Personalization of Supermarket Product Recommendations

R. D. Lawrence, G. S. Almasi, V. Kotlyar, M. S. Viveros, and S. S. Duri
IBM Research Division
Thomas J. Watson Research Center
PO Box 218
Yorktown Heights, NY 10598
{*ricklawr, almasi, vkotlyar, viveros, sastry*}*@us.ibm.com*

December 16, 2002

Abstract. We describe a personalized recommender system designed to suggest new products to supermarket shoppers. The recommender functions in a pervasive computing environment, namely, a remote shopping system in which supermarket customers use Personal Digital Assistants (PDAs) to compose and transmit their orders to the store, which assembles them for subsequent pickup. The recommender is meant to provide an alternative source of new ideas for customers who now visit the store less frequently. Recommendations are generated by matching products to customers based on the expected appeal of the product and the previous spending of the customer. Associations mining in the product domain is used to determine relationships among product classes for use in characterizing the appeal of individual products. Clustering in the customer domain is used to identify groups of shoppers with similar spending histories. Cluster-specific lists of popular products are then used as input to the matching process.

The recommender is currently being used in a pilot program with several hundred customers. Analysis of results to date have shown a 1.8% boost in program revenue as a result of purchases made directly from the list of recommended products. A substantial fraction of the accepted recommendations are from product classes new to the customer, indicating a degree of willingness to expand beyond present purchase patterns in response to reasonable suggestions.

Keywords: recommender systems, personalization, collaborative filtering, data mining, clustering, associations, pervasive computing.

1. Introduction

We describe a personalized recommender system designed to suggest new products to supermarket shoppers based upon their previous purchase behavior. The recommender system has been implemented as part of the "SmartPad" remote shopping system (Kotlyar et al., 1999) developed by IBM and Safeway Stores plc, a major supermarket retailer in the UK. This remote shopping system allows customers to prepare their shopping lists on a personal digital assistant (PDA) device such as a PalmPilot and transmit their order for subsequent pickup at the store without having to walk the aisles of the store. Although this latter feature is viewed as a convenience by a number of shoppers, it does remove the opportunity to suggest new or previously unpurchased products via special displays in the store and so forth. The personalized recommendation system was developed as a substitute "spontaneous purchase" mechanism for this remote shopping system in a weakly-connected "pervasive computing" environment. The recommendations are computed on the server, and delivered to an individual customer's PDA; obviously, the recommendations could also be delivered via more conventional mechanisms such as a web browser, electronic mail, or postal mail.

A number of web-based personalized recommender systems have been proposed recently (Resnick and Varian, 1997; Shardanand and Maes, 1995; Konstan et al., 1997; Borchers et al., 1998; Aggarwal et al., 1999; Personalization Summit, 1999). Personalization works by *filtering* a candidate set of *items* (such as products or web pages) through some representation of a *personal profile.* Two main paradigms for the filtering have emerged: *content-based* and *collaborative.*

A *content-based filtering* system recommends items based on their similarity to what a given person has liked in the past. Typically, both items and profiles are represented as vectors in the space of *features* and their similarity is computed via a standard distance metric, such as cosine coefficient. This approach has its roots in the vector-based model of Information Retrieval (IR), where text documents and user queries (or preferences) are both represented as vectors in the space of keywords or phrases, often referred to as *terms.* The coordinates of the vectors depend on the discriminating value of the respective terms. For example, if we were to recommend web pages from a site that reports news in the computer industry, then the term "computer" has low discriminating value, since it likely occurs in most of the pages. The term "product recommender" is likely to have high discriminating value, since we expect only a subset of pages to refer to product recommenders. A standard metric, called TFIDF ("term frequency / inverse document frequency") is used in the IR literature to quantify discriminating value of document features (see (Salton, 1989; Salton and McGill, 1983) textbooks for details).

In order to use content-based filtering in recommending grocery products, we must define the space of features. We started with product taxonomy that was available in the Safeway database. As discussed in Section 3.2 below, the taxonomy divides products in coarse-grain classes, such as "Pet foods", and, further, into sub-classes, such as "Canned Dog Food". We can directly use the classes and sub-classes as features of products and personal profiles. A person indicates interest in a particular feature by buying products within the corresponding class or sub-class. Products to be recommended can then be determined by computing a measure of distance between vectors representing personal preferences and vectors representing products.

The above strategy suffers from the problem of *overspecialization* ((Balabanovic and Shoham, 1997)): it provides us with no rigorous basis for introducing shoppers to new kinds of products beyond those classes and sub-classes that they already buy. For example, a person who buys dog food might also be interested in carpet cleaners, but we have no way of gauging this interest. We can solve this problem by assigning an implicit feature such as "appeals to dog owners" to both products. This demonstrates both the flexibility and weakness of content-based filtering systems. As new information about relationships between products becomes available it can be incorporated by defining new features. But, for this same reason, it might not be possible to use a content-based filtering system without human intervention, since we have to recognize the need to introduce new features into content.

Collaborative filtering[1] aims to sidestep the problem of feature design by recommending items that other people, who are similar to the person in question, have liked. A collaborative filtering system, such as Ringo (Shardanand and Maes, 1995) and GroupLens (Konstan et al., 1997), works by collecting explicit user ratings of items in question (e.g., movies, CDs or USENET postings). Users are then compared based on how similar their ratings are, and they are recommended items favored by other people with similar interests. To compute these "word-of-mouth" recommendations, it has been suggested in the literature to use clusters in the space of user profiles in order to define prototypical profiles (Ungar and Foster, 1998).

Viewed broadly, collaborative filtering suggests using the information about a group, which can be the whole population of users or a cluster, in order to produce individual recommendations. In our recommender system we use two sources of such information. First, we apply *associations mining* (Agrawal and Srikant, 1994) to customer purchase data in order to derive relationships between product classes and sub-classes. Since these relationships are based on actual purchases, we expect to identify additional product-class relationships (e.g., purchasers of dog-food products also buy carpet cleaners) that are not captured by the product taxonomy (or even by more sophisticated keyword matching). Second, we use *clustering* (Everitt, 1993) to assign customers into groups with similar interests, based on prior purchase patterns. By itself, content-based filtering does not incorporate information about the relative popularity of products among other customers. For this reason,

[1] Also referred to in literature as *social filtering* (Shardanand and Maes, 1995)

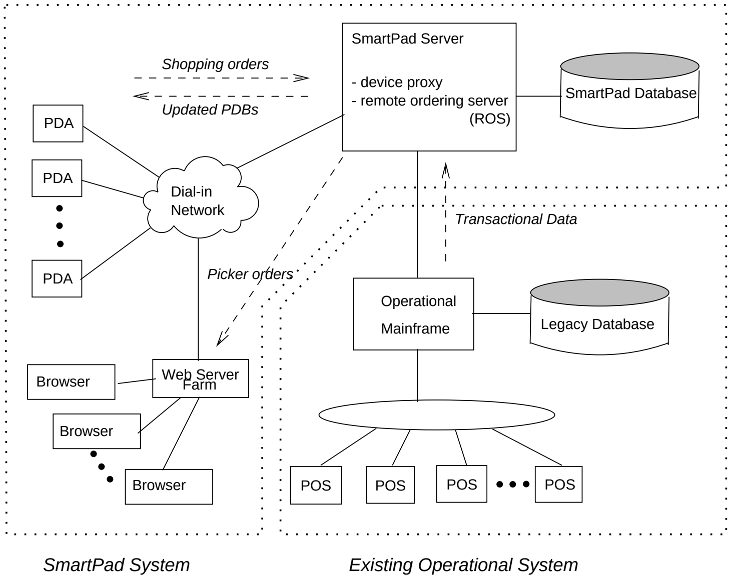


Figure 1. Overview of the SmartPad system

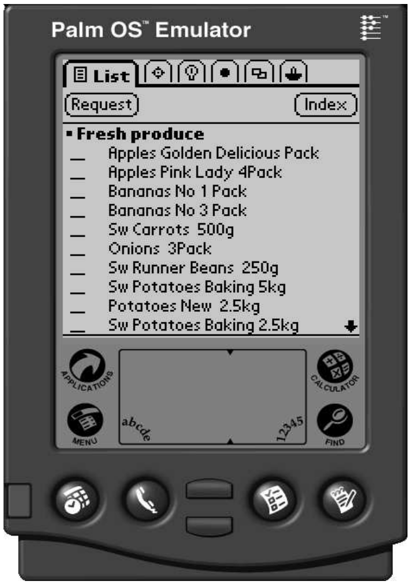
we use the cluster analysis to build ranked lists of the most popular products among customers assigned to each cluster; recommendations for a specific customer are then drawn from products popular among other members of the customer's cluster.

In summary, our recommender system uses content-based filtering at its core, with the ideas from collaborative filtering utilized both to refine the content model and to make recommendations dependent on shared interests within customer clusters.

The organization of this paper is as follows. Section 2 provides a brief overview of the overall SmartPad system, with an emphasis on those aspects relevant to recommender systems. Details of the recommender system are provided in Section 3, while Section 4 describes the associations mining and clustering analyses. Section 5 describes early results and user feedback obtained to date from a field trial of the system, and Section 6 provides an overall summary.

2. The SmartPad Remote Shopping System

Figure 1 provides an overview of the SmartPad environment (Kotlyar et al., 1999). The overall system consists of a network of mobile devices (PDAs) connected to the SmartPad server through a dial-in service. Each customer participating in the program is issued a PDA which



▪ Fresh produce
- _ Apples Golden Delicious Pack
- _ Apples Pink Lady 4Pack
- _ Bananas No 1 Pack
- _ Bananas No 3 Pack
- _ Sw Carrots 500g
- _ Onions 3Pack
- _ Sw Runner Beans 250g
- _ Sw Potatoes Baking 5kg
- _ Potatoes New 2.5kg
- _ Sw Potatoes Baking 2.5kg

Figure 2. SmartPad main screen

runs a consumer application enabling the user to build a shopping list and send it to the server. Products in the order are chosen from three personal databases (PDBs) stored on the PDAs: personal catalog, recommendations, and promotions. (Promotions refer to discounted products offered by the participating Safeway store, while recommendations are products suggested by the system discussed in this paper.) Together, the PDBs serve as a surrogate for the store catalog, which is too large to be stored or shown to the user on the device. Products not appearing on any of the PDBs can be added to the order list manually, either as free-form text or as product numbers from a printed catalog.

As shown in Figure 1, transaction data from the existing Safeway operational system is made available to the SmartPad server. The SmartPad database contains detailed product information as well as customer spending histories for a large number of Safeway customers, including all participants in the actual SmartPad program. As discussed in the following section, this data provides the raw input data for the recommender system including the required data mining analysis.

All computations associated with the personalized recommender system are performed on the SmartPad server. In the current trial program, new sets of recommendations are generated weekly for all SmartPad participants, and stored on the server in the SmartPad database. These recommendations, along with the new promotions and the updated personal catalog, are transmitted to the targeted customer by synchronizing the PDBs on the customer's device at the next server connection initiated by this customer.

Figure 2 shows a view of the PDA main screen for the SmartPad consumer application. The various tabs shown here are used to access lists such as the personal catalog, store promotions, and recommendations, as well as other useful information such as bonus points accumulated by the customer.

3. The Personalized Recommender System

3.1. Overview of the Recommender System

Figure 3 provides an overview of the analysis involved in the personalized recommender system. The input data from the SmartPad database consists of descriptions of approximately 30,000 products and a database containing summarized customer purchase data for roughly 20,000 Safeway customers, including the customers enrolled in the SmartPad pilot program. Not all products are eligible for recommendation; for example, we avoid recommending tobacco, health products, and other inappropriate product classes. As discussed in Sections 3.3 and 4.2, all customers in the database are clustered based on normalized spending vectors constructed using their previous purchase behavior. Each customer is assigned to a single cluster, allowing us to build separate lists for each cluster containing the most frequently purchased products among customers in the cluster. Recommendations for a specific target customer enrolled in the SmartPad program are then drawn from the list of popular products in the assigned cluster for this customer. This cluster-specific list is passed to a "matching engine", which ranks this list of products according to the expected appeal to the target customer. The matching engine also utilizes product relationships or "affinities" computed using associations mining, as described in Sections 3.4 and 4.1. The personalized recommendation list, comprising the 10 to 20 products with the highest scores, are returned to the customer's device during the next server-connection session. It is important to note that the recommendation list, by design, will contain no products previously purchased by this customer. Unlike promotions, no price discounts or other incentives are offered on recommended products.

3.2. Product Taxonomy

Figure 4 illustrates the three-level hierarchical product taxonomy used by Safeway. Products are divided across $G = 99$ product classes. Each product class is subdivided into fewer than 100 subclasses, generating

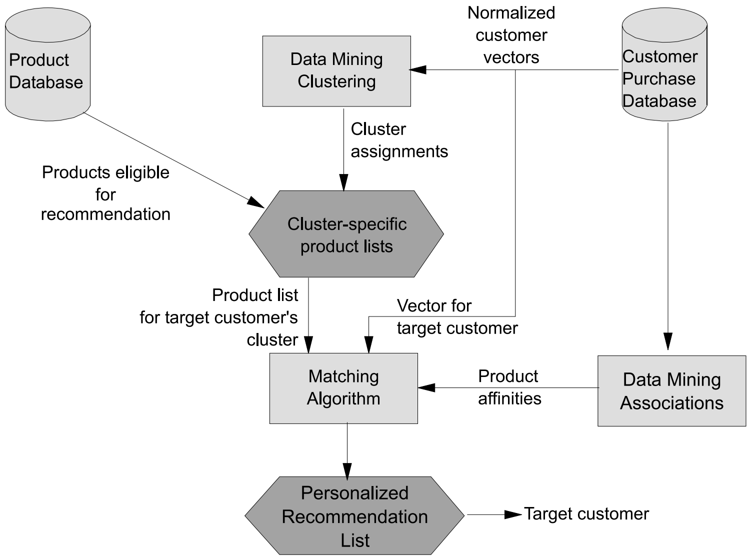

Product Database
Data Mining Clustering
Normalized customer vectors
Customer Purchase Database
Cluster assignments
Products eligible for recommendation
Cluster-specific product lists
Product list for target customer's cluster
Vector for target customer
Matching Algorithm
Product affinities
Data Mining Associations
Personalized Recommendation List
Target customer

Figure 3. Overview of the personalized recommender system

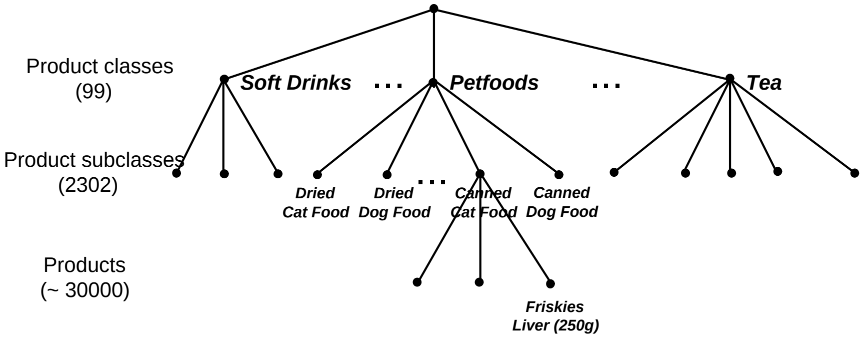


Figure 4. Safeway product taxonomy

a total of $S = 2302$ product subclasses. Absolute customer spending is available at any level in this hierarchy. Although specific to Safeway, the taxonomy used here is typical of the multi-level hierarchy often used to represent retail product catalogs.

Given a specific target customer, the recommendation system seeks to determine products which are best matched to this customer's spending profile. For this reason, we need to be able to construct a profile or vector for each customer representing the customer's "interest" across a range of attributes. This representation is designed to be similar to a vector of explicit ratings typically used in collaborative filtering applications (Shardanand and Maes, 1995). Similarly, we need to be able to represent the "appeal" of each product measured in this same attribute space. We use the subclass level of the Safeway product taxonomy to construct this attribute space. With reference to Figure 4, this defines an $S = 2302$ dimensional space in which we will match products to customers. In the following subsections, we describe the construction of the customer and product vectors.

3.3. Customer Model

We represent the absolute spending for customer m as

$$\mathbf{C}^{(m)} = [C_{m1}, \ldots C_{ms}, \ldots C_{mS}]^T, m = 1, \ldots, M, \tag{1}$$

where C_{ms} denotes the absolute spending of customer m across all products contained in subclass s, M is the total number of customers, and S is the number of product subclasses described above. C_{ms} is computed from the raw transaction data as the sum over the past several months of spending data, and hence reflects this customer's grocery purchases over multiple shopping visits.

We then apply two separate normalizations to this result to obtain the final customer vector. First, we convert absolute spending for each subclass to fractional spending in this subclass by normalizing to this customer's total spending over the period:

$$\hat{C}_{ms} = \frac{C_{ms}}{\sum_{s'=1\ldots S} C_{ms'}} \tag{2}$$

The resulting *fractional spending* vector characterizes this customer's interest in each subclass relative to other subclasses. Since commonly purchased subclasses such as fresh vegetables will tend to dominate the fractional spending, it is also useful to take the ratio of the individual customer's fractional spending in a subclass to the mean value for this subclass taken over all other customers:

$$\hat{\hat{C}}_{ms} = \frac{\hat{C}_{ms}}{\frac{1}{M}\sum_{m'=1\ldots M}\hat{C}_{m's}} \tag{3}$$

This *normalized fractional spending* thus quantifies the customer's interest in this subclass relative to the customer database as a whole, normalized such that entries equal to 1 imply an average level of interest in a subclass relative to all other customers.

3.4. Product Model

We use the following notation to represent the product classifications illustrated in Figure 4:

$$\begin{array}{rcl} \mathcal{S}(n) & \equiv & \text{the product subclass number for the product } n \\ \mathcal{C}(s) & \equiv & \text{the product class for the product subclass } s \end{array} \tag{4}$$

For brevity, we will use "$\mathcal{C}(n)$" instead of "$\mathcal{C}(\mathcal{S}(n))$" to denote the class that product n belongs to.

Each product $n = 1\ldots N$ is represented by an S-dimensional vector $\mathbf{P}^{(n)}$, and hence has the same dimensionality as the customer vectors discussed in the previous section. The individual entries $P_s^{(n)}$, $s = 1\ldots S$, reflect the "affinity" the product has to the subclass s, i.e., the extent to which this product will appeal to a customer with an interest in this subclass. The simplest solution is to set the entries of the vector $\mathbf{P}^{(n)}$ as:

$$P_s^{(n)} = \begin{cases} 1 & \text{if } s = \mathcal{S}(n) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This captures the affinity between different products in a subclass: a purchase of a product in a subclass implies an interest in other products in the same subclass. This simple form leads to straightforward content-based filtering: if you purchased from this product subclass, you will receive recommendations only for other products within this subclass. Clearly, this eliminates the possibility of "cross-selling" potentially interesting products outside of this immediate subclass.

More specifically, Eq. (5) ignores other possible kinds of affinity between products:

1. affinities between products that belong to the same class (but to different subclasses), and
2. affinities that are derived from data-mining analysis such as associations mining (Agrawal and Srikant, 1994) performed independently at both the class and subclass level.

Case (1) is also content-based filtering since only product-taxonomy information is used. However, case (2) incorporates associations derived from “global” customer behavior, and hence reflects aspects of collaborative filtering. In order to account for the first kind of affinity, we have heuristically chosen to set $P_s^{(n)} = 0.5$ for every subclass s that is in the same class as the product n. To account for the second kind, we set $P_s^{(n)} = 0.5$ for every association $\mathcal{C}(n) \Longrightarrow \mathcal{C}(n')$ between the class of the product n and the class of product n'. We also compute associations at the subclass level, and use these as described below to set affinities between associated subclasses.

Some examples of the use of associations to specify the product vector are as follows. If the use of bacon strongly implies the use of eggs (i.e., bacon and eggs are associated classes), then the vector for a particular bacon product will receive a 0.5 in the entries that correspond to all subclasses of the class “Eggs” (e.g., “Eggs/Large” or “Eggs/Medium”). Also, if there is a strong association between the subclasses “Bacon/LowFat” and “Eggs/Whites”, then the vector for the particular low-fat bacon product will receive a 1.0 in the entry for the subclass “Egg/Whites”. Given the class-level association between “Bacon” and “Eggs”, the vector for a particular low-fat bacon product will also receive contributions (0.25) in the entries associated with *all* subclasses with “Eggs”.

Overall, the formula for the entries in the product vector is:

$$P_s^{(n)} = \begin{cases} 1.0 & \text{if } s = \mathcal{S}(n) \quad \text{(within same subclass)} \\ 1.0 & \text{if } \mathcal{S}(n) \Longrightarrow s \quad \text{(within associated subclass)} \\ 0.5 & \text{if } \mathcal{C}(s) = \mathcal{C}(n) \quad \text{(subclass within same class)} \\ 0.25 & \text{if } \mathcal{C}(n) \Longrightarrow \mathcal{C}(s) \quad \text{(within subclass of associated class)} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The multipliers for subclass associations were set higher than those at the product class level in order to encourage customers to try popular new products outside their current shopping pattern, but not too far outside. The results in Section 5 seem to indicate that this intended effect was indeed achieved.

3.5. Matching Algorithm

In the preceeding sections, we have developed descriptions of customers and products in the vector space defined by product subclasses. The final step in the recommendation process [see Figure 3] is to score each candidate product for a specific customer and select the best matches. This score should reflect the degree of similarity between the customer vector and each product vector. Possible metrics include Pearson correlation (Shardanand and Maes, 1995), Euclidian distance (Shardanand and Maes, 1995), and cosine projection (Salton, 1989). As noted in (Salton, 1989), the choice among these methods is heuristic, and we chose cosine projection based on inspection of the generated recommendations. Hence, the score σ_{mn} between customer m and product n is computed using a cosine coefficient between the corresponding vectors, $\mathbf{C}^{(m)}$ and $\mathbf{P}^{(n)}$:

$$\sigma_{mn} = \rho_n \frac{\mathbf{C}^{(m)} \cdot \mathbf{P}^{(n)}}{\|\mathbf{C}^{(m)}\| \|\mathbf{P}^{(n)}\|}, \tag{7}$$

where the $\cdot$ operator denotes inner (dot) product:

$$\mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i, \tag{8}$$

$$\|\mathbf{x}\| = \sum_i |x_i|, \tag{9}$$

and ρ_n is a modulation factor discussed below. Both customer and product vectors are sparse, and we use this sparse structure to minimize the number of arithmetic operations necessary to evaluate Eq. (7).

If $\rho_n \equiv 1$ for $n = 1 \ldots N$ in Eq. (7), all products in a product subclass will have identical scores for a specific customer since the product vectors are constructed at the subclass level. Hence, ρ_n can be used as a "tie-breaker" to differentiate products. In general, ρ_n can reflect the desire to "push" one product over another, for example, due to higher inventories, profit margins, or other market conditions. Since we have product profit margins available, we initially computed this factor as

$$\rho_n = \left[\frac{PM_n}{\overline{PM}}\right]^{\alpha}, \tag{10}$$

where PM_n is the profit margin associated with product n, $\overline{PM}$ is the mean value over all products, and α is an empirical factor designed to control the overall influence. However, we discovered that more profitable items were not necessarily more popular, and we chose not to implement this feature in the production system. Instead, we break ties by random sampling from up to 5 products with the top scores within a specific product subclass. In order to distribute the recommendations across product classes, we limit the number of recommendations for each customer to 1 product per product subclass, and 2 products per product class. In the end, we return a list of approximately 10 to 20 products to the customer's PDA as described in Section 2; this number is determined primarily by the size of the display on the PDA device.

Previously bought products are excluded from the recommend list, since the recommender is meant to broaden each customer's purchase patterns. It is also worth noting that the list of the most popular products in each product subclass is a dynamic list, created by periodic querying of the transaction database. This prevents recommending seasonal items at inappropriate times of the year. Since recommendations are generated on a weekly basis, the list of candidate products could also be updated weekly; in practice, we do it monthly, with an optional weekly edit to comb out special situations such as Easter candy the week after Easter. It should also be noted that several of the 99 product classes, such as Petrol and Tobacco, are excluded from the recommender by design.

4. Data Mining Analysis

Intelligent Miner for Data (Intelligent Miner, 1998) was used to perform the associations mining and clustering analysis shown in Figure 3. Customer transaction data were used as input to both calculations, but, naturally, the data were formatted and normalized differently.

4.1. Associations Mining

As discussed in Section 3.5, associations mining is used to compute the product affinities necessary to construct the product vectors in Eq. (6). We used Intelligent Miner's "*apriori*" associations algorithm (Agrawal and Srikant, 1994) to extract associations (independently) among the 99 product classes and also among the 2302 product subclasses shown in Figure 4. The raw input data for this analysis was 8 weeks of product-level transaction data for 8000 customers with above-average spending. In order to compute product-class associations, the transactions were first binned according to product class, and then aggregated over time. The final input table consisted of (customer ID, product-class ID) tuples, each representing a product class in which the customer has purchased over this 8-week period. An analogous procedure was used to extract associations at the product-subclass level. Note that our objective here differs from conventional market-basket analysis in which associations are computed among products purchased as part of the same transaction.

Consistent with the form of Eq. (6), we computed only simple associations containing a single item in both the body and the head of the rule, e.g., A$\Rightarrow$B. For several reasons, we needed to limit the number of rules we used to about 100 from each of the two levels. These reasons

Table I. Sample associations computed at the product-class and product-subclass levels

Sup	Conf	Lift	Class or Subclass	Relevant Affinities
0.059	0.41	2.4	20(Baby Products)	⇒ 41(Canned Pasta)
0.082	0.47	2.2	66(Table Wines)	⇒ 68(Beer/Lager/Spirits)
0.125	0.50	2.0	90(Fresh Beef)	⇒ 91(Pork/Lamb)
0.025	0.38	9.0	2010(Baby:Disposable Nappies)	⇒ 2007(Baby:Wipes)
0.016	0.33	4.9	2010(Baby:Disposable Nappies)	⇒ 1012(Dairy:Childrens' Yogurt)
0.01	0.33	4.9	2010(Baby:Disposable Nappies)	⇒ 3115(Instore:Babysitting Center)
0.012	0.37	3.4	1020(Dairy:Childrens' Fromage)	⇒ 3115(Instore:Babysitting Center)
0.016	0.52	5.2	2306(Biscuits:Kids Biscuits)	⇒ 3115(Instore:Babysitting Center)
0.022	0.30	4.9	9015(Fresh Beef:Beef Joints)	⇒ 9120(Pork/Lamb:Pork Joints)

included clarity of understanding as well as computational complexity, especially in the case of the product-class rules, since the recommender algorithm assumes that all the subclasses within two related classes are also related via affinities. We used a combination of thresholds on the support, confidence, and lift of a rule to achieve this filtering[2] . After experimenting with several datasets and inspecting the resulting rules, we chose a combination of minimum support in the 1% – 4% range, minimum confidence of 30% to 40%, and minimum lift of 2 to 3. Although heuristic, these choices produced a reasonable number of associations for the various Safeway datasets we examined. Applications to other purchase data may require different limits on these parameters.

The associations analysis is performed periodically as a "batch" process. Rules meeting the filtering criterion are stored in the SmartPad database (see Figure 1) for subsequent access during the computation of the personalized recommendations.

Table I shows a subset of affinities computed at the product-class and subclass levels. The rule head is on the right, so the textual format of the first rule is "5.9% of all customers buy both baby products and canned pasta. When a customer buys baby products, the customer also buys canned pasta in 41% of the cases, which is 2.4 times the rate one would expect if the sales of these two were statistically independent". In

[2] Given the rule A⇒B, where A and B are itemsets and T is the total number of customers, the *support* for the rule, S(A⇒B), is the percentage of customers who have spent in both A and B; the *confidence* in the rule is S(A⇒B)/S(A), and the *lift* is S(A⇒B)/(S(A)*S(B)). The lift is the ratio of actual confidence to the expected confidence, where the latter is computed assuming that A and B are statistically independent. Support and lift are symmetric in A and B; confidence is not.

tags of the form “Baby:Disposable Nappies”, the items before and after the colon refer to product class and product subclass, respectively. The numbers preceeding these classes are actual class and subclass numbers; we will refer to these in the discussion of specific recommendations in Section 5.1.

4.2. Clustering Analysis

Safeway assigns each customer to a pre-defined customer class (or segment) based on purely demographic information derived from questionnaires. Examples of these classes are “empty nesters”, “young parents”, and so on. While generally useful, this demographic-based segmentation does not provide the necessary detail on each customer’s actual spending preferences. For this reason, we did not use these classifications, but instead clustered customers on the basis of their spending in the 99 product classes. We produced lists of the most popular products per subclass for each cluster, and then used the cluster-specific list of popular products as input to the generation of recommendations for a customer in a particular cluster. The five most popular products in a subclass can be quite different in different clusters, as shown in more detail further below.

Given a large number of database records, clustering (Michaud, 1999) can be useful for identifying a small number of prototypes (i.e., cluster centers) which represent dominant characteristics or features present in the input data set. Key issues in any clustering analysis are the dimensionality and the normalization of the input data. If the dimensionality of the input space is too large, and a typical record has only a few nonzero entries, then it is difficult to find records with many nonzero attributes in common. For this reason, we did not cluster at the product spending level. We also found that clustering at the subclass spending level did not produce useful results, and thus all clustering analysis was done using customer spending records at the 99-dimensional product-class level (see Figure 3). The input data for this analysis were normalized fractional customer spending vectors, computed as in Eqs. (1) - (3), but evaluated at the product-class level rather than at the product-subclass level. These values were limited to a maximum value of 5 in order to limit the influence of very high product-class spending.

We applied both the neural clustering algorithm (Lawrence et al., 1999) and the so-called “demographic clustering” algorithm[3] (Michaud, 1999) that are available in Intelligent Miner. Identical input data were

[3] The word “demographic” here refers to a particular clustering *algorithm* that was applied to customers’ spending data, whereas its use above in “demographic-

provided to both algorithms. Using some of the classical measures of cluster validity, these results seemed to suggest that the resulting clusters were relatively diffuse, i.e., were not particularly compact. For example, the demographic clustering method, which seeks to maximize the Condorcet criterion[4] for cluster quality, produced an overall Condorcet criterion of 0.56 for 9 clusters, but did so by putting 97% of the records in one cluster. Neural clustering produced clusters that were much more uniform in size, but the Condorcet criterion was in the range of 0.1 to 0.2, and values of the modified Dunn's index[5] used in (Bezdek, 1998) were in the range 0.3 to 0.8 compared with the values between 1 and 2 that they obtain for "good" clusters.

However, these cluster validity measures tend to favor categorical data with a relatively small number of attributes, and may be less relevant for our continuous data and relatively large number of attributes. There is a good deal of overlap in people's spending, for understandable reasons. One group of people may have extraordinary spending in baby products and none in tobacco, and another may be just the reverse, but both groups presumably need paper towels, milk, bread, lettuce, and many other everyday necessities. For this reason, it seems plausible to evaluate the clusters in terms of their dominant attributes, rather than strictly by their compactness and distinctiveness relative to other clusters.

For example, Figure 5 shows a 3 by 3 self-organizing feature map (Kohonen, 1995) produced by the neural clustering method. Each cell represents a cluster, with the largest cluster containing 17% of the customers, the smallest 9%. The product classes shown represent the 3 attributes which most distinguish members of the cluster from the database background. For example, the cluster in the center of this map is likely to represent "families with young children", based on significantly above-average spending in baby products, clothing, and dairy products. The cluster at the lower right appears to represent "serious bakers", given dominant spending in sugar, home baker products, and canned fruits. The clusters in the upper left focus on specific non-food items, e.g., petrol[6] , tobacco, and wines/beer/spirits. Feature

based segmentation" refers to the demographic *data* (age etc.) that were used as the basis for creating those segments.

[4] The Condorcet criterion (Michaud, 1999) is the difference of two factors, one of which measures how similar the records within a cluster are, while the other measures how different the records within a cluster are from all records not in the cluster. For perfect clustering, the first factor is 1.0 and the second is 0.0.

[5] Dunn's index is the ratio of intercluster distance to cluster diameter. We computed it using the approximation called ν_{53} in (Bezdek, 1998).

[6] The clustering is based on customer spending in all 99 product classes, even though a few classes such as petrol and tobacco are excluded from our lists of recom-

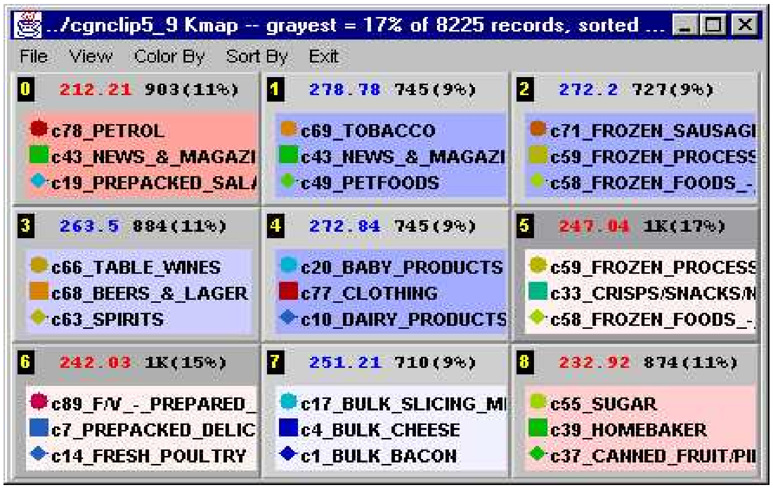


Figure 5. Cluster analysis at the product-class level

Table II. Summary of cluster characteristics

Cluster number	Fraction of records	Most significant product class	Cluster share	Cluster-share enrichment
0	0.11	Petrol	0.24	2.2
1	0.09	Tobacco	0.50	5.6
2	0.09	Frozen Sausage	0.67	7.4
3	0.11	Wines	0.39	3.5
4	0.09	Baby Products	0.56	6.2
5	0.17	Frozen Foods	0.24	1.4
6	0.15	Fresh Vegetables	0.25	1.7
7	0.09	Bulk Sliced Meat	0.34	3.8
8	0.11	Sugar	0.22	2.0

maps generated with more clusters show increasing refinements in these customer purchase patterns.

Table II summarizes these same 9 clusters, showing the most significant product class which defines the cluster, as well as the fraction of total customers assigned to this cluster. We define the "cluster share" of a product class as the spending in that product class by a cluster of

mendable products. We found, for example, that tobacco purchasers also buy more and stronger deodorizers and a different set of wines than non-tobacco purchasers.

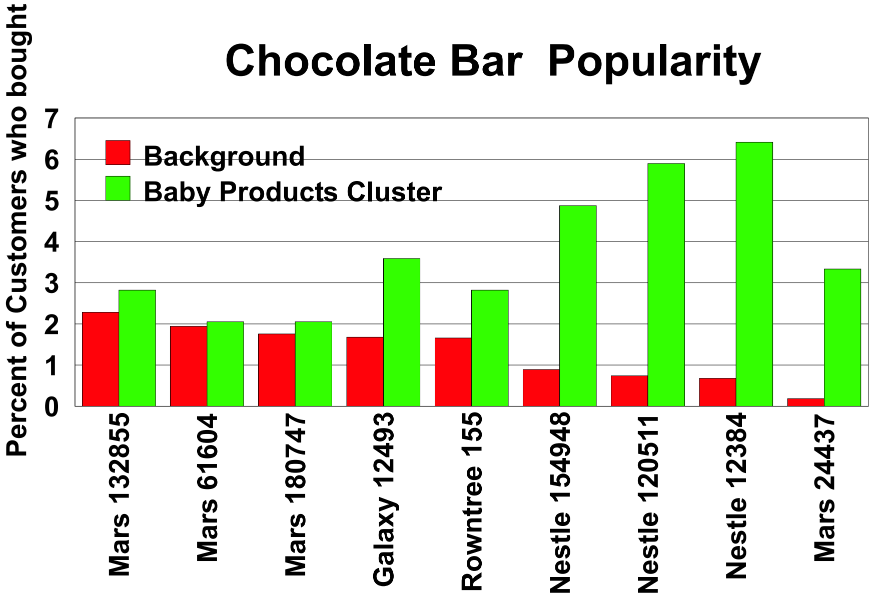


Figure 6. Cluster-specific product preferences

customers divided by the total spending in that product class:

$$\text{Cluster share in class i } = \frac{\sum_{m \in cluster} C_{mi}}{\sum_{m=1,\ldots,M} C_{mi}}, \tag{11}$$

where C_{mi} is the absolute spending for customer m in product class i. The enrichment in cluster share is defined as the ratio of cluster share divided by the fraction of the customer population which is in the cluster. For example, cluster number 4 is characterized by spending in baby products: the 9% of the customers in this cluster generated 56% of the total database spending for baby products, an enrichment by a factor 6.2. Popular products among members of this cluster are likely to be quite different than the most frequently purchased products among members of the "tobacco" cluster shown as cluster 1 in Table II.

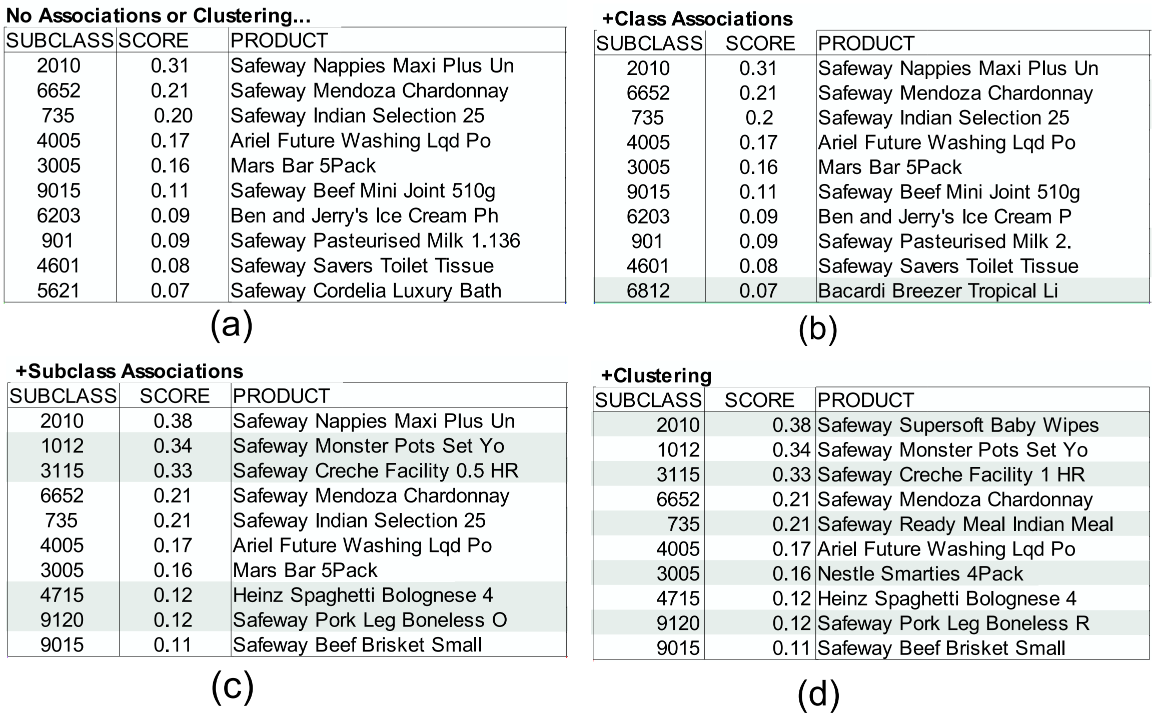
Figure 6 shows a specific example of preferred products in a cluster, compared to popular products across the database as a whole. For the background population, three of the five most popular chocolate bars are made by Mars and none by Nestle, but for cluster 4 (baby products) three of the top five are made by Nestle and only one by Mars, suggesting a preference for Nestle among young children, or at least among those who shop for them.

Table III. Sample customer fractional spending at the product subclass level

Subclass number	Spending fraction	Subclass name
7801	0.077	PETROL:PETROL
2010	0.058	DISPOSABLE NAPPIES:BABY PRODUCTS
735	0.046	INTERNATIONAL:PREPACKED DELICATESS
6652	0.035	WHITE ENGLISH:TABLE WINES
3005	0.032	CHOCOLATE:CONFECTIONERY
7734	0.027	GIRLS PAJAMAS:CLOTHING
4005	0.026	LIQUIDS:HOME LAUNDRY/LIQUID
6203	0.021	LUXURY ICE CREAM:ICE CREAM
9015	0.021	BEEF JOINTS:FRESH BEEF
4001	0.017	SOAP/DETERGENTS POWD:HOME LAUNDRY/LIQUID
2001	0.016	BABY FOOD DRY:BABY PRODUCTS
901	0.015	S/W FULL CREAM MILK:MILK
8004	0.015	ENT UK VIDEO:HOME ENTERTAINMENT
7741	0.015	BOYS BRIEFS:CLOTHING
7733	0.015	BOYS PAJAMAS:CLOTHING
6651	0.015	WHITE SOUTH AFRICA:TABLE WINES
4601	0.014	TOILET TISSUE:PAPER PRODUCTS
905	0.013	S/W SEMI SKIMMED MIL:MILK
6670	0.013	WHITE - NORTH AMERIC:TABLE WINES
2007	0.012	BABY WIPES:BABY PRODUCTS

5. Results

The results that we report here were obtained during the first 8 months of the SmartPad program at Safeway (Kotlyar et al., 1999). Phase 1 of the program lasted approximately 7 months, and included 200 customers from one store. As of this writing, Phase 2 has been in progress for approximately two months, and includes a second participating Safeway store. Different versions of the recommender were used in the two phases. In this section, we first show a sample recommendation list and describe how the use of data mining influences its content. We then describe our experience with the first version of the recommender, the lessons we learned and consequent changes we made, and the performance of the current version. Full implementation of the cluster-specific input lists was not completed for these field tests: the input to the matching process was constructed using popular products within a single cluster consisting of all customers in the database.



No Associations or Clustering...

SUBCLASS	SCORE	PRODUCT
2010	0.31	Safeway Nappies Maxi Plus Un
6652	0.21	Safeway Mendoza Chardonnay
735	0.20	Safeway Indian Selection 25
4005	0.17	Ariel Future Washing Lqd Po
3005	0.16	Mars Bar 5Pack
9015	0.11	Safeway Beef Mini Joint 510g
6203	0.09	Ben and Jerry's Ice Cream Ph
901	0.09	Safeway Pasteurised Milk 1.136
4601	0.08	Safeway Savers Toilet Tissue
5621	0.07	Safeway Cordelia Luxury Bath

(a)

+Class Associations

SUBCLASS	SCORE	PRODUCT
2010	0.31	Safeway Nappies Maxi Plus Un
6652	0.21	Safeway Mendoza Chardonnay
735	0.2	Safeway Indian Selection 25
4005	0.17	Ariel Future Washing Lqd Po
3005	0.16	Mars Bar 5Pack
9015	0.11	Safeway Beef Mini Joint 510g
6203	0.09	Ben and Jerry's Ice Cream P
901	0.09	Safeway Pasteurised Milk 2.
4601	0.08	Safeway Savers Toilet Tissue
6812	0.07	Bacardi Breezer Tropical Li

(b)

+Subclass Associations

SUBCLASS	SCORE	PRODUCT
2010	0.38	Safeway Nappies Maxi Plus Un
1012	0.34	Safeway Monster Pots Set Yo
3115	0.33	Safeway Creche Facility 0.5 HR
6652	0.21	Safeway Mendoza Chardonnay
735	0.21	Safeway Indian Selection 25
4005	0.17	Ariel Future Washing Lqd Po
3005	0.16	Mars Bar 5Pack
4715	0.12	Heinz Spaghetti Bolognese 4
9120	0.12	Safeway Pork Leg Boneless O
9015	0.11	Safeway Beef Brisket Small

(c)

+Clustering

SUBCLASS	SCORE	PRODUCT
2010	0.38	Safeway Supersoft Baby Wipes
1012	0.34	Safeway Monster Pots Set Yo
3115	0.33	Safeway Creche Facility 1 HR
6652	0.21	Safeway Mendoza Chardonnay
735	0.21	Safeway Ready Meal Indian Meal
4005	0.17	Ariel Future Washing Lqd Po
3005	0.16	Nestle Smarties 4Pack
4715	0.12	Heinz Spaghetti Bolognese 4
9120	0.12	Safeway Pork Leg Boneless R
9015	0.11	Safeway Beef Brisket Small

(d)

Figure 7. Progressive development of a sample recommendation list: (a) uninfluenced by data mining, (b) with product-class association rules, (c) with product subclass association rules added, (d) with clustering.

5.1. Analysis of a Sample Recommendation List

Table III shows the fractional spending at the product subclass level for a customer who has spending in the “baby-products” subclasses. Figure 7 (d) shows the recommendation list generated on the basis of this spending profile. The figure also traces the development of this recommendation list through four stages:

(a) As it would appear without the use of either data-mining associations or clustering, i.e., on the basis of content-based filtering alone.

(b) As it would appear if we only used association rules from the product class level. Note the appearance of an item from subclass 6812 due to a boost in its score resulting from the association rule in Table I between class 68 and 66, several of whose subclasses are present in the spending list in Table III.

(c) As it would appear using association rules from both the class and subclass levels, but without clustering. Four new products have appeared in the top ten as a result of product subclass association rules found in Table I. Note that the product that appeared in (b) has disappeared. However, although not shown in this figure, the score for the product from subclass 9120 was influenced by association rules from both the product class and subclass levels.

(d) As it appears when the products for the top ten product subclasses are chosen from the list of popular products for the cluster that this customer falls into, namely, cluster 4 in Figure 5. Note in particular the appearance of the Nestle candy bar instead of the Mars, consistent with the cluster preferences shown in Figure 6.

The much greater effect of subclass association rules (Figure 7 (c) versus (b)), which results from the choice of values in Eq. (6), is reflected in the statistics for the full set of recommendation lists as well: on average, 33% of the items on a recommendation list are in product subclasses that are new to the customer (no spending in the subclass within the past 3 months) and 16% are in product classes new to the customer.

5.2. Results for Phase 1

During Phase 1 of the trial, a total of 1957 complete orders were processed by the SmartPad system. Of these, 120 orders (6.1%) contained at least 1 product chosen from the recommendation list. (It is important to recall that the recommendation list, by design, will contain no products previously purchased by this customer.) An objective of the product recommender is to provide a boost in revenue comparable to the spontaneous purchases a shopper might make while walking through the store or after receiving a flyer in the mail. By this measure, the results for the initial recommender were somewhat disappointing: the corresponding boost in revenue was 0.3% over and above the revenue generated by products bought from the main “personal catalog” shopping list.

As trial program progressed, we noticed with interest that the distribution of spending in the SmartPad product categories[7] was different for items bought from the recommendation list versus the personal catalog (see Figure 8), even though the distribution of items *available* from each list were quite similar. For example, wines accounted for

[7] The 16 SmartPad product categories are a superset of the 99 Safeway product classes, minus those classes like petrol and tobacco that are not available via SmartPad.

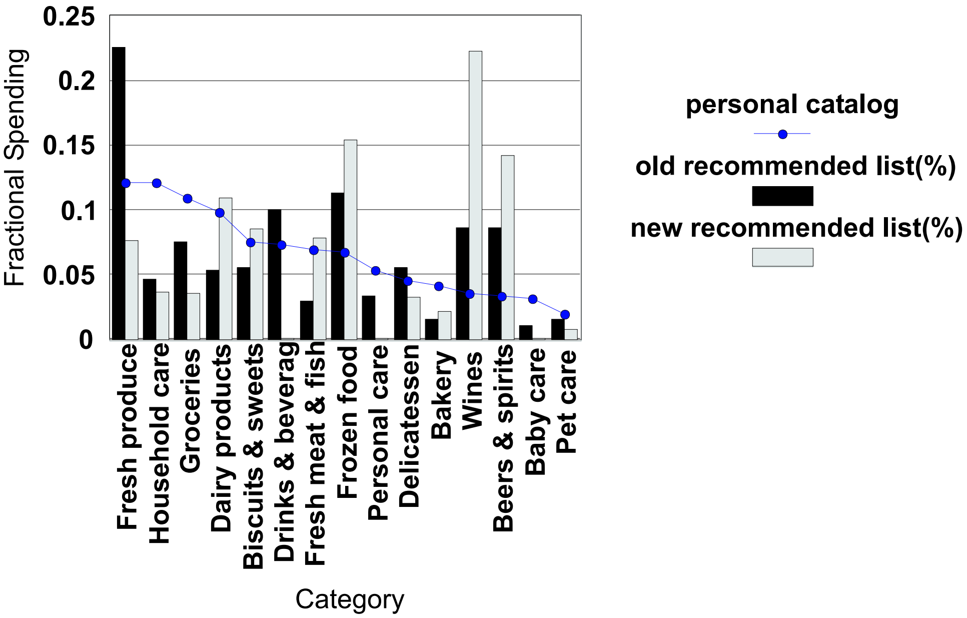


Figure 8. Product mix comparison of items bought from the main shopping list, from the old recommendation list, and from the new recommendation list. The fraction spent in 16 product categories is shown for each list.

only 3.5% of the revenue from the main shopping list, but 8.7% of the revenue from the recommendation list. By contrast, products in the household care category accounted for 12.1% of the revenue from the main shopping list but only 4.6% from the recommendation list. We interpreted these results to mean that there is a set of categories in which recommendations are more welcome than others, and interviews with participating customers confirmed that interpretation. They wanted more “interesting” recommendations, and wines fit that description but household care products did not.

Armed with this insight, we proceeded to trim the list of subclasses from which recommendable products were drawn, emphasizing those product classes in which the spending percentage from the recommendation list exceeded that on the main shopping list, and de-emphasizing the others, with the aim of creating a more “fun” or welcome set of recommendable products. We also added a second source of items eligible for recommendation, namely, new products introduced within the last month. We allowed new products to come from any category, on the rationale that their novelty made them interesting per se. This new recommender went into operation two months before this writing, and is the subject of the next section.

5.3. Results with the Current Version

As mentioned above, a second store and a new set of customers were added to the SmartPad program at the same time that the new recommender was introduced with the changes described above. The number of recommendations sent to the customer's device was increased from 10 products in the initial phase to 20 products in the current version.

For both stores, the results with the new recommender were better than those obtained with the old recommender at the original store, but this was much more pronounced for the new store, bearing out the saying that one gets only one chance to make a first impression.[8] For the old store, the fraction of orders containing at least one recommended product increased from 6.1 to 7.7%, and the revenue boost rose to 0.5 % from its previous value of 0.3 %, a modest increase. But for the new store, the returns were much greater: 25% of the orders included at least one recommendation, with a revenue boost of 1.8%. Safeway considered this boost to be quite respectable given their experience with other promotional methods.

We were reasonably successful in meeting our goal of encouraging shoppers to try new things but not drastically new things: 51% of the acceptances from the recommendation lists corresponded to subclasses in which the shopper had spent no money in the previous three months, but only 4% corresponded to new product classes. For the recommendation lists themselves, 33% of the recommended products on average were from subclasses the shopper had not spent in before, and 16% were from product classes that were new to the customer. Outside the environment of the recommender, the rate of trying new subclasses is substantially lower, and the rate of trying new product classes is practically zero.

5.4. Distributions of Computed Scores

In order to quantify the impact of the recommender system, it would have been useful to have a control group of customers who received "placebo" recommendations, such as a list of randomly chosen products. This approach was not feasible, however, since we were dealing with a live system with real customers doing real shopping. Another approach would be to compute the ratio of accepted recommendations to total recommendations offered. However, looking at the recommendation list is voluntary on the part of the customer (it involves clicking the

[8] It is also possible that SmartPad participants in the new store were inherently more inclined to accept recommendations than those from the original store. We note that the demographics of the two stores are similar.

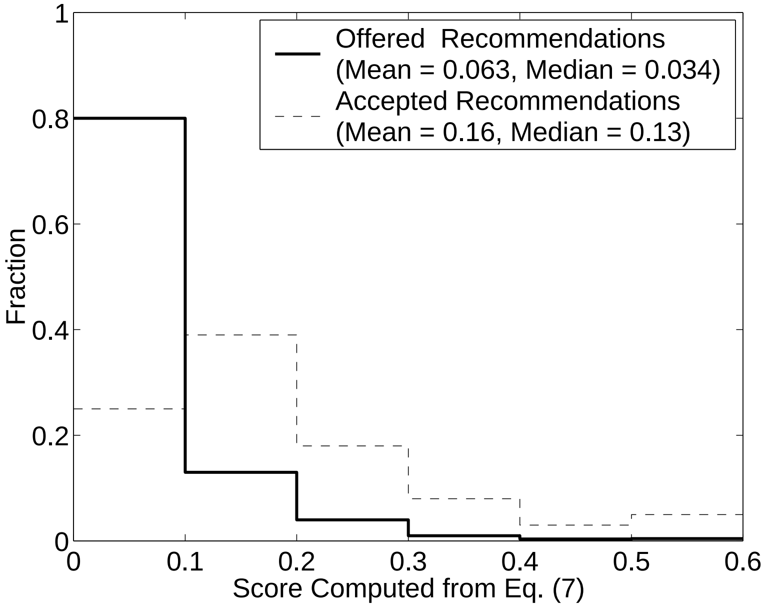


Figure 9. Distributions of scores for offered and accepted recommendations.

“light bulb” tab in Figure 2), and we have no way of knowing whether the customer actually looked at the list of recommended products.[9]

A related issue is the extent to which recommendations with higher scores are accepted preferentially over recommendations with lower scores. We address this issue by comparing the distribution of scores computed from Eq.(7) for accepted recommendations with the analogous distribution for offered recommendations. The results are shown in Figure 9. The scores for the accepted recommendations are based on 243 products accepted from 183 distinct recommendation lists. The distribution for the offered recommendations is taken from approximately 20,000 recommendations made to the customers who accepted at least one recommendation during the pilot program.

Figure 9 shows that the scores of the accepted recommendations are higher than the scores of a large number of offered recommendations. For example, 80% of the products placed onto the recommendation lists have scores below 0.1, but only 25% of the accepted recommendations fall in this lower bin. The mean and median scores for the offered recommendations are 0.063 and 0.034, respectively, while the mean and median scores for the accepted recommendations are 0.16 and 0.13.

[9] Late in the pilot program, a capability was added to the PDA which tagged whether a customer submitting an order had actually looked at the recommendation list.

The difference between the two means, 0.10, falls well within the 95% confidence interval (0.090, 0.105) computed using Student's t-test for the difference between means (Robbins and Van Ryzin, 1975). These results suggest that the score computed using Eq. (7) is indeed a useful indicator of a previously unbought product's appeal to the targeted customer.

6. Summary and Future Work

We have described a product recommendation system developed as part of an overall pervasive computing solution for grocery shopping. The recommendation algorithm combines aspects of content and collaborative filtering to rate new products for a customer based on their prior purchase behavior. Analysis of results obtained during a field test of the system have shown revenue boosts of roughly 1 to 2%, with the interesting observation that people often choose recommendations from product classes in which they have not purchased previously.

The overall SmartPad project is an example of enabling access to server-based data and computation from mobile devices such as PDAs. In our case, the recommendation analysis runs completely on the server, with the PDA providing a mechanism to access this analysis remotely. Given the increasing computational power of these devices, it will be possible to move more of the analysis to the device itself, including compute-intensive portions which heretofore have been possible only on conventional servers. We are exploring this issue in the context of a PDA-based application designed to recommend wine selections based on specific meal choices (Almasi and Lee, 2000). It is likely that many other applications, such as financial analysis, will exploit this opportunity to do increasingly complex analysis on hand-held devices operating in a weakly connected environment.

7. Acknowledgements

We are especially grateful to our colleague Harry Stavropoulos and to Jeremy Wyman of Safeway UK for all their help.

References

Aggarwal, C. C., J. L. Wolf, K-L. Wu, and P. S. Yu. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. *KDD-99 Proceedings*, 201–212, 1999.

Agrawal, R., and R. Srikant. Fast Algorithms for Mining Association Rules. *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, Sept. 1994.

Almasi, G. S., and A. J. Lee. A PDA-Based Personalized Recommender Agent. *Proc. Fifth International Conf. on the Practical Application of Intelligent Agent and Multi Agent Technology*, Manchester, England, 299–309, April 2000.

Balabanovic, M. and Y. Shoham. Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Bezdek, J. C., and N. R. Pal. Some New Indexes of Cluster Validity, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28(3):301–315, June 1998.

Borchers, A., J. Herlocker, J. Konstan, and J. Riedl. Ganging up on Information Overload. *Computer*, 31(4):106-108, April 1998.

Everitt, B. S. Cluster Analysis, London, Edward Arnold, 1993.

First Annual Personalization Summit. *www.personalization.com*, San Francisco, Nov. 15-16, 1999.

Intelligent Miner for Data. *www.ibm.com/software/data/iminer/fordata.*

Kohonen, T. *Self-Organizing Maps.* Springer-Verlag, 1995.

Konstan, J., B. Miller, D. Maltz, J. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of ACM*, 40(3), 1997.

Kotlyar V., M. S. Viveros, S. S. Duri, R. D. Lawrence, and G. S. Almasi. A Case Study in Information Delivery to Mass Retail Markets. *Proceedings of the 10th International Conference on Database and Expert Systems Applications (DEXA)*, Florence, Italy, August/September 1999. Published by Springer-Verlag as Lecture Notes in Computer Science, vol 1677.

Lawrence, R. D., G. S. Almasi, and H. E. Rushmeier. A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems. *Data Mining and Knowledge Discovery*, 3:171–195, 1999.

Michaud, P. Clustering techniques. *Future Generation Computer Systems*, 13(2), 1999.

Resnick, P. and H. R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997. Also see other articles in this special issue.

Robbins, H. and J. Van Ryzin. *Introduction to Statistics.* Scientific Research Associates, Inc., 1975.

Salton, J. *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information By Computer.* Addison-Wesley, 1989.

Salton, J. and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

Shardanand, U. and P. Maes. Social Information Filtering: Algorithms for Automating Word of Mouth. *Proc. CHI 95*, 202–209, ACM Press, 1995.

Ungar, L. H., and D. P. Foster. Clustering Methods for Collaborative Filtering. *Proceedings of 1998 AAAI Workshop on Recommender Systems.* Available as AAAI Technical Report WS-98-08.