Leveraging Comparables for New Product Sales Forecasting

**Problem definition:** Many firms regularly introduce new products. Before the launch of any new product, firms need to make various operational decisions, which are guided by the sales forecast. The new product sales forecasting problem is challenging when compared to forecasting sales of existing products. For existing products, historical sales data gives an indicator of future sales, but this data is not available for a new product. In this paper, we confront the problem of forecasting new product sales by leveraging sales data available from comparable products.

**Academic / Practical Relevance:** Literature on forecasting sales of new products is limited, yet it is a fundamental problem for operation managers. We propose a novel sales forecasting approach that uses the data available from products that were introduced in the past. For this forecasting method, we prove a finite sample prediction error guarantee. Also, we show that the approach improves out-of-sample forecasting metrics, specifically an average reduction of 40% in MAPE and WMAPE.

**Methodology:** We formulate the problem of clustering products and fitting forecasting models to these clusters simultaneously as a quadratic mixed integer optimization problem. Inherently, the model has a large number of parameters, which can lead to an overly complex model. Hence, we add regularization to the model so that it can estimate sparse models. This problem is computationally hard, and as a result, we develop a scalable algorithm that produces a forecasting model with good analytical guarantees on the prediction error.

**Results:** In close collaboration with our industry partner Johnson & Johnson Consumer Companies Inc., a major fast moving consumer goods manufacturer, we test our approach on real datasets, after which we check the robustness of our results with data from a large fast fashion retailer. We show that, compared to several widely used forecasting methods, our approach improves MAPE and WMAPE by 20-60% across various product segments.

**Managerial Implications:** Operations managers in any firm deal with the introduction of new products and their sales forecasts. We show that the current practice of forecasting can be highly suboptimal, and develop a method that is simple to implement, and has both computational and analytical guarantees. Additionally, for the consumer goods manufacturer, we develop a fast and easy-to-use Excel tool that aids managers with forecasting and making decisions before a new product launch.

**Key words:** Data-driven Operations, New Products, Forecasting, Clustering, Regularization, LASSO
1. Introduction

Sales forecasting is a central activity in a firm’s operations. Most operational decision making tools incorporate models describing product sales. Particularly, the sales forecasts of new products guide many of the operational decisions made during product development (e.g., production, inventory, and pricing). Making the right decisions is key to the success of a product launch, and therefore, it is important to forecast the sales of a product accurately. The difficulty in doing this varies considerably between industries, even for existing products. When historical sales data is available, regularly purchased products (e.g., fast moving consumer goods) are easier to predict than temporary products (e.g., fashion clothing). Several studies illustrate this with a mean absolute error (MAE) of roughly 5 units and a mean absolute percentage error (MAPE) of 11% to 19% for grocery brands (Ali et al. 2009, Cohen et al. 2017), which worsens for fashion retailers where it is around 13 units and 68% to 93% (Ferreira et al. 2016). Regular consumption of a product reduces the variability of its sales over time, which means that its historical sales will be a good indicator of future sales.

The main focus of this paper is new product sales forecasting, where typically no historical sales data is available. The interest in this particular problem originates from our collaboration with Johnson & Johnson Consumer Companies Inc., a major fast moving consumer goods manufacturer, and was later verified by the interest of a large fast fashion retailer in our approach. Both industry partners introduce new products to update their assortment frequently: monthly in the fast moving consumer goods industry, and weekly or sometimes daily in the fast fashion industry. Before and during a new product launch, each firm needs to make many decisions that affect the success of the product. These decisions span the entire range of operations: capacity planning, procurement, production scheduling, inventory control, distribution planning, marketing promotions and pricing. As each of these operational decisions are guided by sales forecasting, an accurate sales forecasting model is key to a successful product launch for both industry partners, but also more generally. The importance of this success has grown tremendously over the past decade, as Cecere (2013) estimates that on average new product costs have increased four times over a period of five years.

Current practice has considerable difficulties with predicting new product success. Standard forecasting models use past sales data to predict on the short term. However, predictions for new products need to be made far in advance without any sales history. As a result, many firms, our industry partners included, resort to costly and time-consuming qualitative methods. Kahn (2002) suggests that surveys, expert opinions, and average sales of comparable products are the most widespread techniques for predicting demand of new products. These methods are popular due to their interpretability. This is an essential characteristic, as Armstrong et al. (2015) argue that practitioners should be overly conservative when they do not understand the forecasting procedures. Possibly for this reason, Kahn (2002) observes that at the time only 10% of companies make use of some form of analytics. More recently, Cecere (2013) and Kahn (2014) argue that the usage of analytics is still limited. At the same time only 20% of companies are satisfied with their
approach. Altogether, this literature shows that there are opportunities to improve the new product sales forecasting process significantly, particularly using analytics.

In this paper, we develop an accurate, scalable and interpretable forecasting tool calibrated with our industry partners’ data. These characteristics of the tool are important to both industry partners. For interpretability, we draw inspiration from the practice of our industry collaborators who use comparable products to predict sales of new products. Currently, many practitioners manually determine which products are similar to the new product and use their average sales as a forecast. We propose a cluster-while-estimate model that mimics this, but simultaneously creates clusters of comparable past products and creates forecasting models for each cluster. Adding to the interpretability, the model also includes feature selection methods. Incorporating regularization allows each cluster to prioritize different variables as the most important predictors of sales. As an example, customers might be more price sensitive for generic brands than for premium brands. To address scalability, we devise a fast algorithm whose steps mimic industry practice. The accuracy of the estimated model is established both in theory and in practice. Theoretically, we prove a bound on the prediction error of the estimated model that holds with high probability and scales slowly with the size of the dataset. From a practical standpoint, we estimate our model on data from our two industry partners, and observe significant improvements in out-of-sample forecasting metrics.

1.1. Contributions

Our main contribution is the development of a new product sales forecasting approach based on analytics. We estimate a clustered forecasting model using data of comparable products, and show strong results on the forecasting accuracy for new products introduced by our industry partners. To summarize our contributions:

- **Interpretable clustered forecasting model**: We develop a novel approach to new product sales forecasting. Motivated by practice, we presume that sales of different product clusters are generated by different sales models. In our model formulation, comparable products are clustered together and share a forecasting model that can be any regression model (e.g., linear regression, generalized linear models, regression trees, etc.). As the model is grounded in current practice, it is easy to use for managers who need to understand the model in order to trust it and use it. Furthermore, the approach is general and can account for changes in drivers of product sales such as marketing budget or distribution decisions. Section 2 uses the data of our industry partners to motivate our general model, which is formulated in Section 3.1.

- **Tractable cluster-while-estimate algorithm**: Estimating the cluster forecasting model is not an easy problem. We need an estimation procedure for the model as well as a procedure that assigns new products to clusters. We show that the cluster-while-estimate problem is both NP-hard and practically intractable, as the running times increase rapidly with the problem size. As a result, we develop a fast algorithm that solves the problem. Then, we propose a multiclass classification method that simultaneously assigns the probability with which the new product belongs to a cluster. By weighting each cluster’s forecasting model
with the cluster assignment probability we predict new product sales. Our algorithm is fast, scalable and cheap as it only uses available data to make sales predictions. Section 3.2 discusses the estimation problem, and Section 3.3 describes the cluster assignment problem.

- **Analytical bounds for regularized linear regression:** We focus on the case of regularized linear regression in our applications, due to the interpretability of linear regression coefficients and the additional meaning to the clusters. We show the need for the linear cluster-while-estimate algorithm as the exact estimation problem is NP-hard. Additionally, computational experiments show that it is intractable for small scale datasets with only 100 products and 5 product features. Finally, we prove that the prediction error of the algorithm’s solution is bounded with high probability. This result retrieves the prediction error bound for regular LASSO regression when our model involves a single cluster. Section 4 formulates the linear clustered forecasting model, discusses its estimation, and presents the results.

- **Strong performance on real world data:** Working in collaboration with two large industry partners, we show that our algorithm results in a 20%-70% MAPE improvement and 10%-60% WMAPE improvement over several benchmarks used in practice. These results are particularly robust to external changes in the market as we also incorporate competitor’s data in the estimation of our prediction model, which results in a better understanding of how the market responds to new product releases. In order to further check the robustness of our results, we test our algorithm on various product categories and observe similar improvements. In Section 5, we describe the results on fast moving consumer goods data. Furthermore, Section 6 tests the robustness of the model with fast fashion retail data.

- **Accessible forecasting tool for practitioners:** Out-of-sample results encouraged our fast moving consumer goods manufacturing partner, Johnson & Johnson, to employ the forecasting approach. Our model can be estimated offline, which allows us to code the estimated model into Excel. In this tool, our partners can experiment with a product by changing product features, which immediately gives a report on expected sales, trends in sales over time, and the most important constituents of predicted sales. This allows the managers to identify the key drivers of demand for own and competitive products, in turn this allows them to optimize their new product launch and outperform competitive launches through scenario planning. Positive feedback has encouraged further development of the tool. Section 5 discusses the forecasting tool.

1.2. **Literature Review**

Our work relates to the literature on both sales forecasting and product innovation which have been studied extensively in the operations management and marketing literature. More specifically, our work lies in the intersection of three different streams of literature: product diffusion and innovation in marketing, new products and high dimensional models in operations management, and clustered regression models in machine learning and statistics.

First of all, product diffusion and innovation has been widely studied in the marketing literature. As the seminal paper in this area, Bass (1969) develops a simple yet strong model that estimates how a new product
diffuses through a population. The Bass model predicts lifetime sales based on a few parameters: market size, coefficient of innovation, and coefficient of imitation. For reviews on further research on product innovation and diffusion in marketing we refer to Chandrasekaran and Tellis (2007) and Fan et al. (2017). The Bass model, while widely used, makes lifecycle predictions. Hence, it might make significant prediction error in the introductory sales prediction, the focus of this paper. In comparison, we propose a model that is grounded in practice and is data-driven by incorporating machine learning and statistical tools.

A second stream of literature comes from operations management. In particular, recent studies have considered production, inventory, and pricing of new products. Recently, Hu et al. (2016) use a two step approach to forecasting the sales of new products and show that mean absolute errors reduce by around 2-3%. Their forecasting model fits lifecycle curves to products, then clusters these products, and aggregates the predictions. In contrast to estimating the entire lifecycle of a product, our forecasting problem focuses on the product’s introductory period. After this period, the acquired sales data can be used by existing models (Ali et al. 2009, Huang et al. 2014) to generate better forecasts. Inherently, this means our approach deals with the more complex and most uncertain period during a product’s lifecycle. Furthermore, instead of the two step approach, we propose an algorithm that estimates clusters and forecasting models jointly, while allowing each cluster’s forecasting model to be any machine learning or statistical regression model. Specifically, this allows our model to incorporate other features such as pricing, marketing, and distribution into the prediction models.

With regards to new product pricing, more attention has recently been placed on pricing when the demand curve is unknown. Specifically, a new product is released into the market and dynamic pricing is used as a tool to understand the underlying demand. These studies set prices carefully to maximize revenue while balancing exploration and exploitation. Keskin and Zeevi (2014) study asymptotically optimal policies for pricing a product with linear demand, but assume either no or limited data is available. Ban and Keskin (2017) extend to the setting where customer characteristics are available and pricing policies can be personalized. In certain industries, experimenting with the price of a new product is not desired or allowed. This problem is analyzed by Cohen et al. (2015) who propose a simple pricing policy based on linear demand curves that performs well for many parametric forms of unknown demand curves. Perakis and Singhvi (2017) analyze the case of learning with limited price experimentation under non-parametric demand functions. To contrast with this literature, we assume that historical data is available for comparable products and focus on the forecasting problem.

Concerning production and inventory management of new products, several models have been developed to improve decision making before the product launch, for example, Caro and Gallien (2010), Gallien et al. (2015), Chen et al. (2017), and Ban et al. (2017). These studies optimize production and inventory decisions assuming a particular structure on the demand for the new products. This contrasts with our focus.
on improving the sales forecasting model itself. In turn, our forecasts can also be used to improve these operational decision models.

From a theoretical point of view, this work is related to the recent surge in operational models that involve high dimensional features about people or products. This increase in data enables personalized or product-specific policies. Regularization is used to control model complexity and ensures that decisions can generalize to when new people or products arrive. Among others, Bastani and Bayati (2015), Javanmard and Nazerzadeh (2016), Ban and Keskin (2017) analyze various operations management problems, such as pricing and healthcare delivery, from a high dimensional perspective. Our work differs in that regularization is used to improve the accuracy of predictions instead of prescriptions. Datasets have grown in both observations (e.g., number of people and products) and features (e.g., information on people and products). Therefore, our models involve high dimensional data, and for the aforementioned reason, we use regularization to avoid overfitting. Naturally, the model is to be used to enhance operational models, but our main focus is the forecasting problem itself.

Finally, the model that was applied to the data of our industry collaborators uses a LASSO regularized regression model for each cluster. In this setting, our model is related to clusterwise linear regression. In this problem, the model clusters observations and fits linear regression models to these clusters simultaneously. Bertsimas and Shioda (2007) use integer programming techniques to solve this problem, while others such as Späth (1979) and Park et al. (2016) use heuristic algorithms instead. We expand the model by considering regularized linear regression models for each cluster. Furthermore, our estimation algorithm functions even when the forecasting models of a cluster take non-linear forms such as generalized linear models and regression trees.

2. Motivation and Data from Practice

Before introducing our model, we describe the problem faced by our two industry partners, we discuss their current approaches to new product sales forecasting, and describe the challenges of these approaches. Finally, we use the data of our industry partners to motivate the clustered forecasting model proposed in this paper.

Our industry partners, and firms more generally, invest millions of dollars in innovation every year. The success of a new product is partially dependent on making the right operational decisions surrounding the product launch. These decisions are guided by sales forecasts, and therefore, our industry partners note the importance of accurate forecasting. Though, accuracy is not the only metric of importance. It is also important that the forecasting tool is interpretable and scalable. The model needs to be interpretable because new products are surrounded by large uncertainty and models that are not easy to understand will receive less usage from practitioners. In certain industries, such as fast moving consumer goods and fast fashion, new products are introduced frequently and scalability is important. Their extensive product assortment
leads to large datasets to fit forecasting models on. Additionally, due to frequent product introductions, these firms need to forecast often. In Table 1 we show the number of products that our partners introduced in several consumer goods and fashion categories during the corresponding periods.

<table>
<thead>
<tr>
<th>Industry</th>
<th>Category</th>
<th>Time Period</th>
<th>Number of New Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer goods</td>
<td>Facial care</td>
<td>2012-2016</td>
<td>219</td>
</tr>
<tr>
<td>Consumer goods</td>
<td>Body care</td>
<td>2013-2017</td>
<td>71</td>
</tr>
<tr>
<td>Consumer goods</td>
<td>Baby care</td>
<td>2013-2017</td>
<td>122</td>
</tr>
<tr>
<td>Fashion</td>
<td>Subsegment 1</td>
<td>April-June, 2016</td>
<td>75</td>
</tr>
<tr>
<td>Fashion</td>
<td>Subsegment 2</td>
<td>April-June, 2016</td>
<td>66</td>
</tr>
</tbody>
</table>

Over a period of five years and three different product categories, the consumer goods manufacturer released a total of 412 new products. At an even faster rate, the fashion retailer introduced 143 new products over just three months and two subcategories. The faster pace of innovation also corresponds to products with shorter lifetimes. This does not change the fundamental problem of forecasting sales of new products, but it has an effect on the scale of the problem. For example, yearly forecasts might be adequate for production, procurement, and inventory decisions at a manufacturer, while weekly forecasts are needed for distribution and pricing decisions at a retailer. In particular, where our consumer goods partner is interested in forecasting first year sales in an entire country, the fashion retail partner needs accurate forecasts for the first half-week at a store or regional level.

Forecasting new product sales is a clear challenge for both partners. Often, errors are too large, namely over 50% off. In discussions, our partners explained that using these forecasts would lead to wrong decisions. As an example, applying current forecasting methods to a subcategory, only 16 out of 31 product introductions were predicted accurately enough to aid in effective decision making. To illustrate the difficulty, Figure 1 shows the actual sales over the first six months of two new products that were released by the consumer goods manufacturer. The two products belong to the same product subcategory, yet behave very differently. While the first product shows an increasing trend in sales over the introductory horizon, the sales of the second product decline over the same months after release.

![Figure 1](image1.png)  
**Figure 1**  
Actual sales data of two new products over the first six months after introduction.
Evidently, consumer response to the two new products was very different. The cause of this difference can be attributed to many factors: pricing, promotions, distribution, product attributes (such as size, packaging, and color), and other latent factors. From Johnson & Johnson Consumer Companies Inc., we have access to five year long datasets in which each observation describes the monthly sales of a SKU (Stock Keeping Unit) in an entire trade channel. Interestingly, this dataset includes the sales data from competitor’s products at the same level of granularity. Available product features include the date of product introduction, the product’s categorization, price, promotional events, sizing, packaging, chemical composition, claimed benefits, as well as distribution measures such as the number of stores selling the product and %ACV (Percentage of All Commodity Volume), which is a measure of distribution across stores. The fast fashion retailer gave us access to a dataset containing three months of half-weekly store sales for individual SKUs. The data contains product features such as date of introduction in both the brick-and-mortar and the online channel, product categorization, price, color, style, and prior clickstream data if the product was introduced online earlier (e.g., cumulative views online, cumulative add-to-carts online, cumulative remove-from-carts online). More information on the data can be found in Sections 5 and 6. Unfortunately, even when accounting for all these product features and analyzing a specific subcategory, a single model to predict sales of new products is often unable to capture certain hidden factors. One visible example of these latent factors is the upwards or downwards sales trend in Figure 1. This trend can be hard to predict before a product launch, as no historical sales data is available.

As a remedy, our partners and many other firms in the fast moving consumer goods and fast fashion retailing business, use prediction tools that combine market surveys, expert opinions, and comparable products. Apart from being expensive, these research approaches are time consuming. Moreover, the final product often differs from the prototype product during market research, which undermines the accuracy of these forecasts. Hence, high costs and long prediction lead times, make these tools impractical for most of the smaller and even medium sized new product launches. This problem has been more pronounced in recent times when the frequency of new product launches has been increasing. Therefore, sometimes, firms have to resort to more qualitative methods to forecast new product sales. One collaborator has established a forecasting technique where product managers use their expertise to find products comparable to the new product and then use their actual sales as a forecast. The hope is that these clusters of comparable products capture shared latent factors. Thus, if the right comparables have been selected, the effects of latent factors on sales should be captured by the actual sales of the clustered products. For this reason and the ease that practitioners have with forecasting using comparables, in the remainder of the paper, we develop an algorithm and subsequently a tool which clusters products and simultaneously fits demand forecasting models for these clusters.
3. Cluster-while-Estimate Model

In what follows, we introduce our sales forecasting model for new products. As described, many firms use a cluster-then-estimate model in which experts select past products similar to the new product and use their average sales as a forecast. Unfortunately, initial tests on data from our industry partners showed that this approach produces weak results even when using data-driven clustering (see Sections 5 and 6). Nonetheless, our new product sales forecasting model is related to this practice, namely we propose a cluster-while-estimate model that clusters products and fits sales forecasting models to each cluster simultaneously. The main difference between the two approaches is that the data-driven version of the cluster-then-estimate model clusters based on product feature similarity, fixes these clusters, and then forecasts the cluster’s sales. Instead, our cluster-while-estimate model clusters products and estimates forecasting models simultaneously, thereby clustering on the similarity in terms of both product features as well as sales behavior of the products.

3.1. General Model

Formalizing our approach, we are interested in predicting sales of a new product, \( y_0 \in \mathbb{R} \), based on \( m \) product features that are available before introduction, \( x_0 \in \mathbb{R}^m \). Examples of these product features include the aforementioned data such as the product’s regular price, brand, and sizing. Naturally, firms also use historical sales data to forecast future sales of existing products, but lack this data for new products. As a result, the challenge is to provide an accurate sales forecast for the new product without an indication of its rough sales potential. Through our industry partners we have access to data on past products that were once new. By \( y_i \in \mathbb{R} \) and \( x_i \in \mathbb{R}^m, i = 1, \ldots, n \), we denote the sales and product feature data for \( n \) past new products.

Using this data, we determine a model describing the sales of these past products, which can then be used to predict sales for the new product. Specifically, we consider \( \ell \) clusters of products, each with a different sales generating model, but where the products within a cluster share the same model. To be precise, we propose the following sales generating model:

\[
y_i = \sum_{k=1}^{\ell} z_{ik} f_k(x_i) + \epsilon_i, \quad i = 0, 1, \ldots, n, \quad (1)
\]

where \( z_{ik} \in \{0, 1\} \) indicates whether product \( i \) belongs to cluster \( k \), \( f_k(x_i) \) is the sales forecasting model for a product in cluster \( k \) with features \( x_i \) (i.e., a particular functional form to estimate the conditional expected sales for a product in cluster \( k \) with features \( x_i \)), and \( \epsilon_i \) is assumed to be a zero-mean random noise. The conditional expected sales of each cluster can be a highly non-linear function of the available features. Thus, our estimation approach needs to be able to incorporate a large variety of cluster forecasting models. Examples of \( f_k \) include linear regression models (used in our application), generalized linear models, non-linear regression models, regression trees, and random forests.
In addition, our approach needs to allow for estimating sparse models. To exemplify this necessity, consider the case where \( f_k \) is a linear regression model. The total number of parameters of model (1) is then not just the number of product features, \( m \), but rather a multiple of the number of product features and the number of clusters, \( m\ell \). This means that the dimension of the model can grow quickly as the number of clusters grows. Additionally, some of the features that are included in the model might not affect the sales of certain groups of products. The goal of the regularizer is to guard against fitting an overly complex model. The estimation procedure is likely to find a model that only uses the most important predictors of a cluster’s sales. To account for sparsity, we add a regularization penalty to the objective, which has the added benefit of producing a model that is robust to measurement error (Bertsimas and Copenhaver 2017).

Now, if we want to forecast sales in accordance with model (1), we need to deal with two problems. First, we need to assign past products to clusters, \( \hat{z}_{ik} \), and estimate each cluster forecasting model using the past products’ sales, \( \hat{f}_k \). Second, we still need to determine the new product’s cluster assignment before we can forecast its sales.

3.2. Estimation for Past Products

Before we can forecast sales, we have to describe the estimation procedure of the sales generating model. For model (1), we need to decide in which cluster a product lies, \( \hat{z}_{ik} \), and estimate each cluster’s forecasting model, \( \hat{f}_k \). The general cluster-while-estimate problem is formulated as the following problem \( (P) \):

\[
\min_{z_{ik}/k} \sum_{i=1}^{n} L \left( y_i, \sum_{k=1}^{\ell} z_{ik} f_k(x_i) \right) + \lambda R(f_1, \ldots, f_{\ell})
\]

s.t. \( \sum_{k=1}^{\ell} z_{ik} = 1, \quad i = 1, \ldots, n \) \hspace{2cm} (2a)

\( z_{ik} \in \{0, 1\}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell. \) \hspace{2cm} (2c)

The objective (2a) represents the minimization of regularized prediction error. For each past product \( i \), we observe \( y_i \) sales and forecast \( \sum_{k=1}^{\ell} z_{ik} f_k(x_i) \) sales. For any error in this prediction we incur a loss \( L(y_i, \sum_{k=1}^{\ell} z_{ik} f_k(x_i)) \). In addition, we regularize the cluster forecasting models through a regularizer \( R(f_1, \ldots, f_{\ell}) \) and a penalty parameter \( \lambda \geq 0 \) that balances the loss and regularizer. The form of the loss, \( L \), and regularizer, \( R \), largely depends on the forecasting model, \( f_k \), that is chosen. For example, to estimate the parameters of a LASSO regularized regression model (as in our application), we use the squared error loss with a regularizer that sums the absolute values of the regression parameters. Together, the constraints (2b) and (2c) ensure that each product gets assigned to exactly one cluster.

In general, problem \( (P) \) is hard to solve due to its integer and possibly non-linear structure. Depending on the industry, firms need to solve the cluster-while-estimate model for many different product categories in a limited amount of time. For example, a consumer goods manufacturer might need to solve this problem every month for a few categories, whereas a fast-fashion retailer needs to solve it overnight for many categories. Thus, we develop a general solution approach that quickly finds approximate solutions to problem
We use an algorithm that iteratively pairs products with the least-error cluster forecasting model and then estimates cluster forecasting models based on these new clusters. More formally, our cluster-while-estimate algorithm can be described as follows:

1. Input the number of clusters \( \ell \) and the penalty parameter \( \lambda \). Initialize the assignment of products to clusters \( \hat{z}_{ik}^{(0)} \), either randomly, or through a clustering method like k-means or hierarchical clustering.

2. Approximate the optimal solution of problem \((P)\) by iteratively re-estimating the cluster forecasting model and re-clustering the products. For iteration \( t = 1, \ldots, T \):
   a. Solve \((P)\) with \( z_{ik} = \hat{z}_{ik}^{(t-1)} \) to find \( \hat{f}_k^{(t)} \).
   b. Solve \((P)\) with \( f_k = \hat{f}_k^{(t)} \) to find \( \hat{z}_{ik}^{(t)} \).
   c. Terminate with \( \hat{z}_{ik} \) and \( \hat{f}_k \) if \( t = T \) or \( \hat{z}_{ik}^{(t)} = \hat{z}_{ik}^{(t-1)} \), otherwise return to step 2a.

First off, we note that step 2a and 2b of this algorithm can be solved efficiently for many forms of the cluster forecasting model. In particular, step 2a involves solving \( \ell \) separate estimation problems. For many functional forms of the cluster forecasting model, we can use an existing efficient statistical or machine learning algorithm to solve the estimation problems. Then, step 2b reassigns products to the cluster that would result in the smallest prediction loss for that product. This can be done by sorting the product’s prediction loss in each cluster and selecting the cluster with the smallest loss. Overall, in Proposition 1, we prove that the algorithm converges to a locally optimal solution in polynomial time.

**Proposition 1.** The cluster-while-estimate algorithm has the following two properties:

1. If the estimation method for \( \hat{f}_k \) runs in \( O(p(n,m,\ell)) \) polynomial time, where \( p(n,m,\ell) \) is any polynomial of \( n, m, \) and \( \ell \), then the cluster-while-estimate algorithm runs in \( O(Tp(n,m,\ell) + Tnm\ell) \) polynomial time.

2. The cluster-while-estimate algorithm converges, which happens when \( \hat{z}_{ik}^{(t)} = \hat{z}_{ik}^{(t-1)} \).

**Proof.** We prove the two properties separately:

1. In analyzing the computational complexity of the algorithm, the operations of step 1 are dominated by step 2. The first part of each iteration of step 2 is to determine \( \hat{f}_k \). Thus, this part takes \( O(p(n,m,\ell)) \) time. In the second part, we re-cluster all products. In this part, we calculate for all \( n \) products the prediction of all \( \ell \) clusters, which consists of \( m \) multiplications and additions to obtain \( \sum_{j=1}^{m} \beta_{kj} x_{ij} \). Afterwards we square the difference between the sales and prediction to obtain the prediction loss, and compare this term with the smallest prediction loss until then. Thus, this part takes \( O(nm\ell) \) time. Hence, altogether running \( T \) iterations (or fewer when \( \hat{z}_{ik}^{(t)} = \hat{z}_{ik}^{(t-1)} \) for \( t < T \)), the algorithm runs in \( O(Tp(n,m,\ell) + Tnm\ell) \) time.

2. Let \( \mathcal{L} \) denote the penalized loss of the objective

\[
\mathcal{L}(z_{ik}, f_k) = \sum_{i=1}^{n} L \left( y_i, \sum_{k=1}^{\ell} z_{ik} f_k(x_i) \right) + \lambda R(f_1, \ldots, f_\ell).
\]
During the algorithm, step 2a fixes $\hat{z}_{ik}^{(t-1)}$ and minimizes $\mathcal{L}$ to find $\hat{f}_k^{(t)}$, while step 2b fixes $\hat{f}_k^{(t)}$ and minimizes $\mathcal{L}$ by re-clustering products to find $\hat{z}_{ik}^{(t)}$, which means

$$\mathcal{L}(\hat{z}_{ik}^{(t)}, \hat{f}_k^{(t)}) \leq \mathcal{L}(\hat{z}_{ik}^{(t-1)}, \hat{f}_k^{(t)}) \leq \mathcal{L}(\hat{z}_{ik}^{(t-1)}, \hat{f}_k^{(t-1)}).$$

Thus, the objective decreases in every iteration of step 2. Since $\mathcal{L}$ is lower bounded, the algorithm must converge to a solution in bounded time. If convergence happens at time $t-1$, this can be checked from time $t$ onwards if the solution remains unchanged, i.e., $\hat{f}_k^{(t)} = \hat{f}_k^{(t-1)}$ and $\hat{z}_{ik}^{(t)} = \hat{z}_{ik}^{(t-1)}$ for $\tau \geq t$. For this to happen, we only need that $\hat{z}_{ik}^{(t)} = \hat{z}_{ik}^{(t-1)}$, because the deterministic algorithms of steps 2a and 2b will find the same solutions at every iteration onwards.

Even though the algorithm terminates with a locally optimal solution when $T$ is large enough, the speed of the algorithm allows us to restart the algorithm multiple times from different starting points, thereby increasing the likelihood of finding the globally optimal solution. In addition, this speed enables us to tune the parameters $\ell$ and $\lambda$. We run the algorithm for different combinations of $\ell$ and $\lambda$ on a training set and compute out-of-sample forecasting metrics on a validation set. The tuned parameters are then chosen to yield the best out-of-sample forecasting metrics.

### 3.3. Forecasting for New Products

Having estimated the sales generating model (1), we can now forecast sales for the new product, $y_0$. However, we still need to decide to which cluster the new product belongs, as past products are clustered based on actual sales data which is unavailable for the new product.

For this, we can use any multiclass classification method such as a multinomial logistic regression (which we use in our application), support vector machines, classification trees, and random forests. We propose to train this cluster assignment model by using the cluster assignments for past products, $\hat{z}_{ik}$, as the dependent variable, and the features of past products, $x_i$, as the independent variables. This creates a mapping $\hat{p}_k$ from the product feature space (excluding sales) to the clusters. Depending on the classification method, its predicted assignment for the new product, $\hat{z}_{0k} = \hat{p}_k(x_0)$, will either be in the form of an assignment to a cluster or probabilities of assignment to clusters. Both cases can be captured by letting $\hat{p}_k(x_0)$ give the probability that the new product with features $x_0$ lies in cluster $k$. In the case where the classification method gives a pure cluster assignment the cluster’s corresponding probability is set to 1, while others are set to 0.

In either case, the new product sales forecast is given by

$$\hat{y}_0 = \sum_{k=1}^{\ell} \hat{z}_{0k} \hat{f}_k(x_0) = \sum_{k=1}^{\ell} \hat{p}_k(x_0) \hat{f}_k(x_0).$$

(3)

When $\hat{p}_k(x_0)$ assigns probabilities to clusters, the sales forecast (3) is a weighted average of the cluster forecasts weighted by the cluster probabilities.
4. Application of Regularized Linear Regression

In this section, we specify the cluster forecasting model, \( f_k \), and cluster assignment model, \( p_k \), that we used in collaboration with our industry partners. In this work, we consider \( f_k \) to be a regularized linear regression model, and we use a multinomial logistic regression to estimate \( p_k \). This section will cover how the general cluster-while-estimate problem can be reformulated as a mixed integer quadratic optimization problem in the case of regularized linear regression. As this reformulation is still computationally inefficient, we will discuss our cluster-while-estimate algorithm and prove a bound on its forecasting error. Finally, we will briefly discuss the details of the cluster assignment model.

Formally, we consider the following linear cluster forecasting model:

\[
\tilde{y}_i = \sum_{k=1}^{\ell} \tilde{z}_{ik} \sum_{j=1}^{m} \beta_{kj} x_{ij} + \epsilon_i, \quad i = 0, 1, \ldots, n,
\]

where \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \) for all \( i = 0, 1, \ldots, n \). The linear sales generating model can also be given by the following vector representation:

\[
y = (Z \ast X) \beta + \epsilon,
\]

Here, \( y = (y_1, \ldots, y_n) \in \mathbb{R}^n \) is a column vector of product sales, \( X = (x_{ij}) \in \mathbb{R}^{n \times m} \) is a block matrix whose blocks are rows of product features, \( Z = (z_{ik}) \in \{0, 1\}^{n \times \ell} \) is a block matrix whose blocks are rows of cluster assignments, \( \beta = (\beta_{11}, \beta_{12}, \ldots, \beta_{1m}, \beta_{21}, \ldots, \beta_{m\ell}) \in \mathbb{R}^{m\ell} \) is a column vector that stacks the regression coefficients (this vector is \( s \)-sparse if at most \( s \) elements of \( \beta \) are non-zero), and \( \epsilon = (\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}^n \) is a column vector of zero-mean random error. In this representation, we use the Khatri-Rao product \( Z \ast X \), which is defined as follows:

\[
Z \ast X = \begin{pmatrix}
\tilde{z}_{11} \tilde{z}_{12} \cdots \tilde{z}_{1\ell} \\
\tilde{z}_{21} \tilde{z}_{22} \cdots \tilde{z}_{2\ell} \\
\vdots \ddots \vdots \\
\tilde{z}_{n1} \tilde{z}_{n2} \cdots \tilde{z}_{n\ell}
\end{pmatrix}
\ast
\begin{pmatrix}
x_{11} x_{12} \cdots x_{1\ell} \\
x_{21} x_{22} \cdots x_{2\ell} \\
\vdots \ddots \vdots \\
x_{n1} x_{n2} \cdots x_{n\ell}
\end{pmatrix}
= \begin{pmatrix}
\tilde{z}_{11} x_{11} \tilde{z}_{11} x_{12} \cdots \tilde{z}_{11} x_{1m} & \tilde{z}_{12} x_{11} \tilde{z}_{12} x_{12} \cdots \tilde{z}_{12} x_{1m} & \cdots & \tilde{z}_{1\ell} x_{11} \tilde{z}_{1\ell} x_{12} \cdots \tilde{z}_{1\ell} x_{1m} \\
\tilde{z}_{21} x_{21} \tilde{z}_{21} x_{22} \cdots \tilde{z}_{21} x_{2m} & \tilde{z}_{22} x_{21} \tilde{z}_{22} x_{22} \cdots \tilde{z}_{22} x_{2m} & \cdots & \tilde{z}_{2\ell} x_{21} \tilde{z}_{2\ell} x_{22} \cdots \tilde{z}_{2\ell} x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{z}_{n1} x_{n1} \tilde{z}_{n1} x_{n2} \cdots \tilde{z}_{n1} x_{nm} & \tilde{z}_{n2} x_{n1} \tilde{z}_{n2} x_{n2} \cdots \tilde{z}_{n2} x_{nm} & \cdots & \tilde{z}_{n\ell} x_{n1} \tilde{z}_{n\ell} x_{n2} \cdots \tilde{z}_{n\ell} x_{nm}
\end{pmatrix}.
\]

Under the assumption that the conditional expected sales is also a linear function of the dependent variables, we use \( Z^* = (z_{ik}^*) \) and \( \beta^* = (\beta_{kj}^*) \) to denote the cluster assignments and linear regression coefficients of the ‘true’ model, while our estimates are denoted by \( \hat{Z} = (\hat{z}_{ik}) \) and \( \hat{\beta} = (\hat{\beta}_{kj}) \).

For a linear regression model, we use the ordinary least squares estimation method, which implies that

\[
L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2. \quad \text{To this, we add LASSO regularization on the linear regression coefficients, which means that} \quad R(\beta_1, \ldots, \beta_\ell) = \sum_{k=1}^{\ell} \sum_{j=1}^{m} |\beta_{kj}|. \quad \text{Here, we consider LASSO regularization, but we note that the results below can be extended to the case of ridge regularization. In preliminary experimentation, we estimated the ridge regularized model and it produced worse results, plausibly due to overfitting. Generally,}
\]
ridge regularization is not able to exclude unimportant variables while LASSO regularization can generate sparse models (Bühlmann and van de Geer 2011).

Having specified $f_0$, $L$, and $R$, we adapt problem $(P)$ to formulate the linear cluster-while-estimate problem that estimates the linear sales generating model $(4)$ as the following mixed integer non-linear optimization problem $(P_{\text{Linear}})$:

$$
\min_{z_{ik}, \beta_{kj}} \sum_{i=1}^{n} \left( y_i - \sum_{k=1}^{\ell} \sum_{j=1}^{m} \beta_{kj} x_{ij} \right)^2 + \lambda \sum_{k=1}^{\ell} \sum_{j=1}^{m} |\beta_{kj}| \\
\text{s.t. } \sum_{k=1}^{\ell} z_{ik} = 1, \quad i = 1, \ldots, n \\
\quad \sum_{k=1}^{\ell} \sum_{j=1}^{m} r_{kj} \\
\quad z_{ik} \in \{0, 1\}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell.
$$

(6a)
(6b)
(6c)

Earlier, we mentioned that problem $(P)$ is hard due to its integer decision variables and possibly non-linear objective. Here, problem $(P_{\text{Linear}})$ shows that even a linear regression model without regularization has a non-linear objective, namely biquadratic. This makes the problem hard to solve, even for commercial solvers. In fact, in Megiddo and Tamir (1982), problem $(P_{\text{Linear}})$ is proven to be NP-hard for the case where $\lambda = 0$. Clearly, adding regularization generalizes the problem, which therefore remains NP-hard. Nonetheless, in Proposition 2, we show that problem $(6)$ can be reformulated as a mixed-integer quadratic problem that commercial solvers can attempt to solve.

**Proposition 2.** The linear cluster-while-estimate problem can be reformulated as the following mixed-integer quadratic optimization problem $(P_{\text{Linear-}R})$, where $M$ is a big constant:

$$
\min_{z_{ik}, \beta_{kj}, q_{ikj}, r_{kj}} \sum_{i=1}^{n} \left( y_i - \sum_{k=1}^{\ell} \sum_{j=1}^{m} q_{ikj} x_{ij} \right)^2 + \lambda \sum_{k=1}^{\ell} \sum_{j=1}^{m} r_{kj} \\
\text{s.t. } \sum_{k=1}^{\ell} z_{ik} = 1, \quad i = 1, \ldots, n \\
\quad \sum_{k=1}^{\ell} \sum_{j=1}^{m} q_{ikj} x_{ij} - \sum_{k=1}^{\ell} \sum_{j=1}^{m} \beta_{kj} x_{ij} = M (1 - z_{ik}) - M z_{ik}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m \\
\quad \beta_{kj} \geq 0, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m \\
\quad z_{ik} \in \{0, 1\}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell.
$$

(7a)
(7b)
(7c)
(7d)
(7e)
(7f)
(7g)

**Proof.** Consider problem $(P_{\text{Linear}})$ where $q_{ikj}$ substitutes $z_{ik}\beta_{kj}$ and $r_{kj}$ substitutes $|\beta_{kj}|$, namely,

$$
\min_{z_{ik}, \beta_{kj}} \sum_{i=1}^{n} \left( y_i - \sum_{k=1}^{\ell} \sum_{j=1}^{m} q_{ikj} x_{ij} \right)^2 + \lambda \sum_{k=1}^{\ell} \sum_{j=1}^{m} r_{kj} \\
\text{s.t. } \sum_{k=1}^{\ell} z_{ik} = 1, \quad i = 1, \ldots, n \\
\quad \sum_{k=1}^{\ell} \sum_{j=1}^{m} r_{kj} \\
\quad z_{ik} \in \{0, 1\}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell.
$$

(8a)
(8b)
(8c)
(8d)
(8e)
This substituted problem is not identical to \((P_{Linear})\) as it does not specify the link between \(z_{ik}, \beta_{kj}, q_{ikj},\) and \(r_{kj}\). For this, we need to add constraints that define how \(q_{ikj}\) and \(r_{kj}\) depend on \(z_{ik}\) and \(\beta_{kj}\). The following constraints define that \(q_{ikj} = \beta_{kj}\),

\[
-M(1 - z_{ik}) \leq q_{ikj} - \beta_{kj} \leq M(1 - z_{ik}), \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m.
\]

\[
-Mz_{ik} \leq q_{ikj} \leq Mz_{ik}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m.
\]

Finally, to ensure that \(r_{kj} = |\beta_{kj}|\) we add the following constraints,

\[
\begin{align*}
  r_{kj} &\geq \beta_{kj}, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m, \\
  r_{kj} &\geq -\beta_{kj}, \quad k = 1, \ldots, \ell, \quad j = 1, \ldots, m.
\end{align*}
\]

In the objective, \(\lambda r_{kj}\) appears as an additive term and \(\lambda \geq 0\), and since we are minimizing \(r_{kj}\) is set as small as allowed and pushed to \(|\beta_{kj}|\). Hence, after adding these 4 sets of constraints to the substituted problem it forms \((P_{Linear-R})\) and is equivalent to \((P_{Linear})\).

Unfortunately, this reformulation can not solve large instances of the linear cluster-while-estimate problem in a reasonable time. To illustrate the need for our cluster-while-estimate algorithm, we run computational experiments to test the running time of the \((P_{Linear-R})\). For different settings of \(n, m,\) and \(\ell\), we use Gurobi 7.0.2 to solve \((P_{Linear-R})\) programmed in Julia/JuMP (Dunning et al. 2017) on an Intel Core i5-4690K @ 3.5GHz CPU and 8 GB RAM. The average running time is estimated over 50 randomized instances, where \(z_{ik} \sim Mult(1/\ell, \ldots, 1/\ell) \forall i, \beta_{kj} \sim U(0, 1) \forall k, j, x_{ij} \sim U(0, 1) \forall i, j,\) and \(\varepsilon_{i} \sim \mathcal{N}(0, 1) \forall i\) are drawn randomly from the respective distributions. Setting \(n = 10, m = 5, \ell = 2,\) and averaging over 50 instances, the running time of \((P_{Linear-R})\) is 0.19 seconds. However, when \(n = 100, m = 5, \ell = 2,\) the average running time exceeds 10 minutes. This shows that solving the problem \((P_{Linear-R})\) on a large scale (often there are more than 100 products and 5 product features) is not practical. In comparison, the cluster-while-estimate algorithm (with \(T = 100\) iterations) on average takes 0.02 seconds when \(n = 10, m = 5, \ell = 2,\) while \(n = 100, m = 5, \ell = 2\) takes 0.04 seconds. The cluster-while-estimate algorithm scales well, solving large instances within 8.74 seconds \((n = 10000, m = 100, \ell = 10)\) to 27.72 seconds \((n = 10000, m = 200, \ell = 20)\).

Contrasting with the cluster-while-estimate algorithm, another approximation to problem \((P_{Linear})\) can be obtained by relaxing constraint \((7g)\) to \(0 \leq z_{ik} \leq 1, i = 1, \ldots, n, k = 1, \ldots, \ell.\) This relaxed problem reinterprets \(z_{ik}\) as the probability that product \(i\) belongs to cluster \(k.\) However, we notice that this method overfits. In computations, the first term of the objective \((6a)\) consistently equals 0. Also, we notice that \(z_{ik} = 1/\ell\) in nearly all runs. For large enough \(M,\) this means that the lower and upper bounds of constraints \((7c)\) and \((7d)\) create too wide an interval. In turn, this allows the model to set \(q_{ikj}\) independent of \(\beta_{kj}\). Hence, the model can minimize the loss by setting the forecast \(\sum_{k=1}^{\ell} \sum_{j=1}^{m} q_{ikj} x_{ij}\) to perfectly fit the actual sales \(y_i.\)

Ultimately, this means there is a need for the linear cluster-while-estimate algorithm. In the following, we specify the details of the estimation and forecasting procedure that was used in collaboration with our industrial partners. First, the following linear cluster-while-estimate algorithm is run:
1. Input the number of clusters \( \ell \) and the penalty parameter \( \lambda \). Initialize the assignment of products to clusters \( \tilde{z}_{ik}^{(0)} \) randomly.

2. Approximate the optimal solution of problem \((P_{\text{Linear}})\) by iteratively re-estimating the cluster forecasting model and re-clustering the products. For iteration \( t = 1, \ldots, T \):
   a. For cluster \( k = 1, \ldots, \ell \), fit LASSO regression of sales on the product features of products in cluster \( k \), i.e., fit LASSO regression of \( y_i \) on \( x_i \) for \( i \) such that \( \tilde{z}_{ik}^{(t-1)} = 1 \) to obtain \( \hat{\beta}_{kj}^{(t)} \).
   b. For product \( i = 1, \ldots, n \), compute the distance between product \( i \)'s sales and each cluster’s forecast and assign product \( i \) to the closest cluster, i.e., for \( k = \arg \min \epsilon (y_i - \sum_{j=1}^{m} \hat{\beta}_{kj}^{(t)} x_{ij})^2 \) set \( \tilde{z}_{ik}^{(t)} = 1 \) and \( \tilde{z}_{ik}^{(t)} = 0 \) for \( k' \neq k \).
   c. Terminate with \( \tilde{z}_{ik} \) and \( \hat{\beta}_{kj} \) if \( t = T \) or \( \tilde{z}_{ik}^{(t)} = \tilde{z}_{ik}^{(t-1)} \), otherwise return to step 2a.

Second, the cluster assignment model is estimated by fitting a multinomial logistic regression of the past product cluster assignments \( \tilde{z}_{ik} \), on the available product features \( x_i \) (also available for the new product \( x_0 \)). In particular, this estimation results in the multinomial logistic regression coefficients \( \gamma_{kj} \) which define the following cluster assignment probabilities of the new product:

\[
\tilde{z}_{ik} = \frac{\exp \left( \sum_{j=1}^{m} \gamma_{kj} x_{0j} \right)}{\sum_{k'=1}^{\ell} \exp \left( \sum_{j=1}^{m} \gamma_{k'j} x_{0j} \right)}.
\tag{9}
\]

Finally, plugging (9) into (3), the new product forecast becomes

\[
\hat{y}_0 = \sum_{k=1}^{\ell} \frac{\exp \left( \sum_{j=1}^{m} \gamma_{kj} x_{0j} \right)}{\sum_{k'=1}^{\ell} \exp \left( \sum_{j=1}^{m} \gamma_{k'j} x_{0j} \right)} \sum_{j=1}^{m} \hat{\beta}_{kj} x_{0j}.
\tag{10}
\]

This process assumes that an initial number of clusters \( \ell \) and the penalty parameter \( \lambda \) have been chosen. As we described before, we tune these parameters by running the algorithm on a training set and selecting those parameters that result in the best out-of-sample forecasting metrics on a validation set. In Sections 5 and 6, we describe the results of applying the above procedure to our two industry collaborators. Next, we investigate the theoretical performance of the cluster-while-estimate algorithm. Specifically, we analyze the expected forecasting error of the solution given by the cluster-while-estimate algorithm. We measure the error in terms of the mean squared forecasting error between the sales forecast from the estimated model \( \sum_{k=1}^{\ell} \tilde{z}_{ik} \sum_{j=1}^{m} \hat{\beta}_{kj} x_{ij} = (\hat{Z} \ast X) \hat{\beta} \) and the ‘true’ conditional expected sales \( \sum_{k=1}^{\ell} \tilde{z}_{ik}^* \sum_{j=1}^{m} \beta_{kj}^* x_{ij} = (Z^* \ast X) \beta^* \):

\[
\frac{1}{n} \sum_{i=1}^{n} \left( \sum_{k=1}^{\ell} \tilde{z}_{ik} \sum_{j=1}^{m} \hat{\beta}_{kj} x_{ij} - \sum_{k=1}^{\ell} \tilde{z}_{ik} \sum_{j=1}^{m} \hat{\beta}_{kj} x_{ij} \right)^2 = \frac{1}{n} \| (Z^* \ast X) \beta^* - (\hat{Z} \ast X) \hat{\beta} \|^2_2.
\tag{11}
\]

The mean squared forecasting error (11) is a natural measure for the difference between our forecast and the best possible forecast. This difference comes from the difficulty in estimating \( z_{ik} \) and \( \beta_{kj} \) exactly, which is caused by the fact that we gather noisy measurements (due to \( \epsilon_i \)) instead of the ‘true’ conditional expected sales. Our theoretical results show that, with high probability, the mean squared forecasting error (11) is bounded. In proving our probabilistic guarantees, we use Lemma 1 to show that the prediction error due to the noisy measurements is small with high probability.
LEMMA 1. If $||X_j||^2 \leq \rho$ and $\lambda = 2\sigma \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}$, then for any allowable $Z$ and $0 < \delta < 1$,

$$\mathbb{P}\left(||\varepsilon^T (Z*X)||_{\infty} \leq \frac{\lambda}{2}\right) \geq 1 - \delta.$$ 

**Proof.** This result resembles, but deviates from, a general result in the literature on LASSO regularization, see Bühlmann and van de Geer (2011). We note that $||\varepsilon^T (Z*X)||_{\infty} = \max\{1, \ldots, m\ell\} |\varepsilon^T (Z*X)_j|$, and hence, we examine the distribution of $\varepsilon^T (Z*X)_j$. Since $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ we observe that for any column $c = 1, \ldots, m\ell$,

$$\mathbb{E}[\varepsilon^T (Z*X)_j] = \mathbb{E}[\varepsilon^T] (Z*X)_j = 0$$

$$\text{Var}[\varepsilon^T (Z*X)_j] = (Z*X)^T \text{Var}[\varepsilon^T] (Z*X)_j = \sigma^2 (Z*X)^T (Z*X)_j = \sigma^2 ||(Z*X)_j||^2 \leq \sigma^2 \rho,$$

which implies that $rac{\varepsilon^T (Z*X)_j}{\sigma \sqrt{\rho}} \sim \mathcal{N}(0, v)$ for some $v \leq 1$. In other words, $rac{\varepsilon^T (Z*X)_j}{\sigma \sqrt{\rho}}$ is a zero-mean normal random variable that has lower variance than a standard normal random variable $N \sim \mathcal{N}(0, 1)$. This implies that $\mathbb{P}\left(\frac{\varepsilon^T (Z*X)_j}{\sigma \sqrt{\rho}} > u\right) \leq \mathbb{P}(|N| > u)$. Next, using the probability complement, the union bound, and the Chernoff bound for a standard normal random variable, we obtain

$$\mathbb{P}\left(||\varepsilon^T (Z*X)||_{\infty} \leq \sigma \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}\right) = \mathbb{P}\left(\max\{1, \ldots, m\ell\} |\varepsilon^T (Z*X)_j| \leq \sigma \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}\right)$$

$$= 1 - \mathbb{P}\left(\max\{1, \ldots, m\ell\} |\varepsilon^T (Z*X)_j| > \sigma \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}\right)$$

$$\geq 1 - \sum_{c=1}^{m\ell} \mathbb{P}\left(\left|\frac{\varepsilon^T (Z*X)_j}{\sigma \sqrt{\rho}}\right| > \sqrt{2 \log \left(\frac{2m\ell}{\delta}\right)}\right)$$

$$\geq 1 - \sum_{c=1}^{m\ell} \mathbb{P}\left(|N| > \sqrt{2 \log \left(\frac{2m\ell}{\delta}\right)}\right)$$

$$\geq 1 - 2m\ell \exp\left(-\frac{1}{2} \sqrt{2 \log \left(\frac{2m\ell}{\delta}\right)}\right)$$

$$= 1 - \delta. \quad \Box$$

Theorem 1 below presents a probabilistic guarantee on the mean squared forecasting error (11) of a model that was estimated by running the cluster-while-estimate algorithm on a fixed dataset.

**THEOREM 1.** Consider the linear sales generating model (5) and let $\hat{Z}$ and $\hat{\beta}$ be any estimates of $Z^*$ and $\beta^*$ generated by running the cluster-while-estimate algorithm once. Let $\mathbb{I}(\ell > 1)$ be the indicator function which takes the value of 1 if $\ell > 0$ and 0 otherwise. Let $||X_j||^2 \leq \rho$ and $\lambda = 2\sigma \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}$, then the following probabilistic bound holds for any $0 < \delta < 1$,

$$\mathbb{P}\left(\frac{1}{n}||(Z^* X)\beta^* - (\hat{Z} X)\hat{\beta}||^2_2 \leq \frac{2m\rho ||\beta^*||^2_{\infty}}{n} \mathbb{I}(\ell > 1) + \frac{4\sigma ||\beta^*||_{1}}{n} \sqrt{2\rho \log \left(\frac{2m\ell}{\delta}\right)}\right) \geq 1 - \delta.$$
Proof. This proof can be divided into four steps. First, we use the local optimality of our model to bound its squared forecasting error with two important terms: prediction error due to mistakes in clustering products, and prediction error due to coefficient estimation error that is caused by the systematic error $\epsilon$. Second, we bound the first term in the bound, i.e., error of clustering products wrongly. Afterwards, we find a probabilistic bound on the second term, i.e., error caused by $\epsilon$. Finally, we merge these results to finalize the bound.

(Step 1) First, due to the algorithm’s procedure, we know that $\hat{\beta}$ minimizes the regularized loss function when $\hat{Z}$ is fixed. In particular, we have that

$$||y- (\hat{Z} \ast X)\hat{\beta}||_2^2 + \lambda ||\hat{\beta}||_1 \leq ||y- (\hat{Z} \ast X)\beta^*||_2^2 + \lambda ||\beta^*||_1.$$ 

Plugging $y = (Z^\ast X)\beta^* + \epsilon$ into this inequality, we obtain

$$||(Z^\ast X)\beta^* + \epsilon - (\hat{Z} \ast X)\hat{\beta}||_2 \leq ||(Z^\ast X)\beta^* + \epsilon - (\hat{Z} \ast X)\beta^*||_2^2 + 2\epsilon^T (\hat{Z} \ast X) (\hat{\beta} - \beta^*) + \lambda (||\beta^*||_1 - ||\hat{\beta}||_1),$$

which we can rewrite into

$$||(Z^\ast X)\beta^* - (\hat{Z} \ast X)\hat{\beta}||_2 \leq ||(Z^\ast X)\beta^* - (\hat{Z} \ast X)\beta^*||_2^2 + 2\epsilon^T (\hat{Z} \ast X) (\hat{\beta} - \beta^*) + \lambda (||\beta^*||_1 - ||\hat{\beta}||_1).$$

We note that the left-hand side is the quantity we want to bound, and therefore, we only need to further bound the right-hand side.

(Step 2) For the first term on the right-hand side, we use the fact that for a matrix $A$ and vector $x$ that are compatible, $||Ax||_2 \leq ||A||_F ||x||_2$ where $|| \cdot ||_F$ is the Frobenius norm$^1$, to obtain

$$||(Z^\ast X)\beta^* - (\hat{Z} \ast X)\beta^*||_2 \leq ||Z^\ast X - \hat{Z} \ast X||_2^2 ||\beta^*||_2^2.$$ 

Using the triangle inequality and the assumption that $||X_j||_2^2 \leq \rho$, we obtain

$$||Z^\ast X - \hat{Z} \ast X||_2^2 ||\beta^*||_2^2 \leq \Pi(\ell > 1) \left(||Z^\ast X||_2^2 + ||\hat{Z} \ast X||_2^2\right)||\beta^*||_2^2 \leq 2||X||_2^2 ||\beta^*||_2^2 \Pi(\ell > 1)$$

$$= 2 \sum_{j=1}^m ||X_j||_2^2 ||\beta^*||_2^2 \Pi(\ell > 1) \leq 2m\rho ||\beta^*||_2^2 \Pi(\ell > 1).$$

(Step 3) For the second term on the right-hand side, we use Hölder’s inequality to obtain

$$\epsilon^T (\hat{Z} \ast X) (\hat{\beta} - \beta^*) \leq ||\epsilon^T (\hat{Z} \ast X)||_\infty ||\hat{\beta} - \beta^*||_1.$$

Applying Lemma 1 this yields

$$\mathbb{P} \left( ||\epsilon^T (\hat{Z} \ast X)(\hat{\beta} - \beta^*)||_\infty \leq \frac{\lambda}{2} ||\hat{\beta} - \beta^*||_1 \right) \geq 1 - \delta.$$ 

---

$^1$ The Frobenius norm of the matrix $A = (a_{ij}) \in \mathbb{R}^{m \times m}$ is defined as $||A||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^m a_{ij}^2}$.
(Step 4) Merging our results, using the triangle inequality and \( \lambda = 2\sigma\sqrt{2\rho\log\left(\frac{2m}{\delta}\right)} \) we find

\[
1 - \delta \leq \mathbb{P}\left(\|(Z^* \cdot X)\beta^* - (\hat{Z} \cdot X)\hat{\beta}\|^2 \leq 2m\rho\|\beta^*\|_1^2 \mathbb{I}(\ell > 1) + \lambda (\|\beta^*\|_1 - \|\hat{\beta}\|_1)\right)
\leq \mathbb{P}\left(\|(Z^* \cdot X)\beta^* - (\hat{Z} \cdot X)\hat{\beta}\|^2 \leq 2m\rho\|\beta^*\|_1^2 \mathbb{I}(\ell > 1) + \lambda (\|\beta^*\|_1 + \|\hat{\beta}\|_1) + \lambda (\|\beta^*\|_1 - \|\hat{\beta}\|_1)\right)
\leq \mathbb{P}\left(\|(Z^* \cdot X)\beta^* - (\hat{Z} \cdot X)\hat{\beta}\|^2 \leq 2m\rho\|\beta^*\|_1^2 \mathbb{I}(\ell > 1) + 4\sigma\|\beta^*\|_1 \sqrt{2\rho\log\left(\frac{2m\ell}{\delta}\right)}\right). \]

In proving Theorem 1 we analyze the prediction error by considering two types of error: misclassifying observations to clusters and fitting LASSO regression to the clusters. In particular, we note that for a single cluster (\( \ell = 1 \)) the bound simplifies to \( \frac{4\sigma\|\beta^*\|_1}{n} \sqrt{2\rho\log\left(\frac{2m}{\delta}\right)} \). This is a well known estimation bound for LASSO regularized linear regression (Bühlmann and van de Geer 2011), which shows that our probabilistic bound is a tight generalization to the clustered case.

Additionally, we would like the cluster-while-estimate algorithm to yield consistent forecasts, i.e., the forecasts of models estimated on larger datasets converge to the ‘true’ conditional expected sales. This happens when the mean squared forecasting error converges to 0 as the number of products \( n \) increases.

Studying the bound of Theorem 1, we note that the parameters \( \delta, m, \ell, \beta^* \), and \( \sigma \) are independent of \( n \), as they are either selected or fixed by the ‘true’ model. On the other hand, \( \rho \) depends on \( n \) as we expect \( \|X_j\|^2 = \sum_{i=1}^n x_{ij}^2 \) to grow as \( n \) increases. This means that the bound of Theorem 1 only proves consistency if \( \rho \) grows at a rate slower than \( n \), because otherwise the first part of the bound is a constant. Nonetheless, Theorem 2 presents a probabilistic guarantee on the mean squared forecasting error (11) of a model estimated by restarting the cluster-while-estimate algorithm multiple times, and shows that with enough restarts the forecasts are consistent.

**Theorem 2.** Consider the linear sales generating model (5) and let \( \hat{Z} \) and \( \hat{\beta} \) be the estimates of \( Z^* \) and \( \beta^* \) generated by restarting the cluster-while-estimate algorithm multiple times. Let \( \zeta \) denote the probability that \( \hat{Z} \) and \( \hat{\beta} \) equal the estimates generated by solving \((P_{\text{Linear}})\). Let \( \|X_j\|^2 \leq \rho \) and \( \lambda = 2\sigma\sqrt{2\rho\log\left(\frac{2m}{\delta}\right)} \), then the following probabilistic bound holds for any \( 0 < \delta < 1 \),

\[
\mathbb{P}\left(\frac{1}{n}\|(Z^* \cdot X)\beta^* - (\hat{Z} \cdot X)\hat{\beta}\|^2 \leq \frac{8\sigma\|\beta^*\|_1}{n} \sqrt{2\rho\log\left(\frac{2m\ell}{\delta}\right)}\right) \geq (1 - \delta)\zeta.
\]

**Proof.** This proof can be divided into three steps. First, restarting the cluster-while-estimate algorithm yields a \( \zeta \) probability of finding the optimal solution to \((P_{\text{Linear}})\). Conditional on this event we can bound the squared forecasting error in terms of the prediction error that is caused by the systematic error \( \epsilon \). Finally, we rewrite the error such that the result from the previous lemma can be used to prove a high probability bound on the prediction error.
Applying Lemma 1 this yields

\[ \| y - (\hat{Z} \ast X)\beta \|_2^2 + \lambda \| \beta \|_1 \leq \| y - (Z^* \ast X)\beta^* \|_2^2 + \lambda \| \beta^* \|_1. \]

Plugging \( y = (Z^* \ast X)\beta^* + \epsilon \) into this inequality, we obtain

\[ \|(Z^* \ast X)\beta^* + \epsilon - (\hat{Z} \ast X)\beta \|_2^2 \leq \|(Z^* \ast X)\beta^* + \epsilon - (Z^* \ast X)\beta^* \|_2^2 + \lambda (\| \beta^* \|_1 - \| \hat{\beta} \|_1), \]

which we can rewrite into

\[ \|(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta \|_2^2 \leq 2\epsilon^T((\hat{Z} \ast X)\beta - (Z^* \ast X)\beta^*) + \lambda (\| \beta^* \|_1 - \| \hat{\beta} \|_1). \]

To prove the probabilistic bound, we need to analyze the right-hand side of this event.

(Step 2) For the first term on the right-hand side, we add and subtract \((\hat{Z} \ast X)\beta^*\) to the product’s second term, use Hölder’s inequality, and use the triangle inequality to obtain

\[ \epsilon^T((\hat{Z} \ast X)\beta - (Z^* \ast X)\beta^*) = \epsilon^T((\hat{Z} \ast X)\beta - (\hat{Z} \ast X)\beta^* + (\hat{Z} \ast X)\beta^* - (Z^* \ast X)\beta^*) \]

\[ = \epsilon^T((\hat{Z} \ast X)(\beta - \beta^*) + \epsilon^T(\hat{Z} \ast X - Z^* \ast X)\beta^*) \]

\[ \leq |\epsilon^T((\hat{Z} \ast X)(\beta - \beta^*)) + |\epsilon^T((\hat{Z} \ast X - Z^* \ast X)\beta^*)| \]

\[ \leq |\epsilon^T((\hat{Z} \ast X)|\beta - \beta^*|_1 + |\epsilon^T((\hat{Z} \ast X - Z^* \ast X)\beta^*)|_1 + |\epsilon^T((Z^* \ast X)\beta^*)|_1. \]

Applying Lemma 1 this yields

\[ \mathbb{P} \left( \epsilon^T((\hat{Z} \ast X)\beta - (Z^* \ast X)\beta^*) \leq \frac{\lambda}{2} (|\beta - \beta^*|_1 + 2|\beta^*|_1) \right) \geq 1 - \delta. \]

(Step 3) Merging our results, using the triangle inequality and \( \lambda = 2\sigma \sqrt{2\rho \log \left( \frac{2m\ell}{\delta} \right) } \) we find

\[ 1 - \delta \leq \mathbb{P} \left( |(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta|_2^2 \leq \lambda (|\beta - \beta^*|_1 + 2|\beta^*|_1) + \lambda (|\beta^*|_1 - |\hat{\beta}|_1) \right) \]

\[ \leq \mathbb{P} \left( |(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta|_2^2 \leq \lambda (|\beta|_1 + |\beta^*|_1 + 2|\beta^*|_1) + \lambda (|\beta^*|_1 - |\hat{\beta}|_1) \right) \]

\[ = \mathbb{P} \left( |(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta|_2^2 \leq 2\sigma |\beta^*|_1 \sqrt{2\rho \log \left( \frac{2m\ell}{\delta} \right) } \right). \]

This probabilistic guarantee is conditional on the event \( E \), which implies that

\[ \mathbb{P} \left( |(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta|_2^2 \leq 2\sigma |\beta^*|_1 \sqrt{2\rho \log \left( \frac{2m\ell}{\delta} \right) } \right) \]

\[ \geq \mathbb{P} \left( |(Z^* \ast X)\beta^* - (\hat{Z} \ast X)\beta|_2^2 \leq 2\sigma |\beta^*|_1 \sqrt{2\rho \log \left( \frac{2m\ell}{\delta} \right) } \right) \mathbb{P}(E) \geq (1 - \delta)\xi. \]

\[ \square \]
The bound of Theorem 2 shows that the forecasts of the multiple restart cluster-while-estimate algorithm are consistent as long as $\rho$ grows slower than $n^2$. This is satisfied without loss of generality, because the columns $X_j$ can always be rescaled such that $0 \leq x_{ij} \leq 1$ and $||X_j||_2^2 \leq n$, which implies that $\rho$ grows with at most rate $n$. The main reason behind the consistency of the multiple restart cluster-while-estimate algorithm is that it is more likely to find the optimal solution to $(P_{\text{Linear}})$. As the number of products $n$ in the training set grows, the objective of $(P_{\text{Linear}})$ converges to the expected regularized prediction error, which is minimized by the parameters $Z^*$ and $\beta^*$ of the ‘true’ model. Hence, the estimates $\hat{Z}$ and $\hat{\beta}$ generated by $(P_{\text{Linear}})$ or restarting the cluster-while-estimate algorithm will converge to the ‘true’ parameters $Z^*$ and $\beta^*$, leading the forecasting error to converge to 0.

To illustrate and compare these results, Figure 2 presents both bounds as a function of the number of observations $n$. The figure on the left shows the case where the bound holds with 90% probability ($\delta = 0.10$) and the figure on the right for 99% probability ($\delta = 0.01$). The solid curves form the bound of Theorem 1 and the dotted curves form the bound of Theorem 2, while red and blue indicate $\ell = 2$ and $\ell = 10$ clusters respectively. The other parameters are given by $m = 10$, $||\beta^*||_1 = 2$, $\sigma = 1$, and $\rho = n$.

![Figure 2](image_url)  

**Figure 2**  Probabilistic bounds on the mean squared forecasting error as the number of observations ($n$) changes for several numbers of clusters ($\ell$) and probability to exceed the bound ($\delta = 0.10$ on the left, $\delta = 0.01$ on the right)

The main observation of these figures is that each forecasting error bound declines as the number of observations for past products $n$ increases. Initially, both bounds decrease rapidly, which indicates that the algorithm is accurate even for small datasets. We observe the consistency of the bound in Theorem 2. As more and more data becomes available the estimated model converges to the ‘true’ model, and hence, the prediction error converges to 0. On the other hand, when $\rho = n$, the bound of Theorem 1 is not consistent. As we expected, the estimated model is not guaranteed to find the right clustering, and hence, the prediction...
error converges to the bound’s constant first term when $\rho = n$. Additionally, we observe that the prediction error is nearly identical when there are two or more clusters. Altogether, this validates the idea that introducing the need to learn clusters complicates the model, not how many clusters are added. Finally, comparing the two figures, the bounds are similar when the probability with which the bound holds is increased. Each of the bounds on the left ($\delta = 0.10$) increase by at most 20% on the right ($\delta = 0.01$). Additionally, in Figure 3 we show how the bound evolves with the fraction of non-zero ‘true’ parameters out of all parameters $s/(m\ell)$, where $s = ||\beta^*||_1$ is a rough approximation of the number of non-zero true parameters. The setup is similar, on the left $\delta = 0.10$ and on the right $\delta = 0.01$, the solid curve represents Theorem 1 and the dotted curve represents Theorem 2, the red curve for $\ell = 2$ and the blue curve for $\ell = 10$. Additionally, $n = 100$, $m = 10$, $\sigma = 1$, and $\rho = n$ are fixed.

![Figure 3](image)

**Figure 3** Probabilistic bounds on the mean squared forecasting error as the sparsity level ($s/(m\ell)$) changes for several numbers of clusters ($\ell$) and probability to exceed the bound ($\delta = 0.10$ on the left, $\delta = 0.01$ on the right)

The bound behaves quite differently as a function of the sparsity level $s/(m\ell)$. The forecasting error increases as the model becomes less sparse. Possibly, the difficulty in finding the ‘true’ important parameters is causing this. When only a few regression coefficients have a large effect in the true model, the estimation procedure should be able to find these parameters with greater ease than when there are many regression coefficients with small to medium effects. The difference between the two different probabilistic guarantees is large. At a sparsity fraction of 10% and 2 clusters, the bound of Theorem 1 is 3.2 times larger than that of Theorem 2. The former bound is larger due to the first term in its expression. This term features $||\beta^*||_1^2 = s^2$, which means it increases quadratically with $s$. The latter bound only contains $||\beta^*||_1 = s$, which means it increases linearly with $s$. The key difference is that Theorem 2 is able to learn the correct assignment of products to clusters (by restarting the algorithm), while there is no guarantee on this in Theorem 1. Hence,
these results indicate that restarting the cluster-while-estimate algorithm (therefore being able to learn the right clusters) can lead to a large reduction in forecasting error at a small computational cost.

5. Case Study: Johnson & Johnson Consumer Companies Inc.

In this section, we give a detailed description of our collaboration with Johnson & Johnson Consumer Companies Inc., one of the largest consumer goods manufacturers in the world. While we introduced the collaboration in Section 2 briefly, we will give an extensive description of the data collected, segments tested and performance metrics used. Afterwards, we discuss the forecasting tool that was developed for our collaborators.

5.1. Data

As mentioned before, Johnson & Johnson is highly invested in innovation of its product segments and releases new products frequently. As our partner, they provided us with sales and feature data of new products released in the past. In our analysis, we focus on data from product segments where most of the innovation occurred (i.e., the highest number of new product releases). This leads us to the following categories of products: facial care, body care, and baby care products. These products are further cataloged into subcategories. In Table 2 we list the different segments and subsegments that we consider and along with their corresponding time periods. To aid our method in clustering comparable products, we perform our analysis at the subsegment level where products are more homogeneous. For products in each of these subsegments, we have access to monthly sales and feature data for a period of roughly 4 years.

Through interactions with our industry partner, we realized that a product is considered new for the first 12 months of its lifecycle. Additionally, products that last less than four months are characterized as promotional versions of existing products. Thus, we subset our dataset to first remove products with lifecycle of less than four months. We further subset the data to include only the first twelve months of a product’s sales information. Next, we split the dataset into a train set containing the first two years of data, a validation set containing the third year of data, and a test set containing the last year of data. In other words, we create our prediction models using the cluster-while-estimate algorithm on the train set, use the validation set to tune the hyper parameters (regularization parameter $\lambda$ and number of clusters $\ell$) and then test the model on the hold-out test set.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Subsegments</th>
<th>Time Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facial care</td>
<td>Acne FMT, Non-Acne FMT</td>
<td>2012-2016</td>
</tr>
<tr>
<td>Baby care</td>
<td>Lotion, Bath</td>
<td>2013-2017</td>
</tr>
<tr>
<td>Body care</td>
<td>Cleansers, Moisturizers</td>
<td>2013-2017</td>
</tr>
</tbody>
</table>
Features: We have access to specific product features in different subsegments. These features include product features such as average unit price, brand, claimed benefits, form (e.g., lotion, foam, powder, etc.), distribution features such as %ACV (Percentage of All Commodity Volume), number of stores selling the product, and promotion information such as the usage of display promotions and feature promotions. Another interesting fact is that while the provided datasets contain sales data for the manufacturer’s brand, they also contain sales data for competing brands. This means, we can train our model on both the manufacturer’s as well as the competitors’ data and then test models on collaborator specific brand data.

We now describe how we engineer some of the features related to marketing and distribution in order to efficiently use all the available data. While promotion and distribution are important drivers of product sales, these decisions are possibly dependent on customer response. For instance, a well received new product might see a jump in promotion budget and increased distribution amongst different channels. In contrast, firms might reduce spending on promotion and distribution for products that have seen tepid response from consumers after introduction. Hence, estimating monthly promotion and distribution features for new products is almost as hard as estimating sales for the new product itself. Thus, in order to use these features we first created indicator variables that were then used in the prediction model for sales forecasting. While predicting exact distribution or promotion numbers is hard, managers are often able to make a comparison between the budget allocated to the new product and past products. For instance, for products in a particular category, the manager can compare promotion budgets of all products in the category and decide where the new product in question would lie in comparison to all these existing products on an yearly level. We apply a similar technique to the promotion and distribution related data.

Consider the display promotion feature, which describes whether the product in question was promoted using a display promotion in a given month. Given that we have historical feature and sales data, we have full hindsight information on when this product was promoted during the months after introduction. Clearly, monthly prediction of such features is hard. Hence, we will transform this feature into a new feature describing the intensity of display promotion usage over the first year of introduction. For each product we check whether it falls below the 33rd percentile, between the 33rd and 67th percentile, or above the 67th percentile of display promotion usage. Depending on where the product falls, we classify it as Low/Medium/High on the intensity of display promotions used. Thus, we have simplified the task of predicting monthly display based promotion to calculating an yearly promotion intensity indicator. Not only does this transformation let us use feature information that could not have been directly used, it can also provide insights on how display intensity can impact sales of the new product. Furthermore, this is a more stable feature in comparison to monthly features that can change over the course of a product’s introduction and hence might be hard to predict. We use similar transformation for all features related to promotion and distribution.

5.2. Benchmarks and Results
We now describe the different benchmarks that we used to compare our approach against other machine learning approaches. These benchmarks are motivated from the cluster-while-estimate algorithm.
**OLS Regression:** While our cluster-while-estimate algorithm dynamically clusters and regresses sales for different products, another simpler approach is to assume that there exists only one cluster to which all products belong. Thus, the problem simplifies to finding a sales forecasting function, \( f \) that minimizes the loss function \( L \). Assuming that \( f \) is a linear function and \( L \) is the squared loss function, the cluster-while-estimate problem \((P)\) reduces to the well understood OLS regression. Therefore, we consider OLS regression as one of the benchmarks to test our model against.

**Cluster-then-estimate:** Another simple approach that one could consider is a stepwise method. In the first step, we can cluster products using any clustering method such as k-means or hierarchical clustering. After clustering, one can then fit separate demand models to each cluster that minimize the prediction loss in the cluster. Interestingly, both our industry partners used slight modifications of this stepwise approach to make new product sales predictions.

**Other benchmarks:** We also considered other benchmarks including the Bass model and random forests to forecast sales on the above subsegments. Due to their mediocre performance we only list the natural approaches that were used by our industry partners. As a reason, the Bass model is used to make lifecycle predictions, which means it might make significant prediction error in predicting sales during the introductory period, the focus of this paper. Furthermore, while the Bass model has been extended to incorporate the effect of prices in sales forecasting, it is hard to use other product related features. Our cluster-while-estimate method uses all available feature data related to existing products to make accurate sales predictions. Similarly, while our approach is easier to understand for our partners, we also observe that it outperforms other complex machine learning methods.

### 5.3. Performance Metrics

We now describe the out-of-sample performance metrics that were used to compare different benchmarks.

**Mean Absolute Percentage Error (MAPE):** The MAPE measures the relative difference between the actual and predicted sales, which means that a lower MAPE implies better performance. If we assume that there are \( n \) products in our test set and that \( y_i \) denotes the actual sales while \( \hat{y}_i \) denotes the predicted sales for the \( i \)'th product, then the MAPE of a set of predictions is given by:

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|.
\]  

**Weighted Mean Absolute Percentage Error (WMAPE):** This metric is similar to the MAPE except for the different weighting of observations. Instead of weighting each product in the test set equally, the WMAPE weights products based on the magnitude of their sales. This means that products with higher sales have a higher weight. Using the same notation as before, the WMAPE of a prediction method is given by:

\[
WMAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \frac{y_i}{\sum_{i=1}^{n} y_i}.
\]
In Table 3 and Table 4 we compare the MAPE and WMAPE when applied to 6 different subsegments. Not only do we notice that our method yields the best performance, we also see that it beats the other methods by a considerable margin. Furthermore, the WMAPE and MAPE lie below 0.47 regardless of the dataset considered. In turn, this indicates that our method is robust to different data settings. We explore the question of robustness further in Section 6. The two benchmarks that we consider are inspired by the current industry practice of either fitting a single regression model to the sales of products introduced in the past or using a two step approach of sequentially clustering products and fitting demand models. Our results clearly show that a simultaneous approach of jointly clustering and regressing results in considerable improvements in sales estimation.

*Bull’s Eye Metric:* This measure compares the predicted sales with the actual sales and then buckets products based on percent accuracy with respect to the actual sales. Table 5 shows the Bull’s Eye metric for the cluster-while-estimate method on six different datasets. The end points for different buckets have been created based on consultation with our industry partner. The number 7 in the 70-130 bucket for Baby care (Lotion) represents that our predictions were within 70-130% of the actual sales for 7 out of the 10 products in the dataset. Ideally, the higher the number in the middle bucket, the better the prediction method is. Notice that most of the predictions lie within the desired bracket of 70-130%, hence showing the accuracy of our method for each individual product in these subsegments.

### Table 3 MAPE comparison of various benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OLS</th>
<th>Cluster-then-Estimate</th>
<th>Cluster-while-Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baby care (Lotion)</td>
<td>0.919</td>
<td>4.852</td>
<td>0.214</td>
</tr>
<tr>
<td>Baby care (Bath)</td>
<td>0.733</td>
<td>0.672</td>
<td>0.297</td>
</tr>
<tr>
<td>Body care (Moisturizers)</td>
<td>0.455</td>
<td>0.499</td>
<td>0.288</td>
</tr>
<tr>
<td>Body care (Cleansers)</td>
<td>0.651</td>
<td>0.482</td>
<td>0.294</td>
</tr>
<tr>
<td>Facial care (Acne FMT)</td>
<td>1.189</td>
<td>1.103</td>
<td>0.414</td>
</tr>
<tr>
<td>Facial care (Non-Acne FMT)</td>
<td>0.611</td>
<td>0.904</td>
<td>0.414</td>
</tr>
</tbody>
</table>

### Table 4 WMAPE comparison of various benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OLS</th>
<th>Cluster-then-Estimate</th>
<th>Cluster-while-Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baby care (Lotion)</td>
<td>0.825</td>
<td>2.901</td>
<td>0.251</td>
</tr>
<tr>
<td>Baby care (Bath)</td>
<td>0.790</td>
<td>0.698</td>
<td>0.330</td>
</tr>
<tr>
<td>Body care (Moisturizers)</td>
<td>0.485</td>
<td>0.548</td>
<td>0.360</td>
</tr>
<tr>
<td>Body care (Cleansers)</td>
<td>0.642</td>
<td>0.446</td>
<td>0.421</td>
</tr>
<tr>
<td>Facial care (Acne FMT)</td>
<td>0.891</td>
<td>0.873</td>
<td>0.468</td>
</tr>
<tr>
<td>Facial care (Non-Acne FMT)</td>
<td>0.313</td>
<td>0.608</td>
<td>0.217</td>
</tr>
</tbody>
</table>

5.4. Implementation in Practice

Next, we describe the pilot implementation of the cluster-while-estimate algorithm at Johnson & Johnson. Our pilot tool was created with the objective of simplifying the sales forecasting of new products for managers while giving fast, easy to use and reliable sales predictions. In Figure 4, we describe the workflow...
of the pilot tool. We first apply the cluster-while-estimate algorithm on historical sales data of comparable products to create optimal clusters and prediction models. Then, we use Excel to make a user friendly prediction interface which can be used to make final sales predictions. In the first step, the cluster-while-estimate model works with high dimensional feature data and selects important features for sales predictions. Afterwards, we use these features in the Excel tool as input to make predictions. Figure 5 shows a screen-shot of the Excel tool.

![Figure 4 Workflow for live pilot testing](image)

**User Interaction:** In this section we describe user related inputs that are needed for the Excel tool. The user is asked to provide new product feature information such as the brand, packaging size and unit price of the new product. We provide a range of values that all the input product features can take. This serves dual purposes: first, to provide the user an idea of the kind of values that the input can take, and second, to make sure that our predictions stay reliable and we do not extrapolate our linear models. We also ask the user to provide information on product marketing and the promotion budget as the marketing effort is a decision that can have a large impact on the eventual sales of the new product. This includes deciding on display promotions, feature promotions, and display and feature promotions which includes periods when a product is promoted both through display as well as feature advertising. As explained in the previous section, all these are transformed into intensity indicator variables that compare the level of promotion or distribution in comparison to other products within the same brand. The user inputs Low/Medium/High levels for these features comparing the anticipated levels with those of existing products within the same brand. Similar transformation is also done for distribution related features such as %ACV and others. We
next use the feature information and the already generated clusters and prediction models to make monthly predictions. Note that our task here is to make accurate first year predictions. Nevertheless, in order to make the predictions more interpretable, our tool also illustrates the monthly predictions generated from our model (Figure 6).

As a by product of our prediction model, we also create a list of the predictors that are most predictive of sales for the given input feature values of the new product. This is created by evaluating the derivative of the forecasting model (10) presented in Section 4. This list is useful in guiding important decisions such as pricing or packaging of the product before its introduction to the market. We note that the Excel tool created is very simple to use, descriptive and can guide decision making in various operations issues in new product management. While the creation of cluster and regression model is computationally more expensive, given that new product sales data is weekly. Hence, the Excel tool can be very easily updated to incorporate new features and data into the prediction interface.
**Live testing:** We asked our industry partner to use the pilot tool and present some of the feedback as well as results from the live pilot. In Figure 6 we present an illustrative figure created from the Excel tool. Note that since these product features do not exist in the market yet it was impossible to compare our predictions with actual numbers. Hence, our industry partner tried other products from the test set and compared the final prediction numbers in order to see how well the tool and the models were performing.

![Figure 7: Sample output comparing predicted and actual monthly sales](image)

Figure 7 presents results from one such product. Our prediction model performed well on the product with an error of only 1% of overall sales. We also received positive feedback on other features of the tool such as the list of important predictors and the movement of sales with changing promotion or distribution levels. Overall, our industry partners appreciated the ease of use of the tool as well as a simplified approach of incorporating new data generated through product releases in the model. In all, the feedback that we received on the tool was very positive and the industry partner is now trying to incorporate the tool in the decision making and sales estimation process of new products.

### 6. Robust Results: Fashion Retailer

In order to test the robustness of our approach, and test whether the results of this work applies to other retail settings, we also collaborated with a fashion retailer. Our industry partner is one of the world’s largest fashion retailer. As in the previous case, this fashion retailer also invests considerable resources in new product releases. In what follows, we describe the data and results for forecasting sales of new products for the fast fashion retailer.

#### 6.1. Data

The fast fashion retail industry is characterized by fast paced innovation and product with very short life cycle. For instance, Gallien et al. (2015) states that in 2011, Zara released 8000 new fashion products in the span of just one year. In this case, the products sold by our partner also have a very short life cycle (at most 6 weeks). Our industry partner classifies products into segments and subsegments (e.g., a particular subsegment could include all female trousers or all male dress shirts). As with the consumer goods data,
we estimated and tested our model at the subsegment level. For each product we again divided the data into train-validation-test set. As noted in Section 2, we have access data for different subsegments between April to June of 2016. In consultation with our industry partners, we realized that a product in fast fashion retail is considered new for the first half week of its sales and hence we subset the data to only include the first half week of sales for each product.

Features: We were provided with sales and feature data from different new products at the store level. For each product, we had access to product specific features such as color, price, segment, subsegment and location (distribution) specific information such as total capacity allocated, store capacity allocated etc. Interestingly, we also had e-commerce data at the product level. That is, we had access to number of user clicks for a specific product in a specific region, number of times the product was added in the cart, number of times the product was bought online and other online specific features. We used all these features as potential predictors of sales of the new product. We next present results from our estimation method and compare it with other benchmarks using the performance metrics detailed in section 5.

Benchmark and performance metrics: We use the same benchmarks and performance metrics as described in Section 5. While we were not provided with predictions from the currently used forecasting tool, we do note that the partner’s current approach was closest to the “cluster-then-estimate” benchmark.

6.2. Results

In Table 6 and Table 7, we compare the results for the three benchmarks (OLS regression, cluster-then-estimate and cluster-while-estimate) methods on two different subsegments of products. While we only present the WMAPE results, they are presented at both the individual store level as well as aggregated product level (where sales of products are aggregated over stores). Both these prediction numbers are important for the firm and can aid different decision making problems. For instance, the aggregate predictions can guide overall initial production of the product while the individual store level prediction can guide the distribution of the product. In all instances, our method considerably improves over the other benchmarks. Moreover, this improvement is independent of the aggregation level as well as the subsegment chosen. We note that in absolute terms, predictions at the aggregate level are better. This is expected as prediction at aggregated level is easier than individual store level due to higher variance from store to store.

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>OLS</th>
<th>Cluster-then-Estimate</th>
<th>Cluster-while-Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMAPE (Individual)</td>
<td>0.699</td>
<td>1.031</td>
<td>0.553</td>
</tr>
<tr>
<td>WMAPE (Aggregate)</td>
<td>0.463</td>
<td>0.824</td>
<td>0.291</td>
</tr>
</tbody>
</table>
Table 7  WMAPE comparison of various benchmarks at different aggregation levels for subsegment 2

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>OLS</th>
<th>Cluster-then-Estimate</th>
<th>Cluster-while-Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMAPE (Individual)</td>
<td>0.777</td>
<td>1.598</td>
<td>0.660</td>
</tr>
<tr>
<td>WMAPE (Aggregate)</td>
<td>0.587</td>
<td>0.895</td>
<td>0.370</td>
</tr>
</tbody>
</table>

7. Conclusion

In this paper, we propose a new sales forecasting model that can accurately predict sales of new products by efficiently using available data on comparable products. The forecasting model proposed is general as it is able to estimate a variety of standard demand models for unknown clusters of products. First, we show that the sales forecasting problem is computationally hard. Therefore, we propose a scalable algorithm that has both running time as well as analytical performance guarantees. We then use the proposed algorithm to forecast sales of new products for a consumer goods manufacturer and a fashion retailer. We show robust results on real datasets from various segments and subsegments that significantly improve the prediction error over other benchmarks. Finally, we create and test an Excel pilot tool with our consumer goods manufacturing partner, and observe that its accurate, robust, and fast prediction process considerably simplifies the task of forecasting new product sales for practitioners.

References


Bastani, Hamsa, Mohsen Bayati. 2015. Online decision-making with high-dimensional covariates.


Perakis, Georgia, Divya Singhvi. 2017. Less can be more in price experimentation, an optimistic-pessimistic exploration-exploitation framework.