Refinements of minimum-based ordering in between discrimin and leximin

Henri PRADE
Institut de Recherche en Informatique de Toulouse (IRIT) - CNRS
Université de Paul Sabatier, 118 route de Narbonne
31062 Toulouse Cedex, France. Email : prade@irit.fr

Abstract: The discrimin ordering between two vectors of evaluations of equal length amounts to apply the minimum-based ordering, once identical components having the same rank in the two vectors have been deleted. This short, informal, note suggests extensions of the idea underlying the discrimin ordering, where *subsets* of components in the two vectors are judged to be equivalent (up to a permutation) and can thus be ignored in the comparison.

1. Background

Let $\mathbf{u} = (u_1, ..., u_i, ..., u_n)$ and $\mathbf{v} = (v_1, ..., v_i, ..., v_n)$ be two vectors having the *same length* to be compared. The u_i's and v_i's are assumed to belong to a linearly ordered scale, e.g. [0, 1], or a finite subset of it, such as {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} for instance. The u_i's and the v_i's can be thought of as the values of criteria is for two distinct objects, assuming the commensurateness of the criteria.

Then the discrimin ordering (Fargier *et al.*, 1993 ; Dubois *et al.*, 1996, 1997) can be defined in the following way. Let $D_1(\mathbf{u},\mathbf{v}) = \{i, \text{ such that } u_i = v_i\}$.

Then $\mathbf{u} >_{discrimin} \mathbf{v} \Leftrightarrow \min_{j \notin D_1(u,v)} u_j > \min_{j \notin D_1(u,v)} v_j$.

The discrimin ordering clearly refines the minimum-based ordering and the Pareto ordering:

$\mathbf{u} >_{Pareto} \mathbf{v} \Leftrightarrow \forall i\ u_i \geq v_i \text{ and } \exists_j\ u_j > v_j$;

$\mathbf{u} >_{min} \mathbf{v} \Leftrightarrow \min_i u_i > \min_i v_i$.

Note that in the above definition of discrimin, only identical components having the same *rank* (i.e. corresponding to the same criterion) and the same value are ignored in the comparison.

The leximin ordering, which refines the discrimin ordering, is also based on the idea of deleting identical elements when comparing the two vectors, but *once* the two vectors have been reordered in a non-decreasing order.

Thus with classical discrimin, comparing
$\mathbf{u} = (.2, .5, .3, .4, .8)$ and
$\mathbf{v} = (.2, .3, .5, .6, .8)$
amount to compare vectors $\mathbf{u'}$ and $\mathbf{v'}$ where
$\mathbf{u'} = (.5, .3, .4)$
$\mathbf{v'} = (.3, .5, .6)$
since $u_1 = v_1 = .2$ and $u_5 = v_5 =.8$. Thus, $\mathbf{u} =_{min} \mathbf{v}$ [1] . We still have $\mathbf{u} =_{discrimin} \mathbf{v}$, since $\mathbf{u'} =_{min} \mathbf{v'}$, but $\mathbf{u'} <_{leximin} \mathbf{v'}$ and then $\mathbf{u} <_{leximin} \mathbf{v}$.

---

[1] If $\mathbf{u} \leq_x \mathbf{v}$ and $\mathbf{v} \leq_x \mathbf{u}$, we write $\mathbf{u} =_x \mathbf{v}$ for x = min, discrimin or leximin.

2. Extending the discrimin: the idea

Classical discrimin is based on the elimination of identical singletons at the same places in the comparison process of the two sequences. More generally, we can work with 2 elements subsets which are identical *and* pertain to the same pair of criteria.

Namely in the above example, we may consider that (.5, .3) and (.3, .5) are "equilibrating" each other. Note that it supposes that the two corresponding criteria have the same importance. Then we delete them, and we are led to compare

$\mathbf{u}'' = (.4)$

$\mathbf{v}'' = (.6)$.

Let us take another example:

$\mathbf{u}_2 = (.5, .4, .3, .7, .9)$

$\mathbf{v}_2 = (.3, .9, .5, .4, 1)$

then we would again delete (.5, .3) with (.3, .5) yielding:

$\mathbf{u}'_2 = (.4, .7, .9)$

$\mathbf{v}'_2 = (.9, .4, 1)$.

Note that in this example we do not simplify .4, .9 with .9, .4 since they do not pertain to the same pair of criteria.

Moreover, simplifications can take place only one time. Thus, if the vectors are of the form $\mathbf{u} = (x, y, x, s)$ and $\mathbf{v} = (y, x, y, t)$ (with $\min(x, y) \leq \min(s, t)$ in order to have the two vectors min-equivalent), we may either delete components of ranks 1 and 2, or of ranks 2 and 3, leading in both cases to compare (x, s) and (y, t), and to consider the first vector as smaller in the sense of the order 2-discrimin, as soon as $x < \min(y, s, t)$.

We can now introduce the definition of the (order) 2-discrimin. Let us build a set $D_2(\mathbf{u},\mathbf{v})$ as $\{i,j$, such that $u_i = v_j$ and $u_j = v_i$ and if there are several such pairs, they have no common components$\} \cup \{k, u_k = v_k\}$. Then the 2-discrimin is just the minimum-based ordering once components corresponding to pairs in $D_2(\mathbf{u},\mathbf{v})$ are deleted. Note that $D_2(\mathbf{u},\mathbf{v})$ is not always unique as shown by the above example. However this does not affect the result of the comparison of the vectors after the deletion of the components in $D_2(\mathbf{u},\mathbf{v})$ as it can be checked from the above formal example, since the minimum aggregation is not sensitive to the place of the terms. The 2-discrimin also includes the deletion of identical components as in the ordinary discrimin, since it would be strange to delete pairs of identical values in the comparison but not single identical values (which may blur the comparison).

The 2-discrimin refines the discrimin ordering. It deletes pairs of values which play a neutral role in the aggregation, but which may lead to ties if these values are not ignored. Indeed in the first example $\mathbf{u} =_{\text{discrimin}} \mathbf{v}$, while $\mathbf{u} \leq_{\text{2-discimin}} \mathbf{v}$.

3. Order k - discrimin

Clearly we can work with 3 elements-sets as well or more generally with k elements-sets. The two sub-vectors should be identical up to a permutation. This means that for the corresponding subset of criteria, any symmetrical combination of the three (or more generally k) evaluations yields the same result for each vector. Then these components can be ignored in the comparison of the vectors.

For instance, the two vectors

$\mathbf{u}_3 = (.3, .5, .2, 1, .6)$

$\mathbf{v}_3 = (.6, .3, .4, .7, .5)$

can be simplified into

$\mathbf{u'}_3 = (.2, 1)$

$\mathbf{v'}_3 = (.4, .7)$.

since (0.3, 0.5, 0.6) and (0.6, 0.3, 0.5) are equivalent up to a permutation on the set of ranks (1, 2, 5). It can be done for larger sets as well.

Let us call the classical discrimin '1-discrimin', and the others '2-discrimin', '3-discrimin', and so on. For 3-discrimin, we have to build a set

$D3(\mathbf{u},\mathbf{v}) = \{i,j,k, \exists$ a permutation σ defined on $\{i,j,k\}$, $u_i = v_{\sigma(i)}$ and $u_j = v_{\sigma(j)}$ and $u_k = v_{\sigma(k)}$ and if there are several such permutations they do not overlap$\} \cup D_2(\mathbf{u},\mathbf{v})$.

Note that 3-discrimin incorporates 2-discrimin and 1-discrimin. Again there may exist several ways of building $D_3(\mathbf{u},\mathbf{v})$. Then the situation may become more intricate than with the 2-discrimin as shown by the following example:

Let $\mathbf{u}$ = (0.3, 0.5, 0.8, 1., 0.6, 0.7, 0.5)

$\mathbf{v}$ = (0.6, 0.3, 0.4, 0.7, 0.5, 0.6, 0.7)

There exist two overlapping permutations :

(0.3, 0.5, 0.6) with (0.6, 0.7, 0.5)

and (0.6, 0.3, 0.5) with (0.5, 0.6, 0.7).

Ignoring the values involved in the first permutation, we obtain $\mathbf{v} <_{min} \mathbf{u}$ since $0.4 < 0.5$, while using the second permutation, we get $\mathbf{v} =_{min} \mathbf{u}$ since both vectors of remaining values lead to the same minimum 0.3. This points out that the definition of the 3-discrimin should be further refined by choosing the permutation, which lead to a discriminant situation if possible. Note that this permutation should involve the minimum value of the vectors, to be of interest.

Moreover, let us consider two vectors of the form

$\mathbf{u}$ = (a x b b s)

$\mathbf{v}$ = (b a x a t).

Simplifying (a, x, b) with (b, a, x) leads to compare (b, s) with (a, t), while simplifying the components for i = 1 and i = 4 (i.e., (a, b) with (b, a)) would lead to compare (x, b, s) with (a, x, t). This shows (consider the case where x < min(a, b, s, t) that when triples and pairs are overlapping, the deletion of the triples can be more efficient in the comparison (as soon as we are using the 3-discrimin and not only the 2-discrimin).

However they cannot exist two overlapping permutations which, after ignoring their components in the min-based comparison, would lead to opposite orderings (namely $\mathbf{u} > \mathbf{v}$ and $\mathbf{v} > \mathbf{u}$). This holds for the k-dicrimin as well. This can be seen by considering vectors of the form

$\mathbf{u}$ = (t a b s x t y) and

$\mathbf{v}$ = (b s a t s x z)

where y and z are such that $\mathbf{u} =_{min} \mathbf{v}$. Simplifying by the first four components leads to compare min(x, t, y) with min(s, x, z), while deleting (s, x, t) with (t, s, x) leads to the comparison of min(t, a, b, y) with min(b, s, a, z). Clearly the two comparisons cannot disagree with each other (i.e. cannot lead to $\mathbf{u} > \mathbf{v}$ and $\mathbf{v} > \mathbf{u}$ respectively). This remark could be further generalized by replacing some of the above vector components by sub-vectors. In case of several overlaps, this analysis can be iterated.

So the generalized procedure for applying k-discrimin is to look for permutations of orders 1, 2, ..., k and to explore the different possibilities in case of overlapping permutations until a discriminating one is found.

k discrimin and leximin

Besides, note that 'k-discrimin' amounts to a limited leximin on k-long subsequences. Thus 'k-discrimin' provides orderings in between discrimin and leximin. However the n-discrimin ordering may remain less discriminating than the leximin ordering as shown by the following example.

u = (0.4, 0.5, 0.3, 0.7, 0.6)
v = (0.6, 0.3, 0.5, 0.4, 1).

Then $\mathbf{u} <_{\text{leximin}} \mathbf{v}$ (since 0.7 < 1),
while $\mathbf{u} =_{\text{5-discrimin}} \mathbf{v}$ (since (0.4, 0.7, 0.6) and (0.6, 0.4, 1) have the same minimum), since 5-discrimin and 2-discrimin are equivalent here.

4. Concluding remarks

Discrimin and more generally k-discrimin, which takes advantage of the perfect identity of the components, are well in the spirit of discrete scales.

We may think of further generalizing the above approach in the following way. We can simplify vectors to be compared by deleting components i and j, if for some meaningful aggregation function f, we have $f(u_i, u_j) = f(u_i, v_j)$, here extending the 2-discrimin. For instance, we may simplify (0.4, 0.6) with (0.5, 0.5) because they have the same average inside vectors which are originally min-equivalent (i.e. such as $\min_k u_k = \min_k v_k$). For instance, considering the two vectors **u** = (0.3, 0.4, 0.6, 0.8) and **v** = (0.3, 0.5, 0.5, 0.4), this may lead (together with ordinary discrimin) to find **v** < **u**, while $\mathbf{u} =_{\text{discrimin}} \mathbf{v}$. However, this may be delatable in a situation where **u** is to be compared to **v'** = (0.3, 0.5, 0.5, 0.7) since it would lead to **v'** < **u** while $\mathbf{u} <_{\text{discrimin}} \mathbf{v'}$! This points out that this may be used only for vectors which are discrimin equivalent. Moreover, the non-unicity of the extension of $D_2(\mathbf{u},\mathbf{v})$ may also create difficulties for some functions f.

Another extension of discrimin ordering have been recently proposed by Dubois and Fortemps (1999) for vectors of unequal lengths leading to the general notion of delocalized discrimin where vectors can be completed by introducing the top element of the scale, i. e., 1 here, in places in between components of the original vectors. In that extension, components of the two vectors having the same rank, e.g. u_i, and v_i, may have different ranks once the '1' are added. This clearly corresponds to an extension which differs from the above approach. We may think of combining the two ideas.

References

Dubois D., Fargier H., Prade H. (1996) "Refinements of the maximum approach to decision-making in fuzzy environment ." *Fuzzy Sets and Systems*, 81, 103-122.

Dubois D., Fargier H., Prade H. (1997) "Beyond min aggregation in multicriteria decision: (ordered) weighted min, discri-min, leximin." *The ordered weighted averaging operators. Theory and Applications.* (R.R. Yager, J. Kacprzyk, eds.) Kluwer Academic Publishers, 180-192.

Dubois D., Fortemps Ph. (1999) "Choix des solutions préférées pour la programmation dynamique flexible." *Actes Rencontres francophones sur la Logique Floue et ses Applications* (LFA-99), Cépaduès, Toulouse, 88-95.

Fargier H., Lang J., Schiex T. (1993) "Selecting preferred solutions in fuzzy constraint satisfaction problems.' *In Proc. of the 1st Europ. Congress on Fuzzy and Intelligent Technologies (EUFIT'93),* ELITE, Aachen, Germany, 1128-1134.