Utilização de aprendizado de máquina para a adaptação de estruturas em hipermídia adaptativa

Alfredo Lanari de Aragão

Orientador: *Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências da Computação e Matemática Computacional.

“VERSÃO REVISADA APÓS A DEFESA”

Data da Defesa:	16/02/2004
Visto do Orientador:	

USP – São Carlos
Março/2004

*Utilização de aprendizado de máquina
para a adaptação de estruturas em
hipermídia adaptativa*

Alfredo Lanari de Aragão

A Deus, pelas oportunidades de progresso
Aos meus pais e irmãs, pelo convívio e pela minha formação
À minha esposa Ana Paula, pelo apoio nos momentos difíceis
e por suportar as minhas ausências

Agradecimentos

Ao Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho,
pela orientação, amizade e confiança ao longo do programa de mestrado.

À Universidade Católica Dom Bosco,
pelo apoio financeiro.

Aos administradores do *cluster* Helder, Cleber e Rodrigo,
pela atenção e o suporte técnico nos momentos necessários.
Aos amigos e colegas de trabalho Alexsandro, Arlinda, Amaury, Conceição,
Cláudia, Hermeson, Ivanilde, Jeferson, Leonardo, Reinaldo,
pela compreensão e momentos de descontração,

Aos colegas do LABIC, em especial Katti, Ronaldo, Rodrigo e Cláudia,
pela disponibilidade em auxiliar e trocar experiências

À Ernesto, Marco Antônio, José Antonio, Rogério Furukawa, Cesar,
pela amizade e a oportunidade de aprendizado pela convivência
mesmo que seja com apenas alguns minutos de conversas pelo corredor.

À Marco Antônio e Luciano Gonda,
pela ajuda com gcc for Linux

Sumário

Introdução	1
1 Sistemas Hipermídia	5
1.1 Dimensões da Hipermídia	6
1.2 A World Wide Web	10
1.3 Classificação dos Sistemas Hipermídia	13
1.4 Considerações Finais	16
2 Aprendizado de Máquina	17
2.1 Conceitos de AM Indutivo	18
2.2 Paradigmas de AM	22
2.3 Avaliação de Algoritmos de AM	23
2.4 Técnicas de AM Utilizadas	26
2.4.1 Coeficiente de Correlação de Pearson	26
2.4.2 Classificador Bayesiano	27
2.4.3 Perceptron Multicamadas	28
2.4.4 Máquina de Vetores de Suporte	30
2.5 Considerações Finais	31
3 Sistemas Hipermídia Adaptativos	33
3.1 Motivação para Hipermídia Adaptativa	34
3.2 Modelos para Hipermídia Adaptativa	38
3.3 Modelagem de Usuários	41
3.3.1 Aquisição de Dados	43
3.3.2 Representação de Dados	45
3.3.3 Métodos e Técnicas de Adaptação	46

3.4	Considerações Finais	50
4	Metodologia de Trabalho	51
4.1	Pré-processamento da Base	52
4.2	Algoritmos de AM	58
4.3	Critérios de Avaliação	59
4.4	Considerações Finais	60
5	Realização dos Experimentos	61
5.1	Aplicação	61
5.2	Resultados dos Experimentos	62
5.3	Análise dos Resultados	64
5.4	Considerações Finais	68
6	Conclusão	69

Lista de Figuras

1.1	Máquina abstrata de hipertexto (Campbell & Goodman, 1987).	8
1.2	Modelo de referência Dexter.	9
1.3	Classificação dos sistemas de <i>software</i>	14
1.4	Exemplos de Sistemas Adaptativos.	15
2.1	Tipos de Aprendizado Indutivo.	19
3.1	Modelo de Aplicação para Hipermídia Adaptativa (AHAM).	39
3.2	Processo de personalização em sistemas adaptativos.	43
3.3	Tecnologias de personalização Hipermídia (Brusilovsky, 2001).	48
4.1	Classificador aplicado ao ambiente WWW.	51
4.2	Arquivo MSWeb inicial em formato DST.	53
4.3	Número de páginas acessadas por usuário.	56

Lista de Tabelas

3.1	Métodos para a apresentação adaptativa de texto.	48
3.2	Técnicas para apresentação adaptativa de texto.	49
3.3	Métodos para suporte navegacional adaptativo.	49
3.4	Técnicas para o suporte navegacional adaptativo	50
4.1	Resumo da descrição da base de dados	58
5.1	Taxa de erro dos classificadores em cada base.	63
5.2	As cinco melhores classes para cada classificador em MSWeb-2.	64
5.3	As cinco melhores classes para cada indutor em MSWeb-3.	64
5.4	Classes corretas para o classificador Bayesiano em MSWeb-2.	65
5.5	Classes corretas para o classificador Bayesiano em MSWeb-3.	65
5.6	Classes corretas para SVM em MSWeb-2.	66
5.7	Classes corretas para SVM em MSWeb-3.	66
5.8	Classes corretas para PEARSON em MSWeb-2.	67
5.9	Classes corretas para PEARSON em MSWeb-3.	67

Glossário

Aplicação Hipermédia - banco de dados não-linear para um determinado domínio, formado por nós interligados.

Aprendizado de Máquina (AM) - sub-área da Inteligência Artificial de grande utilidade que simula o aprendizado humano usando um processo de aquisição automática de conhecimentos à partir de novos dados.

Common Gateway Interface (CGI) - interface que estabelece a maneira pela qual programas externos podem trocar informações e serem executados por um servidor HTTP.

Extensible Markup Language (XML) - é uma linguagem de marcação extensível definida pelo usuário.

Filtragem de Informação (FI) - campo de estudo da Computação que desenvolve métodos e técnicas de seleção de informações não estruturadas em grandes bases de dados que supostamente ajustam-se aos interesses dos usuários baseado no grau de semelhança com informações previamente classificadas pelo mesmo indivíduo ou por grupos de indivíduos.

Hipermédia - meio apropriado para a criação e o acesso não-linear à informações de natureza diversa (textos, imagens, áudio, vídeo, animação, etc).

Hipermédia Adaptativa (HA) - a Hipermédia com capacidade de adaptar-se automaticamente à certas características dos usuários.

Inteligência Artificial (IA) - área da Computação voltada para o desenvolvimento de técnicas que permitem ao computador demonstrar comportamentos considerados inteligentes quando observados em seres humanos.

Modelagem de Usuários (MU) - campo de estudo da Computação que preocupa-se com a construção de modelos de usuários

Recuperação de Informação (RI) - campo de estudo da Computação que desenvolve métodos e técnicas que auxiliam o usuário a obter documentos que su-

postamente satisfaçam suas necessidades de informação, auxiliando na resolução de problemas.

Sistema Adaptativo (SA) - sistemas interativos capazes de identificar automaticamente a necessidade de adaptação, podendo realizá-la (ou não) no lugar do usuário.

Sistema de Recomendação (SR) - sistema que implementa técnicas de Filtragem de Informação de maneira colaborativa utilizando perfis de grupos de usuários para realizar indicações a respeito de um objeto qualquer (livro, CD, artigo, etc).

Sistema Hipermídia (SH) - programa de computador que implementa os mecanismos de criação e acesso não-linear.

Sistema Hipermídia Adaptativo (SHA) - sistema hipermídia que reflete características do usuário, de uso ou do ambiente para personalizar conteúdos, ligações e/ou a apresentação de objetos Hipermídia.

Sistema Interativo (SI) - programa de computador que reage à interação com o usuário por meio de diálogos ao longo de sua utilização.

Sistema Não-Interativo (SNI) - programa de computador que executa sem nenhuma interferência do usuário.

Sistema Tutor Inteligente (STI) - sistema especialista em um determinado assunto que é projetado para compartilhar o conhecimento, sendo capaz de guiar de maneira inteligente o estudante no processo de aprendizagem.

World Wide Web (WWW) - serviço da Internet para o armazenamento e a recuperação distribuída de informações Hipermídia.

Resumo

Sistemas Hipermedia são programas capazes de armazenar e recuperar informações não-lineares, estabelecendo uma estrutura complexa e flexível representada por nós interligados. À medida em que aumenta o espaço de navegação, tal como acontece na *World Wide Web* (WWW ou Web), a possibilidade de desorientação do usuário no espaço de navegação torna-se maior. Assim, a Hipermedia Adaptativa investiga métodos e técnicas para a adaptação automática de conteúdos e/ou ligações para características, interesses ou objetivos individuais. Trabalhos recentes em Hipermedia Adaptativa sugerem o uso de técnicas de Aprendizagem de Máquina e Modelagem de Usuários. Este trabalho investiga o uso de técnicas de Aprendizagem de Máquina para a adaptação de estruturas (ligações) em um ambiente Hipermedia, em especial a *World Wide Web*. Para tanto, avalia-se o desempenho de diferentes algoritmos de Aprendizagem de Máquina para a adaptação de ligações em ambiente WWW. Os resultados experimentais obtidos sugerem o potencial do emprego de técnicas de Aprendizagem de Máquina.

Abstract

Hypermedia Systems are programs able to store and recover nonlinear information, establishing a complex and flexible structure represented by linked nodes. At the same time that navigation space increases, such as happens in World Wide Web (WWW or Web), the number of possibilities of user's disorientation in the navigation space becomes larger. Thus, Adaptive Hypermedia investigates methods and techniques for the automatic contents and/or links adaptation for the individual characteristics, interests or objectives. Recent works in Adaptive Hypermedia suggest the use of Machine Learning and Users Modelling techniques. This work investigates the use of Machine Learning techniques for the adaptation of structures (links) in a Hypermedia environment, in special the World Wide Web. Thus, it evaluates the performance of different Machine Learning algorithms for link adaptation in a WWW environment. Experimental results obtained suggest the potential of the employment of Machine Learning in techniques.

Introdução

O computador oferece um conjunto de elementos e uma mídia capaz de armazenar, processar e transmitir informações com rapidez e alto desempenho. A evolução das interfaces de cartões perfurados até os modernos ambientes gráficos para os usuários (*Graphical Users Interfaces* - GUI) permitiu a utilização cada vez mais fácil e direta dos computadores (Preece, 1994). Por esta razão, os computadores tornaram-se ferramentas importantes em diversos setores tanto para o controle automatizado quanto para o armazenamento e a recuperação de informações com rapidez. Entretanto, para que uma grande quantidade de informações diversas possam ser utilizadas de maneira efetiva pelo ser humano, é importante que possam ser recuperadas conforme os interesses ou objetivos dos usuários.

Os Sistemas Hipermídia têm sido usados, tradicionalmente, como um repositório de informações livremente estruturadas que podem ser recuperadas de maneira não-linear. Esse repositório é implementado na forma de nós (fragmentos de informações) interligados, constituindo um banco de dados não-linear também denominado Hiperdocumento ou Aplicação Hipermídia. Há duas maneiras principais de utilizar um Sistema Hipermídia: para autoria ou para navegação. Quando empregado para autoria o usuário recebe a denominação de autor, sendo responsável pela criação do banco de dados não-linear. Já quando o sistema é usado para navegação, o usuário pode, apenas, explorar o repositório de informações que, neste caso, recebe o nome de espaço de navegação. Para tanto, o SH utiliza-se de uma interface interativa simples baseada em janelas que exibem o conteúdo de um nó e âncoras visivelmente destacadas que fazem a ligação com outros nós.

Analisando a essência da Hipermídia, Conklin (1987) identifica dois problemas intrínsecos à natureza não-linear: a sobrecarga de informação e a desorientação no espaço de navegação. Muitos pesquisadores apontam estas falhas da Hipermídia

como um impedimento para a sua aplicação em diversas áreas, provocando discussões calorosas com aqueles que defendem as suas vantagens. Muitas pesquisas têm sido realizadas no sentido de minimizar a ocorrência destes problemas, principalmente com relação à desorientação do usuário, que é justamente a motivação para este trabalho. Através da aplicação de técnicas adequadas, pretende-se indicar as ligações que conduzem o usuário aos conteúdos que possivelmente sejam do seu interesse para atingir um objetivo específico.

De Troyer & Casteleyn (2003) consideram a *World Wide Web* (WWW ou *Web*) como sendo o Sistema Hipermídia mais conhecido do mundo. Sua característica aberta e distribuída aliada à uma crescente expansão fizeram com que uma grande quantidade e diversidade de informações fossem disponibilizadas, aumentando a possibilidade de um usuário perder-se no espaço de navegação. Este cenário dificulta, ainda, a tarefa de encontrar dados que atendam aos interesses ou necessidades dos usuários. O desenvolvimento de tecnologias suplementares como CGI (*Common Gateway Interface*) e XML (*Extensible Markup Language*) e a necessidade de encontrar a informação que atenda aos interesses individuais contribuíram para a criação de mecanismos e o uso de técnicas que auxiliassem o usuário a navegar pela estrutura Hipermídia, bem como encontrar a informação desejada. Desta maneira, a WWW mostra-se um ambiente ideal para a implementação e o teste de mecanismos desta natureza.

A Hipermídia Adaptativa oferece mecanismos para adaptar automaticamente o conteúdo e/ou as ligações de uma Aplicação Hipermídia de acordo com certos aspectos individuais (Brusilovsky, 1996a, 1996b; Kobsa et al., 1999). Para tanto, diversos métodos e técnicas foram desenvolvidos e utilizados para tornar os Sistemas Hipermídia aptos para a produção de adaptações para os usuários. Este trabalho concentra-se no processo de adaptação automática de ligações (*links*) em ambiente WWW. Para tanto, utilizam-se técnicas provenientes da Modelagem de Usuários (MU) e da Inteligência Artificial (IA).

A adaptação automática de ligações pressupõe que o sistema deve decidir como realizá-la. Para tanto, diversas abordagens podem ser utilizadas. As primeiras iniciativas em Hipermídia Adaptativa consistiam na migração dos Sistemas Tutores Inteligentes (STI) para o ambiente WWW, como mostra Brusilovsky (1995, 1996b, 2000). Tratavam-se de aplicações na área da Educação que utilizavam o conheci-

mento a respeito do domínio e dos usuários para decidir como realizar as adaptações. Para tanto, eram utilizadas técnicas simbólicas de IA. Com o passar do tempo e a sistematização dos estudos, novas aplicações foram surgindo permitindo a investigação e a utilização de outras técnicas para o desenvolvimento de Sistemas Hipermídia Adaptativos. Neste processo, o uso da IA simbólica mostrou-se inadequado para uma série de aplicações, forçando a utilização de técnicas diferentes.

O resultado da investigação no campo dos sistemas adaptativos resultou no desenvolvimento de várias aplicações com as mais diferentes tecnologias, tais como os Sistemas de Recomendação (Middleton, 2001; Oard & Kim, 1998; Resnick et al., 1994; Resnick & Varian, 1997), os Sistemas de Filtragem de Informação (Ambrosini et al., 1997; Morita & Shinota, 1994; Nichols, 1997), os Sistemas de Recuperação de Informação (Armstrong et al., 1995; Balabanovic et al., 1996) e os Sistemas Hipermídia Adaptativos (Brusilovsky, 1996a; Brusilovsky et al., 1996).

O Aprendizado de Máquina (AM) é uma sub-área bastante útil da IA que utiliza o conceito de aprendizado (Mitchell, 1997). Suas técnicas costumam ser utilizadas para prever eventos, obter generalizações à partir de um conjunto de características significativas e agrupar dados similares. Os algoritmos de AM apresentam uma fase de treinamento para gerar um classificador capaz de encontrar a classe ou o valor de saída de novas instâncias. Podem ser usados para gerar suposições a respeito dos interesses e objetivos dos usuários com base em dados implícitos obtidos durante a interação com o sistema.

Um Sistema Hipermídia Adaptativo que faz uso de um classificador apresenta algumas vantagens em relação aos que adotam a abordagem simbólica para representar um ou mais modelos usados para gerar as adaptações (Mullier, 1999; Mullier et al., 1999a). Sistemas com abordagem simbólica utilizam as regras existentes em um modelo de adaptação para disparar as personalizações que o autor consideram mais importantes em determinada situação. Para tanto, as regras de adaptação combinam informações provenientes do modelo do domínio - normalmente na forma de Rede Semântica - e de um modelo do usuário - em geral um modelo de sobreposição. Percebe-se que este tipo de sistema é centrado no autor e precisa ser reformulado para cada novo domínio. Já um Sistema Hipermídia Adaptativo baseado em um classificador é centrado no usuário. Ele aprende a maneira como os indivíduos que utilizam o sistema se comportam para atingir seus objetivos. Para isto, basta obter

as informações necessárias para formar uma base de dados que possa servir de treinamento para alguns algoritmos de AM. Desta forma, o classificador torna-se o modelo de usuário do sistema. A utilização de aprendizado incremental pode acrescentar a vantagem de atualizar o modelo de usuário de maneira automática.

O objetivo desta dissertação consiste em avaliar o desempenho de algoritmos de AM para adaptar as ligações de aplicações *Web*. Neste documento, a adaptação de ligações é tratada como um problema de classificação de páginas *Web* baseado na seqüência em que as mesmas são visitadas. Desta maneira, o classificador obtido pode ser utilizado para adaptar a estrutura de navegação de um *website*.

Este trabalho está dividido em seis Capítulos organizados da maneira que se segue. O Capítulo 1 abrange tópicos relacionados à Hipermídia, abordando-a sob aspectos como navegação e autoria. São tratados, também neste Capítulo, as diferenças entre os conceitos e o projeto de Aplicação Hipermídia e Sistema Hipermídia. Além disto, é feita uma classificação baseada na interatividade do sistema com os usuários com a finalidade de estabelecer uma ligação natural com a Hipermídia Adaptativa. O Capítulo 2 aborda conceitos relacionados ao AM indutivo e à avaliação do desempenho dos classificadores obtidos para a produção de suposições que um sistema pode utilizar para tomar decisões e que formam o núcleo do motor adaptativo dos sistemas adaptativos em geral. De maneira geral, o resultado da combinação de técnicas de IA com a Hipermídia e a MU é um sistema capaz de adaptar automaticamente as ligações e/ou os conteúdos de uma Aplicação Hipermídia, que é o assunto discutido no Capítulo 3. Neste Capítulo, também são discutidos os métodos e técnicas de Hipermídia Adaptativa para a adaptação de conteúdos e ligações. A descrição detalhada da metodologia utilizada para realizar os experimentos aparece no Capítulo 4. Uma análise dos experimentos e dos resultados obtidos pela aplicação da metodologia adotada é apresentada no Capítulo 5. Por fim, no Capítulo 6 é feita a conclusão desta dissertação e uma apresentação dos trabalhos futuros neste campo.

Capítulo 1

Sistemas Hipermedia

Em geral, os sistemas computadorizados oferecem, além da capacidade de processamento de informações, uma estrutura hierárquica de acesso linear representada por arquivos e diretórios. No entanto, existe uma série de aplicações que envolvem atividades de comunicação e raciocínio que não podem ser representadas e, tampouco, acessadas desta maneira. Para suportar este tipo de aplicações, é necessária uma organização muito mais rica e complexa, caracterizada pela não-linearidade (Nelson, 1965). Esta é a principal característica da Hipermedia, que pode ser interpretada e implementada de diferentes maneiras conforme a utilização e a aplicação pretendida.

A origem e o desenvolvimento da Hipermedia está diretamente relacionada com a possibilidade de utilização do computador em atividades cognitivas complexas como aprender, documentar, controlar versões de manuscritos e auxiliar na escrita de textos em geral (Nelson, 1965) e (Engelbart, 1962, 1968). Neste documento, o termo Hipermedia é utilizado para definir um meio apropriado para a criação e a recuperação de informações de maneira não-linear. Assim sendo, um Sistema Hipermedia (SH) é definido como um dispositivo automatizado capaz de suportar o espaço de informação não-linear por meio de uma interface interativa simples. O computador é o artefato mais adequado para a construção destes sistemas pelo fato de ser uma máquina algorítmica capaz de manipular, representar e processar informações em estruturas de dados adequadas com rapidez, flexibilidade e eficiência.

Este Capítulo aborda a Hipermedia como um referencial para ajudar a compreender as origens e as motivações para o desenvolvimento da Hipermedia Adaptativa (HA). A relação entre Hipermedia e HA é estabelecida naturalmente pelo fato de

que a HA ser o resultado da aplicação de um processo de adaptação automática em um ambiente Hipermedia, como mostra o Capítulo 3. A Seção 1.1 apresenta os conceitos básicos, os principais aspectos e a distinção entre sistema e aplicação de Hipermedia. Seguindo a evolução dos SHs, a Seção 1.2 aborda a plataforma Hipermedia heterogênea mais conhecida e utilizada em todo o mundo: a WWW. Uma relação entre os SHs tradicionais e a WWW é estabelecida com base nos modelos de propósitos geral para Hipermedia descritos na Seção 1.1. Também são mostradas algumas das tecnologias *Web* mais comuns e utilizadas. Muitas destas tecnologias são bastante significativas para adicionar a capacidade adaptativa no ambiente WWW. Finalmente, a Seção 1.3 apresenta uma classificação dos sistemas de software quanto à interatividade, no qual incluem-se os SHs. Nesta classificação, aparecem os sistemas de *software* capazes de adaptar alguns de seus aspectos para os usuários, representando um progresso com relação aos sistemas interativos tradicionais. Neste contexto, a HA aparece como um campo que estuda o desenvolvimento e a aplicação de métodos e técnicas para produção automática de adaptações em ambiente Hipermedia.

1.1 Dimensões da Hipermedia

A Hipermedia, como já foi dito anteriormente, é um meio apropriado para a criação e a recuperação não-linear de informações e, portanto, um SH é um programa de computador capaz de viabilizar esta estrutura de modo simples. Apesar destas definições aparentemente triviais, o estudo e a tentativa de classificar os Sistemas Hipermedia é uma tarefa bastante árdua devido às diferenças existentes entre eles. Conklin (1987) aponta uma série de fatores que contribuem para isto:

1. a diversidade de interpretações da estrutura Hipermedia;
2. a maneira de implementar os mecanismos de apoio à não-linearidade;
3. a diferença na terminologia para designar um mesmo conceito;
4. a variedade na funcionalidade dos sistemas de acordo com o uso que se pretende fazer da estrutura Hipermedia; e
5. a ênfase dada aos mecanismos de suporte ou à interface com o usuário.

Ainda hoje, estes fatores são responsáveis pela dificuldade de entendimento entre pessoas interessadas no assunto. Por isto, é importante apresentar a interpretação que se faz dos conceitos relacionados à Hipermissão antes de aprofundar os estudos e, no caso deste trabalho, de se compreender a maneira como a HA é abordada. Tendo em vista que o SH é o responsável pelo suporte à estrutura não-linear, recorreremos à interpretação de (Conklin, 1987) que descreve-o como um sistema de janelas associadas a objetos interligados em um banco de dados. O autor mostra que as ligações entre esses objetos são representadas visualmente na forma de ícones e internamente na forma de apontadores. Sob este ponto de vista, o SH possui três aspectos fundamentais:

1. um banco de dados não-linear;
2. um filtro de visão; e
3. uma ou mais janelas.

O banco de dados armazena uma rede de fragmentos de informações livremente conectados em um mesmo domínio. Este banco de dados não-linear é o que muitos autores chamam de Aplicação Hipermissão ou Hiperdocumento. O seu conteúdo é apresentado para os usuários por meio de filtros de visão que estruturam as informações em janelas. Desta maneira, o mesmo SH pode armazenar vários Hiperdocumentos ao mesmo tempo, bem como apresentá-los de maneira distinta para usuários diferentes. O sistema deve prover mecanismos para criar e manter a Aplicação Hipermissão, bem como para recuperar e apresentar as informações para os usuários.

O espaço de um SH, tal como foi descrito anteriormente, pode ser compreendido em três dimensões principais e interrelacionadas, segundo (Halasz, 1987): escopo, navegação autoria e domínio das tarefas. O escopo refere-se à capacidade do sistema em relação ao tamanho da Aplicação Hipermissão e da quantidade de usuários potenciais. A dimensão de navegação diz respeito aos mecanismos de apresentação da informação e de exploração da Aplicação Hipermissão; enquanto que a dimensão de autoria enfatiza os mecanismos para a criação e a modificação da base de dados não-linear. O domínio de tarefas, por sua vez, refere-se à influência que a especi-

ificação de requisitos e o projeto de SHs possuem na determinação das características finais do sistema.

Conklin (1987) salienta que há dois componentes essenciais presentes na organização interna de todo SH: os nós e as ligações. Assim, a navegação e a autoria consistem em operações realizadas sobre estes dois elementos básicos. Logo, os aspectos mais significativos e que geram a diversidade entre os SHs são as particularidades de implementação das facilidades de manipulação de nós e ligações e as especificações e decisões de projeto. A este respeito, é importante observar que cada implementação de SH determina uma arquitetura específica que revela um modelo subjacente para as abstrações mais significativas.

Com o objetivo de prover um modelo genérico e flexível para o armazenamento de baixo nível, Campbell & Goodman (1987) apresentam uma máquina hipertexto abstrata baseada em transações denominada HAM (*Hypertext Abstract Model*). Diversos SHs podem ser construídos usando o modelo de armazenamento HAM que aparece ilustrado na Figura 1.1.

Figura 1.1: Máquina abstrata de hipertexto (Campbell & Goodman, 1987).

O modelo Dexter (Halasz & Schwartz, 1990), por sua vez, foi proposto para facilitar a comparação e desenvolver padrões na terminologia, no intercâmbio e na interoperabilidade entre diferentes SHs. O Dexter é um modelo de referência para o desenvolvimento de SHs dividido em três camadas como mostra a Figura 1.2. A compreensão de modelos como HAM e Dexter ajuda a projetar, entender, comparar e definir meios para integrar diferentes implementações de SHs. Isto acontece porque estes modelos capturam as abstrações mais importantes da Hipermídia em diversas estruturas de organização. Existem outros modelos de propósito geral para SH, tais

como *Trellis Hypertext Reference Model* (Furuta & Stotts, 1990), *Extensible Data Model for Hyperdocuments* (De Bra et al., 1992) e Canto (Nicholas & Rosenberg, 1993).

Modelos de dados de propósito geral para SHs preocupam-se com a abstração de conceitos, a organização e a implementação dos mecanismos de acesso ao banco de dados não-linear, das janelas e filtros de visão. Entretanto, eles não são úteis para o projeto de Aplicações Hipermídia que envolvem a captura e a estruturação das relações entre os conceitos e informações de um domínio complexo tornando-o claro e acessível aos usuários. Desta forma, a descrição de um SH é bastante diferente da modelagem de um Hiperdocumento. Para tanto, (Isakowitz et al., 1995) descreve uma metodologia (RMM - *Relationship Managment Methodology*) e um modelo de dados (RMDM - *Relationship Managment Data Model*) para o projeto e o desenvolvimento de Aplicações Hipermídia. Outros modelos foram propostos com este objetivo, como HDM (Schwabe et al., 1993), OOHDM (Schwabe et al., 1996) e HFPM (Olsina, 1998), entre outros.

Figura 1.2: Modelo de referência Dexter.

Com o aparecimento da WWW na Internet, surge uma plataforma distribuída e aberta para a Hipermídia fortemente baseada em protocolos e padrões bem estabelecidos. A próxima Seção mostra as características e o progresso da WWW desde o seu surgimento até os dias atuais, comparando-a com os modelos tradicionais de propósito geral de Hipermídia.

1.2 A World Wide Web

Segundo (Guelich et al., 2001), a Internet é muito mais um padrão de comunicação do que uma rede propriamente dita. Ela herda o protocolo TCP/IP e uma série de serviços desenvolvidos durante o período de existência da ARPAnet¹. Em 1989, Tim Berners-Lee desenvolve o HTTP (*HyperText Transfer Protocol*) — um protocolo em nível de aplicação baseado no TCP/IP — e um programa para acessar dados no servidor ao qual denominou navegador (*browser*). O objetivo original era o de permitir a comunicação hipertexto distribuída utilizando a Internet (CERN). Dois requisitos principais chamam a atenção no HTTP: a possibilidade de utilizá-lo para acessar outros serviços da Internet e o acesso às informações existentes sem a necessidade de convertê-las. Os autores apontam que estas características aliadas ao desenvolvimento de um navegador com interface gráfica — o Mosaic — foram os responsáveis pela grande explosão do serviço da WWW, a ponto de usuários leigos confundirem-na com a própria Internet.

As principais vantagens da WWW residem na facilidade de uso tanto para autoria quanto para navegação. Assim, o pesquisador em Hipermissão libera-se dos detalhes de baixo nível, como prover suporte para a estrutura de nós e ligações, para estudar problemas de mais alto nível, como os mecanismos de busca e recuperação de informações relevantes ou a adaptação automática de conteúdos e/ou ligações para os interesses dos indivíduos. Por esta razão, a WWW contribuiu muito para o progresso em Hipermissão. No entanto, ela apresenta uma série de limitações no que se refere à funcionalidade quando comparada com todas as possibilidades que um SH pode oferecer. (De Troyer & Casteleyn, 2003) mostram que entre as limitações da WWW quanto à funcionalidade estão:

1. a falta de separação entre conteúdo e ligação;
2. a impossibilidade de criar ligações bidirecionais e restrições para ligações;
3. a ausência de tipos de ligações; e
4. a impossibilidade do próprio usuário controlar a estrutura Hipermissão.

¹uma rede heterogênea mantida pelo Departamento de Defesa dos Estados Unidos, que integra outras redes e computadores por meios da padronização de protocolos e serviços para o compartilhamento e troca de informações.

Ainda assim, as vantagens da WWW superam suas deficiências, principalmente pelo desenvolvimento de tecnologias que acrescentam novas capacidades. As particularidades da *Web* implicam a criação de novos modelos ou a modificação de modelos de dados tradicionais existentes para a Hipermedia. Assim é que foram propostos os modelos OOHDM (Schwabe et al., 1996) e o WebML (Bongio et al., 2000). Surgem novas terminologias — página, no lugar de nó — enquanto outras foram reaproveitadas — âncora do modelo Dexter, por exemplo. A WWW representa uma nova geração no desenvolvimento da Hipermedia até mesmo porque muitas limitações dos SHs tradicionais identificadas por Halasz (1987) têm sido amplamente pesquisadas no ambiente WWW, dentre os quais podem ser citadas:

1. mecanismos de busca e consulta;
2. possibilidade de realizar computação sobre a rede Hipermedia;
3. suporte ao trabalho colaborativo; e
4. extensibilidade ou personalização².

Para (Kappel et al., 2003), as aplicações para a *Web* são caracterizadas por três gerações que diferenciam-se pela tecnologia usada e pelos serviços oferecidos. A primeira geração é formada pelas aplicações apenas de leitura através de páginas estáticas para navegação. Em um segundo momento, as aplicações aumentaram de complexidade e passaram a se caracterizar por uma grande quantidade de informações gerenciadas por sistemas de bancos de dados conectados de diferentes maneiras. Além disto, as páginas *Web* passaram a ser geradas dinamicamente em resposta às requisições do usuário ou às mudanças nos dados subjacentes. Estas são as características da segunda geração de aplicações *Web*. Finalmente, os autores consideram o paradigma de acesso a qualquer tempo, em qualquer lugar e em qualquer mídia como sendo o principal aspecto da terceira geração de aplicações WWW. Isto implica no acesso ubíquo aos serviços com o objetivo de apresentar a coisa certa, no momento certo e do modo correto.

Entretanto, os autores chamam a atenção para o fato de que o conceito de ubiquidade em aplicações *Web* restringe-se ao uso das tecnologias existentes para

²Neste trabalho, o termo personalização é usado como sinônimo de adaptação.

acesso às informações não apenas por meio de navegadores, mas também de outros dispositivos móveis comercialmente disponíveis, personalizando os serviços em relação às diferentes propriedades do contexto usado.

A WWW é baseada em uma arquitetura distribuída formada por vários componentes. Uma série de protocolos e padrões da WWW estabelecem a maneira pela qual as informações devem ser representadas e apresentadas aos usuários. Assim, a linguagem HTML (*Hypertext Markup Language*) restringe a estruturação e a apresentação de informações Hipermídia. O protocolo HTTP, por sua vez, estabelece padrões para o intercâmbio de dados em ambiente distribuído, oferecendo a capacidade de acesso não-linear às informações na Internet.

A arquitetura distribuída da *Web* é formada, basicamente, por clientes representados pelos navegadores responsáveis pelo acesso e pela organização da apresentação das informações; e servidores que armazenam a rede de informações codificadas em unidades denominadas páginas. Sob este aspecto, a WWW assemelha-se ao modelo HAM, uma vez que o servidor oferece mecanismos de acesso às informações armazenadas no sistema de arquivos original. Os clientes, por sua vez, oferecem uma interface simples para o usuário visualizar e navegar pelo espaço de informação.

Entre cliente e servidor, pode existir uma série de *proxies* e *gateways*, definindo uma grande variedade de combinações na arquitetura básica da WWW. Por meio da linguagem HTML, o autor de Aplicações Hipermídia baseadas na *Web* insere elementos de marcação junto ao conteúdo de texto para indicar a maneira como deve ser estruturado e apresentado para o usuário. Além disto, pode-se também definir os pontos de ligação de um elemento de conteúdo (gráficos, animações, *applets*, etc) para outras páginas ou elementos, estabelecendo uma ligação entre elas. Assim, o conjunto de recursos (páginas, *scripts*, imagens, vídeos, etc) que formam um *website* correspondem à camada de componentes do modelo Dexter. As páginas *Web* são arquivos em texto onde aparecem elementos de marcação (*tags*) inseridos no seu conteúdo. O conjunto completo das *tags* oferece funções variadas: de estruturação, de apresentação e de ligação. Assim sendo, atuam como métodos da interface de especificação da apresentação e de ancoramento do modelo de referência Dexter.

Devido à natureza aberta da WWW e à formação do W3C³, novos recursos e tec-

³ *World Wide Web Consortium* — um consórcio de empresas, profissionais e pesquisadores para o desenvolvimento da *Web*.

nologias de suporte puderam ser desenvolvidos com bastante rapidez e consistência. As folhas de estilo CSS (*Cascade Style Sheet*) oferecem mecanismos para aumentar o controle sobre a apresentação e a expressividade do conteúdo. Com as tecnologias XML⁴ é possível acrescentar poder de processamento aos documentos.

O desenvolvimento de mecanismos que permitem ao servidor HTTP executar programas, tais como CGI, ASP (*Active Server Pages*) e *servlets*, possibilitam a criação de Aplicações Hipermissão dinâmicas resultantes do processamento de informações que trafegam pela Internet. Além disto, o conhecimento sobre o funcionamento do protocolo HTTP pode ser útil para prover extensões para suportar características específicas e avançadas, na medida em que se tornem necessárias.

Por ser um serviço da Internet, a *Web* pode atingir um número considerável de usuários em todo o mundo e, por esta razão, Brusilovsky (2001) a considera como plataforma padrão para o desenvolvimento e teste de Aplicações Hipermissão. A próxima Seção mostra uma classificação dos SHs (na qual a WWW está incluída) em relação ao aspecto interativo. A partir desta classificação é que surge a HA com mecanismos para a produção automática de adaptações em ambiente Hipermissão.

1.3 Classificação dos Sistemas Hipermissão

A complexidade e a diversidade de SHs implicam uma variedade de classificações possíveis de acordo com o aspecto colocado em evidência. Este trabalho considera a interação com os usuários como critério para classificação. Como mostra a Seção 1.1, o SH possui uma ou mais janelas como interface que apresenta as informações da Aplicação Hipermissão. O usuário interage com o sistema selecionando aquelas ligações de seu interesse que aparecem destacadas na janela de visualização. Em seguida, o conteúdo indicado pela ligação é recuperado e apresentado na mesma janela ou em outra. Desta forma, pode-se incluir os SHs na categoria dos Sistemas Interativos (SI). Ou seja, aqueles sistemas de computadores que reagem à interação com o usuário por meio de diálogos ao longo da sua utilização. Em situação contrária, os Sistemas Não Interativos (SNI) são aqueles que executam sem nenhuma interferência do usuário, como os antigos sistemas em lote (*batch systems*).

⁴São consideradas tecnologias XML todos os recursos relacionadas à linguagem XML, tais como RDF, XPath e XLink, entre outros.

A Figura 1.3 mostra a classificação dos sistemas de *software* quanto à interação com os usuários. No estudo dos SIs, (Kobsa, 1993) discrimina aqueles que permitem a personalização, isto é, o ajuste de algumas de suas características para os interesses ou preferências de cada usuário. Sob este aspecto, o autor identifica dois tipos de sistemas que permitem personalização:

1. sistemas adaptáveis em que o controle da adaptação parte do usuário; e
2. sistemas adaptativos em que o controle e a iniciativa da personalização partem do sistema,

Figura 1.3: Classificação dos sistemas de *software*.

Os Sistemas Adaptativos (SA) devem ser capazes de identificar a necessidade de adaptação e realizá-la automaticamente (ou não) no lugar do usuário (Kobsa, 1993). Convém observar que nem todo SI permite a configuração de alguns de seus aspectos. Ou seja, a adaptação manual ou automática é uma característica adicional que torna os SIs mais ajustados aos interesses e necessidades dos usuários. Compreende-se, naturalmente, que adicionar capacidade adaptativa aos sistemas de uma forma geral exige um esforço maior e prevê o uso de métodos e técnicas específicas e avançadas por parte dos projetistas.

Uma vez que os SHs são classificados como interativos, alguns aspectos da sua interface também podem ser personalizados para os usuário. (Brusilovsky, 1996a, 1996b) identifica dois elementos da interface passíveis de sofrerem adaptações aos interesses e características dos indivíduos: o conteúdo dos nós e as ligações entre eles. Assim, os SHs capazes de realizar adaptações podem ser classificados como sendo de três tipos, de acordo com (De Bra, 1999):

1. adaptável, quando o usuário estabelece o perfil da Aplicação Hiperfídia através de um diálogo ou questionário sobre as suas preferências de apresentação (cores, tipos de mídia e fonte, por exemplo) e experiências anteriores (qualificação, conceitos conhecidos, entre outros);
2. adaptativo, quando o sistema acompanha e registra o comportamento interativo do usuário com o objetivo de deduzir suas preferências e conhecimentos para personalizar o conteúdo e/ou ligações; e
3. dinâmico, onde o comportamento do usuário é monitorado, mas ao invés de modificar conteúdos e ligações predefinidas o sistema gera uma apresentação a partir da combinação de unidades de informações atômicas.

Todo sistema que provê algum tipo de adaptação automática costuma utilizar recursos de IA e de MU que são abordados nos Capítulos 2 e 3, respectivamente. Muitas de suas técnicas são largamente aplicadas em Filtragem de Informação (FI), Recuperação de Informação (RI), Sistemas Tutores Inteligentes (STI) e também em Hiperfídia Adaptativa. As aplicações derivadas destes campos de estudo exercem uma função adaptativa em relação a alguns de seus aspectos e, portanto, podem ser classificadas como sistemas de *software* adaptativos (Figura 1.4). A diferença entre estas aplicações reside, basicamente, no alvo da adaptação (conteúdos e/ou ligações), no conhecimento necessário ao sistema (do domínio, do usuário e/ou da adaptação) e na tecnologia utilizada para gerar a personalização, isto é, na forma de representação do modelo, no procedimento usado para gerar as deduções e nas técnicas para produzir a adaptação (dedução, indução, analogias).

Figura 1.4: Exemplos de Sistemas Adaptativos.

1.4 Considerações Finais

Este Capítulo mostra que os SHs são ferramentas de *software* projetadas para auxiliar o ser humano no registro de fragmentos de informações livremente interconectados, formando uma estrutura complexa que permite a recuperação destas informações de maneira natural, simples e intuitiva. De modo geral, estes sistemas podem ser usados de duas maneiras distintas e complementares: para autoria ou para navegação. Como ferramenta de autoria, os SHs podem abrigar diferentes Aplicações Hipermedia. Como consequência, os modelos de propósito geral para SHs consideram os aspectos voltados à abstração dos conceitos fundamentais e à implementação dos mecanismos de suporte à manipulação de nós e ligações, enquanto os modelos para Aplicações Hipermedia pretendem capturar a estrutura de domínios complexos. Este Capítulo considerou todos estes aspectos, procurando relacioná-los com a WWW que tornou-se a plataforma Hipermedia padrão por ser uma arquitetura aberta, distribuída e de amplo alcance, mas que apresenta características muito peculiares em relação à verdadeira capacidade dos SHs.

No próximo Capítulo são discutidas as técnicas de IA empregadas neste trabalho para fazer com que o computador produza, de maneira automática, suposições que sejam potencialmente de interesse do indivíduo que esteja utilizando a Hipermedia para prover adaptações de ligações.

Capítulo 2

Aprendizado de Máquina

O computador é considerado o artefato com as melhores chances de exibir algum tipo de inteligência por ser uma máquina algorítmica automática. O estudo dos algoritmos concentra grande parte do aspecto teórico em Computação (Brookshear, 2001). É através dos algoritmos que o ser humano determina o comportamento do computador por meio de um conjunto finito e específico de instruções na execução de uma certa tarefa. Assim, um algoritmo codifica a inteligência necessária para resolver um problema. O desafio da IA está em fazer com que o computador pense ou se comporte como humanos ou de forma racional (Russel & Norvig, 1995).

O Aprendizado de Máquina é uma sub-área da IA que tem se mostrado de grande utilidade, tendo em vista a importância do aprendizado para o comportamento inteligente, segundo Monard et al. (1997). Mitchell (1997) refere-se ao aprendizado como sendo a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Desta maneira, o AM consiste em um programa de computador que realiza um conjunto de atividades capaz de melhorar seu desempenho a cada experiência, em um processo de aquisição automática de conhecimento a partir de novos dados.

Nesta dissertação, será focalizada a aplicação de técnicas de AM com a utilização de raciocínio indutivo para gerar classificadores que terão seu desempenho comparado para determinar qual a melhor técnica para a solução do problema proposto. A Seção 2.1 mostra os principais conceitos relacionados com AM. Os diferentes paradigmas de AM são discutidos na Seção 2.2. Finalmente, a Seção 2.3 mostra como comparar os resultados dos experimentos obtidos com a aplicação de diversos

algoritmos de AM para determinar aquele que possui o melhor desempenho.

2.1 Conceitos de AM Indutivo

O objetivo do AM consiste em modelar computacionalmente o processo de aprendizado humano. Isto é, a aquisição e a representação efetiva de conhecimentos, o desenvolvimento de habilidades e a descoberta de novos fatos e teorias por meio da observação e experimentação. Existem várias maneiras para aprender um novo conceito, determinando estratégias diferentes que são classificadas em aprendizado por hábito, por instrução, por dedução, por indução ou por analogia (Monard et al., 1997).

A estratégia de AM indutivo implica na obtenção de conclusões gerais a partir de fatos particulares. A indução é uma forma de raciocínio que parte do específico para o geral. O problema é que as hipóteses geradas por inferência indutiva nem sempre preservam a verdade. Isto acontece porque o número de observações pode ter sido insuficiente ou os dados relevantes para chegar à conclusão foram mal escolhidos. Ainda assim, o raciocínio indutivo é fundamental e mostra-se útil principalmente para prever eventos futuros, fazer generalizações sobre características específicas, bem como encontrar agrupamentos naturais a partir de características semelhantes. Desta maneira, a inferência indutiva é um dos principais métodos para derivar conhecimento novo e prever eventos futuros (Baranauskas & Monard, 2000).

De modo geral, o aprendizado indutivo divide-se em supervisionado e não-supervisionado, como mostra a Figura 2.1. A diferença está no modo como acontece o aprendizado. No primeiro, cada exemplo¹ é rotulado para uma classe conhecida por meio de um agente externo. No caso do aprendizado não supervisionado, os exemplos não são rotulados por nenhum especialista. O conjunto de padrões é analisado para tentar encontrar *clusters*². Monard et al. (1997) utilizam os termos aprendizado por exemplos e aprendizado por observação e descoberta no lugar de aprendizado supervisionado e não supervisionado, respectivamente.

O AM indutivo com modo de aprendizado supervisionado acontece a partir da

¹outras denominações também costumam ser utilizadas na literatura de IA, como padrões, casos ou instâncias.

²agrupamentos de exemplos baseados em características ou atributos que os definem.

Figura 2.1: Tipos de Aprendizado Indutivo.

análise sobre um conjunto de exemplos (base de dados) fornecido por um processo externo ao sistema. O objetivo do indutor (ou algoritmo de aprendizado indutivo) é encontrar a hipótese de indução - uma função $h()$ que aproxima o conjunto de exemplos - com a menor taxa de erro possível (Russel & Norvig, 1995). O conceito induzido pelo algoritmo de aprendizado - também chamado de classificador - deve encontrar corretamente a classe ou o valor de saída de novas instâncias ainda não consideradas. Este processo pode ser uma classificação, quando os rótulos (classes) possuem valores discretos, ou uma regressão, quando são usados valores contínuos para os rótulos.

Cada exemplo é descrito por uma tupla contendo o valor de características descritas por atributos. No caso do aprendizado supervisionado, adiciona-se a cada exemplo um atributo que é o rótulo da classe associada. O atributo classe serve para descrever o fenômeno de interesse. Representa a meta que se deseja aprender e sobre a qual se pode gerar previsões a respeito. Existem, em geral, dois tipos básicos de atributos: nominal, quando não há nenhuma ordem de importância entre os valores; e contínuo, quando os valores possuem uma ordem linear. Quando o aprendizado é supervisionado, a base de dados é dividida em dois subconjuntos disjuntos: o conjunto de treinamento, usado para o aprendizado do conceito; e o conjunto de teste, para medir o grau de efetividade do conceito aprendido. Muitas vezes, os exemplos contêm ruído, isto é, erros na sua descrição que podem ser expressos por ausência de valores nos atributos ou pela classificação de um padrão com rótulo incorreto.

O algoritmo de aprendizado pode chegar a diferentes hipóteses para uma mesma função ou problema. Neste caso, a escolha da melhor aproximação é determinada por um *bias* de aprendizado que representa o conhecimento adicional necessário para

determinar a alternativa mais adequada (Baranaukas & Monard, 2000). Quando a hipótese for muito específica e ajustar-se excessivamente a um conjunto de treinamento, diz-se que ocorreu um *overfitting*³. Quando a hipótese ajusta-se muito pouco ao conjunto de treinamento, ocorre um *underfitting*⁴. Quando os parâmetros de um algoritmo de aprendizado são ajustados em excesso para otimizar seu desempenho em todos os exemplos disponíveis, diz-se que ocorre um *overtuning*⁵.

O algoritmo de aprendizado indutivo pode ser incremental quando o conjunto de treinamento não está totalmente disponível desde o início do processo de aprendizagem de um conceito. Desta forma, a hipótese é atualizada cada vez que um novo padrão é acrescentado ao conjunto de exemplos. O algoritmo é não incremental quando o aprendizado efetiva-se por meio de um conjunto de treinamento integralmente disponível e estável, isto é, não deverão acontecer mudanças nos dados disponíveis.

O desempenho de um algoritmo de aprendizado é expresso pela taxa de erro (2.1) de um classificador. É obtida quando se compara a diferença entre a classe verdadeira de um exemplo ($h(x)$) e a classe atribuída pelo classificador (y). Quando a saída do indutor classifica corretamente, isto é, quando as classes coincidem, o operador $\|$ retorna 0, caso contrário, retorna 1. A precisão do classificador é o complemento da taxa de erro (2.2), onde n é o número total de padrões que existem no conjunto de exemplos.

$$\text{err}(h) = \frac{1}{n} \sum_{i=1}^n \|y_i = h(x_i)\| \quad (2.1)$$

$$\text{acc}(h) = 1 - \text{err}(h) \quad (2.2)$$

Em problemas de regressão, o erro da hipótese de um classificador é estimado pela distância entre o valor real e o valor atribuído pela hipótese induzida. Duas medidas para o erro da hipótese são normalmente usadas: o erro médio quadrado (2.3) e a distância absoluta média (2.4).

³sobreajuste do algoritmo para o conjunto de dados.

⁴subajuste do algoritmos em relação aos dados.

⁵afinação excessiva dos parâmetros para os dados.

$$\text{mse-err}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h(x_i))^2 \quad (2.3)$$

$$\text{mad-err}(h) = \frac{1}{n} \sum_{i=1}^n |y_i - h(x_i)| \quad (2.4)$$

A distribuição da classe (2.5) representa a proporção de exemplos em cada classe C . A classe com maior distribuição é chamada de majoritária ou prevalente, ao passo que a classe com menor distribuição é denominada de minoritária. Com base na distribuição é possível calcular o erro majoritário (2.6), isto é, o limite máximo para o erro de um classificador T .

$$\text{distr}(C_j) = \frac{1}{n} \sum_{i=1}^n \|y_i = C_j\| \quad (2.5)$$

$$\text{maj-err}(T) = 1 - \max_{i=1, \dots, m} \text{distr}(C_i) \quad (2.6)$$

A classificação ou predição correta para cada classe sobre um conjunto de exemplos pode ser representada em uma matriz de confusão. Nela, aparece o número de classificações corretas e o número de classificações preditas para cada classe. A matriz é formada por elementos $M(C_i, C_j)$, onde $1 \leq i, j \leq k$ e k representa a quantidade de classes diferentes no conjunto de exemplos. Cada elemento da matriz é calculado pelo número de exemplos que realmente pertencem à classe C_i , mas que foram classificados como C_j (2.7).

$$M(C_i, C_j) = \sum_{\{(x,y) \in E : y=C_j\}} \|h(x) = C_i\| \quad (2.7)$$

A diagonal principal da matriz de confusão indica o número de acertos para cada classe. Cada um dos outros elementos de $M(C_i, C_j)$, onde $i \neq j$, representam o erro para um tipo de classificação em particular. A partir da matriz de confusão é possível derivar uma série de erros. Considerando um problema com duas classes (positivo e negativo), os erros possíveis são falso positivo e falso negativo. As classificações corretas podem ser verdadeiro positivo e verdadeiro negativo. Nestas condições, a taxa de erro da classe é definida pela relação existente entre o número de exemplos

classificados como falso negativos e a soma do número de exemplos rotulados pelo classificador como verdadeiro positivo e falso negativo.

2.2 Paradigmas de AM

Para resolver um problema utilizando AM, geralmente seleciona-se um conjunto de algoritmos de aprendizado e conduz-se uma série de experimentos para comparar seus desempenhos. Sistemas de AM podem, ainda, ser de dois tipos. Nos sistemas de caixa-preta, a representação interna não é facilmente interpretada pelo ser humano, ou então não existem meios de explicar de que maneira uma conclusão foi obtida. Assim acontece com as Redes Neurais Artificiais (RNA), por exemplo. Por outro lado, os sistemas orientados a conhecimento fazem uso de uma estrutura simbólica compreensível ao ser humano. A Rede Semântica é um exemplo típico de representação deste tipo de sistema. Os parágrafos a seguir descrevem as características dos paradigmas principais de AM.

Paradigma Simbólico

O aprendizado acontece por representação simbólica de conceitos. Para tanto, são utilizadas representações baseadas em Lógica, em Redes Semânticas, em Regras de Produção ou em Árvores de Decisão. Esses sistemas aprendem descrições simbólicas que representam um conhecimento em alto nível, fazendo fortes suposições estruturais sobre os conceitos a serem adquiridos. Os algoritmos de AM simbólico empregam, em geral, o método de particionamento recursivo para indução em árvores de decisão. Os sistemas ID3 (Quinlan, 1986) e C4.5 (Quinlan, 1994) são representantes típicos deste paradigma.

Paradigma Estatístico

Faz uso de medidas estatísticas para encontrar uma boa aproximação do conceito induzido como método de classificação. As técnicas estatísticas costumam ser empregadas em tarefas em que todos os atributos possuem valores contínuos ou ordinais. Muitos destes algoritmos são paramétricos, isto é, procuram por valores apropriados para os parâmetros a partir dos dados de treinamento. O classificador Bayesiano tem se mostrado bastante efetivo (Friedman et al., 1997). Este é um representante comum da abordagem de aprendizado de máquina estatístico.

Paradigma Baseado em Instâncias

Uma forma bastante comum de classificação consiste em lembrar de casos semelhantes cuja classe seja conhecida e assumir que o novo caso possui a mesma classe. Ou seja, AM baseado em instâncias classifica exemplos com base em outros semelhante, tal como os algoritmos k-vizinhos mais próximos e Raciocínio Baseado em Casos (RBC). Algumas características importantes devem ser consideradas com relação a esta abordagem, tais como quais casos devem ser memorizados e a métrica que será usada para aferir a similaridade entre os casos.

Paradigma Conexionista

As Redes Neurais são sistemas matemáticos relativamente simples inspiradas no modelo biológico do cérebro humano. O conhecimento fica armazenado nas conexões estabelecidas entre os neurônios da rede e que são ajustados durante o processo de treinamento. Esta abordagem tem sido usada com sucesso para resolver problemas que exigem intenso processamento sensorial, como o reconhecimento de voz e a visão. Diferentes modelo de RNAs têm sido usados com sucesso, tais como as redes MLP - *Multi Layer Perceptron* - (Rumelhart et al., 1986) e ART - *Adaptive Resonance Theory* - (Carpenter & Grossberg, 1987).

Paradigma Evolutivo

Fundamenta-se na genética e na seleção natural, onde os indivíduos ou soluções de uma população competem de maneira que os mais fortes proliferam, produzindo variações de si mesmo. Os indivíduos com desempenho mais fraco são descartados enquanto os mais fortes proliferam. Para tanto, alguns operadores são aplicados sobre a população, gerando novos indivíduos a cada geração. Os operadores mais utilizados são: seleção, cruzamento, mutação e inversão. Costumam ser empregados para resolver problemas de otimização e de AM utilizando simulações e regras probabilísticas.

2.3 Avaliação de Algoritmos de AM

Não é possível determinar um único algoritmo de AM que apresente o melhor desempenho em todos os problemas. Por esta razão faz-se necessário empregar algum critério que permita avaliar e comparar estes algoritmos. A metodologia mais comum

usada para atingir este objetivo é a amostragem (*resampling*). Por esta metodologia, o desempenho do classificador induzido é estimado usando um conjunto de exemplos (uma distribuição) em um certo domínio. Seja uma distribuição D e um conjunto de amostras extraídas de D , obtém-se a estimativa da precisão e do erro treinando-se o indutor com um subconjunto desta amostra e testando seu desempenho em amostras fora do subconjunto usado para treinamento. Para que a medida seja verdadeira, a divisão das amostras deve ser aleatória. Existem vários métodos baseado em amostragem que podem ser aplicados para avaliar o desempenho de algoritmos de AM (Baranauskas & Monard, 2000):

1. *holdout*: divide o conjunto de exemplos em porcentagens fixas para o treinamento e o teste. Costuma-se empregar 2/3 do conjunto de exemplos para o treinamento e 1/3 para os testes. Em geral, considera-se o conjunto de exemplos para treinamento maior que 1/2;
2. *cross-validation*: em *k-fold cross validation* os exemplos são aleatoriamente divididos em k partições disjuntas de tamanho n/k , aproximadamente, onde n é o número total de exemplos e k é o número de partições. $(k - 1)$ partições são utilizadas para o treinamento e a partição restante é usada para teste. O processo é repetido k vezes e o erro é a média dos erros calculados em cada k partição; e
3. *leave-one-out*: é um caso especial de *cross-validation* aplicado para pequenos conjuntos de exemplos. São utilizados $(k - 1)$ partições para o treinamento e 1 partição restante para teste. Repete-se o processo testando um exemplo diferente a cada iteração. O erro é obtido pela média dos erros obtidos em cada iteração.

Em geral, para um algoritmo de aprendizado A e um conjunto de exemplos que foi particionado, calcula-se a média (2.8), a variância (2.9) e o desvio-padrão (2.10). Para tanto, considera-se o erro do classificador h_i ($err(h_i)$) de cada partição.

$$\text{mean}(A) = \frac{1}{k} \sum_{i=1}^k \text{err}(h_i) \quad (2.8)$$

$$\text{var}(A) = \frac{1}{k} \left[\frac{1}{k-1} \sum_{i=1}^k (\text{err}(h_i) - \text{mean}(A))^2 \right] \quad (2.9)$$

$$\text{sd}(A) = \sqrt{\text{var}(A)} \quad (2.10)$$

O erro de um algoritmo de AM é expresso por sua média seguida pelo símbolo $\pm$ e pelo seu desvio padrão ($8,30 \pm 1,00$, por exemplo). O desvio padrão representa a robustez do algoritmo. Isto é, as variações nos erros provenientes de classificadores usando diferentes conjunto de treinamento e teste. Quando essa variação é muito grande de um experimento para outro, pode-se dizer que o indutor não é robusto a mudanças no conjunto de treinamento obtido de uma mesma distribuição.

Para determinar, com grau de confiança de 95%, qual algoritmo de AM apresenta melhor desempenho (em taxa de erro de classificação), dever-se assumir o caso geral para determinar se a diferença entre os dois algoritmos é significativa ou não. Suponha que A_s é o algoritmo padrão e A_p o algoritmo proposto. A comparação é efetuada combinando-se a média e o desvio padrão de ambos os algoritmos. Para tanto, os indutores I_s e I_p devem ser comparados em um mesmo domínio, combinando-se a média e o desvio padrão sobre a taxa de erro de cada classificados obtidos pela aplicação de um dos métodos de amostragem anterior. Em seguida, calcula-se a diferença absoluta em desvio padrão. Conforme mostram Baranauskas & Monard (2000), quando $\text{ad}(A_s - A_p) > 0$, então o algoritmo A_p supera A_s e se $\text{ad}(A_s - A_p) \geq 2$, A_p supera A_s com grau de confiança de 95%. Quando $\text{ad}(A_s - A_p) \leq 0$, então o algoritmo A_s supera A_p e se $\text{ad}(A_s - A_p) \leq -2$, A_s supera A_p com grau de confiança de 95%. O mesmo raciocínio pode ser aplicado para obter-se um grau de confiança de 99%, variando-se apenas os valores das fórmulas fornecidas.

$$\text{mean}(A_s - A_p) = \text{mean}(A_s) - \text{mean}(A_p) \quad (2.11)$$

$$\text{sd}(A_s - A_p) = \frac{\sqrt{\text{sd}(A_s)^2 + \text{sd}(A_p)^2}}{2} \quad (2.12)$$

$$\text{ad}(A_s - A_p) = \frac{\text{mean}(A_s - A_p)}{\text{sd}(A_s - A_p)} \quad (2.13)$$

2.4 Técnicas de AM Utilizadas

Esta Seção apresenta uma breve descrição das técnicas de AM utilizadas nos experimentos realizados neste trabalho. Cada uma delas é representativa de um determinado paradigma de AM. Assim, o coeficiente de correlação de Pearson representa o paradigma de AM baseado em instâncias, o classificador Bayesiano e a Máquina de Suporte Vetorias correspondem ao paradigma estatístico e o Perceptron multica-madas participa do paradigma de AM conexionista.

2.4.1 Coeficiente de Correlação de Pearson

A técnica de correlação de Pearson é comum em pesquisas de Filtragem Colabo-rativa (Breese et al., 1998). O objetivo destes sistemas é prever a utilidade de um item a ser recomendado para um usuário corrente⁶ baseado na semelhança com outros usuários que formam uma base de dados colaborativa⁷ (Breese et al. 1998) e (Soboroff & Nicholas, 2000).

A predição ($p_{a,j}$) do interesse do usuário corrente para a página j é calculado pela soma ponderada dos pesos das classificações de outros usuários dada pela Equação 2.14, onde n é o número de usuários no banco de dados colaborativo com pesos diferentes de zero. O valor de k reflete o fator de normalização tal que o valor absoluto dos pesos some uma unidade. As variáveis v_a e v_i correspondem, respectivamente, à média dos votos do usuário corrente a e do i -ésimo usuário da base colaborativa calculadas como mostra a Equação 2.15. Nesta fórmula, v_j indica o grau de interesse do usuário na j -ésima característica.

$$p_{a,j} = v_a + k \sum_{i=1}^n w(a,i)(v_{i,j} - v_i) \quad (2.14)$$

$$v = \frac{1}{|I_j|} \sum_{j \in I_i} v_j \quad (2.15)$$

O peso $w(a, i)$ reflete distância, correlação ou similaridade entre cada usuário i e

⁶aquele que está utilizando o sistema no presente momento.

⁷aqueles que utilizaram o sistema previamente classificando de maneira implícita ou explícita os itens existentes.

o usuário ativo. O interesse do usuário ativo (a) e do i -ésimo usuário da base colaborativa por uma página j é expresso por $v_{a,j}$ e $v_{i,j}$, respectivamente. Os algoritmos de Filtragem Colaborativa baseados em memória, que corresponde ao paradigma baseado em instâncias de AM, distinguem-se entre si em termos dos detalhes de como os pesos são calculados. O coeficiente de correlação de Pearson entre os usuários a e i é calculado como mostra a Equação 2.16.

$$w(a,i) = \frac{\sum_j (v_{a,j} - v_a)(v_{i,j} - v_i)}{\sqrt{\sum_j (v_{a,j} - v_a)^2 \sum_j (v_{i,j} - v_i)^2}} \quad (2.16)$$

Para comparar o algoritmos de correlação de Pearson com outras abordagens de AM é necessário calcular a taxa de erro em relação a predição de um determinado item (página *Web*, no caso). Para tanto, considerou-se um limiar para a predição $p_{a,j}$ que determina quando a página j deve ser recomendada para um usuário a .

2.4.2 Classificador Bayesiano

Em tarefas de reconhecimento de padrões é um procedimento freqüente utilizar o classificador construído a partir de instâncias previamente rotuladas (o conjunto de treinamento) para atribuir uma classe a instâncias descritas por um conjunto de atributos (Friedman et al., 1997). Um classificador Bayesiano aprende utilizando probabilidade condicional de cada atributo dado o rótulo da classe. A classificação é realizada pela aplicação da regra de Bayes para calcular a probabilidade da classe C dado um exemplo particular, predizendo a classe com maior probabilidade posterior, como mostra a Equação 2.17.

$$\Pr(C = c, v_1, \dots, v_n) = \Pr(C = c) \prod_{i=1}^n \Pr(v_i | C = c) \quad (2.17)$$

A essência das abordagens baseadas no Teorema de Bayes consiste na utilização de uma regra matemática que modifica o grau de crença em um evento A condicionado a um conhecimento B . Desta maneira, o problema da incerteza na abordagem Bayesiana é tratada como probabilidades condicionais. A idéia é combinar novos dados com o conhecimento já existente. A regra de Bayes considera a probabilidade condicional sem considerar a probabilidade conjunta de $P(A,B)$. Para tanto,

a fórmula para probabilidade condicional (Equação 2.18) é reescrita utilizando a Equação 2.20 obtida por simetria sobre a Equação 2.19.

$$P(A|B) = \frac{P(A, B)}{P(B)} \quad (2.18)$$

$$P(A, B) = P(A|B)P(B) \quad (2.19)$$

$$P(A, B) = P(B|A)P(A) \quad (2.20)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.21)$$

O resultado é a regra de Bayes (Equação 2.21), onde $P(A|B)$ é a probabilidade posterior de que o modelo A ocorra se a evidência B for verdadeira, calculada em função da probabilidade anterior $P(A)$ do modelo A na ausência da evidência B, da probabilidade $P(B|A)$ de que a evidência B foi produzida dado o modelo A e da probabilidade da evidência $P(B)$. A regra de Bayes permite que $P(A|B)$ seja calculada em termos de $P(B|A)$. Costuma ser empregada em situações onde o cálculo de $P(A|B)$ é difícil de se obter diretamente, ao passo que as informações sobre $P(B|A)$ podem ser facilmente obtidas.

2.4.3 Perceptron Multicamadas

Redes multicamadas do tipo Perceptron (*Multi Layer Perceptron* - MLP) são modelos de RNAs com uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. Seu poder computacional excede a capacidade das redes simples sem camada intermediária como Perceptron e Adaline, podendo tratar dados que não são linearmente separáveis (Braga et al., 2000).

As unidades intermediárias atuam como detectores de características. A determinação da quantidade de camadas intermediárias depende das características dos dados e do problema. Em geral, uma camada intermediária aproxima funções contínuas, enquanto duas camadas são usadas para aproximar qualquer função matemática. A utilização de duas ou mais camadas intermediárias pode facilitar o

treinamento da rede, mas um grande número não é recomendado por propagar o erro durante o treinamento. A quantidade de nodos em cada camada precisa ser obtido empiricamente e depende da distribuição dos padrões de treinamento e validação da rede.

Em uma rede multicamadas, o processamento realizado por um nodo é definido pela combinação dos processamentos realizados pelos nodos da camada anterior que estão conectados a ele. À medida em que avança em direção à camada de saída, as funções implementadas tornam-se cada vez mais complexas. O treinamento de redes multicamadas pode ser feito de duas maneiras, segundo Braga et al. (2000). A primeira consiste em dividir a rede em subredes para cada camada com treinamento independente. A segunda consiste em treinar a rede completa de uma única vez usando uma função de ativação não-linear contínua, diferenciável e preferencialmente não-decrescente. Em geral é utilizada a função sigmoidal, como mostra a Equação 2.22.

$$f = \frac{1}{1 + e^{-x}} \quad (2.22)$$

O *backpropagation* (Rumelhart & McClelland, 1986) é o algoritmo de treinamento supervisionado mais conhecido para as redes MLP (Rumelhart et al., 1986). Para tanto, utiliza pares de entrada associados com a saída desejada para ajustar os pesos da rede por um mecanismo de adaptação por correção de erros em duas fases (*forward* e *backward*). O *backpropagation* baseia-se na regra delta generalizada, recorrendo ao método do gradiente para ajustar os pesos das conexões entre os nodos. A função de custo a ser minimizada pelo *backpropagation* é definida pela soma dos erros quadráticos, isto é, o erro total da rede dado pela Equação 2.23. Onde, n é o número total de padrões, k é o número de nodos de saída, d_i é a i -ésima saída desejada e y_i é a i -ésima saída obtida pela rede. A função de ativação para cada unidade j da rede é definida pelas Equações 2.24 e 2.25.

$$E = \frac{1}{2} \sum_n \sum_{i=1}^k (d_i - y_i)^2 \quad (2.23)$$

$$y_j = f(\text{net}_j) \quad (2.24)$$

$$\text{net}_j = \sum_{i=1}^c x_i w_{ji} \quad (2.25)$$

A Equação 2.26 mostra como é calculada a variação dos pesos das conexões de acordo com o gradiente descendente do erro em relação ao seu peso.

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} = -\frac{\partial E}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ji}} \quad (2.26)$$

$$\delta_j = (d_j - y_j) f(\text{net}_j) \quad (2.27)$$

$$\delta_j = f(\text{net}_j) \sum_l \partial_l w_{lj} \quad (2.28)$$

O erro (δ_j) de uma unidade j é fornecido pela primeira derivada da Equação 2.26. Se o nodo pertencer à camada de saída, o erro é dado pela Equação 2.27. Já quando o nodo pertence a uma camada intermediária, o erro passa a ser definido pela Equação 2.28.

2.4.4 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (*Support Vector Machines* - SVM) constitui uma técnica de AM baseada na Teoria de Aprendizado Estatístico formulada no final da década de 60 por Vapnik e Chervonenkis (Vapnik & Chervonenkis, 1968). Mesmo sendo de aplicação bastante recente (a partir do início dos anos 90), esta técnica desperta grande interesse devido à obtenção de resultados que superam os obtidos por outras abordagens de AM na solução de problemas como a categorização de textos (Dumais, 1998; Hearst et al., 1998) e a detecção de faces (Osuna, 1998; Hearst et al., 1998).

O princípio desta técnica está em encontrar um hiperplano ótimo que separa membros e não-membros de uma classe em um espaço abstrato denominado espaço de características (Vert, 2001). Neste espaço, as classes presentes no conjunto de treinamento se tornam linearmente separáveis e o hiperplano ótimo é definido como aquele para o qual a margem de separação entre as mesmas é maximizada.

Considere f um classificador que mapeia um conjunto de exemplos no espaço de classes e F o conjunto de todos os classificadores que um algoritmo de AM pode gerar. Para escolher o classificador f' que produza o menor erro durante o treinamento, a técnica SVM procura por aquele que produz o menor risco empírico (Equação 2.29), onde $c()$ é uma função de custo que relaciona a classificação $f(x_i)$ com a saída y_i desejada. O processo de busca por f' é denominado minimização do risco empírico, conforme (Lorena & Carvalho, 2003).

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} c(f(x_i), y_i) \quad (2.29)$$

O risco funcional (Equação 2.30) é usado para quantificar a capacidade de generalização, isto é, a probabilidade de que um classificador f' cometa erro na classificação de um novo exemplo gerado seguindo uma distribuição de probabilidade P .

$$R(f) = \int \frac{1}{2} c(f(x), y) dP(x, y) \quad (2.30)$$

A longo do processo de treinamento, o risco empírico pode ser facilmente calculado, mas o risco funcional depende de uma distribuição de probabilidade desconhecida. Por isto, Lorena & Carvalho (2003) mostram que a maioria dos algoritmos de aprendizado SVM busca a minimização do risco empírico com o objetivo de obter um risco funcional menor. Para determinar os limites no risco funcional, utiliza-se o conceito de dimensão Vapnik-Chervonenkis (VC).

Uma propriedade importante das SVMs está na utilização de Kernels. Os Kernels são produtos internos das coordenadas de dois vetores, e são utilizados para construção do hiperplano ótimo no espaço de características sem ser necessário considerar a forma explícita deste (Haykin, 1999), que geralmente é bastante complexa.

2.5 Considerações Finais

Este Capítulo aborda as técnicas de IA como ferramentas úteis para a obtenção de conclusões ou suposições que possam ser utilizadas pelo sistema computadorizado para tomar decisões. Foram apresentados três tipos principais de raciocínio: a dedução, a indução e a analogia. Neste trabalho, avalia-se a aplicação de técnicas

de raciocínio indutivo. O AM indutivo tem sido amplamente utilizado em diversas aplicações por sua capacidade de generalização e predição. Brusilovsky (2001) considera AM importante para o desenvolvimento da HA e que, portanto, precisam ter sua utilização investigada. Para tanto, foi feito, neste Capítulo, um estudo sobre o AM supervisionado e das técnicas que foram utilizadas para a realização dos experimentos, a saber: correlação de Pearson, classificador Bayesiano, MLP e SVM.

O Capítulo seguinte mostra como integrar as técnicas de IA e MU em um processo de adaptação em ambiente Hipermídia. Serão discutidos os fatores que levaram ao desenvolvimento da HA, os aspectos que podem ser personalizados e os modelos específicos para sistemas e aplicações de HA que suportam a adaptação automática.

Capítulo 3

Sistemas Hipermissão Adaptativos

Nos últimos anos, considerável esforço tem sido empregado no progresso da HA. As pesquisas neste campo começaram no início de 1990 e têm atraído pesquisadores de várias outras comunidades (Brusilovsky, 2001). O autor mostra, ainda, que trabalhos independentes procuravam explorar diferentes maneiras de personalizar a saída e o comportamento de SHs à usuários individuais. A partir de 1996, Brusilovsky & Maybury (2002) observam que um aumento de interesse no uso da WWW como plataforma para o desenvolvimento de Hiperdocumentos adaptativos deve-se, especialmente, à diversidade de usuários que podem acessá-la. Para tanto, os autores sugerem uso do termo *Web Adaptativa* para designar os Sistemas Hipermissão Adaptativos (SHA) baseados na WWW.

Partindo de uma perspectiva diferentes de pesquisadores como Peter Brusilovsky, Paul De Bra e Kobsa, (Kappel et al., 2003) emprega o termo customização (*customization*) para referir-se à adaptação dos serviços de uma aplicação em relação às propriedades do seu contexto¹. Assim, o conceito de customização seria mais amplo, abrangendo, inclusive, a personalização que consiste no ajuste de certos aspectos do sistema para as propriedade do usuário (suas características, objetivos, interesses ou preferências individuais). Os autores mostram, ainda, que são três as gerações de SHAs para a WWW:

1. primeira geração, que explorou de métodos e técnicas de apresentação e suporte navegacional, bem como a modelagem do conhecimento e dos objetivos do

¹um contexto pode abranger informações referentes à localização, ao tempo, aos dispositivos usados, à rede, aos usuários, à aplicação, entre outros.

usuário;

2. segunda geração, contribuiu com investigações baseadas na modelagem dos interesses dos usuários, provenientes das pesquisas em Filtragem e Recuperação de Informações; e
3. terceira geração, que procura estender a adaptação não só para os indivíduos, como também para outras propriedades do contexto em que o sistema é acessado.

Este Capítulo está organizado como se segue. A motivação para o desenvolvimento da HA é apresentada na Seção 3.1. Nela, aparecem também as definições para SHAs e as suas principais áreas de aplicação. Pelo fato da HA aumentar a complexidade dos sistemas e aplicações Hipermedia, a Seção 3.2 mostra modelos para sistemas e aplicações de HA, facilitando a sua compreensão e implementação. Além disto, estes modelos também facultam a padronização, o intercâmbio e a comparação entre SHAs distintos. Em geral, são realizadas modificações ou extensões nos modelos existentes para a Hipermedia tradicional de modo a suportarem a capacidade de adaptação adicional. Por fim, a Seção 3.3 mostra como as informações a respeito do usuário podem ser colhidas e armazenadas em um modelo de usuário que permita ao sistema gerar suposições e aplicar adaptações automáticas.

3.1 Motivação para Hipermedia Adaptativa

Os SHs tradicionais que apresentam conteúdos e ligações estáticas não fazem mais do que apresentar o mesmo conteúdo de nós e conjunto de ligações para todos os usuários. Não consideram nem as características individuais e nem o caminho percorrido pelo usuário até o nó corrente, como observam (Calvi & De Bra, 1998), (De Bra et al., 2000) e (Brusilovsky, 2001). Nestas situações, em que o conteúdo dos nós é sempre o mesmo, as informações ou são redundantes ou não podem ser totalmente compreendidas porque não há nenhuma garantia de que o usuário tenha acessado um conteúdo que seja prévio e necessário.

A abordagem tradicional para a produção de Aplicações Hipermedia assume que o autor cria caminhos bem definidos para que os usuários possam navegar, estabelecendo um mapa conceitual sobre um determinado domínio. Neste processo, Isakovitz

et al. (1993) sugerem que apenas as ligações mais significativas sejam consideradas. Além disto, os autores de Aplicações Hipermídia não são capazes de antecipar todos os caminhos para cada indivíduo, segundo (Bodner & Chignell, 1999). Estabelecer ligações entre todos os nós é impraticável para domínios complexos, levando o usuário (e o autor) facilmente à condição de desorientação no espaço de navegação. Assim, os usuários são obrigados a navegar por trilhas pré-estabelecidas e a entender o modelo mental do autor. Portanto, caminhos de navegação que não foram previstos diminuem a compreensão do domínio. Isto acontece porque o autor da Aplicação Hipermídia, ao estabelecer uma ligação entre dois nós em detrimento de outras possibilidades, faz uma suposição sobre os conhecimentos prévios do usuário quando um nó específico é acessado.

Por sua vez, a ajuda navegacional através de mapas também é limitada. Um SH tradicional não registra informações necessárias para identificar as partes da estrutura de ligações que são mais importantes para o usuário. Conhecimentos sobre o domínio, o usuário e a interação com o sistema podem ser usadas para filtrar ou recomendar as ligações mais significativas (De Bra et al., 2000). Desta maneira, Brusilovsky (1996a, 1996b) mostra que a capacidade de adaptação em um ambiente Hipermídia é necessária por dois motivos principais. Primeiro, para atender a uma grande variedade de usuários com objetivos, interesses, preferências, experiências e conhecimentos distintos sobre o assunto coberto pelo sistema. Segundo, para proteger usuários do problema de perder-se no hiperespaço, restringindo-o por meio da sugestão das ligações mais significativas.

A definição de SHA aparece em publicações de diferentes autores com ligeiras diferenças. Para (Brusilovsky, 1996a, 1996b), SHA é todo o SH que reflete características do usuário para adaptar conteúdos e ligações. Uma definição ligeiramente mais abrangente que considera o contexto de trabalho do usuário aparece em (Kobsa et al., 1999): é todo SH que reflete características do usuário, de uso e do ambiente para personalizar conteúdo, ligações e/ou apresentação de objetos Hipermídia. Neste trabalho, utiliza-se a definição mais completa fornecida por Kobsa.

Kappel et al. (2003) estabelece uma estrutura para avaliar sistemas baseados na Web relacionados à terceira geração, na qual SHA está inserido. Com esta finalidade, os autores abstraem três características para classificar esse tipo de aplicação: o contexto, a adaptação e a customização. Para cada abstração consideram-se aspectos

distintos.

Por esta estrutura, pode-se dizer que um modelo é uma representação explícita das propriedades de um determinado contexto. Desta maneira, o modelo de usuário e de uso representam a *propriedade de usuários*. Seguindo o mesmo raciocínio, o modelo de ambiente proposto por Kobsa et al. (1999) corresponde à *propriedade de dispositivo*. Quando se pretende realizar a adaptação de conteúdos, é comum usar um modelo do domínio equivalente à *propriedade da aplicação* do contexto. Ou seja, diversos modelos podem ser gerados para o contexto a partir de fontes distintas, cujas propriedades modificam-se por meio de informações históricas do contexto. Isto implica que certas propriedades podem deixar de ser válidas ao longo do tempo.

Com relação à representação do contexto, os autores consideram aspectos como a possibilidade de reutilizar o modelo e o nível de abstração onde as propriedades do contexto são representadas (nível físico ou lógico). Em se tratando especificamente da adaptação de ligações para os interesses dos usuários, basta considerar o modelo de uso que costuma ser representado na forma de perfil implícito. A principal vantagem na aquisição de dados para esta representação é que se utiliza o método de aquisição passiva. Se o modelo do contexto foi adquirido previamente diz-se que ele é estático, mas se as mudanças no contexto são freqüentes ao longo da execução da aplicação, então diz-se que ele é dinâmico.

Com relação à adaptação propriamente dita, Kappel et al. (2003) consideram o tipo de adaptação em relação às operações, aos mecanismos de extensão, aos efeitos e à complexidade das adaptações. Os métodos de Hipermídia Adaptativa propostos por Brusilovsky (1996a) fazem abstração das operações mais comuns para personalizar conteúdos e ligações. Quanto aos efeitos da adaptação, podem ser: adicionar ou remover certas partes da aplicação ou realizar alguma transformação nos elementos. Já as técnicas referem-se à descrição dos efeitos das operações de adaptação. O assunto da adaptação caracteriza-se pelo nível da aplicação que é afetado (conteúdo, banco de dados não linear e/ou apresentação), os elementos que sofrem as adaptações (páginas, ligações, estruturas de acesso, campos, etc) e a granularidade indicando o número de elementos afetados por certas adaptações. Quanto ao processo de adaptação, considera-se quem é responsável pela adaptação (o sistema, o usuário ou ambos), o momento em que a adaptação é realizada (estático ou dinâmico). A

adaptação pode ser incremental quando as versões adaptadas persistem enquanto as adaptações subsequentes são realizadas com base nos resultados das adaptações anteriores.

Como se pode notar, a HA é capaz de manter a liberdade de navegação ao mesmo tempo em que se beneficia com o conhecimento de diversas fontes para gerar adaptações automáticas. Em geral, um motor inteligente aplica alguma técnica de raciocínio para tomar decisões de adaptação com base no(s) modelo(s) disponíveis. A HA tem sido aplicada a cinco áreas principais (Brusilovsky, 1996a, 1996b, 2001):

1. sistemas educacionais;
2. sistemas de informações *on-line*;
3. sistemas de ajuda *on-line*;
4. recuperação de informações Hipermissão; e
5. visões personalizadas.

Como mostra (Brusilovsky, 2001), as áreas mais populares de pesquisa em HA são a Educação e os sistemas de informações *on-line*. Em seguida, aparecem os sistemas de ajuda *on-line* e recuperação de informações Hipermissão. A escolha da *Web* como plataforma deve-se, segundo o autor, ao sucesso e longa vida dos primeiros sistemas que têm sido atualizados com novas técnicas e usados em estudos experimentais. Brusilovsky & Maybury (2002) referem-se a estudos mostrando que a adaptação automática de conteúdos e ligações aumentam a velocidade de navegação e de aprendizado, além de aumentar a compreensão do conteúdo.

A adaptação automática necessita de um suporte para identificar e produzir a personalização. Se o que se pretende é uma adaptação de conteúdo, é preciso usar métodos e técnicas para representar o domínio da aplicação e produzir inferências sobre ele. Se o alvo é a adaptação de ligações, o modelo de usuário é o principal recurso usado para reconhecer a necessidade de adaptações. Ele contém as suposições do sistema sobre todos os aspectos relevantes para gerar personalizações (Kobsa, 1993, 1994). De acordo com (De Bra, 1999), o modelo de usuário representa o estado mental dos indivíduos que acessam o sistema para decidir como realizar

uma personalização. A HA, propondo o ajuste de conteúdos e ligações para as características dos indivíduos, implica a (re)formulação de arquiteturas e de modelos existentes para Hipermídia de modo que seja possível suportar o desenvolvimento de sistemas e aplicações em HA. Este é o tema abordado na próxima Seção.

3.2 Modelos para Hipermídia Adaptativa

Diversos métodos, técnicas e ferramentas têm sido pesquisados e desenvolvidos para a HA. Alguns focalizam a arquitetura e o motor adaptativo, enquanto outros concentram-se na modelagem de domínios e aplicações de HA. Um método denominado AHDM (*Adaptive Hypermedia Design Method*) proposto em (Alatalo & Peräaho, 2001) oferece auxílio para a modelagem HA que integra a Modelagem de Usuário no processo. Para (Koch & Wirsing, 2001), abordagens de Engenharia de Software orientadas a objetos não são suficientes para suportar o processo de desenvolvimento de SHAs. Tais metodologias não cobrem aspectos importantes para personalização de conteúdos e ligações e modelagem de usuários. Portanto, os autores sugerem uma abordagem baseada na UML modificada para suportar aplicações adaptativas. A esta metodologia, os autores deram o nome de UWE (*UML-based Web Engineering*).

Algumas ferramentas para apoiar a autoria de Aplicações Hipermídia foram desenvolvidas por (Brusilovsky et al., 1996), (Calvi & De Bra, 1998) e (Henze & Nejd, 2000). A cada ano, novas aplicações, metodologias e ferramentas são pesquisadas e desenvolvidas. Por ser uma área relativamente recente, a HA apresenta apenas resultados experimentais e parciais (Brusilovsky, 2000). Portanto, a HA abre perspectivas de investigação em inúmeros pontos, dentre os quais podem ser citados:

1. estabelecimento de arquiteturas para SHAs;
2. desenvolvimento de modelos de dados para Aplicações de HA;
3. estudo do impacto da adaptação Hipermídia em diversos domínios e aplicações;
4. aplicação e criação de tecnologias que melhorem a qualidade da adaptação; e
5. estabelecimento de metodologias, métodos, técnicas e ferramentas que auxiliem na criação de aplicações e sistemas de HA

Uma vez que a HA acrescenta a capacidade de adaptar automaticamente o conteúdo e/ou as ligações em um SH, é natural que o desenvolvimento de sistemas e aplicações desta natureza tenha uma complexidade adicional. Portanto, é importante que se estabeleçam modelos específicos. Em (Brusilovsky, 1996b), surge a primeira tentativa de se especificar uma estrutura interna genérica para SHAs. O autor identifica quatro características mais importantes comuns aos primeiros SHAs:

1. são formados por um conjunto de conceitos;
2. os conceitos são ligados formando uma rede — o modelo do domínio coberto pelo sistema — que pode ser de nível 1 (conceitos independentes), nível 2 (rede semântica) ou nível 3 (baseado em *frames*);
3. cada conceito pode ter uma ou mais páginas de representação externa; uma página pode corresponder a diferentes conceitos; ou cada página pode corresponder a um único conceito; e
4. utilizam modelo de usuários para registrar, manter e atualizar informações sobre usuários, uso e ambiente para as quais serão realizadas as adaptações.

Figura 3.1: Modelo de Aplicação para Hipermedia Adaptativa (AHAM).

Com o objetivo de facilitar o desenvolvimento de SHAs, De Bra et al. (1999b) desenvolveram um modelo de referência baseado no Dexter denominado AHAM (*Adaptive Hypermedia Application Model*). Wu (2001) destaca que a utilização deste

modelo ajuda na escolha das adaptações úteis e utilizáveis, na separação de conceitos, na flexibilização do sistema, além de permitir entender e comparar SHAs. O AHAM aprimora o modelo de referência Dexter para a WWW com a capacidade de realizar adaptações baseadas em um modelo de usuário persistente ao longo da sessão, como mostra a Figura 3.1. A personalização acontece com a utilização de três modelos (De Bra et al., 1999a, 1999b, 2000):

1. modelo de domínio: descreve como as informações são estruturadas e ligadas. Corresponde à camada de armazenamento do modelo Dexter;
2. modelo de usuário: descreve quais informações sobre os usuários um SHA deve manter armazenado; e
3. modelo de adaptação: formado por regras adaptativas que definem como o modelo de domínio e o modelo do usuário são combinados para prover adaptação.

O domínio pode ser descrito em três níveis diferentes: conceitos e relacionamento de conceitos, unidades de apresentação (páginas *Web*) e unidades atômicas ou fragmentos. Além disto, o modelo AHAM oferece um formalismo que permite autores escreverem regras de adaptação sobre conceitos. O motor adaptativo usa regras para gerar especificações de apresentação além de atualizar o modelo do usuário depois de cada evento e pode facilmente ser implementado como um *script* CGI ou um *servlet*. As tarefas realizadas pelo motor adaptativo, quando um usuário segue uma ligação, são detalhadas em (De Bra et al., 1999b, 2000).

Outros modelos e arquiteturas foram propostos para o desenvolvimento de aplicações de HA. Entre eles: XAHM para aplicações baseadas em XML (Cannataro & Pugliese, 2001), ACE que estabelece um ambiente para criação de cursos adaptativos (Oppermann & Specht, 1998) e KBS *Hyperbook System* (Henze & Nejd, 2000) baseado em SHs abertos. Na prática, as aplicações costumam aparecer antes fornecendo uma estrutura subjacente que origina um possível modelo de referência ou arquitetura de sistema.

O problema com modelos de propósito geral para SHAs baseados em conhecimento — tal qual o AHAM — é a sua especificidade para cada domínio. Ou seja, o modelo de usuário é fortemente baseado em características dependentes do domínio (Brusilovsky, 1996b, 2001). Desta maneira, para cada nova Aplicação Hipermídia

com um domínio diferente, um sistema completo com todos os modelos (de domínio, de usuário e de adaptação) deve ser novamente produzido. Para resolver estes problemas, diversos autores propõem um sistema híbrido e genérico para a HA unindo a representação baseada em conhecimento com técnicas mais avançadas e eficientes de IA. A idéia é baseada em uma arquitetura Hipermedia de nós e ligações com um módulo inteligente embutido que usa um modelo do nível de experiência atual do usuário para facilitar a navegação.

A maneira como o domínio é estruturado afeta tanto o processo de autoria quanto de navegação, como mostram (Mullier, 1995, 1999), (Mullier et al., 1999), (Mullier & Dixon, 2000). Por isto, os autores sugerem que os domínios sejam organizados em uma rede Hipermedia usando ligações semânticas — a Hipermedia Semântica. O sistema proposto restringe a navegação de usuários inexperientes às ligações estabelecidas pelos autores ao mesmo tempo em que permite aos indivíduos mais avançados navegarem livremente pela estrutura do domínio. Para garantir um modelo de usuário independente do domínio, (Mullier 1995, 1999, 2000) e (Mullier et al., 1999a, 1999b) sugerem o uso de RNAs para extração de informações como as estratégias de navegação e a habilidade de responder questões.

Com a criação da *Web*, a preocupação em padronizar e estabelecer o intercâmbio de aplicações entre SHs diferentes foi eliminada. Assim, o foco das pesquisas na área foi deslocado tanto para as aplicações quanto para outros campos de interesse como, por exemplo, o trabalho colaborativo e a adaptação automática de conteúdos e/ou ligações. A próxima Seção trata da HA como um processo de adaptação, mostrando como o modelo de usuário pode ser adquirido e usado para gerar personalizações de conteúdos e ligações.

3.3 Modelagem de Usuários

O modelo de usuário é o principal recurso usado para reconhecer a necessidade de adaptações. Ele contém as suposições do sistema sobre todos os aspectos relevantes para a realização de personalizações (Kobsa, 1993, 1994). Por esta razão, o autor recomenda que todo SA tenha um componente de MU que é responsável pelas seguintes tarefas:

1. identificar as informações significativas;
2. armazenar as informações em um sistema de representação apropriado;
3. inferir suposições sobre os dados armazenados;
4. manter a consistência do conjunto de suposições correntes; e
5. suprir outros componentes do sistema com suposições sobre os usuários.

Em (Kobsa et al., 1999) e (Brusilovsky, 1996a), os autores referem-se à MU como parte de um processo de personalização, em especial para aplicações baseadas na WWW. O sistema colhe informações armazenando-as em um modelo que serve como base para o processamento e a realização de adaptações. Assim, o modelo de usuário representa o estado mental dos indivíduos que acessam o sistema, sendo usado para decidir como realizar uma personalização (De Bra, 1999).

A Modelagem de Usuários surgiu a partir de investigações no campo dos Sistemas de Diálogo em Linguagem Natural (Kobsa, 1993). Mostra-se particularmente útil para aplicações que oferecem adaptações automáticas para os perfis de indivíduos. Um conjunto de técnicas é proveniente de outros campos de pesquisas, sobretudo da IA e da Interação Homem-Computador (IHC). Outro grupo foi desenvolvido independentemente, encontrando aplicação em diversos SIs acessados por usuários heterogêneos. É o caso das Interfaces Inteligentes, dos SHs, dos STIs, dos Sistemas de Ajuda Passiva e Ativa, entre outros. A MU está inserida no processo de adaptação automática de SAs através da construção de modelos de usuário que envolve:

1. a coleta de dados significativos;
2. o processamento de dados para construir ou atualizar o modelo de usuário; e
3. a aplicação do modelo para a produção de adaptações.

O processo de adaptação é formado por três fases, como mostra a Figura 3.2. Em cada uma destas fases são selecionados e aplicados métodos e técnicas de MU, IA e HA. As duas primeiras fases tratam da aquisição e representação das informações significativas obtidas de diferentes fontes para a construção de um ou mais modelos de usuários. Em alguns casos é preciso realizar algum tipo de pré-processamento

sobre o modelo obtido. A última etapa – a produção da adaptação – envolve a aplicação de técnicas de inferência da IA sobre o modelo de usuário. Por fim, define-se uma estratégia onde são selecionadas as tecnologias de HA e as suposições obtidas são aplicadas para gerar as adaptações. As Subseções a seguir tratam de tópicos relacionados com cada uma das três fases do processo de adaptação automática. (Pohl & Nick, 1999) consideram as técnicas de aquisição e representação do modelo de usuário como um processo de representação de conhecimento, uma sub-área de estudo da IA. Entretanto, neste documento, será usada a abordagem de MU por ser a mais amplamente utilizada na literatura de HA.

Figura 3.2: Processo de personalização em sistemas adaptativos.

As próximas Subseções abordam a aquisição e a representação do modelo de usuário, bem como as tecnologias para a produção de adaptação. Para maiores detalhes a respeito destes tópicos, consultar (Kobsa et al., 1999).

3.3.1 Aquisição de Dados

A fase de aquisição identifica e colhe as informações - suposições - importantes que estão disponíveis a respeito dos usuários, do seu comportamento de uso e do ambiente pelo qual o sistema é acessado. Além disto, a aquisição torna as informações acessíveis por meio de modelos iniciais de usuário, de uso e de ambiente. Nestes casos, normalmente é preciso realizar algum tipo de pré-processamento sobre os dados brutos.

O modelo inicial de usuário registra informações sobre características individuais, enquanto o modelo inicial de uso colhe dados sobre o seu comportamento observável. As informações de usuários e de uso costumam sobrepor-se, dificultando a distinção

entre elas. Em geral, os dados de usuário abrangem as informações demográficas, os conhecimentos, as habilidades, os interesses, as preferências e os objetivos particulares. Os dados de uso são provenientes do comportamento do usuário, podendo ser diretamente observados, produzindo algum tipo de adaptação simples. Consideram-se informações de uso diretamente observáveis as ações seletivas e aquelas tomadas ao longo da interação com a aplicação, o comportamento temporal e a classificação sobre itens de interesse do indivíduo.

Outra abordagem, usada com frequência na aquisição de dados, consiste em formar uma base de conhecimento onde são registrados os dados de uso para descobrir padrões gerais de regularidade. Alguns destes padrões podem manifestar as preferências, as habilidades, os hábitos ou o nível de conhecimento dos usuários que são necessários para gerar adaptações. Nestes casos, são consideradas como informações de uso passíveis de registro e análise a frequência de uso, a correlação entre situação e ação e a sequência de ações.

Há dois métodos gerais para adquirir os dados em Sistemas Interativos: aquisição ativa - baseada na entrada explícita de dados - e aquisição passiva - baseada na observação implícita do comportamento. As estratégias mais comuns para adquirir dados para um modelo de usuário são: a informação suprida pelo usuário, as regras de aquisição, o reconhecimento de planos e os estereótipos.

No caso de aplicações baseadas na WWW, a coleção de dados sobre o ambiente de *software* pode ser adquirida por meio da análise do cabeçalho das mensagens de pedido de recursos para um servidor HTTP. Algumas vezes, a informação sobre o *hardware* pode ser inferida a partir do tipo de navegador utilizado, mas em geral é difícil de obter este tipo de dado. O conhecimento sobre a largura de banda pode ser útil para decidir quais recursos incluir em uma página para diminuir o seu tempo de carga.

Em situações onde é preciso utilizar o método de aquisição ativa, aconselha-se evitar longos processos de registro ou entrevistas iniciais, limitando a duas ou três questões centrais embutidas no conteúdo principal. Além disto, nem sempre as informações supridas pelos indivíduos são confiáveis. Por esta mesma razão, o sistema que necessita da classificação de itens diretamente do usuário não tem garantia de que todos os elementos serão rotulados. Nem tampouco se esta classificação é significativa ou aleatória.

Quanto à técnica de aquisição passiva de dados de uso, não é recomendável registrar informação em nível de micro interação - como movimentos de mouse - a menos que o propósito desses registros tenha sido especificado. A quantidade de dados é grande, o tempo de processamento para realizar as recomendações é longo e a confiabilidade na qualidade destas adaptações pode ser baixa. Apesar disto, (Kobsa et al., 1999) e (Brusilovsky, 2001) argumentam que a realização de experimento laboratorial com dados desta natureza pode ajudar a orientar o desenvolvimento de métodos na área de HA.

3.3.2 Representação de Dados

A fase de representação de dados concentra-se na expressão do conteúdo dos modelos em sistemas formais apropriados para que sejam acessados e processados. As informações de usuários, de uso e de ambiente são integradas para realizar suposições adicionais e generalizações pela aplicação de técnicas de inferência provenientes da IA.

A principal forma de representação para modelos de usuários baseia-se na Lógica Clássica (Kobsa, 1993, 1994; Kobsa et al., 1999). Neste caso, o modelo encapsula o conhecimento a respeito do domínio ou do usuário na forma de regras. Este conhecimento é usado para disparar as adaptações necessárias. Existem duas restrições com esta abordagem: a capacidade limitada para lidar com a incerteza e com modificações nos modelos. Para representar o raciocínio com incerteza costuma-se empregar outros tipos de Lógica ou modelos probabilísticos que, no entanto, são mais difíceis de utilizar.

Uma forma de representação para o conhecimento dos usuários bastante comum em Sistemas Tutores Inteligentes é o modelo de sobreposição (Brusilovsky, 1996a; Marietto, 2000). Costuma ser aplicado em domínios estruturados como uma rede de conceitos rotulados e inter-relacionados, tal como em uma Rede Semântica. Para cada conceito do domínio, um modelo de sobreposição individual armazena um valor que corresponde à estimativa do nível de conhecimento do usuário sobre este conceito. Valores contínuos são um indicativo de incerteza ou do grau em que determinada característica está associada ao indivíduo. Um esquema como este é facilmente representado por um conjunto de pares de atributo-valor para cada

conceito coberto pelo domínio. Algumas modificações no modelo de sobreposição permitem que outras características dos usuários sejam modeladas.

Um perfil implícito ou explícito pode ser usado para gerar conclusões a respeito do comportamento do usuário. O processamento adicional sobre estes dados pode revelar o interesse, a preferência, o hábito, o nível de experiência ou a habilidade de um ou mais indivíduos. Nesta forma de representação, as características são generalizadas para possibilitar adaptações em novos contextos (Brusilovsky, 1996a).

O perfil de interesse explícito é construído com base na análise das características dos objetos em que o usuário demonstra seu interesse por um mecanismo de votação ou classificação destes objetos. Neste caso, o perfil costuma ser expresso por meio de um vetor de características. Abordagens que consideram as características dos elementos são denominadas tecnologias de filtragem baseadas em conteúdos.

Por sua vez, o perfil de interesse implícito tenta encontrar usuários que apresentem comportamento de interação semelhante. Neste caso, o sistema adapta o ambiente para um usuário individual baseado no comportamento de vizinhos de interesse. A classe à qual um indivíduo pertence reflete um grupo de usuários com comportamentos e interesses similares. Desta forma, há uma forte possibilidade de que os objetos pertencentes a esta classe sirvam como recomendação útil aos seus membros. Essa abordagem é conhecida por alguns autores como tecnologia de filtragem baseada em clique.

A representação dos modelos de usuário, de uso e de ambiente na *Web* pode ser armazenada tanto no cliente quanto no servidor (De Bra, 1999). O armazenamento do modelo é temporário ou permanente. A manutenção de modelos no cliente prevê o uso de *cookies* que apresentam uma série de desvantagens e restrições. Por esta razão, o autor aponta o armazenamento do modelo de usuário no servidor como um importante recurso para a personalização de aplicações baseadas na *Web*.

3.3.3 Métodos e Técnicas de Adaptação

O objetivo da MU é prover um modelo sobre a qual seja possível realizar a produção automática de personalizações sob vários aspectos de um SI. A personalização é produzida depois que o modelo foi adquirido e representado de maneira formal, servindo de base para um módulo inteligente realizar inferências automaticamente.

À medida em que o usuário interage com o sistema, normalmente os modelos de usuário e de uso são atualizados e revistos. Há, basicamente, cinco situações em que os modelos podem sofrer modificações em SHAs, segundo (De Bra et al., 2000):

1. o usuário faz uma seleção/escolha;
2. o usuário realiza um teste;
3. informações sobre os usuários são importadas de um sistema externo;
4. uma preferência do usuário é explicitamente declarada; e
5. uma preferência do usuário é automaticamente inferida a partir do seu comportamento.

A adaptação de conteúdo e de estrutura envolve a aplicação de métodos e técnicas bastante pesquisadas e que têm sido empregados com regularidade em aplicações de HA baseadas na WWW. A primeira ampla classificação de métodos e técnicas com este propósito aparece em (Brusilovsky, 1996a). Recentemente, novos aspectos e tecnologias para HA foram desenvolvidos. Como resultado, a taxonomia foi revista e modificada em (Brusilovsky, 2001), como mostra a Figura 3.3. No entanto, o primeiro trabalho ainda é uma importante referência contendo uma descrição detalhada das principais tecnologias para a personalização no ambiente WWW.

Alguns autores preferem utilizar outros termos para designar os métodos de HA. Assim, o termo adaptação de conteúdo empregado por (Kobsa et al., 1999), (De Bra, 1999) e (De Bra et al., 1999a, 1999b) é chamado de apresentação adaptativa por Peter Brusilovsky. Enquanto o termo adaptação de estrutura utilizado por (Kobsa et al., 1999) é equivalente a adaptação de ligações em (De Bra, 1999; De Bra et al., 1999a, 1999b) e a suporte navegacional adaptativo (Brusilovsky, 1996a, 2001). Por ser mais claro e fácil de compreender, o presente trabalho adota a nomenclatura de Paul De Bra, porém preserva os termos de Brusilovsky nas Tabelas e Figuras que classificam os métodos e técnicas de HA.

O conjunto de métodos e técnicas para **apresentação adaptativa de textos** é descrito com detalhes em (Brusilovsky, 1996a) e aparece resumido nas Tabelas 3.1 e 3.2, respectivamente. Tecnologias mais recentes e menos utilizadas, como adaptação

Figura 3.3: Tecnologias de personalização Hipermídia (Brusilovsky, 2001).

multimídia e de modalidade, não são cobertas com detalhes por (Brusilovsky, 2001) e, por isto, não aparecem nas Tabelas.

Trabalhos como (Brusilovsky, 1995, 1996b), (Mullier, 1995) e (De Bra et al., 1999b) consideram que o usuário inexperiente precisa de explicações adicionais e um controle maior do sistema, enquanto que o usuário avançado está interessado em detalhes e na liberdade de navegação. De acordo com De Bra et al. (1999b), a técnica básica para apresentação adaptativa de texto é o *conditional text* que pode ser facilmente usado para implementar *stretchtext*, *fragment variants* e *page*

Tabela 3.1: Métodos para a apresentação adaptativa de texto.

<i>Additional explanation</i>	oculta partes de informações sobre um conceito em particular que não seja relevante.
<i>Prerequisite explanation</i>	modifica informações apresentadas dependendo do conhecimento sobre conceitos relacionados.
<i>Comparative explanation</i>	oferece explicações comparando conceitos bem conhecidos pelo usuário.
<i>Explanation variants</i>	o sistema mantém uma grande variação de explicações sobre um mesmo conceito.
<i>Sorting</i>	classifica trechos de informações sobre conceitos relevantes ao usuário.

Tabela 3.2: Técnicas para apresentação adaptativa de texto.

<i>Conditional text</i>	divide os conceitos em pedaços menores associados a condições sobre o nível de conhecimento dos usuários.
<i>Stretchtext</i>	a ativação de uma palavra-chave causa a inserção de um texto associado dentro do nó corrente.
<i>Fragment variants</i>	usuário acessa explicações para os conceitos fragmentados em diversos níveis de conhecimento.
<i>Page variants</i>	duas ou mais versões para um mesmo conceito são preparadas para possíveis classes de usuários.
<i>Frame-based technique</i>	informações são apresentadas em <i>frames</i> com variações nas explicações, conceitos, ligações selecionadas por regras especiais.

Tabela 3.3: Métodos para suporte navegacional adaptativo.

<i>Global guidance</i>	Provê informações gerais que auxiliam o usuário a encontrar o menor caminho até a informação que precisa.
<i>Local guidance</i>	Ajuda o usuário a decidir pelo melhor nó a ser visitado em seguida, à partir do estado corrente.
<i>Local orientation support</i>	Ajuda o usuário a localizar sua posição relativa dentro do documento.
<i>Global orientation support</i>	Ajuda o usuário a entender a estrutura geral e sua localização absoluta dentro dele.

variants. Em geral, o SA oculta fragmentos de conteúdos considerados indesejáveis aos interesses do usuário. No entanto, vários autores sugerem que o sistema deve permitir que o usuário veja todos os itens e não apenas o subconjunto recomendado.

A adaptação de ligações modifica ou anota a estrutura de ligações de modo a orientar o usuário em direção à informações interessantes e relevantes (De Bra et al., 1999b). Adaptação de ligações simplifica a estrutura Hipermissão para reduzir o problema da desorientação, ao mesmo tempo em que mantém a liberdade de navegação. Os métodos e técnicas para **suporte navegacional adaptativo** são classificados por (Brusilovsky, 1996a), como mostram as Tabelas 3.3 e 3.4, respectivamente. O suporte à orientação indica quais ligações são preferíveis sobre outras. Para tanto, depende da determinação de contextos e requer um mapa da estrutura de ligações ao redor do nó corrente. Um conjunto de técnicas em (De Bra et al., 1999b) sugere a desabilitação ou remoção de ligações. No entanto, não é desejável utilizar este tipo de recurso, pois diminuem a capacidade de compreensão, além de tornar certos conteúdos inacessíveis.

Tabela 3.4: Técnicas para o suporte navegacional adaptativo

<i>Direct guidance</i>	o sistema decide qual o melhor nó a ser visitado pelo usuário no passo seguinte.
<i>Sorting</i>	utiliza algum critério útil para ordenar um conjunto de ligações colocando os mais significativos em primeiro lugar.
<i>Hiding</i>	restringe o espaço de navegação ocultando ligações pouco relevantes para os usuários.
<i>Annotation</i>	coloca comentários ao lado das ligações que indiquem algo sobre o estado corrente dos nós apontados pelas ligações.
<i>Map adaptation</i>	compreende diferentes maneiras de adaptar mapas hipermídia locais e globais.

Importa observar que não é possível implementar todas as técnicas de adaptação em um único sistema, pois isto o tornaria inutilizável. O projetista de SHAs deve escolher as mais adequadas e desenvolver ou selecionar um sistema que suporte esta escolha. O Capítulo 2 aborda as técnicas da área de IA que permitem gerar suposições através do raciocínio sobre um conjunto de fatos existentes. É com base nestas suposições que um sistema de HA decide como personalizar conteúdos e/ou ligações para cada usuário.

3.4 Considerações Finais

A capacidade adaptativa de SHs incrementa a funcionalidade e possibilita ao usuário encontrar as informações de que necessita, bem como atingir os objetivos a que se propõe de maneira mais rápida e efetiva, como mostra este Capítulo. No entanto, para que isto seja possível, é preciso integrar as técnicas de MU e IA em um processo de adaptação automática que permita ao sistema identificar a necessidade de personalização e produzir as suposições necessárias para realizá-la. É com a integração destes conhecimentos que torna-se possível a criação de SHAs.

O Capítulo a seguir conjuga os conhecimentos apresentados até agora para a formulação de uma metodologia que foi usada nesta dissertação, mostrando como obter uma base de dados para treinamento de indutores supervisionados e como os classificadores podem ser usados para se conseguir a adaptação automática de ligações em ambiente WWW.

Capítulo 4

Metodologia de Trabalho

Neste trabalho, investigou-se a utilização de algoritmos supervisionados de AM representativos de diferentes paradigmas, de modo a obter classificadores que foram avaliados em relação ao seu desempenho para utilização em adaptação de ligações. A Figura 4.1 mostra um esquema onde um classificador deste tipo pode ser utilizado no ambiente WWW. O usuário utiliza um navegador para acessar as páginas que formam a Aplicação Hipermédia armazenadas em um servidor HTTP em três aspectos: conteúdo, ligações e apresentação. O classificador atua como um modelo de usuário para o SHA. Para tanto, são utilizadas as informações de navegação do usuário atual para determinar a(s) página(s) de interesse baseada(s) nas páginas visitadas por outros indivíduos e que foram usadas para o treinamento dos algoritmos de AM. A saída do classificador pode ser usada para alterar as ligações existentes nas páginas pela aplicação dos métodos e técnicas de HA antes de serem enviadas de volta para o navegador.

Figura 4.1: Classificador aplicado ao ambiente WWW.

Para atingir o objetivo proposto, foram realizados experimentos sobre a base de

dados MSWeb da Microsoft (Hettich & Bay, 1999) obtida no repositório da UCI. Os resultados devem indicar o melhor paradigma de AM para trabalhar com dados de implícitos e quais as condições para sua utilização. Este Capítulo descreve a metodologia usada para aplicar os algoritmos de AM. A Seção 4.1 descreve o pré-processamento necessário para adquirir o modelo de usuário, bem como algumas estatísticas e características das bases de dados obtidas após o pré-processamento e que são utilizada nos experimentos. Alguns comentários a respeito dos algoritmos de AM utilizados são feitos na Seção 4.2. Os critérios usados para avaliar e comparar os experimentos são descritos na Seção 4.3.

4.1 Pré-processamento da Base

A base de dados MSWeb foi adquirida por (Hettich & Bay, 1999) utilizando um *script* em Perl no servidor da Microsoft ao longo de uma semana de acessos durante o ano de 1998. Trata-se de uma coleção de dados de uso adquiridos por método passivo. Nela são registradas as páginas visitadas pelos usuários. A base completa conta com 32710 usuários numericamente identificados. De modo semelhante, as 298 páginas do *website*¹ da Microsoft recebem uma numeração única.

As informações de acesso às páginas estão armazenadas em um formato que os autores chamam de DST que contem o registro da sequência de páginas acessadas pelos usuários. Registros que iniciam com a letra A são seguidos de um número de identificação para a página, uma constante (1), o título da página e o seu endereço a partir da raiz <http://www.microsoft.com>. Desta forma, o registro A,1003,1,"Knowledge Base", "/kb" identifica a página intitulada *Knowledge Base* com o número 1003 e que é acessada no endereço <http://www.microsoft.com/kb>. Os registros que começam com a letra C identificam o usuário e são seguidos de um ou mais registros iniciando com V. Assim, o registro C, "10001",10001 refere-se ao usuário 10001 e os registros imediatamente seguintes - V,1000,1 / V,1001,1 / V,1002,1 - mostram as páginas acessadas pelo usuário 10001. Maiores informações a respeito desta base de dados podem ser consultadas em (Hettich & Bay, 1999). Como se pode notar na Figura 4.2, este não é um formato adequado para a aplicação de algoritmos de AM supervisionados. Portanto, o primeiro passo consiste em realizar um pré-processamento sobre

¹uma Aplicação Hipermissão baseada na WWW.

o modelo inicial de usuário para gerar uma representação apropriada ao treinamento destes algoritmos.

Considerando que as informações armazenadas nesta base são resultado da observação do comportamento dos indivíduos no acesso ao *site*, conclui-se que ela constitui o modelo inicial de dados de uso implícito. Como mostra o Capítulo 2, a fase de aquisição do modelo de usuário pode exigir algum tipo de pré-processamento das informações para gerar uma forma de representação adequada ao processamento computacional. Além disto, os algoritmos de AM geram suposições gerais aplicando o raciocínio indutivo sobre um conjunto de exemplos rotulados, onde cada instância é uma tupla contendo o valor das características descritas pelos atributos. Desta maneira, o conjunto de exemplos torna-se uma representação adequada na forma de perfil implícito para o treinamento de indutores supervisionados com o objetivo de obter o modelo de dados dos usuários. O restante desta Seção descreve a maneira como a base original foi processada para gerar os conjuntos de exemplos para a realização dos experimentos.

```
A,1000,1,"regwiz","/regwiz"  
A,1001,1,"Support Desktop","/support"  
A,1002,1,"End User Product View","/athome"  
A,1003,1,"Knowledge Base","/kb"  
A,1004,1,"Microsoft.com Seach","/search"  
C,"10001",10001  
V,1000,1  
V,1001,1  
V,1002,1  
C,"10002",10002  
V,1001,1  
V,1003,1  
C,"10003",10003  
V,1001,1  
V,1003,1  
V,1004,1  
C,"10004",10004  
V,1004,1
```

Figura 4.2: Arquivo MSWeb inicial em formato DST.

Um *script* escrito em Perl realizou o pré-processamento necessário para transformar o modelo de dados de uso inicial no formato DST em um perfil implícito. Portanto, o seu objetivo consiste em obter o conjunto de exemplos adequado para a utilização por diferentes algoritmos de AM supervisionados. Para tanto, parte-se do princípio de que as páginas representam itens sobre os quais um indivíduo pode

ou não estar interessado. Quando o usuário seleciona uma dentre várias ligações existentes em uma página, pode-se concluir que existe algum interesse no conteúdo que será recuperado para atingir um determinado objetivo. Assim, o registro do interesse de todos os n usuários pelas m páginas pode ser representado por uma matriz. Cada linha da matriz (4.1) é um padrão indicando as páginas que foram visitadas por um mesmo indivíduo.

$$\begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} & \dots & v_{1m} \\ v_{21} & v_{22} & v_{23} & v_{24} & \dots & v_{2m} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ v_{n1} & v_{n2} & v_{n3} & v_{n4} & \dots & v_{nm} \end{pmatrix} \quad (4.1)$$

Para gerar o conjunto de exemplos considera-se que os usuários que percorrem a mesma trilha em uma rede Hipermídia possuem interesses e objetivos semelhantes. Tendo em vista que dificilmente dois indivíduos com objetivos similares percorrem seqüências idênticas de nós e ligações, conclui-se que a informação significativa está no conjunto daquelas páginas que foram acessadas, não importando a ordem de visitação. Isto equivale a dizer que quando um usuário visita k das m páginas do *website* ($k \leq m$) e que ao final da navegação tenha atingido o objetivo proposto, então, deve haver uma forte relação entre as k páginas, independentemente da ordem em que elas tenham sido acessadas.

Cada padrão pode ser representado em um perfil implícito na forma de um vetor. Os atributos recebem o valor 0 (zero) quando não foram visitados pelo usuário. Para as páginas visitadas, o atributo recebe o valor 1 (um). Portanto, um padrão do conjunto de exemplos indica as páginas que um usuário acessa enquanto explora o espaço de navegação para atingir um determinado objetivo (4.2). Ligeiras modificações neste procedimento permitem inserir informações no conjunto de exemplos como, por exemplo, a seqüência de acesso às páginas. Deste modo, a página mais recentemente visitada recebe o valor 1 (um), enquanto todas as outras páginas que foram visitadas anteriormente são subtraídas de uma constante c , de maneira que as primeiras páginas visitadas aproximam-se do valor zero (0) à medida em que o usuário navega.

usuário	pág. 1000	pág. 1001	pág. 1002	pág. 1003	pág.1004
10001	1	1	1	0	0
10002	0	1	0	1	0
10003	0	1	0	1	1
10004	0	0	0	0	1

(4.2)

A utilização do modo de aprendizado supervisionado implica que cada padrão no conjunto de treinamento precisa ter uma classe para guiar o aprendizado dos indutores. Entretanto, a base de dados no formato original não possui nenhuma classe explícita. De modo geral, pode-se dizer que de um conjunto V de páginas visitadas por um indivíduo ($V = \{I_1, I_2, \dots, I_k\}$), a página I_p ($1 \leq p \leq k$) é determinada pelas outras $V - I_p$. Em outras palavras, I_p é a classe do exemplo $V - I_p$, produzindo, k padrões para um único usuário. Esta abordagem para gerar exemplos rotulados será chamada de muitos-para-um. A sua aplicação para a primeira instância de (4.2) aparece em (4.3). São características do conjunto de exemplo gerado: a ausência de ruídos e uma alta esparsidade (presença de atributos nulos). A maior quantidade de páginas visitadas por um mesmo usuário na base MSWeb é de 35, restando 263 ($298 - 35$) páginas que não foram visitadas, isto é, que serão fixadas para o valor 0 (zero) na matriz 4.1.

pág. 1000	pág. 1001	pág. 1002	pág. 1003	pág. 1004	...	classe
0	1	1	0	0	...	0 (pág. 1000)
1	0	1	0	0	...	1 (pág. 1001)
1	1	0	0	0	...	2 (pág. 1002)

(4.3)

O procedimento completo descrito anteriormente para gerar um conjunto de exemplos para o aprendizado supervisionado merece algumas considerações. Não há garantias, por exemplo, de que todos os usuário tenham conseguido atingir os objetivos propostos inicialmente ou que estes objetivos permaneçam os mesmos durante a navegação, especialmente se tratando da WWW. Conseqüentemente, os exemplos obtidos podem não ser tão úteis quanto se espera. Além disto, há de se considerar que um indivíduo pode ter visitado uma página por engano e que tenha rapidamente retornado à página anterior ou selecionado uma ligação mais adequada. Assim, devem haver muitas páginas que não contribuem significativamente para atingir a meta desejada. Com isto, os indutores podem ter seu desempenho afetado, pois estarão

aprendendo padrões pouco expressivos. Uma maneira de resolver este problema consiste em restringir o domínio e o uso da Aplicação Hipermídia à tarefas específicas e bem conhecidas, como é o caso da base de dados utilizada neste trabalho. Além disto, o que se pretende é comparar o desempenho de vários algoritmos de AM nesta tarefa, e não medir o grau de utilidade da ligação que o sistema sugere para o usuário.

Figura 4.3: Número de páginas acessadas por usuário.

Outra consequência importante com relação ao procedimento descrito é que o número de exemplos gerados para cada indivíduo é proporcional ao número de páginas que ele visitou. Ou seja, cada indivíduo que visita k das m páginas do *website* ($k \leq m$) deverá produzir k exemplos. Conseqüentemente, quanto mais páginas são visitadas em um *website*, maior o tamanho do conjunto de exemplos (tanto em número de padrões quanto no espaço ocupado em disco), influenciando no tempo de treinamento dos algoritmos de AM. No entanto, esse tamanho pode ser reduzido desde que se admita a hipótese anterior, na qual os exemplos gerados podem não ser tão úteis quanto se espera. A idéia para diminuir o conjunto de exemplos consiste em analisar o número de páginas visitadas por cada indivíduo selecionando apenas aqueles que possuam ligações que possam ser úteis, isto é, que tentam indicar um caminho significativo para se atingir o objetivo proposto.

O histograma da Figura 4.3 mostra a quantidade de usuários que acessam k páginas ($0 \leq k \leq 35$). Existem, no total, 32710 usuários na base de dados original. Tendo em vista que a maioria dos usuários visita de uma a cinco páginas, pode-

se restringir o conjunto de exemplos apenas para usuários que tiveram entre 2 e 5 votos, que somam 18565 usuários, isto é, 56,76% da base original. O conjunto de exemplos gerado pelo método descrito anteriormente para todos os usuários possui 98653 padrões, ao passo que considerando apenas os indivíduos que visitaram de 2 a 5 páginas obtém-se 54784 padrões.

A abordagem usada para gerar o valor de saída (classe) de cada instância do conjunto de exemplos implica que o classificador obtido utiliza o maior número de informações possíveis para determinar qual o próximo interesse do usuário. Assim, quanto mais se explora o espaço de navegação, melhor deverão ser as predições, pois o sistema possui uma quantidade maior de informações. Uma outra abordagem consiste em gerar um exemplo para cada usuário com k páginas visitadas ($k \geq 2$), de modo que cada página é determinada pela demais, uma a uma. Desse modo, do conjunto V de páginas visitadas por um indivíduo ($V = \{I_1, I_2, \dots, I_k\}$), a página I_p ($1 \leq p \leq k$) é determinada pelas demais páginas restantes I_q individualmente, tal que $I_q \in V - I_p$. Em outras palavras, I_p é a classe dos exemplos gerados para cada página em $V - I_p$. Esta abordagem será denominada um-para-um ao longo desta dissertação.

No caso desta segunda abordagem, o classificador é capaz de sugerir ligações supostamente mais significativas mesmo que o usuário não tenha explorado consideravelmente o espaço de navegação. No entanto, o número de exemplos gerados para cada usuário torna-se excessivamente grande, pois para cada uma das k páginas visitadas, são produzidos $k \cdot (k - 1)$ exemplos. O conjunto de exemplos produzido por esta abordagem, considerando todos os 32710 usuários, possui 869288 padrões, inviabilizando o treinamento dos algoritmos de AM.

Em termos práticos, os classificadores devem ser capazes de gerar boas predições para ambas as abordagens. A diferença é que uma aplicação que utiliza um classificador com a primeira alternativa tenderá a melhorar a qualidade da predição à medida em que mais informações a respeito da navegação atuais dos usuários estiverem disponíveis. Enquanto que a segunda alternativa deverá realizar boas predições desde os primeiros atos exploratórios no espaço de navegação. Isto acontece porque, o classificador precisaria, em tese, de apenas uma seleção de ligação do usuário para sugerir um conjunto de páginas a serem seguidas.

A Tabela 4.1 mostra um resumo dos conjuntos de exemplos obtidos para se-

Tabela 4.1: Resumo da descrição da base de dados

Base	#Atributos	#Instâncias	#Classes	Erro majoritário	Valores ausentes
MSWeb-0	298	869288	275	-	não
MSWeb-1	298	98653	275	89,02%	não
MSWeb-2	298	54784	254	86,27%	não
MSWeb-3	123	53858	123	86,54%	não

rem utilizadas neste trabalho. Nela aparecem, para cada conjunto de exemplos, o número de características (atributos), o número de instâncias, a quantidade de classes, o erro majoritário e a existência de valores ausentes. Para obter a base MSWeb-0 foi aplicada a abordagem uma-para-uma sobre todos os usuários da base original. O conjunto de exemplos MSWeb-1 possui padrões rotulados para usuários que visitaram pelo menos 2 páginas gerados pela abordagem muitas-para-um. Os conjuntos MSWeb-2 e MSWeb-3 apresentam instâncias classificadas somente para usuários que visitaram de 2 a 5 páginas com a abordagem muitas-para-um. A diferença é que nesta última, foram extraídos todos os padrões rotulados com a classe com distribuição menor que 30.

4.2 Algoritmos de AM

Depois de selecionar e pré-processar a base de dados MSWeb conforme descrito na Seção 4.1, foram selecionados os algoritmos de AM de diferentes paradigmas para terem seu desempenho comparados. Do paradigma estatístico são utilizados o classificador Bayesiano e o SVM, a rede MLP representa o paradigma conexionista e a correlação de Pearson o paradigma baseado em instâncias.

O classificador Bayesiano foi escrito em Java utilizando a biblioteca WEKA que implementa diversos algoritmos de AM e Mineração de Dados (Witten & Frank, 1999). A classe das instâncias para treinamento do classificador Bayesiano é do tipo nominal. Para os experimentos com os algoritmos SVM e a rede Neural MLP optou-se pela utilização, respectivamente, dos aplicativos SVMTorch (Collobert & Bengio, 2001) e o simulador SNNS - *Stuttgart Neural Network Simulator*².

²disponível na URL <http://www-ra.informatik.uni-tuebingen.de/SNNS/>.

A correlação de Pearson foi implementada em Perl e, posteriormente, convertida para linguagem C por motivos de eficiência na execução e na *bufferização* dos dados. Alguns ajustes e alterações no algoritmo descrito na Seção 2.4 foram realizados para gerar uma classificação, de maneira que o seu desempenho pudesse ser comparados com os demais algoritmos. Assim, para efeitos de classificação e comparação com outros indutores com a metodologia *k-fold cross validation*, considera-se cada exemplo que pertence à partição de validação ou teste como usuário ativo, enquanto a base de dados colaborativa é formada pelo conjunto de treinamento. O usuário ativo (a) é representado pelos exemplos existente na partição usada para testar o conjunto de treinamento, cada exemplo indica os interesses de um usuário. Pode-se estabelecer um critério para a classificação baseado na predição do interesse do usuário ativo em relação à uma página j ($p_{a,j}$). Assim, quando $p_{a,j} < \delta$, então a página não possui força suficiente para ser recomendada e é classificada como não interessante. Por outro lado, se $p_{a,j} \geq \delta$, então a página deve ser do interesse para o usuário. A taxa de erro, neste caso, é calculada em relação à classe que aparece no exemplo corrente. Uma página classificada como interessante que não combina com a classe do exemplo corrente é computada como uma classificação incorreta.

4.3 Critérios de Avaliação

Para atingir os objetivos desta dissertação, foi realizada uma série de experimentos e uma avaliação comparativa do desempenho de vários algoritmos de AM quanto à predição de ligações que possam ser de interesse do usuário. Portanto, pretende-se avaliar o desempenho dos algoritmos e não a utilidade da ligação sugerida. A produção de suposições sobre as ligações que combinam com os interesses do usuário, isto é, a adaptação da estrutura Hiperímídia, pode ser entendida como um problema de classificação. Assim, o caminho que se percorre para chegar a uma página, independentemente da ordem em que seja percorrido, é usado para classificá-la.

Para avaliar os algoritmos foi aplicada a metodologia *10-fold cross validation* para estimar o erro verdadeiro de cada indutor (Mitchell, 1997). Os resultados de cada algoritmo são comparados para determinar aquele que possui melhor desempenho. Para a aplicação do *10-fold cross validation*, foram criados programas para embaralhar os exemplos, dividir o conjunto de exemplos em 10 partições disjuntas

e gerar a sequência de experimentos. Esta última consiste em selecionar, aleatoriamente, das dez partições obtidas uma para validação, outra para teste e as oito partições restantes são usadas para treinamento dos indutores. Os diferentes algoritmos de AM são aplicados sobre os mesmos conjuntos de treinamento, validação e teste para que os resultados colhidos possam ser comparados.

4.4 Considerações Finais

Este Capítulo descreve a metodologia adotada para obter conjuntos de exemplos apropriados, bem como para conduzir os experimentos. A base de dados original é apresentada, bem como todo o pré-processamento feito para transformar as informações em conjuntos de exemplos rotulados passíveis de serem usadas por algoritmos de AM. Algumas considerações a respeito da metodologia são discutidas e hipóteses são consideradas. É feito um resumo das principais características do conjunto de exemplos obtido para os experimentos. Breves comentários sobre a implementação e o uso de ferramentas de AM aparecem neste Capítulo, bem como os critérios e a metodologia de avaliação do desempenho escolhida para comparar os algoritmos.

O Capítulo seguinte descreve o ambiente disponível para a condução dos experimentos, bem como as dificuldades encontradas durante sua realização e as soluções encontradas. Os resultados obtidos na execução de cada algoritmo de AM são exibidos e analisados em relação ao seu desempenho.

Capítulo 5

Realização dos Experimentos

Para a realização dos experimentos foram utilizados, inicialmente, os computadores do Laboratório de Inteligência Computacional do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo em São Carlos (LABIC-ICMC-USP/São Carlos). À partir de abril de 2003, um *cluster* de 12 máquinas com processador Xeon de 2.9 GHz foi colocado à disposição para a realização de pesquisas por professores e pesquisadores ligados ao LABIC. Também foram utilizados alguns computadores do Departamento de Engenharia de Computação da Universidade Católica Dom Bosco (UCDB). A configuração é praticamente a mesma para todas as máquinas, com exceção do cluster: processador Pentium III ou similar com *clock* de 950 MHz à 1 GHz, 256 Mb de memória RAM e 40 Gb de disco rígido. A plataforma operacional de todos os computadores é o Linux.

5.1 Aplicação

Depois de selecionar e pré-processar a base de dados e de implementar os algoritmos de AM que seriam aplicados, conforme aparecem nas Seções 4.1 e 4.2, os experimentos foram iniciados.

A aplicação do classificador bayesiano sobre a base MSWeb-0, que foi gerada com a abordagem uma-para-uma considerando todos os usuários da base original, demorou cerca de 32 dias para treinar utilizando processadores Pentium III 95 Mhz e 256 Mb de memória RAM. Os experimentos com o aplicativo SVMTorch e o simulador SNNS, por sua vez, executavam por algumas semanas no cluster e eram

interrompidos antes de terminar o processamento completo, talvez pelo tamanho demasiadamente grande da base de dados. Tentou-se utilizar outros aplicativos para rodar os experimentos, mas todas as tentativas fracassaram, ocasionando um atraso considerável.

Portanto, dois eram os problemas principais encontrados na realização dos experimentos com a base MSWeb-0. O primeiro deles era o tempo de treinamento. O outro consistia na impossibilidade de aplicar os algoritmos SVM e MLP e na inviabilidade de implementá-los, pois implicaria em um atraso ainda maior. O tempo de processamento poderia ser reduzido com a utilização de máquinas com maior capacidade de processamento. Ainda assim, os experimentos com SVMTorch e SNNS não chegavam ao final e, portanto, foi necessário resolver o problema de outra maneira.

Com o tempo, pensou-se em outras alternativas para que os algoritmos SVM e MLP pudessem ser aplicados. Por esta razão, os resultados dos experimentos com a base MSWeb-0 foram desconsiderados neste trabalho, de modo que uma análise maior pudesse ser feita sobre as bases restantes.

A análise feita sobre o processo de criação das bases de dados para o aprendizado de algoritmos supervisionados discutida na Seção 4.1, conduziu à opção de diminuir o tamanho da base de dados. Assim, o conjunto de exemplos MSWeb-1 foi obtido pela aplicação da abordagem muitos-para-um. Para uma análise mais detalhada, decidiu-se explorar mais os resultados com ligeiras modificações no conjunto de dados. Primeiramente restringindo a base apenas para padrões de usuários que visitaram de 2 a 5 páginas, resultando no conjunto de dados MSWeb-2. Posteriormente, novos experimentos foram realizados sobre a base MSWeb-3 com uma restrição adicional: eliminar os padrões rotulados com as classes com menos de 30 acessos.

5.2 Resultados dos Experimentos

A Tabela 5.1 exibe um resumo do erro de classificação para os classificadores obtidos nos experimentos sobre cada um dos 10 conjuntos de treinamentos, validação e teste para os conjuntos de exemplos MSWeb-1, MSWeb-2 e MSWeb-3 aplicando-se a metodologia *10-fold cross-validation*, conforme descrito no Capítulo 4. Os algoritmos de AM usados foram a Correlação de Pearson (PEARSON), o Classificador

Tabela 5.1: Taxa de erro dos classificadores em cada base.

Base	PEARSON	BAYES	SVM
MSWeb-1	-	76,48±0,005	84,42±7,412
MSWeb-2	86,27±0,006	63,71±0,005	78,98±0,016
MSWeb-3	85,68±0,005	63,23±0,006	83,79±0,030

Bayesiano (BAYES), a rede neural Perceptron Multicamadas (MLP) e as Máquinas de Vetores de Suporte (SVM). Os resultados em negrito indicam o classificador com melhor desempenho em cada base de dados.

O algoritmo PEARSON classifica as páginas considerando $p_{a,j} > \alpha$. Foram realizados experimentos com valores 0, 0,34, 0,49 para α . Os melhores resultados consideram $\alpha = 0$. Para os experimentos com SVM, foram utilizados três tipos de kernel com os parâmetros *default*: linear, polinomial e rbf (*radial basis function*). O treinamento com kernel rbf foi o mais rápido, entretanto, os melhores resultados registrados foram para o kernel linear, apesar do tempo de treinamento ser maior. Com as redes MLP foram feitos experimentos com uma e duas camadas intermediárias, sendo que aquela apresentou melhores resultados. Foram considerados 10, 20, 25 e 50 nodos na camada intermediária, sendo que as redes MLP com 25 nodos na camada intermediária apresentou os melhores resultados. A rede foi treinada com taxa de aprendizado $\eta = 0.2$. Entretanto, uma análise dos erros de classificação com a rede MLP revelam um número muito elevado de padrões desconhecidos (acima de 80%). Assim, os resultados com MLP foram desconsiderados nesta dissertação.

Nos experimentos, verificou-se para que dados cada uma das técnicas apresenta classificações corretas. A análise destes dados revela se há coincidências nos exemplos que são classificados corretamente pelas diferentes técnicas. Assim, as Tabelas 5.2 e 5.3, mostram, respectivamente, as cinco classes que obtiveram melhor índice de acertos para cada classificador aplicado sobre as bases MSWeb-2 e MSWeb-3. Nestas Tabelas, aparecem de cima para baixo, para os classificadores PEARSON, BAYES e SVM, as classes com maior índice de acertos. O número entre parênteses indica a frequência (em número de *folds*) em que a classe aparece com o mesmo índice de acertos aplicando-se *10-fold cross validation*. Estes dados foram obtidos das Tabelas 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9.

Tabela 5.2: As cinco melhores classes para cada classificador em MSWeb-2.

PEARSON	BAYES	SVM
5 (10)	5 (10)	4 (3)
6 (10)	6 (10)	7 (5)
8 (10)	11 (10)	10 (5)
11 (10)	12 (10)	14 (5)
12 (10)	13 (10)	16 (2)

Tabela 5.3: As cinco melhores classes para cada indutor em MSWeb-3.

PEARSON	BAYES	SVM
5 (10)	5 (10)	6 (6)
7 (10)	10 (10)	19 (6)
10 (10)	11 (10)	24 (5)
11 (10)	13 (10)	30 (4)
13 (10)	17 (10)	33 (6)

5.3 Análise dos Resultados

Todos os classificadores obtiveram desempenho abaixo do erro majoritário dos conjuntos de exemplos (Seção 4.1), indicando que o problema de adaptação de estrutura em HA pode ser encarado como um problema de classificação. Assim, podem ser construídos motores adaptativos cujo modelo de usuário seja um classificador.

Os algoritmos BAYES e a SVM apresentaram os melhores resultados de uma maneira geral, sendo que o primeiro supera em todos os experimentos o desempenho do segundo. Ambas são técnicas representativas do paradigma estatístico de AM que demonstrou superar significativamente o paradigma baseado em instâncias (PEARSON) tradicionalmente usado em FC. A abordagem conexionista representada pela rede MLP necessita de maior tempo para que seja melhor analisada e possa ser comparada com os indutores do paradigma estatístico.

As restrições aplicadas à abordagem muitas-para-uma não afetaram o desempenho dos algoritmos de AM, exceto para SVM. Os resultados da Tabela 5.1 mostram que o desempenho dos algoritmos PEARSON e BAYES melhoraram em alguns pontos percentuais, confirmando a hipótese levantada de que os usuários que visitaram mais de 5 páginas não contribuem eficazmente para a tarefa de classificação, como

Tabela 5.4: Classes corretas para o classificador Bayesiano em MSWeb-2.

Fold 1	5	6	11	12	13
Fold 2	5	6	11	12	13
Fold 3	5	6	11	12	13
Fold 4	5	6	11	12	13
Fold 5	5	6	11	12	13
Fold 6	5	6	11	12	13
Fold 7	5	6	11	12	13
Fold 8	5	6	11	12	13
Fold 9	5	6	11	12	13
Fold 10	5	6	11	12	13

Tabela 5.5: Classes corretas para o classificador Bayesiano em MSWeb-3.

Fold 1	5	10	11	13	17
Fold 2	5	10	11	13	17
Fold 3	5	10	11	13	17
Fold 4	5	10	11	13	17
Fold 5	5	10	11	13	17
Fold 6	5	10	11	13	17
Fold 7	5	10	11	13	17
Fold 8	5	10	11	13	17
Fold 9	5	10	11	13	17
Fold 10	5	10	11	13	17

mostra a Seção 4.1. Provavelmente os indivíduos que visitam mais de 5 páginas devem estar perdidos no hiperespaço e ao invés de contribuírem positivamente para a classificação, estavam, na realidade, contribuindo negativamente.

As Tabelas 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9 mostram, respectivamente, as cinco classes com melhor índice de classificação correta em cada partição para os algoritmos PEARSON, BAYES e SVM sobre as bases MSWeb-2 e MSWeb-3. Cada linha das Tabelas indica as cinco classes com melhor aproveitamento (classificações corretas) para cada *fold* separadamente.

A análise das Tabelas revela que não há variação na classificação correta de novos padrões em cada partição para os classificadores BAYES e PEARSON. Isto é o resultado da estabilidade do classificador, que apresenta baixo desvio-padrão. Além disto, as classes de novos padrões corretamente rotuladas por BAYES e PEARSON

Tabela 5.6: Classes corretas para SVM em MSWeb-2.

Fold 1	3	7	9	10	14
Fold 2	2	3	7	8	9
Fold 3	10	14	18	20	21
Fold 4	4	7	10	14	20
Fold 5	2	4	10	14	18
Fold 6	4	7	10	14	16
Fold 7	3	7	10	14	17
Fold 8	7	10	14	16	18
Fold 9	5	6	11	12	13
Fold 10	4	7	10	14	16

Tabela 5.7: Classes corretas para SVM em MSWeb-3.

Fold 1	6	15	30	32	33
Fold 2	19	32	38	48	110
Fold 3	6	19	24	30	33
Fold 4	6	19	24	30	33
Fold 5	8	9	14	21	25
Fold 6	6	19	24	30	33
Fold 7	15	19	30	32	33
Fold 8	6	19	24	30	33
Fold 9	19	32	38	48	120
Fold 10	6	19	24	33	37

são bastante coincidentes na base MSWeb-3. Já o algoritmo SVM apresenta uma variabilidade maior na classificação correta de novos padrões na base MSWeb-3, justamente onde apresenta uma taxa de erro e um desvio-padrão maiores do que os resultados da base MSWeb-2. Isto leva a crer que a segunda restrição, aquela na qual os padrões cujo rótulo sejam classes tenham uma distribuição inferior a 30 são retirados da base, apesar de implicar em menor esparsidade dos dados, influencia de maneira negativa no treinamento do algoritmo SVM. Entretanto, os resultados indicam uma melhoria no desempenho para os classificadores PEARSON e BAYES sobre esta mesma base (MSWeb-3). Estes resultados sugerem que os classificadores PEARSON e BAYES são capazes de lidar bem tanto com dados esparsos quanto não esparsos, ao passo que o classificador SVM apresenta melhor rendimento com dados esparsos do que com dados não esparsos.

Tabela 5.8: Classes corretas para PEARSON em MSWeb-2.

Fold 1	5	6	8	11	12
Fold 2	5	6	8	11	12
Fold 3	5	6	8	11	12
Fold 4	5	6	8	11	12
Fold 5	5	6	8	11	12
Fold 6	5	6	8	11	12
Fold 7	5	6	8	11	12
Fold 8	5	6	8	11	12
Fold 9	5	6	8	11	12
Fold 10	5	6	8	11	12

Tabela 5.9: Classes corretas para PEARSON em MSWeb-3.

Fold 1	5	7	10	11	13
Fold 2	5	7	10	11	13
Fold 3	5	7	10	11	13
Fold 4	5	7	10	11	13
Fold 5	5	7	10	11	13
Fold 6	5	7	10	11	13
Fold 7	5	7	10	11	13
Fold 8	6	7	10	11	13
Fold 9	5	7	10	11	13
Fold 10	5	7	10	11	13

5.4 Considerações Finais

Este Capítulo abordou a realização dos experimentos com os algoritmos BAYES, PEARSON e SVM sobre os conjuntos de exemplos obtidos no Capítulo 4. Foram relatadas as dificuldades encontradas na realização dos experimentos, bem como as soluções encontradas e a análise dos resultados obtidos e das hipóteses levantadas no Capítulo anterior. O próximo Capítulo sintetiza os principais tópicos deste trabalho, além de indicar novas possibilidades de investigação na adaptação de estruturas em HA.

Capítulo 6

Conclusão

O objetivo deste trabalho foi avaliar o desempenho de algoritmos de diferentes AM para gerar suposições a respeito dos possíveis interesses dos usuários que navegam WWW para serem utilizados em sistemas e aplicações de HA. Estas suposições podem ser usadas de diversas maneiras para adaptar a estrutura da Aplicação Hiperfídia, de modo a indicar para o usuário ativo, quais ligações que são possivelmente úteis. Para tanto, foram abordados tópicos de Hiperfídia, IA e MU úteis para o desenvolvimento da HA.

O problema de adaptar as ligações de uma Aplicação Hiperfídia é endereçado, nesta dissertação, com um problema de classificação. A idéia é obter informações a respeito da interação dos indivíduos com o sistema. No caso, foram utilizadas as informações de navegação dos usuários no *website* da Microsoft. Os dados de interação consistem na sequência de páginas visitadas pelos usuários. Considera-se que os usuários pretendam atingir certos objetivos ao navegarem por um *website*. Com isto, foi possível produzir bases de dados com padrões rotulados para a aplicação de algoritmos supervisionados de AM. Todas estas informações determinaram a maneira como as informações brutas foram pré-processadas para obter diferentes conjuntos de exemplos. Estes conjuntos foram utilizados para treinamento de diversos algoritmos de AM e servem como o modelo de usuário para um motor de inferência de um SHA possa decidir como adaptar as ligações que aparecem em uma página.

A saída dos classificadores pode ser aplicada para implementar uma combinação de métodos e técnicas de adaptação de conteúdos e ligações da HA que aparecem na Subseção 3.3.3. A entrada para o classificador consiste nas páginas percorridas

pelo usuário corrente. A saída deve ser a classe (página) com maiores chances de ser do seu interesse. O usuário não pode ficar restrito apenas a esta ligação, por isto a técnica de anotação de ligação (*Annotation*) mostra-se bastante útil, indicando qual das ligações disponíveis é aquela sugerida pelo sistema. Outra possibilidade consiste em considerar não apenas a classe de saída do classificador, mas sim o conjunto de classes que possuem uma forte correlação com a trilha percorrida pelo usuário até o momento. Ambas as abordagens podem ser utilizadas para os algoritmos de AM nos paradigmas baseado em instâncias, estatísticos, baseado em distribuição de probabilidades e RNA. O usuário sempre tem a liberdade de decidir pela seleção ou não da ligação sugerida pelo sistema.

Este trabalho mostra que o uso de AM para a adaptação é possível desde que sejam utilizadas as técnicas mais adequadas. Além disto, a maneira como o conjunto de exemplos é gerado também importa, como discutido no Capítulo anterior. Os resultados de cada classificador sobre bases de dados com diferentes características indicam a melhor abordagem para gerar o conjunto de exemplos. Esta dissertação apresenta duas abordagens diferentes com pequenas variações. Provavelmente resultados diferentes poderiam ser obtidos com a aplicação de metodologias distintas.

O paradigma de AM estatística, em especial o classificador Bayesiano, apresentou resultados melhores que o paradigma baseado em instâncias. Outras abordagens podem ser elaboradas tanto para a obtenção de uma base de dados mais adequada conforme as necessidades e aspirações, quanto na investigação e comparação de novas técnicas de AM. Trabalhos futuros podem considerar:

1. a aplicação de abordagens distintas para a criação do conjunto de exemplos, bem como a realização de experimentos com outros algoritmos de AM com o objetivo de obter melhores classificadores;
2. a investigação de novas metodologias para a aplicação de AM, como, por exemplo, a utilização de aprendizado incremental. Desta forma, o modelo de usuário seria constantemente atualizado, bastando definir os critérios para isto;
3. a avaliação da qualidade da predição das páginas através de experimentos laboratoriais e comparativos entre a interface Hipermídia estática e a interface adaptativa através de métodos e técnicas adequadas de IHC;

4. o desenvolvimento de aplicativos que consideram a metodologia descrita nesta dissertação para diversos setores, como a educação e o comércio, mostrando a viabilidade da utilização do AM em adaptação de estruturas; e
5. a pesquisa de sistemas híbridos que aproveitam as vantagens de técnicas variadas de IA para tentar melhorar o desempenho e a qualidade da predição e sua qualidade, isto é, sua adequação aos interesses e objetivos dos usuários.

Referências Bibliográficas

- (Alatalo & Peräaho, 2001) Alatalo, T.; Peräaho, J. *A Modelling Method for Designing Adaptive Hypermedia*. In: Eight International Conference on User Modeling (UM2001), Sonthofen, Alemanha, 13-17 Julho, Posters Section, 2001.
- (Ambrosini et al., 1997) Ambrosini, L.; Cirillo, V.; Micarelli, A. *A Hybrid Architecture for User-Adapted Information Filtering on the World Wide Web*. Proceedings of The 6th International Conference on User Modeling (UM-97) Jameson, A.; Paris, C.; and Tasso, C. (eds.), New York: Springer Wien New York, pp. 59-61, 1997.
- (Armstrong et al., 1995) Armstrong, R.; Freitag, D.; Joachims, T.; Mitchell, T. *WebWatcher: A Learning Apprentice for the World Wide Web*. In: (AAAI) Spring Symposium on Information Gathering, pp. 6-12, 1995.
- (Balabanovic et al., 1995) Balabanovic, M.; Shoham, Y.; Yun, Y. *An Adaptive Agent for Automated Web Browsing*. Journal of Visual Communication and Image Representation, Vol. 6, No. 4, 1995.
- (Baranauskas & Monard, 2000) Baranauskas, J. A.; Monard, M. C. *Revising Some Machine Language Concepts and Methods*. Relatório Técnico No. 102, ICMC/USP, São Carlos, Fevereiro, 2000.
- (Bodner & Chignell, 1999) Bodner, R.; Chignell, M. *Dynamic Hypertext: Querying and Linking*. ACM Computing Surveys, Vol. 31, No. 4, December, 1999.
- (Bongio et al., 2000) Bongio, A.; Fraternali, P.; Ceri, S. *Web Modeling Language (WebML): a Modeling Language for Designing Websites*. Journal of Computer Networks, Vol. 33, No. 1-6, pp. 137-157, Amsterdam, Netherlands, 2000.
- (Braga et al., 2000) Braga, A. P.; Ludermir, T. B.; Carvalho, A. C. P. L. F. *Redes Neurais Artificiais - Teoria e Aplicações*. LTC, Rio de Janeiro, 2000.

-
- (Breese et al. 1998) Breese, J.; Heckerman, D.; Kadie, C. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publisher, pp. 43-52, Madison, WI, July, 1998.
- (Brookshear, 2001) Brookshear, J. G. *Ciência da Computação - Uma Visão Abrangente*. Bookman, Porto Alegre, 2000.
- (Brusilovsky, 1995) Brusilovsky, P. *Intelligent Tutoring Systems for World-Wide Web. Proceedings of Third International WWW Conference, Darmstadt, pp. 42-45, 1995.*
- (Brusilovsky, 1996a) Brusilovsky, P. *Methods and Techniques of Adaptive Hypermedia*. User Modeling and User-Adapted Interaction 6, Kluwer Academic Publishers, pp. 87-129, 1996.
- (Brusilovsky, 1996b) Brusilovsky, P. *Adaptive Hypermedia: an Attempt to Analyze and Generalize*. P. Brusilovsky, P. Kommers and N. Streitz (eds.), Multimedia, Hypermedia, and Virtual Reality - Lecture Notes in Computer Science, Vol. 1077. Berlin: Springer-Verlag, pp. 288-304, 1996.
- (Brusilovsky, 2000) Brusilovsky, P. *Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education*. G. Gauthier, C. Frasson and K. VanLehn (eds.) Intelligent Tutoring Systems - Lecture Notes in Computer Science, Berlin: Spring Verlag, Vol. 1839, pp. 1-7, 2000.
- (Brusilovsky, 2001) Brusilovsky, P. *Adaptive Hypermedia*. User Modeling and User-Adapted Interaction 11, pp. 87-110, Netherlands: Kluwer Academic Publisher, 2001.
- (Brusilovsky & Maybury, 2002) Brusilovsky, P.; Maybury, M. (2002) *From Adaptive Hypermedia to the Adaptive Web*. Communications of the ACM, Maio, 2002, Vol. 45, No. 5, pp. 31-33.
- (Brusilovsky et al., 1996) Brusilovsky, P.; Schwarz, E.; Weber, G. *A Tool for Developing Adaptive Electronic Textbooks on WWW*. Proceedings of the WebNet'96 Conference, pp. 64-69, San Francisco, 1996.
- (Calvi & De Bra, 1998) Calvi, L.; De Bra, P. *AHA: An Open Adaptive Hypermedia Architecture*. Proceedings of the 2nd Workshop on Adaptive Hypertext and

- Hypermedia (HYPERTEXT'98), Pittsburgh, USA, 20-24 Junho, pp. 5-11, 1998.
- (Campbell & Goodman, 1987) Campbell, B.; Goodman, J. M. *HAM: A General Purpose Hypertext Abstract Machine*. Proceedings of the Hypertext'87 (ACM), November 13-15, pp. 21-32, Chapel Hill, North Carolina, USA, 1987.
- (Cannataro & Pugliese, 2001) Cannataro, M.; Pugliese, A. *XAHM: An XML-Based Adaptative Hypermedia Model and Its Implementation*. In: Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext'01). Arhus, Denmark, August 14-18, 2001. <http://wwwis.win.tue.nl/ah2001/papers/cannataro.pdf>
- (CERN) (Carpenter & Grossberg, 1987) Carpenter, G. A.; Grossberg, S. *A Massively Parallel Architecture for a Selforganizing Neural Pattern Recognition Machine*. Computer Vision, Graphics and Image Processing, Vol. 37, pp. 54-115, 1987.
- (CERN) <http://public.web.cern.ch/Public/ACHIEVEMENTS/web.html>.
- (Collobert & Bengio, 2001) Collobert, R.; Bengio, S. *SVM-Torch: Support Vector Machines for Large-Scale Regression Problems*. Journal of Machine Learning Research, Vol. 1, pp. 143-160, 2001.
- (Conklin, 1987) Conklin, J. *A Survey of Hypertext*. Technical Report MCC Number STP-356-86, Rev. 2, Software Technology Program, (December 3, 1987).
- (De Bra, 1999) De Bra, P. *Design Issues in Adaptive Web-Site Development*. Proceedings of 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 1999. <http://wwwis.win.tue.nl/asum99/debra/debra.html>.
- (De Bra et al., 1992) De Bra, P.; Houben, G-J.; Kornatzky, Y. *An Extensible Data-Model for Hyperdocuments*. In: 4th ACM Conference on Hypertext, Milan, December, 1992, pp. 222-231.
- (De Bra et al., 1999a) De Bra, P.; Houben, G-J.; Wu, H. *Authoring Support for Adaptive Hypermedia Applications*. Proceedings of ED-MEDIA '99, Seattle, pp. 364-369, 1999.
- (De Bra et al., 1999b) De Bra, P.; Houben, G-J.; Wu, H. *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*. Proceedings of ACM Hypertext '99 Conference, Darmstadt, Alemanha, pp. 147-156, 1999.
- (De Bra et al., 2000) De Bra, P.; Houben, G-J.; Wu, H. *Supporting User Adaptation in Adaptive Hypermedia Applications*. Online Conference and

-
- Informatiewetenschap, Rotterdam, 2000. Disponível no endereço Internet <http://wwwis.win.tue.nl/houben/respub/infwet00.pdf>.
- (De Troyer & Casteleyn, 2003) De Troyer, O.; Casteleyn, S. (2003) *Exploiting Link Types During the Conceptual Design of Websites*. International Journal of Web Engineering Technology, Vol. 1, No. 1, pp. 17-40.
- (Dumais, 1998) Dumais, S. *Using SVM for Text Categorization*. IEEE Intelligent Systems, pp. 21-23, Julho/Agosto 1998.
- (Engelbart, 1962) Engelbart, D. C. *Augmenting Human Intellect: A Conceptual Framework*. Summary Report Contract AF 49 638 1024 SRI Project 3578 Stanford Research Institute Menlo Park, California, 1962.
- (Engelbart, 1968) Engelbart, D. C. *A Research Center for Augmenting of Man's Intellect*. Proceedings of 1968 FJCC, 33, Part 1, Montvale, N.J.: AFISP Press, 1968.
- (Furuta & Stotts, 1990) Furuta, R.; Stotts, P. D. *The Trellis Hypertext Reference Model*. Proceedings of the Hypertext Standardization Workshop, National Institute of Standards and Technology, pp. 83-93, 1990.
- (Friedman et al., 1997) Friedman, N.; Geiger, D.; Goldszmidt, M. (1997) *Bayesian Network Classifiers*. Machine Learning Journal, Vol. 29, No. 2-3, pp. 131-163.
- (Guelich et al., 2001) Guelich, S.; Gundavaram, S.; Birznieks, G. *Programação CGI com Perl*. Rio de Janeiro: Editora Ciência Moderna, 2001.
- (Haykin, 1999) Haykin, S. *Neural Networks - A Comprehensive Foundation*, Prentice Hall, 1999.
- (Halasz, 1987) Halasz, F. *Reflections on NoteCard: Seven Issues for the Next Generation of Hypermedia Systems*. Communications of ACM (CACM), Vol. 31 (7), pp. 836-852, 1987.
- (Halasz & Schwartz, 1990) Halasz, F.; Schwartz, M. *The Dexter Reference Model*. In Proceedings of NIST Hypertext Standardization Workshop, pp. 95-133, 1990.
- (Hearst et al., 1998) Hearst, M. A.; Schölkopf, B.; Dumais, S.; Osuna, E.; Platt, J. *Large Margin Rank Boundaries for Ordinal Regression*. In: Smola et al. (1999), pp. 29-45, 1998.

- (Hettich & Bay, 1999) Hettich, S.; Bay, S. D. (1999). The UCI KDD Archive. <http://kdd.ics.uci.edu>. Irvine, CA: University of California, Department of Information and Computer Science.
- (Henze & Nejd, 2000) Henze, N.; Nejd, W. *Extendible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material*. Adaptive Hypermedia and Adaptive Web-based Systems, P. Brusilovsky et al (eds), Springer, 2000, pp. 109-120, 2000.
- (Isakowitz et al., 1995) Isakowitz, T.; Stohr, Edward A.; Balasubramanian, P. *RMM: a Methodology for Structures Hypermedia Design*. Communications of ACM, Vol. 38, No. 8, pp. 34-44, Agosto, 1995.
- (Kappel et al., 2003) Kappel, G.; Pröll, B.; Retschitzegger, W.; Schwinger, W. (2003) *Customization for Ubiquitous Web Applications - a Comparison of Approaches*. International Journal of Web Engineering Technology, Vol. 1, No. 1, pp. 79-111.
- (Kobsa, 1993) Kobsa, A. *User Modeling: Recent Work, Prospects and Hazards*. H. Schneider Hufschmidt, T. Kuhme and U. Malinowski (eds.): Adaptive Users Interfaces: Principles and Practice. Amsterdam, North-Holland, pp. 111-123, 1993.
- (Kobsa, 1994) Kobsa, A. *User Modeling and User-Adapted Interaction*. CHI'94 Tutorial Notes. URL: <http://www.acm.org/sichi/chivas/kobsa.html>.
- (Kobsa et al., 1999) Kobsa, A.; Koenemann, J.; Pohl, W. *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*. Technical Report No. 66 GMD, German National Research Center for Information Technology, St. Augustin, Alemanha, 1999.
- (Koch & Wirsing, 2001) Koch, N.; Wirsing, M. *Software Engineering for Adaptive Hypermedia Applications?* 20th ACM Conference on Hypertext and Hypermedia (HYPERTEXT'01), Arhus, Dinamarca, 14-18 Agosto. <http://wwwis.win.tue.nl/ah2001/papers/koch.pdf>.
- (Lorena & Carvalho, 2003) Lorena, A. C.; Carvalho, A.C.P.L.F. *Introdução às Máquinas de Vetores Suporte (Support Vector Machines)*. Relatório Técnico No. 77, ICMC/USP, São Carlos, Fevereiro, 2003.

-
- (Marietto, 2000) Marietto, M. G. *Definição Dinâmica de Estratégias Institucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW*. Tese de Doutorado, ITA, 2000.
- (Middleton, 2001) Middleton, Stuart E. *Capture Knowledge of User Preferences with Recommender Systems*. Universidade de Southampton, Julho, 2001, Tese de PhD.
- (Mitchell, 1997) Mitchell, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.
- (Monard et al., 1997) Monard, M. C.; Batista, G. E. A. P. A.; Kawamoto, S.; Pugliesi, J. B. Uma Introdução ao Aprendizado Simbólico de Máquina por Exemplos. Notas Didáticas No. 29, ICMC-USP, 1997.
- (Morita & Shinota, 1994) Motira, M.; Shinoda, Y. *Information Filtering Based on User Behaviour Analysis and Best Match Text Retrieval*. Proceedings of the 17th ACM Annual International Conference on Research and Development in Information Retrieval, Dublin, Ireland, pp. 272-281, 1994.
- (Mullier, 1995) Mullier, D. J. *A Hybrid Connectionist/Knowledge-Based Approach to Intelligent Tutoring with Hypermedia*. Acessado em 17/05/2002 na URL: <http://www.lmu.ac.uk/PLYMOUTH.htm>.
- (Mullier, 1999) Mullier, D. J. *The Application of Neural Network and Fuzzy-Logic Techniques to Educational Hypermedia*. Tese de PhD, Inglaterra, 1999.
- (Mullier et al., 1999) Mullier, D. J.; Hobbs, D. J.; Moore, D. J. *A Hybrid Semantic/Connectionist Approach to Adaptivity in Educational Hypermedia Systems*. Proceedings of ED-MEDIA'99, Seattle, WA. <http://www.lmu.ac.uk/ies/comp/staff/dmullier/icai.pdf>.
- (Mullier, 2000) Mullier, D.J. *Examining How Users Interact With Hypermedia Using a Neural Network*. Proceedings of ICAI-00, Las Vegas, 26-29 Junho, 2000. Disponível eletronicamente em http://www.lmu.ac.uk/ies/comp/staff/dmullier/edmedia2000_2.pdf.
- (Mullier et al., 1999b) Mullier, D. J.; Hobbs, D. J.; Moore, D. J. *A Hybrid Semantic/Connectionist Approach to Adaptivity in Educational Hypermedia Systems*. Proceedings of ED-MEDIA'99, Seattle, WA. <http://www.lmu.ac.uk/ies/comp/staff/dmullier/icai.pdf>.

- (Mullier & Dixon, 2000) Mullier, D.J.; Dixon, M.B. *Authoring Educational Hypermedia Using a Semantic Network*. Proceedings of ED-MEDIA 2000, Montreal, Canadá. http://www.lmu.ac.uk/ies/comp/staff/dmullier/edmedia2000_2.pdf.
- (Nelson, 1965) Nelson, T. H. *A File Structure for the Complex, the Changing and the Indeterminate*. Proceeding of 20th ACM National Conference, pp. 84-100, 1965.
- (Nicholas & Rosenberg, 1993) Nicholas, C. K.; Rosenberg, L. H. *Canto: a hypertext data model*. Electronic Publishing, Vol. 6, No. 1, pp. 1-23, March, 1993. Disponível em <http://citeseer.nj.nec.com/nicholas93canto.html>.
- (Nichols, 1997) Nichols, D. *Implicit Rating and Filtering*. Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering, ERCIM (ed.), pp. 31-36, Budapest, Hungria, 1987.
- (Oard & Kim, 1998) Oard, D. W.; Kim, J. *Implicit Feedback for Recommender Systems*. Proceedings of the AAAI Workshop on Recommender Systems, July, 1998. Disponível em <http://citeseer.nj.nec.com/oard98implicit.html>.
- (Olsina, 1998) Olsina, L. (1998). *Building a Web-based Information System Applying the Hypermedia Flexible Process Modeling Strategy*. First International Workshop on Hypermedia Development, HYPERTEXT '98. Disponível em <http://gidis.ing.unlpam.edu.ar/ingles/personas/olsinal/AISWork_HT98.html>.
- (Osuna, 1998) Osuna, E. *Applying SVMs to face detection*. IEEE Intelligent Systems, pp. 23-26, Julho/Agosto 1998.
- (Opperman & Specht, 1998) Opperman, R.; Specht, M. *ACE - Adaptive Courseware Environment*. Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia (HYPERTEXT'98), Pittsburgh, USA, 20-24 Junho, pp. 141-161, 1998.
- (Pohl& Nick, 1999) Pohl, W.; Nick, A. *Machine Learning and Knowledge Representation in the LaboUr Approach to User Modeling*. Proceedings of the 7th International Conference on User Modeling, pp. 179-188, Banff, Canada, 1999.
- (Preece, 1994) Preece, J. *Human Computer Interaction*. Harlow: Addison-Wesley, 1994.
- (Quinlan, 1986) Quinlan, J. R. *Induction of Decision Trees*. Machine Learning, Vol.

-
- 1, No. 1, pp-81-106, 1986.
- (Quinlan, 1994) Quinlan, J. R. *C4.5: Programs for Machine Learning*. Series Machine Learning, New York: Morgan Kaufmann, 1993.
- (Resnick et al., 1994) Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. *Grouplens: An open architecture for collaborative filtering of netnews*. In: Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, pp. 175-186, New York, 1995.
- (Resnick & Varian, 1997) Resnick, P.; Varian, H. (1997). *Recommender Systems*. Communications of ACM, 40(3), pp. 56-58.
- (Rumelhart & McClelland, 1986) Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing*. Vol. 1: Foundations. The MIT Press, 1986 .
- (Rumelhart et al., 1986) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning representations by back-propagating errors*. Nature, 323:533-536, 1986.
- (Russel & Norvig, 1995) Russel, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall, 1995.
- (Schwabe et al., 1993) Schwabe, D.; Garzotto, F.; Paolini, P. *HDM - a Model-based Approach to Hypertext Application Design*. ACM Transactions on Information Systems, Vol. 11, No. 1, pp. 1-26, Janeiro, 1993.
- (Schwabe et al., 1996) Schwabe, D.; Rossi, G.; Barbosa, S. D. J. *Systematic Hypermedia Application Design with OOHD*. Proceedings of the Seventh ACM Conference on Hypertext, pp. 116-128, 1996.
- (Soboroff & Nicholas, 2000) Soboroff, I.; Nicholas, C. *Collaborative Filtering and the Generalized Vector Space Model*. SIGIR, Atenas, 2000.
- (Vapnik & Chervonenkis, 1968) Vapnik, V.; Chervonenkis, A. *On the Uniform Convergence of Relative Frequencies of Events to their Probabilities*. Dokl. Akad. Nauk USSR, No. 181, 1968.
- (Vert, 2001) Vert, J.P. (2001) *Introduction to Support Vector Machines and Applications to Computational Biology*. DRAFT.
- (Witten & Frank, 1999) Witten, I. H.; Frank, E. *Data Mining: Practical Machine Learning Tools with Java Implementations*, Morgan Kaufmann, San Francisco,

1999.

(Wu 2001) Wu, H. (2001). *A Reference Architecture for Designing Adaptive Hypermedia*. 20th ACM Conference on Hypertext and Hypermedia (HYPERTEXT'01), Arhus, Dinamarca, 14-18 Agosto. Disponível no endereço Internet <http://wwwis.win.tue.nl/ah2001/papers/wu.pdf>.