

Sistema de Recomendação Web Usando Agentes

A. Jorge Morais^{2,3,6}, Joaquim Neto^{5,6}, Eugénio Oliveira^{2,4}, Alípio Mário Jorge^{1,3}

¹Faculdade de Ciências da Universidade do Porto ²Faculdade de Engenharia da Universidade do Porto

³Laboratório de Inteligência Artificial e Apoio à Decisão (LIAAD – INESC TEC, L.A.)

⁴Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC – UP)

⁵Laboratório Nacional de Engenharia Civil ⁶Universidade Aberta

Jorge.Morais@uab.pt;jaqf.neto@gmail.com;eco@fe.up.pt;amjorge@fc.up.pt

Resumo

O crescimento da Web trouxe vários problemas aos utilizadores. A grande quantidade de informação existente hoje em dia em alguns sítios Web torna a procura de informação útil muito difícil. Os objetivos dos proprietários dos sítios Web e dos utilizadores nem sempre coincidem. O conhecimento dos padrões de visitas dos utilizadores é crucial para que os proprietários possam transformar e adaptar o sítio Web. Este é o princípio do sítio Web adaptativo: o sítio Web adapta-se de forma a melhorar a experiência do utilizador. Alguns algoritmos foram propostos para adaptar um sítio da Web. Neste artigo, descrevemos uma proposta de um sistema de recomendação Web baseado em agentes que combina dois algoritmos: regras de associação e filtragem colaborativa. Ambos os algoritmos são incrementais e funcionam com dados binários. Os resultados mostram que, em algumas situações, a abordagem multiagente melhora a capacidade preditiva quando comparada com os agentes individuais.

palavras-chave: Web Mining, Sistemas de Recomendação, Sistemas Multiagente

Abstract

The growth of the Web has brought several problems for users. Today the vast amount of information on some web sites makes useful information finding very difficult. The objectives of the owners of the web sites and users do not always coincide. The knowledge of patterns of user visits is crucial to the owners to transform and adapt their web site. This is the adaptive website principle: the website adapts to improve the user experience. Some algorithms have been proposed to tailor a website. In this paper, we describe a proposal for a web recommendation system based on agents that combines two algorithms: association rules and collaborative filtering. Both algorithms are incremental and work with binary data. The results show that, in some situations, the multi-agent approach overcomes the predictive capacity of individual agents.

keywords: web mining, recommendation systems, multi-agent systems

1. Introdução

Hoje em dia, a maioria das organizações tem um sítio Web para divulgar informação para o público em geral. Quando o tamanho deste cresce, a dificuldade para os utilizadores encontrarem informação útil também cresce. Isso levou as organizações a lidar com o problema de organizar toda a informação de forma eficiente, de modo a tornar fácil encontrar todos os produtos ou informação que um utilizador procura.

Vamos considerar, por exemplo, que um cliente está à procura de um determinado produto com características particulares. A estrutura do sítio Web pode ser organizada por diferentes marcas, o que significa que os clientes terão de procurar dentro de cada marca para comparar os modelos. Suponhamos que, a partir de visitas anteriores de outros utilizadores, é descoberto um padrão como "utilizadores que pesquisam este produto também pesquisam aqueles". Descobrir tal padrão pode levar a considerar uma solução, por exemplo, sugerindo produtos similares de outras marcas na mesma página. Um algoritmo que é capaz de fazer este tipo de sugestões é chamado um sistema de recomendação [Ricci et al. 2011].

Uma das abordagens que tem sido propostas para este problema é utilizar agentes autónomos. Sistemas Multiagente [Wooldridge 2009] é uma área de pesquisa que tem estado em grande desenvolvimento ao longo da última década e tem algumas características particulares que se encaixam neste problema. Na verdade, as abordagens multiagente já têm sido aplicadas na automatização da Web [Ardissono et al. 2005], devido à sua flexibilidade e características. Sistemas Multiagente também são usados para recuperação e atualização automática de informação em sítios Web [Albayrak et al. 2005]. Uma implementação de um sistema de recomendação usando esta abordagem já tinha sido proposto por Wei em 2005 [Wei 2005].

Neste trabalho propomos um sistema de recomendação multiagente que combina dois algoritmos diferentes (regras associativas e filtragem colaborativa) e explora um paradigma baseado em mercado, incluindo estratégias de licitação. Ambos os algoritmos de recomendação são incrementais e trabalham com dados binários.

Este artigo começa por apresentar o trabalho anterior na área de sistemas de recomendação e sistemas multiagente, seguido pela descrição da nossa abordagem. Os resultados dos testes com seis conjuntos de dados, as conclusões e trabalho em curso e futuro completam o artigo.

2. Conceitos e trabalho anterior

Um sistema de recomendação típico [Resnick, Varian 1997] usa, geralmente, um modelo construído a partir de dados sobre as visitas ou avaliações de utilizadores anteriores. Dado um utilizador ativo, um sistema de recomendação apresenta as n recomendações principais que correspondem às preferências mais prováveis do utilizador. O modelo pode ser construído usando informação explícita fornecida pelos utilizadores (gosto/não gosto, 1 a 5 estrelas, etc...), ou informação implícita com base no comportamento do utilizador (páginas Web que o utilizador visitou, o tempo gasto em uma determinada página Web, etc...). Embora a informação explícita seja, aparentemente, mais confiável, é mais exigente para o utilizador. A informação implícita só nos dá informação sobre o comportamento do utilizador, e não sobre o que

realmente este pensava sobre um determinado item. No entanto, esta informação é relevante e pode ser utilizada para construir o modelo do sistema de recomendação. Em [Schafer 2006] é afirmado que, em alguns domínios, a informação implícita resulta em modelos mais precisos do que classificações explícitas.

Os sistemas de recomendação são, geralmente, classificados nas seguintes categorias [Balabanovic, Shoham 1997] [Adomavicius, Tuzhilin 2005]: baseados em conteúdo, onde são recomendados itens com conteúdo semelhante aos preferidos anteriormente; ou colaborativos, onde são recomendados itens de que pessoas com gostos e preferências semelhantes gostaram no passado.

Os sistemas de recomendação colaborativos, em vez de se concentrarem num mesmo utilizador, tentam encontrar itens úteis com base nas preferências de outros. De acordo com [Breese, Heckerman, Kadie 1998], os algoritmos para essa abordagem podem ser agrupados em duas classes gerais: baseados em modelo, que usam o conjunto de classificações para gerar um modelo que é usado para fazer previsões; baseados em memória, que fazem previsões com base na combinação de itens armazenados, anteriormente classificados pelos utilizadores.

Os algoritmos baseados em memória podem ser baseados no utilizador ou no item, de acordo com a forma como as semelhanças são calculadas. Em termos práticos, quando seguimos uma abordagem baseada no item, podemos confiar apenas nas semelhanças entre os itens, desconsiderando a informação da relação utilizador-item. A abordagem baseada no item é capaz de fornecer resultados com a mesma qualidade e menos tempo de computação [Sarwar et al. 2001] [Linden, Smith, York 2003].

Em [Adomavicius, Tuzhilin 2005], afirma-se que, uma vez que os sistemas colaborativos usam as classificações de outros utilizadores, eles não têm alguns dos problemas que os sistemas baseados em conteúdo têm. No entanto, também têm as suas próprias limitações, tais como o problema do novo utilizador, o problema novo item e a dispersão. Para os dois primeiros problemas, são sugeridas abordagens híbridas, enquanto que para o último os autores sugerem o uso de filtragem demográfica [Pazzani 1999] e decomposição em valores singulares [Sarwar et al. 2000] [Goldberg et al. 2001] [Canny 2002]. Em [Jannach et al. 2011], foi também sugerida a utilização de um método baseado em grafos, para explorar a eventual transitividade das preferências dos utilizadores [Huang, Chen, Zeng 2004].

Uma visão global sobre sítios Web adaptativos com base na análise de interação do utilizador é dada em [Perkowitz, Etzioni 2000]. Na verdade, foram propostas abordagens menos ambiciosas, como a reorganização do sítio Web [Ishikawa et al. 2002], o uso de recomendações nas páginas [El-Ramly, Stroulia 2004], a categorização automática de ações do utilizador [Berendt 2002], ou procurar caminhos de sequências de páginas Web relevantes usando modelos de Markov [Borges 2000].

Os sistemas de recomendação existentes incluem a combinação de agrupamentos com o algoritmo dos vizinhos mais próximos [Mobasher et al. 2002], cadeias de Markov e agrupamentos [Cadez et al. 2003], regras de associação [Jorge et al. 2003], e filtragem colaborativa e baseada em conteúdo [Basilico, Hofmann 2004].

Algumas plataformas, como o WebWatcher, usam o conhecimento dos utilizadores anteriores para recomendar ligações [Armstrong et al.1995]. O AVANTI implementa uma apresentação adaptativa baseada num modelo construído a partir de ações do utilizador [Fink, Kobsa, Nill 1996]. O sistema WUM infere uma estrutura de árvore de registos de acesso que permite aos peritos encontrar padrões com características pré-definidas [Spiliopoulou, Faulstich 1998]. Em [Masseglia, Teisseire, Poncelet 2001] foi proposta uma ferramenta integrada (HDM) para descobrir padrões de acesso e regras de associação a partir de registos de acesso, a fim de modificar automaticamente a organização do hipertexto.

Em [Ardissono et al. 2005], foi proposto o uso de uma plataforma multiagente para personalização de sistemas baseados na Web. As abordagens multiagente para o desenvolvimento de sistemas complexos, como a adaptação da Web, foram defendidas em [Jennings 2001]. Os agentes inteligentes também pode ser uma contribuição importante para a computação autónoma [Kephart 2005]. Tais sistemas são complexos e têm capacidades de autoadministração, autovalidação, autorregulação e de autocorreção. Os sistemas de adaptação Web também devem ter estas características, porque a dinâmica do sítio Web requer um alto grau de automação do sistema ou alta alocação de recursos humanos. Outro uso importante dos sistemas multiagente é a recolha automática e atualização de informação na Web [Albayrak 2005].

Um trabalho anterior propôs a implementação de um sistema de recomendação usado uma abordagem multiagente [Wei 2005]. O autor implementou um sistema de recomendação baseado no mercado, que é considerado ser eficiente de Pareto, capaz de maximizar o bem-estar social e todos os agentes serem individualmente racionais. O autor faz uma distinção entre Qualidade Interna (INQ) e Perceção do usuário da Qualidade (UPQ). O valor INQ de uma recomendação específica é a soma dos valores da avaliação ponderada de diferentes técnicas em diferentes propriedades de uma recomendação, enquanto UPQ é introduzido pelos utilizadores. Os agentes não têm conhecimento de UPQ, e eles devem inferi-lo do mecanismo de recompensa. Um leilão de primeiro preço em envelope fechado é usado para minimizar o tempo para a execução do leilão e a quantidade de comunicação gerada.

Em [Morais, Jorge 2009] foi apresentada uma plataforma de adaptação Web, que foi a base para este trabalho, com as adaptações posteriores às nossas necessidades particulares [Morais, Oliveira, Jorge 2012]. Dado que a Web é um ambiente dinâmico, os algoritmos que usamos são incrementais. Uma implementação de filtragem colaborativa usando uma abordagem incremental foi apresentada em [Miranda, Jorge 2008].

3. Abordagem multiagente

A abordagem multiagente que seguimos foi implementada em Java, utilizando a plataforma JADE [<http://jade.tilab.com>]. A comunicação com o navegador é implementada usando AJAX [<https://developer.mozilla.org/en/ajax>]. As figuras 1 e 2 mostram como funciona o sistema. A página Web atual é escrita em HTML com AJAX, e um *servlet* HTTP fornece a porta de entrada para o JADE. Um agente de porta de entrada lida com toda a comunicação entre o utilizador e o sistema de recomendação. Dentro do sistema de recomendação há uma base de conhecimento implementada no Apache Derby [Masseglia, Teisseire, Poncelet 2001] com todas as informações úteis,

um conjunto de agentes autónomos e um corretor que lida com as solicitações para os agentes e constrói a recomendação final para o utilizador. Cada agente regista-se no Diretório Facilitador (uma componente do JADE) para que o corretor saiba que agentes estão disponíveis cada vez que uma recomendação é necessária. Isto permite a adição e a remoção de agentes a qualquer momento.

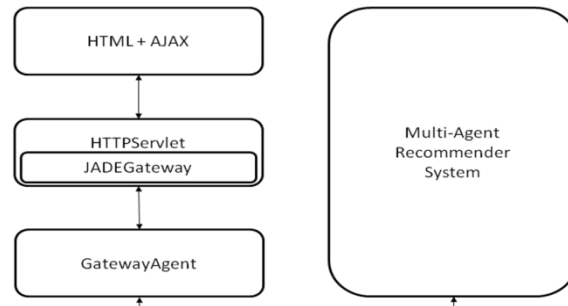


Fig. 1. Abordagem multiagente para adaptação da Web

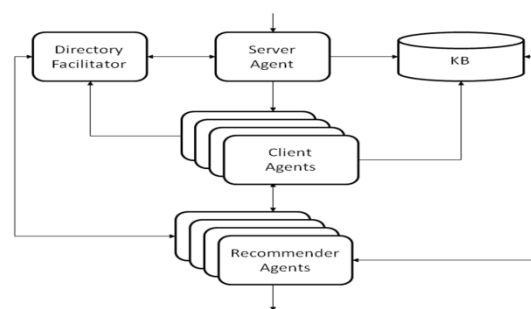


Fig. 2. Sistema de recomendação multiagente

Para os testes utilizamos uma versão *offline* do sistema, para simular acessos ao sítio Web. Os conjuntos de dados de acessos a sítios são lidos por um agente simulador que faz o papel do agente de porta de entrada. Cada registo do conjunto de dados contém o identificador da página Web, o identificador do utilizador e a data e hora do pedido.

Cada pedido é enviado para o agente servidor que o encaminha para o respetivo agente do cliente (se não existir, um novo agente será criado para esse cliente). O agente cliente consulta o Diretório Facilitador e envia um pedido aos agentes disponíveis. Cada agente envia uma oferta e um conjunto de recomendações. O agente cliente regista o vencedor do leilão (se fosse em linha também enviaria as recomendações para o agente de porta de entrada) e aguarda para o próximo pedido. Quando não há pedidos de um utilizador durante 30 minutos, considera-se que a sessão acabou, e a sessão é registada na base de conhecimento, atualizando os modelos de recomendação.

Dois agentes foram criados. O primeiro gera regras de associação com base em critérios de confiança e de suporte, enquanto o segundo utiliza um algoritmo de filtragem colaborativa. Como as recomendações devem ser rápidas, a fim de manter os utilizadores interessados, e tendo em conta que cada nova resposta atualiza os modelos de recomendação, estas abordagens incrementais devem ser capazes de fornecer um conjunto de recomendações numa parcela muito pequena de tempo. Assim, ambos os

algoritmos partilham uma matriz $A_{n \times n}$, onde n é o número de itens (páginas Web) e cada $a_{ij} \in A$ regista a coocorrência dos itens i e j na mesma sessão. A matriz é atualizada cada vez que uma sessão termina.

O agente de regras de associação verifica todas as regras possíveis na forma $i \rightarrow j$, onde i e j são itens, levando em conta dois valores (sendo k o número de sessões):

$$Support_{i \rightarrow j} = \frac{a_{ij}}{k} . \quad (1)$$

$$Confidence_{i \rightarrow j} = \frac{a_{ij}}{a_{ii}} . \quad (2)$$

Assim, se um conjunto de r recomendações é pedido, as r recomendações com maior confiança, que satisfazem valores mínimos de suporte e confiança, são propostas.

O agente da filtragem colaborativa usa a mesma matriz para calcular a similaridade.

$$sim(i, j) = \frac{a_{ij}}{\sqrt{a_{ii}} \sqrt{a_{jj}}} . \quad (3)$$

As licitações dos agentes são baseados numa pontuação acumulada para cada item, obtida a partir das classificações anteriores – as N melhores páginas são ordenadas e se a página selecionada a seguir está nesse conjunto então recebe uma pontuação $N-p+1$, onde p é a posição relativa do item. A esta pontuação adiciona-se a percentagem total, como forma de desempate:

$$Bid_{agent,item} = Score_{item} + \frac{1}{\#requests * N} * \sum_{i \in Items} Score_i . \quad (4)$$

4. Resultados

Os testes foram realizados *offline* e englobaram seis conjuntos de dados. Como a aplicação é capaz de desempenhar vários pedidos em paralelo, foi feita uma modificação de modo a que os pedidos fossem tratados sequencialmente. Cada pedido é processado apenas depois de o anterior ter sido completado. A tabela 1 mostra as características dos conjuntos de dados. Os primeiros quatro foram anteriormente usados em [26], mas este artigo mostra alguns melhoramentos que foram entretanto feitos.

Tabela 1. Características dos conjuntos de dados.

Dataset	#items	# records	#sessions	#records/#sessions	#records/#items
e-com	335	1409	413	3.41	4.21
e-gov	133	4047	1244	3.25	30.43
PE100	100	6070	803	7.56	60.70
PE200	200	2042	200	10.21	10.21
Listener	6428	49505	9570	5.17	7.70
Playlist	5428	25641	4267	6.01	4.72

Cada vez que um conjunto de recomendações é feito, uma das seguintes situações pode ocorrer:

- A. A recomendação é seguida.
- B. Um item não pertencente ao conjunto de recomendações foi seguido.
- C. O conjunto de recomendações estava vazio.
- D. Nenhum item foi seguido - final da sessão.

Vamos medir a precisão do sistema de recomendação como $A / (A + B + C)$.

Uma vez que a situação descrita em D não é considerada, o número de pedidos efetivamente consideradas para a avaliação é $\#requests - \#sessions$.

Para cada conjunto de dados, foram considerados os tamanhos de 1 a 10 para o conjunto de recomendações a serem feitas a cada vez. Consideramos também o suporte mínimo e confiança mínima no agente de regras de associação como tomando valores maiores do que zero.

Houve uma melhoria em ambos os algoritmos: se o algoritmo é incapaz de propor um conjunto completo de recomendações, este é completado com as recomendações anteriores dentro da mesma sessão. Isso significa que estamos assumindo que a relação de semelhança entre duas páginas Web consecutivas pode ser transitiva.

Nas tabelas 2 a 13, apresentamos a percentagem de sucesso para cada agente (AR - regras de associação; CF – filtragem colaborativa), e para o vencedor do leilão (W). As siglas minúsculas correspondem aos resultados dos algoritmos antes de o melhoramento descrito, enquanto as letras maiúsculas é para os algoritmos melhorados. Usamos a percentagem de sucesso para simplificar, e corresponde ao *recall* descrito em [Morais, Oliveira, Jorge 2012]. Em cada algoritmo individual, o resultado é a negrito se o algoritmo obteve o melhor resultado entre eles. O mesmo acontece com o vencedor do leilão, se a percentagem é melhor ou igual do que qualquer um dos algoritmos. Também mostramos os resultados considerando-se apenas os últimos 10% do conjunto de dados, depois de 90% das recomendações terem sido feitas.

Tabela 2. – Percentagem de sucesso de e-com

#R	1	2	3	4	5	6	7	8	9	10
ar	5.22%	9.24%	11.65%	13.25%	15.46%	16.67%	17.27%	18.27%	18.78%	19.38%
cf	4.82%	7.83%	9.84%	10.84%	12.35%	13.25%	14.66%	15.26%	15.96%	16.47%
w	6.02%	10.14%	13.55%	15.36%	17.97%	19.38%	20.78%	22.09%	23.19%	24.20%
AR	5.92%	10.34%	12.95%	15.86%	18.88%	20.28%	21.39%	22.59%	24.20%	24.80%
CF	4.92%	8.23%	10.54%	12.95%	14.66%	15.96%	18.17%	19.38%	20.28%	21.69%
W	6.12%	10.14%	12.85%	15.46%	18.27%	19.48%	21.08%	22.49%	23.59%	24.40%

Tabela 3. – Percentagem de sucesso de e-gov

#R	1	2	3	4	5	6	7	8	9	10
ar	13.20%	19.51%	24.22%	27.68%	31.18%	33.93%	36.46%	38.67%	40.35%	42.06%
cf	11.95%	18.27%	23.40%	27.29%	30.11%	32.36%	34.86%	36.50%	38.21%	39.56%
w	12.49%	19.27%	24.15%	27.79%	31.11%	33.61%	36.35%	38.35%	40.10%	41.96%
AR	13.24%	19.59%	24.40%	28.01%	31.57%	34.39%	37.25%	39.53%	41.42%	43.24%
CF	11.95%	18.52%	23.58%	27.58%	30.40%	32.68%	35.39%	37.39%	39.21%	40.53%
W	12.59%	19.37%	24.44%	28.18%	31.54%	33.93%	36.96%	39.14%	40.85%	42.53%

Tabela 4. – Percentagem de sucesso de PE100

#R	1	2	3	4	5	6	7	8	9	10
ar	6.15%	10.14%	13.35%	16.46%	19.42%	21.95%	24.23%	26.28%	28.52%	30.43%
cf	7.23%	11.98%	15.63%	18.68%	20.96%	23.49%	25.78%	27.64%	29.49%	31.23%
w	7.50%	11.96%	15.74%	18.63%	21.28%	24.09%	26.77%	29.07%	30.91%	32.73%
AR	6.25%	10.21%	13.48%	16.65%	19.63%	22.25%	24.63%	26.64%	28.86%	30.70%
CF	7.27%	12.04%	15.76%	18.83%	21.17%	23.79%	26.03%	27.76%	29.62%	31.35%
W	7.42%	11.90%	15.63%	18.42%	21.17%	24.02%	26.64%	28.73%	30.81%	32.39%

Tabela 5. – Percentagem de sucesso de PE200

#R	1	2	3	4	5	6	7	8	9	10
ar	3.26%	5.92%	7.71%	9.34%	10.53%	12.21%	13.36%	14.71%	16.02%	17.54%
cf	2.93%	5.81%	7.60%	9.28%	10.86%	12.00%	13.46%	14.98%	16.18%	16.83%
w	3.58%	6.41%	8.63%	10.37%	12.00%	13.84%	15.47%	16.99%	18.24%	19.49%
AR	3.69%	6.35%	8.25%	9.77%	11.02%	12.81%	14.28%	15.80%	17.21%	18.68%
CF	3.31%	6.03%	8.03%	9.83%	11.56%	12.70%	14.01%	15.53%	16.83%	17.92%
W	3.75%	6.24%	8.36%	9.93%	11.45%	13.25%	14.50%	16.23%	17.59%	18.78%

Tabela 6. – Percentagem de sucesso de Listener

#R	1	2	3	4	5	6	7	8	9	10
ar	15.37%	23.61%	28.29%	31.23%	33.29%	34.73%	35.85%	36.58%	37.16%	37.69%
cf	17.08%	25.31%	29.39%	31.86%	33.64%	34.81%	35.63%	36.18%	36.70%	37.11%
w	16.64%	24.60%	28.82%	31.59%	33.44%	34.77%	35.80%	36.49%	37.05%	37.60%
AR	15.46%	23.83%	28.62%	31.74%	33.94%	35.53%	36.81%	37.72%	38.44%	39.15%
CF	17.23%	25.62%	29.83%	32.50%	34.43%	35.74%	36.72%	37.44%	38.11%	38.67%
W	16.69%	24.81%	29.28%	32.29%	34.31%	35.83%	37.03%	37.91%	38.60%	39.29%

Tabela 7. – Percentagem de sucesso de Playlist

#R	1	2	3	4	5	6	7	8	9	10
ar	7.51%	13.44%	17.53%	20.61%	22.95%	25.01%	26.52%	27.73%	28.75%	29.67%
cf	8.38%	14.40%	18.71%	22.05%	24.52%	26.33%	27.88%	29.03%	29.92%	30.65%
w	8.27%	14.09%	18.55%	21.69%	24.09%	26.05%	27.57%	28.78%	29.75%	30.60%
AR	9.60%	17.22%	22.93%	26.98%	30.41%	32.89%	34.91%	36.45%	37.66%	38.93%
CF	10.42%	18.08%	23.61%	27.88%	30.96%	33.58%	35.54%	36.80%	38.01%	39.05%
W	10.59%	18.04%	23.93%	28.08%	31.21%	33.63%	35.61%	37.02%	38.23%	39.46%

Tabela 8. – Percentagem de sucesso de e-com (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	8.42%	13.68%	16.84%	20.00%	22.11%	22.11%	23.16%	23.16%	23.16%	25.26%
cf	11.58%	14.74%	15.79%	16.84%	20.00%	20.00%	21.05%	21.05%	22.11%	24.21%
w	11.58%	15.79%	17.89%	21.05%	22.11%	22.11%	24.21%	24.21%	25.26%	27.37%
AR	9.47%	15.79%	18.95%	23.16%	26.32%	26.32%	27.37%	27.37%	28.42%	30.53%
CF	12.63%	16.84%	16.84%	17.89%	23.16%	23.16%	25.26%	26.32%	26.32%	29.47%
W	12.63%	17.89%	20.00%	24.21%	27.37%	27.37%	28.42%	28.42%	28.42%	30.53%

Tabela 9. – Percentagem de sucesso de e-gov (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	13.59%	21.95%	28.57%	32.06%	37.63%	39.37%	40.77%	42.86%	44.95%	47.39%
cf	12.20%	17.77%	24.74%	28.92%	33.45%	36.93%	40.77%	43.21%	44.95%	45.99%
w	13.24%	21.60%	27.18%	31.36%	35.89%	37.98%	41.46%	42.86%	44.25%	46.69%
AR	13.59%	21.95%	28.57%	32.06%	37.98%	39.37%	40.77%	42.86%	44.95%	47.04%
CF	12.20%	17.77%	24.74%	28.92%	33.45%	36.93%	40.77%	43.21%	44.95%	45.99%
W	13.24%	21.25%	27.53%	31.71%	36.59%	37.63%	41.11%	43.21%	44.60%	46.69%

Tabela 10. – Percentagem de sucesso de PE100 (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	4.91%	8.68%	14.72%	18.11%	22.08%	23.58%	25.28%	26.98%	29.25%	31.70%
cf	5.85%	10.38%	15.66%	18.87%	21.51%	23.21%	25.85%	27.55%	29.25%	30.38%
w	6.23%	9.81%	16.60%	20.00%	23.02%	24.72%	27.17%	29.25%	31.51%	32.26%
AR	4.91%	8.68%	14.72%	18.30%	21.89%	23.40%	25.28%	26.79%	29.25%	31.70%
CF	5.85%	10.38%	15.47%	18.87%	21.51%	23.02%	25.66%	27.55%	29.25%	30.38%
W	6.04%	9.25%	16.04%	19.81%	22.64%	24.53%	26.98%	28.87%	31.32%	32.08%

Tabela 11. – Percentagem de sucesso de PE200 (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	5.18%	8.29%	8.81%	10.88%	13.47%	15.03%	16.58%	18.13%	20.21%	22.28%
cf	6.22%	10.88%	12.95%	13.99%	16.58%	17.62%	19.17%	21.24%	23.83%	24.87%
w	7.25%	10.88%	12.44%	14.51%	17.62%	19.17%	20.73%	22.28%	24.35%	25.91%
AR	5.18%	8.29%	8.81%	10.88%	13.47%	15.03%	17.10%	18.65%	21.24%	22.28%
CF	6.22%	10.88%	13.47%	14.51%	17.10%	18.65%	20.21%	21.76%	23.83%	24.87%
W	7.25%	10.88%	13.47%	15.54%	17.62%	19.17%	20.73%	22.80%	23.83%	25.39%

Tabela 12. – Percentagem de sucesso de Listener (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	18.24%	26.28%	31.72%	34.64%	36.80%	38.21%	39.52%	40.20%	40.76%	41.37%
cf	19.87%	29.00%	33.30%	36.00%	37.67%	38.89%	39.83%	40.49%	40.90%	41.41%
w	19.75%	27.91%	32.09%	34.95%	36.75%	38.23%	39.47%	40.17%	40.76%	41.44%
AR	18.39%	26.60%	32.09%	35.15%	37.43%	38.94%	40.32%	41.12%	41.75%	42.43%
CF	20.09%	29.39%	33.79%	36.68%	38.45%	39.74%	40.78%	41.49%	42.00%	42.58%
W	19.80%	28.10%	32.52%	35.68%	37.79%	39.23%	40.59%	41.41%	42.17%	42.90%

Tabela 13. – Percentagem de sucesso de Playlist (últimos 10%)

#R	1	2	3	4	5	6	7	8	9	10
ar	12.12%	21.37%	28.72%	33.05%	36.95%	40.84%	43.18%	45.34%	46.94%	48.80%
cf	12.02%	22.49%	29.16%	33.32%	37.42%	40.60%	44.16%	46.50%	48.46%	49.81%
w	13.68%	22.65%	29.73%	33.90%	37.69%	41.48%	44.23%	45.99%	47.58%	49.48%
AR	12.04%	21.13%	29.47%	33.97%	38.71%	41.61%	44.76%	47.05%	48.45%	49.95%
CF	12.44%	22.23%	28.97%	33.82%	39.11%	43.81%	47.35%	49.25%	50.50%	51.90%
W	13.39%	22.43%	30.02%	34.87%	39.26%	42.61%	45.60%	47.55%	49.00%	50.60%

5. Conclusões e trabalho atual e futuro

Vamos discutir os resultados, separando os dois maiores conjuntos de dados dos outros quatro. A tabela 14 mostra o número de casos em que o sistema de recomendação multiagente (SRMA), considerando o vencedor do leilão, tem uma maior percentagem de sucesso com o melhoramento dos algoritmos.

Tabela 14. – Comparação da percentagem de sucesso relativamente aos algoritmos melhorados

<i>Dados</i>	<i>Todos os dados</i>	<i>Últimos</i>
e-com	8	10
e-gov	10	5
PE100	0	0
PE200	1	3
Listener	10	10
Playlist	10	8

Como podemos observar, nos quatro conjuntos de dados mais pequenos, o SRMA tem melhores resultados em 19 de 40 casos, para o conjunto de dados em geral, e em 18 casos para os últimos 10%. Nos dois maiores conjuntos de dados, temos melhores resultados em todos os 20 casos, para todo o conjunto de dados, e em 18 casos para os últimos 10%.

Na tabela 15, mostramos o número de casos em que o SRMA tem uma percentagem de sucesso maior do que os algoritmos individuais iniciais (R) e os algoritmos melhorados (E)

Tabela 15. – Comparação do SRMA relativamente aos algoritmos individuais

<i>Dados</i>	<i>Todos os dados</i>		<i>Últimos 10%</i>	
	<i>R</i>	<i>E</i>	<i>R</i>	<i>E</i>
e-com	10	1	10	10
e-gov	1	2	1	2
PE100	8	7	9	9
PE200	10	8	9	10
Listener	0	5	1	2
Playlist	0	9	7	5

Na verdade, sem o melhoramento nos algoritmos, o SRMA tem uma precisão melhor do que os dois algoritmos individuais em 29 de 40 casos, para os quatro conjuntos de dados mais pequenos, enquanto isso nunca acontece nos dois maiores conjuntos de dados. Com os melhoramentos, o número de vezes em que a precisão é melhor para os quatro conjuntos de dados menores é de apenas 18 em 40, enquanto nos maiores conjuntos de dados temos agora 14 dos 20 casos em que isso acontece.

Quando consideramos apenas os últimos 10%, o comportamento do SRMA ao utilizar algoritmos sem ou com melhorias agora é mais semelhante: 29/31 de 40 casos para os quatro conjuntos de dados mais pequenos, e 8/7 de 20 casos para os dois maiores conjuntos de dados.

Olhando para esses resultados, podemos considerar, com particular enfoque para os resultados dos maiores conjuntos de dados, que a transitividade da relação de

semelhança entre as páginas Web se torna mais forte quando o número de sessões aumenta.

Olhando para o desempenho do SRMA, ele foi capaz de produzir resultados mais precisos na maioria dos casos, e quando isso não aconteceu, ficou perto do melhor dos dois algoritmos individuais e sempre melhor do que o pior.

Atualmente estão a ser realizados testes *online* cujos resultados estão ainda em fase de análise. Para teste do sistema foi criado um sítio Web de fotografia [www.1000palavras.pt], que foi adaptado de modo a acolher o sistema de recomendação.

O sítio Web inclui 344 fotografias distribuídas por 8 galerias, num total de 353 páginas Web se, além das páginas das fotos, considerarmos também a página principal e as 8 páginas de galerias. Tratando-se de um sítio Web de fotografia, o sistema de recomendação foi adaptado de forma a recomendar fotos pertencentes ao sítio, pelo que se deve considerar um conjunto de dados com 344 itens.

Concluídos os trabalhos de adaptação, os testes decorrem em duas fases. Na primeira fase o sítio é disponibilizado *online*, mas com o sistema de recomendação desligado; na segunda, o sítio Web é disponibilizado com o sistema de recomendação ativo.

No futuro poderemos testar outras abordagens, como a inserção de diferentes agentes no sistema, incluindo abordagens híbridas.

Agradecimentos

Este trabalho foi financiado por: Fundação para a Ciência e a Tecnologia (FCT), Ref. SFRH/BD/27571/2006; ERDF - European Regional Development Fund através do programa COMPETE (operational programme for competitiveness), através do projeto FCT FCOMP -01-0124-FEDER-022701; PRODEI – FEUP.

Referências

Adomavicius, G. and Tuzhilin, A., 2005. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, June.

AJAX (Asynchronous Javascript And XML), Mozilla Developer Center: <https://developer.mozilla.org/en/ajax>. Data de consulta: 20/01/2014.

Albayrak, S., Wollny, S., Varone, N., Lommatzsch, A., and Milosevic, D., 2005. Agent technology for personalized information filtering: the pia-system. In SAC'05: Proceedings of the 2005 ACM symposium on Applied computing, pages 54–59, New York, NY, USA. ACM Press.

Apache Derby, Website: <http://db.apache.org/derby/>. Data de consulta: 20/01/2014.

Ardissono, L., Goy, A., Petrone, G., and Segnan, M., 2005. A multi-agent infrastructure for developing personalized web-based systems. ACM Trans. Inter. Tech., 5(1):47–69.

Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T., 1995. WebWatcher: A learning apprentice for the world wide web. In Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, pages 6–12, California.

Balabanovic, M. and Shoham, Y., 1997. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66-72.

Basilico, J., Hofmann, T., 2004. Unifying collaborative and content-based filtering. In Proceedings of ICML'04, Twenty-first International Conference on Machine Learning. ACM Press, New York.

Berendt, B., 2002. Using Site Semantics to Analyze, Visualize, and Support Navigation. In *Data Mining and Knowledge Discovery*, Volume 6, Issue 1, Pages 37 – 59.

Borges, J. L., 2000. A Data Mining Model to Capture User Web Navigation Patterns, PhD thesis, University College London, University of London.

Breese, J., Heckerman, D., and Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Cooper, G., and Moral, S. (Eds.), Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98),. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 43-52.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S., 2003. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. In *Data Mining and Knowledge Discovery*, Volume 7, Issue 4, Pages 399 – 424.

Canny, J., 2002. Collaborative filtering with privacy via factor analysis. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02). ACM, New York, NY, USA, 238-245.

El-Ramly, M., Stroulia, E., 2004. Analysis of Web-usage behavior for focused Web sites: a case study. In *Journal of Software Maintenance and Evolution: Research and Practice*, 16(1-2), Pages: 129-150.

Fink, J., Kobsa, A., and Nill, A., 1996. User-oriented adaptivity and adaptability in the AVANTI project. In *Designing for the Web: Empirical Studies*, Microsoft Usability Group, Redmond, Washington.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C., 2001. Eigenstate: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133-151.

Huang, Z., Chen, H., and Zeng, D., 2004. Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. *ACM Transactions on Information Systems*, 22(1):116-142.

Ishikawa, H., Ohta, M., Yokoyama, S., Nakayama, J., Katayama, K., 2002. Web usage mining approaches to page recommendation and restructuring. In *International Journal of Intelligent Systems in Accounting, Finance & Management*, 11(3), Pages: 137-148.

JADE (Java Agent DEvelopment Framework), Website: <http://jade.tilab.com>. Data de consulta: 20/01/2014.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G., 2011. *Recommender Systems – An Introduction*. Cambridge.

Jennings, N. R., 2001. An agent-based approach for building complex software systems. In *Communications of the ACM*, vol. 44, no. 4, pp. 35-41.

Jorge, A., Alves, M. A., Grobelnik, M., Mladenic, D., and Petrak, J., 2003. Web Site Access Analysis for A National Statistical Agency. In *Data Mining And Decision Support: Integra-tion And Collaboration*, Mladenic, D., Lavrac, N., Bohanec, M., Moyle, S., Kluwer Aca-demic Publishers.

Kephart, J. O., 2005. Research challenges of autonomic computing. In *ICSE '05: Proceed-ings of the 27th International Conference on Software Engineering*, pages 15–22, New York, NY, USA. ACM Press.

Linden, G.; Smith, B.; and York, J. 2003. Amazon.com recommendations. Item-to-Item Collaborative Filtering. In *IEEE Internet Computing* 3:76–80.

Based Distributed Miner., In *Second International Conference on Web Information Systems Engineering (WISE'01)*, Volume 1, p. 0288.

Miranda, C., and Jorge, A. M., 2008. Incremental Collaborative Filtering for Binary Ratings. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Volume 01 (WI-IAT '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 389-392.

Mobasher, B., Dai, H., Luo, T., Nakagawa, M., 2002. Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. In *Data Mining and Knowledge Discovery*, Kluwer Publishing, Vol. 6, No. 1, pp. 61-82.

Morais, A. J., Oliveira, E., and Jorge, A. M., 2012. A Multi-Agent Recommender System. In S. Omatu et al. (Eds.): *9th International Conference on DCAI'12, AISC 151*, pp. 281-288.

Morais, A. Jorge, 2009. A Multi-Agent Approach for Web Adaptation. In Y. Demazeau et al. (Eds.): *7th International Conference on PAAMS'09, AISC 55*, pp. 349-355. Springer-Verlag Berlin Heidelberg.

Pazzani, M. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6):393-408.

Perkowitz, M. and Etzioni, O., 2000. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(2000):245–275.

Resnick, P. and Varian, H., 1997. Recommender Systems, introduction to special section of *Communications of the ACM*, March, vol. 40(3).

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (Eds.), 2011. Recommender Systems Handbook. Springer.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T., 2000. Analysis of recommendation algorithms for E-commerce. In Proceedings of the 2nd ACM Conference on Electronic Commerce. ACM. vii+271, 158-67.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. In WWW, pages 285–295.

Schafer, J.B., Frankowski, D., Herlocker, J., and Sen, S., 2006. Collaborative Filtering Recommender Systems, in The Adaptive Web, Springer Verlag.

Sítio de fotografia 1000 palavras: <http://www.1000palavras.pt/>. Data de consulta: 20/01/2014.

Spiliopoulou, M. and Faulstich, L. C., 1998. WUM: a tool for web utilization analysis. In Proceedings of the International Workshop on the Web and Databases, pages 184–203, Va-lencia, Spain.

Wei, Y. Z., 2005. A Market-Based Approach to Recommendation Systems, PhD thesis, University of Southampton.

Wooldridge, M., 2009. An Introduction to MultiAgent Systems, 2nd edition. John Wiley & Sons.



A. Jorge Morais, nascido a 12 de Novembro de 1972, em Luanda, Angola. Licenciado em Matemática Aplicada, ramo de Ciência de Computadores, Faculdade de Ciências da Universidade do Porto, 1994. Mestre em Informática, especialização em Ciências da Computação, Universidade do Minho, 1998. Doutoramento em Engenharia Informática, Faculdade de Engenharia da Universidade do Porto, 2013. Lecionou no Instituto Superior de Engenharia do Porto, desde 3 de Janeiro de 1995, nas áreas e respetivos departamentos, de Matemática e Engenharia Informática. Atualmente é Professor Auxiliar na Universidade Aberta, na área de Informática, no Departamento de Ciências e Tecnologia, onde leciona desde 1 de Junho de 2003.