CHAPTER 6. ClusDM Properties

In this chapter, we will study in detail the properties of the ClusDM method. As it is a general procedure rather than a concrete algorithm, we will concentrate on the study of this method when it is applied to a set of qualitative preference criteria. In this case, this decision making technique can be seen as an aggregation operator since we are trying to capture the knowledge provided by a set of qualitative criteria and summarise it with a single overall criterion.

To study some of the properties, we will consider that the vocabulary used in the new preference criterion is specified by the user instead of being selected from the available ones. The reason for this assumption is that if the vocabulary was automatically selected, we could not study some of the properties, since the final vocabulary will not necessarily be the same after applying ClusDM to different data sets.

Formally, we will denote as Θ the ClusDM operator, which is defined over a set of qualitative preference criteria. The result, c_r, is a new qualitative preference criterion that takes into consideration the *p* preferences provided by the data suppliers: $c_1,...,c_p$.

$$\Theta(c_1,...,c_p) = c_r$$

It must be noted that the units that are aggregated are criteria (i.e. the columns in our decision matrix). For this reason, the property of individual independence must not be required to the rows (i.e. alternatives). In this case, it has no sense to obtain the final preference of the alternatives without considering the relationships among them, because the similarity relationships from one alternative to the others will determine its final value of preference.

For example, let us consider the two decision matrices, X1 and X2 of Figure 21. They have a unique common alternative, *a5=b5*. We can see that in the first case, X1, the alternative *a5* is placed in the second position in the ranking, with the value of "good", whereas in the second case, X2, an identical alternative, *b5*, is placed in the first position, with a preference value of "very good". This example shows that the final preference of an alternative depends on the values of the other alternatives to which it is compared.

	c_1	c_2	c_r
a1	vgood	vgood	vgood
a2	vgood	vgood	vgood
a3	vgood	good	good
a4	vgood	good	good
a5	vgood	good	good
a6	bad	good	reg
a7	bad	good	reg
a8	bad	good	reg
a9	bad	vbad	bad
a10	bad	vbad	bad
a11	vbad	vbad	vbad

X_1

	c_1	c_2	c_r
b1	good	good	good
b2	good	good	good
b3	good	good	good
b4	good	vgood	vgood
b5	vgood	good	vgood
b6	good	bad	reg
b7	good	bad	reg
b8	good	bad	reg
b9	bad	good	reg
b10	bad	bad	bad
b11	vbad	bad	vbad

X_2

Figure 21. Qualitative matrices with a common alternative

We will now see the properties that are fulfilled by our Multiple Criteria Decision Making procedure, ClusDM, for qualitative preferences. The definition of each property is followed by an explanation to proof whether ClusDM satisfies the property or not.

6.1 Basic Aggregation Properties

We will begin with the study of three properties that are usually required to aggregation and decision making operators: symmetry, idempotency and monotonicity [Marichal,1999b].

6.1.1 Symmetry

It is also known as commutativity, neutrality or anonymity.

The symmetry property is fulfilled if the ordering of the criteria does not affect to the result. This is required when we combine criteria of equal importance or anonymous expert's opinions.

Θ is a symmetric operator if, for all criteria $c \in C$ and all $\pi \in \Pi_p$ (where Π_p corresponds to the set of all permutations of dimension *p*), we have:

$$\Theta(c_1,...,c_p) = \Theta(c_{\pi(1)},...,c_{\pi(p)})$$

In situations when criteria or individual opinions are not equally important, the symmetry property must be omitted.

Symmetry of ClusDM:

In the case of having equally weighted (or non-weighted) criteria, the symmetry of ClusDM depends on the clustering algorithm. If the clustering does not take into account the order of the criteria, we will obtain the same set of clusters, and the following stages will give the same results.

The clustering builds clusters according to the values in the similarity matrix. So, if the similarity function is symmetric, the grouping stage will not be affected. As we recommend the use of the Manhattan distance and this metric is symmetric, we have that ClusDM is symmetric.

However, if we would like to use another similarity function, we can easily see that the types considered in *Sedàs* [Valls et al.,1997] are also symmetric:

Similarity Calculation	***Properties***	***Sy?***
Functions based on Distances	The result is obtained from a summatory of distances. Being distances symmetric, the addition is also symmetric.	Yes
Association Coefficients	The result is based on some counters that are independent of the position of the criteria	Yes
Correlation Coefficients	Correlation is independent of the ordering of the criteria	Yes

Table 12. Symmetry of similarity functions

6.1.2 Idempotence

This property is also called unanimity. It refers to the case of having a data matrix with equal columns, that is, all the criteria give exactly the same value to the alternatives. In this case, the expected result is to have, at the end, the same value for each alternative.

$$\Theta(c,...,c) = c$$

Idempotence of ClusDM:

To check this property we will analyse the aggregation, ranking and explanation stages separately.

STAGE 1. With regard to the aggregation, the clustering will create a cluster for each term in the vocabulary of this criterion, *c*.

STAGE 2. The ranking stage can be performed using the Principal Components Analysis or the Similarity Ranking. However, if all the criteria have the same vocabulary and semantics, and they give the same value to the same alternatives, the correlation among them will be maximum. That is, we will be in CASE A and apply the PCA.

In this case, the p-dimensional space will be reduced to a 1-dimensional space, since all the dimensions are equal. Thus, the space of criteria is a line, which is the first principal component that we will obtain if we calculate the PCA. The projection of the prototypes of the clusters in this line will not modify the ranking. Moreover, the value attached to each cluster will be the central value of the interval corresponding to the term that has originated the class.

STAGE 3. In this stage we will select the vocabulary of this unique criterion, *c*, to explain the clusters (CASE D). The algorithm that selects the vocabulary will select the same terms that were originally assigned to the alternatives, because the value attached to each cluster is the central point of these intervals.

Therefore, we can say that this method is idempotent.

In addition, it is important to underline that if the vocabularies are different but they have the same granularity and the same semantics, the idempotency property will be fulfilled too.

6.1.3 Monotonicity

This property refers to the fact that increasing (respectively, decreasing) the values of one criterion in a data matrix will produce a result that is greater than the original one.

This is, when $c_k > c'_k$, where $c_k > c'_k$ means that $c_k(a_i) > c'_k(a_i)$, for all $i = 1..m$, we will have a monotonous operator if $\Theta(c_1,...,c_k,...,c_p) \geq \Theta(c_1,...,c'_k,...,c_p)$.

That is, monotonicity is satisfied if the result of applying the decision operator to a data matrix that has better preference values in one criterion, is greater or equal than the result obtained with the other matrix.

Monotonicity in ClusDM

We show with an example, that monotonicity is not satisfied. Let us consider two data matrices, X1 and X2, which only differ in the second criterion (see Figure 22). For X2, criterion c_2+ has better values for all the alternatives than the corresponding criterion c_2 in X1, so $c_2 + > c_2$.

	c_1	c_2
a_1	e	d
a_2	d	d
a_3	d	d
a_4	d	d
a_5	e	c
a_6	c	c
a_7	c	b
a_8	b	b
a_9	b	b
a_{10}	b	b
a_{11}	b	f
a_{12}	f	f
a_{13}	g	c

X1

	c_1	c_2+
a_1	e	c
a_2	d	c
a_3	d	c
a_4	d	c
a_5	e	b
a_6	c	b
a_7	c	a
a_8	b	a
a_9	b	a
a_{10}	b	a
a_{11}	b	e
a_{12}	f	e
a_{13}	g	b

X2

Figure 22. Data matrices for the monotonicity example

Let us consider the case that all criteria have the same vocabulary {g,f,e,d,c,b,a}, with g<f<d<c<b<a. The terms have the same semantics for all criteria, which is given by the following negation function, N:

N(a)={f,g}, N(b)={f}, N(c)={e}, N(d)={d}, N(e)={c}, N(f)={a,b}, N(g)={a}

Let us suppose that we want a result described with 5 terms, then, after applying the clustering method (centroid clustering with Manhattan distance), we will have 5 clusters for each data matrix. Here you have a trace of the clustering process:

STEP 1: Build a dissimilarity matrix and put together those alternatives with minimum value.

In this case, alternatives a_2, a_3 and a_4 are the components of the first class, k_1, and a_8, a_9 and a_{10} form the second class, k_2. The elements of these clusters have dissimilarity 0, which means that they are equal (indistinguishable in the space of preferences).

STEP 2: Calculate the prototype of each new class.

For matrix X1, the prototypes are:

Prototype k_1 = (d,d) = (0.5,0.5) = $a_2 = a_3 = a_4$
Prototype k_2 = (b,b) = (0.72,0.72) = $a_8 = a_9 = a_{10}$

For matrix X2, the prototypes are:

Prototype k_1’ = (d,c) = (0.5, 0.61) = $a_2 = a_3 = a_4$
Prototype k_2’ = (b,a) = (0.72, 0.89) = $a_8 = a_9 = a_{10}$

STEP 3: Modify the dissimilarity matrix, including the new clusters and deleting their components.

	a_1	k_1	a_5	a_6	a_7	k_2	a_{11}	a_{12}	a_{13}
a_1	0	**0.11**	**0.11**	0.33	0.44	0.55	0.61	0.45	0.44
k_1		0	0.22	0.22	0.33	0.44	0.5	0.56	0.55
a_5			0	0.22	0.33	0.44	0.72	0.56	0.33
a_6				0	**0.11**	0.22	0.5	0.78	0.55
a_7					0	**0.11**	0.61	0.89	0.66
k_2						0	0.5	1	0.77
a_{11}							0	0.5	1.05
a_{12}								0	0.55
a_{13}									0

Figure 23. Dissimilarity matrix for X1 (step3)

For matrix X1, we obtain the result in Figure 23 (the red value is the minimum). For matrix X2, we obtain the result in Figure 24.

	a_1	k_1’	a_5	a_6	a_7	k_2’	a_{11}	a_{12}	a_{13}
a_1	0	**0.11**	**0.11**	0.33	0.5	0.61	0.55	0.39	0.44
k_1’		0	0.22	0.22	0.39	0.5	0.44	0.5	0.55
a_5			0	0.22	0.39	0.5	0.66	0.5	0.33
a_6				0	0.17	0.28	0.44	0.72	0.55
a_7					0	**0.11**	0.61	0.89	0.73
k_2’						0	0.5	1	0.83
a_{11}							0	0.5	0.99
a_{12}								0	0.49
a_{13}									0

Figure 24. Dissimilarity matrix for X2 (step 3)

STEP 4: Build another level of clusters.

For matrix X1, we have that the minimum value is 0.11, which creates two new clusters: $k_3 = \{a_1, k_1, a_5\}$ and $k_4 = \{a_6, a_7, k_2\}$.

For matrix X2, we have that the minimum value is also 0.11, which creates two other clusters: $k_3' = \{a_1, k_1', a_5\}$ and $k_4' = \{a_7, k_2'\}$.

STEP 5: Calculate the prototype of each new class.

For matrix X1:

Prototype k_3 = (0.456, 0.522) = average (a_1, a_2, a_3, a_4, a_5)

Prototype k_4 = (0.676, 0.698) = average ($a_6, a_7, a_8, a_9, a_{10}$)

For matrix X2:

Prototype k_3' = (0.456, 0.646) = average (a_1, a_2, a_3, a_4, a_5)

Prototype k_4' = (0.692, 0.89) = average (a_7, a_8, a_9, a_{10})

Notice, that clusters k_4' and k_4 are not equal. The difference is due to the fact that the interval covered by each term has not the same length, that is, we have terms whose interval of possible numerical values is smaller than others.

For matrix X1, we can stop the process because we have the alternatives in 5 groups: { a_1, a_2, a_3, a_4, a_5}, {$a_6, a_7, a_8, a_9, a_{10}$ }, {a_{11}}, {a_{12}} and {a_{13}}.

For the case of matrix X2, we must continue a step forward, to reduce the number of clusters, now equal to 6. So, we recalculate the similarity matrix, introducing k_3' and k_4' and removing their elements (Figure 25).

	k_3'	a_6	k_4'	a_{11}	a_{12}	a_{13}
k_3'	0	**0.23**	0.48	0.52	0.49	0.47
a_6		0	0.25	0.44	0.72	0.55
k_4'			0	0.53	0.97	0.81
a_{11}				0	0.5	0.99
a_{12}					0	0.49
a_{13}						0

Figure 25. Dissimilarity matrix for X2 (step 5)

Using the dissimilarity values in that matrix, we build a new cluster with the elements of k_3' and a_6. The prototype of this new class, k_5', is (0.482, 0.647), which corresponds to the arithmetic average of a_1, a_2, a_3, a_4, a_5 and a_6.

At the end of the aggregation of X2, we have obtained the following 5 groups: {$a_1, a_2, a_3, a_4, a_5, a_6$}, {$a_7, a_8, a_9, a_{10}$ }, {a_{11}}, {a_{12}}and {a_{13}}.

Notice that, the partitions generated by the same methodology with the two data matrices are different. Alternative a_6 belongs to different clusters because the increase of the values of one criterion has modified the relationships among the alternatives.

To establish a ranking of the clusters, we use the Manhattan distance with respect to the *Ideal* alternative, which in this case is I = (1.0,1.0).

	Elements of the class	**Prototype**	**Similarity to the Ideal**	**Rank**
X1	$\{a_1, a_2, a_3, a_4, a_5\}$	(0.456, 0.522)	1.022	2
	$\{a_6, a_7, a_8, a_9, a_{10}\}$	(0.676, 0.698)	0.626	1
	$\{a_{11}\}$	(0.72, 0.22)	1.06	3
	$\{a_{12}\}$	(0.22, 0.22)	1.56	5
	$\{a_{13}\}$	(0.05, 0.61)	1.34	4
X2	$\{a_1, a_2, a_3, a_4, a_5, a_6\}$	(0.482, 0.647)	0.871	2
	$\{a_7, a_8, a_9, a_{10}\}$	(0.692, 0.89)	0.418	1
	$\{a_{11}\}$	(0.72, 0.39)	0.89	3
	$\{a_{12}\}$	(0.22, 0.39)	1.39	5
	$\{a_{13}\}$	(0.05, 0.72)	1.23	4

Table 13. Ranking of X1 and X2 based on similarities

In this example, we can see that if we change the values of a particular criterion (increasing them), we can modify the relations among the alternatives, which produces a new classification. In case of obtaining different clusters, some alternatives can be positioned lower in the new criterion, because now they are more similar to other alternatives with lower values. Alternative a_6, who had the best position in the ranking from X1, has the second position considering data in X2.

In conclusion, ClusDM is not monotonous in general.

This is so because when the increase in the preference value of all the alternatives is not constant, the relationships among them may be modified, which produces new clusters, and some alternatives will decrease its preference in the resulting ranking.

However, if we increase the value of all the alternatives in the same degree[8], we only produce a translation in the space without affecting the relationships among the

[8] This case is possible only if the semantics of the criterion modified is given by the classical negation function, which gives equal informativeness to all the terms.

alternatives. In this case, the result will be greater or equal, depending if the increase is enough to move the clusters to receive the next term in the vocabulary. If it is the case, the property fulfilled could be stated as:

$$\Theta(v_{i1}+d,\ldots,v_{ip}+d)=\Theta(v_{i1},\ldots,v_{ip})+d$$

If we change each term for the one that is *d* positions up or down (a better one or a worse one), the result shows an increase or decrease of the same degree, *d*, in the value of the alternatives.

Let us now study the case that the vocabularies of our criteria have equal or different granularity but with the semantics given by the classical negation function (to assure terms equally informative).

In this case, the clustering process will produce the same clusters with the initial data and with the values increased (or decreased) in *d* units, because the differences between the values do not change if terms are equally informative.

The ranking stage may apply the Principal Components Analysis or the Distance Calculation. With the change of the terms, the values of the numerical prototypes suffer a transformation of Δ_i units in each criterion, where Δ_i depends on the granularity of each criterion c_i, to that:

$$\Delta_i = d \cdot length_i(t) \qquad \text{Eq. 6.1}$$

being $length_i(t)$ the length of the interval corresponding to any term *t* of the vocabulary of the *i*-th criterion.

Notice that, although we modify each criterion with a different value, Δ_i, we are only performing a translation of the points in the p-space. Let us now study in detail the ranking process for the two possible approaches:

- PCA:

The translation of the prototypes is counteracted by the use of a correlation[9] matrix to generate the principal components. Thus, the *U* vectors will be the same than the ones obtained with the original matrix. For this reason, although the clusters have suffered a translation in the original alternatives space, their z-scores will remain the same. However, the values of the *ideal* and *nadir* alternatives will not be modified (they do not increase or decrease *d* units), so, their z-scores will be different. As the values of these to extreme fictitious alternatives are used to scale the z-scores to the interval

[9] Using the correlation matrix we work with centred and standardised values.

[0,1], we will obtain new positions for the clusters, although the ranking will be the same.

To measure the change in the position of the clusters, we must know the change in the position of the reference alternatives (the ideal and nadir points). Both of them will have a new z-score that differs from the one in the original matrix in λ units. Being z_1 the value in the first analysis, calculated as follows,

$$z_1 = u_{11}\left(\frac{x_1 - \overline{x_1}}{s_1}\right) + u_{12}\left(\frac{x_2 - \overline{x_2}}{s_2}\right) + \ldots + u_{1p}\left(\frac{x_p - \overline{x_p}}{s_p}\right)$$

and z_1^{Δ} the score corresponding to the second analysis (after the increment),

$$z_1^{\Delta} = u_{11}\left(\frac{x_1 - \left(\overline{x_1} + \Delta_1\right)}{s_1}\right) + u_{12}\left(\frac{x_2 - \left(\overline{x_2} + \Delta_2\right)}{s_2}\right) + \ldots + u_{1p}\left(\frac{x_p - \left(\overline{x_p} + \Delta_p\right)}{s_p}\right)$$

the difference between them, λ, is given by equation Eq. 6.2.

$$\lambda = -\sum_{i=1}^{p} \frac{\Delta_i}{s_i} u_{1i} \qquad \text{Eq. 6.2}$$

Consequently, if the z-scores of the clusters have not been modified, and the ones of the ideal and nadir points are increased in λ units (see that λ is a negative value), the clusters will be nearer the ideal than in the first analysis.

However, this difference, λ, is given in units without having into account the scaling of the z-scores to the interval [0,1], if we perform this scaling, we have equation Eq. 6.3.

$$\lambda_{01} = \frac{|\lambda|}{z_1(a_{ideal}) - z_1(a_{nadir})} \qquad \text{Eq. 6.3}$$

Following with the ClusDM procedure, the explanation stage will use the vocabulary given by the user to explain the result based on the position of each cluster in the interval [0,1].

Being t any term of this vocabulary, and *length*(t) the length of the interval corresponding to each term, if $(d+1) \cdot length(t) > \lambda_{01} \geq d \cdot length(t)$, the terms selected will be exactly the ones that are *d* positions up or down with respect to the ones selected with the original data.

To check this condition, we will study each inequality separately. So, we must check if the following conditions are true:

Condition 1: $\lambda_{01} \geq d \cdot length(t)$

Condition 2: $(d+1) \cdot length(t) > \lambda_{01}$

Beginning with condition 1, we substitute the value of λ_{01} in equation Eq. 6.3, obtaining:

$$\frac{\sum_{i=1}^{p} \frac{\Delta_i}{s_i} u_{1i}}{z_1(a_{ideal}) - z_1(a_{nadir})} \geq d \cdot length(t)$$

which can be rewritten as:

$$\sum_{i=1}^{p} \frac{\Delta_i}{s_i} u_{1i} \geq d \cdot length\ (t) \cdot \left(z_1(a_{ideal}) - z_1(a_{nadir})\right)$$

if we expand the expression corresponding to Δ_i, we have:

$$\sum_{i=1}^{p} \frac{d \cdot length_i(t)}{s_i} u_{1i} \geq d \cdot length(t) \cdot \left(z_1(a_{ideal}) - z_1(a_{nadir})\right)$$

which can be simplified as:

$$\sum_{i=1}^{p} \frac{length_i(t)}{s_i} u_{1i} \geq length(t) \cdot \left(z_1(a_{ideal}) - z_1(a_{nadir})\right)$$

Substituting $z_1(a_{nadir})$ and $z_2(a_{nadir})$ by their corresponding expression in terms of *U* (Eq. 4.7), we can easily see that if the ideal is an alternative with the highest values (1,1,....,1) and the nadir is an alternative with the lowest values (0,0,....,0), we have that $z_1(a_{ideal}) - z_1(a_{nadir})$ is equal to:

$$u_{11}\left(\frac{1-\overline{x_1}-(0-\overline{x_1})}{s_1}\right)+u_{12}\left(\frac{1-\overline{x_2}-(0-\overline{x_2})}{s_2}\right)+...+u_{1p}\left(\frac{1-\overline{x_p}-(0-\overline{x_p})}{s_p}\right)=\sum_{i=1}^{p}\frac{1}{s_i}u_{1i}$$

Substituting this result in the previous inequality, we obtain that:

$$\sum_{i=1}^{p}\frac{length_i(t)}{s_i}u_{1i} \geq \sum_{i=1}^{p}\frac{length(t)}{s_i}u_{1i}$$

$$\sum_{i=1}^{p}\frac{length_i(t)}{s_i}u_{1i} - \sum_{i=1}^{p}\frac{length(t)}{s_i}u_{1i} \geq 0$$

If we separate *length*(*t*), which is the value we are analysing, we have:

$$\sum_{i=1}^{p}\frac{u_{1i}}{s_i}length_i(t) \geq length(t)\cdot\sum_{i=1}^{p}\frac{u_{1i}}{s_i}$$

So, the length of the terms is constrained according to equation Eq. 6.4.

$$\frac{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}length_i(t)}{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}} \geq length(t) \qquad \text{Eq. 6.4}$$

Now, we proceed to study the condition number 2, to select exactly the terms *d* position up (or down) and not another greater (or smaller).

Substituting λ_{01} we have:

$$\frac{\sum_{i=1}^{p}\frac{\Delta_i}{s_i}u_{1i}}{z_1(a_{ideal})-z_1(a_{nadir})} < (d+1)\cdot length(t)$$

Then, we can perform an analysis equal to the one done with the other inequality, to find out the following expression:

$$d\cdot\sum_{i=1}^{p}\frac{length_i(t)}{s_i}u_{1i} < (d+1)\cdot length(t)\cdot\sum_{i=1}^{p}\frac{1}{s_i}u_{1i}$$

We can rewrite this expression and obtain:

$$d \cdot \sum_{i=1}^{p} \frac{length_i(t)}{s_i} u_{1i} - d \cdot length(t) \cdot \sum_{i=1}^{p} \frac{1}{s_i} u_{1i} < length(t) \cdot \sum_{i=1}^{p} \frac{1}{s_i} u_{1i}$$

which is the same that:

$$d \cdot \left(\sum_{i=1}^{p} \frac{length_i(t) - length(t)}{s_i} u_{1i} \right) < \sum_{i=1}^{p} \frac{length(t)}{s_i} u_{1i}$$

$$\sum_{i=1}^{p} \frac{length(t)}{s_i} u_{1i} - \sum_{i=1}^{p} \frac{d \cdot length_i(t) - d \cdot length(t)}{s_i} u_{1i} > 0$$

Now, we will separate the *length*(*t*) variable to know what is it constrained for. So, we separate the constant elements from the ones depending on the variable *i*.

Putting all together, we have:

$$length(t) \cdot \sum_{i=1}^{p} \frac{u_{1i}}{s_i} + d \cdot length(t) \cdot \sum_{i=1}^{p} \frac{u_{1i}}{s_i} - d \cdot \sum_{i=1}^{p} \frac{u_{1i}}{s_i} length_i(t) > 0$$

$$(d+1) \cdot length(t) \cdot \sum_{i=1}^{p} \frac{u_{1i}}{s_i} > d \cdot \sum_{i=1}^{p} \frac{u_{1i}}{s_i} length_i(t)$$

We obtain that the length of the terms in the vocabulary must be:

$$length(t) > \frac{d}{(d+1)} \cdot \frac{\sum_{i=1}^{p} \frac{u_{1i}}{s_i} length_i(t)}{\sum_{i=1}^{p} \frac{u_{1i}}{s_i}} \qquad \text{Eq. 6.5}$$

Consequently, we have that the property will be true iff the equations Eq. 6.4 and Eq. 6.5 hold.

$$\frac{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}\,length_i(t)}{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}} \geq length(t) > \frac{d}{(d+1)}\cdot\frac{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}length_i(t)}{\sum_{i=1}^{p}\frac{u_{1i}}{s_i}} \qquad \text{Eq. 6.6}$$

The expressions that are restricting the length of the terms in the final vocabulary are weighted averages. It is interesting to note that the weights are proportional to the contribution of the criteria (i.e. experts) to the formation of the first principal component (which is the meaning of the U-vectors, see section 4.1.4). Moreover, we can see that these weights are inversely proportional to the standard deviation of the criterion, which is a measure of the data dispersion.

- Similarity:

The translation of the prototypes makes them to be nearer (or farther) from the ideal alternative in λ units. Thus, if $(d+1)\cdot length(t) > \lambda_{01} \geq d\cdot length(t)$, the terms selected in the final vocabulary will be exactly the ones *d* positions up (or down).

In this case λ depends on the similarity function. Let us study the case of the Manhattan distance, which is the one we recommend. We will see that the interpretation of the result in this case is straightforward.

Taking the Manhattan distance (Eq.3.3), we can see that the distance to the ideal point will be modified with a value equal to:

$$\lambda = \sum_{i=1}^{p}\Delta_i$$

However, to work with values in the interval [0,1], we scale the distance obtained with the Manhattan calculation by dividing it by the distance between the nadir and the ideal points, which is:

$$\lambda_{01} = \frac{\sum_{i=1}^{p} \Delta_i}{\sum_{i=1}^{p} |1-0|} = \frac{\sum_{i=1}^{p} \Delta_i}{p} \qquad \text{Eq. 6.7}$$

Now, we can check the two conditions required:

Condition 1: $\lambda_{01} \geq d \cdot length(t)$

Condition 2: $(d+1) \cdot length(t) > \lambda_{01}$

Beginning with condition 1, we substitute the value of λ_{01} in Eq. 6.7, obtaining:

$$\frac{\sum_{i=1}^{p} \Delta_i}{p} \geq d \cdot length(t)$$

Substituting the value of Δ_I, we have:

$$\frac{\sum_{i=1}^{p} d \cdot length_i(t)}{p} \geq d \cdot length(t)$$

Which can be simplified as:

$$\frac{\sum_{i=1}^{p} length_i(t)}{p} \geq length(t) \qquad \text{Eq. 6.8}$$

Taking, now, condition number 2, we repeat the same analysis:

$$\frac{\sum_{i=1}^{p} \Delta_i}{p} < (d+1) \cdot length(t)$$

$$\frac{\sum_{i=1}^{p} d \cdot length_i(t)}{p} < (d+1) \cdot length(t)$$

Having,

$$\frac{\sum_{i=1}^{p} length_i(t)}{p} < \frac{(d+1)}{d} \cdot length(t) \qquad \text{Eq. 6.9}$$

Finally, we obtain that the two conditions (Eq. 6.8 and Eq. 6.9) can be written as:

$$\frac{\sum_{i=1}^{p} length_i(t)}{p} \geq length(t) > \frac{d}{(d+1)} \cdot \frac{\sum_{i=1}^{p} length_i(t)}{p} \qquad \text{Eq. 6.10}$$

This equation, Eq. 6.10, means that if we want to have a monotonous operation, the length of the terms in the final vocabulary must be less or equal to the arithmetic average of the lengths of the terms used by the different experts (i.e. criteria), and greater than this average length multiplied by a factor related to the increment or decrement applied. Notice, that again we are constrained by an average of the lengths of the terms in the vocabularies of the criteria, however, now we make an arithmetic average while in the Principal Components based ranking we have to perform a weighted average.

After the analysis of the two ranking approaches, we can say that under some conditions, ClusDM is monotonous. In particular, if the length of the terms in the final vocabulary is constrained according to Eq. 6.6 or Eq. 6.10, the property is fulfilled.

6.2 Other Properties

In this section we review other properties studied for aggregation operators: the stability of ClusDM to some modifications in the data values. In particular, we study the behaviour of the method with respect to the negation of all the values in the data matrix, and with respect to the inversion of the preference values of one criterion.

6.2.1 Stability for the negation

An aggregation operator is stable for the negation if the reversal of the scale has no effect on the evaluation.

$$\Theta\left(N(c_1),\ldots,N(c_p)\right)=N\left(\Theta\left(c_1,\ldots,c_p\right)\right)$$

being N the application of a negation operator to all the values in one column of the decision matrix (i.e. to one criterion).

In the numerical case, the negation operator can be the classical one, $N(x_{ij})=1-x_{ij}$, or a strong negation operator of the form $N(x_{ij})=\varphi^{-1}\left(1-\varphi(x_{ij})\right)$.

The rationale of this property is that if we assume that the experts give us values of their non-preference (or distaste), the result should be the opposite to the one obtained in terms of preference.

This property expresses self-duality of Θ, equivalently to the De Morgan laws in fuzzy sets theory [Klir&Yuan,1995].

Stability for the negation of ClusDM

This property can only be applied if the negation of each term is a single term, that is, if it is the classical negation. In this case, the property holds because the relations among the terms will be the same, so the clusters will be the same, and the terms selected will be the ones corresponding to the negation of the ones obtained with the original data. This is proven below.

Instead, if more than one term belong to the negation of another, the substitution of this term by its negation is not possible, since we do not allow to have more than one value in each cell of the decision matrix.

To prove that the terms selected will be the negation of the terms obtained, in case of using the data matrix with the real values, we will study the ranking results.

- PCA: Since all the terms in the vocabulary are equal informative (i.e. have equal interval lengths), we have that the values used to compute the principal components are, 1-x_{ij}, being x_{ij} the numerical value corresponding to the term attached to the i-th alternative for the j-th criterion. As the relationships among the data do not change (they are only inverted), the use of the correlation matrix assures that the eigenvalues and eigenvectors will be the same, since the standardisation of the data will remain the same. Consequently, the z-scores of the cluster prototypes will only change their sign. However, the z-scores of the ideal and nadir alternatives, will not suffer this modification since the negation is not applied to their values.

Therefore, the distance of a z-score to the ideal will be now the distance to this score to the nadir. So, it is as we considered as the reference point 1 the position of the nadir, and as reference point 0 the position of the ideal. With this interchange we will have that all the positions of the clusters will suffer the following modification when they are scaled to the unit interval:

$$z^{N}_{01}(x_i) = 1 - z_{01}(x_i)$$

In the following step, if all the terms are equally informative, the assignment algorithm will select the opposite terms.

- Similarity-Based Ranking: When all the values of the matrix are negated, the objects suffer a translation in the variables space. However, the ideal alternative remains at the same point. So, the clusters obtained with the aggregation method will be the same, but their prototypes will have the negations of the values of the original ones. Now, what we want to prove is that the distance of a cluster prototype in the negated matrix, x_i^N, is equal to 1-z_{01}, where z_{01} is the distance of the same cluster with the original values, x_i.

$$z_{01}\left(x^{N}_{i\,i}\right) = 1 - z_{01}(x_i)$$

Considering the MCD distance, we can write this equation as:

$$\sum_{i=1}^{p} \frac{\left|1 - x^{N}_{i}\right|}{p} = 1 - \sum_{i=1}^{p} \frac{\left|1 - x_i\right|}{p}$$

According to the fact that x_i^N corresponds to the negation of x_i, this expression corresponds to:

$$\sum_{i=1}^{p} \frac{\left|1-\left(1-x_i\right)\right|}{p} = 1 - \sum_{i=1}^{p} \frac{\left|1-x_i\right|}{p}$$

Thus, we have:

$$\sum_{i=1}^{p} \frac{\left|x_i\right|}{p} = 1 - \sum_{i=1}^{p} \frac{\left|1-x_i\right|}{p}$$

which can be rewritten as:

$$\sum_{i=1}^{p} \frac{\left|x_i\right|}{p} + \sum_{i=1}^{p} \frac{\left|1-x_i\right|}{p} = 1$$

As $0 \le x_i \le 1$, we can ignore the absolute value calculation:

$$\sum_{i=1}^{p} \frac{x_i + 1 - x_i}{p} = 1$$

This equation is always true since $\sum_{i=1}^{p} \frac{1}{p}$ is equal to 1. Consequently, we have that the position of the prototype with negated values will be exactly the negation of the position of the prototype with the original values. Therefore, the algorithm for selecting the terms, will select the terms corresponding to the negation of the ones selected in the study with non-negated values.

6.2.2 Stability for the Opposition

In a data matrix with two criteria, if the two criteria have opposite preferences, then the result of the aggregation should be a null degree of preference over the alternatives.

Opposition in ClusDM

This property holds when the semantics of the terms is based on the classical negation function. In this case, if two criteria give completely opposite preferences to all the alternatives in the decision matrix, the result of the aggregation will be a single cluster with the neutral label (the one corresponding to position 0.5 in the interval [0,1]).

What is happening is that the opposite terms compensate each other. So if all the terms in the vocabulary are used, the distances between the objects will be constant and they will be put together in a single cluster with a neutral prototype.

In Figure 26, we can see an example with two criteria with a vocabulary of 5 terms (from best to worst): {a, b, c, d, e}.

	c_1	c_2
a_1	a	e
a_2	b	d
a_3	c	c
a_4	d	b
a_5	e	a

Figure 26. Decision matrix with opposite preferences

The first step in the clustering procedure will calculate the Manhattan distance considering the classical negation function, so we will obtain:

	a_1	a_2	a_3	a_4	a_5
a_1		0.4	0.8	1.2	1.6
a_2			0.4	0.8	1.2
a_3				0.4	0.8
a_4					0.4
a_5					

Figure 27. Dissimilarity matrix for opposite preferences

With this values, we have to build a cluster with objects: a_1, a_2, a_3, a_4 and a_5. They all have a dissimilarity value of 0.4. The prototype of this cluster will have the following values: {0.5,0.5} = {c,c} whose aggregation label will be the neutral one, c.

6.3 Properties with respect to the alternatives

Although our decision making operator is applied to the criteria, it is also interesting to study the results from the point of view of the alternatives, since what we are trying to do is an analysis of the alternatives in order to know which is their individual preference in an overall criterion. For this reason, we will now study the behaviour of the ClusDM methodology according to the values of the alternatives.

We will denote as Φ the application of the ClusDM operator to an alternative. Although we do not mention that the aggregation result of an alternative depends on the value of the other alternatives, it is implicitly taken into account. In fact, to study these properties we assume that the rest of the data matrix does not change. Formally we have,

$$\Phi(a_i) = \Phi(v_{i1}, v_{i2}, ..., v_{ip}) = v_{ir}$$

Considering this new view of the ClusDM operator, in this section we study three properties that are usually fulfilled by aggregation operators: increasingness or monotonicity, positive association and neutrality.

6.3.1 Increasingness

This property is fulfilled when an alternative with a better rating for each criterion evaluates better in the final rating. This is, if there is an alternative a_k such that for all the other alternatives, a_l, satisfying $c_i(a_k) > c_i(a_l)$ for all criteria i=1..p, then we have that $\Phi(a_k) > \Phi(a_l)$.

Increasingness of ClusDM for alternatives:

Let us consider a data matrix where all the alternatives have the same value for all criteria.

When an alternative a_k is better than all the others, it will be also the best one in the result, c_r, because for the idempotency property of ClusDM, the preference value of each alternative in the result, will be the same one that they have in all the criteria.

However, if the values of the alternatives are different according to the criteria, the alternative with best value cannot receive the best term in the result. It will depend on the relationships among the alternatives and the groups produced by the clustering.

Here we have an example of the non-monotonicity for the case of two different columns. Let us consider the case that all criteria have the same vocabulary (from worse to best): {g,f,e,d,c,b,a}. The terms have the same semantics for all criteria, which is given by the following negation function, N:

N(a)={f,g}, N(b)={f}, N(c)={e}, N(d)={d}, N(e)={c}, N(f)={a,b}, N(g)={a}

	c_1	c_2
a_1	a	a
a_2	b	a
a_3	c	a
a_4	d	a
a_5	d	b
a_6	a	d
a_7	f	c
a_8	e	e
a_9	f	f
a_{10}	g	g
a_{11}	d	b
a_{12}	d	b
a_{13}	d	b
a_{14}	d	b

Figure 28. Decision matrix for the example of increasingness of alternatives

The trace of applying the clustering method (centroid clustering with Manhattan distance) is given here:

STEP 1: Build a dissimilarity matrix and put together those alternatives with minimum value.

In this case, alternatives a_5, a_{11}, a_{12}, a_{13} and a_{14} are the components of the first cluster because these alternatives are identical. We will denote this cluster k_1.

STEP 2: Calculate the prototype of the new cluster.

Prototype k_1 = (d,b) = (0.5, 0.72) = a_5 = a_{11} = a_{12}= a_{13} = a_{14}

STEP 3: Build a dissimilarity matrix and put together those alternatives with minimum value.

	a_1	a_2	a_3	a_4	k_1	a_6	a_7	a_8	a_9	a_{10}
a_1	0	0.17	0.28	0.39	0.55	0.39	0.94	1.0	1.33	1.67
a_2		0	**0.11**	0.22	0.39	0.55	0.78	0.83	1.17	1.5
a_3			0	**0.11**	0.28	0.67	0.67	0.72	1.05	1.39
a_4				0	0.17	0.78	0.55	0.61	0.94	1.28
k_1					0	0.61	0.39	0.44	0.78	1.11
a_6						0	0.78	0.61	0.94	1.27
a_7							0	0.39	0.39	0.72
a_8								0	0.33	0.67
a_9									0	0.33
a_{10}										0

Figure 29. Dissimilarity matrix in the 3rd. step

STEP 4: Calculate the prototype of the new cluster.

Prototype k_2 = (0.61, 0.89) = average (a_2, a_3, a_4)

STEP 5: Modify the dissimilarity matrix, including the new cluster and deleting its components.

	a_1	k_2	k_1	a_6	a_7	a_8	a_9	a_{10}
a_1	0	**0.28**	0.55	0.39	0.94	1.0	1.33	1.67
k_2		0	**0.28**	0.67	0.67	0.72	1.06	1.39
k_1			0	0.61	0.39	0.44	0.78	1.11
a_6				0	0.78	0.61	0.94	1.27
a_7					0	0.39	0.39	0.72
a_8						0	0.33	0.67
a_9							0	0.33
a_{10}								0

Figure 30. Dissimilarity matrix in the 5th. step

STEP 6: Build another level of clusters.

We have that the minimum value is also 0.28, which creates another class: k_3 = {a_1, k_2, k_1}.

STEP 7: Calculate the prototype of this class.

Prototype k_3 = (0.58, 0.795) = average (a_1, a_2, a_3, a_4, a_5, a_{11}, a_{12} , a_{13} , a_{14})

We stop the process because we have 6 clusters. Then, to establish a ranking of the clusters, we can use the similarity with respect to the *Ideal* alternative, which in this case is *I* = (a,a) or the Principal Components Analysis (see Table 14).

If we consider that the two criteria are correlated, we can apply the ranking based on the Principal Components Analysis. In this case, the position of the clusters in the ranking is the same that the one obtained with the similarity function (Table 14). We have that the correlation degree between the two criteria is 83%. Moreover, we can see that cluster {a_6} and cluster {a_7} are not properly represented by this ranking, since they have a quality value of 0.207 and 0.593, respectively. Therefore, although a_6 is given as the best alternative, we should not rely on this result, because this alternative has different preference values for the two criteria: criterion 1 says that it is the best (value a) while criterion 2 says that it is normal (value d).

Elements of the class	**Prototype**	**Similarity to the Ideal**	**PCA projection**	**Rank**
{a_1, a_2, a_3, a_4, a_5, a_{11}, a_{12} , a_{13} , a_{14}}	(0.58, 0.795)	0.405	0.66	2
{a_6}	(0.89, 0.5)	0.39	0.79	1
{a_7}	(0.22, 0.61)	0.95	-0.91	3
{a_8}	(0.39, 0.39)	1.0	-1.0	4
{a_9}	(0.22, 0.22)	1.34	-1.97	5
{a_{10}}	(0.05, 0.05)	1.68	-2.94	6

Table 14. Ranking of the alternatives in the increasingness proof

With this example, we can see that ClusDM is not monotonous with respect to the alternative's preference. In the initial data, a_1 was the most preferred (in fact, it is the ideal) and in the result a_6 is considered better than a_1.

However, this situation occurs when, for two alternatives a_k and a_l such that, a_k>a_l, we have that the aggregation makes that a_k become part of a cluster whose centroid is worse than the cluster that a_l belongs to (see Figure 31).

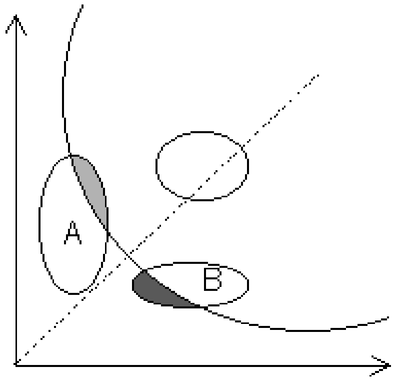

Figure 31. Two clusters with objects that do not fulfil the increasingness property

In this figure, we can see that some elements of cluster A are better that others of cluster B. However, the prototype of A is worse than the one of cluster B.

This situation is more difficult to arise in the case of having qualitative criteria. This is due to the fact that the alternatives can only take a linguistic value that is covering an interval in the numerical domain. In this case, the possibility of generating clusters that produce this effect decreases, since we have to build elliptical clusters, rather than spheripherical ones, which means that we need alternatives with conflicting evaluations (i.e. one criteria says that all the alternatives have the same preference, while the other distinguish quite different preference degrees). It can be seen in the example used to demonstrate that ClusDM in not monotonous.

6.3.2 Positive Association

This property holds when having a set of preference criteria (i.e. experts) whose resulting decision criterion establishes that a_i is preferred to a_j, then making a_i better or a_j worse than before, implies that, in the overall decision criterion, a_i remains preferred to a_j. This can be formalised as follows.

Let $a_i = (v_{i1}, v_{i2}, ..., v_{ip})$ and $a_j = (v_{j1}, v_{j2}, ..., v_{jp})$ such that $\Phi(a_i) > \Phi(a_j)$. Then for all $a'_i > a_i$ (i.e. $v'_{ij} > v_{ij}$ for some *j*) it holds $\Phi(a'_i) > \Phi(a_j)$.

Positive Association in ClusDM:

In Figure 32 we can see a two-variable example of the non fulfilment of this property for the case of ranking according to the PCA results. Note that each dimension on this figure corresponds to a variable and each painted cell represents one or more alternatives (the darker the grey is, the more alternatives with the same value are). In particular, we want to study the behaviour of the x and y alternatives. The alternative y has the values (b,g), which is marked in green colour in the figure. Otherwise, the orange cell corresponds to alternative x, with (c,e) values. According to the first picture, x is preferred to y. Then, if we increase the value of the second criterion of x, obtaining (c,d), this alternative is now closer to another class, whose projection is worse than the one of y. So, in the second picture, y is preferred to x.

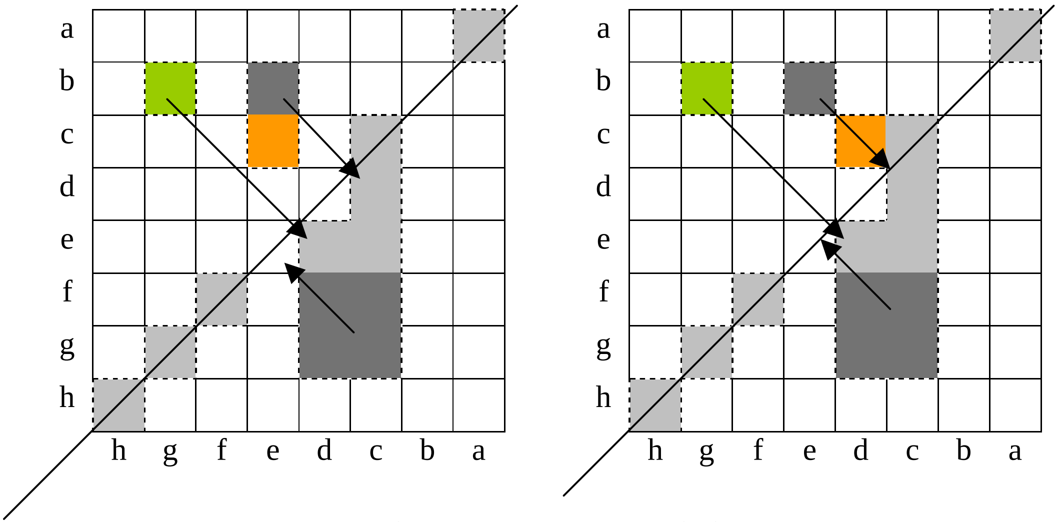

Figure 32. Representation of alternatives that do not fulfil the positive association

As it can be seen in the pictures above, in order to make that an alternative, x, which belongs to a cluster better projected than the one of y, to be moved to another cluster whose projection is worse than the one of y, we need that the cluster that receives the alternative x has an elliptical form, in order to be near the alternative x after the increase of its values but having the gravity centre lower than the one of the cluster of y. Therefore, we believe that it is a non-common case for decision making problems, since it means that alternative with very different values are put together in an elliptical class. With the quality measures we would detect such a cluster with a very low intra-cluster cohesion, which will decrease the confidence on the value attached to it.

In case of using the similarity-based ranking, we need an elliptical cluster whose gravity centre is farther than the one of cluster y (green). So, if an alternative x of the orange cluster receives a better value in the second criterion, it will can move to the big cluster (grey) and then, it will become worse than y. See Figure 33.

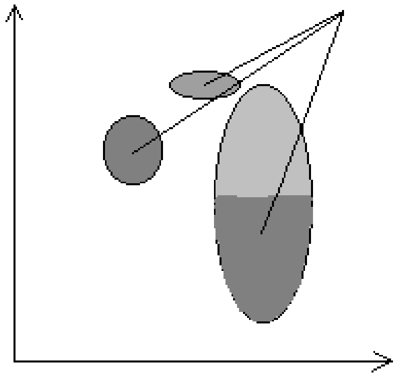

Figure 33. Clusters that do not fulfil the positive association property

This case is similar to the one explained before, in which we have some strange clusters, since one of them is covering a widespread set of alternatives.

In general, it seems reasonable that if an alternative moves to a situation in which there is a large cluster with low intra-cluster cohesion, the new label attached to this alternative is not appropriate, since it is not clear that this cluster can have a unique prototype that determines its value. Therefore, although positive association is not fulfilled, the quality measures that ClusDM uses, probably will show us that the result obtained is not reliable. Thus, although it cannot be proved, in the general case, and specially if criteria are correlated, we will have that the positive association of the alternatives holds.

6.3.3 Neutrality with respect to alternatives

In decision making procedures it is required that any two identical alternatives (such that they have the same value for all the criteria) receive the same preference value in the final ranking.

Having two alternatives $a_i = (v_{i1}, v_{i2}, ..., v_{ip})$ and $a_j = (v_{j1}, v_{j2}, ..., v_{jp})$ so that $v_{ik} = v_{jk}$ for all *k* in 1..*p*, this property can be stated as:

$$\Phi(a_i) = \Phi(a_j)$$

Neutrality in ClusDM

Our method guarantees this property, since the first clusters that are generated in the aggregation process are the ones that put together those indistinguishable objects, because they have a dissimilarity value equal to 0. Therefore, even in the case of cutting the tree at the lowest level, we will have those alternatives in the same class, which means that they will be attached to the same term.