Um Sistema de Recomendação Baseado em Nuvem

Ricardo Batista Rodrigues[1], Frederico A. Durão[3], Rodrigo E. Assad[2], Vinicius C. Garcia[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária – 50.740-560 – Recife – PE-Brasil

[2]Universidade Federal Rural de Pernambuco (UFRPE)
Rua Dom Manoel de Medeiros, Campus Dois Irmãos – 52.171-900 – Recife – PE – Brasil

[3]Instituto de Matemática – Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, Campus de Ondina – 40170-110 – Salvador – BA – Brasil

{rbr,vcg}@cin.ufpe.br, freddurao@dcc.ufba.br, rodrigo.assad@gmail.com

Resumo. *Os sistemas de recomendação têm como objetivo amenizar a sobre-carga de informação auxiliando usuário na busca pela informação desejada. Este artigo apresenta um mecanismo de recomendação de arquivos baseado em nuvem, em um ambiente de armazenamento de dados na nuvem. Com o Sistema de Recomendação, os usuários recebem recomendações de arquivos que são similares as suas preferencias, baseada nos arquivos no qual o usuário salva em sua conta no ambiente. Ao mesmo tempo, que os arquivos recomendados pelo sistema atendem os fatores da nuvem, recomenda ao usuário, arquivos com maior disponibilidade no ambiente.*

Abstract. *The recommendation systems aim to minimize information overload by helping users in searching desired information. This paper presents a recommendation engine for cloud-based files in an environment of data cloud storage. With the Recommendation System, users receive recommendations of files that are similar to their preferences based on files saved by the user in his account on the environment. At the same time that the recommended system files meet the factors of the cloud assures the user file recommendations with greater availability in the environment.*

1. Introdução

Com o advento da computação em nuvens, surgiram os sistemas de armazenamento em nuvem, que possibilitam aos seus usuários armazenar arquivos na nuvem. Com o crescimento da utilização destes sistemas, a massa de dados armazenados na nuvem se tornou humanisticamente impossível de ser processada, implicando na ocultação de informações relevantes aos usuários, que deixam de descobrir novos conteúdos por não disporem de meios eficientes que os auxilie na filtragem de dados em busca de conhecimento relevante e que atenda suas expectativas. Diante deste cenário, sistemas de recomendação se tornam uma alternativa, para auxiliar os usuários na tomada de decisão por qual arquivo escolher e a filtrar informações relevantes em meio a uma imensidão de dados.

Sistemas de recomendação (SR) são *softwares* e técnicas que fornecem sugestões de itens para usuários [Pazzani and Billsus 1997] [Phelan et al. 2009]. Esses sistemas fazem parte de nossas vidas, diariamente nos deparamos com recomendações via *email* ou em páginas na *web*. Muitas lojas *online* e plataformas oferecem serviços de recomendação, por exemplo, Amazon (http://www.amazon.com) e BarnesAndNoble (http://www.barnesandnoble.com) [Melville et al. 2002]. Existem duas abordagens predominantes na construção de SR, Filtragem Colaborativa (CF) e Filtragem Baseada em Conteúdo (CB). Sistemas CF recomendam itens que são similares às características do usuário, por exemplo, o seu perfil em uma rede social. Sistemas CB recomendam ao usuário itens semelhantesãqueles em que ele demonstrou interesse em experiências anteriores. Para tanto, o sistema analisa as descrições dos conteúdos dos itens avaliados pelo usuário para montar o seu perfil, o qual é utilizado para filtrar os demais itens da base [Blanco-Fernandez et al. 2008] [Ricci et al. 2011] [Phelan et al. 2009].´

Os sistemas de recomendação têm por objetivo reduzir a sobrecarga de informação, realizando filtragem de itens baseado nos interesses do usuário. Das diversas´ técnicas existentes para realizar tal tarefa, a abordagem utilizada neste artigo é a Filtragem Baseada em Conteudo, que se baseia em arquivos no qual o usuário demostrou interesse no passado [Shardanand and Maes 1995].

Este trabalho apresenta um mecanismo de recomendação em um ambiente de armazenamento de dados na nuvem. Na geração de recomendações e utilizada a técnica de Filtragem baseada em conteudo e características da nuvem. O objetivo do modelo de recomendação aqui propostoé recomendar ao usuário, arquivos que sejam similares as suas preferências e que atendam os fatores da nuvem, desta forma, um arquivo recomendado ao usuário, sempre estará disponível e acessível no ambiente de armazenamento em nuvem, além de proporcional redução no tempo gasto no *download* de um arquivo recomendado e na filtragem de conteúdo relevante em meio a imensidão de dados disponíveis na nuvem.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos˜ relacionados. A Seção 3 apresenta o modelo proposto e os resultados. Na Seção 4 são apresentadas as conclusões e os trabalhos futuros deste trabalho.

2. Trabalhos Relacionados

Existem alguns trabalhos na literatura que discutem questões a respeito de sistemas de recomendações em nuvem. Nesta Seção, serão apresentados alguns deles, enfatizando a similaridade e as diferenças em relação ao modelo de recomendação proposto nesta pesquisa.

Lee [Lee et al. 2010], apresenta uma proposta de SR que utiliza dados armazenados na nuvem para proverem suas recomendações, distinguindo-se da proposta deste trabalho que além de recomendar arquivos armazenados na nuvem, tem como objetivo utiliza fatores da nuvem para garantir a disponibilidade dos arquivos recomendados aos usuários. Lai [Lai et al. 2011], apresentam o trabalho que mais se aproxima da proposta desta pesquisa. Onde propõem um SR de programas de TV em nuvem, com objetivo principal de oferecer um sistema escalável, que tenha uma alta taxa de disponibilidade para o sistema.

O modelo proposto nesta pesquisa utiliza fatores da nuvem para gerar suas recomendações, para garantir a disponibilidade dos arquivos recomendados e a

economia do tempo gasto para download de arquivos recomendados, e que as recomendações atendam as preferências dos usuários.

3. O Sistema de Recomendação

Esta seção descreve o desenvolvimento do SR resultado desta pesquisa, onde apresentamos o ambiente de desenvolvimento, o modelo de recomendação, os fatores utilizados na geração de recomendações, e os resulados desta pesquisa.

3.1. Ambiente de Desenvolvimento

O SR desenvolvido nesta pesquisa foi implementado em um ambiente real de armazenamento de dados na nuvem, o Ustore (http://usto.re), que consiste em uma solução para armazenamento e *backup* de arquivos em nuvem privada, que tem como proposta principal as funcionalidades de restore de arquivos (*download*, *upload* e compartilhamento), o ambiente tem como objetivo, prover segurança, alta disponibilidade, ganho de desempenho e redução no tempo de resposta, assim como a utilização de recursos computacionais ociosos.

O ambiente Ustore permite a qualquer usuário fazer *upload*, *download* e compartilhamento de arquivos, disponibilizando a opção de tornar os seus arquivos públicos ou por *default* privados. O SR desenvolvido utiliza em suas recomendações os arquivos marcados como públicos para gerar novas recomendações aos usuários. Na geração de novas recomendações, o SR calcula a similaridade entre o conteúdo dos arquivos públicos armazenados na nuvem do ambiente e os arquivos que o usuário demonstrou as suas preferências, e, em seguida, aplica os fatores da nuvem, propostos nesta pesquisa e apresentados nas seções a seguir.

3.2. Modelo de Recomendação

O modelo de recomendação proposto nesta pesquisa e baseado em fatores da nuvem é em conteudo. Seguindo este modelo, são recomendados aos usuários arquivos similares aos que eles demonstraram interesse no passado, representados pelos arquivos salvos pelo usuário em sua conta no ambiente de armazenamento de dados em nuvem, e que atendam os fatores da nuvem. Na geração de recomendações, propomos a utilização de cinco fatores:

- Disponibilidade
- Similaridade
- Taxa de Download
- Quantidade de Downloads
- Tamanho do Arquivo

A seguir, são delineados cada fator e o processo de recomendacão.

Fator Disponibilidade: refere-se ao número de horas em que um peer está disponível na nuvem. Um arquivo só poderá ser recomendado ao usuário se o peer que o armazena estiver disponível, tornando possível o *download* deste arquivo. A Equação a seguir apresenta o cálculo do fator disponibilidade.

$$Dp = h_i \cdot \frac{1}{n}$$

Na equação anterior, é calculado o fator disponibilidade, ondé *h* e a quantidade de tempo em que um peer *i* está disponível na rede, e *n* representa o número total de horas que um *peer* pode tornar-se disponível na rede (24 horas). O número de horas que um *peer i* está disponível na rede e normalizado para um valor entre 0 e 1. O exemplo a seguir mostra como o fator disponibilidade contribui para a geração de uma recomendação: considerando que dois arquivos A e B são semelhantes, o arquivo A está armazenado em um peer que se encontra disponível na rede de 14 a 16 horas, num total de duas horas de disponibilidade. O arquivo B está armazenado em outro *peer*, que está disponível na rede de 14 a 18 horas, totalizando 4 horas de disponibilidade. Desse modo, o arquivo B e o arquivo que deve ser recomendado ao usuário, pois se encontra disponível na rede por um tempo maior do que o arquivo A, possibilitando a realização do *download*. O objetivo é diminuir o risco de o utilizador não conseguir fazer a transferência.

Fator Similaridade: refere-se a similaridade entre o conteúdo do arquivo armazenado na nuvem e o arquivo pelo qual o usuário demonstrou preferência. Para extrair os arquivos foi usado o Apache Lucene (http://lucene.apache.org), um mecanismo de busca de alta performance, e o Apache Tika (http://tika.apache.org), um detector e extrator de conteúdo de metadados e texto estruturado, podendo ser utilizado para a extração de conteúdo de arquivos de diversos formatos, como HTML, XML, OLE2 e OOXML do Microsoft Office, Opendocument Format, PDF, ePUD, RTF, arquivos compactados e empacotados. Este fator é obtido a partir da técnica *Cosine Similarity*, que e a diferença angular entre dois vetores, através do cálculo do cosseno do ângulo entre eles, independente de seus tamanhos [Baeza-Yates and Ribeiro-Neto 1999]. O resultado do cálculo será sempre um valor entre 0 e 1, onde 0 significa 0% de similaridade e 1 significa 100% de similaridade. A Equação a seguir mostra o cálculo da similaridade.

$$St = cos(\theta) = \frac{A \cdot B}{\| A \| \cdot \| B \|}$$

Na equação acima, calcula-se a similaridade de dois vetores A e B, de onde se obtém o produto de A e B e se calcula a magnitude dos vetores A e B. Tais magnitudes são multiplicadas e divididas pelo produto escalar dos vetores A e B.

Fator Taxa de Download: refere-se a taxa de *download* disponível para a realização do *download* de um arquivo que foi recomendado. O objetivo é recomendar arquivos vindos de peers, que são dados com uma conexão melhor para o usuário do *peer*. Este fator e medido de 0 a 20 megabits por segundo (Mbps). A Equação a seguir mostra o cálculo do referido fator.

$$Td = ns \cdot (\frac{1}{n})$$

Na Equação acima é calculado o fator taxa de download *Td*, onde *ns* e a taxa de transferência de rede Mbps, sendo este valor normalizado para um valor entre 0 e 1, em que *n* representa o valor mais alto da taxa de *download* na rede MBps.

Fator Quantidade de Download: corresponde ao número de *downloads* de um determinado arquivo no ambiente. Isso indica a popularidade e a importância social de um arquivo na mesma rede, sinalizando que muitos usuários se interessaram por um mesmo arquivo. A Equação abaixo mostra o cálculo desse fator.

$$Qd = Q_D \cdot (\frac{1}{0})$$

A Equação a cima calcula o fator quantidade de Download. Para cada *download* realizado de um arquivo específico, o contador de *download* de arquivos e incrementado por 1, este valor e medido de 0 a n, onde *n* corresponde ao maior numero de *downloads* realizados de um único arquivo na rede. O valor de *n* é obtido por meio da observação do histórico de *downloads* de arquivos do ambiente. O cálculo do fator, o número de *downloads* de um arquivo *Qd* é normalizado para um valor de 0 a 1, multiplicando a quantidade Q_D de *downloads* pela normalização limiar obtida pela notação $(\frac{1}{n})$ que divide 1 por *n*, que representa o maior número de download de um arquivo realizado no mesmo ambiente.

Fator Tamanho do Arquivo: refere-se ao tamanho do arquivo a ser recomendado. Esse fator penaliza o score de recomendação quando a taxa de *download* e baixa. Por exemplo, um arquivo A e semelhante ao arquivo B, o arquivo A tem tamanho igual a 9 gigabytes e sua taxa de *download* e de 600 Kbps, já o arquivo B possui 2 gigabytes de tamanho e sua taxa de *download* e de 1 Mbps. Assim, o arquivo B terá maior score de recomendação, por apresentar as melhores condições para a realização de seu *download* (menor tamanho e maior taxa de *download*). A equação a seguir mostra o cálculo desse fator.

$$S = (\frac{2^{30}}{S_b}) \cdot \frac{1}{n}$$

Na equação a cima, o fator tamanho e representado por *S*, que corresponde ao tamanho do arquivo a ser recomendado, o qual e obtido em bytes e convertido em gigabytes (GB). A conversão e calculada dividindo-se 1 GB 2^{30} pelo tamanho do arquivo em Bytes S_b, obtendo-se, desta forma, o tamanho do arquivo em gigabytes. O tamanho do arquivo e multiplicado por $\frac{1}{n}$, que será representado por um valor de 0 a 1, sendo o valor 1 dividido por *n* correspondente ao tamanho máximo de um arquivo que pode ser armazenado no ambiente Ustore. Este valor pode ser definido nas configurações pelo administrador, durante o desenvolvimento desta pesquisa, o valor de *n* no sistema foi configurado para 10 gigabytes.

Os fatores são ponderados por pesos ω, a depender da sua relevância na geracão de uma recomendação. Os pesos dos fatores são multáveis e configuráveis pelo administrador do sistema, e podem variar de acordo com as características do ambiente. Inicialmente, foram definidos considerando a relevância de cada fator no sistema de recomendações desenvolvido nesta pesquisa e as características do ambiente de armazenamento utilizado. Na Tabela 1 apresentamos os pesos de cada fator.

O fator Similaridade tem peso 5 e representa 50% do *score* de recomendação, para garantir que o conteúdo de um arquivo recomendado ao usuário seja similar ao conteudo do arquivo que o usuário demonstrou interesse. Um arquivo que possua similaridade 0 atribuída as preferências do usuário não deve ser recomendado.

Tabela 1. Pesos dos Fatores

Fator	Peso
Similaridade	5

Disponibilidade	2
Taxa de Download	1
Quantidade de Downloads	1
Tamanho do Arquivo	1

O fator Disponibilidade tem peso 2, o que representa 20% do *score* de recomendação, por designar o tempo em que um peer de dados está disponível na rede, tornando possível o *download* de um arquivo recomendado. Um arquivo somente poderá ser recomendado ao usuário se o mesmo estiver armazenado em um *peer* que esteja disponível.

O fator Taxa de Download tem peso 1, representando 10% de uma recomendação. Este fator simboliza a taxa de *download* disponível para realizar o *download* de um arquivo recomendado ao usuário. Um arquivo que possua uma baixa taxa de *download* e que tenha um tamanho maior que o dos demais arquivos similares a ele, não deverá ser recomendado ao usuário, pois o seu processo de recomendação demandará mais tempo e processamento.

Ao fator Quantidade de Downloads e atribuído o peso 1, o que corresponde a 10% do *score* de recomendação. Este fator tem o seu peso inferior aos demais por não ser um fator crítico. Desta forma, um arquivo que não seja popular na nuvem pode ser recomendado ao usuário, o mesmo ocorre com arquivos novos na rede.

Ao fator Tamanho do Arquivo e atribuído o peso 1. Este fator tem peso inferior aos demais fatores por não ser um fator crítico. Desta forma, um arquivo que tenha o tamanho máximo aceito pelo ambiente (10 Gigabytes) pode ser recomendado se a sua taxa de *download* for proporcional, garantindo bom desempenho no *download* do arquivo.

As recomendações são representadas pelo *score* de recomendação, que e o resultado do cálculo mostrado na Equação a seguir.

$$Score = (((St \cdot \omega_S) \cdot (Dp \cdot \omega_D)) \cdot ((Td \cdot \omega_T) + (Qd \cdot \omega_Q))) \cdot \frac{1}{n} - ((S \cdot \omega_Z) \cdot \frac{1}{n})$$

No cálculo apresentado na Equação anterior, o *score* de recomendação é igual ao resultado da soma dos fatores Taxa de Download *Td* e Quantidade de Downloads *Qd*, multiplicados por seus respectivos pesos ω_T, ω_Q. O resultado desta notação é multiplicado pelo produto dos fatores Similaridade *St* e Disponibilidade *Dp*, multiplicados por seus pesos ω_S, ω_D, e normalizado por $\frac{1}{n}$, sendo *n* o maior valor possível desta Equação. Recomendações com score igual ou inferior a 0 são descartadas. Um arquivo somente é recomendado ao usuário se os valores de seus fatores Similaridade e Disponibilidadé forem maiores que 0. Desta forma, o conteúdo de um arquivo recomendado ao usuário sempre sera similares preferências do usuário, e sempre estará disponível para *download*. As recomendações são ordenadas de forma decrescente, pelo valor do *score* obtido na Equação anterior.

3.3. Resultados

O experimento realizado neste trabalho foi executado em um ambiente real de armazenamento de dados na nuvem. O experimento apresentado a seguir, apresenta resultados parciais desta pesquisa, gerado pela simulação de usuários utilizando o sistema. Os principais objetivos desses experimentos são: Analisar a relevância dos arquivos recomendado em relação a preferência elicitada pelo usuário.

O experimento realizado neste trabalho avaliou as recomendações realizadas pelo sistema. Neste experimento foi utilizada uma base de dados contendo 100 artigos científicos de domínio público, a partir desta base na nuvem, foram solicitadas recomendações para arquivos de distinto conteúdo. No total foram avaliadas 50 recomendações, que foram avaliadas como *Like* ou *Dislike*. No caso em que uma recomendação não atenda as preferências e expectativas do avaliador a mesma deve receber a avaliação *Dislike*, ou *Like* no caso da recomendação atender as preferências do avaliador. A Figura 1 apresenta o resultado das avaliações realizadas.

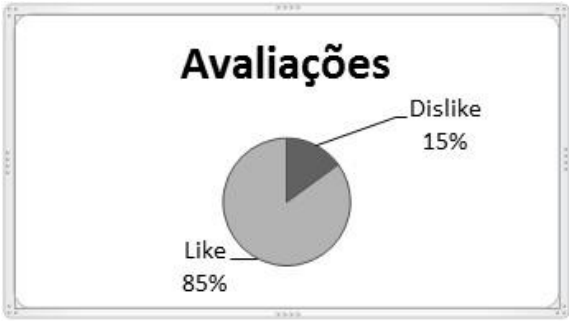

Figura 1. Resultado das Avaliações

A partir da análise dos valores apresentados na Figura 1, inferimos que 85% das recomendações receberam avaliações positivas, o que representa que, a maior parte das recomendações geradas atenderam as expectativas do avaliador. Desta forma, valida as recomendações geradas e o modelo de recomendação proposto.

Para atingir e validar os objetivos desta pesquisa serão realizados experimentos em um ambiente de armazenamento em nuvem disponível em um meio acadêmico, utilizando usuários reais, para elicitarem suas preferências e avaliarem as recomendações recebidas. Assim como a realização de outros experimentos que possam apresentar os ganhos em termo de tempo de *download* dos arquivos recomendados, para que possa se validar o modelo proposto nesta pesquisa.

4. Conclusão e Trabalhos Futuros

Este artigo investiga o impacto de fatores oriundos da nuvem, na geração de recomendações em um ambiente de armazenamento em nuvem. Foi apresentado o modelo proposto nesta pesquisa, bem como os fatores que formam o modelo proposto baseado em nuvem. O desenvolvimento do sistema e os experimentos iniciais foram implementados e executados em um ambiente real de armazenamento de dados na nuvem.

Como trabalhos futuros, julga-se importante refazer e melhorar os experimentos apresentados neste artigo, utilizando usuários reais do ambiente na nuvem, assim como realizar novos experimentos, a fim de comparar os resultados obtidos neste modelo com

os demais modelos disponíveis na literatura. Particularmente, pretende-se propor novos fatores baseados em nuvem, que possam contribuir para a melhoria do modelo proposto.

Referências

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. AddisonWesley Longman Publishing Co., Inc., Boston, MA, USA.

Blanco-Fernandez, Y., Arias, J. J. P., Gil-Solla, A., Cabrer, M. R., and Nores, M. L.´ (2008). Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Trans. Consumer Electronics*, 54(2):727–735.

Lai, C.-F., Chang, J.-H., Hu, C.-C., Huang, Y.-M., and Chao, H.-C. (2011). Cprs: A cloudbased program recommendation system for digital {TV} platforms. *Future Generation Computer Systems*, 27(6):823 – 835.

Lee, S., Lee, D., and Lee, S. (2010). Personalized dtv program recommendation system under a cloud computing environment. *Consumer Electronics, IEEE Transactions on*, 56(2):1034–1042.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Pazzani, M. J. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331.

Phelan, O., McCarthy, K., and Smyth, B. (2009). Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys ’09, pages 385–388, New York, NY, USA. ACM.

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

Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating ”word of mouth”. pages 210–217. ACM Press.