

Recomendação de Recursos utilizando Autômato Adaptativo

P. R. M. Cereda, R. A. Gotardo e S. D. Zorzo

Resumo— Sistemas de Recomendação identificam as preferências de um dado usuário, bem como do restante da comunidade, em relação aos recursos disponibilizados, visando oferecer recursos personalizados aos usuários. Este artigo apresenta um sistema de recomendação que utiliza um autômato adaptativo para analisar a relação entre recursos e usuários e determinar as possíveis recomendações personalizadas. São apresentados os aspectos de implementação e um experimento para verificação e análise das recomendações geradas no sistema desenvolvido.

Palavras-chave:— Sistemas de Recomendação, Autômato Adaptativo, Personalização.

I. INTRODUÇÃO

Com o crescimento e difusão da Internet, a grande maioria dos websites passou a oferecer serviços e recursos personalizados aos usuários. As informações relativas aos seus acessos passaram a ser valiosas para os sites, a ponto de realizarem a coleta destas informações, muitas vezes sem consentimento explícito.

A oferta de serviços e recursos ao usuário é possível com o emprego de diversos métodos. Um deles é o Sistema de Recomendação, que tenta identificar quais são os recursos mais importantes para este. Esses recursos constituem sugestões a serem apresentadas ao usuário, de acordo com um determinado critério.

Este artigo apresenta um sistema de recomendação, denominado *RecomAA*, que utiliza um autômato adaptativo para verificar a existência de alguma relação entre recursos. O autômato adaptativo foi escolhido devido à sua simplicidade, capacidade de modificação autônoma e poder computacional.

A organização deste artigo é a seguinte: a Seção II apresenta as principais vantagens da personalização de serviços e recursos, e as funcionalidades oferecidas aos sites. A Seção III contextualiza os sistemas de recomendação e suas principais características. O sistema proposto, utilizando um autômato adaptativo, é apresentado na Seção IV. A Seção V apresenta um experimento realizado para analisar as recomendações geradas. Os aspectos de implementação são definidos na Seção VI. As conclusões são apresentadas na Seção VII.

II. PERSONALIZAÇÃO DE SERVIÇOS E RECURSOS

A personalização pode ser descrita como tornar algo

peçoal, individual, dependente das características e dos interesses humanos. Ao personalizar um objeto de acordo com um usuário, cria-se uma relação de afinidade. De acordo com Grande [1], “um produto ou serviço pode atender as necessidades fundamentais de uma pessoa por suas funcionalidades e características primárias. Além disso, um serviço, através da personalização, pode possuir determinadas características que o torna mais parecido com um indivíduo. Essas qualidades secundárias são consideradas tão importantes que em muitos casos a escolha do produto ou serviço é regida somente através delas.”

Para utilizar a personalização, algumas informações relevantes necessitam ser obtidas para saber algo a respeito da preferência de um usuário [2].

A personalização apresenta algumas vantagens, tais como:

- **Sistema “mais próximo” do usuário:** o usuário sente-se mais à vontade, pois a maioria das informações que o site retorna o agrada.
- **Perfil diferente para cada usuário:** um site de e-commerce que oferece perfis diferentes para os usuários pode aumentar suas vendas.
- **Tendências de comportamento:** a partir dos perfis dos usuários, é possível prever tendências de comportamento e estabelecer grupos de interesse.
- **Melhorias na navegabilidade:** um site pode prover menus de navegação diferentes, dependendo dos interesses dos usuários.

A personalização pode oferecer a um site algumas funcionalidades. Kimball e Merz [3] citam algumas delas:

- **Reconhecimento de re-visitas:** ter a possibilidade de identificar o usuário como um novo cliente ou um cliente já existente.
- **Interface de usuário e personalização de conteúdo:** o usuário pode ter vários tipos de menus de acesso rápidos, links de produtos, propagandas e anúncios, de acordo com sua preferência.
- **Vendas colaterais:** são recomendações a partir de conhecimentos prévios do cliente.
- **Vendas por impulso:** são sugestões geralmente relacionadas a um determinado grupo de interesse. Por exemplo, se o cliente comprou um determinado livro, o site poderá relatar outros produtos comprados por outros usuários que também compraram o mesmo livro. Frequentemente, esse tipo de personalização tem o título de “*Clientes que compraram este produto também compraram:*”.
- **Filtragem colaborativa ativa:** são sugestões e indicações de clientes. Por exemplo, o cliente tem a

Paulo Roberto Massa Cereda, Reginaldo Aparecido Gotardo e Sérgio Donizetti Zorzo.

Os autores são do Departamento de Computação, Universidade Federal de São Carlos, Rodovia Washington Luís, Km 235, Caixa Postal 676, 13565-905 – SP, (e-mails: paulo_cereda@dc.ufscar.br, reginaldo_gotardo@dc.ufscar.br e zorzo@dc.ufscar.br).

possibilidade de dar nota a um determinado produto, escrever um comentário ou recomendar a um amigo.

- **Eventos de calendário:** os produtos podem ser relacionados de acordo com eventos baseados em feriados ou datas importantes, por exemplo, dia das Mães, Páscoa, Natal, entre outros.
- **Eventos de estilo de vida:** os produtos são relacionados de acordo com situações comuns do dia-a-dia, por exemplo, nascimento de um filho, casamento, formatura, entre outros.
- **Localização:** o conhecimento prévio de um cliente permite a personalização de um site em relação a seu idioma ou localização geográfica.

Há um grande benefício na utilização da personalização por sites, principalmente os de e-commerce. Com isso, diversas técnicas de coleta e ferramentas são criadas ou melhoradas para facilitar a tarefa de prover personalização.

Existem diversos mecanismos para coleta de dados para oferecer personalização na Internet. Geralmente são utilizados análise de dados em formulários e análise de navegação do usuário [4]. Com os dados obtidos, são aplicadas técnicas de mineração de dados para determinar preferências, tendências, entre outros.

É importante ressaltar que os próprios navegadores enviam informações interessantes, como sistema operacional do usuário, idioma, tipo de navegador, a página de referência, e outros, podendo essas informações ser utilizadas para serviços de personalização.

Os mecanismos mais comuns de coleta de dados podem ser implementados por *cookies*, *clickstream*, *logs* de servidor, *web bugs* e formulários, que são detalhados a seguir.

O *cookie* é um grupo de dados trocados entre o usuário e o servidor, armazenados em um arquivo texto criado no computador do primeiro [5]. As informações armazenadas nos *cookies* podem refletir algo sobre o perfil do usuário. O *cookie* tem o objetivo de manter a persistência de sessões HTTP, mantendo o último estado antes de terminar a conexão, e retorná-lo no próximo acesso.

Pelo fato deste mecanismo estar incluído em todos os navegadores (e, na maioria dos casos, ativado por padrão), ele persistirá durante muito tempo como a ferramenta primária para prover personalização [3].

O *clickstream*, também conhecido como seqüência de cliques ou *clickpath*, representa o caminho que o usuário percorre enquanto está visitando um determinado site. A seqüência de cliques obtida, quando aplicadas de forma correta, pode proporcionar informações muito importantes para as empresas [4].

Os dados de *clickstream* também podem ser coletados por Provedores de Serviço de Internet (ISPs), painéis de acesso ou mesmo manualmente, apesar da coleta através dos arquivos de *log* dos servidores ser mais comum [6].

O *clickstream* é muito importante, do ponto de vista comercial, pois é possível identificar as preferências e os padrões de comportamento do usuário; isso inclui qual área lhe interessa, a freqüência que a procura e quais as informações úteis para criar estratégias de marketing mais

direcionadas ao usuário e, conseqüente, maior chance de sucesso [7].

Todos os servidores Web têm a capacidade de registrar as requisições em arquivos de *log* ou banco de dados. Os dados de *log* de um servidor Web são a fonte primária de dados do *clickstream*, pois toda vez que o servidor Web responde uma requisição HTTP, uma entrada é anotada no arquivo de *log*. As entradas do arquivo de *log* seguem um padrão estendido chamado *Extended Common Log Format* (ECLF) [3].

Esse tipo de análise de *log* oferece um desafio: o servidor pode estar, em um determinado momento, mantendo centenas ou até milhares de sessões de usuário simultaneamente. Os registros individuais que abrangem os rastros da sessão de um determinado usuário estarão dispersos por todo o *log* [3].

Outro problema relatado sobre os arquivos de *log* diz respeito aos *Crawlers*, *Bots*, *Spiders* e *Robots*, que são pequenos programas que são executados com a intenção de coletarem dados ao visitarem páginas Web. Esses programas interferem significativamente nos padrões de *clickstream*, poluindo um arquivo de *log* com informações que não foram geradas por um verdadeiro usuário [8].

Web Bugs são mecanismos que tentam obter algum tipo de identificação de um usuário. Geralmente, estão inseridos em mensagens de e-mail ou páginas da Web. Uma página Web ou um e-mail podem ser visualizados sem que se perceba a presença de um *Web Bug*. No momento de carregamento da página Web ou do e-mail, a imagem referente ao *Web Bug* é requisitada a um servidor distinto que, por sua vez, obtém as informações do cabeçalho HTTP da requisição, contendo as informações a respeito do usuário.

Os *Web Bugs* podem trabalhar em conjunto com os *cookies*, monitorando quais sites o usuário visita. Dessa forma, os banners de anúncios são específicos para cada um [9].

A linguagem HTML apresenta controles que permitem enviar informações para um determinado processamento no servidor. Tais informações são preenchidas utilizando formulários eletrônicos; através dos métodos POST ou GET fornecidos pelo protocolo HTTP, o servidor recebe a requisição e realiza o tratamento das informações submetidas.

Através dos formulários, podem ser realizados dois tipos de coleta: a coleta explícita, onde o envio das informações pelo usuário envolve seu consentimento, e a coleta implícita, onde o usuário pode enviar informações adicionais, sem tomar conhecimento da existência delas. Em ambos os casos, as informações são armazenadas no lado do servidor, por exemplo, em um banco de dados.

Os mecanismos de coleta podem ser combinados para um particular sistema de recomendação.

A *Mineração de Dados* é um conjunto de técnicas que visam a aquisição de conhecimentos em bancos de dados. A motivação para esse mecanismo vem da dificuldade em obter conhecimento relevante de grandes volumes de dados, logo, é necessário o uso de ferramentas e técnicas que facilitem essa árdua tarefa [10].

Na Web, a descoberta de informações relevantes de padrões de navegação do usuário, a ponto de descrever seu comportamento, é extremamente valiosa para empresas e

organizações que trabalham com e-commerce.

As técnicas de mineração podem utilizar as mais diversas abordagens de implementação. As mais comuns incluem uso de técnicas estatísticas, raciocínio baseado em casos, redes neurais, árvores de decisão, algoritmos genéticos, entre outros.

A utilização da Mineração de Dados por empresas de e-commerce é imprescindível, pois pode revelar informações valiosas que não podem ser obtidas pelos métodos tradicionais.

Os dados obtidos através dos mecanismos de coleta podem ser analisados utilizando-se mineração de dados para revelar perfis de navegação e comportamento dos usuários.

III. SISTEMAS DE RECOMENDAÇÃO

O *Sistema de Recomendação* é um método tradicional para retornar informações relevantes ao usuário. De um modo geral, um sistema de recomendação tenta identificar quais são os recursos mais importantes para um determinado usuário e recomendá-los [11], [12].

Para recomendar um determinado recurso a um usuário, os sistemas de recomendação podem utilizar-se de duas técnicas para tal:

- **Filtragem colaborativa:** a recomendação é baseada nas preferências do usuário. Para isso, utiliza informações obtidas de outros usuários. Em outras palavras, se os interesses do usuário u_i são semelhantes aos do usuário u_j , os recursos que constituem as preferências de u_j podem ser recomendados a u_i . A filtragem colaborativa tenta avaliar as semelhanças entre os usuários através de seus históricos (associando recursos) ou perfis (associando características).
- **Filtragem baseada em conteúdo:** a recomendação é baseada no perfil do usuário. Para isso, classifica os seus recursos associados e recomenda recursos semelhantes aos que o usuário já avaliou ou possui.

As duas técnicas possuem algumas deficiências. Na filtragem colaborativa, não é possível tratar eficientemente novos usuários ou novos recursos, além da qualidade depender de um grande conjunto de dados históricos. Na filtragem baseada em conteúdo, não é possível explorar novas categorias de recursos além das que o usuário avaliou no passado, e também é necessário um especialista de domínio para avaliar a aquisição de conhecimento. Para minimizar os problemas encontrados, foram criados os *sistemas híbridos*, que combinam a filtragem colaborativa e a filtragem baseada em conteúdo, utilizando os pontos fortes de cada técnica e permitindo uma recomendação mais inerente às necessidades do usuário.

Segundo Reategui e Cazella [13], “os websites de comércio eletrônico são atualmente o maior foco de utilização dos sistemas de recomendação, empregando diferentes técnicas para encontrar os produtos mais adequados para seus clientes e aumentar deste modo sua lucratividade.”

IV. RECOMENDAÇÃO UTILIZANDO AUTÔMATO ADAPTATIVO

Esta seção apresenta o sistema de recomendação proposto utilizando um autômato adaptativo para verificar se uma determinada recomendação é válida.

A. Autômato Adaptativo

O *Autômato Adaptativo*, proposto por Neto [14], é uma extensão do formalismo do Autômato de Pilha Estruturado que permite o reconhecimento de linguagens do tipo 0, segundo a Hierarquia de Chomsky. O termo *adaptativo*, neste contexto, pode ser definido como a capacidade de um dispositivo em alterar seu comportamento de forma espontânea. Logo, um autômato adaptativo tem como característica a possibilidade de provocar alterações em sua própria topologia durante o processo de reconhecimento de uma dada cadeia [15].

Um *autômato adaptativo* M é definido por $M = (Q, S, \Sigma, \Gamma, P, q_0, Z_0, F)$, tal que:

- Q é o conjunto finito de estados, $Q \subseteq Q^A$, Q^A é o conjunto de todos os estados possíveis, Q^A é enumerável,
- S é o conjunto finito de sub-máquinas,
- Σ é o alfabeto de entrada, $\Sigma \subseteq \Sigma^A$, Σ^A é o conjunto enumerável de todos os símbolos possíveis,
- Γ é o alfabeto da pilha, $\Gamma \subseteq \Gamma^A$, $\Gamma = Q \cup \{Z_0\}$, Γ^A é o conjunto enumerável de todos os símbolos possíveis da pilha, $\Gamma^A = Q^A \cup \{Z_0\}$,
- P é um mapeamento $P: Q^A \times \Sigma^A \times \Gamma^A \rightarrow Q^A \times (\Sigma^A \cup \{\epsilon\}) \times (\Gamma^A \cup \{\epsilon\}) \times H^0 \times H^0$, H^0 definido a seguir,
- $q_0 \in Q$ é o estado inicial,
- $Z_0 \in \Gamma$ é o símbolo inicial da pilha, e
- $F \subseteq Q$ é o conjunto de estados finais.

Os conjuntos “ A ” (“All” – para todos) são convenientes porque as *funções adaptativas* podem a) inserir novos estados $q, q \in Q$ mas $q \in Q^A$, e b) usar novos símbolos de pilha $\gamma \in \Gamma$ mas $\gamma \in \Gamma^A$. Em resumo, as *funções adaptativas* podem modificar o autômato, mas os novos símbolos que elas introduzem estão todos nos conjuntos “ A ” [16].

H^0 é o conjunto de todas as funções adaptativas no autômato adaptativo M . Define-se $H^0 = \{f \mid f: E \times G_1 \times G_2 \times \dots \times G_k \rightarrow E\}$, onde f é uma função, $k \in \mathbb{N}$ é o número de argumentos em f , e $G_i = Q^A \cup \Sigma^A \cup \Gamma^A$ [16].

E é o conjunto de todos os autômatos adaptativos que têm o estado inicial q_0 , o símbolo inicial de pilha Z_0 e o conjunto de estados finais F iguais aos do autômato adaptativo M . Define-se $E = \{N \mid N \text{ é um autômato adaptativo } N = (Q', \Sigma', \Gamma', P', q_0, Z_0, F)\}$, onde $Q' \subseteq Q^A$, $\Sigma' \subseteq \Sigma^A$, $\Gamma' \subseteq \Gamma^A$, $P': Q^A \times \Sigma^A \times \Gamma^A \rightarrow Q^A \times (\Sigma^A \cup \{\epsilon\}) \times (\Gamma^A \cup \{\epsilon\}) \times H^0 \times H^0$. Observe que q_0, Z_0 e F são os mesmos em qualquer $N \in E$ [16].

O conjunto de todas as sub-máquinas do autômato adaptativo M é representado por S . Cada *sub-máquina* s_i é definida como $s_i = (Q_i, \Sigma_i, P_i, q_{i0}, F_i)$, onde:

- $Q_i \subseteq Q$ é o conjunto de estados da sub-máquina s_i ,
- $\Sigma_i \subseteq \Sigma$ é o alfabeto de entrada da sub-máquina s_i ,
- $P_i \subseteq P$ é o mapeamento da sub-máquina s_i ,
- $q_{i0} \in Q_i$ é o estado inicial da sub-máquina s_i , e
- $F_i \subseteq Q_i$ é o conjunto de estados finais da sub-máquina s_i .

A linguagem aceita por um autômato adaptativo M é dada por $L(M) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q_f, \epsilon, Z_0)\}$, onde $q_f \in F$. A capacidade de auto-modificação confere ao autômato

adaptativo o poder computacional de uma Máquina de Turing [17].

B. Sistema de Recomendação utilizando Autômato Adaptativo

O autômato adaptativo foi utilizado para modelar um sistema de recomendação simples, chamado *RecomAA*. Dados dois recursos, o sistema verifica se existe alguma relação entre eles. Se a relação existe, o segundo recurso pode ser recomendado em função do primeiro e vice-versa. Por exemplo, dados os recursos *agenda* e *livro*, deseja-se saber se são relacionados; em outras palavras, o sistema responde *sim* ou *não* à seguinte pergunta: *dado o recurso agenda, é possível recomendar o recurso livro?*, ou *dado o recurso livro, é possível recomendar o recurso agenda?*.

Além de verificar se os recursos são relacionados, o sistema *RecomAA* permite que dois recursos sejam avaliados para uma possível criação de relação entre eles. Por exemplo, suponha que os recursos *agenda* e *livro* não são relacionados; entretanto, existem vários clientes que, ao comprar uma agenda, também incluem um livro em sua compra. Os recursos *agenda* e *livro* não são relacionados, mas a *distância* entre eles diminui; isso significa que, em um determinado momento, a distância entre eles indicará a criação de uma relação entre *agenda* e *livro*.

Inicialmente, o autômato adaptativo *M* do *RecomAA* é definido com uma configuração contendo todos os recursos existentes em um determinado cenário, por exemplo, todos os produtos de uma loja de comércio eletrônico. Durante seu tempo de vida, *M* sofrerá alterações em sua topologia para reduzir distâncias entre recursos até o estabelecimento de relações. Isto é feito através da função adaptativa *f* de *M*, que pode alterar o autômato.

Suponha que o sistema *RecomAA* possua um conjunto $\Pi = \{ p_1, p_2, \dots, p_n \}$ de recursos. Assim, o autômato adaptativo *M* possuirá um alfabeto $\Sigma = \Pi \cup \{ link \}$, onde *link* é um símbolo reservado utilizado para reduzir as distâncias entre recursos.

As cadeias a serem submetidas ao autômato adaptativo *M* podem ser nos seguintes formatos:

- $p_i p_k$, representado uma consulta sobre a existência de uma relação entre dois recursos p_i e p_k , e
- $\langle link \rangle p_i p_k$, representando uma avaliação sobre a relação entre os dois recursos p_i e p_k ,

com $p_i, p_k \in \Pi$. Assim, a linguagem reconhecida pelo autômato adaptativo *M* denota a existência de uma relação válida entre dois recursos ou uma avaliação para uma possível criação de uma relação entre tais recursos.

A Figura 1 apresenta um exemplo de configuração inicial do autômato adaptativo *M* utilizado no sistema de recomendação *RecomAA*. O cenário representado possui apenas dois recursos, p_1 e p_2 não relacionados entre si.

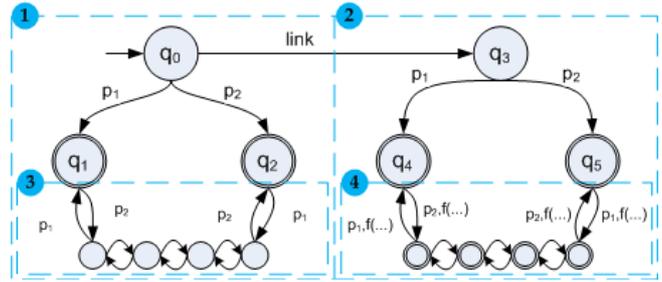


Figura 8. Exemplo do Autômato Adaptativo *M* utilizado no sistema *RecomAA*.

Para ilustrar a execução, o autômato adaptativo *M* (Figura 1) foi dividido em 4 regiões, apresentadas a seguir.

- **Região 1:** constitui a verificação das relações entre os recursos. No exemplo, os recursos p_1 e p_2 estão definidos nessa região.
- **Região 2:** essa região trata da avaliação dos recursos. A partir de cada estado alcançável por uma transição consumindo um símbolo representando um recurso, existe um caminho definido na região 3 que leva até os outros estados. Um *caminho* é constituído por um conjunto de estados intermediários (no exemplo, é representado por estados de tamanho menor) que possuem transições entre si e que conduzem a um determinado estado de destino. O objetivo de um caminho é representar a distância existente para a recomendação de um recurso, dado um outro recurso. No exemplo, como existem apenas dois recursos, existirá apenas um caminho entre q_4 e q_5 , com transições consumindo p_2 de q_4 até q_5 e transições consumindo p_1 de q_5 até q_4 .
- **Região 3:** a região 3 contém todos caminhos de cada estado até os demais. Quando o caminho for representado por apenas duas transições ligando dois estados que representam os recursos, sem estados intermediários compondo esse caminho, existirá relação entre esses dois recursos. Por exemplo, se q_1 possuir uma transição consumindo p_2 até q_2 e se q_2 possuir uma transição consumindo p_1 até q_1 , os dois recursos têm relação entre si e podem ser recomendados.

Seja $\Pi = \{ p_1, p_2, \dots, p_n \}$ o conjunto de recursos. O total de caminhos existentes no autômato adaptativo *M* será igual à $C(n, 2)$ para a região 3 e $C(n, 2)$ para a região 4. $C_{n,i}$ é uma combinação simples de n elementos, i a i , representada pela seguinte fórmula:

$$C_{n,i} = \frac{n!}{i!(n-i)!}$$

(1)

$C_{n,i}$ (Fórmula 1) denota o número total de combinações de n elementos tomados i a i , onde i é o número de elementos presentes em cada combinação.

- **Região 4:** essa região possui os caminhos definidos para cada estado que representa um recurso. Como o caminho representa a distância existente para a recomendação de um recurso, para cada transição existente no caminho, existe uma função adaptativa *f* associada. O objetivo da função *f* é remover um estado intermediário desse

caminho na região 4 e também na região 3, reduzindo assim a distância do caminho existente para a recomendação de um recurso, até que existam apenas duas transições entre os estados de origem e destino desse caminho.

O tamanho de um caminho, isto é, o número de estados intermediários componentes, deve refletir um número razoável de ocorrências de um determinado recurso, dado outro recurso, para que a recomendação seja válida. Valores menores podem criar recomendações não desejáveis, ao passo que valores maiores podem retardar a criação de recomendações mais propensas. No exemplo apresentado na Figura 1, o tamanho do caminho é igual a 4.

É interessante contar com um *especialista de domínio* para analisar o domínio desejado e traçar tamanhos de caminhos mais inerentes à aplicação. Além disso, é possível realizar, de tempos em tempos, análises e eventuais alterações na configuração do autômato de modo a representar com maior consistência as recomendações do sistema.

V. EXPERIMENTO E ANÁLISE

Para verificar as recomendações do sistema *RecomAA*, foi realizado um experimento com 15 alunos de graduação e pós-graduação do Departamento de Computação da Universidade Federal de São Carlos. A primeira parte do experimento consistiu no preenchimento de uma ficha contendo uma relação com 10 livros, apresentados na Tabela I, e 5 campos para resposta. Cada aluno simulou 5 compras e anotou os produtos comprados na ficha. A amostra utilizada no experimento é descrita na Tabela II.

Tabela I
LIVROS UTILIZADOS NO EXPERIMENTO.

Índice	Nome do Livro
p_1	O caçador de pipas
p_2	Quando Nietzsche chorou
p_3	Ensaio sobre a cegueira
p_4	O vendedor de sonhos
p_5	Crime e Castigo
p_6	Crepúsculo
p_7	A cidade e sol
p_8	A menina que roubava livros
p_9	O guardião de memórias
p_{10}	O vencedor está só

Tabela II
AMOSTRA UTILIZADA NO EXPERIMENTO.

Aluno	Compra 1	Compra 2	Compra 3	Compra 4	Compra 5
1	p_2, p_3, p_4, p_7, p_9	$p_1, p_4, p_5, p_6, p_{10}$	p_1, p_2, p_4, p_7, p_8	p_1, p_4, p_5, p_{10}	p_6, p_7, p_8, p_9
2	p_1, p_3, p_4, p_9	p_4, p_5, p_7, p_{10}	p_1, p_2, p_8, p_9	p_6, p_7	p_1, p_2, p_8
3	p_1, p_4, p_8, p_{10}	p_2, p_3	p_1, p_2	p_4, p_7, p_9	p_1, p_{10}
4	p_3, p_7, p_{10}	p_1, p_2	p_4, p_8, p_9	p_4, p_7, p_8	p_3, p_4, p_5
5	p_2, p_5	p_1, p_7, p_8, p_9	p_1, p_2, p_7, p_8	p_7, p_9	p_2, p_5

6	p_3, p_4, p_8	p_2, p_4	p_1, p_9	p_7, p_8	p_3, p_8
7	p_1, p_2, p_7	p_2, p_3, p_5	p_4, p_9	p_{10}	p_1, p_4
8	p_{10}	p_7, p_8	p_5, p_9	p_3, p_9	p_1
9	p_2, p_4, p_6	p_3, p_7, p_9	p_{10}	p_7, p_8	p_4
10	p_4, p_8	p_1, p_2	p_3, p_4	p_9, p_{10}	p_3, p_4, p_7, p_8
11	p_1, p_2, p_3	p_1, p_4, p_7	p_6, p_9, p_{10}	p_2, p_8, p_{10}	p_1, p_8
12	p_5	p_3, p_4	p_{10}	p_7, p_9	p_1, p_8
13	p_3	p_1	p_2	p_8	p_{10}
14	p_3, p_8	p_3, p_9, p_{10}	p_1, p_2	$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$	p_6, p_8, p_9
15	p_6	p_1, p_{10}	p_1, p_4, p_7	p_6, p_8, p_9	p_1, p_4, p_7

A frequência dos produtos utilizados no experimento (Tabela I) é ilustrada na Figura 2.

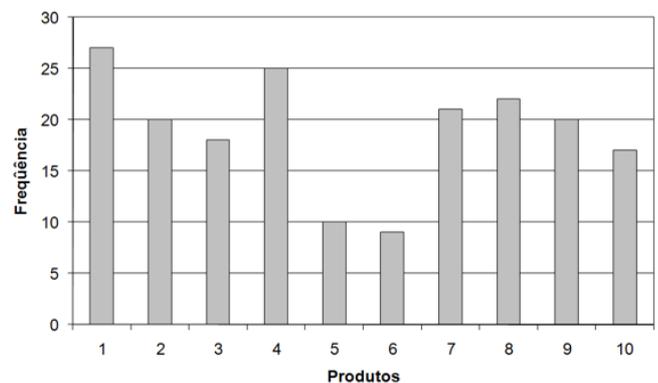


Figura 9. Frequência dos produtos utilizados no experimento.

As fichas preenchidas pelos alunos foram coletadas e cada produto foi relacionado com todos os outros produtos da mesma compra. O número de ocorrências de cada relação de todas as compras de todas as fichas foi registrado na Tabela III. É importante observar que o valor contido na célula (p_i, p_j) corresponde ao mesmo valor da célula (p_j, p_i), pois representam a mesma relação (comprar um *livro* e uma *agenda* também corresponde a comprar uma *agenda* e um *livro*). Compras que possuem apenas um produto não constam na tabela.

Tabela III
PRODUTOS RELACIONADOS DE ACORDO COM O EXPERIMENTO REALIZADO.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	-	10	3	7	3	1	7	8	5	5
p_2	10	-	5	6	4	2	5	5	3	2
p_3	3	5	-	7	3	1	6	4	7	3
p_4	7	6	6	-	5	3	8	7	5	5
p_5	3	4	4	5	-	1	1	0	1	3
p_6	1	2	2	3	1	-	3	2	3	3
p_7	7	5	5	8	1	3	-	9	8	3
p_8	8	5	5	7	0	2	9	-	5	3
p_9	5	3	3	5	1	3	8	5	-	4
p_{10}	5	2	2	5	3	3	3	3	4	-

O autômato adaptativo M do experimento foi modelado contemplando os 10 produtos apresentados anteriormente, $\Pi = \{ p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10} \}$ e alfabeto $\Sigma = \Pi \cup \{ link \}$. Foi assumido o valor 8 como tamanho dos caminhos para M .

A segunda parte do experimento consistiu na submissão de cadeias ao autômato adaptativo M do sistema *RecomAA*, de acordo com os dados obtidos na Tabela III. Cada cadeia possuía o formato $w = \langle link \rangle p_i p_j$ e submetida k vezes ao autômato, onde k é o valor da célula (p_i, p_j) . Por exemplo, a cadeia $\langle link \rangle p_3 p_4$ foi submetida 7 vezes ao autômato adaptativo M . As cadeias no formato $\langle link \rangle p_j p_i$ não foram consideradas por representarem a mesma relação que $\langle link \rangle p_i p_j$.

Após a submissão de todas as cadeias no formato $w = \langle link \rangle p_i p_j$ ao autômato adaptativo M , foram realizadas consultas sobre a recomendação de um produto em função de outro. Para tal, 45 cadeias no formato $u = p_i p_j$ foram submetidas ao autômato, verificando sua pertinência para $L(M)$. As cadeias no formato $p_j p_i$ não foram consideradas por representarem a mesma relação que $p_i p_j$.

A Tabela IV apresenta o resultado das submissões das cadeias no formato $u = p_i p_j$ ao autômato adaptativo M , onde:

- ✓ se $u \in L(M)$, ou
- ✗ se $u \notin L(M)$.

Tabela IV

RESULTADO DAS SUBMISSÕES DAS CADEIAS NO FORMATO $u = p_i p_j$ ao AUTÔMATO ADAPTATIVO M .

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	-	✓	✗	✗	✗	✗	✗	✓	✗	✗
p_2	✓	-	✗	✗	✗	✗	✗	✗	✗	✗
p_3	✗	✗	-	✗	✗	✗	✗	✗	✗	✗
p_4	✗	✗	✗	-	✗	✗	✓	✗	✗	✗
p_5	✗	✗	✗	✗	-	✗	✗	✗	✗	✗
p_6	✗	✗	✗	✗	✗	-	✗	✗	✗	✗
p_7	✗	✗	✗	✓	✗	✗	-	✓	✓	✗
p_8	✓	✗	✗	✗	✗	✗	✓	-	✗	✗
p_9	✗	✗	✗	✗	✗	✗	✓	✗	-	✗
p_{10}	✗	✗	✗	✗	✗	✗	✗	✗	✗	-

A Tabela IV representa as relações existentes entre os produtos. A existência de uma relação entre p_i e p_j indica que p_j é recomendável dado p_i e vice-versa. Por exemplo, de acordo com os perfis de compra dos usuários participantes do experimento, o produto p_8 pode ser recomendado ao usuário, caso este já tenha comprado ou esteja comprando o produto p_1 . Por outro lado, o produto p_8 não é recomendado, caso o usuário esteja comprando o produto p_2 .

VI. IMPLEMENTAÇÃO DO SISTEMA DE RECOMENDAÇÃO UTILIZANDO AUTÔMATO ADAPTATIVO

O Sistema de Recomendação utilizando Autômato Adaptativo (*RecomAA*) foi implementado em Python. A linguagem possui algumas características que contribuíram para sua escolha, tais como velocidade de execução, flexibilidade, sintaxe simples, grande número de módulos de extensão disponíveis e integração com servidores Web.

O conjunto de recursos $\Pi = \{ p_1, \dots, p_n \}$ é representado no *RecomAA* utilizando a linguagem de marcação XML. Cada *tag*

do arquivo XML representa um recurso $p_i \in \Pi$. Um exemplo de arquivo XML pode ser visto a seguir.

```
<?xml version="1.0"?>
<recursos>
  <recurso>recurso 1</recurso>
  <recurso>recurso 2</recurso>
  ...
  <recurso>recurso n</recurso>
</recursos>
```

A partir do arquivo XML contendo o conjunto de recursos, o sistema *RecomAA* constrói o autômato adaptativo.

A configuração e execução do *RecomAA* no *shell* do Python é apresentada a seguir. O arquivo XML contendo o conjunto dos recursos foi criado de acordo com o experimento apresentado na seção anterior.

```
Python 2.5.2 (r252:60911, Feb 21 2008,
13:11:45) [MSC v.1310 32 bit (Intel)]
on win32
...
>>> from RecomAA import *
>>> sistema = RecomAAConf()
>>> sistema.load('recursos.xml')
True
```

Foi criado o objeto *sistema* da classe de configuração do *RecomAA* (*RecomAAConf*). O arquivo *recursos.xml* foi carregado no sistema através do método *load()* e, a partir dos dados contidos nesse arquivo, o autômato adaptativo foi criado.

```
>>> automato = sistema.getAutomaton()
True
>>> automato.getInfo()
RecomAA v1.0.10 (codename wood)
:: Automato Adaptativo criado em
:: 12/10/2008 10:14 (GMT -003)
:: Recursos disponíveis: 10
:: Tamanho do caminho: 8 estados
```

O objeto *automato* é uma instância da classe que implementa o autômato adaptativo do *RecomAA* (*AAutomaton*). O método *getInfo()* retorna as informações sobre o objeto, como data de criação, recursos contemplados pelo autômato e o tamanho do caminho.

A submissão de cadeias ao autômato adaptativo *automato* pode ser realizada através do método *query(...)*. O método retorna *true* se a cadeia pertence ao autômato, ou *false* caso contrário. A submissão de uma cadeia que denota uma consulta sobre a relação entre os produtos p_1 e p_8 é ilustrada a seguir.

```
>>> automato.query('p1', 'p8')
False
```

O método *query('p1', 'p8')* retornou *False*, o que denota que os produtos p_1 e p_8 não possuem relação entre si.

A próxima etapa da execução consistiu na submissão de cadeias no formato $w = \langle link \rangle p_i p_j$ k vezes ao autômato, onde

k é o valor da célula (p_i, p_j) da Tabela III. Essa etapa é igual à segunda parte do experimento apresentado na seção anterior.

```
>>> automato.query('link', 'p1', 'p2')
True
>>> automato.query('link', 'p1', 'p2')
True
...
>>> automato.query('link', 'p9', 'p10')
True
```

Após as submissões das cadeias no formato $w = \langle link \rangle p_i p_j$, a submissão de uma cadeia no formato $u = p_i p_j$ deverá coincidir com os resultados obtidos na Tabela IV. Por exemplo, nesse momento a submissão da cadeia $p_1 p_8$ retornará True, de acordo com a Tabela IV.

```
>>> automato.query('p1', 'p8')
True
```

A implementação do sistema *RecomAA* também permite sua utilização em páginas Web, disponível através do módulo *RecomAAWebInterface*, também escrito em Python. A Figura 3 apresenta a interface Web do *RecomAA*.



Figura 10. Interface Web do *RecomAA*.

A interface Web disponibiliza um campo para consulta que possibilita verificar a existência de alguma relação entre dois recursos. A Figura 4 ilustra a resposta de uma consulta.



Figura 11. Resposta de uma consulta através da Interface Web do *RecomAA*.

O módulo *RecomAAWebInterface* é um script Python CGI que pode ser executado em um servidor Web, através do endereço:

```
/cgi-bin/recomaawebinterface.py
```

A implementação do *RecomAA* tem como objetivo fornecer um conjunto de classes que realizam uma recomendação simples de recursos, utilizando um autômato adaptativo, para os sistemas existentes.

VII. CONCLUSÕES

Este artigo apresentou um sistema de recomendação utilizando um autômato adaptativo para verificar as relações entre recursos. O autômato adaptativo, devido à característica de auto-modificação, é uma escolha interessante para auxiliar na avaliação de quais recursos são mais importantes para um determinado usuário.

A recomendação de recursos é viável do ponto de vista comercial, pois permite que um ambiente torne-se “mais próximo” do usuário, conhecendo seus interesses e hábitos de navegação, e sugerindo recursos que o agradem. A utilização da tecnologia adaptativa na área de personalização de serviços e recursos proporciona soluções consistentes que atendem às necessidades inerentes a esse domínio de aplicação.

REFERÊNCIAS

- [1] R. E. Grande, “Sistema de integração de técnicas de proteção de privacidade que permitem personalização”, Dissertação de Mestrado, Departamento de Computação, Universidade Federal de São Carlos, 2006.
- [2] M. Koch and K. Moeslein, “User representation in ecommerce and collaboration applications”. In *BLED 2003 Proceedings*, 2003.
- [3] R. Kimball and R. Merz, *Data Webhouse: construindo o data warehouse para a Web*. Rio de Janeiro: Campus, 2000.
- [4] S. D. Zorzo, R. A. Gotardo, P. R. M. Cereda, B. Y. L. Kimura, R. A. Rios and R. E. Grande, “Web privacy controlled by user: an approach to treat the user’s preferences about personal data”. In *European Computing Conference 2007*, 2007.
- [5] D. Kristol and L. Montulli, “HTTP state management mechanism”, Bell Laboratories, Lucent Technologies, RFC 2965, 2000.
- [6] A. L. Montgomery, “Using clickstream data to predict www usage”, University of Maryland, Tech. Report, 2003.
- [7] A. L. B. Nogueira and L. R. Oliveira Jr., *Uma análise da aplicabilidade de Data Warehouse em ambientes empresariais*, Faculdade Ruy Barbosa, Salvador, 2004.
- [8] S. Azambuja, “Estudo e implementação da análise de agrupamento em ambientes virtuais de aprendizagem”, Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2005.
- [9] W. Aiello and P. McDaniel, *Lecture 1, Intro: Privacy*, Stern School of Business, NYU, 2004.
- [10] M. P. Silva, C. Boscaroli and S. M. Peres, “Análise de logs da web por meio de técnicas de data mining”. In *I Congresso de Tecnologias para Gestão de Dados e Metadados do Cone Sul*, Universidade Estadual de Ponta Grossa, 2003.
- [11] R. A. Gotardo, C. A. C. Teixeira and S. D. Zorzo, “Predicting user’s interests in web-based educational systems using a collaborative filtering weighted method”. In *11th IEEE International Conference on Computational Science and Engineering – Workshop on Computational Science and Engineering – CSE 2008*, 2008.
- [12] S. Sae-Tang and V. Esichaikul, “Web personalization techniques for e-commerce”. In *Active Media Technology: 6th International Computer Science Conference*, 2005.
- [13] E. B. Reategui and S. C. Cazella, “Sistemas de recomendação”. In *XXV Congresso da Sociedade Brasileira de Computação – A universalidade da computação: um agente de inovação e conhecimento*, 2005.

- [14]J. J. Neto, “Contribuições à metodologia de construção de compiladores”, Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, 1993.
- [15]_____, “Adaptive automata for context-dependent languages”, *SIGPLAN Notices*, vol. 29, no. 9, PP. 115-124, 1994.
- [16]P. R. M. Cereda, “Modelo de controle de acesso adaptativo”, Dissertação de Mestrado, Departamento de Computação, Universidade Federal de São Carlos, 2008.
- [17]R. L. A. Rocha and J. J. Neto, “Autômato adaptativo, limites e complexidade em comparação com máquina de Turing”. In *Proceedings of the Second Congress of Logic Applied to Technology – LAPTEC*. São Paulo, Brasil: Faculdade SENAC de Ciências Exatas e Tecnologia, 2001.



Paulo Roberto Massa Cereda obteve o mestrado em Ciência da Computação pela Universidade Federal de São Carlos em 2008. Atualmente é aluno pesquisador do Grupo de Sistemas Distribuídos e Redes (GSDR) do Departamento de Computação da Universidade Federal de São Carlos. Tem experiência na área de Ciência da Computação, atuando nas áreas de Privacidade e Personalização de Serviços, Teoria da Computação e Tecnologias Adaptativas.



Reginaldo Aparecido Gotardo obteve o mestrado em Ciência da Computação pela Universidade Federal de São Carlos em 2008. Atualmente é professor das Faculdades COC. Tem experiência na área de Ciência da Computação, com ênfase em Linguagens de Programação, atuando principalmente nos seguintes temas: Computação em Grade, GT4, Globus, Web Services e Privacidade e Personalização de Serviços.



Sérgio Donizetti Zorzo obteve o doutorado em Engenharia Elétrica pela Universidade de São Paulo em 1996. Atualmente é professor associado do Departamento de Computação da Universidade Federal de São Carlos. Tem experiência na área de Ciência da Computação, com ênfase em Teleinformática, atuando principalmente em Qualidade, Privacidade e Personalização de Serviços, Multimídia, Computação em Ambientes sem Fio e Tecnologias Adaptativas.