

# Management Of The Marketing Mix, Using Models Based On Household Level Data

by

Árni G. Hauksson

B.Sc. in Electrical Engineering, University of Iceland in Reykjavík (1992)

Submitted to the Department of Electrical Engineering and Computer Science in  
partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

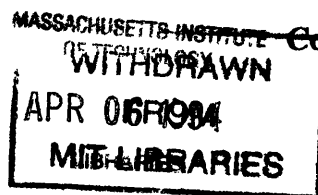
February 1994

© Árni G. Hauksson 1993. All rights reserved.

Author \_\_\_\_\_  
Department of Electrical Engineering and Computer Science  
October 18th, 1993

Certified by \_\_\_\_\_  
John D. C. Little  
Institute professor  
Thesis Supervisor

Accepted by \_\_\_\_\_  
Richard C. Larson  
Codirector, Operations Research Center



---

# **Management Of The Marketing Mix, Using Models Based On Household Level Data**

by

**Árni G. Hauksson**

**B.Sc. in Electrical Engineering, University of Iceland in Reykjavík (1992)**

**Submitted to the Department of Electrical Engineering and Computer Science in  
partial fulfillment of the requirements for the degree of**

**Master of Science in Operations Research**

at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

**February 1994**

**© Árni G. Hauksson 1993. All rights reserved.**

---

# **Management Of The Marketing Mix, Using Models Based On Household Level Data**

by

**Árni G. Hauksson**

**Submitted to the Department of Electrical Engineering and Computer Science in  
partial fulfillment of the requirements for the degree of  
Master of Science in Operations Research**

## **Abstract**

A number of models have been built to describe customer behavior in choosing products to purchase, but none of them has been incorporated into a practical Marketing Decision Support System (MDSS) for everyday use by marketing planners.

In this paper we take the models to this next level. The first problem we encounter, is how to translate high level manufacturer marketing plans into store details. For this we propose two models. The first is a Markov model, in which the in-store merchandising state of the product evolves according to a Markov process. In the second model we input specific store promotions separately and use the Markov model as a back-up for periods and stores where we do not want to get into the specific store promotions.

When using the same seed method for comparing Monte Carlo simulations of scenarios with discrete random variables, we face the problem of "over-switching". In an appendix we solve this problem very efficiently.

Once the in-store merchandising has been generated, we model the households. We use a variation of the recently introduced Little-Anderson model. When building an MDSS, a major practical issue is computation time of the system. The standard simulation method is to use Monte Carlo, but it requires hours of computations. By using a Taylor series approximation for the expected values of the desired output measures, we can reduce the computation time dramatically. Furthermore we suggest a method for making this Taylor series approximation a better representative of the Monte Carlo method by adjusting some of the model parameters.

We illustrate our methods on a juice database from Eau Claire, WI. The database includes the leading 10 brandsizes. In our numerical examples we focus on Ocean Spray Cranberries' 48 oz. Cranberry juice.

Thesis Supervisor: John D. C. Little  
Title: Institute Professor

---

## **Acknowledgments**

I would like to thank my thesis supervisor, John D. C. Little, for his constant encouragement and valuable insight. When I started to get involved, I expected the work to be both challenging and rewarding, and I have not changed my mind in that respect. I would also like to thank Eric Anderson for his invaluable input and help with database work.

I would like to thank Ocean Spray Cranberries Inc., especially Gordon Armstong, for their input on the more practical side of the research.

I thank the staff and students at the Operations Research Center for providing me with a supportive environment. I will not mention any names, because I cannot think of any that I would want to leave out.

Finally I would like to thank my fiancée, parents and family, for being there and for always having been supportive of my decisions.

---

# Contents

1. Introduction .....	9
2. Overview Of Models At The Household Level.....	11
2.1 The Guadagni-Little Model.....	12
2.2 The Little-Anderson Model.....	13
3. Overview Of Intermediate Models.....	16
3.1 Promotion Model.....	16
3.1.1 Categorization Of Stores .....	16
3.1.2 Status Of A Product .....	17
3.1.3 Price.....	20
3.1.4 Determining The "Regular Price" .....	21
3.1.5 Other Products From The Same Manufacturer.....	22
3.1.6 The Model.....	22
3.2 Scenario Evaluation Model .....	23
3.2.1 The Model.....	24
4. Integration Of Models Into A Decision Support System.....	26
4.1 Promotion Calendar And Scenario Evaluator.....	26
4.2 In Store Promotions.....	27
4.3 Consumption.....	27
5. Estimation.....	29
5.1 Household Level.....	29
5.1.1 Estimation Of The Guadagni-Little Model .....	29
5.1.2 Estimation Of The Little-Anderson Model .....	30
5.1.3 Nonlinearity Of The Estimations .....	30
5.2 Intermediate Level .....	31
5.2.1 The Full Procedure .....	31
6. Simulation Techniques.....	33
6.1 Simulation At The Household Level.....	34
6.1.1 The Stochastic Simulation Procedure.....	34
6.1.2 Speeding Up The Simulation Process.....	36
6.1.3 Aggregated Results.....	40
6.1.4 Microscopic View Of The Differences .....	42
6.1.5 Response Estimation.....	45
6.1.6 Adjusting The Model Parameters.....	46
6.1.7 More On Response Estimation.....	50
6.2 Simulation At The Intermediate Level .....	52
6.2.1 The Simulation Procedure.....	52
6.2.2 Making The Markov Model Deterministic.....	53
6.2.3 Model Behavior.....	53
6.3 Integrated Model .....	55

---

7. The Decision Support System In Use.....	57
7.1 The Offer .....	57
7.1.1 Offer Accepted .....	57
7.1.2 Offer Rejected .....	58
7.1.3 Comparing The Output.....	59
7.1.4 What To Do .....	60
7.1.5 Discussion.....	60
8. Conclusions.....	64
Bibliography.....	66
Appendix A: The Method For Generating A Dirichlet Distributed Vector .....	67
A.1 The Dirichlet Density Function .....	67
A.2 The Gamma Density Function .....	67
A.3: The Acceptance - Rejection Method.....	68
A.4: The MATLAB® Code For Generating The Dirichlet Vectors.....	70
f_dirich.m.....	70
f_gamma.m .....	71
Appendix B: The Problem Of Over-Switching When Using The Same Seed	
Method .....	73
B.1 The Over-Switching Problem.....	74
B.1.1 A Continuous Example .....	74
B.1.2 A Discrete Example .....	76
B.1.3 Motivation From A Marketing Model .....	78
B.2 The Hauksson Method For Minimizing Unwanted Switching.....	78
B.2.1 Derivation.....	78
B.2.2 Algorithm .....	79
B.2.3 Queueing System Explanation.....	80
B.2.4 A Discrete Example (Continued).....	81
B.3 Conclusions.....	81
Appendix C: Running Time For A "d_dd" Simulation .....	82
Appendix D: The Transition Matrices .....	83
Appendix E: Variation In Prices.....	85
Appendix F: Proof Of The Validity Of Counting.....	86

---

## List of Symbols

$adj$	index for the adjusted parameters
$b$	estimated parameter for GL loyalty in the MNL model
$C$	the set of available choices for the household
$d$	index for manufacturer promotions
$EISDL$	Estimated Incremental Sales caused by Difference in Loyalty
$h$	index for the households
$i, j, k$	indices for products
$l_{t,k}$	the household's Guadagni-Little loyalty at time $t$ for product $k$
$ML$	index for the maximum likelihood parameters
$n$	index for the purchase occasion of a household
$N$	Total number of purchases
$N_i$	Number of purchases of product $i$
$p_k$	probability of purchasing product $k$
$p_{i,j}$	transition probabilities of Markov chain ( $i \rightarrow j$ )
$P$	probability of all households doing as they did, estimated by the model
$P_d$	transition matrix (consisting of the $p_{i,j}$ elements)
$RSP$	response (difference in probabilities) to a potential promotion
$s$	sum of the $\bar{\alpha}$ parameters of the Dirichlet distribution
$s_0$	sum of population $\tau$ 's
$t$	index for the purchase occasion of a household, sometimes referred to as time
$v_{t,h}$	actual purchase of household $h$ at time $t$

---

$V_k$	EISDL for product $k$
$x_k$	vector of marketing mix for product $k$ . This vector typically includes price, display, feature and a alternative specific constant
$y_{t,k}$	dummy variable of purchase. One if product $k$ purchased at time $t$ , zero otherwise
$\alpha_k(\cdot)$	parameter of the household's loyalties for product $k$
$\beta$	vector of estimated parameters for the MNL model
$\gamma$	smoothing constant for the loyalty vector of the GL model and model constants for the Little-Anderson model
$\Delta_{sales,k}$	estimated change in sales of product $k$ , due to a potential promotion
$\lambda$	parameter for estimating the potential of an offer
$\pi_j$	steady state probabilities of Markov chain
$\tau$	vector for population $\alpha$ 's
$\phi_k$	household loyalty in the Little-Anderson model for product $k$
$\psi_k$	total sales of product $k$ , in items over the time horizon and the households
$\Omega_k$	potential of an offer for product $k$

---

# 1. Introduction

Decision Support Systems (DSSs) can be of great value to marketing managers.

However, as with any other tool, we must understand its functioning to be able to take full advantage of its capabilities.

The goal of the DSS in this paper is to simplify and aid in decision making in marketing.

The decisions involve in-store merchandising and manufacturer-retailer promotions. A somewhat typical question that arises might be: "A retailer is offering to promote the product in all his stores for 3 weeks. Throughout these 3 weeks he is willing to offer the consumers the price of 1.89. In return he wants me to give him \$10,000." When faced with this kind of question we would like to know what we are buying with the \$10,000. Precisely here is where the DSS comes in. With the aid of the DSS we evaluate both alternatives. Then we can make our decision on whether to accept or reject the offer, based on the difference in contributions to profit.

From industry we also have many stories about DSSs training people in decision making.

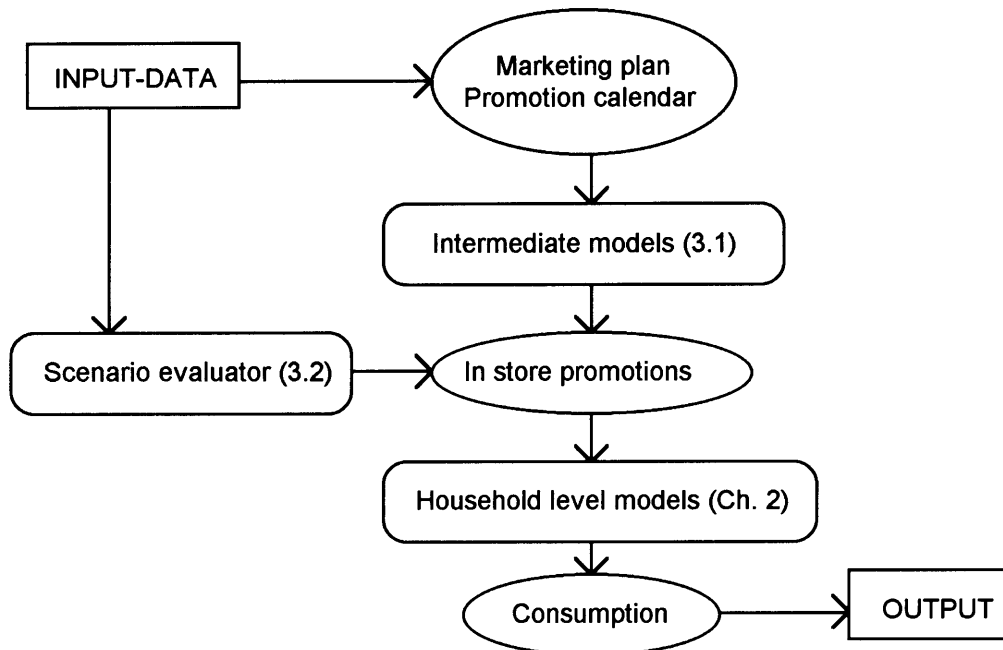
A story from the insurance industry follows:

Before the DSS was developed, people based their daily decisions on rules of thumb. The management felt the need for a better way making the decisions. It was thus decided to invest in a DSS. After two years of development, the system was introduced. It soon became apparent that the system was doing significantly better than the people. This was realized and accepted by the users. Moreover the users were curious how the system behaved differently and through using the DSS gained understanding of the flaws in their previously used rules of thumb. The users adjusted their way of thinking and eventually became much better decision makers than before, often even better than the system.

---

We thus see that not only can a DSS help in making better decision, but it can also train a skillful team of decision makers.

The operation of our DSS for mar marketing managers is outlined in figure 1.0.1.



**Figure 1.0.1: The DSS from a to z**

The users have two options of inputting data:

- a) They can put in the manufacturer promotion periods. The Markov model of section 3.1.6 would then translate this data to individual in-store merchandising.
- b) They can input store specific promotions. The users then specify a few scenarios and the probability of each of them. The scenario evaluator of section 3.2 then translates these to in-store merchandising.

When all the data is in the system, the model of section 2.2 simulates the households of the database. When all the households have been simulated, the system calculates the measures of interest and presents the results to the user.

---

## **2. Overview Of Models At The Household Level**

In this chapter we introduce marketing mix models at the household level. These models focus on how households make their purchase decision as a function of marketing variables they encounter after they reach the store.

A big forward step was taken when Guadagni and Little (1983) presented their model. Besides marketing variables, their model included an important new variable called loyalty. This variable is a weighted average of the previous purchases made by the household. The weights are exponential, assigning higher weights to the more recent purchases. Since the model was first introduced, other scientists have extrapolated the idea, but they have almost always used a loyalty like variable.

The biggest drawback of the Guadagni-Little model is that two different phenomena are contributing to the loyalty. The first is the underlying preference of the household. The second is often referred to as purchase feedback (as the consumer experiences the product his preference change depending on his satisfaction). Guadagni-Little loyalty catches both these effects.

Recently, we have been working with a new model developed by Little and Anderson (1993). This model rests upon loyalty-like variables, but instead of treating them as deterministic, a distribution (Dirichlet distribution) is assigned to them. As a household makes more and more purchases the model tracks its preference and adjusts the parameters of its distribution accordingly. This model decouples the two effects mentioned above. For a detailed discussion of this and other models consult Little and Anderson (1993).

In section 2.1 we introduce the Guadagni-Little model and section 2.2 focuses on the new Little-Anderson model.

---

## 2.1 The Guadagni-Little Model

The Guadagni-Little model introduced brand (and/or size) loyalty. This loyalty is composed of a household specific vector, one element for each product in the category, all of them non-negative and summing to one. It turns out that when modeling household choices, the loyalty vector contributes significantly to the evaluation of the household's future preferences (both brand and size loyalties were contributing significantly to the model). The recursive equation for the loyalties is,

$$l_{t+1,k} = \gamma l_{t,k} + (1 - \gamma) y_{t,k} \quad \forall k \quad (2.1.1)$$

Here  $l_{t,k}$  is the loyalty for product  $k$  at time  $t$  and  $y_{t,k}$  is one if product  $k$  was bought at time  $t$  and zero otherwise.

The model itself, is a multinomial logit (MNL) model with loyalty and other marketing mix variables (usually price, display and feature) as independent variables. The dependent variable is the probability of purchasing a particular product. For further discussion of the MNL model consult Ben-Akiva and Lerman (1974).

The purchase probabilities come out to be,

$$p_k = \begin{cases} \frac{e^{b l_k + \beta x_k}}{\sum_{j \in C} e^{b l_j + \beta x_j}} & k \in C \\ 0 & k \notin C \end{cases} \quad (2.1.2)$$

Here  $p_k$  is the probability of purchasing product  $k$ ,  $C$  is the choice set,  $x_k$  is a vector of marketing mix variables and  $b$  and  $\beta$  are the parameters of the model. A detailed discussion of the model can be found in Guadagni and Little (1983).

---

## 2.2 The Little-Anderson Model

As mentioned above, the main difficulty people have with the Guadagni-Little model is that two phenomenon are contributing to the loyalty. Decoupling these effects was the motivation behind the new Dirichlet model introduced by Little and Anderson.

We postulate that at the beginning of the time horizon a set of random variables,  $\bar{\phi}$ , are Dirichlet distributed, with density function,

$$f(\bar{\phi}|\bar{\alpha}, \bar{x}) = \frac{\Gamma(s)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_{n-1})\Gamma(\alpha_n)} \phi_1^{\alpha_1-1}\cdots\phi_{n-1}^{\alpha_{n-1}-1}(1-\phi_1-\cdots-\phi_{n-1})^{\alpha_n-1} \quad (2.2.1)$$

where the  $\alpha$ 's are a function of  $x$ ,

$$\alpha_k = \alpha_k(x) = \alpha_k(0)e^{\beta x_k} \quad \forall k \quad (2.2.2)$$

and  $s = s(x)$  is the sum,

$$s = \sum_{k=1}^n \alpha_k = s(x) = \sum_{k=1}^n \alpha_k(x) \quad (2.2.3)$$

The  $\phi$ 's are a sort of household specific loyalty variables, whereas the  $\alpha$ 's are parameters for the distribution of  $\bar{\phi}$ .

The mean and the covariance of the  $\phi$ 's are,

$$\bar{\phi}_k = \frac{\alpha_k}{s} \quad \forall k \quad (2.2.4)$$

$$Cov(\phi_i, \phi_j) = \frac{1}{s+1} \frac{\alpha_i}{s} \left( \delta_{ij} - \frac{\alpha_j}{s} \right) \quad \forall (i, j) \quad (2.2.5)$$

We use a multinomial model for the purchases. The nice thing about using the multinomial is that it is the conjugate of the Dirichlet distribution. This means that if we

---

start with Dirichlet distribution, make a multinomial draw using this distribution, and then perform a Bayesian update, our posterior distribution is also Dirichlet. Thus we don't lose the form of the distribution. We only need to update the parameters of the distribution.

The expected probability of purchasing a certain product is calculated, as explained above, from (2.2.4), which is a multinomial logit (MNL) model,

$$p_k(x) = \begin{cases} \frac{\alpha_k(0)e^{\beta x_k}}{\sum_{j \in \{C\}} \alpha_j(0)e^{\beta x_j}} & k \in C \\ 0 & k \notin C \end{cases} \quad (2.2.6)$$

A purchase is made using these probabilities. The resulting purchase vector,  $\bar{y}$ , has all elements equal to zero, except element  $k$ , where  $k$  is the number of the purchased product.

Each purchase occasion gives additional information about the households  $\alpha$ 's. The posterior distribution of these is also Dirichlet (as explained above) and the  $\alpha$ 's must be updated as,

$$\alpha_k^{post}(0) = \alpha_k(0) + y_k e^{-\beta x_k} \quad \forall k \quad (2.2.7)$$

After experiencing with the product, a household's preferences change because the product satisfies some of the needs that the consumer has. Purchase feedback catches this effect. Letting,

$$\bar{\Phi}_k^{post}(0) = \frac{\alpha_k^{post}(0)}{s^{post}(0)} \quad \forall k \quad (2.2.8)$$

the mean is shifted as,

---


$$\bar{\phi}_k^{feed}(0) = \gamma_1 \bar{\phi}_k^{post}(0) + (1 - \gamma_1) y_k \quad \forall k \quad (2.2.9)$$

The learning process also suggests that we should have a forgetting process, meaning that households may gradually forget how they liked a product that they tried a long time ago. Thus this household's preferences shift toward the population preferences as,

$$\bar{\phi}_k^{final}(0) = \gamma_2 \bar{\phi}_k^{feed}(0) + (1 - \gamma_2) \frac{\tau_k}{s_0} \quad \forall k \quad (2.2.10)$$

where,

$$s_0 = \sum_{j=1}^n \tau_j \quad (2.2.11)$$

and the variance may also increase,

$$\frac{1}{s_{t+1}(0)} = \gamma_3 \frac{1}{s_t^{feed}(0)} + (1 - \gamma_3) \frac{1}{s_0} \quad (2.2.12)$$

Now the household can go on to make its next purchase and the starting parameters are,

$$\alpha_{k,t+1}(0) = s_{t+1}(0) \bar{\phi}_{k,t}^{final}(0) \quad \forall k \quad (2.2.13)$$

For a detailed discussion consult Little and Anderson (1993).

---

## 3. Overview Of Intermediate Models

In this chapter we will describe models that generate the merchandising environment.

Section 3.1 will describe a stochastic model that translates the promotion calendar to in-store merchandising. This is a high level promotion model where our only input is the weeks in which a manufacturer promotion is running.

In section 3.2 we describe a more detailed scenario evaluation model. The inputs in this case are scenarios of store specific prices and promotions. The model then generates the merchandising environment from the inputs.

### 3.1 Promotion Model

The promotion model generates in-store merchandising environment from the promotion calendar, as set by the manufacturer. The model is a Markov chain where the state is the in-store status of the product. The Markov transitions are dependent on the promotion calendar and thus we can control the merchandising by adjusting the calendar.

#### 3.1.1 Categorization Of Stores

Stores are inherently different. Some of them run a lot of promotions and are often referred to as Hi-Low stores. Other stores concentrate on offering a good price every day, and are known as Every-Day-Low-Pricing stores (EDLP).

The model for the EDLP stores participating in a trade promotion program, such as Ocean Spray Cranberries' Continuous Allowance Program (CAP) can be made stationary, since under such a program the wholesale prices are more or less the same throughout the year.

On the other hand, when we look at the Hi-Low stores, prices are almost surely not stationary. The biggest reason is that the manufacturer runs promotions for his products.

---

There are many ways to incorporate this into the model. One way would be to use the time since the product was last on a manufacturer promotion.

By doing a big forward buy, the stores build up a big inventory. To turn this inventory over, they are likely to do some kind of merchandising. Determining the causal effect (inventory versus promotion) is not important. But it is important to realize that there should be some correlation between these two, independently of which is caused by the other. Such non-stationarity can be modeled in various ways. One is to make the Markov state space multidimensional. We could have the first dimension as in CAP above, the second would be the time until the next promotion will be held. Another approach would be to let the second dimension be an indicator state, (say promotion or not).

These more complex approaches have however not been implemented because the simple Markov model to be presented in section 3.1.6 has been found to give good results.

### **3.1.2 Status Of A Product**

There are many different states a product's in-store merchandising. The product can just be in its place on a shelf and at the regular price, or it can be featured in last weekend's newspaper and be on an impressive display, beside the cashier. This clearly influences people a lot.

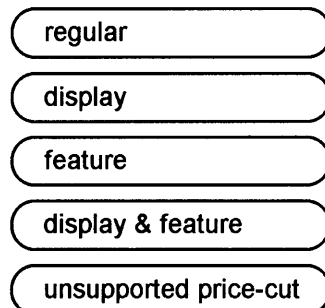
In it's scanner-data, Information Resources Inc. uses the following codes for product status in the store.

Code	Feature Code	Display Code	Special Pack
0	No Feature	No Display	Regular
1	C Feature	Lobby Display	Price Pack
2	B Feature	Front end-aisle Display	Pre-Price Pack
3	A Feature	Mid-aisle Display	Bonus Pack
4	Super A Feature	Back end-aisle Display	Bundle Pack
5		Specialty / All Other	Premium Pack
6		Shipper Display	
7		Promotional Aisle	

**Figure 3.1.1: Status of a product in a store as recorded by IRI**

In the models of chapter 2 we use a single dummy variable for feature and another one for display. This dummy variable is equal to zero if the code in figure 3.1.1 is 0, but one otherwise.

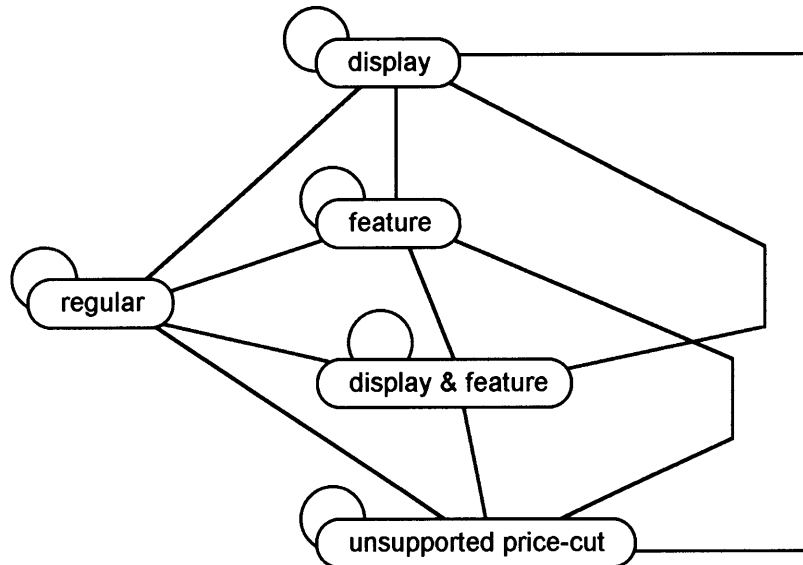
We postulate that a product can be in one of five states. These states are mutually exclusive and collectively exhaustive.



**Figure 3.1.2: Five states of a product in a store**

In the figure, *regular* corresponds to no display, no feature and the regular price (see discussion of regular price below), whereas *unsupported price-cut* means that the product is available at a lower price.

Focusing on only one product we hypothesize that a simple Markov chain models the transitions of the product between these states. We have one transition a week. This is further demonstrated in figure 3.1.3.



**Figure 3.1.3: The Markov chain for the evolution of a product's state**

The transition probabilities can be estimated from data. Sufficient data may not be available for calibration of all the variables for all the products. To cover these cases we can build rules of thumb for cases where there is not sufficient data.

In the test model of chapter 6 we used seven transition matrices. One was used for the first week of manufacturer promotion, the second in the second week, ..., the sixth in the sixth week, and the seventh when there was no manufacturer promotion.

Other properties can be incorporated into the model by making the Markov chain more complicated.

- We can introduce a fixed length of promotion (using the Markov chain unchanged, results in a geometric distribution for the number of weeks).
- We can make the transitions more time dependent by introducing a time component to the states.
- We can introduce an upper bound for the number of promotions in a time period (say max. 3 promotions over 26 weeks).

---

We decided not to follow any of these ideas up at this time.

### **3.1.3 Price**

One of the requirements of the intermediate model is to produce an in-store price. This can be done in various ways with different levels of complexity.

One alternative is to model the prices as a Markov chain. This Markov chain is conditioned on the in-store state of the product described in section 3.1.2. When the product is on promotion a discount is usually offered at the same time. We thus anticipate the price to be lower when the product is on display or if it has recently been featured.

People favor prices ending in a 9 for psychological reasons. It is "obviously" a better buy to buy a soft drink that costs under a dollar (\$0.99) as opposed to another one who costs a dollar (\$1.00). It is clear however that if all prices would end in a 9, the relative advantages would wash out. The retailers therefore often use the 9 pricing for the products that they are promoting. The effect of this pricing policy has been further investigated by Lian (1993).

It is interesting to consider the situation in Japan. The number 9 in Japanese is pronounced "ku", which also means to suffer. The retailers have thus adopted the policy of using 8 instead of 9 in pricing.

In Appendix E we present results from looking at the variations in prices of a single product in a store. It is apparent from the appendix that conditioning on the state of the product, the biggest part of the noise in the pricing has been removed.

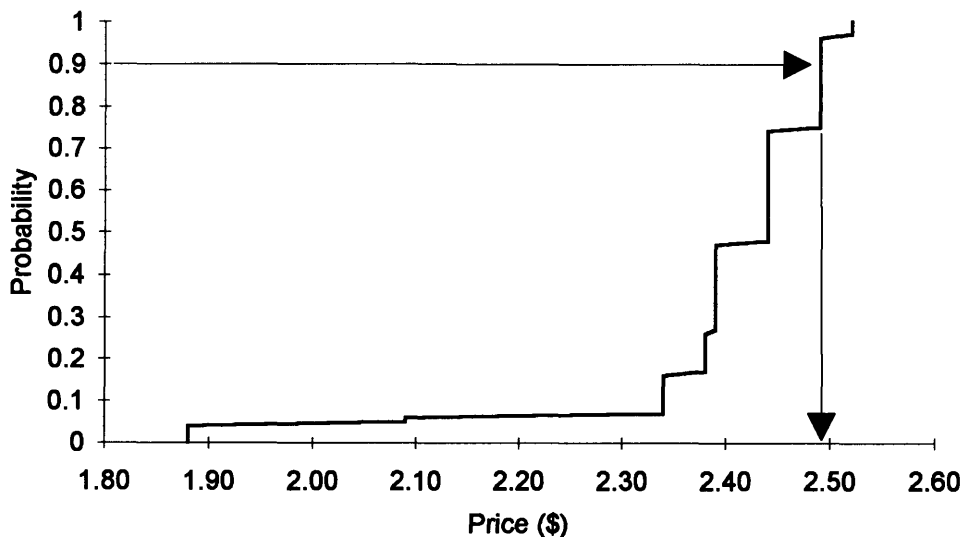
For these reasons we decided not to model the prices stochastically, but to use the expected price conditioned on state and store.

---

### 3.1.4 Determining The "Regular Price"

We propose a method that considers each store, each brand and each size separately. We look at prices when there are no promotions (*feature* or *display*). Take the price that has 10% above it and 90% below. Call this price the **regular price**. All prices above this base value will also be called regular. As far as lower bound is concerned a 10% price cut is allowed from the regular price and we still don't have *unsupported price cut*. On the other hand if the price drops more the 10% below the regular price, we define it as an *unsupported price-cut*.

Let us look at an example to illustrate this. In figure 3.1.4 we have an empirical example of the cumulative probabilities for a product,



**Figure 3.1.4: Cumulative probabilities for price conditioned on store and no display or feature**

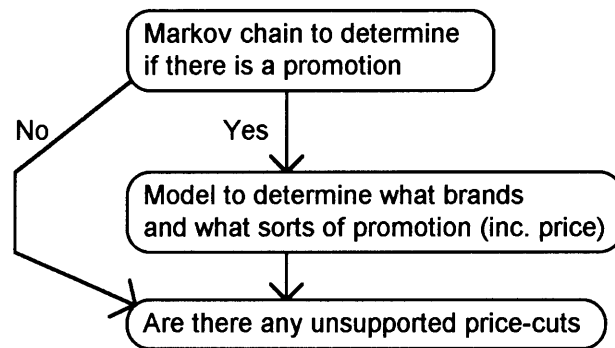
From the figure we see that at the 90% level, we get the regular price of \$2.49. Ten percent below 2.49 is 2.241 and thus we end up with the regular range of \$2.241 (effectively \$2.25) and up.

---

### 3.1.5 Other Products From The Same Manufacturer

We are not just making this model to run for one specific product. We have competing products and also more importantly, we have other products from the same manufacturer (different sizes etc.). It is, from the manufacturers point of view, not very smart to run promotions on two products at the same time. This would just be cannibalizing you own promotion. For this reason we should consider what other products are doing, when calculating the transition probabilities in figure 3.1.3.

One solution to this problem is to use a nested model,



**Figure 3.1.5: Nested model for many brands**

The first level model is for determining if any of the products is on promotion. The second level is to determine which products (taste and size) and at which prices. Finally in some cases stores run unsupported price-cuts of other products at the same time. Therefore it may not be such a bad idea to have the unsupported price-cut *almost* independent of the other promotions. This is done at the lowest box in figure 3.1.5.

### 3.1.6 The Model

The model that we decided to use is a simple one. The benefits of using a computationally cheap model are obvious when the goal is to incorporate the model into a real time DSS.

---

We decided to use a model that handles all stores the same. It uses the five states of a product introduced in section 3.1.2, and does not consider other products as suggested in section 3.1.5. The model uses the expected value for the prices and so does not model them probabilistically. We use seven transition matrices, depending on the promotion calendar.

The state of the product then evolves according to,

$$\bar{\pi}_{t+1} = \bar{\pi}_t P_d \quad (3.1.1)$$

Here  $\bar{\pi}_t$  is the probability vector of being in each of the states and  $P$  is the transition probability matrix. The  $d$  index is a indicator of the promotion calendar,

$$d = \begin{cases} 0 & \text{if no manufacturer promotion} \\ 1 & \text{in the first week of a manufacturer promotion} \\ 2 & \text{in the second week of a manufacturer promotion} \\ \vdots & \vdots \\ 6 & \text{in the last week of a manufacturer promotion} \end{cases} \quad (3.1.2)$$

As mentioned earlier the prices are just the conditional expected values. Thus once the state is determined we have all the marketing mix variables.

## 3.2 Scenario Evaluation Model

In this section we introduce a method to input specific promotions separately, so that the model can simulate store specific promotions and provide the user with response measures.

### 3.2.1 The Model

The scenario evaluation model is fundamentally different from the model of section 3.1. In this model we have a lot more of inputs. Instead of just putting in the promotion calendar as in the previous model, here we focus on each store individually.

For each store, we assign probabilities to scenarios that we think are probable. A schematic diagram of this is shown in figure 3.2.1.

		Week: 670		Week: 671		Week: 672		Week: 673		Week: 674	
Prob. of scenario		Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)
Scenario 1	0.5	df	1.89	df	1.89	df	1.89	f	1.89	M	
Scenario 2	0.2	f	1.99	f	1.99		1.99		1.99		1.99
Scenario 3	0.2		1.99		1.99		1.99		1.99		1.99
Otherwise	0.1										

d = display  
f = feature  
M = Markov chain

**3.2.1: Input into the scenario evaluation model: For each scenario, price and merchandising activity is specified for each week and store.**

What the inputs in this example mean, is that there is a 50% probability that we will have scenario 1. Under this scenario, we will have a display and feature for the first three weeks (670-672) and then for one week we will have a display. After this (at the start of week 674) the system returns to the state generated by the Markov chain. The price throughout this period is \$1.89. Scenarios 2 and 3 are similar, but what the figure refers to as otherwise is that no special action will be taken by the retailer and in that case the Markov chain models his actions.

---

The default situation would be modeled by a Markov chain similar to that of section 3.1. Thus the manufacturer is not doomed to no promotions even though none of the above scenarios take place.

We might observe correlation between stores. What this means is that if store A decides to run the most favorable scenario, then it is more likely that store B will run its most favorable as well. This can be incorporated into the model, but it is not simple nor necessarily attractive to do.

By focusing in this way separately on each store, the users of the system can use their valuable experience, since they often have a pretty good feeling of what a retailer is going to do.

By evaluating two different hypotheses (set of scenarios) we get two different outputs. By investigating the differences of those outputs the users have a tool to evaluate promotions. They can for example ask the question:

*"How much are we going to gain if the probability of scenario 1 in figure 3.2.1 changes from 0.5 to 0.6 and the probability of scenario 3 decreases from 0.2 to 0.1?"*

The procedure would be to input both possibilities. The outputs of the system will assign values to the differences. The users then face the trade-off between the effort of changing the probabilities and the differences in the output. If they would not have access to the DSS they would have to consider the trade-off between the effort and the changes in the probabilities. The system has thus enlightened the problem and enabled the decision makers to base their decision on a clearer trade-off.

---

## 4. Integration Of Models Into A Decision Support System

In this chapter we consider how we integrate the models introduced in chapters 2 and 3 into a Decision Support System (DSS). In figure 4.0.1 (which is identical to figure 1.0.1) we have a box diagram of the process.

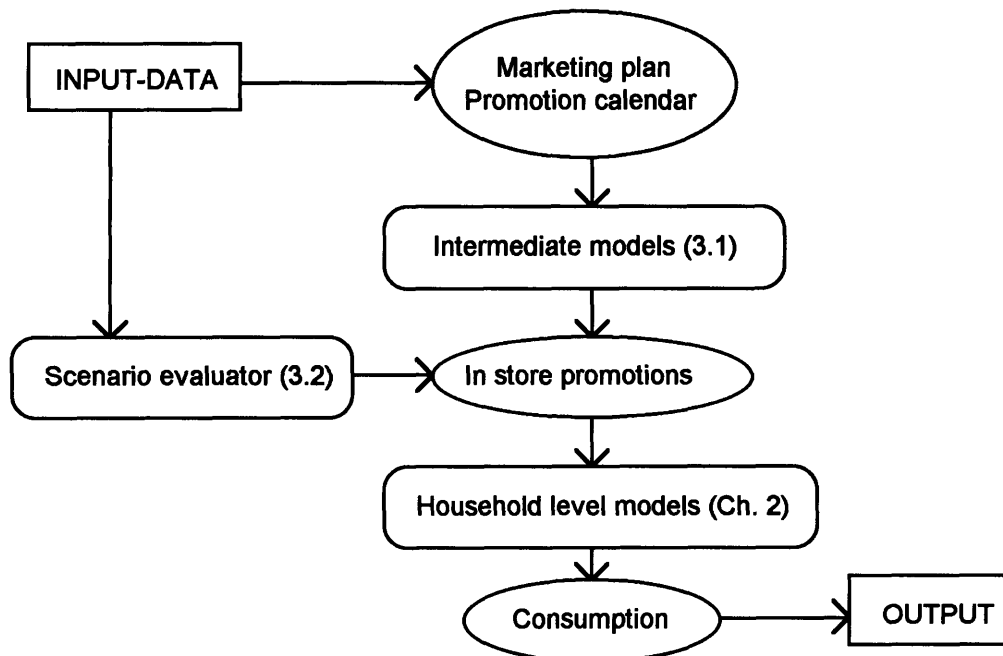


Figure 4.0.1: Integrated model

### 4.1 Promotion Calendar And Scenario Evaluator

Two sets of data need to be input to the model. First we need to determine the promotion calendar. We need to input the beginning and end of all the promotion periods. The other set of data is the store specific promotions. We need to input the promotions we want the scenario evaluator of section 3.1.2 to consider.

For the (Hi-Low) stores the manufacturer typically runs 2-5 promotions a year, each lasting 4 to 6 weeks. During these promotions the retailers can order products at a significantly lower price than the regular wholesale price. These promotions often include

---

some obligation for the retailer to promote the product in the store and undertake other buying enhancing activities.

When a manufacturer promotion is in its last week, the retailer often makes a big order. This builds up a temporary inventory. The retailer incurs some holding cost for this inventory, but saves money on the purchase. The practice is referred to as *forward buying*. After the manufacturer closes his promotion the retailer has thus some incentive to offer a better price, so that he can turn his inventory.

The EDLP (every day low price) stores on the other hand are usually participating in the CAP (continuous allowance program). That means that the wholesale price is constant throughout the year. This price lies between the regular wholesale price and the price offered in a promotion period.

In recent years promotions of another type have also become frequent. These involve so-called *street money*. This refers to marketing funds set aside by the manufacturer for local use. The scenario is that a retailer agrees to run a specific promotion involving, say, a feature and a display and the manufacturer agrees to pay the retailer a lump sum of money.

## **4.2 In Store Promotions**

When the promotion calendar has been designed and the significant promotions have been input to the scenario evaluator, the intermediate models generate the in-store merchandising. These models were introduced in chapter 3.

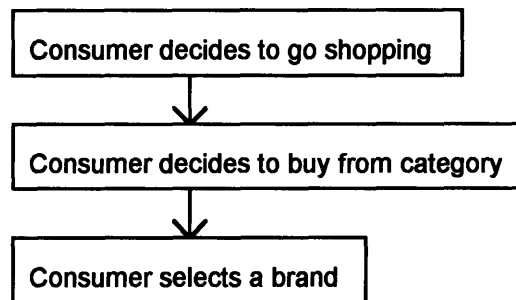
## **4.3 Consumption**

The end users of the products are the consumers. They go into the store, evaluate what they see and make their decisions. Our final step of the simulation is thus to simulate the consumers and their choice of products.

---

One big assumption that we make is that the promotions do not generate new purchases in the category, but simply cause people to switch brands. This is obviously not quite true since some purchases may be incremental to the category. However in large categories this is not likely to be critical because the actions of any one brand will tend not to have much effect on the total category.

Another approach to the problem of purchase generations would be to build a multi-stage model. A box diagram could look like figure 4.3.1,



**Figure 4.3.1: A multistage purchase generation model**

The first stage is a model for shopping trips for all the households in the database. This model would give us the instants and locations at which the households do their shopping. The second stage, conditioning on a shopping trip, we would model the probability of purchasing any product in the category under consideration. Finally the third and last stage, we condition on buying from the category and find which product was bought. For an introduction to these nested models, consult Guadagni and Little (1987).

While this nested structure is clearly closer to what we know about the physical behavior of consumers, it is not evident that the added complexity of the configuration will buy us all that much in getting a better forecast for what is going to happen.

---

## 5. Estimation

In this chapter we explain how the estimation of the relevant model parameters is done. In section 5.1 we look at the household level models and in 5.2 the intermediate models.

### 5.1 Household Level

In this section we explain how we go about estimating the household level models. In essence it is a maximum likelihood estimation.

#### 5.1.1 Estimation Of The Guadagni-Little Model

We need to estimate the following parameters,

$$\begin{array}{l} \beta_a : \text{ for } a \in \{\text{price, feature, display, ASC's}\} \\ \gamma, b \end{array} \quad (5.1.1)$$

The ASC's are alternative specific constants that shift the utility for a specific brand.

At the beginning of the period the loyalties are the same for all the households. A good starting point is to use the market share during the previous year.

Let us define  $v_{t,h}$  as the product that household  $h$  actually bought at purchase occasion  $t$ .

Then the probability that all households would do what they actually did is,

$$P(\beta, b, \gamma, v, x, C) = \prod_{\forall t, h} p_{v_{t,h}}(x) \quad (5.1.2)$$

Here the probabilities  $p$  are calculated as shown in equation 2.1.2. The loyalties,  $\bar{l}$ , evolve according to equation 2.1.1. The maximum likelihood estimate is then the solution to,

$$(\beta_{ML}, \gamma_{ML}, b_{ML}) = \arg \left( \max_{\beta, \gamma, b} (P(\beta, b, \gamma, v, x, C)) \right) \quad (5.1.3)$$

---

## 5.1.2 Estimation Of The Little-Anderson Model

The parameters that we need to estimate are,

$$\begin{aligned} t_i & : \text{ for all products } i \\ g_j & : \text{ for } j \in \{1,2,3\} \\ b_a & : \text{ for } a \in \{\text{price, feature, display}\} \end{aligned} \quad (5.1.4)$$

At the start of the calibration period, all households have,

$$\alpha_k(0) = \tau_k \quad \forall k \quad (5.1.5)$$

The probability that the model assigns to what actually happened is,

$$P(\beta, \gamma, \tau, v, x, C) = \prod_{\forall h,n} p_{v_{h,n}}(x) \quad (5.1.6)$$

Here the probabilities  $p$  are calculated as shown in equation 2.2.6, and  $\bar{\phi}$  and  $\bar{\alpha}$  evolve as in equations 2.2.7 to 2.2.12. The maximum likelihood estimate is then found by solving,

$$(\beta_{ML}, \gamma_{ML}, \tau_{ML}) = \arg\left(\max_{\beta, \gamma, \tau} (P(\beta, \gamma, \tau, v, x, C))\right) \quad (5.1.7)$$

## 5.1.3 Nonlinearity Of The Estimations

The maximization problems in sections 5.1.1 and 5.1.2 are highly nonlinear. Moreover they are not only nonlinear, but also recursively conditional on what happened. This introduces great complexity in the closed form of the likelihood function.

We used two methods for the estimation. The first method relied on the optimization toolbox of MATLAB®. The procedure is very simple. First we constructed a function to evaluate the likelihood function for the models, and then used a minimization function in MATLAB®.

---

The other method is faster, but also requires a much higher detail of input data. It is a variant of the Newton-Raphson method, and we need to provide the likelihood function as well as all its derivatives. For further details see Fader, Lattin and Little (1992).

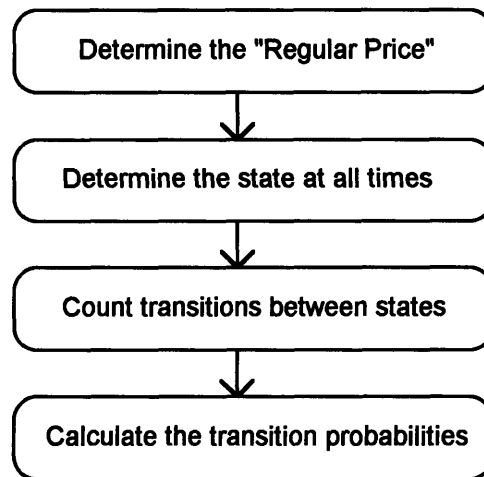
The estimation takes a lot of time. This is however not of deep concern, because estimation is not done frequently. Typically the estimation is performed once in a year or so, whereas we could be running the DSS every single day.

## 5.2 Intermediate Level

Estimating the intermediate model consist of estimating the transition matrices  $P_d$ . In section 5.2.1 we give a method for estimating these probabilities.

### 5.2.1 The Full Procedure

The procedure for the Markov chain is thus as shown in figure 5.2.1.



**Figure 5.2.1: The procedure for estimating the Markov transition probabilities for a product's in-store state**

We begin by finding the regular price. This is done as explained in section 3.1.4. From the regular price and the display and feature data, we can determine the states of the

---

product at all times. We then simply count transitions and construct the probabilities as the relative transition frequencies (see Appendix F for proof).

---

## 6. Simulation Techniques

This chapter describes the procedures that we use in simulating the models. The ultimate goal of our model building is to integrate the models into a decision support system. The role of the simulations is thus to evaluate and compare scenarios. The marketer gives his high level ideas of what he wants to evaluate, and the system responds with market share and other relevant measures.

We use a database originated from Information Resources Incorporated. The data is a sample from Eau Claire, WI from 14 May 1990 to 12 May 1991. We are using the juice market, and our focal product is Ocean Spray Cranberries' 48 oz. Cranberry juice. During this period 237 households made 1634 purchases from the category in 9 stores. In figure 6.0.1 we have the juices considered, which will hereafter be referred to by number.

product #	Manufacturer	Taste	Size
1	Ocean Spray	Cranberry	48 oz.
2	Gatorade	Fruit Punch	32 oz.
3	Hawaiian Punch	Red Punch	46 oz.
4	Gatorade	Original	32 oz.
5	Ocean Spray	Cranberry	32 oz.
6	Ocean Spray	Cranrasberry	48 oz.
7	Gatorade	Lemon-Lime	32 oz.
8	Ocean Spray	Cranberry	48 oz.
9	Ocean Spray	Cranberry	64 oz.
10	Ocean Spray	Cranrasberry	64 oz.

**Figure 6.0.1: Products included in the database**

In the calibration for the Markov matrices for the intermediate model we use a bigger store database. It includes 28 markets, 179 stores and is from 17 June 1991 to 27 December 1992. The only product we use from this database is our focus product, product 1.

---

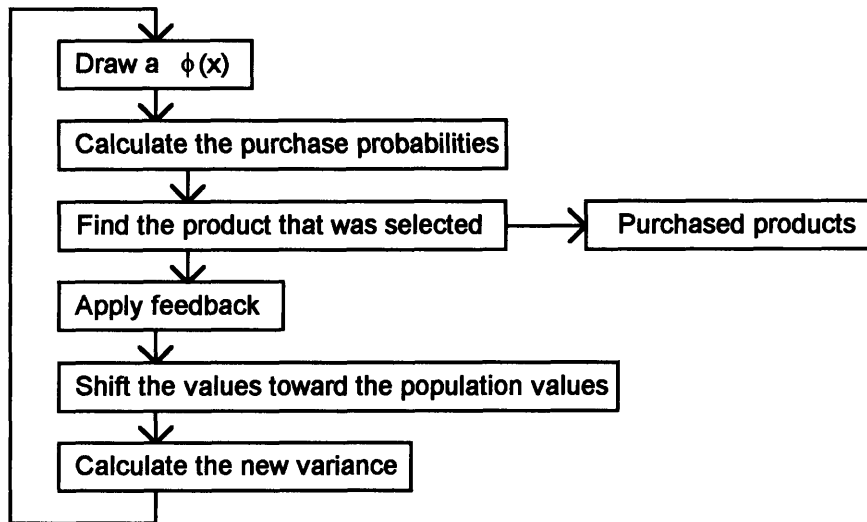
## 6.1 Simulation At The Household Level

Simulating the Little-Anderson model is not trivial. This stochastic model rests upon a Dirichlet distribution and it is not straight forward to generate Dirichlet distributed vectors. The method we used is introduced in Appendix A.

Doing a full Monte-Carlo run is computationally intensive and we introduce methods to speed this up.

### 6.1.1 The Stochastic Simulation Procedure

In figure 6.1.1 we have a box-diagram of how we do the stochastic simulations.



**Figure 6.1.1: The stochastic simulation cycle**

First we consider the first household. Using earlier data or population means, we get a starting value for the  $\alpha$ 's. For the first purchase occasion we then draw a  $\bar{\phi}$  using the marketing the household sees. The draw is thus from,

$$f(\bar{\phi}|\bar{\alpha}, \bar{x}) = \frac{\Gamma(s)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_{n-1})\Gamma(\alpha_n)} \phi_1^{\alpha_1-1}\cdots\phi_{n-1}^{\alpha_{n-1}-1}(1-\phi_1-\cdots-\phi_{n-1})^{\alpha_n-1} \quad (6.1.1)$$

where,

---


$$\alpha_k = \alpha_k(x) = \alpha_k(0)e^{\beta x} \quad \forall k$$

$$s = \sum_{k=1}^n \alpha_k = s(x) = \sum_{k=1}^n \alpha_k(0)e^{\beta x} \quad (6.1.2)$$

Next the purchase probabilities are calculated for all the products in the category. The purchase probabilities are,

$$p_k = \begin{cases} \frac{\phi_k}{\sum_{j \in [C]} \phi_j} & k \in C \\ 0 & k \notin C \end{cases} \quad (6.1.3)$$

Since we have already drawn a  $\bar{\phi}$ , the variance is zero. This effectively means that  $\bar{\alpha}$  is infinite and the Bayesian update does therefore not contribute. We must demarket  $\bar{\phi}$  by,

$$\phi_k(0) = \frac{\phi_k e^{-\beta x_k}}{\sum_{j=1}^n \phi_j e^{-\beta x_j}} \quad \forall k \quad (6.1.4)$$

The purchase feedback is,

$$\phi_k^{feed}(0) = \gamma_1 \phi_k(0) + (1 - \gamma_1) y_k \quad \forall k \quad (6.1.5)$$

Shifting towards the population mean we get,

$$\phi_k^{final}(0) = \gamma_2 \phi_k^{feed}(0) + (1 - \gamma_2) \frac{\tau_k}{s_0} \quad \forall k \quad (6.1.6)$$

Since we are actually drawing a  $\bar{\phi}$  we don't have any variance in it and thus equation 2.2.12 becomes,

---


$$\left. \begin{aligned} \frac{1}{s_{t+1}(0)} &= \gamma_3 \frac{1}{s_t^{feed}(0)} + (1-\gamma_3) \frac{1}{s_0} \\ \frac{1}{s_t^{feed}} &= 0 \end{aligned} \right\} \Rightarrow s_{t+1}(0) = \frac{s_0}{1-\gamma_3} \quad (6.1.7)$$

Finally the starting  $\alpha$ 's for the next purchase occasion are,

$$\alpha_{k,t+1}(0) = s_{t+1}(0) \phi_{k,t}^{final}(0) \quad \forall k \quad (6.1.8)$$

We have now completed the first purchase and are ready for the next. We continue in this manner until we are done with all the purchases of this household. Then we take the other households one by one and follow the same procedure until the households have all been simulated. This all is referred to as a single run (in our case one run consists of 1634 purchases by 237 households over 52 weeks).

We keep in mind that this is just a single run. The households could of course have done something entirely different. For getting a better picture of what is likely to happen, we must simulate this over and over again, until we have a clear picture of what is likely to occur.

After sufficient number of runs, we can then calculate the expected aggregate sales and other measures of interest.

### 6.1.2 Speeding Up The Simulation Process

If the model would be deterministic, only one run would be needed to get the desired result. More than that, it is computationally expensive to generate the Dirichlet vectors and thus the computation time of a single run would decrease as well.

What we are interested in is the total sales. We thus want to estimate,

---


$$\bar{\Psi} = \sum_h \sum_{n=1}^{v(h)} \bar{y}_{h,n} \quad (6.1.9)$$

Here  $\bar{\Psi}$ , represents the total number of items that the households bought of each brand.

Taking the expected value of (6.1.9) gives,

$$E[\bar{\Psi}] = E \left[ \sum_h \sum_{n=1}^{v(h)} \bar{y}_{h,n} \right] = \sum_h \sum_{n=1}^{v(h)} E[\bar{y}_{h,n}] \quad (6.1.10)$$

We thus need the expected value of  $\bar{y}$ , which is,

$$E[\bar{y}|\bar{\alpha}] = \frac{\bar{\alpha}}{s} \quad (6.1.11)$$

But  $\bar{\alpha}$  is not only a random variable, but also a recursive one.

$$\bar{\alpha}_{n+1}(0) = \frac{s_0}{1-\gamma_3} \left\{ \gamma_1 \gamma_2 \bar{\phi}_n(0) + (1-\gamma_1) \gamma_2 \bar{y}_n + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} \quad (6.1.12)$$

We thus need some methods for approximating the  $\alpha$ 's so that we can get a good estimate. We will use Taylor series to represent functions we can not evaluate explicitly.

Let us look at  $\phi_k(0)$ ,

$$\begin{aligned} \phi_k(0) &= \frac{\phi_k e^{-\beta x_k}}{\sum_{j=1}^n \phi_j e^{-\beta x_j}} \\ &= \frac{\bar{\phi}_k e^{-\beta x_k}}{\sum_{j=1}^n \bar{\phi}_j e^{-\beta x_j}} + \sum_{j=1}^n \frac{\partial \phi_k(0)}{\partial \phi_j} (\phi_j - \bar{\phi}_j) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 \phi_k(0)}{\partial \phi_i \partial \phi_j} (\phi_i - \bar{\phi}_i) (\phi_j - \bar{\phi}_j) + \dots \end{aligned} \quad (6.1.13)$$

where the derivatives are evaluated at the expected value. Under some convergence conditions the Taylor series approaches the correct function with probability 1 as we take more and more terms.

We need to calculate the derivatives,

$$\frac{\partial \phi_k(0)}{\partial \phi_j} = \delta_{kj} \frac{e^{-\beta x_k}}{\sum_{n=1}^{10} \phi_n e^{-\beta x_n}} + \frac{-\phi_k e^{-\beta x_k} e^{-\beta x_j}}{\left( \sum_{i=1}^{10} \phi_i e^{-\beta x_i} \right)^2} \quad (6.1.14)$$

and the second derivatives are,

$$\frac{\partial^2 \phi_k(0)}{\partial \phi_i \partial \phi_j} = \delta_{kj} \frac{-e^{-\beta x_k} e^{-\beta x_i}}{\left( \sum_{n=1}^{10} \phi_n e^{-\beta x_n} \right)^2} + \delta_{ki} \frac{-e^{-\beta x_k} e^{-\beta x_j}}{\left( \sum_{i=1}^{10} \phi_i e^{-\beta x_i} \right)^2} + \frac{2\phi_k e^{-\beta x_k} e^{-\beta x_i} e^{-\beta x_j}}{\left( \sum_{i=1}^{10} \phi_i e^{-\beta x_i} \right)^3} \quad (6.1.15)$$

Inserting this for the first two terms of (6.1.13) results in,

$$\begin{aligned} \phi_k(0) &= \frac{\bar{\phi}_k e^{-\beta x_k}}{\sum_{j=1}^n \bar{\phi}_j e^{-\beta x_j}} \\ &+ \sum_{j=1}^n \delta_{kj} \frac{e^{-\beta x_k}}{\sum_{i=1}^{10} \bar{\phi}_i e^{-\beta x_i}} + \frac{-\bar{\phi}_k e^{-\beta x_k} e^{-\beta x_j}}{\left( \sum_{i=1}^{10} \bar{\phi}_i e^{-\beta x_i} \right)^2} (\phi_j - \bar{\phi}_j) \\ &+ \sum_{j=1}^n \frac{-e^{-\beta x_k} e^{-\beta x_i}}{\left( \sum_{i=1}^{10} \bar{\phi}_i e^{-\beta x_i} \right)^2} (\phi_k - \bar{\phi}_k) (\phi_j - \bar{\phi}_j) \\ &+ \sum_{i=1}^n \sum_{j=1}^n \frac{\bar{\phi}_k e^{-\beta x_k} e^{-\beta x_i} e^{-\beta x_j}}{\left( \sum_{i=1}^{10} \bar{\phi}_i e^{-\beta x_i} \right)^3} (\phi_i - \bar{\phi}_i) (\phi_j - \bar{\phi}_j) \\ &+ \dots \end{aligned} \quad (6.1.16)$$

Taking the expected value we get,

---


$$\begin{aligned}
E[\phi_k(0)] &= \frac{\bar{\phi}_k e^{-\beta x_k}}{\sum_{j=1}^n \bar{\phi}_j e^{-\beta x_j}} \\
&+ \sum_{j=1}^n \frac{-e^{-\beta x_k} e^{-\beta x_j}}{\left(\sum_{n=1}^{10} \bar{\phi}_n e^{-\beta x_n}\right)^2} \text{Cov}(\phi_k, \phi_j) \\
&+ \sum_{i=1}^n \sum_{j=1}^n \frac{\bar{\phi}_k e^{-\beta x_k} e^{-\beta x_i} e^{-\beta x_j}}{\left(\sum_{n=1}^{10} \bar{\phi}_n e^{-\beta x_n}\right)^3} \text{Cov}(\phi_i, \phi_j) \\
&+ \dots
\end{aligned}
\tag{6.1.17}$$

In the beginning of the forecast period the  $\alpha$ 's are known. But right after the first purchase  $\bar{\alpha}$  is a random variable. We found that using the first term in its Taylor series, i.e. the expected value of  $\bar{\alpha}$ , is a very good approximation. Taking the expected value of (6.1.12) yields,

$$\begin{aligned}
E[\bar{\alpha}_{n+1}(0)] &= E\left[\frac{s_0}{1-\gamma_3} \left\{ \gamma_1 \gamma_2 \bar{\phi}_n(0) + (1-\gamma_1) \gamma_2 \bar{y}_n + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\}\right] \\
&= \frac{s_0}{1-\gamma_3} \left\{ \gamma_1 \gamma_2 E[\bar{\phi}_n(0)] + (1-\gamma_1) \gamma_2 E[\bar{y}_n] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\}
\end{aligned}
\tag{6.1.18}$$

Using the first two terms of the Taylor series expansion for  $E[\bar{\phi}_n(0)]$  we get,

$$\begin{aligned}
E[\bar{\alpha}_{n+1}(0)] &\approx \frac{s_0}{1-\gamma_3} \left\{ \gamma_1 \gamma_2 \left[ \frac{\bar{\phi}_k e^{-\beta x_k}}{\sum_{j=1}^n \bar{\phi}_j e^{-\beta x_j}} + \sum_{j=1}^n \frac{-e^{-\beta x_k} e^{-\beta x_j}}{\left(\sum_{n=1}^{10} \bar{\phi}_n e^{-\beta x_n}\right)^2} \text{Cov}(\phi_k, \phi_j) + \sum_{i=1}^n \sum_{j=1}^n \frac{\bar{\phi}_k e^{-\beta x_k} e^{-\beta x_i} e^{-\beta x_j}}{\left(\sum_{n=1}^{10} \bar{\phi}_n e^{-\beta x_n}\right)^3} \text{Cov}(\phi_i, \phi_j) \right] + (1-\gamma_1) \gamma_2 E[\bar{y}_n] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} \\
&\tag{6.1.19}
\end{aligned}$$

We shall now try to identify the effects of using the expected value for those random variables. We have five settings, and these are indexed in figure 6.1.2,

---

Index	Equation 2.2.1	Equation 2.2.6
d_rr	random	random
d_rd	random	1st order Taylor
d_dr	1st order Taylor	random
d_dd	1st order Taylor	1st order Taylor
d_d2	2nd order Taylor	1st order Taylor

**Figure 6.1.2: Indices for simulations**

Using the d\_dd or d\_d2 method has a tremendous advantage. Because of the deterministic behavior of the simulation, we only need one run. It is sufficient to make two runs to compare two slightly different scenarios. Using the d\_rr method on the other hand is computationally much more expensive. Not only is each iteration much more expensive, but we must also make a lot of them to get a grip of the resulting differences of the scenarios.

A method for decreasing the variance of the difference of two scenarios is to use the *same seed*. However, since the households' choices are discrete, we encounter the problem of over-switching. A solution to this is introduced in Appendix B.

A potential problem with using the deterministic methods, is that it prohibits us from identifying *switchers*. Switchers are people that would have bought A but for some outside reason (merchandising) decide to buy B, they thus switch from A to B. When a household at each purchase instant makes a fractional purchase, it is impossible to identify this phenomenon. An approximation would be to use the most likely product. A household would then be said to switch when the most likely product changes.

### 6.1.3 Aggregated Results

Let us look at sales figures for a whole year. We use a juice database from Eau Claire including the leading ten brandsizes. We have 234 households that over 52 weeks made 1634 purchases. In figure 6.1.3 we have the simulated aggregated sales using different simulation methods,

product	d_dd	d_dr	d_rd	d_rr	actual	d_d2
1	269.0	267.4 (22.0)	243.3 (24.8)	242.9 (30.4)	259	253.5
2	182.6	183.5 (20.2)	191.4 (23.0)	192.0 (28.0)	158	192.1
3	167.1	165.9 (17.1)	170.4 (20.7)	170.9 (25.6)	179	170.5
4	158.7	159.6 (19.5)	166.6 (21.8)	165.4 (27.0)	157	167.1
5	123.3	126.3 (15.6)	129.9 (19.3)	132.3 (23.9)	154	119.9
6	166.4	168.1 (18.0)	164.3 (23.0)	164.3 (26.0)	159	158.2
7	186.5	181.9 (20.3)	191.3 (22.4)	188.6 (28.6)	160	198.7
8	93.3	93.8 (13.6)	93.0 (16.2)	93.3 (20.2)	152	87.8
9	176.4	176.7 (18.6)	174.8 (20.7)	175.1 (26.1)	131	176.3
10	110.6	110.7 (14.1)	108.9 (16.2)	109.2 (20.2)	125	109.9
# of runs	1	2000	2000	10000		1
running time	14.0	20.0	141	146		36.6
RMS Error (act)	28.76	28.11	29.62	29.17		32.30
RMS Error (rr)	9.59	9.12	1.23	0.00	29.17	6.63

**Figure 6.1.3: Simulation results of aggregated sales:  
10 brandsizes, 234 households, 52 weeks, 1634 purchases  
Standard deviations of a single run are in the parenthesis  
Running time is the time in seconds that a single run takes  
on a 50 MHz 486 (DELL 486P/50)**

In figure 6.1.3,

$$\text{RMS error(actual)} = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (N_i(\text{method}) - N_i(\text{actual}))^2} \quad (6.1.20)$$

and,

$$\text{RMS error(rr)} = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (N_i(\text{method}) - N_i(\text{rr}))^2} \quad (6.1.21)$$

where  $N_i(\text{method})$  is the mean of the simulated annual sales of product  $i$ , using the indicated method.

Let's now compare the results. We use RMS errors as a measure of quality. Comparing the results with what actually happened in the market, we see that the d\_dd simulation gives results closest to what actually happened. The other results are however not much further off.

---

One thing we need to keep in mind is that the actual is similar to just one  $d_{rr}$  run. We see that for the biggest difference between the actual and  $d_{rr}$  is for product 8, where the difference is 2.906 standard deviations. The probability that a normally distributed random variable is outside 2.906 standard deviations is 0.37%. The probability that all of 10 random variables are less than 2.906 standard deviations from their means is 96.46%.

If we were to hypothesize that "actual" was a single  $d_{rr}$  run using a 95% confidence interval, we would reject the hypothesis (96.35% > 95%). Among the reasons why "actual" is so far off, is that there are marketing activities (coupon drops, etc.) that the model does not include. We must thus keep in mind that the model is far from being perfect.

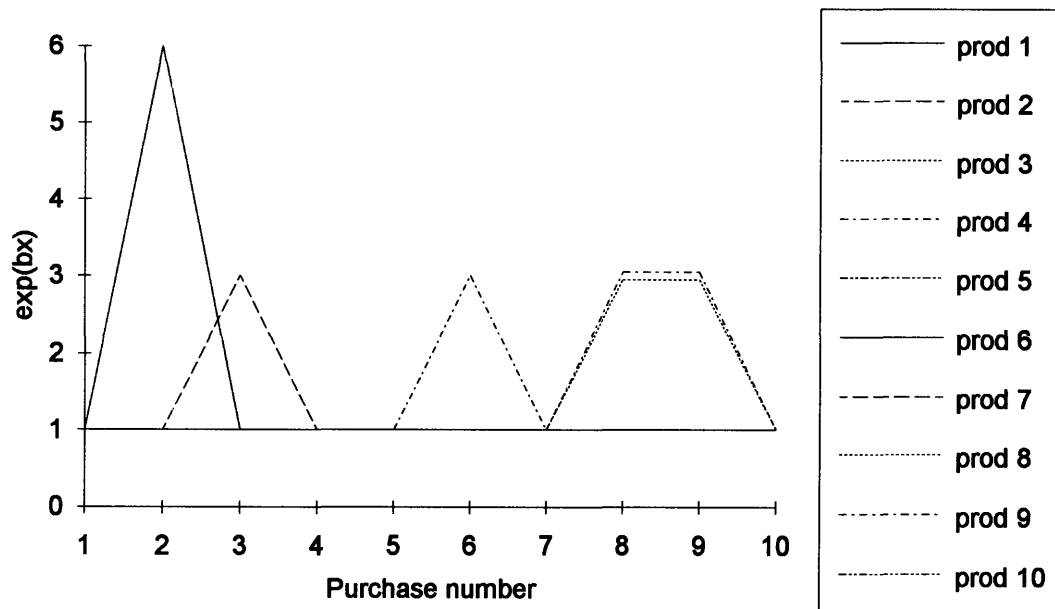
Since  $d_{rr}$  is a complete simulation of the model, we feel that  $d_{rr}$  gives the desired result. We thus from now on consider  $d_{rr}$  to give the correct results.

When comparing how closely the other methods are to  $d_{rr}$  we observe from figure 6.1.3 that  $d_{dd}$  has a RMS error of 9.59, whereas the  $d_{d2}$  RMS error is only 6.63. This suggests that there is a lot to gain by using the 2nd order Taylor series expansion. Looking at the standard deviations and running times of the  $d_{rd}$  and  $d_{dr}$  methods, we decide to discard them. We know that  $d_{rr}$  gives the desired results and if we have the time to do either  $d_{rd}$  or  $d_{dr}$  we might as well just go all the way and use  $d_{rr}$ .

#### **6.1.4 Microscopic View Of The Differences**

Let us take a closer look at the differences in the  $d_{d2}$  and  $d_{rr}$  simulations. The goal is to understand where those differ, and from that understanding to conclude about the price we pay for using the computationally feasible  $d_{d2}$  method.

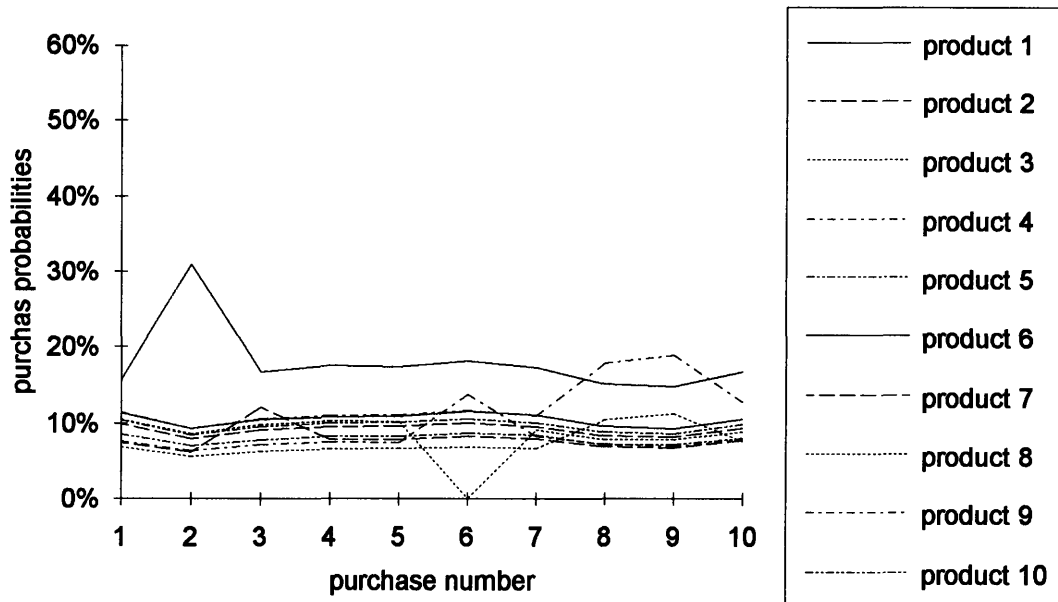
We designed a hypothetical marketing scenario, shown in figure 6.1.4,



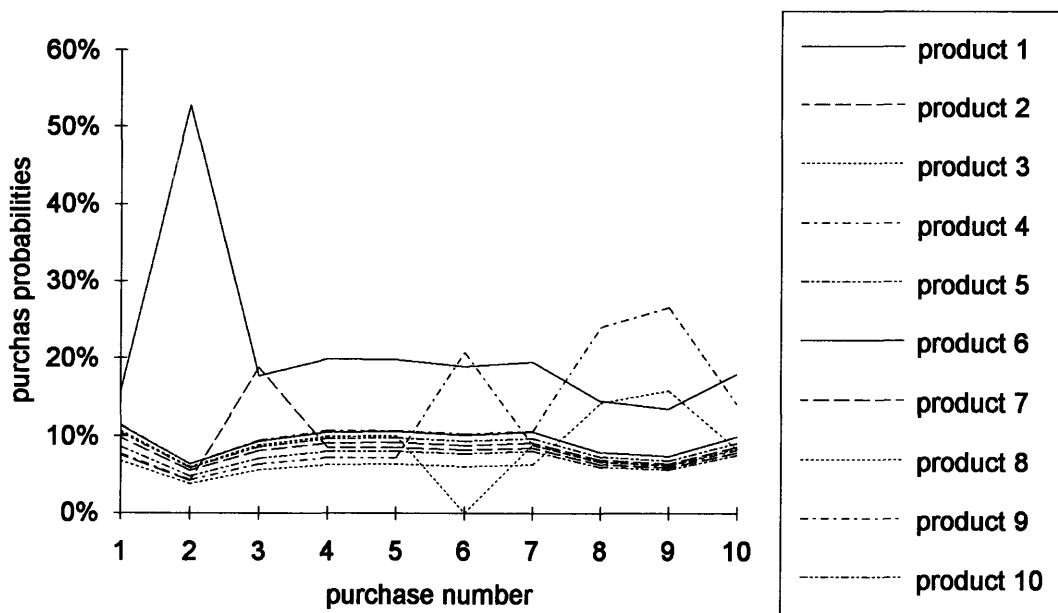
**Figure 6.1.4: Marketing activities,  $e^{\beta x_j}$**

This marketing scenario is a little unusual in that often there is more marketing noise. This means that it is not likely that almost all the products are committed to reference marketing,  $e^{\beta x} = 1$ , as above.

Next let us look at the results. Figure 6.1.5a, shows the results when using the d\_rr method and figure 6.1.5b show the results of using d\_d2.



**Figure 6.1.5a: Simulation result using the d\_rr method**



**Figure 6.1.5b: Simulation results using the d\_d2 method**

From the figures we observe that the d\_d2 method overestimates the power of promotions. To look further into this issue let's do a response estimation.

---

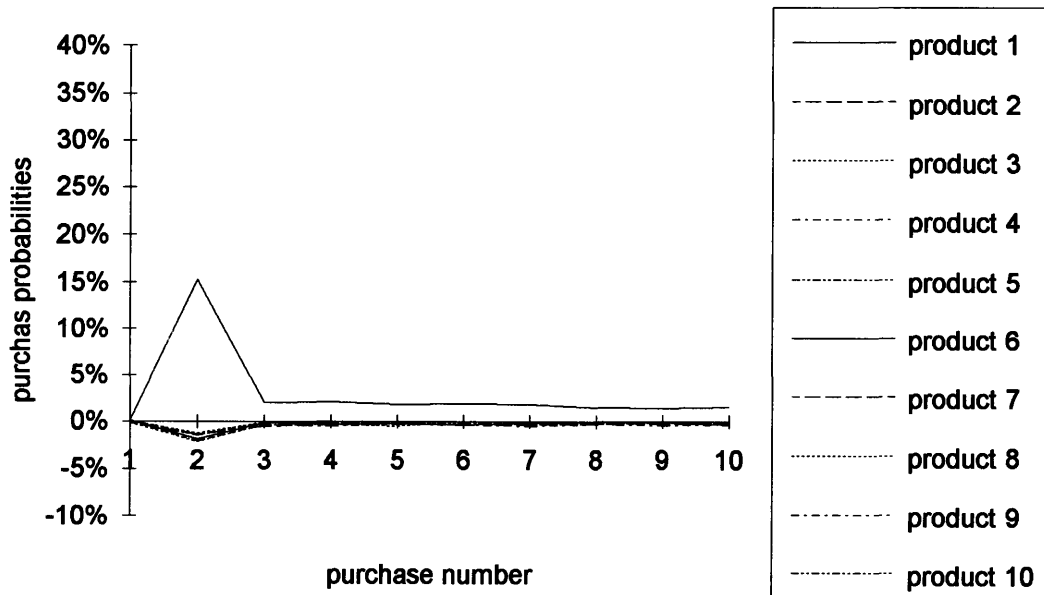
## 6.1.5 Response Estimation

Here we investigate how the model reacts to changes in the marketing environment.

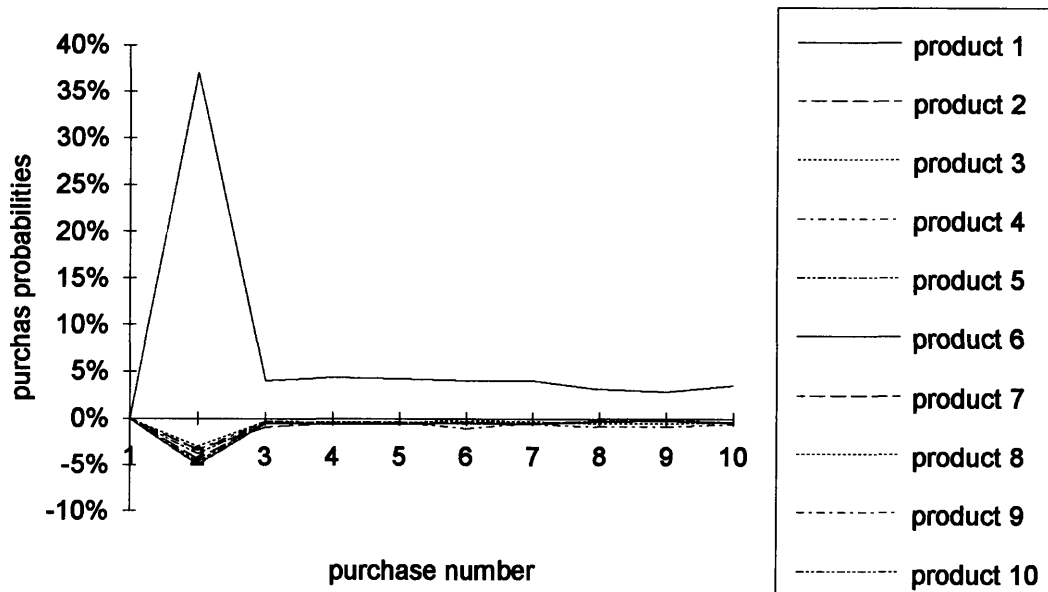
Consider the example of evaluating the promotion of product 1 at  $t = 2$  in figure 6.1.4.

Ideally we would first do a full Monte-Carlo run for the marketing plan with the promotion, and then do another run without the promotion of product 1 in period 2. By comparing the differences we get a picture of the effect of the promotion.

In figure 6.1.6, we have displayed the changes in purchase probabilities (i.e. probabilities with the promotion minus the probabilities without the promotion). Figure 6.1.6a displays the results for the Monte Carlo run (d\_rr), but figure 6.1.6b for the deterministic runs (d\_d2).



**Figure 6.1.6a: Response to the promotion using a full Monte-Carlo simulation**



**Figure 6.1.6b: Response to the promotion using the d\_d2 method**

From figure 6.1.6a we see that during the promotion period we raise the purchase probabilities about 15%. By running the promotion we invest in the loyalties of the household and this increases the probabilities about approximately 2% for the rest of the period.

From figure 6.1.6b we see that the deterministic method greatly overestimates the response to the promotion. This is clearly a problem, since we are primarily going to use the model for answering these kinds of response questions. We must thus look for ways of closing this gap that is apparent between the d\_d2 and d\_rr results.

### 6.1.6 Adjusting The Model Parameters

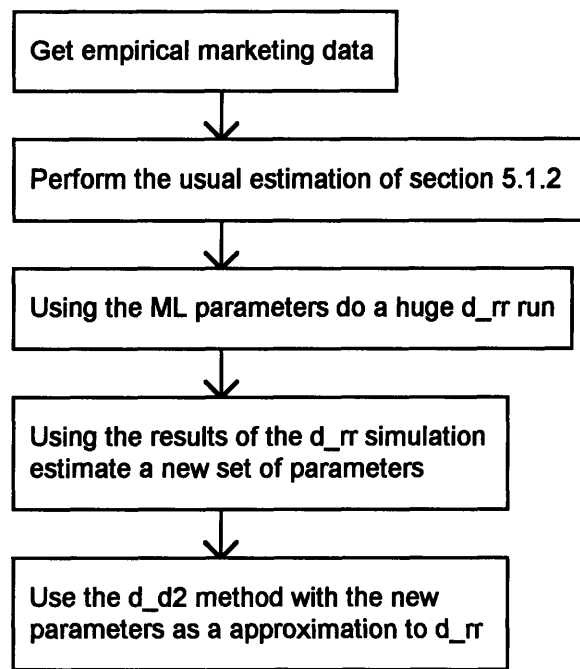
A priori we believe that the d\_rr results are the "true" results. We also know from section 6.1.5 that using the d\_d2 method gives results that overestimate the response to promotions.

---

We thus need a way of getting a quick estimate of what the full  $d_{rr}$  results would be if we would not have the time to wait for the exact simulation results ( $d_{rr}$ ).

The method proposed here is to adjust the parameters (and thus the dynamics) of the  $d_{d2}$  simulation method, such that it resembles the  $d_{rr}$  results as closely as possible. We then plan to use the computationally feasible  $d_{d2}$  method as an approximation of the correct  $d_{rr}$  simulation.

The proposed procedure is illustrated in figure 6.1.7



**Figure 6.1.7: The procedure for parameter adjustments**

Using the data we have, we first do the usual Maximum Likelihood estimation of the model parameters of section 5.1.2. Using those parameters we run a huge correct ( $d_{rr}$ ) simulation. In our research we did 10,000 runs to get a comfortable estimate of the purchase frequencies. Using the relative purchase frequencies we fit the deterministic  $d_{d2}$  model to our results. We minimize the power of the noise in the purchase probabilities, i.e.,

$$(\beta, \gamma, \tau)_{adj} = \arg \left( \min_{\beta, \gamma, \tau} \left( \sum_{h,j,t} (p_{d2,h,j,t}(\beta, \gamma, \tau) - p_{rr,h,j,t})^2 \right) \right) \quad (6.1.22)$$

Here  $p_{d2,h,j,t}(\beta, \gamma, \tau)$  are the purchase probabilities for household  $h$ , product  $i$  and purchase number  $t$ , using the  $d\_d2$  model, and  $p_{rr,h,j,t}$  is the average purchase probabilities from the  $d\_rr$  runs. These average purchase probabilities are essentially the relative purchasing frequencies.

We decided to keep  $\gamma_{3,adj}$  fixed at the  $\gamma_{3,ML}$  value, because it enters the  $d\_d2$  simulation method in a funny way.

The ML and the new parameters for our data are shown in figure 6.1.8,

	ML	adjusted	ratio
beta_price	-5.4618	-3.4812	0.64
beta_display	1.0188	0.6894	0.68
beta_feature	0.1439	0.1259	0.88
gamma1	0.8790	0.9034	1.03
gamma2	0.9781	0.9807	1.00
gamma3	0.9850	0.9850	1.00
tau_1	0.1710	0.0491	0.29
tau_2	0.1068	0.0324	0.30
tau_3	0.1165	0.0350	0.30
tau_4	0.0839	0.0256	0.31
tau_5	0.1135	0.0339	0.30
tau_6	0.1233	0.0361	0.29
tau_7	0.0822	0.0259	0.32
tau_8	0.0738	0.0219	0.30
tau_9	0.1244	0.0366	0.29
tau_10	0.0936	0.0276	0.29
opt_criterion	1.4579	0.3220	0.22

**Figure 6.1.8: The Maximum Likelihood (ML) model parameters and the adjusted ( $d\_d2$ ) model parameters. The `opt_criterion` is the value of the penalty function (6.1.22)**

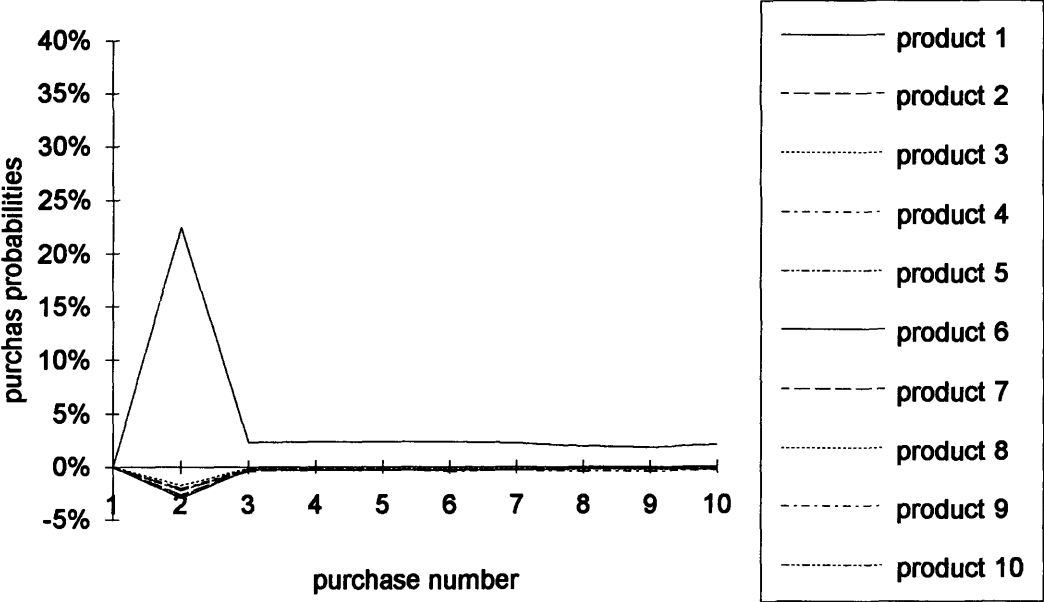
From the figure we observe that the marketing variables (the  $\beta$ 's) have approximately been reduced to 65% of their previous value. The  $\gamma$ 's stay more or less unchanged. The  $\tau$ 's are scaled down by a factor of approximately 0.3.

---

All these changes serve the purpose of adjusting the dynamics of the  $d\_d2$  simulation method so as to mimic the  $d\_rr$  results as closely as can be.

Another approach for approximating the  $d\_rr$  results would be to build new, and not necessarily related models solely for the purpose of reproducing the  $d\_rr$  results.

In figure 6.1.9 we have displayed the estimated effect of the promotion of section 6.1.6 by using these revised parameters.



**Figure 6.1.9: Response to the promotion using the  $d\_d2$  method and adjusted parameters**

From the figure we observe that after these parameter changes, the  $d\_d2$  simulation method (here after referred to as  $d\_d2\_adj$  when using the adjusted parameter values) is a good approximation of the  $d\_rr$  method.

By doing the parameter updates the RMS value of the differences in the response probabilities, i.e.

$$\sqrt{\frac{1}{100} \sum_{i=1}^{10} \sum_{j=1}^{10} \left( (RSP_{d\_d2\_adj,j} - RSP_{rr,j})^2 \right)} \quad (6.1.23)$$

changes from 2.387% to 0.777%. This is a very significant change and was a very nice surprise when we first saw how good it was.

Looking at the aggregated results of section 6.1.3 again to see how the adjustments change the errors, we get the results presented in figure 6.1.10.

	d_dd	d_rr		actual	d_d2	d_d2_adj
product	mean	mean	(std)	total	mean	mean
1	269.0	242.9	(30.4)	259	253.5	239.5
2	182.6	192.0	(28.0)	158	192.1	192.7
3	167.1	170.9	(25.6)	179	170.5	172.0
4	158.7	165.4	(27.0)	157	167.1	166.2
5	123.3	132.3	(23.9)	154	119.9	133.2
6	166.4	164.3	(26.0)	159	158.2	163.0
7	186.5	188.6	(28.6)	160	198.7	190.2
8	93.3	93.3	(20.2)	152	87.8	92.3
9	176.4	175.1	(26.1)	131	176.3	175.2
10	110.6	109.2	(20.2)	125	109.9	109.7
# of runs	1	10000			1	1
running time	14.0	146			36.6	36.6
RMS Error (act)	28.76	29.17			32.30	29.73
RMS Error (rr)	9.59	0.00		29.17	6.63	1.42

Figure 6.1.10: Aggregated results (as figure 6.1.3 ) with adjusted parameters as well

Again we are stunned by how closely d\_d2\_adj approximates the sales of the d\_rr method.

### 6.1.7 More On Response Estimation

When we are evaluating the effect of a promotion, we look at the increase in market share. We must also remember that from running the promotion we not only get a bigger market share when the promotion is running, but also after the promotion closes. In this section we develop a measure that indicates how much a investment in the loyalty vector of a household is worth.

---

Suppose we are evaluating a promotion. We run the model for the market when there is no promotion and get  $\bar{\alpha}_{no}$  as a finishing loyalty vector for a certain household. When running the model with the promotion we get  $\bar{\alpha}_{yes}$ . We now want to get a measure the value in having  $\bar{\alpha}_{yes}$  instead of  $\bar{\alpha}_{no}$ .

Let's look at what the difference in sales would be if starting from these  $\alpha$ 's we would have reference merchandising and an infinite sequence of purchase occasions. Under reference merchandising,  $e^{\beta x} = 1$ , we have,  $E[\phi] = E[y]$ , and thus the updates for  $\bar{\alpha}$  is,

$$E[\bar{\alpha}_{n+1}] = \frac{s_0}{1-\gamma_3} \left\{ \gamma_2 E[\bar{\phi}_n] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} \quad (6.1.24)$$

Normalizing we get,

$$E[\bar{\phi}_{n+1}] = \left\{ \gamma_2 E[\bar{\phi}_n] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} \quad (6.1.25)$$

The difference in purchases probabilities is,

$$\begin{aligned} E[\bar{y}_{n+1,yes} - \bar{y}_{n+1,no}] &= E[\bar{\phi}_{n+1,yes}] - E[\bar{\phi}_{n+1,no}] \\ &= \left\{ \gamma_2 E[\bar{\phi}_{n,yes}] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} - \left\{ \gamma_2 E[\bar{\phi}_{n,no}] + (1-\gamma_2) \frac{\bar{\tau}}{s_0} \right\} \\ &= \gamma_2 \left\{ E[\bar{\phi}_{n,yes}] - E[\bar{\phi}_{n,no}] \right\} \\ &= \gamma_2 \left\{ E[\bar{y}_{n,yes}] - E[\bar{y}_{n,no}] \right\} \end{aligned} \quad (6.1.26)$$

We thus end with a geometric sum and total difference in expected sales is,

$$\sum_{n=n_1}^{\infty} E[\bar{y}_{n,yes} - \bar{y}_{n+1,no}] = \frac{E[\bar{\phi}_{n_1,yes}] - E[\bar{\phi}_{n_1,no}]}{1-\gamma_2} \quad (6.1.27)$$

Henceforth we shall refer to this measure as EISDL (estimated incremental sales from difference in loyalty).

---

If we take (6.1.27) one step back, we can evaluate the value of making a person buy product A as opposed to the person buying product B.

$$\sum_{n=n_1}^{\infty} \{E[y_{n,yes}] - E[y_{n,no}]\} = \frac{E[\phi_{n_1,yes}] - E[\phi_{n_1,no}]}{1 - \gamma_2} = \frac{1 - \gamma_1}{1 - \gamma_2} \Delta y \quad (6.1.28)$$

Using our parameters the multiplicative factor,

$$\frac{1 - \gamma_1}{1 - \gamma_2} \approx 5.0 \quad (6.1.29)$$

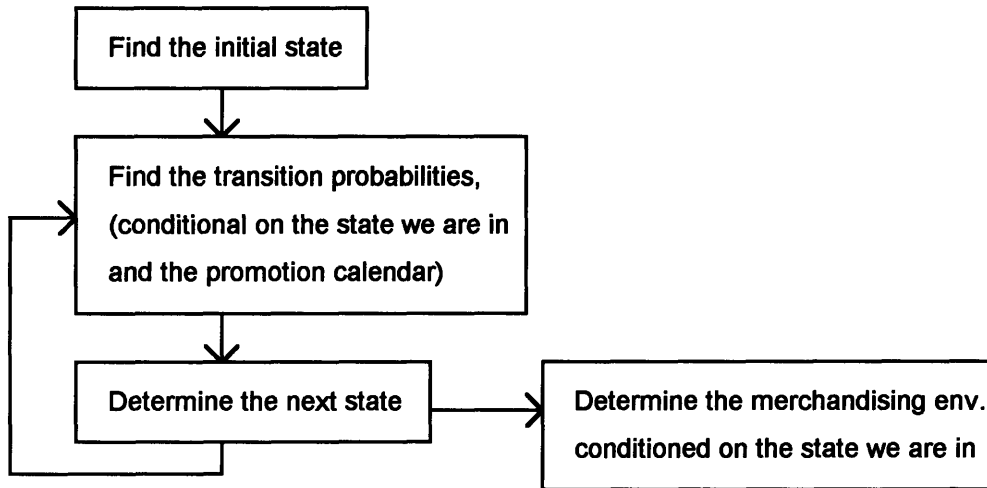
We see that in addition to the actual sale, we increase the expected value of future sales about 5.5.

## 6.2 Simulation At The Intermediate Level

In this section we look at the methods we used to simulate the intermediate Markov chain model of section 3.1.

### 6.2.1 The Simulation Procedure

In figure 6.2.1 we have a diagram explaining how we went about doing the simulations.



**Figure 6.2.1: The simulation cycle for the intermediate Markov model**

The estimated transition matrices are shown in Appendix D.

## 6.2.2 Making The Markov Model Deterministic

When making the Markov model deterministic we have a few approaches. We can use the expected values of the derived parameters (the merchandising here), and then use those as an approximation. There are two versions of this method, depending on where we take the expected value,  $E[e^{\beta x}]$  or  $e^{E[\beta x]}$ , i.e.,

$$\begin{aligned} \beta x &= \sum_{i=1}^5 \Pr(i) \beta x | i \\ e^{\beta x} &= \sum_{i=1}^5 \Pr(i) e^{\beta x | i} \end{aligned} \tag{6.2.1}$$

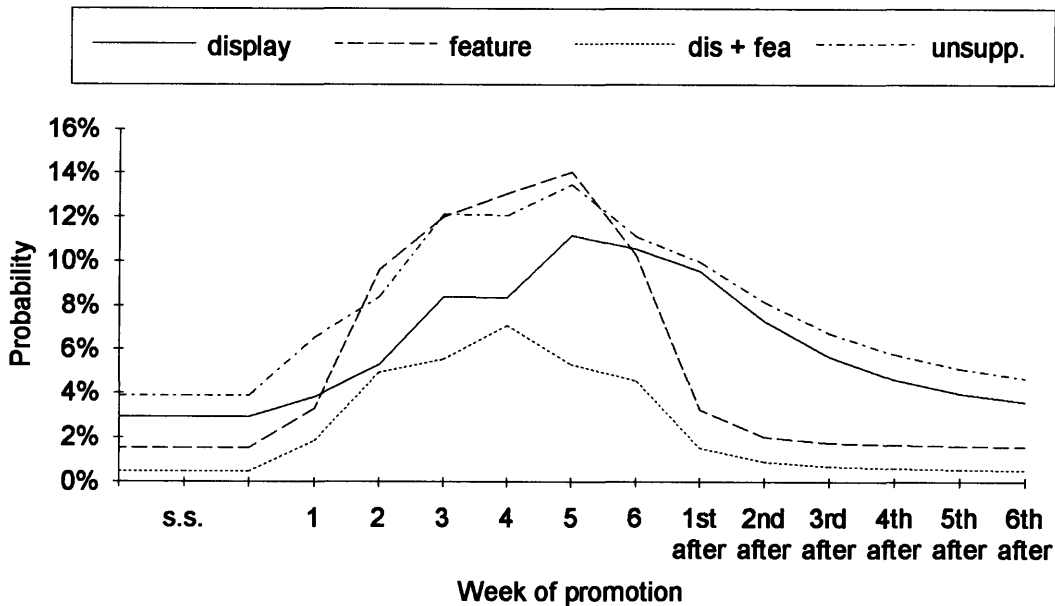
The approach we used, proved to be much better than those briefed above. Since we have only 5 states it is feasible to make one calculation for each state. We calculate the purchase probabilities and the new  $\bar{\alpha}$  for all five states. Then take the expected value of  $\bar{\alpha}$  and  $\bar{y}$  and use those for the next iteration, i.e.

$$\bar{\alpha}_{n+1} = \sum_{i=1}^5 \Pr(i) \bar{\alpha}_{n+1} | \beta x(i) \quad (6.2.2)$$

$$\bar{y}_n = \sum_{i=1}^5 \Pr(i) \bar{y}_n | \beta x(i)$$

### 6.2.3 Model Behavior

In figure 6.2.2 we have displayed the probability of being in each of the 5 states as estimated by this Markov model.



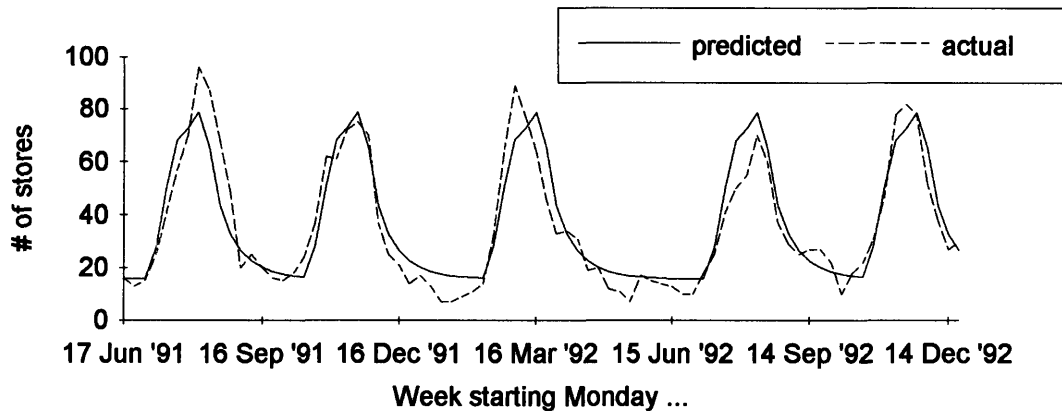
**Figure 6.2.2: Probability of being in a state**

From the figure we observe the following

- The most active time period is the 5th week of the promotion
- Display seems to lag the other activities
- Feature is primarily within the promotion period
- The unsupported price-cuts are lagging the promotion period

---

Let us now look at how well this model fits the number of stores that are running a promotion. In figure 6.2.3 we have plotted the actual number of stores running a promotion and also the expected number of stores using the model. There are in total 179 stores.



**Figure 6.2.3: Number of promoting stores**

From the figure we see that the model tracks what is actually happening surprisingly well. This suggests that at the aggregate level the Markov chain model is quite good for our data.

### 6.3 Integrated Model

We now integrate the Markov model and the Little-Anderson model. The Markov model first generates the marketing conditions probabilistically. Next the households make their purchases on this randomly generated market ( $d_{rr}$ ). The results using this procedure are shown in figure 6.3.1 under  $d_{dss}$ .

As earlier we want to get an approximation for the results of  $d_{dss}$ . This is done by integrating the method introduced in 6.2.2 for the Markov model and the adjusted parameter method of section 6.1.6. The results are shown in the figure under  $d_{dss\_d2\_adj}$ .

product	d_rr	d_d2_adj	dss_d2	dss_d2_adj	dss_r
1	242.9 (30.4)	239.5	250.6	242.7	247.7 (36.7)
2	192.0 (28.0)	192.7	187.9	190.3	189.6 (28.1)
3	170.9 (25.6)	172.0	166.2	169.6	168.1 (25.0)
4	165.4 (27.0)	166.2	163.7	164.2	163.3 (27.0)
5	132.3 (23.9)	133.2	118.4	132.2	131.2 (23.6)
6	164.3 (26.0)	163.0	177.7	170.4	171.6 (26.4)
7	188.6 (28.6)	190.2	194.4	187.8	185.2 (28.7)
8	93.3 (20.2)	92.3	93.6	94.4	95.6 (19.7)
9	175.1 (26.1)	175.2	173.3	173.7	174.3 (25.5)
10	109.2 (20.2)	109.7	108.0	108.6	107.3 (19.5)
# of runs	10000	1	1	1	4008
running time	146	36.6	36.7	36.7	156.0
RMS Error (act)	29.17	29.73	30.76	28.70	28.22
RMS Error (rr)	0.00	1.42	7.13	2.18	3.42
RMS Error (dss)	3.42	4.71	5.52	2.04	0.00

**Figure 6.3.1: Aggregated results (as figure 6.1.3) with Markov model as well**

From the figure we see that after adding the Markov chain to the model we are still close to the d\_rr runs. This is somewhat surprising because when inspecting the data it was difficult to determine when the manufacturer for product 1 (which is the product that the Markov chain was used) was running his promotions. We thus made an educated guess and it seems to be very good.

Also we observe that the d\_dss\_d2\_adj is very close to the d\_dss results (2.04 = 0.125 share points), which verifies that the deterministic approximation is very good for this data.

---

## 7. The Decision Support System In Use

In this chapter we look at an example of how the user can use the DSS to aid in the making of a decision. We do this by hypothesizing an offer and then use the DSS to calculate relevant measures that make the trade-off between accepting and rejecting the offer more explicit.

### 7.1 The Offer

**A retailer has proposed that for \$20,000 in promotion funds, he would run a promotion for product 1. He would have a feature in the weekly chain flyer (effective for two weeks, from 2 July to 15 July) and a three week "front end-aisle" display (2 July to 22 July) followed by a two week (23 July to 5 August) unsupported price cut. The price throughout this 5 week period (2 July to 5 August) would be 15% below the average price. Would this be profitable for the manufacturer?**

This retailer is in Eau Claire, WI. Last year, a sample of 237 households made 1634 purchases (from 14 May '90 to 12 May '91) from a category of the leading 10 juices. The retailer has 4 of the 9 stores included in the database.

To decide whether to accept the offer, we must look at what the \$20,000 buys. We thus evaluate the cases we have and from the results, make a decision.

#### 7.1.1 Offer Accepted

Let's forecast what will happen if the manufacturer accepts the offer. For all the stores involved (stores 337, 339, 340 and 341) we need to put the scenario into the scenario evaluator. The input would look similar to that shown in figure 7.1.1.

											d = display f = feature M = Markov chain
		Week: 2 Jul		Week: 9 Jul		Week: 16 Jul		Week: 23 Jul		Week: 30 Jul	
	Prob. of scenario	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)	Disp. Feat.	Price (\$)
Scenario 1	1.00	df	1.89	df	1.89	d	1.89		1.89		1.89
Scenario 2											
Scenario 3											
Otherwise	0.00										

**Figure 7.1.1: The input for the scenario evaluator if offer is accepted**

From prior experience we know that it can happen that the retailer just takes the money and does nothing. This can be incorporated into the scenario evaluator, but here we ignore this possibility.

In figure 7.1.2 we have summarized last year's data on consumer behavior.

Time Period		chain		non-chain		total	
from	to	purch.	hh	purch.	hh	purch.	hh
14 May '90	1 Jul '90	222	81	95	48	317	117
2 Jul '90	5 Aug '90	127	66	71	37	198	94
6 Aug '90	12 May '91	719	172	400	101	1119	215
14 May '90	12 May '91	1068	194	566	119	1634	237

**Figure 7.1.2: Last year's consumer behavior: "Chain" refers to the stores of the retailer making the offer and "non-chain" refers to all other stores. In the "purch." column we have the number of purchases by week and chain. The "hh" column has the number of households that made those purchases**

The final step is then to run the household level model.

### 7.1.2 Offer Rejected

We now estimate what happens if we reject the offer. In this case we don't need to use the scenario evaluator. The Markov model generates the merchandising environment as if

---

nothing had happened. We then run the household level model and using the results we are ready to compare the output to the output of section 7.1.1.

### 7.1.3 Comparing The Output

In figure 7.1.3 we have the forecasted sales figures for this population for both cases. In the figure "yes" refers to accepting the offer, and "no" refers to rejecting it.

product	yes	no	yes - no	EISDL
1	281.87	242.69	39.18	190.74
2	184.19	190.31	-6.13	-24.47
3	165.53	169.60	-4.07	-22.06
4	159.36	164.18	-4.81	-19.48
5	128.49	132.21	-3.72	-20.39
6	166.53	170.44	-3.90	-23.79
7	181.65	187.76	-6.11	-23.57
8	92.07	94.43	-2.36	-14.56
9	168.73	173.73	-5.01	-25.51
10	105.57	108.64	-3.08	-16.91

**Figure 7.1.3: Aggregate sales and EISDL (Estimated Incremental Sales caused by Difference in Loyalty, see section 6.1.7) for "yes"-accepting the offer and "no"-rejecting the offer**

For our population we gain 39.18 items (2.40% share). Hypothesizing that the households of the database represent the population of the Eau Claire market, we estimate that accepting the offer increases the share in Eau Claire about 2.40%. Further more we observe that for the households of the database we gain 190.74 potential sales estimated by EISDL (Estimated Incremental Sales caused by Difference in Loyalty, see section 6.1.7).

From figure 7.1.3 we also observe that we have cannibalization from other Ocean Spray products. Products number 1, 5, 6, 8, 9, 10 (see beginning of chapter 6 for product names) are manufactured by Ocean Spray Cranberries. In figure 7.1.4 we have summed up the results of figure 7.1.3 for each manufacturer.

---

Manufacturer	yes	no	yes - no	EISDL
Ocean Spray	943.26	922.15	21.12	89.58
Gatorade	525.20	542.25	-17.05	-67.52
Hawaiian Punch	165.53	169.60	-4.07	-22.06

**Figure 7.1.4: Aggregated sales and EISDL by manufacturer**

We observe that Ocean Spray's share (in # of items) increases from 56.44% to 57.73% (1.29%) and the potential sales of the households of the database increase about 89.58 items.

### 7.1.4 What To Do

We now have a much better idea of the potential of the offer. From the analysts prior experience he decides if it is worth the \$20,000 to get the share increase and potential sales forecasted by the DSS.

### 7.1.5 Discussion

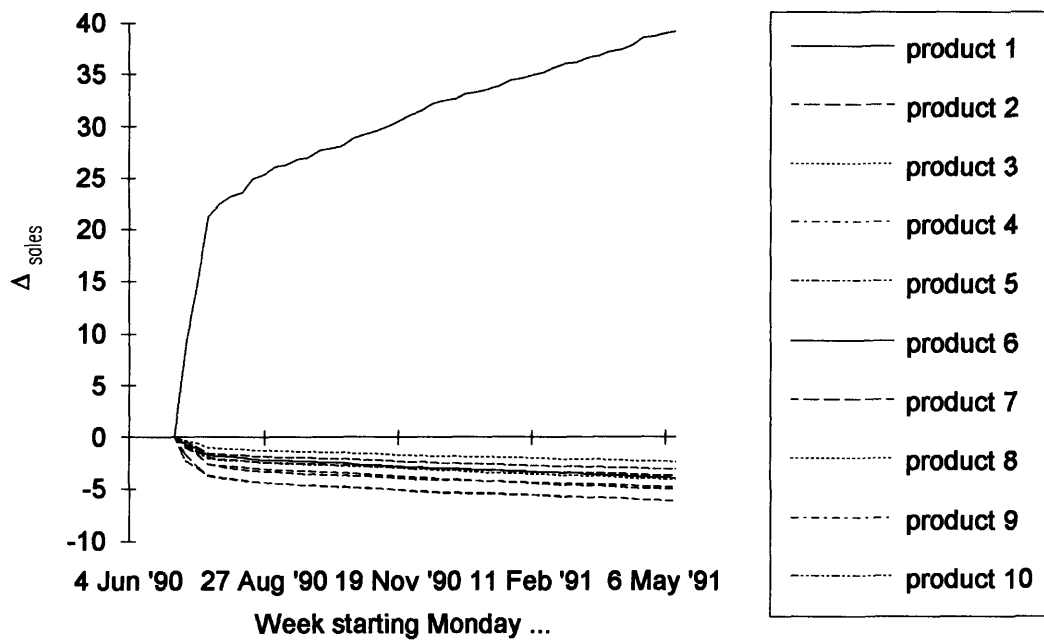
The offer that we are evaluating above affects the customers early in our time horizon. If the offer would be say for April '92, we must observe that the promotion only affects sales after it opens. We would thus see a much smaller rise in the market share, but the EISDL would be higher.

After gaining some experience with the system the user should determine a factor,  $\lambda \approx 1$ , and then use the incremental sales plus  $\lambda$  times EISDL for evaluating offers, i.e.,

$$\bar{\Omega} = \bar{\Delta}_{sales} + \lambda \bar{V} \quad (7.1.1)$$

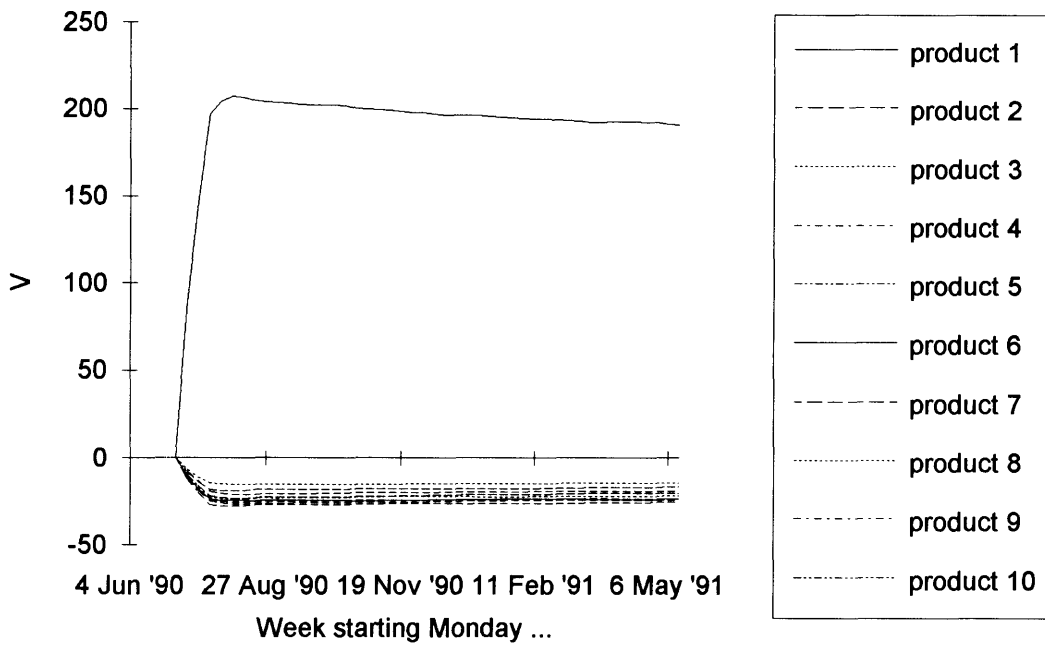
Then  $\bar{\Omega}$  is a measure of the potential of this offer in this market,  $\bar{\Delta}_{sales}$  is the incremental sales of the offer ("yes - no" in figures 7.1.3 and 7.1.4), and  $\bar{V}$  is the EISDL.

Let's look closer at how  $\bar{\Omega}$  behaves. In figure 7.1.5 we have the incremental sales (in number of items) of the promotion if the time window is from 21 May '90. As expected this is a monotonically increasing function for product 1.



**Figure 7.1.5: Incremental sales of promotion using time horizon from 14 May '90 to the date shown on the x-axis**

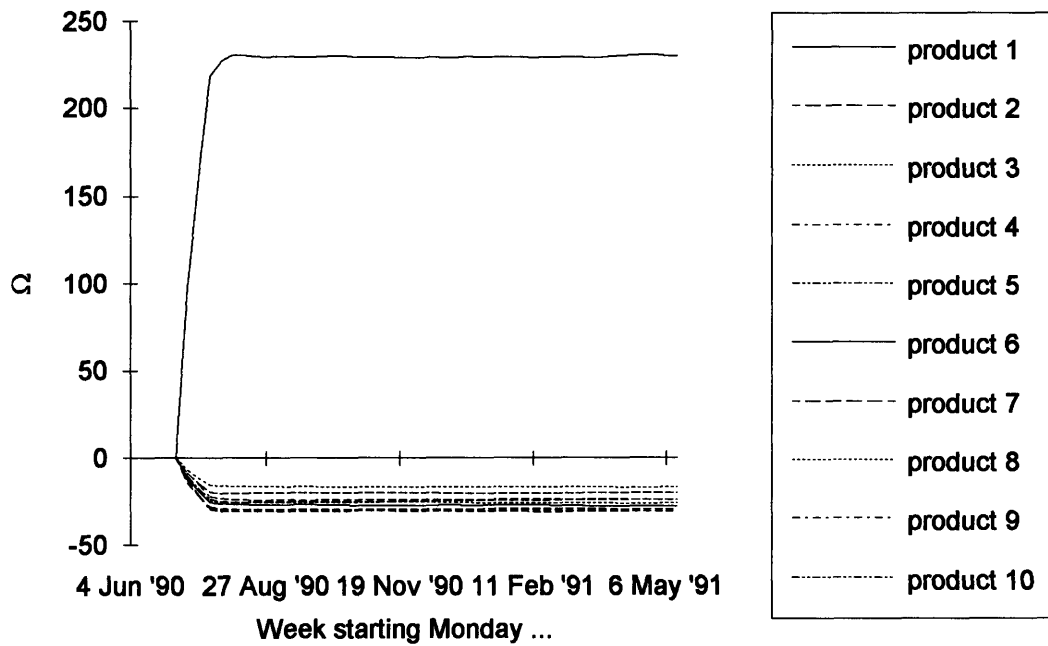
In figure 7.1.6 We have similarly displayed the potential sales (EISDL) for all the products as a function of the end of the time horizon.



**Figure 7.1.6: EISDL using time horizon from 14 May '90 to the date shown on the x-axis**

We see that after the promotion closes the EISDL for product 1 slowly decreases. This is to be expected because as we get longer from the promotion the less effect it has.

Summing up for  $\bar{\Omega}$  (using  $\lambda = 1$ ) we get the graph shown in figure 7.1.7.



**Figure 7.1.7: EISDL plus incremental sales using time horizon from 14 May '90 to the date shown on the x-axis**

We see from the figure that after the promotion closes (5 August) the potential is very close to being a constant. We thus feel good about using  $\bar{\Omega}$  as an indicator of the offer potential.

---

## 8. Conclusions

We have reviewed a few models that model customer behavior. The primary goal of these models has been to help the model developers to understand people's choice behavior. We have here taken one step further and used the models to simulate people and therefore markets. By building a "human simulator" we can hypothesize scenarios and forecast what would happen in different markets under different circumstances. This can be a very useful tool if used correctly.

We developed and tested models for generating the merchandising environment using higher level inputs (the manufacturer promotions). This tool can be used by manufacturers to determine the optimal timing for promotions.

Combining the two types of models above, we have a complete system. We used this system to evaluate an example promotion.

We developed shortcuts to make the models computationally practical for marketing planners. Doing the full Monte Carlo simulation is the most desirable way of evaluating offers, but it is at present unfeasible, because of the computational burden. Many hours are required for a single evaluation. We developed deterministic methods that approximate the Monte Carlo. By doing this we decrease the computation time enormously. We need only to make a single run through the households to get their expected purchases of each product. Also, each run of the deterministic model is computationally cheaper than a single Monte Carlo run (thousands of individual runs are required for a Monte Carlo evaluation). The price we have to pay for using this approach is that it only approximates the desired results. However, in the examples we have tried the accuracy is very good.

We introduced methods for changing the model parameters for the deterministic approximation of the Monte Carlo simulations. Depending on the goal of the DSS we can

---

use the most suitable criterion for selecting the parameters for the deterministic approximation.

We defined a new variable,  $\bar{\Omega}$ , that estimates the potential of an offer. Comparing two offers that don't intersect in time is much easier using this potential variable than if we would not have it.

We did not build a user interface. A computer package, including the input interface, the Decision Support System and an output interface should be built. This package would be of great use to a lot of people, and make their every-day decisions easier.

---

## Bibliography

Arnold, Steven F., 1990, *Mathematical Statistics*, Prentice Hall, New Jersey, pp. 538-541

Law, Averill M., W. David Kelton, 1991, *Simulation Modeling & Analysis*, 2nd edition, McGraw-Hill, Inc., New York, pp. 487-490

Guadagni, Peter M. and John D. C. Little, Summer 1983, *A logit Model of Brand Choice Calibrated on Scanner Data*, Marketing Science, Vol. 2, No. 3, pp 203-238

Guadagni, Peter M. and John D. C. Little, August 1987, *When and What to Buy: A Nested Logit Model of Coffee Purchase*, WP 1919-87, MIT, Cambridge, MA 02139

*Matlab® Reference Guide*, January 1993, The Math Works Inc., Cochituate Place, 24 Prime Park Way, Natick, MA 01760

Fader, Peter S., James M. Lattin, John D. C. Little, Fall 1992, *Estimating Nonlinear Parameters in the Multinomial Logit Model*, Marketing Science, Vol. 11, No. 4, pp. 372-385

Little, John D.C. and Eric Anderson, 8 June 1993, *A Product Choice Model with Marketing, Filtering and Purchase Feedback*, MIT Sloan School of Management, Cambridge, MA 02139

Lian, Elizabeth, 1993, : *UROP Project with John D. C. Little*, MIT, Cambridge, MA 02139

Ben-Akiva, Moshe and Steven R. Lerman, 1985, *Discrete Choice Analysis*, MIT Press, Cambridge, MA 02139

---

## Appendix A: The Method For Generating A Dirichlet Distributed Vector

In this appendix we explain how we go about generating the Dirichlet vectors.

### A.1 The Dirichlet Density Function

The Dirichlet density function is,

$$f(\vec{\phi}|\vec{\alpha}) = \frac{\Gamma(\alpha_1 + \dots + \alpha_{n-1} + \alpha_n)}{\Gamma(\alpha_1) \dots \Gamma(\alpha_{n-1}) \Gamma(\alpha_n)} \phi_1^{\alpha_1-1} \dots \phi_{n-1}^{\alpha_{n-1}-1} (1 - \phi_1 - \dots - \phi_{n-1})^{\alpha_n-1} \quad (\text{A.1.1})$$

To generate a vector with this density function we follow the following procedure:

- Generate  $n$  gamma( $\alpha_j, 1$ ) distributed scalars, one for each element of alpha, call the resulting draws  $g_i$
- Calculate  $d_i = \frac{g_i}{\sum_{j=1}^n g_j} \quad \forall i$
- Then  $\vec{d} = (d_1, d_2, \dots, d_{n-1})$  is Dirichlet distributed (A.1.1) with parameters  $\vec{\alpha}$ .

For a proof that  $\vec{d}$  is Dirichlet distributed consult Arnold (1990).

We have thus reduced the problem to generating a gamma( $\alpha, 1$ ) distributed scalar.

### A.2 The Gamma Density Function

In this section we describe the procedure for generating a gamma distributed scalar.

The gamma( $\alpha, \beta$ ) density function is,

$$f(x) = \begin{cases} \frac{\beta^{-\alpha} x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha)} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.2.1})$$

---

Observe that  $\beta$  is merely a scaling factor for the density so we set it equal to one.

The cumulative density function does not exist in closed form if  $\alpha$  is not an integer. Thus, we need to use an alternative method. The method used here is known as the *acceptance-rejection method*.

### A.3: The Acceptance - Rejection Method

The intuition for this method can be explained as follows: Plot the density function on a piece of paper and randomly choose any point on the paper (uniformly distributed). If this point is under the density function, use the corresponding  $x$ -value. If it is outside the density function, reject the point and choose another one. This is shown in figure A.3.1.

The accepted points are solid points and the rejected ones are open circles.

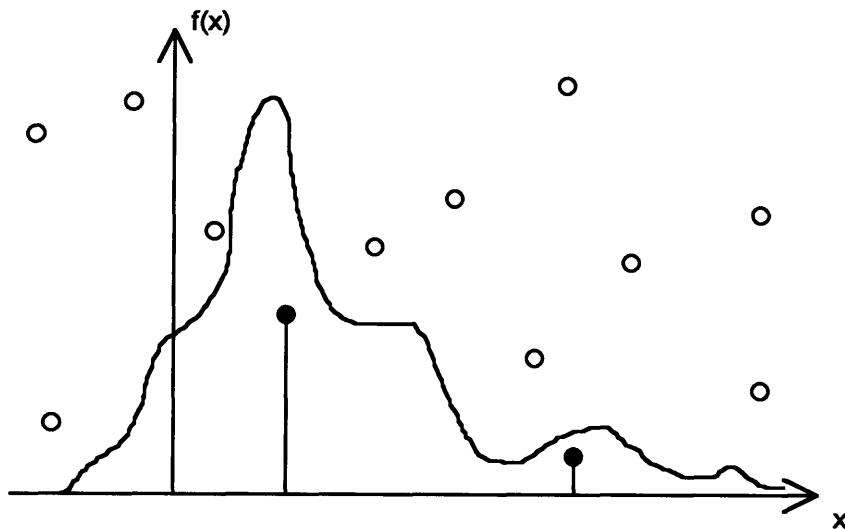


Figure A.3.1: The density function and accepted/rejected points

The efficiency of the method can often be greatly increased by using a majoring function,  $m(t)$ .

$$m(t) \geq f(t) \quad \forall t \in \mathbb{R} \quad (\text{A.3.1})$$

---

The two constraints for the majoring function are that it satisfies equation A.3.1 and that it has a closed form integral.

With such a majoring function, the first step is to draw a random number  $z_1$ ,

$$z_1 \sim U\left(0; \int_{-\infty}^{\infty} m(t) dt\right) \quad (\text{A.3.2})$$

We find the corresponding  $x$  by taking the inverse, i.e.,

$$z_1 = \int_{-\infty}^x m(t) dt \quad (\text{A.3.3})$$

The second step is to determine if this random point is below the density function. We do this by choosing another random number,

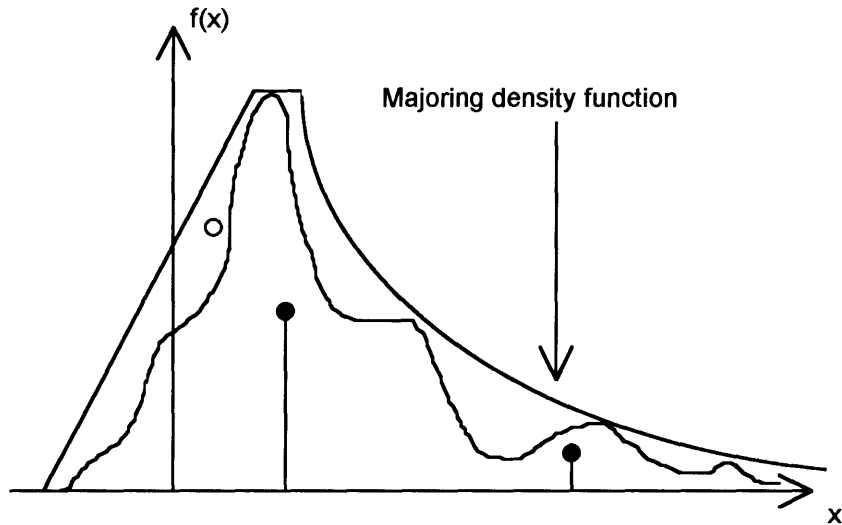
$$z_2 \sim U(0; m(x)) \quad (\text{A.3.4})$$

If,

$$z_2 \leq f(x) \quad (\text{A.3.5})$$

then  $x$  is accepted, otherwise we must try again from step one (observe that we must choose another  $z_1$ ).

Figure A.3.2 illustrates the increased efficiency when using a good majoring function for the density of figure A.3.1



**Figure A.3.2: The density function and accepted/rejected points using a majoring function**

For the purpose of different majoring function, generating a  $\text{gamma}(\alpha, 1)$  scalar is split into three cases,  $\alpha < 1$ ,  $\alpha = 1$ ,  $\alpha > 1$ . We shall not go further into the details here, but a complete discussion is in Arnold (1990), and the algorithms we used for generation are from Law and Kelton (1991).

## **A.4: The MATLAB® Code For Generating The Dirichlet Vectors**

Below we have the code for generating the Dirichlet vectors. The function `f_gamma.m` generates a gamma scalar, and `f_dirich.m` generates the Dirichlet vector.

### **f\_dirich.m**

```
function w = f_dirich(alfa) ;
% f_dirich      f_dirich(alfa) returns a random vector, using the
%              dirichlet distribution and parameters alfa.

% Written: January 25, 1993 by Arni G. Hauksson
% This routine generates n-gamma distributed numbers and then
% normalizes the corresponding vector to get a Dirichlet distributed
% vector.

lengd = length(alfa) ;
```

---

```
for cou = 1:lengd,
    x(cou) = f_gamma(alfa(cou)) ;
end ;
```

```
w = x/sum(x) ;
```

## **f\_gamma.m**

```
function w = f_gamma(alfa) ;
% f_gamma    f_gamma(alfa) returns a random number from the gamma
%            distribution with parameter alfa.

% Written:   January 25, 1993 by Arni G. Hauksson
% This routine uses the algorithms proposed in: Law, Averill M. and
% W. David Kelton, Simulation Modeling & Analysis Second Edition,
% pages 487 - 490.
% There are three cases, for alfa < 1, alfa = 1 and alfa > 1
% The method is to find a majorizing function for the density and then
% apply a acceptance-rejection technique.

%           Case 3:  alfa > 1
if alfa > 1,
    a = inv(sqrt(2 * alfa - 1)) ;
    b = alfa - log(4) ;
    q = alfa + 1 / a ;
    p = 4.5 ;
    d = 1 + log(p) ;
    ok = 0 ;
    while ok == 0,
        u = rand(1,2) ;
        v = a * log(u(1) / (1-u(1))) ;
        y = alfa * exp(v) ;
        z = u(1)^2 * u(2) ;
        f = b + q * v - y ;
        if f + d - p * z >= 0,
            w = y ;
            ok = 1 ;
        elseif f >= log(z),
            w = y ;
            ok = 1 ;
        end ;
    end ;
end ;

%           Case 2:  alfa = 1  (--> exponencial distrib.)
elseif alfa == 1,
    u = rand ;
    w = -log(u) ;

%           Case 1:  alfa < 1
else
    b = 1 + alfa * exp(-1) ;
    ok = 0 ;
```

---

```
while ok == 0,
    u = rand(1,2) ;
    p = b * u(1) ;
    if p <= 1,
        y = p^(1 / alfa) ;
        if u(2) <= exp(-y),
            w = y ;
            ok = 1 ;
        end ;
    else
        y = -log((b-p)/alfa) ;
        if u(2) <= y^(alfa - 1),
            w = y ;
            ok = 1 ;
        end ;
    end ;
end ;
end ;
```

---

## Appendix B: The Problem Of Over-Switching When Using The *Same Seed* Method

In this appendix we take a look at a problem encountered when using the *same seed* method to explore differences in discrete event simulation.

The idea of the same seed method is to reduce the variance of the difference of two results, by stimulating both systems (models or whatever) with the same noise. The method helps to minimize stochastically driven differences between the two systems.

A simple qualitative explanation can be given. When subtracting two random variables,  $x$  and  $y$ , the variance of the result,  $z$ , is,

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2 - 2\rho_{xy} \sigma_x \sigma_y \quad (\text{B.0.1})$$

When doing two independent runs the correlation is zero,  $\rho = 0$ , and we have,

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2 \quad (\text{B.0.2})$$

If we would have a method that gives a strong correlation,  $\rho \cong 1$ , between the two, the variance can be decreased significantly. When we for example need to get a certain level of significance, we get along with much fewer iterations by using such a technique. This is exactly what the same seed method does.

The method is used with great successes in simulating continuous systems. When simulating discrete event systems, special measurements must be taken to preserve the properties of the method.

---

## B.1 The Over-Switching Problem

When using the same seed method for discrete systems, we run into a problem. We experience switching from say event A to B despite keeping the ratio of the probabilities of these events unchanged. The problem is best illustrated by looking at an example.

### B.1.1 A Continuous Example

Let us compare two models for a continuous process. The two models (A and B) are `armax(1)` and `armax(2)`,

$$\begin{aligned} u_{t+1} &= 0.8u_t + e_{t+1} && \text{Model A} \\ u_{t+1} &= 0.9u_t - 0.1u_{t-1} + e_{t+1} && \text{Model B} \end{aligned} \tag{B.1.1}$$

Here  $u_t$  is the output signal (could for example be some control signal) and  $e_t$  is the uniformly distributed (between zero and one) error term. In figure B.1.1 we have two scenarios of interest.

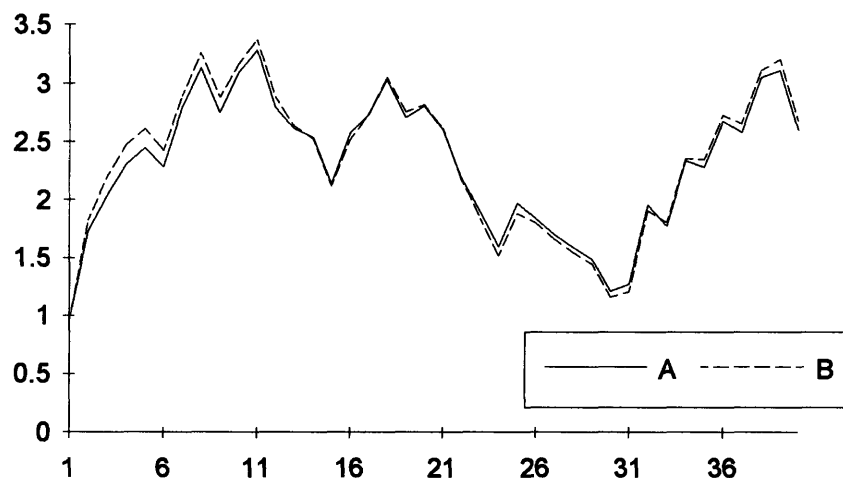
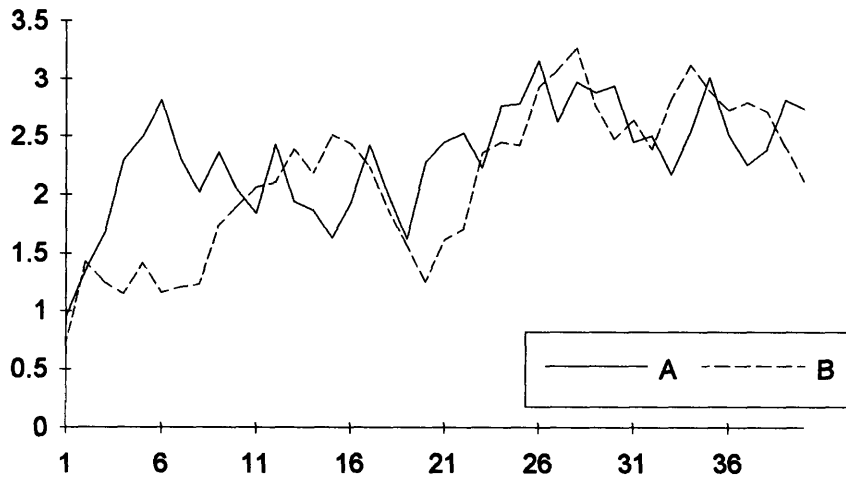
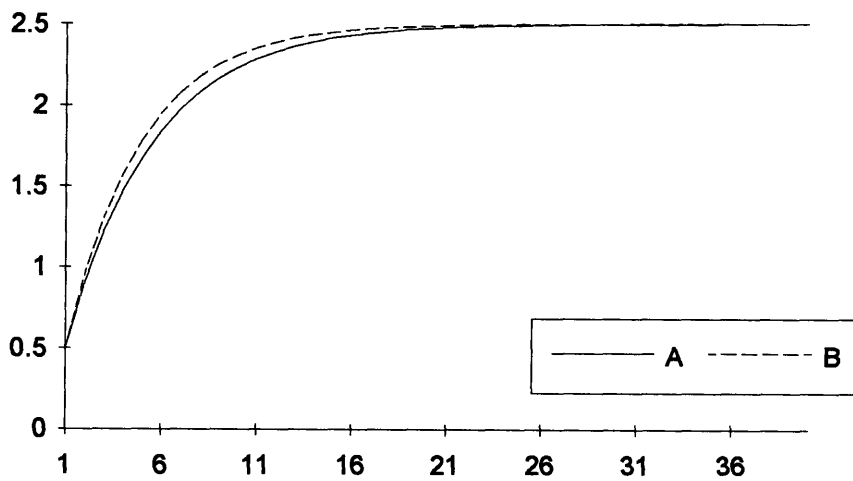


Figure B.1.1a.: Same seed simulations

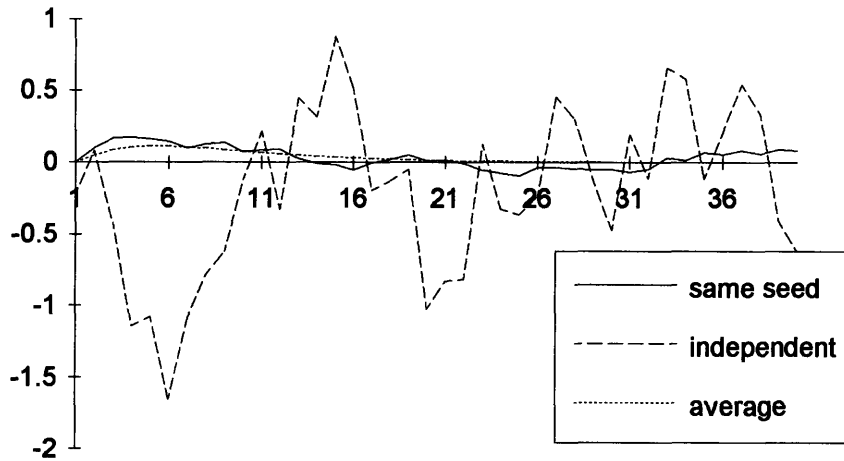


**Figure B.1.1b: Independent simulations**

Figure B.1.1b shows two independent runs of the models, and figure B.1.1a illustrates the output where the same seed is used. We see that figure B.1.1a gives a much better feel for the differences of the two models, whereas the difference in figure B.1.1b is clearly mostly caused by differences in the random input to the system. Next we did many independent runs of the models. Figure B.1.2a shows the average of the tracked value,  $u_t$ . Finally in figure B.1.2b we plotted the difference of the outputs of the models (model B - model A).



**Figure B.1.2a: Average of many runs**



**Figure B.1.2b: Results of model B - model A**

We immediately observe that using the same seed gives a much better picture of the different responses of the models (i.e., model A filtering the high frequency terms more). The reason is that both models are simulated with the same stochastic effects. Thus their responses are very similar and all differences are caused by true differences in the models and not by different inputs.

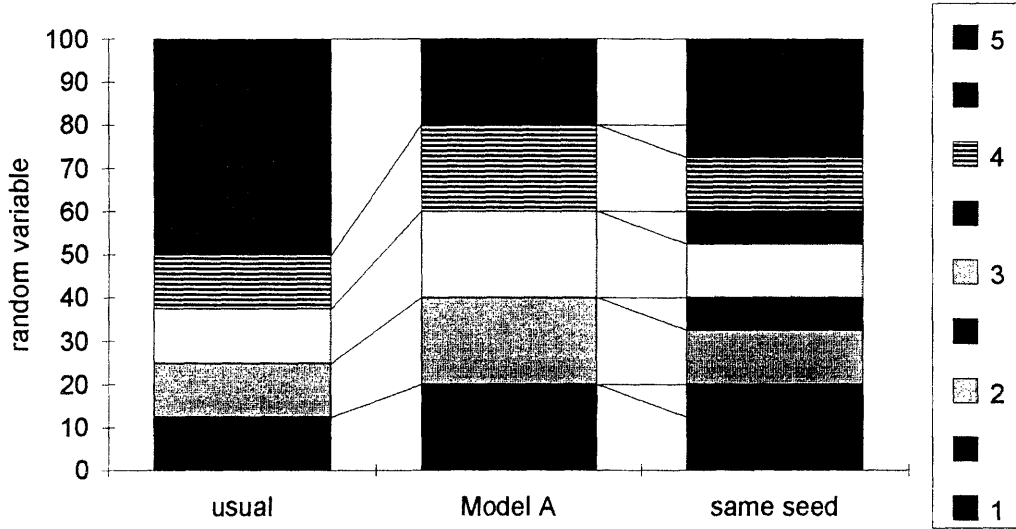
### **B.1.2 A Discrete Example**

Now let's look at a more complex and interesting example. Suppose we are deciding which of several discrete events happened (1, 2, 3, 4 or 5). We have two models,

$$\begin{aligned}
 p(i) = \frac{1}{5} \quad i \in \{1,2,3,4,5\} \quad \text{Model A} \\
 \left. \begin{aligned}
 p(i) = \frac{1}{8} \quad i \in \{1,2,3,4\} \\
 p(i) = \frac{1}{2} \quad i = 5
 \end{aligned} \right\} \text{Model B}
 \end{aligned}
 \tag{B.1.2}$$

The difference between the models is that the probability of event 5 has increased and the probability of the other events has decreased. Observe that for events 1 through 4 we have IIA between the models, i.e.,

$$\left. \frac{p(i)}{p(j)} \right|_{\text{Model A}} = \left. \frac{p(i)}{p(j)} \right|_{\text{Model B}} \quad \forall (i, j) \neq 5 \quad (\text{B.1.3})$$



**Figure B.1.3: Representation of problem**

The obvious way to generate an event is to select a random number ( $r$ ) uniformly distributed between 0 and 1, and  $k$  occurred iff,

$$\sum_{j=1}^{k-1} p(j) < r \leq \sum_{j=1}^k p(j) \quad (\text{B.1.4})$$

Let us generate a series of events using the same seed,

Model A: 34121343553143454214244522544414523432221355153325  
 Model B: 55141455555154555315355532555515534542332455155535

We observe that we have transitions  $3 \rightarrow 5$  and  $4 \rightarrow 5$  as expected (a transition is the change when going from model A to model B). However, we also have  $1 \rightarrow 2$ ,  $2 \rightarrow 3$ ,  $2 \rightarrow 4$  and  $3 \rightarrow 4$  and we see that all 4s change to 5s. We have thus lost the main advantage of using the same seed method. Ideally we would only have transitions  $1 \rightarrow 5$ ,  $2 \rightarrow 5$ ,  $3 \rightarrow 5$  and  $4 \rightarrow$

---

5, and none of the other. This is the problem of **over-switching**. In section B.2 we present a solution to this problem.

### **B.1.3 Motivation From A Marketing Model**

When we are doing a full simulation of the Little-Anderson model, one of the steps is to determine which product was purchased. Determining this is a typical discrete event simulation as introduced in B.1.2.

We want to estimate the effects of reducing the price of one product,  $k$ , by 10%. This should encourage some people who otherwise would buy other products to switch to  $k$ , but it should not make people change from  $i$  to  $j$  (both different from  $k$ ). The method introduced in this appendix, attacks exactly this problem.

When thinking of the Little-Anderson model, this also becomes even more serious than in the simple stationary example of section B.1.2. The reason is that the parameters of the model are updated **according** to the purchase,  $\bar{y}$ . Therefore we must do everything possible to avoid switching from  $i$  to  $j$  (both different from  $k$ ) where we don't want them.

## **B.2 The Hauksson Method For Minimizing Unwanted Switching**

In this section we present a solution to the problem encountered in section B.1.2. We give a simple proof and an algorithm to use.

### **B.2.1 Derivation**

Consider  $n$  Poisson processes, each of rate  $a(i)$ . Let  $X(i)$  be the inter arrival times. The combined process is defined as,

$$X = \min_i \{X(i)\} \quad (\text{B.2.1})$$

The combined process is itself a Poisson process of rate  $a = \sum a(i)$ . Let  $k$  be the argument that gives the minimum. The probability that  $k = j$  is then,

$$\Pr(k = j) = \frac{a(j)}{\sum_i a(i)} \quad \forall j \quad (\text{B.2.2})$$

We thus see, that by letting  $a(i) = p(i) \quad \forall i$  the probability that  $k$  equals  $j$  is just  $p(j)$ , where  $p(i)$  is a pdf.

### B.2.2 Algorithm

We present a step-by-step algorithm. This algorithm is used to generate discrete events.

Step 1	Generate $n$ random numbers, uniformly distributed between 0 and 1.  $r(i) \propto U[0;1] \quad \forall i$
Step 2	Calculate the value of the random variables  $x(i) = \frac{-\log(1-r(i))}{p(i)}$
Step 3	Find the argument the attains the minimum  $k = \arg\left(\min_i \{x(i)\}\right)$
Step 4	Then event $k$ happened.

**Figure B.2.1: The Hauksson algorithm for generating discrete events**

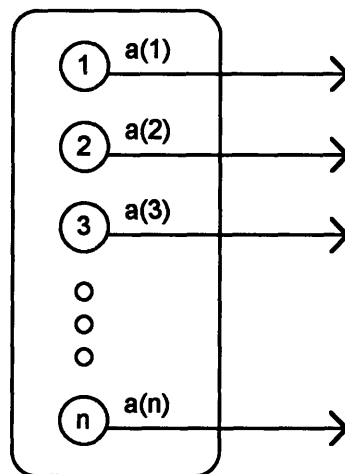
Considering the concavity of the logarithmic function it is clear that steps two and three can be done more efficiently by,

Step 2	<p>Calculate the value of the random variables</p> $x^*(i) = r(i)^{1/p(i)}$
Step 3	<p>Find the argument that attains the maximum</p> $k = \arg\left(\max_i \{x^*(i)\}\right)$

**Figure B.2.2: The extended Hauksson algorithm for generating discrete events**

### B.2.3 Queueing System Explanation

To get a better understanding of how the method works, let us look at a queueing system.



**Figure B.2.3: There are  $n$  competing Poisson processes**

---

Imagine that we have  $n$  customers in  $n$  service stations. The service time for each of them is exponential and independent. Let the service rates at each station be  $a(i)$ . The probability that the customer at station  $k$  is the first out is then,

$$\Pr(\text{customer } k \text{ is the first one out}) = \frac{a(k)}{\sum_i a(i)} \quad (\text{B.2.3})$$

To determine which customer gets out of the system first we could follow the following procedure. First determine the time at which each customer gets out. For doing that we need to generate  $n$  exponentially distributed random variables. Next, simply find who gets first out.

We now observe that letting  $a(i) = p(i) \quad \forall i$ , the probability that customer  $k$  is the first one out is exactly  $p(k)$ .

### B.2.4 A Discrete Example (Continued)

Here we re-simulate the models in section B.1.2, using the extended Hauksson algorithm.

Model A: 11234553115511352232512224551253534133251232133223  
 Model B: 51235555155511355232552254551553555135255255533225

From the results we see that we only have transitions  $1 \rightarrow 5$ ,  $2 \rightarrow 5$ ,  $3 \rightarrow 5$  and  $4 \rightarrow 5$ .

## B.3 Conclusions

We now have a method to generate one of many discrete events, that does not suffer from the problem explained in section B.1.2. The method uses  $nn$  random numbers (instead of only one), transforms these numbers, and finds the minimum.

We have used the method for simulations of marketing models, with great success.

---

## Appendix C: Running Time For A "d\_dd" Simulation

In this section we present a model approximating the time it takes to do a full d\_dd simulation.

The goal is to estimate a model of the time it takes to do a full d\_dd simulation of the Little-Anderson model. Assume the functional form is:

$$T = \beta_0 N^{\beta_1} B^{\beta_2} H^{\beta_3} \quad (\text{C.1})$$

where

- $T$  is the time that the simulation takes in milliseconds
- $N$  is the number of purchases
- $B$  is the number of brands in the category
- $H$  is the number of households
- $\bar{\beta}$  are the parameters to be estimated

The estimated model is,

$$T = 8.46 N^{1.009} B^{0.055} H^{-0.031} \quad (\text{C.2})$$

However, it does not make sense to have  $\beta_3 < 0$  so we set it equal to 0 and then get the model,

$$T = 8.03 N^{0.992} B^{0.056} \quad (\text{C.3})$$

This is very close to what we expected a priori. The parameter corresponding to  $N$  should be very close to 1 and the parameter corresponding to  $B$  should be close to, but greater than zero. Finally, restricting the parameter for  $N$  to be one, we get the result,

$$T = 7.6 N B^{0.0576} \quad (\text{C.4})$$

---

## Appendix D: The Transition Matrices

Below we present the results of estimating the seven transition matrices for the intermediate Markov model. These estimates are based on the store database introduced in chapter 6.

from\to	1	2	3	4	5
1	96.47%	0.91%	1.15%	0.22%	1.25%
2	31.45%	55.20%	0.68%	2.71%	9.95%
3	59.31%	7.36%	16.88%	3.46%	12.99%
4	31.46%	42.70%	4.49%	11.24%	10.11%
5	31.99%	4.24%	4.66%	2.12%	56.99%

**Figure D.1: Non-promotional transition matrix**

from\to	1	2	3	4	5
1	90.14%	1.97%	2.63%	1.31%	3.94%
2	42.11%	47.37%	0.00%	0.00%	10.53%
3	27.27%	0.00%	45.45%	9.09%	18.18%
4	60.00%	20.00%	0.00%	20.00%	0.00%
5	10.34%	13.79%	5.17%	10.34%	60.34%

**Figure D.2: Transition matrix for first week of promotion**

from\to	1	2	3	4	5
1	80.62%	2.67%	8.43%	2.95%	5.34%
2	18.18%	42.42%	12.12%	15.15%	12.12%
3	55.17%	6.90%	24.14%	3.45%	10.34%
4	22.22%	38.89%	0.00%	33.33%	5.56%
5	10.29%	7.35%	19.12%	17.65%	45.59%

**Figure D.3: Transition matrix for the second week of promotion**

from\to	1	2	3	4	5
1	76.20%	2.98%	10.08%	4.63%	6.12%
2	21.28%	53.19%	2.13%	12.77%	10.64%
3	41.86%	8.14%	25.58%	1.16%	23.26%
4	27.27%	36.36%	2.27%	15.91%	18.18%
5	9.09%	10.39%	24.68%	7.79%	48.05%

**Figure D.4: Transition matrix for the third week of promotion**

---

from\to	1	2	3	4	5
1	77.82%	1.72%	9.75%	2.68%	8.03%
2	13.51%	48.65%	4.05%	25.68%	8.11%
3	55.24%	5.71%	30.48%	1.90%	6.67%
4	45.83%	31.25%	2.08%	16.67%	4.17%
5	7.41%	6.48%	24.07%	17.59%	44.44%

**Figure D.5: Transition matrix for the fourth week of promotion**

from\to	1	2	3	4	5
1	78.19%	4.32%	9.04%	2.95%	5.50%
2	23.29%	50.68%	2.74%	5.48%	17.81%
3	33.04%	6.09%	42.61%	4.35%	13.91%
4	20.97%	40.32%	8.06%	19.35%	11.29%
5	15.24%	5.71%	19.05%	9.52%	50.48%

**Figure D.6: Transition matrix for the fifth week of promotion**

from\to	1	2	3	4	5
1	88.24%	1.26%	5.04%	2.10%	3.36%
2	23.71%	54.64%	7.22%	6.19%	8.25%
3	55.00%	6.67%	20.83%	3.33%	14.17%
4	25.00%	29.55%	15.91%	13.64%	15.91%
5	17.80%	9.32%	21.19%	11.02%	40.68%

**Figure D.7: Transition matrix for the sixth week of promotion**

---

## Appendix E: Variation In Prices

These results are based on the store database introduced in the beginning of chapter 6.

Average over all the stores and all 80 week the average price is \$2.4549 and the standard deviation is 0.2757.

Conditioning on the state we get,

state	mean	std
1	2.53	0.23
2	2.26	0.27
3	2.15	0.24
4	2.07	0.22
5	2.15	0.19
all	2.45	0.28

Conditioning on the state and store we get,

	All states		regular		display		feature		dis+fea		unsupp	
store	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
124	2.29	(0.13)	2.31	(0.10)	2.23	(0.08)	2.11	(0.25)	1.79			
125	2.30	(0.13)	2.32	(0.09)	2.19	(0.00)	2.05	(0.26)				
127	2.53	(0.31)	2.67	(0.04)			1.99	(0.33)	1.89	(0.14)	1.96	(0.08)
128	2.39	(0.19)	2.46	(0.12)	2.29		2.03	(0.09)			2.17	(0.15)
131	2.36	(0.11)	2.39	(0.00)	2.12	(0.23)	2.09	(0.14)	1.94	(0.08)	1.99	
136	2.23	(0.22)	2.33	(0.08)	2.12	(0.26)	1.91	(0.27)	1.86	(0.12)	1.85	(0.09)
223	2.32	(0.08)	2.33	(0.04)			2.19	(0.00)	1.99	(0.00)		
3289	2.63	(0.12)	2.66	(0.06)	2.58	(0.09)	2.32	(0.22)	2.39	(0.11)		
3363	2.05	(0.13)	2.07	(0.16)	1.99	(0.10)	2.01	(0.06)	1.96	(0.06)	2.08	(0.08)
3364	2.01	(0.08)	2.02	(0.08)	1.97	(0.04)	1.99		1.89	(0.00)	1.91	(0.02)
3365	2.07	(0.23)	2.36	(0.13)			1.88	(0.00)			1.91	(0.03)
3374	2.16	(0.14)	2.26	(0.06)	2.06	(0.11)			1.92	(0.06)	1.98	(0.02)
3381	2.37	(0.28)	2.56	(0.10)	2.00	(0.01)	1.99	(0.00)	2.00	(0.00)	1.99	(0.00)
mean	2.46	(0.17)	2.53	(0.07)	2.28	(0.13)	2.12	(0.15)	2.07	(0.10)	2.15	(0.08)
median	2.45	(0.16)	2.53	(0.07)	2.29	(0.11)	2.09	(0.13)	2.00	(0.10)	2.14	(0.07)

---

## Appendix F: Proof Of The Validity Of Counting

In this section we show that finding a maximum likelihood estimate for a Markov-chain is equivalent to counting the transitions between states.

The problem is to estimate a matrix of probabilities,  $P$ , such that the likelihood of what the data shows is maximized (when doing this maximization we always condition on the state that we are in and find pdf for the next state). Let us look at one state, say state  $i$ . From state  $i$  count the transitions and let these numbers be  $b_{i,j}$  for  $j = 1, \dots, n$  (we also count self transitions, that is transitions from  $i$  to  $i$ ). The maximum likelihood problem can then be written as a mathematical program with linear constraints but non-linear objective function,

$$\begin{aligned} \max z_i &= \prod_{j=1}^n (p_{i,j})^{b_{i,j}} \\ \text{s.t. } \sum_{j=1}^n p_{i,j} &= 1 \\ p_{i,j} &\geq 0 \quad \forall j \end{aligned} \tag{F.1.1}$$

Writing out the Lagrangian and differentiating results in,

$$u \frac{b_{i,j}}{p_{i,j}} \prod_{k=1}^n (p_{i,k})^{b_{i,k}} = \prod_{k=1}^n (p_{i,k})^{b_{i,k}} \tag{F.1.2}$$

Canceling the common term it is clear that the maximum is where the probabilities are proportional to the number of times these transitions occur. Thus the maximum likelihood estimate is,

$$p_{i,j,ML} = \frac{b_{i,j}}{\sum_{j=1}^n b_{i,j}} \quad \forall i, \forall j \tag{F.1.3}$$

---

Let  $\bar{\pi}$  be the steady state probability vector. It is straight forward to show that solution to the equilibrium equations,  $\bar{\pi} = \bar{\pi}P$ ,

$$\pi_j = \frac{\sum_{i=1}^n b_{i,j}}{\sum_{j=1}^n \sum_{i=1}^n b_{i,j}} = \frac{\sum_{i=1}^n b_{j,i}}{\sum_{j=1}^n \sum_{i=1}^n b_{i,j}} \quad \forall j \quad (\text{F.1.5})$$

We thus have that the steady state probabilities are the same as the proportion of time spent in each state in the underlying data used for calibration of the transition matrix.