ANAND V. BODAPATI*

Many firms use decision tools called “automatic recommendation systems” that attempt to analyze a customer’s purchase history and identify products the customer may buy if the firm were to bring these products to the customer’s attention. Much of the research in the literature today attempts to recommend products that have a high probability of purchase (conditional on the customer’s history). However, the author posits that the recommendation decision should be based not on purchase probabilities but rather on the sensitivity of purchase probabilities to the recommendation action. This article attempts to model carefully the role of firms’ recommendation actions in modifying customers’ buying behaviors relative to what the customers would do without such a recommendation intervention. The author proposes a simple consumer behavior model that accommodates a transparent role for a firm’s recommendation actions. The model is expressed in econometric terms so that it can be estimated with available data. The author studies these ideas using purchase data from a real e-commerce firm and compares the performance of the proposed main model with the performance of benchmark models. The author shows that the main model is better than benchmark models on key measures.

Keywords: customer relationship management, one-to-one marketing, up-selling, cross-selling, automatic recommendation systems, Internet marketing, direct marketing, stochastic choice models, hierarchical Bayes models, dual latent class models

Recommendation Systems with Purchase Data

A cornerstone idea in customer relationship management (CRM) is that a firm should emphasize selling more products to existing customers rather than merely acquiring more customers. This idea of “add-on selling” makes sense if the customer acquisition cost is high relative to the marketing cost required to induce purchases from existing customers (Blattberg, Getz, and Thomas 2001; Rust, Zeithaml, and Lemon 2000). To achieve add-on selling, many firms use decision tools called “automatic recommendation systems” that attempt to analyze a customer’s purchase history and identify products the customer may buy if the firm were to bring these products to the customer’s attention. For example, firms such as Amazon.com make fairly heavy use of such recommendation systems in an attempt to achieve add-on selling. This article addresses the question of how customer histories should be analyzed to identify products for the firm to recommend to a specific customer. Much research in the marketing science, computer science, and artificial intelligence literature streams focuses on this issue. I review this literature subsequently in this section. However, this article differs from others in two important respects.

First, to the best of my knowledge, all the work in the literature to date attempts to recommend products that have a high probability of purchase (conditional on the customer’s history) rather than products whose purchase probabilities have high sensitivity to recommendations. A simple example helps illustrate this point: Suppose that a customer with purchase history H is likely to buy product B_1 with probability $p(B_1|H) = .95$ and product B_2 with probability $p(B_2|H) = .75$. A conditional probability, such as $p(B_2|H)$, can be estimated either with an econometric model or, as is often done in industry practice if H is short, with the simple

*Anand V. Bodapati is Associate Professor of Marketing, Anderson School of Management, University of California, Los Angeles (e-mail: bodapati@ucla.edu). The author is grateful to Scott Neslin and the two anonymous *JMR* reviewers for their careful reading, detailed comments, and insightful suggestions on multiple versions of this article. They made this article vastly better than it otherwise would have been.

ISSN: 0022-2437 (print), 1547-7193 (electronic)

empirical fraction obtained from counting the number of customers who have bought B_1 among customers with purchase history H. Which product should the firm recommend to the customer? The position taken implicitly or explicitly by all the articles in the literature to date is that the firm should recommend product B_1 because a customer with purchase history H is more likely to purchase B_1 than B_2. However, a case can be made for recommending B_2 instead: The numbers suggest that the customer is very likely to buy product B_1. Given that he or she has purchase history H, the customer will probably buy product B_1 anyway, even if the firm does not make an explicit recommendation for B_1. This follows from the way the conditional probabilities are computed and interpreted. Therefore, it can be argued that the firm should recommend product B_2 because it is less certain that the customer will buy that product anyway. Again, I pose the key question: Assuming that only one product can be recommended, which product should the firm recommend to the customer? Should the firm's marketing message recommend B_1, or should it recommend B_2? Let the two messages be labeled R_1 and R_2, respectively; the trouble is that there are not enough data to answer the question. To decide on R_1 versus R_2, assuming that recommendations do not have cross-effects, it is necessary to know not just $p(B_1|H)$ versus $p(B_2|H)$ but $p(B_1|H, R_1)$ versus $p(B_2|H, R_2)$ as well. In other words, the recommendation decision should be based not only on the raw purchase probability but also on the purchase probability conditional on the recommendation; thus, it is important to consider the sensitivity of the purchase to the recommendation. Subsequently, I consider various decision rules for recommendation systems. Different rules are correct under different settings, but most optimal rules account not only for $p(B_1|H)$ and $p(B_2|H)$ but also for $p(B_1|H, R_1)$ and $p(B_2|H, R_2)$ because it is necessary to consider the response with and without a certain marketing intervention.

The idea that marketing actions should be chosen on the basis of expected response to the actions is well entrenched in marketing science, but this idea has somehow been ignored in recommendation systems. Indeed, this kind of argument is fundamental and is routinely made early in many MBA courses. In MBA courses on pricing, for example, students are taught that price discrimination across two customers should be done based not only on just their sales volumes but also on their price elasticities. Similarly, in MBA courses on direct marketing, students are taught that the decision of whom to mail a marketing communication to should be based not on the response probabilities but rather on the response probability sensitivity of the mailing. The analogous point in the current setting is that the recommendation decision should be based not only on purchase probabilities but also on the sensitivity of purchase probabilities to the recommendation action. This is a fundamentally important framing that has been absent in all literature to date.

This article attempts to model carefully the role of firms' recommendation actions in modifying customers' buying behaviors relative to what the customers would do without such recommendation interventions. This is accomplished by examining data on customers' purchases and firms' recommendation actions and customers' subsequent behavior. It is not possible to make optimal recommendations to the customer without an explicit understanding of the effect of those recommendations on purchase behavior. In this article, I propose a simple consumer behavior model that accommodates a transparent role for the firm's recommendation actions. The model is expressed in econometric terms so that it can be estimated with available data.

The second aspect of differentiation between this article and much of the work in the literature is that this article assumes that the firm has access only to purchase behavior data. The overwhelming majority of articles assume that the firms have access to ratings data from customers—for example, when each customer rates each item on a five-point scale to indicate the extent to which he or she likes or dislikes an item. In many contexts, this can be unattractive because getting good ratings data entails a proper survey mechanism that can collect such data accurately and efficiently. This may be difficult because of demand effects and low cooperation rates from customers in such surveys. To address this problem, a small number of articles work with binary data; in these studies, each user is assumed to give a binary response (i.e., "thumbs up" versus "thumbs down") to indicate like or dislike of a product. A commonly advanced argument is that thumbs up versus thumbs down does not require survey mechanisms but can be inferred from just purchase behavior. However, this is not true because purchase data should not be viewed as binary data but rather as "unary" data in the sense that a customer purchasing a product can reasonably be taken as the customer liking the product (thumbs up), but a customer not purchasing a product cannot be taken as a customer disliking the product (thumbs down). This is because the nonpurchase of the product may have been because the customer was just not aware of the product rather than because of his or her dislike of the product. Herlocker (2000) coined the term "unary data" to capture the inherent one-sidedness in the inferences that can be drawn from purchase data about like versus dislike of a product. Other authors have also suggested that a purchase behavior data set is more correctly treated as unary data than as binary data, though they have not used the term "unary" (see, e.g., Breese, Heckerman, and Kadie 1998). However, for technical difficulties that I explain subsequently, thus far, no article has proposed a statistical model to handle unary data in recommendation systems. As far as I can tell, this article is the first to treat purchase behavior data strictly as unary data.

The rest of this article is organized as follows: I begin with a review of the literature. Then, I propose a statistical model for unary data that accommodates an explicit role for the influence of the firm's recommendations actions. I go on to discuss the implication of the model for forecasting customer behavior and identifying the best recommendations. I then discuss methods to estimate the model under a latent class structure for heterogeneity in the products' attributes and the customers' response parameters. This is followed with an illustration of uses of the model using real purchase data from an e-commerce firm. In the final section, I offer some ideas for future work.

THE LITERATURE TO DATE: DATA MODELS AND COMMENTS

The literature on automatic recommendation systems operates on three different kinds of data models; in general, these can be labeled as (1) the ratings data model, (2) the binary data model, and (3) the unary data model. Imagine that the data are laid out as a matrix, with rows corresponding to users and columns corresponding to items. In ratings data, each user reports a vote for each item in a subset of items on a scale, for example, from 1 to 5. If the user is unaware of an item or is otherwise unable or unwilling to report the vote for an item, the vote is considered missing data, and the corresponding matrix element is represented as "N.A.," which stands for not available. Different users have different subsets of items with missing data. Typically, most users are unaware of most items, so the ratings data matrix will be overwhelmed by N.A. responses. The binary data can be viewed as a truncated version of the ratings data. Each user expresses a positive valence (which is recorded as a 1) for an item by purchasing the item or giving it a rating that meets some threshold; the user expresses a negative valence (recorded as a 0) if he or she indicates an intention not to purchase the item or if the rating falls below the threshold. If the user's rating or purchase intention is missing, the valence is also considered missing and, again, is recorded as N.A. The unary data are a censored version of the binary data. If it is assumed that only positive valences are observed and negative valences are censored, each data element in which a negative valence is expressed also becomes missing and is recorded as N.A. The important point to note is that an N.A. for an item indicates that the user is either unaware of that item or aware of the product but has negative valence toward it.

The vast majority of the work in the automatic recommendation systems literature operates on the ratings data model (see, e.g., Herlocker, Konstan, and Riedl 2002). There have been numerous approaches to predict unknown values in the ratings matrix. The most common approach is nearest-neighbor collaborative filtering (Herlocker, Konstan, and Riedl 2002), in which a user's rating for an item is imputed as the mean of others' ratings for that item; more similar users' ratings are given greater weight. Other approaches include the use of nonparametric methods, such as neural networks and tree classifiers (Breese, Heckerman, and Kadie 1998). The literature also proposes model-based methods, such as those in Chien and George (1999), Ungar and Foster (1998), and Ansari, Essegaier, and Kohli (2000). Ansari, Essegaier, and Kohli's work is a singularly important contribution because it posits a stochastic framework that is immediately appealing to marketing scientists; each user's rating for an item arises from the user evaluating a linear utility function on the attributes of the item. The clever observation in their work is that the attributes, even if unknown to the analyst, can be imputed from the observed ratings data in a data augmentation framework. As I show subsequently, one of the modeling devices in the current article follows this idea and applies it in the context of unary data. Similar to Ansari, Essegaier, and Kohli's work, Ying, Feinberg, and Wedel's (2006) work operates under ratings data.

There is limited research that uses the binary data model. Notable exceptions include the works of Iacobucci, Arabie, and Bodapati (2000), Weiss and Indurkhya (2001), Mild and Reutterer (2003), and Linden, Smith, and York (2003), all of whom consider extensions of the usual nearest-neighbor collaborative filtering approach from the ratings data case to the binary data case. The articles by Breese, Heckerman, and Kadie (1998), Moon and Russell (2005), and Ansari and Mela (2003) are important because they are among the few examples that offer model-based collaborative filtering with models considered specifically for binary data.

In real-world systems, explicit self-reports of ratings are not observed as frequently as behavioral data in the form of purchases; a purchase may be interpreted as indicating positive valence for an item, but there is no direct way to observe negative valence. Therefore, real-world data are closer to the unary data model than to either of the other two data models. However, the few articles that work with purchase data simply assume that the data are binary. By this, I mean that instead of considering that a nonpurchase event can imply either an explicit negative valence or a lack of awareness of the item, much of the research to date treats nonpurchases as if they correspond explicitly to negative valences. There are no examples that I am aware of that treat unary data properly and correctly model the corresponding data generation process.

The mainstream problem formulation in recommendation systems assumes that preference and goals are largely stable and stationary. The operating assumption is that the sequence of purchases is not informative and that inference is to be drawn from just the collection of choices regardless of ordering. The current article adopts this mainstream problem framing. However, in recent years, a secondary research stream has developed that assumes a sequential setup for goals, needs, and preferences and uses knowledge about purchase sequences for models that support recommendations for cross-selling. Notable examples in this stream are the works of Kamakura and colleagues (2003), Kamakura, Kossar, and Wedel (2004), Knott, Hayes, and Neslin (2002), and Li, Sun, and Wilcox (2005).

A MODEL FOR THE EFFECT OF FIRM RECOMMENDATIONS ON CUSTOMER PURCHASE BEHAVIOR

In this section, I develop a model for the effect of a firm's recommending an item on customer purchase behavior. The model operates on purchase data and applies the unary data model; thus, purchases are taken to indicate positive valence, but nonpurchases are not necessarily assumed to indicate negative valence.

A brief, big-picture summary of the modeling approach of the current article is as follows: Purchases are viewed as being of two conceptually distinct types:

- •Firm-initiated purchases, or purchases the consumer makes as a consequence of the firm making recommendations, and
- •Self-initiated purchases, or purchases other than firm-initiated purchases.

I make the reasonable assumption that firms have access to data on these two types of purchases. In the paragraphs that follow, I propose a model of consumer behavior that implies specific likelihood function forms for these two types of data. I then use the estimated parameters from the model to forecast the probabilities that a specific customer will purchase a certain item without a recommendation and with a recommendation. These two probabilities can be used together to determine the impact of the firm recommending that particular item to that particular customer.

I propose the following consumer behavior process: Let $\mathrm{x_i}$ be a vector of characteristics for item i that are observed by the user but not by the analyst. I propose that to buy a product i, the consumer goes through two logically distinct steps:

- •Awareness (A): becoming aware of product i and its characteristics x_i; and
- •Satisfaction (S): evaluating x_i and buying the product if the anticipated postconsumption utility for the product exceeds some threshold.

A customer buys an item only if both A and S occur. Not buying a product—in other words, observing an N.A. for a product—is taken to mean that the consumer is either unaware of the product or aware of the product but considers the product of low utility.

Before proceeding further, I must explain briefly the motivation behind the assumption that the attribute vector x_i is observed by the user but not by the analyst, particularly because this assumption complicates the estimation methodology to some degree because the x_i must be statistically inferred from the purchase data. In many e-commerce and direct-marketing contexts, customers are presented product descriptions in the form of photographs and verbose English language (or some other natural language) text. Presumably, the customer processes these descriptions to generate an internal representation of the product attributes, or x_i. Because the descriptions a firm presents often fail to lend themselves to easy codification and because the translation to the internal representation can happen in a highly context-dependent manner, it is difficult for the analyst to specify an appropriate x_i for use in a model. This problem is more severe in some industries than in others. In consumer electronics, there are many items that can be adequately described in terms of well-accepted technical specifications. Conversely, in the clothing industry, customers' perceptions of a product shown in a catalog are so holistic and affect driven and so sensitive to the customers' assessment of what other consumers value that even "experts" in the industry find it difficult to generate the right set of attributes for codification. In this sense, the x_i can be taken to be observed by the user but not by the analyst.

To define the model properly, it is necessary to come up with specifications for the probabilities of events A and S. I use a simple logistic function form for both events. The likelihood of event A_{ui}, the consumer u becoming aware of product i, is posited as

$$p(A_{ui}) = \text{logistic}(\alpha_u^T x_i), \tag{1}$$

where α_u is a user-specific parameter vector of response coefficients for awareness and the logistic function is defined as usual by logistic(z) = [exp(z)]/[1 + exp(z)]. The likelihood of event S_{ui}, conditional on event A_{ui}, is posited as

$$p(S_{ui}|A_{ui}) = \text{logistic}(\beta_u^T x_i), \tag{2}$$

where β_u is a user-specific parameter vector of response coefficients for satisfaction.

I now discuss how this model implies certain likelihood functions for the self-initiated purchase data and the firm-initiated purchase data. For reasons I discuss subsequently, in the typical application, the overwhelming majority of the data are of the self-initiated type. Thus, I first discuss the likelihood for that data type.

Let y_{ui} denote the data element in the self-initiated purchase data set: y_{ui} takes the value of 1 if consumer u purchases item i and N.A. if otherwise. For a purchase to happen, the consumer needs to go through both the awareness step, A, and the satisfaction step, S. Therefore, the probability of consumer u purchasing item i is

$$\begin{aligned} p(y_{ui} = 1|\alpha_u, \beta_u, x_i) &= p(A_{ui}, S_{ui}) \\ &= p(A_{ui})p(S_{ui}|A_{ui}) \\ &= \text{logistic}(\alpha_u^T x_i)\text{logistic}(\beta_u^T x_i). \end{aligned} \tag{3}$$

The likelihood of observing an N.A. is simply the complement:

$$p(y_{ui} = \text{NA}|\alpha_u, \beta_u, x_i) = 1 - p(y_{ui} = 1|\alpha_u, \beta_u, x_i). \tag{4}$$

Most work in the literature uses ratings data. Of the few articles that use purchase data, it seems that none distinguish between the events A and S. In general, they treat an observed product purchase as an indicator that S_{ui} is true and the lack of a product purchase as an indicator that S_{ui} is false. In consumer behavior terms, however, these two events are more correctly interpreted as $A_{ui} \cap S_{ui}$ being true and $A_{ui} \cap S_{ui}$ not being true, respectively. Therefore, these other articles' attempts to predict S_{ui} often end up being predictions of $A_{ui} \cap S_{ui}$. Accordingly, the recommendation lists constructed usually present items with the highest values of $p(A_{ui}, S_{ui})$, the likelihood of this joint event. However, the goal of a recommendation list is to make people aware of items they would buy if they were to become aware of the products. Therefore, the recommendation list construction should account for the values of $p(S_{ui}|A_{ui})$, the likelihood of the conditional event rather than the joint event. Because these two likelihoods will typically impute different orderings of items, the consequent recommendation lists will differ.

Recall that it is assumed that the item characteristics x_i are unobserved and thus are to be treated as parameters to be estimated. Furthermore, assume that x_i, α_u, and β_u are each of dimensionality d.

If there are U users and I items, the unknown parameters to be estimated are

$$\Theta \equiv \left\{\{\alpha_u, \beta_u\}_{u=1}^{U}, \{x_i\}_{i=1}^{I}\right\}. \tag{5}$$

Given observed unary data $\{y_{ui}\}_{u=1,i=1}^{U,I}$, the likelihood function can be written as

$$L_y(\Theta) = \prod_{u=1,i=1}^{U,I} p(y_{ui}|\alpha_u, \beta_u, x_i). \tag{6}$$

From the point of view of building recommendation lists, interest centers on β_u because that is what will play a key role in determining which items should be presented to the user. However, as things stand, it is not possible to identify β_u properly, because its position in the likelihood function is completely symmetric and interchangeable with that of α_u (see Equation 3).

To be able to identify β_u, some point of differentiation with α_u is needed; such a point of differentiation is obtainable with the second kind of data, namely, the firm-initiated purchase data that represent data on responses to recommendations. Such data are frequently collected with the clickstream or call-center logs, in which the firm records occasions when a recommendation is made and whether the customer buys the product being recommended. When such data are available, it becomes possible to identify α_u and β_u separately. Several models may be proposed to exploit such data, but this article pursues a simple model.

Let r_{ui} denote the data element in the firm-initiated purchase data set that takes the value of 1 if the firm recommends item i to user u and 0 if otherwise. The term n_{ui} is defined as the binary complement of r_{ui}, and it takes the value of 1 if the firm does not recommend item i to user u and 0 if otherwise. Suppose that the firm makes a recommendation for a product x_i to user u and succeeds in forcing an awareness of the product i on the user and making the event A_{ui} happen. The response to the recommendation is denoted as v_{ui}, which takes the value of 1 if the recommendation is accepted (meaning that the recommended product is purchased) and 0 if otherwise. Assume that the acceptance of the recommendation entails the same utility valuation process the consumer goes through as when the awareness of the product is created by the consumer him- or herself rather than by the firm. This assumption leads to the following:

$$\begin{aligned}(7) \qquad p(v_{ui} = 1|\beta_u, x_i) &= p(S_{ui}|A_{ui})p(A_{ui}|r_{ui} = 1)\\ &= p(S_{ui}|A_{ui});\\ &= \text{logistic}(\beta_u^T x_i)\\ p(v_{ui} = 0|\beta_u, x_i) &= 1 - p(v_{ui} = 1|\beta_u, x_i).\end{aligned}$$

The transition from the first part to the second part of Equation 7 assumes that when the firm recommends an item to the user, it successfully creates awareness of that item for that user and that $p(A_{ui}|r_{ui} = 1)$ is equal to 1. If data are available on some values of v_{ui}, it is possible to identify β_u and α_u separately.

I now summarize these ideas and formalize them so that the distinction between the self-initiated purchase data and the firm-initiated purchase is made more precise. The term **R** denotes the set of user–item pairs (u, i) when the firm has recommended item i to user u and the value of v_{ui} is observed. Similarly, the term **N** denotes the set of user–item pairs (u, i) when the firm has not recommended item i to user u. Precisely stated,

$$\mathbf{R} = \{(u, i) : r_{ui} = 1\}$$

$$\mathbf{N} = \{(u, i) : n_{ui} = 1\}.$$

The total data can be viewed as consisting of two logically distinct components with different forms for the likelihood of the responses. First, there are responses in the recommendation set **R**: For every user–item pair in **R**, response v_{ui} is observed, which can take a value of 0 or 1. The likelihood for the data in **R** would be the likelihood for the collection of observed $\{v_{ui}\}$ and would be given by

$$(8) \qquad L_R(\Theta) = \prod_{(u,i) \in R} p(v_{ui}|\beta_u, x_i).$$

The expression for $p(v_{ui}|\beta_u, x_i)$ is given in Equation 7.

Second, there are responses in the nonrecommendations set **N**: For every user–item pair in **N**, response y_{ui} is observed, which can take the value of N.A. or 1. The likelihood for the data in **N** would be the likelihood for the collection of observed $\{y_{ui}\}$ and would be given by

$$(9) \qquad L_N(\Theta) = \prod_{(u,i) \in N} p(y_{ui}|\alpha_u, \beta_u, x_i).$$

The expression for $p(y_{ui}|\alpha_u, \beta_u, x_i)$ is given in Equation 3 and Equation 4.

The recommendation set **R** corresponds to firm-initiated purchases in the sense that the awareness step for these purchases is facilitated by the firm's recommendation actions. The nonrecommendation set **N** corresponds to self-initiated purchases in the sense that the awareness step for these purchases is achieved by the consumer using his or her own awareness generating resources.

The net likelihood for Θ, based on the overall data from both the recommendation set **R** and the nonrecommendation set **N**, is given by

$$\begin{aligned}(10) \qquad L_o(\Theta) &= L_R(\Theta) \times L_N(\Theta)\\ &= \prod_{(u,i) \in R} p(v_{ui}|\beta_u, x_i) \prod_{(u,i) \in N} p(y_{ui}|\alpha_u, \beta_u, x_i).\end{aligned}$$

Note that in this likelihood, the positions of β_u and α_u are not interchangeable; therefore, they can be estimated separately. Given that β_u and α_u appear everywhere only in a dot-product with x_i, it may seem that these three constructs are not properly separable and identifiable. However, as in the similar situation in factor analysis, placing certain constraints makes all the constructs fully identified. I discuss this particular issue in greater detail subsequently.

At first, it may appear that the model requires that awareness and satisfaction be driven by the same factors. However, this is not true. No assumption is made that the same x variables influence awareness and satisfaction or that the products are described by a restricted set of attributes that apply to all products. Strictly speaking, the x contains the union (and not the intersection) of attributes that influence awareness and satisfaction over all items. If the driver variables are completely distinct in the model—for example, x_A drives awareness, and x_S drives satisfaction—the x should be taken as $[x_A x_S]$, the concatenation of the two sets of variables. The elements in the α vector corresponding to the x_S terms and the elements in the β vector corresponding to the x_A terms would take the value of 0 in the estimates to adapt to the notion that the awareness responds only to the x_A terms and satisfaction responds only to the x_S terms. Statistically speaking, if x_A and x_S are indeed distinct, in principle, this can be picked out by inspecting the estimated values of x, α, and β. A similar argument would apply for the case of different x descriptors applying to different items.

I now briefly discuss the desired characteristics of the **R** set. For statistical efficiency of the estimates obtained by maximizing the likelihood in Equation 10, it would be desirable to have the recommendation set **R** be randomly drawn from a set of products that widely span the space of attributes represented by the x vectors. However, even if the set is not randomly drawn, consistent (and thus asymptotically unbiased) estimates of the parameters are still obtained. In particular, consistency is obtained even if recommendations are based on the customer's previous purchases. The key assumption behind the consistency argument is that the response probability depends only on the product characteristic x and the parameter vector β. As long as this key assumption is met, under fairly broad conditions, consistency is obtained regardless of the mechanism by which the x vectors (and, equivalently, the recommended items) are selected. Conversely, if this key assumption is violated, the estimates can be biased and inconsistent. A possible route to such violation is when the recommendation set affects the valuation process parameter β itself. This is akin to prices affecting the price coefficient in scanner data models or stimuli in conjoint analysis affecting the partworths. In such a scenario, a kind of endogeneity is encountered in which the usual estimates become inconsistent.

USING THE MODEL PARAMETERS TO MAKE PRODUCT RECOMMENDATIONS: A DECISION FRAMEWORK FOR THE FIRM

An Adjustment When Forecasting for Durations Different from the Calibration Duration

Implicit in the model estimates is a specific length of time over which purchases are modeled as occurring or not occurring. Specifically, the construct $p(A_{ui}, S_{ui})$ is the probability that a self-initiated purchase for user u and item i occurs in the data over the length of time represented in the data. Strictly speaking, this should be interpreted as the purchase probability in a replication of the data generation process over a comparable period. What happens if the forecast is being made for a duration considerably different from the duration of the calibration data? For example, suppose that the model was estimated with three months of data and that the model estimate for a consumer u buying a certain as-yet-unpurchased item i is $p(A_{ui}, S_{ui}) = .03$. If the data-generating process for the next three months can be argued to be a replication of the calibrating three months, the probability of the customer purchasing this item in the next three months is also .03. What is the probability of purchasing in the next one month? Intuition suggests that this probability would be approximately .03/3 = .01. Formally, let T_c be the time interval length of the calibration data, let T_f be the time interval length for the forecast, and let $p(A_{ui}, S_{ui})$ be the purchase probability in the T_c duration. The purchase probability in the T_f duration is closely approximated by $(T_f/T_c)p(A_{ui}, S_{ui})$.[1] The formal argument appears in Appendix A, which models awareness as a memoryless exponential arrival process and derives an exact expression for a purchase occurring in a time interval of a certain length. In general, the manager wants to make forecasts of customer behavior in a decision-planning period that will often be significantly shorter than the data calibration period. Therefore, it is important that the adjustment factor (T_f/T_c) be applied.

A Decision Framework for the Optimal Recommendation

I now investigate what would be the best recommendations for the firm to make. As discussed previously, the usual policy in the recommendation systems literature is to recommend products with high predicted preference ratings or purchase probabilities based on historical buying patterns. In terms of the notation used herein, this corresponds to recommending items with high values of $p(y_{ui} = 1)$, which is $p(S_{ui}|A_{ui})$. This policy does not explicitly consider what happens in response to a recommendation. To formalize the decision framework for the firm, consider the purchase behavior of the customer in a future observation period that is equivalent in offerings and competitive positioning to the historical observation period in which the existing data have been collected. Let T_c be the time interval length of the calibration data, and let T_f be the time interval length for the planning period of the recommendation decision.

The probability of consumer u purchasing item i in the absence of a recommendation is $(T_f/T_c)p(A_{ui}, S_{ui})$, as discussed previously. If the firm recommends item i, in line with the logic used for Equation 7, the purchase probability is $p(A_{ui}, S_{ui})$. This means that the expected incremental number of purchases generated from recommending item i, relative to the situation in which no item is recommended, is

$$\Delta_i = \left[p(S_{ui}|A_{ui}) - \frac{T_f}{T_c}p(A_{ui}, S_{ui})\right]. \tag{11}$$

If all items have the same profit contribution,[2] the best item to recommend is the one with the highest value of Δ_i. So the best item is

$$k = \arg\max_i \left[p(S_{ui}|A_{ui}) - \frac{T_f}{T_c}p(A_{ui}, S_{ui})\right]. \tag{12}$$

If the expressions from Equations 3 and 7 are substituted, this becomes

$$k = \arg\max_i \text{logistic}(\beta_u^T x_i)\left[1 - \frac{T_f}{T_c}\text{logistic}(\alpha_u^T x_i)\right]. \tag{13}$$

Instantaneous Policy Versus Broad-Range Policy

I now consider the special case in which the planning period T_f shrinks to 0. If T_f is set to 0 in Equation 11, the expected incremental number of purchases from recommending i becomes

$$\Delta_i = p(S_{ui}|A_{ui}). \tag{14}$$

So, the best item to recommend is

$$k = \arg\max_i p(S_{ui}|A_{ui}) \tag{15}$$

$$= \arg\max_i \text{logistic}(\beta_u^T x_i). \tag{16}$$

I call the policy for this situation the "instantaneous policy." For the more general situation of Equation 12, in which T_f is moderately large so that T_f/T_c is not negligible, I use the term "broad-range policy."

The broad-range policy takes the view that over the course of the planning period, there are some items—specifically, those with high values of $(T_f/T_c)p(A_{ui}, S_{ui})$—

[1] The approximation is accurate for small values of (T_f/T_c) and $p(A_{ui}, S_{ui})$.

[2] If each item i has a different profit contribution—for example, ρ_i—the best item to recommend is the one with the highest value of $\rho_i\Delta_i$. As the "Empirical Illustration" section shows, because there are no financial data, all ρ_i are considered equal to 1.

that the customer would buy anyway, even without the firm intervening with a recommendation, so making a recommendation would not increase the purchase probability by much, and such a recommendation does not produce much incremental revenue. The broad-range policy wants to recommend items for which the recommendation boosts the purchase probability as much as possible relative to what the customer would have bought without the recommendation.

In contrast, the instantaneous policy attempts to maximize sales in the immediate term and specifically just for the next occasion the consumer visits the firm; therefore, it tries to recommend items with the highest recommendation acceptance probability. Which policy is better? Industry practitioners are divided on this issue. On the one hand, the instantaneous policy camp would point out that there is no guarantee that the consumer will indeed keep visiting the firm in the planning period to purchase at the level assumed by the broad-range policy. Particularly, if there is a threat of competition from new firms, it may be imprudent to count on such purchases occurring in the future. In such situations, it may be better to focus on the near term and get the consumer to buy as much as possible right away. A bird in the hand is worth two in the bush, as the saying goes. On the other hand, if the customer is loyal, if the firm has high "stickiness," and if there is some assurance that the anticipated level of self-initiated purchasing will indeed occur, the broad-range policy can be argued to be the better one.

HETEROGENEITY MODELING AND ESTIMATION METHODOLOGY

Unless there is a large number of observed purchases and responses to recommendations, it is not possible to estimate α_u and β_u for each user u or x_i for each item i. To circumvent this issue, a hierarchical Bayes model with certain generating priors can be posited so that inference about α_u and β_u for a certain user can be drawn from the behaviors of other users. There are various potential models for generating priors, but the easiest in this case would be a simple point-mass density, so that the resultant model is a kind of latent-class model. It is the "easiest" in the sense that the iterations for estimation can be executed cleanly and efficiently, especially when compared with models for which the generating prior is from the Gaussian or a Dirichlet process.

A Dual Latent Class Model for Heterogeneity in User Space and Item Space

If it is assumed that there are C_{item} latent classes for the items and C_{user} latent classes for the users, it is necessary to estimate a vector π_{item}, which gives the relative frequencies for each of the C_{item} classes of items, and a vector π_{user}, which gives the relative frequencies for each of the C_{user} classes of users. In addition, for each user class, an α vector and a β vector must be estimated, and for each item class, an x vector must be estimated; recall that these vectors are all of dimensionality d. Therefore, the total number of parameters is $(C_{item} - 1) + (C_{user} - 1) + (d \times C_{item}) + (2 \times d \times C_{user})$.

It is must be emphasized that this latent class model is a "dual latent class" model, which is distinct from the usual latent class models encountered in the choice modeling literature. In the usual latent class model, the predictor variables x are observed, and a latent class structure is posited only on the response coefficients (here, α and β). In the current situation, however, even the predictor variables are unobserved, and a latent class structure is posited on the predictor variables as well. Because the predictors are unobserved in addition to the response coefficients, the model is more akin to a factor analysis model than a response model, with the added complexity of a latent class structure imposed on both sets of variables. Therefore, this model is considerably more difficult to estimate than the usual latent class models encountered in the econometric choice modeling literature (for a discussion on the typology of latent variable models, see Bartholomew and Knott 1999).

Estimation by Markov Chain Monte Carlo

Let s = 1, …, C_{user} be used to index the latent classes in user space and t = 1, …, C_{item} be used to index the latent classes in item space. The vector of attributes for every item in the tth latent class for items is denoted by x_t. As in every latent class model, the assumption is that the members within a latent class are all homogeneous. For this reason, all the items within a certain latent class possess the same vector of attributes. The awareness and satisfaction coefficient vectors for each user in the sth latent class are denoted by α_s and β_s, respectively. The term Z_u^{user} is an integer indicator variable that takes a value between 1 and C_{user}, indicating the latent class to which user u belongs. Similarly, for each item i, the indicator variable Z_i^{item} takes a value between 1 and C_{item}. The entire collection of parameters for this latent class model, organized as seven blocks of variables, is denoted by Φ:

$$(17)\quad \Phi \equiv \Big\{\pi_{user}, \pi_{item}, \{Z_u^{user}\}_{u=1}^{U}, \{Z_i^{item}\}_{i=1}^{I}, \{\alpha_s\}_{s=1}^{C_{user}}, \{\beta_s\}_{s=1}^{C_{user}}, \{x_t\}_{t=1}^{C_{item}}\Big\}.$$

The Markov chain Monte Carlo (MCMC) theory is used to draw from the posterior density of Φ. Specifically, each of the seven blocks of variables is cycled through by drawing from each block's conditional distribution given the parameters in the other blocks. It is assumed that the reader is familiar with MCMC methods applied to hierarchical Bayes models from accounts such as Gelman and colleagues (2004), Carlin and Louis (2000), and Rossi, Allenby, and McCulloch (2006). From these sources, the reader could anticipate how to construct and draw from these full conditional distributions. Therefore, the discussion here is brief, focusing mainly on the less obvious aspects. Complete specifications of the conditional densities and mechanisms for generating random variates from each density are available in the Web Appendix (http://www.marketingpower.com/jmrfeb08).

Drawing values of π_{user} and π_{item}. If a Dirichlet prior is assumed for π_{user}, the posterior is also Dirichlet. The sufficient statistics for this Dirichlet are the number of users who fall into each class in the iteration at hand. The posterior distribution for π_{item} is constructed similarly from the values in $\{Z_i^{item}\}_{1=1}^{I}$.

Drawing values of Z_u^{user} and Z_i^{item}. In the π_{user} vector, let the element corresponding to the sth latent class be denoted by $\pi_{user}(s)$. The likelihood for user u's data under the assumption that the user belongs to class s is denoted by $L^{user}_{u,s}$. This likelihood is easily constructed from Equation 10 by (1) eliminating all terms that do not involve user u, (2) using for α_u and β_u the values of the α and β vectors for segment s, and (3) using for each x_i the x vector for the latent class to which item i belongs. The posterior conditional distribution of Z^{user}_u is multinomial with sample size 1, and the probability for the sth latent class is

$$\frac{\pi_{user}(s)L^{user}_{u,s}}{\sum_{s'}\pi_{user}(s')L^{user}_{u,s'}}.$$

The posterior conditional distribution of each Z^{item}_i is constructed analogously, ignoring every term in Equation 10 that does not involve item i.

Drawing values of $\{\beta_s\}^{C_{user}}_{s=1}$. For each latent class's β vector, assume a Gaussian prior with zero mean and a precision matrix that is close to zero and fixed. The posterior density is obtained by multiplying this prior by the conditional likelihood. The conditional likelihood for the β vector for latent class s is obtained from Equation 10 by (1) eliminating all terms that do not involve users from latent class s, (2) using for α_u and β_u the values of the α and β for the sth latent class, and (3) using for each x_i the x vector for the latent class to which item i belongs. There is no simple mechanism to draw from the posterior density of β_s. It might initially appear that it is necessary to use the Metropolis–Hastings algorithm, with the accompanying difficult task of constructing a suitable proposal density mechanism, but this is not the case. Because the conditional posterior of β_s turns out to be globally log-concave, Gilks and Wild's (1992) adaptive rejection sampling algorithm embedded in Gibbs sampling iterations can be used. In the current application, this is more computationally efficient than using the Metropolis–Hastings algorithm.

Drawing values of $\{\alpha_s\}^{C_{user}}_{s=1}$. These values are drawn using ideas close to what was discussed for the β vectors.

Drawing values of $\{x_t\}^{C_{item}}_{t=1}$. The conditional likelihood for the x vector for latent class t is obtained from Equation 10 by (1) eliminating all terms that do not involve items from latent class t, (2) using for x_i the value of the x vector for the tth latent class, and (3) using for each α_s and β_s the α and β vectors for the latent class to which user u belongs. (For details on drawing variates from this posterior, see the Web Appendix at http://www.marketingpower.com/jmrfeb08. This part is done using the random-walk Metropolis–Hastings algorithm.)

Identification Issues

The matrix with as many columns as the number of latent classes of users is denoted by **A**. Let each column correspond to the vector α from each segment. Recall that α, β, and x are each of dimensionality d. Therefore, **A** is of the size d × C_{user}. Similarly, let the matrix **B** contain the β vectors for all latent classes of users. It would also be of the size d × C_{user}. Finally, let the matrix **X** be a matrix of the size d × C_{item} and contain in each column the x vector for each latent class of items. An inspection of the likelihood function in Equation 10, along with Equations 8, 7, 9, 3, and 4, shows that the parameters α, β, and x appear only in inner products with one another. The likelihood function can be expressed fully in terms of the elements of the matrices $\mathbf{A}^T\mathbf{X}$ and $\mathbf{B}^T\mathbf{X}$. This means that the likelihood function identifies α, β, and x only through $\mathbf{A}^T\mathbf{X}$ and $\mathbf{B}^T\mathbf{X}$. However, these two matrix products identify the constituent factors only up to a linear transformation. To understand this, consider any invertible matrix **L** of the size d × d and construct matrices **A′**, **B′**, **X′**, such that

$$\mathbf{A}' = \mathbf{L}^{-1}\mathbf{A}$$

$$\mathbf{B}' = \mathbf{L}^{-1}\mathbf{B}$$

$$\mathbf{X}' = \mathbf{L}^{T}\mathbf{X}.$$

The likelihood value evaluated at [**A′**, **B′**, **X′**] is the same as when it is evaluated at [**A**, **B**, **X**]. Therefore, the likelihood function does not fully identify **A**, **B**, **X**.

To correct the identification problem, it is necessary to reduce the parameter space so that the likelihood identifies all the parameters in this reduced space. How much does the parameter space need to be reduced? Because the linear transformation matrix **L** has d^2 elements, there is an indeterminacy of d^2 parameters. Therefore, it is necessary to impose d^2 independent constraints on the parameters to reduce the space enough to identify all parameters. Indeterminacy due to linear transformations is a problem that frequently occurs in latent factor models (see Bartholomew and Knott 1999). The commonly used approaches suggest the following two routes to impose d^2 constraints:

1. *Orthogonality and unit norming*: Suppose that the rows of **X** are restricted to be mutually orthogonal [d(d – 1)/2 constraints] and of unit length (d constraints) and that the rows of **B** are restricted to be mutually orthogonal [d(d – 1)/2 constraints]; suppose also that **A** is unrestricted. Then, the total number of constraints is d^2, and in this restricted space, the **A**, **B**, **X** would be fully identified in the likelihood.
2. *Anchoring*: Suppose that any d columns of the **X** matrix are fixed to the d columns of some fixed, invertible d × d matrix. For example, the first d columns of **X** could be set to be the columns of the identity matrix of the size d × d. This fixes d^2 of the parameters, and the rest of the parameters are fully identified.

However, both routes have problems. The first route is the route most commonly used in areas such as factor analysis. However, it creates problems in the MCMC iterations. In the log-likelihood corresponding to Equation 10, an x vector for one item never appears in the same term as the x vector for another item. Therefore, one x vector is conditionally independent of another x vector. This simplifies the MCMC iterations. Imposing any orthogonality restriction would destroy this conditional independence and greatly complicate the drawing procedure. For this reason, the first route is largely unworkable.

The problem with the second route is that it is necessary to choose (1) which columns of **X** to anchor and (2) what to anchor to. Anchoring to an invertible matrix ensures identifiability, but to which specific invertible matrix? It is not necessary to worry about the choices on Points 1 and 2 if the choices do not matter. Unfortunately, these choices affect the behavior of the MCMC iterations. This is because different anchorings lead to different shapes for the likelihood functions of the unanchored parameters and impute different levels of autocorrelations in the MCMC iterations (for a discussion of this issue, see Loken 2005). Consequently, some anchorings can cause the MCMC iterations to go much slower than other anchorings, and poor choices can be costly. A few anchoring schemes were experimented with, and one scheme was identified that works reasonably well. This scheme is described in the Web Appendix (http://www.marketingpower.com/jmrfeb08).

Slow Convergence of the MCMC Procedure

The MCMC procedure outlined previously converges slowly. With random starting points and for problem sizes and data distributions similar to what is described in the "Empirical Illustration" section of this article, it took approximately 81 hours of central processing unit (CPU) time for the Markov chain to arrive within three standard deviations of the posterior mean. A procedure that takes about three days to run is certainly not objectionable if it is to be run just once. However, in any empirical application, the estimation procedure must be run several times for different choices of the model size parameters: the number of latent classes for items and users and the factor dimensionality d. In the current empirical application, more than 250 sets of choices for the model size parameters were run. To estimate so many different models in a reasonable time, it would be necessary to use several computers in parallel.

There are two reasons for the slowness. First, the random-walk Metropolis–Hastings iterations procedure for the x parameters moves slowly. Alternatives such as the hit-and-run method and the independence-chain approach were considered, but each has problems, and in these limited experiments, they do not help much in this situation. Second, the x parameters' draws tend to be strongly correlated with the α and β parameters. In the current model, β is like the coefficient vector in a logistic regression with the x variables as predictors, and the x is like the coefficient vector in a logistic regression with the β variables as predictors. As a result, they are strong functions of each other, and this induces correlations between them. A similar situation holds with x and α. The literature on Bayesian computation (Carlin and Louis 2000; Gelman et al. 2004) has pointed out that correlations between blocks of variables can greatly inhibit mixing, induce high autocorrelation, and increase the required burn-in time.

A Prelude Procedure for Acceleration: Mode Finding Based on Eigenvalue–Eigenvector Calculations

Two suggestions are commonly made for situations such as those discussed in the previous paragraph. First, it has been suggested to reparameterize the model so that blocks of variables are as uncorrelated as possible to increase mixing. However, this strategy cannot be pursued, because a suitable reparameterization could not be found for the current problem. Second, it has been suggested to use numerical methods to find the mode in the joint space of the correlated blocks of variables and to start the iterations from the mode. This approach recognizes that drawing β conditional on x and x conditional on β (and similarly with α) can lead to poor mixing if there are strong dependencies and that it may be better to work with the joint distribution rather than the three sets of conditional distributions. If there is a mechanism to find the mode in the joint space, the iterations can be started from there to avoid the long burn-in. As it turns out, there is indeed a way to find the mode in the joint space of α, β, and x. As pointed out previously in the paragraph on identification issues, the likelihood can be expressed in terms of matrix factorizations $\mathbf{A}^T\mathbf{X}$ and $\mathbf{B}^T\mathbf{X}$, where the $\mathbf{B}$ and $\mathbf{X}$ are parameters to be estimated. This situation is akin to that in factor analysis or principal components analysis, in which the likelihood is expressed using the product of a loading matrix and a factor score matrix, which are then estimated using an eigenvalue–eigenvector calculation.[3] The current situation is an extension of the factor analysis situation, with two differences: First, this article deals with data not from the Gaussian distribution but rather from the binomial distribution after a logistic transformation of the matrix products. Second, the likelihood is a function of two matrix factorizations that interact nonlinearly, largely because of the likelihood for the self-initiated purchasing, such that the logistic transformations get multiplied elementwise (see Equations 9, 3, and 4). Even with these two extensions, the current problem retains many of the basic characteristics of factor analysis and principal components analysis, and the likelihood maximizer can be obtained with an iterative sequence of eigenvalue–eigenvector calculations. Appendix B explains the reduction of the mode-finding problem to eigenvalue–eigenvector calculations. Because eigenvalue–eigenvector calculations are rapid, the mode-finding algorithm is fast, typically taking only approximately 12 CPU seconds in the empirical illustrations.

The mode-finding algorithm is used to accelerate the burn-in period as follows: The MCMC procedure is run as described previously in this section, except that instead of setting the α, β, and x parameters as random draws from the density of each conditional on the other two, they are set at the maximizer for the joint likelihood function given the values of Z_u^{user} and Z_i^{item} at that iteration. The mechanism for setting the Z_u^{user}, Z_i^{item}, π_{user}, and π_{item} at each iteration remains unchanged. Running this procedure for 30–40 iterations is typically sufficient to complete burn-in and have the parameters approach within three standard deviations of their posterior mean of the stable MCMC. This means that the entire burn-in procedure is completed in fewer than 10 minutes of CPU time. This is far less than the 81 hours it would take otherwise, when burn-in is implemented merely by running the MCMC algorithm for many iterations.

When the burn-in is completed, the MCMC procedure is allowed to run in its pure form, cycling through the seven blocks of parameters as described previously. However, because burn-in is already completed and the mode is near, approximate log-concavity for the x distribution can be assumed, and the adaptive rejection sampling framework of Gilks and Wild (1992) can be used, thus avoiding the Metropolis–Hastings algorithm altogether in the entire estimation process. The MCMC iterations were run for 1000 iterations following the prelude procedure. Estimates of the x vector for each item and the α and β vectors for each user are obtained by averaging the respective variates from their posterior distributions in the MCMC sequence.

Some Theory on the Prelude Procedure: The Stochastic-EM Algorithm

The prelude procedure is a variant of the EM algorithm in which the auxiliary data-augmentation variables (which are Z_u^{user} and Z_i^{item} in this case) are set not at their expected values, as the EM algorithm requires, but rather at random

[3] Sincere thanks go to an anonymous reviewer for pointing out the connection with principal components analysis and for referring to the work of Schein, Saul, and Ungar (2003), in which the connection is explicated.

draws from their conditional densities. This variant of the EM algorithm has been treated in the literature and is referred to as the stochastic EM algorithm (see Diebolt and Ip 1996; Ip 2002; Nielsen 2000). The stochastic EM algorithm was proposed for situations such as the current problem, in which the complete data log-likelihood is not a linear function of the sufficient statistics and the original EM algorithm becomes invalid. Three theoretical results pertaining to the stochastic EM are worth mentioning. First, the average of the stochastic EM converges to a stable quantity. Second, this stable quantity asymptotically approaches the maximum likelihood estimate. Third, the rate for this asymptotic approach is the same as the rate at which the Bayesian posterior mean approaches the maximum likelihood estimate. Together, these three results indicate that the burn-in acceleration prelude procedure by itself, without any further MCMC iterations, can produce good estimates of the model parameters. However, there are no simple ways of obtaining standard errors from the stochastic EM algorithm. To obtain confidence intervals and other measures of uncertainty, perhaps the most practical approach would be to follow the prelude procedure with proper MCMC iterations and to use those draws to estimate parameter uncertainty. Some other convergence issues are discussed in the Web Appendix (see http://www.marketingpower.com/jmrfeb08).

AN EMPIRICAL ILLUSTRATION

A moderately sized Internet firm based in the United States provided the data for the empirical illustration. The firm sells a variety of household articles, ranging from consumer electronics to school material to home tools and devices. Customers visiting the Web site are exposed to product recommendations in two distinct ways. First, the “home page” lists a recommendation based mainly on the sales volumes of the various items at that time. Second, on many occasions, when a user adds a product to the online shopping cart, a recommendation is made for a product related to the items in the shopping cart. Most of these recommendations are based on the usual market basket analysis used by several e-commerce firms, but many recommendations are picked randomly from a set of items the retailer wants to promote.[4]

I now attempt to map this scenario onto the model proposed in this article. There is no usable attribute information on any of the items in the Web site. The Web site gives verbose English language descriptions of each product; these descriptions do not readily lend themselves to codification for a computational statistics model. Therefore, although the computer analysis algorithms cannot make sense of the verbose product descriptions, the customer presumably can. In this sense, the situation matches the model’s position that the attribute vector x_i is observed by the user but not by the analyst. Furthermore, the reasonable assumption is made that the customer mentally records the attributes x_i and takes them into account in his or her preference functions.

Recall the basic conceptualization that a customer needs to go through two stages to make purchase: awareness (A) and satisfaction (S). The A and S probabilities for every item are each influenced by the product’s attribute vector x_i. How reasonable is this in the current setting? The effect of x_i on S is largely self-evident. The effect of x_i on A would probably occur because the Web site is loosely organized under various categories. Therefore, if a customer is interested in consumer electronics, he or she is likely to browse the consumer electronics section of the Web site and thus be more aware of these products.

The data set used here has 932 customers and 1681 products. Thus, the number of (u, i) pairs is 1,566,692. The total number of purchases is 60,790. This means that the U × I table is 96.12% sparse. The total number of recommendations presented was 73,996. This number gives the number of pairs in the set **R**, which was introduced in the discussion of the modeling framework. Consequently, the number of pairs in the set **N** is 1,566,692 – 73,999 = 1,492,693. Note that this implies that 1 – 73,999/1,566,692 = 95.28% of the possible (u, i) pairs never get recommended. This number is not atypical of e-commerce settings, even for a moderately sized firm such as the one in this study, for which the numbers of customers and products are not very large. With much larger numbers of customers and products, the sparsity can go up greatly. The number of items the firm offers can be so large and the number of recommendation opportunities can be so small that the probability of a random product being recommended to a random user is low. Therefore, it becomes even more important for the firm to do the right analytics to make the best use of these few opportunities. The total number of recommendations accepted is 6807. Thus, the average probability of a recommendation being accepted is 9.2%.

Estimation and Setting Model Size Parameters

Recall that the model has three exogenous parameters: the number of user latent classes C_{user}, the number of item latent classes C_{item}, and the dimensionality of the parameter space d. The estimation procedures described assume that these three parameters are externally specified. The “right” values were chosen by predictive posterior fit on an external data set. From the data set described previously, only 85% of the observations for model estimation were used. A random 5% subset of the observations was set aside for figuring out the best values of the exogenous parameters. (A second 10% subset was used for forecast accuracy assessment, as I describe in greater detail subsequently in this section.)

The predictive posterior fit is computed for the 5% external data subset in the following manner, which is common in Bayesian analysis (see Gelman et al. 2004): At each iteration of the MCMC, the negative log-likelihood (NLL) for the external data was evaluated using the values of the parameter set Φ (see Equation 17) at that iteration. Then, all iterations were averaged to obtain the mean value of the NLL. This mean value is referred to as the predictive posterior NLL, or PPNLL. In general, goodness-of-fit measures evaluated in-sample need to be adjusted for the number of parameters, as in the Akaike information criterion or Bayesian information criterion. However, the PPNLL is like a cross-validation fit in that more parameters do not automatically lead to better fit. The PPNLL can be interpreted “as is,” without needing to adjust for model size.

To evaluate a certain set of values of C_{user}, C_{item}, and d, the estimation procedure was run for a model of that size,

[4] For an introduction to market basket analysis, see Berry and Linoff (2004).

and then PPNLL was evaluated for the resultant estimates. The PPNLL was evaluated for several values of C_{user}, C_{item}, and d, and the setting that yielded the lowest value of PPNLL was selected. An exhaustive search was not done. Instead, a "greedy search" was employed, which worked as follows: Only one of the three exogenous variables was varied, and the best value, when the other two variables were held fixed, was identified. All three variables were cycled through until every variable was at its optimal setting, conditional on the other two variables. The conditional function was assumed to be convex, and a discrete variant of the golden section search method was used to pick trial values when doing this conditional optimization. Values of d from 1 to 15 were considered, and values of C_{user} and C_{item} from d to 100 were considered. (The number of latent classes must exceed d for identification.) It took approximately 250 evaluations of PPNLL to decide on the final values: C_{user} = 26, C_{item} = 33, and d = 7. The value of PPNLL for this model size was 6584.898. To get a sense of the sensitivity of the PPNLL to the model size parameters, see Table 1, which reports the values of the PPNLL when each of the three model size parameters is varied around the "best" values and the other model size parameters are held fixed.

Changes to d affect predictive fit the most. Between C_{user} and C_{item}, fit appears to be more sensitive to changes in C_{item} than to changes in C_{user}.

Understanding the X Space for "Cold-Starting" Recommendations on New Items

As pointed out previously during the discussion on identification issues, the parameters **A**, **B**, **X** are not interpretable by themselves, because they can be linearly transformed without affecting fit if $\mathbf{A}^T\mathbf{X}$ and $\mathbf{B}^T\mathbf{X}$ are unaltered. However, it may be worth trying to understand how the various items lie relative to one another in the x space. Under linear transformations of the x space, the coordinates themselves may change greatly, but in general, objects relatively close together remain so, even under the transformation. Understanding the relative locations of items in the x space may be useful in "cold-starting" recommendations on new items. The cold-starting problem in recommendation systems (see Schein et al. 2002) refers to the problem of making predictions for previously unobserved items. To use the model strictly, it is necessary to wait until enough purchase data have accumulated for the x parameter to be estimated for that item. In the current situation, estimating the x parameter for an item amounts to determining the class to which the item belongs. What might be a cold-start solution in this case if there are no purchase data on an item?

A qualitative solution would be to use subjective or objective characteristics to match the new item to the existing items in the data set for which x values have already been estimated. As a tentative x value for this new item, the x value of a similar item can be used. This suggested process of matching can be facilitated if there is a map of a representative set of products in the x space. Figure 1 shows such a map. For each latent class, a list was compiled of items most highly identified with that latent class in terms of posterior segment membership probabilities. Each list was then reduced to a small set of one to three "representative items" to give a sense of what kinds of items get put into that particular latent class. This list was then placed on a two-dimensional grid using just the first two coordinates of the x vector for that latent class. The prelude procedure is like principal components analysis in that it organizes the coordinates in decreasing order of explained fit. Therefore, the first two coordinates can be viewed as being the most important coordinates. (This ordering may get destroyed when the MCMC iterations start to operate on the solution from the prelude procedure, but in this specific application, the MCMC procedure does not alter the solution from the prelude procedure that much.) Table 2 gives the coordinates for each latent class. Figure 1 depicts the lists of representative items using the given coordinates. As an example of how the map can be used to determine approximate x coordinates for a new item, if the new item is a toy, it would likely be placed in the upper-right-hand corner.

Table 1
PPNLL VALUES WHEN VARYING THE MODEL SIZE PARAMETERS

C_{user}	*PPNLL*	C_{item}	*PPNLL*	*d*	*PPNLL*
24	6668.3	31	6685.4	5	7217.9
25	6628.8	32	6635.4	6	6859.1
26	6584.9	33	6584.9	7	6584.9
27	6599.6	34	6607.4	8	6646.2
28	6615.1	35	6625.9	9	6700.8

Benchmarks and Holdout Sample Performances

The main idea in this article is the distinction between the self-initiated purchase data set **N** and the firm-initiated purchase data set **R** and its consequent use in separately identifying the satisfaction and awareness parameters with a unary model. To evaluate the improvement this idea offers for predictions, a benchmark procedure should be created that is as close as possible to the main model described herein, except that the benchmark should not make a distinction between **N** and **R**. Instead, the benchmark should operate on the pooled data from **N** and **R**, and a binary model should be used. In the pooled data, purchase is denoted as y_{ui} = 1, and the lack of purchase is denoted as y_{ui} = 0. As discussed previously, the binary data model implicitly views all purchases as liking the product and all nonpurchases as disliking the product. In effect, the binary model interprets all responses the way the main model interprets the **R** data. Accordingly, in the benchmark procedure, y_{ui} is modeled as in Equation 7, but the user's coefficient vector is denoted as γ_u to avoid confusion with the β_u from the main model. So, for the benchmark model,

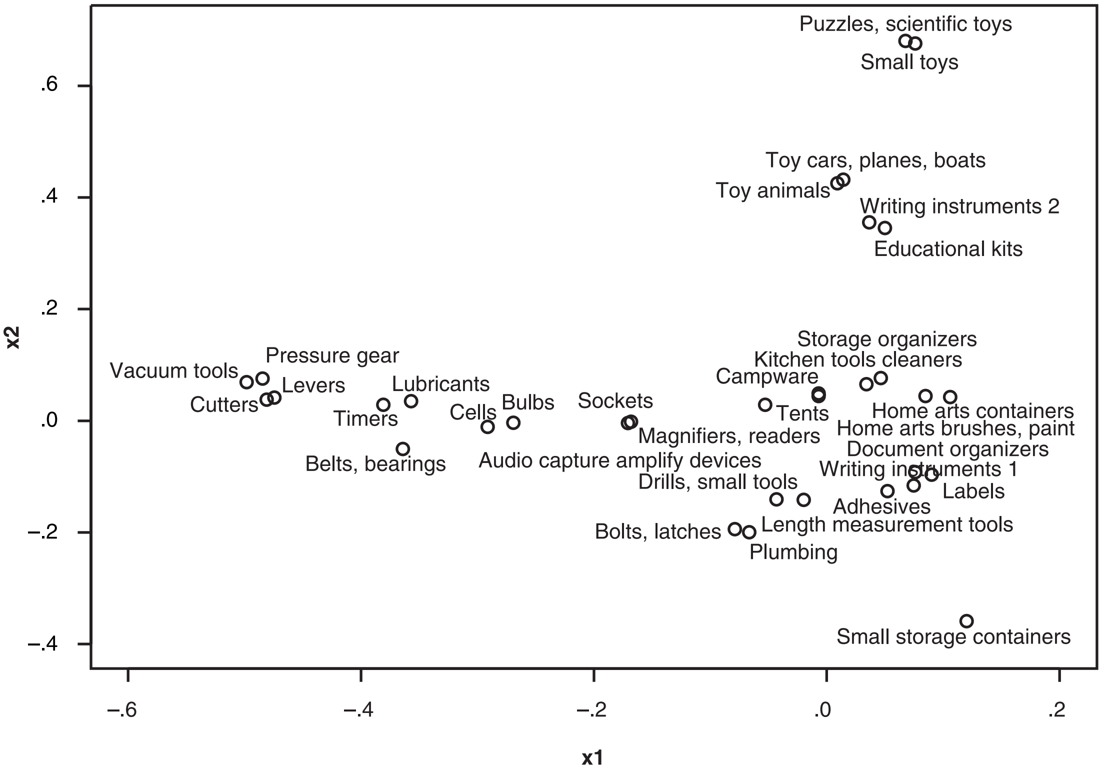
$$(18) \qquad p(y_{ui} = 1) = \text{logistic}(\gamma_u^T x'_i), \text{ and}$$
$$p(y_{ui} = 0) = 1 - \text{logistic}(\gamma_u^T x'_i).$$

The x′ and γ parameters come from a dual latent class model, which is estimated using a preamble procedure and MCMC iterations similar to that of the main model. This model is referred to as the "binary benchmark." Note that item i's characteristics vector is denoted by x_i' (i.e., with a prime) to indicate that the vector obtained from the binary benchmark may differ from the vector x_i obtained from the main model.

Another benchmark is the procedure that the typical commercial implementation of recommendation systems would apply for purchase data—that is, nearest-neighbor

Figure 1

A MAP SHOWING THE LIST OF REPRESENTATIVE ITEMS FOR EACH LATENT CLASS OF ITEMS



collaborative filtering used in its binary data version rather than the more dominant ratings data version. Several metrics can be employed to measure between-user similarity in binary data; here, the Tanimoto similarity index is employed, as in Mild and Reutterer (2003). This benchmark is referred to as the "CF benchmark."

The holdout data are a 10% stratified sample of the full data set. No new users or items are contained in this data set. This means that there are no cold-start problems to address. In the **N** part of the holdout, there are 149,269 observations, which resulted in 5398 purchases. In the **R** part, there are 7400 observations, which resulted in 681 purchases. The results of a decile analysis on these data are not shown. For the main model, each observation in **N** and **R** is scored using Equations 3 and 7, respectively. For the binary benchmark, observations in both **N** and **R** are scored using Equation 18. For the CF benchmark, observations in both **N** and **R** are scored using purchase probability estimates based on behaviors of similar users. For each model, for each of **N** and **R**, observations are sorted in decreasing order of score and grouped into deciles; then, the number of purchases in each decile are counted. If the score is predictive of purchasing and capable of separating observations with purchases and observations with no purchases, the distribution of the purchases should be concentrated in the upper deciles. The higher the concentration, the better the score is at separating.

The decile analysis results appear in Table 3. It should not be surprising that the main model outperforms the benchmark models because it is the only one of the three that makes a distinction between the stochastic processes governing **R** and **N**. Therefore, it is considerably better equipped to make predictions in the two situations. Conversely, the benchmark models try to come up with a single scoring mechanism that works for the pooled data from both data sets. What may be surprising, however, is the magnitude of differences. In general, managerial interest centers on the highest deciles because therein lie the items that managers consider recommending. An examination of just the top decile in the **R** data shows that the main model performs 99% better than the binary benchmark and 123% better than the CF benchmark. The difference for the **N** is less impressive but still sizable. Between the two benchmarks, the CF benchmark is somewhat comparable to the binary benchmark for the **R** data set but inferior for the **N** data set. Thus, the CF benchmark was dropped from further analysis.

Some confusion matrices for predictions from the main model are now shown. If the purchase probability exceeds a threshold number, a 1 is predicted. The threshold is .455 for **N** and .485 for **R**. These thresholds were picked because they minimized overall error (false positives plus false negatives) in the 5% external data set discussed previously for picking the model size parameters.[5] Because firms may consider making recommendations only if the recommendation acceptance probability is fairly high—for example,

[5] An anonymous reviewer suggested this approach for picking the threshold.

Table 2

REPRESENTATIVE ITEMS FOR EACH LATENT CLASS OF ITEMS

Class	x_1	x_2	*Representative Items for Class*
1	.04	.36	Writing instruments
2	–.01	.05	Tents
3	.07	.68	Puzzles, scientific toys
4	–.38	.03	Timers
5	–.36	.04	Lubricants
6	–.01	.04	Campware
7	–.08	–.19	Bolts, latches
8	.01	.43	Toy animals
9	.03	.07	Kitchen tools cleaners
10	.05	.08	Storage organizers
11	.08	.68	Small toys
12	.05	–.13	Adhesives
13	–.17	.00	Sockets
14	–.07	–.20	Plumbing
15	–.48	.08	Pressure gear
16	.11	.04	Home arts brushes, paint
17	–.27	.00	Bulbs
18	–.36	–.05	Belts, bearings
19	–.04	–.14	Drills, small tools
20	–.29	–.01	Cells
21	.08	–.09	Document organizers
22	–.05	.03	Magnifiers, readers
23	–.17	.00	Audio capture amplify devices
24	–.02	–.14	Length measurement tools
25	.07	–.12	Labels
26	–.48	.04	Levers
27	.08	.04	Home arts containers
28	–.5	.07	Vacuum tools
29	–.47	.04	Cutters
30	.05	.34	Educational kits
31	.01	.43	Toy cars, planes, boats
32	.12	–.36	Small storage containers
33	.09	–.10	Clips, tape

Table 3

DECILE ANALYSIS ON HOLDOUT DATA

	Main Model		*Binary Benchmark*		*CF Benchmark*	
	***N** Data*	***R** Data*	***N** Data*	***R** Data*	***N** Data*	***R** Data*
Best decile	4472	402	3779	202	3157	180
2	637	172	919	110	1238	123
3	209	57	468	108	495	89
4	50	28	140	49	265	85
5	14	8	57	64	148	36
6	11	7	27	64	67	52
7	5	5	7	40	21	40
8	0	2	1	20	7	49
9	0	0	0	19	0	18
Worst decile	0	0	0	5	0	9
Total purchases	5398	681	5398	681	5398	681

Notes: These results show that the main model is the best for both data sets. The purchase counts are concentrated in the top deciles more for the main model than for the benchmarks. This implies that the main model's scoring is better at discriminating between purchases and nonpurchases.

75%—the confusion matrix for **R** with a threshold of .75 is also shown. Table 4 reports the confusion matrices.

Of particular interest from the second matrix is the ratio 323/(131 + 323) = 71%, which gives the accuracy rate when recommendation acceptance is forecast using the 48.5% threshold. With a 75% threshold, the accuracy rate is 171/(36 + 171) = 82.6%. These accuracy figures are good, considering that the base rate for recommendation acceptance is 9.2%. The accuracy rate of purchase predictions in the **N** data set is 1809/(885 + 1809) = 67%, which again is good relative to the base rate of 3.6%.

Table 4

CONFUSION MATRICES COMPARING PREDICTED BEHAVIOR WITH ACTUAL BEHAVIOR

*A: Confusion Matrix for **N** Data with .455 Threshold Comparing Prediction of Self-Selected Purchases with Actual Purchases*

	Actually Not Purchased	*Actually Purchased*
Predicted to not purchase	142,986	3589
Predicted to purchase	885	1809

*B: Confusion Matrix for **R** Data with .485 Threshold Comparing Prediction of Recommendation Acceptance with Actual Acceptance*

	Actually Rejected	*Actually Accepted*
Predicted to reject	6588	358
Predicted to accept	131	323

*C: Confusion Matrix for **R** Data with .75 Threshold Comparing Prediction of Recommendation Acceptance with Actual Acceptance*

	Actually Rejected	*Actually Accepted*
Predicted to reject	6683	510
Predicted to accept	36	171

Differences in Recommendation Lists Between the Main Model and the Binary Benchmark Model

Recall from the previous discussion that the main model implies distinct recommendation-picking criteria under the instantaneous policy and the broad-range policy. The instantaneous policy is considered first.

Comparison under the instantaneous policy. In the instantaneous policy scenario of the main model, the expected incremental number of purchases generated from recommending item i is $p(S_{ui}|A_{ui})$ or $\text{logistic}(\beta_u^T x_i)$ (see Equations 14 and 7). Therefore, when the recommendation list under the main model is constructed for user u, items with the highest values of $\text{logistic}(\beta_u^T x_i)$ are picked. In contrast, for the recommendation list under the binary benchmark model, items with the highest values of $\text{logistic}(\gamma_u^T x'_i)$ are picked (see Equation 18). How different are the choices from these two criteria? The correlation between the two scores for the typical user is approximately .65. Because this correlation is so high, it may seem that the two criteria result in similar recommendation lists. From the firm's point of view, the similarity just near the top end of the lists is of interest because this is the area from which the firm will draw its recommendations. To study this, just the top k items in each of the two lists are considered, and the items that appear in both the length-k lists are counted. Values of k from 5 to 60 are examined, and the items that overlap between the two lists are computed and averaged across users. This overlap number appears in Table 5. The overlapping fraction is only approximately 12% at k = 60 and considerably smaller for lower k.

The recommendation list from the main model is chosen to maximize the expected incremental number of purchases generated from recommendations. Although most of the

Table 5

OVERLAP IN LISTS AND PERFORMANCE COMPARISON UNDER THE INSTANTANEOUS POLICY SCENARIO FOR THE k BEST RECOMMENDATIONS FOR VARIOUS VALUES OF k

k	*Overlap Number*	*Δ from Main Model Recommendations*	*Δ from Benchmark Recommendations*
5	.29	4.45	1.67
10	.60	8.81	3.19
20	2.03	17.13	6.01
30	4.38	24.80	8.54
40	6.99	32.06	10.93
50	9.00	38.84	13.20
60	11.30	45.15	15.32

Notes: Numbers reported are across-user averages.

items in the binary benchmark's list are different, and thus suboptimal, it may be that the extent of suboptimality is not much. Therefore, the natural question is, What is the incremental purchase volume generated when the firm uses one recommendation list versus the other? The incremental purchase volume generated from user u from a recommendation list as a whole is the sum of logistic($\beta_u^T x_i$) over all items i on the list. The incremental purchase volume generated by recommending the top k items from the main model's score was computed in this way. Similarly, the incremental purchase volume figures for the binary benchmark model were computed (see Table 5).

Because the main model picks the items with the highest values of purchase increment logistic($\beta_u^T x_i$), the incremental purchase volume from the main model's recommendation list is, by construction, greater than the incremental purchase volume from the benchmark list for each value of k. Therefore, it is a foregone conclusion that in Table 5, the numbers in the last column will be smaller than the numbers in the second-to-last column. What is of interest here is the magnitude of difference: The incremental sales generated from the main model are greater by a factor of 2.8, which is substantial.

The average value of logistic($\beta_u^T x_i$) across all items is approximately .08. This means that if, instead of choosing recommendation items according to the main model's score or the benchmark's score, k items are picked randomly with equal probability, the incremental sales generated will be approximately .08 × k. At k = 50, this is 4, compared with 38 from the main model and 13 from the binary benchmark. This means that the binary benchmark's list is not bad, because it is superior to picking items randomly. However, it is possible to do substantially better by using the main model's list.

Comparison under the broad-range policy. As discussed previously, in the broad-range policy scenario of the main model, the expected incremental number of purchases generated from recommending item i is

$$\left[p(S_{ui}|A_{ui}) - \frac{T_f}{T_c}p(A_{ui}, S_{ui})\right], \text{ or}$$

$$\text{logistic}(\beta_u^T x_i)\left[1 - \frac{T_f}{T_c}\text{logistic}(\alpha_u^T x_i)\right].$$

See Equations 11 and 13. Accordingly, for the main model's recommendation list for user u, the items that had the highest values for the preceding expression were picked. Calibration data were collected over a six-month interval. If a three-month planning horizon for the broad-range policy is assumed, the ratio (T_f/T_c) to use would be 1/2.

A similar analysis to that of the instantaneous policy scenario was conducted. Again, the overlap between the main model's list and the benchmark model's list was examined, and the incremental purchase volume for each of the two lists was computed at various levels of k. The resultant numbers appear in Table 6. The incremental sales generated from the main model are higher on average by a factor of 3.8, which is an even greater improvement than that observed for the instantaneous policy. This is because the mismatch between the two lists is higher. Because the main model's list is optimal for this scenario by construction, greater mismatches imply greater departures from optimality. The correlation between the two scores is approximately 48%, which is substantially lower than the 65% in the instantaneous policy scenario (see Table 6).

A cautionary note. When interpreting Tables 5 and 6, it is critical to remember the following: The forecasts for incremental sales volume are based directly on the theoretical expressions and are being presented only to give an approximate sense of the magnitude of difference between the main model and the binary benchmark. No data are available to validate these projections for incremental sales volume directly. The best validation would be through a field experiment. To test under the broad-range policy, three groups of customers over duration T_f would need to be studied and the following would be measured: (1) sales for a customer group that gets recommended nothing, (2) sales for a group that gets recommended by the main model, and (3) sales for a group that gets recommended by the benchmark model. The results from Group 2 – Group 1 would serve to validate the figures in the second-to-last column of Table 6, and the results from Group 3 – Group 1 would serve to validate the figures in the last column. The field experiment results could depart from the theoretical results as a result of misspecification issues or sampling issues.

DISCUSSION AND FURTHER WORK

The chief contributions of this article are as follows: It (1) suggests the idea that purchasing can be viewed as the outcome of awareness and satisfaction, (2) points out that the two events can be separated out and identified and estimated by combining two kinds of data sets that firms have

Table 6

OVERLAP IN LISTS AND PERFORMANCE COMPARISON UNDER THE BROAD-RANGE POLICY SCENARIO FOR THE k BEST RECOMMENDATIONS FOR VARIOUS VALUES OF k

k	*Overlap Number*	*Δ from Main Model Recommendations*	*Δ from Benchmark Recommendations*
5	.01	4.16	1.05
10	.07	7.88	2.01
20	.51	14.86	3.89
30	1.34	21.28	5.65
40	2.71	27.11	7.35
50	4.41	32.54	8.99
60	7.39	37.47	10.54

Notes: Numbers reported are across-user averages.

access to—namely, self-initiated purchase data and recommendation response data—and (3) proposes a decision framework for recommendations that makes use of the distinction between awareness and satisfaction. The model can be viewed as the simplest model that rides on these three ideas. However, much more work is needed because there are many enhancements to be carried out to reflect the many characteristics of business situations in e-commerce.

This article concludes with a discussion of two deficiencies in the modeling framework, which further research might try to remedy. The deficiencies are related to the discrete nature of the awareness event and the satisfaction event. The model assumes that the firm's making a recommendation for an item results in creating an awareness of that item—specifically, an awareness of sufficient intensity that the user becomes knowledgeable about the attribute vector x. For the empirical application presented herein, this assumption is probably a reasonable one. To understand why, a little more needs to be said about how the particular firm studied herein deployed recommendations. For example, this particular firm makes relatively few recommendations. When it makes a recommendation for a product, it presents only one recommendation, and this recommendation is displayed prominently in mid-center of the top screen of the Web page with a picture and full description of the product. With the firm under study, it is more certain than usual that the recommendation is actually processed by the consumer, in the sense that the recommendation probably creates awareness, as is assumed in the development of Equation 7. However, not all e-commerce Web sites operate this way. Some Web sites clutter the screen by offering four or five recommendations at any point. In some instances, the recommendations appear near the bottom of the Web page, which the user may not actually see. In such situations, it is unreasonable to assume that the firm's making a recommendation is equivalent to the consumer developing full awareness of x_i, as demanded by the model. It may make more sense to view the user as having developed partial awareness and knowledge about x_i. In other words, there needs to be a model that allows A_{ui} to take a continuum of realizations, not just a value of 0 or 1, depending on what the data indicate about the nature of the exposure to the recommendation.

Now, consider the discrete nature of the satisfaction event. In the current model, this event is true or false, depending on whether (conditional on awareness) there is a purchase or nonpurchase. The discreteness of purchase versus nonpurchase is limiting. There are intermediate states that are also informative of satisfaction—for example, adding an item to the shopping cart but not actually buying the item or repeatedly visiting the product Web page for a certain item. A model examining such intermediate states of satisfaction will be able to process more data and thus possibly outperform the main model described herein. It is hoped that other researchers will pursue these extensions.

APPENDIX A: FORECASTING PURCHASE PROBABILITIES FOR AN INTERVAL WITH DURATION DIFFERENT FROM THE CALIBRATION DURATION

Let $p(A^c_{ui})$ be the probability of awareness occuring in a period of length T_c. To derive a proper expression for awareness probabilities for an interval different from that in the calibration sample, it is necessary to posit a model of how awareness probability is influenced by the time interval. Specifically, it is necessary to allow that greater passage of time leads to greater awareness probability. Let T_c be the time interval length of the calibration data and T_f be the time interval length for the forecast. Let $p(A^c_{ui})$ be the probability of awareness occuring in a period of length T_c. Similarly, let $p(A^f_{ui})$ be the probability of awareness occurring in a period of length T_f. To derive an expression for $p(A^f_{ui})$, it is proposed that awareness follows a memoryless exponential process, so that for a given period T and arrival rate λ_{ui}, the probability of awareness happening between time 0 and T is $1 - \exp(-\lambda_{ui}T)$. This implies the following:

$$p(A^c_{ui}) = 1 - \exp(-\lambda_{ui}T_c)$$

$$p(A^f_{ui}) = 1 - \exp(-\lambda_{ui}T_f).$$

Solving for λ_{ui} from the first equation and substituting into the second and simplifying gives the following:

$$\text{(A1)} \qquad p(A^f_{ui}) = 1 - \left[1 - p(A^c_{ui})\right]^{(T_f/T_c)}.$$

This expression is invariant with respect to the scale of time measurement. It does not matter whether T_f and T_c are measured in months or days or hours. Only the ratio is relevant. Applying the binomial theorem to first order in Equation A1 gives

$$\text{(A2)} \qquad p(A^f_{ui}) \approx \frac{T_f}{T_c} p(A^c_{ui}),$$

and applying it to second order gives

$$\text{(A3)} \qquad p(A^f_{ui}) \approx \frac{T_f}{T_c} p(A^c_{ui}) - \frac{1}{2}\left(\frac{T_f}{T_c}\right)\left(\frac{T_f}{T_c} - 1\right)\left[p(A^c_{ui})\right]^2.$$

The first-order approximation corresponds to the intuition that the probabilities should simply be scaled up or down according to the ratio of the time interval lengths. As an example, suppose that T_c is three months and $p(A^c_{ui})$ is .03. When T_f = one month, the expression for $p(A^f_{ui})$ from Equation A1 gives .0101017, which is close to the number .01 produced from the first-order approximation. For the range of values of $p(A^c_{ui})$ obtained in the empirical illustration, the first-order approximation seems sufficiently accurate.

The self-initiated purchase probability is $p(A) \times p(S|A)$. If the foregoing approximation is used, the probability of a self-initiated purchase in T_f can be taken as follows:

$$p(A^f_{ui}, S_{ui}) = p(S_{ui}|A^f_{ui}) \times p(A^f_{ui})$$

$$\approx \frac{T_f}{T_c} p(S_{ui}|A_{ui}) p(A_{ui}).$$

APPENDIX B: MODE FINDING IN THE JOINT SPACE OF α, β, AND X

Mode finding is conditional on knowing the classes for all the items and all users. Because all items within an item class are considered to have the same x value and because all users within a user class are considered to have the same α value and the same β value, all user–item pairs that fall within a user-class–item-class pair can simply be aggregated. Specifically, the quantities y_{ui}, n_{ui}, v_{ui}, and r_{ui}, defined at the level of user–item pairs, are aggregated into the quantities Y_{st}, N_{st}, V_{st}, and R_{st}, defined at the level of user-class–item-class pairs. For example, the aggregate quantity Y_{st} for the user-class–item-class pair, where user class is s and item class is t—is given by the sum of y_{ui} over all user–item pairs (u, i), where user u belongs to class s and item i belongs to class t. The quantities $\{Y_{st}\}$ are arranged into a matrix **Y** so that Y_{st} is the element in the sth row and tth column of **Y**. Similarly, the other quantities are arranged into matrices **N**, **V**, and **R**. In addition, recall that the α, β, and x parameters are organized into matrices **A**, **B**, and **X**. From Equation 2, V_{st} is binomial with success probability logistic($\beta_s^T x_t$) and number of trials R_{st}. Similarly, from Equation 3, Y_{st} is binomial with success probability logistic($\alpha_s^T x_t$) × logistic($\beta_u^T x_i$) and number of trials N_{st}. Therefore, **V** is binomial with success probability logistic($\mathbf{B}^T\mathbf{X}$) and number of trials **R**, and **Y** is binomial with success probability logistic($\mathbf{A}^T\mathbf{X}$) ⊗ logistic($\mathbf{B}^T\mathbf{X}$) and number of trials **N**. The ⊗ operator denotes matrix multiplication that is carried out elementwise. For identification, the following constraints are imposed: **X** is orthonormal, **B** is orthogonal, and **A** is unconstrained.

Consider the following model, which is referred to as the "rank-reduced binomial regression" model: **E** is binomial with success probability logistic($\mathbf{F}^T\mathbf{G}$) and number of successes **H**. Here, **E**, **F**, **G**, and **H** are all matrices. The logistic operation is carried out elementwise as before, and **F** and **G** are conformable matrices of rank lower than that of **E**. The problem is to estimate **F** and **G**, given **E** and **H**. This problem can be expressed as a sequence of something called "weighted Frobenius norm minimizations," which in turn can be expressed as a sequence of eigenvalue–eigenvector calculations. Schein, Saul, and Ungar (2003) and Srebro and Jaakkola (2003) offer solutions for the case in which **E** is Bernoulli rather than binomial. Here, their solution is extended to the Binomial case in the Web Appendix (see http://www.marketingpower.com/jmrfeb08). The implication of this is that the mode-finding problem reduces to a sequence of eigenvalue–eigenvector calculations.

To return to the mode-finding problem, finding the maximum likelihood estimates for **A**, **B**, and **X** can be expressed as a sequence of rank-reduced binomial regressions. Estimation is done with the EM algorithm. Each M step is a rank-reduced binomial regression. This is now explained as follows.

The **V** is decomposed to be a result of two binomial processes, each with success probability given by a single logistic expression. To construct this decomposition, the following result is used: If random variable Y is binomial with success probability A × S and number of trials N, then Y can be expressed as an outcome of two binomial processes with success probabilities A and S as follows: First, random variable n is drawn from the binomial process with number of trials N and success probability A. Second, Y is drawn from the binomial process with number of trials n and success probability S. Third, the distribution of n conditional on Y is

$$\text{(B1)}\quad \mathrm{p}(n|Y, N, A, S) = \frac{(1-A)^{Y-n}A^{n-Y}S^{n-Y}\left(\frac{1-AS}{1-A}\right)^{Y-N}(N-Y)!}{(n-Y)!(N-n)!},$$

which implies that the conditional expectation of n is

$$\text{(B2)}\qquad \mathrm{E}(n|Y, N, A, S) = \frac{(1-A)Y + AN(1-S)}{1-AS}.$$

Proofs of these results are available on request. The immediate implication of this result is that if there is a binomial process with success probability given by logistic(αx) ⊗ logistic(βx), this can be expressed as two binomial processes with success probabilities given separately by logistic(αx) and logistic(βx). Recall that the **Y** counts are considered a consequence of an awareness step (A) and a satisfaction step (S). Therefore, the result just described has the following appealing interpretation: It helps decompose **Y** into counts attributable separately to the awareness process and the satisfaction process. In this, notation was chosen for the various probabilities and counts to emphasize this interpretation: A can be viewed as the awareness probability given by logistic(αx), and S can be viewed as the satisfaction probability given by logistic(βx) (see Equations 1 and 2). The Y and N in this result can be viewed as corresponding to the **Y** and **N** used in the discussion of Step M.

A consequence of the preceding discussion is that it is possible to decompose the **Y** binomial counts from **N** trials and success probability logistic(αx) ⊗ logistic(βx) as the following two processes in expectation:

1. **n** successes from **N** trials and success probability logistic(αx), with the expectation E(**n**) given by

$$\text{(B3)}\qquad \mathbf{A} = \text{logistic}(\alpha\mathbf{x}),$$

$$\mathbf{S} = \text{logistic}(\beta\mathbf{x}), \text{ and}$$

$$\mathrm{E}(\mathbf{n}) = \frac{(1-\mathbf{A})\mathbf{Y} + \mathbf{AN}(1-\mathbf{S})}{1-\mathbf{AS}}.$$

2. In the last of these three equations, all the operations are performed elementwise on the matrices.
3. **Y** successes from **n** trials and success probability logistic(βx). The EM algorithm requires the expectation, not the actual realization, of the sufficient statistic corresponding to the auxiliary data, which are **n** in this situation.

Apart from the **Y** counts decomposed as shown, there are also **V** counts. Recall that the **V** counts are binomial with **R** trials and success probability logistic(βx). Thus, the Step M data as a whole can be expressed as follows:

1. **n** successes from **N** trials and success probability logistic(αx), and
2. **Y** + **V** successes from **n** + **R** trials and success probability logistic(βx).

Now this collection of data falls under the framework of rank-reduced binomial regression. To see the connection explicitly, it is necessary to stack the data matrices from the two binomial processes. Denote by {**C**, **D**} the matrix obtained by stacking a matrix **C** above another matrix **D** with the same number of columns. The stacked matrix {**C**, **D**} has as many columns as each of **C** and **D** and as many rows as **C** and **D** combined. Consider the matrix {**n**, **Y** + **V**}, whose top half consists of the counts from the first binomial process given previously and whose bottom half consists of the counts from the second binomial process. As a whole, this matrix comes from a binomial process whose success probability matrix is logistic ({α, β}**x**) and whose matrix for number of trials is {**N**, **n** + **R**}. This shows that the singular value decomposition factorizations in the reduced-rank binomial regressions directly produce the maximum likelihood estimates of {α, β} and **x**.

Note that this is an iterative process. To obtain the estimates of {α, β} and **x**, estimates of **n** are required, which are shown to depend on previous estimates of α and β. This iteration sequence converges rapidly, so that the computation burden is not high. A small numerical example illustrating the mode-finding procedure at work is given in the Web Appendix (http://www.marketingpower.com/jmr feb08).

REFERENCES

Ansari, Asim, Skander Essegaier, and Rajeev Kohli (2000), "Internet Recommendation Systems," *Journal of Marketing Research*, 37 (August), 363–75.

——— and Carl F. Mela (2003), "E-Customization," *Journal of Marketing Research*, 40 (May), 131–45.

Bartholomew, D.J. and M. Knott (1999), *Latent Variable Models and Factor Analysis*, 2d ed. Oxford: Oxford University Press.

Berry, M.J.A. and G. Linoff (2004), *Data Mining Techniques*, 2d ed. Indianapolis: John Wiley & Sons.

Blattberg, R.C., G. Getz, and Jacquelyn Thomas (2001), *Customer Equity*. Cambridge, MA: Harvard University Press.

Breese, J.S., D. Heckerman, and C. Kadie (1998), "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, G.F. Cooper and S. Moral, eds. San Francisco: Morgan Kaufmann, 43–52.

Carlin, B.P. and T.A. Louis (2000), *Bayes and Empirical Bayes Methods for Data Analysis*, 2d ed. Boca Raton, FL: Chapman & Hall/CRC.

Chien, Y.-H. and Edward George (1999), "A Bayesian Model For Collaborative Filtering," working paper, Department of Statistics, The Wharton School, University of Pennsylvania.

Diebolt, J. and E.H.S. Ip (1996), "Stochastic EM: Method and Application," in *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, eds. New York: Chapman and Hall, 259–73.

Gelman, A., J.B. Carlin, H.S. Stern, and D.B. Rubin (2004), *Bayesian Data Analysis*, 2d ed. New York: Chapman & Hall/CRC.

Gilks, W.R. and P. Wild (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41 (2), 337–48.

Herlocker, J.L. (2000), "Understanding and Improving Automated Collaborative Filtering Systems," doctoral dissertation, Department of Computer Science and Engineering, University of Minnesota.

———, J.A. Konstan, and J. Riedl (2002), "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," *Information Retrieval*, 5, 287–310.

Iacobucci, D., P. Arabie, and A.V. Bodapati (2000), "Recommendation Agents on the Internet," *Journal of Interactive Marketing*, 14 (3), 2–11.

Ip, E. (2002), "On Single Versus Multiple Imputation for a Class of Stochastic Algorithms Estimating Maximum Likelihood," *Computational Statistics*, 17 (4), 517–24.

Kamakura, W.A., B.S. Kossar, and M. Wedel (2004), "Identifying Innovators for the Cross-Selling of New Products," *Management Science*, 50 (8), 1120–33.

———, M. Wedel, F. de Rosa, and J.A. Mazzon (2003), "Cross-Selling Through Database Marketing: A Mixed Data Factor Analyzer for Data Augmentation and Prediction," *International Journal of Research in Marketing*, 20 (1), 45–65.

Knott, A., A. Hayes, and S.A. Neslin (2002), "Next-Product-to-Buy Models for Cross-Selling Applications," *Journal of Interactive Marketing*, 16 (3), 59–75.

Li, Shibo, Baohong Sun, and Ronald T. Wilcox (2005) "Cross-Selling Sequentially Ordered Products: An Application to Consumer Banking," *Journal of Marketing Research*, 42 (May), 233–39.

Linden, G., B. Smith, and J. York (2003), "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, 7 (1), 76–80.

Loken, E. (2005), "Identification Constraints and Inference in Factor Models," *Structural Equation Modeling*, 12 (2), 232–44.

Mild, A. and T. Reutterer (2003), "An Improved Collaborative Filtering Approach for Predicting Cross-Category Purchases Based on Binary Market Basket Data," *Journal of Retailing and Consumer Services*, 10 (3), 123–33.

Moon, S. and G.J. Russell (2005), "Predicting Product Purchase from Inferred Customer Similarity: An Autologistic Model Approach," working paper, Tippie College of Business, University of Iowa.

Nielsen, S.F. (2000), "The Stochastic EM Algorithm: Estimation and Asymptotic Results," *Bernoulli*, 6 (3), 457–89.

Rossi, P., G. Allenby, and R. McCulloch (2006), *Bayesian Statistics and Marketing*. Hoboken, NJ: John Wiley & Sons.

Rust, R.T., V.A. Zeithaml, and K.N. Lemon (2000), *Driving Customer Equity*. New York: The Free Press.

Schein, A., A. Popescul, L. Ungar, and D. Pennock (2002), "Methods and Metrics for Cold-Start Recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Micheline Beaulieu, ed. New York: Association for Computing Machinery, 253–60.

———, L. Saul, and L. Ungar (2003), "A Generalized Linear Model for Principal Component Analysis of Binary Data," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Christopher M. Bishop and Brendan J. Frey, eds., (accessed October 29, 2007), [available at http://research.microsoft.com/conferences/aistats2003/proceedings/index.htm].

Srebro, N. and T. Jaakkola (2003), "Weighted Low-Rank Approximations," working paper, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Ungar, L. and D. Foster (1998), "Clustering Methods for Collaborative Filtering," working paper, Department of Computer and Information Science, University of Pennsylvania.

Weiss, S.M. and N. Indurkhya (2001), "Lightweight Collaborative Filtering Method for Binary-Encoded Data," in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery in Freiburg*, Luc de Raedt and Arno Siebes, eds. New York: Springer, 484–91.

Ying, Yuangping, Fred Feinberg, and Michel Wedel (2006), "Leveraging Missing Ratings to Improve Online Recommendation Systems," *Journal of Marketing Research*, 43 (August), 355–65.