Integrating Domain Expertise With Aggregate Data Using Evolutionary Computation

Bruce Edmonds and Scott Moss
Centre for Policy Modelling
Manchester Metropolitan University Business School
http://cfpm.org

Introduction

In marketing there are (at least) two separate worlds of knowledge: information about consumers (from surveys, panel data, etc.) and information about purchase decisions (from aggregate sales data and similar). Relating these two worlds is difficult because the consumer views cannot be easily traced to a part of the sales data (or vice versa) – there is no "psychological" model that bridges this gap (although there have been some attempts at making one). Part of the problem is that the standard method is to take a model structure that is *a prior* acceptable (to academics and other experts), parameterise it to fit the data and finally present the results to practitioners (those who actually have to try and make sense of the results in terms of costly marketing decisions). The data is thus used to try and adapt a theoretical model to specific market of concern. However this means that the *meaning* of the model is in the hands of the academic or expert who formulates and manipulates the model – the practitioners are left with a black-box model and (if they are lucky) some results. Unless the model-maker and the person with experience in the market happen to be the same person then, necessarily, the model will have to be applied somewhat 'blindly' to the market and its formulation will be unable to make use of the context-dependent domain knowledge of the practitioner.

In contrast to this traditional 'division of labour', we propose an approach whereby the final model is constructed, in part, by the practitioner (and optionally a computer) but within a framework devised by academic modellers. Thus we seek to *delay* some of the model specification. This is done at three different levels:

1. we have specified a general framework for relating a certain type of consumer preference model to aggregate data, based on expert advice, our experience, and multiple trials of frameworks to find what works – this can be thought of as a meta-model because it relates a class of specific models to results (this will be called the 'generic framework');
2. a marketing practitioner specifies some information to constrain the possibilities inherent in the general framework to those that are relevant to a particular market based on their knowledge of their market – this provides

the semantic framework for the specific models (we will call the 'market context');

3. within these constraints consumer preference models can be posited and judged – these are the most specific models which can then be directly related to the aggregate data to find out (among other things) how good a fit it is – we will call this the 'preference model'.

These three levels are illustrated in figure 1.

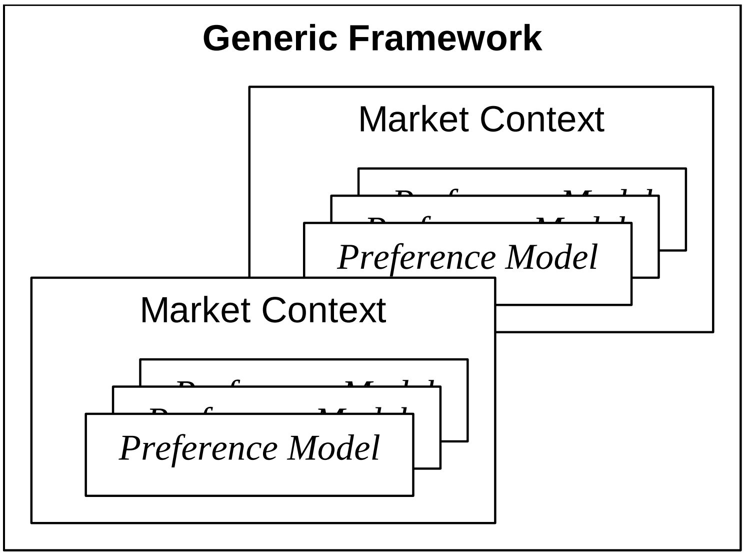


Fig. 1. The three levels: generic framework, market context and preference model

Within each of these levels there is a different feedback process. I will start at the most constrained level, that of particular preference models. Once a preference models is suggested, the generic framework and market context can be used to simulate the aggregate level of sales over time. The simulated demand can be compared with the historical demand, and if the discrepancy is too great the preference model might well be rejected and another preference model sought. The preference model gains its meaning in terms of the market context specified by the practitioner, and thus can be interpreted in the practitioner's terms. If it seems impossible to formulate a preference model that seems credible to the practitioner (*both* in terms of the model's meaning and the extent of its agreement with historical aggregate sales data) then this might lead the practitioner to change the specification of the market context, for example by extending the context by adding a product characteristic that might be significant but was omitted. If it seemed impossible to specify a market context that was both reasonable to the practitioner and so that it was possible to formulate an acceptable preference model within it then this would lead one to reject validity of the generic framework. These nested processes of feedback and re-specification is shown in figure 2. The authors have gone through all these feedback loops many times in the development of the model and so, whilst all models have some limitations in terms of applicability, the practitioner should not have to traverse the outermost loop.

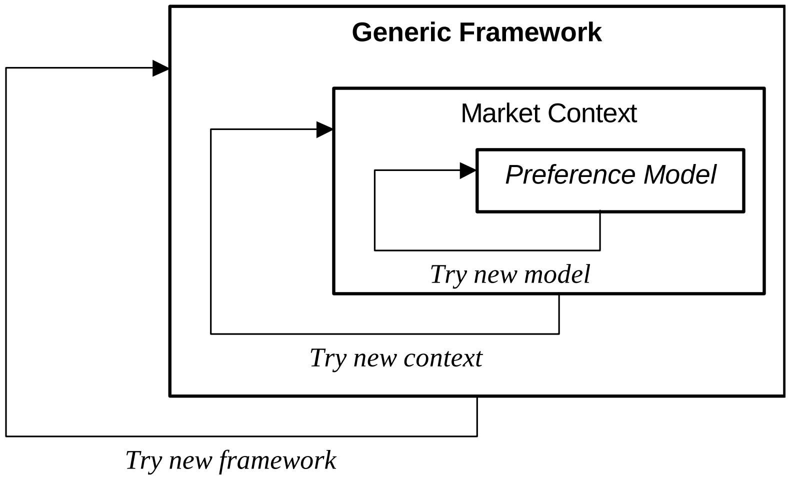


Fig. 2. Nested processes of feedback at the three levels.

In order to facilitate the most specific feedback cycle the preference models can be compared with the aggregate sales via a straight forward computation to simulate the aggregate projected sales from the model specification (using the generic framework and market context). Here the preference models are being compared within the space of possible aggregate sales time series. However, it is also useful to be able to compare preference models within the space of their characteristics in terms of their specification in terms of the properties of the consumer clusters specified in the preference models. To facilitate this it is useful to be able to do the reverse – that is, to 'calculate' preference model specifications from the aggregate data. In other words it is useful to be able find the preference models (within a given market context) that most closely matches the aggregate data when simulated. This is difficult to calculate directly since the equations embedded in the generic framework are not solvable in an analytic fashion. Thus we use a technique from machine learning to perform a search for such models. This technique is only feasible because the market context constrains the search space. Of course, just as in the simulation of aggregate sales from preference models, the result of this search, a preference model, might well be rejected because it is unacceptable in the view of the practitioner. Continual failure of the search process to discover a model with an acceptable meaning in terms of the market context might cause the practitioner to re-specify the market context[1]. Figure 3 illustrates the simulation and search processes.

[1] For more on the dynamics of learning with contexts see (Edmonds 2002).

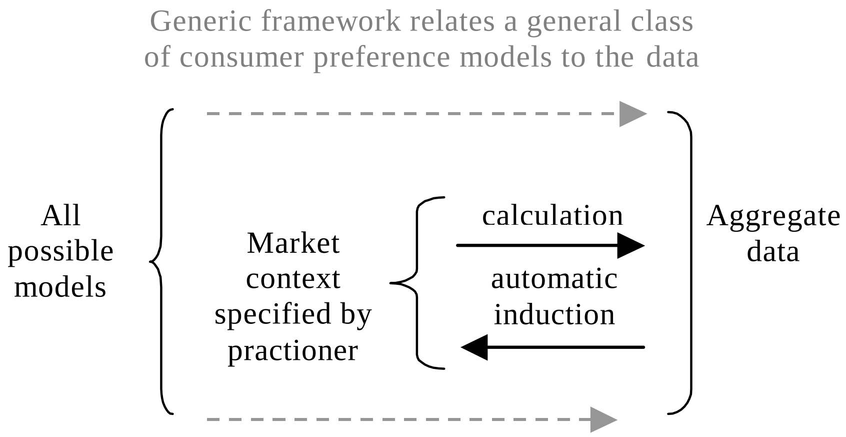

Generic framework relates a general class of consumer preference models to the data
All possible models
Market context specified by practioner
calculation
automatic induction
Aggregate data

Fig. 3. Calculation and preference model induction within the market context

In summary, the idea is that the suggested structure allows practitioners to experiment with preference models within a market context they have specified but all of this embedded within a generic framework which allows the preference models to be related to aggregate data. In this way a preference model might be developed that is *at the same time* meaningful in terms of the practitioner's viewpoint *and* coherent with the aggregate data.

This chapter starts by describing the three layers of this model: market context, preference model and generic framework. It then briefly describes the machine learning technique used to search for preference models before describing the results and experience in a specific application: the UK market for one type of liquor. It ends with a discussion of some of the issues raised. In order to make the main text more readable a lot of the technical information has been relegated to appendices. The first appendix describes some of the assumptions behind the generic framework; the second briefly describes and lists the equations of the generic framework; and the third describes the algorithms used in more detail.

The Three Layers

In this section we describe the three layers. We start with the market context because this provides the meaning for the preference models which we describe second. Lastly we outline the generic framework because this is the most technical part, being composed on a system of equations.

The Market Context

The market consists of a set of products and consumer clusters. These are described using an attribute space, each of whose dimensions is a single characteristic that might be used by the consumers to choose between the various brands of the product type. These characteristics are the perceived characteristics of the products; they can be either objectives or subjective. These could include such as: price, size, social acceptability, how much of a bargain they are, how special they are, the extent that they look foreign etc. If characteristics that turn out to be irrelevant for a particular model are included this is not disastrous – it would mean that the results in terms of irrelevant characteristics were not very meaningful and it would slow up the search process. However if an important characteristic is missed out this can result in no acceptable preference models being possible. Thus the first thing the practitioner does is give labels to and list the relevant characteristics for a particular market.

In the example (a market for liquor in the UK) discussed below, the relevant characteristics were suggested to be: price, size, expensiveness, bargainness, specialness and uniqueness. Price and size were directly available for each brand; expensiveness was supposed to represent the general perception of how expensive the brand was – it was calculated as the average price over the time period; bargainness was the relative price of the brand compared to the average price (the idea being that sometimes one can be offered an expensive product at a bargain price); specialness and uniqueness were subjective measures, the first posited to represent how much the brand was used as a self-reward and the second was the inverse of how socially acceptable the brand is (non-unique might be brought to offer to one's friends) – these were entered using a Likhart scale by the practitioner. Note how the attributes overlap – the set of characteristics do not have to be independent, merely sufficient to distinguish the brands in the ways that are important to the customers.

Once the characteristics have been decided upon, the relevant brands are listed and labelled. For the purposes of this model different sizes of the same brand are given a separate label, e.g. "`GloopyShampoo70cl`" and "`GloopyShampoo100cl`". Also sometimes pseudo-brands that are representative of a group of identically perceived brands are listed, so the "own label" brands offered by the different shampoos might be conflated into a single label, e.g. "`OwnLabel70cl`".

It is necessary that for each such brand that a reasonably lengthy time series of aggregate sales and price data must be available. In general you need one time series for aggregate demand and one for each characteristic of the brands that significantly change during the chosen time frame of the series. For perceived characteristics that do not change the degree of that characteristic for each of the brands needs to be calculated, entered or estimated by an expert, e.g. the practitioner. The availability of such data restricts the application of the technique to where they are available.

To summarise, the practitioner needs to provide the following information in order to specify the market context:

- A list of labels for each of the brands (and pseudo-brands) considered;
- A list of labels for each of the relevant product characteristic that are used by consumers to distinguish between these brands;

- For each product:
    - For each characteristic:
        - A number representing the perceived intensity of that characteristic associated with that brand.

The Consumer Preference Models

The consumer preference model consists of a series of (possibly overlapping) consumer purchase clusters. Each cluster is supposed to represent a different set of purchases in a particular purchasing context by similar groups of consumers. For example, the attributes one might want of a bottle of drink one buys for a party might be different to those one desires of a bottle of drink one buys to celebrate a personal achievement. Thus the same customer can be associated with different clusters in different circumstances – it is not assumed that the customer has the same wishes across purchasing contexts (although there may some commonality).

For each such cluster, for each characteristic, the following needs to be specified: the ideal value of the characteristic, the tolerance to deviations from this ideal; and the importance of that characteristic for that cluster. So, for example, when considering buying a bottle of drink for a party one is throwing you might have the following values:

- price: ideal=low, tolerance=medium, importance=high (they would like a low price but are willing to consider considerable deviation from this but this could make a big difference to the final decision);
- size: ideal=high, tolerance=low, importance=medium (they want a big bottle and nothing else but it is not the end of the world if this is not available);
- expensiveness: ideal=high, tolerance=medium, importance=low (they would like a bottle that was perceived as expensive, but this is not an important matter);
- uniqueness: ideal=low, tolerance=low, importance=high (they want a known brand and this is important for the party);
- etc.

Also one has to specify the proportion of total sales that results from decisions in each cluster. This could be thought of as the size of the cluster. Finally there are the ‘global’ parameters: two to determine the relative impact of income effects via the price of brands and the effect of the matching of the brands to the consumer clusters preferences; and one to determine the ‘scale’ of the attribute space.

Thus in total the specification of a preference model necessitates values for the following:

- Attribute impact parameter;
- Price impact parameter;
- Attribute scaling parameter;
- A list of labels, one for each consumer cluster;
- For each consumer cluster;
    - A number representing the ‘size’ of the cluster;

- For each characteristic;
    - A number representing the ideal value of that characteristic for that cluster;
    - A number representing the tolerance that the cluster has to deviations in this characteristic from the ideal;
    - A number representing the 'weight' in their decision making (i.e. how much the cluster's evaluation of the match would vary with the extent of the mismatch between their ideal value and the actual value associated with the brand).

Clearly the more characteristics one has and the more clusters one posits in the preference model the more flexibility one has to 'fit' the aggregate data. The more latitude one has in 'fitting' a model to the data, the more data in the form of changing time series and multiple brands one needs to constrain the possible preference models. Thus there an unavoidable trade-off between the granularity of the preference model one posits and the extent to which that model is constrained by the data. If one has only a few clusters and much data (i.e. many brands and time series for several different characteristics over many time periods), then there will be few (if any) acceptable preference models so any ones that are discovered will reflect that data. If one has a preference model with many clusters and relatively little data, then the data provides only loose constraints on the models, leaving the practitioner more free to 'construct' preference models as desired.

The Generic Framework

The purpose of the generic framework is to map consumer preference models specified for a particular market context onto aggregate sales and price data. It consists of a series of equations that were initially developed to reflect the views of experts and practitioners but then adapted in the light of its success and appropriateness over a period of time over different markets. It was found that, in many cases that the *exact* form of the equations were not critical to the results but the general shape of the functions were. The framework is thus designed to satisfy the following criteria:

- To simulate the aggregate sales that would result in a given market context as the result of the existence of the consumer clusters specified in the preference model;
- To allow the flexible choice of market context which will indicate the meaning of the parameters in the preference models;
- To make the automatic search for preference models that correspond to the aggregate data feasible.

Here I will give a qualitative account of the key ideas encapsulated in the generic framework. The major assumptions behind the framework are listed and briefly discussed in appendix 1 and the full set of equations are in appendix 2.

The extent to which a product is satisfactory for a cluster with respect to a single characteristic is determined by a transformed normal curve. The peak of the curve is at the ideal of the cluster and is at height 1. The tolerance represents how stretched the curve is sideways – it would correspond to the deviation. The importance is the extent that it is stretched downwards. If the curve goes below zero this indicates that the product is totally unsatisfactory, regardless of how desirable it might be in other respects – in other words it fails to be a member of acceptable products of that type. Some such curves are illustrated below in figures 4 and 5.

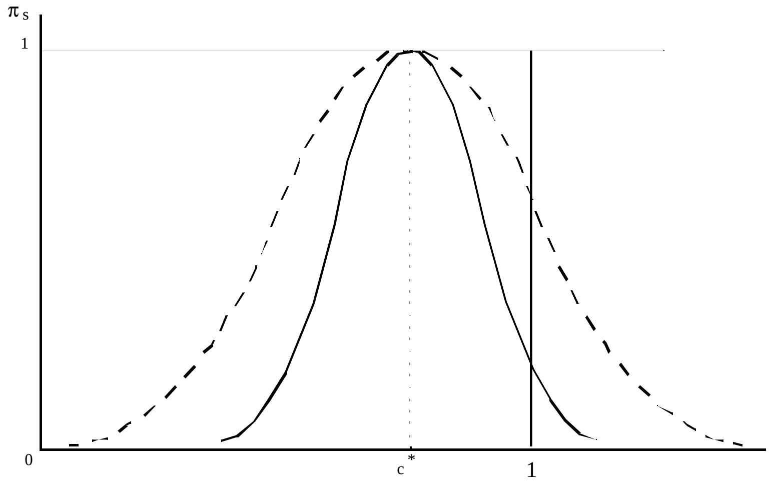


Fig. 4. Satisfaction curves with different tolerances to deviations from the ideal

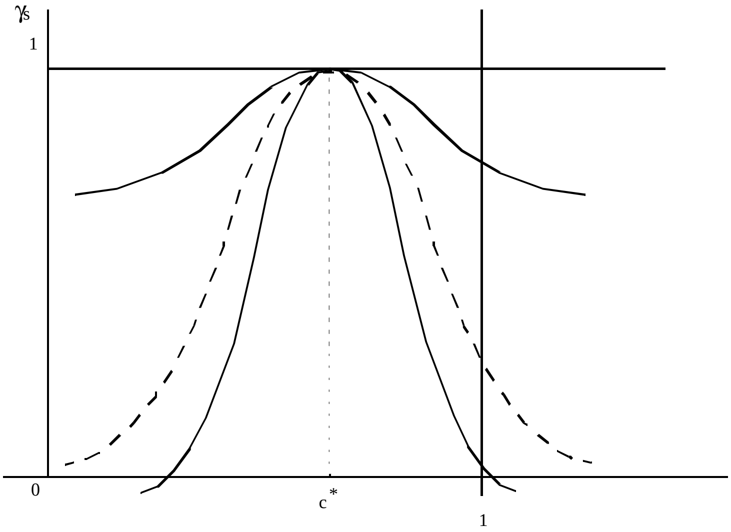


Fig. 5. Satisfaction curves with different importances

The separate satisfaction values are multiplied together to give a general preference index for that product with respect to the cluster.

The second main idea is that of the ‘distance’ of one product from another in the space of characteristics. If products are far apart then they do not compete strongly, whilst if they are close (i.e. are similar) they compete strongly. Basically products that match the cluster’s ideals (according to the satisfaction functions described above) ‘steal’ customers from nearby clusters which are less satisfactory. The closer they are the more they steal. The distance measure is a Cartesian distance on the vector of product characteristics but heavily moderated so that small changes in nearby products make more difference than small changes in the characteristics of distant products.

The desirability of the products and the interaction between products and clusters is combined with a traditional economic model of the income effect – i.e. the fact that a limited amount of money to spend will limit the purchase of more expensive products. The total effects are then combined and scaled to represent the market share of each product.

The Automatic Induction of Preference Models

In order to facilitate the interaction between the representation of the preference models in the space of characteristics and in the space of aggregate data, we applied some techniques from the field of Machine Learning in order to automatically induce appropriate preference models from aggregate time series given a fully specified market context.

A series of techniques were tried. The most successful was based around the technique of Genetic Programming (Koza 1992) – this is the one described here. It is an extension of Genetic Algorithms (Holland 1975, Goldberg, 1989) and is a particular type of evolutionary computation. Evolutionary computing evolves a population of models in a manner directly analogous to that of breeding animals. Each generation the models with the most desirable traits are selected for 'crossing' with others to produce the next generation. In this case the models were selected according to how well their simulated market shares fitted those of the aggregate sales data over an initial number of time periods, in other words the models with the lowest error were preferentially propagated and 'bred' into the next generation. Over many generations this process results in models which fit the aggregate data well. Genetic Programming is a particular type of evolutionary computing where the models can be represented as a general class of tree structures. This is particularly suitable because it allows the *structure* as well as the content to be evolved – in this case the number of clusters does not have to be specified in advance but preference models with different clusters can be explored by the search process. Genetic Programming works best were large populations are evolved slowly over many time periods, it thus requires a reasonable amount of computational power but is quite feasible to do on a high range PC overnight.

By itself Genetic Programming has a tendency to produce more and more complex models as the search progresses, so (like many others) we introduced a 'parsimony' component to the model selection. That is to say given two preference models with roughly the same level of error the model with fewer clusters is chosen. The stronger this pressure is the more likely that the process will come up with preference models with fewer clusters but this might mean that it misses a very good model with more clusters.

The efficient application of any machine learning technique (including the one described here) depends upon the exploitation of knowledge about the domain. In this case the framing that the practioners provide in the market context greatly facilitates the automatic induction of preference models. Given the state of current easily-available computational resources, the technique would not be feasible without this constraint. The algorithm is described in much more detail in appendix 3 and it performance in a particular domain described in the next section.

An Example Application – the UK Market for Liquor

The technique was initially tested on data from the UK market for liquor. This consisted of the aggregate volume and average price for 4 products[2] in each of 2 sizes over 96 weeks. A marketing practitioner suggested that the relevant attributes were: relative price, expensiveness (a long-term average of the price), size, 'specialness' (ranging from the well-known to a special treat), and 'uniqueness'

[2] One of these 'products' is the pseudo product of all the 'Own Label' products conflated together.

(ranging from the ordinary to the starkly different). The practitioner estimated the perceived values of the specialness and uniqueness attributes for each product on a Likehart scale of (low, below average, average, above average, and high) which was mapped onto the values [0.1, 0.3, 0.5, 0.7, 0.9]. The other attributes were scaled so that their values mapped onto the [0,1] interval (except price which was mapped onto the [0,0.1] interval so as not to contribute much to the perceived distance between products).

The plot for the real and simulated market shares from the best preference model found for these five products is shown in figure 6, below. The model was learnt on only the first 21 weeks of data. We see that the simulated shares track the actual shares well for well over a year. The RMS error was 3.2% on the 76 weeks of *out-of-sample* data over all five products.

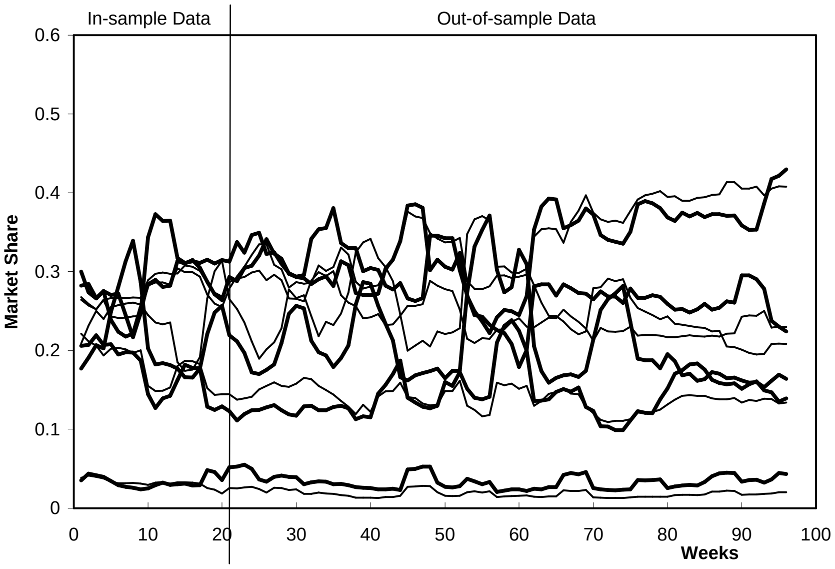


Fig. 6. Real vs. Predicted Market Share for 5 Brands (thick=actual, thin=model generated)

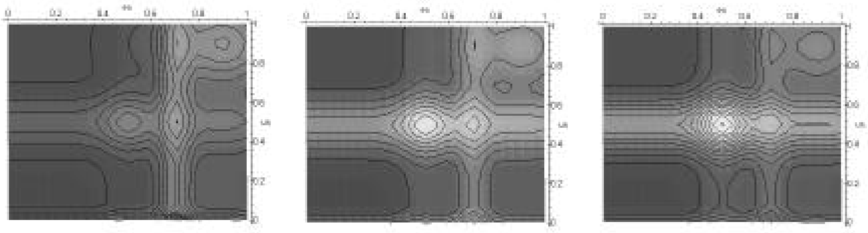
Table 1 shows the attributes for the three clusters found for this market. The entries are scaled from 0 to 10 for ease of reading. The bold entries are those where the cluster response was particularly sharp. The blank entries are where the customer response was too flat to be important. We see from this that almost half the market is determined by a cluster (cluster B) that values a low relative price and slightly larger bottles. Cluster A likes above average expense along with no uniqueness – in other words they value a well-known product that they see as being a little expensive. Cluster C has less definite preferences, but seems to be biased towards expensive and unique products. As it turns out the abstract attribute of specialness turns out not to be very important for any of the clusters, probably because this attribute is dominated by how expensive the product is perceived to be.

Cluster	Relative Price	Expensiveness	Size	Specialness	Uniqueness
A (21%)	1	**7**	6		**0**
B (49%)	**1**	5	**8**		5
C (29%)	2	9		3	9

Table 1. The significant attributes of the three clusters found

In this case the marketing practitioner interpreted these clusters in the following way: cluster A was the “social” purchases (e.g. buying to serve to friends); cluster B was the “functional” purchases (i.e. bought for the alcoholic content alone); and cluster C was the “reward” purchases (i.e. bought as a self reward or reward for others). This was similar, but not identical to a preference model the practitioner himself suggested – the difference being that he suggested a fourth cluster of “novelty” purchases (i.e. those seeking something different). Given that this particular market was for quite a traditional type of liquor it was not surprising that this group was omittable.

It is not easy to visualise the underlying preference models. We attempt this in two ways. The first method is by plotting an index of the intrinsic desirability of various values of the attributes to the clusters, which is the weighted sum of the desirability of a product with notional attributes. This is done in figures 7 and 8.

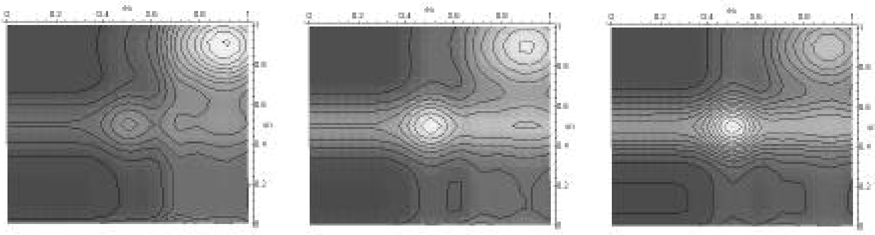


Low Relative Price

Average Relative Price

High Relative Price

Fig. 7. Index of the Intrinsic Desirability of various values of the attributes expensiveness and uniqueness, for three prices, size 70cl (white= high desirability, black=low desirability).

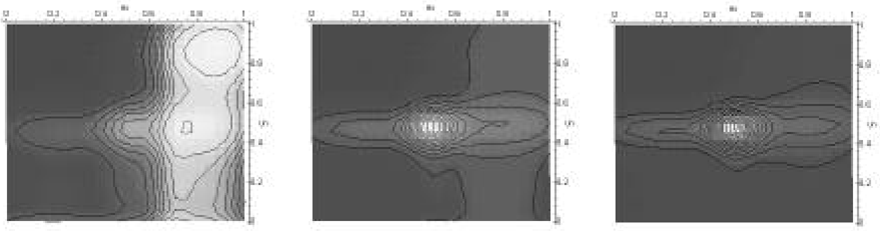


Low Relative Price Average Relative Price High Relative Price

Fig. 8. Index of the Intrinsic Desirability of various values of the attributes expensiveness and uniqueness, for three prices, size 1L (white= high desirability, black=low desirability).

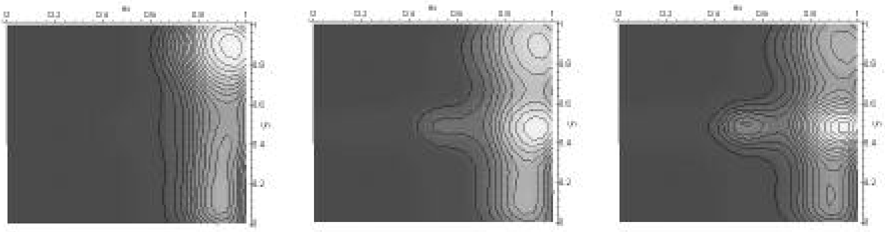
These diagrams should be interpreted with caution. *Firstly*, they represent one set of models that is consistent with the judgemental information and the aggregate data, there may be others as well. *Secondly*, the models will only be accurately induced around the values of the attributes that were input (either by the practitioner or in the aggregate date) – for example, if all the products had the same value for some attribute, then the results for other values of this attribute would be arbitrary. This is not surprising as no induction method (automatic or otherwise) can work in regions where there is no data. *Thirdly*, the diagrams in figures 15 and 16 do not take into account the effects of the other products in the market – it would be no use positioning a product with attributes that are desirable by this index if this region was already oversupplied with products.

The second method is by plotting what demand there would be for a notional new product that was positioned with various different attributes values in such a market. Figures 17 and 18 show the equivalent plots to figures 15 and 16 but for demand rather than an index of desirability.



Low Relative Price Average Relative Price High Relative Price

Fig. 9. Notional demand for a new product with various values of the attributes expensiveness and uniqueness, for three prices, size 70cl (white= high demand, black=low demand).



Low Relative Price

Average Relative Price

High Relative Price

Fig. 10. Notional demand for a new product with various values of the attributes expensiveness and uniqueness, for three prices, size 1L (white= high demand, black=low demand).

Thus the first figure in Figure 10 indicates that give we have two 70cl bottles of liquor both having the same relatively low price, then there will be higher demand for the product perceived as more expensive. That is people buying such products like a bargain – buying an expensive bottle at a cheap price. What is perhaps more surprising is that at higher relative prices there would be more demand for a product with a medium level of uniqueness (and for 70cl bottles).

Again a great deal of caution is needed to avoid over interpreting these plots. They represent the notional demand for a new product positioned only against the products selected by the competitive set filter, on the assumption that the underlying discovered preference model is correct and that this new product does not change the perceptions of the existing products. But they are useful indications that might guide further research into these markets and it does illustrate how the technique can give a result that is readily interpretable in terms that are meaningful to marketing professionals. The fact that the framework is somewhat specified by these people themselves and the models mapped back into this framework ensures this.

Performance of the GP-Algorithm

The nature of the GP algorithm is that it finds an *acceptable* solution to a very difficult problem, it is in no way an optimising algorithm – this is in contrast with a regression algorithm which finds the *best* solution to a relatively simple problem. If there was a reliable and practical way of finding the best model to fit the data within the presented framework the GP algorithm would not be necessary. In a real sense the GP algorithm is a *creative* algorithm – it *creates* possible solutions within the rugged landscape of possible models. Thus the nature of the problem makes it unlikely that the algorithm will come up with the same model each time, and this is indeed the case.

Figure 11 below shows the distribution of errors of the best model trained on the first 21 periods of data at the end of each run. The error is with respect to the total data. The average RMSE was 4.4% and the minimum just under 3%. This is consistent with the indications provided by the non-linear regression models discussed below. No runs ended with a best model with RMSE greater than 7.5% on the total data. One thing that could be done to the present algorithm to make it more consistent is to add a hill-climbing back-end so that it (locally) optimised the model it found to a canonical one for the local minima it had evolved for.

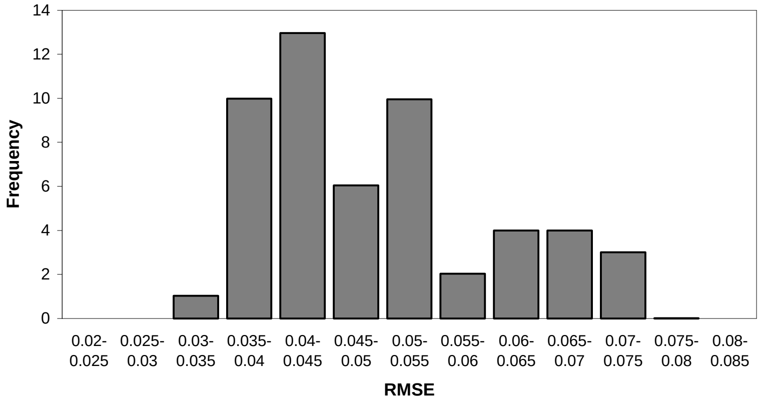


Fig. 11. A histogram of model errors from 53 runs of the GP algorithm with a population of 1000 for 500 generations (average RMSE 4.4%, SD 1.1%)

Comparison with a Non-Parametric Model

To give an idea of the extent that the technique finds a model which captures the marketing relationships we compare the performance with non-parametric model, namely that resulting from a non-linear regression analysis (Clevelan et al. 1988). The non-linear regression fits a sequence of local linear models smoothed into a continuous curve. The degree of locality is controlled by a parameter. The result is effectively a curve of a given smoothness that is fitted to the data. Such a model is almost completely free of *prior* theoretical restriction and is thus free to fit the data as closely as possible. Thus, given enough in-sample data and a low smoothness one would expect it to out-perform any other technique in terms of in-sample error, since if it was fitted on all the data with almost no smoothing it would simply "track" the data. However such a model has little *explanatory* value – a good explanation explains substantial out-of-sample data in terms of the in-sample data in terms that are meaningfully interpreted into the market being modelled. The local linear models give only local linear explanations and a series of curve for the global model but no idea of the curve's significance. However it *does* give one an idea of how much of the data is capturable in systems of curves even if these curves do not have any comprehensible analytic form.

Model	Size Of In-Sample Data	RMSE
best GP fitted	21	0.031087
non-linear regression	21	0.216146
non-linear regression	30	0.074595
non-linear regression	48	0.043716
non-linear regression	96	0.038016

Table 2. Comparison of models

Table 2, shows the error rates (over the whole of the data) of the described model fitted by the GP algorithm on 21 periods of in-sample date, compared with the non-linear regression model fitted on 21, 30, 48 and 96 periods of in-sample data. The non-linear model fitted on 96 periods is comparable in level of error to the described model fitted on 21 periods, although this is not achieved evenly across the four brands (Figure 12). The non-linear model fitted over the entire 96 time periods has an RMSE of 3.8%, indicating that this is around the limit for what can be fitted from the time series alone. This may indicate that the market context entered by the practitioner added predictive value beyond that in the aggregate data.

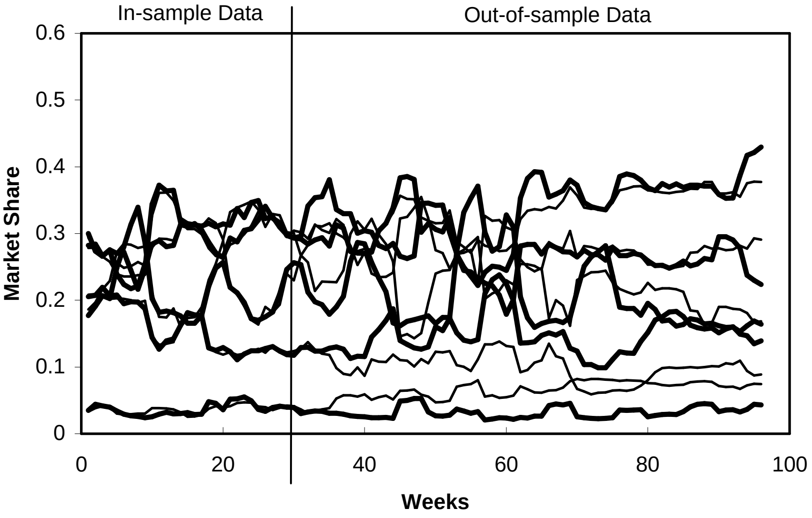


Fig. 12. Real vs. Predicted Market Share (thick=actual, thin=model generated) for Non-linear model fitted on 30 time periods

Discussion

A model can be have predictive value by processing inputs to anticipate outcomes before they happen. Such a model can be a purely 'black-box' model, there is no necessity for the workings of the model to be of any particular form. A model can have explanatory value by providing a process from the initial conditions to the outcomes that can be interpreted back into the modelled domain. In this case the model mechanism needs to be, in some sense, a representation of the causal mechanisms of the process being modelled. In other words there must be a finer grained correspondence between model and modelled than merely the initial and final stages - some of the intermediate stages must also correspond in some way.

Prediction and explanation in these senses are key concerns of modellers and knowledge engineers, they concern the relation of their representations with the data models of the phenomena that concern them. However most people do not work with such formal representations but rather with semantically rich representations of their world - what might be called phenomenological models. In complex social situations (such as markets) human informal reasoning, story telling, pattern matching and intuition can easily out-perform formal syntactic models. Thus there arise two different 'worlds' of knowledge about complex phenomena - that based on formal syntactic modelling and that based on phenomenological understanding.

The creation of meaning in a phenomenological sense is far from being completely understood. However a plausible and persuasive line of thought (Wittgenstein 1963) holds that such meaning is acquired though use in context. It seems that it is through a dialogue inside a shared context that meaning is indicated, checked and cemented. In the situation here the practitioner imports referents that are already meaningful from the experiential domain and uses those to specify and label the market context. The interplay between preference models and their results in terms of aggregate data is a sort of dialogue in which models are repeatedly used in context and which provides feedback in meaningful terms. It helps to give the models phenomenological meaning in terms of the outputs as well as their meaning in terms of the (already meaningful) labels. This 'playing' with the models is facilitated by the generic framework which 'hides' the detailed model mechanics from the practitioner who works solely within the market context they have specified and the graphs of the aggregate outcomes. The formal consistency of the preference model and the simulated outcomes is the job of the generic framework made by the modellers. The set-up described in this chapter thus allows a preference model to be meaningful in *both* phenomenological and modelling terms. It can be a bridge between the informal but rich understanding of consumer attitudes (informed by surveys, anecdotal accounts etc.) and the aggregate time series data.

The net effect is that we have 'devolved' some of the modelling to the practitioner in a structured way providing some intelligent tools to facilitate this. This means that the practitioners have some freedom in how they use the set-up. As mentioned above these uses can be placed on a constructivist-realist line. If preference models with relatively few clusters are used in the presence of a lot of data then the preference models will be very constrained by that data resulting in a near-unique reflection of that data. If preference models with many clusters are chosen only loosely constrained by limited data then the choice might be more driven by the practioners' intuitions and biases – in other words it might be more accurate to say they construct the preference models.

Clearly there is a danger that a practitioner might 'con' themselves into misinterpreting a constructed model as a reflective one. However it is not cleat that there is any more of a danger than with any other modelling technique where the process of interpretation from the formal to the phenomenological world is implicit and unclear.

References

Angeline, P. J. and Kinnear, K. E. Jr. (eds.) (1996) Advances in Genetic Programming, Volume 2. Cambridge, MA: MIT Press.

Blattberg, C.R. and Hoch, J.S. (1990). Database models and managerial intuition - 50% model + 50%. Management Science, 36:887-899.

Campbell, M., Moss, S. and Simms, C. (1996). Rulebase-Driven Non-Linear Analysis of Competitive Structure. CPM-report 96-13. (http://www.cpm.mmu.ac.uk/~scott/es96web/wmwork/www/es96-1_1.html).

Cleveland W. S., Devlin S. J., Grosse E. (1988) Regression By Local Fitting - Methods, Properties, And Computational Algorithms. Journal Of Econometrics 37: 87-114.

Bronnenberg B. J., VanHonacker W. R. (1996) Limited choice sets, local price response, and implied measures of price competition. Journal of Marketing Research 33:163-173.

Dawes, R. M. (1972). Fundamentals of attitude measurement. New York: Wiley.

Deaton A. and Muellbauer J. (1980). An almost ideal demand system. American Economic Review, 70:312-326.

Edmonds, B. (2001) Learning Appropriate Contexts. In: Akman, V. et. al (eds.) Modelling and Using Context - CONTEXT 2001, Dundee, July, 2001. Lecture Notes in Artificial Intelligence, 2116:143-155.

Goldberg, D. E. (1989) Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.

Guadagni, P.M., Little, J.D.C. (1998). When and what to buy: a nested logit model of coffee purchase. Journal of Forecasting, 17:303-326

Holland, J. (1975). Adaptation In Natural and Artificial Systems. Ann Arbour: The University of Michigan Press,.

Horsky, D. and Rao, M. R. (1984). Estimation Of Attribute Weights From Preference, Management Science, 30:801-822.

Koza, J. R. (1992) Genetic Programming: on the programming of computers by means of natural selection. Cambridge, MA: MIT Press.

Meehl, P. E. (1954). Clinical versus statistical prediction : a theoretical analysis and a review of the evidence. Minneapolis: University of Minnesota Press.

Morgan-Jones, J. and Zufryden, F. S. (1980) Adding Explanatory Variables to a Consumer Purchase Behavior Model: An Exploratory Study. Journal of Marketing Research, 17:323-334.

Moss, S., Gaylard, H., Wallis, S. and Edmonds, B. (1998). SDML: A Multi-Agent Language for Organizational Modelling. Computational and Mathematical Organization Theory, 4:43-69.

Murphy A. H., Brown B. G. (1984) A Comparative-Evaluation Of Objective And Subjective Weather Forecasts In The United-States. Journal of Forecasting 3:369-393.

Spector, L., Langdon, W. B., O'Reilly, U. and Angeline, P. J. (eds.) (1999) Advances in Genetic Programming, Volume 3. Cambridge, MA: MIT Press.

Wittgenstein (1963) Philosophical Investigations. Basil Blackwell, Oxford.

Wolpert, D. H., and Macready, W. G. (1995) No Free Lunch Theorems for Search. Technical Report, Santa Fe Institute, Number SFI-TR-95-02-010. (http://www.santafe.edu/sfi/publications/Working-Papers/95-11-101.ps)

Acknowledgements

The system and results reported here were obtained as part of the Intelligent Marketing Integrated System (IMIS) Project conducted in collaboration with United Distillers PLC as part of the Information Systems Integration Programme of the

UK Engineering and Physical Sciences Research Council and the Department of Trade and Industry under contract number IED/4/8022. The programming language used in developing IMIS is SDML, designed and implemented by the Centre for Policy Modelling using the VisualWorks 2.5 development environment supplied by Sincom Systems Ltd.. The data was provided by United Distillers who have generously supported IMIS from its inception. Clive Sims of Competitive Analysis International suggested basing demand analysis on need-states which we formalized as context-dependent purchase clusters. We thank our colleague Steve Wallis and our collaborators in IMIS, Michael Campbell and Paul Ormerod for their encouragement and help in establishing the programme of which this chapter is a result.

Appendix 1 – Some Assumptions Behind the Generic Framework

In this appendix I list some of the assumptions behind the generic framework specification. Some of these assumptions are explored in (Bronnenberg et al. 1996, Dawes, 1972, Guadagni and Little 1998, Horsky and Rao 1984, Meehl 1954, Morgan-Jones and Zufryden 1980 and Murphy and Brown 1984).

Assumption 1: In each market there will be a number of dimensions representing the relevant characteristics that the consumer uses to decide amongst competing products.

There can be any number of these both of numerical and binary type. These dimensions are selected by the practitioner concerned as those that are relevant. It is important that all the characteristics that are important for distinguishing products are included but it is not critical if some irrelevant ones are included[3]. Price is almost always included as a relevant characteristic. These characteristics are intended to be those as perceived by customers – they are not necessarily linked to physically measurable attributes and can be quite abstract. For example one dimension could be whether a beer is imported or not and another how expensive the product is perceived to be.

Assumption 2: The perceived values of the characteristics of the products are known.

That is, once the relevant characteristics have been decided then the rough position of the products in terms of their perception by consumers in terms of these attributes are known. Practitioners are often far more confident about the perceptions of a product than how these are used by consumers to make choices.

[3] The algorithm will discover this irrelevance, resulting in clusters that are extremely tolerant to deviations in these characteristics. The inclusion of irrelevant attribute charactersistics will, however, make the search harder as each extra attribute characteristics will expand the search space.

Assumption 3: There are meaningful clusters of purchasing decisions that will have broadly similar goals in terms of these dimensions.

These clusters correspond with potential customer purchase contexts rather than customers. For example a customer could desire different attributes of a bottle of spirits if buying for a social event and when buying for themselves as a reward for some achievement. There can be any number of these clusters.

Assumption 4: The desirability of the product (in the absence of competition) is strongly related to the price and the extent to which the product meets the ideals of the cluster on the relevant characteristics separately.

This is not such a restrictive assumption as it may seem as the characteristics need not act independently and they could be quite abstract such as *expensiveness* (which could be modelled as the average price) or *the extent to which an item is a bargain* (which could be modelled as the extent to which the current price is less than its average price). However it does rule out extremely non-linear combinations of dimensions, for example where *a single cluster* desires one attribute on one dimension and another on a second dimension *but not both together*. In the application to a market for spirits reported below, for example, the clusters induced could be characterised as: social, functional and reward. The attributes of the products sold in that market which were specified by the marketing professionals as uniqueness, specialness and expensiveness. Expensiveness is not the same as price or relative price since an "expensive" drink can sometimes be acquired (relatively) cheaply in a sales promotion. Also expensiveness might have an upward-sloping demand curve as opposed to a typical downward-sloping curve for price.

The next assumption concerns the distance between competing products.

Assumption 5: The extent to which products compete is strongly (and inversely) related to the "distance" between products expressed in terms of their characteristics in terms of these key dimensions.

That is to say that items that are perceived as having very different values of these relevant characteristics will not be strongly competing with each other and ones more similar will be more strongly competing. There are more assumptions about the relevant characteristics and shape of this distance function which we discuss below.

Assumption 6: When products are sufficiently close to each other (in terms of perceived distance), the extent to which sales can be "poached" will depend on the closeness, the price and the extent to which the product meets the ideals of the cluster.

Note that these effects do not have to be independent of each other; for price can also be a attribute used to judge distance between products and also their general desirability in a non-decreasing way.

We make several assumptions about the preferences of these groups

Assumption 7: Each group has an ideal, such that it desires the product more the closer the characteristics of that product matches its ideal.

Note that we do *not* assume that these ideals are very important in all dimensions, hence the next assumption.

Assumption 8: Groups will have different tolerances to deviation from this ideal in the extent to which their desire for the product decreases with this deviation.

Assumption 9: Groups will have different residual desires for products when there is a large deviation from their ideal.

That is to say that some dimensions will be of the nature of an 'optional extra' so that even if the desired attribute is not at all present they still have a high basic desire for the product. For other groups, products and attributes there will be a sharp 'cut-off' point beyond which the group would not consider purchasing the product at all.

We have chosen a space of transformed normal curves as the basis for our preference functions in any one dimension for a group. This has the advantage that there is a region around the ideal where small deviations from the ideal are not perceptible and that the effect of deviation can drop of slowly with large deviations. In trials it was found that the exact shape of this function was not critical for the overall technique, for example it was found that a triangular distribution about the ideal did almost as well.

Appendix 2 – The Equations of Generic Framework

The most basic function maps the extent to which a characteristic of a brand is satisfactory for a cluster (compared to its ideal). The extent of the satisfaction of the cluster with a brand depends upon: what the value of the brand's characteristic is; its ideal for this characteristic; how tolerant it is with respect deviations from this ideal; and how important this is for the cluster's overall decision. The mathematical form of this function is:

$$g = \max(0, \frac{2m_c e^{-36\left(\frac{c^*-c}{t_c}\right)^2} + 1 - m_c}{1 + m_c}) \tag{1}$$

where c is the value of attribute C, c^* is the ideal value, m_c is the index of the importance and t_c is the corresponding tolerance index. This is merely a parameterised version of the equation for a normal curve obtained by stretching about c^* by t_c in the c-axis and downwards by m_c down from 1 in the $g$axis.

The overall level of satisfaction of a cluster with a brand is given by the preference index. The preference index corresponding to the cluster *s* for product *b,* denoted G_{sb}, is the product of the preference indices for actual attribute value associated with the product. Formally,

$$\Gamma_{sb} = \prod_{c \in C} g_{sc} \tag{2}$$

where *C* is the set of defined characteristics. These indexes are then scaled for all the products in the set being considered so they sum to one. Thus, the *strength* of product *b* w.r.t. cluster *s* is as follows.

$$s_{sb} = \frac{\Gamma_{sb}}{\sum_{j} \Gamma_{sj}} \tag{3}$$

The perceived 'distance' between brands was modelled as a function of the value of the brands' characteristics as vectors. The function used in the model reported here was:

$$d_{ij} = \tanh\left(2\left|\Theta_i - \Theta_j\right|\right) \tag{4}$$

This has the effect of squashing the distance between brands, so that changes in the perceived characteristics of brands had little effect when they are very dissimilar but more effect when they are similar. This distance function models the extent of product differentiation.

Because product differentiation need not have the same impact in all markets, we specify the differentiation effect as being determined by the distance between the products in characteristic space and a *distance scaling parameter* to be denoted as I_d. This function is

$$d_{ij} = e^{-\left(I_d\, d_{ij}\right)^2} \tag{5}$$

where δ_{ij} is the distance between products *i* and *j* in attribute space. This is a scaled index of relevance of one product to another.

The price effect is a standard economic demand function. It models the income effect of prices (i.e. the fact that as goods become more expensive fewer people can afford them). The effect of the relative prices is

$$\Pi_{ij} = e^{-\frac{p_i}{p_j} d_{ij} I_p} \tag{6}$$

where $\mathrm{I_p}$ is the price effect parameter, p_I is the price of one product and p_I that of the other.

The effect of the relative strengths of two products with respect to a cluster, *s*, is

$$\Sigma_{sij} = \frac{\tanh\left(\left(\frac{s_{si}}{s_{sj}} - 1\right) d_{ij} I_s\right) + \tanh\left(d_{ij} I_s\right)}{1 + \tanh\left(I_s\right)} \tag{7}$$

where I_s is the *strength scaling parameter* and is the distance effect index introduced above. The larger the value of the this parameter, the higher the value of Σ_{sij} for any value of the strength ratio. This function is just a shifted logistic curve, its range is from 0 to 1 and it gets steeper when the distance effect is greater (i.e. products are closer to each other).

Finally the price and strength effect are multiplied together to give the reach of one product over another:

$$r_{sij} = \Sigma_{sij} \Pi_{ij} \tag{8}$$

Finally the ‘reach’ function of product *i* over product *j* is the weighted sum of the reaches over all the clusters. This represents the total reach of one product over another, in other words this represents the proportion of sales lost to each of the other products.

$$r_{ij} = \sum_{s \in S} w_s \, r_{sij} \tag{9}$$

where *S* is the set of purchasing clusters.

Thus the proportion of sales lost to all other products is given by the product of the proportion lost to each of the others:

$$R_b = \prod_{j \neq b} r_{jb} \tag{10}$$

Now the notional demand is the sum over clusters of the strength (weighted by their size) reduced by this loss rate.

$$\Delta_b = R_b \sum_s w_s \, \hat{s}_{sb} \tag{11}$$

Finally, to get the simulated market shares we normalise these notional demands to the size of the market.

$$m_b = \frac{\Delta_b}{\sum_i \Delta_i} \tag{12}$$

This gives the model’s prediction of the market shares of each product at each time.

Appendix 3 – The Automatic Induction of Preference Models

The heart of the procedure for automatically searching for a preference model whose output would approximate the aggregate data is a genetic programming algorithm (Koza 1992). This can be made more robust with a random search front-end to ensure a viable initial population of possible models and then a final hill-climbing algorithm afterwards to tune the models found. Recent collections covering extensions to the technique can be found in (Kinnear 1994, Angeline and Kinnear 1996, Spector et al. 1999). Here we will just describe the basic techniques.

Genetic programming module

Genetic programming (GP) differs from the familiar genetic algorithms in that the gene is a labelled tree rather than a string. The basic GP algorithm is:

1. Specify the possible branching and terminal nodes that the trees can be built from and the fitness function for evaluating them.
2. Generate an initial population of random trees of a given depth using these nodes.
3. Evaluate this population using the fitness function.
4. Find the best gene and, if it is good enough, stop.
5. Otherwise generate a new population of trees using one of two methods (according to a fixed proportion determined by the programmer):
    - drawing pairs of trees randomly from the current population with a probability related to their fitness and producing two new offspring by choosing a random node in each and swapping the sub-trees that are rooted at these nodes (tree-crossover) or,
    - randomly choosing trees with fitness-related probabilities for propagation to the new population.
6. Go to step 3.

In our case the tree-structure covered possible preference models. A gene was an instance of the following specification:

```
gene := parameter list, weight list, cluster list,
  parameter list := price value parameter, strength value parameter, distance scaling parameter
  weight list := list of non-negative numbers (of same length as list of clusters)
  cluster list := list of cluster specifications, one for each cluster
    Cluster specification := list of preference specifications, one for each property
      preference specification := a triple of numbers: the ideal value, its importance and the tolerance to variation
```

The fitness function was the RMSE error of the predicted market shares compared to the actual shares over a sample period for the competitive set with a small discount to bias the algorithm in favour of models with fewer clusters.

Our crossover operator was constrained to produce only well-formed genes, i.e. if one chosen sub-tree was a preference specification the other would be also. Also if the domain expert had previously entered any trial preference models, these would be seeded into the initial population, so that variations of these would be tried along side the randomly generated ones.

Competitive set front-end

One problem we encountered is that although practitioners may know the total set of products in any market and, for their own products, have strong and well articulated views about the main competing products, they are uncertain about the effects of other products on their own. Putting the complete range (perhaps 2500 products for which data is available) into the algorithm would needlessly waste computational time since most of these products have little effect on one another. Moreover, practitioners frequently want a model centred around a particular product. For these entirely pragmatic reasons, we added a front end which applies statistical algorithms to the full data set to filter out products characterised by insignificant (though not typically symmetrical) cross price elasticises with a designated *focus* product.

A three-stage filtering algorithm was developed to identify the set if competitors of any, arbitrarily focus product. At each stage, the marketing practitioners were able to retain products discarded by the algorithm or discard retained products. The stages were:

OLS regression of market share of the focus product on the relative price in logs of each of the other products for which data is held as well as regressing the shares of the other products on the same price variable of the focus product. Products were retained if both t-ratios exceeded a critical value and discarded otherwise. The critical value was chosen to yield enough degrees of freedom for the second stage.

Multiple OLS regressions on the logs of the prices of all remaining products and the log of total sales volume. This was an iterative process in which the product with the lowest standard error on its price coefficient at each iteration was discarded until the coefficients on all of the log prices were significant at the 99% confidence level. The regression equation used in this stage was taken from the Deaton-Muellbauer AIDS algorithm (Deaton and Muellbauer 1980) but without the symmetry restriction. In general, the marketing professionals were interested in the half-dozen or so most important competitors. Leaving 15 to 20 products in the competitive set at this stage gave the practitioners confidence that all of the most important six to eight competitors were included for the third stage.

Further elimination of products from the competitive set together with analysis of the changes in competitive structures over the data period was based on a nonlinear generalization of the second stage based on the local regression algorithm of

Cleveland and Devlin (1988).[4] The particular advantage of this stage was that it yielded a time series of cross-price elasticises indicating that some products appeared significant in linear regressions because of a few large and systematic fluctuations in volumes and prices due to special offers or other ephemeral events during the observation period. A rulebase was developed to identify such products and also to identify products that were becoming less competitive with the focus product. All such products were discarded from the analysis.

[4] We actually used a refinement of this technique developed by our collaborator on this project, Michael Campbell. A full description of the algorithm used is reported by Campbell *et al.* (1997).