Clustering of preference criteria[1]

J. Díez, J.J. del Coz, O. Luaces and A. Bahamonde
Centro de Inteligencia Artificial. Universidad de Oviedo at Gijón,
Campus de Viesques,
E-33271 Gijón (Asturias), Spain
{jdiez, juanjo, oluaces, antonio}@aic.uniovi.es

Abstract. Learning preferences is a useful task in application fields such as collaborative filtering, information retrieval, adaptive assistants or analysis of sensory data provided by panels. From training sets of preference judgments, using SVM, it is possible to induce ranking functions that map vectors representing objects into real numbers. In this paper we present a new algorithm to build clusters of people with closely related tastes, and hence people whose preference judgment sets can be merged in order to learn more reliable ranking functions. The key insight is that these functions can be used to guide the clustering, since they codify the rationale for the preferences gathered in training sets. Experimental results using the EachMovie database illustrate the satisfactory performance of our approach.

Keywords: learning preferences, clustering, adaptive assistants, analysis of sensory data, information retrieval.

1 Introduction

Supervised inductive learning deals with sets of training examples; these represent pairs of input and the attached outputs of a function that has to be found in a given family of hypotheses. The input is described by a set of attribute values, while the output is in fact another attribute of the examples called class; its type determines the approach and even the name of the learning task. Regression is used when the class is a continuous number, and categorical classification is employed when the class or output of training examples is one of a finite set of symbolic categories.

In this paper we tackle a slightly different problem: learning people's preferences for consumable products, or for system configurations, or for responding to information requests. Here the training material can be expressed as in regression problems: the description of each object is then followed by a number that assesses the degree of satisfaction. Alternatively, training examples can be represented by *preference judgments*: pairs of vectors (**v**, **u**) where someone expresses the fact that he or she

[1] The research reported in this paper has been supported in part under MCyT and Feder grant TIC2001-3579.

prefers **v** to **u**. In other words, training sets are samples of binary relations between objects described by the components of vectors of real numbers.

As pointed out in (Cohen et al., 1999; Dumais et al., 2003), obtaining preference information may be easier and more natural than obtaining the labels needed for a classification or regression approach. Moreover, this type of information is more accurate, since people tend to rate their preferences in a relative way, comparing objects with the other partners in the same batch. There is a kind of *batch effect* that often biases the ratings. Thus, an object presented in a batch surrounded by worse objects will probably obtain a higher rating than if it were presented together with better objects.

There are a number of algorithms in the literature able to learn preferences. Sometimes they aim to classify pairs of objects (**v**, **u**), deciding whether **v** is preferable to **u** or not, as in (Branting and Broos, 1997; Cohen et al., 1999). Another approach consists in learning a real *preference* or *ranking function* **f** from the space of objects considered in such a way that **f(v) > f(u)** whenever **v** is preferable to **u**. This functional approach can start from a set of objects endowed with a (usually ordinal) rating, as in regression (Herbrich et al., 1999; Crammer and Singer, 2001; Shashua and Levin, 2002), or can stem from sets of preference judgments, as in (Tesauro, 1989; Utgoff and Clouse, 1991; Freund et al., 1998; Fiechter and Rogers, 2000; Joachims, 2002; Díez et al., 2002).

In this paper we present a new algorithm for clustering preference criteria. The next section is devoted to explaining the usefulness of clusters of preference criteria in different areas of application. The general idea is that given a family of preference judgment sets of a number of people, the algorithm discovers groups with homogeneous tastes. We can then merge the sets of preference judgments of members of the same group, thus attaining more useful and reliable knowledge about peoples' preferences. The novelty of our proposal is based on the assumption that the ranking function learned from each preference judgment set codifies the criteria used to make these preferences. Therefore we will try to merge data sets with similar ranking functions. We learn the new ranking function from the merged data set, aggregating the data sets if the estimated accuracy is higher, to then adopt the new ranking function as the criteria of the group thus constituted. The algorithm stops when no more merges can be achieved.

In the rest of the paper, following a detailed discussion of our approach, we conclude with a report of the experiments conducted to evaluate the clustering algorithm. For this purpose we use EachMovie (McJones, 1997), a publicly available collaborative filtering database for movie ratings.

2 How clusters can be useful

The learning tasks involved in *recommender* systems (Resnick and Varian, 1997) can be considered as special cases of ordinal regression. Here users rate one kind of object and receive recommendations about objects that they are likely to prefer. Such advice can be elaborated according to the relationship of the properties of the objects and the user's past ratings; this is the *content-based* model (Basu et al., 1998; Pazzani, 1999). Or, on the other hand, in the model called *collaborative* or *social filtering*, the recommendations are induced from the user and other users' ratings, formulating them as a learning task (Goldberg et al., 1992; Resnick et al., 1994; Shardanand and Maes, 1995).

Within this context, as pointed out in (Hofmann and Puzicha, 1999), there is another fundamental problem in addition to the prediction of ratings: the discovery of meaningful groups or *clusters* of persons and objects able to explain the observed preferences by some smaller number of typical preference patterns. This point of view gives rise to the *latent* class model (Cheung et al., 2000). See also (Ungar and Foster, 1998).

There are other application fields where clusters and preferences appear together as a desirable mixture. For instance, in (Joachims, 2002), Joachims presents an information retrieval system equipped with a ranking function learned from click-through data collected from user interaction with a www search engine. To improve his proposal, the author acknowledges the need to obtain feasible training data. This raises the question of the convenience of developing clustering algorithms to find homogeneous groups of users. An Adaptive Route Advisor is described in (Fiechter and Rogers, 2000); the system is able to recommend a route to lead users through a digitalized road map taking into account their preferences. An interesting extension discussed in the paper is to modify route recommendations depending on the time of the day or the purpose of the trip. The approach suggested includes an algorithm that clusters user preferences into contexts.

In addition to these problems involving clusters and preferences, the field of application that motivates the research reported in this paper is the analysis of sensory data used to test the quality (or study the acceptability) of market products that are principally appreciated through sensory impressions. An excellent survey of the use of this type of data in the food industry can be found in (Murray et al., 2001; Buck et al., 2001); for a Machine Learning perspective, see (Corney, 2002) and (Goyache et al., 2001a, 2001b; Díez et al., 2002, 2003).

From a conceptual point of view, sensory data include the assessment of products provided by two different kinds of panels. The first one is made up of a small group of expert, trained judges; these will describe each product by attribute-value pairs. Expert panelists are thus required to have enough sensory accuracy so as to discriminate between different and similar products; note that experts are not

necessarily asked to rate the overall quality of products. This panel will play the role of a bundle of sophisticated sensors, probably acting in addition to some chemical or physical devices. To achieve this performance, 2-3 times as many panelists as those required are screened through a selection or casting process that may take several months.

The second kind of panel is made up of untrained consumers; these are asked to rate their degree of acceptance of the tested products on a scale. The aim is to be able to relate sensory descriptions (human and mechanical) with consumer preferences in order to improve production decisions.

Market studies will start out from tables such as Table 1. Each row represents a product rated by a consumer in a given session; this is important, since we can interpret consumer ratings relative to each session (Joachims, 2002; Díez et al., 2002). In this way, we do not need to assume that a rating of "7" means the same thing to every consumer and in every session (Cohen et al., 1999).

Table 1. Sensory data collected from panels of experts and consumers. Each product is described by expert assessments (Att_j) in addition to other ($O\text{-}Att_i$) chemical or physical analysis outputs

Expert sensory appreciations							Other descriptive attributes			Consumer preferences		
Expert-1				Expert-k								
Att_1	…	Att_m	…	Att_1	…	Att_m	$O\text{-}Att_1$	…	$O\text{-}Att_n$	Session	Consumer	Rating
<num>	…	<num>	…	<num>	…	<num>	<num>	…	<num>	<i>	<Id>	<num>
	…		…		…					…	…	…
<num>	…	<num>	…	<num>	…	<num>	<num>	…	<num>	<i>	<Id>	<num>

On the other hand, expert descriptions are ratings in an ordinal scale of different aspects of products related to their taste, odor, color, etc.. Here we must assume that a rating of "7" (in say, texture) means the same for a given expert in every product; though not necessarily for every expert. In other words, the most essential property of expert panelists, in addition to their discriminatory capacity, is their own coherence, not the uniformity of the group. Therefore, the selection of expert panelists must check this capacity of candidates throughout a number of experiments.

Our proposal for the selection of expert panelists explicitly uses an algorithm to cluster preference criteria. Thus for each candidate we propose one experiment for each product attribute (Att_j) that will be used in Table 1. For any collection of products, the data set used for these experiments will include the candidate assessments and an extensive description of products obtained from chemical or physical analysis outputs. Let us point out that the amount of mechanical data in this stage is greater and more expensive than that used in the interaction with consumers (see Table 1). In fact, this kind of data will frequently not be needed at all, if experts behave like well-calibrated, feasible sensors.

Within this framework of experiments, candidate adequacy may be estimated by the accuracy of the ranking function that can be learned from experimental data sets; and the reliability of the estimation will rely on the number of preference judgments used for training. Therefore, should we be able to merge data sets of some candidates with

similar assessment criteria without decreasing the accuracy, then the credibility of the results will increase, and the selection of experts for the panel will be more objective and reliable. And this is exactly what clustering can do.

3 How to cluster preferences

Let us start this section by presenting a framework that will be used to introduce our clustering algorithm. Thus, E will be a set of vectors describing a collection of objects that will be ranked by N people. Given a space of ordinal values, Scale, we have a family

$$r_i: E \rightarrow \text{Scale}, i = 1, \ldots, N \quad (1)$$

of ranking functions, one for each person, that in general are partially defined. From another point of view, the pairs (E, r_i) can be seen as a collection of N data sets for ordinal regression. In the following subsections, we will show that this framework can be slightly more general if we assume that (E, r_i) may be sets of preference judgments.

To illustrate our proposals, we used EachMovie. This database contains ratings of 1,628 movies provided by 72,916 people. The scale used has 6 values {0, 0.2, 0.4, 0.6, 0.8, 1}, but less than only 2.4% of the possible values are filled. Thus, for instance, 11,651 people have not given any rating to any movie.

As many authors do in order to avoid the extreme sparsity of data, we considered a subset of EachMovie. We only considered movies with at least 1,000 ratings; there are 504 movies in such conditions. We likewise only took into consideration people who had submitted at least 200 ratings for these movies; this makes a total of 908 people.

To accomplish our setting for clustering, we considered the 100 people with more ratings as our sample to study possible similarity of preference criteria; i.e. we set $N = 100$. Therefore, the remaining available ratings were considered as the set E of descriptions of the objects considered in our experiment. In other words, we represented each movie by a vector of 808 (= 908 – 100) ratings.

3.1 Clustering ratings or ranking functions

To measure the similarity between the preferences of two people i and j, a first undertaking is to compare the vectors $(r_i(\mathbf{x}): \mathbf{x} \in E)$ and $(r_j(\mathbf{x}): \mathbf{x} \in E)$ of their ratings. To do this, we must realize that we will normally have a substantial number of missing values in both rating vectors.

Different tools have been employed to make comparisons when using these preference vectors for prediction tasks in collaborative filtering (see (Breese et al., 1998)); the most obvious and unsuccessful being Euclidean distance, used in nearest

neighbor algorithms. Pearson's correlation or the cosine of the vectors has been put forward to take into account the possible differences in the scales used by different people.

However, these comparison techniques devised for prediction purposes are not easily extendable to clustering. To illustrate this point, let us consider one person p with a coherent preference criterion, and let us divide p's rating vector in two parts:

$$(r_p(\mathbf{x})\colon \mathbf{x} \in E_1),\ (r_p(\mathbf{x})\colon \mathbf{x} \in E_2), \text{ with } E_1 \cap E_2 = \emptyset. \qquad (\mathbf{2})$$

These two vectors would not have anything in common for any reasonable comparison measure. However, both vectors represent the same rating criterion, and p rating E_1 and p rating E_2 must be included in the same cluster; in fact p *is* the same person.

Therefore, the proposal presented in this paper is to represent preference criteria explicitly by means of linear ranking functions, and then use the similarity of these functions as a heuristic to aggregate individuals into a cluster. In the next subsection we will review how to compute these ranking linear functions.

3.2 Linear separation and preferences

Given a set of descriptions E and a rating vector $(r(\mathbf{x})\colon \mathbf{x} \in E)$, we can try to use regression to induce a function that maps object descriptions into ratings. However, this is not a faithful way of capturing people's preferences; see (Freund et al., 1998; Herbrich et al., 1999; Fiechter and Rogers, 2000; Shashua and Levin, 2002; Joachims, 2002; Díez et al., 2002). The main reason is that ratings are relative orderings instead of absolute values. Thus, linear regression is not a good candidate for finding a representation of preference criteria. Instead, we will use ranking functions.

Following (Herbrich et al., 1999; Joachims, 2002; Díez et al., 2002), our approach is to reduce the induction of a ranking function to a problem of linear separation of examples into two discrete classes. Additionally, in order to capture the relative character of people's preferences, we are not going to consider rating vectors any more. So, let us assume that

$$PJ = \{\mathbf{v}_j > \mathbf{u}_j\colon j = 1, \ldots, m\} \qquad (\mathbf{3})$$

is a training set of preference judgments of vectors of real numbers in $E \subset \Re^d$ that represents a given kind of object.

If we have rating vectors, like in the EachMovie case, a preference judgment set can be obtained from a rating vector $(r_i(\mathbf{x})\colon \mathbf{x} \in E)$ of a spectator i considering for each $\mathbf{x} \in E$ a number of t (usually t will be 10) vectors with different ratings. Thus, given two movies $\mathbf{x}$ and $\mathbf{y}$ ranked by spectator i, we include $(\mathbf{x}, \mathbf{y})$ in PJ_i if and only if $r_i(\mathbf{x}) > r_i(\mathbf{y})$. However, this is not the usual situation; in the kind of applications described in section 2 above, we will have sets of preference judgments in a natural way; notice

that, in this sense, we are using EachMovie data in order to simulate a typical preferences situation extracted from a publicly available dataset.

In any case, from a dataset like (3), we are looking for an ordering-preserving (monotone) function $\mathbf{f}:\Re^d \to \Re$ that will be called *preference* or *ranking function*, such that it maximizes the probability of having $\mathbf{f(v) > f(u)}$ whenever $\mathbf{v > u}$. We will assume that $\mathbf{f}$ must be a linear function; then our function will have the form $\mathbf{f_w(x) = w \cdot x}$ for a given $\mathbf{w}$. From a geometrical point of view, the output of this map is proportional to the distance to the hyperplane of vectors perpendicular to $\mathbf{w}$; see Figure 1. Thus, given $\mathbf{v > u}$, we need $\mathbf{w}$ and $(\mathbf{v-u})$ to be vectors with a positive cosine, i.e. with a positive scalar product; equivalently $\mathbf{w\,(v - u) > 0}$. In symbols,

$$\mathbf{f_w(v) > f_w(u) \Leftrightarrow 0 < f_w(v) - f_w(u) = f_w(v - u) = w \cdot (v - u)} \qquad (4)$$

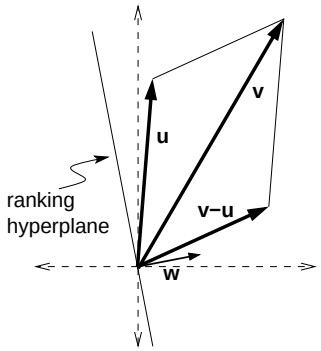

Fig. 1. The difference vector $\mathbf{v-u}$ is on the positive side of the hyperplane with a normal vector $\mathbf{w}$. Therefore, $\mathbf{f_w(x) = w \cdot x}$ coherently ranks the preference judgment $\mathbf{v > u}$

Thus we are searching for a hyperplane able to separate increasing or positive differences (like $\mathbf{v - u}$ when $\mathbf{v > u}$) from decreasing or negative differences (like $\mathbf{u - v}$). We will employ an SVM classifier (Vapnik, 1998) to find this $\mathbf{w}$; the implementation used is Joachims' SVM^{light} (Joachims, 1998).

3.3 The clustering algorithm

Let $((PJ_i, \mathbf{w}_i): i = 1,\ldots, N)$ be a collection of N preference judgment sets PJ_i endowed with the director vector $\mathbf{w}_i$ of their respective ranking functions. Taking into account the applications of clusters reviewed in Section 2, the goal is to be able to merge data sets in order to induce more reliable ranking functions. This yields the following

Definition.- Two people have *similar preference criteria* if and only if the estimated accuracy of their ranking functions is lower than the estimated accuracy of the ranking function induced from the union of their respective preference judgments.

If two director vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are similar, we can expect $PJ_1 \cup PJ_2$ to be learned with a similar accuracy to that achieved by $\mathbf{w}_1$ and $\mathbf{w}_2$ separately. In fact, the ranking function of the union will have a director vector similar to $\mathbf{w}_1$ and $\mathbf{w}_2$. To compute the similarity of this kind of vector, we will use their cosine, defining

$$\text{similarity}(\mathbf{w}_1, \mathbf{w}_2) = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\|\mathbf{w}_1\| \cdot \|\mathbf{w}_2\|} \quad (5)$$

However, this measure is solely a heuristic to suggest pairs with similar preference criteria. The reason for this is that it is not necessary for $\mathbf{w}_1$ and $\mathbf{w}_2$ to be similar vectors in order to guarantee that two people (1 and 2) have similar *preference criteria*. Due to possible overfitting of the learning algorithm used, and the typical sparsity of data, sometimes rather different director vectors may codify similar preference criteria; a schematic example of this situation can be seen in Figure 2.

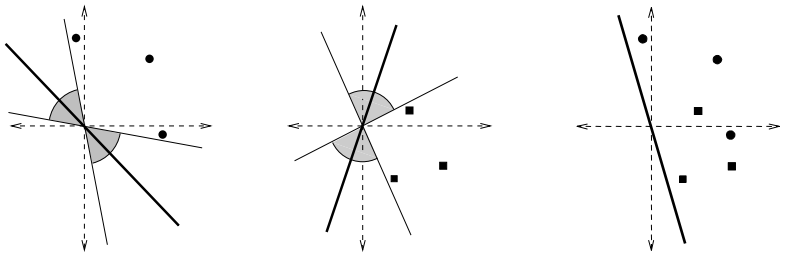

Fig. 2. In the leftmost picture we have three positive differences represented by circles; there are a wide range of possible hyperplanes that can situate themselves on the positive side. In the central picture, in a similar case, there are another three positive differences represented by squares. Apparently both situations reflect distant preference criteria; however, it is possible to find a common criterion for circles and squares, namely the hyperplane of the rightmost picture

The algorithm that we propose to compute clusters of preference criteria is detailed in Table 2 with the title *clustering algorithm*. It is a straightforward clustering algorithm; it starts by considering each individual as a cluster and iteratively tries to merge the clusters with the most similar director vector of their respective ranking functions. The condition to merge clusters (PJ_1, $\mathbf{w}_1$) and (PJ_2, $\mathbf{w}_2$) with similar $\mathbf{w}_1$ and $\mathbf{w}_2$ is the increase of the estimation of accuracy.

Thus, if $\mathbf{w}$ is the director vector induced from the union $PJ_1 \cup PJ_2$, we aggregate the clusters whenever the estimated number of classification errors $\mathbf{w}$ is lower than the sum of estimated errors of $\mathbf{w}_1$ plus those of $\mathbf{w}_2$. The algorithm stops when no more merges can be achieved. If the available training data allow us to separate a significant part as *verification* data, then to estimate the number of errors we can use the family of those sets, (V_i: $i = 1,\ldots, N$). In this case, we first compute the confidence interval of the probability of error when we apply each ranking function to the corresponding verification set; we use the confident level $\alpha = 0.05$. Let $[L_1,R_1]$ $[L_2,R_2]$

be those intervals for $\mathbf{w}_1$ and $\mathbf{w}_2$ separately. At the same time, we compute the same interval for $PJ_1 \cup PJ_2$: [L,R]. Finally, the clusters are merged if

$$R \cdot |V_1 \cup V_2| \leq R_1 \cdot |V_1| + R_2 \cdot |V_2| \quad (6)$$

However, according to the availability of data, it is possible that we can not afford to have a family of separate verification dataset. In those cases, we should use any available tool to estimate the generalization error using only the training set. In some cases, the Xi-alpha estimator (Joachims, 2000) is a good candidate for this job.

Table 2. ***Clustering Algorithm***. Starting from a list of sets representing the preference judgments of N people, this algorithm outputs a list of clusters of people with similar preference criteria. Additionally, each cluster is endowed with its learned ranking function

```
A list of clusters ClusterPreferencesCriteria
  (a list of sets of preference judgments (PJi: i = 1,…, N)) {
    Clusters = ∅;
    for each i = 1 to N {
        wi = Learn a ranking hyperplane from (PJi);
        Clusters = Clusters U {(PJi, wi)};
    }
    repeat {
        let (PJ1, w1) and (PJ2, w2) be the clusters with most similar w1 and w2;
        w = Learn a ranking hyperplane from (PJ1 U PJ2);
        if (the estimated number of errors of w <=
            (the estimated number of errors of w1 +
             the estimated number of errors of w2))
        then replace the clusters (PJ1, w1) and (PJ2, w2)
              by (PJ1 U PJ2, w) in Clusters;
    } until (no new merges can be tested);
    return Clusters;
}
```

4 Experimental results

As explained in Section 3, in order to show the performance of the clustering algorithm proposed in this paper, we considered the preference judgments of the N = 100 spectators with more ratings in EachMovie: $PJ_1, \ldots, PJ_{100}$. Let us recall that from each rating of these spectators, we built 10 preference pairs selecting randomly other 10 different ratings.

The movies are described in our experiments by vectors of 808 components: the ratings provided by the rest of the spectators who submitted at least 200 ratings for the movies with at least 1,000 ratings. The resulting 504 movies were then randomly

separated into three subsets: training 60%, verification 20%, and test 20%. We will call these datasets the *808-Collection*.

Additionally, to check the results that our algorithm can achieve with smaller sets, we also considered the *89-Collection*, where each movie is now described by the ratings of the 89 spectators, from the set of 808, with more than 275 ratings.

Notice that both collections deal with the same 504 movies. The distribution of ratings in the description of the movies, that is in the attribute space, and those provided by our 100 spectators is quite similar and uniform in the two collections considered; see Table 3.

To take into account explicitly the absence of ratings that is so frequent in these collections, and as is usual when dealing with EachMovie, we moved the scale of ratings to {-1.5, -1, -0.5, 0.5, 1, 1.5}, assuming zero as the missing value. Then, for coherence, to compute the difference of two vectors representing movies, we only considered components with ratings different to zero; in all other cases the result of the difference was considered missing and then set to zero.

Applying our clustering algorithm to 808-Collection and 89-Collection, we obtained clusters of sizes varying from 35 in the first case, or 54 in the second, to only 1 individual. The size of a cluster is the number of spectators whose preference criteria are considered similar for our algorithm. Let us emphasize that in both cases the largest clusters sum a significant part of the whole 100 spectators, which is a perfect result for selecting expert panelists, as explained in Section 2. The errors of ranking misclassifications were computed on the test sets devised for this purpose, and were recorded both for the ranking functions of the clusters and for each individual separately. The scores thus obtained are reported in Tables 4 and 5.

The main achievement of our algorithm is that, instead of having ranking functions whose accuracy was estimated with an average of about 600 test examples, now the ranking functions of the clusters have been tested with several thousands of test examples; see the second and fourth columns of Tables 4 and 5. And the price to be paid in terms of accuracy is very modest; in fact, the differences in error percentages are very slight, especially in the biggest clusters.

The consequences of having our 100 spectators arranged in these clusters are very useful. Notice that we have discovered in the biggest clusters the currents of significant opinions available in our population, and the outliers represented by the clusters with only one spectator. Therefore, the applications mentioned in section 2 can be strengthened using the ranking functions of the biggest clusters instead of those induced from individual preference judgments.

Table 3. Rating distribution in data sets used for the experiments

Movies' Rating	In attribute space				In the 100 spectator ratings	
	808-Collection		89-Collection			
	Number	Percentage	Number	Percentage	Number	Percentage
0	30,170	15.87%	4,392	17.02%	6,665	19.32%
0.2	12,073	6.35%	1,955	7.57%	2,651	7.68%
0.4	26,888	14.14%	3,813	14.77%	4,801	13.92%
0.6	48,689	25.61%	6,373	24.69%	8,193	23.75%
0.8	45,742	24.06%	5,774	22.37%	7,471	21.66%
1	26,572	13.98%	3,503	15.57%	4,716	13.67%
Totals	190,134		25,810		34,497	

Table 4. 808-Collection. Generalization errors of ranking functions of the biggest clusters computed with test examples and the average of individuals separately. The clusters are ordered by the number of spectators included

Number of spectators	By Clusters		Individually		Error Difference
	Test ex.	Errors	Avg. test ex.	Errors	
35	22,590	20.06%	645	18.32%	-1.74%
13	7,770	21.47%	605	20.75%	-0.72%
5	3,420	17.02%	684	14.82%	-2.19%
4	2,280	37.50%	570	33.29%	-4.21%
4	2,650	28.19%	662	23.55%	-4.64%

Table 5. The scores of Table 4, for the 89-Collection

Number of spectators	By Clusters		Individually		Error Difference
	Test ex.	Errors	Avg. test ex.	Errors	
54	34,990	21.16%	648	21.20%	0.03%
18	12,240	27.60%	680	26.08%	-1.52%
9	5,210	27.85%	579	29.02%	1.17%
4	2,040	35.64%	510	32.40%	-3.24%
4	2,590	33.78%	647	33.59%	-0.19%

5 Conclusions

We have presented a new algorithm to build clusters of preference criteria. Starting from a collection of preference judgments of different people, our algorithm discovers groups of people with closely related tastes, that is to say people whose preference judgments can be merged in order to learn more reliable ranking functions able to express the preferences of the people involved. The key insight involved here is that ranking functions, learned from each preference judgment set, codify the rationale for these preferences. Experimental results, using the EachMovie database, show the satisfactory performance of our approach.

We believe that the contributions of this paper fall mainly within the applications of preference learning. Thus, clustering is useful for strengthening applications such as collaborative filtering, information retrieval or adaptive assistants. Moreover, this algorithm opens the possibility of developing new applications to handle sensory data provided by panels. This type of data is very important in the food industry for the design and control of product quality.

Acknowledgements

The research reported in this paper has been supported in part under Spanish Ministerio de Ciencia y Tecnología (MCyT) and Feder grant TIC2001-3579. The authors would additionally like to thank the Company Compaq for providing us with the EachMovie database (McJones, 1997), and Thorsten Joachims for his SVM^{light} (Joachims, 1998) used in the experiments.

References

Basu, C., Hirsh, H., and Cohen, W.W. 1998. Recommendation as classification: Using social and content-based information in recommendation. In Proc. AAAI. pp. 714-720.

Branting, K.L. and Broos, P.S. 1997. Automated acquisition of user preferences. International Journal of Human-Computer Studies 46:55-77.

Breese, J.S., Heckerman, D., and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Proc. Conference on Uncertainty in Artificial Intelligence. pp. 43-52.

Buck, D., Wakeling, I., Greenhoff, K., and Hasted, A. 2001. Predicting paired preferences from sensory data. Food quality and preference 12:481-487.

Cheung, W.K., Kwok, J.T., Law, M.H., and Tsui, K.C. 2000. Mining customer preference ratings for product recommendations using the support vector machine and the latent class model. In Proc. International Conference on Data Mining Methods and Databases for Engineering. pp. 601-610.

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

Corney, D. 2002. Designing food with bayesian belief networks. In Proc. International Conference on Adaptive Computing in Engineering Design and Manufacture. pp. 83-94.

Crammer, K. and Singer, Y. 2001. Pranking with ranking. In Proc. Conference on Neural Information Processing Systems. pp. 641-647.

Díez, J., Bahamonde, A., Alonso, J., López, S., del Coz, J.J., Quevedo, J. R., Fernández, I., Luaces, O., Alvarez, I., Royo L.J., and Goyache, F., 2003. Artificial intelligence techniques point out differences in classification performance between light and standard bovine carcasses. Meat Science, 64:249-258.

Díez, J., del Coz, J.J., Luaces, O., Goyache, F., Peña, A. M., and Bahamonde, A. 2002. Learning to assess from pair-wise comparisons. In Proc. Ibero-American Conference on Artificial Intelligence. Lecture Notes in Computer Sciences Vol. 2527:481-490.

Dumais, S., Bharat, K., Joachims, T., and Weigend, A. 2003. Workshop on implicit measures of user interests and preferences. In ACM SIGIR Conference, Toronto, Canada.

Fiechter, C.N. and Rogers, S. 2000. Learning subjective functions with large margins. In Proc. International Conference on Machine Learning. pp. 287-294.

Freund, Y., Iyer, R., Schapire, R.E., and Singer, Y. 1998. An efficient boosting algorithm for combining preferences. In Proc. International Conference on Machine Learning. pp. 170-178.

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

Goyache, F., Bahamonde, A., Alonso, J., López, S., del Coz J.J., Quevedo, J.R., Ranilla, J., Luaces, O., Alvarez, I., Royo, L., and Díez, J. 2001a. The usefulness of artificial intelligence techniques to assess subjective quality of products in the food industry. Trends in Food Science and Technology 12 (10):370-381.

Goyache, F., del Coz, J.J., Quevedo, J.R., López, S., Alonso, J., Ranilla, J., Luaces, O., Alvarez, I., and Bahamonde, A. 2001b. Using artificial intelligence to design and implement a morphological assessment system in beef cattle. Animal Science 73:49-60.

Herbrich, R., Graepel, T., and Obermayer, K. 1999. Support vector learning for ordinal regression. In Proc. International Conference on Artificial Neural Networks. pp.97-102.

Hofmann, T and Puzicha, J. 1999 Latent class models for collaborative filtering. In Proc. International Joint Conference on Artificial Intelligence. pp.688-693.

Joachims, T. 1998. Making large-scale SVM learning practical. Advances in Kernel Methods - Support Vector Learning, MIT Press, Cambridge MA. Chapter 11.

Joachims, T. 2000. Estimating the generalization performance of a SVM efficiently. In Proc. International Conference of Machine Learning.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In Proc. ACM Conference on Knowledge Discovery and Data Mining.

McJones, P. 1997. EachMovie collaborative filtering dataset. DEC (now Compaq) Systems Research Center. http://www.research.compaq.com/SRC/eachmovie/.

Murray, J.M., Delahunty, C.M., and Baxter, I.A. 2001. Descriptive sensory analysis: Past, present and future. Food Research International 34:461-471.

Pazzani, M.J. 1999. A framework for collaborative filtering, content-based and demographic filtering. Artificial Intelligence Review 13 (5-6):393-408.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. 1994. GroupLens: An open architecture for collaborative filtering of networks. In Proc. Conference on Computer Supported Cooperative Work. pp. 175-186.

Resnick, P. and Varian, H. 1997. Introduction to special section on recommender systems. Communicatons of the ACM 40(3):56-58.

Shardanand, U. and Maes, P. 1995. Social information filtering: Algorithms for automatic "Word of Mouth". In Proc. Conference on Human Factors in Computing Systems. pp. 210-217.

Shashua, A. and Levin, A. 2002. Ranking with large margin principle: Two approaches. In Proc. Neural Information and Processing Systems.

Tesauro, G.1989. Connectionist learning of expert preferences by comparison training. In Proc. Neural Information and Processing Systems. pp. 99-106.

Ungar, L.H. and Foster, D.P. 1998. Clustering methods for collaborative filtering. In Proc. AAAI'98 Workshop on Recommendation Systems. pp. 112-125.

Utgoff, J. P. and Clouse, J.1991. Two kinds of training information for evaluation function learning. In Proc. AAAI'91. pp. 596-600.

Vapnik, V. 1998. Statistical learning theory. John Wiley, New York.