new product intended to replace an old one. Since we seek to focus on forecasting the “naturally occuring” product life cycle, rather than orders that occur during an actively managed end-of-life, we exclude customer orders near the end of the life cycle. In our data set, the length of the product life cycle can be artificially extended past when a product’s life has essentially ended, since a single customer order may occur or a return or cancellation may be made weeks later. Figure 7 shows the first behavior (managed end-of-life) in the left subfigure and the second behavior (artificially extended PLC) in the right subfigure.

In addition to managerial decisions and single orders or returns affecting end-of-life orders, the end-of-life is ‘far away’ at time 0 (launch) and thus there is less value to forecasting it well versus forecasting the nearer term majority of the PLC. Let θ^t be the fractional portion of the length of the beginning of the life cycle we keep, and let θ^v be the fractional portion of the beginning life

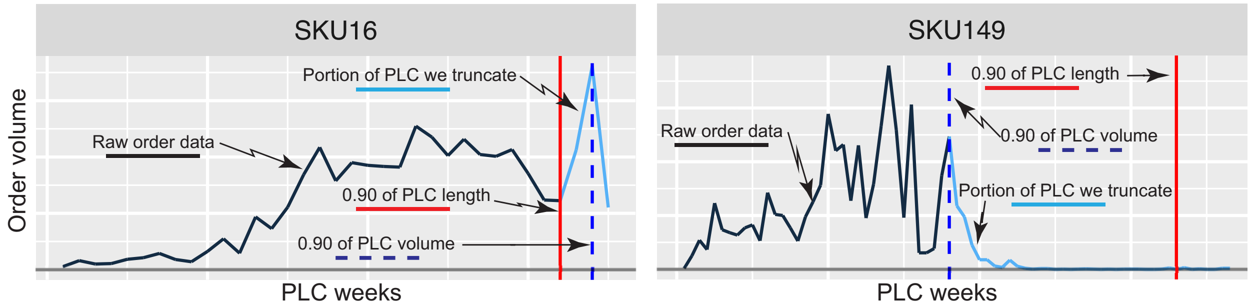

Figure 7 Illustration of our end-of-life truncation approach. The solid red line represents the cut-off based on PLC length, while the dashed blue line represents the cut-off based on volume. Our method cuts off all data points that occurred after the earliest cut-off point from either method.

cycle order volume we keep. Thus, we exclude the last $1-\theta^t$ fraction of the weeks of the products' PLCs and we exclude the last $1-\theta^v$ fraction of the total volumes of the products. We initially set $\theta^t = \theta^v = 0.9$. Figure 7 shows how the time cutoff and the volume cutoff affect the raw data, where the time cutoff dominates in the left subfigure and the volume cutoff dominates in the right subfigure. The processed data is now defined as $D_t^{C[3],i} = D_t^{C[2],i}$ for t satisfying the above criteria. We note that even though we examine products whose life cycles are longer than 12 weeks, by truncating the end of life of products some resulting life cycles in our data set may be less than 12 weeks. T_i is redefined to be a smaller value as appropriate to account for the cut off data points.

Normalization of data After data preparation, we normalize the data so that for each product, the lifetime cumulative sum of customer orders is equal to 1. Thus, we obtain the normalized customer order series $\tilde{D}$, with $\tilde{D}_t^i = D_t^{C[3],i}/D^{C[3],i}$. The lack of a t subscript denotes summation: $D^i \equiv \sum_{t=1}^{T_i} D_t^i$. Thus, $\sum_t \tilde{D}_t^i = 1 \quad \forall i$. All the following analyses are carried out based on this normalized data series $\tilde{D}$. In this way, PLCs for products with dramatically different volumes or launch seasons can be compared with each other.

4. PLC Curves Fitting

Having cleaned and normalized customer order data, we can proceed with fitting life cycle curves to these data for each product. Previous literature has suggested the following three families of curves: n^{th}-order polynomial curve (poly-n), Bass diffusion curves (Bass), and piecewise-linear 'curves' (trapezoids and triangles as their subset). We test and compare models from all three families, comparing both fitting accuracy and model complexity. Figure 8 shows one example product (SKU150, the same product in Figure 1) fit by each of these curves.

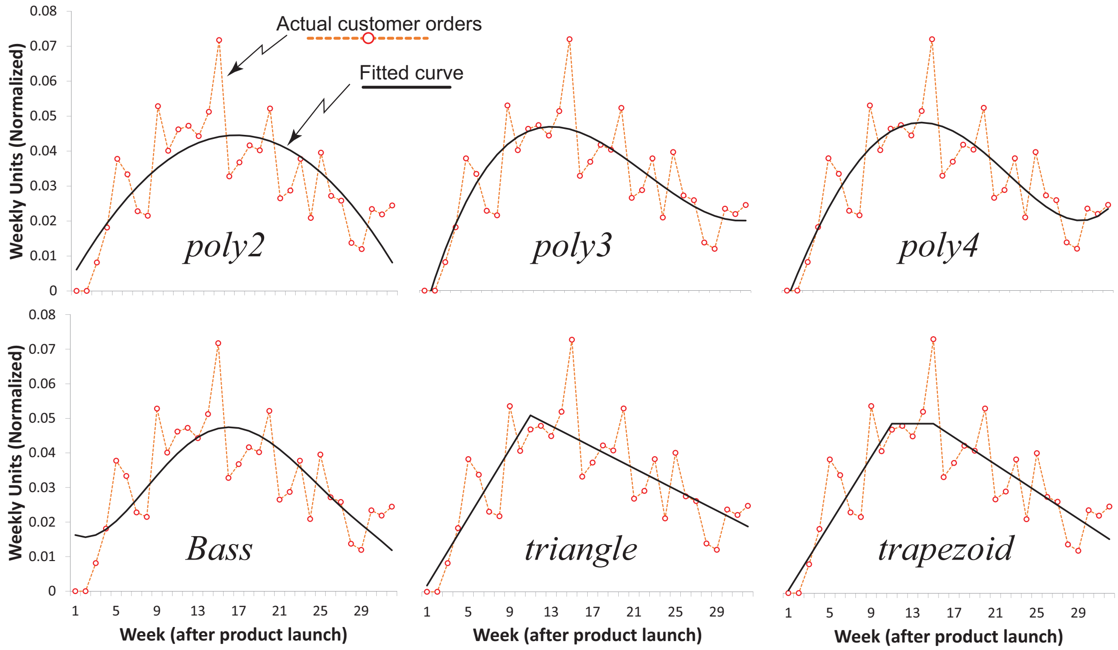

Figure 8 **Six PLC curves fit to one product (SKU150). A second order polynomial and the Bass curve overestimate demand in first few weeks. The fourth order polynomial might be overfitting the last few weeks of the life cycle. While the trapezoid curve allows for a maturity stage, it is very short and visually it is difficult to identify that a clear maturity stage even exists (from the firm's point of view). Visually, the third order polynomial and triangle seem to provide 'good' fits in this example.**

We first describe the curve families, then relate the quality of the fits of each curve family on our data. However, we note that the curve family that best fits the data (even when fit is measured using criteria that penalize complexity) may not lead to the curve family that generates the best forecasts or the curve family that is most managerially friendly. The fit results below in this section are not intended to lead to a decision as to which curve family to use for forecasting; we revisit the forecast quality (the main metric we are trying to optimize in this research) associated with each curve family later in Section 5.

4.1. Overview of PLC category families

The Bass diffusion model uses three parameters (p, q, m). The parameter m represents lifetime volume, which we normalize to 1. The resulting normalized, continuous-time Bass model has instantaneous customer order rate $\eta(t) = p + (q-p)N_t - q(N_t)^2$, where $N_t = \int_{s=0}^{t} \eta(s)ds$ is the cumulative sum of customer orders up to time t, and p and q are shape parameters. Kumar and Swaminathan (2003) present the solution of the differential equation:

$$\breve{D}_t^{Bass} = \frac{p(p+q)^2 e^{-(p+q)t}}{(p+qe^{-(p+q)t})^2} \tag{2}$$

The $\breve{\cdot}$ notation denotes a customer order estimate provided by a particular PLC shape whose cumulative volume is 1.

The family of piecewise-linear curves fit the PLC with connected straight line segments. We explore two types of 'curves' in this family: the triangle (using two connected line segments) and the trapezoid (using three connected line segments with the middle segment forced to be flat). The triangle is suggested by Goldman (1982) and the trapezoid allows identification of maturity stage. The below piecewise-linear functions are valid for any lifetime sum of customer orders. When we force lifetime sum of customer orders to be 1, in terms of model complexity, the triangle and trapezoid functions will have 1 less free parameter each.

The triangle PLC requires four[2] parameters (a, b, c, τ), and is defined as:

$$\breve{D}_t^{triangle} = \begin{cases} at + b & 0 < t < \tau \\ c(t - \tau) + (a\tau + b) & \tau \leq t \leq T_i \end{cases} \tag{3}$$

where $\breve{D}_t^{triangle}$ is the order at time t, τ marks the period of transition and (a, b, c) are shape parameters.

The trapezoid is defined by five parameters $(a, b, c, \tau_1, \tau_2)$ which characterize the PLC as:

$$\breve{D}_t^{trapezoid} = \begin{cases} at + b & 0 < t < \tau_1 \\ a\tau_1 + b & \tau_1 \leq t < \tau_2 \\ c(t - \tau_2) + (a\tau_1 + b) & \tau_2 \leq t \leq T_i \end{cases} \tag{4}$$

where $\breve{D}_t^{trapezoid}$ is the customer order at time t, τ_1 and τ_2 mark the beginning and the end of the maturity (flat) stage and (a, b, c) are shape parameters: a and c are two slopes and $\breve{D}_t^{trapezoid}$ will equal $a\tau_1 + b$ at transition time τ_1.

The family of polynomial curves capture the PLC with smooth curvature. According to the literature (Crompton and Bonk 1978, Crompton 1979, Headen 1966), polynomial functions up to fourth orders are sufficient to capture a wide range of PLC curves. The n^{th} degree polynomial PLC curve is:

$$\breve{D}_t^{poly-n} = \sum_{i=0}^{n} a_i t^i \tag{5}$$

where $\breve{D}_t^{poly-n}$ is the order at time t, and a_i for $i = 0, \ldots, n$ are the shape parameters. In terms of model complexity, when lifetime volume is forced to equal 1, an n^{th} order polynomial will have n free parameters.

[2] A horizontal triangle starting at $t = 0$ and returning to 0 at T_i requires only 3 parameters. Based on the data, we do allow the start and end values to differ from 0 leading to 4 parameters.

	poly2	poly3	poly4	Bass	triangle	trapezoid
RMSE						
- mean	0.0192	0.0179	$\boxed{0.0169}$	0.0187	0.0174	$\boxed{0.0170}$
- 1st quantile	0.0078	0.0077	$\boxed{0.0073}$	0.0080	0.0075	$\boxed{0.0073}$
- median	0.0113	0.0109	$\boxed{0.0106}$	0.0113	$\boxed{0.0106}$	$\boxed{0.0106}$
- 3rd quantile	0.0171	0.0169	$\boxed{0.0159}$	0.0181	0.0167	$\boxed{0.0166}$
Loglikelihood (sum over SKUs)	21,284	21,514	$\boxed{21{,}745}$	21,267	21,556	$\boxed{21{,}729}$
AIC	-41,549	-41,669	$\boxed{-41{,}790}$	-41,515	-41,753	$\boxed{-41{,}758}$
Number of parameters	3	4	5	3	4	5
For how many SKUs is each curve the best fit						
-RMSE	-	-	68	1	23	78
-AIC	39	20	27	31	39	14

Table 3 Summary statistics of PLCs' fits to the data. Boxes with a double (single) outline denote the best (second best) value in each row. The poly4 curve fits the data the best. Trapezoid is the best curve (smallest RMSE) for the most number of SKUs without considering complexity, while poly2 and triangle are the best (smallest AIC) for the most number of SKUs when complexity is considered.

4.2. Quality of PLC fits

For each product, we fit all candidate PLC curves to its processed and normalized customer order data. The parameters for each curve are found through minimizing the root-mean-squared error (RMSE) of the candidate PLC applied to the customer order values across the weeks of the product's life cycle.[3] We utilized RMSE to not only fit the PLC curves, but also to evaluate the quality of the fit. In order to adjust the quality of fit for the model complexity (that is, the number of parameters) we report a measure intended to do just that: Akaike information criterion (AIC), defined as $2k - 2Loglikelihood$ where k is the total number of parameters to be estimated across all SKUs. (Recall that minimizing RMSE is equivalent to maximizing loglikelihood when errors are normally distributed.) For AIC, smaller values (in general, more negative) imply a better model. We summarize the performance of curve fits in Table 3.

We observe that smooth curves fit marginally better than piecewise-linear curves, when measuring either pure fit (RMSE and loglikelihood) or a fit penalizing complexity (AIC). In terms of unpenalized raw fit, triangles and trapezoids are the best curve for $(23+78)/170 = 59\%$ of the products. When complexity is accounted for, most of the 170 SKUs are somewhat evenly distributed among poly2, triangle, Bass, and poly4, with poly3 and trapezoid assigned only 20% of the 170 SKUs in total.

[3] The poly-n curves are fit using ordinary least squares regression, while Bass and piecewise linear curves are based on an optimization routine, both of which are implemented in R (R Core Team 2017).

While poly4 is the best fitted curve family for all measures, trapezoid is the second best for each of the same measures (and dominates for forecasting, as we will see in the next section). In addition, trapezoids and triangles (a special case of trapezoids) have two additional advantages that are important in practice: they are intuitive to explain and estimate. One needs only two slopes (growth and decline rates), and either one or two turning points/times. This intuitive interpretation is helpful as our approach must be integrated into demand planners' workstreams given that demand planners must provide life cycle lengths, volumes, and assign clusters as outlined below. This concept of three stages is already incorporated into Dell's planning and forecasting.

Polynomial curves, on the other hand, 'look nice and smooth' but are not as easy to interpret from a managerial perspective. It is not clear—without plotting the curve—what impact each parameter has on a product's life cycle shape. Thus, from a managerial perspective, the piecewise linear curves are more desirable and any degradation in fitting or forecasting ability must be balanced against ease of interpretation. In the next section (evaluating forecast quality) we will compare all curve types in terms of their ability to accurately predict demand. We show below in Section 5 that despite the superior fit of the poly4 curves to the raw data, trapezoid curves lead to better forecasts.

To investigate the length of the maturity stage of product life cycles, we fit a trapezoid and measure the relative length of its middle, flat portion. Figure 9 shows the distribution of this relative length. About 20% of product life cycles have no maturity stage (and are best fit by a triangle). Half of the products have maturity stages less than 13% of the entire life cycle length and three quarters of products have maturity stages less than 35% of the entire product life cycle length. Of all the weeks (across all 170 SKUs) that we observe in our data set, only 22% of those weeks are in a maturity stage. This lack of a maturity stage has implications on operational planning because many of the most common inventory models used in practice assume stationary demand. However, we show in this faster-paced technology environment that stationary demand models may not be appropriate most of the time.

5. PLC Forecasting

Armed with the PLC curves of past products, we can generate a normalized forecast for a new product that is similar to past products; i.e., the new product's normalized PLC curve is similar to those of past products. To operationalize the notion of "similar past products" we cluster the past similar products. Our overall approach for generating a forecast for a specific new product can be broken into three steps:

1. Choose a set or 'cluster' of similar products. As similarity metrics we consider:
    (a) *Features* such as RAM, processor, hard drive size, etc.

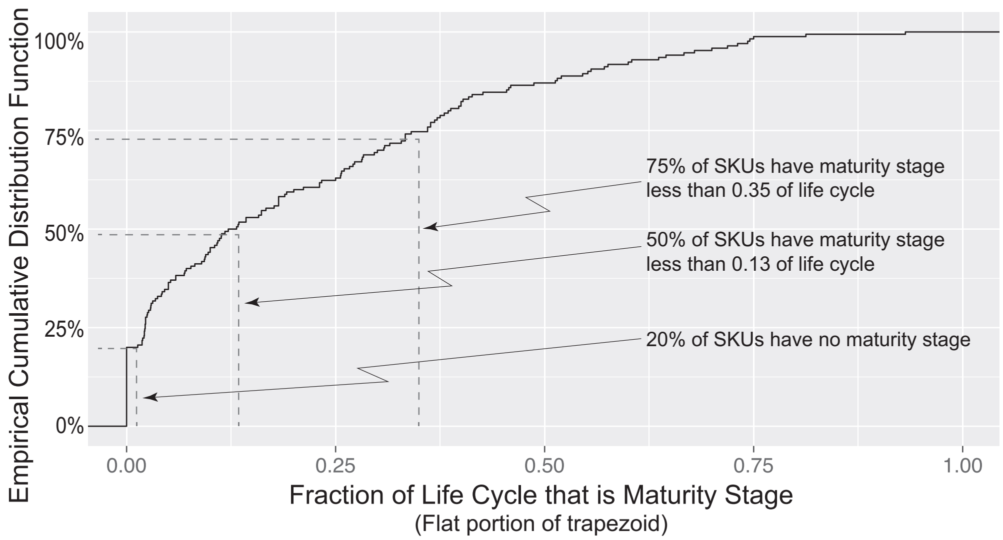

Figure 9 **About 20% of the 170 product life cycles have no maturity stage, as calculated by the relative length of the flat middle segment in its best-fitted trapezoid. Three quarters of the products have maturity stages of less than a third of their entire life cycle length.**

(b) Product *category* such as fixed workstation, laptop, desktop, etc.

(c) A specific distance metric used in a *data-driven* clustering algorithm explained below.

The efficacy of these three clustering methods is compared to using no clustering; i.e., assuming that all products are homogeneous and fall in one cluster.

2. For each cluster, generate a single representative "cluster PLC curve" using one of two methods:

(a) Take the average of all the curves in a cluster;

(b) Fit the best curve (from a family selected by the user) to the set of products' data points in the cluster.

Note that a cluster need not contain PLCs with the same functional curve. The average can be taken over, say, trapezoids and poly4 curves in one cluster. Fitting the best curve from a given family, however, makes most sense when all curves in that cluster are from the same family.

3. Identify the cluster of the new product and use its cluster PLC curve to generate the forecast by adjusting the cluster curve with estimated PLC time and volume, as provided by a demand planner, and adding seasonality and other known (promotion) effects if applicable. More generally than identifying a specific cluster, one could have a probabilistic identification that the new product belows with probability p_i to cluster i and then use the probability-weighted average cluster curve.

We first describe the three different ways of clustering. We then outline how we translate a cluster of curves into a forecast for a specific SKU.

5.1. Step 1: Choose a set or cluster of similar products

Among the three clustering approaches that we will use to test the performance of our forecasting method, 'features' and 'category' clustering are what demand planners at various organizations may be practicing currently. Data-driven (using time-series) clustering, which can be automated, is what we propose as part of our new approach.

5.1.1. Features: For a subset of 86 products, we know attributes such as processor, hard drive size, RAM, and optical drive type. We aggregate the products with the exact same values of these attributes into the same cluster. This results in 20 clusters with cluster size ranging from 2 to 13 SKUs.

5.1.2. Category: For all 161 products with known category, we know which of the following product categories they fall into (numbers in parentheses denote the number of SKUs in each category): fixed workstation (7), laptop (22), mobile workstation (5), desktop (127). We consider each category as a cluster.

5.1.3. Data-driven clustering using time-series: Each PLC curve is in essence a time series and thus we can utilize time-series clustering as outlined in Chouakria and Nagabhushan (2007). This clustering method is based on proximity of both scale and behavior. The authors present a distance measure to address the proximity of magnitude in two time series at the same point in time as well as the temporal correlation for behavior similarity. We outline the exact implementation of their ideas as well as the selection of number of clusters in Appendix B. We note that their distance measure ($\delta_{CORT}(X_t, Y_t)$) is model-free: it allows us to cluster the fitted PLC curves based on their features in terms of temporal structure regardless of whether a polynomial, Bass, or piecewise-linear curve was used to model the PLC shape.

We note that in order to use any clustering, we first normalize the cumulative volume and the length of each PLC to 1 and 100 time periods, respectively. (If the PLC has less than 100 time data points, we use interpolation to fill in missing data points.) Thus a 30 week and 60 week PLC might be clustered together if they have the same shape, scaled by time.

Once a distance measure is established, we need to determine an appropriate number of clusters. One commonly uses a scree plot to select the optimal number of clusters. We first plot 'sum of squared distances within clusters' versus number of clusters in Figure 10. Using our judgement, we choose four clusters because beyond four, there is not much improvement in 'sum of squared distances' (although we test the robustness of forecast accuracy to number of clusters in the results section).

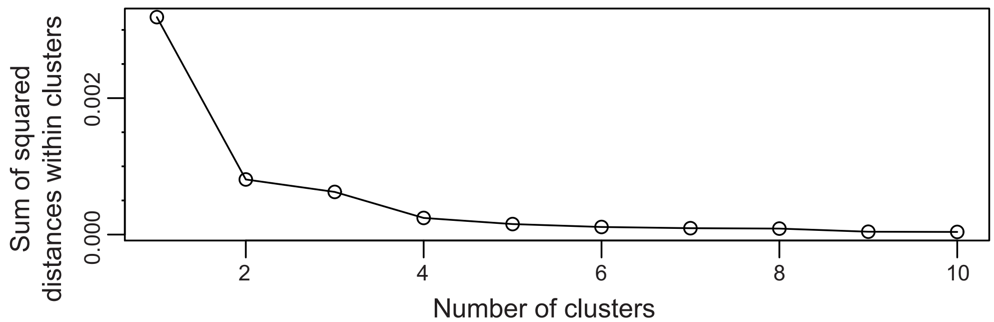

Figure 10 **Sum of squared distances ($\delta_{CORT}(X_t, Y_t)$) within clusters versus number of clusters. We chose four clusters because there is little reduction in sum of squared distances for values above 4.**

Cluster group	Product category: Fixed Workstation	Laptop	Mobile Workstation	Desktop	Unknown	Fraction of total
1	5	9	3	94	4	0.68
2	1	11		26	1	0.23
3	1	2	2	5	2	0.07
4				2	2	0.02

Table 4 **Breakdown of time-series clusters' product categories (which were not actually used in the clustering process). Some category-cluster pairs that emerge are consumer desktops and workstations in cluster 1 and laptops and desktops in cluster 2. Clustering was done using trapezoid curves for fitting.**

The results of time-series clustering is shown in Figure 11 which displays the four clusters when using trapezoid fits to all SKUs' PLCs (bottom), poly4 curves (middle), and $BestForEach$ curves (top) which uses for each SKU the best-fitted curve family (which results in lowest RMSE). Note that in this figure, while clusters generally line up across curve families (trapezoid cluster 1 looks similar to poly4 cluster 1), this will not always occur. Different curve families may lead to different SKUs being clustered with each other using the time-series algorithm; thus a SKU that is in cluster 1 under trapezoid may be under cluster 4 under poly4. A product's fitted curve may shift clusters when using different functional forms for fitting.

Tables 4 and 5 show that overall, products with differing attributes are spread across the time-series clusters. This suggests that time-series clustering may be able to identify hidden product attributes not represented in the raw data itself and possibly not known to demand planners. Thus, the mere act of clustering (without forecasting anything yet) may by itself be beneficial to firms in forecasting "almost-new" products.

Table 6 shows for three of the clustering approaches outlined here (feature, category, time-series, excluding 'overall') the breakdown of count, volume, and PLC length.

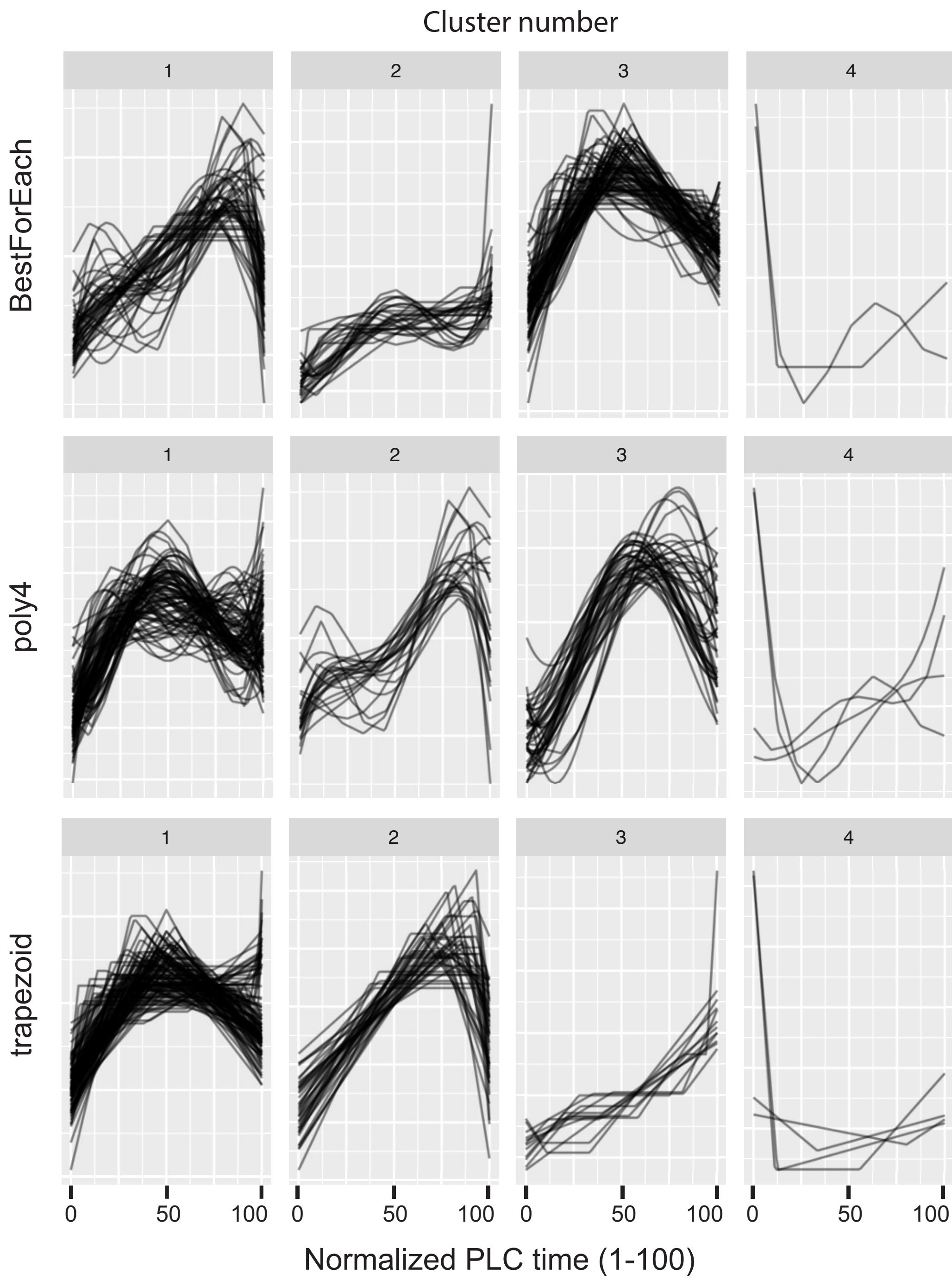

Figure 11 **Illustration of time-series clusters based on three imposed curve families: trapezoid, poly4, and the** $BestForEach$ **curves. Regardless of the curve family imposed, similar overall shapes are clustered together: cluster 1 peaks early- to mid-life cycle and its ending volume is higher than the starting volume; cluster 2 peaks later in the life cycle and its peaks are more pronounced; cluster 3 tends to increase in demand throughout the life cycle; cluster 4 tends to start with high demand and then stabilize (except for poly4).**

Cluster group	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec
1	0.38	0.11	0.23	0.27
2	0.23	0.33	0.28	0.15
3	0.42	0.25	0.17	0.17
4	0.50	0.25	0.00	0.25

Table 5 Breakdown of time-series clusters by launch month. Clusters 1, 3, and 4 tend to have January to March launches while other clusters' launches are either spread out across the year or slightly concentrated in July to September. Clustering was done using trapezoid curves for fitting.

Clustering method	Cluster group	SKU count	Mean total volume	Mean life cycle length (weeks)
Feature-Based 34 SKUs	i3/4/500/RW	12	30,113	41
	i5/4/500/RW	13	50,884	46
	i5/8/500/RW	11	58,344	53
	...	...	...	...
Product-Based 161 SKUs	Fixed WorkStation	7	2354	43
	Laptop	22	15,364	34
	Mobile WorkStation	5	3,101	31
	Desktop	127	28,767	43
Cluster-Based (Trapezoid) 170 SKUs	1	115	26,194	43
	2	39	25,528	38
	3	12	2,074	23
	4	4	1,265	11
Overall	1	170	23,752	40

Table 6 Breakdown of clusters by volume and life cycle length.

5.2. Steps 2 and 3: Generating forecasts from clusters

In order to forecast the weekly orders of new products, we will use the information from the fitted curves of similar products as well as the company's knowledge of launch date and its estimates of lifetime volume ($\hat{D}^i$) and life cycle length ($\hat{T}_i$), as mentioned in Section 3.1. Here, the $\hat{\cdot}$ notation denotes the company's estimates, not necessarily the true values. The exact steps we propose to move from the normalized PLC curve to actual forecast are as follows: generate a PLC curve from a given cluster; scale by time and add seasonality if applicable; scale by volume.

Step 2: **Generate the normalized PLC curve for the ready-to-launch product.**

We propose two options for obtaining the normalized PLC curve: averaging the curves within a cluster time-step by time-step or fitting a specific curve through all the superimposed datapoints of all curves within a cluster.

(a) **Taking the average of similar curves** (*GenerateAvg*)**.** Recall that for the normalized curves, regardless of the actual life cycle length, we discretize the curve into 100 time epochs via interpolation. Let us define $\bar{D}_t^{i,PLC}$ for $t=1,\cdots,T=100$ as the demand in epoch t for product i for $PLC \in \{trapezoid, poly4, Bass...\}$ where the $\bar{(\cdot)}$ notation denotes that the demand curve's volume has been normalized to total volume equal to 1 over 100 time epochs. Then, to take the average curve of a set, for each of these 100 time epochs we average over the curves in this cluster at that time point. We express this explicitly as

$$\bar{D}_t^{i,PLC,GenerateAvg} = \frac{\sum_{i' \in J_k(-i)} \bar{D}_t^{i',PLC}}{N_k - 1}, \text{ for } t=1,\cdots,T=100$$

where $N_k - 1$ is the number of fitted curves in cluster k not including product i and $J_k(-i)$ is the set of fitted curves in cluster k also not including product i. When generating a curve from a cluster for a specific SKU's forecast, we exclude that SKU's data. This is to generate an 'out-of-sample' PLC forecast curve whose shape is not directly influenced by the SKU whose life cycle curve it is estimating.

(b) **Fitting the best curve through the data points** (*GenerateFit*)**.** After clustering, we have $T=100$ time epochs and $N_k - 1$ curves for a total of $T \times (N_k - 1)$ datapoints (again, we exclude SKU i when generating a cluster's curve). To fit the best curve through these datapoints, we formulate an optimization problem that finds parameters in one of the equations (2) (Bass) through (5) (polyn) to minimize the resulting sum of squared errors of the $T \times (N_k - 1)$ datapoints relative to the resulting cluster curve. We add a constraint to this optimization problem to ensure that no segment of any curve is negative. This results in the curve $\bar{D}_t^{i,PLC,GenerateFit}$ for $t=1,\cdots,T=100$.

Figure 12 illustrates the two approaches (averaging or imposing a curve family) to generating a cluster curve from a given set of products for both the poly4 family and the trapezoid family.

Step 3: **Scale the time, add the seasonality effect (if applicable) based on the information of launch week, scale by volume.** We first scale this PLC shape to the actual life cycle length. Thus, $\bar{D}_t^{i,PLC,\eta}$ for $t=1,\ldots,100$ is transformed into $\breve{D}_t^{i,PLC,\eta}$ for $t=1,\ldots,\hat{T}_i$ by scaling the life cycle length by the *estimated* length of the PLC of $\hat{T}_i$ for product i. $\eta \in \{GenerateAvg, GenerateFit\}$ denotes the method of generating a representative curve from a cluster. With a weekly normalized forecast projected onto an actual calendar, this is where one would add a seasonality effect. The resulting curve is $\breve{D}_t^{i,PLC,\eta,Season}$ for $t=1,\cdots,\hat{T}_i$.

We then scale by the estimate of total volume $\hat{D}^i$. The forecast is then

$$\hat{D}_t^i = \hat{D}^i \frac{\breve{D}_t^{i,PLC,\eta,Season}}{\sum_t \breve{D}_t^{i,PLC,\eta,Season}}, \text{ for } t=1,\cdots,\hat{T}_i \tag{6}$$

where we suppress the PLC and η notation from the $\hat{D}_t^i$ as it should be obvious from context.

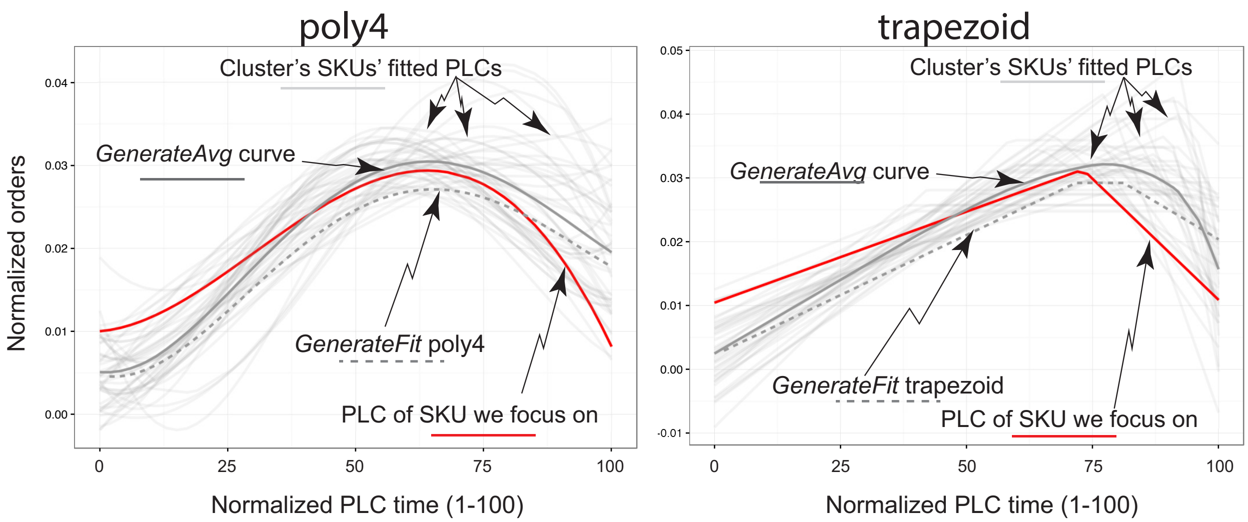

Figure 12 **Illustration of generating normalized forecast curves by either taking simple average over the cluster or fitting a best trapezoid or poly4 curve through all the fitted curves in the cluster.**

6. Out-of-Sample Forecast Evaluation

We evaluate the quality of the forecast $\hat{D}_t^i$ generated in the previous section using the following out-of-sample approach: Each product is part of the clustering process to mimic that the company would know which cluster each new product belongs to. As we mentioned above, in our evaluation of the forecast of product i, we remove that product from its cluster and then produce that cluster's representative PLC curve (out-of-sample) and use that curve as the normalized forecast for product i.

We first define how we measure forecast accuracy. We then measure forecast accuracy of our forecast versus Dell's internal forecasts on the 45 products for which we have forecasts. The section concludes with a comparison of the forecast accuracy of the four different clustering methods on a larger data set for which we do not have Dell's forecasts, but for which we can determine the drawbacks and benefits of the different clustering methods. Under all of these conditions, the time-series clustering approach results in the best overall forecasts.

6.1. Measuring forecast accuracy

We report on two measures of forecast accuracy: mean absolute scaled error (MASE) and sum of absolute errors (SAE). MASE is scale-independent and hence can be used to compare forecast accuracy across products. SAE provides a portfolio-wide metric but weights large-volume SKUs more heavily. Together, MASE and SAE provide a full picture of the forecast accuracy of different methods.

MASE is a measure of forecast accuracy proposed by Hyndman and Koehler (2006). It scales forecast errors by the errors resulting from a naive forecast in which a previous period's realized demand is used to predict demand in the current period. It is defined as such for SKU i:

$$\text{MASE}(i) = \frac{1}{T_i}\sum_{t=1}^{T_i}\left(\frac{|\hat{D}_t^i - D_t^i|}{\frac{1}{T_i-1}\sum_{t'=2}^{T_i}|D_{t'}^i - D_{t'-1}^i|}\right) = \frac{\sum_{t=1}^{T_i}|\hat{D}_t^i - D_t^i|}{\frac{T_i}{T_i-1}\sum_{t=2}^{T_i}|D_t^i - D_{t-1}^i|}, \tag{7}$$

where $\hat{D}_t^i$ is the estimate of demand and D_t^i is the observed demand.

MASE has the desirable properties that it is invariant to scale and it is not skewed when the data points are near 0. Mean absolute percentage error (MAPE), on the other hand, divides the period t error by period t demand and thus can be undefined (very large) when a period's demand is zero (near zero). We observe zero and near-zero demands in our data—it is non-stationary often ramping up from near-zero demand in the early weeks and often ramping down to near-zero in the end of life cycle—and thus MAPE is often undefined in our data set. Mean absolute error (MAE) is not scaled for volume or underlying variation, and so comparing MAE across products has little meaning. MASE is a nice compromise between being scaled (to enable cross-product comparisons) and providing reasonable values even for products with periods of no demand.

The sum of absolute errors (SAE) is defined as:

$$\text{SAE} = \sum_{\text{product } i}\sum_{t=1}^{T_i}|\hat{D}_t^i - D_t^i|. \tag{8}$$

SAE provides us with an unnormalized portfolio-wide accuracy metric which (because it is unscaled) places higher weights on higher volume products, longer life cycle products, and products with higher absolute errors (product types which may be of more interest to demand planners and managers).

6.2. Comparing our forecasts to Dell's

As mentioned above in Section 3, we have access to Dell's forecasts for 45 products. Demand planners at Dell typically forecast demand 34 weeks into the future from the start of a quarter, even if the life cycle is known to be longer than that. Each quarter when Dell planners update their forecasts, the new forecast is extended 3 months into the future, so that each *updated* forecast—even if it is made in the middle of a product's life cycle—is approximately 34 weeks into the future.

Given this workflow at Dell, we compare our forecasts to Dell's in two different ways. For either way, we first forecast a product's entire life cycle using our method. Then, we either compare Dell's *initial* forecast to the corresponding initial portion of our forecast, or we compare a 'stitched together' Dell's *updated* forecast (which we describe below) into a full Dell life cycle forecast to our entire life cycle forecast. Here, we emphasize that *updated* implies that Dell was able to update its forecast throughout the life cycle as it observed demand; our forecast, on the other hand, is static and makes an entire PLC forecast before the first demand is realized.

6.2.1. Comparison to Dell's initial forecasts across partial life cycle: We follow these steps in evaluating forecast accuracy in the initial stage of a product's life cycle (recall *initial* denotes the part of the life cycle for which we have a pre-launch forecast from Dell, which may be less than the full life cycle):

1. In week 0 (pre-launch), we have access to the company's initial forecast week by week (but made in week 0). We call these estimates $\hat{D}_t^{i,comp}$ for $t \in 1 \dots T^{i,\cap}$. (*comp* is company and $T^{i,\cap}$ is the minimum of the latest week of observed demand and latest week for which Dell has made a forecast for product i; that is, $\cap$ denotes the intersection of Dell's forecast horizon and the actual life cycle.) Note for this section $T^{i,\cap}$'s will in general (although not always) be shorter than the actual life cycle lengths.

2. In week 0, we have generated a PLC curve for each product that is normalized by volume and time for that product's entire life cycle. We compare several ways to generate this curve: base curve shape (Bass, poly-n, trapezoid, and $BestForEach$ in which each SKU is assigned the curve family with the best RMSE value), cluster type (for now we focus on time-series only), and method of generating a representative cluster curve ($GenerateAvg$ or $GenerateFit$). In order to provide a fair comparison so that our forecast is not advantaged over Dell's by knowing the actual volume, we scale forecasts in one of two ways. First, we scale our normalized PLC curve so that the volume in the *initial* weeks equals the *company's* estimates of volume during this same initial period ($ScaleForecast$). Second, to evaluate the quality of *shape* of our PLC forecast relative to Dell's forecast *shape*, we scale our PLC curve *and* Dell's forecast so that the volume in the initial weeks equals the true initial observed volume ($ScaleActual$). We test this second method because Dell's forecast may be significantly biased in one direction or another, resulting in poor forecasts for both Dell and for us. In either method, our *initial* volume equals Dell's *initial* volume.

Table 7 shows the summary of improvement in MASE and SAE across the SKUs. Clustering is done via data-driven time-series clustering. This table shows that our approach is robust both to modeling choice (which curve to impose, how to generate a curve), to method of evaluation ($ScaleForecast$ or $ScaleActual$), and to forecasting metric (MASE or SAE). The method that works the best is to use a trapezoid to represent each product's historical PLC, and then to generate forecasts with trapezoid shape to actually fit a trapezoid through the cluster's datapoints. Fitting the best curve to each product ($BestForEach$) works fairly well when forecasts are scaled by forecasted volumes, but results in low improvement when forecasts are scaled by actual volumes. Among the families of curves that we test, Bass yields the smallest forecasting improvement.

6.2.2. Comparison to Dell's updated forecasts across full life cycle: Here, we utilize Dell's updated forecasts across each product's life cycle. For each time epoch, we select Dell's latest forecast which includes that epoch. We call this Dell's *updated* forecast. We approach volume scaling in the same two ways described above ($ScaleForecast$ and $ScaleActual$), except that $ScaleActual$

Base curve family	Cluster curve generation ($GenerateFit$ or $GenerateAvg$)	Relative reduction in our forecast errors over Dell's *initial* forecasts			
		MASE		SAE	
		$ScaleForecast$	$ScaleActual$	$ScaleForecast$	$ScaleActual$
Bass	Fit	1.03%	1.13%	0.05%	0.34%
Bass	Avg	0.75%	0.75%	-0.06%	-0.04%
poly2	Fit	1.94%	2.48%	0.59%	2.09%
poly2	Avg	1.94%	2.49%	0.59%	2.09%
poly3	Fit	2.01%	$\boxed{3.08\%}$	0.66%	2.21%
poly3	Avg	2.01%	$\boxed{3.08\%}$	0.66%	2.22%
poly4	Fit	2.60%	2.33%	0.90%	2.52%
poly4	Avg	2.56%	2.23%	0.89%	2.49%
trapezoid	Fit	$\boxed{\boxed{2.91\%}}$	$\boxed{\boxed{3.92\%}}$	$\boxed{1.01\%}$	$\boxed{\boxed{3.70\%}}$
trapezoid	Avg	2.56%	3.01%	0.97%	$\boxed{2.83\%}$
triangle	Fit	2.66%	2.41%	0.93%	1.88%
triangle	Avg	$\boxed{2.88\%}$	2.68%	$\boxed{\boxed{1.24\%}}$	2.54%
BestForEach	Avg	2.68%	1.80%	0.76%	2.04%

Table 7 Change in MASE and SAE as a result of using our forecast versus Dell's for different forecasting approaches during the initial portion of the life cycle. For MASE, the values represent the mean of the individual percentage changes in MASE across SKUs. SAE is measured across the entire portfolio of SKUs. Positive values imply that our method decreased forecast errors. Each row represents a choice that a demand planner can make in generating forecasts, while the four columns represent different ways to evaluate forecast accuracy improvement. The double box denotes the best value in each column, while the single box denotes the second best value.

requires an adjustment to account for the updating. For $ScaleForecast$ we take Dell's updated forecasts as is, and scale our forecast so that the lifetime forecast volumes are equal to each other. For $ScaleActual$, we still scale our normalized PLC curve by the actual lifetime volume. However, we scale Dell's forecast one of two ways: scaling the entire forecast or scaling segment-by-segment. In the first way, we 'stitch' together Dell's forecast at the points where two forecasts meet. We then scale this 'stitched' forecast so that its total adjusted lifetime volume equals the the actual lifetime demand. We call this *ScaleActualStitch*. It is possible scaling Dell's stitched forecast to the actual volume may significantly disadvantage Dell; on many SKUs we observed that for the first quarter of a product's lifetime, Dell might have severely biased forecasts, but that subsequent quarters would be much less biased or even overcorrected. Scaling the entire stitched curve up or down might result in a curve that does not represent a useful (or accurate) PLC shape, due to the discontinuity at the point where Dell started making better forecasts and reduced its bias.

Figure 13 Illustration of stitched forecasts for two SKUs. The top figure shows $ScaleActualStitch$**, the middle shows** $ScaleActualSegment$**, while the bottom shows** $ScaleForecast$**. The vertical lines show the stitch points (the epochs at which one forecast horizon ends and we adopt the next quarter's forecasted values).**

The second way—$ScaleActualSegment$—is to adjust Dell's forecasts to the actual total in each updated segment. This gives Dell's updating forecasts the biggest advantage. For Dell's forecast, between the stitch points, we scale each segment of Dell's forecast so that, in each segment, the sum of Dell's forecasts equals the sum of the actual demand. Figure 13 shows examples of SKUs whose forecasts are stitched together, and to which both $ScaleActualStitch$ and $ScaleActualSegment$ are applied. If anything, this approach should favor Dell because every three months planners are able to update the volume and shape.

In Table 8 we report on the improvement of forecast accuracy. Piecewise linear curves (namely trapezoids and triangles) perform very well among the other curves we tried. Because Dell was able to update their forecast throughout the life cycle (unlike us) and because we tried to account for this in a few different ways, our forecast performance varies significantly: it ranges between an almost 15% improvement over Dell's forecast to a 2% decline. Given that $ScaleActualStitch$

Base curve family	Cluster curve generation (*GenerateFit* or *GenerateAvg*)	Relative reduction in our forecast errors over Dell's *updated* forecasts					
		MASE			SAE		
		Scale−Forecast	*ScaleActual* *Stitch*	*ScaleActual* *Segment*	*Scale−Forecast*	*ScaleActual* *Stitch*	*ScaleActual* *Segment*
Bass	Fit	9.70%	12.30%	-0.10%	10.90%	12.90%	-4.70%
Bass	Avg	9.50%	11.90%	-0.70%	10.60%	12.40%	-5.30%
poly2	Fit	10.10%	12.80%	0.20%	11.30%	13.10%	-4.40%
poly2	Avg	10.10%	12.80%	0.30%	11.30%	13.10%	-4.40%
poly3	Fit	10.30%	13.20%	0.70%	11.50%	13.60%	-3.80%
poly3	Avg	10.30%	13.20%	0.70%	11.50%	13.60%	-3.80%
poly4	Fit	10.80%	12.80%	-0.20%	12.00%	13.30%	-4.10%
poly4	Avg	10.80%	12.70%	-0.30%	12.00%	13.30%	-4.20%
trapezoid	Fit	11.20%	13.90%	1.40%	12.70%	14.70%	-2.50%
trapezoid	Avg	11.30%	13.50%	0.90%	12.60%	14.10%	-3.10%
triangle	Fit	11.00%	13.60%	0.40%	11.50%	13.60%	-3.80%
triangle	Avg	11.40%	13.60%	0.60%	12.30%	14.10%	-3.20%
BestForeach	Avg	11.00%	12.80%	0.10%	12.20%	13.70%	-3.60%

Table 8 Change in forecast accuracy as a result of using our forecast versus Dell's for different forecasting approaches. The interpretation is the same as in Table 7, except this is for the updated forecasts.

perhaps favors our forecast too much while the *ScaleActualSegment* favors Dell's too much, we posit that the truth is likely in the middle and that if Dell provided us with full life cycle forecasts pre-launch, our approach would have similar or better performance as reported in Table 7: 3-4% improvement.

In Figure 14 we show the distribution of MASE for our and Dell's forecasts, for *ScaleActual* for the initial forecasts and for *ScaleActualStitch* and *ScaleActualSegment* for the updated full life cycle forecasts. This figure highlights that the overall improvements in MASE reported in Tables 7 and 8 are not due to a few outliers. Rather, our approach stochastically dominates Dell's approach with respect to the empirical cumulative distribution function (ECDF) of the values of MASE observed across all of the 45 products for which we have Dell forecasts. That is, for most (but not all) values of MASE: $F_{OurMethod}(MASE) \leq F_{DellsMethod}(MASE)$, where $F(\cdot)$ is the ECDF.

6.3. Comparing clustering approaches

For most of the SKUs in our data set (125 out of 170), we do not have Dell's forecasts. However, we can compare different ways of clustering products together and the resulting forecast accuracy of each clustering approach. We compare the following clustering approaches: features, product category, overall, and data-driven-time-series (each of these is defined in Section 5.1). Because we

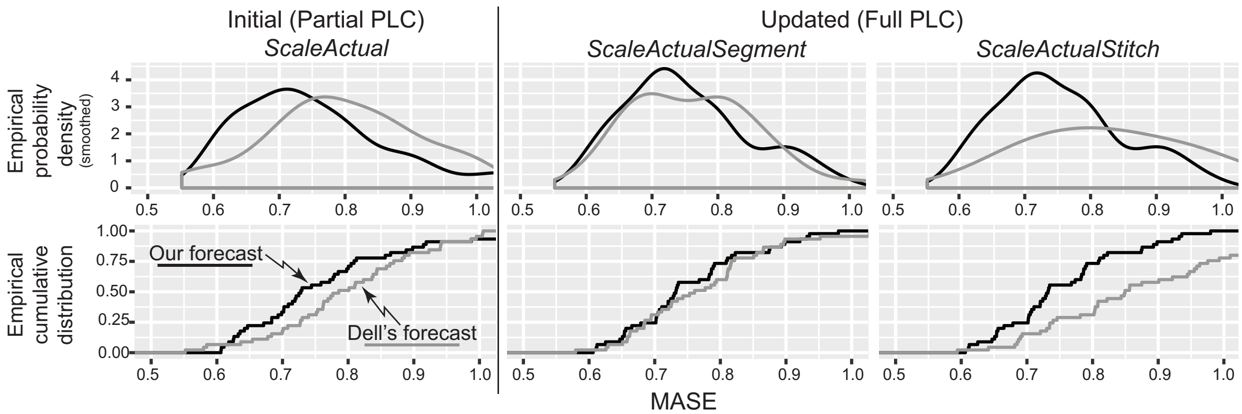

Figure 14 **Distribution of MASE across 45 SKUs for three different forecast evaluations. The empirical probability density function is above while the empirical cumulative distribution function is below. The horizontal axis is truncated at 1 to better show details within the relevant range. Our forecasts are based on the trapezoid curve using** $GenerateFit$**.**

do not have forecasts, we will scale all normalized PLC curves using the actual volume (defined as $ScaleActual$ earlier). While it is not realistic to expect to know the true lifetime volume prior to launch, all clustering approaches are similarly advantaged. So, we believe the *relative* improvement of one approach over another will transfer to practice.

Recall that we do not have all data on all products (see Figure 4). Thus, we make as many comparisons as we can, with each set of comparisons on a different subset of the data. Tables 9 and 10 show the relative change in forecast errors for Dell's forecasts (using $ScaleActual$) and our forecasts using various clustering approaches when compared to no clustering. For the time-series clustering approach, a trapezoid shape is the base curve, and we impose this shape using $GenerateFit$. The results in the tables show that, of the different clustering approaches available to a planner, the time-series clustering algorithm leads to the best forecasts.

6.4. Robustness summary

Some of the parameters used in our method are user-chosen. We have already reported on the forecast accuracy of different versions of some of these parameters: choice of family of curves, $GenerateFit$ versus $GenerateAvg$, imposing a single shape for all SKUs' curves or using $BestForEach$. To test how robust our method is (or is not) to other user-selected parameters, we vary the following inputs: whether or not to adjust for seasonality, values of end-of-life truncation parameters θ^v and θ^t, and the number of clusters for the time-series algorithm (see the scree plot in Figure 10). Table 11 shows the summary of the robustness results. Our method is relatively robust to user-defined parameters; under some conditions the MASE improvement of 3.92% can

Limit to SKUs for which we know:			**Count of SKUs**	**Reduction in SAE over no clustering**			
Features	Dell forecasts	Product category		Dell's own forecasts	Our forecasts' clustering methods *Features*	*Product*	*Time series*
✓		✓	86		1.84%	2.50%	4.66%
✓	✓	✓	34	-0.35%	0.29%	0.63%	1.97%
	✓	✓	45	-2.30%		0.55%	1.50%
		✓	161			3.42%	4.80%

Table 9 Summary of improvement of Dell's forecast and various clustering approaches over no clustering when measured using $ScaleActual$**. Each row's forecast improvements are evaluated only on the subset of SKUs for which we have all the data for which there are check marks in the leftmost three columns. Our time-series clustering approach (using trapezoid and** $GenerateFit$**) results in the best improvement relative to using no clustering at all.**

Limit to SKUs for which we know:			**Count of SKUs**	**Reduction in MASE over no clustering**			
Features	Dell forecasts	Product category		Dell's own forecasts	Our forecasts' clustering methods *Features*	*Product*	*Time series*
✓		✓	86		1.71%	2.15%	4.83%
✓	✓	✓	34	-0.13%	0.02%	0.54%	2.38%
	✓	✓	45	-1.93%		0.45%	1.97%
		✓	161			3.44%	5.00%

Table 10 This table has the same structure as Table 9 except it is measuring improvement in MASE.

decrease to a 1.15% decline, while the SAE improvement of 3.70% can decrease to 0.5%. However, our forecasting approach improves upon Dell's internal forecasts in all cases except one, although the amount by which it improves varies with a median improvement in MASE of 2.29% and in SAE also of 2.29%.

7. Additional case study of PLC fitting and forecasting

In addition to the Dell data we analyze above, we apply our PLC fitting and forecasting approach to data from ACME, an anonymized name for a computer accessories manufacturer. ACME provided us with over three years of monthly sales data (2010 to 2013), which included complete life cycle sales for three SKUs as well as partial sales for direct successors of each of those SKUs. We will adopt the notation of SKU0X-p to denote a predecessor, SKU0X-s to denote its successor, and SKU0X to denote the product line as a whole. Summary statistics on the predecessor SKUs are in the left hand side of Table 12.

Parameter	Value(s)		Relative reduction in forecast errors MASE	SAE
End-of-life truncate	θ^t	θ^v		
	0.90	0.90	3.92%*	3.70%*
	0.90	1.00	1.51%	1.56%
	1.00	0.90	2.47%	2.19%
	0.95	0.95	1.38%	2.93%
	0.95	1.00	-1.12%	0.50%
	1.00	0.95	2.47%	2.19%
	1.00	1.00	1.25%	2.41%
Number of time-series clusters	3		3.27%	3.47%
	4		3.92%*	3.70%*
	5		2.66%	2.03%
Seasonality	In		2.11%	2.38%
	Out		3.92%*	3.70%*
Median reduction			**2.29%**	**2.29%**

Table 11 **Robustness summary of forecast accuracy change of our forecast versus Dell's when different user parameters are adjusted. Our forecast utilizes time-series clustering, trapezoid is the base curve, curves are generated using** $GenerateFit$**, our and Dell's forecasts are scaled by** $ScaleActual$**, and forecasts are compared for the initial portion of the product life cycle and do not involve stitching any forecasts. Positive values imply that our forecast has better accuracy than Dell's. (* indicates base case reported earlier in the paper. These values were counted only once in the median calculations reported.)**

SKU	**Predecessor information** PLC volume (units)	PLC length (months)	**Forecast performance** ACME MAE	Our MAE	Improvement
SKU01	1,337,602	17	33,542	14,611	56%
SKU02	77,630	16	2,108	1,764	16%
SKU03	18,284	18	498	973	-96%
TOTAL (SAE)			612,906	294,125	52%

Table 12 **ACME SKU summary statistics and comparison of forecasting performance. Our PLC fitting and forecasting approach results in a substantial reduction in SAE overall.**

In addition to historical sales, ACME provided us with multiplicative seasonal factors for each of the product lines. Using deseasonalized historical sales, we apply our data preparation and fitting procedure. We truncate end-of-life with $\theta^t = \theta^v = 0.95$, and there were no negative orders or outliers. Results are reported in Table 13. Consistent with Dell PLC fitting results, trapezoid and poly4 curves provide the best PLC fits.

Config	Bass	poly2	poly3	poly4	triangle	trapezoid
RMSE						
- SKU01-p	0.01877	0.01699	0.01471	$\boxed{\boxed{0.00960}}$	0.01233	$\boxed{0.01095}$
- SKU02-p	0.04083	0.04189	0.04080	0.03855	$\boxed{0.03369}$	$\boxed{\boxed{0.03369}}$
- SKU03-p	0.03374	0.03498	0.03495	$\boxed{0.03305}$	0.03330	$\boxed{\boxed{0.03298}}$
Log likelihood (sum)	103.01	103.57	106.28	$\boxed{\boxed{114.98}}$	112.85	$\boxed{114.93}$
AIC	-188.02	-189.14	-188.56	$\boxed{\boxed{-199.96}}$	-221.7	$\boxed{-199.86}$

Table 13 Summary statistics of PLC fits to the deseasonalized ACME data. Boxes with a double (single) outline denote the best (second best) value in each row. Consistent with Dell PLC fitting results (see Table 3), trapezoid and poly4 curves most effectively fit the data.

We generate forecasts for the successor SKUs by using the PLC curve, length and volume of the predecessors. Figure 15 shows this fitting and forecasting process start-to-finish for SKU01: (a) predecessor sales (after EOL truncation) are deseasonalized, (b) a trapezoid curve is fit to the deseasonalized data, (c) the trapezoid curve is scaled by the PLC length and volume of it's predecessor, and (d) seasonal factors are applied to create the forecasts. (Analogous figures for SKU02 and SKU03 are in Appendix C)

Rather than forecast the entire life cycle at once, ACME historically made three-month-ahead forecasts (since their products have a three month lead time.) We have ACME forecasts for a six month window, and we compare the accuracy of our forecasts to theirs. Results are summarized in Table 12. Our approach results in substantially better accuracy for the two higher volume items (SKU01, SKU02) and much lower accuracy for the lowest volume item. While it is encouraging that our approach results in a 52% reduction in sum of absolute errors, one should be cautious in drawing conclusions based on this sample of only three pairs of items.

8. Conclusion

In this paper, we address the problem of generating forecasts for new products that are similar to past products. To accomplish this, we fit curves of several functional forms (Bass, piecewise linear and polynomial) to normalized product life cycle historical data. Using complete product life cycle order history for 170 products from Dell, we evaluate our data preparation and PLC fitting approach.

We find that simple, piecewise-linear curves are effective in fitting historical PLC curves. In particular, simple triangles and trapezoids performed very well on our data. Both have the advantage that they are easy to explain and therefore easy to implement. In addition, we found that about 20% of the product life cycles in our data set feature no maturity stage, and 75% have maturity

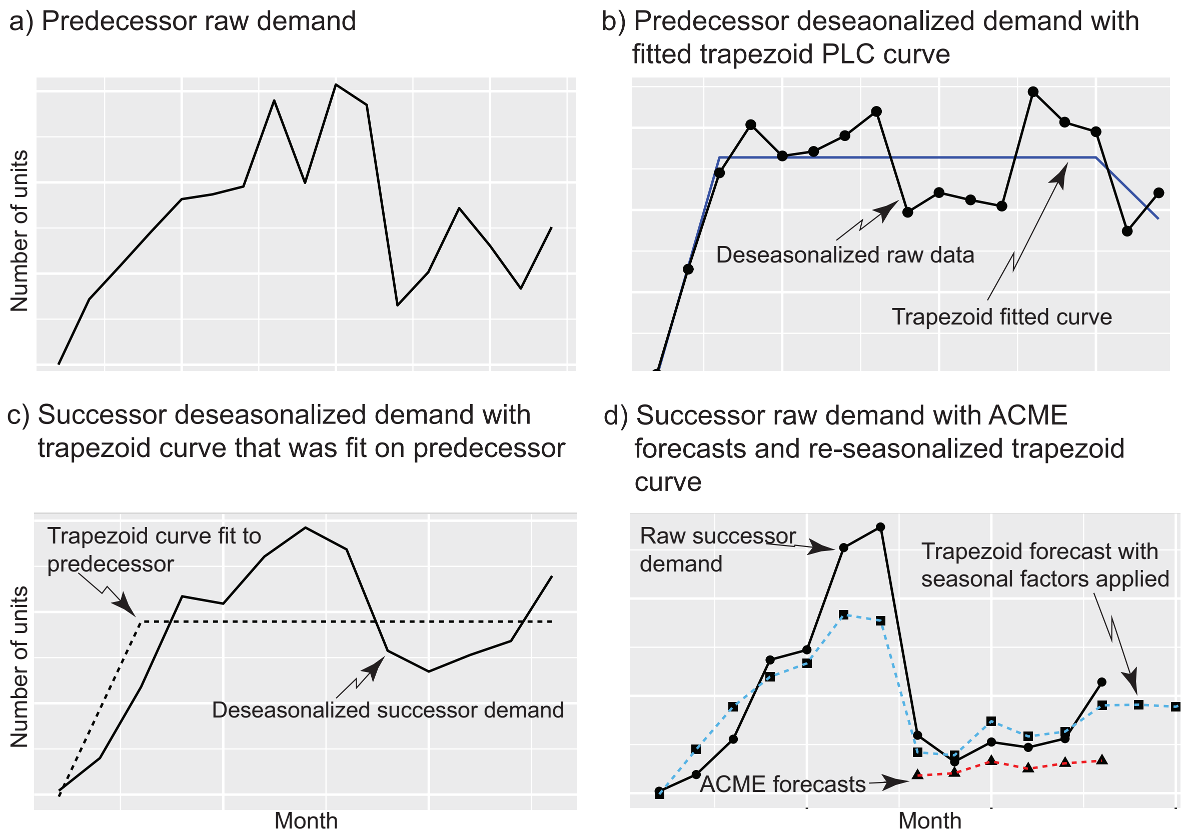

Figure 15 **PLC forecasts and actual data for ACME**

stages of less than 1/3 of their PLC. While our approach is general and could be applied to other industries, our empirical finding of a very short maturity stage is of course specific to the industrial setting studied here. Indeed, an opportunity for future research is to apply a PLC fitting approach such as ours to data sets from other industries.

We use the normalized PLC curves (which were fit to historical data) for forecasting by first clustering similar PLC curves and finding a representative curve for each cluster. We clustered using information provided by the company as well as by using a data-driven time-series clustering technique. When we know which cluster a new product belongs to (e.g., the cluster of its most similar predecessor), we simply use that cluster's representative curve as the new product's forecast. Since our approach uses normalized curves, we must scale the appropriate normalized curve for a new product by a lifetime quantity forecast. Given that there are many ways to evaluate forecast accuracy improvements, we used several benchmarks to provide a fair and conservative evaluation. Our out-of-sample forecast evaluation indicates our approach results in a reduction in absolute errors of approximately 3-4% (see Table 7). Our fitting and forecasting method can be applied in different industries but may be particularly useful for the electronics sector we study here as well as the fashion industry given the long production lead times and short product life cycles.

Effective new product forecasting is critical for many companies, and many new products fall into the category of "similar" to past products; thus, our approach would be applicable. Several possibilities for further study are desirable, especially incorporating forecast updating and the firm's possibility of quick response to supply-demand mismatches (given that the PLC length typically exceeds the lead time). We hope our work, and the normalized data set that we make available, stimulates new research in this area.

References

Ban, Gah-Yi, Jérémie Gallien, Adam J. Mersereau. 2017. Dynamic procurement of new products with covariate information: The residual tree method. SSRN Scholarly Paper ID 2926028, Social Science Research Network, Rochester, NY. URL `https://papers.ssrn.com/abstract=2926028`.

Bass, Frank M. 1969. A new product growth for model consumer durables. *Management Science* **15**(5) 215–227.

Boute, Robert N, Jan A Van Mieghem. 2014. Global dual sourcing and order smoothing: The impact of capacity and lead times. *Management Science* **61**(9) 2080–2099.

Buzzell, Robert Dow, Robert E Nourse. 1967. *Product innovation in food processing, 1954-1964*. Harvard University Press, New York.

Chen, Chung, Lon-Mu Liu. 1993. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association* **88**(421) 284–297.

Chen, George H., Stanislav Nikolov, Devavrat Shah. 2013. A latent source model for nonparametric time series classification. *Working paper* URL `http://arxiv.org/abs/1302.3639`. ArXiv: 1302.3639.

Chouakria, Ahlame Douzal, Panduranga Naidu Nagabhushan. 2007. Adaptive dissimilarity index for measuring time series proximity. *Advances in Data Analysis and Classification* **1**(1) 5–21.

Cooper, Robert G, Scott J Edgett. 2012. Best practices in the idea-to-launch process and its governance. *Research-Technology Management* **55**(2) 43–54.

Cox, William E. 1967. Product life cycles as marketing models. *The Journal of Business* **40**(4) 375–384.

Crompton, John L. 1979. Recreation programs have life cycles, too. *Parks and Recreation* **Oct.** 52–57.

Crompton, John L, Sharon Bonk. 1978. An empirical investigation of the appropriateness of the product life cycle to municipal library services. *Journal of the Academy of Marketing Science* **6**(1-2) 77–90.

Diermann, Christoph, Arnd Huchzermeier. 2016. Judgmental demand forecasting for online sales of product collections: Segmentation by type or age? SSRN Scholarly Paper ID 2782002, Social Science Research Network, Rochester, NY. URL `https://papers.ssrn.com/abstract=2782002`.

Fildes, Robert, Paul Goodwin. 2007. Against your better judgment? how organizations can improve their use of management judgment in forecasting. *Interfaces* **37**(6) 570–576.

Fisher, Marshall, Ananth Raman. 1996. Reducing the cost of demand uncertainty through accurate response to early sales. *Operations Research* **44**(1) 87–99.

Frederixon, Martin Shelton. 1969. An investigation of the product life cycle concept and its application to new product proposal evaluation within the chemical industry. Ph.D. thesis, Michigan State University.

Gallien, Jérémie, Adam J Mersereau, Andres Garro, Alberte Dapena Mora, Martín Nóvoa Vidal. 2015. Initial shipment decisions for new products at Zara. *Operations Research* **63**(2) 269–286.

Golder, Peter N, Gerard J Tellis. 2004. Growing, growing, gone: Cascades, diffusion, and turning points in the product life cycle. *Marketing Science* **23**(2) 207–218.

Goldman, Arieh. 1982. Short product life cycles: implications for the marketing activities of small high-technology companies. *R&D Management* **12**(2) 81–90.

Graves, Stephen C. 1999. A single-item inventory model for a nonstationary demand process. *Manufacturing & Service Operations Management* **1**(1) 50–61.

Hayes, Robert H, Steven C Wheelwright. 1979. Link manufacturing process and product life cycles. *Harvard Business Review* **57**(1) 133–140.

Headen, Robert Speir. 1966. *The Introductory Phases of the Life Cycle for New Grocery Products: Consumer Acceptance and Competitive Behavior*. Graduate School of Business Administration, George F. Baker Foundation, Harvard University.

Ho, Teck-Hua, Sergei Savin, Christian Terwiesch. 2002. Managing demand and sales dynamics in new product diffusion under supply constraint. *Management Science* **48**(2) 187–206.

Hyndman, Rob J, Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* **22**(4) 679–688.

Kahn, Kenneth B. 2002. An exploratory investigation of new product forecasting practices. *Journal of Product Innovation Management* **19**(2) 133–143.

Kapuscinski, Roman, Rachel Q Zhang, Paul Carbonneau, Robert Moore, Bill Reeves. 2004. Inventory decisions in dell's supply chain. *Interfaces* **34**(3) 191–205.

KPMG. 2016. U.s. CEO outlook. *https://assets.kpmg.com/content/dam/kpmg/pdf/2016/07/2016-ceo-survey.pdf* .

Kumar, Sunil, Jayashankar M Swaminathan. 2003. Diffusion of innovations under supply constraints. *Operations Research* **51**(6) 866–879.

Kurawarwala, Abbas A, Hirofumi Matsuo. 1996. Forecasting and inventory management of short life-cycle products. *Operations Research* **44**(1) 131–150.

Levitt, Theodore. 1965. Exploit the product life cycle. *Harvard Business Review* **18** 81–94.

Niu, Shun-Chen. 2006. A piecewise-diffusion model of new-product demands. *Operations Research* **54**(4) 678–695.

PWC. 2017. 20^{th} CEO survey. *http://www.pwc.com/gx/en/ceo-agenda/ceosurvey/2017/us* .

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rink, David R, John E Swan. 1979. Product life cycle research: A literature review. *Journal of Business Research* **7**(3) 219–242.

Roberts, SW. 2000. Control chart tests based on geometric moving averages. *Technometrics* **42**(1) 97–101.

Stark, John. 2015. Product lifecycle management. *Product Lifecycle Management*. Springer, 1–29.

Tigert, Douglas, Behrooz Farivar. 1981. The Bass new product growth model: a sensitivity analysis for a high technology product. *The Journal of Marketing* **45** 81–90.

Wu, S David, Berrin Aytac, Rosemary T Berger, Chris A Armbruster. 2006. Managing short life-cycle technology products for agere systems. *Interfaces* **36**(3) 234–247.

Zhu, Kaijie, Ulrich W Thonemann. 2004. An adaptive forecasting algorithm and inventory policy for products with short life cycles. *Naval Research Logistics* **51**(5) 633–653.

Appendix

A. Adjusting for seasonality and normalization

This seasonality step would be after the 'Detecting cancellations' step and before the 'Excluding build-to-order customers' step outlined in Section 3.3. If one had enough data to group products together by market (namely, by seasonal buying patterns of primary customers of each product), or if seasonality factors were given by the firm, we suggest one approach here to adjust for seasonality:

1. Normalize data so that cumulative volume of each product equals 1. In this way, the seasonal effect will not be disproportionally affected by high volume products.

2. For each group, apply an additive or multiplicative seasonal effect model to the normalized data. This model would estimate customer orders based on the following independent variables:

   (a) Month effect (the seasonality to be estimated)

   (b) A generalized group-wide PLC that the model would estimate

   (c) Fixed effects for year and other group attributes

Hence, the seasonality-adjusted data is then $D_t^{C[n+1],i} = f_{Season}^{-1}\left(D_t^{C[n],i}\right)$, where $f_{Season}^{-1}(\cdot)$ denotes the deseasonalization function derived from the seasonal effects model. For the reasons mentioned above, we set $f_{Season}^{-1}(\cdot)$ to be the identity function; that is, $f_{Season}^{-1}(x) = x$ and $f_{Season}(x) = x$.

At the end of the forecasting process, in order to turn a PLC curve into a forecast, seasonality must also be considered. This corresponds to step 3 in Section 5.2. In this step, the fitted PLC curves are updated as:

$$\breve{D}_t^{i,PLC,\eta,Season} = f_{Season}\left(\breve{D}_t^{i,PLC,\eta}\right), \text{ for } t = 1, \cdots, \hat{T}. \tag{9}$$

For Dell, we do not adjust for seasonality. Thus, we define $f_{Season}(\cdot)$ as the identity function (that is, $f_{Season}(x) = x$). For our analysis of data at ACME, we do apply the firm's multiplicative seasonality factors.

B. Time series clustering

The time series distance metric proposed by Chouakria and Nagabhushan (2007) measures proximity by two dimensions: behavior and value. The proximity of *behaviors* between two series X and Y is evaluated by means of the first order temporal correlation coefficient, which is defined by

$$CORT(X_t, Y_t) = \frac{\sum_{t=1}^{T-1}(X_{t+1} - X_t)(Y_{t+1} - Y_t)}{\sqrt{\sum_{t=1}^{T}(X_{t+1} - X_t)^2}\sqrt{\sum_{t=1}^{T-1}(Y_{t+1} - Y_t)^2}} \tag{10}$$

$CORT(X_t, Y_t)$ falls into $[-1, 1]$ with 1 meaning that two series behave similarly, i.e. their increase or decrease at any instant of time are similar in direction and rate, -1 meaning that the two series have similar rate of change but opposite in direction, and 0 meaning that the two series are stochastically linearly independent.

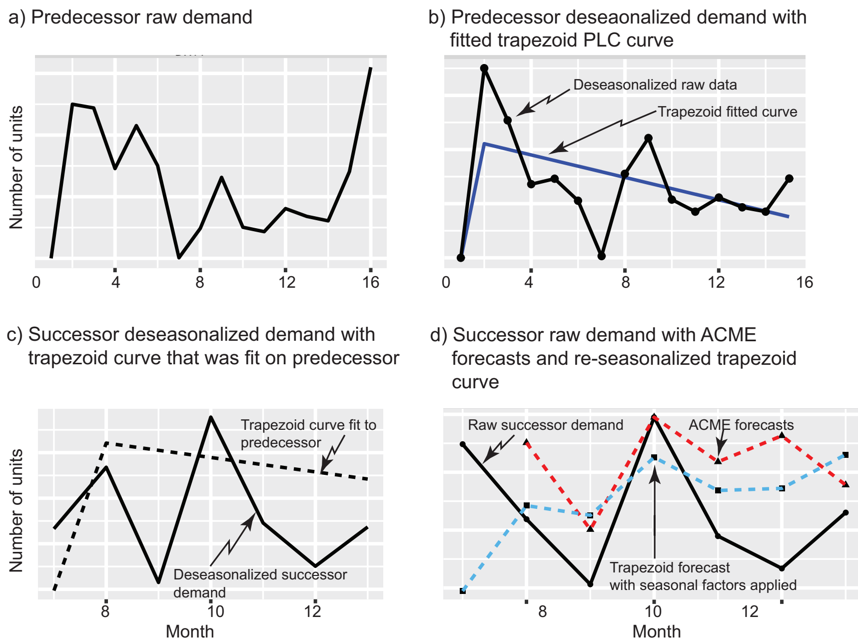

Figure 16 PLC fitting and forecasting for ACME SKU02

The proximity on *values* are measured as the conventional Euclidean distance with $d(X_t, Y_t) = \sqrt{\sum_{t=1}^{T}(X_t - Y_t)^2}$.

The dissimilarity index to measure the proximity between series X_t and Y_t is proposed as

$$d_{CORT}(X_t, Y_t) = \phi_m[CORT(X_t, Y_t)]d(X_t, Y_t) \tag{11}$$

where $\phi_m(\cdot)$ is an adaptive tuning function to adapt the distance metrics $d(X_t, Y_t)$ to the temporal correlation $CORT(X_t, Y_t)$. With m to be the tuning parameter, the function $\phi_m(u)$ is written as

$$\phi_m(u) = \frac{2}{1 + e^{m\mu}}, m \geq 0.$$

In our case, we use m equal to 2, which is the default choice recommended by Chouakria and Nagabhushan (2007). When using other values for m, the results change very little. Note the dissimilarity measure $d_{CORT}(X_t, Y_t)$ is model-free, it allows us to cluster the fitted PLC curves based on their features in terms of temporal structure and scales.

C. Additional figures for ACME forecasting

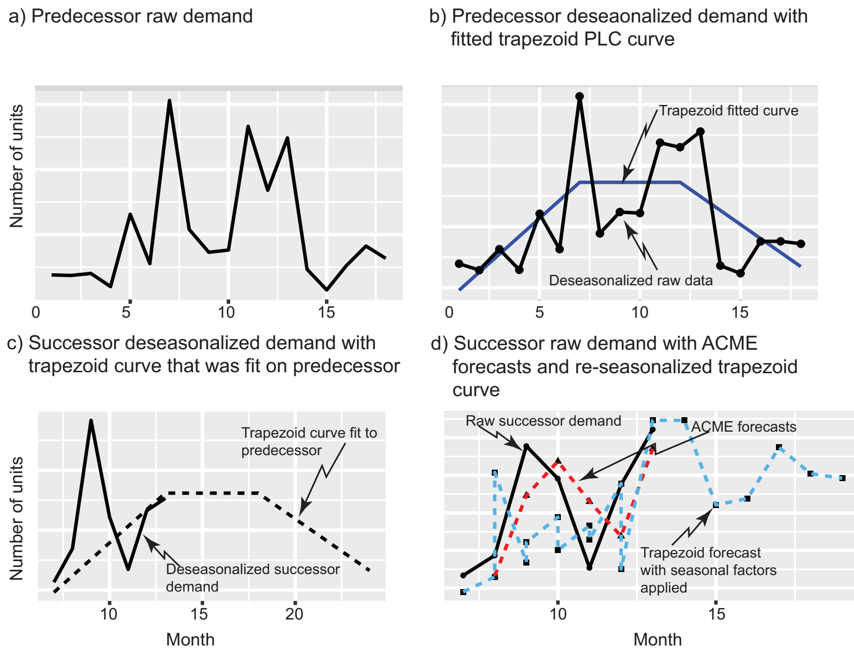

Figure 17 **PLC fitting and forecasting for ACME SKU03**