Universidade do Minho
Escola de Engenharia

Alice Monteiro Marques

Identificação de Perfis de Utilização Web Baseada em Clickstreams

Novembro de 2009

Identificação de Perfis de Utilização Web Baseada em Clickstreams
Alice Monteiro Marques
UMinho | 2009

Universidade do Minho
Escola de Engenharia

Alice Monteiro Marques

Identificação de Perfis de Utilização Web Baseada em Clickstreams

Dissertação de Mestrado
Mestrado em Informática

Trabalho efectuado sob a orientação do
Professor Doutor Orlando Manuel de Oliveira Belo

Novembro de 2009

Aos meus pais, e á minha irmã

Agradecimentos

Queria deixar expressos os meus agradecimentos a todos aqueles que contribuíram para a realização desta tese, em particular:

- Ao meu orientador Professor Doutor Orlando Manuel de Oliveira Belo, que pelo seu incentivo, e que e através das suas criticas e sugestões, tornou possível o elaboração desta dissertação.
- Ao Nélio Guimarães por todas as horas que passamos a discutir os problemas desta dissertação.
- Aos gabinete de sistemas de informação(GSI) pela disponibilização dos ficheiros de *log*, utilizados no caso de estudo.
- A Anália Lourenço a ao Eurico Borges pela disponibilização dos seus trabalhos de doutoramento e mestrado, que em muita ajudaram na recolha de bibliografia, e na organização deste dissertação.
- A minha família e amigos pelo apoio durante o processo de realização desta tese.
- Por fim, aos meus pais e á minha irmã pelo seu incondicional apoio e compreensão, e sem os quais nada disto teria sido possível.

Resumo

Identificação de Perfis de Utilização Web Baseada em Clickstreams

A Web é cada vez mais um mercado apelativo para as grandes organizações, seja como forma de divulgarem as suas actividades ou até mesmo ampliar os seus negócios. Contudo, sendo a Web um mercado tão alargado e acessível, facilmente um utilizador pode saltar para um *site* da concorrência, caso não encontre aquilo que procura ou caso algum aspecto no *site* não seja do seu agrado. Torna-se por isso essencial conhecer que tipo de utilizadores visitam os *sites* Web, de forma a garantir que os produtos, serviços e informações que procuram neles estejam disponíveis. Cada vez mais as organizações optam por analisar questões relativas à utilização de *sites* Web, de forma a conseguirem responder a questões sobre a utilização do *site*: tais como, qual a página mais acedida, qual a página mais utilizada como página de entrada, entre muitas outras. O estabelecimento de perfis de identificação tem como principal objectivo agrupar os utilizadores de um dado *site* Web, em grupos de utilizadores com características de utilização e interesses semelhantes. Com os perfis estabelecidos, pretende-se caracterizar o tipo de utilizadores desse *site*, e, assim, melhorar os serviços fornecidos aos utilizadores, garantido que os produtos, informações e serviços que eles procuram se encontram disponíveis, de acordo com as suas expectativas. Nesta dissertação, são apresentadas algumas das técnicas de mineração de dados utilizadas para identificação de perfis de utilização Web. São também descritas todas as etapas do processo de mineração de dados de utilização Web, desde a identificação das fontes até a geração dos perfis. Assim, neste trabalho, começamos por estudar as fontes de dados utilizadas em técnicas de mineração de dados Web bem como algumas técnicas de transformações de dados utilizadas para preparar o conjunto de dados para a aplicação de técnicas de mineração de dados.

Posteriormente estudámos alguns dos algoritmos de mineração de dados mais utilizados na identificação de perfis de utilização Web. Por fim, depois de analisado todo o processo de identificação de perfis, aplicámos as técnicas estudadas a um caso de estudo previamente estabelecido e com grande aplicação prática. Os resultados obtidos da aplicação de técnicas de identificação de perfis ao caso prático, definiram quatro grupos de utilizadores distintos para o *site* Web em análise. Através da análise de cada um dos grupos obtidos foi possível implementar algumas melhorias, de modo a melhorar os serviços fornecidos aos utilizadores.

Palavras-Chave: *Clickstreams*, *WebLogs*, Agentes Automáticos *Data Webhouse, Data Warehouse*, Mineração de Dados, Mineração de Dados de Utilização *Web*, *Cluster*, Regras de Associação, Padrões Sequenciais, Cadeias de *Markov,* Extracção, Transformação, Integração, Sessões, Perfis*,* Extracção, Transformação, Integração, Web, Site, Página.

Abstract

Identification of Web Usage Profiles Based on Clickstreams

The Web increasingly presents itself as a tasty* market for big organizations, as a way to show their activities or even to amplify their business. However, the Web is a large and accessible market, so a user may easily go to a site of a competing company if he doesn't find what he is looking for or if there is something on the site that isn't of his liking. So it's essential to know the type of users that visit the Web sites, to warranty that the products, services and information that they are looking for are available. With increasing frequency we see organizations choosing to analyse Web site utilization issues, in order to answer site utilization problems, for example the most accessed page, the page mostly used as an entry page, etc. The establishment of identification profiles has the main objective of agglomerating a Web site's users in user groups with the same user characteristics and interests. With established profiles, we want to categorize the site's user types, which would lead to better services for these users, giving the warranty of available products, information and services for the users that seek them, in accordance to their expectations. In this thesis, we present some data mining techniques used to identify Web usage profiles. We also describe all web usage mining process steps, from source identification to profile generation. In this work, we start studying data sources used on Web data mining techniques, as well as some data transformation techniques used to prepare the datasets on which we will apply data mining techniques. Following this, we will study some of the most used data mining algorithms used in web using profile identification. Finally, after analysing the profile identification process, we will apply the studied techniques on a study case previously established. The results of the application of identification profiles techniques to the study case defined four distinct sets of

users to the Web site. Through the analysis of each group obtained is possible to implement some improvements in order to improve the services provided to the users.

Keywords: Clickstreams, Log Files, Crawlers, Data Webhouse, Data Warehouse, Data Mining, Web Mining, Web Usage Mining, Clusters, Association Rules, Sequential Patterns, Markov Chains, Extraction, Transformation, Integration, Sessions, Profiles, Web, Site, Page.

Índice

Introdução 1

1.1 A World Wide Web 1
1.2 Perfis de Utilização Web 4
1.3 Motivação e Objectivos 6
1.4 Estrutura da Dissertação 8

Aplicação de Técnicas de Mineração de dados à Web 11

2.1 Mineração de dados 11
2.2 Mineração de dados Web 12
2.2.1 Mineração de Dados de Conteúdo e de Estrutura Web 15
2.2.2 Mineração de Dados de Utilização Web 16

Preparação de Dados na Aplicação de Técnicas de Mineração de Dados de Utilização Web 19

3.1 Processamento de *Clickstreams* 19
3.2 Fontes e Tipos de Dados 20
3.2.1 Dados de Utilização Web 20
3.2.2 Outras Fontes de dados 24
3.3 Junção e Limpeza de Dados 25
3.4 Identificação de Utilizadores 26
3.5 Identificação de Agentes Automáticos 29
3.6 Identificação de Sessões 32
3.6.1 Estratégias Pró-activas 33
3.6.2 Estratégias Reactivas 34

3.7 Identificação de Páginas 37
3.8 Reconstrução de Caminhos 37
3.9 Integração dos Dados 40

Descoberta e Análise de Padrões 43

4.1 Modelação de Dados 43
4.2 Análise Estatística 45
4.3 Clusters 46
4.4 Regras de Associação 50
4.5 Padrões Sequenciais 53
4.6 Classificação 58

Avaliação Experimental 61

5.1 Descrição do Site Alvo de Estudo 61
5.2 As Fontes de Dados 65
5.3 Modelação Dimensional 68
5.4 O Processo de Extracção, Transformação e Integração dos Dados 72
5.5 Aplicação de Técnicas de Mineração de Dados 76
5.5.1 Conjunto de Treino 79
5.5.2 Algoritmos de Clustering 81
5.5.3 Algoritmos de Regras de Associação 85
5.5.4 Cadeias de Markov 88
5.6 Considerações Finais Acerca dos Resultados Obtidos 91

Conclusões e Trabalho Futuro 95

6.1 Síntese do problema 95
6.2 Considerações à Abordagem Desenvolvida 96
6.3 Comentários à Avaliação Prática 98
6.4 Contributos e Limitações desta Dissertação 99
6.5 Trabalho Futuro 100

Bibliografia 103

Referências WWW 111

Índice de Figuras

Figura 1.1: Evolução do crescimento do número de *sites* na Web [WWW 2]..... 3
Figura 2.1: Descrição genérica da aplicação de mineração de dados de utilização Web 17
Figura 3.1: Exemplo de identificação de Utilizadores usando IP + Agent 28
Figura 3.2: Exemplo da identificação de sessões usando a heurística H1 35
Figura 3.3: Exemplo da identificação de sessões usando a heurística H2 35
Figura 3.4: Exemplo da identificação de sessões usando a heurística H-REf..... 36
Figura 3.5: Estrutura de um *site* Web e páginas visitadas por um utilizador..... 39
Figura 4.1: Construção de uma matriz de transacções 44
Figura 4.2: Exemplo de quatro clusters de dados 47
Figura 4.3: Separação do conjunto em três cluster 48
Figura 4.4: Geração de perfis de utilização através de clusters 50
Figura 4.5 : Visualização de uma cadeia de Markov 56
Figura 4.6: Árvore de caminhos mais frequentes suporte = 0.1 e confiança = 0.4 57
Figura 5.1: Página principal do *site* do *RepositoriUM*..... 63
Figura 5.2 Distribuição de pedidos e sessões por hora 64
Figura 5.3: Excerto de um ficheiro de logs..... 65
Figura 5.4: Ficheiro de IPS 66
Figura 5.5: Tabela com motores de pesquisa conhecidos..... 67
Figura 5.6: Ttabela com as extensões dos ficheiros. 67
Figura 5.7: Tabela com os *logs* conhecidos..... 68
Figura 5.8 : *Datamart* para sessões Web 68
Figura 5.9: Esquema global do processo de ETI 73

Figura 5.10: P1- Extracção e carregamento dos ficheiros IPs..... 73
Figura 5.11: Recolha e carregamento dos ficheiros de *Log*..... 74
Figura 5.12: Povoamento das várias dimensões 75
Figura 5.13: Povoamento da tabela de factos e da tabela ponte. 76
Figura 5.14: Limpeza de tabelas área de retenção. 76
Figura 5.15: Várias fases da metodologia *CRISP-DM* [WWW 15]..... 77
Figura 5.16:Percentagem de pedidos por página em cada cluster. 82
Figura 5.17: Representação de uma cadeia de *Markov* 89

Índice de Tabelas

Tabela 1.1: Estatísticas disponibilizadas pelo Internet World Statistcs[WWW 1] sobre a utilização de Internet em termos mundiais 2
Tabela 3.1: Exemplo de um ficheiro de *log* no formato CLF 21
Tabela 3.2: Exemplo de um ficheiro de *log* no formato ECLF 22
Tabela 3.3: Campos de CLF e do ECLF 22
Tabela 3.4: Exemplo de um ficheiro de *logs* no formato ELF 24
Tabela 3.5: Vantagens e Desvantagens dos métodos de Detecção de Agentes Automáticos 31
Tabela 4.1: Conjunto de transacções Web 55
Tabela 4.2: Caminhos mais frequentes com suporte 0.1 e confiança 0.4 57
Tabela 5.1: Caracterização da Tabela de Factos de sessões Web 69
Tabela 5.2: Caracterização da dimensão Tempo 70
Tabela 5.3: Caracterização da dimensão Data 70
Tabela 5.4: : Caracterização da dimensão computadorUtilizador 71
Tabela 5.5: Caracterização da dimensão *Agent* 71
Tabela 5.6: Caracterização da dimensão Referente 71
Tabela 5.7: Caracterização da dimensão *Request* 72
Tabela 5.8: Caracterização da Tabela Ponte 72
Tabela 5.9: Perfil de utilização associado ao cluster 0 83
Tabela 5.10: Perfil de utilização associado ao cluster 1 83
Tabela 5.11: Perfil de utilização associado ao cluster 2 84
Tabela 5.12: : Perfil de utilização associado ao cluster 3 85
Tabela 5.13: *Itemsets* mais frequentes com suporte mínimo de 5% 86

Tabela 5.14: Resultado das regras de associação 87
Tabela 5.15: Caminhos mais frequentes probabilidade 10% e suporte 1%. 90

Lista de Siglas e Acrónimos

AR	*Área de Retenção*
CERN	*Center for European Nuclear Research*
CRISP-DM	*CRoss-Industry Standard Process for Data Mining*
CLF	*NCSA Common Log Format*
DW	*Data Warehouse*
CPU	*Central Processing Unit*
Dweb	*Data Webhouse*
ETI	*Extracção, Transformação e Integração*
ETL	*Extraction, Transformation and Loading*
ECLF	*NCSA Extended Common Log Format*
ELF	*W3C Extended Log File Format*
FAQ	*Frequently Asked Questions*
GB	*Giga Bytes*
GMT	*Greenwich Mean Time*
HTTP	*Hypertext Transfer Protocol*
HTML	*HyperText Markup Language*
IP	*Internet Protocol*
ISP	*Internet Service Provider*
MB	*Mega Bytes*
NCSA	*National Center for Supercomputing Applications*
OLAP	*Online Analytical Processing*
RAM	*Random Access Memory?*

SSIS	*SQL Server Integration Services*
UM	*Universidade do Minho*
URI	*Uniform Resource Identifier*
W3C	*World Wide Web Consortium*
WWW	*World Wide Web*
XML	*Extensible Markup Language*

Capítulo 1

Introdução

1.1 A World Wide Web

A *World Wide Web (WWW)*, uma das maiores invenções dos últimos anos, foi apresentada no CERN em 1992. Inicialmente, tinha como principal objectivo a partilha de informação entre os seus diversos utilizadores. No entanto, a sua simplicidade e potencialidades abriram novos horizontes em termos de partilha de informação. Apesar do seu objectivo inicial estar confinado à comunidade específica do CERN, este foi mais tarde expandido à escala global, onde qualquer utilizador, em qualquer parte do mundo, pode partilhar e aceder a informação contida numa rede mundial. Surgiu assim, o conceito de Internet ou, simplesmente, o de Web.

É do conhecimento geral que, ao longo dos últimos anos, o número de utilizadores Web tem aumentado de forma exponencial. Este crescimento, deve-se tanto à facilidade com que qualquer pessoa pode aceder *a* Web*,* como ao aumento e diversidade de serviços disponibilizados na Internet. Hoje em dia é possível utilizar a Internet para realizar uma grande diversidade de actividades, podendo estas irem desde a simples comunicação com outros utilizadores, até ao encomendar de produtos online, passando por enviar e receber *emails*, pesquisar informações, tirar cursos *on-line*, pagar contas ou consultar extractos, entre outras. Existem mesmo alguns serviços que apenas se encontram disponíveis na Internet. Por este motivo cada vez mais pessoas a usam no seu dia-a-dia para realizar um sem número de tarefas. Este factor, torna a Web um

mercado muito apetecível para as empresas, uma vez que estas vêem nela uma boa maneira de publicitar os seus produtos, ou até mesmo como uma nova forma de expandir o seu negócio. Hoje em dia, uma empresa que não possua um *site Web* pode ser preterida em relação a uma empresa do mesmo ramo com representação online. Isto acontece pois os utilizadores antes de recorrer aos serviços de alguma empresa tentam primeiro, pesquisar na internet algum tipo de informação que os possa ajudar a decidir sobre o melhor local para encontrar aquilo que procuram. No sentido de aproveitarem este mercado, e dele retirarem vantagens competitivas, cada vez mais as empresas apostam no desenvolvimento de *sites* Web como forma de publicitar os seus serviços.

Região	População (2008)	População (% do mundo)	Utilizadores de Internet	%Utilização de Internet	%Utilização no Mundo	Crescimento da utilização 2000-2008
África	975,330,899	14.5%	54,171,500	5.6%	3.4 %	1,100.0 %
Ásia	3,780,819,792	56.3%	657,170,816	17.4%	41.2 %	474.9 %
Europa	803,903,540	11.9%	S393,373,398	48.9%	24.6 %	274.3 %
Médio Oriente	196,767,614	2.9%	45,861,346	23.3%	2.9 %	1,296.2 %
América do Norte	337,572,949	5%	251,290,489	74.4 %	15.7 %	132.5 %
América Latina e Caraíbas	581,249,892	8.7%	173,619,140	29.9 %	10.9 %	860.9 %
Oceânia e Austrália	34,384,384	0.5%	20,783,419	60.4 %	1.3 %	172.7 %
Total	6,710,029,070	100%	1,596,270,108	23.8%	100.0 %	342.2 %

Tabela 1.1: Estatísticas disponibilizadas pelo Internet World Statistcs[WWW 1] sobre a utilização de Internet em termos mundiais

Sendo a Web um espaço a que qualquer pessoa tem facilmente acesso, seja para procurar ou disponibilizar informação, esta tornou-se num dos maiores repositórios de informação a nível mundial. Esta característica leva a que sempre que se faz uma pesquisa na Internet sejam encontrados vários *site*s relacionados com esse assunto. Esta enorme variedade de informação, permite aos utilizadores escolher qual o *site* que preferem consultar. Caso o *site* escolhido por algum motivo não agrade ao utilizador, é possível mudar de *site* com apenas um simples *click*.

Esta facilidade em mudar de *site*, juntamente com a grande quantidade de *sites* dedicados ao mesmo assunto, faz com que na Web exista uma forte concorrência entre as diversas organizações. Esta concorrência, verifica-se em especial em *sites* dedicados a comercio electrónico, em que os utilizadores são simultaneamente clientes e interessa por isso mantê-los. Por este motivo cada vez mais as organizações investem recursos na melhoria dos serviços Web fornecidos.

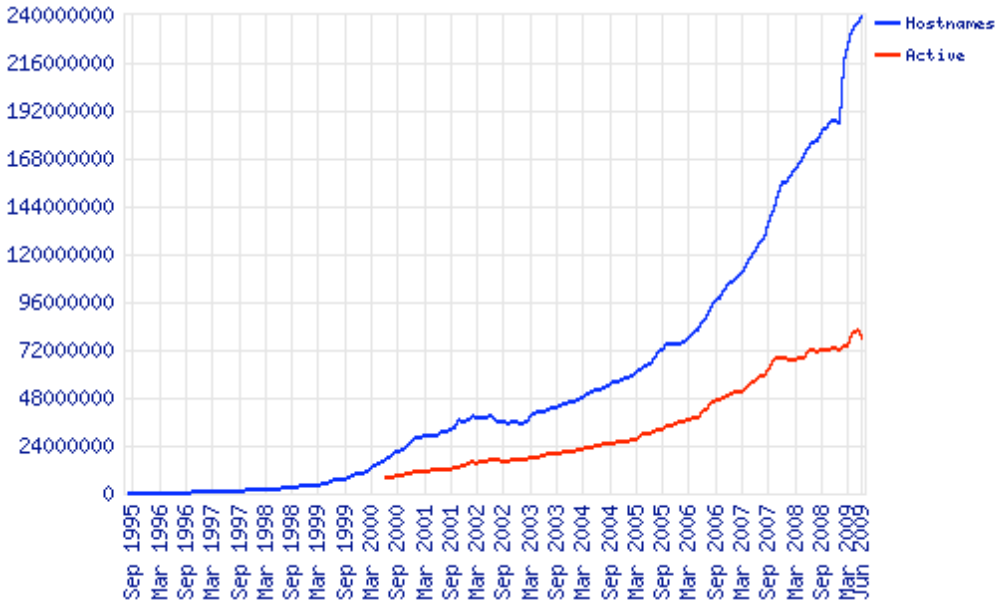


Figura 1.1: Evolução do crescimento do número de *sites* na Web [WWW 2]

Existem vários factores que podem ditar o sucesso ou insucesso de um *site* dos quais podemos destacar dois: a divulgação do *site*, pois se esta não existir muitos utilizadores não vão sequer chegar a saber da existência dele, e o *design*, sendo este um factor muito importante uma vez que é o primeira cartão de visitas de um *site*. Um *design* apelativo, profissional, que transmita confiança aos clientes tem mais probabilidade de ser bem aceite pelo utilizador. Contudo, um bom *design* e uma boa divulgação por si só não garantem o sucesso. Pois caso o *site* não tenha o que o cliente pretende ou demore muito tempo a satisfazer o pedido, o utilizador vai rapidamente desistir e no limite abandonar o próprio *site*. O sucesso de um *site* vai por isso muito para além de questões meramente estéticas. É necessário conhecer, tal como em todos os meios tradicionais de negócio, o tipo de clientes que utiliza o *site*, quais os seus interesses e perspectivas, uma vez que são eles os principais elementos que condicionarão a sua evolução, assim como por influência a evolução da própria organização. Por este motivo, cada vez mais as organizações tentam analisar

questões relacionadas com a forma como os clientes interagem com o *site*, como por exemplo, quais as páginas mais e menos visualizadas, quais as principais páginas de entrada e de saída, qual o país de onde são efectuados mais pedidos, quais as horas de maior utilização, qual a frequência com que um utilizador visita o *site*, entre muitas outras que possam permitir compreender o utilizador e melhorar o *site*. A análise deste tipo de questões vai permitir detectar perfis de utilização dos *sites*. Uma vez estabelecidos os perfis, é possível personalizar o *site* de acordo com as necessidades dos utilizadores, promovendo uma navegação mais fácil e uma melhor capacidade de resposta ás necessidades dos clientes.

1.2 Perfis de Utilização Web

Considerada como um mercado em franca expansão e ao qual recorrem cada vez mais utilizadores, a Web*,* despertou nas empresas um interesse superior, na medida em que perceberam as suas mais valias e que poderiam retirar vantagens competitivas estudando e interpretando o negócio "Web" e o comportamento dos seus utilizadores. Para esta análise de negócios e comportamentos, podem ser aplicadas técnicas de mineração de dados, sobre os dados Web de cada empresa alvo. Este tipo de técnicas são ferramentas de análise bastante úteis e importantes, na medida em que permitem detectar padrões e relacionamentos entre os diferentes dados, que até então passavam despercebidos, não tendo sido por isso considerados. Estas técnicas possibilitam ainda, um maior conhecimento sobre os *sites* Web de cada empresa, conhecimento esse que vai permitir estabelecer e caracterizar perfis de utilização do *site*. A aplicação de técnicas de mineração de dados a informação proveniente da Web*,* é chamada mineração de dados Web*.*

Os dados recolhidos da Web podem ser classificados, essencialmente, em três categorias: dados sobre o conteúdo do *site* – dados referentes aos conteúdos publicados no *site*, conteúdos estes que podem ser texto, gráficos, entre outros; dados de estrutura – dados referentes a estrutura do *site* e à sua organização; e, por último, dados de utilização do *site* - dados recolhidos da interacção dos utilizadores com o *site*. Como existem três tipos diferentes de dados Web, também as técnicas de mineração de dados Web, um pouco à semelhança dos dados recolhidos, se dividem em três áreas diferentes: mineração de dados sobre o conteúdo Web*,* mineração de dados de estrutura Web e, por fim, mineração de dados de utilização Web*.*

A mineração de dados de conteúdo Web*,* consiste no desenvolvimento de técnicas que procuram encontrar padrões no conteúdo de um *site* Web. Quanto à Mineração de dados da estrutura Web permite desenvolver técnicas que retiram conhecimento dos *hiperlinks* dos *sites*, quer estes sejam externos ou internos ou *site*. Por fim a mineração de dados de utilização Web, foca-se no desenvolvimento de técnicas, que visam encontrar padrões de comportamento dos utilizadores de um *site*.

As técnicas de mineração de dados de utilização Web, são úteis tanto para o administrador do *site* como para os seus utilizadores. Conhecendo o tipo de utilizadores que frequentam o *site*, o administrador pode melhorar os serviços fornecidos, de acordo com as suas preferências. Essas melhorias podem ir desde a personalização de páginas Web de acordo com os utilizadores, disponibilizando o que estes mais procuram, até à promoção de conteúdos específicos. A promoção de conteúdos, passa por colocar produtos, ou informações sobre produtos, em locais do *site* no qual sejam facilmente visíveis aos clientes. A personalização de *sites* Web visa aumentar o grau de satisfação dos seus clientes, aumentando assim o tempo que estes passam no *site* e a probabilidade de no futuro voltar a visitá-lo Esta pode ser feita através da análise do comportamento de utilizadores que tenham visitado o *site* anteriormente. Através desta análise é estabelecer perfis de utilização do *site*. Com os perfis de utilização estabelecidos é possível efectuar melhorias no *site* Web, de modo a ir de encontro ás necessidades dos utilizadores. Estas melhorias podem ir desde disponibilizar novos conteúdos de acordo com as preferências dos utilizadores, até a colocação de *hiperlinks* entre as páginas mais visitadas por um determinado perfil de utilização, facilitando assim a procura de conteúdos por parte dos utilizadores. Os perfis de utilização podem também ser utilizados em campanhas *marketing* para promover novos produtos, colocando referências aos novos produtos nas páginas mais visitadas pelos perfis em que os produtos se encaixam. A análise de comportamentos, permite também descobrir quais as páginas, em que mais utilizadores abandonam os *sites*. Esta informação pode ajudar os administradores, a perceber quais os motivos que levam os utilizadores a abandonar o *site* em determinada página Web*, e* com essa informação tentar melhorar o conteúdo/*design* de modo a evitar que os utilizadores saiam do *site* depois de consultar a página.

Quando os utilizadores Web interagem com o *site*, todos os *clicks* executados são guardados num ficheiro de *log*, que num *site* de tamanho médio pode guardar vários megabytes de informação por dia. Os dados de *log* são guardados em formato de texto, em que cada linha vai representar um

pedido feito por um utilizador. Actualmente, existem no mercado várias ferramentas de análise de ficheiros de *log* [WWW 3], [WWW 4]. Contudo, estas ferramentas têm uma capacidade de análise um pouco limitada, uma vez que, na generalidade, apenas disponibilizam dados estatísticos sobre os acessos ao *site*. Presentemente, existem várias pesquisas na área de mineração de dados de utilização Web que tentam retirar toda a informação possível existente nos ficheiros de *log*. Uma das abordagens utilizadas na aplicação de técnicas de mineração de dados de utilização Web, é mapear as informações existentes num ficheiro de *logs* para um *Data Warehouse* (DW) - neste caso especifico é designado por *Data Webhouse* (DWeb) - e seguidamente aplicar técnicas de mineração de dados, como cluster ou regras de associação, sobre os dados guardados como forma de conseguir extrair padrões.

1.3 Motivação e Objectivos

O enorme volume de informação que é inserido e mantido ao longo do tempo nos ficheiro de *log* dos servidores Web das empresas, usualmente não serve para nada. Na realidade, a maioria das organizações que mantêm *sites* nem sabe da existência desses ficheiros, nem das enormes vantagens que a exploração dessa informação lhes poderia trazer em termos do conhecimento sobre os seus próprios clientes. Para contrariar um pouco este enorme desperdício, nesta dissertação, pensámos em explorar essa enorme quantidade de dados, de forma a transformá-la em conhecimento útil para um dado organismo que teve (ou não) a preocupação de os guardar. Esse conhecimento poderá ser utilizado para melhorar as actividades de negócio suportadas pelos *sites* em actividade, procurando ir ao encontro das necessidades dos utilizadores que frequentam esses *sites* e, consequentemente, aumentar o seu grau de satisfação. Este processo, porém, não é fácil. O trabalho de definição de perfis de utilização – padrões de comportamento dos utilizadores Web – é uma actividade sistemática que envolve a exploração exaustiva de todos os registos de *clickstream*, transformando-os e analisando-os com as técnicas mais recentes de mineração de dados.

Actualmente existem várias ferramentas comerciais de análise de ficheiros de *log.* Contudo estas ferramentas apresentam uma capacidade de análise limitada, fornecendo apenas informações estatísticas sobre os acessos ao *site.* No entanto, existem vários trabalhos de investigação que se dedicam à exploração e análise dos dados de *cikstreams*. Estes trabalhos enquadram-se em

diferentes áreas de aplicação, e tem diferentes objectivos. Algumas das áreas em que a exploração de dados *clikstream*, se torna bastante útil são:

- **Personalização.** Esta consiste em personalizar um *site* de acordo com as necessidades dos utilizadores. É uma das áreas em que mais projectos foram desenvolvidos em mineração de dados de utilização Web. É neste tipo de técnicas, que assentam os sistemas de recomendação, uma vez que o produto recomendado depende de utilizador para utilizador. Para esta área, o estabelecimento de perfis de utilização, através da análise do comportamento de utilizadores anteriores, é bastante útil, uma vez que a personalização pode ser feita de acordo com o perfil em que o utilizador se encaixa. Algumas das mais conhecidas pesquisas desenvolvidas na área de personalização utilizando mineração de dados Web podem ser consultados em [Joachins et al. 97], [Ngu and Wu 97], [Lieberman 95],[Mobasher et al 99].

- **Adaptação de *Sites*.** O aspecto de um *site*, quer em termos de conteúdo quer em termos de estrutura, é um ponto crucial para o sucesso de um *site* Web. A mineração e dados de utilização Web providencia informações sobre as visitas anteriores dos clientes, dando aos *designers* informações sobre a direcção que uma futura reestruturação do *site* deve seguir. Em [Perkowitz and Etzioni 99],[Perkowitz and Etzioni 98], [Perkowitz and Etzioni 00] podemos ver dois projecto de adaptação de *sites*, focado na reestruturação automática de *sites* com base em padrões de utilização descobertos.

- ***Bussiness Inteligence*.** Ajuda a organização nos processos de tomada de decisão dando respostas a questões relacionadas com o negócio através da análise dos produto Web. A informação de como um *site* esta a ser utilizado pelos seus clientes é útil para os processos de tomada de decisão de um empresa. Em [Buchner and Mulvenna 98] é-nos apresentado um processo de extracção de conhecimento que visa extrair informação de *sites* Web.

- **Caracterização de Utilizadores.** Permite prever qual o comportamento futuro de um utilizador através da análise da estratégia de navegação em visitas anteriores desse mesmo utilizador, ou de utilizadores com perfil idêntico.

- **Melhoria do Sistema.** Analisar a performance dos serviços fornecidos, por exemplo através da análise de tráfego, vai permitir melhorar os serviços fornecidos pelo *site*. Isto

pode ser feito investindo mais recursos nos serviços mais utilizados e retirando recursos dos serviços menos utilizados. Tornando assim os serviços mais rápidos e mais interessantes para o utilizador. Alguns dos trabalhos realizados na área de melhoria de sistemas podem ser consultados em [Almeida et al 96],[Schechter et al 98].

Nesta dissertação temos como objectivo aproveitar a informação contida nos ficheiros de *log* e, através dela, construir um modelo que permita identificar perfis de utilização Web. De forma mais concreta, pretendemos:

1. Justificar a necessidade de identificar perfis de utilização Web, e mostrar quais as vantagens da aplicação deste tipo de técnicas.

2. Estudar todos os passos da aplicação de técnicas de mineração de dados a dados Web, mais propriamente a dados de utilização Web.

3. Estudar as várias técnicas de pré-processamento de dados, ou seja as técnicas de transformação de dados necessárias para preparar os dados para a aplicação dos diversos algoritmos de mineração de dados.

4. Analisar quais as técnicas de mineração de dados mais utilizadas na implementação de técnicas de mineração de dados de utilização Web, assim como estudar quais as suas principais áreas de aplicação.

5. Aplicar as várias técnicas de estudadas ao longo da dissertação a um caso de estudo real e analisar os resultados obtidos.

O objectivo desta dissertação não é disponibilizar um guia para a implementação de um modelo para identificação de perfis, mas sim apresentar quais os principais algoritmos utilizados para a criação de perfis e suas principais diferenças. A escolha dos algoritmos utilizados está dependente, obviamente, do tipo de *site* em análise, pelo que não existe uma receita única para a identificação de perfis Web.

1.4 Estrutura da Dissertação

Além do presente capítulo, em que fazemos uma breve introdução ao processo de identificação de perfis de utilização Web, explicando alguns dos conceitos envolvidos, a presente tese integra mais cinco capítulos, que estão organizados da seguinte forma:

- O Capítulo 2 começa por fazer uma pequena introdução às técnicas de mineração de dados. De seguida são apresentados alguns conceitos de mineração de dados Web assim como algumas das suas principais características. No final, é feita uma breve apresentação dos conceitos de mineração de dados de conteúdo Web, mineração de dados de estrutura Web e mineração de dados de utilização Web. Refira-se a especial atenção dada à mineração de dados de utilização Web uma vez que esta é a área base do desenvolvimento desta dissertação.
- O Capítulo 3 enumera as principais etapas do processamento de dados para a aplicação de técnicas de mineração de dados Web. No início faz-se uma descrição das principais fontes de dados, dando especial ênfase aos dados de *clickstream.* Depois são descritas as principais etapas do processo de transformação dos dados, tais como: filtragem dos dados, identificação de sessões, identificação de agentes automáticos entre outros. Por fim é descrito o processo de integração dos dados depois de processados os *logs* num *DWeb*, *data webhouse*.
- No Capítulo 4 é feita a descrição das principais técnicas de mineração de dados aplicadas a Web. Aqui, faz-se a descrição do processo de modelação dos dados para a aplicação das técnicas, sendo de seguida apresentada uma pequena descrição de cada uma das técnicas, seguida das vantagens e desvantagens da sua aplicação a dados Web. As técnicas apresentadas neste capítulo são: *clusters,* regras de associação, padrões sequenciais (cadeias de *markov*), análise estatística e, por fim, classificação.
- O Capítulo 5 apresenta um caso de estudo sobre o qual vão ser aplicadas as técnicas apresentadas anteriormente. Começa-se por descrever o *site* sobre o qual foi feita a análise e, de seguida, são apresentadas as fontes de dados utilizadas para aplicação de técnicas de mineração dos dados. Posteriormente é apresentada a estrutura do *Dweb* utilizado para integrar os dados do *site*, seguida de uma pequena explicação do processo de extracção, transformação e integração dos dados. Seguidamente, é feita a aplicação de técnicas de mineração de dados, onde é descrito o conjunto de treino e são avaliados os resultados da aplicação das várias técnicas de mineração utilizadas. Por fim apresenta-se uma análise comparativa entre os resultados das diversas técnicas aplicadas e algumas referências ao desempenho do sistema desenvolvido.
- Por último, no Capítulo 6 são apresentadas as principais vantagens, desvantagens e dificuldades que podemos encontrar num processo de desenvolvimento de perfis de

utilização Web, resultantes do estudo e do trabalho realizado ao longa desta dissertação. Adicionalmente, são apresentadas e discutidas algumas questões pertinentes acerca do trabalho realizado nesta dissertação, assim como indicados alguns dos principais trabalhos que podem ser feitos como complemento ao projecto inicialmente estabelecido.

Capítulo 2

Aplicação de Técnicas de Mineração de dados à Web

2.1 Mineração de dados

O processo de mineração de dados, também conhecido por descoberta de conhecimento em base de dados, é uma área de pesquisa focada no desenvolvimento de ferramentas, que procuram descobrir padrões não perceptíveis num grande volume de dados. Os padrões descobertos devem apresentar determinadas características, tais como: serem válidos, inovadores, úteis e compreensíveis.[Fayyad et al 96]. Existem diversas técnicas de mineração de dados com diferentes objectivos e cuja aplicação envolve usualmente diversas áreas de trabalho, nomeadamente: a Estatística, as Bases de Dados, a Inteligência Artificial , os *Data Warehouses,* entre outros. Das várias técnicas existentes as mais utilizadas são: a classificação, a previsão, o *clustering*, as regras de associação e os padrões sequenciais.

O processo de descoberta de conhecimento pode ser dividido em três fases: preparação dos dados, descoberta de padrões e interpretação dos padrões obtidos. A preparação de dados é a etapa responsável pela recolha, transformação e integração dos dados. É nesta fase que se remove o ruído dos dados, se decide qual a técnica a utilizar para o tratamento de nulos, se removem os atributos desnecessários, se aplicam os vários algoritmos de transformação de dados,

entre outras coisas. Em resumo, no fim desta fase, devemos ter um conjunto de dados sobre o qual vamos aplicar as técnicas de mineração de dados. Na fase seguinte, a descoberta de padrões, o conjunto de dados processado vai ser passado a um algoritmo de mineração de dados que vai tentar descobrir padrões nos dados. Por fim, a última fase, a interpretação dos padrões obtidos é responsável por verificar o sucesso da aplicação das técnicas escolhidas. A análise dos padrões obtido pode sugerir uma melhor filtragem dos dados ou a aplicação de diferentes parâmetros nos algoritmos de mineração de dados - nem todos os padrões descobertos são úteis. Nesta fase identificam-se quais os padrões que realmente trazem vantagens e aqueles que se traduzem em conhecimento considerado como novo. Para uma melhor percepção de todo o processo de mineração de dado*s* recomenda-se a leitura de [Frawley et al 91], [Fayyad et al 96], [Chen and Yu 96]

2.2 Mineração de dados Web

A aplicação de técnicas de mineração de dados a dados provenientes da Web é designada normalmente por mineração de dados Web [Etizione 96], [Mena 99]. A aplicação de técnicas de mineração de dados Web*,* foca-se em torno dos seguintes objectivos: desenvolver técnicas que ajudem a melhorar a qualidade dos *sites* e o grau de satisfação de um utilizador quando este está a navegar num *site* Web.

O desenvolvimento de técnicas que permitam melhorar a qualidade do *site* é devido à necessidade que os administradores têm de ter de algum apoio, no que diz respeito, aos negócios provenientes do *site*. Um dos exemplos em que a mineração de dados Web é útil para os administradores é no desenvolvimento de campanhas de marketing na Web*.* Isto acontece, pois, uma vez conhecendo o comportamento anterior dos utilizadores Web, é mais fácil definir quais os utilizadores que uma determinada campanha pode afectar. Para além de ajudar em campanhas de *marketing*, as técnicas de mineração de dados podem ainda indicar qual o caminho que uma reestruturação do *site* pode (ou deve) seguir, de forma a atingir os objectivos propostos. A análise de informação referente ao tráfico de rede é também importante pois permite determinar quais os pontos onde se deve investir de forma a melhorar a qualidade de serviço fornecido.

Por outro lado, o estudo da aplicação de técnicas de mineração de dados Web também tem como objectivo fornecer ao utilizador final ferramentas e serviços que aumentem a sua satisfação,

durante a navegação no *site*. A aplicação de técnicas de mineração de dados, permite obter conhecimento sobre o comportamento dos utilizadores, que no caso de ser aproveitado pelos administradores do *site* vai trazer benefícios aos seus utilizadores. Com recurso a esta informação, pode-se, por exemplo, criar um sistema de recomendação, prever o comportamento futuro do cliente ou disponibilizar novos conteúdos de acordo com o tipo de utilizadores. A criação de um sistema de recomendação pode ser feita com recurso ao estudo do comportamento de utilizadores anteriores, sendo recomendado a um dado utilizador as páginas que foram visualizadas por utilizadores com perfil de navegação semelhante. A utilização destes sistemas pode levar o utilizador a descobrir artigos que lhe interessam e que inicialmente não tinha conhecimento da sua existência. Os sistemas de previsão tentam estimar qual vai ser a próxima página pedida pelo utilizador, o que pode permite aos administradores melhorar de forma significativa o desempenho do sistema. Uma vez que prevendo qual será a próxima página pedida pelo utilizador é possível disponibilizá-la antecipadamente, depois, quando o utilizador pedir de uma próxima vez essa mesma página o seu carregamento será mais rápido. Por fim, o conhecimento do tipo de utilizadores vai permitir ajustar os conteúdos do *site*, de forma a fornecer aos utilizadores mais conteúdos que satisfaçam o seu interesse.

O rápido crescimento da Web nos últimos tempos torna-a num dos maiores repositório de informação do mundo. Por este motivo a Web apresenta características únicas que fazem com que a aplicação e técnicas de mineração de dados seja um grande desafio. Algumas dessas características apresentados em [Liu 08] são descritas de seguida:

- A quantidade de dados existente actualmente na Web é enorme e continua a crescer. Esta informação abrange diversas áreas pelo que podemos dizer que os dados Web são bastante amplos e diversificados, sendo possível encontrar informação sobre qualquer assunto na Internet.
- Na Web existem dados dos mais variados tipos, tais como: tabelas, páginas Web, blocos de texto ou ficheiros multimédia (filmes, musicas, etc.).
- A informação existente na Web é bastante heterogénea. Devido a diversidade de pessoas que desenham páginas Web*,* páginas diferentes podem abordar o mesmo assunto usando formatos e palavras completamente diferentes. Isto torna a integração de dados provenientes de vários *sites* num desafio interessante e complexo.

- Uma quantidade significativa de dados Web encontra-se interligada. Existem *hiperlinks* tanto entre páginas dentro do mesmo *site*, como entre de páginas de *sites* diferentes. Dentro de um *site*, os *hiperlinks* são utilizados como um mecanismo para organizar a informação. Os *hiperlinks*, que ligam *sites* diferentes, são utilizados para redireccionar os utilizadores para outro *site* que contém informação mais completa sobre o assunto. Normalmente, as páginas que têm muitos *hiperlinks* a redireccionar para elas são páginas onde a qualidade da informação é boa uma vez que várias pessoas confiam nela.
- Os dados Web apresentam bastante ruído. Na informação proveniente da Web existem duas fontes principais de ruído. Primeiro, uma página Web contém vários tipos de informação, como por exemplo o conteúdo principal da página, links ou politicas de *copyright*. Para uma aplicação em particular apenas uma parte destes dados vão ser utilizados, sendo os restantes considerados ruído. A segunda fonte de ruído, é o facto de não haver qualquer controlo da informação publicada na Web. Isto é, qualquer pessoa pode ter uma página na Web e construí-la de forma que bem lhe apetecer (ou quase). Assim sendo, existe uma grande quantidade de dados na Web com baixa qualidade. Na aplicação de técnicas de mineração de dados o ruído deve ser removido.
- A Web não se limita a um simples repositório de informações, também disponibiliza serviços aos seus utilizadores A grande maioria dos *sites* de comércio online, permite que os seus utilizadores façam diversas operações, tais como: comprar produtos, pagar compras entre outros.
- A informação disponibilizada na Web é dinâmica, o que significa que a informação disponibilizada está a ser constantemente alterada. Manter-se informado sobre as mudanças que ocorrem na Web e monitorizá-las é um factor crucial em muitas aplicações de mineração de dados.
- A Web não serve apenas como repositório de informação ou como meio de disponibilizar serviços, mas também como uma meio de interacção entre pessoas, organizações, ou entre pessoas e organizações. Um utilizador pode comunicar através da Web com outra pessoa que se encontra do outro lado do mundo em qualquer altura. A Web permite também que os seus utilizadores expressem em fóruns, *blogs* entre outros, a sua opinião sobre os mais diversos assuntos. Em resumo, podemos assim dizer que a Web é uma sociedade virtual.

Todas estas características apresentam um desafio e oportunidades únicas para a aplicação de técnicas de mineração de dados Web com o objectivo de descobrir padrões. Contudo a aplicação de técnicas de mineração de dados Web não se limita apenas a aplicar técnicas tradicionais de mineração de dados. Devido à riqueza, diversidade e outras características próprias dos dados Web, foram desenvolvidos alguns algoritmos próprios utilizados na mineração de dados Web.

As aplicações de mineração de dados Web, dividem-se essencialmente em três áreas:

- **Informação ou conteúdos –** consiste na aplicação de técnicas de mineração de dados aos dados do conteúdo do *site* Web.
- **Estrutura Web –** aplicação de técnicas de mineração de dados aos dados que constituem a estrutura de um *site*, *hiperlinks.*
- **Comportamento dos utilizadores -** Aplicar técnicas de mineração de dados aos dados referentes a navegação dos utilizadores do *site*.

2.2.1 Mineração de Dados de Conteúdo e de Estrutura Web

O processo de mineração de dados de conteúdo Web tem como objectivo extrair conhecimento a partir do conteúdo de uma página Web. O conteúdo de uma página Web normalmente integra textos, gráficos, imagens, tabelas e blocos de dados. Os objectivos da aplicação deste tipo de técnicas podem ser os mais variados. De referir: classificar páginas Web, agrupar páginas Web e extrair informação do dados de conteúdo. A aplicação de classificação ou de *clustering*, para agrupar as páginas de acordo com o seu conteúdo é similar à aplicação tradicional de técnicas de mineração de dados. Contudo, podemos também descobrir padrões nas páginas Web que permitam extrair informações importantes tais como descrições de produtos, mensagens em fóruns entre outros. Este tipo de aplicação é bem diferente das técnicas tradicionais de mineração de dados. Alguns dos projectos desenvolvidos na área de mineração de dados de conteúdo Web podem ser consultados em [Mendelzon et al. 96], [Za¨ıane and Han, 00], [Craven et al., 98], [Brin 98].

O principal objectivo da aplicação de técnicas de mineração de dados sobre dados da estrutura Web, é extrair conhecimento através dos *hiperlinks* que representam a estrutura dos *sites*. Através da análise dos diversos *hiperlinks* podemos descobrir páginas importantes. Este é, aliás, um dos

factores chave da tecnologia usada pelos motores de pesquisa. As técnicas de mineração de dados tradicionais não abordam este assunto, uma vez na generalidade dos casos não existe uma estrutura nos dados guardados numa base de dados. Para uma visão mais ampla do processo de mineração de dados de estrutura Web consultar [Brin and Page 98],. [Page et al. 98], [Kleinberg 99], [Chakrabarti et al.99].

No contexto da análise de *hiperlinks,* podemos dizer que o número de *hiperlinks* que existe para uma dada página, indica o grau de popularidade dessa página. E o número de *hiperlinks* que sai de uma dada página indica a variedade de assuntos que são abordados nessa página.[Spertus 97]

2.2.2 Mineração de Dados de Utilização Web

A mineração de dados de utilização Web refere-se à descoberta e análise de padrões em dados de *clickstreams* em conjunto com outros dados colectados ou gerados, como resultado da interacção dos utilizadores com *sites* Web. O objectivo da aplicação de técnicas de mineração de dados de utilização Web*,* é capturar padrões de comportamento e perfis de utilização Web, estabelecidos com base na análise da interacção dos utilizadores com o *site* [Cooley et al 97], [Mobasher 06], [Srivastava et al 00]. Os perfis definidos são normalmente conjuntos de recursos, páginas e objectos que são frequentemente acedidos por grupos de utilizadores com interesses semelhantes. Os ficheiros de *log* são a principal fonte de dados utilizada no processo de mineração de dados de utilização Web, estes são gerados automaticamente pelos servidores Web. Eventuais fontes de dados adicionais podem também ser usadas na fase de transformação dos dados.

O processo de mineração de dados de utilização Web está dividida em três fases independentes: o processamento de dados, a descoberta de padrões e a análise de padrões. Na fase de pré-processamento, os dados são recolhidos das várias fontes, sendo depois transformados e integrados num sistema próprio. As principais etapas na transformação de dados são as seguintes: identificação de sessões, identificação de *crawlers*, identificação de páginas, reconstrução de caminhos, entre outros. Depois de transformados os dados, passamos à fase de descoberta de padrões, onde vão ser aplicados os algoritmos de mineração de dados. As técnicas de mineração de dados mais utilizadas em mineração de dados de utilização Web são: o *clustering*, as regras de associação, os padrões sequenciais, a classificação e a previsão. Por fim, depois de gerados os

padrões, vamos analisá-los e seleccionar aqueles que são relevantes para a definição de perfis de utilização.

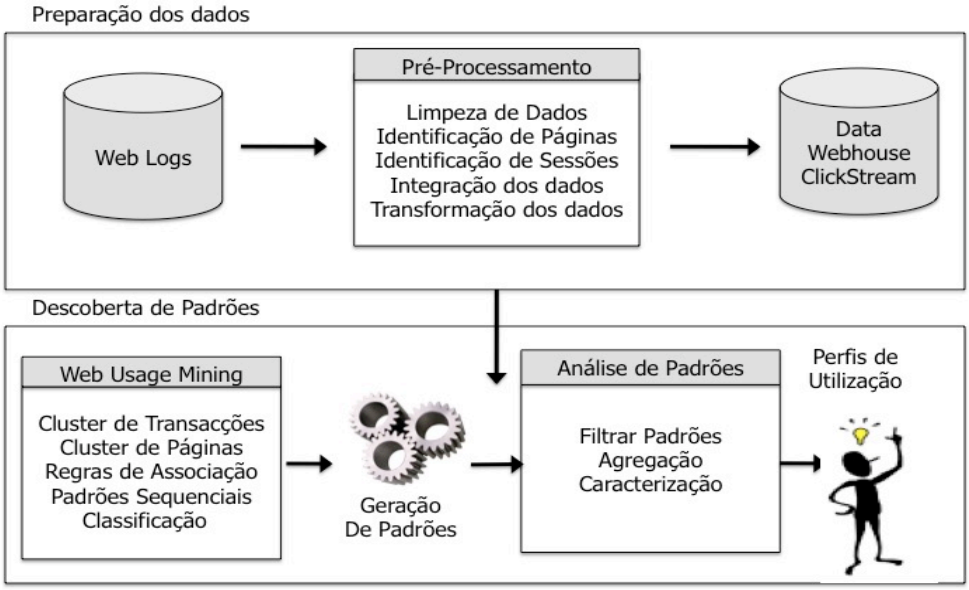


Figura 2.1: Descrição genérica da aplicação de mineração de dados de utilização Web

O processo de mineração de dados de utilização Web é similar ao processo de aplicação de técnicas de mineração de dados convencional. A diferença entre eles encontra-se, essencialmente, na fase de pré-processamento. Na aplicação de técnicas de mineração de dados, os dados encontram-se frequentemente guardados numa base de dados. Em mineração de dados de utilização Web a fase de recolha de dados tem especial importância, uma vez que a quantidade de dados envolvida é bastante grande. Depois dos dados recolhidos, o processo segue os mesmos passos da aplicação de técnicas tradicionais de mineração de dados, podendo haver algumas diferenças em cada um dos passos. Estas diferenças encontram-se sobretudo na fase de preparação dos dados. Uma vez que nesta fase são aplicados algoritmos específicos de mineração de dados de utilização Web. Como é o caso da identificação de sessões ou da identificação de páginas.

Capítulo 3

Preparação de Dados na Aplicação de Técnicas de Mineração de Dados de Utilização Web

3.1 Processamento de *Clickstreams*

Em qualquer aplicação de mineração de dados, uma das etapas mais importantes é a criação de um conjunto de dados apropriado, sobre o qual possam ser aplicados os algoritmos de mineração de dados. Esta etapa é particularmente importante em mineração de dados de utilização Web, devido às características específicas dos dados dos *clickstreams* e à sua integração com as restantes fontes de dados. Em termos gerais, este é o processo que consome mais tempo e mais recursos computacionais durante o processo de mineração de dados de utilização Web, sendo que geralmente recorre a algoritmos e heurísticas pouco utilizadas noutros domínios. A aplicação de técnicas de mineração de dados a um conjunto de treino inapropriado pode levar à descoberta de padrões que não representam a informação contida nos dados, o que faz com que a fase de pré-processamento seja uma das etapas mais críticas na geração de padrões. Durante esta etapa é feita a integração das diferentes fontes de dados e a sua posterior transformação, para que, posteriormente, possam ser passados como *input* aos algoritmos de mineração de dados.

Nos últimos anos têm sido desenvolvidas várias pesquisas em mineração de dados de utilização Web que se focam na fase de processamento e integração dos dados. A preparação de dados dos *clickstreams* coloca alguns desafios únicos, que lidam com uma enorme variedade de algoritmos e

heurísticas. Alguns dos principais desafios, colocados na fase de processamento de dados, são os seguintes: identificação de utilizadores, identificação de sessões, identificação de páginas, fusão e limpeza de dados, reconstrução de caminhos, entre outros [Cooley et al. 99]. O sucesso da aplicação de técnicas de mineração de dados está directamente relacionada com a correcta aplicação dos algoritmos de transformação de dados.

3.2 Fontes e Tipos de Dados

A principal fonte de dados utilizada em técnicas de mineração de dados de utilização Web são os ficheiros de *log* gerados pelos servidores Web, que acolhem todos os pedidos feitos aos *sites* sobre o controlo desses servidores. Adicionalmente, existem outras fontes de dados essenciais, quer na fase de preparação de dados, quer na fase de geração de padrões. Estas fontes podem ser, por exemplo: bases de dados de sistemas operacionais, informações sobre a estrutura do *sites* e dados sobre o conteúdo. [Cooley et al. 99] e [Srivastava et al 00] dividem os dados obtidos através das várias fontes em quatro tipos: dados de utilização Web, dados de conteúdo Web*,* dados de estrutura Web e dados sobre os utilizadores Web*.*

3.2.1 Dados de Utilização Web

Os ficheiros de *log*, capturados automaticamente pelos servidores Web*,* são a principal fonte de dados para a aplicação de técnicas de mineração de dados de utilização Web, uma vez que registam todos os pedidos HTTP efectuados pelos navegadores dos utilizadores ao *site*. Por cada pedido HTTP feito por um utilizador fica registado no ficheiro de *logs* uma entrada com informações sobre esse pedido. Por este motivo a informação guardada nesse ficheiro de *log* é bastante útil já que nos fornece informação pertinente para analisar o comportamento dos utilizadores de um *site* Web.

O formato de um ficheiro de *log* e, consequentemente, da informação que disponibiliza, varia de servidor para servidor - dentro de um servidor podem existir diferentes formatos. Contudo, independentemente do formato utilizado, um ficheiro de *log* deve conter sempre informações, tais como: a data e hora do pedido, o IP de origem e o pedido efectuado. Actualmente, existem vários formatos de *log* com nível de detalhe variável [WWW 5]. Os formatos disponibilizados, assim como o nível de detalhe, dependem do servidor Web utilizado. Existem alguns formatos de *log*

standard, os mais utilizados foram criados pelo *National Center for Supercomputing Applications(NCSA)* e pela *World Wide Web Consortium(W3C)* [WWW 6]. Actualmente, os mais utilizados são: *NCSA Common Log Format,(CLF), NCSA Extended Commom Log Format(ECLF) e W3C Extended Log Format(ELF).*

NCSA Common Log Format.

O *Common Log Format* definido em [WWW 7], [WWW 8] surgiu pela primeira vez com o servidor HTTPD [WWW 8] desenvolvido pelo NCSA. Este formato tornou-se o primeiro formato *standard* e surgiu como forma de tentar unificar todos os formatos de *log* proprietários existentes no mercado. Com a sua aparição, os vários servidores de *log* começaram a trabalhar com um formato único, o que tornou o desenvolvimento de programas de análise estatística mais simples. Dos três exemplos apresentados este formato é aquele que apresenta menos nível de detalhe. Isto é justificado pelo facto de ter sido o primeiro a aparecer. Aqueles que surgiram posteriormente puderam adaptar o seu formato, de forma a colmatar as falhas que o CLF apresenta.

Exemplo CLF	
125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043	
Campo	**Significado**
125.125.125.125	Endereço IP do cliente que efectuou o pedido ao servidor.
-	Identificação do utilizador com autenticação efectuada pelo processo ident a correr do lado do cliente. O hífen quer dizer que a informação não está disponível.
Dsmith	Identificador do utilizador autenticado através do processo HTTP. Caso a informação seja desconhecida coloca-se um "-".
[10/Oct/1999:21:15:05 +0500]	A data e hora em que o servidor terminou de servir o pedido.
"GET /index.html HTTP/1.0"	Pedido feito ao utilizador. Neste caso foi usado o método get para aceder á página índex.html, o protocolo usado foi HTTP/0.1.
200	Estado HTTP, resultante da execução do pedido, retornado ao cliente.
1043	Número de bytes enviados ao cliente na resposta ao pedido.

Tabela 3.1: Exemplo de um ficheiro de *log* no formato CLF

NCSA Extended Commom Log Format

O *Extended Common Log Format(ECLF)* [WWW 8], assim como o CLF, é disponibilizado pelo servidor Web da NCSA. Resulta de uma extensão do CLF pelo acréscimos dos campos *Referrer* e *User agent.* Vejamos então um exemplo de um ficheiro de *log* no formato ECLF (Tabela 3.2). Neste caso apenas vamos explicar os campos *Referrer* e *User Agent*, uma vez que todos os outros são comuns ao formato CLF.

Exemplo ECLF	
125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043 ***"http://www.ibm.com/" "Mozilla/4.05 [en] (WinNT; I)"***	
Campo	**Descrição**
"http://www.ibm.com/"	Indica a página que referenciou o pedido HTTP. Ou seja indica a página pedida anteriormente, que tinha um apontador para a página pedida. O valor pode ser desconhecido nesse caso coloca-se um "-".
/" "Mozilla/4.05 [en] (WinNT; I)"	Campo User agent. Contém o cliente HTTP utilizado para efectuar o pedido.

Tabela 3.2: Exemplo de um ficheiro de *log* no formato ECLF

De seguida, na Tabela 3.3 vamos apresenta uma comparação entre os dois formatos apresentados, assim como uma breve descrição de cada um dos campos existentes nos ficheiros de *log*.

Campo	**Exemplo**	**CFL**	**ECLF**	**Descrição**
RemoteHost	192.168.1.5	X	X	Endereço IP do cliente.
Ident	"-"	X	X	Identidade remota fornecida pelo pedido HTTP atribuída pelo processo Ident do lado do cliente.
Authuser	Dsmith	X	X	Nome ou código com que o utilizador se autenticou.
Date	[10/Oct/1999:21:15:05 +0500]	X	X	Data e Hora em que foi efectuado o pedido.
Request	GET /index.html HTTP/1.0"	X	X	Pedido feito ao servidor. Para além do pedido indica o método e o protocolo HTTP.
Status	200	X	X	O código de estado HTTP retornado para o cliente resultante do pedido efectuado.
Bytes	1043	X	X	Número de bytes enviados pelo servidor ao cliente HTTP.
Referrer	"www.google.pt"		X	Valor passado no campo referrer do cabeçalho HTTP. Indica por quem o pedido foi referenciado. Caso o seu valo seja desconhecido aparece um hífen("-").
User-Agent	/" "Mozilla/4.05 [en] (WinNT; I)"		X	O cliente HTTP utilizado no pedido. Normalmente identifica o navegador Web.

Tabela 3.3: Campos de CLF e do ECLF

W3C Extended Log Format ELF

Este formato foi desenvolvido pela *World Wide Web Consortiun*(W3C) com a intenção de obter um formato standard, que satisfizesse tanto as necessidades dos clientes como as dos servidores [Hallam and Behlendorf 96ª], [Hallam and Behlendorf 96b]. Ao contrário dos dois formatos anteriores, em que os campos apresentados eram fixos, o ELF apresenta um formato variável no qual se pode escolher quais os campos que ficam registados no ficheiro de *logs*.

Um ficheiro de *logs* no formato ELF é identificado por um cabeçalho, que precede todos os registos do ficheiro de *logs*. Este cabeçalho, contém diversas directivas que, nos vão dar informações sobre os dados recolhidos. Todas as directivas começam pelo símbolo "#". As directivas que normalmente se encontram no cabeçalho de um ficheiro de *logs* no formato ELF são as seguintes:

- *version* – a versão do formato utilizado;
- fields - indica quais os campos e a ordem em que aparecem no ficheiro;
- s*oftware* – identifica o software que gerou o ficheiro de *logs*;
- *start-date* – a data e hora a que o registo de *logs* começou;
- *end-date* - a data e hora a que o registo terminou;
- *date* - data e hora do registo da directiva;
- *GMT-Offset* - desvio do tempo em relação à hora *Greenwiich Mean Time(GMT)*;
- *remark* – comentários em texto livre.

Um ficheiro de *logs* pode ter mais do que cabeçalho, neste caso os *logs* seguem as directivas do cabeçalho que os precede no ficheiro.

No formato ELF os campos são separados por espaços, embora também seja possível usar tabulações. Os espaços contidos no URL do pedido são codificados, de forma a garantir que não são interpretados como separadores de campos. Tal como nos dois formatos anteriores, quando não há informação disponível sobre um campo é colocado um "-". De seguida apresentamos um exemplo de um ficheiro no formato ELF (tabela 3.4).

Ficheiro ELF		
#Software: Microsoft Internet Information Server 6.0 *#Version: 1.0 #Date: 1998-11-19 22:48:39* *#Fields: date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-bytes cs-bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referrer)* 1998-11-19 22:48:39 206.175.82.5 - 208.201.133.173 GET /global/images/ - 200 540 324 157 HTTP/1.0 Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95) USERID=CustomerA;+IMPID=01234 http://www.google.pt		
Campo	**Exemplo**	**Descrição**
Date	*1998-11-19*	Data em que o servidor terminou de servir o pedido.
Time	*22:48:39*	Hora em que o servidor terminou de servir o pedido.
c-ip	*206.175.82.5*	Endereço IP do cliente que fez o pedido.
Cs-username	-	Identificador usado para efeitos de autenticação.
s-ip	*208.201.133.173*	Endereço do servidor que respondeu ao pedido.
Cs-method	*GET*	Método HTTP utilizado.
Cs-uri-sterm	*/global/images/*	Identificação do nome e caminho do pedido.
Cs-uri-query	-	Componente de parâmetros variável incluída no URI.
Sc-status	*200*	O código HTTP enviado ao cliente, resulta do pedido.
Sc-bytes	*540*	Número de bytes enviados do servidor para o cliente.
Cs-bytes	*324*	Número de bytes enviados do cliente para o servidor.
Time-taken	*157*	Tempo que o servidor demorou a servir o pedido.
Cs-version	*HTTP/1.0*	A versão do protocolo HTTP usado.
Cs(user-agent)	*Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95)*	Contém o cliente HTTP utlizado no pedido.
Cs(cookie)	*USERID=CustomerA;+IMPID=01234*	Identificadores utilizados para a troca de informação de estado.
Cs(referrer)	*http://www.google.pt*	Indica por quem o pedido foi referenciado.

Tabela 3.4: Exemplo de um ficheiro de *logs* no formato ELF

3.2.2 Outras Fontes de dados

Como vimos anteriormente, os ficheiros de *log* são a principal fonte de dados na aplicação de técnicas de mineração de dados de utilização Web. Adicionalmente, existem outras fontes de dados que utilizadas em conjunto com os ficheiros de *log,* fornecem informações que podem melhorar os padrões gerados. De seguida, vamos apresentar algumas dessas fontes de dados.

Os dados de conteúdo são colecções de objectos que são visualizados pelo utilizador, ou seja, tudo aquilo que pode ser visualizado num *site* Web. Na maioria dos casos estes dados são compostos

por uma combinação de texto e imagens, mas podem conter também ficheiros de música, vídeos, entre outros. As fontes de dados utilizadas para gerar este tipo de dados podem ir desde páginas de HTML/XML, estáticas, até segmentos de páginas, gerados dinamicamente através de scripts ou outras aplicações, passando por imagens, ficheiros de vídeo e som, registos em bases de dados, etc. Para além da informação visível aos utilizadores, os dados de conteúdo também incluem metadados semânticos ou estruturais, que estão embebidos no *site* ou apenas em determinadas páginas, tais como descrição de palavras chave, atributos do documento, *tags* semânticas, variáveis HTTP e outras.

Os dados da estrutura são referentes à organização das páginas num *site*. A estrutura do *site* é capturada através dos *hiperlinks* para páginas internas, ou seja *hiperlinks* que direccionam os utilizadores para páginas do próprio *site* existentes nas suas várias páginas. Para além da organização das páginas, os dados de estrutura contém ainda a organização do conteúdo dentro da própria página. Por exemplo, os ficheiros em HTML/XML podem ser representados por uma estrutura em árvore, gerada através das várias *tags* espalhadas pelo documento. A organização dos *hiperlinks* é normalmente capturada por ferramentas que geram automaticamente o mapa do *site*. Este tipo de ferramentas deve ter a capacidade de capturar e representar as ligações entre as diversas páginas, bem como as relações entre os diversos conteúdos dentro de um página.

A última fonte de dados que vamos apresentar são as bases de dados operacionais que suportam um *site* Web. Estas, podem conter informações adicionais sobre os clientes, produtos vendidos entre outros. Dessa informação fazem parte dados sobre utilizadores registados: tais como dados demográficos, nome, idade, sexo, histórico de visitas, produtos mais comprados, páginas mais consultadas e últimas encomendas, entre outras informações, que nos ajudem a caracterizar o cliente. Alguns destes dados podem ser capturados anonimamente, sem saber quem realmente é o cliente, desde que seja possível distinguir entre diferentes utilizadores.

3.3 Junção e Limpeza de Dados

Usualmente, os *sites* de grande dimensão encontram-se alojados em vários servidores Web*,* pelo que os pedidos feitos por um utilizador ao *site* podem estar espalhados por diferentes ficheiros de *log* . Em alguns casos são mesmo utilizados vários servidores com informação redundante de modo a reduzir o tempo de carregamento das páginas. A junção de dados refere-se à fusão de ficheiros

de *log* provenientes de diferentes servidores Web. Para que isto seja possível, é necessário que exista uma sincronização global entre todos os servidores Web. Nos casos em que existe um identificador da sessão à qual pertence o pedido, e esse identificador é comum a todos os servidores Web*,* a junção dos ficheiros torna-se bastante simples, bastando juntar todos os pedidos com o mesmo identificador. Na falta de um marcador que identifique a sessão a que pertence cada pedido, podem ser utilizados alguns métodos heurísticos, baseados no campo "*referrer*" dos ficheiros *log* em conjunto com técnicas de identificação de utilizadores e sessões, para juntar *logs* provenientes de dois ficheiros de *log* distintos. A junção de ficheiros de *logs* é, assim, um passo essencial na aplicação de técnicas de mineração de dados de utilização Web, sobretudo em *sites* que se encontram alojados em mais do que um servidor, ou seja, registam os pedidos em diferentes ficheiros de *log* [Tanasa and Trousse 04].

O processo de limpeza de dados é normalmente especifico de cada *site* e envolve tarefas como a remoção de referências externas para objectos embebidos que não se mostram interessantes para as análises que se pretende fazer. Nas referências removidas podem-se incluir ficheiros de estilos, imagens ou ficheiros de som. No processo de limpeza podem também ser removidos alguns campos do ficheiro de *logs*, que contêm informação que não vai ser utilizada, como, por exemplo, o protocolo HTTP usado ou o método, que não sejam relevantes para o estudo. A limpeza de dados inclui ainda remover todos os pedidos efectuados por agentes automáticos, programas que se tentam passar por humanos para recolher informação sobre o conteúdo do *site* muito utilizados por motores de pesquisa como é o caso do *Google.* Não é incomum que um ficheiro de *logs* de um *site* tenha uma percentagem muita alta de acessos feitos por agentes automáticos. Uma grande percentagem destes agentes automáticos pode ser identificada e removida. Posteriormente, neste mesmo capítulo, vamos apresentar algumas das técnicas mais utilizadas na captura e remoção de pedidos feitos por agentes automáticos.

3.4 Identificação de Utilizadores

A identificação de utilizadores deve ser feita de forma a atribuir um identificador único a cada um deles. Em certos casos, esta identificação pode servir para identificar um utilizador em visitas posteriores. Contudo, nem sempre é possível identificar o utilizador o que faz com que nestes casos a identificação sirva apenas para distinguir diferentes utilizadores. Um utilizador pode fazer mais do que uma visita ao *site* Web. Nesta situação, os servidores Web registam mais do que uma

sessão por utilizador. Existem várias técnicas utilizadas na identificação de utilizadores. Algumas delas permitem reconhecer o utilizador em visitas posteriores, enquanto outras apenas atribuem um identificador anónimo a cada utilizador. Quando é possível escolher uma de entre várias técnicas deve-se pesar bem, obviamente, as vantagens e desvantagens associadas a cada técnica.

Uma das técnicas mais simples passa por obrigar os utilizadores a autenticarem-se no *site*, para que eles possam ter acesso aos conteúdos disponibilizados. A autenticação pode ser feita de duas maneiras: pelo servidor Web, ou a nível aplicacional. Se for feita pelo servidor Web, o identificador fica guardado no campo cs_*username* ou *authuser*, dependendo do formato do ficheiro de *logs*. Se for feito a nível aplicacional, terá de haver um mecanismo que relaciona o controlo de acessos feito pela aplicação com a informação contida nos ficheiros de *log*, isto para ser possível identificar correctamente os utilizadores no processamento dos *logs.* Este mecanismo poderá ser um *cookie* ou o adicionar de informação ao URI. Para que a autenticação funcione correctamente é necessário que o pedido de autenticação seja feito logo na primeira página, pois, caso contrário, as páginas visitadas antes de se autenticar não seriam correctamente identificadas. A autenticação apresenta algumas desvantagens, sendo a principal a desconfiança por parte dos utilizadores. A maioria dos utilizadores Web são um pouco relutantes em efectuar qualquer registo e evitam entrar em qualquer *site* que lhes peça para se autenticar. Este factor pode afastar muitos utilizadores, pelo que devem ser bem avaliados todos os prós e contas que esta técnica possa apresentar. Caso não seja necessário identificar os utilizadores o identificador de utilizador poderá ser atribuído pelo servidor Web ou por uma aplicação, servindo neste casos apenas como uma distinção entre utilizadores anónimos. Nesta situação, podem ser utilizados *cookies* para armazenar o identificador do utilizador. Nem todos os *sites* utilizam *cookies* e, devido a politicas de privacidade, estes são muitas vezes desactivados no lado dos utilizadores.

As técnicas apresentadas anteriormente atribuem identificadores de utilizador durante a interacção dos utilizadores com o *site*. Contudo, caso não tenha sido aplicada nenhuma delas é necessário atribuir algum tipo de identificador ao utilizador durante o processamento dos ficheiros de *log*. De seguida vamos apresentar uma técnica utilizada para atribuir identificadores de utilizadores durante o processamento de ficheiros de *log.*

A combinação de "*IP*+ *user agen*t" é um técnica muita simples que se baseia no seguinte método: todos os pedidos que têm o mesmo IP e o mesmo *User Agent* são atribuídos ao mesmo utilizador.

Consideremos o exemplo apresentado na Figura 3.1. Do lado esquerdo dessa figura podemos ver um exemplo de um ficheiro de *logs, e* do lado direito apresentamos o resultado da identificação de utilizadores utilizando a técnica "*IP+User Agent*". No ficheiro temos dois *IPs* distintos, o que indica que existem pelo menos dois utilizadores diferentes. Para verificar se são dois ou mais utilizadores é necessário analisar o campo *User Agent*. Para o *IP* 1.4.5.6 o valor do campo *User Agent* é sempre o mesmo pelo que temos apenas um utilizador. Contudo para o utilizador 1.2.3.4 temos dois valores diferentes do campo *User Agent* (IE6 e Safari), pelo que podemos dizer que temos dois utilizadores distintos com o mesmo *IP.* Podemos assim concluir que no bloco de *logs* apresentado foram identificados três utilizadores distintos.

Time	IP	URL	Ref	Agent
00:01	1.2.3.4	A	-	IE6
00:05	1.2.3.4	B	A	IE6
00:06	1.4.5.6	C	-	Safari
00:07	1.4.5.6	B	C	Safari
00:08	1.2.3.4	A	-	Safari
00:10	1.2.3.4	C	B	IE6
00:12	1.4.5.6	D	C	Safari
00:15	1.2.3.4	B	A	Safari
00:20	1.2.3.4	D	C	IE6

Utilizador 1	00:01	1.2.3.4	A	-	IE6
	00:05	1.2.3.4	B	A	IE6
	00:10	1.2.3.4	C	B	IE6
	00:20	1.2.3.4	D	C	IE6
Utilizador 2	00:06	1.4.5.6	C	-	Safari
	00:07	1.4.5.6	B	C	Safari
	00:12	1.4.5.6	C	D	Safari
Utilizador 3	00:08	1.2.3.4	A	-	Safari
	00:15	1.2.3.4	B	A	Safari

Figura 3.1: Exemplo de identificação de Utilizadores usando IP + Agent

A combinação *IP e User Agent,* como meio de identificar utilizadores, apresenta algumas fragilidades, uma vez que o mesmo *IP* pode ser utilizado por vários utilizadores. Isto pode acontecer devido a utilização de *proxies* que podem ter vários fins entre eles o de esconder a própria identidade. Nestes casos, será o endereço *IP* do servidor *proxy*, que serve múltiplos utilizadores, que vai aparecer no pedido. Temos também o caso das redes publicas, em que temos vários utilizadores distintos ligados em rede, mas cujos pedidos aparecem todos associados ao mesmo *IP.* Uma heurística possível de ser utilizada para resolver alguns dos problemas desta técnica, é apresentada em [Cooley et al. 99]. Neste estudo, é utilizado o referente como complemento a esta técnica.

3.5 Identificação de Agentes Automáticos

Nem todos as visitas Web são efectuadas por utilizadores humanos. São muitos os casos em que os pedidos feitos ao *site* são efectuados por agentes automáticos. Segundo um estudo realizado em [Kohavi and Parekh 03], os agentes automáticos podem representar entre 5% a 40% das visitas registadas num *site* Web. Estes agentes automáticos são conhecidos por *robots, spiders, Webbbots, worms, Web Crawlers, Web ants, wanderers e harverests,* entre outros. Em [Borges 04] podemos ver vários dos objectivos deste tipo de programas, nomeadamente:

- Indexação – agentes automáticos normalmente utilizados por motores de pesquisa. Percorrem todas as páginas de um *site* Web de modo a indexar a informação disponibilizada pelo *site*.
- Validação de páginas HTML - programas que verificam se determinadas páginas Web continuam activas.
- Validação de Apontadores - programas que percorrem todos os apontadores de um *site* Web, com o objectivo de verificar se estes são válidos.
- Alertar utilizadores sobre nova informação disponível no servidor – programas que enviam informações aos utilizadores, quando detectam alguma alteração no conteúdo de um *site* Web.
- Replicação de conteúdos entre servidores – agentes automáticos utilizados para percorrer e copiar toda informação disponibilizada numa página Web.
- Descarregar conteúdos do servidor Web para o computador do utilizador para permitir uma consulta quando desconectado da Internet – programas que percorrem todo o *site* Web e guardam em disco a informação disponibilizada no *site*, para posteriormente ser consultada.
- Recolha de emails para serem usados como alvos de campanha de emails não solicitados – programas que capturam os endereços de email dos utilizadores do *site*, com o objectivo de posteriormente enviarem a informação não solicitada.
- Contactos com o servidor Web para recolha de dados para elaboração de estatísticas diversas – programas que comunicam com o servidor de um *site* Web de modo a obter dados que depois de analisados podem ser bastante úteis.

Um agente automático pode ser definido como qualquer programa que utiliza o protocolo *HTTP* para percorrer documentos Web, quer seja através de *hiperlinks* ou utilizando outro método qualquer [Chau and Chen 03].

Existem agentes automáticos que trazem benefícios para o *site* Web, como é o caso dos que fazem indexação dos conteúdos do *site*. Estes agentes automáticos são utilizados sobretudo por motores de pesquisa, que utilizam a informação indexada para redireccionar utilizadores para o *site*. Contudo, existem também aqueles que não trazem nenhuma vantagem para o *site*, podendo mesmo prejudica-lo, uma vez que estão a consumir recursos que poderiam estar a ser utilizados por outros utilizadores.

Os acessos feitos por agentes automáticos provocam um grande número de registos no ficheiro de *logs*, esses registos podem prejudicar as análises feitas com base nos ficheiros de *log.* Por exemplo, quando se pretende analisar o comportamento dos utilizadores durante a navegação no *site*, os registos de *crawlers* podem alterar os resultados. Uma vez que os agentes automáticos tem estratégias de navegação diferentes dos utilizadores regulares. No estudo do comportamento dos utilizadores essas estratégias vão aparecer como comportamentos que os utilizadores do *site* apresentam normalmente, mas na realidade não traduzem o comportamentos dos utilizadores do *site*, mas sim o comportamento dos agentes automáticos que visitam os *site.* Assim os resultados obtidos da análise dos ficheiros de *log* não representa apenas o comportamento dos utilizadores do *site*, mas também o comportamento dos agentes automáticos. Por este motivo é importante a correcta identificação dos pedidos que foram feitos por agentes automáticos. Depois de identificados os registos atribuídos a *crawlers* podemos filtrar e seleccionar os registos de *log* que interessam analisar - esta filtragem é feita de acordo com os objectivos que se pretende atingir. Os administradores de *sites* Web têm à sua disposição um conjunto de técnicas que ajudam a tentar impedir o acesso ao *site* por agentes automáticos [Koster 94]. Contudo, nem sempre é possível impedir o acesso dos *crawlers* ao *site*, uma vez que existem alguns bastante mal comportados que não obedecem às regras estabelecidas, não podendo por isso ser detectados antes de entrar. Por este motivo é necessário identificar quais os registos feitos por *crawlers* na análise dos ficheiros de *log.*

Método	Vantagens	Desvantagens
Comparar valor do campo *user-agent* com uma lista de agentes automáticos construída previamente.	Identificar agentes automáticos previamente conhecidos.	Manutenção da lista de agentes automáticos. Existem registo em que o campo *user-agent* não se encontra preenchido. Alguns *crawlers* escondem a sua identidade para não serem identificados.
Comparar a origem dos pedidos com uma lista previamente conhecida.	Permite identificar agentes automáticos mesmo quando estes não tem valor no campo *User-agent.*	Manutenção da lista dos *crawlers* conhecida. Apenas identifica agentes conhecidos.
Verificar acessos á página robots.txt.	Permite detectar agentes automáticos desconhecidos.	Nem todos os *crawlers* visitam esta página.
Colocar armadilhas, com apontadores invisíveis ao ser humano.	Permite detectar agentes desconhecidos.	Implica alterar o *site* Web.

Tabela 3.5: Vantagens e Desvantagens dos métodos de Detecção de Agentes Automáticos

Existem várias técnicas que podem ser utilizadas na identificação de *crawlers,* sendo umas mais eficazes do que outras. Na Tabela 3.5 apresentamos uma breve descrição com as principais vantagens e desvantagens de algumas das técnicas mais utilizadas. Pode ser feita uma comparação entre o campo *User Agent,* existente nos ficheiros de *log,* e uma lista com todos os agentes automáticos conhecidos. Esta técnica não pode ser utilizada caso o campo *User Agent* não exista nos ficheiros de *log*, ou no caso do valor ser desconhecido Caso isso aconteça podemos usar o *IP* e comparar com uma lista de *IPs* de agentes automáticos conhecidos. Estas duas técnicas obrigam a manter uma lista sempre actualizada com o *IP* e o nome de todos os agentes automáticos conhecidos. Mesmo que se mantenha a lista sempre actualizada esta técnica nem sempre detecta os agentes automáticos, uma vez que apenas detecta agentes automáticos conhecidos. Por outro lado, existem agentes automáticos que tentam esconder a sua identidade fazendo-se passar por utilizadores normais passando valores no campo *User Agent* iguais aos de um navegador Web tradicional.

No caso de a técnica de comparações baseadas em listas não funcionar, o acesso à página *robots.txt* pode ser uma alternativa. Esta página aparece apenas em alguns *sites* Web, e contém um conjunto de regras que os agentes automáticos devem cumprir durante o acesso ao *site*. Esta página é visitada quase exclusivamente por agentes automáticos, apenas alguns utilizadores mais

curiosos ou com fins menos claros acedem a pagina. Assim podemos dizer que todos os pedidos feitos por um utilizador que acedeu a esta página foram feitos por agentes automáticos.

Outra das técnicas que pode ser utilizada é a de colocação de páginas ratoeira no *site*. Isto é páginas desprovidas de conteúdo e cujo acesso se faz por *hiperlinks* invisíveis aos seres humanos. Se estas páginas forem pedidas por alguma utilizador podemos atribuir todos esses pedidos a um agente automático [Kohavi and Parekh 03]. A aplicação desta técnica implica a alteração da estrutura do *site*, pelo que não é uma das técnicas mais adequadas. Existem outros métodos que podem ser utilizados na identificação de *crawlers*, por exemplo temos as técnicas que estudam os padrões de navegação dos *Web crawlers*. Em [Lourenço 04] é seguido este caminho, sendo utilizado um esquema que recorre a modelos de classificação para detectar padrões Web.

3.6 Identificação de Sessões

Após feita a identificação de utilizadores, é agora necessário definir quais os limites de uma sessão, ou seja quando começa e termina a sessão. A identificação de sessões consiste em fragmentar os registos de um utilizador em sessões, em que cada sessão representa uma visita diferente ao *site*. De acordo com [W3C99], uma sessão é composta pelo conjunto de actividades que um utilizador realiza, desde o momento em que entra no *site* até ao momento em que o abandona. A reconstrução de sessões visa agrupar sobre um identificador único todos os pedidos feitos por um utilizador desde que entra até sair do *site*. Tornando desta forma possível estudar o comportamento dos utilizadores, durante o período de tempo que esteve a navegar no *site*.

A identificação de sessões está ligada a identificação de utilizadores, uma vez que sem identificar os utilizadores não podemos identificar sessões. Quando a identificação do utilizador é feita sem ser necessário saber a sua identidade, ou seja é apenas uma distinção entre utilizadores anónimos, as técnicas utilizadas para reconstrução de sessões são praticamente idênticas às técnicas utilizadas para identificar os utilizadores.

As técnicas utilizadas para reconstrução de sessões dividem-se em dois grupos: pró-activas e reactivas [Spiliopoulou et al 03]. As estratégias pró-activas passam pela atribuição de um identificador único de sessão a cada pedido enquanto o utilizador está a interagir com o *site*. As estratégias reactivas tentam atribuir um identificador de sessão a cada pedido após estes terem

ocorrido, com base nos registos de pedidos HTTP registados nos ficheiros de *log*. Estes pedidos, incluem não só os pedidos efectuados directamente pelos utilizadores, como também os pedidos efectuados automaticamente a objectos incluídos nas páginas servidas. Existem vários *sites* que optam por utilizar estratégias pró-activas na identificação de sessões. Contudo o mecanismo de identificação de sessões pode falhar havendo pedidos que ficam sem identificador. Este factor indica que para uma reconstrução de sessões ser completa é necessário utilizar, em conjunto com as técnicas pró-activas, técnicas reactivas. Neste caso as técnicas reactivas servem como complemento, atribuindo identificadores de sessão aos pedidos que por alguma motivo ficaram sem identificador. No fim da aplicação das técnicas de reconstrução espera-se que todos os pedidos HTTP tenham um identificador de sessão. Contudo, devido à omissão de informação ou existência de informação incompleta nos dados de *clickstream* pode haver pedidos que fiquem sem nenhum identificador de sessão ou então colocados numa sessão errada.

3.6.1 Estratégias Pró-activas

A estratégia mais simples de reconstrução de sessões é quando o próprio servidor Web atribuí automaticamente um identificador de sessão a cada pedido HTTP que recebe. Neste caso, não é necessário atribuição posterior de um identificador único, uma vez que este fica registado nos ficheiros de *log*. A reconstrução de sessão vai ser feita, simplesmente, juntando todos os pedidos com o mesmo identificador. Há que considerar os casos em que atribuição de um identificador de sessão por parte de servidores Web não seja único. Neste caso vamos ter de aplicar outro método de atribuição de sessão de modo a garantir que as sessões ficam bem identificadas, pois caso contrario podemos ter utilizadores distintos a fazer parte da mesma sessão.

Os identificadores de sessão são tipicamente armazenados em *cookies*. Devido ao funcionamento em modo pedido resposta do protocolo HTTP, será apenas após a primeira resposta do servidor Web que um *cookie* pode ser passado ao navegador Web. Como tal, o registo do ficheiro de *log* referente ao primeiro pedido nunca terá informação da *cookie* que contém o identificador de sessão, esta *cookie* apenas aparecerá em pedidos posteriores, e sendo assim identificação da primeira página fica comprometida.

A utilização de *cookies* apresenta contudo outro problema, uma vez que o seu uso pode não estar autorizado no navegador do visitante. Nesta situação os sítios Web devem estar preparados para

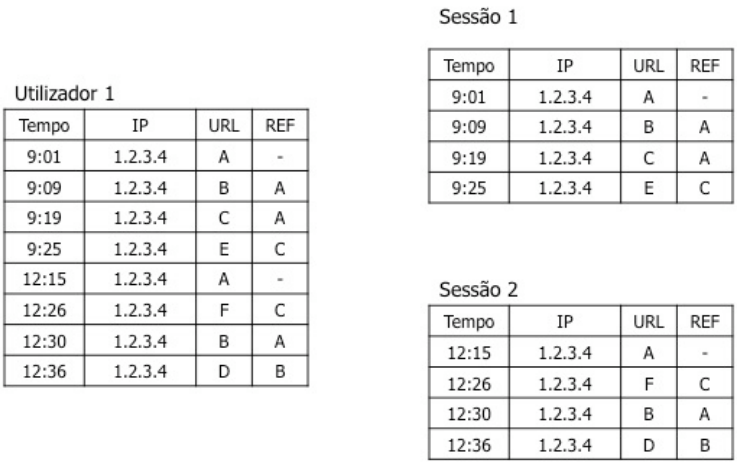
recorrer á colocação do identificador de sessão embebido nos parâmetros do URI. Caso sejam usados os dois tipos de estratégia em simultâneo por um *site* Web*,* no processo de reconstrução de sessões é preciso ter em conta que o identificador de sessões pode aparecer em dois locais diferentes.

3.6.2 Estratégias Reactivas

Caso não tenha sido atribuído um identificador de sessão durante a ocorrência dos pedidos HTTP, este terá de ser atribuído posteriormente durante o processamento dos ficheiros de *log.* Nestes caso a análise dos registos de *log* terá que em primeiro identificar quais os pedidos HTTP efectuados pelo mesmo utilizador e depois identificar quando começa e termina uma sessão. Vamos então apresentar algumas heurísticas que podem ser usadas na reconstrução de sessões. Primeiro vamos definir um valor para o tempo que os utilizadores podem ficar inactivos durante uma visita ao *site* Web. Em [Catledge and Pitkow 95] foi estimado um valor de inactividade médio dentro de um *site*, e chegaram ao valor de 25 minutos. Vários sistemas arredondaram esse valor para 30 minutos e passaram a utilizar esse tempo como o máximo tempo em que um utilizador posso estar inactivo no *site*. Isto significará que, numa mesma sessão, não poderá haver dois pedidos consecutivos separados por mais de 30 minutos.

Como exemplo de algumas heurísticas de identificação de sessões apresentamos de seguida três heurísticas[Spiliopoulou et al 03]. Cada heurística h recebe como input os *logs* do servidor Web particionados por utilizador, e dá como output o conjunto de sessões construídas. São elas:

- **H1 – Tempo total de Sessão.** Esta é a heurística mais simples de aplicar e consiste em estabelecer um limite máximo para uma sessão. O tempo total de sessão não pode exceder uma valor θ passado como input. Sendo t_0 o tempo atribuído ao primeiro pedido da sessão *S*, qualquer pedido pertencente a sessão *S* tem de ter uma etiqueta temporal *t* , em que $t+t_0 < \theta$.



Utilizador 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C
12:15	1.2.3.4	A	-
12:26	1.2.3.4	F	C
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Sessão 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C

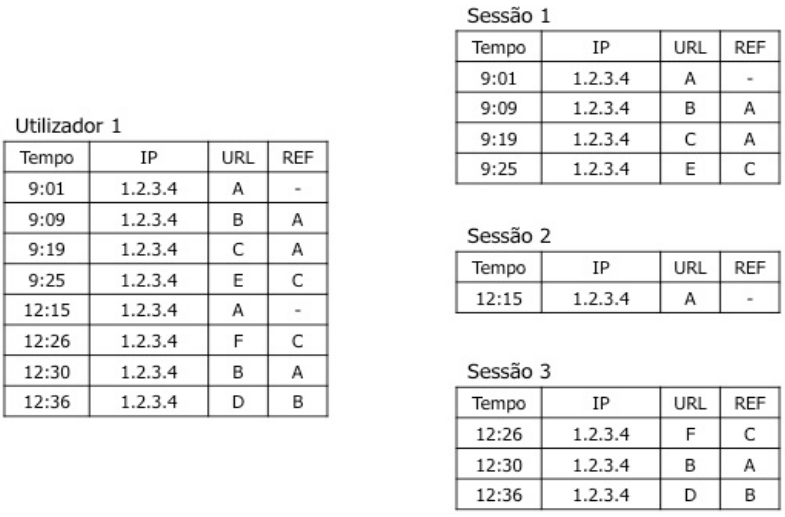
Sessão 2

Tempo	IP	URL	REF
12:15	1.2.3.4	A	-
12:26	1.2.3.4	F	C
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Figura 3.2: Exemplo da identificação de sessões usando a heurística H1

Na figura acima podemos ver um exemplo da reconstrução de sessões utilizando a heurística H1, foi utilizado um valor θ de trinta minutos para fazer a divisão dos pedidos por sessão.

- **H2 – Tempo máximo de inactividade.** O tempo total passado numa página não deve exceder um valor θ passado como input. Ou seja dado um pedido com etiqueta temporal t_n pertencente a uma sessão *S,* o próximo pedido com etiqueta temporal t_{n+1} pertence a sessão *S* se $t_{n+1} - t_n < \theta$. Ou seja, dois pedidos consecutivos pertence a mesma sessão se não tiveram passado mais de θ minutos entre eles.



Utilizador 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C
12:15	1.2.3.4	A	-
12:26	1.2.3.4	F	C
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Sessão 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C

Sessão 2

Tempo	IP	URL	REF
12:15	1.2.3.4	A	-

Sessão 3

Tempo	IP	URL	REF
12:26	1.2.3.4	F	C
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Figura 3.3: Exemplo da identificação de sessões usando a heurística H2

Na figura anterior podemos ver um exemplo da reconstrução de sessões utilizando a heurística H2, o tempo máximo de diferença entre dois pedidos utilizado foi de 10 minutos.

- **H-Ref – Baseado no Referente.** Um pedido *q* é adicionado a uma sessão *S* se o referente de *q* for previamente invocado em *S.* Caso contrário *q* é usado como o inicio de uma nova sessão. Contudo, utilizando esta heurística, é possível que *q* pertença a mais do que uma sessão que se encontre aberta, uma vez que o referente de *q* pode ter sido acedido em várias sessões Também existem os casos em que o referente de *q* pode ir vazio. Nestes dois casos usa-se a heurística ***H2*** para atribuir uma sessão a q. Ou seja vai se buscar a sessão *S* cujo último pedido tem uma diferença menor para o tempo de *q,* e caso essa diferença seja menor que um valor θ passado como input *q* vai pertencer a essa sessão caso contrário inicia-se uma nova sessão com *q*.

Utilizador 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C
12:15	1.2.3.4	A	-
12:26	1.2.3.4	F	C
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Sessão 1

Tempo	IP	URL	REF
9:01	1.2.3.4	A	-
9:09	1.2.3.4	B	A
9:19	1.2.3.4	C	A
9:25	1.2.3.4	E	C
12:26	1.2.3.4	F	C

Sessão 2

Tempo	IP	URL	REF
12:15	1.2.3.4	A	-
12:30	1.2.3.4	B	A
12:36	1.2.3.4	D	B

Figura 3.4: Exemplo da identificação de sessões usando a heurística H-REf

Na figura acima utilizamos o método de reconstrução de sessões H-REF aplicado ao mesmo conjunto de *logs* de um utilizador. Neste caso quando o pedido da página F é feito existem duas sessões activas, mas ainda que a sessão dois tenha tempos mais próximos do tempo de F, F vai ser incluído na sessão 1 pois é nessa sessão que se encontra o pedido de C que é o referente de F. O pedido de B do 12:30 podia ser incluído em ambas as sessões uma vez que ambas têm um pedido A. Contudo, como a sessão 2 é mais próxima em termos de

tempo, o pedido vai ser incluído nessa sessão. O mesmo acontece com o pedido D, que tem B como referente.

3.7 Identificação de Páginas

Os servidores Web registam todos os pedidos feitos ao servidor por um navegador Web. Para além das páginas pedidas pelos utilizadores, também ficam registados pedidos para todos os componentes da página, como por exemplo: imagens, ficheiros de música, entre outros. Os pedidos têm assim de ser analisados, de modo a poderem ser classificados, como sendo páginas Web ou simplesmente como um componente de uma página Web. A identificação de páginas está dependente da estrutura do *site* Web, assim como dos conteúdos e do domínio de conhecimento sobre o qual o *site* actua. Uma página Web pode ser vista conceptualmente, como um conjunto de objectos ou recursos que representam um evento específico de um utilizador. Este eventos podem ser: carregar num *hiperlink,* abrir a página de um produto ou adicionar um produto ao carrinho de compras.

Num *site* que apenas utiliza páginas sem frames, cada página HTML pode ser vista como uma página do *site*. Contudo, para um *site* que utiliza páginas com vários frames, várias páginas HTML podem fazer parte da mesma página Web. Para *sites* dinâmicos, ou seja, *sites* que estão constantemente a mudar as suas páginas Web, uma página pode representar um conjunto de templates estáticos e conteúdos gerados por aplicações com base em parâmetros que lhe foram passados.

Para facilitar a aplicação de técnicas de mineração de dados um conjunto de informações deve ser registado em conjunto com as páginas. Estas informações são geralmente constituídas por um identificador único da página (normalmente um URI único que representa a página), o tipo de página (página de informações, produtos, categoria, etc.) e outros metadados, tais como informação sobre o conteúdo das páginas.

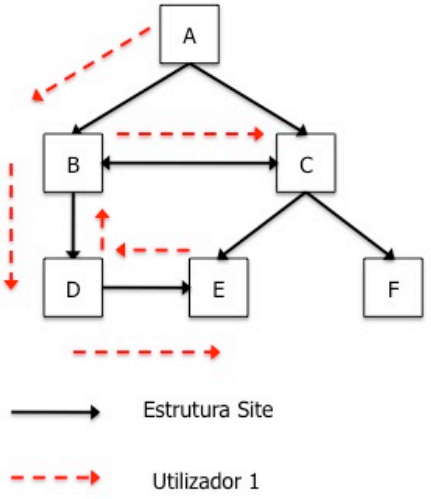
3.8 Reconstrução de Caminhos

Depois de feita a identificação de sessões, é necessário reconstruir a sequência das páginas visitadas pelo utilizador. Para isso as páginas visitadas devem ser identificadas e interpretadas no

conceito de sessão, cada página deve ser identificada através de um número de sequência dentro da sessão. Um factor que se deve ter em conta durante a reconstrução das sessões é o efeito que as *caches* causam. A utilização de *caches* é um método utilizado para melhorar os tempos de resposta e diminuir o tráfego na rede. Contudo a sua utilização traz alguns problemas na captura de ficheiros *de log,* uma vez que os sistemas de *cache* existentes entre o utilizador e o servidor guardam localmente cópias das páginas que foram acedidas. Devido à utilização de sistemas de *cache* é provável que, sempre que um utilizador carrega na botão de retroceder no seu navegador, esteja a aceder a páginas guardadas numa *cache,* em vez de estar a fazer o pedido directamente ao servidor. Consequentemente, como o pedido não foi feito ao servidor, este registo não ficará guardado nos ficheiros de *log* gerados pelos servidores. A não inclusão destes pedidos no ficheiro de *logs* faz como que a sequência de páginas visitadas na sessão não esteja completa, uma vez que faltam pedidos. As *caches* são um dos principais motivos para que muitos programas de análise de ficheiros de *log* não funcionem correctamente.

Existem alguns métodos que podem ser utilizados de forma a reduzir a falta de registos nos ficheiros de *log,* devido a utilização das *cache*. Em [Borges 04] podemos ver alguns dos métodos mais utilizados, tais como: a utilização de páginas dinâmicas, ou de apenas um componente dinâmico na página, atribuir ás paginas Web uma data de validade de maneira a forçar um novo carregamento sempre que são pedidas, incluir no cabeçalho HTML directivas que indiquem que a página não deve ser mantida em *cache,* entre outros.

Caso não seja utilizado nenhum método para reduzir os efeitos da cache sobre os ficheiros de *log*, é possível através da análise dos registos e da estrutura do *site* descobrir uma grande parte dos registos que foram omitidos. Isto é feito com recurso a análise dos referentes de cada um dos pedidos de uma sessão. Depois de ordenados todos os pedidos dentro de uma sessão, é possível através da análise dos referentes descobrir pedidos omitidos. Se o referente de uma página *P* não for igual à página que a antecedeu na sessão, podemos dizer que uma ou mais páginas foram servidas, utilizando mecanismo de *cache*, entre esses dois pedidos. Caso a página referente de *P* tiver sido pedida anteriormente nessa sessão, isso pode ser um indicador de que utilizou o botão de retroceder, visualizando a página armazenada em *cache* pelo navegador Web. Na figura seguinte mostramos um exemplo retirado de [Mobasher et al 01a] de uma estrutura de um *site* Web, e dos *logs* gerados por um utilizador numa visita ao *site* Web.



Percurso de Navegação Utilzador 1

A -> B -> D -> E -> D -> B -> C

Ficheiros de Log		
Hora	*Pedido*	*Referente*
9:00	A	-
9:05	B	A
9:07	D	B
9:08	E	D
9:10	C	B

Figura 3.5: Estrutura de um *site* Web e páginas visitadas por um utilizador

No exemplo apresentado na figura 3.5 temos representada a estrutura de um *site* (setas contínuas) e um caminho feito por utilizador hipotético (setas a tracejado). Temos também representados numa tabela as entradas no ficheiro de *logs* que o caminho feito pelo utilizador registou. Analisando a tabela dos *logs,* podemos observar que a última página servida *C.html* tem como referente a página *B.html*. Ora dado que esta página não corresponde à página servida no pedido anterior *E.html*, podemos dizer que o utilizador utilizou a tecla de retroceder até chegar à página *B.html*, o referente da página *C.html*. Os pedidos feitos enquanto o utilizador usou o botão de retroceder não foram registados pelos servidores de *log,* uma vez que, como já tinham sido pedidos anteriormente, ficaram guardados na *cache* do navegador *Web*. Considerando a estrutura do *site Web* podemos considerar dois caminhos que o utilizador possa ter feito : *1) - A->B->D->E->D->B->A->B->C* e *2) - A->B->D->E->D->B->C.* Nos casos em que existe mais do que um caminho possível, em [Cooley et al. 99] é sugerida uma heurística para a escolha do caminho mais curto. O caminho escolhido seria, assim, o segundo. Neste caso, faltam dois registos entre os pedidos *E* e *C* do ficheiro de *log*. Estes dois pedidos podem ser colocados no ficheiro de *log*, contudo não temos uma referência temporal para os pedidos. Essa referência pode ser calculada com base no tempo médio de permanência dos utilizadores, em páginas do mesmo tipo.

Como observámos anteriormente, um factor essencial para a reconstrução de caminhos é conhecer a estrutura do *site* em análise. Para *sites* estáticos é bastante simples manter informação sobre a

estrutura do *site*, contudo para *sites* dinâmicos, *sites* em que as páginas e o seu conteúdo estão constantemente a ser alterados, é necessário encontrar alguns mecanismos que permitam obter a estrutura do *site*.

3.9 Integração dos Dados

O resultado da última etapa do processamento de dados é um conjunto de sessões, em que cada uma delas é composta pela sequência de páginas visitadas nessa sessão. Contudo para um melhor estudo de padrões de navegação é preciso agora integrar as outras fontes de dados com os dados de *clickstream* processados. Este processo é extremamente útil em *sites* dedicados ao comércio *online* onde é necessário integrar dados referentes aos utilizadores registados(dados demográficos, lista de produtos comprados, principais interesses), bem como dados referentes aos produtos que se encontram disponíveis no *site*(categorias, famílias). Estes dados normalmente encontram-se em bases de dados operacionais, que em conjunto com os dados de *clickstream* podem ser utilizados no processo de mineração de dados, tanto para descobrir importantes métricas para o negócio como para tentar identificar padrões de navegação [Kohavi et al 04].

A integração dos dados é normalmente feita num *Data Webhouse (Dweb)* [Kimball et al 98], uma instância de uma *Data Warehouse (DW*) [Kimball and Mertz 00], em que uma das fontes de dados são os ficheiros de *log* gerados pelos servidores Web. Um dos pontos fortes dos sistemas de DW é a modelação dimensiona. Ao contrário do tradicional modelo relaciona baseado em entidades interligadas entre si, a modelação dimensional assenta no conceito de dimensão e tabelas de facto [Kimball et al 98]. Sendo um DWeb uma instancia de um DW, em que as fontes de dados são ficheiros de log de servidores Web, é igualmente possível fazer estudos segundo vários eixos de análise e assim obter conhecimento, que de outra forma seria algo extremamente difícil. Assim como na modelação de um DW, também na modelação de um DWeb a escolha do nível de detalhe (tradicionalmente chamado de granularidade [Kimball et al 98]) é uma tarefa extremamente importante, visto que é a partir desta definição que se poderá conhecer o tipo de factos a analisar. Na modelação do DWeb normalmente são utilizados dois tipos de detalhe. Por um lado temos a tabela de facto cujo grão é pedidos feitos pelo utilizador, em que cada entrada na tabela de factos representa um pedido feito ao servidor. Por outro temos tabela de factos cujo grão é uma sessão do *site* Web, em que cada entrada na tabela corresponde a uma sessão de utilizador no *site*. A

tabela de factos que guarda cada um das sessões tem informação menos especifica uma vez que agrupa os pedidos por sessão.

A utilização de um *DWeb* para a integração dos dados torna-se muito útil, pois permite visualizar a informação contida nos ficheiros de *log* sobre diferentes perspectivas Permite por exemplo consultar todos as sessões efectuadas num dado período de tempo, analisar apenas os pedidos feitos de um dado país entre muitas outras análises. Os dados armazenados num sistema de *Data Webhousing* podem posteriormente ser utilizados para alimentar sistemas *OLAP*, sistemas utilizados para fazer análises sobre os dados guardados num *Dweb.*

Capítulo 4

Descoberta e Análise de Padrões

4.1 Modelação de Dados

Usualmente, o processamento de dados de utilização Web resulta num conjunto de páginas $P = \{p_1,p_2,.....p_n\}$ e num conjunto de sessões (ou transacções) $S = \{S_1, S_2, ..., S_t\}$, em que cada uma das sessões pertencentes a S é um subconjunto de P. Conceptualmente cada sessão pode ser vista como uma sequência ordenada de pares de páginas [Liu 08]:

$$t = <(p_1^t, w(p_1^t)),(p_2^t,w(p_2^t)),...,(p_n^t,w(p_n^t))>$$

em que cada $p_n^t = p_j$ para algum $j \in (1,2,....,n\}$, e $w(p_n^t)$ é o peso associado à página p_n na sessão i, ou seja representa a sua importância da página dentro da sessão. O peso de uma página dentro de uma sessão pode ser atribuído de várias maneiras, dependendo do tipo de análise que se pretende efectuar. Na maioria das aplicações de mineração de dados de utilização Web os pesos podem ser de dois tipos: binários, que representam a existência ou não dessa página na sessão geralmente é utilizado o zero caso a página não exista e o um caso exista; ou, então, representam o tempo que o utilizador esteve a visitar a página Web durante a sessão. No caso da utilização do tempo passado é de notar que o tempo da última página não está disponível. Uma das maneiras de resolver este problema é calcular a média do tempo que o utilizador passou em todas as páginas da sessão e atribuir esse tempo à última página.

Para muitas aplicações de mineração de dados Web, que envolvam clusters regras de associação ou classificação, em que a ordem das páginas não é importante, podemos representar cada sessão de um utilizador como um vector de páginas. Dado a transacção t referida em cima, o vector v que representa a transacção é dado por:

$$V = (w^t_{p1}, w^t_{p2},...,w^t_{pn})$$

em que cada $w^t_{pi} = w(p^t_j)$ para algum $j \in \{1,2,...,n\}$, caso p_j aparecer na transacção t, e igual a zero no caso contrário. Quando se utilizam pesos binários, w^t_{pj} é igual a um caso p_j aparece em t ,e igual zero casa não apareça. Assim, conceptualmente, podemos visualizar o conjunto de todas as transacções como uma matriz *T x M* de transacções de utilizadores. Na Figura 4.1 podemos ver um exemplo de uma dessas matrizes, em que, neste caso, os pesos atribuídos a cada página são binários.

Páginas

P = {Index., Desporto., Tecnologia,
Politica, Economia, Pais}

Sessões

S = {(user0, [index, Desporto., Tecnologia]);
(user1 ,[Politica, Economia]);
(user2 ,[index, Economia, Pais]);
(user3 ,[index, Desporto., Tecnologia]);
(user4 ,[Politica, Economia]);
(user5 ,[index, Economia, Pais]);
(user6 ,[index, Desporto., Tecnologia]);
(user7 ,[Politica, Economia]);
(user8 ,[index, Economia, Pais, Politica]);
(user9 ,[Desporto., Tecnologia, Politica]);
}

Matriz de Transacções

	Index	Desporto	Tecnologia	Politica	Economia	Pais
U0	1	1	1	0	0	0
U1	0	0	0	1	1	0
U2	1	0	0	0	1	1
U3	1	1	1	0	0	0
U4	0	0	0	1	1	0
U5	1	0	0	0	1	1
U6	1	1	1	0	0	0
U7	0	0	0	1	1	0
U8	1	0	0	1	1	1
U9	0	1	1	1	0	0

Figura 4.1: Construção de uma matriz de transacções

Dado um conjunto de sessões na forma de uma matriz de transacções de utilizadores, podem ser aplicadas sobre ela várias técnicas de extracção de conhecimento. Algumas dessas técnicas tais como clusters de sessões podem levar à descoberta de importantes grupos de utilizadores. Outras técnicas, como as regras de associação, podem levar à descoberta de relacionamentos importantes entre páginas através da análise dos padrões de navegação dos utilizadores no *site*. Também é possível integrar outras fontes de dados nestes processos, tais como dados sobre o conteúdo do *site*. Para estes casos as matrizes de transacções vão apresentar algumas diferenças de modo a conseguir mostrar os vários tipos de informação [Liu, 08]

4.2 Análise Estatística

Uma das formas mais comuns de análise dos dados de utilização Web é a geração de dados estatísticos sobre as sessões processadas. Neste tipo de análises, os dados são agrupados por unidades predeterminadas, como por exemplo: o dia, o mês, o ano, o país, a cidade, o referente ou o tipo de pedidos, entre outros. A análise deste tipo de estatísticas pode ser muito útil na descoberta de padrões de comportamento de utilizadores. Este tipo de análise, consiste, basicamente, na geração de relatórios com os mais diversos tipos de informação, podendo incluir informação sobre as páginas mais requisitadas, o tempo médio de visita a uma página, o número médio de páginas visitadas numa sessão, as principais páginas de entrada e saída dos *sites* Web, os países de onde são feitos mais pedidos, as páginas menos requisitadas, os principais referentes para o *site*, etc. Este tipo de análise fornece informação bastante útil para fazer eventuais reestruturações de *site* ou para melhorar a sua *performance*. Embora este seja um processo relativamente simples, apresenta algumas falhas de relevo – em alguns casos não é feita a identificação de *crawlers*, o que pode induzir em erro, fornece apenas dados estatísticos sobre os acessos ao *site*, esta informação pode ser incompleta, por exemplo uma página no total de acessos pode ser muito pouco visualizada no entanto, pode ser bastante vista por um grupo minoritário de utilizadores do *site*, mas que são bastante importantes para o negócio.

Para além da geração de relatórios, outra forma de análise estatística sobre os dados integrados é a utilização de técnicas *OLAP* (*Online Analytical Processing*). A fonte de dados para um sistema *OLAP* é normalmente um sistema de *data warehousing*. Neste caso será um *data warehouse* que integrará os dados dos ficheiros de *log* e de outras eventuais fontes adicionais. As ferramentas *OLAP* permitem efectuar análises aos dados com diferentes níveis de agregação, bem como mudar

o nível de agregação de cada uma das dimensões ao longo do processo de análise. A análise de dados Web numa estrutura deste tipo vai ter como base os vários campos do ficheiro de *logs* tais como: tempo de duração, referente ou *user agent*. Os *output* dos sistemas *OLAP* pode ainda ser posteriormente utilizado com *input* de algoritmos de mineração de dados [Buchner and Mulvenna 99].

A maioria dos programas de análise de ficheiros de *logs* existente no mercado limita-se a gerar alguma informação estatística sobre os dados obtidos. No entanto, e devido às inúmeras fragilidades deste tipo de abordagem, as ferramentas comerciais de análise de ficheiros de *log* tentam, cada vez mais, melhorar as suas implementações. Para isso apostam em incluir nos seus produtos a aplicação de alguns algoritmos de mineração de dados, com o objectivo de fornecer informação mais completa aos seus clientes. No *site* [WWW 3,] [WWW 4] podemos ver alguns exemplos de ferramentas que geram estatísticas a partir de ficheiros de *log.*

4.3 Clusters

Os clusters são uma das técnicas de aprendizagem não supervisionada mais utilizadas em mineração de dados. Basicamente, a aplicação de clusters consiste em dividir um conjunto de itens em vários grupos, em que os elementos de cada grupo apresentam semelhanças entre si. Um cluster não é mais do que um conjunto de itens que, por um lado, apresentam semelhanças entre si e, por outro, apresentam diferenças relativamente aos elementos dos outros clusters [Kaufman L. and Rousseeuw P.J., 90], [Berkhin 02], [Jain and Dubes 88]. Na Figura 4.2 podemos ver uma representação em duas dimensões de um conjunto de dados e a sua organização em quatro grupos de elementos distintos. Cada um desses grupos pode ser considerado um cluster. O objectivo da aplicação de técnicas de clusters, neste caso, é descobrir os quatro grupos de itens escondidos nos dados. Num caso a duas dimensões, como o descrito de seguida, é fácil de identificar cada um desses clusters, através da simples observação dos dados. Contudo num conjunto de dados com mais de três dimensões torna-se complicado, senão mesmo impossível, fazer a sua visualização. Existem ainda algumas aplicações em que a divisão dos clusters não é assim tão visível como a representada.

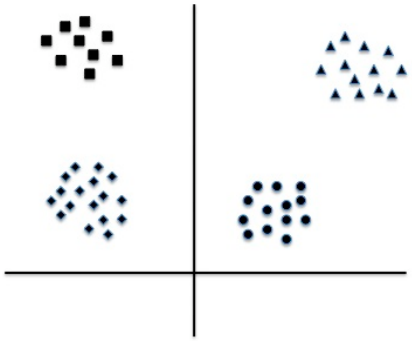


Figura 4.2: Exemplo de quatro clusters de dados

Existem vários tipos de algoritmos de *clustering.* Dois dos tipos mais conhecidos são os algoritmos de cluster hierárquicos [Murtagh 83] e os particionais[Boley 98]. Como o próprio nome hierárquico indica, os clusters gerados com algoritmos hierárquicos estão ligados uns aos outros estabelecendo uma hierarquia, podendo os vários clusters identificados ficarem incluídos num cluster mais abrangente. Isto permite que um dado item possa ser encontrado em mais do que um cluster. Nos clusters particionais, os clusters gerados não têm qualquer ligação uns com os outros e cada elemento do conjunto de dados apenas pode fazer parte de um cluster. Na aplicação de *clustering* é necessário definir uma função de semelhança, que nos indique qual a semelhança de dois pontos distintos, ou então uma função de distância que determine a distância entre dois pontos.

No domínio da mineração de dados de utilização Web existem dois tipos de cluster que interessa analisar: clusters de utilizadores e clusters de páginas. Os cluster de utilizadores (sessões) são uma das técnicas mais utilizadas em mineração de dados Web, que tendem a estabelecer grupos de utilizadores com o mesmo perfil de navegação no *site*. O conhecimento destes perfis torna-se especialmente útil, uma vez que estes permitem descobrir quais os segmentos de mercado que o *site* está a atingir. Além disso, permitem também descobrir os hábitos dos utilizadores e, através desta informação, personalizar o conteúdo do *site* Web de forma a agradar a futuros utilizadores, com o mesmo tipo de perfil de utilização. A análise futura de grupos de utilizadores com base nos seus atributos demográficos (idade, país e localidade) pode levar à descoberta de informações importantes sobre o negócio. Os clusters de utilizadores, podem servir para identificar diferentes comunidades de utilizadores, estas comunidades reflectem os interesses dos utilizadores [Paliouras et al 02]. Para além disso, podem também ser utilizados em sistemas de recomendação dinâmica para aplicações de personalização Web[Mobasher et al 02].

Neste trabalho, a aplicação de técnicas de *clustering* vai ser feita sobre uma dada matriz de transacções. Neste tipo de matriz, cada linha representa uma sessão no *site*, enquanto que as colunas representam as suas páginas. Dadas todas as transacções de utilizadores numa matriz de transacções, os algoritmos de *clustering* (como, por exemplo, o *K-means* [Hartigan and Wong 79], [Hartigan 75] ou *EMClustering* [McLachlan and Krishnan 96], [Dempster et al 77] particionam a matriz em grupos de utilizadores com características semelhantes. Esta separação é feita com base numa função que mede a distância entre transacções. Os clusters de transacção obtidos deste modo representam grupos de utilizadores estabelecidos com base no seu comportamento durante a navegação no *site* Web. Contudo, apenas a indicação dos clusters de transacções não é suficiente para estabelecer um perfil de navegação, uma vez que cada cluster pode conter milhares de sessões e envolver centenas de páginas. Isto torna a informação fornecida pelos clusters inútil, já que é demasiada informação para se poder analisar. Devido a isso, é necessário definir uma maneira de agregar a informação de forma a que seja possível analisá-la de forma concreta. O último objectivo dos clusters de transacções é, assim, arranjar uma maneira que permita analisar os dados de cada cluster, para que, posteriormente, possam ser utilizados em diversas áreas, como nos casos que envolvam a personalização de *sites* Web.

Matriz de Transacções

	index	desporto	tecnologia	politica	economia	pais
u0	1	1	1	0	0	0
U1	0	0	0	1	1	0
U2	1	0	0	0	1	1
U3	1	1	1	0	0	0
U4	0	0	0	1	1	0
U5	1	0	0	0	1	1
U6	1	1	1	0	0	0
U7	0	0	0	1	1	0
U8	1	0	0	1	1	1
U9	0	1	1	1	0	0

Clusters

Cluster 1

	index	desporto	tecnologia	politica	economia	pais
U1	0	0	0	1	1	0
U4	0	0	0	1	1	0
U7	0	0	0	1	1	0

Cluster 2

	index	desporto	tecnologia	politica	economia	pais
U0	1	1	1	0	0	0
U3	1	1	1	0	0	0
U6	1	1	1	0	0	0
U9	0	1	1	1	0	0

Cluster 3

	index	desporto	tecnologia	politica	economia	pais
U2	1	0	0	0	1	1
U5	1	0	0	0	1	1
U8	1	0	0	1	1	1

Figura 4.3: Separação do conjunto em três cluster

Uma maneira simples de obter uma vista agregada sobre os dados existentes em cada cluster é através da computação do vector principal de cada um dos cluster. O valor de cada página no

vector principal é calculado através da soma do valor dessa página em todas as sessões pertencentes ao cluster, a dividir pelo número total de sessões existentes no clusters. No caso em que os valores da matriz são apenas zeros e uns, o valor da página representa a percentagem de sessões em que a página aparece no cluster. Isto indica-nos a popularidade da página dentro do cluster. O valor de cada página no vector principal representa qual a importância da página dentro do cluster. Depois de calculado esse valor, é construída uma lista ordenada das várias páginas dentro do cluster, em que a ordenação será feita de acordo com o valor da página no vector principal. Depois de termos a lista construída, as páginas com menos importância dentro do cluster podem ser eliminadas, uma vez que não aparecem num número significativo de sessões. O conjunto de páginas e seus respectivos pesos obtidos podem ser vistos como um perfil de utilização Web, que representa as páginas mais interessantes para um dado grupo de utilizadores.

De uma maneira mais formal, dado um conjunto de clusters de transacções *c*, podemos construir o perfil de utilização *p,* que representa c da seguinte maneira:

$$P = \{(página, peso(página, c)) \mid peso(página, c) > X\}$$

em que *X* representa o valor mínimo que uma página deve ter para fazer parte do perfil. Quanto ao peso, este é dado pela seguinte expressão:

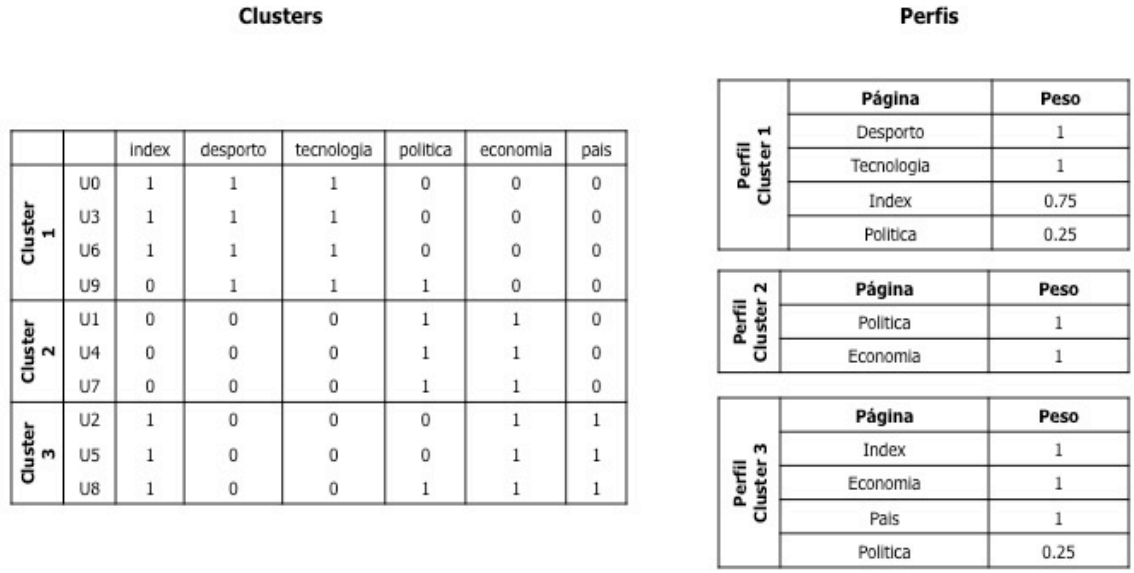
$$Peso(p, c) = \{\sum_{s \in c} w(p,s) / |c|\}$$

em que:

- *|c|* é o número de elementos de c;
- *w(p,s)* é o peso da página p na transacção s do cluster c.

Cada um dos perfis obtidos pode ser utilizado em vários modelos de previsão, como também em sistemas de recomendação. Dado um utilizador *u* que acedeu a um determinado conjunto de páginas até ao momento, podemos medir a proximidade entre o conjunto de páginas visitadas e os perfis gerados e, desta forma, definir em que perfil se encaixa o utilizador. Depois de definido o

perfil podemos recomendar ao utilizador que visite as páginas que compõem o perfil e que ainda não foram visitadas por ele.



Clusters

		index	desporto	tecnologia	politica	economia	pais
Cluster 1	U0	1	1	1	0	0	0
	U3	1	1	1	0	0	0
	U6	1	1	1	0	0	0
	U9	0	1	1	1	0	0
Cluster 2	U1	0	0	0	1	1	0
	U4	0	0	0	1	1	0
	U7	0	0	0	1	1	0
Cluster 3	U2	1	0	0	0	1	1
	U5	1	0	0	0	1	1
	U8	1	0	0	1	1	1

Perfis

Perfil Cluster 1	Página	Peso
	Desporto	1
	Tecnologia	1
	Index	0.75
	Politica	0.25

Perfil Cluster 2	Página	Peso
	Politica	1
	Economia	1

Perfil Cluster 3	Página	Peso
	Index	1
	Economia	1
	Pais	1
	Politica	0.25

Figura 4.4: Geração de perfis de utilização através de clusters

Os clusters de páginas ou itens podem ser estabelecidos com base em dados de utilização Web (por exemplo, através das sessões de utilizadores), ou com base no conteúdo associado às páginas (atributos de um produto, sessão a que pertence uma notícia, etc.). No caso baseado em conteúdos Web, os resultados podem ser grupos de filmes pertencestes a uma mesma sessão, enquanto que quando baseado em dados de mineração Web, o resultado será os grupos de páginas que costumam aparecer frequentemente em conjunto nas sessões Web - também podem ser utilizados para sugerir *hiperlinks* relacionados com os utilizadores, sendo esta sugestão proposta com base em navegações anteriores.

4.4 Regras de Associação

As regras de associação são uma das técnicas de mineração de dados mais exaustivamente estudas até ao momento. O objectivo da aplicação destas técnicas é a descoberta de associações entre *itens* que ocorrem frequentemente juntos. Desde que foram pela primeira vez introduzidas em 1993 por *Agrawal* [Agrawal and Srikant 94,] [Agrawal and Srikant 95], têm atraído bastante

atenção sobre elas próprias e, tal circunstância, fez com que surgissem inúmeros processos de investigação que originaram muitos algoritmos eficientes.

A aplicação de regras de associação pode ser definida da seguinte maneira: dado um conjunto de *itens* $I=\{i_1,i_2,..,i_n\}$ e um conjunto $T=\{t_1,t_2,...,t_n\}$ de transacções sobre uma base de dados, em que cada transacção t_i é um subconjunto de *I*, uma regra de associação é uma implicação da forma:

$$X \rightarrow Y$$

na qual $X \subseteq I$, $Y \subseteq I$ e a intersecção entre *X* e *Y* é o conjunto vazio. *X* e *Y* são conjuntos de *itens* chamado *itemsets*. Uma transacção t_i contêm uma *itemset X*, caso *X* seja um subconjunto de t_i. O suporte de um *itemset X* num conjunto de transacções *T*, é o número de transacções em *T* que contém *X*. Uma regra de associação é avaliada pelo seu suporte e pela sua confiança.

O suporte de uma regra $X \rightarrow Y$ é a percentagem de transacções em *T* que contêm $X \cup Y$, podendo ser visto como a probabilidade de *X* e *Y* aparecerem juntos na mesma transacção. Por outras palavras podemos dizer que o suporte determina com que frequência X e *Y* aparecem juntos na mesma transacção. O suporte é uma medida muito útil, porque caso seja muito baixo essa associação pode ter sido feita por acaso. Uma regra que cubra um número muito baixo de transacções pode não ser útil e não fazer qualquer sentido na área de negócio em questão.

$$Suporte(X \rightarrow Y) = |X \cup Y|$$

A confiança de uma regra $X \rightarrow Y$ é dada pela percentagem de transacções em *T* em que *Y* ocorre sabendo que *X* também ocorre. Ou seja é a probabilidade de *Y* aparecer numa transacção onde *X* apareça também. Isto pode ser visto como a probabilidade condicionada *P(Y|X)* e é dada pela seguinte expressão:

$$Confiança(X \rightarrow Y) = |X \cup Y| \,/\, |X|$$

Em que *|XUY|* representa o número de transacções em que *X* e *Y* aparecem em conjunto e *|X|* representa o número de transacções em que *X* aparece. A confiança determina assim a previsibilidade de uma regra. Caso a confiança seja muito baixa, não se pode inferir ou prever que *Y* ocorre quando *X* ocorre.

O objectivo da aplicação de regras de associação é, dado um conjunto de transacções *T,* descobrir todas as regras de associação em *T,* que apresentem valores de suporte e confiança maiores ou iguais aos valores que o utilizador estabelece como suporte mínimo e confiança mínima.

Os algoritmos de regras de associação estão normalmente divididos em duas fases: geração de um conjunto de *itemsets* frequentes e a geração das próprias regras de associação. Na primeira parte são gerados todos os *itemsets* frequentes com suporte superior ao valor mínimo para o suporte. Esta fase é a mais pesada em termos computacionais, uma vez que é necessário percorrer todo o conjunto de transacções. A segunda é mais simples e consiste em utilizar os conjuntos de *itemsets* gerados na fase anterior e calcular as regras de associação cuja confiança seja superior ao valor mínimo definido pelo utilizador. Um dos algoritmos mais conhecidos que usa este método para gerar regras de associação é o algoritmo *Apriori* [Agrawal and Srikant 94], [Han and Kamber 07].

A descoberta de regras de associação em transacções Web traz muitas vantagens para os administradores Web*,* dado que lhes permite descobrir relacionamentos que, anteriormente, estavam escondidos entre as páginas Web*.* Por exemplo, num *site* de noticias uma regra de associação com confiança muita alta, como por exemplo /país, /desporto/ -> /internacional/, indica que usualmente quem acede a páginas de notícias sobre o país e desporto tem uma probabilidade muita alta de também vir a aceder a notícias sobre o mundo. Este tipo de regras também é utilizado para melhorar a estrutura do *site*. Por exemplo se um *site* não tiver um *hiperlink* directo entre duas páginas *A* e *B*, a descoberta de uma regra de associação *A->B* indica que colocar um *hiperlink* directo de *A* para *B* pode ajudar os utilizadores a obterem a informação que necessitam. A análise de regras de associação, tanto sobre os produtos como sobre as páginas Web, é bastante utilizada em técnicas de personalização de *sites* Web e na criação de sistemas de recomendação [Herlocker et al 04], [Mobasher et al 01b], [Sarwar et al 00].

Um dos problemas da utilização de regras de associação em sistemas de recomendação é devido a que o sistema não pode dar recomendações, caso o conjunto de dados seja muito disperso (o que é muito frequente em aplicações de mineração de dados Web). A razão da grande dispersão de dados ocorre normalmente durante uma sessão quando um utilizador apenas visita uma pequena fracção das páginas disponíveis e, assim, torna-se difícil encontrar páginas em comum em várias

das sessões Web realizadas. Em [Sarwar et al 00] é proposto um sistema que reduz este problema.

4.5 Padrões Sequenciais

A aplicação de regras de associação não considera a ordem em que os *itens* aparecem numa transacção. Contudo, existem vários tipos de aplicações em que a ordem dos *itens* é importante. Por exemplo, numa análise de vendas de produtos é interessante saber qual a sequência de itens comprados, para que se possa saber qual a regra geral que um utilizador segue – por exemplo, primeiro compra uma consola e depois pode vir a comprar jogos para essa consola.

Na mineração de dados de utilização Web as técnicas de padrões sequenciais tentam encontrar padrões dentro das sessões, tais como a presença de um grupo de *itens* seguido por um outro *item* num determinado conjunto de sessões. Utilizando estas técnicas é possível prever padrões de visita futuros que vão ser úteis, por exemplo, para a colocação de avisos no *site* para avisar ou informar um determinado grupo de utilizadores sobre assuntos pelos quais costumam ter algum interesse. No contexto de mineração de dados Web este tipo de técnicas podem também ser utilizadas para calcular caminhos mais frequentes.

No contexto de mineração dos dados de utilização Web, padrões sequenciais capturam as sequências de páginas mais visitadas pelos utilizadores. Os padrões sequenciais podem ser vistos como sequências de páginas que ocorrem frequentemente num grande número de transacções. Seja $P=\{p_1,p_2,...,p_n\}$ um conjunto de páginas de um *site* Web, uma sequência $S=<s_1,s_2,...,s_n>$ ocorre numa transacção $T=<p_1,p_2,...,p_m>$ em que $n<m$, caso S seja uma subconjunto de *T*. Contudo a ordem de s tem de ser preservada na transacção *T,* ou seja qualquer par s_i, si_{+1} deve aparecer em posições consecutivas na transacção *T*.

A visualização de transacções Web, como uma sequência de páginas permite utilizar um grande número de modelos para descobrir e analisar padrões de navegação. Uma destas abordagens é modelar as actividades de navegação num *site* Web através de cadeias de *Markov*. As cadeias de *Markov* podem ser vistas como um grafo em que os estados representam as páginas e os arcos são a probabilidade de se passar de uma página para a outra. Esta probabilidade é calculada através da contagem do número de utilizadores que passam de uma página para a outra. Esta representação permite o conhecimento de informações tanto sobre o *site* como sobre os

utilizadores. Por exemplo permite calcular qual a probabilidade de uma pessoa ler uma notícia sobre futebol sabendo que antes leu uma notícia sobre economia. As cadeias de *Markov* foram propostas como um mecanismo para previsão de *hiperlinks* [Deshpande and Karypis 04], [Sarukkai 00]. O objectivo deste tipo de aplicações é conseguir prever qual vai ser a próxima página pedida por um utilizador, com base no comportamento em sessões anteriores de utilizadores com o mesmo perfil. As cadeias de *Markov* são também utilizadas para prever quais as sequências de páginas com maior probabilidade de ocorrer num *site* Web [Borges and Levene 99].

Formalmente, uma cadeia de *Markov* é caracterizada por um conjunto de estados $\{s_1, s_2, s_3,...,s_n\}$ e uma matriz de transacção T, de ordem n, que nos dá a probabilidade de passar de um estado para outro. As cadeias de *Markov* são especialmente úteis para criar sistemas de previsão sobre sequências de páginas contínuas.

Um conceito associado com as cadeias de *Markov* é o da ordem da cadeia. No caso da aplicação de técnicas de mineração de dados Web, a ordem da cadeia de *Markov* representa o número de páginas que são utilizadas para prever qual será a próxima página consultada pelo utilizador. Assim podemos dizer que numa cadeia de *Markov* de ordem 1 usa-se apenas a página actual que o utilizador está a visualizar para tentar prever a próxima página. Já numa cadeia de *Markov* de ordem *n*, utilizamos as *n*-1 páginas anteriores para conseguir prever qual a próxima página que o utilizador pretende visualizar. Quanto maior for a ordem das cadeias de *Markov,* maior será o número de estados. Logo quando maior for a ordem, pior vai ser o desempenho do sistema, mas também maior será a probabilidade de acertar na página seguinte. Outras das características associadas com as cadeias de *Markov* é o facto de elas poderem ser incrementais. Isto significa que é possível acrescentar informação sobre novas sessões a uma cadeia já construída, não sendo necessário reconstrui-la novamente a partir do zero. Para se ver como uma cadeia de *Markov* pode modelar um conjunto de transacções Web, observe-se o exemplo apresentado de seguida (Tabela 4.1), que poderia representar as sessões de um jornal de noticias *online*

ID	Sessão
1	/Índex/ ->/País/ ->/Desporto ->/Economia/ ->/Mundo/
2	/Índex/ ->/País/ ->/Desporto/
3	/Índex/ ->/País/ ->/Desporto ->Mundo/
4	/País/ -> /Economia/->/Mundo/
5	/País/ -> /Economia/ -> /Mundo/ -> /Desporto/
6	/País/ -> /Economia/ -> /Índex/ -> /Mundo/
7	/Economia/ -> /Índex/ -> /País/ -> /Mundo/

Tabela 4.1: Conjunto de transacções Web

A cadeia de *Markov* de ordem um que vai representar as transacções que aparecem na Tabela 4.1 vai ser construída da seguinte maneira. Primeiro vai ser colocado na cadeia um estado inicial que vai ser representado por *S.* Depois, a partir deste estado vai sair uma ligação para cada uma das páginas que aparecem no *log* de transacções. A probabilidade de passar do estado inicial para qualquer uma das outras páginas representa a probabilidade inicial de cada página. De seguida, vamos colocar um estado final representado por *F*. Todas as páginas que apareçam como sendo a última página numa sessão vão ter uma ligação para o estado final. A probabilidade de passar de uma página qualquer para o estado final representa a probabilidade do utilizador sair do *site* depois de visualizar essa página. A probabilidade inicial de uma página *P* é, dada pela seguinte equação:

$$ProbInic(P) = |P|/total\_pedidos.$$

em que |*P*| representa o número de vezes que a página *P* foi pedida ao servidor e total_pedidos representa o número total de pedidos feitos ao servidor. A probabilidade de se passar de uma página *A* para uma página *B* é dada pela seguinte equação:

$$P(A->B) = S(B|A)/S(A)$$

Nesta equação, S(B|A) representa o número de sessões em que a página *B* apareceu a seguir à página *A,* e o S(A) representa o número de pedidos da página *A* feitos ao servidor, ou seja o numero de sessões em que a página *A* aparece.

Observando o exemplo de transacções da Tabela 4.1 podemos calcular as probabilidades da seguinte maneira: a probabilidade de passar do estado *S* para o estado /*index*/ é de 5/27 = 0.19, a probabilidade de ir da páginas /*país*/ para a página /*economia*/ é de 3/7=0.42 e, por fim, a probabilidade de ir do página /*mundo*/ para o estado final é de 5/6 = 0.83. De seguida na Figura 4.5 apresentamos a cadeia de *Markov* gerada a partir das transacções dadas na Tabela 4.1.

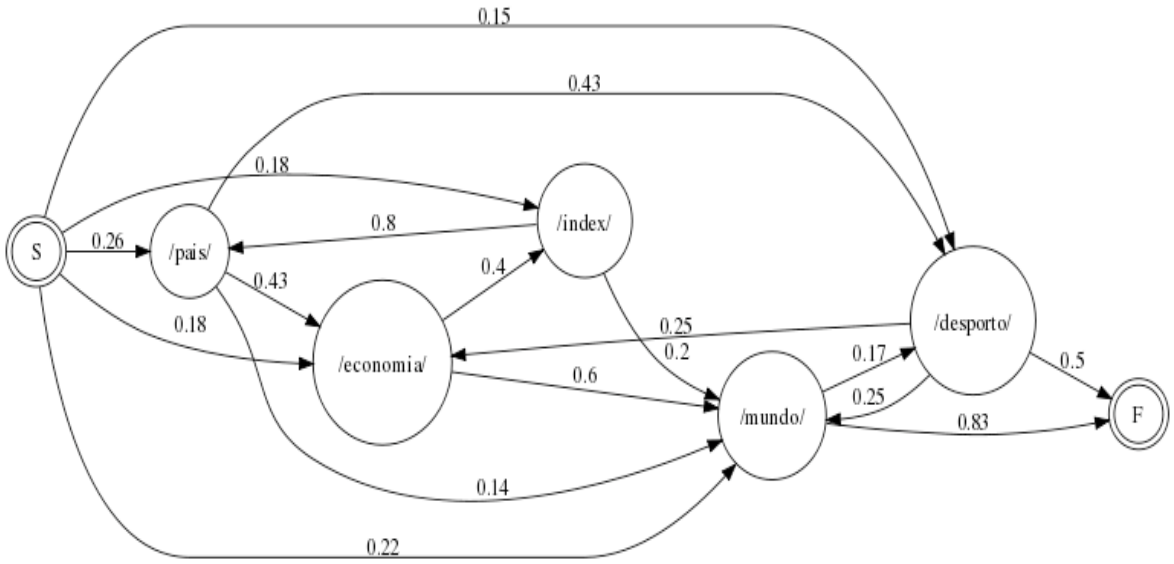


Figura 4.5 : Visualização de uma cadeia de Markov

Depois de construída a cadeia de *Markov*, passamos agora à segunda parte: o cálculo dos caminhos mais frequentes. Neste caso, como a cadeia representa páginas Web, falta calcular as sequências de páginas que aparecem frequentemente nas sessões. Uma vez que as cadeias de *Markov* são implementadas como grafos pode-se usar algoritmos de pesquisa em grafos para calcular todos os caminhos possíveis na cadeia. Para isso, podemos usar algoritmos tanto de pesquisa em largura, *Breath-First Search* [Weiss 93], como de pesquisa em profundidade, *Depth-First Search* [Tarjan 62].

Dois conceitos muito importantes, relacionados com a descoberta de caminhos mais frequentes, são o suporte e a confiança. O suporte representa a probabilidade inicial de um caminho, sendo muitas vezes definido como a média das percentagens iniciais de todas as páginas Web. Por sua vez, a confiança é a probabilidade de um utilizador percorrer um dado caminho e pode ser calculada através da multiplicação das probabilidades de cada uma das ligações da cadeia. De

seguida na Figura 4.6 apresentamos um pequeno exemplo da geração dos caminhos mais frequentes para suporte igual a 10% e confiança igual a 40%.

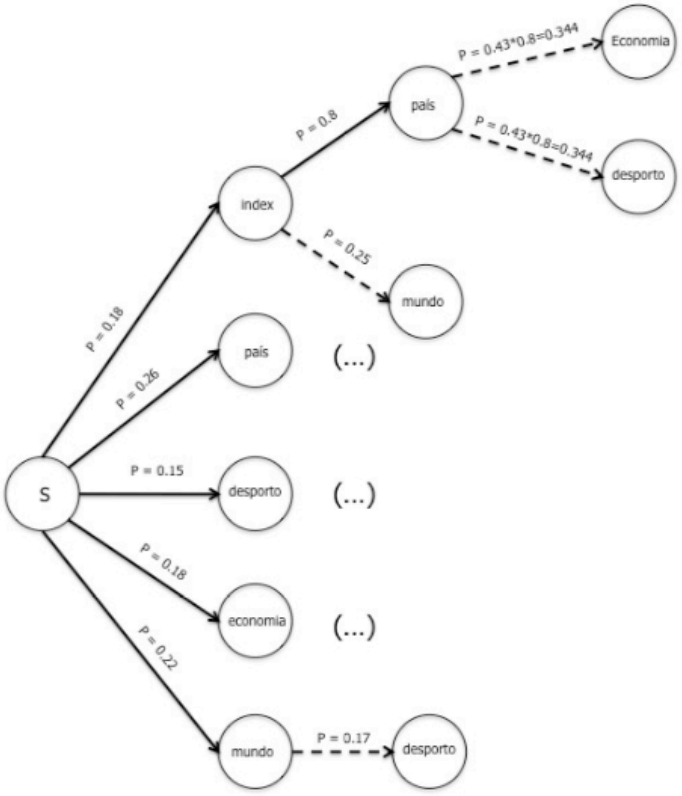


Figura 4.6: Árvore de caminhos mais frequentes suporte = 0.1 e confiança = 0.4

Na Tabela 4.2 podemos ver os caminhos mais frequentes gerados através da cadeia de *Markov* apresentada na Figura 4.5.

Caminhos	**Probabilidade**
/país/ -> /economia/	0.43
/país/ -> /desporto/	0.43
/economia/ -> /índex/	0.4
/economia/ -> /mundo/	0.6
/índex/ -> /país/	0.8

Tabela 4.2: Caminhos mais frequentes com suporte 0.1 e confiança 0.4

4.6 Classificação

A classificação tem tido um grande sucesso em aplicações do mundo real. Esta técnica é utilizada em quase todos os domínios, desde a Inteligência Artificial até á Medicina. Este tipo de aprendizagem é semelhante ao processo de aprendizagem dos seres humanos através de experiências passadas, de forma a adquirir conhecimento que lhes permita melhorar a forma como vai reagir em situações futuras. O objectivo das técnicas de classificação é construir um modelo que permita classificar novas instâncias, com base no conhecimento adquirido através de exemplos anteriores.

Um conjunto de treino, sobre o qual podem ser aplicadas técnicas de classificação, necessita de ter um conjunto de atributos de previsão e um atributo especial, o atributo objectivo. Para técnicas de classificação o atributo objectivo tem de ter um conjunto de valores predefinidos. Cada um dos valores do atributo objectivo é chamado classe. Cada uma das estâncias do conjunto de treino é classificada de acordo com o valor do atributo objectivo. Um conjunto de treino é, simplesmente, uma tabela relacional, em que cada uma das instâncias corresponde a um acontecimento passado. Podemos, por isso, dizer que um conjunto de treino consiste, basicamente, num conjunto de exemplos ou instâncias. Dado um conjunto de treino D, o objectivo de um mecanismo de classificação é construir um classificador, que relacione os vários atributos de previsão com o atributo objectivo. O classificador pode depois ser usado para classificar novas instâncias de acordo com o valor dos vários atributos de previsão.

No domínio da Web, um dos interesses da aplicação de técnicas de classificação é desenvolver um perfil de utilizadores que pertencem a uma determinada classe ou a uma categoria predefinida. Isto requer a extracção e a selecção das características que melhor descrevem cada uma das classes. A classificação pode ser feita através da utilização de algoritmos de aprendizagem supervisionada, tais como as árvores de decisão [Han and Kamber 07], os classificadores *Naive Bayes* [Han and Kamber 07] ou os classificadores *k-nearest neighbo*r [Han and Kamber 07], entre outros. É também possível utilizar a classificação em conjunto com *clustering* e regras de associação. Nestes casos, começamos por utilizar os clusters para dividir o conjunto de treino e, de seguida, utilizamos os algoritmos de classificação sobre o conjunto, sendo o atributo objectivo o

cluster a que pertence. O modelo de classificação gerado serve depois para classificar novas instâncias que seguem ao conjunto, atribuindo-lhes o cluster ao qual pertencem.

As técnicas de classificação desempenham um papel bastante importante na análise de informação Web, uma vez que permitem catalogar os utilizadores de acordo com um conjunto de métricas predefinidas. Por exemplo dado um conjunto de sessões feitas por um utilizador é possível calcular a quantidade de produtos que ele comprou na loja. Um modelo de classificação pode ser construído com base nestes dados. Os utilizadores vão, assim, ser classificados em dois grupos: aqueles que tem uma grande probabilidade de comprar algum produto e aqueles que muito provavelmente não vão adquirir nada. Esta classificação tem em conta tanto os atributos demográficos do cliente, como as actividades de navegação dos clientes.

Outra importante aplicação das técnicas de classificação é na construção de sistemas de recomendação. A maioria dos sistemas de recomendação usa o classificador *k-nearest neighbo*r para prever se um utilizador se vai interessar ou não por um determinado *item*, medindo para isso a proximidade entre o utilizador actual (que pode ser simplesmente a lista de paginas visitadas até ao momento) e os perfis de utilizadores passados. Deste modo pode-se encontrar utilizadores anteriores com as mesmas características ou com as mesmas preferências do utilizador, e, com este conhecimento, recomendar-lhe as páginas que o utilizador anterior visitou [Herlocker et al 04], [Mobasher 05], [Pierrakos et al 03].

Capítulo 5

Avaliação Experimental

A avaliação das técnicas de identificação de perfis apresentadas anteriormente é uma fase muito importante, uma vez que é a através dela que vamos consolidar o estudo dos métodos apresentados. A escolha de casos de estudo para a Web nem sempre é uma tarefa fácil, já que a diversidade de cenários é tão grande que dificulta muito essa tarefa. Não é possível encontrar dois *site*s Web idênticos. Cada *site* apresenta características que o tornam único. Sendo assim, o mecanismo de identificação de perfis que mais se adapta a um determinado *site* pode não ser o mesmo para todos os *sites*. Mesmo tendo o cuidado de escolher casos de estudo que representem um dado cenário, a fase de experimentação deve ter em conta as características específicas do *site* em estudo, bem como os objectivos que se pretendem atingir. Neste capítulo vamos apresentar e descrever um caso de avaliação experimental da aplicação das técnicas estudadas nos capítulos anteriores. Mais especificamente, vamos começar por descrever o caso de estudo e, em seguida, a aplicação dos métodos estudados no capítulo anterior. Depois da aplicação dos métodos passamos a análise dos resultados obtidos.

5.1 Descrição do Site Alvo de Estudo

O caso de estudo utilizado, para avaliar as várias técnicas utilizadas na identificação de perfis de utilização Web corresponde ao *site* do *repositóriUM* [WWW 9]. O *repositóriUM* é o repositório

institucional de Universidade do Minho constituído com o objectivo de armazenar, preservar, divulgar e dar acesso à produção intelectual da Universidade do Minho [WWW 10] em formato digital. O *repositóriUM* pretende reunir, num único sitio, o conjunto de publicações científicas da UM contribuindo, desta forma, para o aumento da sua visibilidade.

O *repositóriUM* está organizado em torno de comunidades, que correspondem às unidades orgânicas (Escolas, Departamentos e Centros de Investigação). Cada comunidade pode organizar os seus documentos em colecções, podendo existir um número ilimitado de documentos dentro de cada colecção. Cada comunidade possui uma página própria com informação, notícias e *hiperlinks* que reflectem os seus interesses, bem como uma listagem das colecções dentro da comunidade.

A organização geral do *site* do *repositóriUM* segue uma estrutura típica de um *site* bem estruturado. Tem uma página principal, através da qual podemos aceder a todas as páginas do *site*. Esta, por sua vez, está dividida em três partes principais: do lado esquerdo apresenta um menu que disponibiliza as várias opções de navegação no *site*, no lado direito, na parte superior, tem um campo para efectuar pesquisas, e na parte inferior um bloco de notícias sobre o *repositóriUM* e uma lista com os últimos documentos depositados. O menu do lado esquerdo está dividido em quatro secções: percorrer, entrar, ajuda e sigam-nos. A secção percorrer, permite aos utilizadores visualizar uma lista de todos o documentos por autor, comunidade, data de publicação entre outros. A secção entrar, apenas pode ser utilizada por utilizadores registados e tem disponíveis vários serviços, como por exemplo: aceder à área pessoal, editar conta, etc. A secção *ajuda* disponibiliza informações úteis aos utilizadores, tendo disponível tópicos com: *FAQs,* uma secção com perguntas e repostas normalmente feitas por utilizadores; *guias*, que disponibilizam informações sobre como utilizar o *repositóriUM;* e *Copyright,* que contém informações sobre as regras de cópia e distribuição associados aos documentos publicados no *site*. Por fim, na secção *sigam-nos* temos um *hiperlink* para a página do *Twitter* do *repositóriUM,* no qual se podem acompanhar as últimas novidades sobre o *site*.

Figura 5.1: Página principal do *site* do *RepositoriUM*.

O público alvo deste *site* é, sobretudo, os estudantes, sejam eles parte do corpo da Universidade do Minho ou não. Contudo, existem outros tipos de utilizadores que acedem com alguma regularidade ao *site*. À partida, podemos identificar quatro grupos de utilizadores distintos: os alunos da UM, administradores, utilizadores registados e outros. O grupo de alunos da UM, caracteriza-se por serem utilizadores que têm conhecimento da existência do *site* e o utilizam para pesquisar artigos sobre os mais diversos assuntos. Relativamente ao grupo dos administradores, estes utilizam o *site* sobretudo para consultar estatísticas sobre a utilização do *site* e verificar a existência de algum problema nas páginas principais. Quanto ao grupo dos utilizadores registados, são na sua maioria professores ou investigadores, que têm autorização para publicar os seus artigos no repositório. Por fim, temos o grupo dos outros, onde se encaixam os utilizadores que não fazem parte de nenhum dos grupos anteriores. Estes, são normalmente utilizadores que entram no *site* referenciados por um motor de pesquisa, o que significa que fizeram algum tipo de pesquisa num determinado motor de busca e foram posteriormente redireccionados para o *site*. Este tipo de utilizadores pode depois vir a utilizar os mecanismos de pesquisa do *site* para encontrar outros artigos que sejam do seu interesse.

Em termos de volume de acessos, o *site* do *repositorioUM,* não é muito homogéneo. Isto acontece, uma vez que a maioria dos utilizadores são estudantes e, sendo assim, durante a época de exames (Janeiro ou Junho) há um aumento do número de acessos, sucedendo o caso inverso durante a época de férias escolares no qual há, obviamente, um decréscimo no número de acessos. O período de maior acesso ao *site* vai desde as dez da manhã até cerca da meia-noite, notando-se uma quebra no número de pedidos durante período da meia-noite ás nove da manha, período este correspondente à altura em a maioria dos utilizadores do *site* estão a dormir. No período de maior acesso ,verifica-se um pico entre as 15 e as 19 horas, sucedendo uma ligeira quebra entre as 20 e as 21 horas, voltando a aumentar depois das 22 horas. Esta quebra justifica-se pois este período corresponde normalmente ao período de jantar da grande maioria dos utilizadores.

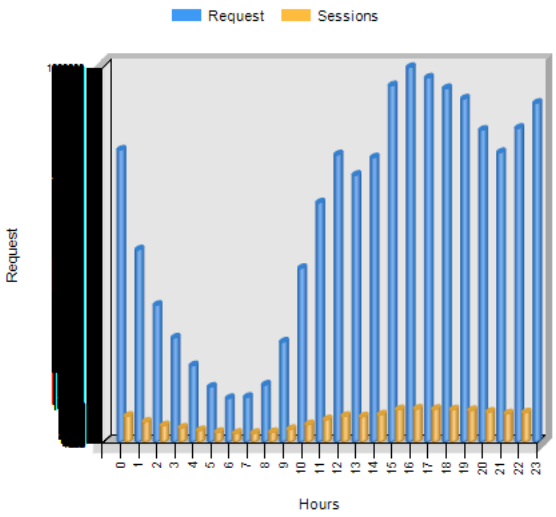


Figura 5.2 Distribuição de pedidos e sessões por hora

A maioria dos acessos efectuados ao *repositorioUM* são provenientes de Portugal. Contudo, existem outros países com uma percentagem significativa de acessos no *site*. Alguns desses países são de língua portuguesa como é o caso do Brasil, havendo outros como por exemplo os Estados Unidos que não o são. O acesso por parte de países de língua portuguesa é fácil de justificar, uma vez que muitos dos artigos publicados são em português. Já o caso de haver muitos acessos por parte de países como os Estados Unidos é devido a acessos feitos por agentes automáticos. Estes agentes automáticos, são normalmente programas utilizados pelos motores de pesquisa para indexar a informação contida no *site*.

5.2 As Fontes de Dados

Como já mencionado anteriormente, usualmente utilizam-se várias fontes de dados em processos de mineração de dados de utilização Web. Neste caso, em particular, foram utilizadas cinco fontes de dados para alimentar o *data webhouse em questão*. As fontes de dados utilizadas foram as seguintes:

1. Ficheiros de *logs* do servidor Web onde o *site* se encontra alojado.
2. Base de dados com informação geográfica acerca dos *IPs.*
3. Lista de agentes automáticos conhecidos.
4. Lista de motores de pesquisa conhecidos.
5. Lista com as extensões de ficheiros, o nome associado assim como a classe a que pertencem.

Os ficheiros do *log* gerados pelo servidor Web seguem o formato ECLF. De seguida, na Figura 5.3, apresentamos um pequeno extracto dos ficheiros de *log* do servidor.

```
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/UM3_over.gif HTTP/1.0" 200 2208 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/login.gif HTTP/1.0" 200 597 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/home.gif HTTP/1.0" 200 389 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/contactos.gif HTTP/1.0" 200 681 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/faqs.gif HTTP/1.0" 200 560 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
201.89.105.58 - - [05/Jan/2008:16:16:16 -0500] "GET /image/email_web.gif HTTP/1.0" 200 615 "http://
repositorium.sdum.uminho.pt/handle/1822/2999" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)"
```

Figura 5.3: Excerto de um ficheiro de logs

A localização geográfica de um pedido é feita com recurso ao *IP* do pedido. A empresa MaxMind [WWW 11], disponibiliza gratuitamente no seu *site* uma base de dados com a associação entre *IPs* e o seu respectivo país. Esta base de dados é fornecida em formato csv. Cada linha do ficheiro

contém um intervalo de *IPs*, dando o limite superior e inferior do intervalo, e o país ao qual essa gama de *IPs* foi atribuída. A informação deste ficheiro vai ser carregada para uma base de dados de forma a facilitar a comparação entre *IPs*. Na Figura 5.4 podemos ver um exemplo do ficheiro de *IPs.*

```
"62.96.128.32","62.96.128.79","1046511648","1046511695","DE","Germany"
"62.96.128.80","62.96.128.103","1046511696","1046511719","GB","United Kingdom"
"62.96.128.104","62.96.128.111","1046511720","1046511727","DE","Germany"
"62.96.128.112","62.96.128.199","1046511728","1046511815","GB","United Kingdom"
"62.96.128.200","62.96.128.231","1046511816","1046511847","DE","Germany"
"62.96.128.232","62.96.128.247","1046511848","1046511863","GB","United Kingdom"
"62.96.128.248","62.96.128.255","1046511864","1046511871","DE","Germany"
"62.96.129.0","62.96.129.31","1046511872","1046511903","GB","United Kingdom"
"62.96.129.32","62.96.129.47","1046511904","1046511919","DE","Germany"
"62.96.129.48","62.96.129.55","1046511920","1046511927","GB","United Kingdom"
"62.96.129.56","62.96.129.63","1046511928","1046511935","DE","Germany"
"62.96.129.64","62.96.129.95","1046511936","1046511967","GB","United Kingdom"
```

Figura 5.4: Ficheiro de IPS

A lista de motores de pesquisa conhecidos está guardada numa tabela da base de dados. Desta lista fazem parte tanto os motores de pesquisa mais conhecidos, como é o caso do *Google* [WWW 12], como aqueles menos conhecidos e que apenas são utilizados nos seus países de origem, como é o caso do Sapo [WWW 13]. Esta lista é mantida pelo administrador do *data webhouse* e necessita de ser regularmente actualizada, de forma a fornecer sempre a informação mais actualizada. De seguida na Figura 5.5 podemos ver uma parte da tabela onde estão guardados os motores de pesquisa conhecidos.

	SEARCH
15	busca.uol
16	radix
17	eniro
18	search.yahoo
19	busca.yahoo
20	sea.search
21	goggle
22	guggle
23	search
24	scholar
25	citeseerx
26	sapo

Figura 5.5: Tabela com motores de pesquisa conhecidos

A lista com extensões de ficheiros, bem como a lista de motores de pesquisa, é mantida numa tabela da base de dados. Nesta tabela estão incluídos os tipos de ficheiros mais conhecidos, a sua extensão e a família a que pertencem. Esta lista (Figura 5.6) é mantida pelo administrador do *site* Web, sofrendo poucas alterações ao longo do tempo.

	prefix	httpRequestClass	explanation	httpRequestClass	definition
39	fig	IM	A standard image file.	IM	Image files
40	gif	IM	A standard image file. (Graphic Interchange File) C...	IM	Image files
41	gz	CE	GNU zip file. A popular format for compressing a si...	CE	Compressed and Encoded files
42	gzip	CE	GNU zip file. A popular format for compressing a si...	CE	Compressed and Encoded files
43	h	AS	C File.	AS	Ascii documents
44	hqx	CE	ASCII-encoded binary file. A common format on th...	CE	Compressed and Encoded files
45	ht	AS	HyperTerminal Data File.	AS	Ascii documents
46	htm	ID	htm is synonymous but not identical to .html Bot...	ID	Internet files and Downloadable
47	html	ID	Hyper Text Mark-up Language. This file is a simpl...	ID	Internet files and Downloadable
48	html2	ID	Hyper Text Mark-up Language. This file is a simpl...	ID	Internet files and Downloadable
49	ico	IM	Windows icon.	IM	Image files

Figura 5.6: Ttabela com as extensões dos ficheiros.

Por fim, a lista com os agentes automáticos. Tal como as duas listas apresentadas anteriormente, esta também é guardada num tabela da base de dados, que é preenchida, inicialmente, com a informação fornecida pela página *robotstxt.org* [WWW 14]. Esta organização mantém uma lista actualizada, com o nome e algumas características dos agentes automáticos conhecidos. Uma dessas características, talvez das mais importantes, é o texto identificador do robot, que é o valor enviado no campo *User Agent* de um pedido HTTP. Esta característica vai ser guardada na tabela, e vai permitir que durante a fase de processamento dos *logs* se consigam identificar correctamente os pedidos feitos pelos *crawlers.*

	ua	tipo
11	googlebot	crawler
12	feedfetcher-google	crawler
13	voilabot	crawler
14	twiceler	crawler
15	goldfire	crawler
16	grub	crawler
17	cazoodlebot	crawler
18	rome client	crawler
19	yahoo! slurp	crawler
20	yahoo! de slurp	crawler
21	exabot	crawler
22	msnbot	crawler

Figura 5.7: Tabela com os *logs* conhecidos

5.3 Modelação Dimensional

O modelo dimensional construído para integrar os dados provenientes dos ficheiros de *logs* e de outras fontes adicionais é composto por: uma tabela de factos, *TF_sessions*, seis dimensões: *Data, Tempo, Request, Referrer, Agent, ComputadorUtilizador*, uma tabela ponte *TP_sessions*, que estabelece a relação entre uma sessão e os pedidos feitos nessa sessão, e por catorze medidas.

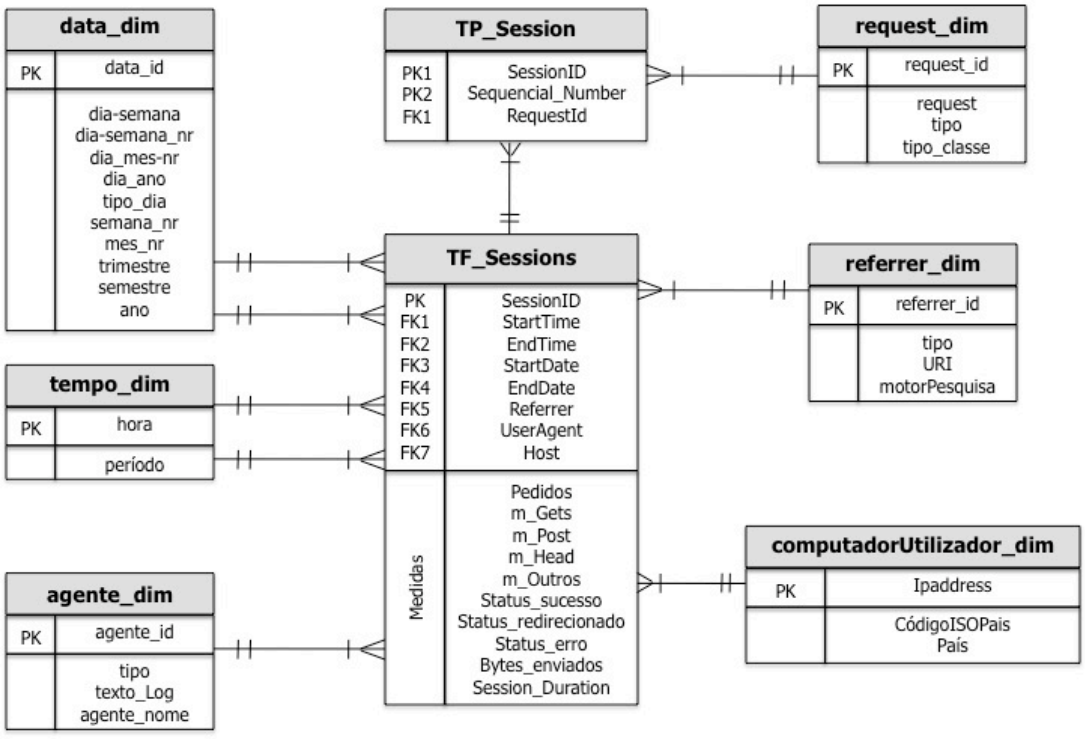


Figura 5.8 : *Datamart* para sessões Web

A tabela de factos *TF_Sessions,* guarda o registo de todas as sessões identificadas nos ficheiros de *log*. É composta por vinte atributos, sendo que desses atributos sete são chaves estrangeiras para as várias dimensões, e dez são medidas, métricas que permitem analisar os valores das sessões sobre os mais diversos pontos de vista.

A identificação de sessões é feita utilizando a técnica H2, apresentada na sessão 3.6. Para isso, primeiro identificam-se todos os pedidos feitos pelo mesmo utilizador, de seguida, ordenam-se os pedidos por data e hora, atribuindo-se um número de sessão a todos os pedidos. Caso existam dois pedidos consecutivos efectuados pelo mesmo utilizador, com um intervalo de tempo superior a quinze minutos atribuímos uma nova sessão a todas os pedidos ocorridos a partir desse ponto.

Atributos	**Descrição**	**Tipo**
SessionID	Identificador de Sessão	Chave Primária
StartTime	Data e Hora do inicio da sessão	
EndTime	Data e Hora do fim da sessão	
StartDate	Data do inicio da sessão	Chave Estrangeira
EndDate	Data do fim da sessão	Chave Estrangeira
StartHour	Hora do inicio da sessão	Chave Estrangeira
EndHour	Hora do fim da sessão	Chave Estrangeira
Referrer	*Site* que referenciou a página	Chave Estrangeira
UserAgent	Programa utilizado para efectuar os pedidos	Chave Estrangeira
Host	IP do utilizador que efectuou o pedido	Chave Estrangeira
Pedidos	Número de pedidos feitos na sessão	Medida
m_gets	Número de pedidos com o método Get	Medida
m_post	Número de pedidos com o método Post	Medida
m_head	Número de pedidos com o método Head	Medida
m_outros	Número de pedidos com outros métodos	Medida
Status_sucesso	Número de pedidos respondidos com sucesso	Medida
Status_redirecionado	Número de pedidos redireccionados	Medida
Status_Erro	Número de pedidos cuja resposta foi erro	Medida
Bytes_enviados	Número de bytes enviados pelo servidor	Medida
Session_Duration	Tempo de duração da sessão	Medida

Tabela 5.1: Caracterização da Tabela de Factos de sessões Web

As dimensões Data e Tempo, as duas dimensões mais simples de perceber, caracterizam a altura em que a sessão ocorreu. A dimensão data caracteriza o dia em que a sessão ocorreu, enquanto a sessão tempo caracteriza a hora. Estas dimensões são alimentadas apenas uma vez, antes de o

data webhouse começar a integrar os ficheiros de *log.* Para o povoamento da dimensão tempo geramos os registos para as vinte e quatro horas do dia, para a dimensão data, determinamos o período de validade do *Dw*eb, e geramos os registos dos dias necessários .No caso estudado o tempo de vida atribuído ao *Dweb* foram 10 anos.

Atributo	**Descrição**	**Exemplo**
Hora	Chave Identificadora da dimensão tempo. Indica a hora	8
Período	Período do dia.	"Manha

Tabela 5.2: Caracterização da dimensão Tempo

Atributo	**Descrição**	**Exemplo**
Data_id	Chave identificadora do dia. Indica o dia	01-01-08
Dia Semana	Nome do dia semana	"Terça"
Dia_Semana_nr	Número do dia da semana	3
Dia_mes_nr	Número do dia no mês	1
Dia_ano_nr	Número do dia no ano	1
Tipo_Dia	Fim-de-semana ou dia de trabalho	"Semana"
Semana_nr	Número da semana no ano	1
Mês	Nome do mês	Janeiro
Mês_nr	Número do mês	1
Quarter	Trimestre correspondente a data	1
Semestre	Semestre corresponde a data	1
Ano	Ano da data	2008

Tabela 5.3: Caracterização da dimensão Data

A dimensão *computadorUtilizador* caracteriza o utilizador que fez os pedidos ao *site*. Uma vez que neste caso não é possível identificar o utilizador, apenas se caracteriza o local de onde é feito o pedido. O preenchimento desta tabela é feito através do *IP* do utilizador que fez o pedido. A resolução dos *IPs,* é feita com recurso a uma tabela auxiliar, onde se encontram guardados vários intervalos de *IPs*, juntamente com o país a que foram atribuídos. Esta tabela é preenchida com recurso a um ficheiro de texto disponível na Web, que tem a alocação de *IPs* por país.

Atributo	Descrição	Exemplo
ipAdress	IP do cliente que fez o pedido	89.84.45.174
codigoISOPaís	Código do país do cliente	"PT"
País	País de onde foi feito o pedido	"Portugal"

Tabela 5.4: : Caracterização da dimensão computadorUtilizador

A dimensão *Agent* caracteriza os programas utilizados para aceder ao *site* Web. Nos casos em que os pedidos são feitos por navegadores Web, identifica qual o navegador utilizado. Nesta dimensão, é feita a classificação dos utilizadores em agentes automáticos e utilizadores regulares. Esta separação é feita com recurso a uma lista de agentes automáticos previamente conhecidos, comparando-se o valor do campo *User Agent* com os valores da lista caso esse valor esteja na lista a sessão é atribuída a um agentes automáticos.

Atributo	Descrição	Exemplo
agentID	Atributo identificador do agente.	1
Tipo	Tipo de agent	"crawler"
agent_name	Nome do agent	"googleBot"
texto_log	Texto existente no ficheiro de log referente ao agente	

Tabela 5.5: Caracterização da dimensão *Agent*

A dimensão *referrer* é preenchida através do valor do campo *referrer* do ficheiro de *logs*. Esta dimensão, caracteriza o *site* que referenciou o *site* Web em estudo, indicando-nos qual o *site* que o utilizador visitou antes de entrar nos *site*. Nesta dimensão, existe um campo motor de pesquisa que indica se o *referrer* é ou não motor de pesquisa. Este campo é preenchido com recurso a uma tabela onde se encontram os motores de pesquisa mais conhecidos.

Atributo	Descrição	Exemplo
Referrer_id	Atributo identificador do referrer.	1
Tipo	Diz se o referrer é uma página do próprio *site*, ou uma página externa.	"externa"
URI	Endereço WWW do *site* que referenciou	"www.google.pt"
Motor_pesquisa	Indica se o referente é ou não um motor de busca	"y"

Tabela 5.6: Caracterização da dimensão Referente

A dimensão *request* caracteriza o pedido efectuado ao *site*. Este campo é preenchido através do valor do campo *request* do ficheiro de *logs.* A identificação do tipo de ficheiro pedido é feita com recurso a uma tabela onde se encontram os tipos de ficheiros mais comuns, assim como as suas extensões e o grupo a que pertencem. Comparando a extensão do pedido com as extensões na tabela obtemos qual o tipo de ficheiro pedido.

Atributo	Descrição	Exemplo
Request_id	Atributo identificador do pedido.	1
Request	Texto do pedido que ficou registado no ficheiro de *log.*	"/"
Tipo	Tipo de ficheiro pedido	"html"
Classe	Classe a que pertence o ficheiro pedido	"IM"

Tabela 5.7: Caracterização da dimensão *Request*

Por último a tabela ponte *TP_sessions*, estabelece a ligação entre a *TF_sessions* e a dimensão *request*. Esta tabela existe porque uma sessão pode ter mais do que um pedido, e um pedido pode fazer parte de mais do que uma sessão. Esta tabela, guarda todos os pedidos que foram efectuados numa sessão, assim como a ordem do pedido na sessão.

Atributo	Descrição	Exemplo
Session_id	Identifica a sessão	1
Request_id	Identifica o pedido	"externa"
número sequencial	Diz-nos a ordem do pedido dentro da sessão.	"www.google.pt"

Tabela 5.8: Caracterização da Tabela Ponte

5.4 O Processo de Extracção, Transformação e Integração dos Dados

A implementação do *Dweb* usada para recolher os dados de *log* foi feita sobre uma base de dados Microsoft SQL Server 2008 para o sistema operativo Windows XP. O fluxo de carregamento de dados foi implementado utilizando o Microsoft SQL Server Integration Services (SSIS). De seguida apresentamos um esquema global do processo de *ETI* para alimentação do *Dweb* construído.

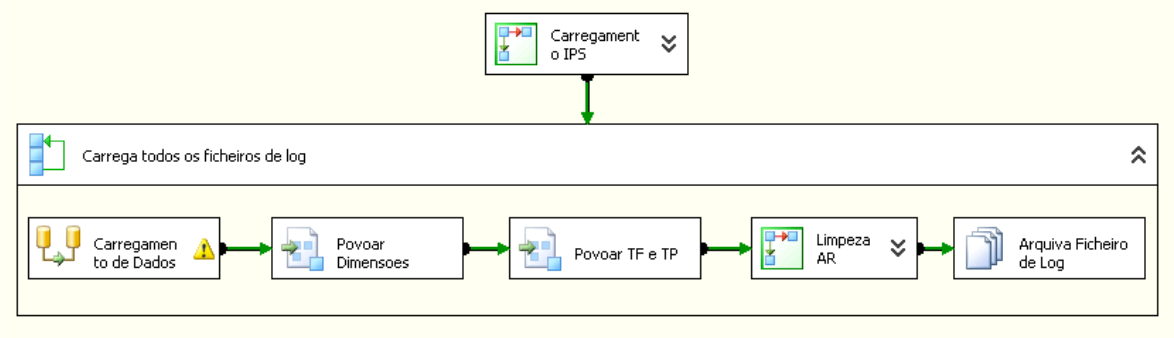


Figura 5.9: Esquema global do processo de ETI

Como podemos ver pela figura anterior, o processo de recolha, transformação e integração (ETI) dos dados está dividido em cinco processos:

- P1: Recolha e carregamento dos dados referentes a base de dados geográfica.
- P2: Recolha e carregamento na área de retenção, dos dados dos ficheiros de *log* do servidor Web.
- P3: Processamento das dimensões na área de retenção, e passagem dos dados para o *Dweb*.
- P4: Criação da tabela de factos e da tabela Ponte na área de retenção e posterior passagem dos dados para o *Dweb*.
- P5: Limpeza das tabelas da área de retenção.

O primeiro processo (P1) (Figura 5.10) começa por fazer o download do ficheiro com informação geográfica associada aos *IPs*, sendo este feito com recurso ao método *wget*. Posteriormente, a informação contida no ficheiro é carregada para uma tabela na área de retenção onde vai ser mantida - novos dados substituirão sempre aqueles que já se encontravam na tabela.

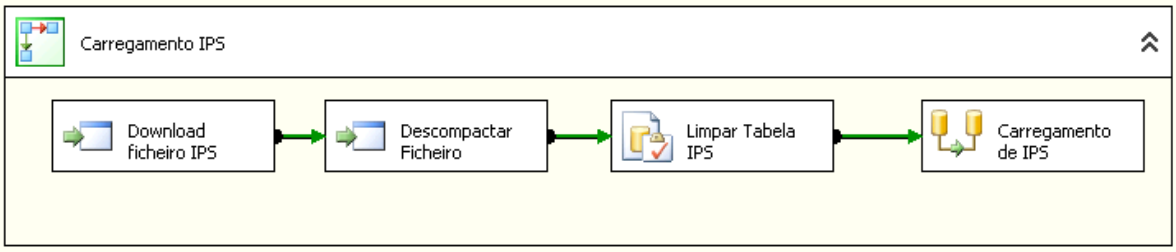


Figura 5.10: P1- Extracção e carregamento dos ficheiros IPs

A fase de carregamento dos ficheiros de *log,* começa por extrair todos os ficheiros que se encontram na directoria na qual o servidor coloca os ficheiros de *logs*. Cada um dos ficheiros é depois processado individualmente. Primeiro, começamos por fazer um *parser* ao ficheiro, uma vez que os *logs* se encontram no formato ECLF, utilizamos o espaço para separar cada um dos campos. Depois, os dados do ficheiro são carregados para uma tabela da área de retenção, tabela *logs,* onde vão ficar guardados até o processo de ETI estar completo.

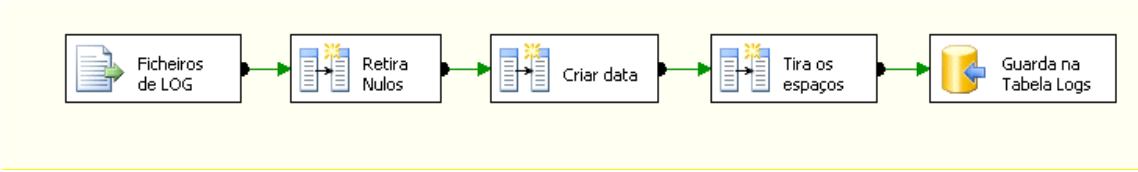


Figura 5.11: Recolha e carregamento dos ficheiros de *Log*

O carregamento das dimensões é feito em duas fases distintas. As dimensões tempo e data apenas são carregadas uma vez antes de se carregar o primeiro ficheiro de *logs*. Nessa altura é determinado um período de vida para o *Dweb*, sendo gerados então todos os registos da dimensão até essa data. Por sua vez, na dimensão tempo geram-se os registos com todos as horas do dia. As outras restantes quatro dimensões são carregadas à medida que os ficheiros de *log* vão sendo processados. Por precaução, na área de retenção são mantidas cópias destas dimensões para o caso de haver algum problema durante o carregamento.

Depois de carregados todos os dados do ficheiro de *logs* para a base de dados é feito o processamento das dimensões. Cada uma das dimensões vai à tabela de *logs* buscar os valores do atributo necessário para as alimentar. No povoamento da dimensão *request* vamos buscar todos os valores do atributo *request* há tabela de *logs*, fazendo-se a eliminação de todos os valores repetidos. Os valores que ainda não existem na dimensão vão, então, ser preparados para a integração na dimensão, sendo para isso atribuído uma chave de substituição a cada pedido determinando-se qual extensão do pedido. No fim da realização destes passos, os registos estão prontos para serem integrados na dimensão *request* da AR e do DW.

O povoamento da dimensão *agent* é idêntico à dimensão *request*, com uma pequena diferença depois de identificar todos os novos *user agent* antes de integrar na dimensão é preciso determinar se este é um agente automático ou um utilizador regular. Na dimensão

computadorUtilizador, não é necessário gerar uma chave de substituição uma vez que o próprio *IP* é usado para isso, depois de identificar todos os novos *IPs* vamos atribuir um país a cada *IP*. Depois de atribuído país é feita a integração dos *IPs* na dimensão *computadorUtilizador*. Por último a dimensão *referrer* povoa-se de maneira idêntica á *request*, sendo que depois de identificar os novos *referrer* é preciso verificar se o *referrer* é motor de pesquisa antes de ser integrado na respectiva dimensão.

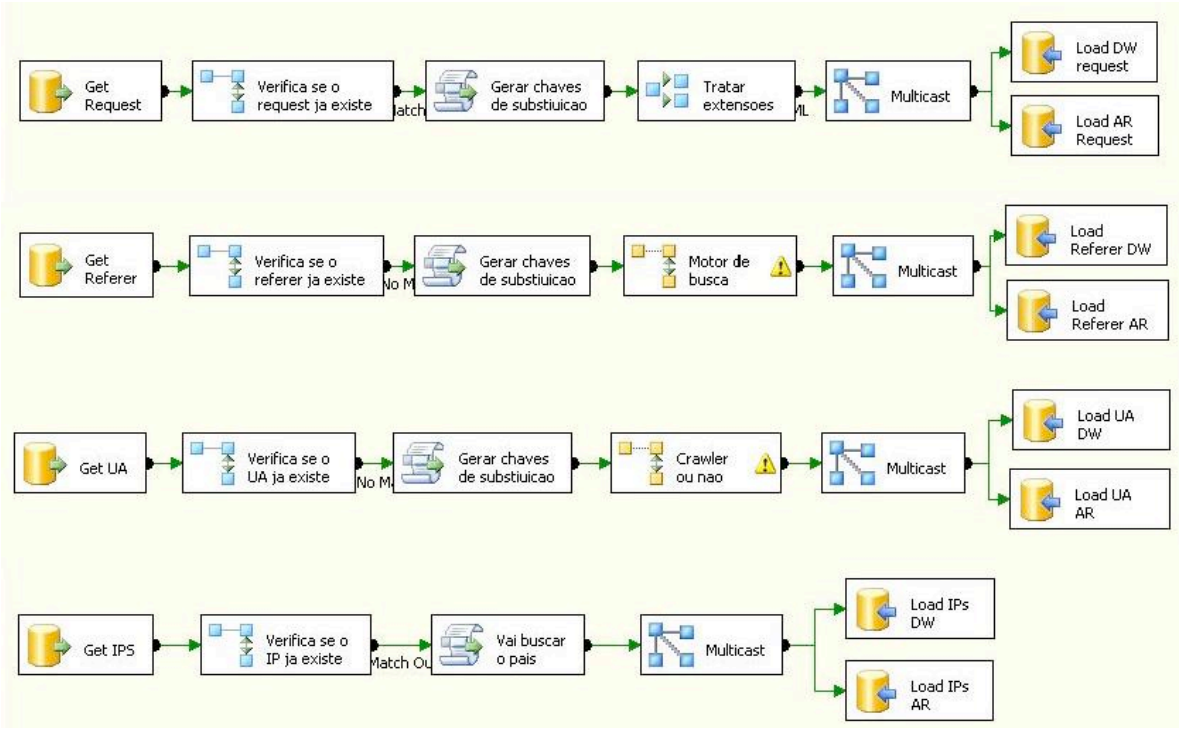


Figura 5.12: Povoamento das várias dimensões

Por fim, depois de processadas todas as dimensões, faltam apenas carregar os dados para a tabela de factos, Figura 5.13. Antes do preenchimento da tabela de factos é necessário fazer a identificação de sessões, onde a cada pedido da tabela de *logs* vai ser atribuído um identificador. Depois de atribuído o identificador podemos passar os dados para a tabela de factos e para a tabela ponte. Os dados não são passados directamente para o *Dweb,* como forma de precaução, primeiro são colocados numa tabela com a mesma estrutura na AR e posteriormente são passados para o *DWeb*. Nesta primeira fase o preenchimento dos dados na tabela de factos e tabela ponte é feito em simultâneo. Depois de preenchidas as tabelas na AR vão ser copiadas para o *DWeb*,

primeiro são passados os dados da tabela de factos, e apenas quando esta passagem estiver concluída são passados os dados da tabela ponte.

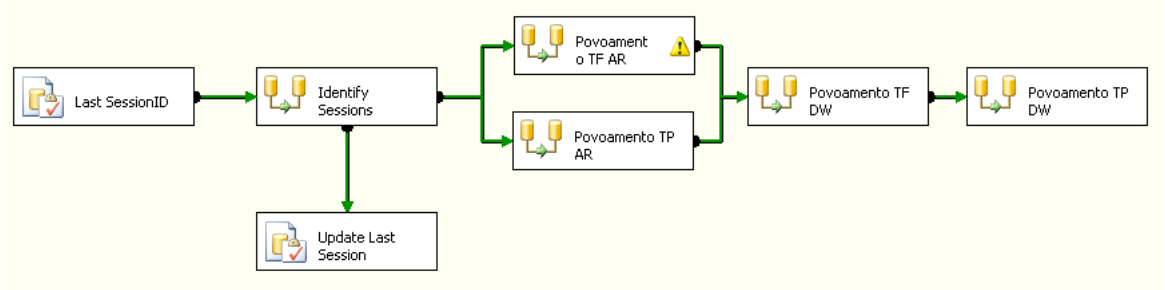


Figura 5.13: Povoamento da tabela de factos e da tabela ponte.

Depois de todos os dados integrados falta apenas realizar um último passo: a limpeza das tabelas auxiliares da área de retenção.

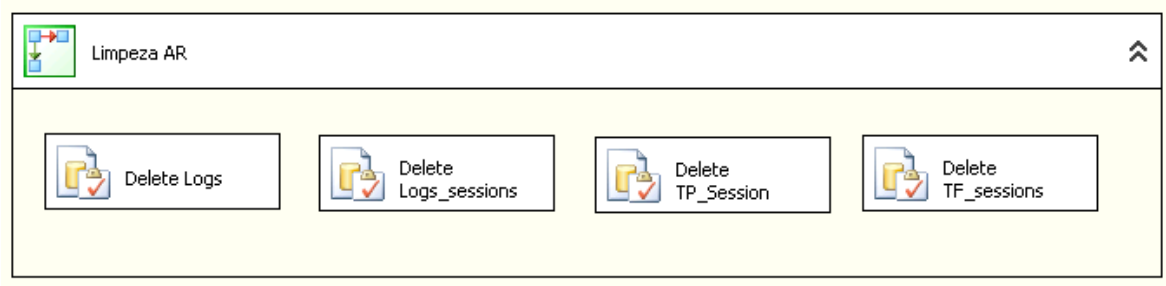


Figura 5.14: Limpeza de tabelas área de retenção.

5.5 Aplicação de Técnicas de Mineração de Dados

No capítulo 4 foram apresentadas várias técnicas de mineração de dados usadas em mineração de dados de utilização Web. Contudo, na avaliação experimental vamos utilizar apenas três das técnicas apresentadas: *clustering* [Berkhin 02], regras de associação[Agrawal and Srikant 94] e padrões sequenciais [Liu, 08], mais propriamente cadeias de *markov*.

A aplicação das técnicas de mineração de dados seguiu a metodologia de CRISP-DM *(CRoss-Industry Standart Process for Data Mining)* [WWW 15]. Esta metodologia fornece uma visão muito

concreta e prática sobre o ciclo de vida de um projecto de mineração de dados. Na metodologia podemos ver todas as fases de desenvolvimento de um projecto de mineração de dados e a forma como cada uma delas se encontra relacionada entre si. Num projecto de mineração de dados existem relacionamentos entre todas as suas fases de acordo com os objectivos que se pretendem atingir. Na Figura 5.15, retirada de [WWW 15] podemos observar o ciclo de vida de um projecto de mineração de dados segundo a metodologia de *CRISP-DM*.

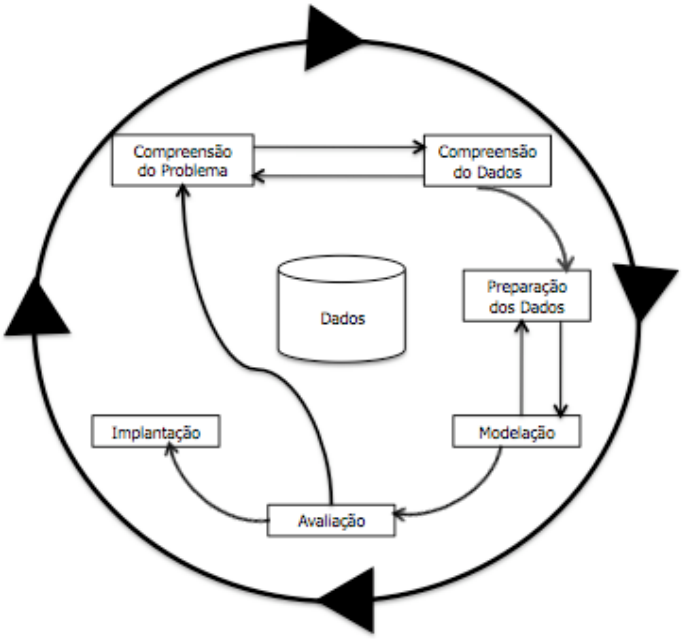


Figura 5.15: Várias fases da metodologia *CRISP-DM* [WWW 15]

O ciclo de vida de um projecto de da mineração de dados está dividido em seis fases: compreensão do problema, compreensão dos dados, preparação dos dados, modelação, avaliação e implementação. A ordem das fases não é restrita, andar para a frente e para trás em cada uma das fases é bastante comum. Isto depende dos resultados obtidos na passagem de uma fase para a outra, ou se existir uma etapa na fase seguinte que necessite de alguns resultado especial. As setas da imagem indicam as dependências mais frequentes entre as diversas fases.

De seguida é apresentada uma pequena descrição de cada uma das fases:

- **Compreensão do Problema :** Esta é a primeira etapa num projecto de mineração de dados. Foca-se na compreensão dos objectivos e dos requisitos do projecto, e na posterior conversão do problema num problema de mineração de dados. Nesta fase, é traçado o plano inicial com que se pretende atingir os objectivos que levaram a criação do projecto.
- **Compreensão dos dados:** Nesta etapa é feita uma avaliação inicial ao conjunto de dados sobre o qual vão ser aplicados os algoritmo de mineração de dados. De forma a conhecer melhor os dados são aplicados diversos algoritmos, que têm diferentes objectivos tais como: identificar problemas na qualidade dos dados, descobrir os primeiros padrões, detectar subconjuntos de dados que fornecem informações importantes entre outros.
- **Preparação dos dados:** A preparação dos dados é uma das etapas de maior importância num projecto de mineração de dados. Nesta fase são feitas todas as transformações necessárias sobre os dados de modo obter o *dataset* final, o conjunto de treino sobre o qual vão ser aplicados os algoritmos de mineração. Dentre as várias tarefas que fazem parte da fase de preparação de dados incluem-se: selecção de registos, selecção de atributos, transformação e limpeza dos dados, tratamento de nulos, entre outros.
- **Modelação:** Nesta fase vão ser aplicadas diversas técnicas de mineração de dados de modo a obter um modelo. Tipicamente, podem ser aplicadas várias técnicas ao mesmo problema de mineração de dados. Algumas dessas técnicas têm requisitos específicos para o tipo de dados, por este motivo é muitas vezes necessário voltar à fase de preparação de dados.
- **Avaliação:** Nesta fase já temos um ou vários modelos construídos que aparentam ter um grande qualidade sobre o ponto de vista dos negócios. Antes de passar para a fase final do projecto é necessário avaliar o modelo e rever todos os passos seguidos até a obtenção deste, de maneira a verificar se foram cumpridos todos os objectivos definidos. No fim desta fase deve haver uma decisão sobre usar ou não os modelos gerados.
- **Implementação**: A criação do modelo não é a ultima tarefa num projecto de mineração de dados. Mesmo quando o objectivo do projecto é aumentar o conhecimento sobre os dados, é necessário organizar o conhecimento obtido de modo a poder ser usado pelos clientes. Dependo dos requisitos a fase de implementação pode ser tão simples como a geração de um relatório ou tão complexo como um processo de mineração de dados que se repete ciclicamente. Em muitos casos é o cliente e não o analista dos dados que se vai preocupar com esta fase. Contudo, mesmo que o analista não trate da parte de

implementação é importante que o cliente perceba que tipo de tarefas tem de fazer para assim poder fazer uso do modelo gerado.

Actualmente existem no mercado várias ferramentas para aplicação de técnicas de mineração de dados. Na avaliação prática deste projecto optamos por utilizar uma ferramenta *open source*, uma vez que este é um trabalho com fins académicos. Dentre as várias ferramentas *open source*, optamos por utilizar o *RapidMiner* versão 4.5[WWW 16]. Os principais motivos que levaram a escolha desta ferramenta foram os seguintes:

- A ferramenta disponibiliza uma versão *open source*.
- Possui uma grande número de algoritmos implementados para as diferentes técnicas(Regras de Associação, Clusters, Regressão e Classificação).
- Possui directivas gráficas que possibilitam uma melhor visualização dos resultados
- Pode ser instalado em vários sistemas operativos uma vez que está escrito em java
- Interface gráfica é bastante intuitiva, possibilitando uma rápida aprendizagem e uma rápida evolução
- Disponibiliza vários exemplos práticos o que facilita a aprendizagem
- A cotação da ferramenta na comunidade cientifica.

O *Data Webhouse* onde os dados se encontram guardados, esta a correr sobre um máquina com 1 GB de *RAM*, e um CPU, *Pentiun* 4 CPU 3.0 Ghz, com o sistema operativo *Windows XP.* O sistema de gestão de base de dados utilizado para guardar o *Dweb* foi o Microsoft SQL Server 2008. No entanto o sistema utilizado para a aplicação de técnicas de mineração de dados foi diferente, os processos correram sobre um máquina com Windows XP *Service Pack* 3, 3 GB de *RAM*, e um CPU , Intel Pentium Dual CPU T3200 2.0GHz

5.5.1 Conjunto de Treino

Os dados utilizados em qualquer uma das técnicas de mineração de dados estudadas, encontram-se guardados num *webhouse* construído com os ficheiros de *log* gerados pelo servidor do *RepositoriUM*. A maior parte do processamento de dados já foi feito durante o processo de alimentação do *webhouse* pelo que não será necessário efectuar muitas operações de transformações de dados.

No *data webhouse* encontram-se guardadas todas as sessões efectuadas desde o dia 01-01-08 até ao dia 31-06-08. Durante este período foram identificadas 1433771 sessões, dessas 330531(23%) foram atribuídas a agentes automáticos, sendo as restantes atribuídas a utilizadores regulares. No total das sessões foram efectuados 21948090 pedidos, sendo a média de pedidos por sessão 15. Foram também identificados 54437 pedidos distintos desses apenas cerca de metade 26228 são páginas *webs*, os restantes são imagens, *pdfs,* entre outros. As sessões foram efectuadas de 208 países diferentes. Durante estes seis meses foram identificados 3334 referentes dos quais 1032 são motores de pesquisa e 60116 *user agents* dos quais 536 foram identificados como agentes automáticos.

Devido ao volume de dados armazenado no *webhouse* ser muito grande optou-se por utilizar um conjunto de dados mais reduzido, de forma a não existirem problemas de processamento durante a fase de aplicação dos algoritmos. Optou-se assim por reduzir o conjunto de dados a um período de dois meses, os meses escolhidos foram Março e Abril, uma vez que são meses de aulas normais. Desde o dia 01-03-08 até ao dia 30-04-08 foram identificadas 579567 sessões das quais 158694 são atribuídas a *crawlers* e 420873 são atribuídas utilizadores normais, a média de pedidos por sessão neste período foi de 14. Durante estes dois meses foram também identificados 42375 pedidos distintos dos quais 20109 são identificados como páginas

Tendo identificado o período de tempo e consequentemente as sessões que vão ser utilizadas, passamos a fase seguinte que consiste em aplicar alguns algoritmos de pré-processamento aos dados. Dado que o *data webhouse* guarda tanto sessões pertencentes a agentes automáticos como a utilizadores regulares, é necessário seleccionar apenas as sessões que foram atribuídas a utilizadores normais. Uma vez que a utilização de sessões atribuídas a agentes automáticos pode baralhar a identificação de perfis de utilização Web.

Para a aplicação das técnicas de cluster e de regras de associação é necessário construir a matriz de transacções sobre a qual vão ser aplicados os algoritmos. A matriz de transacções é como vimos na secção 4.1 uma matriz de sessões por páginas. Uma vez que o número de páginas identificadas é muito alto a matriz fica com muitas colunas, isto torna os algoritmos de mineração de dados muito lentos. Por este motivo optou-se por retirar todas as páginas que não aparecem em mais de 0.5% das sessões, dado que estas páginas não afectam a identificação de perfis visto

que não aparecem num número significativo de sessões. Retiraram-se também todas as sessões que tinham apenas um pedido pois como o objectivo da identificação de perfis é definir um conjunto de páginas que sejam acedidas por um grupo de utilizadores, as sessões com apenas uma página não vão acrescentar informação significativa aos perfis. Depois de eliminadas as páginas e as sessões ficamos com um total de 476 páginas e 27637 sessões. O resultado da matriz de transacções é guardado num ficheiro que depois vai ser passado como input aos algoritmos no *rapidminer.*

5.5.2 Algoritmos de Clustering

Como referido anteriormente a ferramenta de mineração de dados escolhida para a aplicação de algoritmos de mineração de dados foi o *RapidMiner*[WWW 16]. O *RapidMiner* tem implementados vários algoritmos de cluster, alguns desses algoritmos são particionais outros hierárquicos, devido ao volume e complexidade dos dados optamos por escolher um algoritmo de particionamento para a aplicação de clusters.

Dentro os vários algoritmos disponíveis o algoritmo escolhido foi o *k-means.* O *k-means* particiona o conjunto de dados em k *clusters* distintos, sendo k o número de clusters escolhidos pelo utilizador. O *k-means* funciona de uma maneira muito simples, começa por escolher *k* pontos que vão ser o centro de cada um dos *cluster.* De seguida calcula a distancia de cada um dos pontos restantes ao centro de cada um dos *clusters.* Sendo, cada um dos pontos restantes atribuído ao *cluster* cujo centro se encontra mais próximo. Seguidamente depois de todos os pontos estarem colocados num *clusters,* o centro do *cluster* é recalculado, utilizando todos os pontos do cluster. Repetindo-se de seguida outra vez todo o processo com os novos centros, até que a condição de paragem seja satisfeita. Normalmente a condição de paragem pode ser: até que não existam pontos colocados num clusters diferente do da iteração anterior, ou até não haver alteração do centro. Esta seria a condição de paragem ideal, mas uma vez que é muito complicada de conseguir, muitos algoritmos usam também como condição de paragem, um número máximo de iterações definido pelo utilizador.

Algumas das principais vantagens do *k-means* são as seguintes: é simples, fácil de perceber e implementar é eficiente a sua complexidade está dependente do número de instâncias, do número de clusters e do número de iterações. Dado que na maioria das vezes o número de clusters e o

número de transacções são baixos a eficiência do *k-means* está depende sobretudo do número de instancias. O algoritmo apresenta também algumas desvantagens dentre elas apenas pode ser utilizado quando o meio de um clusters está definido é necessário definir o número de clusters, é bastante sensível a *outliers*, pontos que se encontram muito longe do centro do clusters, é bastante sensível a uma má escolha dos pontos escolhidos inicialmente para gerar os clusters. Apesar de todas estas desvantagens o *k-means* é um dos algoritmos de clusters mais utilizados em especial devido a sua simplicidade e eficiência. Para além disso não podemos comparar os resultados obtidos pelo *k-means* com resultados de outros algoritmos, uma vez que não conhecemos os clusters que pretendemos obter.

Para a geração dos clusters passamos como input ao *rapidminer* o ficheiro onde está guardada a matriz de transacções. Depois a esse ficheiro vamos aplicar o algoritmo *k-means*, é preciso definir alguns parâmetros antes de por o algoritmo a correr. O parâmetro mais importante a ser definido é o número de clusters utilizado pelo algoritmo, neste caso o número de cluster escolhido foi quatro um vez que foi este o número de grupos utilizadores definidos a partida. Como *output* o *rapidminer* fornece-nos um gráfico com a percentagem em que cada página aparece em cada um dos clusters.

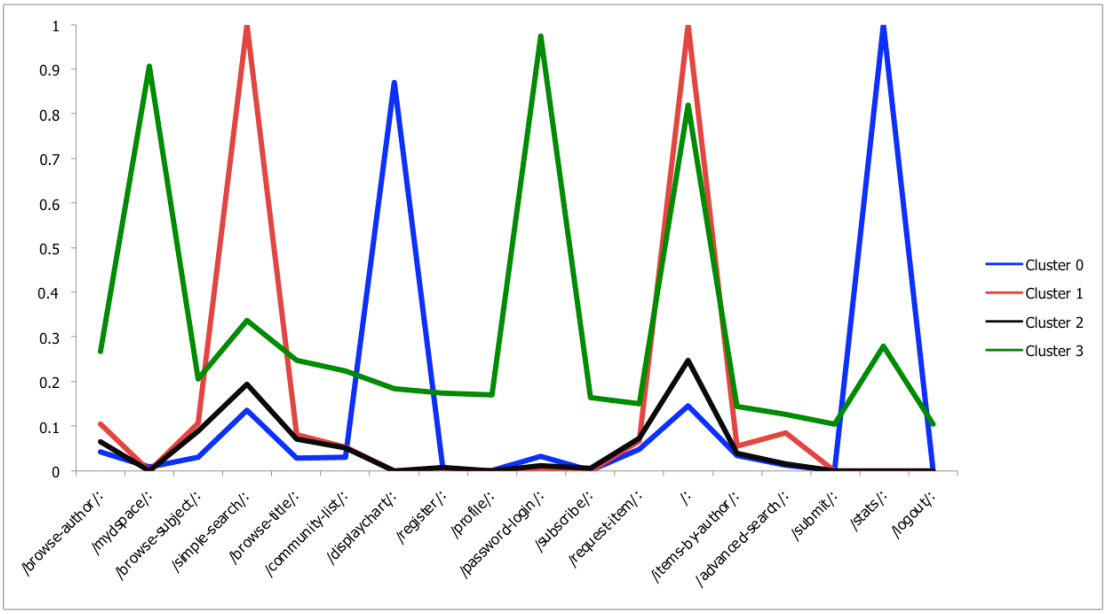


Figura 5.16:Percentagem de pedidos por página em cada cluster.

Para além do gráfico o *rapidminer* fornece uma lista com o cálculo do centro de cada cluster, que neste caso representa a percentagem que cada página apresenta em cada um dos clusters. De seguida apresentamos uma tabela com as páginas mais vistas em cada cluster, mais propriamente com as páginas vistas em mais de 5% das sessões de cada cluster.

ID	**Cluster 0 – 2499 Sessões** **Página**	**Peso**
1	/stats/:	1
2	/displaychart/:	0.871
3	/handle/1822/7484/:	0.185
4	/:	0.146
5	/simple-search/:	0.136
6	/sdum/stats/:	0.0.053

Tabela 5.9: Perfil de utilização associado ao cluster 0

Como podemos observar, o número de sessões atribuídas ao *cluster 0* representa apenas 10% das sessões do *site*. Podemos ver que em todas as sessões os utilizadores consultaram a página /*stats*/, página que fornece algumas estatísticas sobre os acessos ao *site*. Devido estes dois factores, podemos dizer que este perfil diz respeito aos funcionários/administradores do *site*, uma vez que serão eles quem maioritariamente vai consultar este tipo de páginas.

ID	**Cluster 1 – 5885 Sessões** **Página**	**Peso**
1	/simple-search/:	1.000
2	/:	1.000
3	/browse-subject/:	0.107
4	/browse-author/:	0.105
5	/advanced-search/:	0.086
6	/browse-title/:	0.081
7	/request-item/:	0.068
8	/items-by-author	0.056
9	/community-list/:	0.054

Tabela 5.10: Perfil de utilização associado ao cluster 1

O *cluster 1*, representa cerca de 25% dos acessos ao *site*. Todos os utilizadores colocados neste *cluster* consultam a página inicial do *site* e a pesquisa simples. O facto de entrarem pela página inicial e as páginas mais visualizadas estarem todas relacionadas com pesquisas, podemos dizer que os utilizadores deste perfil já tinham conhecimento prévio do *site*. Por estes motivos, podemos dizer que este perfil representa os alunos ou investigadores da UM, uma vez que estes têm conhecimento do *site* e utilizam-no para efectuar pesquisas.

ID	Cluster 2 – 17143 Sessões Página	Peso
1	/:	0.248
2	/simple-search/:	0.194
3	/browse-subject/:	0.089
4	/request-items/:	0.074
5	/browse-title/:	0.071
6	/browse-author/:	0.065
7	/handle/1822/3/:	0.059
8	/items-by-subject/:	0.058
9	/handle/1822/2/:	0.057
10	/community-list/:	0.051

Tabela 5.11: Perfil de utilização associado ao cluster 2

O *cluster 2* representa a maioria das sessões do *site*, mais própria mente representa cerca de 60% das sessões. Este perfil é complicado de caracterizar, uma vez que não existe uma página que se destaque das restantes. O facto de todas as páginas terem uma percentagem baixa, leva a crer que a maioria dos utilizadores não utiliza a página inicial para entrar no *site*. Muito provavelmente os utilizadores entram no *site* redireccionados pelos diversos motores de pesquisa existentes. E alguns desses utilizadores aproveitam depois para utilizar o *site* para fazer pesquisas. É por este motivo a página da / e o /simple-search/ apresentam as percentagens mais altas neste perfil. Este perfil é então atribuído a outros, que são os utilizadores que não tinham conhecimento do *site*, chegaram a ele através de motores de pesquisa, e alguns desses utilizadores depois de entrarem no *site* utilizaram-no para fazer pesquisas.

ID	Cluster 3 – 2011 Sessões Página	Peso
1	/password-login/:	0.975
2	/mydspace/:	0.907
3	/:	0.820
4	/simple-search/:	0.339
5	/stats/:	0.281
6	/browse-author/:	0.268
7	/browse-title/:	0.249
8	/community-list/:	0.225
9	/browse-title/:	0.207
10	/displaychart/:	0.185
11	/register/:	0.176
12	/profile/:	0.171
13	/subscribe/:	0.165
14	/request-item/:	0.152
15	/items-by-author/:	0.146
16	/advanced-search/:	0.127
17	/submit/:	0.108
18	/logout/:	0.105

Tabela 5.12: : Perfil de utilização associado ao cluster 3

Por ultimo o *cluster 3,* assim como o *cluster 0,* representa apenas 10 % dos utilizadores do *site*. O tipo de utilizadores deste *cluster* é fácil de identificar, são os utilizadores registados. Observando as páginas mais acedidas no *cluster* podemos ver que praticamente todos os utilizadores deste perfil se autenticam no *site*. Este perfil representa os professores e investigadores que estão autorizados a publicar artigos no *site*, ou que por algum outro motivo possuam uma conta no repositório.

5.5.3 Algoritmos de Regras de Associação

Tal como na aplicação de cluster a ferramenta utilizada para gerar regras de associação foi o *rapidminer* [WWW 16]. O *rapidminer* disponibiliza apenas uma implementação de um algortimode regras de associação Contudo para além desta implementação disponibiliza outras, que não foram implementados no *rapidminer* mas sim em outra ferramenta de mineração de dados o *weka* [WWW 17]. O algoritmo escolhido para a aplicação das regras de associação foi a implementação feita pelo *rapidminer.*

O algoritmo encontra-se dividido em duas partes distintas, a primeira parte, pode ser usada sem a segunda, consiste na geração de grupos de *itemsets* frequentes, ou seja grupos de *items* que tenham um valor de suporte maior do que o mínimo determinado. A segunda parte não funciona sem a primeira, e consiste em calcular as regras de associação com confiança superior ao valor mínimo fornecido Para gerar as regras de associação é necessário definir primeiro um suporte e uma confiança mínimas.

O suporte mínimo escolhido foi de 5%, este valor foi escolhido com recurso aos resultados obtidos pelos clusters, basicamente a escolha deste valor foi feita com o objectivo de conseguir gerar regras que englobem os quatro clusters. Assim calculamos qual a percentagem de sessões do cluster mais pequeno, deu aproximadamente 10%, e definimos um valor de suporte mais pequeno que essa percentagem neste caso o escolhido foi 5%. De seguida apresentamos uma tabela com todos os *itemsets* com suporte maior que 5%.

ID	Nº Itens	Suporte	Grupos
1	1	0.443	/
2	1	0.374	/simple-search
3	1	0.112	/stats
4	1	0.096	/browse-subject/
5	1	0.092	/displaychart/
6	1	0.087	/browse-author/
7	1	0.083	/password-login/
8	1	0.082	/browse-title/
9	1	0.076	/request-item/
10	1	0.068	/mydspace/
11	1	0.062	/community-list/
12	1	0.053	/items-by-subject/
13	1	0.051	/items-by-author/
14	2	0.244	/, /simple-search/
15	2	0.054	/, /browse-subject/
16	2	0.061	/, /browse-author/
17	2	0.059	/, /password-login/
18	2	0.052	/, /browse-title/
19	2	0.054	/, /mydspace/
20	2	0.092	/stats/ , /displaychart/
21	2	0.065	/password-login/ , /mydspace/
22	3	0.052	/, /password-login/ , /mydspace/

Tabela 5.13: *Itemsets* mais frequentes com suporte mínimo de 5%

A confiança mínima escolhida foi de 50%. Este valor não é muito alto, pois havendo tantas páginas no *site*, os pedidos feitos pelos utilizadores podem ser muito diversos, não havendo por isso regras que apresentem valores para a confiança muita altos. Contudo o valor também não pode ser muito baixo, uma vez que é necessário que as regras ocorram num número significativo de sessões, de forma a poder estabelecer perfis com algum grau de confiança.

ID	Regras de Associação	Suporte	Confiança
1	{/, /mydspace/} ->{/password-login/}	0.052	0.957
2	{/displaychart/}->{/stats/}	0.092	0.999
3	{/mydspace/}->{/password-login/}	0.065	0.953
4	{/, /password-login}->{/mydspace/}	0.052	0.873
5	{/stats/}->{/displaychart/}	0.092	0.821
6	{/password-login/}->{/mydspace/}	0.065	0.785
7	{/mydspace/}->{/}	0.054	0.796
8	{/mydspace/}->{/, /password-login/}	0.052	0.762
9	{/password-login/}->{/}	0.059	0.720
10	{/browse-author/}->{/}	0.061	0.709
11	{/browse-title}->{/}	0.052	0.629
12	{/password-login/}->{/, /mydspace/}	0.052	0.628
13	{/simple-search/}->{/}	0.244	0.652
14	{/password-login/, /mydspace/ }->{/}	0.052	0.800
15	{/browse-subject/}->{/}	0.054	0.563
16	{/}->{/simple-search/}	0.244	0.550

Tabela 5.14: Resultado das regras de associação

Observando o conjunto de regras gerado podemos verificar que as regras com a confiança mais alta são regras relacionados com o perfil dos utilizadores autenticados e com o perfil de administradores. As regras um, três, quatro, seis, oito, nove, doze e treze, são regras que se encaixam no perfil de utilizadores autenticados, cluster 3. Todas estas regras são referentes a três páginas /*mydspace/*, /*password-login/* e /. Podemos então dizer que estas três páginas têm uma grande probabilidade de serem acedidas numa mesma sessão. Por exemplo podemos ver que a probabilidade de um utilizador que visite as páginas /*mydspace*/ e / e depois visitar a página /*password-login/* é de quase 95%. As regras dois e cinco estão relacionados com o perfil dos administradores/funcionários do site Web, uma vez que são duas páginas que fornecem estatísticas. Por fim as regras dez, onze, treze, quinze e dezasseis, são regras que podem pertencer a qualquer um dos perfis, mas traduzem uma informação importante: mais de 50% das

pessoas que acedem ao *site* para pesquisar sobre um dado assunto, quase sempre visitam a página inicial. A regra dezasseis é uma regra que pode definir muitas vezes os alunos, já que indica os utilizadores que utilizam a barra e a pesquisa simples.

5.5.4 Cadeias de Markov

Por último vamos tentar identificar perfis de utilização através da modelação de sessões com recurso a cadeias de *Markov*. A implementação das cadeias de *Markov* foi feita em Java [WWW 18]. Ao contrário das outras duas técnicas nas cadeias de *markov,* interessa saber a ordem em que cada pedido ocorreu na sessão e, por este motivo, não se pode cortar as páginas que aparecem em menos de 0.5% das sessões. Isto porque, ao cortar-se essas páginas perde-se o caminho feito pelo utilizador, ficando-se assim com sessões descontínuas.

A implementação das cadeias de *Markov* inicia-se pela reconstrução das sessões, que depois vão ser usadas para construir o grafo que vai representar a cadeia. Para isso, começa-se por ir buscar todas as páginas para as quais foram feitos pedidos e de seguida acede-se à base de dados para obter todos os pedidos, ordenados por sessão e número sequencial. Posteriormente, tendo todas as sessões reconstruídas e todas as páginas guardadas, calcula-se a probabilidade inicial de cada página. A probabilidade inicial de uma página *P* é calculada do seguinte modo: calcula-se o número de pedidos de *P* e divide-se pelo número total de pedidos. Depois de calculadas todas as probabilidades iniciais calcula-se a probabilidade de uma página aparecer a seguir a outra. Para isso é necessário percorrer todas as páginas e, para cada uma delas, buscar todas as sessões nas quais cada uma delas aparece. Tendo todas as sessões em que uma página *P* aparece, é necessário consultar qual a página que aparece a seguir a *P* em todas as sessões. Caso página P*1* apareça a seguir a *P,* vamos verificar se já existe algum contador para a página *P1*. Caso exista, incrementamos o valor, caso contrário inicializamos o contador com o nome da página e o valor 1. No fim temos um *array* com todos as páginas que aparecem de seguida e o número de vezes que aparecem. O passo seguinte é fazer o cálculo do número de vezes que as páginas aparecem depois de uma página *P*. A probabilidade da página seguinte *P1* é calculada da seguinte forma: o número de vezes que a página P1 aparece a seguir a P, sobre o número de vezes que *P* aparece.

Depois de calculadas todas as probabilidades o grafo vai ficar guardado numa tabela específica da base de dados. De seguida, apresentamos uma parte da cadeia de *Markov* gerada pela ferramenta

GraphViz [WWW 19] - não é possível visualizar a cadeia na sua totalidade devido ao elevado número de nós e arcos existentes.

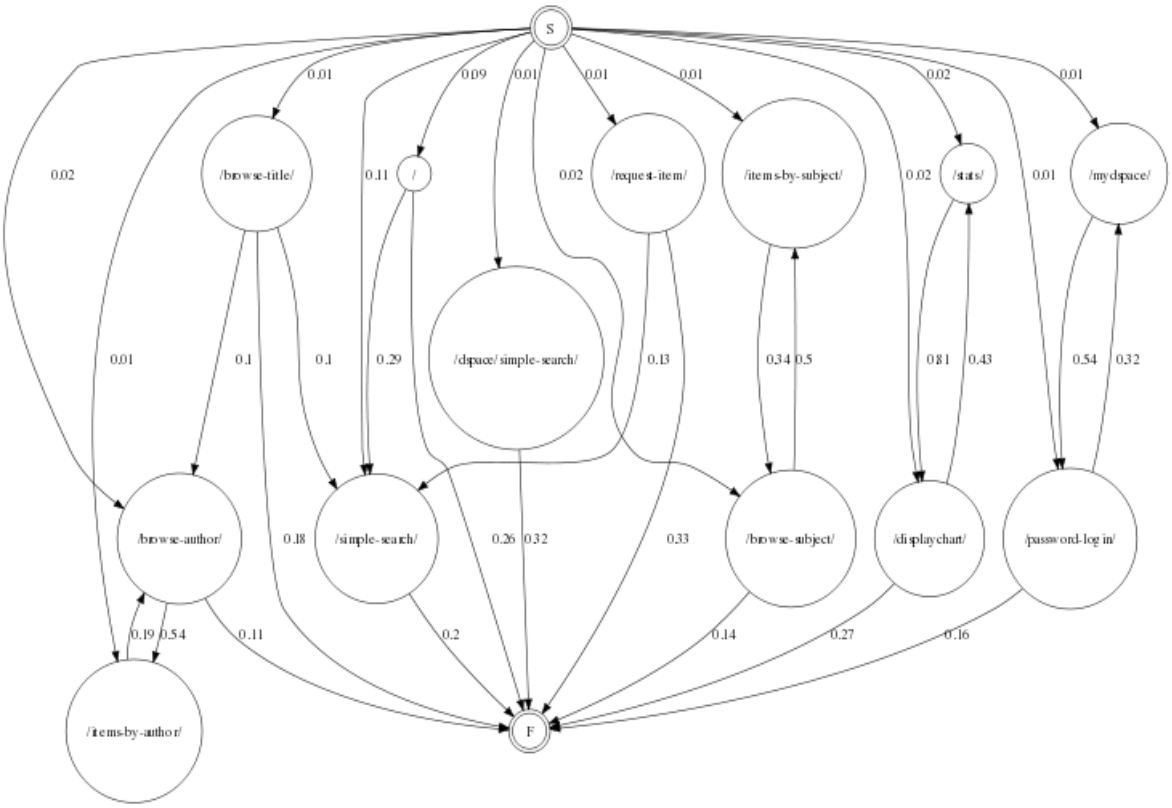


Figura 5.17: Representação de uma cadeia de *Markov*

Depois de gerada a cadeia de *markov,* com os dados referentes aos acessos ao *site* durante o mês de Março e Abril, foi possível calcular quais os caminhos mais frequentes, ou seja os caminhos mais percorridos pelos utilizadores. Para fazer esse cálculo é preciso definir, em primeiro lugar, o suporte e a confiança mínimos. Para isso calculou-se a média de todas as probabilidades iniciais e usou-se esse valor como suporte mínimo. Para a confiança escolhemos um valor de 10%, que foi o valor escolhido já que existe um grande número de páginas Web que o utilizador pode escolher para visualizar. Assim, como a escolha é muito diversificada, se se tivesse sido escolhido um valor muito alto para a confiança, muitas das cadeias não iriam ser capturadas.

A geração dos caminhos mais frequentes é feita do seguinte modo: primeiro, começamos por ir buscar todas as páginas que têm probabilidade maior do que o suporte definido; depois, utilizamos o algoritmo de pesquisa sobre grafos *depth-first* [Tarjan 62] para calcular os caminhos cuja

confiança é mais alto que o valor dado; por fim, apresentamos uma tabela com caminhos mais frequentes gerados, com um suporte de 1% e uma confiança de 10%.

ID	Caminho	Probabilidade
1	/stats/ -> /displaychart/	0.81
2	/mydspace/ -> /password-login/	0.54
3	/browse-author/ -> /items-by-author/	0.54
4	/browse-subject/ -> /items-by-subject/	0.5
5	/displaychart/ -> /stats/	0.43
6	/items-by-subject/ -> /browse-subject/	0.34
7	/password-login/ -> /mydspace/	0.32
8	/ -> /simple-search/	0.29
9	/items-by-author/ -> /browse-author/	0.19
10	/request-item/ -> /simple-search/	0.13
11	/browse-title/ -> /browse-author/	0.1
12	/browse-title/ -> /simple-search/	0.1

Tabela 5.15: Caminhos mais frequentes probabilidade 10% e suporte 1%.

Os caminhos mais frequentes obtidos vão de encontro aos resultados alcançados, quer nos *clusters* quer nas regras de associação. Observando os caminhos gerados, pode-se ver claramente que as regras um e cinco se enquadram perfeitamente no cluster 0, atribuído a funcionários e administradores do *site*. Também podemos observar que as regras sete e dois estão relacionadas com o perfil utilizadores autenticados, o cluster 3. As regras três, quatro, seis e nove podem fazer parte de qualquer um dos perfis, mas encontram-se relacionadas sobretudo com os perfis, alunos e outros.

Todas estas regras dizem respeito à navegação do *site*. Por exemplo, pela regra três pode-se ver que, depois de visualizar todos os autores, existe 54% de probabilidade do utilizador ver os artigos publicados por um dado autor. Esta regra é complementar à regra nove em que um utilizador estando a visualizar todos os artigos de um autor tem 19% de probabilidade de visualizar a lista de todos os autores. As regras quatro e seis são semelhantes, mudando-se apenas o tema de pesquisa, que aqui em vez de ser autor é assunto. A regra cinco é uma regra que pode estar associada ao perfil estudantes, uma vez que indica que as pessoas que estão na página principal do *site* tem uma grande probabilidade de seguida utilizar a pesquisa simples. Por último as duas ultimas regras, onze e doze, que apresentam a probabilidade mais baixa, também não podem ser

atribuídas a um perfil em particular. A regra onze pode ser interessante, uma vez que indica que 10% das pessoas que procuram por título de seguida pesquisam por autor. Por fim, a última regra indica que 10% dos utilizadores prefere primeiro consultar a lista de títulos disponíveis e só depois fazer uma pesquisa simples no repositório.

5.6 Considerações Finais Acerca dos Resultados Obtidos

Na avaliação experimental foram aplicadas três técnicas diferentes de mineração de dados de modo a conseguir identificar perfis de utilização Web. Os resultados obtidos com cada uma das técnicas foram convergentes. Isto significa que, independentemente da técnica utilizada, os resultados obtidos foram idênticos. A técnica de *clustering utilizada* foi aquela que deu os resultados mais fáceis de ser interpretados, uma vez que o próprio *output* do algoritmo utilizado já apresentava quais as páginas mais acedidas em cada um dos *clusters.* Para as outras duas técnicas a identificação de perfis foi feita de forma idêntica: associamos cada uma das regras com os perfis predefinidos. Para os perfis de utilizadores autenticados e administradores e funcionários, não foi muito complicado associar as regras. Contudo, para os outros dois tipos foi um pouco mais complicado, uma vez que não existe uma página, ou conjunto de páginas que caracterize inequivocamente cada um destes perfis.

O sistema implementado apresenta alguns problemas em termos de *performance* devido ao tipo de máquina utilizada e às suas características pouco adequadas. Assim, não foi possível utilizar todo o conjunto de dados disponível devido ao facto de o volume de dados ser muito grande e, quando passado ao *rapidminer*, não ser possível produzir resultados devido a problemas de memória. Também nos deparámos com alguns problemas na utilização de alguns dos algoritmos, devido, essencialmente, ao número de atributos envolvidos. Por este motivo, não foi possível fazer uma comparação entre os diferentes algoritmos de *clustering* e de regras de associação. Apenas conseguimos correr um algoritmo de *clustering,* o *k-means*, que como se sabe é um dos mais simples. Os outros algoritmos experimentados não produziram resultados em tempo útil. O mesmo aconteceu para os algoritmos de regras de associação. As cadeias de *Markov ,* uma vez que foram construídas em Java e guardadas numa base de dados não causaram grandes problemas de memória nem de desempenho - demoraram cerca de quinze minutos a ser geradas.

Devido a diversidade de documentos publicados no repositório, não foi possível estabelecer perfis mais específicos, sendo apenas definidos perfis mais abrangentes. Isto aconteceu porque existem muitas possibilidades de consulta sobre os documentos no repositório, não existindo um número significativo de pessoas que sigam um determinado padrão de comportamento nesse processo. Os documentos consultados pelos utilizadores são muito diversos, o que faz com que o número de pessoas que consulta um dado documento não seja muito significativo, tendo em conta o número de sessões existentes.

Os quatro perfis estabelecidos, *utilizadores registados, administradores/funcionários, alunos e outros*, são perfis bastante abrangentes. Analisando os resultados das técnicas aplicadas podemos caracterizar cada um deles do seguinte modo: o perfil de *administradores* representa cerca de 10% das visitas feitas ao *site* durante os dois meses em estudo, consultando sobretudo páginas relacionadas com estatísticas de acesso ao *site*. Muitos destes utilizadores acabam também por visitar a página principal. O perfil dos *utilizadores registados* refere-se aos professores ou investigadores da universidade que podem publicar artigos ou que têm artigos publicados no repositório. Estes caracterizam-se por se autenticarem no *site* e que consultam sobretudo as páginas referentes à sua conta ou aos artigos publicados. O perfil de *alunos* representa cerca de 25% dos acessos ao *site*, caracterizando-se por englobar sessões em que todos os utilizadores entram pela página principal do *site* e, de seguida, utilizam a pesquisa simples, podendo pelo meio utilizar outros tipos de pesquisa. Este perfil é atribuído a *alunos*. Contudo pode também englobar sessões feitas por professores ou investigadores que tenham utilizado o *site* para efectuar alguma pesquisa sem que, para isso, se tenham autenticado. Por último o perfil *outros*, que diz respeito a todos os utilizadores que entraram no *site*, mas que não foram incorporados em nenhum dos perfis anteriores. Provavelmente, estes registos dizem respeito a utilizadores que entraram no *site* depois de terem feito uma pesquisa num motor de busca, como por exemplo o *Google* [WWW 12] – este depois redireccionou-os para o *site* do repositório. É por este motivo que, neste perfil, ao contrário dos quatro perfis anteriores não temos uma página com uma percentagem muita alta, porque os pedidos estão espalhados pelas diversas páginas do *site*.

Através da caracterização dos perfis de utilização podemos, agora, sugerir algumas melhorias que podem ser implementadas no *site*, de forma a ir de encontro daquilo que os utilizadores procuram. Na página principal pode ser colocada uma hiperligação para uma página onde se encontrem as várias estatísticas existentes sobre o *site*. Deste modo estaremos a facilitar a navegação dos

utilizadores que se encaixem no perfil administradores. Em relação ao perfil utilizadores autenticados não há grandes melhorias a apontar, uma vez que estes já têm um menu onde podem consultar todas as opções sobre a sua conta. Em relação ao perfil alunos também não existem grandes melhorias a fazer uma vez que estes se limitam a entrar no *site* e a utilizar as funcionalidades de pesquisa simples para procurar documentos. Alguns, para além disso, podem também fazer outro tipo de consulta como, por exemplo, por autor, assunto ou escola. Por fim, para o perfil outros é complicado estabelecer que melhorias aplicar, já que estes utilizadores visitam páginas com características muito distintas. Uma melhoria que podia ser feita seria a de colocar uma pequena explicação sobre o repositório nas páginas dos documentos. Isto poderia chamar a atenção de um utilizador que entre no *site*, via um motor de pesquisa, e fazer com que ele começe a utilizar o repositório para efectuar algumas das suas pesquisas.

Uma outra melhoria seria a de beneficiar todos os perfis, estabelecendo um sistema de recomendação. O sistema poderia funcionar do seguinte modo: em cada uma das páginas de documento teríamos uma lista de documentos relacionados com o documento; esta lista poderia ser criada através das cadeias de *markov,* uma vez que estas nos fornecem quais as páginas com probabilidades mais altas de ser consultadas a seguir a uma dada página. Este tipo de sistema de recomendação seria bastante benéfico, uma vez que poderia indicar aos utilizadores possíveis artigos relacionados com o que ele pesquisou e que, previamente, o utilizador não tinha conhecimento.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Síntese do problema

A Web transformou-se num dos maiores repositórios de informação a nível mundial, constituindo actualmente um poderoso meio de difusão de informação, bem como um importante centro de negócios. Por estes motivos, cada vez mais as empresas migram os seu negócios para a Internet de forma a aproveitar as vantagens que lhe estão associadas. O estudo do comportamento dos utilizadores Web, é um tema que interessa sobretudo às empresas que migram os seu negócios para a Web, uma vez que nestes casos o utilizador é também um potencial cliente e, como tal, interessa mantê-lo. É neste sentido que as organizações estudam os seus dados Web de forma a compreender o tipo de utilizadores que normalmente visitam o seu *site.* Nestas situações, interessa sobretudo estudar algumas questões bastante pertinentes relacionadas com a forma como os utilizadores interagem com o *site,* por exemplo:

- Quais as páginas mais e menos acedidas?
- Quais os países dos utilizadores do *site*?
- Quais as páginas mais utilizadas para entrar e sair do *site*?
- Qual o browser mais utilizado pelos clientes?
- Quais as horas de maior e menor utilização do *site*?
- Qual a frequência com que um utilizador visita o *site*?

A análise deste tipo de informação pode trazer mais valias competitivas em relação a outras organizações que não façam este tipo de estudo. Uma vez que sendo a Web um mercado tão competitivo, em que um utilizador pode mudar de *site* com um simples *click*, conhecer o tipo de utilizadores que utiliza um *site* pode ajudar a implementar melhorias que agradem os utilizadores, evitando assim que estes abandonem o *site*.

Os servidores dos *site*s Web, registam num ficheiro de *log* todos os acessos feitos ao *site* por um determinado utilizador. Estes ficheiros constituem uma das principais fontes de dados para o estudo do comportamento dos utilizadores Web. Através da análise da informação contida nestes ficheiros, e depois de aplicadas algumas técnicas de mineração de dados, é possível estabelecer perfis de utilização para o *site* em estudo. O estabelecimento de perfis de utilização vai permitir estabelecer melhorias no *site* que vão de encontro ao tipo de utilizador que frequenta um *site.* Estas melhorias envolvem usualmente várias áreas e têm diferentes objectivos, tais como: podem ser utilizadas para personalizar o *site*, criar sistemas de recomendação, adaptar os conteúdos aos seus utilizadores, fornecer novas métricas de negócio, criar sistemas de previsão, melhorar desempenho do *site* ou ajudar a criação de campanhas de marketing .

6.2 Considerações à Abordagem Desenvolvida

A metodologia para a identificação de perfis de utilização Web seguida nesta dissertação, seguiu todas as fases de desenvolvimento de um projecto de mineração de dados de utilização Web. A experiência desenvolvida ao longo da fase de investigação, nas quais foram vistos vários projecto que envolvem aplicação de técnicas de mineração de dados a utilização Web, levou-nos a definir três etapas distintas no processo de identificação de perfis. A primeira etapa consistiu no pré-processamento aos dados, a segunda na aplicação de técnicas de mineração de dados e por fim a terceira etapa na análise dos padrões gerados.

A primeira etapa, começou com a recolha dos ficheiros de *log*, que são, como sabemos, a principal fonte de dados na aplicação deste tipo de técnicas. Depois de recolhidos os dados do ficheiros de *logs*, estes foram transformados para que posteriormente pudessem ser integrados num *Data Webhouse.* Os principais algoritmos aplicados, durante essa fase, actuaram na realização de tarefas como: a identificação de utilizadores, a identificação de sessões, a identificação de sessões ou a reconstrução de caminhos. A identificação de utilizadores foi feita com recurso ao campo *IP* e

User Agent de um ficheiro de *log*. Porém, esta identificação de utilizadores não é muito fiável pois torna possível que diferentes utilizadores possam ter o mesmo *IP* e usar o mesmo *browser*. A identificação de sessões também apresenta problemas de fiabilidade, uma vez que depende da identificação de utilizadores. A técnica de identificação de *crawlers* não é muito eficaz uma vez que está dependente de uma lista de *crawlers* previamente conhecidos. Ora os *crawlers* que não estão nessa lista não vão ser correctamente identificados. Depois de aplicados todos os algoritmos de transformação, os dados foram integrados num *Data Webhouse*.

Existem vários tipos de técnicas de mineração de dados utilizadas na identificação de perfis, nesta dissertação optámos por apresentar cinco das técnicas mais utilizadas: análise estatística, *clustering*, regras de associação, cadeias de *Markov* e classificação. Das cinco técnicas apresentadas apenas três foram utilizadas no estudo do caso prático: *clustering*, regras de associação e cadeias de *Markov*, uma vez que estas são as técnicas mais adequadas na identificação de perfis. Decidimos, também, por não utilizar as sessões atribuídas a agentes automáticos, uma vez que podiam influenciar os resultados de forma negativa, e as sessões com apenas um pedido, dado que estávamos a tentar identificar perfis de utilização (grupos de páginas acedidos frequentemente por um grupo de utilizadores), sabendo que estas sessões não acrescentariam informação relevante.

A implementação de técnicas de *clustering* e regras de associação foi feita com recurso ao programa *RapidMiner*. Para isso foi necessário construir a matriz de transacções e guardá-la num ficheiro de texto que depois foi passado ao *RapidMiner* como *input*. A construção da matriz foi feita através da linguagem Java, não apresentando grandes problemas. Para isso, apenas foi necessário percorrer todas as sessões feitas por utilizadores regulares e guardá-las num ficheiro. A aplicação de algoritmos de *clustering* e de regras de associação levantou alguns problemas de desempenho, sendo que apenas foi possível testar um algoritmo para cada uma das técnicas, uma vez que todos os outros que foram testados apresentaram sérios problemas de desempenho devido ao grande volume de dados.

Por sua vez, a implementação das cadeias de *Markov,* ao contrário dos algoritmos de *clustering* e de regras de associação não requereu a utilização de qualquer ferramenta externa. As cadeias de *Markov* foram, também, implementadas em Java. A implementação apresentou alguns problemas de memória devido ao grande número de estados (páginas) gerados, pelo que o grafo que

representa a cadeia teve de ser guardado numa tabela numa base de dados. Por este motivo o cálculo dos caminhos mais frequentes tornou-se mais ineficiente, já que foi necessário estar sempre a executar operações de consulta sobre a base de dados. Para visualizar as cadeias de *Markov* foi utilizado o *Graphviz*. Este também apresentou problemas, em grande parte devido ao número de estados gerados, o que fez com que se conseguisse representar apenas uma pequena parte da cadeia.

6.3 Comentários à Avaliação Prática

O caso de estudo escolhido diz respeito ao *site* do *repositóriUM*, um *site* institucional da Universidade do Minho cujo principal objectivo é armazenar, divulgar a produção intelectual da Universidade. Este *site* apresenta um volume de acessos de média dimensão, apesar de haver em algumas épocas (como a de exames) um aumento significativo de consultas. Os utilizadores do *site* do *repositóriUM* podem ser catalogados à partida em quatro grupos distintos: alunos, utilizadores registados, administradores e funcionários, e outros. Os três primeiros grupos, dizem respeito a utilizadores que já tinham conhecimento do *site*, enquanto que o último grupo diz respeito a utilizadores que muito provavelmente entraram no *site* através de motores de pesquisa.

A utilização do *site* do *repositoriUM* como caso de estudo permitiu avaliar algumas das técnicas de identificação de perfis estudadas ao longo desta dissertação. Os resultados obtidos com cada uma das técnicas foram de encontro aos grupos de utilizadores que se esperava obter inicialmente. Isto significa, que a nossa categorização inicial de utilizadores não se afastou muito da realidade. Desses quatro grupos, dois deles são simples de identificar, os utilizadores autenticados, que consultam sempre páginas relacionadas com o seu *login* e a conta que têm definida para o *site*, e o grupo dos funcionários e de administradores. O grupo de alunos foi um pouco mais complicado de identificar, uma vez que é difícil identificar qual o comportamento de um aluno perante o *site*. No entanto, podemos dizer que o perfil alunos corresponde aos utilizadores que entram no *site* pela página principal, procedendo depois com vários tipos de processos de pesquisa. No perfil alunos podem também incluir-se sessões referentes a professores, que não se tenham autenticado no *site*, uma vez que estes têm um perfil semelhante ao dos alunos. Por último, o perfil outros, foi o mais complicado de se definir, uma vez que representa uma grande quantidade e variedade de utilizadores. Este último perfil representa os utilizadores que entram no *site* referenciados por um motor de pesquisa, sendo que alguns deles utilizam depois este *site* para efectuar outras pesquisas

que sejam do seu interesse. Neste perfil nenhuma página apresenta uma percentagem alta de acessos, o que pode ser justificado pelo facto de muitos dos utilizadores entrarem apenas na página do documento que lhes interessa abandonando o *site* logo de seguida.

Os perfis identificados no *site* do *repositorioUM*, são perfis bastante abrangentes, que não estão relacionados com uma determinada área de pesquisa. Isto acontece porque existe uma grande quantidade de documentos publicados no repositório, tendo os utilizadores várias possibilidades de consulta, e, como tal, os utilizadores não seguem qualquer padrão de consulta. Ou seja, os documentos consultados pelos utilizadores são muito diversificados e, assim, o número de pessoas que visita um determinada conjunto de documentos acaba por não ser significativo em relação ao número total de sessões existentes no *site*.

Em termos gerais, as técnicas utilizadas para identificação de perfis foram tão efectivas quanto o necessário para permitir atingir os objectivos traçados para este trabalho, uma vez que identificaram de forma muito aproximada os quatro perfis que estavam previstos à partida. Contudo, isto não significa que quando aplicadas a outro *site,* com objectivos diferentes, os resultados obtidos sejam os mais adequados e com o mesmo tipo de efectividade. Para uma melhor avaliação das técnicas estudadas, deveríamos aplicá-las a casos de estudo diferentes, de modo a poder verificar como se comportam em *sites* com outros propósitos. Adicionalmente, o caso de estudo utilizado permitiu fazer uma comparação entre a efectividade dos vários tipos de técnicas de mineração de dados utilizadas sobre o caso seleccionado.

6.4 Contributos e Limitações desta Dissertação

A abordagem proposta nesta dissertação, para a identificação de perfis de utilização Web, representa apenas um ponto de partida para o estudo dos padrões de comportamento dos utilizadores de um *site* Web. As técnicas apresentadas para identificação de perfis de utilização Web, englobam as várias fases do processo, que vão desde o processamento dos ficheiros de *log,* até aplicação de técnicas de mineração de dados, de modo a descobrir grupos de utilizadores. De entre os diversos contributos desta tese, decidiu-se destacar os seguintes:

- Um conjunto de processos de tratamento de ficheiros de *Log* no formato ECLF.
- Um esquema (ainda) básico de detecção de Web *Crawlers.*

- Um modelo dimensional para um *data webhouse* capaz de acolher a informação contida num ficheiro de *logs.*
- Uma técnica de construção de matrizes de transacção sobre o qual podem posteriormente ser aplicados diversos algoritmos de mineração de dados.
- Um algoritmo para construção de cadeias de *Markov,* de grau 1, utilizando para isso a informação guardada num *data webhouse*.
- A implementação de um algoritmo de pesquisa em profundidade numa cadeia de *Markov* de modo a descobrir quais os caminhos mais frequentes efectuados por um utilizador dentro de um *site* Web.

As principais limitações deste projecto estão relacionadas sobretudo com questões operacionais, nomeadamente, aspectos de desempenho e problemas cujo resolução foi equacionada apenas a médio longo prazo. Dessas questões, destacam-se:

- A utilização de um método de identificação de utilizadores mais fiável, uma vez que o utilizado se limita a distinguir utilizadores apenas por *IP e User Agent*.
- Elaboração de uma lista de motores de pesquisa mais extensa.
- Um método de identificação de *crawlers* mais eficiente, que não obrigue a utilizar uma lista de *crawlers* previamente conhecidos.
- Uma maior diversidade de casos de estudo, com volumes de informação mais significativos e necessidades de análise e segurança mais específicas.
- A utilização de um maior número de algoritmos de clusters e regras de associação de modo a poder comparar resultados.
- A implementação de um modelo de cadeias de *Markov* de ordem superior a 1.

6.5 Trabalho Futuro

As principais linhas de trabalho futuro seguem em duas direcções. Uma das linhas assenta em implementar melhorias no trabalho desenvolvido. O processo de identificação de perfis de utilização Web é um processo bastante complexo, que envolve uma grande quantidade de dados. Os algoritmos estudados na elaboração desta dissertação apresentam várias limitações. Com vista a melhorar os resultados obtidos e, consequentemente, os perfis identificados, podemos aplicar

outros algoritmos de identificação de perfis. Algumas das melhorias que podem ser implementados no futuro são, nomeadamente:

- **Utilizar um conjunto de dados maior.** Aplicar os vários algoritmos estudados a um conjunto de dados maior, uma vez que a amostra utilizada - dois meses de dados - pode não ser significativa ou suficiente para capturar todos os possíveis perfis.
- **Melhorar a implementação das cadeias de *Markov*.** Implementar cadeias de Markov de ordem superior a 1, uma vez que estas podem ser mais eficientes a prever os caminhos feitos pelos utilizadores.
- **Aplicar um modelo misto de cadeias de *Markov* e *Clustering*.** Aplicar os algoritmos de cadeias de *Markov* a cada um dos clusters identificados. Deste modo poderemos caracterizar as sequências de páginas mais visualizadas pelos utilizadores de cada um dos clusters identificados.
- **Aplicar um modelo misto de *Clustering* e Regras de associação.** Utilizar algoritmos de regras de associação sobre cada um dos clusters identificados. Deste modo poderemos caracterizar as páginas visualizadas pelos utilizadores de cada um dos clusters.

A segunda linha de trabalho futuro estará relacionada com a utilização da informação obtida, ou seja, dos perfis gerados para implementar algumas funcionalidades no *site* em estudo. Alguns dos projectos que podem ser desenvolvidos com recurso aos perfis estabelecidos poderão ser os seguintes:

- **Criar um modelo de classificação.** Utilizar os resultados obtidos pela aplicação de algoritmos de *clustering* para criar um modelo de classificação, que classifique as novas instâncias, usando a matriz como atributos de previsão e o cluster como atributo objectivo.
- **Criar um sistema de recomendação**. Com base nos resultados obtidos pela aplicação de algoritmos de regras de associação, criar um sistema de recomendação. Este sistema pode ser criado colocando *hiperlinks* entre as páginas que aparecem frequentemente juntas na mesma sessão.
- **Criar um sistema de previsão**. Com base no resultado da aplicação das cadeias de *Markov*, criar um sistema de previsão que consiga prever o comportamento de um utilizador a partir do momento em que este entra no *site*. Esta previsão poderá ser feita com base no percurso realizado por utilizadores anteriores

Bibliografia

[Agrawal and Srikant 94] Agrawal R., Srikant R. "Fast Algorithms for mining Association Rules". In Proceedings of the 20th Conference on Very Large Data Bases, Santiago do Chile, Setembro 1998.

[Agrawal and Srikant 95] Agrawal R., Srikant R., "Mining Sequential Patterns". In proceedings of the international conference on data Engineering (ICDE'95) Taiwan, March 1995.

[Almeida et al 96] Almeida V., Bestravos A, Crovella M., Oliveira A. "Characterizing reference Locality in the WWW". Technical Report TR96-11, Boston, University, 1996.

[Berkhin 02] Berkhin P., "A Survey of Clustering Data Mining Techniques", Springer, 2002

[Boley 98] Boley D.L., "Principal Direction Divisive Partitioning", Data Mining and Knowledge Discovery, 1998.

[Borges 04] Borges E., "Sistemas de Data Webhousing: Análise, Desenho, Implementação e Exploração em Sistemas Reais", Universidade do Minho, 2004

[Borges and Levene 99] Borges J., Levene M. "Data Mining of User Navigation Patterns. In Web Usage Analysis and User Profiling",LINAI 1836m, Springer, 1999.

[Brin 98] Brin, S. "Extracting patterns and relations from the world wide web". Em International Workshop on the Web and Databases, Valencia, Espanha,1998.

[Brin and Page, 98] Brin, S. and Page, L. "The anatomy of a large-scale hypertextual web search engine". Em Proceedings of the seventh International World Wide Web Conference, Brisbane, Australia., 1998

[Buchner and Mulvenna 98] Buchner A., Mulvenna M. D. "Discovering Internet market Intelligence through online analytical web usage mining". SIGMOD Record, 1998.

[Buchner and Mulvenna 99] Buchner A. Mulvenna M. D. "Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining". In Proc. of the ACM SIGMOD Intl. Conf on Management of Data (SIGMOD'99), 1999.

[Catledge and Pitkow95] Catledge L., Pitkow J., "Characterizing browsing behaviors on the World Wide Web". Computers Network and ISDN Systems, Elsevier Abril, 1995.

[Chakrabarti et al.99], Chakrabarti, S., Dom, B. E., Gibson, D., Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. "Mining the link structure of the world wide web." IEEE Computer, 1999.

[Chapman et al 00] Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., Wirth R. "CRISP-DM 1.0 Step-by-Step data mining guide", http://www.crisp-dm.org/CRISPWP-0800.pdf, 2000.

[Chau and Chen 03] Chau M., Chen H. "Personalises on Focused Web Spiders". Web-Intelligence, Springer- Verlag, Setembro, 2003.

[Chen and Yu 96] Chen, M.-S. and Yu, P. S. "Data mining: An overview from a database perspective". IEEE Transactions on Knowledge and Data Engineering, 1996.

[Cooley et al 97] Cooley R., Mobasher B., Srivastva J., "Web Mining: Information and Pattern Discovery on the World Wide Web". Em Proc. of the 9th IEEE Intl. Conf. On Tools With Artificial Intelligence (ICTAI'97), 1997.

[Cooley et al. 99] Cooley R., B. Mobasher B., Srivastava J. "Data preparation for mining World Wide Web browsing patterns". Journal of Knowledge and Information Systems, 1(1), 1999.

[Craven et al., 98] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. "Learning to extract symbolic knowledge from the world wide web". In Proceedings of the fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin, 1998.

[Dempster et al 77] Dempster A., Laird N., Rubin D., "Maximum Likelihood from Incomplete Data Via the EM Algorithm", Journal of the Royal Statistical Society, 1977.

[Deshpande and Karypis 04] Deshpande M., Karypis G. "Selective Markov Models for Predicting Web Page Accesses". ACM Trans. on Internet Technology, 2004.

[Etizione 96] Etzioni O. "The World Wide Web: a quagmire or gold mine? " Communications of the ACM, 1996.

[Frawley et al 91] Frawley W. J., Piatetsky-Shapiro G., C. J. Matheus C. J. "Knowledge discovery in databases: An overview". In Piatetsky-Shapiro, G. and Frawley, W. J., editors, Knowledge Discovery in Databases,. AAAI/MIT Press, 1991.

[Fayyad et al 96] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. "From data mining to knowledge discovery: An overview". Em Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[Hallam and Behlendorf 96a] Phillip M. Hallam-Baker, Brian Behlendorf: "Extended Log Format". W3C Working Draft WD-logfile-960323, 1996, http://www.w3.org/pub/WWW/TR/WD-logfile.html.

[Hallam and Behlendorf 96b] Phillip M. Hallam-Baker, Brian Behlendorf: "Extended Log File Format". World Wide Web Journal: The Web Five Years After, Volume1, Number 3, O'Reilly & Associates, Setembro, 1996, http://www.w3journal.com/3/s2/hallam.htm.

[Han and Kamber 07] Han J., Kamber M. "Data Mining: Concepts and Techniques". Morgan Kaufmann, 2007.

[Hartigan 75]Hartigan J.A., "Clustering Algorithms", Wiley, 1975.

[Hartigan and Wong 79]Hartigan J.A., Wong M., "Algorithm as136: A K-Means Clustering Algorithm", Applied Statistics, 1979.

[Herlocker et al 04] Herlocker J. L., Konstan J., Terveen L., Riedl J. "Evaluating Collaborative Filtering Recommender Systems". ACM Transactions on Information Systems, 2004.

[Jain and Dubes 88]Jain A., Dubes C., "Algorithms for Clustering Data", Prentice-Hall, NJ, 1988

[Joachins et al. 97] Joachins T., Freitag D, Mitchell T. "Webwatcher: A tour guide for the world wide web. In the 15th International Conference on Artificial Intelligence, Nagoya, Japão 1997.

[Kaufman L. and Rousseeuw P.J., 90] Kaufman L., Rousseeuw P.J., "Finding Groups in Data: An Introduction to Cluster Analysis", Wiley, New York, 1990.

[Kimball et al 98] Kimball R., Reeves L., Ross M, Thornthwalte W. "The *Data Warehouse* Lifecycle Toolkit – Expert Methods for Designing, Developing and Deploying, *Data Warehouses".* Jonh Wiley

[Kimball and Mertz 00] Kimball R., Mertz R. "The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse. Jonh Wiley & Sons Inc., 2000.

[Kleinberg 99] Kleinberg, J. M. "Authoritative sources in a hyperlinked environment." Journal of the ACM, 1999.

[Kohavi et al 04] Kohavi R., Mason L., Parekh R., Zheng Z. "Lessons and Challenges from Mining Retail E-commerce Data". Machine Learning, 2004.

[Kohavi and Parekh 03] Kohavi R., Parekh R., "Ten supplementary analyses to improve E-commerce Web Sites". Em proceedings of Fifith International Workshop on Konwledge Discovery in the Web WEBKDD'2003- Webmining as a premisse to Effective and Intelligent Web Applications, 2003.

[Koster 94] Koster M., " A standard for Robot Exclusion." 1994, http://www.robotstxt.org/wc/norobots.html.

[Lieberman 95] Lieberman H. "Letizia: Agent that assists web browsing". In proc. of the 1995 International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995

[Liu, 08] Bing Liu, Web Mining: " Hyperlinks, Contents and Usage Data". Springer, 2008

[Lourenço 04] Lourenço A., "Detecção e previsão comportamental de Web Crawlers baseadas na análise de padrões de navegação", Universidade do Minho, 2006

[McLachlan and Krishnan 96] McLachlan G.J., Krishnan T., "The EM Algorithm and Extensions", Wiley, 1996.

[Mena 99] Mena, J. "Data Mining your Website". Digital Press, Boston, Massachusetts, 1999.

[Mendelzon et al., 96] Mendelzon, A. O., Mihaila, G. A., and Milo, T. "Querying the World Wide Web". In Proceedings of the fourth International Conference on Parallel and Distributed Information Systems, Miami Beach, Florida, 1996.

[Mobasher et al 99] Mobasher B., Cooley R., Sriivastava J. "Creating adaptive Web Sites through usage based clustering of urls". In Knowledge and Data Engineering Workdshop, 1999.

[Mobasher et al 01a] Mobasher B., Berendt, B., Spiliopoulou, M. "KDD for Personalization – PKDD 2001 Tutorial", Setembro 2001.

[Mobasher et al 01b] Mobasher B., Dai H., Luo T., Nakagawa N. "Effective Personalization Based on Association Rule Discovery from Web Usage Data." In Proc. of the 3rd ACM Workshop on Web Information and Data Management(WIDM01), 2001.

[Mobasher et al 02] Mobasher B., Dai H., Luo T., Nakagawa M. " Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization". Data Mining and Knowledge Discovery, 2002.

[Mobasher 05] Mobasher B. "Web Usage Mining and Personalization". In Munindar P. Singh(ed.) Pratical Handbook of Internet Computing, 2005.

[Mobasher 06] Mobasher B. "Web Usage Mining". In Jonh Hang(Eds), Encyclopedia of Data Warehousing and Mining, Idea Group, 2006.

[Murtagh 83] Murtagh F., "A Survey of Recent Advances in Hierarchical Clustering Algorithms", Computer Journal,1983.

[Ngu and Wu 97] Ngu D.S.W., Wu X. "SiteHelper: A localized agent that helps incremental exploration of the world wide web", In 6th International World Wide Web Conference, Santa Clara, 1997.

[Page et al., 98] Page, L., Brin, S., Motwani, R., Winograd T. "The PageRank citation ranking: Bringing order to the web". http://google.stanford.edu/ backrup/pageranksub.ps, 1998.

[Paliouras et al 02] Paliouras G., Papatheodorou C., Karkaletsis V., Spyropoulos. "Discovering User Communities on the Internet Using Unsupervised Machine Learning Techniques". Interacting with Computers Journal, 2002.

[Perkowitz and Etzioni 98] Perkowitz M., Etzioni O. "Adaptive Web Sites: Automatically synthesizing web pages ". In Fifteenth National Conference on Artificial Intelligence, 1998.

[Perkowitz and Etzioni 99] Perkowitz M., Etzioni O. "Adaptive Web Sites: Conceptual Cluster Mining". In Sixteen International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999.

[Perkowitz and Etzioni 00] Perkowitz, M. and Etzioni, O. "Towards adaptive web sites: Conceptual frame- work and case study". Artificial Intelligence, 2000.

[Pierrakos et al 03] Pierrakos G., Paliouras G., Papatheodorou C., Spyropoulos C. "Web Usage Mining as a Tool for Personalization: a Survey". User Modeling and User –Adapted Interaction, 2003.

[Sarukkai 00] Sarukkai R. R. "Link Prediction and Path Analysis Using Markov Chains. In proc. of the 9th Intl.World Wide Web Conf., 2000.

[Sarwar et al 00] Sarwar B., Karipys G., Konstan J., Riedl J. "Application of Dimensionality Reduction in Recommender Systems – A Case Study. In Proc. of the KDD Workshop on WebKDD'2000, 2000.

[Schechter et al 98] Schechter S., Krishnan M., Smith M. D. "Using Path Profiles to predict HTTP requests". In 7th International World Wide Web Conference , Australia, 1998.

[Spertus 97] Spertus, E. "ParaSite: Mining structural information on the web".Computer Networks and ISDN Systems, 1997.

[Spiliopoulou et al 03] Spiliopoulou, M., Mobasher B., Berendt, B., Nakagawa M.: "A framework for the evaluation of session reconstruction heuristics in Web Usage Mining". Informs Journal on Computing, Abril, 2003.

[Srivastava et al 00] Srivastava J., Cooley R., Deshpande M., Tan P. "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations , 2000.

[Tanasa and Trousse 04] Tanasa D., Trousse B. " Advanced Data Preprocessing for Intersite Web Usage Mining. IEEE Intelligent Systems, 2004.

[Tarjan 62] Tarjan R., "Depth-first search and linear graph algorithms". Siam Journal on Computing, 1962.

[Weiss 93] Weiss M. A., " Data Structures and Algorithm Analysis in C". The Benjamin/Cummings Publishing Company, Redwood City, Califórnia.

[W3C99] World Wide Web Commitee Web Usage Characterization Activity: "W3C Working Draft: Web Characterization Terminology & Defenitions Sheet". 1999, Http://www.w3c.org/1999/05/WCA-terms/.

[Za¨ıane and Han, 00]. Za¨ıane, O. R. and Han, J. "WebML: Querying the world-wide web for resources and knowledge". Em Proceedings of the Workshop on Web Information and Data Management, Washington, 2000

Referências WWW

[WWW 1] "Internet World Statistics" http://www.internetworldstats.com/

Este *site* fornece informação estatística sobre a utilização da Internet, nos vários países do mundo.

[WWW 2] http://news.netcraft.com/

Este *site* fornece informação sobre o número de *site*s alocados nos vários servidores Web existentes. Fornece também informação sobre o número total de *site*s existentes em todos os domínios Web.

[WWW 3] http://www.analog.cx/

Site da ferramenta de análise de *logs Analog*. Esta ferramenta é uma das mais populares ferramentas gratuitas de análise de *logs*. No *site* é possível fazer download do ferramenta, assim como ver alguns exemplos de relatórios gerados com a ferramenta.

[WWW 4] http://www.weblogexpert.com/

Site da ferramenta de análise de *logs WebLog Expert*. Permite fazer download gratuito de uma versão reduzida do programa, ou então comprar a versão total. Disponibiliza informação sobre o tipo de *logs* que a ferramenta suporta entre outras. Fornece também um exemplo de uma relatório gerado com a informação dos *logs.*

[WWW 5] http://www.summary.net/manual/log_formats.html

Site com informações sobre os diversos formatos de ficheiros de Logs Web. Disponibiliza informações sobre os diversos servidores Web e qual o tipo de ficheiro de logs que cada um deles regista.

[WWW 6] "World Wide Web Consortium" http://www.w3.org/

Site da comunidade W3C. O W3C é uma comunidade internacional que desenvolve *standards* que visam garantir o crescimento sustentável da Internet.

[WWW 7] http://www.w3.org/Daemon/User/*Config*/Logging.html.

Site com a descrição do formato de ficheiros de *log CLF.*

[WWW 8] http://hoohoo.ncsa.uiuc.edu/docs/

Fornece documentação do servidor "NCSA HTTPd". NCSA. Na documentação estão incluídas informações sobre os formatos de ficheiros CLF, ECLF usados pelo servidor HTTPd.

[WWW 9] "RepositóriUM" http://repositorium.sdum.uminho.pt/

Este *site* é um repositório institucional da Universidade do Minho, constituído com o objectivo de armazenar, preservar, divulgar e dar acesso à produção intelectual da Universidade do Minho em formato digital. Pretende reunir num único sitio, o conjunto das publicações científicas da UM contribuindo deste modo para o aumento da sua visibilidade e impacto e garantindo a preservação da memoria intelectual da Universidade do Minho.

[WWW 10] "Universidade do Minho" http://www.uminho.pt

Site institucional da Universidade do Minho. Fornece as mais variadas informações da universidade, desde noticias sobre a universidade, até informações sobre os cursos existentes.

[WWW 11] http://www.maxmind.com/download/geoip/database

Este *site* disponibiliza um ficheiro com a atribuição de *Ips* por pais.

[WWW 12] http://www.google.com

Site do motor de pesquisa Google. Este *site* permite fazer pesquisas sobre os mais diversos assuntos, tendo indexado *sites* de todo o mundo.

[WWW 13] http://www.sapo.pt

O sapo é um motor de pesquisa português, que permite fazer pesquisas sobre os mais diversos assuntos.

[WWW 14] http://www.robotstxt.org/

Este *site* fornece informação sobre os *crawlers* conhecidos até ao momento. Permite aos utilizadores fazerem download de forma gratuita, de um ficheiro com diversas informações sobre os *crawlers* catalogados.

[WWW 15] http://www.crisp-dm.org/

Este *site* apresenta a metodologia *Crips-DM.* Fornece informação detalhada sobre todas as fases envolvidas no processo de aplicação de técnicas de Mineração de Dados.

[WWW 16] http://rapid-i.com/

Site da ferramenta *rapidminer.* Este *site* disponibiliza uma versão gratuita, *da* ferramenta *rapidminer* para *download*. Disponibiliza também uma versão paga, com mais funções. Para além de disponibilizar a ferramenta fornece também informação sobre esta. Esta informação torna a interacção dos utilizadores com a ferramenta mais simples.

[WWW 17] http://www.cs.waikato.ac.nz/ml/weka/

Site da ferramenta Weka. Neste *site* é disponibilizada uma versão gratuita da ferramenta Weka para download. Para além da ferramenta fornece também informação sobre esta. Esta informação vai permitir uma melhor interacção entre a ferramenta e os utilizadores.

[WWW 18] http://www.java.sun.com/

Este *site* fornece uma elevada quantidade de informação acerca da linguagem orientada aos objectos Java. É possível encontrar uma vasta gama de recursos que vai desde as versões mais actuais do Java, passando por uma extensa documentação, até grupos de discussão e investigação.

[WWW 19] http://www.graphviz.org/

Site ferramenta *Graphviz*, ferramenta que permita a visualização de grafos entre outras coisas. Este *site* disponibiliza gratuitamente uma versão da ferramenta. Para além de disponibilizar a ferramenta fornece também informação sobre esta, bem como alguns exemplos de como utilizar a ferramenta.