Anais do XVIII Encontro de Iniciação Científica – ISSN 1982-0178
Anais do III Encontro de Iniciação em Desenvolvimento Tecnológico e Inovação – ISSN 2237-0420
24 e 25 de setembro de 2013

ALGORITMO HÍBRIDO PARA SISTEMAS DE RECOMENDAÇÃO UTILIZANDO FILTRAGEM COLABORATIVA E ALGORITMO GENÉTICO

Renan de Oliveira Yamaguti
Faculdade de Engenharia de Computação / CEATEC
renan.yamaguti@terra.com.br

Juan Manuel Adán Coello
Grupo de Pesquisa em Sistemas Inteligentes
CEATEC
juan@puc-campinas.edu.br

Resumo: *A expansão da Internet trouxe uma grande quantidade de informações, tornando cada vez mais trabalhoso encontrar informações de real interesse para os usuários. Os sistemas de recomendação baseados em filtragem colaborativa surgem como uma alternativa para aliviar esse problema, na medida em que procuram sugerir itens de interesse para os usuários. Nesta abordagem, recomendações são baseadas nas avaliações explicitas ou implícitas que um usuário faz dos itens acessados. Entre os principais desafios desse tipo destaca-se oferecer precisão aceitável, especialmente quando a matriz onde se registram as avaliações da comunidade de usuários é esparsa. Este artigo apresenta um algoritmo híbrido que combina os resultados de dois algoritmos de filtragem colaborativa utilizando um algoritmo genético. Na avaliação experimental realizada, o algoritmo híbrido obteve maior precisão que os dois algoritmos de filtragem colaborativa isolados, porém com um elevado custo computacional.*

Palavras-chave: *Sistemas de recomendação, filtragem colaborativa, algoritmo genético.*

Área do Conhecimento: *Ciências Exatas e da Terra –Ciência da computação.*

1. INTRODUÇÃO

A quantidade e diversidade de informações disponíveis na Internet desafiam os seres humanos a encontrarem algo de real relevância. Neste contexto, os Sistemas de Recomendação (SR) surgem como uma ferramenta útil para minerar a informação no mar da Internet. Tais sistemas varrem de forma autônoma o espaço de opções e preveem se itens ainda não acessados pelos usuários podem ser de seu interesse. Um item corresponde a qualquer recurso de informação, como um livro, uma página Web, uma música, um filme, um usuário a ser seguido no *Twitter* ou adicionado no *Facebook* ou ainda um canal do *Youtube*.

Entre os atributos usados para classificar os SR destaca-se a abordagem empregada para fazer a filtragem da informação a recomendar, sendo as principais a filtragem demográfica, a filtragem colaborativa e a filtragem baseada em conteúdo. Muitos sistemas combinam duas ou mais dessas abordagens, dando origem aos métodos híbridos de filtragem [1].

A filtragem demográfica caracteriza os usuários por informações demográficas como idade, local de residência e preferências, podendo criar estereótipos, por exemplo: meninas de 14 anos que moram no Reino Unido gostam de Jonas Brothers. O principal problema dessa abordagem é que as recomendações são muito genéricas, não conseguindo capturar os gostos específicos de pessoas com um mesmo perfil demográfico.

A Filtragem Colaborativa (FC) é uma abordagem que leva em conta o histórico das avaliações feitas por um usuário (ou recebidas por um item), e a sua relação com as avaliações feitas por outros usuários (ou recebidas por outros itens). A FC funciona construindo uma matriz usuário-item, na qual cada posição da matriz contém a avaliação (*rating*) dada para um item por um usuário.

Uma das principais dificuldades da FC é lidar com matrizes esparsas, ou seja, matrizes com poucas posições preenchidas. Outro problema importante é denominado de partida a frio (*cold-start),* quando se tem um novo usuário ou um novo item. Neste cenário, o usuário não avaliou nenhum item e o item não foi avaliado por nenhum usuário, o que impede que se estabeleçam relações com as avaliações feitas por outros usuários (ou recebidas por outros itens).

Na filtragem baseada em conteúdo, o SR coleta informações que descrevem os itens e então, baseado nas preferências dos usuários, prevê que itens serão do interesse do usuário. Entre os problemas desta abordagem destaca-se a superespecialização, já que neste tipo de sistema só serão recomendados itens semelhantes a outros que foram avaliados previamente pelo usuário, não havendo a exploração de novas

categorias de itens.

Os métodos híbridos compõem duas ou mais abordagens de filtragem a fim de reduzir as suas limitações ou aumentar o seu desempenho.

O objetivo do trabalho descrito neste artigo é precisamente esse: combinar duas abordagens conhecidas de filtragem colaborativas, visando melhorar a precisão das recomendações realizadas.

O restante deste artigo está organizado da seguinte forma: na seção 2 aprofunda-se a discussão sobre filtragem colaborativa, apresentando detalhes dos algoritmos usados no trabalho descrito neste artigo; na seção 3 faz-se uma introdução aos algoritmos genéticos, abordagem usada neste trabalho para criar um algoritmo híbrido; na seção 4 apresenta-se o algoritmo híbrido proposto e implementado que combina dois algoritmos de filtragem colaborativa usando um algoritmo genético; na seção 5 os resultados de uma avaliação experimental desse algoritmo são apresentados e, finalmente, na seção 6 são apresentadas algumas considerações finais.

2. FILTRAGEM COLABORATIVA (FC)

Os algoritmos de filtragem colaborativa utilizam uma matriz usuário-item para prever a nota que um usuário *u* daria para um item *i* ainda não avaliado por esse usuário. No restante desta seção serão apresentadas duas abordagens usadas com esse propósito. A primeira baseia-se no algoritmo dos k vizinhos mais próximos (*K-Nearest-Neighbors* – KNN) e a segunda nas tendências verificadas de cada ao avaliar itens e dos itens ao serem avaliados.

2.1. O KNN

Muito utilizado na FC o algoritmo KNN busca encontrar semelhanças entre usuários (ou entre itens), o conjunto dos k usuários (ou itens) mais semelhantes a um usuário (ou item) recebe o nome de vizinhança desse usuário (item). A partir da vizinhança prevê-se uma avaliação para um item da matriz usuário-item ainda não avaliado pelo item alvo.

A filtragem colaborativa usando o KNN pode ser baseada em itens ou em usuários. Em ambos os tipos, busca-se estabelecer uma vizinhança formada pelos K usuários ou itens mais parecidos com o usuário ou item alvo.

No KNN baseado em usuários, se $\boldsymbol{S(u)^k}$ for a vizinhança definida pelos k usuários mais similares ao usuário *u,* a nota que *u* atribuiria ao item *i,* $\hat{\boldsymbol{r}}_{\boldsymbol{u,i}}$ pode ser estimada pela equação (1):

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in S(u)^k} sim(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in S(u)^k} sim(u,v)} \qquad (1)$$

Nesta equação, *sim(u,v)* expressa o grau de semelhança entre o usuário *u* e um usuário *v* de sua vizinhança, e $\bar{\boldsymbol{r}}_{\boldsymbol{u}}$. é o valor médio atribuído por *u* aos itens por ele avaliados.

No KNN baseado em itens, se $\boldsymbol{S(i)^k}$ for a vizinhança definida pelos *k* itens mais similares ao item *i,* a nota que o usuário *u* atribuiria ao item *i,* $\hat{\boldsymbol{r}}_{\boldsymbol{u,i}}$, pode ser estimada pela equação (2):

$$\hat{r}_{u,i} = \frac{\sum_{j \in S(i)^k} sim(i,j) r_{u,j}}{\sum_{j \in S(i)^k} sim(i,j)} \qquad (2)$$

Em (2), *sim(i,j)* expressa o grau de semelhança entre os itens *i* e *j* e $\boldsymbol{r_{u,j}}$ a nota que o usuário *u* atribuiu ao item *j*..

O cálculo da semelhança entre dois usuários *u* e *v* para a aplicação da equação (1) pode ser feito utilizando a correlação de Pearson, conforme apresentada na equação (3), onde I é o conjunto de itens avaliados por *u* e por *v*.

$$sim(u,v) = \frac{\sum_{i \in I}(r_{u,i} - \overline{r_u})(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \qquad (3)$$

Para a aplicação da equação (2), a similaridade entre dois itens *i* e *j* pode também ser determinada pela correlação de Pearson, conforme a equação (4). Nesta equação, *U* é o conjunto de todos os usuários que avaliaram os itens *i* e *j,* $\bar{\boldsymbol{r}}_{\boldsymbol{i}}$ é o valor médio das avaliações recebidas por *i* e $\bar{\boldsymbol{r}}_{\boldsymbol{j}}$ é o valor médio das avaliações recebidas por *j.*

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}} \qquad (4)$$

2.2. Filtragem Baseada em Tendências

Cacheda e Carneiro, em [2], propõem uma nova maneira de prever uma nota para um elemento da matriz usuário-item, não utilizando nem a vizinhança baseada em itens nem a baseada em usuários. O novo método baseia-se na interpretação das tendências ou diferenças entre usuários e itens.

A FC baseada em vizinhanças (item ou usuários) considera que se dois usuários mostram um padrão de avaliações similares provavelmente as previsões para itens não avaliados irão também coincidir. Contudo, como discutido anteriormente, essas técnicas requerem uma grande quantidade de informação e capacidade de processamento, de modo que esses algoritmos, baseados no cálculo da similaridade entre usuários ou itens, enfrentam sérios problemas com matrizes esparsas. A alternativa proposta consiste em procurar as diferenças entre usuários ou itens, ao

invés das semelhanças. Os autores defendem a abordagem a partir da constatação de que usuários parecidos podem avaliar itens de maneiras diferentes: alguns são inclinados a dar notas boas deixando notas baixas somente para os itens realmente ruins, outros guardam as melhores notas somente para os melhores itens deixando a maior parte com avaliações ruins.

Desse modo, a equação (5) define a tendência de um usuário, τ_u, como a diferença média entre as suas avaliações e a média do item, sendo I_u o conjunto de itens avaliado pelo usuário *u*, $r_{u,i}$ a nota que o usuário u atribuiu ao item *i* e $\overline{r_i}$ é a média de notas que o item *i* recebeu.

$$\tau_u = \frac{\sum_{i \in I_u}(r_{u,i} - \overline{r_i})}{|I_u|} \quad (5)$$

De forma análoga, a tendência de um item τ_i é definida pela equação (6), onde U_i, é o conjunto de usuários que avaliaram o item *i* e $\overline{r_u}$.a média que o usuário *u* atribuiu aos itens por ele avaliados.

$$\tau_i = \frac{\sum_{u \in U_i}(r_{u,i} - \overline{r_u})}{|U_i|} \quad (6)$$

Para prever a avaliação de um item são consideradas as médias do usuário e do item bem como as suas tendências

Se tanto o usuário como o item tem tendência positiva é feita uma previsão que é superior à ambas médias, conforme mostrado na equação (7).

$$\hat{r}_{u,i} = \max(\overline{r_u} + \tau_i, \overline{r_i} + \tau_u) \quad (7)$$

Em caso contrário, quando ambos, usuário e item, têm tendência negativa, é usada a equação (8).

$$\hat{r}_{u,i} = \min(\overline{r_u} + \tau_i, \overline{r_i} + \tau_u) \quad (8)$$

Uma terceira situação ocorre quanto se tem um usuário com tendência negativa e um item com tendência positiva. Se ambas médias corroboram as tendências (usuário com média baixa e item com média alta), a previsão ficará entre os dois, conforme expresso pela equação (9), onde β, uma constante que varia entre 0 e 1, controla a contribuição da média do item e do usuário na previsão.

$$\hat{r}_{u,i} = \min\big(max\big(\overline{r_u}, (\overline{r_i} + \tau t_u)\beta + (\overline{r_u} + \tau_i)(1 - \beta)\big), \overline{r_i}\big) \quad (9)$$

Finalmente, há uma quarta situação na qual as médias não corroboram as tendências. Isto se verifica quando um usuário com uma tendência negativa avalia um item com média baixa (a previsão deve ser ruim) porém a média do usuário é alta e a tendência do item é positiva. Neste caso, a previsão é feita usando a equação (10).

$$\hat{r}_{u,i} = \overline{r_i}\beta + \overline{r_u}(1 - \beta) \quad (10)$$

Em [2], mostra-se que o algoritmo baseado em tendências é tão preciso quanto a maioria dos métodos modernos, e ao mesmo tempo computacionalmente mais eficiente. Tendo sido esse o motivo para também empregar essa abordagem no trabalho aqui descrito.

3. ALGORITMOS GENÉTICOS

Algoritmos genéticos (AG) são métodos para buscar soluções aproximadas para problemas de otimização e busca. Esse tipo de algoritmo é inspirando na biologia evolutiva, utilizando mecanismos como a hereditariedade, a mutação, a reprodução e a seleção natural. Esses mecanismos biológicos são simuladas computacionalmente para buscar soluções para problemas. Estas soluções são tratadas como indivíduos de uma população.

O Algoritmo 1 mostra o pseudocódigo de um algoritmo genético, tal como apresentado em [3]. No pseudocódigo, *t* representa o número da geração atual e P(*t*) é a população na geração *t*. T é uma constante que indica o número máximo de gerações que será produzido na busca por uma solução. Passado esse número de gerações, o algoritmo retorna a melhor solução encontrada até o momento, a qual que poderá ser utilizada ou não, dependendo de sua qualidade e dos requisitos do problema. Se antes disso uma solução (um indivíduo) adequada for encontrada, o algoritmo para e retorna essa solução.

Algoritmo 1: Algoritmo Genético
t = 0 Inicia P(*t*) Avalia P(*t*) Enquanto (*t* <T e ∉ em P(*t*) uma solução) faça   Seleciona indivíduos em *P(t)*   Cruzamento em P(*t*) criando P(*t+1*)   Mutação em P(*t+1*)   Avalia P(*t+1*)   *t=t+1* Fim

Os indivíduos da população são as possíveis soluções para o problema, sendo usualmente representados por cadeias de bits. No início do algoritmo, para *t=0,* são gerados aleatoriamente

indivíduos para compor a primeira geração.

A avaliação de uma população é feita usando uma função de adaptação (*fitness*), a função objetivo que se pretende otimizar. Ela serve para selecionar os indivíduos melhor adaptados (as melhores soluções até o momento) para a próxima geração. Os melhores indivíduos de uma população são escolhidos a partir do grau de adaptação (*fitness)* de cada um. A ideia é que quanto mais adaptado o individuo mais perto ele está da solução, por esse motivo procura-se manter para a próxima geração os indivíduos que correspondem à melhor solução para o problema até o momento (seleção elitista).

Assim como na biologia, nos algoritmos genéticos há a troca de material genético dos indivíduos de uma população para gerar um novo indivíduo, realizada por um operador de reprodução ou cruzamento (*crossover).* Há diversas implementações do operador de reprodução, sendo o comum o cruzamento em um ponto (*one-point crossover*). em que se escolhe um mesmo ponto nas cadeia de bits que representa cada um dos pais (que têm o mesmo tamanho). O material genético (os bits) além desse ponto em cada um dos pais é trocado, dando origem a dois novos indivíduos (filhos).

A mutação modifica aleatoriamente parte do cromossomo dos indivíduos visando aumentar a diversidade da população o que permite ir para regiões do espaço de busca não exploradas pelos indivíduos da geração atual.

4. FC HÍBRIDA COM ALGORITMO GENÉTICO

Com base na proposta de [4], foi desenvolvido um algoritmo híbrido que utiliza um algoritmo genético para ponderar as notas previstas por uma implementação do algoritmo KNN e do algoritmo de tendências, visando melhorar a precisão das previsões feitas. O objetivo é encontrar os valores ótimos para os pesos a atribuir a cada previsão, conforme mostrado na equação (11). Nesta equação, $\hat{r}_{j_{u,i}}$ é o valor previsto pelo algoritmo *j* para a nota que o usuário *u* daria ao item *i.* Na implementação ora descrita *j=1* corresponde ao

$$\hat{r}_{u,i} = \frac{\sum_j (\hat{r}_{j_{u,i}} . w_j)}{\sum_j w_j} \qquad (11)$$

algoritmo KNN e *j=2* ao algoritmo de tendências.

Os indivíduos foram representados como sequência de 16 bits, onde os primeiros oito bits correspondem ao peso a ser atribuído para a primeira previsão (*j=1*) e os últimos oito para a segunda previsão (*j=2*).

Na implementação da reprodução foi usado um operador de cruzamento em um ponto (*single-point crossover)*, com corte no oitavo bit, ocorrendo a troca entre os pesos das previsões de cada algoritmo para a produção dos novos indivíduos. A taxa de reprodução é de 75%, ou seja, são escolhidos 75% dos indivíduos da população para gerar descendentes. Os indivíduos sofrem mutação em um de seus genes com probabilidade de 5%. Como cada gene é representado por um bit, ocorre uma alteração em um de seus bits aleatoriamente (se o bit for igual a 0 torna-se 1 e vice-versa).

A função de *fitness* avaliará o erro médio absoluto

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \left| r_{u,i} - \hat{r}_{u,i} \right| \qquad (12)$$

(Mean Average Error -*MAE*) correspondente às previsões feitas utilizando na equação (11) com os pesos associados ao genes do indivíduo. O MAE é calculado pela equação (12).

Nesta equação, *n* é o número total de previsões feitas (quantos $\hat{r}_{u,i}$ foram previstos). $\hat{r}_{u,i}$ é o valor previsto para a nota que o usuário *u* atribuiria ao item *i,* e $r_{u,i}$ é valor real para essa nota. O cálculo do MAE pressupõe que se dispõe de uma matriz usuário-item com os valores de $r_{u,i}$.

5. AVALIAÇÃO EXPERIMENTAL

Para a avaliação dos algoritmos implementados foi construída uma matriz usuário-item com 100 usuários e 200 itens.

A precisão das previsões realizadas foi medida utilizando o MAE, em três etapas. Na primeira foi utilizado apenas o algoritmo KNN com vizinhança baseada em usuários, em seguida foi usado o algoritmo de tendências e, finalmente, o algoritmo híbrido que combina as previsões dos dois algoritmos precedentes usando um algoritmo genético, conforme descrito na seção 4. Foram feitos testes com diversos graus de esparsidade da matriz usuário-item, conforme mostrado na Tabela 1. O número de avaliações preenchidas na matriz variou de 32 a 512.

Quando da aplicação do AG para definir os pesos da composição, foi feita uma fase de treinamento na qual 80% dos usuários participaram, tendo-se como condições de parada do AG obter um par de pesos *w* que ao aplicar a equação (11) produzisse um *MAE menor que* 1,5m ou atingir 2000 gerações. Ao encontrar esses pesos, a composição foi testada com os 20% usuários restantes na base, seguindo os procedimentos adotados em [4]. A Tabela 1 apresenta os resultados obtidos utilizando a equação (11) com os pesos obtidos.

Tabela 1. MAE para os algoritmos avaliados

Avaliações	KNN	Tendências	*Composição*
32	2.2440	1.6667	1.4466
64	2.2661	1.2265	1.1886
128	2.2882	1.2265	1.1977
256	2.5816	1.3198	1.3088
512	2.3604	1.3198	1.3088

Conforme pode ser observado na Tabela 1, o algoritmo baseado em tendências apresenta um MAE inferior ao KNN em todos os casos testados. De forma semelhante, as previsões feitas a partir da composição dos valores do KNN e do algoritmo de tendências, usando os pesos determinados pelo Algoritmo Genético, apresentam MAE inferior ao dos dois algoritmos isoladamente em todos os casos testados. Conforme esperado, o MAE tende a diminuir à medida que a matriz usuário-item torna-se mais densa.

6. CONCLUSÃO

O objetivo do trabalho descrito neste artigo era: combinar duas abordagens conhecidas de filtragem colaborativas, visando melhorar a precisão das recomendações realizadas. Os resultados obtidos experimentalmente, utilizando uma matriz usuário-item artificialmente construída, mostram que isso é possível, ainda que a um custo computacional elevado. Pretende-se, dar continuidade ao trabalho, realizando nova e mais elaborada avaliação experimental utilizando dados reais, entre eles a base Lenskit[1] de 100 mil avaliações. Pretende-se também adicionar outros algoritmos à composição, tentando diminuir ainda mais o erro.

AGRADECIMENTOS

Ao CNPq pela bolsa de iniciação tecnológica, à PUC-Campinas pelas instalações que permitiram o desenvolvimento desse artigo, e ao Prof. Dr. Juan Manuel Adán Coello pela atenção, motivação e orientação.

REFERÊNCIAS

[1] Celma, Ò. (2010), The Recommendation Problem, in *Music Recommendation and Discovery*, Berlin Heidelberg: Springer-Verlag, pp. 15-41.

[2] Cacheda, F., et al. (2011), Comparison of Collaborative Filtering Algorithm: Limitations of Current Techniques and Proposals for Scalable, High-Performace Recommender Systems, *ACM Transactions on the Web (TWEB)*, Volume 5 Issue 1, Article No. 2.

[3] Nanas, N., e De Roeck, A., (2010), A review of evolutionary and imune-inspired information filtering, *Natural Computing*, vol 9. n. 3, pp. 545-573.

[4] Bobadilla, J. , et al. (2011), Improving collaborative filtering recommender system results and performance using genetic algorithms, *Knowledge-Based System*s, vol. 24, pp. 1310-1316.

[1] http://lenskit.grouplens.org/