Price Optimization

Brendan Kitts
Kevin Hetherington-Young

Vignette Corp
19 Newcrossing Road,
Reading, MA. 01867. USA. Email: bj@datasage.com
Web: http://www.cs.brandeis.edu/~brendy

Abstract

Price is one of the most important tools at the disposal of retailers. Empirical studies by Lambin (1976) have found that price elasticity is about 20 times higher than advertising elasticity. We introduce a price optimization system which combines a variety of variables which are available to a vanilla POS system. We show that cross-elasticities increase prediction accuracy from 5% of variance to 55%. We validate our model using 40,000 items at a real retailer. We also find a vector of price suggestions which could be implemented to increase profit by several hundred percent. The current study differs from previous work in size, accuracy, and the use of scan-level POS data alone, making this study easily deployable by retailers using data which has already been collected.

INTRODUCTION

Retailers have a limited range of options to influence shopper behaviour. Retailers can influence behaviour through four general mediums: (a) price, (b) advertising (eg. weekly newspaper, tv, radio and banner advertising direct mail, and store displays), (c) in-store location (including shelf position, page-hierarchy location, on-page location) and (d) assortment

Of these, price has traditionally been observed to be one of the most influential. Classic work by Lambin concluded that price elasticity is 20 times higher than advertising elasticity (Lambin, 1976). Price changes can also be performed with little preparation, and with immediate effects. These advantages contrast particularly with advertising which requires resources to implement. (Simon, 1989).

This paper focuses on how to set price to optimize total store profit. We introduce a price optimization system which uses cross-prices to optimize total store profit. Our solution is scalable, and we validate our model using 40,000 items at a real retailer.

Problems with price optimization

Selecting prices for 40,000 items in a store is a difficult proposition. The Professional Assignments Group (Pag, 2000) reports that the current practice indicates that retailers might be over-discounting products, with as many as 25%-30% of items being sold at some price discount. With profit margins so low already (1.5% according to the Food Manufacturer's Institute REF), many retail stores may not be able to sustain aggressive price discounting[1].

Pricing is also made difficult by the fact that products interact with each other. Decreasing the price of one juice item to increase traffic, may merely result in the cannibalization of a more profitable juice brand, (as consumers switch from one brand to the other), without increasing demand (if that was the intention). Similarly, raising prices may have pronounced consequences across category boundaries, such as decreasing the number of items bought in distant categories due to a general depression in store traffic.

Retailers have long been aware of these price-demand interactions, and have developed various strategies for coping with these effects. One common pricing strategy is the use of "loss leaders". Loss leaders are products which are kept at greatly discounted prices, because they are known to be high-profile, common, and easily comparable between retailers. Typical loss leaders include milk, bread, eggs, and juice.

Loss leaders are presently determined by retailer experience. However, analysis of transaction data should be able to reveal which products are true "loss leaders" and which are not. This can prevent unnecessary discounting. Similarly, not all items need to be kept at discounted prices, and in some cases it makes sense to raise prices.

As a result, a comprehensive approach to pricing - incorporating knowledge of product interactions, consumer demand, and store-wide effects - needs to be developed.

Previous work on price optimization

Classic work by x has examined the optimization of price for various products. However, typically this work does not involve interactions between products.

TEXT GOES HERE paras on prev work & descriptions

Contribution

Our work differs from previous work in the literature in that we attempt to develop a large-scale, and comprehensive approach to price optimization that includes product interactions. We develop a model that naturally incorporates cross-elasticities of products in any category of the store by examining price-demand timeseries correlations, thereby capturing interactions across category boundaries. We also develop demand models for 40,000 products in the store, and provide this to a global optimization code to optimize price. The model is constructed from POS data supplied by a live retail chain.

[1] Aggressive price discounting may have been to blame in part for many recent bankruptcies and downsizes, including Vons (CA, laid off 250 executives in 1993), Bradlees (MA, Chapter 11 bankrupt in 1998) and Caldor (MA, bankrupt in 1999).

To validate our system, we ran a large-scale price experiment involving 8 stores, with results collected over 120 days. The results of the experiment were an increase of profit at the item level of 18% unadjusted, or 29% seasonally adjusted, with category profit increases of around 7%.

The outline of this paper is as follows. In section 1 we describe the demand model. In section 2, we describe methods for optimizing price using analytical derivatives from the demand model. In section 3, we optimize prices in a retail chain, and examine the results.

THE DIFFICULTY OF FORECASTING DEMAND

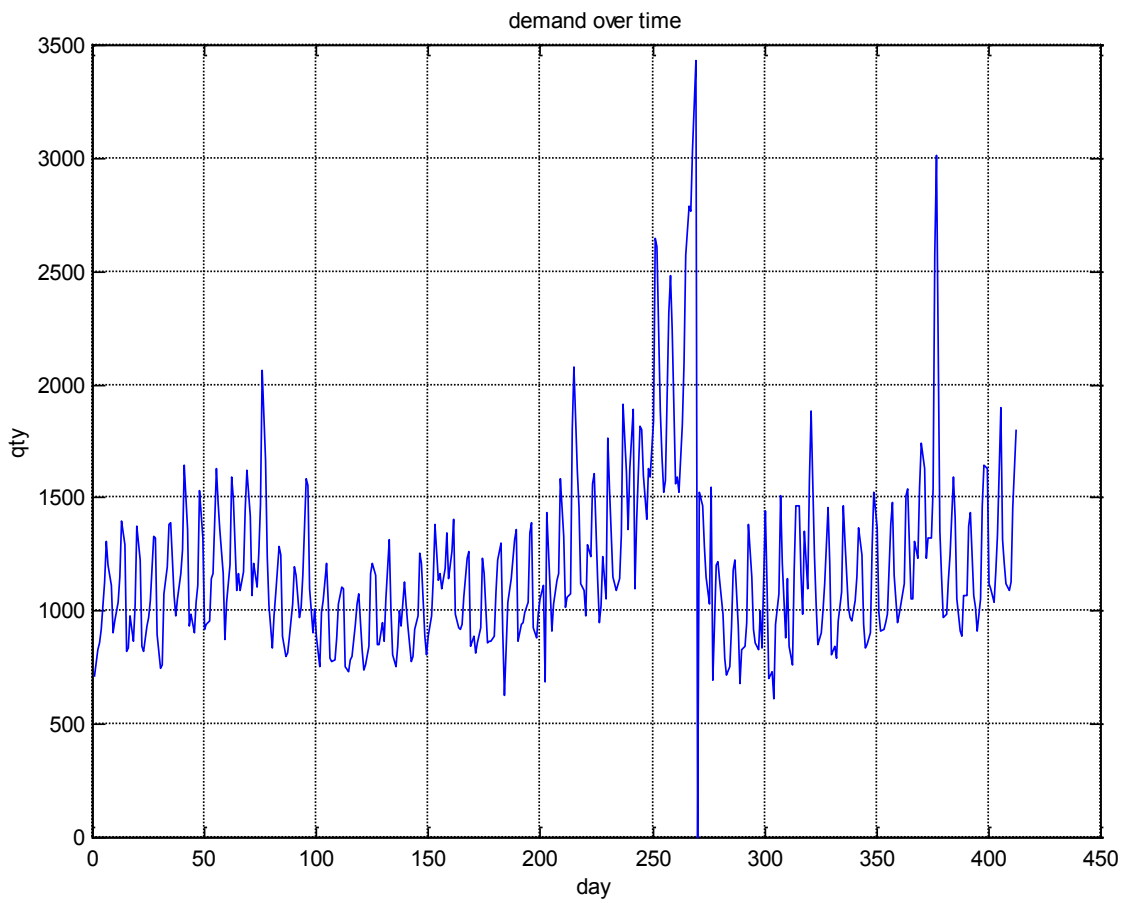


Figure: A typical General Merchandiser's demand over 400 days.

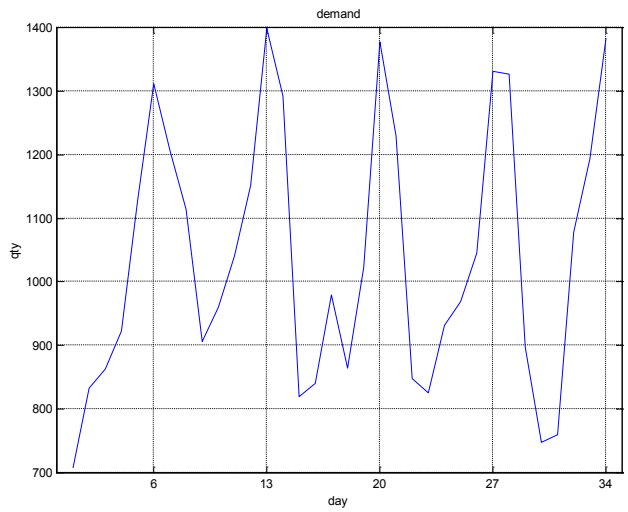


Figure: Close-up of the General Merchandiser's demand series (first 35 days). Peaks are 7 days apart.

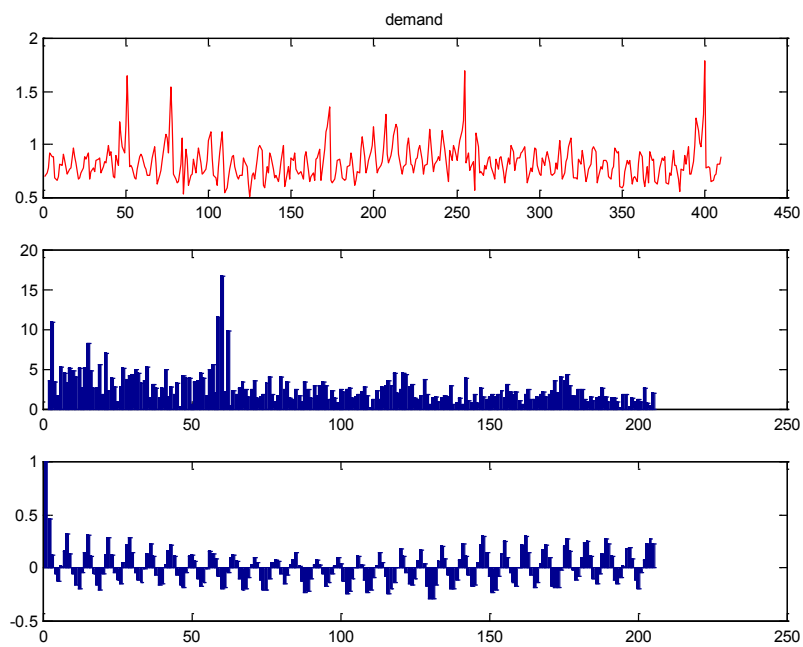


Figure: Bethany demand over time (top) with Fourier spectrum (middle) and auto-correlation (bottom)

Despite being considered very different businesses, there are actually many important similarities between General Merchandise and Grocery retail data. Below we describe some of the most salient characteristics of both GM and grocery retail data, and describe how this presents problems for inferring price-demand models.

1. Day of week effects

GM, grocery data, and web-retail data all show 7-day customer visit periods. In figure x each panel from top to bottom shows in order the demand series, a fourier magnitude spectrum of this demand series, and auto-correlation. The fourier magnitude plot shows the strength of different frequencies in the data. The x-axis is the number of cycles in the 412 day timeseries. The spike at around 60 cycles corresponds to a peak every 412 days/60 cycles = 6.8 days. The autocorrelation plot shows the correlation of the series with itself if that series were shifted by 1,2,3, and so on, days into the past. This shows a strong correlation every 7 days, meaning that the correlation of the series is very strong at 7 days into the future.

GM, grocery, and e-retail timeseries are shown in figure x. For GM and grocery the most popular shopping day is Saturday followed by Sunday - weekends. For eRetail, the most popular shopping day is Wednesday – the middle of the work week at which time presumably most people have the highest chance of access to work computers!

This omnipresent day-of-week phenomenon means that price demand curves can be unreliable if a price is tested for just a few days or weeks. Based on the days over which the experiment was run, the demand at a price point could be several times higher or lower than the average level that would be achieved if the item were observed over a longer period of time. Hence, inferring price demand curves on small amounts of data can lead to erroneous predictions.

2. Small number of price changes

In our grocery store, 29% of items underwent some price change over 412 days. (13,009 of 44,791). A similarly percentage were reported for the General Merchandiser (get Lucy % data). The average number of price changes (for items undergoing some change) during the 412 day period was 5.78. A distribution of number of price changes (given a price change took place) is shown in figure x. This shows that most items underwent 2 changes in around 13.5 months.

In practice this presents the following difficulty. Because so few prices undergo any price change, for most items it is impossible to optimize their price, since no historical data is available to view the effect of shifting this parameter on store-wide and item sales.

[take out - finding] A further difficulty is that many items have the potential to significantly affect demand of other items in the store, but this influence is not seen because those items lay "dormant" in the store until a future time, at which time a price change occurs, and they throw the forecasts.

Thus, any demand model that incorporates cross-price information, may be "taken by surprise" when an item which has hitherto not changed, undergoes its first price change. Without historical observation of the effect of that price change, the model could not assess the cross-elastic effect of that item, and demand forecasting accuracy may decrease.

3. Most items are slow-moving and have sporadic demand which is difficult to forecast

We have found that in web, GM and grocery businesses, around 95% of items in a store's inventory can be categorized as "slow-moving". Slow moving items are items which often sell less than one unit per day, and so have discontinuous and spikey timeseries.

Spikey series can be hard to predict. Let's say that we have a slow moving item with a spikey series. Our prediction will appear as a single spike on a particular day. Even if the prediction is just one day "off", we get no points for predicting that item.

Spikey series also cause difficulty for forecasting methods such as ARMA (Masters, 1996). In response to a series of spikes, ARMA will forecast an averaged level which minimizes its error. This continuous prediction will tend to be 100% off on most days, since it is forecasting SKUs when none are sold. There are also fewer points for ARMA to fit a curve which satisfies the data (a couple of spikes and lots of zeros), and finally, ARMA will be penalized for not hitting the spike and then dropping to zero right away. All in all, this is a difficult problem to fit a continuous curve to.

4. Seasonal changes

Items behave differently in different seasons. Figure x shows an example of a category which is strongly seasonal (Cold and flu medicine). Many other products exhibit these seasonal changes, including demand for beverages, cocoa, soup, and insect repellant. Other products experience high demand during special events such as Thanksgiving.

5. Industry-wide price level changes

Point of sales data only records the price of products in the store. It does not reveal the price that other competitors in the area have adopted. Loss-leader products especially freuquently change price to match competitor pricing, or changes in basic costs associated with the product. For instance, the price of butter increases and decreases due to the price of butter fat. Any demand curves inferred from this data can be misleading, since competitors were matching prices.

6. Each product responds to a different set of variables

Some products are seasonal, others are driven by competitor prices. Incidental or impulse items presumably are driven largely by traffic increases due to store-wide events and special promotions. Because of all of these varying

Production or storage cost

The optimization of profit presented in the previous section ignores a variety of other quantities which affect profitability. For instance, it is also common in Economics to attach a cost for storing or producing the items which are being sold. This function $C(q)$ is a function of the quantity of items which are being held at the store. According to a survey by Wied-Nebbeling (1975), 37% of companies have a linear cost function, and 52% an exponentially decreasing function (economies of scale). In retail, there are many incidental areas of cost, including floor, shelf, or firdge space, utilities, and so forth. In the work presented, we have ignored these costs in profit optimization.

Market expansion or population growth

If you expand your patronage, or just the population grows by a few percent, the mean qty purchased will increase. This plays havoc with the elasticity, incorrectly suggesting you have increasing elasticities (when in fact the patrons might have been reducing their individual purchases and going to competitors). This effect is very marked in the simulator, where we can see the results of increased market share in a very short space of time.

Inflation

If price increased over time, again you would see increasing elasticities, when actually patronage has remained the same but just bought at the same rate at the higher inflationary price.

Seasonality

Seasons will also hurt us, for example, especially high prices can be achieved over Christmas. This might lead us to think you can increase the price of a seasonal item during January, but this isn't the case because its high price was a special price only maintainable during Christmas. The elasticities for seasonal items are only comparable within comparable seasons.

Industry-wide price increases

Accounting for Population growth

We can account for greater populations by normalizing the qties purchased by the number of people now travelling to the store. This amounts to a smoothed timeseries of the number of customers, over a very long
period. For example, an average number of patrons per six months. This is just a count of all customers, grouped by six month intervals - easy to compute. Once we have this, we assume this is the baseline population travelling to the store. We then normalize the mean_qty by this total number. Again, this should be adjustable within the CRM interface, as we will want to experiment to find out what the best setting is (or no setting as the case may be).

Accounting for Inflation

Inflation should be entered at the CRM interface, and then used to uniformly scale all prices. Scaling prices by an inflation factor will change the gradients.

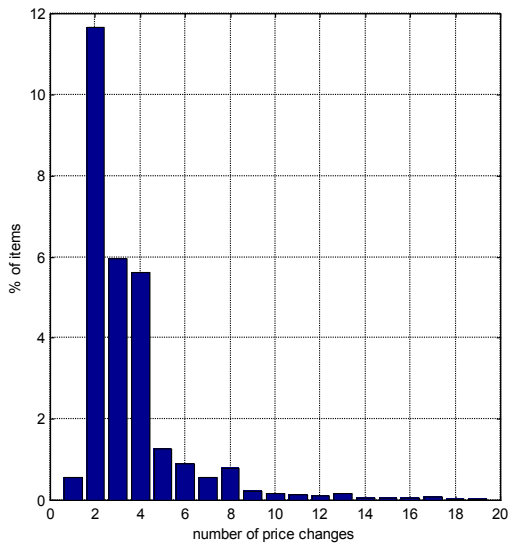
Accounting for Seasonality

Goods that have a seasonal aspect need to be keyed with a mode which indicates a different season they could be in, so that we can compare those products for the current season. So for each item, we have an elasticity for Christmas period, and for other period. Or for each item, we have an elasticity for Summer, winter, and other periods.

Season codes can be typed in by the user, or we could also use algorithms to try to identify significantly different prices during the year, which can define the seasons of the product. I favour the manual typing in of seasons, since this builds expert category manager knowledge into the system.

Accounting for monopolistic price rises

A record needs to be kept of these price increases, and when the occurred. Then we can add those deltas to the historical price, so that the current set of prices are consistent with the past.



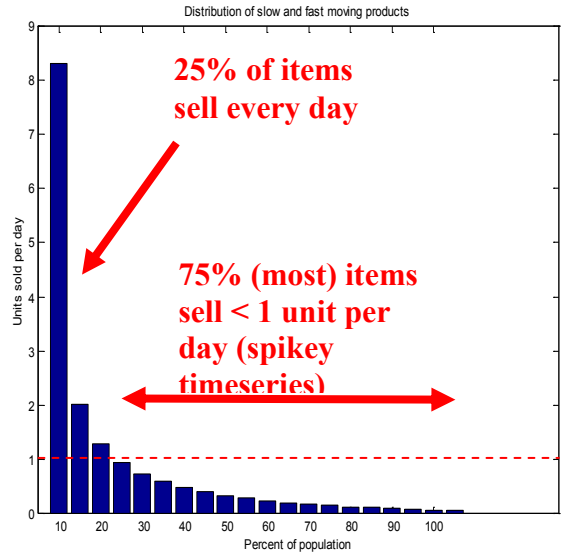
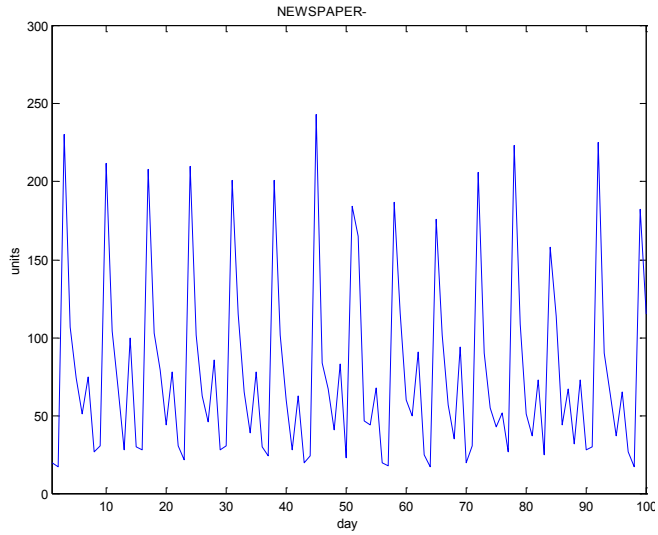


Figure:



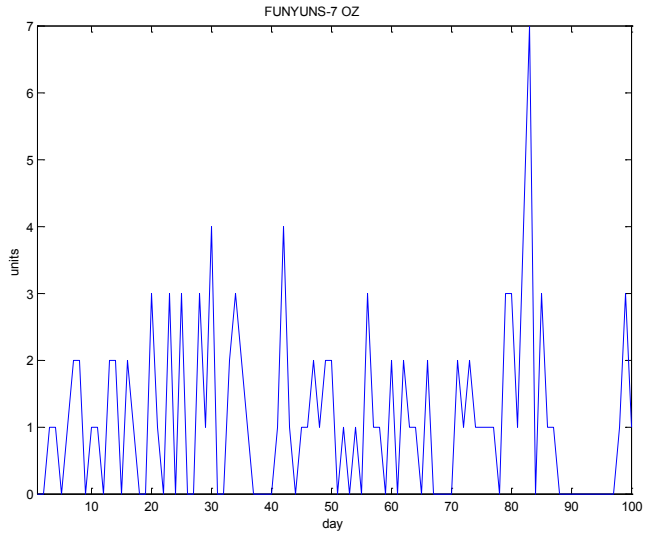


Figure:.

THE DEMAND MODEL

There are three significant problems making demand forecasting and price elasticity analysis difficult in retail:

1) 95% of items in the store's inventory are slow-moving. These items have sporadic and discontinuous demand (figure x).
2) Each product responds to a different set of variables and external factors
3) 70% of items undergo no price change: Many items have the potential to significantly affect demand of other items in the store, but this influence is not seen because those items lay "dormant" in the store until a future time, at which time a price change occurs, and they throw the forecasts.

There are also a range of lesser, but still important confounds which can impair models. These include:

1) Industry-wide price level changes: For instance, the price of butter increases and decreases due to the price of butter fat. It may appear that the price of butter can be increased, but in fact all competitors have increased price at the same time. Inflation also can cause industry-wide price level changes.
2) Seasonal changes: Items behave differently in different seasons

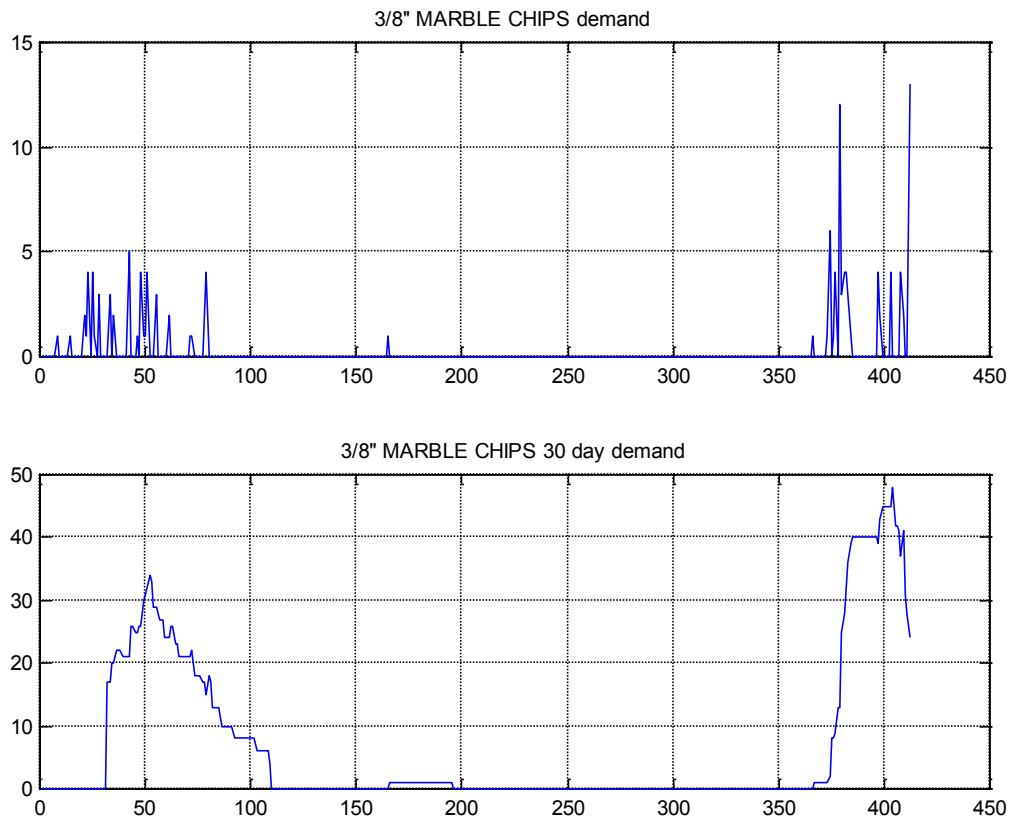


Figure: Effect of 30 day window on demand of Marble Chips. Notice that the 30-day summed series is shifted slightly from the position of the spikes, due to summing the past.

We will not deal with the latter two problems. In practice they can be solved with in the following ways: Seasonal items can be keyed with this information, and then literally treated as different products in their different seasons. (other solutions no doubt exist). Industry-wide price changes can be identified, and then past price levels adjusted correspondingly up or down (similar to "correction for inflation").

We will now proceed to describe how we tackle the first 3 problems. We address the first problem by performing stepwise regression on a number of variables, and selecting those which are important to each particular item. We address the second problem by transforming the demand series into a smoothed 30-day quantity. Finaly, we address the final problem by introducing underlying attributes common to all items.

1. Representation of the demand series

Spikey series can be hard to predict. Let's say that we have a slow moving item with a spikey series. Our prediction will appear as a single spike on a particular day. Even if the prediction is just one day "off", we get no points for predicting that item.

Spikey series also cause difficulty for forecasting methods such as ARMA (Masters, 1996). In response to a series of spikes, ARMA will forecast an averaged level which minimizes its error. This continuous prediction will tend to be 100% off on most days, since it is forecasting SKUs

when none are sold. There are also fewer points for ARMA to fit a curve which satisfies the data (a couple of spikes and lots of zeros), and finally, ARMA will be penalized for not hitting the spike and then dropping to zero right away. All in all, this is a difficult problem to fit a continuous curve to.

We experimented with various methods to mitigate this problem. One method was to encode the demand as a complex value with "time-to-spike" on the real axis, and "size-of-spike" on the complex. We also experimented with reducing the timeseries to a weekly or monthly value. This resulted in much fewer training and test points (from 412 days to 64 for weekly, and 15 for monthly) and so poorer models due to less information.

The approach we eventually used was to recode the demand series from amount sold per day, into a "rolling summed amount" over the last week or last month. The forecast and actual are therefore interpreted as "expected amount over the next 7 days" or "expected amount for the next 30 days".

$$q_i = \sum_{i \in \left\lfloor \frac{-W}{2} \right\rfloor .. \left\lceil \frac{W}{2} \right\rceil} q_i \quad \text{where } W=30$$

This operation converts a spikey series into a continuous series. The tradeoff is that we loose accuracy in pinpointing exactly where in time where the demand took place. This loss of accuracy in time is caused because we have actually applied a low-pass filter to the series, causing us to loose resolution in the high frequency components of the demand series.

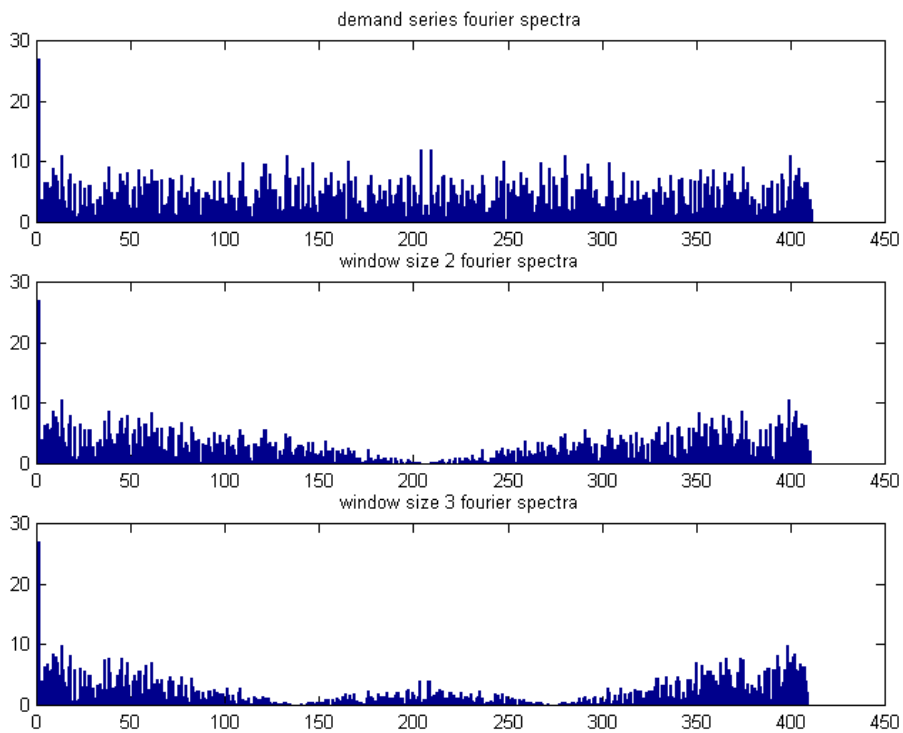


Figure: Fourier spectra for demand series in figure x, resulting from a 1, 2, and 3 width window. Note that high frequencies are depressed using the windowing scheme.

Since we perform an average for timeperiods of size w, all sin and cos curves which have half-periods less than w will be impossible to see in the new series, since they will have been turned into averages. Another way to see this is with Shannon's sampling theorem. Shannon showed that if we need to reproduce a series perfectly, then we need to sample the series at half the wavelength of the smallest frequency component. To reproduce a high frequency, we need to use a window size which is half its period or smaller. If we sample at over half the wavelength, we won't be reproducing those frequencies of size *w* perfectly.

Due to the loss of information, we need to find a good window size. The problem is as follows: If we have a fast-moving item which already has continuous demand each day, then the effect of the applying a moving average filter will be to muddle up the shape of the series and simply reduce foreacasting accuracy (since we're not using all of the frequency components of the series to forecast demand). But for a slow moving spikey product we want a large window.

Therefore, faster moving items should be modeled with smaller windows. This could give rise to Wavelet-like models parameterized for different items, with each item being modeled at its best time/frequency resolution.

In this project we sidestepped this and simply modeled all items using the same window. We experimented with various sized windows, and found that 30 day windows resulted in the highest accuracy over all items, and were intuitively appealing, since forecasts were interpreted as “x units per 30 days”.

A result of this choice of window, was that slow-moving items were actually more accurately forecasted, than fast-moving items! The windowing spread out infrequent spikes to provide easier demand prediction, but on fast-movers blurred their timeseries.

2. Variables used in demand model

The demand model was constructed by performing a stepwize regression for each item on a large pool of variables. These variables were:

- Demand-at-lag terms
- Day-of-week and month-of-year
- Average price, zscore price, Percent-of-normal price
- Own price: the price of the item being forecasted
- Cross-price: the price of other items in the store

Demand-at-lag or auto-regressive (AR) terms

A lag demand term is the demand <lag> days in the past. eg. a lag 5 term is a variable which is the demand 5 days in the past. The Autoregressive part of ARMA is based on putting a series of these lag terms together, and trying to infer future demand. It can be shown that even using a linear regression model, a wide variety of curved demand series can be predicted using this simple model (Press et. al. REF).

Figure x shows that at an aggregate level, the total store demand clearly has a 7-day period. This means that a term with lag 7 would be a good variable to include in our model for store demand.

We also explored the use of cross-elastic lag demand terms. The idea was that perhaps high demand in a product some weeks ago in a different product could cause high demand in this particular product in the future. This didn't give higher results than using the cross-elastic price.

In addition, lag-demand terms had implementation difficulties. To use demand at lag terms in an operational system, we would need access to the latest POS data to measure demand, so that we could predict as far ahead as possible. However, buyers operate on planning cycles of up to 2 weeks, which means that only demand terms 2 weeks in the past or more could be used for forecasting, since that's when the decision to buy is made. Another drawback of demand lag is that it is impossible to get cross-demand terms for future forecasts. Prices, however, can be planned in advance since they are set by the retailers, and therefore long horizon future prediction can be possible.

Day-of-week and month-of-year dummy variables:

A day-of-week vector has 7 elements, with a "1" in the day which is Monday. These variables can be useful if some event occurs on a particular day. For example, perhaps all demand should be multiplied by a factor of 2 on Saturdays. If this were true, then the model could put a weight of 2 on the Saturday component of the week vector. Thus when Saturday comes around, the demand is multiplied by two. Similar vectors can be created for special events such as 4th of July or Christmas, although this study did not include special events in the model, since the goal was to predict daily demand[2].

In our tests no model selected day-of-week or month-of-year variables as components of an item model.

Store-wide price

When a store undergoes a sale, a large number of product prices drop, resulting in a surge in demand. If an item's demand was affected by this sale, then these variables should be important in predicting demand.

We created three storewide price measures - mean price, mean Z-Score price and mean percent of baseline price. Figure x shows each of these series for retailer Lucy. The Z-score and percent of baseline metrics were designed to be sensitive to whether the present pricing was higher or lower than normal.

Dummy variables were created as follows: Let *period* be the number of separate positions that the variable could be in (number of separate "states"). For example day-of-week has 7 possible states. Month-of-year has 12 possible states. Let *width* be the number of days filled by any state. For example, day-of-week has a width of 1. Month-of-year has a width of 30. The dummy vector is a Toeplitz matrix formed using a vector with *width*period* elements, the first *width* elements equal to 1 and the remaining equal to 0. For example, day-of-week dummy variable is a Toeplitz of [1 0 0 0 0 0 0] (width 1, period 7). Month-of-year is a Toeplitz of a 360 element vector with the first 30 elements equal to 1.
[2]

$$meanprice(d) = \frac{1}{N}\sum_{i=1}^{N} price(d)_i$$

$$pricezscore(d) = \frac{1}{N}\sum_{i=1}^{N} \frac{price(d)_i - \mu_i}{\sigma_i}$$

$$pricepercent(d) = \frac{1}{N}\sum_{i=1}^{N} \frac{price(d)_i}{\mu_i}$$

Own price

Own-price is the price of an item under consideration. For instance, in forecasting demand for Juice, own-price will be the price for juice. Intuitively, own-price should normally be an important variable for predicting changes in demand. However only 29% of items undergo any own-price change in 412 days. Therefore, for 71% of items, own-price will have no predictive value at all.

Despite this observation, some items show compelling price-demand behaviour. Figure x shows a very slow moving item. Despite this, every time the price series drops, there is a spike in sales. This graphically demonstrates that some products do react to changes in their own price.

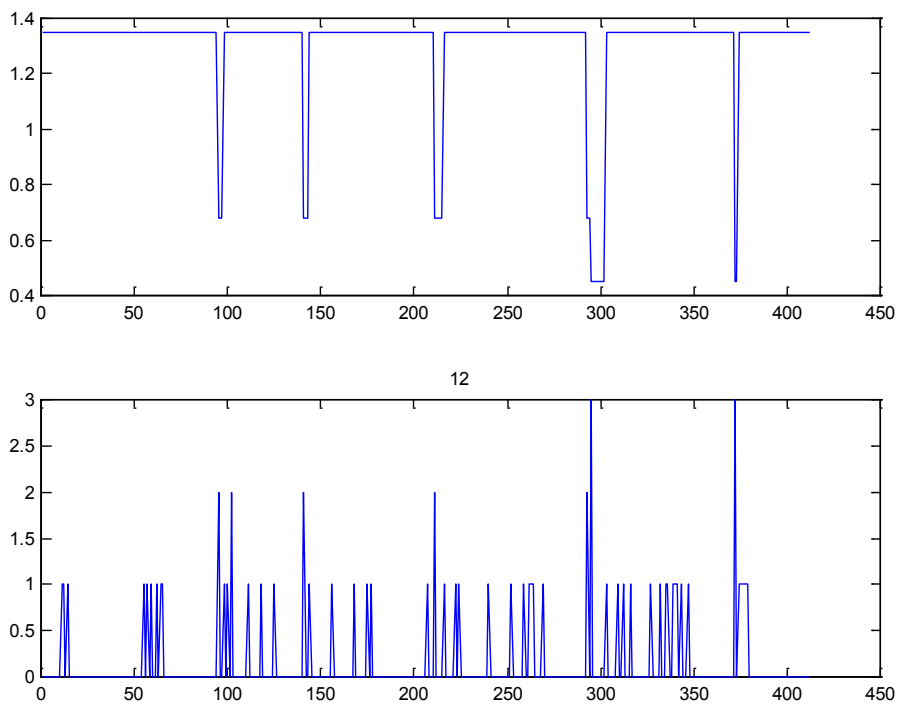


Figure: Common effects seen in the data. Top is price, bottom is units sold. Whenever price drops for a time, we get a demand spike.

Cross-price

The use of cross-price variables is curtailed by their sheer number. For any given item, there are up to 40,000 other items at the store which could be affecting its sales. This means 40,000 regression equations need to be tested each step of the stepwize procedure!

This problem was solved in the following way: we created a full correlation matrix *Corr* between the demand series of the target item, and the price series for all other 40,000 items so that for the *(i,j)*th element of this matrix:

$$= \frac{\sum_{t=1}^{N}\left((price_{ti} - E[price_i]) \cdot (qty_{tj} - E[qty_j])\right)}{\sqrt{\sum_{t}^{N}\left((price_{ti} - E[price_i])^2 \cdot (qty_{tj} - E[qty_j])^2\right)}}$$ [3]

We then sorted the correlations into order, and then selected the top n positive and negative correlations. We then performed a stepwize regression using only those best n correlated prices. This approach was equivalent to doing a linear regression with all variables, but avoided the need to invert any matrices, and so was an extremely fast operation.

Figure x shows a sort of the top positive and negative correlations for item x. This shows that only a tiny number of items have a strong correlation with the target item. The strength of those correlations drops off very rapidly in an exponential curve. (the inverse sigmoid occurs because we have separated positive and negative correlations. If we took the absolute value of the correlation, the distribution would of course be a single exponential sloping towards zero). Therefore, although there are 40,000 candidates, only a tiny number of items have anything to do with the target item and should be incorporated into the stepwize variable pool.

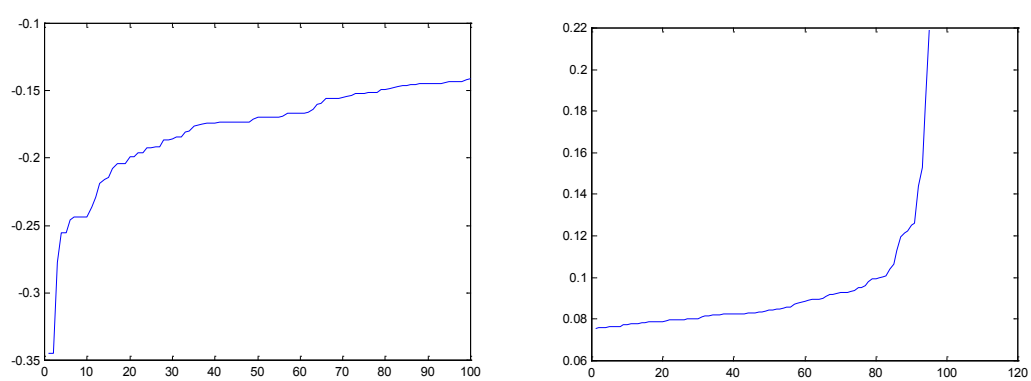


Figure: Correlations for item 100 (item 1 description). The correlation between its own price and demand was -0.0139, which was negative (good), but 3854th out of all correlations.

Correlation set for DM Squeeze Ketchup–40 Oz

R	Driver Item
-0.45	KEY WEST PINK SHRIMP-PER LB
-0.43	IGA FLOUR-5 LB
-0.39	JIMMY'S SPINACH DIP TUB-16 OZ
-0.39	BC AU GRATIN POTATOES-5.5 OZ
-0.39	SUBS TO ORDER LG REDBIRD-
-0.37	SUBS TO ORDER LG BADGER-
-0.36	BEEF CHUCK SHLDR STEAK BNLS.-P
-0.34	OM REG SL BOLOGNA 10X11-1 LB
-0.34	G MILLS TEAM CHEERIOS-13.7 OZ
-0.34	missing

Figure: Items with strongest price-demand cross-correlation with DM Squeeze ketchup. Items that "make sense" are interspersed with items that seem to be spurious. For instance, a price drop on Key west pink shrimp or subs to order is associated with increased demand in Squeezeketchup. However, this is also the case when flour goes on sale.

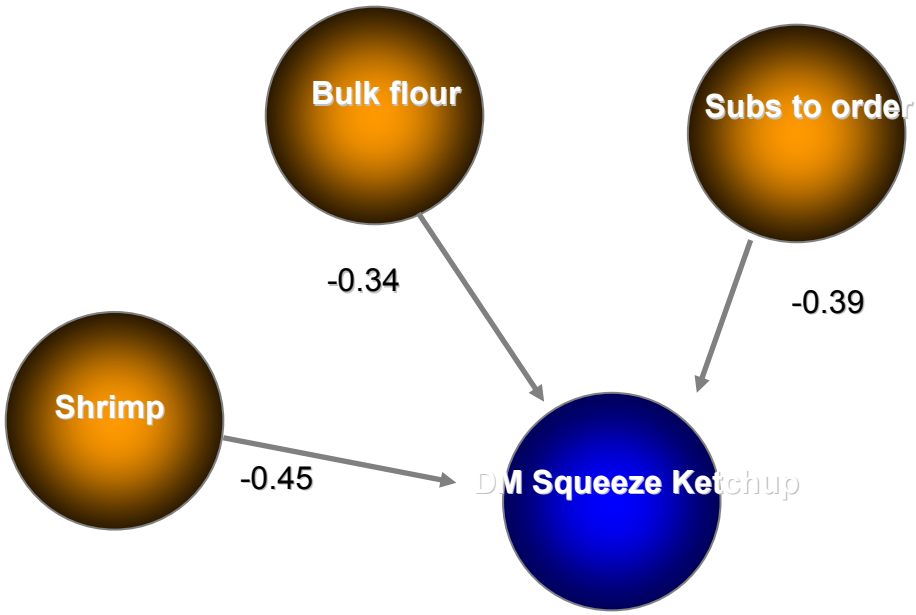


Figure: Correlation set for DM Squeeze ketchup represented as a graph

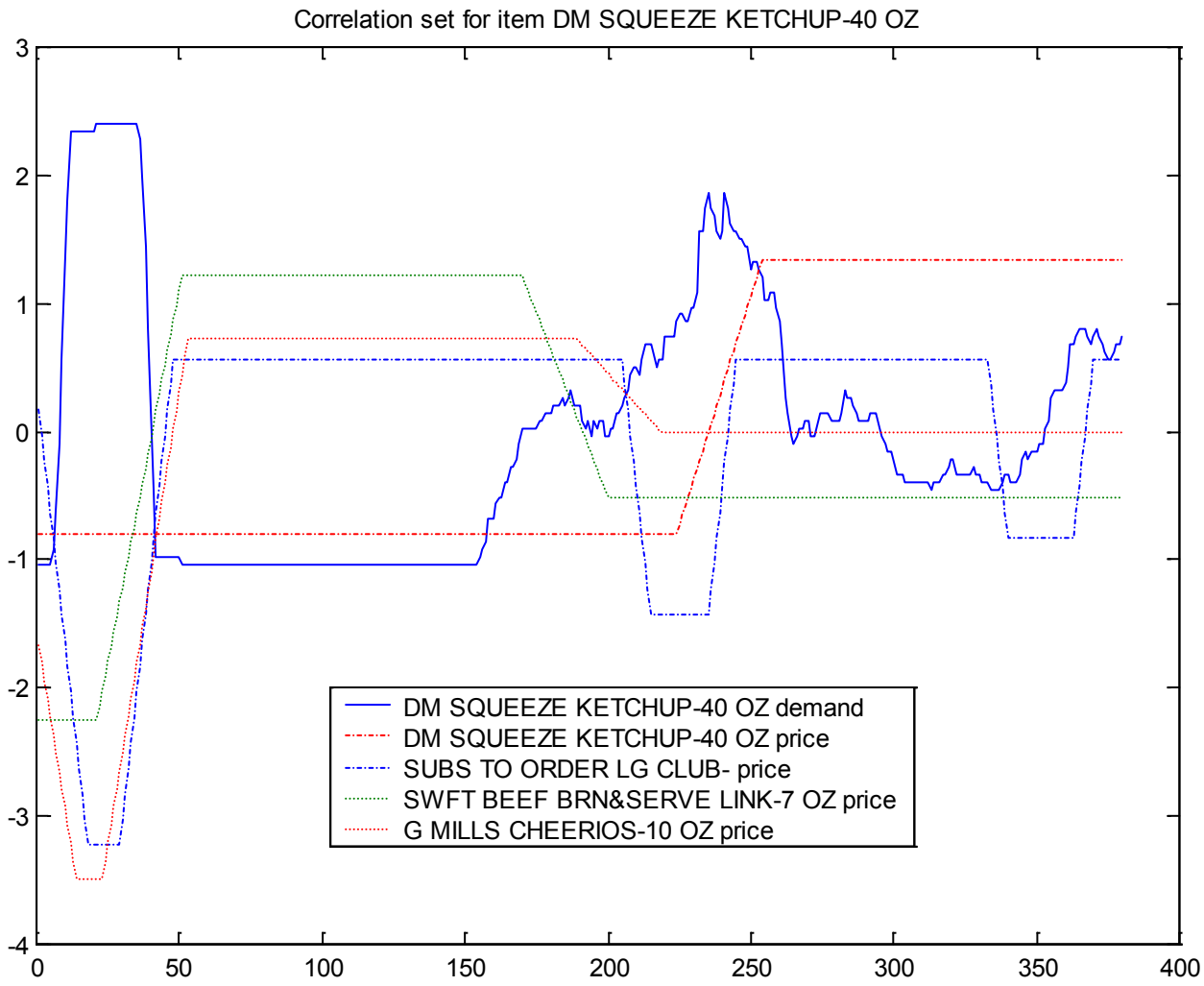


Figure: Correlation between demand for DM Squeeze ketchup and price of four items including Squeeze ketchup, subs to order, and Beef links.

This method has proved extremely sucessful. For example, relaxing the stepwize set from 5 items to 50 results in only a miniscule improvement in accuracy, meaning when we select only the top 5 items, we infrequently capture the most important driver items.

Another interesting fact is that of the stepwize candidates, the item's own price is invariably one of the poorer predictors! In figure x, the item's own price is ranked 860th in correlation with demand. Figures x (price demand) also show that many items can have a flat price, and yet be driven by price changes occurring in other products.

This has a variety of implications. Whilst it is true that changing an item's price can change demand, many customers are more strongly attracted by the prices of other items at the store. For example, customers may see cheap lettuce and tinned soup, and then be encouraged to buy many more products at the store.

Variables left out of the model

The model did not include any of the following: manufacturer and retail-store advertising, coupons, and in-store location.

Model form

Several functions have been proposed for describing consumer demand. These include linear, multiplicative (exponentially reducing), attraction (sigmoid shaped) and Gutenberg (inverse sigmoid shaped) functions (Simon, 1989). Although the Gutenberg model is often favoured, Simon (1982) and Kucher (1985) found that all four models provide a reasonably good fit to the observed data. Simon (1989) hypothesized that this might be because all models behave very similary through their middle range, and most prices observed are not extreme prices.

We have chosen to implement a linear model. There are three reasons:

1) A linear model admits easy formalization and differentiation.
2) Tests with more elaborate models showed only slight improvements. For the case of higher degree polynomial models, the model also becomes rank deficient, increasing the numerical instability of the model. The results of these experiments are provided in figure x.
3) Since this model is a first step towards automated pricing and what-if modeling, we wanted to stress simplicity, and allow elaboration later. Small tweaks / improvements to the error, which are important to a real system, can be pursued at a later time. However, for now we need to know if the method works.

The model is:

$$\mathbf{Q} = \mathbf{P}.\mathbf{W} + \mathbf{c}$$

Where **P** is a row vector of prices for each item, and **W** is a square matrix with weights for each item in the model **i** as a column of weights, and **c** is a row vector of constants. Figure x shows the form of the cross-price component of the model. The **W** matrix is a symmetric matrix giving the demand influence (positive or negative) of each item on another item.

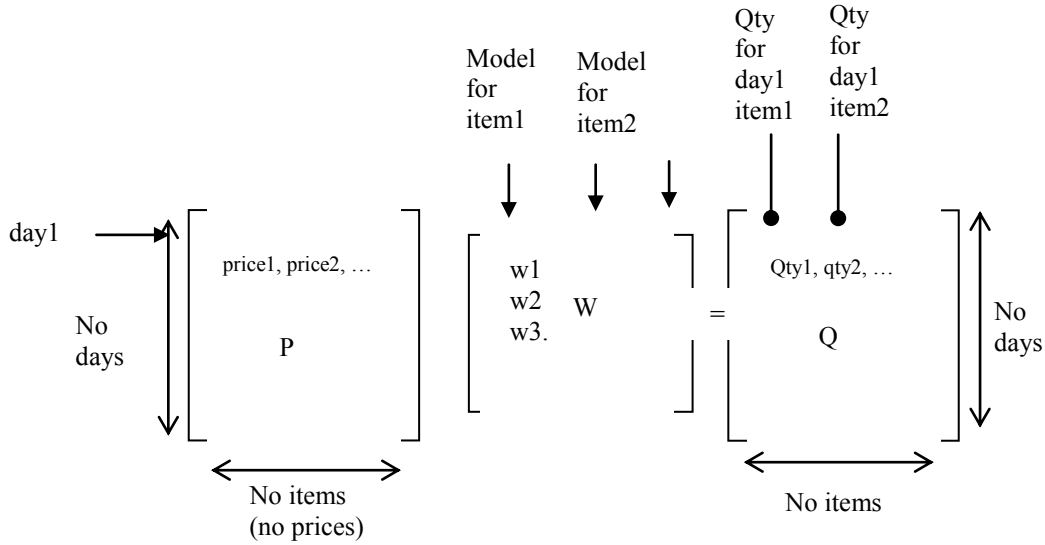


Figure 2: Matrices for Price and Weights. Price is cross-multiplied with weights to give the model estimate for quantity. Constants are added in a separate step.

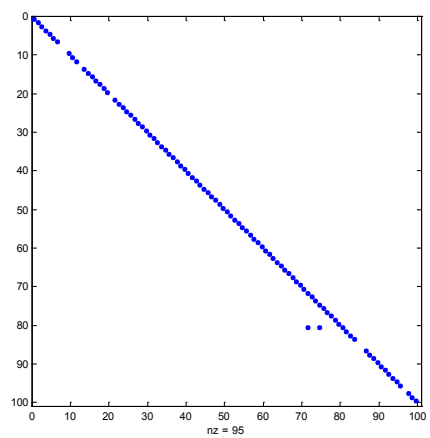


Figure: Sparsity pattern of the W matrix

RESULTS

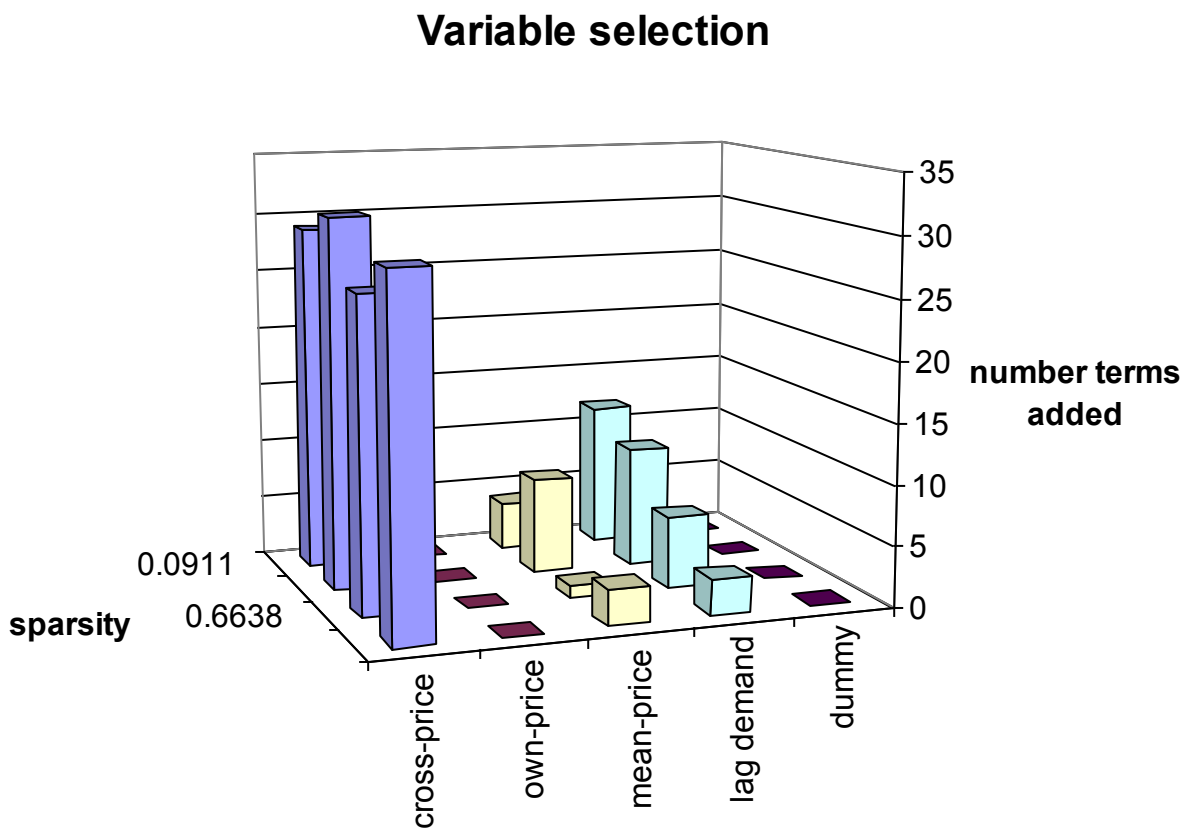
The model was trained on Lucy's data using the first 230 days as a training set, and the remaining 382 days for testing. A similar period, 250 days training and remainder test, was used for Bethany. Since less than a year of data was used in training and testing, seasonal effects from the Christmas period should have impacted the model's accuracy. For example, if Christmas occurred in the training set, then test set predictions should have been higher than normal. However, even on this seasonally compromised period of data, the model was surprisingly accurate. The stepwize procedure used an R criterion to determine goodness of fit, and halted if R improvement was less than 0.03.

Variable selection

We determined which variables were most often selected by running 100 items from different UPC ranges through the model. We then counted the number of each variable which was added to the model.

Cross-price terms were the best variables. For sparse items cross-prices were selected 10 times more than lag-demand. For continuous items, cross-price terms were still selected 2.4 times more than lag demand.

No dummy variables or own-price variables were selected in our test sample. This was surprising, as intuitively one imagines that if you change the price of an item, then the demand should be affected. However, this result indicates other prices actually have a stronger influence on demand. In the concluding section we will explore this in more detail.



Lag terms weren't selected as often as cross-prices, however, lag-terms became more useful if: we decreased the latency of the lag-demand terms, or we used less sparse timeseries. Figures x and y show the difference between using minimum lag of 14 and 5. If lag 5 terms are available, the model uses them as much as the cross-price terms.

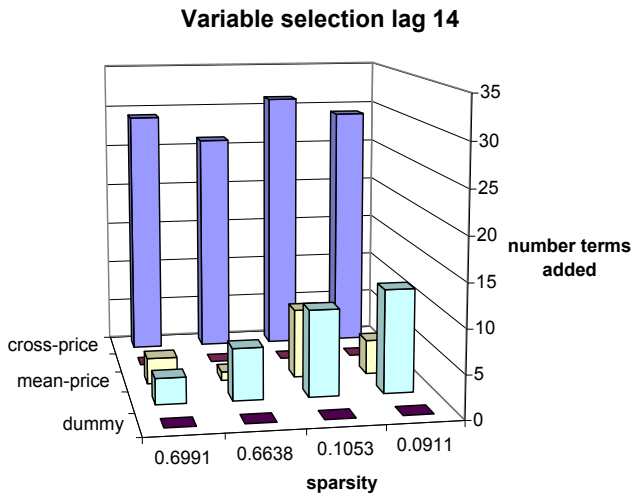


Figure A: Rate of selection when only lag terms of latency two weeks or higher are allowed into the model. Price is the most favoured variable for determining demand.

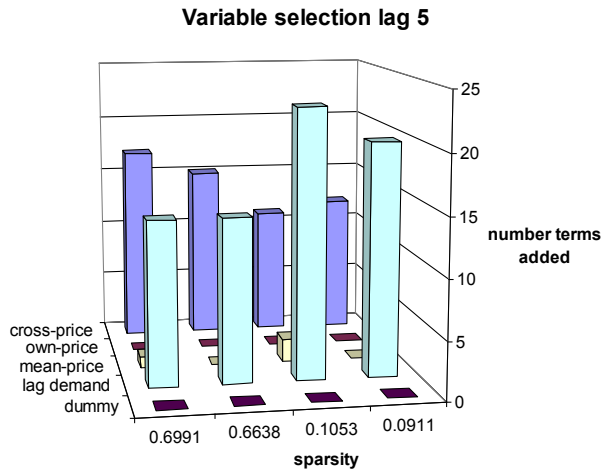
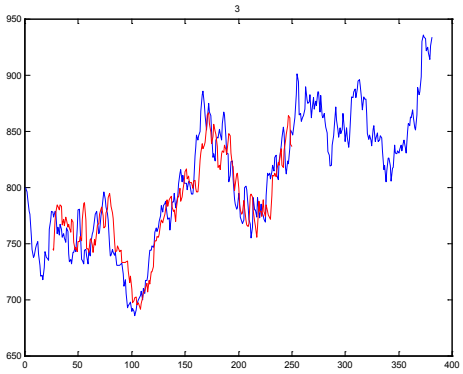


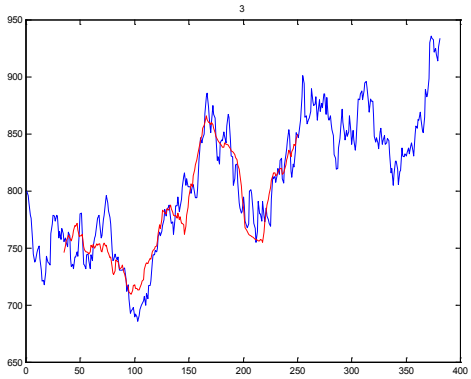
Figure A: Rate of selection when lag terms with latency 5 days or higher are allowed to be selected. The model places much more emphasis on lag terms, and on continuous timeseries, even selects them at a higher rate than the cross-price terms. The cross-price terms are still selected at a high rate, however.

DICKS MILK 2% 1/2 GAL-1/2 GAL

Lag 5



Lag 14



Price only

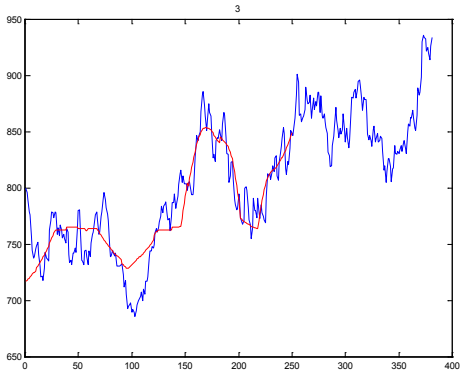


Figure: Item 3 fit on training set using lag terms of latency 5 or higher, 14 or higher, and none at all. If lag 5 terms are allowed, then the model changes its behaviour and begins tracking the series of 5 days in the past. This kind of behaviour minimizes its error, however results in forecasts which can't predict changes. A lag 14 model doesn't show this tracking behaviour, and a model which uses price only also doesn't show this behaviour.

However, if these short latency (eg. Lag 5) demand terms are used, then a curious phenomenon occurs. The predicted demand series begins to trail the actual demand series – ie. the forecasts become "late". This phenomenon does not occur when using longer latency lag terms, such as lag 14 (figure x), and nor does it occur on every item – some items such as 48 the lag terms hit the series "dead on" (figure x). This means the effect is not caused by a programming error.

For example, look at figure x. This shows that at lag 5, the predicted series carefully follows the actual series, and is late. However, at lag 14 this effect seems to have vanished (if it was a coding error, the series should trail by 14). Why does the model prefer to be late and simply track the series if short-latency lag terms are available?

For short latency demand series which don't fluctuate too much, the error resulting from following actual levels can be quite small. We have documented previously for Wilco that the ARMA process will often simply track the actual level. The same seems to be occurring here. Although this behaviour decreases the error, it may be undesirable. A late forecast provides little comfort to retailers who need to plan inventory levels into the future. In contrast, the lag 14 model both "looks" better (figure x), and predicts level changes when they occur, even it its errors are a higher.

Forecasting accuracy

In order to quantify the accuracy of the forecasting method, and compare its accuracy to other possible variants, we ran the following experiment. We took 16 categories (chosen because they would be involved in a price optimization experiment described later), and used the first 230 days for training. We set aside the remaining 187 days as a test set.

We next tested four variants of forecasting methods. These were:

1daycont:
1 day continuous timeseries. This model provided all variables to a stepwize selection procedure except cross-price, however, demand and own-price were daily timeseries. Daily timeseries have considerable problems with sparsity and non-continuity as described previously, and this would show how easily a regression-based model could utilize these day-level variables for predicting demand.

30daycont:
30 day continuous timeseries. This model provided all variables to a stepwize selection procedure, except cross-price, where demand and price series were represesented as 30-day moving averages.

pp:
pricepoint method. This model utilized single price-points to construct a demand curve for each of those points. Price was then used to forecast demand.

cross-price:
cross-elastic method. This model utilized all variables including cross-price terms, and represented demand and price both as 30-day moving averages.

Results from each of these methods is provided in table x, and examples of the forecasting performance of the methods are shown in figure x through y. The 30daycont model scored a correlation of R=0.26 which was higher than the 1daycont model R=0.21.

The cross-price model clearly outperformed all of these other models at R=0.86.

The stunning success of the cross-price model brings up a number of issues. If we look at the correlation graphs provided in figures x,y,z, we can see what other item-prices are being used as a basis for forecasting demand. These other prices sometimes are intuitive, such as Ketchup being driven by price drops in weiners, and peanut butter being driven by price drops in strawberry jam. However, they sometimes appear to be spurious, such as Ketchup being driven by flour or cheerios.

Essentially, what we believe has happened is that groups of products are probably undergoing sales at roughly the same time, for instance, when category managers design promotions they may advertise a set of usual suspects. Thus, correlations and anticorrelations become "built into" the demand/price series. In addition, because there are so many items being advertised at the same time, the item-price series that is best correlated with the demand may not be the "best" driver item.

If this was so, then we still need to ask why the model is so effective in forecasting the test series. Surely if these drivers were spurious, in the 187 day test set, this would become apparent, and forecasting accuracy would go down.

The only explanation that seems to make sense is that category management practices may well have remained stable into the test set, enabling the forecasting algorithm to continue to correctly forecast demand for those items, even with sometimes spurious driver items.

Model Fragility

Despite the excellent peformance on this test set, we believe that ultimately this forecasting method suffers from several weaknesses, which may be exposed if more elaborate price experimentation and promotion was performed.

The first problem is related to the picking up of spurious driver items in developing a forecasting model, discussed in the previous section. If category managers begin advertising different sets of products, forecast accuracy will break down.

The second problem in focusing on individual item-level prices, is that the model will always be "taken by surprise" by a sale of a product that has never experienced a price drop; for example, a sale on blueberries. When this sale happens for the first time, and exerts cross-influences on other products, the forecasting method's predictions may be grossly inaccurate.

Because of these problems, we believe that ultimately forecasting should be improved by removing the focus from individual items, to perhaps category-level price, or a indicator timeseries which shows how many products are under-normal price in a category.

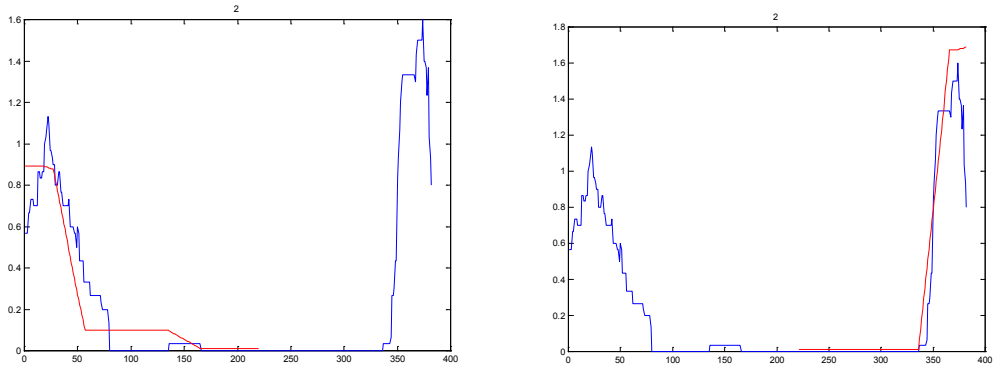


Figure: (top): Training set, (bottom): Test set. The vertical axis is the average number of units per day. For slow movers, this average number of items will tend to be less than one. The axis should be multiplied by 30 to give the average number of units over the next 30 days, which is what the model is designed to predict. The city-block-like patterns are a result of smoothing the demand out over 30 days, which I have found improves prediction markedly on sparse timeseries.

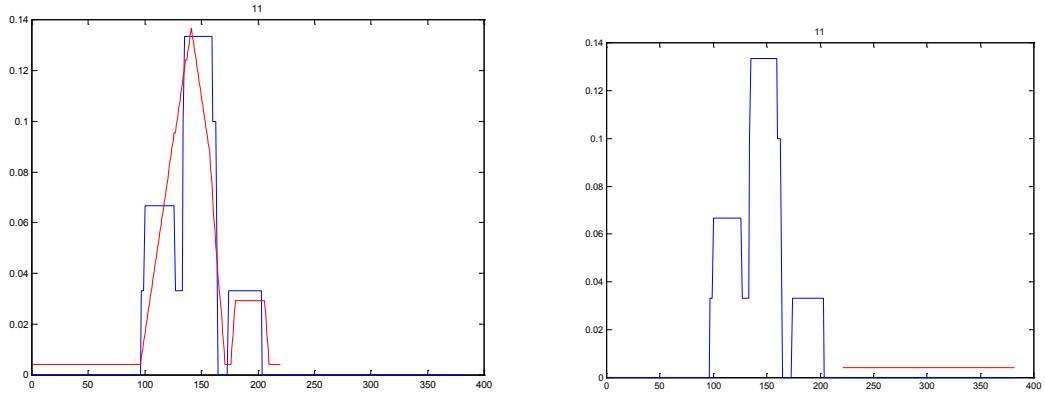


Figure: Corrctly predicts zero demand over the next 200 days. Note that the predicted level is a little above zero. This is because the model tries to minimize its squared error, and because there have been cases where demand occurred on a non-low-price day, it compromises by elevating its level a little. In terms of cases predicted over the next 200 days, however, this is close to accurate.

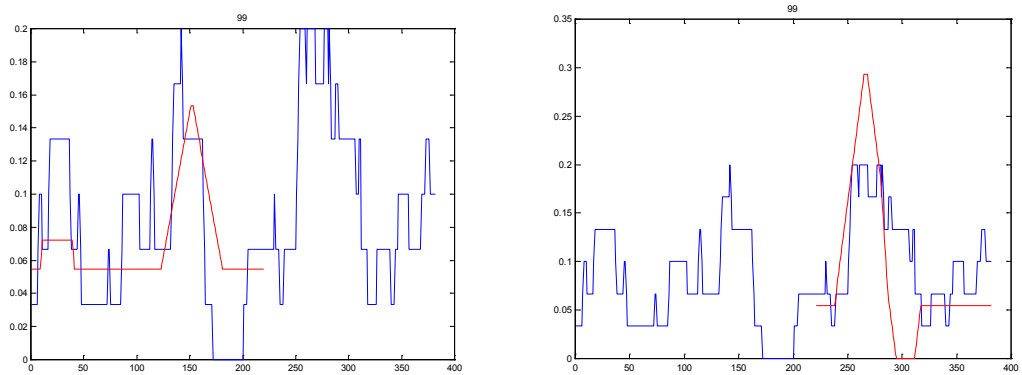
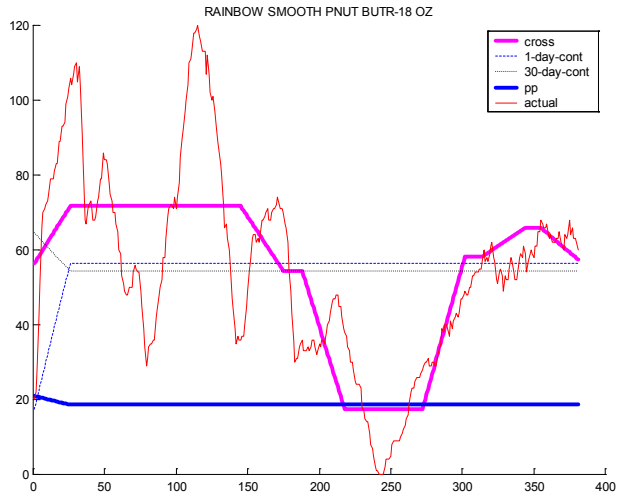
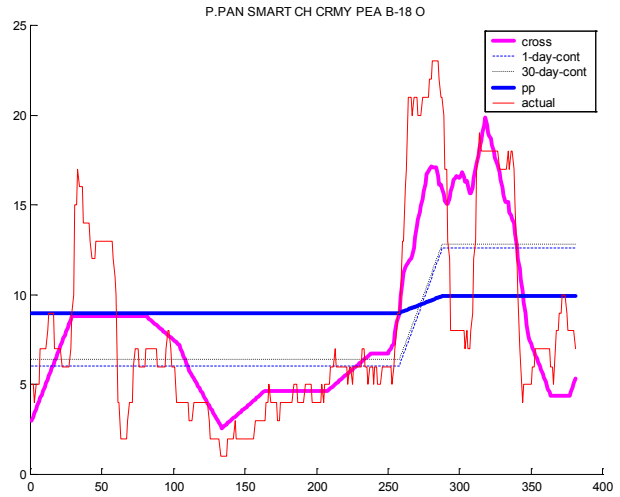
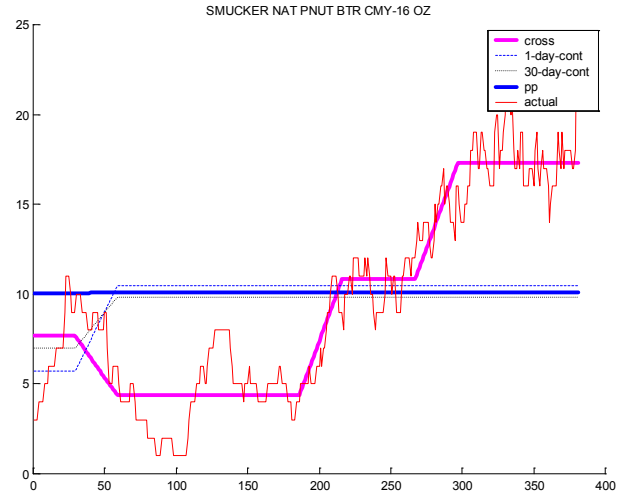
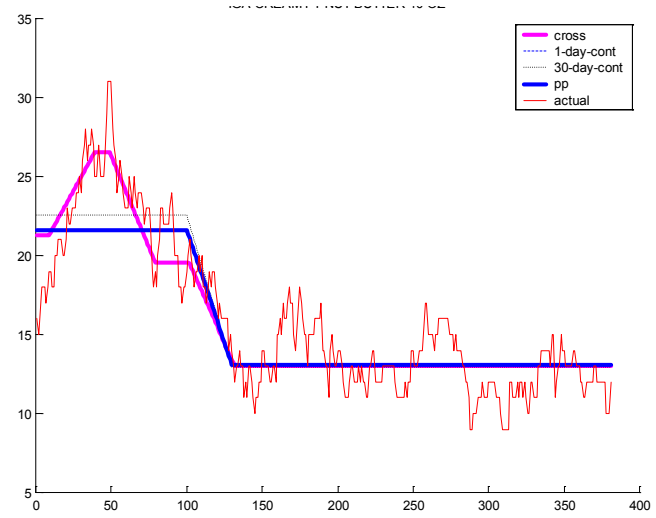


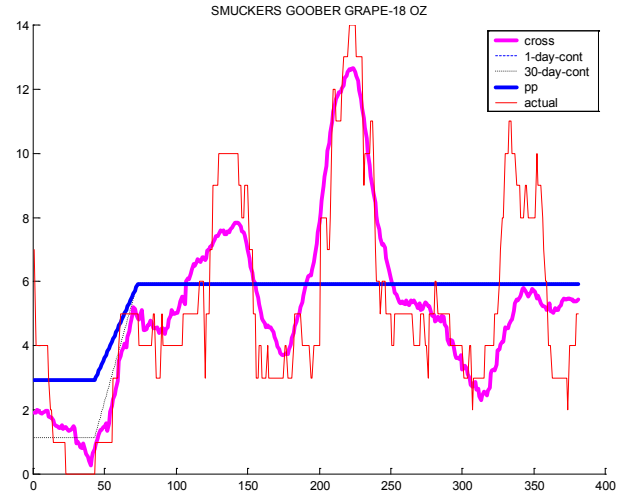
Figure: This is one of the few timeseries that moves relatively fast (a unit being bought at least every 30 days). Again, the training cases allow the model to "hit" the correct demand with startling accuracy.

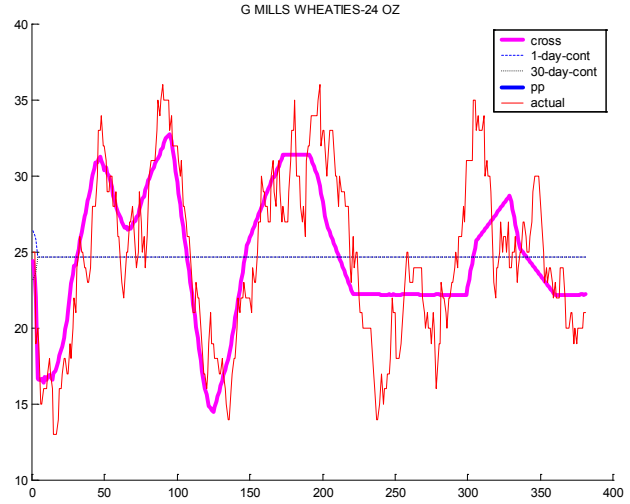

RAINBOW SMOOTH PNUT BUTR-18 OZ
cross
1-day-cont
30-day-cont
pp
actual

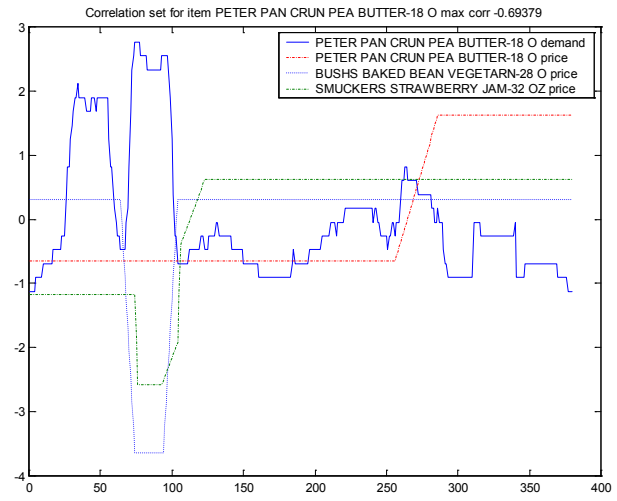

P.PAN SMART CH CRMY PEA B-18 O
cross
1-day-cont
30-day-cont
pp
actual

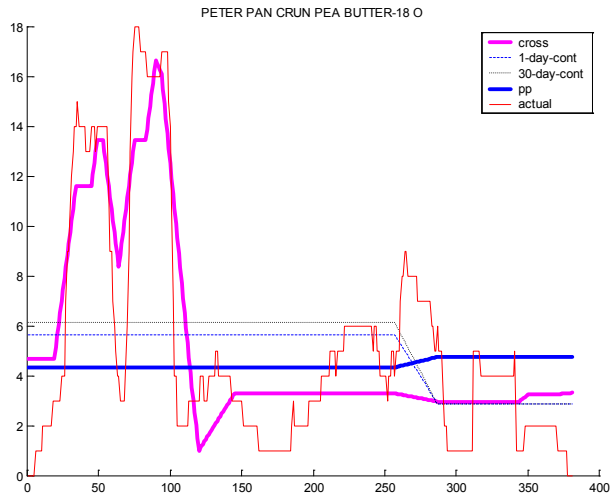

SMUCKER NAT PNUT BTR CMY-16 OZ
cross
1-day-cont
30-day-cont
pp
actual

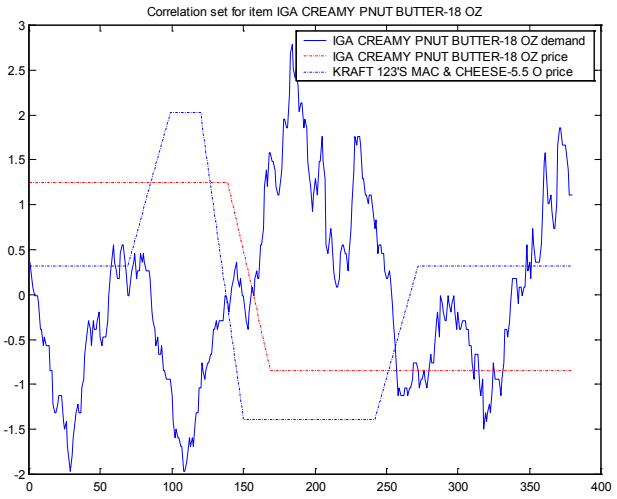

cross
1-day-cont
30-day-cont
pp
actual

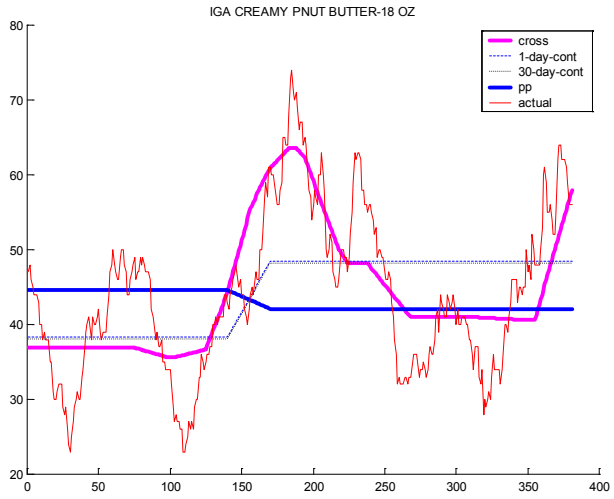

SMUCKERS GOOBER GRAPE-18 OZ
cross
1-day-cont
30-day-cont
pp
actual

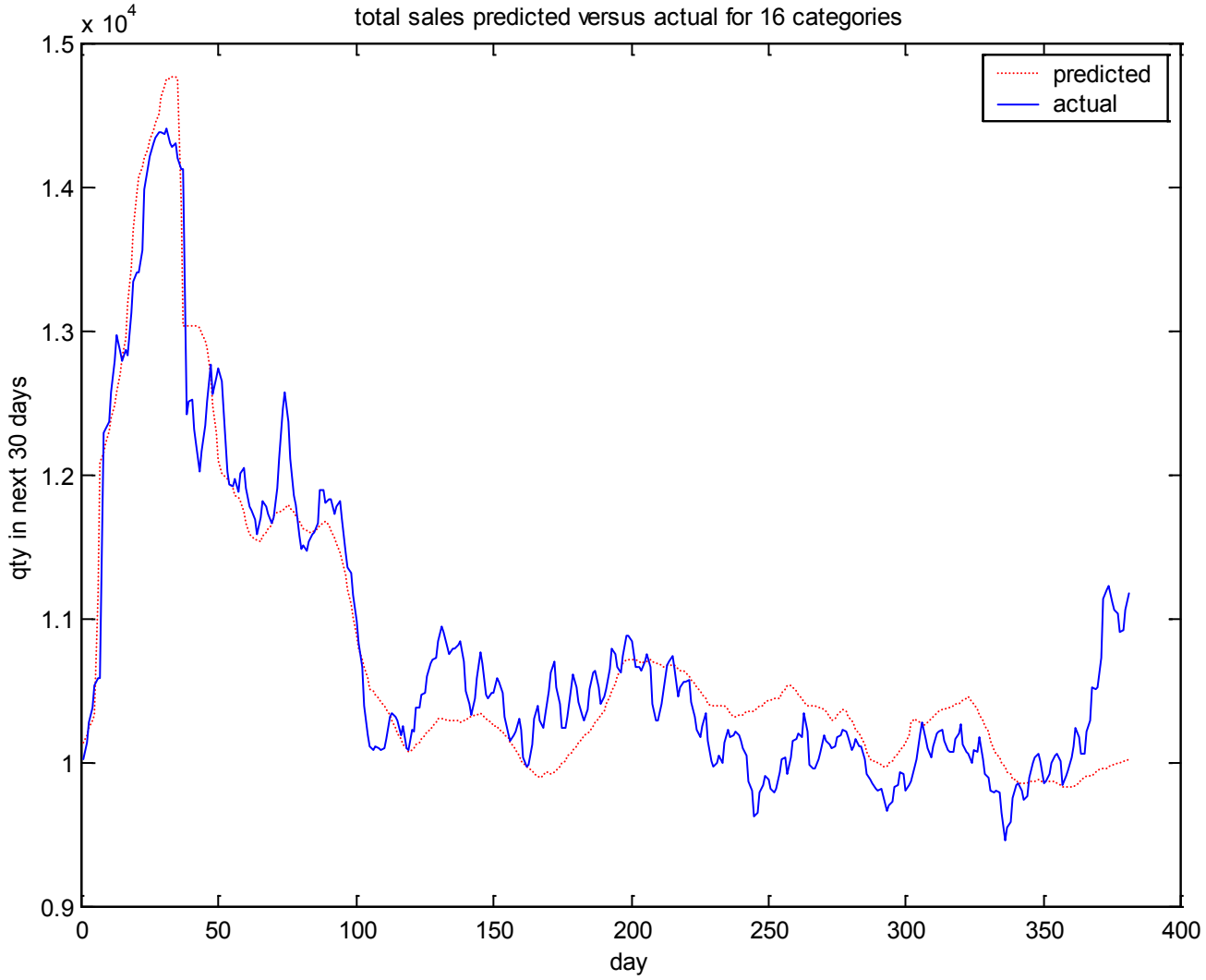

G MILLS WHEATIES-24 OZ
cross
1-day-cont
30-day-cont
pp
actual


Correlation set for item PETER PAN CRUN PEA BUTTER-18 O max corr -0.69379
PETER PAN CRUN PEA BUTTER-18 O demand
PETER PAN CRUN PEA BUTTER-18 O price
BUSHS BAKED BEAN VEGETARN-28 O price
SMUCKERS STRAWBERRY JAM-32 OZ price


PETER PAN CRUN PEA BUTTER-18 O
cross
1-day-cont
30-day-cont
pp
actual


Correlation set for item IGA CREAMY PNUT BUTTER-18 OZ
IGA CREAMY PNUT BUTTER-18 OZ demand
IGA CREAMY PNUT BUTTER-18 OZ price
KRAFT 123'S MAC & CHEESE-5.5 O price


IGA CREAMY PNUT BUTTER-18 OZ
cross
1-day-cont
30-day-cont
pp
actual



method	R	meanabs	meanerr
cross	0.826	0.28%	-0.14%
1daycont	0.229	11.49%	9.06%
30daycont	0.261	8.84%	8.83%
pp	0.056	57.50%	-30.31%

PRICE OPTIMIZATION

In this section we will provide a survey of price optimization methods, as they relate to our problem of optimizing price in retail stores. We will use our model for predicting demand, and re-apply it for price optimization. The problem is to find a vector of prices across all 40,000-100,000 items in the store, which results in the highest profit for the store.

Problem Difficulty

Search space

There are many hurdles to optimizing price. The item-price search space is huge. Let's say we decided to test only prices that an item had been set to at some time in the past. If we searched all of these, the number of price combinations would be number of prices for item1 times number of prices for item2 times the number of prices for item3, … to itemn:

$$\prod_i prices_i$$

If all items had the same number of prices (for instance), then the number of combinations would be:

$$\text{Number\_of\_prices}^{\text{number of items}}$$

which is exponential. In reality the number of combinations won't be quite as high because some items will have more prices than other items. However, according to real data, for a sample of 42 randomly chosen items out of the 40,000, multiplying out the number of possible price combinations resulted in an astounding 10,000 trillion possible price combinations.

The Traveling Salesman Problem (TSP) also exhibits this behaviour, in that sequentially testing all possible routes grows exponentially with the number of cities. The similarity, it turns out, is more than coincidental. Price optimization, like TSP, properly belongs to the class of NP-Complete problems (proof in Appendix A). NP Complete problems are normally regarded as intractable but could be solved non-deterministically if vastly parallel machines existed. Until Quantum or DNA computers become available (see Adelman, 1995), there are no known techniques for perfectly solving these problems.

Because these problems are intractable, they must be approached using heuristic approaches. The techniques discussed below all use heuristics of one form or another, for instance, relying on analytic gradients to provide information on the direction of more desirable regions.

SINGLE ITEM OPTIMIZATION

Single item optimization involves optimizing profit by adjusting only an item's own price. This is distinguished from Global optimization, which tries to take into account these changes.

Single item optimization can be an attractive method, because the optimum for each item can be found exactly. Below we have also examined a method called Selective pricing which looks at past price-demand pairs, and simply selects the price which resulted in the highest profit. These methods are robust, simple, and cheap to implement, and so they provide a good baseline to compare more elaborate (and possibly fragile) strategies against.

SINGLE ITEM BEST PRICE

The first single-price optimization method we will present will be called *selective pricing*. In our data 29% of the items changed price at some time during the 412 days. As a result, we can measure the average demand each day for each item at each of its different price thresholds.

The simplest way to choose the best price, is simply to choose *p* such that revenue is maximized. In other words, we have a set of prices, qties, and profits, and we just choose the price which resulted in the best profit.

$$\mathbf{max}\ p : (q_{ip}(p-c))\ \forall p$$

The drawbacks to this method are that it can only choose prices which were tried at some time in the past. This is also the method's primary advantage - unlike the inference method we shall consider next, there is some degree of certainty about the price and effect on demand, since the product was tried at that price in the past. For instance, unusual effects such as ".99-ending" prices will be automatically handled, while they may be mis-estimated using one of the methods below that treats prices as continuous numerical variables.

SINGLE ITEM ANALYTIC OPTIMUM

Analytic optimum finds the price which results in the highest profit (or revenue) for the partiuclar item. The optimum price is easy to calculate. Recall the equation for revenue (or profit).

We will ignore all cross-elastic price terms, leaving only the item's own price in the equation. This gives us the following much simplified equation

$$\begin{aligned} rev_i &= (price_i w_i + c) \cdot price_i \\ &= w_i price_i^2 + cprice_i \end{aligned}$$

$$\begin{aligned} prof_i &= (price_i w_i + c) \cdot (price_i - \cos t_i) \\ &= w_i price_i^2 + cprice_i - w_i price_i \cos t_i - c \cos t_i \end{aligned}$$

Now we can take the derivative of this equation with respect to the item's own price.

$$\frac{drev_i}{dprice_i} = 2 price_i w_i + c$$

$$\frac{dprof_i}{dprice_i} = 2 price_i w_i + c - w_i \cos t_i$$

An optimum for the revenue (profit) equation will occur when this derivative is 0. Therefore, we set this derivative to 0 and solve for pricei at this point.

$$at0,\ price_i^{\ rev} = \frac{-c}{2w_i}$$
$$= -cons \cdot \frac{1}{2} diag(W)^{-1}$$

$$at0,\ price_i^{\ prof} = \frac{w_i \cos t_i - c}{2w_i}$$
$$= \left(\cos t \cdot diag(W) - cons\right) \cdot \frac{1}{2} diag(W)^{-1}$$

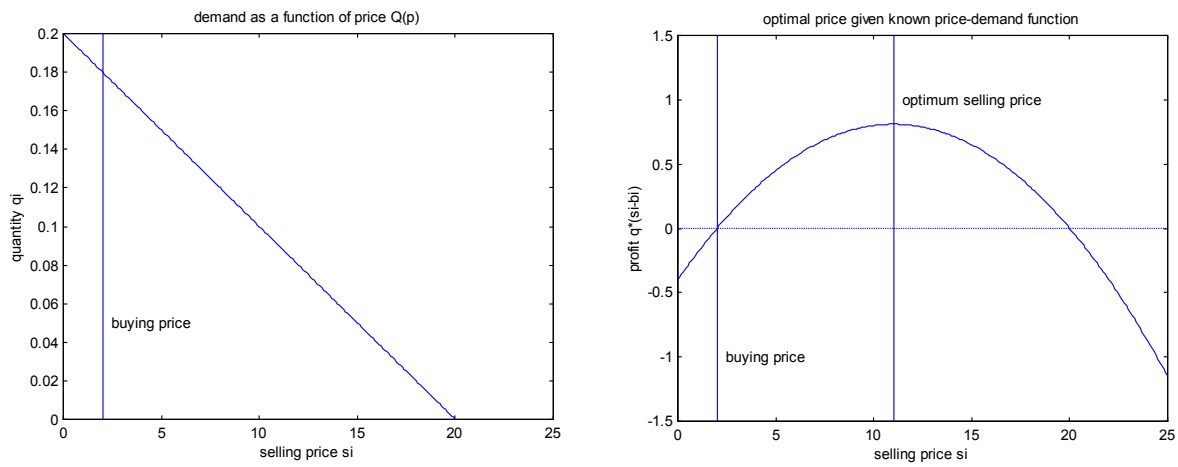


Figure: (left) Graph of quantity versus selling price (price-demand, estimated). (right) Graph of profit versus selling price. An optimum can be found analytically after we estimate the price-demand function. *b=2;s=[0:0.1:25];m=-0.01;c=0.2;profit=(s-b).*(m.*s+c); sopt=(b.*m-c)./2.*m;*

A final step was to constrain prices not to exceed minimum or maximum historical values. This prevents wild values from being generated. It would be easy to relax this if desired, for instance, allowing values to be within 10% - or 20% of the minimum and maximum values. For consistency of comparison, all models tested in this paper, if a price exceeds its minimum or maximum, it is set to the minimum or maximum value.

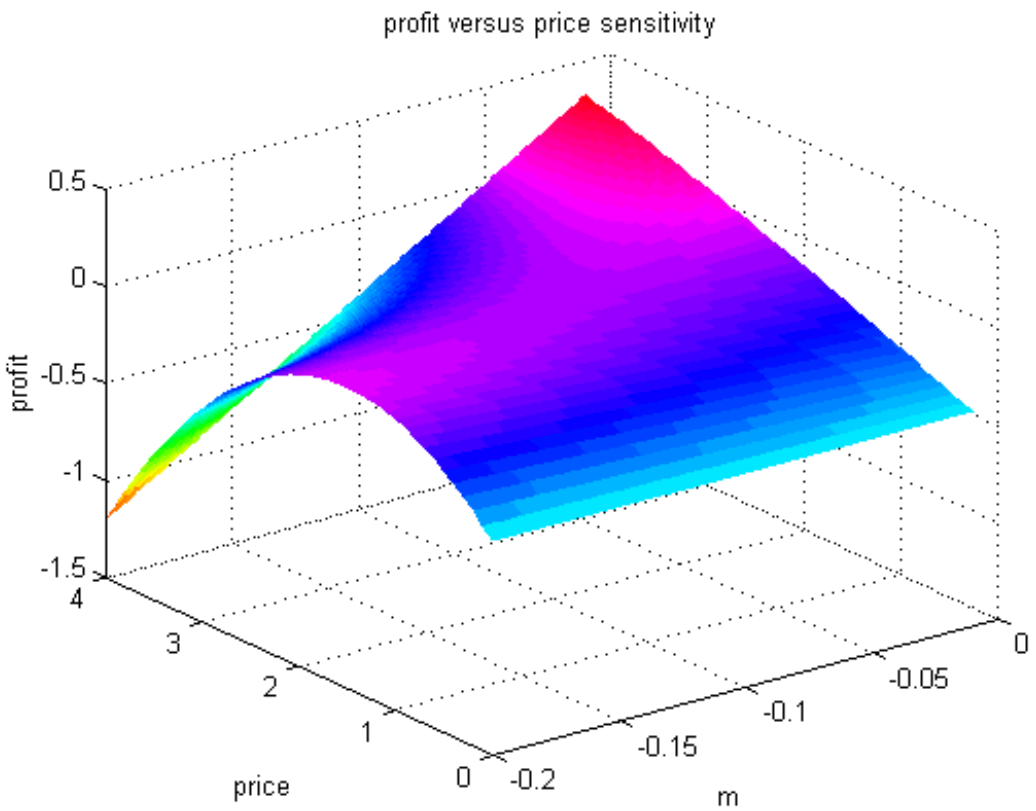


Figure: Profit versus selling price and demand gradient. The quadratic nature of profit can be seen clearly in a slice from the side. As m gets flatter, (consumers are less affected by price increases), the selling price optimum moves farther out.

GLOBAL OPTIMIZATION

Single item optimization finds the best price of an item without regard to its effect on other items in the store. For example, say that 2 Litre Tropicana Orange Juice sells for $2.50. Single item optimization might predict an optimal profit on Tropicana alone, at a price at $2.60. However, even if were true that $2.60 led to the best profit on Tropicana, this optimum doesn't take into account the store-wide implications of making that price change. Perhaps Tropicana Orange juice at the lower price draws in important general sales in the store. In this case, Tropicana might contribute more by actually dropping its price and becoming a consumer drawcard.

The complete demand model which proved to be so accurate in section 1 is used for this multiple, cross-elastic price optimization. The implications of using this model are fairly important. Firstly it is more accurate than the single item-price demand model (Correlation of 5% versus 55%), so the optimization predictions will hopefully be more accurate also. Secondly, the model is completely aware of interactions between a price change on one product, and the impact on up to 40,000 other products in the store. It can conceivibly home in on critical items in which price should be dropped very low, for the greater good of the store. This kind of savvy optimization has a lot of potential to find revenue opportunities which human beings mightn't see purely because

of the number of items involved, and the fact that interactions occur across category boundaries (something, again, the model has no problems with).

Unfortunately, unlike the single item optimization case, there is no closed form solution to the optimum price vector for the store. Profit optimization becomes a large quadratic programming problem, which must be solved using numerical techniques.

We will explore several methods for attacking this problem over the following pages. Gradient descent adjusts price in the direction of the gradient to increase the predicted profit. Newton's method is a faster method which uses the second derivative to jump to price vectors at which the derivative is close to 0. Stochastic and direct methods ignore gradient information and instead rely on various heuristics to find good price vectors. We have also proposed a hybrid method which uses both gradient information, and discrete pricepoints to rapidly skip through the price search space.

In all of the optimization methods below, price values were constrained to move within their minimum and maximum historical values. Therefore, if a method used gradient descent, and one of the prices increased beyond its upper bound, that price was set to the upper bound, (meaning no movement for that dimension), and the perhaps smaller movement in the other dimensions was allowed, meaning the optimization slid along the boundary in its allowable directions.

GRADIENT DESCENT

Gradient descent uses the first derivative of the profit (or revenue) function to guide the algorithm on which direction to move to increase profit (revenue).

$$\frac{drev}{dprice_i} = P\left(W + W^T\right) + 1_{n,1} \cdot c$$

$$\frac{dprof}{dprice_i} = \frac{drev}{dprice_i} - C \cdot W$$

There are many variants of gradient descent.

<u>Line search</u>

Instead of calculating a weight adjustment each iteration, setting line search to true will allow the algorithm to find weight change once, and then continue to update weights in the same direction, until error starts increasing. This decreases the number of costly weight change calculations which need to be performed, and so speeds up the algorithm.

<u>Conjugate gradient minimization</u>

The conjugate gradient has been termed by Masters (1994) one of the “Seven miracles of modern mathematics”. Instead of moving in the direction of steepest descent, this keeps track of the previous direction of movement. It then estimates a new direction which is conjugate to the previous directions. For example, if the algorithm is zig-zagging down a ravine, then the conjugate gradient algorithm will choose a direction straight down the middle of the zig-zag. As a result, the algorithm can result in tremendous speed improvements.

Adaptive step size

The learning constant in backpropagation and many other numerical packages is fixed. A common change is to allow the step-size to increase by a constant factor, as long as error is still dropping. If error increases after an iteration, this means the algorithm has overshot a minima, and has moved to a worse location. When this happens, the learning rate is braked at a geometric rate, to make it small enough to again follow the gradient down into the basin of the minima.

Momentum

Momentum was on of the first and most common modifications to backpropagation. Instead of choosing the direction of steepest descent, it factors in the previous direction also, and moves in this combined direction. The greater the momentum, the greater the emphasis on the previous search direction. This causes the algorithm to have a kind of inertia which pushes it over local minima (it barrels straight past those small basins, since it is relying on previous search direction).

NEWTON'S METHOD

Newton's method uses the second derivative of the profit (revenue) function to provide data on where the first derivative is zero. Let's say the first derivative is given by the equation, y=mx+c. Since our profit (revenue) function is a quadratic, and the first derivative is linear, this is actually a very good assumption! When the first derivative to 0, the equation is:

$$y = mx + c$$
$$c = y - mx$$

$$\begin{aligned} aty = 0, x &= \frac{-c}{m} \\ &= \frac{-(y - mx)}{m} \\ &= x - \frac{y}{m} \\ &= x - \frac{d^2 prof}{dprice_i^{\,2}}^{-1} \cdot \frac{dprof}{dprice_i} \end{aligned}$$

Therefore, given that we know the second derivative of both profit and revenue with respect to pricei are

$$\frac{d^2 rev}{dprice_i^{\,2}} = \frac{d^2 prof}{dprice_i^{\,2}} = W + W^T$$

a price which makes the gradient equal to 0 can be found by performing

$$P = P - \left(W + W^T\right)^{-1} \cdot Q$$

Since the inverse Hessian is calculated numerically and can have errors, repeated application of this adjustment will move the price closer and closer to the true point at which the first derivative is 0.

INTEGER PROGRAMMING AND STOCHASTIC METHODS

It is also possible to represent the problem of finding prices for each item as an integer programming problem, where legal values are the pool of discrete prices items were tested at in the past. It is also possible to use simulated annealing, dynamic hillclimbing, or other methods, however, these methods tend to run much slower than the gradient descent and Newton methods described previously, and as a result, we will ignore them in these experiments.

EXPERIMENT

Design of Experiment

In order to test the effectiveness of the price optimization method, we ran a live experiment. Our participating retailer was a 9 store chain in the mid-west. The retailer identified 7 categories which they were interested in manipulating prices, comprising 430 individual items.

Of those products, we only considered an item a candidate for price optimization if it had undergone at least one price change in the past. Items with no price change carried no information on how their own-price could be adjusted to improve global profit. This made it impossible to use them as an adjustable variable. This constraint ruled out 70% of all UPCs, since they underwent no price change.

We next removed products which had known external dependencies, or had unusual demand curves which suggested external influences. For example, the retailer pointed out that price reductions in the butter category were inappropriate, as prices were directly tied to the price of butter fat, which fluctuated and drove prices at competitor stores as well. This resulted in demand curves which couldn't be trusted.

After eliminating these items, we were left with $\frac{23}{430}$ = 5% items as our final list of candidates.

Price optimization method

We ran a gradient descent procedure to determine optimal prices for each of the remaining products. We hard-limited this method so that if an optimal price was selected which was greater than the maximum historical price, the maximal historical price was chosen as the optimum. Similarly for minimum historical prices.

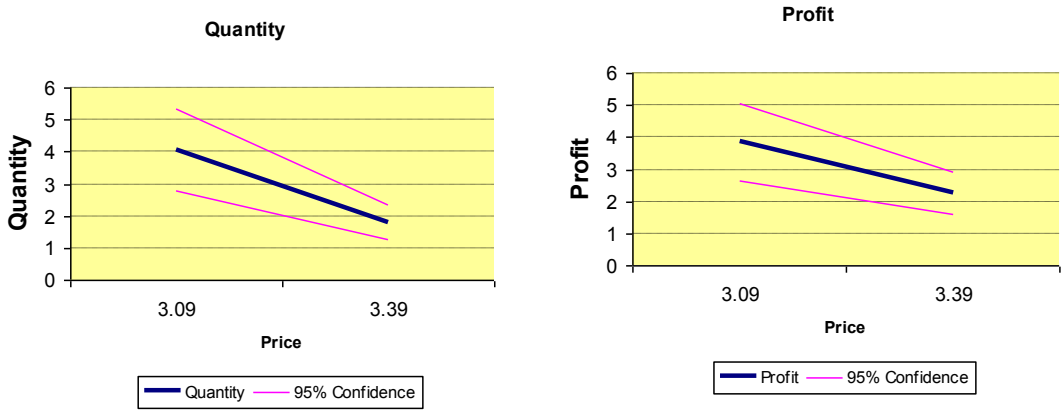
For each product, two stores were created as controls, and 2 stores for experimentation. Price changes were implemented at the experimental stores, and compared against the control stores. The price experiment was performed over 120 days.

The final price changes for each of these items is shown in table x. The price changes consisted of 12 decreases and 14 increases. The average price change was –$0.05.

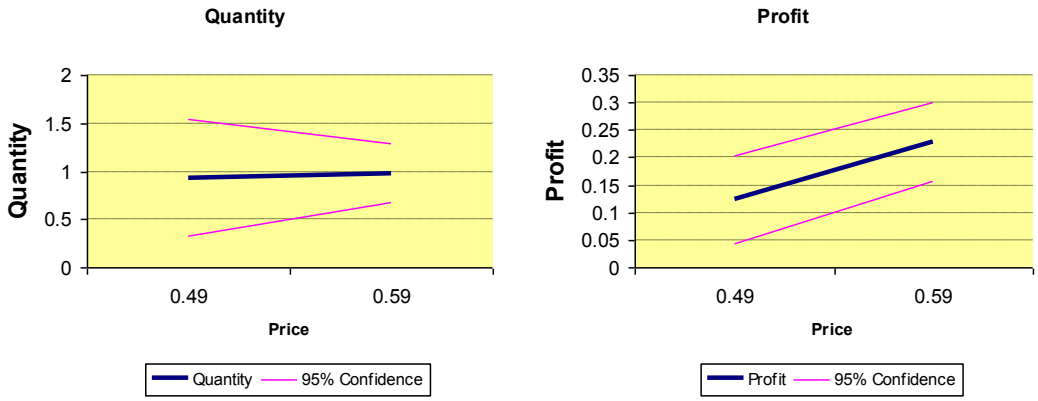
Recommended Price changes

Item description	Intervention type	Price before	Price after	Change
IGA CUT ASPARAGUS-14.5 OZ	increase	1.19	1.29	0.10
IGA 3 SV CUT GREEN BEANS-8 OZ	increase	0.39	0.43	0.04
IGA CUT GREEN BEANS-14.5 OZ	increase	0.49	0.59	0.10
IGA FRENCH STYLE GRN BEAN-14.5	increase	0.49	0.59	0.10
IGA FR STY GREEN BEANS-8 OZ	increase	0.39	0.43	0.04
IGA DICED CARROTS-14.5OZ	increase	0.49	0.59	0.10
IGA MEDIUM SLICED CARROTS-14.5	increase	0.49	0.59	0.10
IGA CREAM STYLE CORN-14.5 OZ	increase	0.49	0.59	0.10
IGA CREAM STYLE CORN-8 OZ	increase	0.39	0.43	0.04
IGA WHOLE KERNEL CORN-15.25	increase	0.49	0.59	0.10
IGA WHOLE KERNEL CORN-8.0 OZ	Increase	0.39	0.43	0.04
IGA MIXED SWT PEAS-15 OZ	Drop	0.59	0.49	-0.10
IGA SLICED POTATOES-15 OZ	increase	0.59	0.67	0.08
IGA WHOLE POTATOES-15 OZ.	increase	0.59	0.67	0.08
SKIP CHUNK PEANUT BUTTER-18 OZ	Drop	2.19	2.05	-0.14
SKIP CRMY PEANUT BUTTER-18 OZ	Drop	2.19	2.05	-0.14
SKIPPY R.FAT CHUNKY P.BUTTER-1	Drop	2.19	2.05	-0.14
SKIPPY R.FAT CREAMY P.BUTTER-1	Drop	2.19	2.05	-0.14
TROP PURE PREM ORANGE JCE-64 O	Drop	3.39	3.09	-0.30
TROP PURE PREM HOMESTYLE-64 OZ	Drop	3.39	3.09	-0.30
TROP PURE PREM GROVESTAND-64 O	Drop	3.39	3.09	-0.30
TRP PURE PREM + CALCIUM-64 OZ	Drop	3.39	3.09	-0.30
			Average	**-0.04**

Tropicana Pure Premium Homestyle – 64 Oz



IGA Diced Carrots – 14.5 Oz



IGA Cream Style Corn – 14.5 Oz

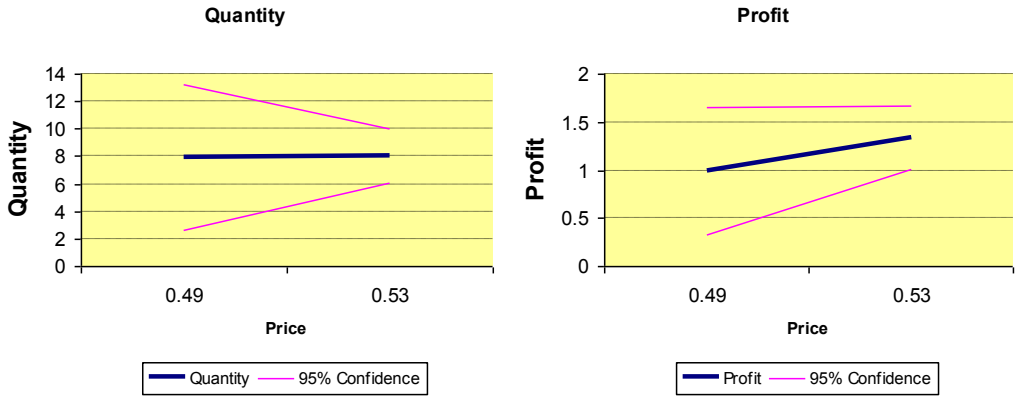
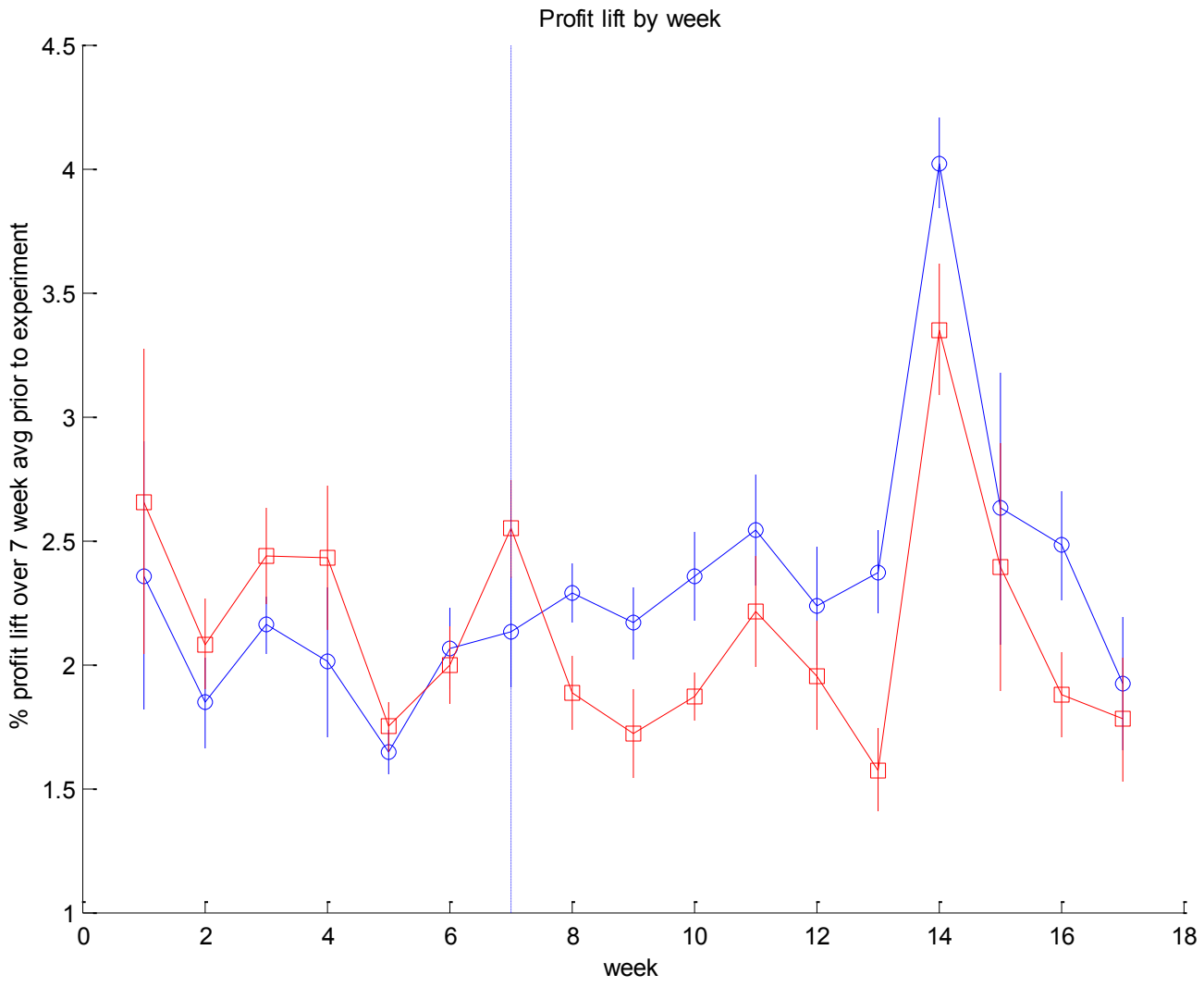


Figure: d df

Profit change due to intervention

descript	intervention type	profbefore control	profbeforestd control	profduring control	profafterstd control	profbefore exp	profbeforestd exp	profduring exp	profafterstd exp	%profchange	%controlchange	absolute diff
G MILLS CIN TOAST CRUNCH-14 OZ	drop	2.63	5.95	1.66	2.71	3.33	6.80	0.65	0.70	-80%	-37%	-1.71
KEL CRISPIX-12 OZ	drop	1.36	1.16	0.96	0.80	1.12	0.97	0.52	0.49	-54%	-29%	-0.20
KEL FST MINI WHEATS-24.3 OZ	increase	0.83	0.94	1.08	0.94	1.37	0.96	1.82	1.45	33%	30%	0.20
KELLOGG RICE KRISPIES-13.5 OZ	drop	1.68	1.27	1.64	1.47	2.11	2.24	1.37	1.19	-35%	-2%	-0.71
IGA CUT ASPARAGUS-14.5 OZ	increase	0.50	0.50	0.43	0.49	0.37	0.45	0.53	0.68	45%	-13%	0.23
IGA 3 SV CUT GREEN BEANS-8 OZ	increase	0.13	0.15	0.11	0.13	0.09	0.12	0.14	0.17	53%	-14%	0.07
IGA CUT GREEN BEANS-14.5 OZ	increase	1.81	3.25	2.07	2.88	1.85	2.49	2.90	3.78	56%	15%	0.78
IGA FRENCH STYLE GRN BEAN-14.5	increase	2.31	3.00	1.20	1.70	1.57	1.88	1.65	1.24	5%	-48%	1.19
IGA FR STY GREEN BEANS-8 OZ	increase	0.06	0.11	0.07	0.11	0.06	0.10	0.09	0.14	43%	7%	0.02
IGA DICED CARROTS-14.5OZ	increase	0.18	0.23	0.16	0.22	0.17	0.23	0.23	0.36	37%	-6%	0.07
IGA MEDIUM SLICED CARROTS-14.5	increase	0.43	0.31	0.40	0.37	0.40	0.31	0.63	0.53	55%	-6%	0.25
IGA CREAM STYLE CORN-14.5 OZ	increase	1.52	0.99	0.81	0.61	1.25	1.15	1.28	0.90	2%	-47%	0.73
IGA CREAM STYLE CORN-8 OZ	increase	0.13	0.20	0.16	0.20	0.20	0.25	0.18	0.23	-12%	25%	-0.06
IGA WHOLE KERNEL CORN-15.25	increase	3.04	1.66	1.68	1.09	2.62	2.16	3.16	1.69	21%	-45%	1.90
IGA WHOLE KERNEL CORN-8.0 OZ	increase	0.18	0.17	0.15	0.15	0.13	0.15	0.18	0.19	42%	-17%	0.08
IGA MIXED SWT PEAS-15 OZ	drop	1.16	0.90	0.47	0.44	1.53	1.27	0.95	0.66	-38%	-59%	0.10
IGA SLICED POTATOES-15 OZ	increase	0.29	0.36	0.33	0.44	0.23	0.32	0.24	0.35	3%	14%	-0.03
IGA WHOLE POTATOES-15 OZ.	increase	0.33	0.36	0.44	0.60	0.30	0.45	0.28	0.44	-4%	36%	-0.13
SKIP CHUNK PEANUT BUTTER-18 OZ	drop	0.31	0.33	0.32	0.38	0.30	0.37	0.23	0.23	-22%	4%	-0.08
SKIP CRMY PEANUT BUTTER-18 OZ	drop	1.11	0.74	1.10	0.70	0.94	0.72	0.61	0.45	-34%	-1%	-0.31
SKIPPY R.FAT CHUNKY P.BUTTER-1	drop	0.18	0.30	0.21	0.27	0.20	0.30	0.12	0.17	-41%	15%	-0.11
SKIPPY R.FAT CREAMY P.BUTTER-1	drop	0.34	0.38	0.38	0.43	0.39	0.36	0.24	0.25	-38%	12%	-0.19
TROP PURE PREM ORANGE JCE-64 O	drop	2.74	2.42	2.90	2.57	2.31	1.96	2.54	2.33	10%	6%	0.06
TROP PURE PREM HOMESTYLE-64 OZ	drop	1.69	1.73	1.96	1.96	1.79	1.61	2.69	2.41	50%	16%	0.61
TROP PURE PREM GROVESTAND-64 O	drop	2.24	1.73	2.25	2.00	1.79	1.65	2.62	2.09	46%	0%	0.81
TRP PURE PREM + CALCIUM-64 OZ	drop	2.74	1.97	2.83	2.43	1.97	1.92	2.43	2.34	23%	3%	0.36
Total	**All**	**1.15**		**0.99**		**1.09**		**1.09**		**6%**	**-5%**	**0.15**


Profit lift by week
% profit lift over 7 week avg prior to experiment
week
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12
14
16
18

All Price Changes

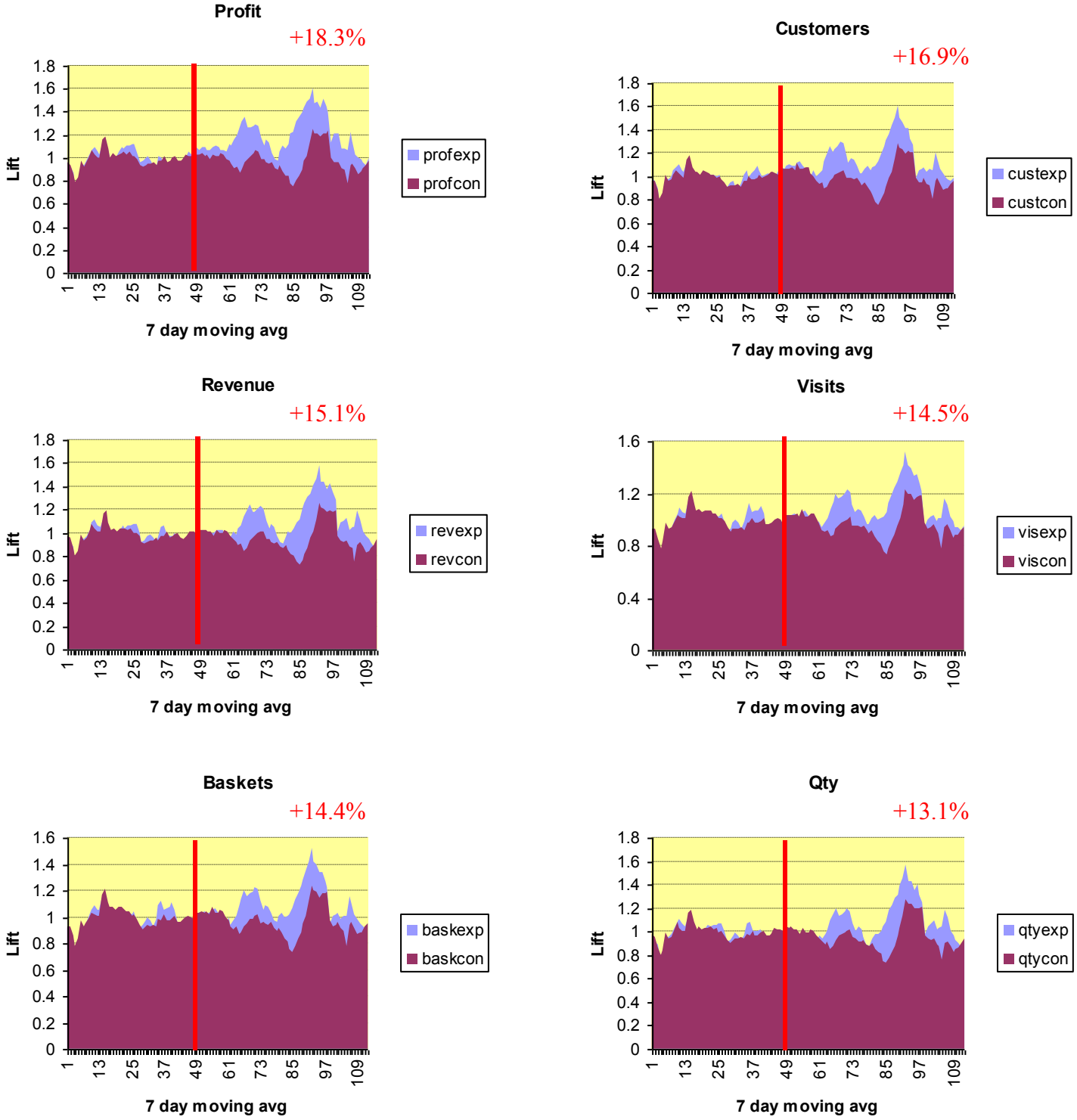
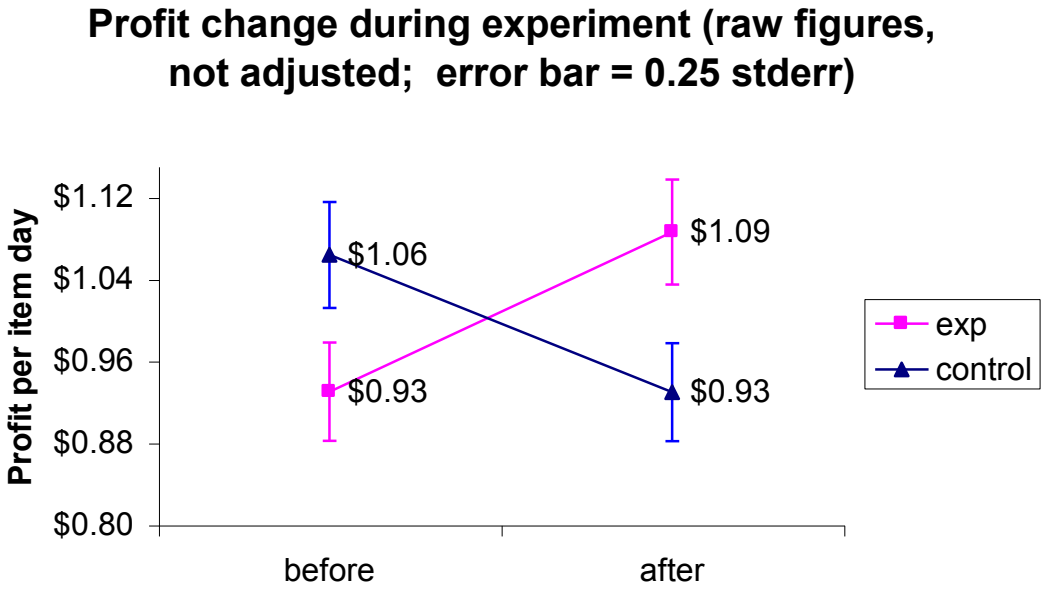
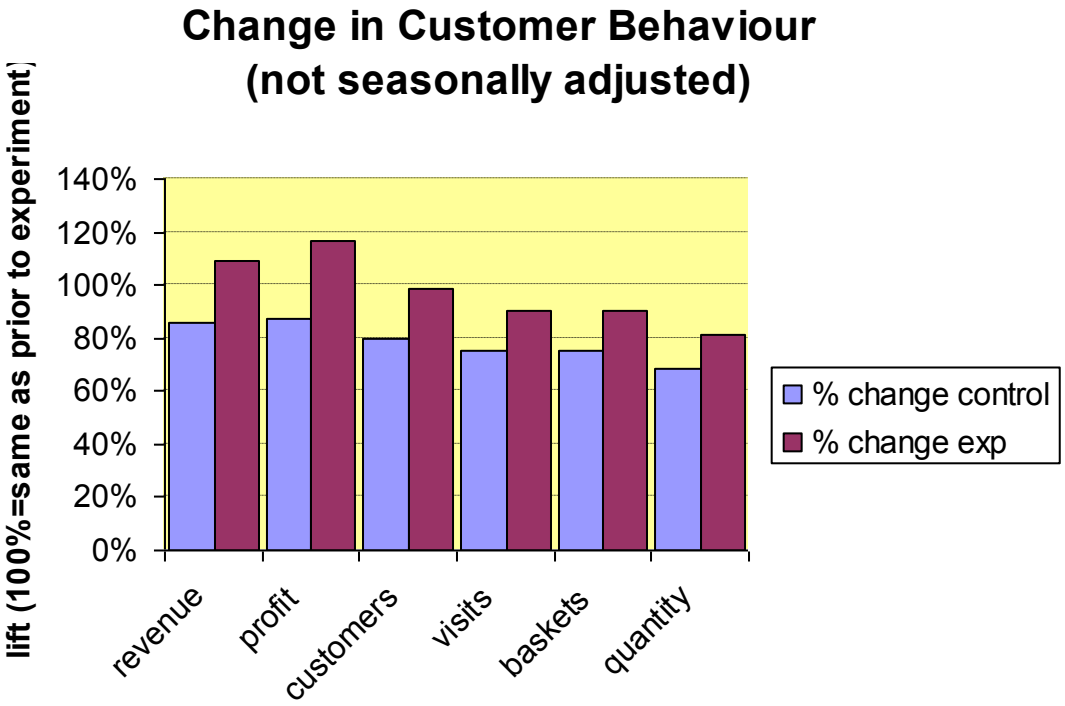


Figure:

Change in customer behaviour

Pricing	beforecontrol	aftercontrol	% change control	beforeexp	afterexp	% change exp	seasonally adjusted lift	controlbeforestd	controlafterstd	expbeforestd	expafterstd
revenue	$3.69	$3.17	86%	3.14	3.44	109%	**23.54%**	3.23	3.00	3.05	2.99
profit	$1.06	$0.93	87%	0.93	1.09	117%	**29.37%**	0.99	0.92	0.92	0.98
customers	2.56	2.04	79%	2.06	2.04	99%	**19.50%**	2.13	1.85	1.97	1.62
visits	2.76	2.07	75%	2.29	2.07	90%	**15.47%**	2.21	1.90	2.29	2.07
baskets	2.75	2.06	75%	2.29	2.07	91%	**15.52%**	2.19	1.89	2.06	1.65
quantity	4.47	3.07	69%	3.73	3.02	81%	**12.12%**	3.96	3.04	3.73	2.58




Change in Customer Behaviour
(not seasonally adjusted)
lift (100%=same as prior to experiment)
140%
120%
100%
80%
60%
40%
20%
0%
revenue
profit
customers
visits
baskets
quantity
% change control
% change exp

Results

In order to quantify the experimental changes, we introduce two metrics. The first is Lift, the second Seasonally-adjusted lift.

Lift

Lift quantifies the increase in a customer measure, measured against the score prior to the intervention. To calculate lift we divided the measure after the intervention by the measure before the intervention.

$$lift_G = \frac{mean(R_G{}^{t+1})}{mean(R_G{}^{t})} - 1$$

For instance, profit rose by an average of 18% per product in the experimental group, and decreased by an average of 13% per product in the control (negative lift).

Seasonally adjusted lift

Because we measure quantities through time, various external influences may be occurring during the measurement period. These external influences, such as seasonality, should ideally be factored out when reporting lifts. For instance, if juice increased by 10% in a control group, and 12% in an experimental group, we should normally assume that the juice would have experienced a 10% increase in either group. Therefore, the actual lift for the experimental group might be only 12% - 10% = 2%.

Seasonally adjusted lift is therefore the experimental lift minus the control lift. This results in the number of percentage points higher that the experimental group was compared to the control group.

$$seasonallyadjustedlift_G = \frac{mean(R_G{}^{t+1})}{mean(R_G{}^{t})} - \frac{mean(R_C{}^{t+1})}{mean(R_C{}^{t})}$$

In our experiment, all customer measures decreased in the control group, with profit dropping by 13%. In contrast, absolute profit in the experimental group increased by 17%. Seasonally adjusted profit lift was therefore very high at 29%.

Multiple measures of behaviour

Finally, rather than just report profit change, which was the objective of the price optimization, we will also examine the effect of our intervention on five other behavioural measures, including number of visits, number of distinct customers buying a product, number of units bought of a product, and revenue generated by product, and number of baskets with product. These additional measures are useful to determine if any of our price optimizations are “harming” other critical business metrics, such as customer visitation, which could happen as a result of price increases.

Overall Effect on Items in Experiment

In the control group, all behavioural measures declined during the experiment. Profit declined -13%, revenue -14%, visits -25%, and quantity -31%.

In the experimental group *declines were smaller* than in the control group. Further, profit and revenue *increased*. In the experimental group, profit increased by +17%, revenue +9%, visits -10%, and quantity -19%. These drops are much less severe than in the control group.

The raw profit change compared to the control group is shown in figure x. Seasonally adjusted lifts for all metrics were positive, meaning the experimental groups all performed better than the control groups. Profit showed the largest seasonally adjusted lift, with an increase of 29.37%. It is somewhat comforting that profit showed the greatest increase, since this was the objective of the experiment.

When we examine individual items, the experimental group items still do better on all measures than the control group items. In terms of number of products showing increase over control, revenue increased in 73.1% of the products affected, profit increased in 61.5%, customers 76.9%, visits 73.1%, baskets 73.1%, quantity 57.7%.

Significance tests

<u>1. Overall Magnitude of difference between experimental and control groups</u>

In order to calculate the chance that control and experimental groups are the same, (lift was due to chance), we employed a variety of statistical techniques. We applied three rank-based tests, and one t-test for comparison. We applied these tests to the lift measures described previously. Details on these methods can be found in REF. In all cases, every behavioural metric tested was significantly higher than the control group ($p<0.01$). We should therefore conclude that the improvements in the experimental group were unlikely to have been caused by chance.

<u>2. Improvement for each item</u>

We also applied tests on individual items to determine whether the profit improvement in each item could be significant. 7 items showed significant ($p<0.05$) changes in profit for the experimental group, versus 4 items in the control group.

<u>2. Number of items greater than control</u>

A second significance test was used to determine how likely it was to observe *k* products being greater than control, given that the chance of an item being higher or lower than control is random.

The binomial distribution gives the chance of observing *k* heads from a coin-flip, given *n* flips of the coin. In our case, the coin-flip is whether a product's change was higher or lower than its control. We are assuming that the control group should grow as a percentage of its baseline at the same rate as the experimental group.

$$\Pr(\text{observing} >= k \text{ outcomes})$$
$$=1-\Pr(p,0..k,n)$$

$$=1-\sum_{i=0}^{k}\left[\binom{n}{i}p^{i}(1-p)^{(1-i)}\right]$$

The probability of observing the number of increases under a binomial distribution with p=0.5, n=26, and k=the number of observed positive outcomes, is shown in table x. All behavioural indicators except profit and quantity are significant at the p<0.05 level. Profit is significant at the p<0.09 level.

Overall significance of increases in experimental group

Behavioural Measure	Ranksum p value	Signtest p value	Signrank p value	Ttest p value
Profit	0.000000	0.000000	0.000000	0.000000
Revenue	0.000000	0.000006	0.000000	0.000000
Customers	0.000000	0.000351	0.000000	0.000000
Visits	0.000126	0.000708	0.000003	0.000042
Baskets	0.000002	0.000708	0.000000	0.000002
Quantity	0.000124	0.000351	0.000001	0.000027

Effect of intervention on items in experiment

Pricing	no items > control (out of 23)	significance
revenue	18	0.04%
profit	15	2.62%
customers	18	0.04%
visits	18	0.04%
baskets	18	0.04%
quantity	14	6.69%

Figure: Effect on customer behaviour, including profit, number of visits to store per day, spending, and quantity purchased per day.

Items showing significant increase on six behavioural measures

Item	revsig control	revsig exp	profsig control	profsig exp	custsig control	custsig exp	visitssig control	visitssig exp	basketsig control	basketsig exp	qttysig control	qttysig exp
IGA CUT ASPARAGUS-14.5 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA 3 SV CUT GREEN BEANS-8 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA CUT GREEN BEANS-14.5 OZ	1	0	0	0	1	0	1	1	1	1	1	1
IGA FRENCH STYLE GRN BEAN-14.5	1	0	1	0	1	0	1	1	1	1	1	1
IGA FR STY GREEN BEANS-8 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA DICED CARROTS-14.5OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA MEDIUM SLICED CARROTS-14.5	0	0	0	1	0	0	0	0	0	0	0	0
IGA CREAM STYLE CORN-14.5 OZ	1	1	1	0	1	1	1	1	1	1	1	1
IGA CREAM STYLE CORN-8 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA WHOLE KERNEL CORN-15.25	1	0	1	1	1	0	1	1	1	1	1	1
IGA WHOLE KERNEL CORN-8.0 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA MIXED SWT PEAS-15 OZ	1	1	1	1	1	0	1	1	1	1	1	1
IGA SLICED POTATOES-15 OZ	0	0	0	0	0	0	0	0	0	0	0	0
IGA WHOLE POTATOES-15 OZ.	0	0	0	0	0	0	0	0	0	0	0	0
SKIP CHUNK PEANUT BUTTER-18 OZ	0	0	0	0	0	1	0	0	0	0	0	1
SKIP CRMY PEANUT BUTTER-18 OZ	0	0	0	1	0	0	0	0	0	0	0	0
SKIPPY R.FAT CHUNKY P.BUTTER-1	0	0	0	0	0	0	0	0	0	0	0	0
SKIPPY R.FAT CREAMY P.BUTTER-1	0	1	0	1	0	0	0	0	0	0	0	0
TROP PURE PREM ORANGE JCE-64 O	0	0	0	0	0	1	0	1	0	1	0	1
TROP PURE PREM HOMESTYLE-64 OZ	0	1	0	1	1	1	0	1	0	1	0	1
TROP PURE PREM GROVESTAND-64 O	0	1	0	1	0	1	0	1	0	1	0	1
TRP PURE PREM + CALCIUM-64 OZ	0	0	0	0	0	1	0	1	0	1	0	1
Sum	**5**	**5**	**4**	**7**	**6**	**6**	**5**	**9**	**5**	**9**	**5**	**10**

Cannibalization

The profit increases reported have been for the individual items that we influenced during the experiment. However, maybe those improvements came at the "cost" of other items in the category.

For example, our price drop on Tropicana juice might have increased the sales of Tropicana, but decreased the sales of Florida's natural juice. This is because Florida juice customers started switching over and buying the cheaper Tropicana juice. This would have given the illusion that Tropicana was more profitable, when the category level volume and profit may have been unchanged or even declined. Therefore, we needed to determine whether our results were affected by cannibalization.

Our technique for quantifying cannibalization is to examine the profitability of categories in which products were either changed or held the same. If the profitability of the category stays the same or decreases, cannibalization could have occurred.

We examined 4 stores. In 2 stores all products were experimental, and in the other 2 all products were controls. There was no mixed effect at the category level. The total expected lift in revenue, assuming no interactions arising from our price intervention, is therefore equal to

$$ExpectedLift_{cat} = ItemContribution_{cat} \cdot ItemLift_{cat}$$

We next examined the actual observed lift at the category level, where lift is defined as the percentage change in experimental group minus percentage change in control group.

$$ActualLift_{cat} = Lift_{cat,ex} - Lift_{cat,control}$$

If ActualLift is lower than the expected lift, cannibalization will be assumed to have taken place. The ActualLift and expected lifts are shown in table x. In all categories except one (Peanut butter), ActualLift exceeded ExpectedLift. Therefore we will conclude that for 5 of 6 categories, no cannibalization was apparent, and in fact, the reverse – increases in profit due to follow-on affects – may have been taking place. Overall the contribution of the experiment to global store profitability seems to have been overwhelmingly positive.

Revenue Cannibalization

Category	control % change	Exp % change	Actual lift at cat level%	expected lift at cat level%	Excess lift%
CANNED - CORN	75.59%	83.33%	7.75%	0.22%	7.53%
CANNED - GREEN & WAXED BEANS	79.45%	84.09%	4.64%	0.20%	4.44%
CANNED - MISC. VEGETABLES	84.06%	89.51%	5.45%	0.02%	5.42%
CANNED - PEAS	78.94%	90.38%	11.44%	0.02%	11.41%
PEANUT BUTTER	97.39%	95.44%	-1.96%	0.00%	-1.96%
REFRIGERATED JUICES & CITRUS	90.47%	93.71%	3.24%	0.05%	3.19%

APPENDIX A

Analytic Derivatives for qty, rev, and profit

Quantity derivatives

Lemma 1: $\frac{d}{dP} PW = W^T$

Proof

Let qi=the ith column sum from P to W. The following holds:

dqi/dpj = wji

Therefore, the derivative of price at position j across is equal to the weight at position j down, and position 1..n across. This applies for all I,j combinations. Therefore, the derivative matrix of dq/dp is the transpose of W.

Lemma 2: dqtyi/dpricej = $I_{1,n}.W^T$

Revenue Derivatives

Lemma 3: Revenue derivative of an item i with respect to a change in its own price $\frac{drev_i}{dprice_i} = w_i\, price_i + qty_i \;\; = P(I \cdot diag(W)) + P \cdot W + 1_{n,1} \cdot c$

Proof

$$\begin{aligned}
\frac{drev_i}{dprice_i} &= \partial / \partial price_i \left[w_i\, price_i^{\,2} + w_j\, price_j\, price_i + \ldots + cprice_i \right] \\
&= 2w_i\, price_i + w_j\, price_j + \ldots + c \\
&= w_i\, price_i + \left(w_i\, price_i + w_j\, price_j + \ldots + c \right) \\
&= w_i\, price_i + qty_i
\end{aligned}$$

We can also express the revenue derivative revi/pricei in matrix form. The provides a compact representation of revenue derivatives. If we take the derivative of revi with respect to pricei, for each i, then we end up with a matrix of derivatives for each item i (ie. drevi/dpricei for each item i).

$$\frac{drev_i}{dprice_i} \forall i = \begin{bmatrix} \frac{drev_1}{dprice_1} & \frac{drev_2}{dprice_2} & \ldots & \frac{drev_n}{dprice_n} \end{bmatrix}$$

$$\begin{aligned}
\frac{drev_i}{dprice_i} &= \left[(2w_1 price_1 + w_{.} price_{.} + \ldots + c) \;\; (2w_2 price_2 + w_{.} price_{.} + \ldots + c) \;\; \ldots \;\; (2w_n price_n + w_{.} price_{.} + \ldots + c) \right] \\
&= \left[w_1 price_1 \quad w_2 price_2 \quad \ldots \quad w_n price_n \right] + \left[qty_1 \quad qty_2 \quad \ldots \quad \right]
\end{aligned}$$

Substituting Q = P.W+1.c (equation x) we have

$$\frac{drev_i}{dprice_i} = \begin{bmatrix} w_1 price_1 & w_2 price_2 & \dots & w_n price_n \end{bmatrix} + PW + 1_{n,1} \cdot c$$
$$= P \cdot \left(I \cdot diag(W) \right) + PW + 1_{n,1} \cdot c$$

<u>Lemma 4:</u> $\frac{drev}{dprice_i} \forall i = P \cdot W^T + P \cdot W + 1_{n,1} \cdot c$

Proof

Total revenue across the store is the sum of all revenues accrued from individual items.

$$rev = \sum_{i=1}^{n} rev_i = \sum_{i=1}^{n} (w_i price_i^{\ 2} + w.price. + \ldots + c)$$
$$= \begin{array}{l} w_1 price_1^{\ 2} + w.price. + \ldots + c_1 \\ + w_2 price_2^{\ 2} + w.price. + \ldots + c_2 \\ + w_3 price_3^{\ 2} + w.price. + \ldots + c_3 \\ + w_4 price_4^{\ 2} + w.price. + \ldots + c_4 \\ \ldots \end{array}$$

If we take a the derivative of this total sum with respect to pricei, then we have the following matrix of deriviatives (one derivative for each i)

$$
\begin{aligned}
\frac{drev}{dprice_i} &= \begin{bmatrix} \frac{drev}{dprice_1} & \frac{drev}{dprice_2} & \frac{drev}{dprice_3} \end{bmatrix} \\
&= \begin{bmatrix} \frac{d}{dprice_1}\sum_{j=1}^{n} rev_j & \frac{d}{dprice_2}\sum_{j=1}^{n} rev_j & \frac{d}{dprice_3}\sum_{j=1}^{n} rev_j \end{bmatrix} \\
&= \begin{bmatrix} \frac{d}{dprice_1}\left(rev_1 + rev_2 + \ldots + rev_n\right) & \frac{d}{dprice_2}\left(rev_1 + rev_2 + \ldots + rev_n\right) & \frac{d}{dprice_3}\left(rev_1 + rev_2 + \ldots + rev_n\right) \end{bmatrix} \\
&= \begin{bmatrix} \frac{drev_1}{dprice_1} & \frac{drev_1}{dprice_2} & \frac{drev_1}{dprice_3} \\ +\frac{drev_2}{dprice_1} & +\frac{drev_2}{dprice_2} & +\frac{drev_2}{dprice_3} \\ +\frac{drev_3}{dprice_1} & +\frac{drev_3}{dprice_2} & +\frac{drev_3}{dprice_3} \\ +\ldots & +\ldots & +\ldots \end{bmatrix} \\
&= \begin{bmatrix} 2w_1 price_1 + w.price. + \ldots + c_1 & w.price_1 & w.price_1 \\ +w.price_2 & 2w_2 price_2 + w.price. + \ldots + c_2 & +w.price_2 \\ +w.price_3 & +w.price_3 & 2w_3 price_3 + w.price. + \ldots + c_3 \\ +\ldots & +\ldots & +\ldots \end{bmatrix} \\
&= \begin{bmatrix} w_1 price_1 + w.price. + \ldots + c_1 & w_2 price_2 + w.price. + \ldots + c_2 & w_3 price_3 + w.price. + \ldots + c_3 \end{bmatrix} \\
&\quad + \begin{bmatrix} w_1 price_1 & w.price_1 & w.price_1 \\ +w.price_2 & w_2 price_2 & +w.price_2 \\ +w.price_3 & +w.price_3 & w_3 price_3 \\ +\ldots & +\ldots & +\ldots \end{bmatrix} \\
&= PW^T + Q = PW^T + PW + 1_{n,1} \cdot c
\end{aligned}
$$

Profit derivatives

<u>Lemma 5:</u> $\frac{dprof}{dprice_i} \forall i = \frac{drev}{dprice_i} - C \cdot W$

Proof

$$
\begin{aligned}
&\frac{dprof}{dprice_i} \forall i = \frac{d}{dprice_i}(rev - qty \cdot \cos t) \\
&= \frac{drev}{dprice_i} - \left(PW + 1_{n,1} \cdot c\right) \cdot C
\end{aligned}
$$

Appendix B

CONFOUNDS

Production or storage cost

The optimization of profit presented in the previous section ignores a variety of other quantities which affect profitability. For instance, it is also common in Economics to attach a cost for storing or producing the items which are being sold. This function *C(q)* is a function of the quantity of items which are being held at the store. According to a survey by Wied-Nebbeling (1975), 37% of companies have a linear cost function, and 52% an exponentially decreasing function (economies of scale). In retail, there are many incidental areas of cost, including floor, shelf, or firdge space, utilities, and so forth. In the work presented, we have ignored these costs in profit optimization.

Market expansion or population growth

If you expand your patronage, or just the population grows by a few percent, the mean qty purchased will increase. This plays havoc with the elasticity, incorrectly suggesting you have increasing elasticities (when in fact the patrons might have been reducing their individual purchases and going to competitors). This effect is very marked in the simulator, where we can see the results of increased market share in a very short space of time.

Inflation

If price increased over time, again you would see increasing elasticities, when actually patronage has remained the same but just bought at the same rate at the higher inflationary price.

Seasonality

Seasons will also hurt us, for example, especially high prices can be achieved over Christmas. This might lead us to think you can increase the price of a seasonal item during January, but this isn't the case because its high price was a special price only maintainable during Christmas. The elasticities for seasonal items are only comparable within comparable seasons.

Industry-wide price increases

Accounting for Population growth

We can account for greater populations by normalizing the qties purchased by the number of people now travelling to the store. This amounts to a smoothed timeseries of the number of customers, over a very long
period. For example, an average number of patrons per six months. This is just a count of all customers, grouped by six month intervals - easy to compute. Once we have this, we assume this is the baseline population travelling to the store. We then normalize the mean_qty by this total number. Again, this should be adjustable within the CRM interface, as we will want to experiment to find out what the best setting is (or no setting as the case may be).

Accounting for Inflation

Inflation should be entered at the CRM interface, and then used to uniformly scale all prices. Scaling prices by an inflation factor will change the gradients.

Accounting for Seasonality

Goods that have a seasonal aspect need to be keyed with a mode which indicates a different season they could be in, so that we can compare those products for the current season. So for each item, we have an elasticity for Christmas period, and for other period. Or for each item, we have an elasticity for Summer, winter, and other periods.

Season codes can be typed in by the user, or we could also use algorithms to try to identify significantly different prices during the year, which can define the seasons of the product. I favour the manual typing in of seasons, since this builds expert category manager knowledge into the system.

Accounting for monopolistic price rises

A record needs to be kept of these price increases, and when the occurred. Then we can add those deltas to the historical price, so that the current set of prices are consistent with the past.

APPENDIX C

Experimental Confounds: Intervention itself caused the improvements

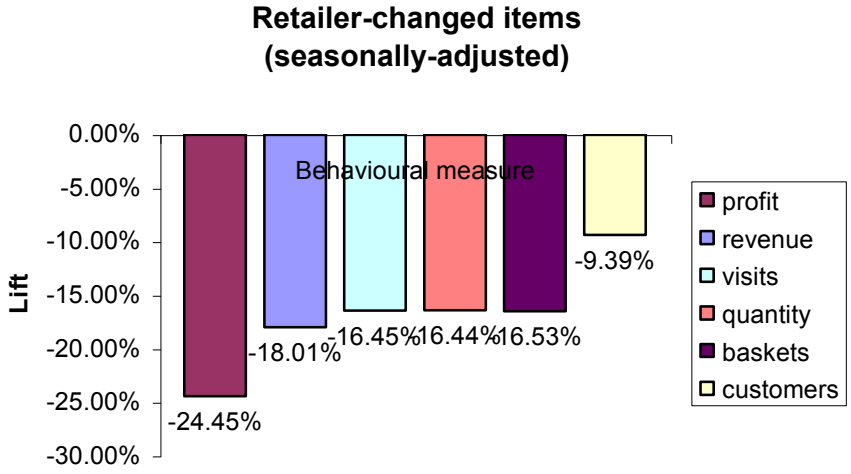
A potential confound is that the practice of changing the price on certain items, irrespective of the price selected, might have led to increase sales behaviour. We controlled for this problem in the following way. At the same time as the experimental price changes were implemented, the retailer also implemented price changes on 4 other items in the Cereal category. These price changes included increases and decreases, and were selected by the retailer.

It is possible that other price changes were implemented during the same period, but we are unaware of those. These cereal changes were implemented specifically in connection with our experiment.

Price changes (implemented by retailer, not optimal)

Item description	Intervention type	Price before	Price after	Change
G MILLS CIN TOAST CRUNCH-14 OZ	Drop	3.49	3.23	-0.26
KEL CRISPIX-12 OZ	Drop	3.19	2.98	-0.21
KEL FST MINI WHEATS-24.3 OZ	Increase	3.39	3.69	0.30
KELLOGG RICE KRISPIES-13.5 OZ	Drop	3.49	3.19	-0.30
Total	All			**-0.27**

These changes resulted in universal drops in profitability, revenue, and sales.



APPENDIX D

Experimental Confounds: Influence of Exogenous Store Effects on Experiment

It is possible that the increases reported were due to promotions that were run at the stores during the experiment. If not all stores ran exactly the same promotions, this could have meant that some stores showed increases in customers due to exogenous factors.

This scenario contradicts a fundamental design of experiment assumption, that experimental and control stores were identical except for the price intervention. However, if that assumption was violated, could we still say something about the effectiveness of the promotion?

Even with store-inhomogeneity, the experimental intervention still contributed to some percentage of lift. This can be argued based on the following fact: experimental and control items were distributed across several different stores. Therefore, if some stores underwent exogenous promotions, that would have affected both control and experimental items that were defined at that store. Exogenous store factors could not have been responsible for *all* of the lift.

Exogenous store factors could be factored out completely by randomly allocating experimental and control items to each of the 8 stores. We will call this "optimal mixing" or maximum entropy. Unfortunately, this was not done due to practical reasons, such as the need for adjusting prices of all canned goods at the same store. The actual number of experimental and control items allocated to each store is shown in table x.

To quantify the degree of mixing, we can use entropy which is defined as:

$$E_{observed} = -\sum_{s} p_s \ln(p_s)$$

where $p_s = \frac{\#ex_s}{\#control_s + \#ex_s}$ is the proportion of experimental items versus control items found at store *s*. Because the number of items involved in the experiment were different at each store, we will also weight this entropy measure by the number of items in each store to arrive at a measure we will call weighted entropy:

$$EW_{observed} = -\sum_{s} \left(n_s \Big/ \sum_{i} n_i \right) \cdot p_s \ln(p_s)$$

Maximum entropy (perfect random mixing between stores) will only be achieved if each store has exactly half of its items tagged as experimental, and half tagged as controls. Since we had 8 stores, maximum entropy will be exactly

$$\begin{aligned} EW_{\max} &= -\sum_{8} 0.5 \ln(0.5) \\ &= -4 \cdot \log_2(0.5) \\ &= -4 \end{aligned}$$

Higher values of entropy mean imperfect mixing. We can define MixingPerfection as the observed entropy expressed as a percentage of maximum entropy, or

$$Mixing = \frac{EW_{observed}}{EW_{\max}}$$

Mixing is equal to –0.2185/-4 = 5%. Therefore, in the worst possible scenario, 5% of the lift will still have been caused by non-store affects. Given a 17% lift in profit, we can therefore conclude that 1% lift would still have been caused by price intervention.

store	control items	exp items	% exp	number items	weighting
1401	15	33	69%	48	19.83%
1402	0	19	100%	19	7.85%
1403	25	23	48%	48	19.83%
1404	0	25	100%	25	10.33%
1406	4	21	84%	25	10.33%
1408	4	25	86%	29	11.98%
1409	27	0	0%	27	11.16%
1410	21	0	0%	21	8.68%
			Total storeitems	242	
			E	-1.276111	
			EW	-0.218539	
			Emax	-4	
			MixE	31.90%	
			MixEW	5.46%	

Appendix D

Item timeseries

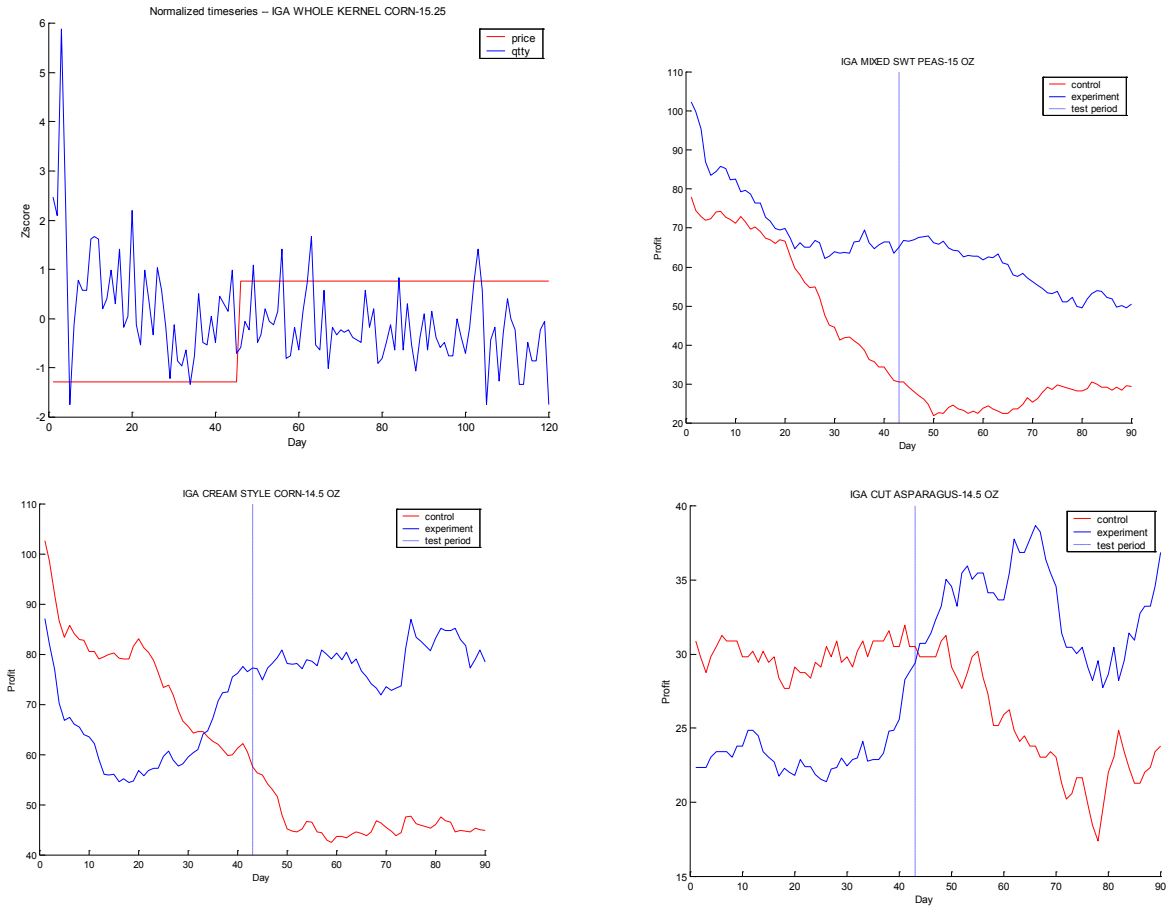


Figure: Example of a successful price change in canned goods. **(top left)** profit at experimental store plotted against price level at experimental store. Both timeseries have been converted to zscores to show on the same scale. This shows that the price was increased around day 42 (week 6) of the experiment. The other canned goods in this figure also experienced similar increases. **(top right, bottom left, bottom right)** 7 day moving average of profit series (which is why the price change on day 42 appears to be late) for three canned goods. In many of these products the demand is declining initially in both experimental and control stores. Possibly this is a seasonal change arising because the store is moving into a summer period in which canned vegetable sales are lower. When the price change comes into effect, the demand for canned goods continues to decline or be low in the experimental store, but because of the profit increase, the profit margin jumps to a higher level.

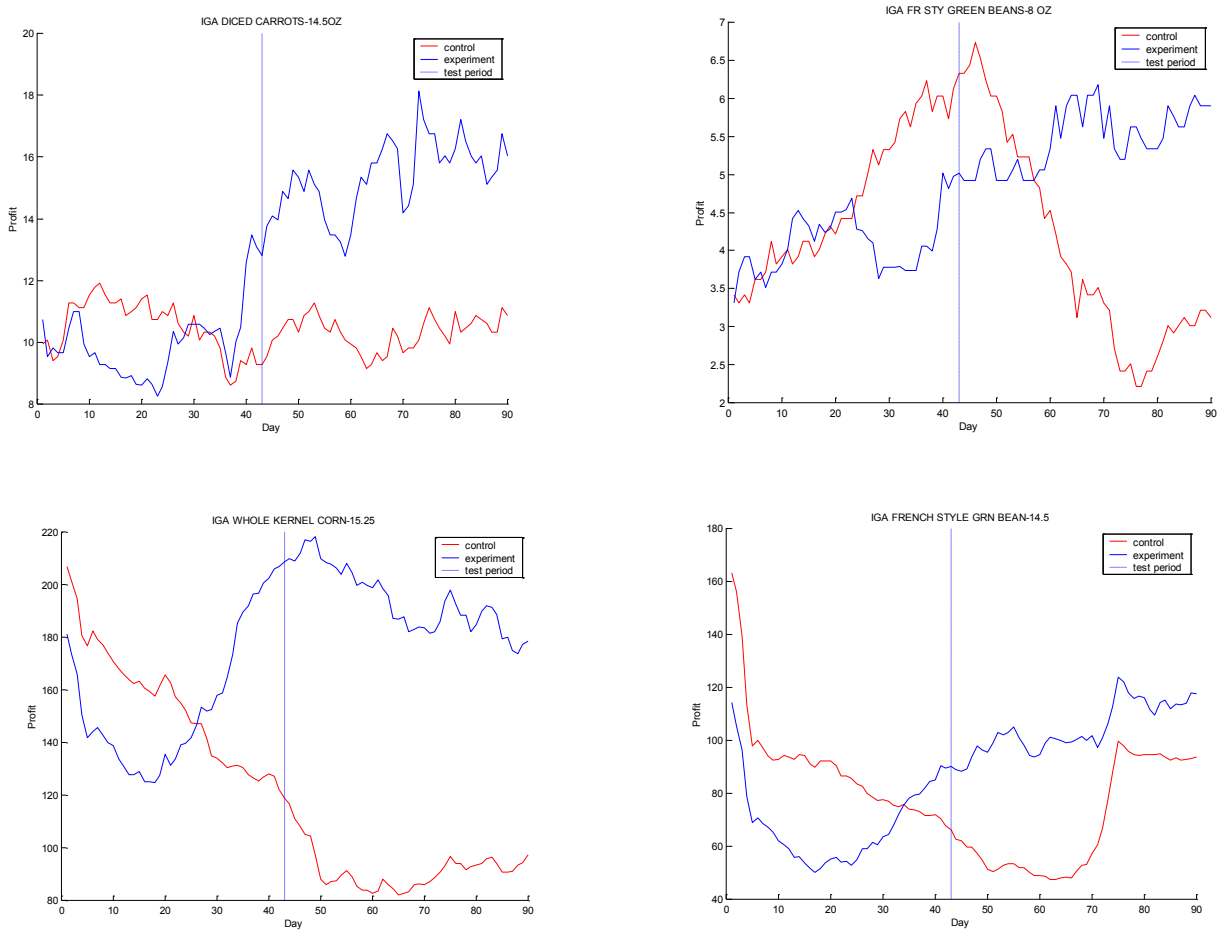


Figure: Example of a successful price change in canned goods. **(top left)** profit at experimental store plotted against price level at experimental store. Both timeseries have been converted to zscores to show on the same scale. This shows that the price was increased around day 42 (week 6) of the experiment. The other canned goods in this figure also experienced similar increases. **(top right, bottom left, bottom right)** 7 day moving average of profit series (which is why the price change on day 42 appears to be late) for three canned goods. In many of these products the demand is declining initially in both experimental and control stores. Possibly this is a seasonal change arising because the store is moving into a summer period in which canned vegetable sales are lower. When the price change comes into effect, the demand for canned goods continues to decline or be low in the experimental store, but because of the profit increase, the profit margin jumps to a higher level.

APPENDIX E

Model cleaning

Before moving to the optimization step, we need to clean up the demand models which we obtained in the training phase. The reason is that the optimization will exploit any anomolies it finds in the structure of the models. For example, say that an item has a positive elastcitiy – ie. increasing its price increases the number of units sold. Perhaps on the training data, the data presented to the model was such that this helped it fit the curve. However, now when we try to optimize this model, the optimization algorithm will pick up on this positive weight, and increase the price of that item to the maximum possible. This could both give unrealistically high estimates for predicted proft, and will also result in price movement which doesn't make sense in reality.

To prevent this problem, the each model is checked and removed if they exhibit any pathologies which could cause a problem in the optimization. These items are then flagged to signal to the user that these items need to be taken care of manually. Types of removal performed include:

<u>Models with low prediction accuracy</u>

An inaccurate model is a model which generated a poor score on its test set. Accuracy can be measured using R^2, mean absolute error, or another measure. If an item is below a certain threshold defined by the user, that model is considered unreliable, and the item is removed from the optimization. Removal is done by zeroing out that item's weight matrix (so that derivatives are zero) and freezing that item's upper and lower price constraints to be equal to its last known price. (so it cannot move outside of those bounds)

<u>Improperly formed models</u>

An improperly formed demand model is a model which doesn't conform to what we know about the nature of demand. There are two types of improperly formed models:

(1) a model which has a negative intercept is improperly formed, since this states that at 0 price the demand is negative, which is impossible. In this case the model is deleted
(2) A model with a positive main effect weight. We could repair the model by setting the main effect weight to zero. Zeroing out the main effect weight will decrease the sum of squared errors which the model accounts for, however, the remaining model terms will continue to provide a contribution to squared error, and so the model minus its main effect can still be used. Since the model is linear, the form of the model with its main effect taken out is the same as the form of the model if it was induced without a main effect variable. But then it would become a pawn of the other items. The other items would adjust it to its max or min, depending on whether it contributes positively or negatively to them. To prevent this, we should delete it completely, and again freeze its price to the price on the last day, disallowing further movement and setting all derivatives from that item to 0.

General outlier removal

Outliers are usually caused by bad data – extremely large or small values occurring in the data.

Poorly supported models

A poorly supported model is a model which was constructed from data with very few price or demand changes. In this case, the model could infer a relationship (eg. price rose and demand dropped), which was just an anomaly caused by there being so few real datapoints. Normally these items can be removed before inducing models, saving the expense of building the model for these items which we won't be able to make reliable conclusions from.

In Lanco's data approximately 50% of all items were removed prior to the optimization step due to having problems of the sort above.

The work reported in this paper differs from a previous work in retail in the following ways:

1) Pricing recommendations are made for around 40,000 UPCs. Past optimization experiments have been run using smaller domains, for instance, a single category of a few dozen items.

2) All results have been obtained with un-enriched Point-Of-Sales data, which is recorded by all POS systems. Some past studies have used expensive data sources such as Neilsen panel data to build models, which may not be economically feasible to acquire on a continuing basis.

3) The system has been validated with 200 days of test data. The prediction accuracy is comparable or higher than similar results reported elsewhere in the literature.

4) We test our price optimization method in a live price experiment at a participating retail chain, using 6% of the UPCs from 7 categories.

Lucy is a large General Merchandise chain. We were provided with 412 days of POS data for analysis related to price elasticity. Total sales over time for Lucy are shown in figure x. This shows that Lucy has a regular 7-day cycle of lows and highs. The large increase in the middle, and sudden drop to 0 sales the next day is Christmas.

Bethany is a small grocery chain. We acquired 411 days of data, and selected the fastest moving 16,000 items for analysis. Bethany's sales profile is shown in figure x. This shows that grocery stores have different profiles to general merchandisers. This series has a similar 7-day period to Lucy, however, has a number of peaks, roughly corresponding to Thanksgiving and Christmas.