Resúmenes Métodos Multicriterio

Pedro Luis Luque

Construcción de Funciones de Valor o Utilidad

1.3 Métodos de construcción de funciones de valor o utilidad

1.3.1 Métodos de maximales (no borroso).

Partimos de una estructura de preferencia-indiferencia completa.

Consideramos las clases de equivalencia (indiferencia) del conjunto. Si se representa este conjunto de clases de equivalencia a través de un grafo, **éste no puede ser cíclico**.

Un **elemento a_k es maximal para la preferencia estricta** si no existe ningún a_i tal que $a_i > a_k$. Bajo las condiciones de partida, se sabe que existe al menos un elemento maximal.

ALGORITMO: Sea M_N el conjunto de elementos maximales para la preferencia estricta.

Se asigna $u(a_j) = N, \forall a_j \in M_N$. Obsérvese que:

(a) $a_i \sim a_j, \forall a_j \in M_N$.

(b) Si $a_i \in M_N$ y $a_k \sim a_i$ entonces $a_k \in M_N$.

Se eliminan el conjunto de elementos maximales del conjunto X, y volvemos a realizar las operaciones anteriores con los elementos que quedan.

1.3.2 Método construcción con estructura borrosa.

Nota: puede darse el caso de que $a_i \succ a_k$ y $a_k \succ a_i$ de forma borrosa (valoraciones en el intervalo $[0,1]$, función borrosa $\mu(a_i, a_j) \in [0,1]$).

Definiciones: Se definen las funciones de **flujo saliente (DOMINA A)**: $v^+(a_i) = \sum_{l=1}^{m} \mu(a_i, a_l)$, y de **flujo entrante (DOMINADO POR)**: $v^-(a_i) = \sum_{l=1}^{m} \mu(a_l, a_i)$.

Se define la función de **flujo neto** como:

$$v(a_i) = v^+(a_i) - v^-(a_i)$$

La función de valor o utilidad será la dada por el valor del flujo neto.

Homogeneización de Columnas en Tablas de Decisión

Podemos encontrarnos que cada columna de la tabla de decisión venga medida en distintas unidades de medida. En este caso, en primer lugar debemos transformar la tabla de decisión en otra nueva en la que **todas las columnas estén en la misma escala**. Se utilizan varias métodos.

MÉTODO NADIR

EJEMPLO:

```
X = matrix(c(100,1,40,0,60,0,70,1),nrow=4,ncol=2,byrow=TRUE);
rownames(X) = c('a1','a2','a3','a4');
colnames(X) = c('w1','w2');
X
```

```
##     w1 w2
## a1 100  1
## a2  40  0
## a3  60  0
## a4  70  1
```

```
(vcol.max = apply(X,2,max))
```

```
##  w1  w2
## 100   1
```

```
(vcol.min = apply(X,2,min))
```

```
## w1 w2
## 40  0
```

```
(m.vcol.min = matrix(vcol.min,4,2,byrow=T))
```

```
##      [,1] [,2]
## [1,]   40    0
## [2,]   40    0
## [3,]   40    0
## [4,]   40    0
```

```
(m.vcol.max = matrix(vcol.max,4,2,byrow=T))
```

```
##      [,1] [,2]
## [1,]  100    1
## [2,]  100    1
## [3,]  100    1
## [4,]  100    1
```

```
Nueva.X = (X-m.vcol.min)/(m.vcol.max - m.vcol.min)
Nueva.X
```

```
##           w1 w2
## a1 1.0000000  1
## a2 0.0000000  0
## a3 0.3333333  0
## a4 0.5000000  1
```

MÉTODO PROMETHEE

Para cada columna construimos una matriz en la que se represente una relación binaria borrosa entre las alternativas.

```
matriz.relacion.borrosa = function(vcolumna,delta.min,delta.max) {
  X.rel.bor = matrix(NA,length(vcolumna),length(vcolumna));
  for (i in 1:length(vcolumna)) {
    for (j in 1:length(vcolumna)) {
      X.rel.bor[i,j] = vcolumna[i] - vcolumna[j];
      if (X.rel.bor[i,j] <= delta.min) {
        X.rel.bor[i,j] = 0;
      } else if (X.rel.bor[i,j] >= delta.max) {
        X.rel.bor[i,j] = 1;
      } else {
        X.rel.bor[i,j] = (X.rel.bor[i,j]-delta.min)/(delta.max - delta.min);
      }
    }
  }
  return(X.rel.bor)
}

calculo.flujo.neto = function(X.matriz) {
 flujo.saliente = rowSums(X.matriz);
 flujo.entrante = colSums(X.matriz);
 flujo.neto = flujo.saliente-flujo.entrante;
 return(flujo.neto)
}
```

```
X
```

```
##       w1 w2
## a1   100  1
## a2    40  0
## a3    60  0
## a4    70  1
```

Para la columna 1:

```
X.rel.bor1 = matriz.relacion.borrosa(X[,1],0,20);
X.rel.bor1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    1  1.0    1
## [2,]    0    0  0.0    0
## [3,]    0    1  0.0    0
## [4,]    0    1  0.5    0
```

```
v.flujo.neto1 = calculo.flujo.neto(X.rel.bor1);
v.flujo.neto1
```

```
## [1]  3.0 -3.0 -0.5  0.5
```

ASÍ PARA EL RESTO DE COLUMNAS

```
##      v.flujo.neto1 v.flujo.neto2
## [1,]           3.0             2
## [2,]          -3.0            -2
## [3,]          -0.5            -2
## [4,]           0.5             2
```

NOTA

LOS DOS MÉTODOS ANTERIORES SUPONEN QUE TODAS LAS COLUMNAS SE REFIEREN A CRITERIOS DE BENEFICIOS O GANANCIAS (MAYOR VALOR MEJOR)

SI ALGUNA COLUMNA FUERA UN COSTO, SE DEBE CAMBIAR EL SIGNO A TODOS SUS ELEMENTOS