# Análise de Conectividade
# Extracção de Conhecimento da Internet
# Mestrado em Inteligência Artificial e Sistemas Inteligentes

António Jesus Monteiro de Castro
Aluno Nº 040594004

Faculdade de Engenharia da Universidade do Porto
frisky.antonio@gmail.com

10 de Julho de 2005

## Resumo

Neste o objectivo é fazer a análise de conectividade de um determinado site/tópico de pesquisa. Para o efeito, será criado um grafo de hiperligações e, depois, para cada página, será determinada a sua importância como *Hub* e como *Authority.* O algoritmo utilizado para realizar este trabalho chama-se HITS [1] e fiz o meu trabalho com base no livro *Mining the World Wide Web* [2] e na sebenta da disciplina [3]. A implementação foi realizada, tal como sugerido pelo enunciado do trabalho, em R [4].

O programa foi testado em dois sites, *www.liacc.up.pt* e *www.appia.pt* e os resultados obtidos fazem parte deste relatório.


**Palavras-chave:** HITS, HUB, AUTHORITY, Análise de conectividade.

# 1 Resumo da Metodologia Utilizada

Ao fazer a análise de conectividade com HITS e dando cumprimento ao enunciado deste trabalho, iremos obter o grafo de hiperligações (Matriz de Adjacências) e, também, o peso das páginas como *Hub* (páginas que referem muitas outras) e como *Authorities* (páginas referidas por muitas outras). Basicamente, o processo será o seguinte:

- Obter as páginas que digam respeito a um determinado tópico, utilizando um motor de pesquisa para o efeito.
- Das páginas obtidas anteriormente, escolher as mais relevantes e, para cada uma delas, obter: (1) as páginas que a referem (*In Links*) e (2) as páginas referidas por essa página (*Out Links*).
- Criar uma matriz de adjacências (que será o Grafo de Hiperligações) com as páginas obtidas pelos dois passos anteriores. As células deverão ser preenchidas com o algarismo 1 nos casos em que houver uma relação de ligação entre as páginas envolvidas e com o algarismo 0 nos restantes casos.
- Aplicar o algoritmo HITS à matriz de adjacências, tal como explicado na documentação [2][3].
- A comunidade correspondente ao tópico procurado, será composta pelas páginas com o valor de *Authority* mais elevado e pelas páginas com o valor de *Hub* mais elevado.

Para realizar os passos anteriores foram desenvolvidas algumas funções em R e/ou adaptadas algumas das existentes na documentação. Durante a utilização do código foram detectados alguns problemas, nomeadamente ao nível do *parser* XML que não consegui resolver. Essa foi a razão principal por não ter utilizado (apesar de a ter criado) uma função que, dado o tópico, realizasse automaticamente todos os passos e apresentasse os resultados. Assim, para cada um dos passos anteriores, optei por chamar a função R adequada e, depois de analisados os resultados, passar para a fase seguinte. De qualquer maneira, a função INICIAR(tópico a procurar) que incluo no código em anexo, serviria para fazer todo o ciclo do programa, não fossem os erros obtidos pelo meio e que fazem com que o programa pare.

# 2 Aplicação do Programa ao Tópico 'www.liacc.up.pt'

Em primeiro lugar é necessário obter o conjunto de páginas que estão relacionadas com o tópico que nos interessa. Neste caso vamos começar pelo tópico www.liacc.up.pt . Para o efeito vou utilizar a função **`> urlsS<- obterS('www.liacc.up.pt')`** que, utilizando o motor de busca Google vai retornar as páginas que fazem parte do site (comando site:www.liacc.up.pt). A lista devolvida é a seguinte:

```
> urlsS
 [1] "www.google.com/webhp?hl=en"
 [2] "images.google.com/images?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8&sa=N&tab=wi"
 [3] "groups-beta.google.com/groups?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-
8&sa=N&tab=wg"
 [4] "news.google.com/news?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8&sa=N&tab=wn"
 [5] "froogle.google.com/froogle?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-
8&sa=N&tab=wf"
 [6] "local.google.com/local?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8&sa=N&tab=wl"
 [7] "/preferences?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8"
 [8] "www.liacc.up.pt/~ltorgo/Yails/yails_report-3.html"
 [9] "www.liacc.up.pt/NCC/Projs/projectos_10.html"
[10] "www.liacc.up.pt/~amjorge/Aulas/Excel97.html"
[11] "www.liacc.up.pt/~csoares/aulas/dp3.html"
[12] "www.liacc.up.pt/~ltorgo/Ensino/MIAC/ModelosDeRegressao/sld009.htm"
[13] "www.liacc.up.pt/~ltorgo/OtherLearningAlgs/"
[14] "www.liacc.up.pt/ML/METAL/Consortium/classification/FLAT.README"
[15] "www.liacc.up.pt/ML/METAL/Consortium/table_class2.html"
[16] "www.liacc.up.pt/~amjorge/Aulas/ProgramBD.html"
[17] "www.liacc.up.pt/seminarios_antigos/25_01_01.html"
[18] "/swr?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8&swrnum=7070"
[19] "/language_tools?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8"
[20] "/quality_form?q=site:www.liacc.up.pt&hl=en&lr=&ie=UTF-8"
[21] "www.google.com/"
```

Como se pode verificar os resultados que interessam são os que correspondem aos índices 8 ao 17 sendo os restantes pertencentes ao *Google* (esta é uma questão a melhorar no programa).
Agora vamos obter as páginas que referenciam cada uma das páginas entre o índice 8 e 10 (escolhi estas três apenas para não tornar o exemplo muito grande), ou seja, os *In Links*. Para isso utilizo o seguinte comando **`> xyIN<-inLinks(urlsS[8])`** repetindo-o até chegar ao índice 10.
Da mesma forma que é necessário obter os *In Links* também precisamos de obter os *Out Links* para cada uma das páginas anteriores. Para isso utilizo o comando **`> xyOUT<-outLinks(urlsS[8])`** repetindo-o até chegar ao índice 10. De notar que, para este exemplo, as páginas não referenciam nenhum outra página, pelo que não posso mostrar resultados. No final deste ciclo teremos uma matriz com os *In Links* e outra com os *Out Links* (que, como referi anteriormente e para este exemplo, está vazia). Assim a matriz final dos *In Links* é:

```
> xyIN
     [,1]                                        [,2]
[1,]"www.google.com/webhp?hl=en"   "www.liacc.up.pt/~ltorgo/Yails/yails_report-3.html"
[2,]"www.google.com/"              "www.liacc.up.pt/~ltorgo/Yails/yails_report-3.html"
[3,]"www.google.com/webhp?hl=en"   "www.liacc.up.pt/NCC/Projs/projectos_10.html"
[4,]"www.liacc.up.pt/NCC/Projs/projectos_6.html"
"www.liacc.up.pt/NCC/Projs/projectos_10.html"
[5,]"www.liacc.up.pt/NCC/Projs/projectos.html"
"www.liacc.up.pt/NCC/Projs/projectos_10.html"
[6,]"www.google.com/"              "www.liacc.up.pt/NCC/Projs/projectos_10.html"
[7,]"www.google.com/webhp?hl=en"   "www.liacc.up.pt/~amjorge/Aulas/Excel97.html"
[8,]"www.google.com/"              "www.liacc.up.pt/~amjorge/Aulas/Excel97.html"
```

O grafo de hiperligações (ou matriz de adjacências), será obtido criando uma nova matriz em que as linhas e as colunas correspondem às páginas do conjunto base (designadas neste exemplo por s1, s2 e s3 e correspondentes aos índices 8 a 10 acima) mais as encontradas pelos *In Links* (indicadas neste exemplo como x1, x2 até x8) e as encontradas pelos *Out*

*Links* (que não foram encontradas neste exemplo mas que teriam a designação y1, y2, ...). O comando para fazer esta matriz é **> matLigacoes(xyIN,xyOUT).**
A matriz obtida foi a seguinte:

```
> xyALL
   S1 S2 S3 X1 X2 X3 X4 X5 X6 X7 X8
S1  0  0  0  0  0  0  0  0  0  0  0
S2  0  0  0  0  0  0  0  0  0  0  0
S3  0  0  0  0  0  0  0  0  0  0  0
X1  1  0  0  0  0  0  0  0  0  0  0
X2  1  0  0  0  0  0  0  0  0  0  0
X3  0  1  0  0  0  0  0  0  0  0  0
X4  0  1  0  0  0  0  0  0  0  0  0
X5  0  1  0  0  0  0  0  0  0  0  0
X6  0  1  0  0  0  0  0  0  0  0  0
X7  0  0  1  0  0  0  0  0  0  0  0
X8  0  0  1  0  0  0  0  0  0  0  0
```

Esta matriz servirá de entrada à aplicação do algoritmo HITS através do comando **> hits(xyALL,its=10).** O resultado obtido foi:

```
> hits(xyALL,its=10)
       [,1]    [,2]    [,3]    [,4]
 [1,] 0.125 0.00000 0.03125 0.00000
 [2,] 4.000 0.00000 1.00000 0.00000
 [3,] 0.125 0.00000 0.03125 0.00000
 [4,] 0.000 0.03125 0.00000 0.03125
 [5,] 0.000 0.03125 0.00000 0.03125
 [6,] 0.000 1.00000 0.00000 1.00000
 [7,] 0.000 1.00000 0.00000 1.00000
 [8,] 0.000 1.00000 0.00000 1.00000
 [9,] 0.000 1.00000 0.00000 1.00000
[10,] 0.000 0.03125 0.00000 0.03125
[11,] 0.000 0.03125 0.00000 0.03125
```

As linhas nesta matriz correspondem às páginas e as colunas correspondem aos seguintes valores:
Coluna 1: O peso como Authority não normalizado.
Coluna 2: O peso como Hub não normalizado.
Coluna 3: O peso como Authority normalizado (este é o valor que interessa).
Coluna 4: O peso como Hub normalizado (este é o valor que interessa).

Daqui podemos concluir que a página com *Authority* mais elevado é a S2 (índice 9 na lista de urlsS base), ou seja, `"www.liacc.up.pt/NCC/Projs/projectos_10.html"` e as páginas com *Hub* mais elevado são as X3 a X6, ou seja, www.google.com/webhp?hl=en, www.liacc.up.pt/NCC/Projs/projectos_6.html, www.liacc.up.pt/NCC/Projs/projectos.html e www.google.com/.

# 3 Aplicação do Programa ao Tópico 'www.microsoft.com'

Dado que o processo de utilização já foi descrito no tópico anterior, vou limitar-me a apresentar os vários resultados e a comentar os valores finais.

```
> urlsS<- obterS('www.microsoft.com')
> urlsS
[8]www.microsoft.com/mac/downloads.aspx?pid=download&location=/mac/do
wnload/misc/msn_401.xml&secid=35&ssid=4&flgnosysreq=True"
[9] "www.microsoft.com/technet/traincert/virtuallab/mom.mspx"
[10] "www.microsoft.com/mac/downloads.aspx"
[11] "www.microsoft.com/technet/archive/IIS3/iischp5.mspx"
[12] "www.microsoft.com/latvija/partners/partnerprogram/enroll.asp"
[13] "www.microsoft.com/communities/webcasts/default.mspx"
[14]"www.microsoft.com/emea/refurbishers/fr/marsSupplyTo.aspx?country
=Malte"
[15] "www.microsoft.com/BusinessSolutions/navision/community.mspx"
[16] "www.microsoft.com/miserver/"
```

```
[17]"www.microsoft.com/windows2000/es/professional/help/toc/dl/toc_850.asp"
```

**> xyIN<-inLinks(urlsS[14..16]) retornou 16 URLS**
**> xyOUT<-outLinks(urlsS[14..16]) retornou 123 URLS**

Dada a limitação de espaço e como os URLs obtidos são muitos, optei por apenas indicar o valor obtido. Contrariamente ao exemplo anterior, já foi possível obter resultados dos *Out Links*. Pela mesma razão e dado o tamanho do Grafo/Matriz de adjacência (142 linhas e 142 colunas) optei por não o apresentar, passando a apresentar a matriz com os resultados do HITS:

**> hits(xyALL,its=10)**

```
> hits(xyALL,its=2)
            [,1]      [,2]       [,3]       [,4]
  [1,] 0.03278689 1.9166667 0.03278689 0.37704918
  [2,] 0.03278689 5.0833333 0.03278689 1.00000000
  [3,] 0.19672131 3.2500000 0.19672131 0.63934426
  [4,] 0.37704918 0.1666667 0.37704918 0.03278689
  [5,] 0.37704918 0.1666667 0.37704918 0.03278689
  [6,] 0.37704918 0.1666667 0.37704918 0.03278689
  [7,] 0.37704918 0.1666667 0.37704918 0.03278689
  [8,] 0.37704918 1.0000000 0.37704918 0.19672131
  [9,] 0.37704918 1.0000000 0.37704918 0.19672131
 [10,] 0.37704918 1.0000000 0.37704918 0.19672131
 [11,] 0.37704918 1.0000000 0.37704918 0.19672131
 [12,] 0.37704918 1.0000000 0.37704918 0.19672131
 [13,] 0.37704918 1.0000000 0.37704918 0.19672131
 [14,] 0.37704918 1.0000000 0.37704918 0.19672131
 [15,] 0.37704918 1.0000000 0.37704918 0.19672131
 [16,] 0.37704918 1.0000000 0.37704918 0.19672131
 [17,] 0.37704918 1.0000000 0.37704918 0.19672131
 [18,] 0.37704918 1.0000000 0.37704918 0.19672131
 [19,] 0.37704918 1.0000000 0.37704918 0.19672131
 [20,] 0.37704918 0.0000000 0.37704918 0.00000000
 [21,] 0.37704918 0.0000000 0.37704918 0.00000000
 [22,] 0.37704918 0.0000000 0.37704918 0.00000000
 [23,] 0.37704918 0.0000000 0.37704918 0.00000000
 [24,] 0.37704918 0.0000000 0.37704918 0.00000000
 [25,] 0.37704918 0.0000000 0.37704918 0.00000000
 [26,] 0.37704918 0.0000000 0.37704918 0.00000000
 [27,] 1.00000000 0.0000000 1.00000000 0.00000000
 [28,] 1.00000000 0.0000000 1.00000000 0.00000000
 [29,] 1.00000000 0.0000000 1.00000000 0.00000000
```

A matriz de HITS apresentada é parcial mas podemos verificar que boas páginas *Authority* são as que correspondem aos índices 27, 28 ou 29 e que são, respectivamente:
/emea/refurbishers/fr/receivingDonations.mspx,
/emea/refurbishers/fr/frequentlyAskedQuestions.mspx e
/emea/refurbishers/fr/freshStart.mspx.

Boas páginas *HUB* são as correspondentes aos índices 2 e 3, respectivamente:
www.microsoft.com/BusinessSolutions/navision/community.mspx e
www.microsoft.com/miserver/.

# 4 Comentários Finais

O principal problema do programa em R que envio em anexo, prende-se com o facto de, devido aos erros de *Parsing* e ao facto de eu não saber como os "contornar", ter tido que executar grupos de instruções em separado em vez de utilizar a função que criei para o efeito (INICIAR('www.microsoft.com') ) e que permitiria executar o programa mais facilmente.
Para além disso existem outros problemas com os endereços URL devolvidos pela função ObterS que não deveria incluir os URL's que são específicos do *Google*.

## Referências

[1] J. M. Kleinberg – Authoritative Sources in a Hyperlinked Environment, In Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 1998.

[2] G. Chang, M. Healey, J. McHugh, J. Wang – Mining The World Wide Web, An Information Search Approach, 2001.

[3] A. Jorge – Análise da Estrutura da Web, Sebenta da Cadeira de ECI, 2005.

[4] The R Project for Statistical Computing. http://www.r-project.org.

## Anexos

Ficheiro **ECI_TRAB_AJ1_AntonioCastro.R** com o código R utilizado neste trabalho.