İŞLETME ARAŞTIRMALARI DERGİSİ
JOURNAL OF BUSINESS RESEARCH-TURK
2020, 12(4), 3319-3332
https://doi.org/10.20491/isarder.2020.1043

Regression Clustering for Estimating Product-Level Price Elasticity with Limited Data

Enis KAYIŞ [a]

[a] Ozyegin University, Department of Industrial Engineering, Istanbul, Turkey. enis.kayis@ozyegin.edu.tr

ARTICLE INFO

Keywords:
Regression clustering
Price elasticity
Heuristics

Received 7 September 2020
Revised 15 December 2020
Accepted 20 December 2020

Article Classification:
Research Article

ABSTRACT

Purpose – Pricing is a strategic competitive leverage and firms increasingly utilize data-driven pricing methods. Estimates of product-level price elasticities are needed to determine the best prices for each product, hence reliable estimation is of first-order importance. However, due to the increasing number of products and dynamics of customer behavior, relevant historical data is often limited.

Design/methodology/approach – The objective of this paper is to jointly cluster products with similar price elasticities and estimate this cluster-specific quantity using regression clustering. An extension of the regression clustering problem. Two heuristics are proposed: The gradient descent-based heuristic iterates through feasible solutions to increase cluster-specific regression fit. The categorical ordering heuristic fits a regression for each product, orders the products based on the mean response, and splits them at the largest gap. Using simulated and real-world datasets, a comparative performance analysis is conducted.

Findings – Using the gradient descent-based heuristic with multiple starting solutions gives the best performance. The computational times could decrease with smart initial solutions, which is especially critical if the number of products is large. The categorical ordering heuristic, the fastest method, performs better when there are more than two clusters but suffers from robustness problems.

Discussion – The findings show that offered heuristics are effective to estimate product-specific price elasticity with limited data. Firms could leverage these estimates to increase revenues and profits by better aligning product prices with demand. Given that many products have limited relevant data, the extent of the applications of our method is quite large which, in turn, could help firms stay competitive.

1. Introduction

In today's highly competitive and increasingly transparent markets, product pricing is a critical success factor for firms to attract and retain customers. With the development of the revenue management field and its uptake in the industry, scientific methods to develop the optimal product pricing are gaining momentum as an alternative to manual and simple ruled-based (e.g., cost plus a fixed margin) pricing strategies (Talluri and Van Ryzin, 2006). One of the key inputs to such methods is to quantify how much the product demand would change as the product price is modified. Hence one needs to reliably estimate the price elasticity for each product which measures the percentage change in demand as the price increases by one percent, keeping all the other factors that affect demand constant.

Price elasticity estimation has been studied in many earlier works, albeit mostly at the category level. For example, Andreyeva et al. (2010) reviews earlier studies on the price elasticity of food products and provides a list of mean elasticities for some of the food categories such as soft drinks, milk, eggs, and beef. However, product (e.g., stock keeping unit SKU) specific elasticity is required to calculate the right price for each product in the assortment.

Estimation of product-level price elasticity is not an easy task. According to the Food Marketing Institute, a typical US supermarket carries somewhere between 16000 to 60000 stock keeping units on average (Peer 2019). With an increasing number of SKUs, estimating product-level price elasticity is ever challenging: Price elasticity is dynamic (Fibich et al., 2005) and there is generally limited amount of relevant data to estimate (Bauer and Jannach, 2018). For firms that employ consistent pricing instead of "everyday low pricing" strategy,

Suggested Citation

Kayış, E. (2020). Regression Clustering for Estimating Product-Level Price Elasticity with Limited Data, *Journal of Business Research-Turk*, 12 (4), 3319-3332.

the amount of relevant data is even scarcer: For example, using a supermarket sales data Cohen et al. (2017) states that a particular brand of ground coffee was promoted (i.e. discounted price) in only 8 weeks out of the total 35 weeks during which the price was kept constant. Thus, there is a real need for methods to estimate product-level elasticity in the face of limited historical data. Fitting a different regression equation for each product to estimate price elasticity could simply lead to statistically insignificant results due to this data limitation.

In this paper, our objective is to jointly classify products based on their price elasticity and estimate single price elasticity for all the products in the group. Since more frequent price changes are observed in the collection of products in the group as compared to a single product, our method circumvents the aforementioned data limitation problem by using all the price change information for the products in the group. A modified version of the regression clustering methodology is employed to solve this problem. As the exact solution of this problem takes too much computational time, fast and close-to-optimal algorithms are designed to solve this clustering problem for large assortments.

A limited number of approaches have been proposed in the literature to deal with the aforementioned problem. Bauer and Jannach (2018) employs machine learning techniques to estimate the optimal product price under sparse sales data. They overcome the data availability problem by combining kernel regression results with information on the products within the same subcategory using a Bayesian inference approach. Greenstein-Messica and Rokach (2020) clusters products sold by an e-commerce retailer based on similarity using semantic features and clickstream behavior of customers. To find the optimal prices for an Airbnb listing-night, which is by nature a quite unique product with sparse data, Ye et al. (2018) presents a customized regression model tied to a booking probability model.

Regression clustering, also known as clusterwise regression, aims to form groups of data points that follow the same regression hyperplane given the independent variables. It was Charles (1977) who first introduced the problem in the literature. Since this problem is common across many diverse sets of fields, regression clustering is used in many applications such as customer segmentation (Wedel and Kistemaker, 1989 and Brusco et al., 2003), groundwater remediation system design (He et al., 2008), monthly rainfall prediction (Bagirov et al., 2017), wine classification (Costanigro et al., 2009), and stroke diagnosis (McClelland and Kronmal, 2002). Späth (1979) presents an exchange algorithm to solve for the optimal hard cluster memberships for each data point such that the total sum of squares (SSE) is minimized. DeSarbo (1988) provides a soft clustering technique as an alternative that aims to maximize the log-likelihood function of each data point belonging to a specific cluster.

There are two main decisions one has to solve for in regression clustering problems: the optimal number of clusters and the cluster memberships. This paper develops algorithms for optimal binary partitioning, hence papers that study the latter problem are discussed next. In order to solve for the optimal soft binary partitioning, a nonlinear mixed-integer programming formulation is presented in Lau et al. (1999) which is then solved by an expectation-maximization heuristic. A mixed logical-quadratic programming formulation is proposed in Carbonneau et al. (2011) which is shown to generate numerically stable solutions that are also global optimal using both real and synthetic datasets. As an extension, Carbonneau et al. (2012) offers to solve this combinatorial problem using a repetitive branch and bound algorithm which is found to be much faster than simply using a commercial optimization solver. An incremental algorithm is proposed in Bagirov et al. (2017) based on a non-smooth nonconvex formulation. The authors analyze the effect of the quality of the starting solutions and argue that the method is effective even with large but not too sparse datasets with a limited number of outliers. Kayış (2020) presents a gradient descent-based heuristic to solve a modified version of the regression clustering problem for small datasets. Joki et al. (2020) introduces a support vector machine-based formulation coupled with the L_1 norm for the objective function that performs well even with datasets that include outliers. The performance of several metaheuristics (e.g., column generation, genetic algorithm, Späth's algorithm) are proposed and compared in Park et al. (2017) using synthetic and real-world datasets.

In this paper, regression clustering is applied to estimate product-level price elasticity based on cluster memberships. Our approach differs from the existing literature in regression clustering as each data point from the same predefined subgroup (e.g., same SKU) has to belong to the same cluster. For example, a data point representing weekly price and corresponding sales for a specific product can be a member of a cluster if

all data points belonging to the same product are in the same cluster as well. The aim of this paper is to find cluster membership of each data point subject to the aforementioned constraint in order to minimize the total sum of squares resulting from fitting a regression equation using all the data points in each cluster. Each regression equation will use demand as the dependent variable and other independent variables, such as price, that may affect demand. Hence one could use the cluster-specific regression equation to estimate price elasticity for all the products within the same cluster.

In this paper, the optimal binary partitioning problem for product-level price elasticity is investigated. In other words, the products are grouped into high and low price elasticity clusters. This assumption is without loss of generality: Creating a higher number of clusters is possible via successive application of our proposed algorithm into the resulting clusters after each binary split. However, finding even the optimal binary split is not easy when the number of predefined subgroups is large. Assuming the number of predefined subgroups is L, the number of possible nonempty binary partitions is $2^{L-1}-1$. For example, when there are more than 20 subgroups, the number of possible binary partitions would become more than one million making full enumeration a computationally prohibitive alternative. Hence this paper proposes two simple heuristics to address this problem.

The first heuristic borrows the idea of gradient descent search developed for continuous search spaces and modifies it for integer search spaces. The algorithm starts with an initial solution and iterates through other feasible solutions until there is no improvement. Three variants of this algorithm are proposed that differ by the initial solution used. The second heuristic is a faster alternative to the gradient-descent based heuristics. This algorithm simply fits a separate regression equation for each subgroup and orders them by the mean response. Through synthetic datasets the performance of each heuristic is studied, and a real-world dataset is used to apply the proposed algorithms to estimate product-level price elasticity.

The rest of this paper is organized as follows. The formulation of the problem and the heuristics are introduced in Section 2. The computational performances of proposed heuristics are evaluated using simulated datasets in Section 3. The same section also presents the results of applying the presented algorithms to estimate product-level price elasticities using a real-world dataset. Section 4 presents a discussion of this work and the potential implications on product pricing. Finally, Section 5 concludes the paper with a summary of this study and potential opportunities for future research.

2. Problem Formulation and Methodology

In this section, the formal definition and the mathematical formulation of the problem are presented. As the formulation gets increasingly difficult to solve, several heuristics are developed to generate good solutions to our problem.

The objective is to cluster a group of products using a single categorical variable $s \in \mathbb{R}$ with L unique values (levels). These levels could be different stock keeping units (SKUs), product categories, or any other predefined product groupings. Let $q \in \mathbb{R}$ be the sales quantity and $\boldsymbol{x} \in \mathbb{R}^p$ represent the vector of independent variables, including but not limited to the product price, used to explain variability in the sales quantity. (In this paper, boldface is used to denote vectors or matrices and $\boldsymbol{X}'$ denotes the transpose of matrix $\boldsymbol{X}$.) The linear regression relationship between q and $\boldsymbol{x}$ is assumed to be different for each value of s. For simplicity of exposition, it is assumed that there is a single splitting variable. However, the proposed method(s) can be generalized to settings with multiple splitting variables by either forming factors through the combination of original factors or searching for optimal partition variable-wise.

Let $(\boldsymbol{x}'_{jt}, q_{jt}, s_{jt})$ denote the observations for product j in period t. In this notation, $s_{jt} \in \{1,2,\dots,L\}$ denotes which level does product j belongs to in period t. Note that product j's level is allowed to change across time, if needed. The partitioned regression model can now be written as follows:

$$q_{jt} = \sum_{m=1}^{M} \boldsymbol{x}'_{jt}\beta_m w_m(s_{jt}) + \varepsilon_{jt}$$

where $w_m(s_{jt}) \in \{0,1\}$ indicates whether the jtth observation belongs to the mth cluster or not. Since each observation has to belong to a cluster, it is required that $\sum_{m=1}^{M} w_m(s_{jt}) = 1$ for any $s_{jt} \in \{1,2,\dots,L\}$.

In this study, only binary partitions are considered, i.e., $M = 2$. However, multiple partitions could be extended straightforwardly. For example, using a greedy approach similar to the classification and regression trees (CART) method to generate decision trees, one can apply the methods presented in this paper consecutively to the resulting partitions to form new two subgroups until a termination condition is met. The result would be a multiple partitioning of the whole dataset.

To simplify our notation in binary partitioning, assume that $w_{jt} = w_1(s_{jt}) \in \{0,1\}$. Clearly, $w_1(.)$ is a mapping from $\{1,2,\dots,L\}$ to $\{0,1\}$. Moreover, atomic weights for each level are denoted by $\{\tau_1, \tau_2, \dots, \tau_L\} \in \{0,1\}^L$, where $\tau_l = 1$ ($\tau_l = 0$) implies that the l[th] level is in cluster 2 (1). Thus, one can write the observation-level weights as $w_{jt} = \sum_{l=1}^{L} \tau_l I_{(s_{jt}=l)}$.

Let the level association vector $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_L)'$ and $\boldsymbol{W}$ be the diagonal matrix with entries equal to w_{jt}. The sales quantities of all the products across available time periods are stored in vector $\boldsymbol{q}$ and $\boldsymbol{X}$ denotes the matrix for independent variables. Using the ordinary least squares regression theory, one could then write the total sum of squared error (SSE) of the two groups as follows:

$$Q(\boldsymbol{\tau}) \coloneqq SSE_1 + SSE_2 \coloneqq \|\boldsymbol{q} - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{q}\|^2 + \|\boldsymbol{q} - \boldsymbol{X}(\boldsymbol{X}'(\boldsymbol{I}-\boldsymbol{W})\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{I}-\boldsymbol{W})\boldsymbol{q}\|^2 \coloneqq Q_1(\tau) + Q_2(\tau). \quad (1)$$

Our objective is to minimize $Q(\boldsymbol{\tau})$ over the feasible set of $\boldsymbol{\tau}$'s.

The optimization problem stated above is combinatorial in nature: With L levels of the splitting variable s, there are $2^{L-1} - 1$ possible solutions. The stated problem is also a special case of the general regression clustering problem explained in Section 1. In the original formulation, each data point could belong to any cluster. However, the formulation studied in this paper requires each data point with the same level to be a member of the same partition. Since exact solutions to the original problem do not exist, our problem is also quite challenging to solve.

2.1 Heuristics

Our problem has a combinatorial nature, thus an exact solution to the problem is not computationally feasible. An exhaustive search that considers all possible groupings of the levels into two disjoint clusters would provide an exact solution. However, even with a moderate number of levels, this is computationally too expensive: For example, a dataset with 25 products requires a search of 16,777,215 possible partitions, which is computationally infeasible. Hence heuristics are developed to generate good solutions to the problem. This paper considers two simple heuristics and their variants: gradient descent-based and categorical ordering.

The gradient descent-based algorithm implements the gradient descent idea on an integer space. This algorithm initially L levels are randomly assigned into two nonempty and non-overlapping clusters, then all the levels are cycled through and each level's cluster membership is flipped sequentially. The resulting L cluster assignments are evaluated in terms of the SSE criterion $Q(\boldsymbol{\tau})$ as defined in (1). Then the clustering that minimizes $Q(\boldsymbol{\tau})$ is chosen as the current assignment and iteration continues until the algorithm converges. Let's define two partitions as adjacent if they differ only by one level. Then our algorithm can be considered as a variant of the classical gradient descent on the space of possible partitions. A local optimum is guaranteed to be reached after the gradient descent algorithm terminates. The complexity of our algorithm is polynomial in the number of levels, assuming that the SSE is locally convex near the initial solution.

Three variants of the gradient descent-based algorithm are studied in this paper. The first variant runs the algorithm multiple times, each time using a different random starting partition. The best solution out of these multiple runs is hoped to be a global optimum for the original problem. As the starting points are selected randomly, there is some uncertainty about what the converged solution will be. The second version, defined as "All-In-One Initial," assigns all the levels into the same cluster initially and iterates through until convergence. The last variant, defined as "Smart Initial," first fits a regression equation to each level independently. Then the estimated price elasticity coefficients are ranked, ignoring the statistical significance of the resulting estimate, and the levels are split into two groups from the largest gap in the ordered coefficients. Using this as the initial solution, the gradient descent-based algorithm is run till termination. Unlike the first variant, there is no randomness about what the final solution will be using the last two variants.

The second heuristic, designed to alleviate the computational burden, is to order the categories in a way similar to CART. In a piecewise constant model like CART, Hastie et al. (2009) reviews the idea of ordering the

categories by the mean response in each category and then treating the categorical variable as if it were an ordinal variable. Thus, the computational complexity is reduced from exponential to linear. The simplification was justified by Fisher (1958) in an optimal splitting setup and is exact for a continuous response regression problem. In the regression clustering context, let $\hat{\beta}_l$ represent the least squares estimate of β given the observations in the lth level. Now, lth level's fitted regression line would be $\boldsymbol{x}'\hat{\beta}_l$. A strict ordering of the $\boldsymbol{x}'\hat{\beta}_l$'s as functions of $\boldsymbol{x}$ may not exist, thus an approximate solution is suggested. Our method uses $\bar{\boldsymbol{x}}'\hat{\beta}_l$ to rank the L levels where $\bar{\boldsymbol{x}}$ is the mean of $\boldsymbol{x}$, and thus could handle the categorical variable as ordinal. This approximation performs well when there is a clear separation in the fitted models, however a locally optimal partitioning is not guaranteed.

3. Computational Results

To measure the quality of the solutions generated via the heuristics, a computational study is performed using simulated and real-world datasets. Through simulated datasets, the performance of our algorithms is studied as the number of levels (e.g., products), the number of data points, the magnitude of the variability in the residuals and the underlying regression equation vary. Since the optimal partitioning in some of the simulated datasets is known by construction, one could determine if our heuristics could generate solutions that are close to the optimal. Then the proposed heuristics are applied on a real-world sales dataset to estimate product-level price elasticity and the resulting product groups and the estimated price elasticities in each group are discussed.

3.1 Results with the Simulated Datasets

In this computational study, the performances of the three variants of the gradient descent-based heuristic as well as the categorical ordering heuristic are compared. The first variant of the gradient descent-based heuristic is run 5 times, each time with a different random starting solution, and the best solution out of these 5 replications is recorded. Since there is no randomness in the other two variants, they are run only once. The comparison between the three variants of the gradient descent-based heuristics is used to understand the effect of using starting solutions with different qualities. Moreover, the quality of the solutions generated by the categorical ordering heuristic, which is expected to be a faster alternative to the gradient descent-based heuristic, is evaluated using the generated datasets.

As a benchmark, the random search method is used to search through $\min(4000, 2^{L-1}-1)$ randomly generated unique partitions and return the partition with the smallest total SSE as defined in (1). This method is replicated 5 times and the best solution out of these 5 replications is recorded. Notice that when the number of levels is less than or equal to 12, the possible number of unique partitions is less than 2047. Hence the benchmark gives the optimal partitioning when the number of levels is low. However, the performance of the benchmark is expected to deteriorate exponentially fast especially when the number of levels is more than 12.

Simulated datasets are created under the following generation rules: It is assumed that there is a single independent variable, price p, that affects the sales quantity. Three different parameter settings are considered: In the first two parameter settings, a linear demand model is used where the optimal partition is binary. In the first setting, the underlying regression equation is assumed to take the form $q = 1000 - 8 * p + \varepsilon$ for the first partition and $q = 500 - p + \varepsilon$ for the second one. Even-numbered levels are assigned to the first partition and odd-numbered levels are assigned to the second partition. The second setting is very similar to the first one. The only difference is that only the first two levels are in the first partition and the rest of the levels belong to the second partition. A comparison of the results obtained under these two settings will help us understand the performance of our methods under equal and unequal number of levels in the optimal partitions. Multiple splits could be optimal in many applications as well which is considered in the third setting: There are 8 partitions and the underlying regression equation in each partition is assumed to be $q = 1000 - (s\ mod\ 8) * 700 - ((s\ mod\ 8) + 1) * p + \varepsilon$. Clearly, the optimal number of partitions is 8 in this last setting. However, since only binary partitions are considered, the optimal binary partition depends on the individual dataset and is not known in advance. Across all these three settings, price p is randomly generated uniformly from the interval [500,1000] and regression error ε is generated from a normal distribution with mean zero and standard deviation σ.

In total 528 datasets are created. In each dataset, one of the following parameters is varied: The number of levels changed from 8 to 48 in increments of 4. Given the number of levels, the number of data points is selected from the set $\{15L, 30L, 60L, 90L\}$. The standard deviation of the residuals varied from 100 to 400 in increments of 100. The result is a full factorial design with 11*4*4*3=528 different datasets. All computations are carried out on a machine with Intel® Core™ i7-8565U CPU @ 1.80 GHz processor and 16 GB RAM.

The quality of the results obtained from each method is measured using the following formula:

$$PR = \frac{SSE_0 - SSE_f}{SSE_0}$$

where SSE_0 is the total SSE assuming all the levels are in the same group and SSE_f is the total SSE in the two partitions arrived using the method at hand. Thus, the percentage reduction in the SSE (PR) is used to evaluate the methods. Table 1 provides the summary statistics of the 5 methods across all the generated datasets. The average PR is highest with the first variant of the gradient descent-based heuristic. Using a smart initial solution decreases the performance only slightly. The average performance of the category ordering heuristic is very similar, but there are a few datasets where the performance is quite low (see the minimum and the 5th percentile of the PR reported for this method). Using all-in-one initial solution deteriorates the performance of the gradient-descent-based heuristic to an extent. Finally, the random search method has the lowest average PR of the five methods, as expected. Even though this method could find the optimal partitioning when $L \leq 12$, the performance worsens exponentially fast as L increases. Also, note that the random search method is used only when $L \leq 28$ due to computational time limits. Since the PR is known with increasing L, a direct comparison is not appropriate. In the rest of this section, how design factors affect the performances of the methods is addressed.

Table 1. Descriptive statistics about the percentage SSE reductions of the five methods across 528 datasets

	Descent Search	**Descent Search (Smart Initial)**	**Descent Search (All-In-One Initial)**	**Categorical Ordering**	**Random Search**
Average	71.09%	70.57%	67.81%	70.19%	59.64%
Min	16.34%	16.34%	16.34%	5.71%	9.37%
5th Perc.	32.34%	32.39%	32.39%	28.70%	19.00%
25th Perc.	63.42%	62.64%	58.99%	63.05%	47.85%
Median	74.79%	74.63%	72.79%	74.76%	62.50%
75th Perc.	82.15%	82.29%	76.18%	82.29%	75.07%
95th Perc.	96.41%	96.43%	95.27%	96.43%	94.09%
Max	97.15%	97.15%	97.15%	97.15%	97.15%

Figure 1 presents the average PR for the 5 methods as the number of levels varies. As the random search method could lead to much lower PRs, the secondary vertical axis is used to display the average PR for this method. Notice that the average PRs are the same when $L \leq 12$ with the gradient descent-based and random search methods. The performance of random search decreases rapidly, and the average PR reduces to 39% when $L = 28$. The decrease in the performance is also observed for our heuristics, albeit at a much smaller rate. Of the four heuristics we have proposed, the first variant of the gradient descent-based heuristic performs the best across almost all the different number of levels. Using the smart initial solution does not decrease the performance much, and even has a better performance when $L = 48$, which may suggest that initial solution quality is critical especially when the number of levels is high. The performance of the categorical ordering heuristic could be significantly worse than the former two heuristics for some instances. Running the gradient descent-based heuristic with the all-in-one initial solution results in the worst performance of the four heuristics; however, the gap in the performance decreases as one increases the number of levels. Also notice that when $L\ (mod\ 8) = 0$, due to our generation rules of the simulated datasets under the third regression equation setting (i.e., the one with multiple partitions), there is an equal number of levels in each underlying partition. For these cases, starting with the all-in-one initial solution increases the likelihood that the heuristics stuck in a local optimum that is far away from the global one. However, when there are an unequal number

of levels in each partition, there are high quality solutions which are close to all-in-one initial solution, hence the performance is better when $L\ (mod\ 8) \neq 0$.

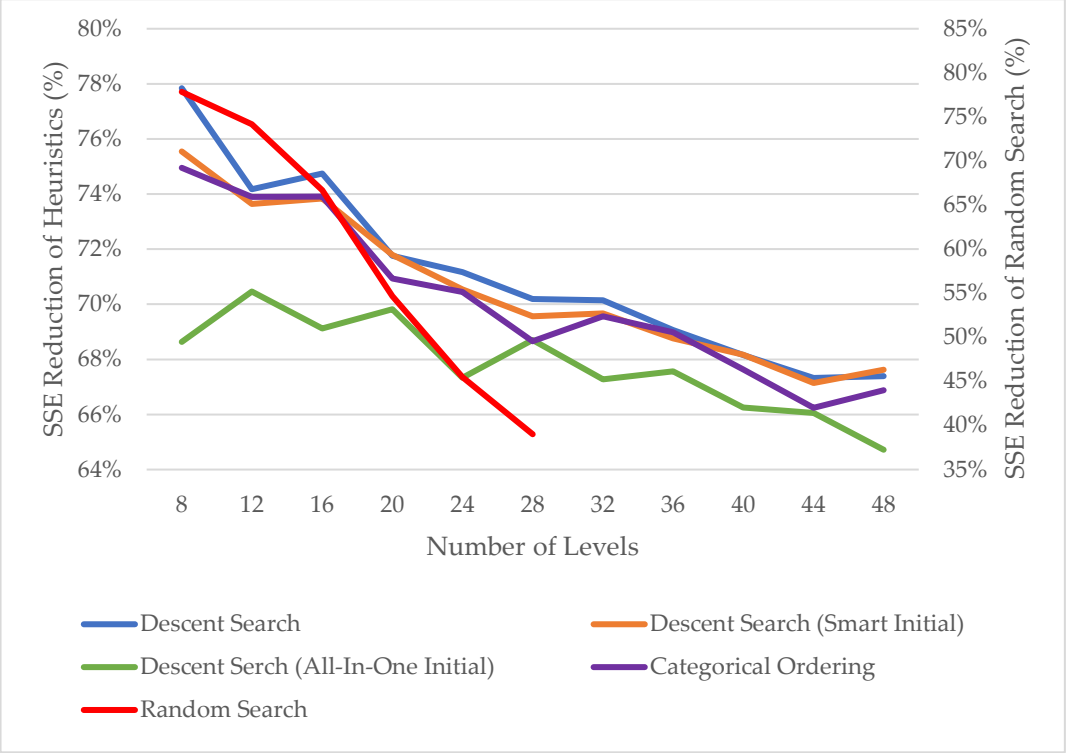


Figure 1. The average percentage SSE reductions of the five methods (Numbers for the Random Search method is displayed on the secondary y-axis.)

By our construction of the simulated datasets, the optimal binary partitioning is known a priori under the first two regression equation settings. For these two settings, 352 datasets in total, the maximum number of misclassified levels are computed given the number of levels for each method. The results are presented in Table 2. The performances of the gradient descent-based heuristics are quite good in terms of misclassification, explaining their comparatively better performance especially for high number of levels (i.e., $L > 12$). Starting with a smart initial solution never misclassifies a level. The comparatively lower performance of this variant of the heuristic compared to the first variant as depicted in Figure 1 is due to lower performance for the cases with multiple partitions that are not included in the analysis behind Table 2. Moreover, it is interesting that in some cases categorical ordering heuristic misclassifies more levels than the simple random search. This observation supports the fact that the worst-case PR of this heuristic is lower than that of the random search, further questioning the robustness of the categorical ordering method.

Table 2. The maximum number of misclassified levels across the five methods

L	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering	Random Search
8	0	0	1	3	0
12	0	0	1	1	0
16	0	0	1	1	1
20	1	0	1	3	2
24	1	0	1	3	4
28	0	0	1	7	6
32	0	0	1	4	
36	0	0	1	4	
40	1	0	1	9	
44	1	0	1	5	
48	1	0	1	5	

The average PR of each method under three different regression settings is presented in Table 3. In line with our earlier discussions, the first and the smart initial variants of the gradient descent-based heuristic performs the best, though the performance of the latter is lower under the third regression setting. Starting with all-in-one initial solution has the best performance out of the methods under the second regression setting where the number of levels in each partition is different (i.e., 2 versus L). In this setting, most of the levels belong to the same partition, hence the optimal partition could be reached in two iterations if one starts the gradient descent method with the all-in-one solution. On the other hand, the categorical ordering performs especially well under the third setting (i.e. multiple partitions). Finally, the random search method performs comparatively worst under the first two settings with binary partitions. Given the fact that this method is only computed for datasets with $L \leq 28$ and the PR reduces with L, the gap over all the 528 datasets would be larger.

Table 3. The average percentage SSE reductions of the five methods under different regression settings

Regression Setting	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering	Random Search
1	80.26%	80.47%	74.32%	78.40%	65.82%
2	58.52%	58.83%	58.83%	57.57%	46.45%
3	74.48%	72.41%	70.28%	74.59%	66.65%
All	**71.09%**	**70.57%**	**67.81%**	**70.19%**	**59.64%**

Regarding the remaining two factors, the number of data points does not have a significant impact on the average PR, except the categorical ordering heuristic (see Table 4) which requires a sufficient number of data points for better performance. The average PR as the standard deviation of the regression error is shown in Table 5. As expected, the performance of all the heuristics decreases with increased standard deviation. The decrease is largest with the categorical ordering heuristic, which shows that this method is very dependent on a strong relationship between price and demand.

Table 4. The average percentage SSE reductions of the five methods across different number of data points

Number of Data Points	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering	Random Search
15*L	71.17%	70.78%	68.89%	67.62%	60.20%
30*L	71.16%	70.77%	67.38%	71.05%	59.58%
60*L	70.92%	70.71%	67.64%	70.99%	59.59%
90*L	71.10%	70.02%	67.35%	71.10%	59.19%

Table 5. The average percentage SSE reductions of the five methods across varying standard deviation of the residual error

σ	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering	Random Search
100	85.98%	85.93%	82.62%	86.40%	68.59%
200	75.75%	75.21%	70.86%	75.83%	63.25%
300	65.52%	64.62%	62.64%	64.07%	55.91%
400	57.10%	56.52%	55.14%	54.46%	50.82%

Given the performances of the five methods regarding the solution quality, the next question is the computational times required to run each of these methods. Table 6 presents the average computation time for each method, as a function of the number of levels. The categorical ordering heuristic is the fastest heuristic, owing to the fact that it does not require any iteration. All the variants of the gradient descent-based heuristic slow down as the number of levels increases, as a higher number of iterations are required for convergence. However, starting with a smart initial solution requires a comparatively smaller computational time especially when the number of levels is high. Notice the high computational times required to run the random search

method. When $L \geq 16$, the upper limit of 4000 searches is exceeded, hence this method only computes 4000 solutions independent of L. Hence the solution time is steady around half a minute for $L \geq 16$.

Table 6. The average computational times (seconds) of the five methods

L	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering	Random Search
8	0.22	0.03	0.04	0.02	0.17
12	0.68	0.09	0.12	0.04	4.13
16	0.97	0.10	0.17	0.04	31.59
20	1.39	0.13	0.23	0.05	28.80
24	1.97	0.15	0.32	0.05	29.39
28	2.70	0.22	0.42	0.06	28.68
32	3.60	0.23	0.57	0.08	
36	4.65	0.32	0.72	0.08	
40	5.82	0.29	0.91	0.09	
44	7.24	0.43	1.11	0.11	
48	8.70	0.47	1.30	0.12	
All	**3.45**	**0.22**	**0.54**	**0.07**	**20.46**

3.3 Results with a Real-World Dataset

To understand the performances of the proposed heuristics in a real-world dataset, a retail sales dataset from a small firm offering several products is used. The dataset includes the weekly sales as well as average weekly sales prices across 156 weeks for 11 products which are coded SKU A through SKU K in this study. SKU E is removed from the study due to incomplete data. The data collection period is from 25 September 2016 through 15 September 2019.

To capture the dynamic nature of product-level price elasticity, the dataset is divided into 13 equal sized subsets, each covering 12 consecutive sales periods (i.e., a three-month span). The underlying regression equation which is used to fit for each cluster is given below:

$$\log(q_{jt}) = \beta_0 + \beta_1^j t + \beta_2 \log(p_{jt}) + \varepsilon_{jt}$$

This demand model, also known as the log-log model, is used frequently in the literature to estimate price elasticity (see, for example, Cohen et al. 2017). Any product-specific trend is captured in our model with the β_1^j coefficients. The coefficient β_2 measures the product-price elasticity for the all the products in the particular cluster. The adjusted R^2 values of the regression fits in the final clusters are found to be above 90%, hence other independent variables, such as competitor prices, are not incorporated into the regression in line with other works (Cohen et al. 2017).

Table 7 presents the percentage SSE reduction after applying our heuristics across the thirteen subsets. In each row of the table, the best performance is highlighted in boldface. In all but one subset, the first variant of the gradient-descent heuristic offers the highest percentage SSE reduction. The difference between the other variants is significant for subsets numbered 1, 2, and 12, but quite small or does not exist for the rest of the subsets. The categorical ordering heuristic could lead to significantly poor performances in almost half of the subsets. This result is in line with the observations regarding the poor robustness results of this heuristic.

Table 7. The percentage SSE reductions of the four methods across 13 subsets

Subset	Descent Search	Descent Search (Smart Initial)	Descent Search (All-In-One Initial)	Categorical Ordering
1	**9.16%**	6.70%	6.70%	6.28%
2	12.91%	**15.79%**	12.91%	12.40%
3	**15.68%**	**15.68%**	**15.68%**	15.63%
4	**1.13%**	1.03%	1.03%	0.44%
5	**10.78%**	**10.78%**	**10.78%**	6.47%
6	**6.68%**	**6.68%**	**6.68%**	6.03%
7	**18.42%**	**18.42%**	**18.42%**	18.42%
8	**3.11%**	**3.11%**	**3.11%**	2.56%
9	**3.92%**	**3.92%**	3.69%	3.17%
10	**18.82%**	6.34%	**18.82%**	6.34%
11	**20.70%**	**20.70%**	**20.70%**	**20.70%**
12	**15.07%**	12.94%	12.94%	12.94%
13	**18.23%**	**18.23%**	**18.23%**	**18.23%**

Given that the best performing method is the first variant of the gradient descent-based heuristic, product-specific price elasticities are estimated for each cluster across the 13 subsets using this variant of the heuristic. The first part of Table 8 provides the estimated elasticities, and the associated statistical significance levels, for the products in each cluster as well as cluster memberships. The products in the first group have higher price elasticities as compared to the products in the second group. Also notice that the estimated price elasticities for the first group are always statistically significant, whereas the estimated price elasticities for the second group are not significant at 1% level in six subsets. Some products do not change cluster membership across the 13 subsets (e.g. SKU B), whereas the opposite is true for some other products (e.g., SKU J).

What would happen if, instead of clustering products into two groups, all the products are combined in a single group to estimate the price elasticity? The last column of Table 8 lists the estimated category-specific price elasticity assuming all the products are in the same category and thus have the same elasticity. Notice two problems with this approach. First, it may lead to underestimation and overestimation of price elasticity for some products. Second, statistically significant category-specific price elasticities are not available in the last two subsets. Hence price optimization using price elasticity estimates would lead to suboptimal pricing or is not possible due to the unavailability of elasticity estimates.

Table 8. The price elasticity of the two groups as well as the group membership of the SKUs across 13 subsets (p-values for the price elasticity coefficient are displayed inside the parenthesis where *** (**) implies a p-value that is less than 0.001 (0.01))

Subset	Group 1		Group 2		All Products
	Price Elasticity	SKU List	Price Elasticity	SKU List	Price Elasticity
1	-3.59 (***)	A,B,C,G	-1.52 (***)	D,F,H,I,J,K	-1.71 (***)
2	-2.89 (***)	A,B,C,D,G,H,I,J	Not Sig.	F,K	-1.18 (***)
3	-4.01 (***)	B,I	-1.55 (***)	A,C,D,F,G,H,J,K	-1.97 (***)
4	-1.55 (***)	A,B,G,I,J,K	Not Sig.	C,D,F,H	-1.48 (***)
5	-2.88 (***)	B,G,H,K	-1.09 (**)	A,C,D,F,I,J	-1.99 (***)
6	-3.38 (***)	B,D,F,G,J	-1.37 (**)	A,C,H,I,K	-1.78 (***)
7	-1.81 (***)	A,B,C,D,F,J,K	Not Sig.	G,H,I	-1.69 (***)
8	-2.13 (***)	A,B,F,G,H,I,J,K	Not Sig.	C,D	-2.04 (***)
9	-1.8 (***)	A,B,F,G,K	Not Sig.	C,D,H,I,J	-1.62 (***)
10	-6.56 (***)	B,C,G,J	-0.92 (***)	A,D,F,H,I,K	-1.38 (***)
11	-4.36 (***)	B	-0.98 (***)	A,C,D,F,G,H,I,J,K	-1.36 (***)
12	-1.36 (***)	B,C,D,F,I,K	Not Sig.	A,G,H,J	Not Sig.
13	-1.58 (**)	A,B,C,D,F,G,H,I,K	Not Sig.	J	Not Sig.

Another alternative to clustering products is to estimate product-specific price elasticity using only that product's sales data. Table 9 displays the number of products in each dataset that would have statistically insignificant price elasticity, along with the same figure using our regression clustering method. In total, our approach is able to estimate statistically significant product-specific price elasticities for additional 67 product-subset pairs that single product analysis could not provide. Thus, our method provides necessary input in order to calculate the optimal price for these additional 67 cases, which represents 51% of total cases.

Table 9. The number of products with statistically insignificant price elasticity estimates (assuming a p-value of 0.01) using different methods across 13 subsets

Subset	Regression Clustering Method	Single Product Analysis
1	0	4
2	2	7
3	0	6
4	4	8
5	0	7
6	0	8
7	3	4
8	2	6
9	5	7
10	0	7
11	0	7
12	4	9
13	1	8
Total	**21**	**88**

4. Discussion

Revenue management is increasingly leveraged by firms to increase revenues and use available resources more effectively (see, for example, Klein et al. 2020). It is an effective strategy especially for firms with limited product availability and price-sensitive demand. Firms in the service sector adopted this method first, followed by firms in the retail sector. For example, Yazgan et al. (2019) utilizes demand prediction and mathematical optimization to find optimal ticket prices under capacity constraints for a given route of an airline. On the retail side, Cohen et al. (2017) employs revenue management to design promotional pricing strategies for an FMCG company.

Product price optimization is one of the instruments revenue management uses. Given the competitive pressures of today's marketplaces, setting the right price is critical to stay competitive. If the product price is set too high, then the firm would face low demand. If it is set too low, then the product margin would be very thin. The best tradeoff requires a complete understanding of how demand is affected by prices, namely the price elasticity. This paper provides a novel method to estimate product-level price elasticity under limited data.

In order to obtain a reliable estimate of the product-level price elasticity, one needs to observe sufficient price changes in the historical sales data for a given product. For example, Yazgan et al. (2019) uses 165 weeks of sales data for demand estimation for a fight leg of an airline, which is known to use dynamic pricing. However, price changes infrequently for many products of firms serving in other sectors. In the retail sector, for example, Bonomo et al. (2020) states that on a regular day only 0.5% of the products are repriced. Moreover, dynamic customer behaviors render data older than a couple of months useless to estimate up-to-date price elasticity. Using only a single product's sales data would lead to statistically insignificant product-specific price elasticity estimates due to infrequent price changes. In our analysis, the proposed method is able to estimate statistically significant price elasticities for an additional 51% of the cases as compared to single product analysis, which is quite substantial. Another alternative on the opposite side of the spectrum is to estimate category-specific price elasticity using all the data for all the products in the predefined category. Our results also highlight the fact that this approach may lead to underestimation and overestimation of price elasticity, which in turn would

lead to suboptimal pricing. Finally, one could rely on managerial insights on setting prices, which may be biased and is far from perfect. Our results show that firms could increase revenues by better aligning product prices with the up-to-date product demand, even in cases with limited relevant sales data by jointly finding the set of product clusters with similar price elasticities and estimating the unique price elasticity for the products in the cluster.

Given the limited data due to infrequent price changes and the huge number of SKUs, generating product-specific elasticity for each product is challenging. Our method helps to estimate product-specific price elasticities, which in turn could be used to set optimal prices, for these kinds of products. The result is increased profitability without any cost increase, which is the real promise of revenue management. Cohen et al. (2017) estimates that using sales data of a grocery retailer, revenue management could improve the profits of the retailer by 5%. This is only true for the products with sufficient relevant historical data. Our method, which could be used to estimate price elasticity even under limited historical data, could extent this profit gain to almost all the products in the portfolio.

For many sectors, product demand is dynamic and affected by changes in competitor prices, seasonal factors, and customer behaviors. The techniques presented in this paper could be used to update price-elasticity estimates more frequently. Our results show that there could be significant changes in price elasticity estimates for the same product across different time periods (see Table 8). Hence the advantage of using updated price elasticity estimates in setting optimal prices will take into consideration the dynamic and seasonal demand patterns.

The methodology presented in this paper could be incorporated into decision support systems for product pricing. Our clustering algorithm could generate price elasticities for thousands of products since minimal manual intervention is necessary only to filter out inconsistent results. Even with a larger number of products, the gradient descent-based heuristic with smart initial solution provides a solution to help determine the optimal product prices within a reasonable time, while compromising little from solution quality. Hence automatic product price changes for the full product portfolio are possible.

Automatic and frequent price changes could prove to be problematic from technical and customer relationship management points of view. On the technical side, one has to keep aware of system nervousness, as changing prices too frequently would increase the possibility of making operational mistakes like wrong price tags displayed on the product or the IT systems. Moreover, each price change has operational costs required to execute this decision. Hence these associated costs have to be considered while making price change decisions. Bonomo et al. (2020) considers economies of scale in changing prices of multiple products at the same time and conducts a general equilibrium analysis of a monopolistic firm. On the customer side, Yıldırım and Mert (2019) argues that customers may deem some pricing policies unethical and flag them as excessive pricing or discounted pricing. Thus, price changes have to be controlled and any price adjustment has to be made while considering these constraints and intangible costs.

5. Conclusion

The focus of this paper is to devise methods to estimate product-level price elasticities for products with limited relevant historical sales data. Since fitting separate regression equations for each product to estimate price elasticity could lead to statistically insignificant results as relevant data is limited, products are intended to be grouped into two clusters based on the magnitude of the resulting price elasticity. Since this is a challenging problem due to its combinatorial nature, especially when the number of products is large, this paper designs two heuristics to solve this problem. The gradient descent-based heuristic is an iterative algorithm that travels along with the search space and the category ordering heuristic fits a regression equation for each product and simply orders the products based on the mean response and splits them at the largest gap. Using synthetic datasets, the gradient descent-based heuristic is found to offer better performance at the expense of a longer computational time. Using smart initial solutions is found to be quite effective especially when the number of products is large and the solution time is a hard constraint for implementation. The categorical ordering heuristic suffers from robustness problems but could be effective in datasets with multiple underlying clusters. Next, real-world sales data is used to apply the devised methods to estimate product-level price elasticity. Although the performance difference between the two methods decreased, the gradient descent-based heuristic still yields better performance.

Future research could investigate several topics to deepen our understanding of this problem. First, an exact solution to this problem could be developed, possibly with the use of mathematical programming and optimization techniques. When the number of products is very large, exact solution methodologies could take a very long time, hence meta-heuristics could be developed to generate close-to-optimal solutions.

In this work, products are clustered into two groups. One could successively use the binary clustering approach to the resulting clusters of our method to generate multi-cluster solutions. Another extension of our work is to study the optimal number of product clusters. One could tune the number of clusters parameter of our method using a validation set, yet excessive computing times needed with exact solutions may render this option impossible. Finally, our heuristics could be evaluated using other real-world datasets, possibly with a higher number of products, to determine the extent of our conclusions.

References

Andreyeva, T., Long, M. W. and Brownell, K. D. (2010). The impact of food prices on consumption: a systematic review of research on the price elasticity of demand for food, *American Journal of Public Health*, 100(2), 216–222.

Bagirov, A. M., Mahmood, A. and Barton, A. (2017). Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach, *Atmospheric Research*, 188, 20–29.

Bauer, J. and Jannach, D. (2018). Optimal pricing in e-commerce based on sparse and noisy data, *Decision Support Systems*, 106, 53-63.

Bonomo, M., Carvalho, C., Kryvtsov, O., Ribon, S. and Rigato, R. (2020). Multi-product pricing: Theory and evidence from large retailers in Israel, *https://www.bankofcanada.ca/wp-content/uploads/2020/04/swp2020-12.pdf* (Last Accessed: 15 December 2020).

Brusco, M. J., Cradit, J. D. and Tashchian, A. (2003). Multicriterion clusterwise regression for joint segmentation settings: An application to customer value, *Journal of Marketing Research*, 40(2), 225–234.

Carbonneau, R. A., Caporossi, G. and Hansen, P. (2011). Globally optimal clusterwise regression by mixed logical-quadratic programming, *European Journal of Operational Research*, 212(1), 213–222.

Carbonneau, R. A., Caporossi, G. and Hansen, P. (2012). Extensions to the repetitive branch and bound algorithm for globally optimal clusterwise regression, *Computers & Operations Research*, 39(11), 2748–2762.

Charles, C. (1977). R´egression typologique et reconnaissance des forms, PhD thesis.

Cohen, M. C., Leung, N. H. Z., Panchamgam, K., Perakis, G. and Smith, A. (2017). The impact of linear optimization on promotion planning, *Operations Research*, 65(2), 446-468.

Costanigro, M., Mittelhammer, R. C. and McCluskey, J. J. (2009). Estimating class-specific parametric models under class uncertainty: Local polynomial regression clustering in an hedonic analysis of wine markets, *Journal of Applied Econometrics*, 24(7), 1117–1135.

DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression, *Journal of Classification*, 5(2), 249–282.

Fibich, G., Gavious, A. and Lowengart, O. (2005). The dynamics of price elasticity of demand in the presence of reference price effects, *Journal of the Academy of Marketing Science*, 33(1), 66-78.

Fisher, W. D. (1958). On grouping for maximum homogeneity, *Journal of the American statistical Association*, 53(284), 789-798.

Greenstein-Messica, A. and Rokach, L. (2020). Machine learning and operation research based method for promotion optimization of products with no price elasticity history, *Electronic Commerce Research and Applications*, 40, 100914.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements Of Statistical Learning: Data Mining, Inference, And Prediction*, New York, Springer Science & Business Media.

He, L., Huang, G. and Lu, H. (2008). Health-risk based groundwater remediation system optimization through clusterwise linear regression, *Environmental science & technology*, 42(24), 9237–9243.

Joki, K., Bagirov, A. M., Karmitsa, N., Mäkelä, M. M. and Taheri, S. (2020). Clusterwise support vector linear regression, *European Journal of Operational Research*, 287(1), 19-35.

Kayış, E. (2020). A gradient descent based heuristic for solving regression clustering problems, in *Proceedings of the 9th International Conference on Data Science, Technology and Applications, online, 7-9 July 2020*, Portugal, SciTePress, 102–108.

Klein, R., Koch, S., Steinhardt, C. and Strauss, A. K. (2020). A review of revenue management: recent generalizations and advances in industry applications. *European Journal of Operational Research*, 284(2), 397-412.

Lau, K.-N., Leung, P.-l. and Tse, K.-k. (1999). A mathematical programming approach to clusterwise regression model and its extensions, *European Journal of Operational Research*, 116(3), 640–652.

McClelland, R. L. and Kronmal, R. (2002). Regression based variable clustering for data reduction, *Statistics in Medicine*, 21(6), 921–941.

Park, Y. W. , Jiang, Y. , Klabjan, D. and Williams, L. (2017). Algorithms for generalized clusterwise linear regression, *INFORMS Journal on Computing*, 29 (2), 301–317.

Peer, D. (2019). What does SKU mean in the grocery business?, *https://smallbusiness.chron.com/sku-mean-grocery-business-75577.html* (Last Accessed: 15 December 2020).

Späth, H. (1979). Algorithm 39 clusterwise linear regression, *Computing*, 22(4), 367-373.

Talluri, K. T. and Van Ryzin, G. J. (2006). *The Theory and Practice Of Revenue Management*, New York, Springer Science & Business Media.

Wedel, M., and Kistemaker, C. (1989). Consumer benefit segmentation using clusterwise linear regression, *International Journal of Research in Marketing*, 6 (1), 45-59.

Yazgan, H.R., Candan, G., Ataman, M. (2019). Talep tahmini ve dinamik fiyatlandırma ile havayolu bilet fiyatlarının belirlenmesi, *İşletme Araştırmaları Dergisi*, 11(2), 732-742.

Ye, P., Qian, J., Chen, J., Wu, C. H., Zhou, Y., De Mars, S. and Zhang, L. (2018). Customized regression model for Airbnb dynamic pricing, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in London, United Kingdom, 19-23 August 2018, New York, Association for Computing Machinery, 932-940.

Yıldırım, E., Mert, K. (2019). Etik dışı fiyatlandırma uygulamaları karşısında tüketicilerin düşünce ve davranışlarının incelenmesine yönelik bir araştırma, *İşletme Araştırmaları Dergisi*, 11(4), 2876-2892.