Active Exploration in Instance-Based Preference Modeling

L. Karl Branting

Department of Computer Science
University of Wyoming
P.O. Box 3682
Laramie, WY 82972, USA
karl@uwyo.edu

Abstract. Knowledge of the preferences of individual users is essential for intelligent systems whose performance is tailored for individual users, such as agents that interact with human users, instructional environments, and learning apprentice systems. Various memory-based, instance-based, and case-based systems have been developed for preference modeling, but these system have generally not addressed the task of selecting examples to use as queries to the user. This paper describes *UGAMA*, an approach to learning preference criteria through active exploration. Under this approach, Unit Gradient Approximations (UGAs) of the underlying quality function are obtained at a set of reference points through a series of queries to the user. Equivalence sets of UGAs are then merged and aligned (MA) with the apparent boundaries between linear regions. In an empirical evaluation with artificial data, use of UGAs as training data for an instance-based ranking algorithm (1ARC) led to more accurate ranking than training with random instances, and use of UGAMA led to greater ranking accuracy than UGAs alone.

1 Introduction

Knowledge of the preferences of individual users is essential for intelligent systems whose performance is tailored for individual users, such as advisory agents and self-customizing systems. While some simple preferences are easily elicited (e.g., the preference for one soft-drink over another), more complex preference criteria may be difficult or extremely inconvenient for users to articulate (e.g., preferences among designs, schedules, plans, or other configurations).

A variety of approaches to automated preference acquisition are possible, varying in the attentional cost, or cognitive load, that they impose on the user. At one extreme is *a priori* knowledge, such as group membership, stereotypes, and default models, which can be determined at no attentional cost to the user. For example, collaborative filtering systems typically base their preference models on easily-obtained group membership information [GNOT92]. A second approach that has no attentional cost is *passive observation*, simply recording the user's choices.

A more active approach is the "candidate/revision" or "learn-on-failure" approach, under which the system makes suggestions based on its current model and revises the model whenever a suggestion is rejected. This approach has been applied to text retrieval [HGBSO98], telescope observing schedules [BB94], acquisition of "interface agents" [MK93], calendar management [DBM+92], and information filtering [Mae94].

At the opposite end of the spectrum of demands on the user from passive learners are approaches involving queries posed to the user. One approach to querying the user is *criteria elicitation*, in which the user's preference criteria are explicitly elicited through an extended interview process [KR93]. The attentional and time costs of explicit criteria elicitation make it infeasible for most automated systems. However, *exploration*, querying the user with pairs to be ranked (or larger collections from which the best instance should be selected) can potentially lead to faster acquisition of preference models than passive observation with less burden on the user than explicit criteria elicitation.

The choice among the methods for acquisition of user-specific information depends on the relative importance of the accuracy of the preference model and the cognitive load on the user. If the burden on the user is unimportant and accuracy of the preference model is of paramount importance, then a lengthy elicitation process should be followed. If, by contrast, no queries of any sort are permitted, then only *a priori* information and passive observations are available.

If, as is more typically the case, a small number of elicitations, such as candidates or queries, are permitted, the timing and contents of the elicitations are critical for maximizing the trade-off between ranking accuracy and cognitive load.

This paper describes *UGAMA*, an approach to acquiring instances for learning preference criteria through active exploration. The next section defines the preference learning task and describes previous approaches to preference learning by passive observation. Section 3 describes UGAMA, and Section 4 sets forth an empirical evaluation showing that for many target quality functions UGAMA leads to much faster acquisition of preference criteria than learning with an equal number of random observations. The scope of the results and its implications for representation design are described in the last section.

2 The Preference Learning Task

The preference learning task arises in many domains—typified by design and configuration problems—in which the relevant characteristics of problem-solving states can be identified by users or by experts, but users differ as to or are unable to articulate evaluation criteria for problem solving states in terms of these attributes.

For example, in the task of modeling individual preferences for two-dimensional designs, experts in design can identify the characteristics of designs that determine their quality, such as symmetry, contrast, and balance. Moreover, each of these characteristics can be expressed as a numerical or symbolic feature. But the precise manner in which these characteristics combine to determine the overall effectiveness of a design varies with each individual and is quite difficult for a given individual to articulate. Similarly, in the personal scheduling task, the relevant characteristics of schedules may be easy to identify, but their relative importance and interaction may both vary among individuals and be difficult for each individual to articulate.

A preference of user u is a binary relation, P_u such that $P_u(S_1, S_2)$ is satisfied whenever user u prefers S_1 to S_2. Various approaches have been taken to representing such relations. One approach rests on the assumption that a value function, $v_u(S)$, expressing the quality of state S, underlies P_u [KR93]. Thus, $P_u(S_1, S_2)$ is satisfied whenever $v_u(S_1) > v_u(S_2)$. A second approach subsumes preference model acquisition under supervised concept acquisition by viewing the problem of determining whether state S_1 is preferred to state S_2 as equivalent to determining whether the concatenation of S_1 with S_2, $concat(S_1, S_2)$, is an instance of the category "is-preferred-to." Under this approach each ranked pair $< S_1, S_2 >$ for which $P_u(S_1, S_2)$ is converted into a pair of training instances: $concat(S_1, S_2) \in$ "is-preferred-to" and $concat(S_1, S_2) \notin$ "is-preferred-to". For example, perceptron learning and decision-tree induction were applied to preference acquisition in [US87], [UC91], and [CFR91].

A third, intrinsically instance-based, approach represents preference pairs as arcs in feature space and ranks new pairs through nearest-neighbor algorithms, such as 1ARC or CIBL [BB94,BB97]. For example, the set of ranked pairs $\{P_u(A, B), P_u(C, D), P_u(E, F)\}$ can be represented as shown in Figure 1 by the preference arcs $\overleftarrow{AB}$, $\overleftarrow{CD}$, and $\overleftarrow{EF}$ (where $\overleftarrow{AB} \equiv P_u(A, B)$).

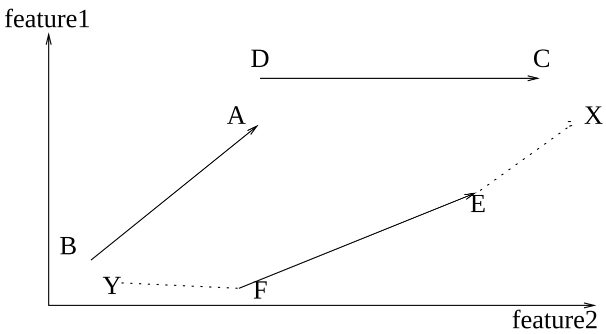

Fig. 1. X is ranked higher than Y by 1ARC because of the match between hypothesis $\overleftarrow{XY}$ and preference arc $\overleftarrow{EF}$. The dissimilarity between $\overleftarrow{XY}$ and $\overleftarrow{EF}$ is the sum of the Euclidean distances represented by dotted lines.

In the 1ARC algorithm, a new pair of objects, X and Y is ranked by determining whether $\overleftarrow{XY}$ or $\overleftarrow{YX}$ has the better match to a ranked pair in the training set. The dissimilarity between a hypothesis, e.g., $\overleftarrow{XY}$, and a ranked pair is measured by the sum of the Euclidean distances between (1) Y and the tail of the ranked pair and (2) X and the head of the preference pair. In Figure 1, for example, the ranked pair $\overleftarrow{EF}$ best matches $\overleftarrow{XY}$ with a dissimilarity of $dist(Y,F)+dist(X,E)$, represented by the sum of the lengths of the dotted lines. The best match for the alternative hypothesis $\overleftarrow{YX}$ is determined in the same way. In this case, $\overleftarrow{XY}$ matches ranked pair $\overleftarrow{EF}$ more strongly than $\overleftarrow{YX}$ matches any ranked pair, so $P_u(X,Y)$ is predicted.

Common to all these previous approaches to preference predicate acquisition is the assumption that the learning algorithm has no control over the choice of instances.

3 UGAMA

This section explores the implications of relaxing the assumption that a preference learning is not permitted to choose instances to learn from, proposing an approach based on two ideas: acquisition of Unit Gradient Approximations (UGAs); and merging and alignment of UGAs with respect to inflections (*i.e., changes in derivative sign* in the underlying quality function (UGAMA).

3.1 Unit Gradient Approximations

An estimation of the gradient of a quality function at a single point in feature space can be obtained as follows. Let R be a point (termed a *reference point*) in feature space. For each dimension d, create a pair $< R_{-d}, R_{+d} >$ by subtracting (respectively, adding) a small increment δ from (to) the value of R in the d dimension. If the user ranks $< R_{-d}, R_{+d} >$ as equal, the d dimension is irrelevant at R. If R_{-d} is ranked better than R_{+d}, Q has negative slope with respect to d at R; if R_{+d} is preferred, the slope is positive at R. For example, Figure 2 shows how points P1 and P2 are δ larger and smaller, respectively, than reference point R in dimension 1, and points P3 and P4, are δ larger and smaller, respectively, than R in dimension 2. If user ranks $P_u(P2,P1)$ and $P_u(P4,P3)$, the UGA has a slope of $< 1, -1 >$.

If there are n dimensions, then n queries are sufficient to determine the relevance and polarity of each dimension. This information can be expressed in a single pair, $\overleftarrow{HT}$, called a *unit gradient approximation* (UGA), in which H and T are identical to R in irrelevant dimensions, H is δ greater than and T δ less than R in dimensions with positive slope, and H is δ less than and T δ greater than R in dimension with negative slope.

If the quality function happens to be a linear function whose coefficients are all either k, $-k$, or 0, for some constant k, then the UGA will be parallel to the gradient of the function.[1] Under these circumstances, a single UGA is a sufficient

[1] For example, suppose that the quality function $Q(x_1,x_2,x_3,x_4) = 2x_1 - 2x_3 + 2x_4$, the reference point is $< .5, .5, .5, .5 >$, and $\delta = .1$. Under these circumstances, the UGA

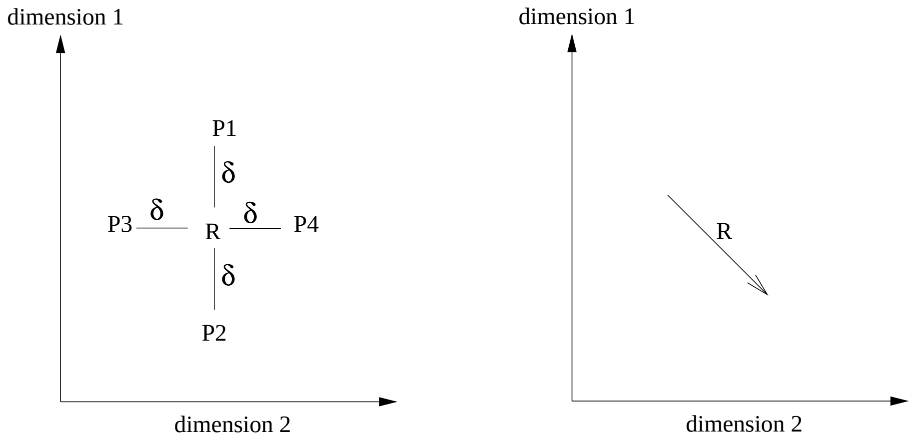

Fig. 2. Determining the relevance and polarity of each dimension, and forming a UGA. If user ranks $P_u(P2, P1)$ and $P_u(P4, P3)$, the UGA has a slope of $< 1, -1 >$

training set for 1ARC to achieve perfect accuracy, that is, correctly rank all pairs (see Theorem 1, Appendix). As shown in Table 1, ranking accuracy given a 4-dimensional linear function defined on $[0, 1]^4$ with 50% irrelevant features is 100% for both perceptron and 1ARC with a single UGA as training data, as compared to only 69.9% for 1ARC and 71.1% for perceptron with a training set consisting of 4 random instances (in 10 trials of 128 random test cases each).

Table 1. Ranking accuracy with linear quality function in 4 dimensions, two of which are irrelevant and two of which have identical weights.

	1ARC	Perceptron
Random	69.9	71.1
UGA	100	100

Of course, if the coefficients of the underlying quality function differ by factors other than 1, -1, or 0, the UGA will no longer be parallel to the gradient and will therefore no longer guaranteed to rank new pairs correctly. For example, given quality function $Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} 2^i x_i$, the ranking accuracy with a single UGA is considerably lower than with unit weights. However, as shown in Table 2, the ranking accuracy is still higher than with an equal number of random instances (4 dimensions, 10 trials of 128 test cases each). In practice, such extreme variations in the weight of relevant attributes coefficients (*i.e.,* in the coefficients of the quality function) seem unlikely.

---

will be $(< .6, .5, .4, .6 >< .4, .5, .6, .4 >)$. The slope of this instance is $< .2, 0, -.2, .2 >$, which is parallel to the gradient of Q, $< 2, 0, -2, 2 >$.

Table 2. Ranking accuracy with linear quality function in 4 dimensions with coefficient 2^d for dimension d.

	1ARC	Perceptron
Random	67.6	72.2
UGA	79.9	80.0

3.2 Inflected Quality Functions

The nature of the quality function underlying a person's preferences depends both on the preferences themselves and on the representation of the attributes used to characterize the instances. A quality function may be linear when defined on an optimal set of attributes, but nonlinear when defined on suboptimal attributes. Ideally, a preference learning task should be defined in such a way that user's quality functions defined on those attributes should be linear. But in practice it seems unlikely that a representation guaranteed to lead to linear quality functions for all users can be found for all domains.

For example, the width-to-height ratio of two-dimensional designs is a factor that affects many peoples' preferences for designs. Some people may prefer a width-to-height ratio near the "golden mean," $(1+\sqrt{5})/2$, while others may prefer a unit width-to-height ratio. If the width-to-height ratio attribute of designs were replaced with a distance-from-golden-mean attribute, the function would become linear in the attribute for people in the first group, but the unit width-to-height ratio would be indistinguishable from $\sqrt{5}$ (since both are an equal distance from $(1+\sqrt{5})/2$). Similarly, if a distance-from-unit-ratio attribute were used, the golden mean could no longer be distinguished from $2-(1+\sqrt{5})/2$. Thus, width-to-height ratio itself must be used as a feature if both preferences are to be precisely expressible. However, if the width-to-height ratio is used, then there will be an inflection in the quality function at the golden-mean for people in the first group and at 1 for people in the second group. This example shows that it may not always be feasible to devise a representation that is linear in all attributes because users may differ as to the values of an attribute that they consider optimal.

Clearly, a single UGA is not sufficient to represent a preference predicate based on a nonlinear quality function. If the quality function has inflections, then multiple UGAs must be obtained. Only if at least one UGA has been obtained for each linear region is an accurate preference model possible. Since each UGA requires n queries, where n is the number of dimension, the user's patience is likely to be exhausted if the number of dimensions and linear regions is high. Therefore, it appears that the key condition under which an algorithm for preference acquisition through exploration must work is when the number of inflections in the users' quality function is greater than zero but not too large.

A single perceptron is not capable of expressing nonlinear concepts. However, the 1ARC algorithm is capable of modeling nonlinear quality functions provided that there is at least one ranked pair per linear region. This suggests the strategy of eliciting a set of UGAs at random points and using them as the training set for 1ARC.[2]

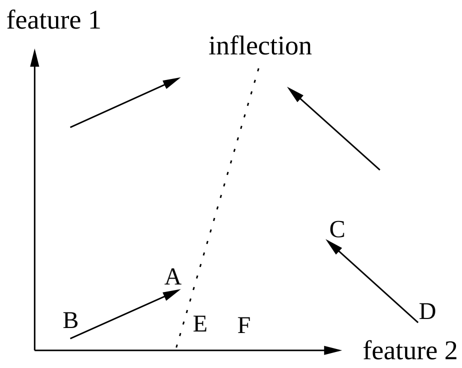

Fig. 3. The pair $< E, F >$, for which $Q(E) > Q(F)$, is misranked because $\overleftarrow{FE}$ matches $\overleftarrow{AB}$ more closely than $\overleftarrow{EF}$ matches $\overleftarrow{CD}$.

The limitation of this approach is that because 1ARC is a nearest-neighbor algorithm, the position of the UGAs within each linear region affects ranking accuracy. An example is illustrated in Figure 3, in which the dotted line represents the inflection between two linear regions. Since $\overleftarrow{AB}$ is much nearer to the inflection than $\overleftarrow{CD}$, the $\overleftarrow{FE}$ matches $\overleftarrow{AB}$ more closely than $\overleftarrow{EF}$ matches $\overleftarrow{CD}$. As a result, the pair $< E, F >$ is misranked.

3.3 Merging and Aligning UGAs

Merging and alignment is a procedure to reduce this effect. As set forth in Figure 4 and illustrated in Figure 5, UGAs with identical slopes that are closer to each other than to UGAs with different slopes are merged. Merging a set S of arcs consists of forming the arc $\overleftarrow{H_m, T_m}$, where H_m is the mean of the heads of the arcs in S and T_m is the mean of the tails of the arcs in S. The merged UGAs from adjacent regions are then displaced, without changing their slope, until their heads (or tails, if the tails are closer to each other than the heads) coincide at the midpoint between their original positions. The purpose of this displacement is to align the endpoints of the UGAs so as to coincide as closely as possible with the inflection in the quality function. Choosing the midpoint of the heads (or tails) is simply a heuristic for estimating the position of the inflection. As shown in Theorem 2, Appendix, if two arcs each parallel to the

[2] Of course, if domain knowledge exists from which one point per linear region can be selected, this knowledge should be used to create the minimal set of UGAs. However, in the general case it is not known how many linear regions there are.

```
Procedure MERGE-AND-ALIGN(UGASET)
Input:  UGASET is a list of UGAs
Output: UGAMASET is list of merged and aligned UGAs
1. Let MERGERS and UGAMASET be {}
2. Let ECLASSES be a partition of UGASET into sets with equal slope
3. For each class C in ECLASSES do
  a. Let SC be a partition of C into the largest sets such that every
     member of SC is closer to some other member of SC than to any member
     of UGASET with a different slope.
  b. For every partition P in SC do
       Add the arc M consisting of mean of every arc in P to MERGERS
4. For each pair of arcs (A1, A2), where A1, A2 are in MERGERS
    LET M be the mean of A1 and A2.
    IF A1 and A2 have different slopes AND M is closer
    to A1 [equivalently, A2] than to any other arc in MERGERS
    THEN {IF the heads of A1 and A2 are closer to each other than the tails
          THEN {Let A1' and A2' be the result of displacing A1 and A2 so that
          their heads coincide at the mean of the heads' original positions}
          ELSE {Let A1' and A2' be the result of displacing A1 and A2 so that
          their tails coincide at the mean of the tail's original positions}
          Add A1' and A2' to UGAMASET.}
5. Return UGAMASET
```

Fig. 4. The merge-and-adjust algorithm.

gradient are symmetric around a single inflection and share a common endpoint, 1ARC will correctly rank all pairs, given the two arcs as a training set. The entire procedure of forming UGAs through successive queries, then merging and aligning the UGAs is termed *UGAMA*.

4 Experimental Evaluation

Theorem 2's guarantee of ranking correctness does not extend to functions with multiple inflections. How well does UGAMA perform with functions with multiple inflections, which are likely to be more typical of actual user quality functions? To answer this question, an evaluation was performed with a set of artificial quality functions.

The experiments were performed on a 4-dimensional feature space, $[0,1]^4$ with 6 artificial quality functions intended to resemble human quality functions. The first quality function, `independent`, shown in Figure 6, is linear in even-numbered dimensions and inflected at 0.5 in odd-numbered dimensions. This corresponds to a domain, like 2-dimensional design, where some dimensions (e.g., width-to-height ratio) are inflected and others (e.g., balance) are not. In `dependent` the quality function is inflected in the sum of successive pairs of dimensions, e.g., for 2 dimensions if $d1 + d2 < 1$, $Q(d1, d2) = d1 + d2$, otherwise $Q(d1, d2) = 2 - (d1 + d2)$. This corresponds to a quality function

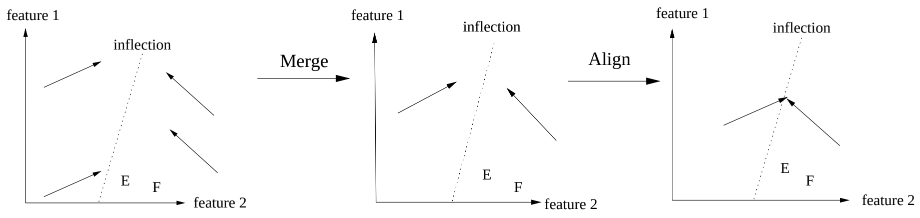

Fig. 5. An example of merging and aligning UGAs. The pair $< E, F >$, incorrectly ranked by the original UGAs, is correctly ranked by the merged and adjusted UGAs

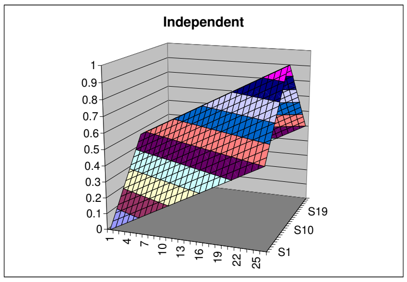

Fig. 6. The 2-dimensional analog of quality function `independent`. The vertical axis represents quality as a function of two features.

with pairwise interactions between dimensions. In `sinusoid .5`, Q is the sine of the sum of the dimensions normalized to range from $[0..\pi]$. `Exponential` is $Q(d1, d2, d3, d4) = 1 - e^{\sqrt{(d1^2+d2^2+d3^2+d4^2)/4}}$. In `double fold`, shown in Figure 7, Q consists of 4 linear regions with inflections perpendicular to the line d1 = d2 = d3 = d4, and `pyramid` consists of 4 linear regions intersecting at (0.5,0.5,0.5,0.5).

In each test, 8 random reference points were selected to create 8 UGAs (through 32 queries to the test function). The accuracy in ranking randomly selected pairs using the UGAs both before and after merging and alignment was compared to accuracy using 32 random instances. Each function was tested with 10 repetitions of 128 random testing instances each.

Figure 8 sets forth the results using 1ARC as the learning mechanism. For each function, UGAs resulted in higher ranking accuracy than did the random training instances, and merging and alignment produced an additional improvement in every function except `exponential`. Merging and alignment produces no improvement in `exponential` because merging results in a single arc.

Non-instance-based learning methods are benefited relatively little by the UGAMA approach. Briefly, perceptron performs at the chance level on inflected quality functions. UGAMA does not improve the performance of decision-tree in-

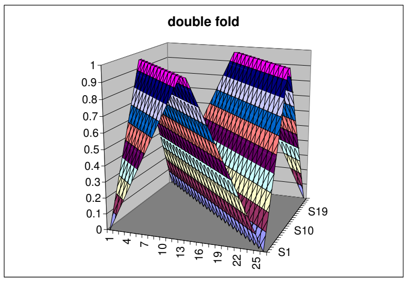

Fig. 7. The 2-dimensional analog of quality function `double-fold`. The vertical axis represents quality as a function of two features.

duction (ID3) or backpropagation, which perform with random instances, UGAs, and UGAMA at approximately the same level as 1ARC given random instances. This result is consistent with previous research, which has shown that instance-based learning methods tend to work better than greedy generalizers when there is a very small number of training instances [BB97,Aha92], such as result from the elicitation of UGAs. Identification of exploration techniques that are appropriate for these alternative preference-learning methods is an open question.

5 Conclusion

This paper has presented an approach for acquiring instance-based preference models through active exploration. The empirical evaluation showed that UGAMA lead to more rapid acquisition of preference predicates than training sets of random instances. The results with *independent* showed that a ranking accuracy of over 80% can be obtained on a quality function with inflections in 2 different dimensions after 32 queries.

The next step in research in acquisition of preference predicates through exploration should be testing with human subjects. The actual complexity of human preference criteria in representative domains is unknown. The performance requirements for preference acquisition algorithms will be better understood when there are well-analyzed sets of human preference data. A second issue in human preference testing is the number of queries that users will tolerate. This probably depends on the complexity of the instances being ranked and on the level of useful assistance that can be expected from the investment of users' effort. A third issue is the amount of noise or inconsistency in human preference rankings. This factor determines the extent to which preference learning algorithms must be noise tolerant.

In view of the dramatic effect that quality function complexity has on the number of instances needed to learn a preference model, design of representations

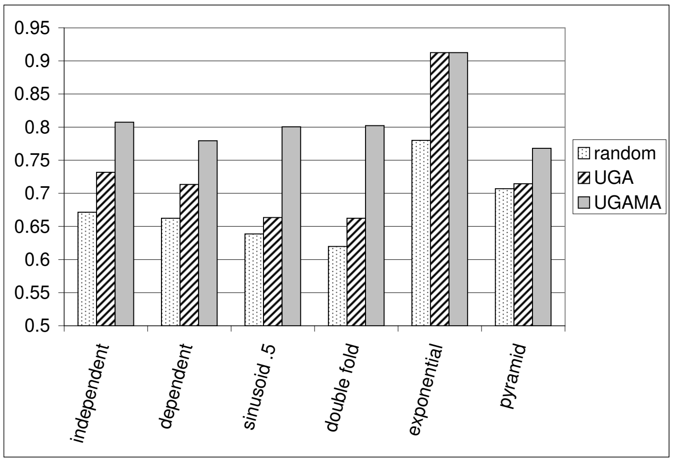

Fig. 8. A comparison of the ranking accuracy of 1ARC using random instances, UGAs, and UGAMA on 6 quality functions.

for which users' quality functions are as nearly linear as possible is clearly essential. However, in many domains some nonlinearity appears to be unavoidable. The UGAMA approach may therefore be a valuable tool for active preference predicate acquisition for such domains.

Acknowledgments

This research was supported in part by a German-American Fulbright Commission Senior Scholars grant, a University of Wyoming Flittie sabbatical award, and the University of Kaiserslautern Center for Learning Systems and Applications.

Appendix

Theorem 1.
With a training set consisting of a single preference instance parallel to the gradient of a linear quality function, 1ARC correctly ranks all pairs with respect to the quality function.

Proof.
Let $Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i$ be a linear quality function of n features. The gradient of Q is the vector G=$< a_1, \ldots, a_n >$. A ranked pair parallel to G in feature space must be of the form

$$P_Q(A, B) \equiv (< u_1 + ca_1, \ldots, u_n + ca_n >< u_1, \ldots, u_n >) \tag{1}$$

where c is a positive constant.

Let (W,Z) $\equiv$ ($< w_1, \ldots, w_n >< z_1, \ldots, z_n >$) be a testing pair. 1ARC ranks W and Z by finding whether $\overleftarrow{AB}$ more closely matches $\overleftarrow{WZ}$ or $\overleftarrow{ZW}$. The distance between $\overleftarrow{AB}$ and $\overleftarrow{WZ}$ is

$$dist(A, W) + dist(B, Z) = \sum_{i=1}^{n}[(u_i + ca_i - w_i)^2 + (u_i - z_i)^2] \tag{2}$$

Similarly, the distance between $\overleftarrow{AB}$ and $\overleftarrow{ZW}$ is

$$dist(A, W) + dist(B, Z) = \sum_{i=1}^{n}[(u_i + ca_i - z_i)^2 + (u_i - w_i)^2] \tag{3}$$

Thus, 1ARC will rank W as preferable to Z only if

$$\sum_{i=1}^{n}[(u_i + ca_i - w_i)^2 + (u_i - z_i)^2] < \sum_{i=1}^{n}[(u_i + ca_i - z_i)^2 + (u_i - w_i)^2] \tag{4}$$

However, this inequality can be simplified to

$$\sum_{i=1}^{n} w_i a_i > \sum_{i=1}^{n} z_i a_i \tag{5}$$

which is equivalent to $Q(W) > Q(Z)$.
Similarly, 1ARC will rank Z as preferable to W only if

$$\sum_{i=1}^{n}[(u_i + ca_i - w_i)^2 + (u_i - z_i)^2] > \sum_{i=1}^{n}[(u_i + ca_i - z_i)^2 + (u_i - w_i)^2] \tag{6}$$

which can be simplified to

$$\sum_{i=1}^{n} w_i a_i < \sum_{i=1}^{n} z_i a_i \tag{7}$$

which is equivalent to $Q(W) < Q(Z)$. Thus, the single training pair P_Q(A,B) correctly ranks all testing pairs with respect to Q.

Theorem 2.
Let Q be a piecewise linear function symmetrical around a single linear inflection, that is, let the components of the slope on both sides of the inflection have the same magnitude, but let at least one component differ in sign. Then with a training set consisting of two ranked pairs that (1) share a common endpoint, (2) are reflections of one another across the inflection, and (3) are each parallel to the gradient of the quality function, 1ARC correctly ranks all pairs with respect to Q.

Proof.
Assume without loss of generality that the shared endpoint is the preferred point, as pictured in Figure 9 for training pairs $\overleftarrow{E_1E_2}$ and $\overleftarrow{E_1E_3}$ (an analogous argument can be made if the shared endpoint is the less-preferred point). Any two

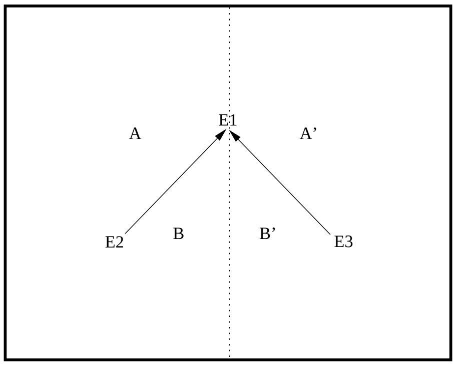

Fig. 9. The inflection between two linear regions is indicated by the dotted line.

pairs to be ranked must either both be on the same side of the inflection or they must be on different sides of the inflection.

(a) Same side
Let A and B be points to be ranked, and suppose that their actual ranking is $\overleftarrow{AB}$, i.e., $\mathrm{Q}(A) > \mathrm{Q}(B)$ (if not, rename the points). Under Theorem 1,

$$dist(A, E_1) + dist(B, E_2) < dist(A, E_2) + dist(B, E_1) \tag{8}$$

That is, $\overleftarrow{AB}$ is correctly ranked by $\overleftarrow{E_1E_2}$. Thus, $\overleftarrow{AB}$ could be misranked only if

$$dist(A, E_3) + dist(B, E_1) < dist(A, E_1) + dist(B, E_2) \tag{9}$$

However, since E_2 and E_3 are symmetrical across the inflection, the inflection represents the set of points equidistance from E_2 and E_3. Any point on the same side of the inflection as E_2 is closer to E_2 than to E_3. Therefore, $dist(A, E_3) > dist(A, E_2)$, so

$$dist(A, E_2) + dist(B, E_1) < dist(A, E_3) + dist(B, E_1) \tag{10}$$

Inequalities 8 and 10 together imply that:

$$dist(A, E_1) + dist(B, E_2) < dist(A, E_3) + dist(B, E_1) \tag{11}$$

which contradicts inequality 9. Therefore, $\overleftarrow{AB}$ will be correctly ranked by $\overleftarrow{E_1E_2}$.

(b) Different sides
Let A and B' be points to be ranked, and again suppose without loss of generality that their actual ranking is $\overleftarrow{AB'}$.

$\overleftarrow{AB'}$ could not be incorrectly ranked unless either

$$dist(A, E_2) + dist(B', E_1) < dist(A, E_1) + dist(B', E_3) \tag{12}$$

or

$$dist(A, E_3) + dist(B', E_1) < dist(A, E_1) + dist(B', E_3) \tag{13}$$

Let B be the reflection of B' across the inflection and let A' be the reflection of A across the inflection. Then

$$dist(B, E_2) = dist(B', E_3). \tag{14}$$

and

$$dist(A, E_1) = dist(A', E_1) \tag{15}$$

Theorem 1 implies that

$$dist(A, E_1) + dist(B, E_2) < dist(A, E_2) + dist(B, E_1) \tag{16}$$

and

$$dist(A', E_1) + dist(B', E_3) < dist(A', E_3) + dist(B', E_1) \tag{17}$$

Substituting $dist(B', E_3)$ for $dist(B, E_2)$ in 16 we obtain:

$$dist(A, E_1) + dist(B', E_3) < dist(A, E_2) + dist(B, E_1) \tag{18}$$

which contradicts 12. Moreover, substituting $dist(A, E_1)$ for $dist(A', E_1)$, and $dist(A', E_3)$ for $dist(A, E_2)$ in 17 we obtain:

$$dist(A, E_1) + dist(B', E_3) < dist(A, E_3) + dist(B', E_1) \tag{19}$$

which contradicts 13.

Since, $dist(A, E_1) + dist(B', E_3)$ is less than either $dist(A, E_2) + dist(B', E_1)$ or $dist(A, E_3) + dist(B', E_1)$, $\overleftarrow{AB'}$ will be correctly ranked by $\overleftarrow{E_1E_3}$.

References

[Aha92] D. Aha. Generalizing from case studies: A case study. In *Proceedings of the Ninth International Workshop on Machine Learning*, pages 1–10, 1992.

[BB94] P. Broos and K. Branting. Compositional instance-based learning. In *Proceedings of the Twelfth National Conference Conference on Artificial Intelligence (AAAI-94)*, Seattle, Washington, July 31–August 4, 1994.

[BB97] K. Branting and P. Broos. Automated acquisition of user preferences. *International Journal of Human-Computer Studies*, 46:55–77, 1997.

[Bra99] K. Branting. Learning user preferences by exploration. In *The Sixteenth International Conference on Machine Learning*, 27–30 June 1999 1999. Under review.

[CFR91] J. Callan, T. Fawcett, and E. Rissland. Adaptive case-based reasoning. In *Proceedings of the Third DARPA Case-Based Reasoning Workshop*, pages 179–190. Morgan Kaufmann, May 1991.

[DBM+92] L. Dent, J. Boticario, J. McDermott, T. Mitchell, and D. Zabowski. A personal learning apprentice. In *Proceedings of Tenth National Conference on Artificial Intelligence*, pages 96–103, San Jose, CA, July 12–16 1992. AAAI Press/MIT Press.

[GNOT92] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[HGBSO98] Ralf Herbirch, Thore Graepel, Peter Bollmann-Sdorra, and Klaus Obermayer. Learning preference relations for information retrieval. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*. AAAI Press, July 26–27 1998.

[KR93] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, second edition, 1993.

[Mae94] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–40, 1994.

[MK93] P. Maes and R. Kozierok. Learning interface agents. In *Proceedings of Eleventh National Conference on Artificial Intelligence*, pages 459–466, Washington, D.C., July 11–15 1993. AAAI Press/MIT Press.

[UC91] P. Utgoff and J. Clouse. Two kinds of training information for evaluation function learning. In *Proceedings of Ninth National Conference on Artificial Intelligence*, pages 596–600, Anaheim, July 14–19 1991. AAAI Press/MIT Press, Menlo Park, California.

[US87] P. Utgoff and S. Saxena. Learning a preference predicate. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 115–121, 1987.