Optimal Recommendation Sets: Covering Uncertainty over User Preferences

Bob Price
Department of Computer Science
University of Alberta
Edmonton, AB T6G 2E8 Canada
price@cs.ualberta.ca

Paul R. Messinger
School of Business
University of Alberta
Edmonton, AB T6G 2R6 Canada
paul.messinger@ualberta.ca

Abstract

We propose an approach to recommendation systems that optimizes over possible sets of recommended alternatives in a decision-theoretic manner. Our approach selects the alternative set that maximizes the expected valuation of the user's choice from the recommended set. The set-based optimization explicitly recognizes the opportunity for passing residual uncertainty about preferences back to the user to resolve. Implicitly, the approach chooses a set with a diversity of alternatives that optimally covers the uncertainty over possible user preferences. The approach can be used with several preference representations, including utility theory, qualitative preferences models, and informal scoring. We develop a specific formulation for multi-attribute utility theory, which we call maximization of expected max (MEM). We go on to show that this optimization is NP-complete (when user preferences are described by discrete distributions) and suggest two efficient methods for approximating it. These approximations have complexity of the same order as the traditional k-max operator and, for both synthetic and real-world data, perform better than the approach of recommending the k-individually best alternatives (which is not a surprise) and very close to the optimum set (which is less expected).

Introduction

Recommender systems can help people select from large numbers of alternatives. Approaches to recommender systems include content-based systems (Basu et al. 1998), collaborative filtering (Konstan et al. 1997), hybrids (Popescul et al. 2001), case-based retrieval (Bradley and Smyth, 2001), and qualitative preference inference (Faltings et al. 2004). Some of these systems rely on explicit preference elicitation, which has been studied in computing science (Boutilier 2002) and in marketing science as conjoint analysis (Green et al. 1990) and stated-choice models (Louviere et al. 2000).

Each of these systems relies on some kind of representation of users' preferences. Since information about users is incomplete and uncertain and since preference representations are often based on simplified models of users and alternatives, the single best alternative for the user is uncertain. The common solution to this problem is to present the user with a set of the top-k alternatives (typically $k \leq 10$) in the hope that one of these alternatives will be the true best.

Unfortunately, providing the top-k alternatives may not provide for diversity in the recommendation set. Imagine that a user is seeking a book from an online retailer, and that the retailer has used elicitation or data-mining to determine that the user is interested in one of its large number of "machine learning" texts. Using a utility framework, the retailer might recommend the top-k books ranked highest according to a measurement scale consisting of the user's utility function. Using a qualitative preference model, the retailer might recommend a set of books most consistent with the user's stated preference constraints according to a measure of the extent of constraint violations. Neither approach explicitly takes into account diversity in the set of possible user preferences. And for both approaches, one could easily end up with a recommendation set containing ten editions of the same book. The problem is that including an alternative in a set that already contains similar alternatives does not add potential value for the user. Instead, an ideal text recommendation set would incorporate diversity in price, technical depth, availability, number of pages, authorship, etc. This diversity should be tailored to the system's uncertain prior beliefs about the user. For example, if most users are non-experts, then most of the selection should be tailored to non-experts.

Related problems arise in collaborative filtering. Here, the user is matched with the previously encountered users with the most similar profiles of past choices. Typical nearest neighbor approaches do not explicitly take into account possible diversity of the selected neighbors or their past choices. The optimal recommendations should again be chosen to cover this diversity.

The simplest way to provide diversity in the recommendation set is to suppress duplicated alternatives. In the book-selling context, class reasoning could be used to recognize that different editions are equivalent. Class reasoning, however, does not handle partial similarity.

Alternatively, we could apply a dissimilarity heuristic to ensure that the recommendation set does not become degenerate (*e.g.*, Bradley and Smyth 2001). Dissimilarity measures guarantee that the set contains alternatives that differ in attributes, but these measures do not take into account partial knowledge about the user's preferences. As a result, diversity in the recommendation set will not be tailored to individual users. In any case, the use of a diversity heuristic may push some of the highest-ranked alternatives out of the set resulting in the need to make a difficult tradeoff between including alternatives that are preferred *a priori* to those that provide diversity.

Our main contribution is to show that recommendation sets can be constructed optimally using a simple formulation of the expected value of a user's *choice from a set* and that the resultant set optimally matches the diversity of alternatives in the recommendation set to the uncertainty present in the system's prior beliefs about the user's preferences.

In the remainder of the paper, we review paradigms for representing uncertain preferences; we propose specific optimization criteria to implement the notion of covering preference uncertainty; we provide an extended illustration of such an optimization; we show that one formulation of this optimization problem is NP-complete; we introduce two algorithms that approximate this optimization, and we apply these algorithms using synthetic and real-world data to explore the extent of the gain from our approach.

Uncertain Preference Representations

We consider a user facing a choice from a large set of alternatives $S = \{1,\ldots,n\}$. To help find the best alternative, the user consults a recommender system that makes use of information about the user's preferences. This information can be represented in a variety of forms.

In multi-attribute utility theory (MAUT), the utility of an alternative i is represented by a function u of the attributes of the alternative, described by the vector $\mathbf{a}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,m}]$ (for the textbook example, attributes might include price, technical depth, and number of pages). The framework is very flexible: several representative classes of utility functions appear in Table 1. Utility functions with complex dependencies can also be described by explicit conditional functions represented by a graph structure (Boutilier et al. 2001).

Uncertainty in the MAUT framework can be described by a distribution $p(\boldsymbol{\theta})$ over parameters $\boldsymbol{\theta}$ of a family of utility functions $u(\mathbf{a}_i, \boldsymbol{\theta})$. Often the distribution can be factored into per coefficient components (*e.g.*, $p(\boldsymbol{\theta}) = \prod_j p(\theta_j)$)

Table 1. Representative utility functions

$u(\mathbf{a}_i, \boldsymbol{\theta}) = \theta_1 a_{i,1} + \cdots + \theta_1 a_{i,m}$	Linear model
$u(\mathbf{a}_i, \boldsymbol{\theta}) = \sum_j \theta_j (a_{i,j} - \mu_j)^2$.	Ideal-point model
$u(\mathbf{a}_i, \boldsymbol{\theta}) = \theta_1 a_{i,1} a_{i,2} a_{i,3} (1 - a_{i,1})$	Dependencies and correlations
$u(\mathbf{a}_i, \boldsymbol{\theta}) = \log(a_{i,j})$	Diminishing utility

Elicitation in the MAUT framework makes use of Bayesian techniques to integrate prior beliefs about $\boldsymbol{\theta}$ with information obtained subsequently from explicit preference elicitation or data-mining of past choices. Coefficients can be estimated directly from numerical user preference ratings[1] analyzed with (conjoint) regression techniques or indirectly from users' choices from alternatives described by attribute "profiles"[2] analyzed with logit or probit (choice) models. The latter (choice modeling) eliminates the need for a user to assign explicit numerical ratings to alternatives.

Recommendation in the MAUT framework has traditionally used expected utility. One calculates the expected value of each alternative under all possible utility parameter realizations (for the discrete distribution case, $EU(\mathbf{a}) \equiv \sum_t u(\mathbf{a}; \boldsymbol{\theta}_t) \Pr(\boldsymbol{\theta}_t)$) and chooses the k-highest expected utility alternatives.

Recently there has been a resurgence of interest in qualitative preference representation. *Ceteris paribus* reasoning (Boutilier et al. 2004) allows the user to express preferences through comparisons of specific alternatives or properties of these outcomes conditional on other aspects of the world. Graphical representations with clear semantics (*e.g.*, CP-nets) make it easy to decompose complex dependencies. The avoidance of specific numerical probabilities can simplify elicitation.

Algorithms have been developed to determine non-dominated sets of alternatives in CP-nets without explicit use of utility functions. If no recommendation satisfies the user's preferences, recommendations can be made by choosing alternatives that minimize some function of preference violations (Faltings et al. 2004). Uncertainty in qualitative preferences, represented by missing preference information, can lead to large increases in the size of the set of non-dominated alternatives. When the set is too large to directly display, recommendations can be made by randomly sampling from this set (*ibid.*).

[1] An example of a numerical preference rating might be "rate the importance of battery life from 1 to 10."

[2] An example of a choice set consisting of two attribute profiles might be
(1) Long battery life, built in networking, small display;
(2) Short battery life, an expansion slot, large display.

Formalizing Preference-Covering Sets

We now formalize our intuition about covering preference uncertainty. We focus on covering algorithms using the multi-attribute utility theory framework, because MAUT is currently the most widely-used framework for uncertain preference representation in economics, marketing, finance, medicine, and other fields. We comment briefly on covering set approaches for other preference representations at the end of the section.

Traditional methods rank alternatives individually based on a (one-dimensional) scoring criterion and recommend the set consisting of the k-highest alternatives. We refer to such an approach as a *k-individually best* (KIB) method. For example, in the MAUT framework, the KIB method scores alternatives individually by expected utility.

Note that the KIB method's selection of alternatives individually prevents it from optimizing the diversity of the resulting set. As we argued in the introduction, any system that fails to explicitly optimize diversity in the recommendation set to cover the uncertainty in user preferences will be suboptimal. In stochastic programming, it has long been recognized that optimization in an uncertain environment requires explicit consideration of the uncertainty present in the system (Sengupta 1972). We now formally show how this insight applies to recommendation set generation.

We begin by observing that the expected utility to users of a recommendation set is dependent on the set elements jointly, not on each element individually. We accordingly want to maximize the expected value of a user's selection from a *set M* of *k* alternatives offered jointly.

Suppose that the system knows that each user has a utility function in a known parameterized family. In particular, a user with parameter vector $\boldsymbol{\theta}$ will obtain utility $u(\mathbf{a}_i;\boldsymbol{\theta})$ from an alternative $\mathbf{a}_i$. Such a user will choose the alternative with the highest expected utility, and the value of the user's maximal alternative is $u^*_{\boldsymbol{\theta}} = \max_{i \in S} u(\mathbf{a}_i, \boldsymbol{\theta})$.

For any given user, however, the system does not know with certainty the parameter vector $\boldsymbol{\theta}$; but we assume that the system knows the probability (or probability density) $p(\boldsymbol{\theta})$ that any particular $\boldsymbol{\theta}$ is realized. This distribution integrates prior beliefs, past experiences with the present user or other users in the population, and any explicit elicitation we have been able to perform.

We compute the expected value of the user's *maximal* alternative $u^*_{\boldsymbol{\theta}}$ with respect to our uncertainty about the user's utility function. We call this criterion "emax." Equations (1) and (2) give the discrete and continuous forms of the *emax* criterion, where *M* is the proposed set of alternatives and *p* is the distribution over the utility parameter vector $\boldsymbol{\theta}$.

$$\text{emax}(M,p) = \sum_t \max_{i \in M}\{u(\mathbf{a}_i;\boldsymbol{\theta}_t)\}\, p(\boldsymbol{\theta}_t) \qquad (1)$$

$$\text{emax}(M,p) = \int_{\boldsymbol{\theta}} \max_{i \in M}\{u(\mathbf{a}_i;\boldsymbol{\theta})\}\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta} \qquad (2)$$

In this paper, we assume that, given a set of alternatives, the user can determine the utility of each alternative and normatively chooses the best one. The expected value of the user's choice from a set *M* is, thus, the expected utility of the maximal alternative in *M* as described by the *emax* criterion. In this context, the *emax* criteria is the best available summary measure (and the correct normative measure) of the utility that a recommendation set is expected to provide users.

We can directly *maximize* the "emax" criterion (continuous or discrete) to find the optimal recommendation set M^*. We call M^* the maximal expected max set, or the MEM set, for short.[3]

$$M^* \;=\; \underset{M \subset S,\, |M|=k}{\text{argmax}} \left[\text{emax}(M,p)\right] \qquad (3)$$

Unlike KIB methods, MEM-set methods directly optimize the normative *emax* measure. Since the MEM set is maximal over all possible *k*-element recommendation sets, including the KIB set, it always has *emax* score at least as high as the KIB set. Therefore, MEM-set methods are normatively superior to KIB methods.

Qualitative preference models are not the focus of this paper, but an analogous formulation can be developed for this case.[4]

---

[3] The argmax operator returns the set of items that maximizes the expression given in its argument:

$$\underset{i \in S}{\text{argmax}}[f(i)] = \{i \mid f(i) = \max_{j \in S} f(j)\}$$

[4] Here we must address two cases. When preferences over-constrain the recommendation set, we use standard qualitative preference concepts to choose alternatives that break the fewest and "least important" preference constraints. Since we are choosing a set, however, we need not include alternatives that break the same preference constraints. Formally, let $\text{vio}(i;\boldsymbol{\theta})$ be the set of preference violations of alternative i in the CP-net with parameter vector $\boldsymbol{\theta}$. Let $\#(i;\boldsymbol{\theta})$ be the count of preference violations for alternative i. Let $\#(M;\boldsymbol{\theta}) = \sum_{i \in M} \#(i;\boldsymbol{\theta})$ be the total violations in set M. We then optimize:

$$M^* = \underset{M \subset S\ s.t.\ |M|=k,\ \forall i,j \in M,\ \text{vio}(i) \neq \text{vio}(j)}{\text{argmin}} \#(M;\boldsymbol{\theta})$$

When preferences are under-constrained, we can select alternatives that increase diversity.

Extended Example

In this section we demonstrate the *emax* and optimal MEM set concepts with a simple example that assumes extreme diversity in consumer preferences.

For the purposes of the example we consider a simple laptop computer market. The i^{th} alternative in the market, $\mathbf{a}_i$, corresponds to a specific laptop configuration. Each laptop alternative is described by a two-element vector of real-valued attributes, $[a_1, a_2]$, representing clock-rate and battery life, respectively. We might imagine that our market contains a high performance laptop, $\mathbf{a}_h = [.05, 1]$, two medium performance laptops, $\mathbf{a}_{m1} = [0.5, 0.6]$ and $\mathbf{a}_{m2} = [0.6, 0.5]$, and a low performance laptop, $\mathbf{a}_l = [1, .05]$. In this representation, each laptop can be viewed as a point in attribute space (see Figure 1).

We assume that the space of user preferences is characterized by two discrete groups of users: "video editors" and "traveling executives." In this example, we assume that video editors want raw processing speed and traveling executives want maximum battery life; we represent these preferences by vectors $\boldsymbol{\theta}_v = [0,1]$ and $\boldsymbol{\theta}_e = [1,0]$ respectively. We represent each preference type as a vector with its tail at the origin of attribute space and its tip at the point given by the $\boldsymbol{\theta}$-vector. By construction, these two preference vectors coincide with the vertical and horizontal axes in Figure 1.

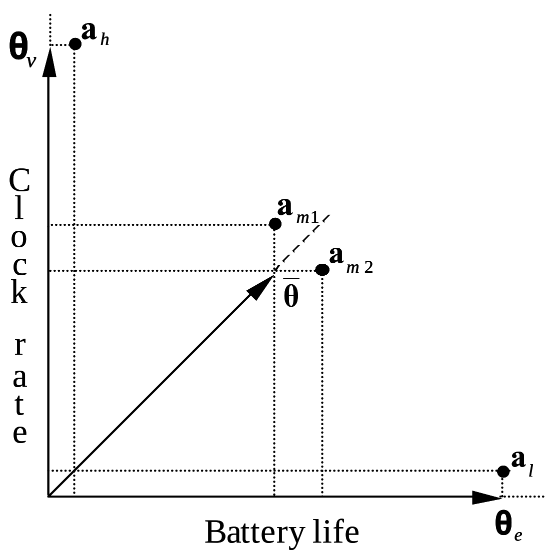

Figure 1. Attribute space showing alternatives as points and user preferences as vectors

We express user t's utility function as a $\boldsymbol{\theta}_t$-weighted linear combination of the alternative's attributes

$$u(\mathbf{a}_i, \boldsymbol{\theta}_t) = \boldsymbol{\theta}_t \cdot \mathbf{a}_i = \theta_{t,1} a_{i,1} + \ldots + \theta_{t,d} a_{i,d} . \qquad (4)$$

Equation 4 can be geometrically interpreted as the unnormalized projection of alternative $\mathbf{a}_i$ onto user t's preference vector $\boldsymbol{\theta}_t$. In Figure 1, we see that the high performance laptop $\mathbf{a}_h$ projects very high on the video editor's preference vector $\boldsymbol{\theta}_v$, but very low on the traveling executives vector $\boldsymbol{\theta}_e$. The situation is reversed for the low performance laptop $\mathbf{a}_l$.

Suppose we would like to construct a two-element recommendation set. A KIB approach would compute the expected value of each alternative and choose the top two.

Let us assume *a priori* that either user type is equally likely, so that $p(\boldsymbol{\theta}_v) = p(\boldsymbol{\theta}_e) = 0.5$. The expected utility of an individual alternative is, thus,

$$\begin{aligned} E_{\boldsymbol{\theta}}(u(\mathbf{a}_i;\boldsymbol{\theta})) &= \textstyle\sum_t (\boldsymbol{\theta}_t \cdot \mathbf{a}_i) p(\boldsymbol{\theta}_t) \\ &= (\textstyle\sum_t \boldsymbol{\theta}_t p(\boldsymbol{\theta}_t)) \cdot \mathbf{a}_i \qquad (5) \\ &\equiv \overline{\boldsymbol{\theta}} \cdot \mathbf{a}_i . \end{aligned}$$

Therefore, the expected utility function of a single alternative is essentially the utility of the alternative for the "average" user with preferences described by $\overline{\boldsymbol{\theta}} = (0.5, 0.5)$. The expected utility of each of our laptops, individually, is as follows:

Alternative	$\mathbf{a}_h$	$\mathbf{a}_{m1}$	$\mathbf{a}_{m2}$	$\mathbf{a}_l$
E.U.	0.525	0.55	0.55	0.525

In contrast to expected utility of individual alternatives, the *emax* criterion is calculated on sets. It provides the expected utility of a user's choice from an alternative set (recognizing possible user types). The *emax* values for three particular two-element sets are as follows:

Alt. Set	$\{\mathbf{a}_{m1}, \mathbf{a}_{m2}\}$	$\{\mathbf{a}_{m2}, \mathbf{a}_h\}$	$\{\mathbf{a}_h, \mathbf{a}_l\}$
Emax	0.6	0.8	1.0
	KIB Set	Greedy Set*	MEM Set

*Discussed later in the section on the GMEM algorithm.

A KIB method, based on individual expected utility, would choose the set of two alternatives that project highest on $\overline{\boldsymbol{\theta}}$, namely $\{\mathbf{a}_{m1}, \mathbf{a}_{m2}\}$. Unfortunately, neither of these alternatives is most preferred for either video editors or traveling executives. Indeed, the KIB method generates the worst possible two-element recommendation set in this example.

The MEM set method chooses the set of two alternatives that directly maximizes the expected utility of the user's choice from that set. In this example, the MEM set will consist of $\{\mathbf{a}_h, \mathbf{a}_l\}$ and will provide each user type with his or her preferred alternative.

In general, a k-element recommendation set permits sufficient diversity of alternatives to allow the recommendation system to provide k types of users with their optimal alternatives. When there are more user types than elements in the recommendation set, the MEM set method will choose sets that optimally substitute more generic alternatives capable of satisfying more than one user type. In either case, the MEM set method will provide an *optimal cover set*.

Computational Complexity of MEM-Sets

It is not obvious that a tractable optimal MEM set algorithm exists. A naïve version of a MEM set optimizer must enumerate all *k*-element subsets of the *n* alternatives requiring $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$ *emax* evaluations. In particular, we show below that the discrete version of the Maximum-expected-max (MEM) set optimization is NP Complete. We give an outline of the reduction from the NP complete k-medians problem (see Young 2000 for the setup of the k-medians decision problem).

Theorem: MEM solves any instance of k-medians

In *k*-medians, we are given a set of locations *L*, and a distance metric $d_{i,j}$ between locations i, j. The solution is the k-element subset of *L*, called *M* (*i.e.*, the *medians* of *L*), such that the sum of the distances from every member of *L* to the *nearest* member of *M* is minimized (equivalently, we have consumer's living in *n* cities and choose to locate stores in the subset of *k* cities that minimizes the total commute time from any city to a city with a store). The *k*-medians optimization is given by the first line of Equation (6).

$$\begin{aligned} M &= \underset{M' \subseteq L, |M'|=k}{\arg\min} \left[\sum_{i \in L} \min_{j \in M'} \{d_{ij}\} \right] \\ &= \underset{M' \subseteq L, |M'|=k}{\arg\max} \left[\sum_{i \in L} \max_{j \in M'} \{-d_{ij}\} \right] \\ &= \underset{M' \subseteq L, |M'|=k}{\arg\max} \left[\sum_{i \in L} \max_{j \in M'} \{-d_{ij}\} \frac{1}{m} \right] \end{aligned} \quad (6)$$

The k-medians optimization can be transformed to a MEM set optimization. We can substitute max's for min's by negating the distance metric. We can introduce the constant $1/m$, where m is the number of discrete user types, without altering the solution to the optimization. The result is equivalent in form to the MEM set optimization (Equations (3) and (1)) with $S = L$, $\{\boldsymbol{\theta}\}$=L, $\forall \boldsymbol{\theta}. p(\boldsymbol{\theta}) = 1/m$ and $u(\mathbf{a}_i; \boldsymbol{\theta}_j) = -d_{i,j}$ where $\{\boldsymbol{\theta}\}$ is a set of all user type parameters. Intuitively, one could imagine that every location in *L* is both a possible alternative and a user type. We connect users to alternatives via negated distance. Computing the MEM-set on the transformed problem results in the set of nodes with minimum sum of minimum distance to all other nodes as required. Since the k-medians problem is NP complete, we conclude that MEM-set problem is NP complete. One can also transform the MEM-set problem into a constrained k-medians problem (proof available from the authors). This might allow one to approximate MEM-set problem solutions using sophisticated non-metric k-medians algorithms (see Young 2000 for details).

Real applications may involve specific subclasses of alternative and user type distributions. It may be possible to exploit properties of these distributions to provide tractable optimal algorithms for the MEM-set problem.

MEM-set Algorithms

One could use sophisticated approximation algorithms to get guaranteed performance, however, we are primarily interested in quickly computing recommendation sets on-line. As a starting point, we focus on relatively simple, fast heuristic algorithms.

1. GMEM is a *greedy* approach to constructing MEM sets that successively adds the alternative to *M* that most improves the *emax* score of the current recommendation set. We start with the empty set M_0 and iteratively add to it until we have k alternatives in the set:

$$M_{j+1} = M_j \cup \left\{ \underset{i \in (S - M_j)}{\arg\max} \ \mathrm{emax}(M_j \cup i, p) \right\} \quad (7)$$

This algorithm is linear in the number of alternatives *n* and the set size *k* (*i.e.*, O(*nk*)). The complexity of GMEM is only a constant factor more than that of KIB methods.

2. 2MEM is a greedy local-search algorithm. We start by calculating a complete k-element greedy set M_k using GMEM as above. We then successively re-optimize each element of M_k greedily, holding the other elements constant. We continue as long as we are making progress (*i.e.*, until we reach the fix point defined by Equation (8)). This technique is a variation of the 2-opt heuristic (Croes, 1958) in the combinatorial optimization literature.

$$\begin{aligned} &\text{For all } i = 1, \ldots, k, \\ &M_k = M_k / i \cup \underset{j \in (S - M_k) \cup i}{\arg\max} \mathrm{emax}((M_k / i) \cup j, p) \end{aligned} \quad (8)$$

3. CMEM is a complete enumeration method based on a direct implementation of Equation (3). It simply computes *emax* on all possible k-element subsets. It is optimal, but

intractable on large sets of alternatives. We include it here for benchmarking purposes.

Empirical Investigation of MEM sets

In this section we assume that preference elicitation has been completed and that we are trying to find a desirable set of alternatives to recommend to a user first entering a website. We examine the benefits of MEM-sets empirically in two settings. In the first setting, we analyze the gains from MEM-sets as a function of the distributions of alternatives and user types. In the second setting, we demonstrate the benefits of MEM-sets on the real-world application of recommending apartments to students.

Sensitivity of MEM-set gains to Distributions

MEM sets improve over KIB sets by selecting alternatives that cover the diversity in user types. The user types can be covered only if there is also diversity in the alternative set. The possible gain of the MEM set will therefore be moderated by the diversity of both the alternatives and the user types. An equivalent line of reasoning starting from the no diversity case leads to the same conclusion: If there is little diversity in alternatives, then any alternative from the set will satisfy any consumer about as well (*i.e.*, the notion of a recommender system is redundant). Similarly, if there is a single user type, there is no uncertainty about preferences, and a single best alternative can always be recommended.

In our simple example, both alternatives and user types are expressed in a three-dimensional space (*e.g.*, laptops with clock rate, battery life, and hard disk storage). Within this space, alternatives and user types are assumed to lie on the surface of the positive octant of the unit sphere. This simplifying assumption roughly models the fact that dominated alternatives (those falling inside the sphere) are generally eliminated by marketplace dynamics. Where this is not the case, recommendation is not necessary since we can give everyone the same dominating alternative.

We model diversity in both user types (distribution of parameter vectors θ) and alternatives as ranging from uniform over the whole space to highly clustered. The range is represented formally by the "bathtub" density in Equation (9) formed from an α-normalized combination of two easy to sample polynomial densities. The density is uniform when the clustering parameter *c* is zero, increasingly convex with increasing values of *c*, and largely concentrated at 0 and 1 as *c* goes to infinity.

$$\Pr(Y = x; c) = \alpha \begin{cases} (x - 0.5)^c & 0.5 > x \geq 1.0 \\ (0.5 - x)^c & 0 \geq x \geq 0.5 \end{cases} \qquad (9)$$

Since the surface of the sphere is two-dimensional, we draw samples of alternatives and samples of user types from a two-dimensional version of this "bathtub" function. Both alternatives and user types use the same value of c. When *c* is 0, the alternatives and user types are spread out uniformly over the sphere's surface. When c is infinite, the alternatives and utility functions are concentrated at the corners of the positive octant of the sphere.

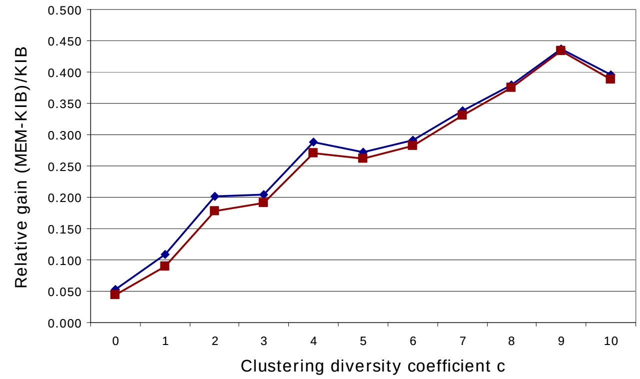

Figure 2. MEM set gain as a function of segmentation

We then generated artificial markets to empirically compare KIB and MEM methods. For each market, we randomly drew 20 alternatives and 5 utility functions. We then evaluated the *emax* criteria on CMEM, GMEM, 2MEM, and KIB sets. Gain was calculated as shown in Equation (10):

$$gain = \frac{\text{emax}(MEM) - \text{emax}(KIB)}{\text{emax}(KIB)}. \qquad (10)$$

Figure 2 shows the average gain in expected utility for using MEM sets over KIB sets. Each point in the graph represents 20 simulated markets. We can see that the gain is roughly the same for optimal CMEM (top line) and greedy GMEM (bottom line). 2MEM gains are only marginally worse than CMEM. For uniformly distributed alternatives and user types (*c*=0) the gain is about 5%. The gain increases steadily as the clustering coefficient c increases and reaches an asymptotic upper bound as *c* becomes large. We have seen gains in the 90% range at the extreme end of the clustering parameter, but these do not necessarily reflect realistic markets. We also observed that the execution time of CMEM grows rapidly with set size while greedy GMEM grows linearly as predicted.

Recommendations for Apartment Renters

Since MEM set gains are sensitive to input distributions, it is important to test the algorithms on real-world data. We chose apartment recommendations for our test. Our basic model is that students approach the university housing office with an imprecise notion of the market. We hoped to generate small samples representing the market to give them an idea of what is available.

We obtained an apartment utility model from a study done of students at a North American university (Elrod et al. 1992). The study concluded that four factors explained student preferences well: number of bedrooms in the apartment, the monthly rent, distance to campus and perceived safety of the neighborhood. Regression analysis suggested that a simple random coefficients model of the form

$$u(x) = a_{room}x_{room} + a_{rent}x_{rent} + a_{dist}x_{dist} + a_{safe}x_{safe}$$

explained students' preferences. Elrod's estimates of the parameters appear below:

Table 2. Coefficients for student utility function

	Bedrooms	Rent	Distance	Safety
Mean	0.45	-0.51	-0.13	0.21
Std. dev.	0.41	0.37	0.17	0.23

To simulate market diversity, we sampled utility functions randomly using the mean and variances given above. We obtained our alternatives from the off-campus apartment listings. We selected two months of 1 and 2 bedroom apartments with a particular configuration of utilities (roughly 60 apartments) and removed 2 apartments that strictly dominated all others. We note that the problem as formulated, has relatively little cluster structure in the user type distribution due to the linear utility family employed. We would therefore expect moderate gains. Results were averaged over 20 runs and appear in Table 3.

Table 3. Results of apartment recommendations

Method	KIB	CMEM	GMEM
Avg. value in utils	-0.024	0.168	0.165
Std. dev. in utils	0.0638	0.0646	0.0653
Approx. gain in $s	N/A	$90	$89
Recommended sets from one run	{6 11 9}	{22 4 16}	{22 4 6}

The results represent net utility (utility – cost). Both greedy and complete MEM sets return higher expected utilities than KIB (2MEM results were essentially identical to CMEM). The differences between the greedy and complete methods are negligible. Note that the greedy method used one of the sub-optimal apartments from KIB.

The gain in utility can be interpreted in dollars. From the coefficients we see that one bedroom is worth 0.45 "utils." In the relevant rental market, average rents are $526, $774, and $949 for 1, 2, and 3 bedroom apartments. An extra bedroom is worth roughly $212, so a util is worth around $470 (=212/.45). Apartments have few parameters with a small number of discrete levels. Uncertainty is resolved very quickly and segmentation is low. Despite this conservative scenario we still see gains of roughly 10% in expected utility. Furthermore, these gains can be obtained by GMEM, which has time complexity only a constant factor more than the traditional KIB method.

Future Work

In this paper we have presented an approach for generating recommendation sets. Several extensions and promising avenues for future research suggest themselves:

1. *Determine the optimal set size k*.* This can be done by calculating optimal MEM-sets, and associated *emax* scores, for successive values of k. The optimal set size is chosen using a selection criterion consisting of the *emax* score less a penalty for larger set sizes (associated with cognitive or economic costs).

2. *Include the recommender's interests in the objective.* Rather than maximize expected user utility, one might, for example, optimize expected recommender profits or minimize expected inventory holding cost.

3. *Model the ordering of alternatives within the set.* The recommender system can be made more useful as decision aid if alternatives with higher incremental expected utility contributions are ranked higher in the set.

4. *Derive preference covers for qualitative preference and collaborative filtering paradigms.*

5. *Develop new preference cover algorithms.* In particular, one could apply currently available polynomial time ε-optimal approximation algorithms developed for related k-medians and facility location problems. It may also be desirable to consider implicit enumeration and traditional AI methods, such as A-star.

6. *Consider the computational complexity of MEM-set optimization.* One could explore ways to exploit knowledge about common preference distributions to get better bounds on gains and time complexity.

7. *Extend adaptive preference elicitation* (Boutilier 2002) and adaptive conjoint analysis (Johnson 1974) to generate sets rather than individual recommendations.

8. *Replace the presentation and modification of single alternatives in critiquing approaches (Burke, Hammond, and Young 1997) with the presentation of a MEM-set and selection of representative elements in this set.* The MEM set represents a cover of the space of alternatives consistent with what is known of user preferences. Selection of an alternative from the set would communicate preference information that would then be used to generate a refined MEM set.

9. *Explore applications of MEM set optimization in various domains,* including (a) standard e-commerce applications (to help online consumers focus on subsets of relevant products in a desired category), (b) retail assortment planning (to help retail buyers or store managers select products in a given category to carry in a dedicated amount of shelf space); (c) real estate and similar services (to help real estate agents recommend a selection of houses for clients to visit); and (d) team selection (to help managers of teams or companies choose the set of players or employees that maximize the probability of achieving a desired goal, such as a point score or profits). The first two applications could involve movie, apparel, or

other large-assortment categories. The first three applications would be based on explicit preference elicitation or classification of users into consumer segments calibrated on conventional market research studies. Personalization can be achieved using dialog scripts to invoke the relevant segment preferences.

Conclusions

In contrast with traditional recommenders that consider each alternative separately, the MEM-set approach developed in this paper explicitly optimizes over possible sets of alternatives. The approach produces a diversity of alternatives in the recommendation set that optimally covers the uncertainty over possible user preferences. The resulting sets generally have higher expected utility (and never less). Though intractable to compute optimally, heuristic algorithms can provide large gains over traditional "k-individually best" methods. While the gains from MEM sets are moderated by diffuse distributions of alternatives and user types, gains can be found even in cases like the study of apartment recommendations described above. From a practical point of view, higher expected utility can be achieved with exactly the same information about the user at the price of a small increase in computation. We conclude that MEM sets offer a simple, practical, and effective approach to building recommenders that fully maximize utility.

Acknowledgements

We would like to thank our anonymous reviewers and our colleagues Russell Greiner, Gerald Häubl, and Xin Ge for helpful suggestions, Todd Mortimer and Marc Dumouchel of Whitematter Services and the University of Alberta Student's Union for access to apartment data. This research was supported by the Social Sciences and Humanities Research Council of Canada, through its Initiative on the New Economy Research Alliances Program (SSHRC grant 538-02-1013), and the Alberta Ingenuity Fund for Machine Learning.

References

Basu, C.; Hirsh, H.; and Cohen, W. 1998. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* 714-720. Madison, WI: AAAI Press.

Ben-Akiva, M., and Lerman, S. R. 1985. *Discrete Choice Analysis*. Cambridge, Mass: MIT Press.

Boutilier, C.; Bacchus, F.; and Brafman, R. 2001. UCP-Networks: A Directed Graphical Representation of Conditional Utilities. In *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence,* 56 - 64. Seattle, WA: Morgan Kaufmann.

Boutilier, C. 2002. A POMDP Formulation of Preference Elicitation Problem. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* 239-246. Edmonton, Alberta: AAAI Press.

Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research* 21: 135-191.

Bradley, K., and Smyth, B. 2001. Improving Recommendation Diversity. In *Procedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science*, 85-94. Maynooth, Ireland.

Burke, R., Hammond, K., and Young, B. 1997. The FindMe Approach to Assisted Browsing. *IEEE Expert* 12(4): 32-40.

Croes, G. A. 1958. A Method for Solving Traveling-Salesman Problems. *Operations Research* 6(6): 791-812.

Elrod, T.; Louviere, J. J.; and Davey, K. S. 1992. An Empirical Comparison of Ratings-based and Choice-based Conjoint Models*.* *Journal of Marketing Research* 29(3): 368-377.

Faltings, B.; Torrens, M.; and Pu, P. 2004. Solution Generation with Qualitative Models of Preferences. *Computational Intelligence* 20(2): 246-263.

Green, P. E., and Srinivasan, V. 1990. Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice. *Journal of Marketing* 54(4): 3-19.

Johnson, R. M. 1974. Trade-off Analysis of Consumer Values. *Journal of Marketing Research* 11(2):121-127.

Konstan, J. A.; Maltz, D.; Herlocker, J. L.; Gordon, L. R.; and Riedl, J. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the Association for Computing Machinery* 40 (3): 77 – 87.

Louviere, J. J.; Hensher, D. A.; and Swait, J. D. 2000. *Stated Choice Methods*. Cambridge, England: Cambridge University Press.

Popescul, A.; Ungar, L.; Pennock, D.; and Lawrence, S. 2001. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence,* 437-444. Seattle, WA: Morgan Kaufmann.

Sengupta, J. K. 1972. *Stochastic Programming: Methods and Applications*. Amsterdam: North Holland Company.

Young, N. E. 2000. K-Medians, Facility Location, and the Chernoff-Wald Bound. In *ACM-SIAM Symposium on Discrete Algorithms*.