Vitor Passos Rios

Trabalho Final

Proposta B - Sequências Comportamentais

A função recebe os dados, verifica se as entradas do usuário estão de acordo com a formatação exigida, e realiza os cálculos. Primeiro são calculadas as durações de cada comportamento subtraindo-se o valor de tempo do comportamento atual daquele do comportamento seguinte. O modo de cálculo da duração depende do formato de entrada do tempo, utilizando a função *diff.time* para o tempo formatado em *HH:MM:SS*, e subtração simples para o tempo em segundos. Depois, é utilizada a função *table* para calcular a matriz de transição entre os comportamentos e as frequências dos comportamentos, guardados em objetos separados. A função *tapply* é utilizada para calcular a duração total de cada tipo de comportamento, somando as ocorrências individuais destes. Um *data.frame* é criado para resumir os dados de frequência, duração total e duração média dos comportamentos, e a função retorna uma lista com este *data.frame* e a matriz de transição. A opção de plotar os dados não estava incluída na proposta original, porém facilita muito a visualização da matriz de transição. Para isto é utilizado o pacote *diagram*, que possui funções específicas para plotar fluxogramas. A função não permite passar argumentos para as funções de plotagem, porém estas podem ser alteradas diretamente no código.

sequence.r meu_help.txt

Função Sequence

```
sequence=function(seq, plot=0)
{
  ### testando validade do input
  if(plot!=0 & plot!=1 & plot!=2)
  {
    plot=0
    message("Tipo de plot selecionado incorreto, nenhum gráfico será
produzido")
  }
  ###calculo das durações de cada comportamento
  if(is.numeric(seq[,2]) )
  {
    ## calcula a duração de cada comportamento com tempo de entrada em
segundo
    duration=function(seq)
    {return(c(seq[2:nrow(seq),2],seq[nrow(seq),2])-seq[,2])}
    seq[,3]=duration(seq)
  }
  else{
```

```
    if(is.character(seq[,2]))
    {
      ## calcula a duração de cada comportamento com tempo de entrada
"0:00:00"
      duration2=function(seq)
      {return (difftime(c(seq[2:nrow(seq),2],seq[nrow(seq),2]),seq[,2]))}
      seq[,2]=as.POSIXct(strptime(seq[,2], format="%H:%M:%S"))
      seq[,3]=duration2(seq)
    }
    else{stop("Tipo incorreto de formato de tempo. Formatos válidos são
HH:MM:SS ou segundos")}
  }
  trans.matrix=table(seq[-nrow(seq),1],seq[-1,1])
  trans.matrix.prob=round(trans.matrix/rowSums(trans.matrix),digits=3)
  ###calculo das frequencias de cada comportamento
  freq=as.data.frame(table(seq[,1]))
  dura=data.frame(as.factor(freq[,1]),
                  freq[,2],
                  round(tapply(X=seq[,3], INDEX=seq[,1], FUN=sum),digits=3),
                  0)
  colnames(dura)=c("Comportamento","Frequencia","Duracao","Media")
  dura[,4]=round(dura[,3]/dura[,2],digits=3)
  if (plot==1)
  {
    require(diagram)
    x11()
    plotweb(trans.matrix.prob,legend=T, arr.col="blue")
  }
  if(plot==2)
  {
    require(diagram)
    x11()
    plotmat(trans.matrix.prob, shadow.size=0,box.type="none",
arr.lcol="black", txt.col="blue")
  }
  res=list(trans.matrix.prob, dura)
  names(res)=c("Matriz.Trans","Duracao")
  return(res)
}
```

Help da função

```
Descrição
Calcula matrizes de transição, frequências e durações médias de
comportamentos em uma série temporal.

Uso
sequence=function(seq, plot=0)
```

```
Argumentos
seq Sequência comportamental a ser analisada. Um data.frame com duas coluna,
uma contendo a  série de comportamentos e a
        outra contendo os tempos. Formatos aceitos são "Comportamento"
"HH:MM:SS" ou "Comportamento" "segundos".
plot    Requer o pacote "diagram”. Define o tipo de gráfico a ser plotado.
Se =1, chama a função "plotweb", se =2, chama a
        função "plotmat”. Qualquer outro valor resulta em nenhum plot.

Detalhes
A função calcula a matriz de transição entre os comportamentos e  frequência
e duração média destes. A coluna com os
comportamentos pode estar em qualquer formato aceito pelo R. É assumido que
os comportamentos são estados, e o tempo é o momento
de início do comportamento, que deve estar em segundos ou no formato
"HH:MM:SS". No último caso, a string de caracteres é
convertida um objeto do formato POSIXct para calculo dos tempos. A matriz de
transição calcula a probabilidade de um
comportamento ser seguido por outro dentro da sequencia fornecida.

A plotagem da matriz de transição utiliza as funções “plotweb” e “plotmat”
do pacote “diagram”, específicas para plotar
fluxogramas. “plotweb” plota uma rede com a espessura das arestas
proporcional ao valor da probabilidade de transição entre os
comportamentos, enquanto “plotmat” plota uma rede com os valores de
transição ao lado de cada aresta.

Valor
A função retorna uma lista com dois itens, a matriz de transição e um
data.frame com as frequências e durações médias de cada
comportamento.  A duração dos comportamentos é mostrada em segundos,
independente do tipo de entrada. No caso de inputs
incorretos, retorna uma mensagem de aviso.

Autor
Vitor Passos Rios

Package diagram by Karline Soetaert, Netherlands Institute of Sea Research
(NIOZ)
(http://cran.r-project.org/web/packages/diagram/vignettes/diagram.pdf)

Ver também
plotmat, plotweb, diagram

Exemplos
library("sampling")
l=srswor(100,1000)
times=(1:1000)[l==1]
behav=sample(x=paste(letters,letters),size=length(times),replace=T)
seq=data.frame(behaviour=as.factor(behav),time=times)
a=sequence(seq, plot=1)
```

a

Propostas de Trabalho Final

A

Implementar o algoritmo DBSCAN para detecção de clusters espaciais de forma e tamanho arbitrários. O algoritmo calcula clusters baseado em uma densidade mínima de vizinhos para cada ponto em um determinado raio de vizinhança. Os pontos são atribuídos ao mesmo cluster se estiverem um no raio de vizinhança do outro(desde que haja uma densidade mínima de pontos), ou se for possível estabelecer entre eles uma sequencia de pontos cujos raios de vizinhança se sobreponham. Pontos que não se encaixam nessas condições são considerados ruído. O algoritmo é descrito em detalhes em Ester et al. (1996) “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”.Ester et al. A função deve receber uma tabela de pontos com coordenadas X e Y, um valor de número mínimo de vizinhos e um raio de vizinhança, e retornar uma lista com os clusters (com uma lista de seus respectivos pontos) e uma lista dos pontos considerados ruído.

B

Criar uma função que faça uma análise de sequências de comportamentos animais e retorne para cada categoria comportamental a frequência, duração média, e as probabilidade que cada categoria comportamental tem de ser seguida por cada uma das outras na sequência. A função deve receber uma tabela com as séries temporais de comportamentos, com o momento de inicio de cada comportamento, como na tabela abaixo:

comportamento a	hh:mm:ss
comportamento b	hh:mm:ss
comportamento a	hh:mm:ss
comportamento d	hh:mm:ss

Comentários

Vou comentar a proposta B, mais alguém virá para opinar sobre a A! 

A função parece bem viável e útil. Frequência e duração média por categoria serão facilmente obtidas com funções table, família apply e aggregate. A probabilidade da sequência de comportamentos dará um pouco mais de trabalho, mas acho que é possível se você pensar bem em que critérios da sequência de comportamentos observada serão importantes e como entrarão para o calculo da probabilidade - esses critérios talvez devam ser explicados no Help, assim o usuário saberá o que foi usado para calcular a probabilidade do comportamento acontecer (frequência com que o comportamento ocorreu após um outro; duração do comportamento após outro, ainda que a frequência não seja a maior etc). Se você conseguir contruir essa função com facilidade, pode pensar em outras observações que seriam interessantes de se obter (período do dia em que algum comportamento foi mais frequente etc). Pense também em como você usará os dados de horário;

como eles entrarão na função e como irá subtrair um do outro para saber a duração do comportamento. — *Tauana Junqueira da Cunha 2013/03/20 10:08*

Comentario

Gostei muito do plano A, um desafio interessante. Tente ficar longe do algoritmo que eles publicaram no artigo ou corre o risco de fazer apenas uma traduçao. Uma possivel ampliaçao do algoritmo poderia ser pensado para dados com mais dimensoes, para agrupar objetos em espaco multidimensional, no caso da ecologia isso seria muito util (mas isso fica para uma outra fase!!). Fica a seu criterio plano A ou B, ambos sao viaveis e factiveis.

— *Alexandre Adalardo de Oliveira 2013/03/25 11:55*

Exercícios

exec