Pairwise Preference Learning and Ranking

Johannes Fürnkranz
Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Wien, Austria
juffi@oefai.at

Eyke Hüllermeier
Informatics Institute, Marburg University
Hans-Meerwein-Str., Lahnberge, D-35032 Marburg, Germany
eyke@mathematik.uni-marburg.de

Technical Report OEFAI-TR-2003-14

Abstract

We consider supervised learning of a ranking function, which is a mapping from instances to total orders over a set of labels (options). The training information consists of examples with partial (and possibly inconsistent) information about their associated rankings. From these, we induce a ranking function by reducing the original problem to a number of binary classification problems, one for each pair of labels. The main objective of this work is to investigate the trade-off between the quality of the induced ranking function and the computational complexity of the algorithm, both depending on the amount of preference information given for each example. To this end, we present theoretical results on the complexity of pairwise preference learning. We also carry out some controlled experiments investigating the predictive performance of our method for different types of preference information, such as top-ranked labels and complete rankings. The domain of this study is the prediction of a rational agent's ranking of actions in an uncertain environment.

1 Introduction

The increasing trend to treat consumers, computer users and patients as *individuals* has produced, among other things, user-adapted software and operating systems (Horvitz et al., 1998), e-commerce personalization of products and services (Riecken, 2000), and systems for patient-centered medical care (Couch, 1998). A

key prerequisite in all of these applications is the ability of discovering and capturing an individual's *preferences*, a problem often referred to as *preference elicitation*.

We consider the acquisition of preferences in the context of supervised learning. Roughly speaking, this means to generalize given examples to a "preference structure-valued" function, that is, a function which assigns preference structures to instances (computer users, customers, patients, ...). This problem, which can obviously be seen as an extension of learning a classification function, will be referred to as *preference learning*. It should be distinguished from preference elicitation in a more narrow sense, where the goal is to learn about the preferences of a single individual, and where specific questions can be asked to that individual.[1]

The problem of learning with or from preferences has recently received a lot of attention within the machine learning literature. The problem is particularly challenging because it involves the prediction of complex structures, such as weak or partial order relations, rather than single values. Moreover, training input will not, as it is usually the case, be offered in the form of complete examples but may comprise more general types of information, such as relative preferences or different kinds of indirect feedback.

More specifically, the learning scenario that we will consider in this paper consists of a collection of training examples which are associated with a finite set of decision alternatives. Following the common notation of supervised learning, we shall refer to the latter as *labels*. However, contrary to standard classification, a training example is not assigned a single label, but a set of *pairwise preferences* between labels, expressing that one label is preferred over another.

The goal is to use these pairwise preferences for predicting a total order, a *ranking*, of all possible labels for a new training example. More generally, we seek to induce a *ranking function* that maps instances (examples) to rankings over a fixed set of decision alternatives (labels), in analogy to a *classification function* that maps instances to single labels. To this end, we investigate the use of *round robin learning* or *pairwise classification*. As will be seen, round robin appears particularly appealing in this context since it can be extended from classification to preference learning in a quite natural manner.

The paper is organized as follows: In the next section, we introduce the learning problem in a formal way. The extension of pairwise classification to pairwise preference learning and its application to ranking are discussed in section 3. Section 4 provides some results on the computational complexity of pairwise preference learning. Results of several experimental studies investigating the predictive

[1] Here, the major problem is to ask such questions in a clever way, so as to find a good approximation of the individual's preference structure with an as small as possible number of questions.

performance of our approach under various training conditions are presented in section 5. We conclude the paper with an overview of related work in section 6 and some complementary final remarks in section 7.

2 Learning Problem

We consider the following learning problem:

Given:

- a set of *labels* $L = \{\lambda_i \,|\, i = 1 \dots c\}$
- a set of *examples* $E = \{e_k \,|\, k = 1 \dots n\}$
- for each training example e_k:
  - a set of *preferences* $P_k \subseteq L \times L$, where $(\lambda_i, \lambda_j) \in P_k$ indicates that label λ_i is preferred over label λ_j for example e_k.

Find: a function that orders the labels $\lambda_i, i = 1 \dots c$ for any given example.

We will abbreviate $(\lambda_i, \lambda_j) \in P_k$ with $\lambda_i \succ_k \lambda_j$, or even $\lambda_i \succ \lambda_j$ if the particular example e_k doesn't matter or is clear from the context.

This setting has been previously introduced as *constraint classification* by Har-Peled et al. (2002). As has been pointed out in their work, the above framework is a generalization of several common learning settings, in particular (see ibidem for a formal derivation of these and other results)

- *ranking:* Each training example is associated with a total order of the labels, i.e., for each pair of labels (λ_i, λ_j) either $\lambda_i \succ \lambda_j$ or $\lambda_j \succ \lambda_i$ holds.
- *classification:* A single class label λ_i is assigned to each example. This implicitly defines the set of preferences $\{\lambda_i \succ \lambda_j \,|\, 1 \leq j \neq i \leq c\}$.
- *multi-label classification:* Each training example e_k is associated with a subset $S_k \subseteq L$ of possible labels. This implicitly defines the set of preferences $\{\lambda_i \succ \lambda_j \,|\, \lambda_i \in S, \lambda_j \in L \setminus S\}$.

As pointed out before, we will be interested in predicting a ranking (total order) of the labels. Thus, we assume that for each instance, there exists a total order of the labels, i.e., they form a transitive and asymmetric relation. For many practical applications, this assumption appears to be acceptable at least for the *true* preferences. Still, more often than not the observed or *revealed* preferences will be

incomplete or inconsistent. Therefore, we do not require the *data* to be consistent in the sense that transitivity and asymmetry applies to the P_k. In fact, this property is not compulsory for our learning algorithm. Yet, we do make the reasonable assumption that P_k is irreflexive ($\lambda_i \not\succ \lambda_i$) and anti-symmetric ($\lambda_i \succ \lambda_j \Rightarrow \lambda_j \not\succ \lambda_i$). (Note that $0 \leq |P_k| \leq c(c-1)/2$ as a consequence of the last two properties.)

3 Pairwise Preference Ranking

A key idea of our approach is to learn a separate theory for each of the $c(c-1)/2$ pairwise preferences between two labels. More formally, for each possible pair of labels (λ_i, λ_j), $1 \leq i < j \leq c$, we learn a model m_{ij} that decides for any given example whether $\lambda_i \succ \lambda_j$ or $\lambda_j \succ \lambda_i$ holds. The model is trained with all examples e_k for which either $\lambda_i \succ_k \lambda_j$ or $\lambda_j \succ_k \lambda_i$ is known. All examples for which nothing is known about the preference between λ_i and λ_j are ignored.

At classification time, an example is submitted to all $c(c-1)/2$ theories, and each prediction is interpreted as a vote for a label. If classifier m_{ij} predicts $\lambda_i \succ \lambda_j$, we count this as a vote for λ_i. Conversely, the prediction $\lambda_j \succ \lambda_i$ would be considered as a vote for λ_j. The labels are ranked according to the number of votes they receive from all models m_{ij}. Ties are first broken according to the frequency of the labels in the top rank (the class distribution in the classification setting) and then randomly.

We refer to the above technique as *pairwise preference ranking* or *round robin ranking*. It is a straight-forward generalization of pairwise or one-against-one classification, aka round robin learning, which solves multi-class problems by learning a separate theory for each pair of classes. In previous work, Fürnkranz (2002) showed that, for rule learning algorithms, this technique is preferable to the more commonly used one-against-all classification method, which learns one theory for each class, using the examples of this class as positive examples and all others as negative examples. Round robin has also been successfully used in other fields, in particular in the area of support vector machines (Hsu and Lin, 2002, and references therein). We refer to Section 8 of (Fürnkranz, 2002) for a brief survey of related work on pairwise classification.

More importantly, however, Fürnkranz (2002) showed that, despite its complexity being quadratic in the number of classes, the algorithm is no slower than the conventional one-against-all technique. We will generalize these results in the next section.

4 Complexity

Consider a learning problem with n training examples and c labels.

Theorem 4.1 *The total number of training examples over all $c(c-1)/2$ binary preference learning problems is*

$$\sum_{k=1}^{n} |P_k| \leq n \max_k |P_k| \leq n \binom{c}{2} = n \frac{c(c-1)}{2}$$

Proof: Each of the n training examples will be added to all $|P_k|$ binary training sets that correspond to one of its preferences. Thus, the total number of training examples is $\sum_{k=1}^{n} |P_k|$. As the number of preferences for each example is bounded from above by $\max_k |P_k|$, this number is no larger than $\max_k |P_k| n$, which in turn is bounded from above by the size of a complete set of preferences $nc(c-1)/2$. □

From this immediately follows a result of Fürnkranz (2002):

Corollary 4.2 *For a classification problem, the total number of training examples is only linear in the number of classes.*

Proof: A class label expands to $c-1$ preferences, therefore $\sum_{k=1}^{n} |P_k| = (c-1)n$. □

Note that we only considered the number of training examples, but not the complexity of the learner that runs on these examples. For an algorithm with a linear run-time complexity $O(n)$ it follows immediately that the total run-time is $O(dn)$, where d is the maximum (or average) number of preferences given for each training example. For a learner with a super-linear complexity $O(n^a), a > 1$, the total run-time is much lower than $O((dn)^a)$ because the training effort is not spent on one large training set, but on many small training sets. In particular, for a complete preference set, the total complexity is $O(c^2 n^a)$, whereas the complexity for $d = c - 1$ (round robin classification) is only $O(cn^a)$ (Fürnkranz, 2002).

For comparison, the only other technique for learning in this setting that we know of (Har-Peled et al., 2002) constructs twice as many training examples (one positive and one negative for each preference of each example), and these examples are projected into a space that has c times as many attributes as the original space. Moreover, all examples are put into a single training set for which a separating hyper-plane has to be learned. Thus, under the (reasonable) assumption

that an increase in the number of features has approximately the same effect as a corresponding increase in the number of examples, the total complexity becomes $O((cdn)^a)$ if the algorithm for finding the separating hyper-plane has complexity $O(n^a)$ for a two-class training set of size n.

In summary, the overall complexity of pairwise constraint classification depends on the (maximum or average) number of preferences that are given for each training example. While being quadratic in the number of labels if a complete ranking is given, it is only linear for the classification setting. In any case, it is more efficient than the technique proposed by Har-Peled et al. (2002). However, it should be noted that the price to pay is the large number of classifiers that have to be stored and tested at classification time.

5 Empirical Results

The previous sections have shown that an extended version of round robin learning can induce a ranking function from a set of preferences instead of a single label. Yet, it turned out that computational complexity might become an issue. Especially, since a ranking induces a quadratic number of pairwise preferences, the complexity for round robin ranking becomes quadratic in the number of labels. In this context, one might ask whether it could be possible to improve efficiency at the cost of a tolerable decrease in performance: Could the learning process perhaps ignore some of the preferences without decreasing predictive accuracy too much? Apart from that, incomplete training data is clearly a point of practical relevance, since complete rankings will rarely be observable.

The experimental evaluation presented in this section is meant to investigate issues related to incomplete training data in more detail, especially to increase our understanding about the trade-off between the number of pairwise preferences available in the training data and the quality of the learned ranking function. For a systematic investigation of questions of such kind, we need data for which, in principle, a complete ranking is known for each example. This information allows a systematic variation of the amount of preference information in the training data, and a precise evaluation of the predicted rankings on the test data. Since we were not aware of any suitable real-world datasets, we decided to conduct our experiments with synthetic data.

5.1 Synthetic Data

We consider the problem of learning the ranking function of an expected utility maximizing agent. More specifically, we proceed from a standard setting of ex-

pected utility theory: $A = \{a_1, \ldots, a_c\}$ is a set of actions the agent can choose from and $\Omega = \{\omega_1, \ldots, \omega_m\}$ is a set of world states. The agent faces a problem of *decision under risk* where decision consequences are lotteries: Choosing act a_i in state ω_j yields a utility of $u_{ij} \in \mathbb{R}$, where the probability of state ω_j is p_j. Thus, the *expected utility* of act a_i is given by

$$\mathbb{E}(a_i) = \sum_{j=1}^{m} p_j \cdot u_{ij}. \tag{1}$$

Expected utility theory justifies (1) as a criterion for ranking actions and, hence, gives rise to the following preference relation:

$$a_i \succ a_j \;\Leftrightarrow\; \mathbb{E}(a_i) > \mathbb{E}(a_j). \tag{2}$$

Now, suppose the probability vector $p = (p_1, \ldots, p_m)$ to be a parameter of the decision problem (while A, Ω and the utility matrix matrix $U = (u_{ij})$ are fixed). We denote by $\succ_p$ the ranking of actions induced by the vector p according to (2).

The above decision-theoretic setting can be used for generating synthetic data for preference learning. The set of instances corresponds to the set of probability vectors p, which are generated at random according to a uniform distribution over $\{p \in \mathbb{R}^m \,|\, p \geq 0,\, p_1 + \ldots + p_m = 1\}$. The ranking function associated with an example e_k is given by the ranking $\succ_{e_k}$ as defined in (2). Thus, an experiment is characterized by the following parameters: The number of actions/labels (c), the number of world states (m), the number of examples (n), and the utility matrix which is generated at random through independent and uniformly distributed entries $u_{ij} \in [0, 1]$.

5.2 Experimental Setup

In the following, we will report on results of experiments with ten different states ($m = 10$) and various numbers of labels ($c = 5, 10, 20$). For each of the three configurations we generated ten different data sets, each one originating from a different randomly chosen utility matrix U. The data sets consisted of 1000 training and 1000 test examples. For each example, the data sets provided the probability vector $p \in \mathbb{R}^m$ and a complete ranking of the c possible actions.[2] The training examples were labeled with a subset of the complete set of pairwise preferences as imposed by the ranking in the data set. The subsets that were selected for the experiments are described one by one for the experiments.

[2]The occurrence of actions with equal expected utility has probability 0.

We used the decision tree learner C4.5 (Quinlan, 1993) in its default settings[3] to learn a model for each pairwise preference. For instances in the test set we obtained a final ranking using simple voting (and tie breaking) as described in section 3. The predicted ranks were then compared with the actual ranks on the test set, and evaluation measures were computed as follows: Denote by $(\rho_k^1, \ldots, \rho_k^c)$ the true ranking of a test example e_k, where ρ_k^1 is the top-ranked label (action). Likewise, denote by $(\tau_k^1, \ldots, \tau_k^c)$ the predicted ranking, again with τ_1^k being the label that has been assigned the top rank. Further, we use $r_k(\lambda_i)$ to denote the true rank of label λ_i for example e_k. The following four evaluation metrics were computed:

Error, the percentage of examples for which the *top rank* was incorrect:

$$\frac{1}{n}\sum_{k=1}^{n}\delta(\tau_k^1, \rho_k^1) \times 100\%,$$

where $\delta(i, j) = 1$ if $i \neq j$ and 0 if $i = j$.

Average Deviation, the average of the (average absolute) deviation of the predicted rank from the true rank:

$$\frac{1}{cn}\sum_{k=1}^{n}\sum_{r=1}^{c}|r - r_k(\tau_k^r)|$$

Maximum Deviation, the average of the maximum (absolute) deviations of the predicted rank from the true rank of each example:

$$\frac{1}{n}\sum_{k=1}^{n}\max_{r=1..c}|r - r_k(\tau_k^r)|$$

Correlation, the average Spearman rank correlation coefficient:

$$\frac{1}{n}\sum_{k=1}^{n}1 - \frac{6\sum_{r=1}^{c}(r - r_k(\tau_k^r))^2}{c(c^2-1)} \tag{3}$$

Note that this coefficient assumes values between -1 (for reversed rankings) and $+1$ (for identical rankings).

[3] Our choice of C4.5 as the learner was solely based on its versatility and wide availability. If we aimed at maximizing performance on this particular problem, we would resort to algorithms that can directly represent the separating hyperplanes for each binary preference.

Table 1: Comparison of ranking (a complete set of preferences is given) vs. classification (only the preferences for the top rank are given). Also shown are the results for the complementary setting (all preferences for the top rank are omitted).

c	prefs	error	avg dev.	max dev.	rank corr.
5	ranking	13.380 ± 8.016	0.295 ± 0.096	0.663 ± 0.201	0.907 ± 0.038
	classification	14.400 ± 8.262	0.567 ± 0.234	1.236 ± 0.537	0.783 ± 0.145
	complement	32.650 ± 14.615	0.401 ± 0.120	0.864 ± 0.248	0.872 ± 0.051
10	ranking	15.820 ± 8.506	0.594 ± 0.121	1.823 ± 0.293	0.940 ± 0.018
	classification	16.670 ± 9.549	1.559 ± 0.312	4.103 ± 0.757	0.711 ± 0.108
	complement	24.310 ± 9.995	0.617 ± 0.116	1.858 ± 0.287	0.937 ± 0.018
20	ranking	24.030 ± 4.251	1.012 ± 0.057	3.461 ± 0.204	0.966 ± 0.004
	classification	26.370 ± 5.147	3.320 ± 0.389	10.526 ± 1.125	0.697 ± 0.066
	complement	32.300 ± 3.264	1.026 ± 0.055	3.479 ± 0.191	0.966 ± 0.004

5.3 Ranking vs. Classification

Figure 1 shows experimental results for the cases where pairwise preferences are selected as follows: First, when using the full set of $c(c-1)/2$ pairwise preferences. Second, for the classification setting which uses only the $c-1$ preferences that involve the top label. Third, for the complementary setting that uses the $(c-1)(c-2)/2$ preferences that do *not* involve the top label.

There are several interesting things to note for these results. First, the difference between the error rates of the classification and the ranking setting is comparably small. Thus, if we are only interested in the top rank,[4] it may often suffice to use the pairwise preferences that involve the top label. The advantage in this case is of course the reduced complexity which becomes linear in the number of labels. On the other hand, the results also show that the complete ranking information can be used to improve classification accuracy, at least if this information is available for each training example and if one is willing to pay the price of a quadratic complexity.

The results for the complementary setting show that the information of the top rank preferences is crucial: When dropping this information and using only those pairwise preferences that do not involve the top label, the error rate on the top rank increases considerably, and is much higher than the error rate for the classification

[4] It should be noted that there is nothing special about the top rank. We expect that the same type of results can be observed if we focus on any arbitrary rank (e.g., the bottom rank or the median rank).

setting. This is a bit surprising if we consider that in the classification setting, the average number of training examples for learning a model m_{ij} is much smaller than in the complementary setting. Interestingly, the effective number of training examples for the top labels might nevertheless decrease. In fact, in our learning scenario we will often have a few *dominating* actions whose utility degrees are systematically larger than those of other actions. In the worst case, the same action is optimal for all probability vectors p, and the complementary set will not contain any information about it. While this situation is of course rather extreme, the class distribution is indeed very unbalanced in our scenario. For example, we determined experimentally for $c = m = 10$ and $n = 1000$ that the probability of having the same optimal action for more than half of the examples is $\approx 2/3$, and that the expected Gini-index of the class distribution is $\approx 1/2$.

With respect to the prediction of complete rankings, the performance for learning from the complementary set of preferences is almost as good as the performance for learning from the complete set of preferences, whereas the performance of the ranking induced from the classification setting is considerably worse. This time, however, the result is hardly surprising and can easily be explained by the amount of information provided in the two cases. In fact, the complementary set determines the ranking of $c-1$ among the c label, whereas the top label alone does hardly provide any information about the complete ranking.

As another interesting finding note that the classification accuracy decreases with an increasing number of labels, whereas the rank correlation increases (this is also revealed by the curves in Figure 2 below). In other words, the quality of the predicted rankings increases, even though the quality of the predictions for the individual ranks decreases. This effect can first of all be explained by the fact that the (classification) error is much more affected by an increase of the number of labels. As an illustration, consider random guessing: The chances of guessing the top label correctly are $1/m$, whereas the expected value of the rank correlation (3) is 0 regardless of m. Moreover, one might speculate that the importance of a correct vote of each individual learner m_{ij} decreases with an increasing number of labels. Roughly speaking, incorrect classifications of individual learners are better compensated on average. [5] This conjecture is also supported by an independent experiment in which we simulated a set of homogeneous learners m_{ij} through biased coin flipping with a prespecified error rate. It turned out that the quality measures for predicted rankings tend to increase if the number of labels becomes large (though the dependence of the measures on the number of labels is not necessarily monotone, see Fig. 1).

[5] This gives some intuitive support to the interpretation of round robin learning as an ensemble learning technique (Fürnkranz, 2003).

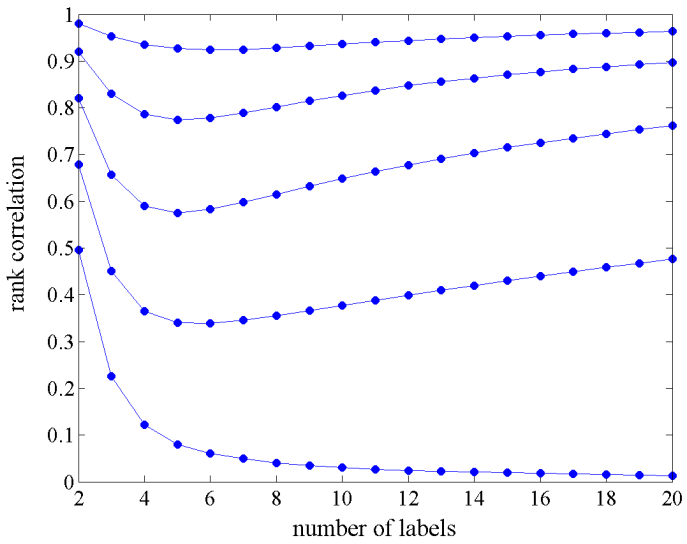

Figure 1: Expected Spearman rank correlation as a function of the number of labels if all learners m_{ij} have an error rate of ϵ (curves are shown for $\epsilon = 0.1, 0.2, 0.3, 0.4, 0.5$).

5.4 Missing Preferences

While the previous results shed some light on the trade-off between utility and costs for two special types of preference information, namely top-ranked labels and complete rankings, they do not give a satisfactory answer for the general case. The selected set of preferences in the classification setting is strongly focused on a particular label for each example, thus resulting in a very biased distribution. In the following, we will look at the quality of predicted rankings when selecting subsets of pairwise preferences from the full sets with equal right.

Figure 2 shows the curves for the classification error in the top rank and the average Spearman rank correlation of the predicted and the true ranking over the number of preferences. To generate these curves, we started with the full set of preferences, and ignored increasingly larger numbers of them. This was implemented with a parameter p_i that caused any given preference in the training data to be ignored with probability p_i ($100 \times p_i$ is plotted on the x-axis).

The similar shape of the three curves (for 5, 10, and 20 labels) suggests that the decrease in the ranking quality can be attributed solely to the missing preferences while it seems to be independent of the number of labels. In particular, one is

p

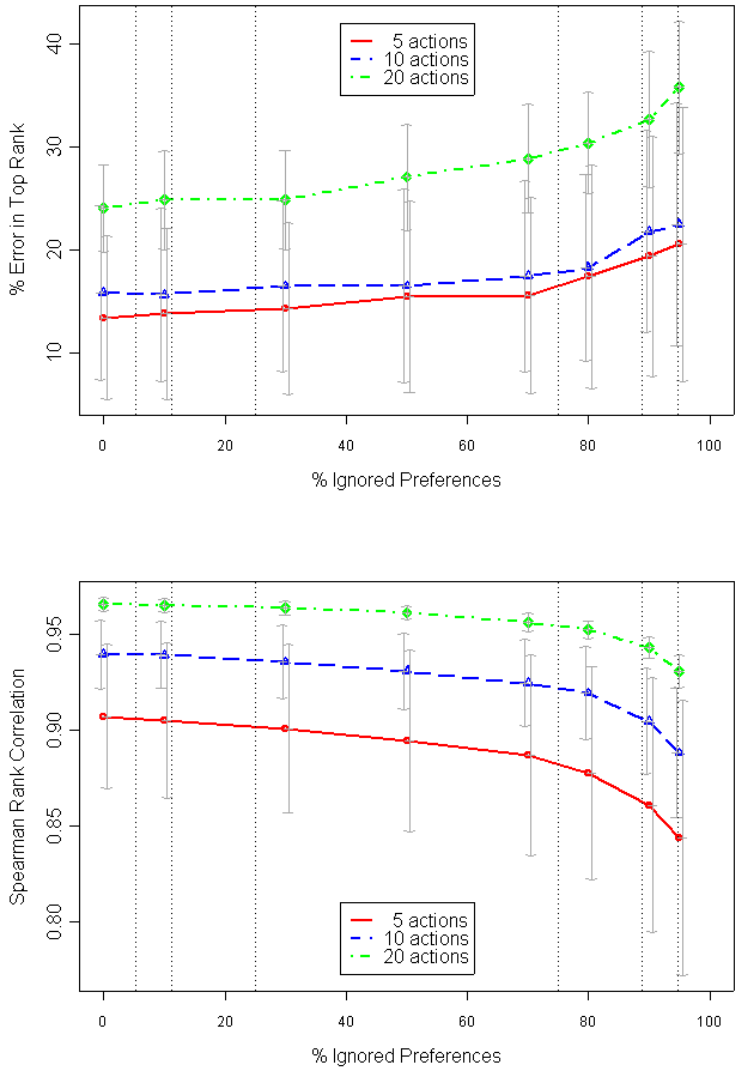

Figure 2: Average error rate (left) and Spearman rank correlation (right) for various percentages of ignored preferences. The error bars indicate the standard deviations. The vertical dotted lines on the right indicate the number of preferences for classification problems (for 5,10, and 20 classes), those on the left are the complementary sizes.

inclined to conclude that—contrary to the case where we focused on the top rank—it is in general *not* possible to reduce the number of training preferences by an order of magnitude (i.e., from quadratic to linear in the number of labels) without severely decreasing the ranking quality. This can also be seen from the three dotted vertical lines on the right. These lines indicate the percentage of preferences that were present in the classification setting for 5, 10, and 20 labels (from inner-most to outer-most). A comparison of the error rates, given by the intersection of a line with the corresponding curve, to the respective error rates in Figure 1 shows an extreme difference between the coincidental selection of pairwise preferences and the systematic selection which is focused on the top rank.

Nevertheless, one can also see that about half of the preferences can be ignored while still maintaining a reasonable performance level. Even though it is quite common that learning curves are concave functions of the size of the training set, the descent in accuracy appears to be remarkably flat in our case. One might be tempted to attribute this to the redundancy of the pairwise preferences induced by a ranking: In principle, a ranking ρ could already be reconstructed from the $c-1$ preferences $\rho_1 \succ \rho_2, \ldots, \rho_{c-1} \succ \rho_c$, which means that only a small fraction of the pairwise preferences are actually needed. Still, one should be careful with this explanation. First, we are not trying to reconstruct a single ranking but rather to solve a slightly different problem, namely to learn a ranking function. Second, our learning algorithm does actually not "reconstruct" a ranking as suggested above. In fact, our simple voting procedure does not take the dependencies between individual learners m_{ij} into account, which means that these learners do not really cooperate. On the contrary, what the voting procedure exploits is just the redundancy of preference information: The top rank is the winner only because it is preferred in $c-1$ out of the $c(c-1)/2$ pairwise comparisons.

Finally, note that the shape of the curves probably also depends on the number of training examples. We have not yet investigated this issue because we were mainly interested in the possibility of reducing the complexity by more than a constant factor without losing too much of predictive accuracy. It would be interesting, for example, to compare (a) using $p\%$ of the training examples with full preferences and (b) using all training examples with $p\%$ of the pairwise preferences.

5.5 Mislabeled Preferences

Recall that our learning scenario assumes preference structures to be complete rankings of labels, that is transitive and asymmetric relations. As already pointed out, we do not make this assumption for *observed* preferences: First, we may not have access to complete sets of preferences (the case studied in the previous section). Second, the process generating the preferences might reproduce the underly-

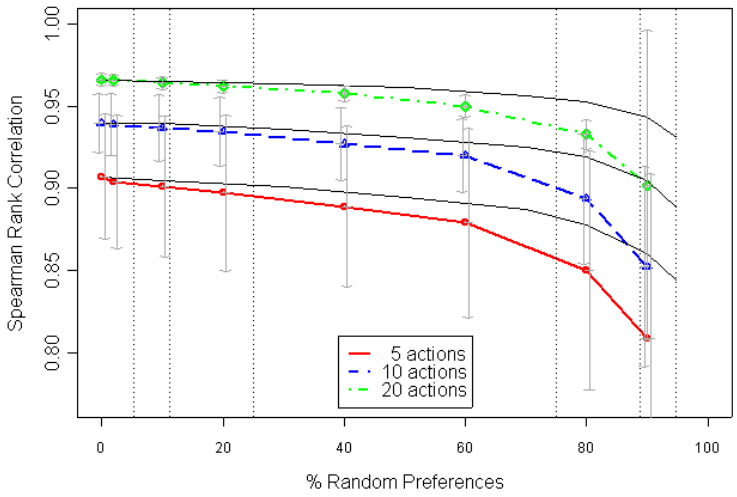

Figure 3: Average Spearman rank correlation over various percentages of random preferences. The error bars indicate the standard deviations. The solid thin lines are the curves for ignored preferences (Figure 2).

ing total order incorrectly and, hence, produce inconsistent preferences. The latter problem is quite common, for example, in the case of human judgments.

To simulate this behavior, we adopted the following model: Proceeding from the pairwise preferences induced by a given ranking, a preference $\lambda_i \succ \lambda_j$ was kept with probability $1 - p_s$, whereas with probability p_s, one of the preferences $\lambda_i \succ \lambda_j$ and $\lambda_j \succ \lambda_i$ was selected by a coin flip. Thus, in approximately $p_s/2$ cases, the preference will point into the wrong direction.[6] For $p_s = 0$, the data remain unchanged, whereas the preferences in the training data are completely random for $p_s = 1$.

Figure 3 shows the average Spearman rank correlations that were observed in this experiment. Note that the shape of the curve is almost the same as the shape of the curves for ignored preferences. It is possible to directly compare these two curves because in both graphs a level of $n\%$ means that $100 - n\%$ of the preferences are still intact. The main difference is that in Figure 2, the remaining $n\%$ of the preferences have been ignored, while in Figure 3 they have been re-

[6] In fact, we implemented the procedure by selecting $p_s/2$ preferences and reversing their sign.

assigned at random. To facilitate this comparison, we plotted the curves for ignored preferences (the same ones as in Figure 2) into the graph (with solid, thin lines).

It is interesting to see that in both cases the performance degrades very slowly at the beginning, albeit somewhat steeper than if the examples are completely ignored. Roughly speaking, completely omitting a pairwise preference appears to be better than including a random preference. This could reasonably be explained by the learning behavior of a classifier m_{ij}: If m_{ij} does already perform well, an additional correct example will probably be classified correctly and thus improve m_{ij} only slightly (in decision tree induction, for example, m_{ij} will even remain completely unchanged if the new example is classified correctly). As opposed to this, an incorrect example will probably be classified incorrectly and thus produce a more far-reaching modification of m_{ij} (in decision tree induction, an erroneous example might produce a completely different tree). All in all, the "expected benefit" of m_{ij} caused by a random preference is negative, whereas it is 0 if the preference is simply ignored.

From this consideration one may conclude that a pairwise preference should better be ignored if it is no more confident than a coin flip. This can also be grasped intuitively, since the preference does not provide any information in this case. If it is more confident, however, it clearly carries some information and it might then be better to include it, even though the best way of action will still depend on the number and reliability of the preferences already available. Note that our experiments do not suggest any strategy for deciding whether or not to include an *individual* preference, given information about the uncertainty of that preference. In our case, each preference is equally uncertain. Thus, the only reasonable strategies are to include all of them or to ignore the complete sample. Of course, the first strategy will be better as soon as the probability of correctness exceeds $1/2$, and this is also confirmed by the experimental results. For example, the correlation coefficient remains visibly above 0.8 even if 80% of the preferences are assigned by chance and, hence, the probability of a particular preference to be correct is only 0.6. One may conjecture that pairwise preference ranking is particularly robust toward noise, since an erroneous example affects only a single classifier m_{ij} which in turn has a limited influence on the eventually predicted ranking.

6 Related Work

As pointed out before, especially relevant for our work is the framework of *constraint classification*, introduced as an extension of standard classification by Har-Peled et al. (2002). The learning method proposed in this work constructs two training examples for each preference $\lambda_i \succ \lambda_j$, where the original d-dimensional

training examples are mapped into a cd-dimensional space. The positive example copies the original training vector into the components $d(i-1)+1 \ldots di$ and its negation into the components $d(j-1)+1 \ldots dj$ of a vector in the new space. The remaining entries are filled with 0, and the negative example has the same elements with reversed signs. In this cd-dimensional space, the learner tries to find a separating hyperplane. For classifying a new example e, the labels are ordered according to the response resulting from multiplying e with the i-th d-element section of the hyperplane. This technique also compares favorably to a one-against-all approach.

There has also been some recent work on ranking algorithms. For example, Crammer and Singer (2003) consider a variety of on-line learning algorithms for the problem of ranking possible labels in a multi-label text categorization task. However, we are only aware of one work that actually uses a *complete* ranking of the available labels for each example for training or evaluation: Brazdil et al. (2003) investigate the meta-learning task of ranking learning algorithms according to their suitability for a new dataset, based on the characteristics of this dataset.

Some authors have investigated the problem of preference elicitation in a more narrow sense, that is, the learning of one single preference function. For example, Cohen et al. (1999) propose a two-step approach for ranking a set of objects (and not a set of labels associated with the objects as in our approach) given feedback in the form of preference judgments. Similarly, Haddawy et al. (2003) assume training data to be available in the form of pairwise comparisons of objects. Given such data, they train an artificial neural network that takes as input two objects and outputs either 0 or 1, depending on whether or not the first object is preferred to the second one. (A somewhat similar approach has already been proposed by Wang (1994)). Joachims (2002) analyzes "click-through data" in order to rank documents retrieved by a search engine according to their relevance. This is a nice example of a kind of *indirect* preference information. Using this information, learning of a retrieval function is accomplished by training a support vector machine.

The problem of learning a preference relation over a set of labels L can also be approached in a somewhat indirect way, namely through learning a value or utility function that assigns a utility degree to each label. Note that the preference relation induced by a utility function is necessarily *complete* (linear) in the sense that all tuples of labels are assumed to be comparable. Moreover, note that learning a utility function can be considered a more difficult problem than learning a (linear) preference relation, since the latter subsumes the former but not vice versa.

Depending on the underlying utility scale one can distinguish between learning a numeric function and learning a function that maps into an ordinal (ordered categorical) scale. These two cases involve, respectively, a problem of standard regression and ordinal regression (also called ordinal classification). Ordinal regression has been investigated thoroughly in statistics and econometrics (McCullagh and

Nelder, 1983) and has recently also received attention in machine learning. For example, a method for ordinal regression based on a modification of regression tree learning has been proposed by Kramer et al. (2001). Frank and Hall (2001) suggest a method for translating an ordinal regression problem into a set of ordinary (binary) classification problems. In (Herbich et al., 2000), ordinal regression is approached in the context of support vector machines, using a special type of loss function suitable for comparing predictions on an ordered categorical scale.

The problem of learning (eliciting) *real-valued* utility functions has been investigated in fields such as decision theory and economics for a long time, and has more recently become a topic of research in AI and machine learning as well. A particularly elegant approach is due to Tesauro (1989), who proposes a symmetric network architecture that can be trained with representations of two states and a training signal that indicates which of the two states is preferable. The elegance of this *comparison training* approach comes from the property that one can replace the two symmetric parts of the network with a single network, which can subsequently provide a real-valued evaluation of single states. More recently, Chajewska et al. (1998) simplify the elicitation of utility functions by clustering exemplary utility functions, deriving prototypes from the clusters, and inducing a decision tree whose inner nodes are associated with properties of utility functions (questions that can be asked to a person) and whose leaf nodes are identified with the prototypes. The idea of Chajewska et al. (1999) is to simplify elicitation by exploiting the *additive independence* of variables. Given a database of exemplary utility functions, statistical learning (model selection) methods are used in order to induce a factorization of utility functions into additive subutility functions. Chajewska et al. (2000) accomplish learning of a utility function by treating utility as a random variable. Starting with some prior distribution (derived from analyzing a database of available utility functions), the model is incrementally updated based on information elicited from the user. In order to decide on which questions should be asked next to the user, the authors fall back on the principle underlying the *value of information*. Chajewska et al. (2001) study the problem of learning the utility function that determines the behavior of an agent which is rational in the sense of expected utility theory. The approach proposed by the authors proceeds from a prior probability distribution over a class of utility functions having a certain (linear) structure. The agent's decisions are then used for defining constraints on its true utility function (see (Ng and Russell, 2000) for a quite similar approach). Finally, these constraints are employed in order to turn the prior distribution over the class of utility functions into a posterior distribution.

Learning preferences is also a key topic in recommender systems and collaborative filtering (Goldberg et al., 1992; Resnick and Varian, 1997; Kautz, 1998). Methods proposed in this field are closer to learning utility functions, but are of-

ten specifically adjusted to commercial applications where the set of alternatives (labels) to be recommended is usually very large. The method of choice is quite often a case-based or memory-based approach, where the basic idea is to estimate a user's preferences from the preferences of other users that appear to be *similar* (see e.g. Ha and Haddawy (2003); Nakamura and Abe (1998); Billsus and Pazzani (1998)).

7 Concluding Remarks

We have introduced pairwise preference learning as an extension of pairwise classification to constraint classification, a learning scenario where training examples are labeled with a preference relation over all possible labels instead of a single class label as in the conventional classification setting. From this information, we also learn one model for each pair of classes, but focus on learning a complete ranking of all labels instead of only predicting the most likely label. Our main interest was to investigate the trade-off between ranking quality and the amount of training information (in terms of the number of preferences that are available for each example). We experimentally investigated this trade-off by varying parameters of a synthetic domain that simulates a decision-theoretic agent which ranks its possible actions according to an unknown utility function. Roughly speaking, the results show that large parts of the information about pairwise preferences can be ignored in round robin ranking without losing too much predictive performance. In the classification setting, where one is only interested in predicting the top label, it also turned out that using the full ranking information rather than restricting to the pairwise preferences involving the top label does even improve the classification accuracy, suggesting that the lower ranks do contain valuable information. For reasons of efficiency, however, it might still be advisable to concentrate on the smaller set of preferences, thereby reducing the size of the training set raises by an order of magnitude.

The main limitation of our technique is probably the assumption of having enough training examples for learning each pairwise preference. For data with a very large number of labels and a rather small set of preferences per example, our technique will hardly be applicable. In particular, it is unlikely to be successful in collaborative filtering problems (Goldberg et al., 1992; Resnick and Varian, 1997; Breese et al., 1998), although these can be mapped onto the constraint classification framework in a straightforward way. A further limitation is the quadratic number of theories that has to be stored in memory and evaluated at classification time. However, the increase in memory requirements is balanced by an increase in computational efficiency in comparison to the technique of Har-Peled et al. (2002). In

addition, pairwise preference learning inherits many advantages of pairwise classification, in particular its implementation can easily be parallelized because of its reduction to independent subproblems.

There are several directions for future work. First of all, it is likely that the prediction of rankings can be improved by combining the individual learners' votes in a more sophisticated way. Several authors have looked at more sophisticated ways for combining the predictions of pairwise theories into a final ranking of the available options. Proposals include weighting the predicted preferences with the classifiers' confidences (Fürnkranz, 2003) or using an iterative algorithm for combining pairwise probability estimates (Hastie and Tibshirani, 1998). However, none of the previous works have evaluated their techniques in a ranking context, and some more elaborate proposals, like error-correcting output decoding (Allwein et al., 2000), organizing the pairwise classifiers in a tree-like structure (Platt et al., 2000), or using a stacked classifier (Savicky and Fürnkranz, 2003) are specifically tailored to a classification setting. Taking into account the fact that we are explicitly seeking a ranking could lead to promising alternatives. For example, we are thinking about selecting the ranking which minimizes the number of predicted preferences that need to be reversed in order to make the predicted relation transitive. Departing from the counting of votes might also offer possibilities for extending our method to the prediction of preference structures more general than rankings (total orders), such as weak preference relations where some of the labels might not be comparable.

Apart from theoretical considerations, an important aspect of future work concerns the practical application of our method and its evaluation using real-world problems. Unfortunately, real-world data sets that fit our framework seem to be quite rare. In fact, currently we are not aware of any data set of significant size that provides instances in attribute-value representation plus an associated complete ranking over a limited number of labels.

Acknowledgments

Johannes Fürnkranz is supported by an APART stipend (no. 10814) of the *Austrian Academy of Sciences*. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture.

References

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

D. Billsus and M. Pazzani. Learning collaborative information filters. In *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, pages 46–54. Morgan Kaufmann, 1998.

P. B. Brazdil, C. Soares, and J. P. da Costa. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3): 251–277, March 2003.

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, Madison, WI, 1998. Morgan Kaufmann.

U. Chajewska, L. Getoor, J. Norman, and Y. Shahar. Utility elicitation as a classification problem. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in AI (UAI-98)*, pages 79–88, 1998.

U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, pages 35–42, 2001.

U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 363–369, 2000.

U. Chajewska, M. Kuppermann, and D. Koller. Discovering the structure of utility functions based on additive and conditionally additive independence properties between utility attributes. In *Proceedings of the 21st Annual Meeting of the Society for Medical Decision Making (MDM-99)*, 1999.

W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

J. Couch. Disease management: An overview. In J. Couch, editor, *The Health Care Professional's Guide to Disease Management: Patient-Centered Care for the 21st Century*. Aspen Publishers, 1998.

K. Crammer and Y. Singer. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, 3:1025–1058, 2003.

E. Frank and M. Hall. A simple approach to ordinal classification. In L. D. Raedt and P. Flach, editors, *Proceedings of the 12th European Conference on Machine Learning (ECML-01)*, pages 145–156, Freiburg, Germany, 2001. Springer-Verlag.

J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

J. Fürnkranz. Round robin ensembles. *Intelligent Data Analysis*, 7(5), 2003. to appear.

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

V. Ha and P. Haddawy. Similarity of personal preferences: theoretical foundations and empirical analysis. *Artificial Intelligence*, 2003. To appear.

P. Haddawy, V. Ha, A. Restificar, B. Geisler, and J. Miyamoto. Preference elicitation via theory refinement. *Journal of Machine Learning Research*, 2003. To appear.

S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification. In N. Cesa-Bianchi, M. Numao, and R. Reischuk, editors, *Proceedings of the 13th International Conference on Algorithmic Learning Theory (ALT-02)*, pages 365–379, Lübeck, Germany, 2002. Springer.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10 (NIPS-97)*, pages 507–513. MIT Press, 1998.

R. Herbich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In A. J. Smola, P. J. Bartless, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring goals and needs of software users. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in AI (UAI-98)*, pages 256–265, 1998.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

T. Joachims. Optimizing search engines using clickthrough data. In *KDD–2002, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 2002.

H. Kautz, editor. *Recommender Systems: Papers from the AAAI Workshop*, Menlo Park, CA, 1998. AAAI Press. Technical Report WS-98-08.

S. Kramer, G. Widmer, B. Pfahringer, and M. DeGroeve. Prediction of ordinal classes using regression trees. *Fundamenta Informaticae*, XXI:1001–1013, 2001.

P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman & Hall, London, 1983.

A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, pages 395–403. Morgan Kaufmann, 1998.

A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*, 2000.

J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12 (NIPS-99)*, pages 547–553. MIT Press, 2000.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

P. Resnick and H. R. Varian. Special issue on recommender systems. *Communications of the ACM*, 40(3), 1997.

D. Riecken. Perzonalized views of personalization. *Communications of the ACM*, 43(8):26–29, 2000.

P. Savicky and J. Fürnkranz. Combining pairwise classifiers with stacking. Submitted for publication, 2003.

G. Tesauro. Connectionist learning of expert preferences by comparison training. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1 (NIPS-88)*, pages 99–106. Morgan Kaufmann, 1989.

J. Wang. Artificial neural networks versus natural neural networks: a connectionist paradigm for preference assessment. *Decision Support Systems*, 11:415–429, 1994.