CADERNOS DO IME – Série Estatística
Universidade do Estado do Rio de Janeiro - UERJ
ISSN impresso 1413-9022 / ISSN on-line 2317-4536 - v.47, p. 1 - 16, 2019
DOI: 10.12957/cadest.2019.51611

ALGUMAS TÉCNICAS ESTATÍSTICAS DA COMPOSIÇÃO PROBABILÍSTICA DE PREFERÊNCIAS

Annibal Parracho Sant'Anna
Universidade Federal Fluminense - UFF
annibal.parracho@gmail.com

Resumo

Este artigo reúne um conjunto de procedimentos estatísticos empregados em diferentes estágios na Composição Probabilística de Preferências. Estes procedimentos incluem a estimação de probabilidades de preferência para a formação de escores de preferência com base na distribuição empírica acumulada conjunta, em substituição a médias ponderadas de avaliações nas diferentes dimensões ou outras formas de composição quaisquer. Em outra aplicação, é apresentado um procedimento de estimação de parâmetros de distribuição beta garantindo a unimodalidade e empregando o tamanho da amostra na estimação da dispersão. Outro procedimento é apresentado em que, para um conjunto qualquer de critérios, é empregado o máximo da amostra no processo de determinação de capacidades de Choquet e da importância de critérios.

Palavras–chave: *Composição Probabilística de Preferências; Distribuição Empírica; Distribuição Beta; Valor de Shapley; Capacidade de Choquet.*

Recebido em 09/06/2020; Aceito em 20/07/2020

1. Introdução

Na área do Apoio a Decisão Multicritério, é oferecida ao tomador de decisão farta variedade de técnicas para lidar com problemas cuja modelagem estatística é complicada principalmente pela multidimensionalidade. Esta área recorre a uma simplificação dos modelos ao substituir a modelagem estatística das variáveis representadas em cada dimensão do problema, em geral variáveis contínuas, pela modelagem de um conjunto discreto de situações, identificadas como alternativas de possíveis soluções para o problema de escolha da melhor alternativa.

Um conjunto de técnicas que leva em conta a imprecisão das medidas dos valores que constituem essas alternativas é reunido na Composição Probabilística de Preferências (CPP) (SANT'ANNA, 2015). A CPP obedece a um princípio de concentração das preferências que faz ampliar a distância entre as alternativas de maior preferência, de modo a aumentar a importância das avaliações com mais alta preferência. Isto é feito através da transformação probabilística, que substitui a avaliação numérica inicial resultante da aplicação de cada critério separadamente pela probabilidade de ser a alternativa de maior preferência segundo tal critério. Esse princípio pode ser aplicado diretamente também na atribuição de importância aos critérios, mas é difícil encontrar um hipercritério que gere comparação inicial adequada dos critérios. Para esta situação, uma saída é aplicar o princípio da concentração das preferências (SANT'ANNA & SANT'ANNA, 2019) para derivar a importância dos critérios (ou dos conjuntos de critérios) da sua capacidade de indicar uma alternativa como a de maior preferência.

São discutidos aqui três pontos em que métodos estatísticos originais são aplicados na CPP. O primeiro, de aplicação mais geral, é o da resolução direta do problema de escolha, com a determinação da alternativa de maior preferência sem recorrer seja à modelagem da distribuição dos critérios, seja mesmo à determinação de qualquer uma regra de composição dos critérios. O método é baseado na função de distribuição acumulada da distribuição empírica conjunta das avaliações pelos diversos critérios (SANT'ANNA *et al.*, 2020). A CPP fornece, então, uma regra de decisão completamente não supervisionada (KOJADINOVIC, 2008). Como a CPP não interfere na escolha dos critérios nem na sua aplicação, constitui, com o uso deste método, uma forma de composição de preferências completamente neutra.

A segunda diz respeito à transformação probabilística, baseada na determinação dos parâmetros de uma distribuição de probabilidades para a avaliação de cada alternativa segundo cada critério. Aqui, também, a mínima interferência do modelo é buscada. A informação disponível, constituída pelo único valor observado, ou por um conjunto de valores fornecidos por um conjunto de especialistas chamados a aplicar o critério, é usada junto com a escolha da forma da distribuição, que deve ser a mais simples entre as que satisfaçam as especificações do problema pelo tomador de decisão. Apresentamos aqui a solução para este problema a partir da escolha de uma distribuição triangular ou, para maior generalidade, de uma distribuição beta. Para este caso, apresenta-se um desenvolvimento para a estimação dos parâmetros que explora a diminuição esperada da variância da distribuição com o aumento do tamanho da amostra (SANT'ANNA *et al.*, 2016).

A terceira aplicação diz respeito à geração de informação sobre a importância dos critérios quando se admite a interação ente eles. Apresentamos para este problema um novo uso do máximo. A importância do critério é dada pelo seu valor de Shapley (SHAPLEY, 1953) ou, do mesmo modo, por outra medida de importância (BANZHAF 1966; HOLLER & PACKEL, 1983; JOHNSTON, 1978), em relação à capacidade empregada na integral de Choquet usada na composição dos critérios. Fundamental para isto é a determinação da capacidade a partir das avaliações de preferência observadas. Isto é feito, com o uso do máximo, recorrendo-se apenas à aplicação da transformação probabilística e do princípio da concentração das preferências.

2. Probabilidades de Maximização na Distribuição Conjunta Empírica

Devido ao grande número de comparações realizadas, a influência da precisão da forma de distribuição sobre as probabilidades de maximização de preferência obtidas pela CPP é limitada. Usar a distribuição empírica torna-se, então, um caminho mais simples para a composição de preferências.

A função de distribuição acumulada obtida a partir de uma amostra aleatória de observações de avaliações de uma alternativa qualquer segundo um dado critério é obtida calculando para cada avaliação dessa alternativa o resultado da divisão, pelo número total de comparações, da soma de duas parcelas: o número de outras alternativas com avaliações menores mais o inverso do número de alternativas com avaliação igual. A soma das probabilidades de maximização assim obtida é exatamente 1.

Com um grande número de especialistas aplicando independentemente cada critério, distribuições discretas muito detalhadas são obtidas dessa forma, mas mesmo com um único avaliador, uma distribuição adequada é obtida para cada alternativa. Esta distribuição é a representação mais precisa da posição relativa da alternativa no conjunto de dados observado. Primeiro, o princípio de estimar pela proporção do número de casos favoráveis contados pode ser usado para determinar a distribuição de preferências entre as alternativas por cada critério, isoladamente. Permanece, neste caso, em aberto a regra de composição dos critérios.

Formalmente a computação das probabilidades de preferência pode ser detalhada da seguinte forma, para Crit(j,i) denotando a probabilidade de que a i-ésima alternativa maximize a preferência segundo o j-ésimo critério e A(k,j,i) a avaliação da alternativa segundo o j-ésimo critério pelo k-ésimo especialista.

Seja D1 o indicador de ser a preferida. Isto, é, seja

D1(k,j,i)=1 se A(k,j,i)<A(k,j,m) para nenhum m, m variando livremente ao longo do conjunto de alternativas, e

D1(k,j,i)=0 caso contrário.

Quando D1(k,j,i)=1, seja D2(k,j,i) o número de alternativas empatadas com a alternativa i na avaliação pelo especialista k segundo o critério j.

Então Crit(j,i) é dado pela média ao longo do conjunto de todos os especialistas da razão D1(k,j,i)/D2(k,j,i).

Mais interessante é que, para evitar a etapa da composição dos critérios, podemos derivar diretamente para cada alternativa sua função de distribuição acumulada multidimensional (as dimensões são os critérios, que são assim implicitamente ajuntados). Esta distribuição conjunta é a distribuição acumulada empírica dos vetores de avaliação da alternativa pelos múltiplos critérios.

A probabilidade de maximizar a preferência global é, então, determinada simplesmente pela contagem da proporção de pares que apresentam, para a alternativa sendo avaliada, avaliação de preferência superior à de cada uma das demais alternativas, na comparação uma a uma entre as avalições, segundo cada critério e cada especialista. Cabe notar que esta abordagem atribui a todos os critérios e a todos os especialistas igual importância, ainda que ponderações possam ser aqui inseridas, se houver motivo para trazer de volta as complicações da avaliação dos critérios e dos especialistas.

O passo central do procedimento de cálculo desta probabilidade de ser a alternativa de maior preferência global, P, pode ser descrito precisamente da seguinte forma.

Sejam T o número total de alternativas, K o número total de especialistas e J o número total de critérios. Para levar em conta a possibilidade de empates, seja B(i,j) o número total de pares de alternativas (i,m) para o qual, para algum par de especialistas (k,q), A(k,j,i)>A(q,j,m) mais metade do número de pares (k,q) para os quais A(k,j,i)=A(q,j,m). P(i) é dado pela média dos B(ij). Isto é,

$P(i) = \sum_j B(i,j)/\{T(T-1]/2\}/J/K^2$.

O procedimento abaixo destina-se ao cálculo no R (R CORE TEAM, 2017) de probabilidades de maximização de risco em um conjunto de dados de uma análise de Severidade, Ocorrência e Detecção da FMEA (US MILITARY, 1949).

```
A1 <- as.matrix(read.table("Severidade.txt")) # matriz de avaliações segundo um critério
A2 <- as.matrix(read.table("Ocorrencia.txt")) # matriz de avaliações segundo um critério
A3 <- as.matrix(read.table("Detectabilidade.txt")) # matriz de avaliações segundo um critério
I<-20 #número de alternativas
J<-3 #número de critérios
K<-30 #número de especialistas
N<-2 #número de estágios na determinação da classe
A<-rep(0,times=K*J*I) #todas as avaliações
dim(A)<-c(K,J,I)
for (i in 1:I)
{for (k in 1:K)
{A[k,1,i]<-A1[k,i]
A[k,2,i]<-A2[k,i]
A[k,3,i]<-A3[k,i]
}}
Score<-rep(0, times=I) #saída
Bi<- rep(0,times=I*J*I*N)
dim(Bi)<-c(I,J,I,N)
for (i in seq.int(from=1, to=I))
{for (m in seq.int(from=1, to=I))
{for (j in seq.int(from=1, to =J))
{for (k in seq.int(from=1, to=K))
{for( q in seq.int(from=1, to=K))
{if (A[k,j,i]>A[q,j,m]) {Bi[m,j,i,N]<-Bi[m,j,i,N]+1}
else if(A[k,j,i]==A[q,j,m]) {Bi[m,j,i,N]<-Bi[m,j,i,N]+1/2}
}}}}
Score[i]<-sum(Bi[,,i,N])/((I+1)*I/2)/J/K/K
}
Score
write.table(Score, file="scores.txt")
```

3. Estimação de Parâmetros da Distribuição de Preferência por cada Critério

A CPP considera que cada avaliação inicial é uma realização de uma variável aleatória e, se outras medidas fossem tomadas para o mesmo objeto de avaliação, valores ao seu redor seriam observados. As medidas iniciais são então usadas para determinar a distribuição de probabilidades dessa variável aleatória. Uma vez que as medidas são obtidas pelas avaliações de vários especialistas em cada critério, os conjuntos de medidas para uma mesma alternativa podem ser tratados como amostras dessa distribuição e podem ser utilizados para estimar não apenas o parâmetro de localização, mas também os outros parâmetros que determinam a distribuição.

Uma abordagem simples para essa inferência no caso de uma avaliação única y_{ij} da alternativa i-ésimo pelo critério j-ésimo consiste em usar essa avaliação como moda de uma distribuição triangular com extremos, A_j e B_j, fixos por critério. Esses extremos podem ser os extremos observados no conjunto de todas as avaliações segundo o critério, ou valores mais afastados levando em conta expectativas de possíveis outros valores. Na prática, para facilitar comparações esses extremos são tornados iguais para todas as avaliações, por meio de simples padronização. O extremo inferior é zero, o extremo superior é 1 e a moda y_{ij} é substituída por $(y_{ij} - A_j)/(B_j - A_j)$. Embora simples, essa distribuição propicia uma dispersão assimétrica em torno do valor observado.

Esta observação y_{ij} pode ser a avaliação única da alternativa pelo critério, mas, pode ser também uma média, moda ou mediana amostral, no caso de se dispor de avaliações de vários especialistas aplicando cada critério. Neste caso, se o número de especialista for suficientemente grande, a amostra pode ser usada para estimar outros parâmetros além do parâmetro de locação.

Para um número de especialistas K maior que 3, uma modelagem de maior generalidade é obtida com o uso da distribuição beta.

A densidade da distribuição beta é:

$$f(y)=[(y-L)/(U-L)]^{\alpha-1}[(U-y)/(U-L)]^{\beta-1}/\mathrm{Beta}(\alpha,\beta)$$

para y variando entre L e U e Beta denotando a função Beta de parâmetros positivos α e β.

A média desta distribuição é $L+\mu(U-L)$ para $\mu=\alpha/(\alpha+\beta)$. A variância decresce com a soma $\alpha + \beta$. Assim, pode-se modelar o ganho de precisão com o aumento do tamanho da amostra fixando $\alpha+\beta = K$. Isto, associado à relação de $\alpha/(\alpha+\beta)$ com a média amostral, leva a estimar os parâmetros α_{ij} e β_{ij} que determinam a distribuição da avaliação da alternativa i-ésima pelo critério i-ésimo, a partir do vetor de avaliações $(y_{ij1}, \dots, y_{ijk})$ pelos K especialistas por KY_{ij} e $K(1-Y_{ij})$ para $Y_{ij}= (\Sigma_k y_{ijk}/K-A_j)/(B_j-A_j)$.

O procedimento abaixo calcula no R as probabilidades de maximização com a distribuição beta para um conjunto de dados de avaliações da FMEA. Outras alternativas de computação estão disponíveis no pacote CPP (GAVIÃO *et al.*, 2018) do R.

```
A <- as.matrix(read.table("dadosfmea6.txt")) #avaliações iniciais
N<-A
Alfa<-A
Beta<-A
ASAI<-A
 for (j in 1:ncol(A))
 { for (i in 1:nrow (A))
  {
N[i,j]<- 7 #número de especialistas constante igual a 7
Alfa[i,j] <-  N[i,j]*(A[i,j]/6)
Beta[i,j] <- 7*(1-(A[i,j]/6))
}}
for (j in 1:ncol(A))
 { for (i in 1:nrow (A))
  {
 ASAI[i,j] <- (integrate(Vectorize(function(x){prod(pbeta(x,Alfa[,j][-i],Beta[,j][-
i]))*dbeta(x,Alfa[,j][[i]],Beta[,j][[i]])}),0,1))$value
}}
print(as.data.frame(Alfa))
print(as.data.frame(Beta))
print(as.data.frame(ASAI))
write.table(ASAI, file="saidadedadosbeta.txt")#probabilidades de maximizar
Score<-A[1,]
for (i in 1:nrow (A))
 {Score[i]<-prod(ASAI[i,])
}
print(as.data.frame(Score))
write.table(Score, file="scorebeta.txt")#probabilidades de maximizar
```

4. Importância dos Critérios com Interação

A ideia de destacar as manifestações de maior preferência é a ideia básica da CPP. A transformação em probabilidade de ser a alternativa de maior preferência é a principal aplicação dessa ideia. Mas, a mesma ideia de privilegiar as maiores preferências pode ser aplicada também na determinação da regra de composição dos critérios. Por exemplo, a composição otimista, que usa como escore a probabilidade de ser a melhor alternativa segundo algum critério – em vez de por todos, é uma forma de destacar os valores maiores possíveis.

Outra forma de aplicação dessa ideia é assumir máxima dependência entre os critérios. Ela se combina com aquela acima referida ao se calcular a preferência segundo algum critério pelo máximo da preferência segundo cada um deles. De fato, ser a melhor segundo cada um é ser a melhor segundo a união e a probabilidade da união é, no caso de máxima dependência, igual à probabilidade de um deles, o de maior probabilidade.

O princípio de obter a máxima informação sobre as preferências na distância entre as manifestações de maior preferência está presente, também, em um procedimento de determinação de capacidades de Choquet (CHOQUET, 1953) pela CPP, usado para se determina a importância dos critérios, no caso de interação e ausência de informação inicial sobre pesos. Esta seção apresenta esse procedimento.

A importância relativa de cada critério j para os resultados da composição segundo uma capacidade μ é dada, entre outros índices, pelo seu valor de Shapley (Shapley, 1953), denotado por $S_K(\{j\})$.

O valor de Shapley de um subconjunto unitário dos critérios, {c}, para uma capacidade μ é dado pela expressão

$$\text{Shapley}(\mu,c)=\sum_{K\subset C\backslash\{c\}}((\#(C\backslash\{c\}\backslash K))!(\#(K))!/(\#(C))!)(\mu(K\cup\{c\})-\mu(K)).$$

Capacidades de Choquet em um conjunto S são, em linguagem comum para suscitar uma intuição, medidas não aditivas que expressam, para cada subconjunto de S, a importância subjetiva associada a esse conjunto. A integral de Choquet é uma forma de agregação usada em substituição à média ponderada com pesos associados aos elementos de S, quando se supõe que interação entre os critérios invalida o uso da adição compensatória. Para qualquer função $x= (x_1,..., x_m)$, de domínio $S= \{1, \dots , m\}$ e valores em R^+, a integral de Choquet de x em relação à capacidade μ em S (no nosso caso, m é número de critérios) associa a x o real não negativo

$$C_\mu(x) = \sum_{j=1}^{m}(x\tau(j) - x\tau(j-1))\mu(\{\tau(j),\dots,\tau(m)\}),$$

para τ denotando a permutação de S tal que $x\tau(1)\leq x\tau(2)\leq\ldots\leq x\tau(m-1)\leq x\tau(m)$ e $x\tau(0)=0$.

Para $x: S\rightarrow R^+$ e μ uma capacidade em S, a integral de Choquet de x em relação a μ satisfaz

$$C\mu(x)=\sum_{j=1}^{m} x(\tau(j))[\mu(Z\tau(j))-\mu(Z\tau(j+1))]$$

para $Z\tau(j) = \{\tau(j), \dots , \tau(m)\}$, para todo j de 1 a m, e $Z\tau(m+1)=\phi$.

Para chegar às capacidades e aos valores de Shapley aplicando o princípio da concentração das preferências, propõe-se, primeiro, computar, para cada subconjunto de critérios, o máximo, ao longo das alternativas, das probabilidades de maximização de preferência de acordo com algum dos critérios do subconjunto. As capacidades serão proporcionais a esse vetor de máximos. Seu valor será obtido rescalonando-o de modo que a capacidade 1 seja atribuída ao conjunto de todos os critérios.

Formalmente, o algoritmo de estimação da capacidade do subconjunto de s critérios $\{C_1, \dots , C_s\}$, terá os dois passos seguintes. Primeiro, compute-se, para T o número de alternativas e $M_{Cj}(A_i)$ a probabilidade de a alternativa A_i ser a preferida pelo critério C_j,

$$M(\{C_1, \ldots , C_s\}) = \max_{i\in\{1, \ldots , T\} j\in\{1, \ldots , s\}}(M_{Cj}(Ai)),$$

A capacidade é obtida com a padronização final, que consiste em dividir pelo maior valor:

$$\mu \{C_1, .., C_s\}) = M(\{C_1, .., C_s\})/M(S).$$

O cálculo dos valores de Shapley, assim como da capacidade, pode empregar o pacote kappalab (GRABISCH *et al*, 2008) do R. O procedimento abaixo realiza esse cálculo, para o caso de nove critérios. Nele, o arquivo de entrada probmax.txt é formado pelas probabilidades de maximização de preferências por um conjunto de alternativas segundo nove critérios.

```
#probabilistic entrance
A <- as.matrix(read.table("probmax.txt"))
#capacity generation
B<-rep(2,times=(2**(ncol(A)-1)))
for (j in 1:ncol(A))
B[j]<-max(A[,j])
for (j in (ncol(A)+1):(2*ncol(A)-1))
B[j]<-max(A[,1],A[,j-ncol(A)+1])
for (j in (2*ncol(A)):(3*ncol(A)-3))
B[j]<-max(A[,2],A[,j-2*ncol(A)+3])
for (j in (3*ncol(A)-2):(4*ncol(A)-6))
B[j]<-max(A[,3],A[,j-(3*ncol(A)-6)])
for (j in (4*ncol(A)-5):(4*ncol(A)-1))
B[j]<-max(A[,4],A[,j-(3*ncol(A)-1)])
for (j in (4*ncol(A)):(5*ncol(A)-6))
B[j]<-max(A[,5],A[,j-(4*ncol(A)-6)])
for (j in (5*ncol(A)-5):(5*ncol(A)-3))
B[j]<-max(A[,6],A[,j-(4*ncol(A)-3)])
for (j in (5*ncol(A)-2):(5*ncol(A)-1))
B[j]<-max(A[,7],A[,j-(4*ncol(A)-1)])
B[5*ncol(A)]<-max(A[,8],A[,9])
for (j in (5*ncol(A)+1):(5*ncol(A)+7))
B[j]<-max(A[,1],A[,2],A[,j-(4*ncol(A)+7)])
for (j in (5*ncol(A)+8):(6*ncol(A)+4))
B[j]<-max(A[,1],A[,3],A[,j-(5*ncol(A)+4)])
for (j in (6*ncol(A)+5):(7*ncol(A)))
B[j]<-max(A[,1],A[,4],A[,j-6*ncol(A)])
for (j in (7*ncol(A)+1):(7*ncol(A)+4))
B[j]<-max(A[,1],A[,5],A[,j-(6*ncol(A)+4)])
for (j in (7*ncol(A)+5):(7*ncol(A)+7))
B[j]<-max(A[,1],A[,6],A[,j-(6*ncol(A)+7)])
for (j in (7*ncol(A)+8):(8*ncol(A)))
B[j]<-max(A[,1],A[,7],A[,j-7*ncol(A)])
B[8*ncol(A)+1]<-max(A[,1],A[,8],A[,9])
for (j in (8*ncol(A)+2):(8*ncol(A)+7))
B[j]<-max(A[,2],A[,3],A[,j-(7*ncol(A)+7)])
for (j in (8*ncol(A)+8):(9*ncol(A)+3))
B[j]<-max(A[,2],A[,4],A[,j-(8*ncol(A)+3)])
for (j in (9*ncol(A)+4):(9*ncol(A)+7))
B[j]<-max(A[,2],A[,5],A[,j-(8*ncol(A)+7)])
for (j in (9*ncol(A)+8):(10*ncol(A)+1))
B[j]<-max(A[,2],A[,6],A[,j-(9*ncol(A)+1)])
for (j in (10*ncol(A)+2):(10*ncol(A)+3))
B[j]<-max(A[,2],A[,7],A[,j-(9*ncol(A)+3)])
B[10*ncol(A)+4]<-max(A[,2],A[,8],A[,9])
for (j in (10*ncol(A)+5):(11*ncol(A)))
B[j]<-max(A[,3],A[,4],A[,j-10*ncol(A)])
```

```
for (j in (11*ncol(A)+1):(11*ncol(A)+4))
B[j]<-max(A[,3],A[,5],A[,j-(10*ncol(A)+4)])
for (j in (11*ncol(A)+5):(11*ncol(A)+7))
B[j]<-max(A[,3],A[,6],A[,j-(10*ncol(A)+7)])
for (j in (11*ncol(A)+8):(12*ncol(A)))
B[j]<-max(A[,3],A[,7],A[,j-11*ncol(A)])
B[12*ncol(A)+1]<-max(A[,3],A[,8],A[,9])
for (j in (12*ncol(A)+2):(12*ncol(A)+5))
B[j]<-max(A[,4],A[,5],A[,j-(11*ncol(A)+5)])
for (j in (12*ncol(A)+6):(12*ncol(A)+8))
B[j]<-max(A[,4],A[,6],A[,j-(11*ncol(A)+8)])
for (j in (13*ncol(A)):(13*ncol(A)+1))
B[j]<-max(A[,4],A[,7],A[,j-(12*ncol(A)+1)])
B[13*ncol(A)+2]<-max(A[,4],A[,8],A[,9])
for (j in (13*ncol(A)+3):(13*ncol(A)+5))
B[j]<-max(A[,5],A[,6],A[,j-(12*ncol(A)+5)])
for (j in (13*ncol(A)+6):(13*ncol(A)+7))
B[j]<-max(A[,5],A[,7],A[,j-(12*ncol(A)+7)])
B[13*ncol(A)+8]<-max(A[,5],A[,8],A[,9])
for (j in (14*ncol(A)):(14*ncol(A)+1))
B[j]<-max(A[,6],A[,7],A[,j-(13*ncol(A)+1)])
B[14*ncol(A)+2]<-max(A[,6],A[,8],A[,9])
B[14*ncol(A)+3]<-max(A[,7],A[,8],A[,9])
for (j in (14*ncol(A)+4):(15*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,3],A[,j-(14*ncol(A))])
for (j in (15*ncol(A)+1):(15*ncol(A)+5))
B[j]<-max(A[,1],A[,2],A[,4],A[,j-(14*ncol(A)+5)])
for (j in (15*ncol(A)+6):(16*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,5],A[,j-(15*ncol(A))])
for (j in (16*ncol(A)+1):(16*ncol(A)+3))
B[j]<-max(A[,1],A[,2],A[,6],A[,j-(15*ncol(A)+3)])
for (j in (16*ncol(A)+4):(16*ncol(A)+5))
B[j]<-max(A[,1],A[,2],A[,7],A[,j-(15*ncol(A)+5)])
B[16*ncol(A)+6]<-max(A[,1],A[,2],A[,8],A[,9])
for (j in (16*ncol(A)+7):(17*ncol(A)+2))
B[j]<-max(A[,1],A[,3],A[,4],A[,j-(16*ncol(A)+2)])
for (j in (17*ncol(A)+3):(17*ncol(A)+6))
B[j]<-max(A[,1],A[,3],A[,5],A[,j-(16*ncol(A)+6)])
for (j in (17*ncol(A)+7):(18*ncol(A)))
B[j]<-max(A[,1],A[,3],A[,6],A[,j-(17*ncol(A))])
for (j in (18*ncol(A)+1):(18*ncol(A)+2))
B[j]<-max(A[,1],A[,3],A[,7],A[,j-(17*ncol(A)+2)])
B[18*ncol(A)+3]<-max(A[,1],A[,3],A[,8],A[,9])
for (j in (18*ncol(A)+4):(18*ncol(A)+7))
B[j]<-max(A[,1],A[,4],A[,5],A[,j-(17*ncol(A)+7)])
for (j in (18*ncol(A)+8):(19*ncol(A)+1))
B[j]<-max(A[,1],A[,4],A[,6],A[,j-(18*ncol(A)+1)])
for (j in (19*ncol(A)+2):(19*ncol(A)+3))
B[j]<-max(A[,1],A[,4],A[,7],A[,j-(18*ncol(A)+3)])
B[19*ncol(A)+4]<-max(A[,1],A[,4],A[,8],A[,9])
for (j in (19*ncol(A)+5):(19*ncol(A)+7))
B[j]<-max(A[,1],A[,5],A[,6],A[,j-(18*ncol(A)+7)])
for (j in (19*ncol(A)+8):(20*ncol(A)))
B[j]<-max(A[,1],A[,5],A[,7],A[,j-(19*ncol(A))])
B[20*ncol(A)+1]<-max(A[,1],A[,5],A[,8],A[,9])
for (j in (20*ncol(A)+2):(20*ncol(A)+3))
B[j]<-max(A[,1],A[,6],A[,7],A[,j-(19*ncol(A)+3)])
B[20*ncol(A)+4]<-max(A[,1],A[,6],A[,8],A[,9])
B[20*ncol(A)+5]<-max(A[,1],A[,7],A[,8],A[,9])
for (j in (20*ncol(A)+6):(21*ncol(A)+1))
B[j]<-max(A[,2],A[,3],A[,4],A[,j-(20*ncol(A)+1)])
for (j in (21*ncol(A)+2):(21*ncol(A)+5))
B[j]<-max(A[,2],A[,3],A[,5],A[,j-(20*ncol(A)+5)])
for (j in (21*ncol(A)+6):(21*ncol(A)+8))
B[j]<-max(A[,2],A[,3],A[,6],A[,j-(20*ncol(A)+8)])
for (j in (22*ncol(A)):(22*ncol(A)+1))
B[j]<-max(A[,2],A[,3],A[,7],A[,j-(21*ncol(A)+1)])
B[22*ncol(A)+2]<-max(A[,2],A[,3],A[,8],A[,9])
for (j in (22*ncol(A)+3):(22*ncol(A)+6))
B[j]<-max(A[,2],A[,4],A[,5],A[,j-(21*ncol(A)+6)])
for (j in (22*ncol(A)+7):(23*ncol(A)))
B[j]<-max(A[,2],A[,4],A[,6],A[,j-(22*ncol(A))])
for (j in (23*ncol(A)+1):(23*ncol(A)+2))
```

```
B[j]<-max(A[,2],A[,4],A[,7],A[,j-(22*ncol(A)+2)])
B[23*ncol(A)+3]<-max(A[,2],A[,4],A[,8],A[,9])
for (j in (23*ncol(A)+4):(23*ncol(A)+6))
B[j]<-max(A[,2],A[,5],A[,6],A[,j-(22*ncol(A)+6)])
for (j in (23*ncol(A)+7):(23*ncol(A)+8))
B[j]<-max(A[,2],A[,5],A[,7],A[,j-(22*ncol(A)+8)])
B[24*ncol(A)]<-max(A[,2],A[,5],A[,8],A[,9])
for (j in (24*ncol(A)+1):(24*ncol(A)+2))
B[j]<-max(A[,2],A[,6],A[,7],A[,j-(23*ncol(A)+2)])
B[24*ncol(A)+3]<-max(A[,2],A[,6],A[,8],A[,9])
B[24*ncol(A)+4]<-max(A[,2],A[,7],A[,8],A[,9])
for (j in (24*ncol(A)+5):(24*ncol(A)+8))
B[j]<-max(A[,3],A[,4],A[,5],A[,j-(23*ncol(A)+8)])
for (j in (25*ncol(A)):(25*ncol(A)+2))
B[j]<-max(A[,3],A[,4],A[,6],A[,j-(24*ncol(A)+2)])
for (j in (25*ncol(A)+3):(25*ncol(A)+4))
B[j]<-max(A[,3],A[,4],A[7],A[,j-(24*ncol(A)+4)])
B[25*ncol(A)+5]<-max(A[,3],A[,4],A[,8],A[,9])
for (j in (25*ncol(A)+6):(25*ncol(A)+8))
B[j]<-max(A[,3],A[,5],A[6],A[,j-(24*ncol(A)+8)])
for (j in (26*ncol(A)):(26*ncol(A)+1))
B[j]<-max(A[,3],A[,5],A[,7],A[,j-(25*ncol(A)+1)])
B[26*ncol(A)+2]<-max(A[,3],A[,5],A[,8],A[,9])
for (j in (26*ncol(A)+3):(26*ncol(A)+4))
B[j]<-max(A[,3],A[,6],A[,7],A[,j-(25*ncol(A)+4)])
B[26*ncol(A)+5]<-max(A[,3],A[,6],A[,8],A[,9])
B[26*ncol(A)+6]<-max(A[,3],A[,7],A[,8],A[,9])
for (j in (26*ncol(A)+7):(27*ncol(A)))
B[j]<-max(A[,4],A[,5],A[,6],A[,j-(26*ncol(A))])
for (j in (27*ncol(A)+1):(27*ncol(A)+2))
B[j]<-max(A[,4],A[,5],A[,7],A[,j-(26*ncol(A)+2)])
B[27*ncol(A)+3]<-max(A[,4],A[,5],A[,8],A[,9])
for (j in (27*ncol(A)+4):(27*ncol(A)+5))
B[j]<-max(A[,4],A[,6],A[7],A[,j-(26*ncol(A)+5)])
B[27*ncol(A)+6]<-max(A[,4],A[,6],A[,8],A[,9])
B[27*ncol(A)+7]<-max(A[,4],A[,7],A[,8],A[,9])
for (j in (27*ncol(A)+8):(28*ncol(A)))
B[j]<-max(A[,5],A[,6],A[7],A[,j-(27*ncol(A))])
B[28*ncol(A)+1]<-max(A[,5],A[,6],A[,8],A[,9])
B[28*ncol(A)+2]<-max(A[,5],A[,7],A[,8],A[,9])
B[28*ncol(A)+3]<-max(A[,6],A[,7],A[,8],A[,9])
for (j in (28*ncol(A)+4):(28*ncol(A)+8))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,j-(27*ncol(A)+8)])
for (j in (29*ncol(A)):(29*ncol(A)+3))
B[j]<-max(A[,1],A[,2],A[,3],A[,5],A[,j-(28*ncol(A)+3)])
for (j in (29*ncol(A)+4):(29*ncol(A)+6))
B[j]<-max(A[,1],A[,2],A[,3],A[,6],A[,j-(28*ncol(A)+6)])
for (j in (29*ncol(A)+7):(29*ncol(A)+8))
B[j]<-max(A[,1],A[,2],A[,3],A[,7],A[,j-(28*ncol(A)+8)])
B[30*ncol(A)]<-max(A[,1],A[,2],A[,3],A[,8],A[,9])
for (j in (30*ncol(A)+1):(30*ncol(A)+4))
B[j]<-max(A[,1],A[,2],A[,4],A[,5],A[,j-(29*ncol(A)+4)])
for (j in (30*ncol(A)+5):(30*ncol(A)+7))
B[j]<-max(A[,1],A[,2],A[,4],A[,6],A[,j-(29*ncol(A)+7)])
for (j in (30*ncol(A)+8):(31*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,4],A[,7],A[,j-(30*ncol(A))])
B[31*ncol(A)+1]<-max(A[,1],A[,2],A[,4],A[,8],A[,9])
for (j in (31*ncol(A)+2):(31*ncol(A)+4))
B[j]<-max(A[,1],A[,2],A[,5],A[,6],A[,j-(30*ncol(A)+4)])
for (j in (31*ncol(A)+5):(31*ncol(A)+6))
B[j]<-max(A[,1],A[,2],A[,5],A[,7],A[,j-(30*ncol(A)+6)])
B[31*ncol(A)+7]<-max(A[,1],A[,2],A[,5],A[,8],A[,9])
for (j in (31*ncol(A)+8):(32*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,6],A[,7],A[,j-(31*ncol(A))])
B[32*ncol(A)+1]<-max(A[,1],A[,2],A[,6],A[,8],A[,9])
B[32*ncol(A)+2]<-max(A[,1],A[,2],A[,7],A[,8],A[,9])
for (j in (32*ncol(A)+3):(32*ncol(A)+6))
B[j]<-max(A[,1],A[,3],A[,4],A[,5],A[,j-(31*ncol(A)+6)])
for (j in (32*ncol(A)+7):(33*ncol(A)))
B[j]<-max(A[,1],A[,3],A[,4],A[,6],A[,j-(32*ncol(A))])
for (j in (33*ncol(A)+1):(33*ncol(A)+2))
B[j]<-max(A[,1],A[,3],A[,4],A[,7],A[,j-(32*ncol(A)+2)])
B[33*ncol(A)+3]<-max(A[,1],A[,3],A[,4],A[,8],A[,9])
```

```
for (j in (33*ncol(A)+4):(33*ncol(A)+6))
B[j]<-max(A[,1],A[,3],A[,5],A[,6],A[,j-(32*ncol(A)+6)])
for (j in (33*ncol(A)+7):(33*ncol(A)+8))
B[j]<-max(A[,1],A[,3],A[,5],A[,7],A[,j-(32*ncol(A)+8)])
B[34*ncol(A)]<-max(A[,1],A[,3],A[,5],A[,8],A[,9])
for (j in (34*ncol(A)+1):(34*ncol(A)+2))
B[j]<-max(A[,1],A[,3],A[,6],A[,7],A[,j-(33*ncol(A)+2)])
B[34*ncol(A)+3]<-max(A[,1],A[,3],A[,6],A[,8],A[,9])
B[34*ncol(A)+4]<-max(A[,1],A[,3],A[,7],A[,8],A[,9])
for (j in (34*ncol(A)+5):(34*ncol(A)+7))
B[j]<-max(A[,1],A[,4],A[,5],A[,6],A[,j-(33*ncol(A)+7)])
for (j in (34*ncol(A)+8):(35*ncol(A)))
B[j]<-max(A[,1],A[,4],A[,5],A[,7],A[,j-(34*ncol(A))])
B[35*ncol(A)+1]<-max(A[,1],A[,4],A[,5],A[,8],A[,9])
for (j in (35*ncol(A)+2):(35*ncol(A)+3))
B[j]<-max(A[,1],A[,4],A[,6],A[,7],A[,j-(34*ncol(A)+3)])
B[35*ncol(A)+4]<-max(A[,1],A[,4],A[,6],A[,8],A[,9])
B[35*ncol(A)+5]<-max(A[,1],A[,4],A[,7],A[,8],A[,9])
for (j in (35*ncol(A)+6):(35*ncol(A)+7))
B[j]<-max(A[,1],A[,5],A[,6],A[,7],A[,j-(34*ncol(A)+7)])
B[35*ncol(A)+8]<-max(A[,1],A[,5],A[,6],A[,8],A[,9])
B[36*ncol(A)]<-max(A[,1],A[,5],A[,7],A[,8],A[,9])
B[36*ncol(A)+1]<-max(A[,1],A[,6],A[,7],A[,8],A[,9])
for (j in (36*ncol(A)+2):(36*ncol(A)+5))
B[j]<-max(A[,2],A[,3],A[,4],A[,5],A[,j-(35*ncol(A)+5)])
for (j in (36*ncol(A)+6):(36*ncol(A)+8))
B[j]<-max(A[,2],A[,3],A[,4],A[,6],A[,j-(35*ncol(A)+8)])
for (j in (37*ncol(A)):(37*ncol(A)+1))
B[j]<-max(A[,2],A[,3],A[,4],A[,7],A[,j-(36*ncol(A)+1)])
B[37*ncol(A)+2]<-max(A[,2],A[,3],A[,4],A[,8],A[,9])
for (j in (37*ncol(A)+3):(37*ncol(A)+5))
B[j]<-max(A[,2],A[,3],A[,5],A[,6],A[,j-(36*ncol(A)+5)])
for (j in (37*ncol(A)+6):(37*ncol(A)+7))
B[j]<-max(A[,2],A[,3],A[,5],A[,7],A[,j-(36*ncol(A)+7)])
B[37*ncol(A)+8]<-max(A[,2],A[,3],A[,5],A[,8],A[,9])
for (j in (38*ncol(A)):(38*ncol(A)+1))
B[j]<-max(A[,2],A[,3],A[,6],A[,7],A[,j-(37*ncol(A)+1)])
B[38*ncol(A)+2]<-max(A[,2],A[,3],A[,6],A[,8],A[,9])
B[38*ncol(A)+3]<-max(A[,2],A[,3],A[,7],A[,8],A[,9])
for (j in (38*ncol(A)+4):(38*ncol(A)+6))
B[j]<-max(A[,2],A[,4],A[,5],A[,6],A[,j-(37*ncol(A)+6)])
for (j in (38*ncol(A)+7):(38*ncol(A)+8))
B[j]<-max(A[,2],A[,4],A[,5],A[,7],A[,j-(37*ncol(A)+8)])
B[39*ncol(A)]<-max(A[,2],A[,4],A[,5],A[,8],A[,9])
for (j in (39*ncol(A)+1):(39*ncol(A)+2))
B[j]<-max(A[,2],A[,4],A[,6],A[,7],A[,j-(38*ncol(A)+2)])
B[39*ncol(A)+3]<-max(A[,2],A[,4],A[,6],A[,8],A[,9])
B[39*ncol(A)+4]<-max(A[,2],A[,4],A[,7],A[,8],A[,9])
for (j in (39*ncol(A)+5):(39*ncol(A)+6))
B[j]<-max(A[,2],A[,5],A[,6],A[,7],A[,j-(38*ncol(A)+6)])
B[39*ncol(A)+7]<-max(A[,2],A[,5],A[,6],A[,8],A[,9])
B[39*ncol(A)+8]<-max(A[,2],A[,5],A[,7],A[,8],A[,9])
B[40*ncol(A)]<-max(A[,2],A[,6],A[,7],A[,8],A[,9])
for (j in (40*ol(A)+1):(40*ncol(A)+3))
B[j]<-max(A[,3],A[,4],A[,5],A[,6],A[,j-(39*ncol(A)+3)])
for (j in (40*ncol(A)+4):(40*ncol(A)+5))
B[j]<-max(A[,3],A[,4],A[,5],A[,7],A[,j-(39*ncol(A)+5)])
B[40*ncol(A)+6]<-max(A[,3],A[,4],A[,5],A[,8],A[,9])
for (j in (40*ncol(A)+7):(40*ncol(A)+8))
B[j]<-max(A[,3],A[,4],A[,6],A[,7],A[,j-(39*ncol(A)+8)])
B[41*ncol(A)]<-max(A[,3],A[,4],A[,6],A[,8],A[,9])
B[41*ncol(A)+1]<-max(A[,3],A[,4],A[,7],A[,8],A[,9])
for (j in (41*ncol(A)+2):(41*ncol(A)+3))
B[j]<-max(A[,3],A[,5],A[,6],A[,7],A[,j-(40*ncol(A)+3)])
B[41*ncol(A)+4]<-max(A[,3],A[,5],A[,6],A[,8],A[,9])
B[41*ncol(A)+5]<-max(A[,3],A[,5],A[,7],A[,8],A[,9])
B[41*ncol(A)+6]<-max(A[,3],A[,6],A[,7],A[,8],A[,9])
for (j in (41*ncol(A)+7):(41*ncol(A)+8))
B[j]<-max(A[,4],A[,5],A[,6],A[,7],A[,j-(40*ncol(A)+8)])
B[42*ncol(A)]<-max(A[,4],A[,5],A[,6],A[,8],A[,9])
B[42*ncol(A)+1]<-max(A[,4],A[,5],A[,7],A[,8],A[,9])
B[42*ncol(A)+2]<-max(A[,4],A[,6],A[,7],A[,8],A[,9])
B[42*ncol(A)+3]<-max(A[,5],A[,6],A[,7],A[,8],A[,9])
```

```
for (j in (42*ncol(A)+4):(42*ncol(A)+7))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[,j-(41*ncol(A)+7)])
for (j in (42*ncol(A)+8):(43*ncol(A)+1))
B[j]<-max(A[,1],A[,2],A[,3],A[,4], A[,6],A[,j-(42*ncol(A)+1)])
for (j in (43*ncol(A)+2):(43*ncol(A)+3))
B[j]<-max(A[,1],A[,2],A[,3],A[,4], A[,7],A[,j-(42*ncol(A)+3)])
B[43*ncol(A)+4]<-max(A[,1],A[,2],A[,3],A[,4],A[,8],A[,9])
for (j in (43*ncol(A)+5):(43*ncol(A)+7))
B[j]<-max(A[,1],A[,2],A[,3],A[,5],A[,6],A[,j-(42*ncol(A)+7)])
for (j in (43*ncol(A)+8):(44*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,3],A[,5], A[,7],A[,j-(43*ncol(A))])
B[44*ncol(A)+1]<-max(A[,1],A[,2],A[,3],A[,5],A[,8],A[,9])
for (j in (44*ncol(A)+2):(44*ncol(A)+3))
B[j]<-max(A[,1],A[,2],A[,3],A[,6], A[,7],A[,j-(43*ncol(A)+3)])
B[44*ncol(A)+4]<-max(A[,1],A[,2],A[,3],A[,6],A[,8],A[,9])
B[44*ncol(A)+5]<-max(A[,1],A[,2],A[,3],A[,7],A[,8],A[,9])
for (j in (44*ncol(A)+6):(44*ncol(A)+8))
B[j]<-max(A[,1],A[,2],A[,4],A[,5],A[,6],A[,j-(43*ncol(A)+8)])
for (j in (45*ncol(A)):(45*ncol(A)+1))
B[j]<-max(A[,1],A[,2],A[,4],A[,5], A[,7],A[,j-(44*ncol(A)+1)])
B[45*ncol(A)+2]<-max(A[,1],A[,2],A[,4],A[,5],A[,8],A[,9])
for (j in (45*ncol(A)+3):(45*ncol(A)+4))
B[j]<-max(A[,1],A[,2],A[,4],A[,6], A[,7],A[,j-(44*ncol(A)+4)])
B[45*ncol(A)+5]<-max(A[,1],A[,2],A[,4],A[,6],A[,8],A[,9])
B[45*ncol(A)+6]<-max(A[,1],A[,2],A[,4],A[,7],A[,8],A[,9])
for (j in (45*ncol(A)+7):(45*ncol(A)+8))
B[j]<-max(A[,1],A[,2],A[,5],A[,6], A[,7],A[,j-(44*ncol(A)+8)])
B[46*ncol(A)]<-max(A[,1],A[,2],A[,5],A[,6],A[,8],A[,9])
B[46*ncol(A)+1]<-max(A[,1],A[,2],A[,5],A[,7],A[,8],A[,9])
B[46*ncol(A)+2]<-max(A[,1],A[,2],A[,6],A[,7],A[,8],A[,9])
for (j in (46*ncol(A)+3):(46*ncol(A)+5))
B[j]<-max(A[,1],A[,3],A[,4],A[,5],A[,6],A[,j-(45*ncol(A)+5)])
for (j in (46*ncol(A)+6):(46*ncol(A)+7))
B[j]<-max(A[,1],A[,3],A[,4],A[,5], A[,7],A[,j-(45*ncol(A)+7)])
B[46*ncol(A)+8]<-max(A[,1],A[,3],A[,4],A[,5],A[,8],A[,9])
for (j in (47*ncol(A)):(47*ncol(A)+1))
B[j]<-max(A[,1],A[,3],A[,4],A[,6], A[,7],A[,j-(46*ncol(A)+1)])
B[47*ncol(A)+2]<-max(A[,1],A[,3],A[,4],A[,6],A[,8],A[,9])
B[47*ncol(A)+3]<-max(A[,1],A[,3],A[,4],A[,7],A[,8],A[,9])
for (j in (47*ncol(A)+4):(47*ncol(A)+5))
B[j]<-max(A[,1],A[,3],A[,5],A[,6], A[,7],A[,j-(46*ncol(A)+5)])
B[47*ncol(A)+6]<-max(A[,1],A[,3],A[,5],A[,6],A[,8],A[,9])
B[47*ncol(A)+7]<-max(A[,1],A[,3],A[,5],A[,7],A[,8],A[,9])
B[47*ncol(A)+8]<-max(A[,1],A[,3],A[,6],A[,7],A[,8],A[,9])
for (j in (48*ncol(A)):(48*ncol(A)+1))
B[j]<-max(A[,1],A[,4],A[,5],A[,6], A[,7],A[,j-(47*ncol(A)+1)])
B[48*ncol(A)+2]<-max(A[,1],A[,4],A[,5],A[,6],A[,8],A[,9])
B[48*ncol(A)+3]<-max(A[,1],A[,4],A[,5],A[,7],A[,8],A[,9])
B[48*ncol(A)+4]<-max(A[,1],A[,4],A[,6],A[,7],A[,8],A[,9])
B[48*ncol(A)+5]<-max(A[,1],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (48*ncol(A)+6):(48*ncol(A)+8))
B[j]<-max(A[,2],A[,3],A[,4],A[,5],A[,6],A[,j-(47*ncol(A)+8)])
for (j in (49*ncol(A)):(49*ncol(A)+1))
B[j]<-max(A[,2],A[,3],A[,4],A[,5], A[,7],A[,j-(48*ncol(A)+1)])
B[49*ncol(A)+2]<-max(A[,2],A[,3],A[,4],A[,5],A[,8],A[,9])
for (j in (49*ncol(A)+3):(49*ncol(A)+4))
B[j]<-max(A[,2],A[,3],A[,4],A[,6], A[,7],A[,j-(48*ncol(A)+4)])
B[49*ncol(A)+5]<-max(A[,2],A[,3],A[,4],A[,6],A[,8],A[,9])
B[49*ncol(A)+6]<-max(A[,2],A[,3],A[,4],A[,7],A[,8],A[,9])
for (j in (49*ncol(A)+7):(49*ncol(A)+8))
B[j]<-max(A[,2],A[,3],A[,5],A[,6], A[,7],A[,j-(48*ncol(A)+8)])
B[50*ncol(A)]<-max(A[,2],A[,3],A[,5],A[,6],A[,8],A[,9])
B[50*ncol(A)+1]<-max(A[,2],A[,3],A[,5],A[,7],A[,8],A[,9])
B[50*ncol(A)+2]<-max(A[,2],A[,3],A[,6],A[,7],A[,8],A[,9])
for (j in (50*ncol(A)+3):(50*ncol(A)+4))
B[j]<-max(A[,2],A[,4],A[,5],A[,6], A[,7],A[,j-(49*ncol(A)+4)])
B[50*ncol(A)+5]<-max(A[,2],A[,4],A[,5],A[,6],A[,8],A[,9])
B[50*ncol(A)+6]<-max(A[,2],A[,4],A[,5],A[,7],A[,8],A[,9])
B[50*ncol(A)+7]<-max(A[,2],A[,4],A[,6],A[,7],A[,8],A[,9])
B[50*ncol(A)+8]<-max(A[,2],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (51*ncol(A)):(51*ncol(A)+1))
B[j]<-max(A[,3],A[,4],A[,5],A[,6], A[,7],A[,j-(50*ncol(A)+1)])
B[51*ncol(A)+2]<-max(A[,3],A[,4],A[,5],A[,6],A[,8],A[,9])
```

```
B[51*ncol(A)+3]<-max(A[,3],A[,4],A[,5],A[,7],A[,8],A[,9])
B[51*ncol(A)+4]<-max(A[,3],A[,4],A[,6],A[,7],A[,8],A[,9])
B[51*ncol(A)+5]<-max(A[,3],A[,5],A[,6],A[,7],A[,8],A[,9])
B[51*ncol(A)+6]<-max(A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (51*ncol(A)+7):(52*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[6],A[,j-(51*ncol(A))])
for (j in (52*ncol(A)+1):(52*ncol(A)+2))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,5] ,A[,7],A[,j-(51*ncol(A)+2)])
B[52*ncol(A)+3]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[,8],A[,9])
for (j in (52*ncol(A)+4):(52*ncol(A)+5))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,6], A[,7],A[,j-(51*ncol(A)+5)])
B[52*ncol(A)+6]<-max(A[,1],A[,2],A[,3],A[,4],A[,6], A[,8],A[,9])
B[52*ncol(A)+7]<-max(A[,1],A[,2],A[,3],A[,4],A[,7], A[,8],A[,9])
for (j in (52*ncol(A)+8):(53*ncol(A)))
B[j]<-max(A[,1],A[,2],A[,3],A[,5],A[,6], A[,7],A[,j-(52*ncol(A))])
B[53*ncol(A)+1]<-max(A[,1],A[,2],A[,3],A[,5],A[,6],A[,8],A[,9])
B[53*ncol(A)+2]<-max(A[,1],A[,2],A[,3],A[,5],A[,7],A[,8],A[,9])
B[53*ncol(A)+3]<-max(A[,1],A[,2],A[,3],A[,6],A[,7],A[,8],A[,9])
for (j in (53*ncol(A)+4):(53*ncol(A)+5))
B[j]<-max(A[,1],A[,2],A[,4],A[,5],A[,6], A[,7],A[,j-(52*ncol(A)+5)])
B[53*ncol(A)+6]<-max(A[,1],A[,2],A[,4],A[,5],A[,6],A[,8],A[,9])
B[53*ncol(A)+7]<-max(A[,1],A[,2],A[,4],A[,5],A[,7],A[,8],A[,9])
B[53*ncol(A)+8]<-max(A[,1],A[,2],A[,4],A[,6],A[,7],A[,8],A[,9])
B[54*ncol(A)]<-max(A[,1],A[,2],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (54*ncol(A)+1):(54*ncol(A)+2))
B[j]<-max(A[,1],A[,3],A[,4],A[,5],A[,6], A[,7],A[,j-(53*ncol(A)+2)])
B[54*ncol(A)+3]<-max(A[,1],A[,3],A[,4],A[,5],A[,6],A[,8],A[,9])
B[54*ncol(A)+4]<-max(A[,1],A[,3],A[,4],A[,5],A[,7],A[,8],A[,9])
B[54*ncol(A)+5]<-max(A[,1],A[,3],A[,4],A[,6],A[,7],A[,8],A[,9])
B[54*ncol(A)+6]<-max(A[,1],A[,3],A[,5],A[,6],A[,7],A[,8],A[,9])
B[54*ncol(A)+7]<-max(A[,1],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (54*ncol(A)+8):(55*ncol(A)))
B[j]<-max(A[,2],A[,3],A[,4],A[,5],A[,6], A[,7],A[,j-(54*ncol(A))])
B[55*ncol(A)+1]<-max(A[,2],A[,3],A[,4],A[,5],A[,6],A[,8],A[,9])
B[55*ncol(A)+2]<-max(A[,2],A[,3],A[,4],A[,5],A[,7],A[,8],A[,9])
B[55*ncol(A)+3]<-max(A[,2],A[,3],A[,4],A[,6],A[,7],A[,8],A[,9])
B[55*ncol(A)+4]<-max(A[,2],A[,3],A[,5],A[,6],A[,7],A[,8],A[,9])
B[55*ncol(A)+5]<-max(A[,2],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
B[55*ncol(A)+6]<-max(A[,3],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
for (j in (55*ncol(A)+7):(55*ncol(A)+8))
B[j]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[,6], A[,7],A[,j-(54*ncol(A)+8)])
B[56*ncol(A)]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[,6],A[,8],A[,9])
B[56*ncol(A)+1]<-max(A[,1],A[,2],A[,3],A[,4],A[,5],A[,7],A[,8],A[,9])
B[56*ncol(A)+2]<-max(A[,1],A[,2],A[,3],A[,4],A[,6],A[,7],A[,8],A[,9])
B[56*ncol(A)+3]<-max(A[,1],A[,2],A[,3],A[,5],A[,6],A[,7],A[,8],A[,9])
B[56*ncol(A)+4]<-max(A[,1],A[,2],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
B[56*ncol(A)+5]<-max(A[,1],A[,3],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
B[56*ncol(A)+6]<-max(A[,2],A[,3],A[,4],A[,5],A[,6],A[,7],A[,8],A[,9])
#Choquet
F<-max(B)
E<-B/F
library(kappalab)
c9<-capacity(c(0,E,1))
score<- rep(1,nrow(A))
for (i in 1:nrow(A))
score[i]<- Choquet.integral(c9,A[i,])
write.table(score, file="scoremax9.txt")
Shapley<-Shapley.value(c9)
write.table(Shapley, file="Shapleymax9.txt")
```

5. Conclusão

Diferentes procedimentos estatísticos empregados em circunstâncias não usuais foram aqui apresentados. Primeiro, a distribuição empírica conjunta foi empregada na determinação de preferências por múltiplos critérios em substituição a formas de composição que exigem a obtenção de mais informação sobre a importância a ser atribuída a cada critério. Na segunda aplicação, o aspecto mais interessante é o uso que é feito do tamanho da amostra na estimação dos parâmetros da distribuição beta. Finalmente, um novo uso do máximo da amostra é apresentado, empregado em processo de determinação de capacidades de Choquet e da importância de um conjunto numeroso de critérios.

Referências

BANZHAF, J. Weighted voting doesn't work : A Mathematical Analysis, **Rutgers Law Review**, 19, 317-343, 1965.

CHOQUET G. Theory of capacities. **Annales de l'Institut Fourier**, 5,131–295, 1953.

GAVIÃO, L.O.; SANT'ANNA, A. P.; LIMA, G. B. A.; GARCIA, P. A. CPP: Composition of Probabilistic Preferences. R package version 0.1.0. 2018. Disponível em https://cran.r-project.org/package=CPP.

GRABISCH M,; KOJADINOVIC I; MEYER, P. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. **European Journal of Operational Research**, 186,766–785, 2008.

HOLLER, M. J.; PACKEL, E. W. Power, Luck and the Right Index, **Journal of Economics**, 43, 21-29, 1983.

JOHNSTON, R. J. On the measurement of Power: Some Reactions to Laver, **Environment and Planning** A, 10, 907-914, 1978.

KOJADINOVIC, I. Unsupervised aggregation of commensurate correlated attributes by means of the Choquet integral and entropy functionals. **International Journal of Intelligent Systems**, 23,128-154, 2008.

R CORE TEAM **R: A language and environment for statistical computing**. R Foundation for Statistical Computing,Vienna. Disponível em https://www.R-project.org/, 2017.

SANT'ANNA, A. P. **Probabilistic composition of preferences: theory and applications**. Springer, Heidelberg, 2015.

SANT'ANNA, A. P.; MARTINS, E. F.; LIMA, G. B. A.; FONSECA, R. A. Beta distributed preferences in the comparison of failure modes. **Procedia Computer Science**, 55, 862-869, 2015.

SANT'ANNA, A.P.; LIMA, G. B. A.; SANT'ANNA, L. A. F. P.; GAVIÃO, L. O. Two-Stage Composition of Probabilistic Preferences. **Annals of Data Science**. Disponível em https://link.springer.com/article/10.1007%2Fs40745-018-0177-9, 2020, no prelo.

SANT'ANNA A. P.; SANT'ANNA, J. L. A Principle of Preference Concentration applied to the unsupervised evaluation of the importance of multiple criteria. **Pesquisa Operacional**, 39, 317-338, 2019.

SHAPLEY, L. A value for n-person games. In Contributions to the Theory of Games, Vol. II. **Annals of Mathematics Studies**, 28. Kuhn H.; Tucker A. (Eds.) Princeton U. Press: Princeton, 307–317, 1953.

US MILITARY, **Procedures for Performing a Failure Mode, Effects and Criticality Analysis**, United States Military Procedure MIL-P-1629, 1949.

SOME STATISTICAL TECHNIQUES OF PROBABILISTIC COMPOSITION OF PREFERENCES

Abstract

This article brings together a set of statistical procedures employed at different stages in Composition of Probabilistic Preferences. These procedures include estimating probabilities of preference for the formation of preference scores based on the joint cumulative empirical distribution, replacing weighted averages of evaluations in different dimensions or other forms of composition. In other application, an alternative of estimation of beta distribution parameters is presented, ensuring unimodality, and using the sample size in the estimation of dispersion parameters. Another procedure is presented where, for a set of up to nine criteria, the maximum limited sample is used in the process of determining capacities and the importance of criteria.

Keywords: *Probabilistic Composition of Probabilistic Preferences; Empirical Distribution; Beta Distribution; Shapley value; Choquet capacity.*