Report on Pricing Optimization

By: Daniel Byun, Ricky Chen, Jayson Faulds, May Shao

1. Introduction

Pernalonga, a successful marketing chain operating over four hundred stores across the region of Lunitunia, is looking to implement promotions to offer customers. Pernalonga has two main avenues to induce sales, which include shelf price and promoted price. To preface, product-level shelf prices are increased to drive contribution and lowered to drive sales. These promotions are intended to be adjusted for list (shelf) prices to improve revenues. Going into a particular week of April of this year, we are tasked with developing a pricing scheme for numerous Pernalonga products. More specifically, we must recommend a pricing strategy that will maximize store revenues while still maintaining profits. Additionally, there are a set of constraints that we must abide by. These constraints include 100 products that are to be priced, belonging to 2 product categories, to be offered at 10 stores. Thus, we must find the combination of products, categories, and stores that optimize total revenue based on the new prices we set. In this report that follows, we walk through our methodology and assumptions for generating these new prices, and subsequently selecting the optimal combination of said prices. We will report the 100 products that are to be targeted along with their new prices. Lastly, we will report the anticipated changes in sales, sales quantity, and profits for all 10 stores that are selected.

2. Methodology

Here we will give a brief overview of our strategy for tackling the problem laid out in the previous section. The first major step is to generate the new prices for certain Pernalonga products. To generate these prices, we created a price response function for each product-store combination. In this case, a price response function is a model that represents total quantity purchased as a function of list price, pocket price, and a host of added factors (Like information on substitutes and complements, more on that later). A short-form example of a price response function is shown below, where D refers to demand for a product-store combination, and addedFactors include an assortment of features such as substitute/complement information:

$$D_{ps} = \alpha + \beta_p * pocketPrice_{ps} + \sum_i \gamma_{pi} * addedFactors_{psi}$$

Because the price response function models total quantity purchased, taking the product of the model output and the pocket price will yield the total revenue generated from this product-store combination. Holding the added factors constant, we can find the pocket price that maximizes total revenue expected for this product using calculus concepts. Going through this process, then, should leave us with a set of product-store combinations, each with an optimized price level and an expected revenue.

The second major step involves selecting which of these product-store combinations to utilize. If you recall from the intro section, we are constrained to 100 products of 2 overarching product categories, and are only limited to 10 stores to introduce our optimized prices. We must select the combination of products-categories-stores that will maximize the total revenue that we can achieve while adhering to these constraints. To do this, we will run through each possible combination of product, store, and category, in an effort to identify the optimal combination that maximizes revenue.

3. Data Preparation

3.1 Dimension Reduction

To effectively solve this problem, it is paramount that we understand Pernalonga's product offerings and their fiscal performance. After various efforts to clean up the data for proper analysis, we were left with over 10,000 unique products, coming from 429 product categories, being sold at 421 different stores. One of the first hurdles is handling the sheer size and dimensionality of this data. Making a singular price response function for each product-store combination at this moment would be unfeasible. To alleviate this, our focus will be on Pernalonga's stores and product categories that account for sizable portions of its sales revenue.

Fig. 1 Accumulated Sales Share (Stores)

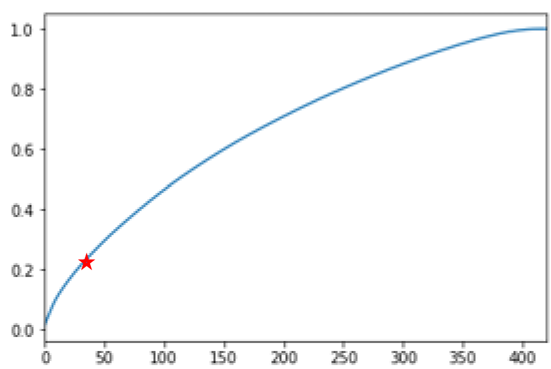
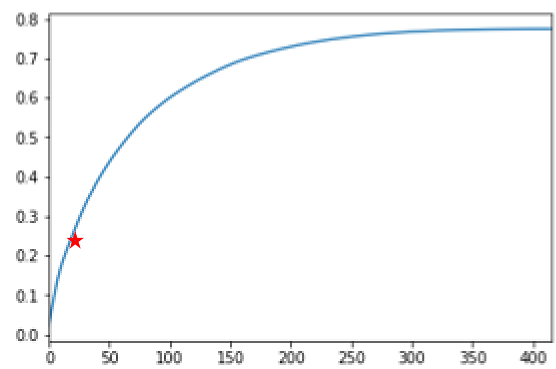


Fig. 2 Accumulated Sales Share (Categories)



In the figures above, the red star represents the point that represents the 25% mark for sales share

As can be seen in **Fig. 1** above, Pernalonga's top 40 stores account for 25% of total revenues. Identically, we repeat this process in **Fig. 2** and find that the top 20 product categories account for 25% of total revenues. These are small samples of the total populations of stores and categories, and yet represent significant value in terms of total revenue. To ease the computational power necessary to perform our revenue optimization algorithm, we will be limiting our focus to these 20 product categories and 40 stores which account for a sizable portion of Pernalonga's total sales revenue.

We must also mention an extra step of handling that went into the product category reduction process. We did not simply take the top 20 product categories, because Pernalonga also stated the restriction that fresh products cannot be targeted for this price optimization campaign. For this reason, we selected the top 20 non-fresh product categories to target. After retrieving the top 20 categories, taking into account historical sales data in the 04/13/2017-04/19/2017 period, we were left with a total of 1,712 unique products in our data set.

3.2 Feature Generation

After completing the filtering process above, it was necessary to obtain the various features to insert into our model, to allow for proper prediction of demand for each product-store combination. To do this, the first necessary step was to generate the dependent variable itself: the demand. In this case, we define demand as the quantity of the product purchased at that store in each month, so that we could take seasonality into consideration. Because the data provided is at the transactional level, we had to aggregate by price, product, store, and month, summing up the total quantity purchased in the process.

Additionally, we had to generate the two significant independent variables, which are related to the direct pricing of the product. We have two pricing components that we will be incorporating into the model, and these include the list price as well as the pocket price, where the pocket price is the list price before taking into account any promotion being offered by the store. While list price is already provided in the data, we had to generate the pocket price by running calculations on the list price and promotions utilized in each transaction.

Lastly, we need to generate features that represent the pricing effects of substitute and complement products on the demand for the targeted product. For the substitute pricing, it is far too labor-intensive to manually identify a list of substitutes for each product that we are considering targeting. Instead, we make the assumption that the other products that exist within a product's category are substitutes for that product. As an example, if we take "Cereal" as a product category, and Fruit Loops is the targeted product, then the other cereals within this product category are considered substitutes. These other cereals can include Lucky Charms, Frosted Flakes, etc. Because each product can have a different number of substitutes, we take the median of the substitute prices to represent the price most substitute products hover around.

The problem with this is that there are some substitutes that cost noticeably less than this median value mentioned above. To continue with the cereal example above, these low-cost cereals could potentially be the private-label brands, or simply brands that aren't exceptionally popular. These are products that represent the lowest-priced substitutes, and its possible a lot of customers will not purchase our target product because this is the cheapest option available. As a result, in addition to the median price of substitutes, we also include the minimum price of a substitute to capture this "cheapest option" effect.

We define complementary products as products that are often purchased together. We sifted through two years of transaction records for the 1,712 products in order to find the most frequently purchased product combinations. After getting the list of each product's complements, our next step is to calculate the average price of those complements in each store in each month, which is similar to our previous process for finding the list price and pocket price for our 1,712 products. And with this information regarding complementary products, we have all of the data and factors desired to build our price response models. An excerpt of the data is shown as follows:

prod_id	category_id	store_id	list_price	pocket_price	median_sub	min_sub	complement_price	original_demand
156953010	95856	157	2.69	2.69	1.79	0.4	1.05	1.0
156953010	95856	308	2.69	2.69	2.69	0.4	1.01	1.0
156953010	95856	335	2.69	2.69	1.79	0.4	1.05	1.0
156953010	95856	341	2.69	2.69	1.69	0.4	1.05	1.0
156953010	95856	343	2.69	2.69	2.69	0.4	1.05	1.0

4. Modeling

With the final data set complete, we enter the modeling phase. Here, we are constructing a singular price response model for each product-store combination. There will be 53,663 combinations of stores and products that have historical data in 2016 and 2017. By personalizing each product by its store as well, we are able to focus on the shift in demand that would be caused by a shift in the product's price without worrying about the effect of the store.

For each product-store combination, we built two response functions. The first, by using linear regression with the pure, untransformed data. The second model was created after applying a log transformation to the demand and price features (assumption of constant-elasticity). The forms of the response functions are shown as below:

$$D_{ps} = \alpha + \beta_p * L_{ps} + \varepsilon_p * P_{ps} + \sum_i \sigma_{psi} * S_{psi} + \lambda_p * K_{ps} + \sum_i \gamma_{psi} * season_{psi}$$

$$\log D_{ps} = \alpha + \beta_p * \log L_{ps} + \varepsilon_p * \log P_{ps} + \sum_i \sigma_{psi} * \log S_{psi} + \lambda_p * \log K_{ps} + \sum_i \gamma_{psi} * season_{psi}$$

Where

D is the historical demand for product p in store s

L is the historical list price for product p in store s

P is the historical pocket price for product p in store s

S is the historical price for product p's substitutes in store s

K is the historical pocket price for product p's complementary goods in store s

Season is the seasonal factor, which is 12 months

We use RMSE (root mean square error) as the criterion to choose which response function will be used for each combination of product and store. For each combination, we choose the model with lower RMSE. Among 53,663 combinations, 36,771 use linear regression without log transformation, and the other 16,892 use linear regression with log transformation. Afterwards, we will use the response function to represent demand with pocket price of each product in each store in that week in April, and our next step is to generate the best price for each combination of product and store in order to maximize revenue.

5. Optimization

In this part, we focus on finding the optimal price of each product to reach the maximum revenue, as well as find the top 100 products that can bring us the maximum revenue with 10 stores.

As mentioned above, we will be selecting from a sample of 20 categories across 40 stores. These 20 categories have 1,712 products, so we need to consider 68,480 (1,712×40) product-store combinations when optimizing the price for each product. After looking closely into our data, we only found 53,663 combination in the 2-year transaction. Products that are not sold in specific stores for 2 years in the month of April are no longer considered as our potential pricing target, so we choose to focus on the 53,663 combinations and optimize the product price in order to reach maximum revenue.

Before we optimize the price, we need to assemble the info necessary to generate predictions using our price response models that we created. What we need is a data frame containing the following information for the 53,663 combinations:

Columns	Function
Product ID	Identifier for products
Store ID	Identifier for stores
List Price	Used for calculating the demand & the promotion rate
Pocket Price	Used for calculating the demand, the promotion rate & the original revenue
Substitute Info	Used for calculating the demand
Complement Info	Used for calculating the demand
Original Demand	Used for comparison after optimization & calculating the original revenue and profit

After retrieving the above information from the data and using the response functions from the modeling section, we can optimize the price by looking into the relationship between revenue and price. Since we use RMSE as our model selection criteria, the two forms of response function might be used on different combinations. Also, in order to simplify the model, the only variable considered for optimization is the list price. The pocket price should be expressed in the format of list price together with promotion rate, which we regard as a constant. All the other prices, including substitute price and complement price, are also regarded as constant.

For the response function without the log transformation:

$$revenue = D_{ps} * P_{ps} = \left(\mu + \sigma * L_{ps}\right) * L_{ps} * (1 - promotion\_rate) = \sigma' * L_{ps}^2 + \mu' * L_{ps}$$

For the response function with the log transformation:

$$\begin{aligned} \log(revenue) &= \log(D_{ps} * P_{ps}) = \log D_{ps} + \log\left(L_{ps} * (1 - promotion\_rate)\right) \\ &= \mu + \sigma * \log L_{ps} + \log L_{ps} + \log(1 - promotion_{rate}) = (\sigma + 1) * \log L_{ps} + \mu' \end{aligned}$$

Where

D is the demand for product p in store s

L is the list price for product p in store s

P is the pocket price for product p in store s

μ, σ, μ’, σ’ are all constants

After we find the response functions revealing the relationship between revenue and list price, we can use derivatives to help us find the optimal list price that can generate best revenue. For the response function without the log transformation, the optimal price depends on the coefficients of this polynomial. Besides the mathematical restrictions, we also set an upper limit and a lower limit from business understanding to get a reasonable optimal price. We use 125% of the original list price as the upper limit and Pernalonga’s cost for the product as the lower limit in order to make profits. After considering all the constraints, our optimal price can only be from the following three values: the lower limit, the upper limit and the mathematical optimal, $-\frac{\mu'}{2\sigma'}$. We compare the revenue of the three possible values, and find the optimal price.

For the response function with the log transformation, we notice it is a monotone function and the optimal revenue is reached when the variable is at the limit. As a result, we only need to examine the upper limit and the lower limit to find the optimal price.

After getting these two prices, we use the optimal price to calculate the demand with the response function above. One thing worth mentioning is that we also set limits on demand according to historical data. For those products not bought during Apr 13th, 2017 – Apr 19th, 2017, we set an upper limit of 10 since there were no demand before. For those products in the transaction history during the week, we set an upper limit of twice as much as they were sold during the week in 2017, avoiding possible unreasonable increase in sales.

Finally, with the new optimal price and the new demand, we can calculate the revenue and profit after optimizing the price and compare the results with the historical data.

We are now left with a set of product-store combinations, where each combination contains a product, product category, store, new price level, and expected sales revenue for that product. If you recall the original constraints established by Pernalonga, we can only implement 100 of these products at 2 categories at 10 stores. We still have a dimension that is larger than these constraints; hence, we will have to select the combination of products-categories-stores that maximizes our expected revenue.

To do this, we implement an algorithm that will run through each store and find the optimal combination of 100 products for each set of two product category pairs. This process is more intuitively laid out in the diagram below:

Store	Category 1	Category 2	Product ID List	Revenue
157	1	2	1, 3, 14, 75, …	700
157	1	3	2, 4, 14, 22, …	300
158	1	2	1, 3, 12, 57, …	200
158	1	3	2, 14, …	500

With the different combinations laid out in this manner, we can then loop over each possible combination of two categories by store. We already have the top 100 products for each combination as listed above. Now, all that is necessary is to combine the 100 products with the optimal store and category combinations. Again, here is a diagram to show the transformation that occurred:

Store	Category 1	Category 2	Product ID List	Revenue
157	1	2	1, 3, 14, 75, …	700
157	1	3	2, 4, 14, 22, …	300
158	1	2	1, 3, 12, 57, …	200
158	1	3	2, 14, …	500

Store List	Product ID List	Category List	Revenue
157, 158, …	1, 3, 14, …	95798, 95798	1000
342, 346, …	72, 45, 97, …	90210, 95425	800

With the last table seen in the diagram above, all we need to do is find the highest value in the Revenue column, and this row would contain the optimized list of 10 stores, 100 products, and two product categories.

6. Results

Following the optimization process outlined above, we are left with a combination of stores, products, and product categories that maximizes expected revenue while still remaining within the constraints set forth by Pernalonga. Interestingly, the two product categories that were optimal included Fine Wines as well as Washing Machine Detergent. Both of these products are fairly expensive when looking at the full repertoire of a grocery store, and both having fairly wide pricing options because of that.

By taking the total expected revenue from our optimal product-category combinations, and summing the results up for each of the ten stores, we get an expected profit of approximately $4,330, with an expected revenue of approximately $19,000 for the promotional period. Dividing this profit figure across the ten stores, and subsequently dividing the profit by day, we get around $60 profit per store per day for these items during the promotional period. This is met with a 3,800 increase in units of fine wine and detergent products sold during the week across the ten stores.

7. Conclusion

To summarize, the objective of this project is to adjust the price of 100 products across two categories within 10 stores, in order to achieve the maximum revenue possible. We start with basic data cleaning such as creating new transaction ID as identifier for each transaction, which we use for defining our substitutes and complements later.

Then we try to reduce the dimensions of data by considering fewer stores and products before picking the required 10 stores and 2 categories. We select 40 stores and 20 categories that are associated with about 25% of the total sales for Pernalonga. This left us with a total of 1,712 products.

When building our models, we use historical sales data to fit the relationship between demand and price into a linear regression model. We also use logged transformation to test different types of linear regression model to see if we could get improved performance. By comparing the RMSE on the two models, we can choose the model which performs better on the specific product-store combination.

The models indicate the formula which uses price info to calculate demand. Since revenue equals demand times price, we can get the exact relationship between revenue and price. After that, we use derivatives to find the best price that can generate most revenue. When we finished finding the optimal price for each product-store combination, we can calculate the updated demand, as well as the updated revenue and profit, so we can compare the results with the original profit and check the revenue increase.

Finally, we used an optimization algorithm to select the best combination of store, category, and products, which ultimately led to our results. To conclude, we will add a brief commentary on the modeling and its effect on the results. We do think the profit estimates are higher than they should be, and that is because of the assumption we made during the modeling process. By holding everything constant aside from the pocket price for the target product, we are assuming that things like substitute and complement prices will not change as a result, something that in reality would be much more fluid. This may be contributing to overestimates for profit and revenues for the upcoming promotional period.

With this being stated, even if we remain on the side of conservatism and expect profits and revenue figures far below what is reported, we can still expect at least a noticeable increase in profits for such a small sample of products (100) to optimize.