DD-PREF: A Language for Expressing Preferences Over Sets

Marie desJardins
University of Maryland Baltimore County
Computer Science and Electrical Engineering Department
1000 Hilltop Circle, Baltimore, MD 21250
mariedj@csee.umbc.edu

Kiri L. Wagstaff
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
kiri.wagstaff@jpl.nasa.gov

Abstract

In many application domains, it is useful to be able to represent and reason about a user's preferences over sets of objects. We present a representation language, DD-PREF (for Diversity and Depth PREFerences), for specifying the desired diversity and depth of sets of objects where each object is represented as a vector of feature values. A strong *diversity* preference for a particular feature indicates that the user would like the set to include objects whose values are evenly dispersed across the range of possible values for that feature. A strong *depth* preference for a feature indicates that the user is interested in specific target values or ranges. Diversity and depth are complementary, but are not necessarily opposites.

We define an objective function that, when maximized, identifies the subset of objects that best satisfies a statement of preferences in DD-PREF. Exhaustively searching the space of all possible subsets is intractable for large problem spaces; therefore, we also present an efficient greedy algorithm for generating preferred object subsets. We demonstrate the expressive power of DD-PREF and the performance of our greedy algorithm by encoding and applying qualitatively different preferences for multiple tasks on a blocks world data set. Finally, we provide experimental results for a collection of Mars rover images, demonstrating that we can successfully capture individual preferences of different users, and use them to retrieve high-quality image subsets.

Introduction and Motivation

In many application domains, it is useful to be able to express preferences or utilities over sets of objects. Consider:

- a search engine user wants to see ten relevant yet diverse documents;
- an autonomous Mars rover must select fifteen images to download to scientists on Earth;
- a college wants to select a group of 100 incoming students from the 1500 applications it received.

In all of these cases, the preferences among the possible choices *interact* with each other. For example, the search user does not want to see duplicate documents, despite the fact that they will be ranked equally and therefore could both appear in the top ten documents. The college wants to have highly ranked students who also comprise a diverse incoming class. These examples show that the simple approach of ranking the items and taking the top k as the selected subset will not necessarily yield the optimal subset. To enable the study of this phenomenon, we introduce two important, sometimes competing, aspects of set preferences: the desired *diversity* and *depth* of the selected subset. A strong diversity preference for a particular feature indicates that the user would like the set to include objects whose values are evenly dispersed across the range of possible values for that feature. A strong depth preference for a feature indicates that the user is interested in specific target values or ranges. While selecting the top k items from a ranked list can address a depth preference, it cannot satisfy a diversity preference, since diversity is inherently a set-based property.

Previous work on preference modeling for sets of objects has focused on capturing interactions between propositional objects. That is, the objects are not represented by any features. While this framework is useful for resolving combinatorial auctions (e.g., Nisan, 2000; Boutilier, 2002), it precludes the possibility of generalization to new sets. Preference modeling that does account for feature values has focused on ranking the individual items in a list, rather than selecting the optimal subset of items (e.g., Crammer & Singer, 2001). This work seeks to address the gap between these two areas by developing a method for modeling, applying and, eventually, learning feature-based preferences over sets of objects.

The primary contribution of this paper is DD-PREF (for Diversity and Depth PREFerences), a language that allows a user to specify the degree to which individual features should be varied (diversity) or focused on particular values (depth). We also present a simple greedy algorithm for computing high-quality subsets. We evaluate the ability of DD-PREF to encode meaningful preferences for (1) three separate tasks in an artificial blocks world domain, where we observe that the optimal subset of k objects often differs from the top k individually ranked objects, and (2) a Mars rover image retrieval task, where we automatically infer and apply preferences for multiple users.

Related Work and Discussion

Preference modeling has been widely studied in decision theory, machine learning, and multi-agent systems. However, most of this work focuses only on preferences over individual objects, rather than sets (Crammer & Singer 2001); or on preferences over sets of propositional objects, rather than feature-based objects (Cramton, Shoham, & Steinberg 2005). CP-Nets (Boutilier *et al.* 2004), which capture preferential independence and conditional independence relationships among object features in a graphical representation, have been applied to a number of different domains, including web page configuration (Domshlak, Brafman, & Shimony 2001). However, CP-Nets capture only interactions among features, not among objects in a set.

Barberà, Bossert, & Pattanaik (2004) survey the literature on ranking sets of objects. Most of this work focuses on modeling preferences over *opportunity sets*, where the important issue is freedom of choice within mutually exclusive alternatives. Barberà, Bossert, & Pattanaik use the term *joint alternatives* for the problem in which we are interested (collections of objects that may all be useful or relevant). Again, research in this area focuses only on propositional domains.

Recommender systems have primarily focused on rating individual objects, but several authors have identified the *portfolio effect*—in which including highly similar objects in a recommendation set may be undesirable—as an important issue. In a survey of hybrid recommender systems, Burke briefly mentions the portfolio effect; the only suggestion given is to avoid objects that the user has already seen. Ali & van Stam (2004) indicate that the portfolio effect is a major concern in the domain of TV show recommendations; they mention a domain-specific set reduction technique (use the next episode of a TV show to represent that series), and suggest that one should avoid "imbalance" in a recommendation set, but do not give specific methods for doing so.

More recently, Ziegler *et al.* performed an extensive user study to examine the effects of a technique they call *topic diversification* on book recommendation (Ziegler *et al.* 2005). They found that by balancing a recommendation list using a user's interests, although accuracy is reduced, user satisfaction is increased. Their technique is domain-specific, and treats objects as having only a single relevant attribute. However, their findings reinforce our claim that modeling the diversity of a set of objects can increase performance.

DD-PREF: Specifying Preferences over Sets

To address these limitations, we present the DD-PREF language for capturing feature-based preferences over sets of objects. We assume a scenario in which a user wishes to select a subset S of k objects from U, the universe of n objects, where each object is represented as a vector of m feature values. DD-PREF supports two important preference notions: *diversity* and *depth*. A diversity preference specifies the desired amount of variability among objects in the selected subset, and a depth preference specifies preferred feature values.

We will use the term *candidate subsets* to refer to all $C(n, k)$ (n choose k) possible subsets of size k that can be constructed from n objects; *selected subset* to refer to the subset of k objects that is returned by an algorithm; and *optimal subset* to refer to the best available subset (i.e., the subset of size k that maximizes the DD-PREF objective function, as defined below). Each object x_i is represented as a feature vector, $(x_i^1, \ldots, x_i^m)$, where $x_i^f \in \mathcal{V}_f$, and $\mathcal{V}_f$ is the domain of feature f. In this paper, we focus on real-valued features, but the methods can be applied to integer-valued and categorical (discrete) features as well.

In our language, a preference statement $\mathcal{P}$ is a collection of individual feature-based preferences $\mathcal{P}_f$, $f = 1, \ldots, m$. Each preference $\mathcal{P}_f$ is expressed as a tuple: $< q_f, d_f, w_f >$ indicates that we prefer subsets of objects that exhibit an amount of diversity $d_f \in [0, 1]$, subject to a "quality" function q_f that maps feature values to their relative desirability, with a weight of $w_f \in [0, 1]$. The feature weight w indicates a feature's importance relative to other feature preferences. For example, a Mars rover scientist might identify "percent of image that is rock" as an important feature ($w_f = 1.0$), preferring sets containing images that are 50-80% rock (q_f), with high diversity ($d_f = 0.8$). In this work, we examine preferences explicitly specified by a user as well as preferences inferred from independent item rankings provided by the user.

Depth. The quality functions q_f specify the preferred *depth* of the optimal subset; that is, which feature values are most desirable in the selected objects. In the simplest case, the quality function can simply be represented as a step function, where values in a desired range $[v_{min}, v_{max}]$ are mapped to 1 and all other values are mapped to 0. This preferred range can also be interpreted as a soft constraint on the feature values, by penalizing values outside the range to varying degrees. Other examples of quality functions might be a bimodal preference, indicating that very large and very small values are preferred to medium values, or a monotonic preference, indicating that larger values are preferred to smaller ones.

Given the m quality functions in $\mathcal{P}$, we define the depth of an object $x \in S$ as the weighted average quality of its feature values:

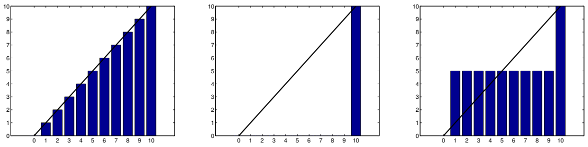
$$\text{object-depth}(x, \mathcal{P}) = \frac{1}{\sum_f w_f} \sum_{f=1}^{m} w_f q_f(x^f),$$

where x^f is the value of the f-th feature for x. The depth of a set S is the average depth of the objects in the set:

$$\text{depth}(S, \mathcal{P}) = \frac{1}{|S|} \sum_{x \in S} \text{object-depth}(x, \mathcal{P}).$$

The depth of a set will always be in the range $[0, 1]$.

Note that this definition of depth does not model interactions between objects or between features. Interactions between different objects are modeled in DD-PREF by the diversity preference, as explained next. In some domains, a concept of depth that is sensitive to inter-feature dependencies might be useful; for example, one might only be interested in web pages that mention machine learning if they also mention preference modeling. We plan to investigate ways to model such interactions, perhaps by using modified CP-Nets (Boutilier *et al.* 2004). CP-Nets capture preferential independence and conditional independence relation-



(a) Skew 0.00 (b) Skew 1.00 (c) Skew 0.21

Figure 1: Sets with different skew values.

ships among features in a graphical representation, but they have not been applied to object sets.

Diversity. Diversity is a set-based property that captures the degree to which values for a particular feature are grouped close together (low diversity) or widely and evenly dispersed across the range of values (high diversity). The diversity measure that we define ranges from 0 to 1, where 0 corresponds to a preference for minimal diversity (all objects have the same value for that feature) and 1 represents a maximal diversity preference (objects have values that are maximally distinct and spread evenly across the desired range).

We define diversity in terms of a complementary notion we call *skew.*[1] Skew quantifies the amount by which a set of real values diverges from an even distribution of values over the range. (Skew can also be defined for integer and categorical values, but we focus on real-valued features here.) A low skew value means a very even distribution, corresponding to high diversity; high skew corresponds to low diversity.

We calculate the skew $\sigma(V)$ of a list of $k > 1$ sorted values $V =< v_{min}, \ldots, v_{max} >$ as the normalized *squared loss function* for a linear fit through v_{min} and v_{max}. This loss function is normalized by the maximum possible squared loss. Specifically,

$$\sigma(V) = \frac{\sum_{i=1}^{k}(v_i - v_i')^2}{M(V)},$$

where v_i', the ith value in an evenly distributed list of k values bounded by v_{min} and v_{max}, is computed by:

$$v_i' = v_{min} + (v_{max} - v_{min})\frac{i-1}{k-1},$$

and $M(V)$, the maximum squared loss for a list with the same v_{min}, v_{max}, and length as V, is:

$$M(V) = \sum_{i=1}^{k-1}(v_i' - v_{min})^2. \qquad (1)$$

The maximum squared loss occurs when there are only two distinct values (v_{min} and v_{max}) in the list, and the values are distributed in a maximally uneven fashion. Without loss of generality, we let there be one value at v_{max} and the rest ($k-1$ values) at v_{min}, yielding Equation 1. By this definition, skew is undefined for a list composed of only one distinct value; for completeness, we set $\sigma(V) = 0$ in this case.

Figure 1 shows three different sets of 11 values, all with the same minimum (0) and maximum (10) values. The linear fit through 0 and 10 is shown by a solid line. Figure 1(a) matches this distribution exactly, and has a skew of 0.0. Figure 1(b) has ten values at 0 and one value at 10, so it is maximally divergent from the constrained linear fit and has a skew of 1.0. Figure 1(c) has all nine intermediate values set to 5; the resulting skew is 0.21.

```
BASIC-GREEDY($\mathcal{P}$, $U$, $s$, $k$, $\alpha$)
1. Initialize candidate set $S$ with seed object $\{s\}$.
2. For $j$ from 2 to $k$:
   (a) Select the object $x \in (U - S)$ that maximizes
       $\mathcal{F}_{dd}(S \bigcup \{x\}, \mathcal{P}, \alpha)$.
   (b) Set $S = S \bigcup \{x\}$.
3. Return $S$.
```

Figure 2: Pseudocode for greedy subset selection.

Since low skew corresponds to high diversity and vice versa, we define the *diversity of feature f* over a candidate subset $S = \{x_1, \ldots, x_k\}$ as 1 minus the skew of that feature's values in S:

$$\text{div}_f(S) = 1 - \sigma(\text{sort}(< v_i^f | i = 1, \ldots, k >)).$$

The *actual* diversity of a set is then the weighted average of the diversity values for each feature:

$$\text{actual-div}(\mathcal{S}) = \frac{1}{\sum_f w_f} \sum_{f=1}^{m} w_f \text{div}_f(S).$$

To capture the degree to which a subset's diversity matches the desired diversity, d_f, we calculate the average squared diversity error, weighted by the feature preferences:

$$\text{div}(S, \mathcal{P}) = \frac{1}{\sum_f w_f} \sum_{f=1}^{m} w_f (d_f - \text{div}_f(S))^2.$$

This measure of diversity will always be in the range $[0, 1]$.

As with depth, because we measure the diversity of each feature independently, DD-PREF cannot capture interactions between features in the diversity measure. It is not clear that such interactions arise in practice; in the domains we have looked at, there are no obvious cases where inter-feature interactions are important in modeling diversity. However, it may be useful in the future to model such interactions.

Objective Function. We define an objective function $\mathcal{F}_{dd}(S, \mathcal{P}), \alpha$ that assesses the value of a candidate subset S, given a preference statement $\mathcal{P}$. The objective function includes a parameter α (referred to as the *diversity weight*), which allows the user to emphasize the relative importance of depth and diversity.

$$\mathcal{F}_{dd}(S, \mathcal{P}, \alpha) = (1 - \alpha)\,\text{depth}(S, \mathcal{P}) + \alpha\,\text{div}(S, \mathcal{P}) \qquad (2)$$

Greedy Algorithm for Subset Selection

To evaluate the expressive power of DD-PREF, we implemented a greedy algorithm for selecting a subset of k objects, given a preference statement $\mathcal{P}$ (see Figure 2). This algorithm takes as input the preference $\mathcal{P}$, the set of objects U, the diversity weight α, and a "seed object" $s \in U$ to serve

[1] We use this term for its intuitive meaning of bias or unevenness, not in the statistical sense of skewness.

as the starting point for the selected subset. This is necessary because diversity cannot be evaluated over a set with only one member. The greedy algorithm repeatedly adds the best object (according to the DD-PREF objective function) to the subset until the set includes k objects.

We observe that the greedy algorithm gives close to optimal performance in practice. However, it can be sensitive to the seed object that is selected. Recall that for singleton sets, the diversity function always returns 0, so only the depth value matters in the objective function. We define and analyze three variants with different seed selection mechanisms. The Basic-Greedy algorithm simply chooses a seed item randomly from the equivalence set of items in U with maximal depth (i.e., with maximal weighted quality). The Wrapper-Greedy method (which we use in our experiments) invokes the greedy algorithm n times, trying each seed object in turn, and then selects the subset with the overall best value for the objective function. Finally, LA-Greedy (Lookahead-Greedy) searches exhaustively over all subsets of size 2 to find the optimal such subset, and then uses this subset to initialize the greedy algorithm.[2]

We compare our approach to three baselines: exhaustive search, random selection, and the Top-K algorithm, which selects the top k blocks independently, according to the depth function only.

Complexity analysis. The big-O complexity of each of the six algorithms is given in the following table, and is briefly justified in the following text. (Due to space limitations, the mathematical details are omitted.) The complexity of calculating the depth or diversity of a set of k objects is $O(mk)$; therefore, the objective function $\mathcal{F}_{dd}$ is also $O(mk)$. We assume that $n >> k$.

Algorithm	Complexity
Basic-Greedy	$O(mk^2n)$
Wrapper-Greedy	$O(mk^2n^2)$
LA-Greedy	$O(mkn^2)$
Exhaustive	$O(mkn^k)$
Top-K	$O(mk^2n)$
Random	$O(k)$

Basic greedy search evaluates n subsets of size 1, $n-1$ subsets of size $2, \ldots$, and $(n-k+1)$ subsets of size k, so it is $O(\sum_{i=1}^{k} m\,(n-i+1)\,(i))$. *Wrapper greedy search* applies greedy search for each of n seed objects. *Lookahead greedy search* exhaustively searches all subsets of size 2 (which is $O(C(n,2)\,mk) = O(n^2mk)$), then uses those as the first two seed objects and grows the rest of the set greedily. For large n, the n^2 term in the lookahead dominates the k^2 term in greedy search.

Exhaustive search evaluates $C(n,k)$ (n choose k) subsets. $C(n,k)$ is $O(n^k)$ for $n >> k$. *Top-K search* computes depth only ($O(mk)$) for n objects and maintains a sorted list of length k with $O(k)$ comparisons at each step. *Random search* generates k random selections.

Experimental Results

Encoding Qualitatively Different Preferences

To illustrate how DD-PREF can be used to model preferences for different types of tasks, we developed a simple blocks-world data set, which includes three different tasks with very different preferences. We show the results of applying five different algorithms to generate preferred subsets from a collection of blocks: (1) the Basic-Greedy algorithm, (2) the Wrapper-Greedy algorithm, (3) the Top-K algorithm, (4) random subset selection, and (5) exhaustive search.

Blocks world data. Objects in the synthetic blocks domain have four attributes: size (a real value from 0 to 100), color (represented as integers from 0–6), number-sides (an integer value from 3 to 20), and bin (representing sequential locations in a storage area; an integer from 0 to 100). We have developed sample preference statements for three distinctly different tasks. Although these tasks (and the specific values associated with each task) are artificial, they are intuitively reflective of the types of real-world preferences one might wish to model in such a domain. The quality functions q_f are step functions, specified by the minimum and maximum desired values.

Task 1: Construct a mosaic. This task requires various block sizes—but all fairly small; with varied colors; and many different shapes. Location is less important, but the blocks should be close together if possible. Thus, we have the following feature preferences $< q_f, d_f, w_f >$:

$\mathcal{P}_{size} =$	$< [0, 25],$	$0.8,$	$1.0 >$
$\mathcal{P}_{color} =$	$< [0, 6],$	$0.75,$	$0.8 >$
$\mathcal{P}_{number-sides} =$	$< [3, 20],$	$1.0,$	$0.6 >$
$\mathcal{P}_{bin} =$	$< [0, 100],$	$0.1,$	$0.6 >$

Task 2: Build a uniform tower. Here we want large, similar-size, similar-color blocks, all the same shape and with few sides. The location doesn't matter (i.e., $w_{bin} = 0$).

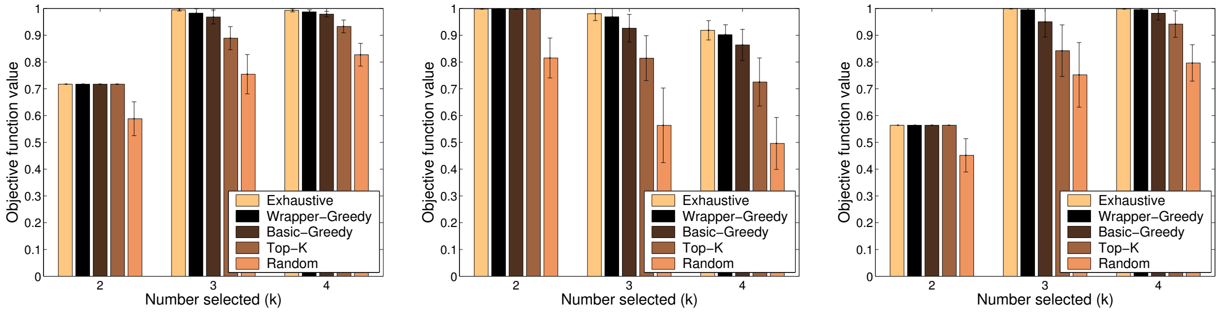
$\mathcal{P}_{size} =$	$< [50, 100],$	$0.1,$	$1.0 >$
$\mathcal{P}_{color} =$	$< [0, 6],$	$0.0,$	$1.0 >$
$\mathcal{P}_{number-sides} =$	$< [4, 8],$	$0.0,$	$1.0 >$

Task 3: Select blocks for a child. The blocks for this task must be medium-sized for grasping, in many different colors and shapes, and located close together.

$\mathcal{P}_{size} =$	$< [10, 100],$	$1.0,$	$1.0 >$
$\mathcal{P}_{color} =$	$< [0, 6],$	$1.0,$	$0.8 >$
$\mathcal{P}_{number-sides} =$	$< [3, 20],$	$1.0,$	$0.8 >$
$\mathcal{P}_{bin} =$	$< [0, 100],$	$0.2,$	$0.4 >$

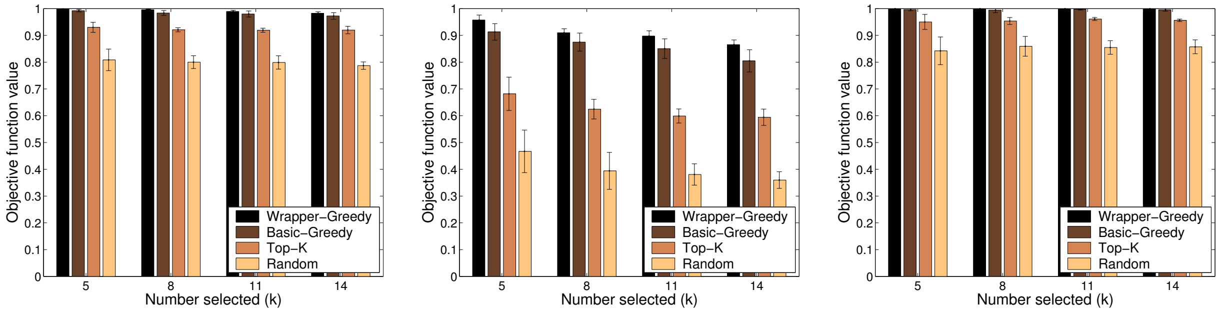
Methodology. The same n blocks were used with all algorithms, all values of k, and all preferences in a given trial. The diversity weight α is set to 0.5 for all trials. The first set of experiments (Figure 3) applied exhaustive search to small problems to measure the true optimal value of the objective function; we used this baseline to assess the performance of the other four algorithms. In these experiments, n was set to 50, and k was set to 2, 3, and 4. Exhaustive search is impractical for larger problems, so we also ran a set of experiments using only the first four methods (Basic-Greedy, Wrapper-Greedy, Top-K, and Random) on 200 blocks, with $k = 5, 8, 11,$ and 14 (Figure 4).

[2] Although we mention LA-Greedy and give its complexity analysis, we did not evaluate this algorithm in the experiments presented in the paper. We plan to explore this method and other variants more thoroughly in future work.



(a) Task 1 (Mosaic) (b) Task 2 (Tower) (c) Task 3 (Child)

Figure 3: Blocksworld results on small data sets ($n = 50, \alpha = 0.5$), averaged over 20 trials. Bars show +/- one standard deviation.



(a) Task 1 (Mosaic) (b) Task 2 (Tower) (c) Task 3 (Child)

Figure 4: Blocksworld results on large data sets ($n = 200, \alpha = 0.5$), averaged over 20 trials. Bars show +/- one standard deviation.

Discussion. Figure 3 shows that the Wrapper-Greedy algorithm performs very well, identifying subsets whose objective function values are close to the optimal value in all cases tested. For two-block subsets, all of the algorithms but Random perform equally well. This is because any pair of instances will yield a diversity of 1.0. so the diversity match is the same for all algorithms. (The Random algorithm performs worse because it will sometimes select blocks that do not satisfy the depth preference.) Note that this highlights a potential weakness of our current definition of diversity, which does not measure the degree to which a set of values actually covers the possible range of values. This latter property can be termed *coverage*; we are currently investigating ways to measure coverage, either by modifying the definition of diversity, or by adding an additional term to the objective function.

For $k > 2$, Wrapper-Greedy significantly outperforms Top-K and Random in the 50-block experiments (Figure 3) and in the larger 200-block experiments shown in Figure 4.

For Task 1 and Task 3, on the large data sets (Figures 4), both the objective function value and the relative performance of the algorithms stay fairly constant as k increases. By contrast, on Task 2, the objective function value for all of the algorithms decreases as k increases. This is because Task 2 is highly constrained (size to $[50, 100]$ and numbersides to $[4, 8]$). As a result, very few blocks actually satisfy the constraints, so the depth preference cannot be satisfied for most of the blocks when k is large. In Task 2, there is also a strong preference for blocks having the same color, which is difficult to satisfy given the random generation of the blocks.

As shown by the complexity analysis previously, the greedy algorithm is far more computationally efficient than exhaustive search. Since it yields near-optimal performance, it is a good choice for larger problems. Wrapper-Greedy consistently yields higher performance than Basic-Greedy, but is is computationally more expensive (by a factor of n). Also, for large k, the difference between Wrapper-Greedy and Basic-Greedy is generally less than the difference between Basic-Greedy and Top-K. In Tasks 1 and 3, the difference between Wrapper-Greedy and Basic-Greedy is barely noticeable. Therefore, in some domains, the Basic-Greedy algorithm may be preferred to Wrapper-Greedy. Hybrid methods such as Lookahead-Greedy can provide a compromise between these two algorithms, but were not studied in these experiments.

The Impact of Preference Specification. Next, we investigated the importance of being able to specify different preferences. Given that Wrapper-Greedy identifies good subsets for each task, are these subsets similar to each other or quite distinct? If we find that approximately the same subset is selected when using different preferences, then specifying the preferences may not be very important.

Table 1 shows the objective function values obtained when generating the best block subset (k = 14, n = 200) according to one preference statement but evaluating it against

Generating Preference	**Evaluating Preference**		
	Task 1 (mosaic)	**Task 2** (tower)	**Task 3** (child)
Task 1	**0.9827 ± 0.0054**	0.3054 ± 0.0306	0.9249 ± 0.0302
Task 2	0.7128 ± 0.0073	**0.8654 ± 0.0175**	0.6811 ± 0.0156
Task 3	0.8761 ± 0.0224	0.2646 ± 0.0212	**0.9994 ± 0.0007**
Random	0.7813 ± 0.0166	0.4051 ± 0.0382	0.8542 ± 0.0267

Table 1: Preference comparison across blocks world tasks, averaged over 20 trials ($k = 14, n = 200, \alpha = 0.5$). Rows represent different selected subsets; columns represent different preference evaluations.

other preferences. The first three rows represent the subsets obtained when running Wrapper-Greedy with preferences for Task 1 (mosaic), Task 2 (tower), and Task 3 (child). The fourth row represents the subsets obtained by random selection. The columns give the average objective function for the selected subsets, over 20 trials, when evaluated against each of the three preferences. The best value for each column appears in boldface.

We find that the diagonal entries, where the generating and evaluating preferences are identical, have the highest objective function values. This demonstrates that the Wrapper-Greedy algorithm is doing a good job of tailoring the selected subset to the specified preferences. In addition, the preferences have significant impact: the subsets generated according to other preferences do a poor job of satisfying the evaluating preference. Further, we find that the block subsets for one task, when measured by the other task preferences, are often worse than a *randomly* selected set of blocks would be. For example, the subsets generated for Task 1, when evaluated in the context of Task 2's preference, yield a worse objective function value (0.3054 on average) than a randomly selected block subset (0.4051 on average). The diagonal entries also have the smallest standard deviations—that is, subsets selected for a given task have lower variability with respect to that task's preferences than do other subsets. We conclude that preference specification can have a large impact on the results, and specifying the right set of preferences is critical.

Applying Preferences to Retrieve Larger Sets

The ability to encode and apply set-based preferences is particularly useful for large retrieval scenarios. For example, a user may be willing to indicate preferences over a small number of objects in exchange for getting back the top fraction of objects from a much larger data set. We tested this hypothesis by encoding user preferences for Mars rover images in DD-PREF and then applying them to retrieve a matching subset from a larger group of images. We also explored the impact of different values for the diversity weight, α.

We obtained a set of 100 grayscale images taken by a Mars field test rover, on Earth. Each image is represented by six features: the percent of the image composed of sky, rock, rock layers, dark soil, light soil, and shadow. Figure 5(a) shows one of the images and its feature values. For evaluation purposes, each user identified, without examining the image features, the best subset of 20 images, S_{top20}.

Inferring Preferences. First, we identified each user's top five ranked images, S_{top5}, from a separate task in which the same users were asked to fully rank 25 randomly selected images from the same set of 100 (Castaño *et al.* 2005). Our goal was to see if preferences inferred from S_{top5} could be used to identify a subset similar to S_{top20}. We set the desired diversity d_f to the diversity of S_{top5}. The quality function q_f was defined as a soft constraint over the range of values in S_{top5} for feature f. Specifically, $q_f(x_i^f) = 1$ if $x_i^f \in [v_{min}^f, v_{max}^f]$ and $(1 - \min(|v_{min}^f - x_i^f|, |x_i^f - v_{max}^f|)^2)$ otherwise. We set $w_f = 1.0$ for all features. For example, User 1's preferences were:

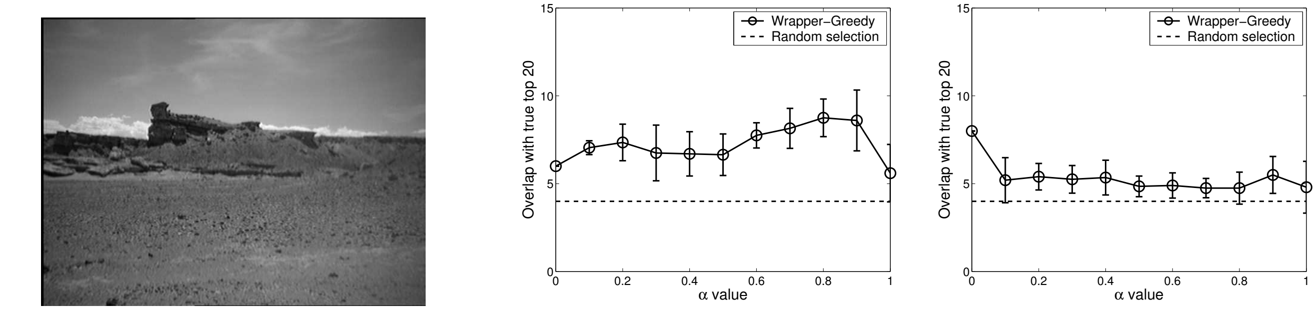
$\mathcal{P}_{sky} =$	$< [33, 50],$	$0.94,$	$1.00 >$
$\mathcal{P}_{rock} =$	$< [2, 16],$	$0.92,$	$1.00 >$
$\mathcal{P}_{layers} =$	$< [0, 2],$	$0.81,$	$1.00 >$
$\mathcal{P}_{lightsoil} =$	$< [1, 6],$	$0.97,$	$1.00 >$
$\mathcal{P}_{darksoil} =$	$< [35, 47],$	$0.99,$	$1.00 >$
$\mathcal{P}_{shadow} =$	$< [0, 1],$	$0.94,$	$1.00 >$

In contrast, User 2 preferred images with less sky, more rocks, and more light soil.

Methodology. For these experiments, we used the inferred preference statements and the Wrapper-Greedy method to select a subset of $k = 20$ items from the full data set of $n = 100$ items. We evaluated the selected subsets in terms of their overlap with the S_{top20}, varying $alpha$ from 0 to 1 in increments of 0.1. The overlap expected by random chance is four images.

Discussion. Figures 5(b,c) show that Wrapper-Greedy performed significantly better than random chance for each user, despite their different preferences. The best result for User 1 was obtained with a high diversity weight ($\alpha = 0.9$). Note that this result is better than a focus purely on depth ($\alpha = 0.0$, equivalent to a Top-K approach) or diversity ($\alpha = 1.0$). In contrast, the best result for User 2 was obtained with a focus purely on depth.

We observe that although the mean overlap for each user's best results never exceeds 8 (of a possible 20), the objective function values for these subsets are very high (0.999 in all cases). We found that the true S_{top20} was actually valued lower (0.995 for User 1 and 0.994 for User 2), which suggests that (a) the users may not have been consistent in their selection of S_{top5} and S_{top20}, and/or (b) the features cannot adequately capture what the users value in the images. For example, User 1's preferences specify a range of 33-50% for the sky feature, but in this user's S_{top20}, the sky feature ranges from 1-54%. Similarly, the desired (based on S_{top5})



Sky	Rock	Layers	Light soil	Dark soil	Shadow
32.6	10.1	2.0	4.9	47.1	0.8

(a) Rover image 9 of 100 and its feature values.

(b) Results for User 1

(c) Results for User 2

Figure 5: Rover image results ($n = 100, k = 20$), averaged over 20 trials; bars show +/- one standard deviation.

and actual (based on S_{top20}) diversity values differ greatly for some features. Other methods for eliciting preferences, and a richer feature set, will likely improve performance.

These experiments demonstrate that DD-PREF can successfully—albeit not perfectly—encode user-specific preferences over subsets. One remaining challenge is the issue of how to combine multiple users' preferences, when they must agree on a single subset (such as when a Mars rover must select images that satisfy multiple scientists' preferences).

Conclusions and Future Work

We presented the DD-PREF language for representing preferences over sets of objects, and introduced two key supporting concepts: *diversity* and *depth*. We showed that DD-PREF can be used to encode qualitatively different preferences in two domains, and presented an efficient and effective greedy algorithm for generating preferred subsets.

We identified a simple way to infer preferences from ranked image data. More generally, we are interested in developing automated algorithms to learn DD-PREF preference statements from ranked objects and object sets, including negative (non-preferred) training data.

We also plan to explore several extensions to DD-PREF, including the use of CP-Nets (Boutilier *et al.* 2004) to represent interactions between features for depth and diversity preferences, ways to capture inter-object interactions for depth preferences, and additional kinds of inter-object interactions, such as coverage, complementarity, and substitutability.

Acknowledgments

This work was supported by NSF ITR #0325329 and the Interplanetary Network Directorate. Thanks to Eric Eaton and other members of the UMBC MAPLE Laboratory for feedback and suggestions on the research; to Rebecca Castaño and JPL's Machine Learning Systems Group for the rover image data; and to the anonymous reviewers for their suggestions and comments.

References

Ali, K., and van Stam, W. 2004. TiVo: Making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 394–401. ACM Press.

Barberà, S.; Bossert, W.; and Pattanaik, P. K. 2004. Ranking sets of objects. In Barberà, S.; Hammond, P. J.; and Seidl, C., eds., *Handbook of Utility Theory*, volume 2: Extensions. Springer. Chapter 17.

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. CP-Nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of AI Research* 21:135–191.

Boutilier, C. 2002. Solving concisely expressed combinatorial auction problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 359–366.

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12(4):331–370.

Castaño, R.; Wagstaff, K.; Song, L.; and Anderson, R. C. 2005. Validating rover image prioritizations. *The Interplanetary Network Progress Report* 42(160).

Crammer, K., and Singer, Y. 2001. Pranking with ranking. In *Proceedings of the Neural Information Processing Systems Conference*, 641–647.

Cramton, P.; Shoham, Y.; and Steinberg, R., eds. 2005. *Combinatorial Auctions.* MIT Press.

Domshlak, C.; Brafman, R. I.; and Shimony, S. E. 2001. Preference-based configuration of web page content. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 1451–1456.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 1–12.

Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 15th International World Wide Web Conference*. In press.