

Automatic Construction of Personalized Customer Interfaces

Bob Price

Computing Science
University of Alberta
price@cs.ualberta.ca

Russ Greiner

Computing Science
University of Alberta
greiner@cs.ualberta.ca

Gerald Häubl

School of Business
University of Alberta
gerald.haeubl@ualberta.ca

Alden Flatt

Computing Science
University of Alberta
alden@cs.ualberta.ca

ABSTRACT

Interface personalization can improve a user's performance and subjective impression of interface quality and responsiveness. Personalization is difficult to implement as it requires an accurate model of a user's intentions and a formal model of how an interface meets a user's need. We present a novel model for tractable inference of consumer intentions in the context of grocery shopping. The model makes unique use of a priori temporal relations to simplify inference. We then present a simple interface generation framework that was inspired by viewing user interface interaction as a channel coding problem. The resulting model defines a simplified but clear notion of a user's utility for an interface. We demonstrate the effectiveness of the research prototype on some simple data, and explain how the model can be augmented with richer user modeling to create a deployable application.

Categories and Subject Descriptors

H.5 Information Interfaces and Presentation

G.3 Probability and Statistics

General Terms

Algorithms, Human Factors

Keywords

Interface personalization, e-commerce, temporal demand models, particle filters, interface cost models, online grocery shopping.

1. INTRODUCTION

The grand vision of personalization is to be able to automatically optimize the layout and features of a computer-based user interface to improve a user's subjective perception of the interface and/or to increase an objective measure of the user's ability to use

the interface to complete a task. Automatic personalization frees the user from the need to know what optimizations are possible and from the burden of learning how to express these optimizations. It can even meet needs users have not consciously realized they have. Finally, automatic personalization that successfully anticipates the user's needs can make the user feel valued and respected by the interface and the vendor. While explicitly eliciting user needs can be useful, the concept of automatic personalization achieves its most effective form when user needs can be unobtrusively inferred from past interactions of the user with the interface or prior information gained through other systems.

Technologies such as personalized interfaces, also known as adaptive or dynamic interfaces, and the related field of ubiquitous computing bring technologies into increasingly intimate areas of users' lives and therefore need to be considered in the context of a broad set of issues. The notion of a user interface itself has vastly increased in scope from the traditional display and keyboard, to personal digital assistants or cell phones, and to ubiquitous technologies such as radio-frequency ID tags [16]. The notion of personalization itself can be understood in different ways. We can personalize the appearance, size, and manipulation features of an interface – technologies that have roots in studies of interface usability and menu design [5]. The automatic formatting of content to suit different interface technologies is an important example of this approach. A generic planner can be given a formal description of the functions of interface components, generic background knowledge about interface layout, and the user's functional requirements and current platform in a formal language. If possible, the planner then generates an interface that satisfies the requirements. This paradigm has been employed for control of AV equipment [7]. More recently, a similar concept has been used in a broader vision of having personal digital assistants (PDAs) dynamically download interfaces from appliances like copiers, fax machines, etc. [17]. The emphasis in these models is on satisfying a formal specification.

We can also personalize the content provided by an interface such as an information retrieval system or a stream processing system for email or news [2] [14] [20]. Researchers have also investigated infrastructure issues such as resolution of user identity from ambiguous IP addresses.

Beyond the technological issues, there are larger societal considerations to be taken into account. Users are increasingly concerned about privacy [25]. Will information unobtrusively collected by one entity be communicated to others? Despite emerging legal protections, enforcement of privacy policies and detection of breaches to privacy policies remain challenging

problems. Of course, the social context of users can also be used in positive ways such as explicitly using trust networks to produce better recommendations [18].

While the scope of research into personalization has expanded tremendously, core problems remain in what is variously called preference elicitation, user profiling, or user modeling, and in creating a framework for expressing possible personalizations. Presently, an entirely generic framework for personalization does not appear to be feasible. We have, therefore, decided to focus on personalization in a specific domain with rich information content and frequently repeated but subtle behaviors: online grocery shopping. Specifically, we investigate how to formalize an ongoing user interaction with a grocery shopping interface that adapts both between sessions and during a single session to update its beliefs about the consumer's current demands for products. In the context of grocery shopping, we consider consumers that make weekly shopping trips, charged to a regular account, which eliminates the problems of user identification and log analysis.

The issues that make the problem interesting are that (1) stores are increasingly stocking a huge inventory of items (50,000 in a large grocery store) of which a consumer might purchase fewer than 50 during a single shopping trip, and (2) the consumer makes repeated purchases over time. The substantial size of the inventory precludes communication of the entire catalog to the consumer, who will only select a minute fraction of the items anyway. The repeated purchases aspect introduces the possibility for both the consumer and the vendor to learn about each other and use this learning to adapt to each other.

Unlike the problem of long-term menu optimization, the user's demands are highly correlated and can change rapidly over time. Correlations can be complex, even during a single transaction (e.g., we may observe many customers purchasing related items such as spaghetti sauce, noodles, garlic, red wine, and stain remover on a single trip). This analysis over joint purchase properties is commonly done in market basket analysis in data mining. However, groceries also have *temporal* dependencies. Past purchases influence present ones through stocking and inter-product consumption. Traditional approaches can result in many complex rules. Inferences about a consumer's demand profile can also change rapidly in real time as he or she shops. Informative purchases can change our beliefs about what else a consumer will purchase during the same trip. We will have more to say about the nature of the user modeling problem below. For now, we simply wish to underline the complex dependencies and dynamic optimization inherent in the problem.

We introduce two main ideas in this paper. First, we describe a novel framework for modeling household demand that is capable of representing rich behaviors while maintaining tractable inferences. Second, we reconceptualize the problem of personalization in terms of a novel bi-directional communication model. Both concepts are illustrated with simple domains.

2. CONSUMER MODELS

The key to personalizing customer interfaces is to accurately predict the preferences of individuals and households. Consumer preferences have been analyzed at a variety of levels and in various dimensions by many research communities. Early work in modeling and eliciting preferences for individual products can be found in the marketing literature. Marketing has focused, more so than other disciplines, on measuring the effects of various combinations of product attributes on consumer demand through

techniques such as conjoint analysis (e.g., [9]). Recent extensions, such as fast polyhedral conjoint estimation [24], use interactive computer programs to maximize the efficiency of one-on-one preference elicitation. These techniques make explicit queries to the user. The data-mining and web-mining communities have been more focused on estimation of preferences from *passively* obtained transaction data. The term "user profiling" is more common. These communities have brought insights from the machine-learning and data-base communities to create high-throughput, automated and semi-automated techniques for analyzing and exploring very large databases. Market basket analysis techniques [1] allow one to efficiently extract patterns of items that frequently appear together. Under reasonable assumptions, these correlations can be used to infer demand for one alternative given observed choices for correlated alternatives. Most research to date has favored the discovery of within basket patterns in data, though interest in temporal aspects appears to be growing (see [13]).

Interest in modeling temporal patterns has been a focus of several disciplines. In marketing, techniques borrowed from the health sciences and the reliability community (i.e., survival or reliability analysis) have been used to estimate consumer demand as a function of time and promotional options. In economics and finance, considerable sophistication has been achieved in fitting time series models to data (ARMA, GARCH, etc. [3]). A number of prominent problems in computer science have similar features, including speech recognition, video sequence processing, gene sequencing, and protein prediction. Techniques based on probabilistic models such as hidden Markov models (HMMs) [19] and sequential decision processes [22] are popular.

"Stream processing" [20] is also a rapidly growing field of analysis. The goal here is to analyze large collections of continuously generated items such as messages, news items, e-mails, or stock reports over time and provide filtered reports to a user in a continuous fashion. The roots of this field can be found in work such as the GroupLens project [14] that analyzed news-net messages. Current applications include Google news.

2.1 Structured Models

Grocery purchase patterns have a rich structure that can be exploited in a number of ways. First, grocery purchases exhibit both patterns over time and within-basket. Temporal patterns are evident in purchases of items such as laundry detergent, which occur at regular, predictable intervals. Within-basket patterns can be seen in many items such as the simultaneous purchase of noodles and spaghetti sauce. We also, however, have patterns that are both temporal and cross-sectional. Purchases such as ketchup and hotdogs are correlated in both time and within basket. Our beliefs about the consumer's demand for ketchup will be influenced both by prior belief about the average inter-purchase interval for ketchup, but also by our knowledge that ketchup consumption tends to increase with hotdog consumption. Thus, during summer BBQ season, an increase in hotdog consumption would allow our system to learn to predict shorter inter-purchase intervals for ketchup.

Secondly, the dynamics of consumer grocery purchasing derive in part from the functional structure of household consumption and purchasing. If we assume a general time-series model, we must learn things we already know (e.g., the fact that the unused portion of a household's stock of ketchup persists over time). A highly structured model that integrates such knowledge has the potential to learn complex behaviours with much less data.

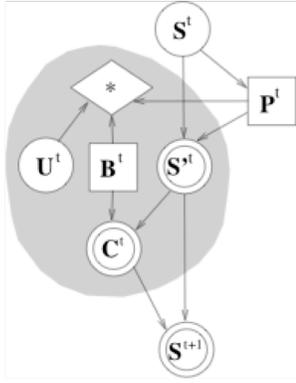


Figure 1: Household Decision Process

In our structured model, we view the household behavior as being organized into a sequence of intervals (we will think of these as “weeks”). At the beginning of interval t , the household starts with a stock of items S^t . This stock S^t is a vector giving a real-valued quantity or amount of each possible item currently on hand in the household. The household then makes a joint stochastic decision of what bundles B^t to consume and what items P^t to purchase. Each bundle corresponds to a meal plan. A “spaghetti dinner” bundle might include noodles, sauce, bread, garlic, salad, and red wine. The bundle consumption decision B^t is a vector of integers representing the number of times each bundle type b_k will be consumed during the interval. We emphasize that consumption is determined in terms of bundles, each of which commits the household to consuming a collection of underlying items. We note that these bundles can be used in mixtures in much the same way that Hofmann defines aspects [11]. The purchase decision P^t is a vector of integers representing the number of units of each item that the household has decided to purchase at the beginning of the interval. In contrast to consumption, purchase decisions are in terms of individual items. In our spaghetti dinner example, the household might have noodles and wine on hand. The household might therefore decide to purchase sauce, bread, and salad to complete the ingredients required for the spaghetti bundle. The household makes a joint decision (B^t, P^t) in order to maximize its utility.

A number of non-linear constraints are required to describe this process. First, stock levels, consumption amounts and purchase quantities are always positive. An obvious corollary is that consumption is bounded above by the sum of prior stock and present purchases. Second, we assume that the actual consumption decision is the result of a non-linear utility maximization process. The utility function of the household is described by parameters U^t . This utility function is the place where individual behavior is represented and thus constitutes our user model for the purposes of personalization.

We use a non-compensatory Cobb-Douglas style utility function to represent the household’s utility, where U_k^t is a scalar parameter for the k^{th} bundle. The non-compensatory aspect ensures that household consumption must be balanced among items. One cannot make up for the need for bread by purchasing additional laundry detergent. The form of the model appears in equation 1.

$$U(B^t) = \prod_k (B_k^t)^{U_k} \quad (1)$$

Presently, we use a simple greedy planner to optimize the bundles that should be selected under the utility function. We believe this constitutes a reasonable model of a household’s decision process. It repeatedly selects the bundle with the highest marginal utility until a sufficient number of bundles have been generated to cover the total household consumption. Total household consumption is a free parameter that can also be estimated from the data and represents an additional parameter through which the model may be personalized.

The structured aspect of the representation comes from the fact that the update of the household stock process is deterministic given the household’s decisions about which items to purchase and which bundles to consume. A resource matrix κ can be used to specify the ingredients of bundles. (Each row is identified with some “meal”, and each column with an ingredient, so that each entry k_{ij} specifies how much of ingredient i appears in bundle j .)

The consumption during an interval C^t is therefore the resource requirements κ times bundle choices B^t , or $C^t = \kappa B^t$. The final stock of the household at the end of the interval is the original stock S^t plus purchases P^t minus consumption κB^t , or $S^{t+1} = S^t + P^t - \kappa B^t$. A graphical view of the model appears in Figure 1.

Actual purchases and consumption may differ due to events in the household or spontaneous shopping behavior. We have discovered that the addition of independent additive Gaussian noise to the purchases and consumption allows us to maintain an efficient algorithm for updates while allowing for random departures from deterministic behavior. The noise must be added carefully to preserve invariants (household stock, purchases and consumption are always positive and should be consistent with the update rule).

The household decision process model leads naturally to a Markov-chain-like structure, which models a consumer’s purchase history. The stock state of the household is the latent state of the process. The utilities become model parameters. The household’s purchases are the observables. Nominally, we can use standard approaches to maintain a distribution over hidden states given observables (e.g., Kalman filters, see p. 551 of [21]). However, because the distribution of purchase probabilities is created by a complex non-linear optimization process, it is difficult to capture in terms of parametric forms. Researchers have attempted to address this issue through techniques such as mixtures of Gaussians, truncated exponentials, scented filters and other forms. In this work, we elect to employ particle filtering [6], a non-parametric representation of the distribution over possible household states.

In particle filtering, we maintain a sample of households – each with its own stock state. We then calculate the likelihood of each household given the observed purchases. The samples are then weighted according to these likelihoods. The process is illustrated in Figure 2. From these weighted samples we can compute statistics such as the expected demand for each product that will be purchased. When normalized, we can use these weighted samples as a probability distribution from which to choose new households for the next stage. From the vendor’s perspective, the household decision process is a stochastic process whose parameters can be estimated. Unlike an ARMA model,

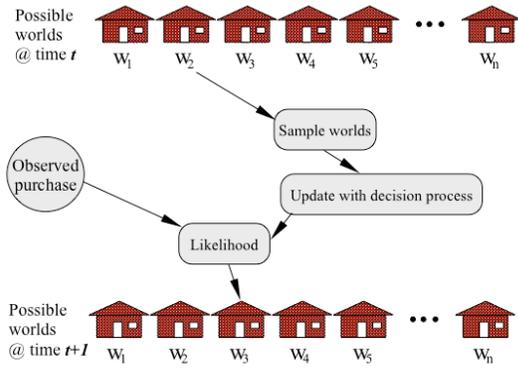


Figure 2: Particle Filtering Update

however, the model explicitly encodes the structure of the household’s decision making.

To explore the properties of the structured temporal-cross-sectional models, we have performed some simulations with synthetic data. In these studies we have predefined a set of consumption bundles. In the simulation reported here, there were 5 possible consumption bundles each consisting of 2 out of 5 possible ingredients. The bundles, along with some arbitrarily chosen utility assignments appear in Table 1.

Table 1 Examples of Bundles

2 Item Bundles	Ingredients	Utils
Peanut butter sandwich	0.05 PB 0.1 bread	0.3
Banana sandwich	0.2 Banana+0.1 bread	0.3
Tomato sandwich	1.0 Tomato+0.1 bread	0.7
Jam sandwich	0.15 Jam + 0.1 bread	0.3
Wheaties breakfast	0.15 wheaties+0.25 milk	0.7

To test the model, we used the given utility parameters to simulate an artificial household purchase history. We then fed just the observed purchases to a consumer utility inference mechanism based on the process model and particle filtering approaches described above.

Figure 3 shows that the inference mechanism is able to recover the relative utility assignments given by the household to various

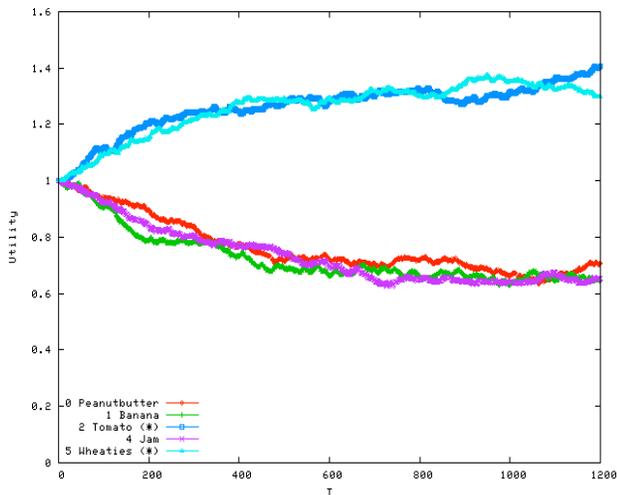


Figure 3: Inferred Bundle Utilities

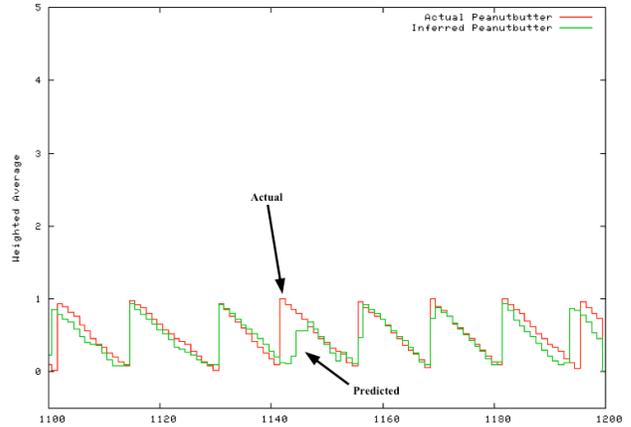


Figure 4 Actual vs. Inferred Peanut Butter Purchases

consumption bundles using only the observed purchases. Of course, modeling a realistic store inventory would require hundreds of bundles for each consumer. We present this simple illustration to highlight how the household process model intuitively captures purchasing behavior.

Accurate estimates of utilities allow the system to infer the rate at which various items are consumed within the household. Due to random noise, the model’s predictions can be off. Figure 4 shows the actual stock level of peanut butter in the simulated household together with the predicted level of peanut butter (weighted average of all household particle predictions). Generally, the model tracks peanut butter consumption well. At some points, the model fails to predict the true stock level. Eventually, however, those household particles that better explain observed purchases will be re-weighted more highly and the prediction will again start tracking actual consumption levels. Note, as seen in, Figure 4, it is possible for the model to recover from tracking errors before actually observing a purchase of the item by making inferences from other correlated items such as bread and bananas.

In Figure 5, we see the household’s actual purchases, together with the purchase predictions generated from the stock level model. We see that when stock is predicted to fall below consumption needs in Figure 4, a purchase is predicted in Figure 5. Where consumption predictions were off in Figure 4, purchase predictions were off in Figure 5. Ultimately, the erroneous purchase predictions drive the process back to the proper state.

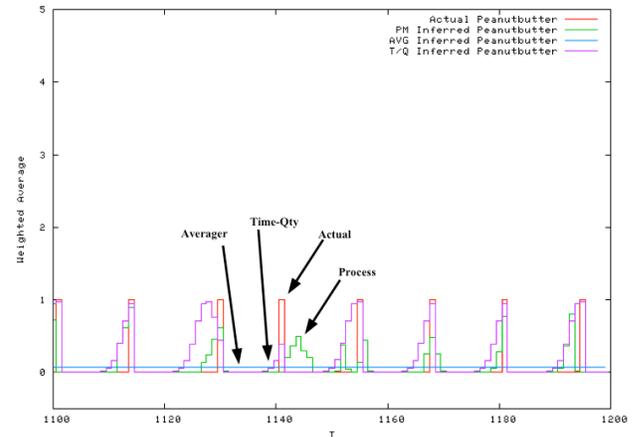


Figure 5 Actual vs. Inferred Peanut Butter Purchases

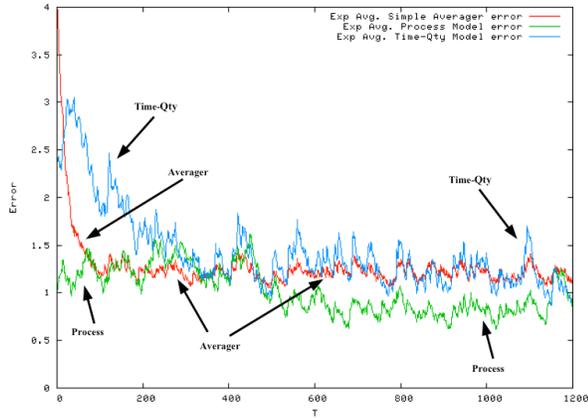


Figure 6 Prediction Error for Various Models

The overall accuracy of purchase predictions is the most important measure of the model’s capability. We have measured accuracy as the Euclidean distance of the point representing the predicted purchases at time t to the actual household purchases at time t .

In Figure 6, the performance of several different prediction techniques are compared. The first technique is a “simple averager.” It computes the average amount purchased on each time interval and uses this average as its prediction for the next time interval. So for instance, a household that buys laundry detergent every 4 weeks would have a 0.25 chance of buying detergent every week under this model. The second model is the time-quantity model. This style of model has been used in Marketing [10] to assess the sensitivity of purchase timing to promotions. The time of the purchase and the quantity to be purchased at this time are estimated separately and then multiplied to get an expected quantity. This model gives improved estimates in theory as it would predict a high probability of laundry detergent around week four and lower estimates for other weeks. When there is significant variance in the timing of purchases, however, the predictions tend to be smeared out over several time steps and accuracy suffers. The final model is our structured temporal-cross-sectional process model. It is able to use its household model to track purchases even if they do not occur with perfect inter-purchase regularity (often the case due to least common divisor type interactions between consumption rates of various ingredients). It can beat both other models on this simple problem.

The utility parameter inference provided by process models therefore seem to be a promising technique for characterizing purchasing behaviour of consumers. Furthermore, the output of the particle filter has a clean and intuitive interpretation. The likelihood of each household particle can be used to assign a probability to each particle’s purchase prediction forming a probability distribution $\Pr(D|H)$ over product demands D conditioned on the user’s purchase history H .

3. ADAPTATION OF INTERFACES

In the previous section, we introduce a rich representation for consumer purchase behaviors. In this section, we show how one can use a model of the form described above to drive the generation of a personalized user interface.

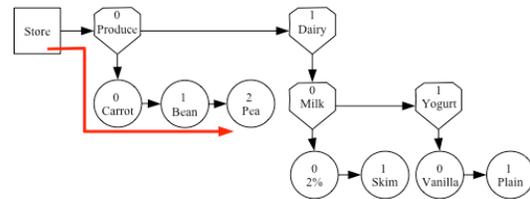


Figure 7 Code corresponds to tree branch indexes

A classic approach to creating flexible user interfaces is to adopt a restricted family of interface layouts and then choose a layout within this family. A common choice is to employ a tree-structured directory hierarchy. In one example [23], a tree structure provides access to web resources on mobile phone displays. Popular nodes are promoted up the hierarchy in the tree to reduce users’ access cost. The optimization is based on a weighted metric that measures distance to items based on the number of selections and amount of scrolling required to access items. Relevant work has also been done in optimization of hierarchical computer menus [15].

Using a communications channel analogy, we could recast the problem of trying to minimize the effort for users to find desired products (eg, with a minimum expected number of keystrokes) as

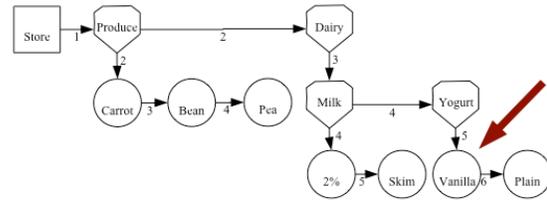


Figure 8 Reading and navigation costs given by tree

the problem of constructing an information channel with an optimal code to express the desired messages. Specifically, the sequence of choices the user makes as they descend a tree-structured hierarchy correspond to sending successive digits in an n -ary code. In figure at the top, “Peas” have the code “02” and plain yogurt “111”.

The navigation pattern imposes a cost on the user. In the figure above, a simple cost model assigns costs to reading and link following. Reaching “plain yogurt” has a cost of “6”. We have used simple and equal integer costs here for clarity. The model, however, admits a general cost function.

One can easily optimize the tree to minimize the cost to obtain “plain yogurt”, however, the cost of all other items will increase. The best organization to use in personalizing the interface for a given consumer is not obvious and will depend on the user’s demand distribution, $\Pr(D|H)$. We note that the demand distribution, $\Pr(D|H)$, corresponds to the message transmission frequency in coding theory. Given transmission cost and

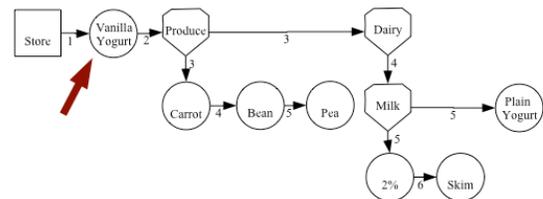


Figure 9 Costs in yogurt optimized tree

transmission frequency, Huffman [12] provides an efficient optimal method for finding codes spelled with n-ary digits. The original work, however, assumes that all digits have equal transmission cost. This assumption fails to hold in our navigation problem as options with higher branch numbers require additional reading (and possibly interface manipulation) to reach. This problem is called a code with unequal letter costs and is known to be solvable approximately for useful cases [8].

So far, we have only considered communication from consumer to vendor. It captures the behaviour of a user who has a specific grocery category in mind (e.g., “peas”) and navigates through the hierarchy to find it. However, the user must go from an abstract category to a specific stock keeping unit or SKU (*i.e.*, a box with a barcode on it). So, the category “peas” might be refined by brand, format, and organic status to become “Everfresh organic frozen peas, 2kg bag”. To perform this specialization, the user needs to scan the SKU’s related to the category of interest.

Table 2: Greedy Directory Tree Generation

```
Greedy_generate(Items,D)
    Return Greedy_generate_helper(makeTree(Items),D)

Greedy_generate_helper(T,D)
    [ a* , T* ] = max_{a ∈ A} { EC(partition(T,a),D) }
    R = new_root_node()
    foreach n in T*
        R += greedy_generate_helper(n,D)
    Return R
```

In our coding analogy, this means that SKU’s which are related should have similar prefixes in the code so that they will fall under the same branch. We are unaware of specific extensions to coding theory to this. We therefore extended coding to a new paradigm we call “coherent coding” which enforces the idea that messages with related content should have codes with similar prefixes.

In Table 2, we present a high-level outline of the core ideas for an algorithm to compute coherent codes. We start with a flat tree T containing all products. Given the user demand distribution p , we attempt to partition the nodes of the tree using

an attribute to create a new tree structure T' so as to minimize the expected cost of finding each item, $EC(T', p)$. Intuitively, expected cost will be minimized by the attribute that separates items likely to be demanded by the consumer from items unlikely to be demanded by the consumer. For example, some consumers might find the attribute “organic” to be a useful separator, while other consumer’s might find an ethnic category more useful. In our implementation, we also allow promotion of highly demanded nodes to rise above the split as in [23]. The algorithm also has the important property that the resulting directory tree is always complete in the sense that all items are always accessible.

To test the algorithm, we developed a simple simulation scenario. We simulate a store with a 5,000 item inventory. Each item has 10 attributes (weight, brand, etc). Each attribute has 5 levels (e.g., 100g, “Fresh Pick”). We define a spread parameter S which gives the spread of products over the attributes. The usefulness of attributes for partitioning products will depend on how the products are spread over the levels of the attribute. If the products are spread out over all levels, the attribute will be informative (e.g., grocery department). If the products are concentrated on one level, the attribute will be uninformative (e.g., all products in store are non-organic). We generated 1,000 customers with randomized demand functions. Each consumer randomly chooses an item to request from the store. Two stores are created: one which is personalized for the user’s personal demand distribution, $Pr(D|H)$ in as much as attributes allow it and one that is optimized to the average demand distribution of the 1000 customers in our sample. In Figure 10, we can see that when attributes are highly discriminative ($S=1/5$), savings in navigation time created by personalization are about 81 percent. When attributes are completely uninformative ($S=5/5$), there are no savings, as expected. This simple scenario is intended only to give the reader a qualitative understanding of how interface personalization can be used to reduce user communication overhead.

4. FUTURE WORK

A key goal for us in the coming months is to apply our framework to some real-world data. As part of this project we are implementing automatic bundle discovery. Beyond immediate applications, the grocery domain introduces a number of challenging properties into consumer modeling and interface generation. Products can be substitutable and complementary. Optimizers must be able to recognize this so that demand for 2% milk is not treated independently of demand for 1% milk. Similar reasoning is required to address multiple quantities of the same item. The number of samples for any one consumer is small, and thus information about the population must be used to give us information about the individual. Techniques such as shrinkage and Hierarchical-Bayes estimation can be used here. We are also in the process of comparing our techniques to traditional ARMA style time series analysis.

Another key problem for this area of research is defining measures of customer satisfaction. Simple matching of the predicted demand to actual demand seems problematic for a number of reasons. First, satisfaction is asymmetric. One is better off to occasionally meet a user’s needs in the excess than to undersupply a need. Traditional matching measures also introduce subtle biases that skew the notion of closeness of two time series in unintuitive ways. For instance, two time series that are identical but shifted in phase would show high absolute pair-wise error, whereas we would want this to have low error.

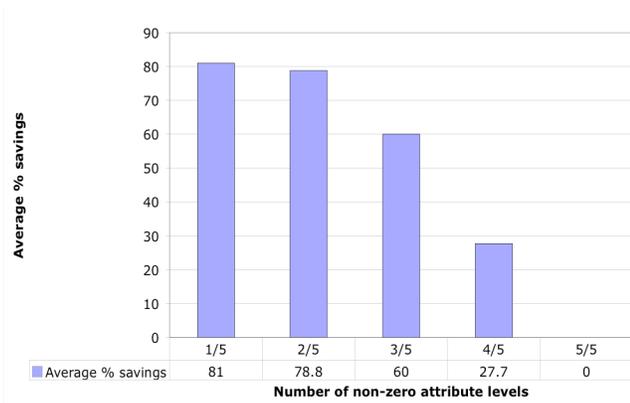


Figure 10: Product Search Savings as a Function of Product Distribution over Attributes

5. CONCLUSIONS

Grocery shopping poses special challenges for personalization. It involves complex inter-temporal and inter-item correlations. It also requires unobtrusive and incremental real-time analysis. Our novel approach to consumer modeling leverages a household behavior model to achieve tractable incremental inference of these complex correlations using only past observations of shopping behavior. The inference uses household utility parameters to represent preferences. The preferences generate a demand model which is then used to personalize the interface. Finally, we showed that this personalization can be implemented using channel coding concepts to optimize an expressive family of interfaces and that these optimized interfaces have the potential to reduce objective interface manipulation costs for the user.

6. ACKNOWLEDGMENTS

We would like to thank The Alberta Ingenuity Center for Machine Learning, The National Science and Engineering Research Council of Canada, and The Social Science and Humanities Research Council of Canada for funding this research. We would also like to thank Terry Elrod and Peter Popkowski-Leszczyc for their helpful advice on this project.

7. REFERENCES

- [1] Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules *Proc. 20th International Conference on Very Large Data Bases (VLDB)*. 1994
- [2] Billsus, D., Hilbert, D.M., and Maynes-Aminzade, D. Improving Proactive Information Systems, *International Conference on Intelligent User Interfaces 2005*
- [3] Bollerslev, T. and Engle, R.F. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307-327, 1986.
- [4] Boutilier, C. A POMDP Formulation of Preference Elicitation Problem, In: *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, Edmonton: AAAI Press, 239-246. 2002
- [5] Card, S., Moran, T.P., and Newell, A. *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: L. Erlbaum Associates, 1983
- [6] Doucet, A., de Freitas, N., and Gordon, N. *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag. 2001
- [7] Gajos, K. and Weld D. SUPPLE: Automatically Generating User Interfaces. International Conference on Intelligent User Interfaces. Island of Madeira, Portugal. 13-16 Jan 2004.
- [8] Golin, M. J., Kenyon, C., and Young, N. E. Huffman Coding with Unequal Letter Costs, *34th ACM Symposium on Theory of Computing*. 2002
- [9] Green, P. E. and Srinivasan, V. Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice, *Journal of Marketing*, 54, 4, (1990) 3-19.
- [10] Gupta, S. Impact of Sales Promotions on When, What, and How Much to Buy, *Journal of Marketing Research*, Vol. XXV (November 1988), 342-55.
- [11] Hoffmann, T. and Puzicha, J. Latent Class Models for Collaborative Filtering. *Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July 31-August 6, 1999
- [12] Huffman, D. A. A method for the construction of minimum-redundancy codes, *Proceedings of the I.R.E.*, Sept. 1952, 1098-1102.
- [13] ICDM 2004 Workshop on Temporal Data Mining: Algorithms, Theory and Applications November 01-04, 2004, Brighton, UK.
- [14] Konstan, J. A., Maltz, D., Herlocker, J.L., Gordon, L. R. and Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News, *Communications of the Association for Computing Machinery*, 40, 3, (1997) 77-87.
- [15] Lane, D.M., Napier, H.A., Batsell, R.R., and Naman, J.L. Predicting the Skilled Use of Hierarchical Menus With the Keystroke-Level Model *Human-Computer Interaction*, Vol. 8, No. 2 (1993), Pages 185-192
- [16] Loebbecke, C. (2005) Emerging Information Systems Applications in Brick-and-mortar Supermarkets: A Case Study of Content Provision Devices and RFID-Based Implementations, *Pacific Asia Conference on Information Systems (PACIS)*, Bangkok, Thailand, July.
- [17] Nichols, J, and Faulring, A. Automatic Interface Generation and Future User Interface Tools *ACM CHI 2005 Workshop on The Future of User Interface Design Tools*, Pittsburg.
- [18] O'Donovan, J. and Smyth, B, *Trust in Recommender Systems* International Conference on Intelligent User Interfaces, San Diego, California, January 9-12, 2005
- [19] Rabiner, L. R. and Juang, B. H. An introduction to hidden Markov models, *IEEE ASSP Magazine*, January 1986, 4-15
- [20] Riabov, A. V. and Liu, Z. Planning for Stream Processing Systems. The Twentieth National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania July 9-13, 2005.
- [21] Russell, S., and Norvig, P. *Artificial Intelligence: Modern Approach Second Edition*, Pearson Education Ltd. Toronto, 2003.
- [22] Shani, G., Heckerman, D., and Brafman, R.I., An MDP-Based Recommender System, *JMLR* 6 (Sep): 1265-1295, 2005
- [23] Smyth, B. and Cotter, P. Intelligent navigation for Mobile Internet Portals, *IJCAI Workshop on AI Moves to IA: Workshop on Artificial Intelligence, Information Access, and Mobile Computing. The 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico. 2003
- [24] Toubia, O., Duncan S., Hauser, J., and Dahan, E. Fast Polyhedral Adaptive Conjoint Estimation, *Marketing Science*, 22, 3, (2003) 273-303.
- [25] Volokh, E. Personalization and Privacy, *Communications of the ACM*, Volume 43, Number 8 (2000), Pages 84-88.