

Algoritmos Especializados para Sistemas de Recomendação de Músicas

André C. Silva¹, Marcel L. Oliveira¹, Aloísio de M. Vilas-Boas¹,
Hendrik T. Macedo¹

Departamento de Computação – Universidade Federal de Sergipe (UFS) São
Cristóvão – SE – Brazil

{andrecs,marcello,aloisiomvb}@dcomp.ufs.br, hendrik@ufs.br

Abstract. Recommendation Systems (RS) use information filtering to solve the information overload issue. Mainstream RS approaches make use of collaborative algorithms due to its low cost, good accuracy and for being generic enough. We claim though that more specialized algorithms can improve recommendation quality. This paper presents a music recommendation system with an innovative specialized algorithm, considering feature extraction and user models. Experiments were conducted considering a database of 20 users and 335 ratings. The quality of recommendation was measured by means of two different precision metrics and two different diversification metrics. Results show that specialized algorithms contribute to the improvement of both accuracy and diversification.

1 Introdução

Filtragem de Informação separa a informação relevante do usuário da informação irrelevante, agindo como mediadores entre a informação e o usuário [1] de forma a resolver o problema da sobrecarga de informação. Sistemas de Recomendação são sistemas de filtragem de informação que procuram gerar recomendações aos usuários [2][3]. A abordagem clássica para geração de recomendações utiliza KNN (*K-Nearest Neighbors*), embora outras abordagens existam principalmente relacionadas ao campo de aprendizado de máquina (*Machine Learning*) [4] [1]. Como apontado em [3] a maioria destas técnicas utilizam poucas informações sobre o item e o usuário, excluindo informações contextuais (como data e localidade) e informações mais profundas do item (e.g. no caso de músicas: ritmo e melodia), que poderiam ser abordadas por algum algoritmo de forma especializada, para uma filtragem mais útil.

Neste trabalho são abordados algoritmos especializados para a recomendação de músicas. A seção 2 apresenta uma formalização de sistemas de recomendação necessária para entendimento dos algoritmos especializados propostos. Vale ressaltar, a importância da modelagem de usuário definida e utilizada no processo de recomendação. A seção 3 discute as métricas e meios para avaliar a qualidade

da recomendação. A seção 4 apresenta a composição do sistema de recomendação de músicas utilizado neste trabalho, utilizando métricas de similaridade derivada do modelo de usuário apresentado e processos de extração de características de arquivos musicais. A seção 5 define uma experimentação e discute seus resultados e a seção 6 apresenta as conclusões do trabalho.

2 Sistemas de Recomendação

Matematicamente um sistema de filtragem de informação tenta modelar uma função de utilidade para itens levando em consideração o usuário. Considerando $p : I \times U \rightarrow \mathbb{R}$ a relação que descreve a preferência do usuário $u \in U$ com relação a um item $i \in I$. Costuma-se utilizar um subconjunto de \mathbb{R} (e.g. valores $[0,1]$) ou uma discretização em níveis para o resultado da previsão. A relação $p(i, u)$ é definida apenas para um pequeno subconjunto $I \times U$ que são os itens que o usuário avaliou explicitamente ou que o sistema inferiu de alguma forma. A ideia é estimar ou definir $\hat{p}(i, u)$ para o resto do espaço $I \times U$ de acordo com alguma heurística ou critério. Em geral o resultado de uma filtragem é uma recomendação, o item que obtém a maior previsão $\operatorname{argmax}_{i \in I} p(i, u)$ ou simplesmente uma lista ordenada de acordo com as preferências estimadas.

Dois abordagens diferentes costumam ser amplamente citadas na literatura: filtragem baseada em conteúdo (*content-based filtering*) e filtragem colaborativa (*collaborative filtering*). Filtragem baseada em conteúdo analisa o conteúdo do item para verificar sua utilidade ao usuário, enquanto a filtragem colaborativa verifica a opinião de outros usuários similares ao usuário alvo da filtragem sobre um determinado item. A filtragem baseada em conteúdo analisa o conteúdo de itens para identificar a utilidade desses itens para um usuário [5][3][6]. Em geral a função de preferência para um item i_k é estimada $\hat{p}(i_k, u_l)$ de acordo com os itens que tem a função de preferência $p(i, u_l)$ definida. O princípio básico é que se um usuário u_l considera um item i_a relevante, então um item i_b que é similar a i_a também será relevante.

Algoritmos baseados em vizinhos próximos utilizam um determinado número de vizinhos para estimar $\hat{p}(i_a, u_l)$. Um exemplo de função de estimação seria simplesmente uma média ponderada:

$$\hat{p}(i_a, u_l) = \frac{\sum_{i \in I_{u_l k}} p(i, u_l) s(i_a, i)}{\sum_{i \in I_{u_l k}} s(i_a, i)} \quad (1)$$

Sendo $I_{u_l k}$ o conjunto dos k itens mais próximos de i_a para o qual $p(i, u_l)$ seja definido. Caso sejam considerados níveis não reais, uma simples maioria pode servir. A função de similaridade pode ser definida ou pode ser utilizada uma distância euclidiana caso esteja-se trabalhando com vetores em espaços euclidianos. A função de similaridade define o espaço de vizinhança dos itens, logo tem caráter crucial na previsão.

O sucesso de uma filtragem baseada em conteúdo depende da representação do item. O algoritmo não tem como analisar uma característica de um item que

não está representada na descrição do item[6][3]. A filtragem depende exclusivamente das avaliações do usuário, o que faz surgir problemas como superespecialização e baixa exploração [5][3].

A representação pode ir desde texto até tabelas ou tuplas que são formas mais estruturadas. Itens costumam ser representados como vetores em um espaço multi-dimensional de atributos $i = \{a_1, a_2, \dots, a_n\}$. Um problema associado é que as abordagens que utilizam classificadores necessitam que o item seja bem representado em um espaço euclidiano, o que muitas vezes não é o caso [6]. Nesses casos utilizam-se abordagens de classificação baseadas em similaridade[7], que necessitam que seja definida uma função de similaridade $s : I \times I \rightarrow \mathbb{R}$ bem definida. A abordagem clássica baseada em vizinhança (i.e. KNN) pode ser considerada uma classificação baseada em similaridade. É importante frisar que a função de similaridade, tanto quanto a representação do item, terá um peso decisivo no processo de filtragem baseada em conteúdo[7][5].

Na filtragem colaborativa as experiências de outros usuários são levadas em conta para previsão $\hat{p}(i, u_l)$ [2][3][5]. Basicamente o sistema parte do princípio de que se um item i_a é relevante para u_l que é parecido com u_k então i_a é provavelmente relevante a u_k .

A abordagem mais comum e utilizada é a baseada em vizinhança, utilizando o KNN [2][3]. Em comparação ao método baseado em conteúdo, que utiliza um cálculo da similaridade entre itens, é utilizada uma similaridade entre usuários $s : U \times U \rightarrow \mathbb{R}$. Uma das relações para previsão mais utilizadas [2] é:

$$\hat{p}(i_a, u_l) = \overline{p_{u_l}} + \frac{\sum_{u \in U_{u_l}} p(i_a, u) s(u_l, u)}{\sum_{u \in U_{u_l}} s(u_l, u)} \quad (2)$$

Sendo $\overline{p_{u_l}}$ a média das avaliações do usuário u_l e U_{u_l} os k usuários mais próximos ao usuário u_l . Em outras palavras a previsão de um item i_a é feita de acordo a avaliação do mesmo item por outros usuários suficientemente próximos no espaço de usuário U . Esta equação leva em conta que usuários diferentes tem uma interpretação diferente da escala de notas na avaliação [3], por isso é levado em conta o desvio dos usuários da média, para que a estimação mantenha a mesma escala absoluta do usuário. A similaridade entre usuário é geralmente calculada pela correlação de pearson [2], mostrada na equação 3.

$$s(u_a, u_b) = \frac{\sum_{i=1}^m [(p(i, u_a) - \overline{p(i, u_a)}) (p(i, u_b) - \overline{p(i, u_b)})]}{\sqrt{\sum_{i=1}^m (p(i, u_a) - \overline{p(i, u_a)})^2 \sum_{i=1}^m (p(i, u_b) - \overline{p(i, u_b)})^2}} \quad (3)$$

O sucesso da filtragem colaborativa depende diretamente do sucesso da comunidade de usuários do sistema. Usuários com preferências muito incomum terão poucos usuários similares, diminuindo a confiabilidade e precisão da recomendação [3]. Uma vantagem com relação a filtragem baseada em conteúdo é que a representação do item é irrelevante [3].

2.1 Representação do Usuário

De maneira resumida o usuário é representado no sistema por qualquer informação que ajude no processo de definição de $\hat{p}(u_k, i_l)$, sendo u_k o usuário. Abordagens clássicas levam em conta apenas o histórico de transações do usuário com o sistema, como itens avaliados e itens previstos.

Neste trabalho, além do histórico de interação do usuário com o sistema, é mantido um modelo de usuário baseado em grafos[8]. Este modelo captura a relação entre categorias cuja similaridade não é bem representada (e.g. gêneros musicais). Imaginando-se dois gêneros subjetivamente similares g_a e g_b o melhor que pode ser inferido sem informações adicionais é uma similaridade binária (i.e. $s(g_a, g_b) = 1$ apenas se $g_a = g_b$), o que pode prejudicar a definição da vizinhança.

Seja então a definição de um grafo $G(V, E, \mu, \nu)$, sendo que $v \in V$ representa cada categoria relacionada, $e = (v, w) \in E : V \times V$ representa as arestas (i.e. relações), $\mu : V \rightarrow L_V$ uma função que rotula vértices e $\nu : E \rightarrow L_E$ uma função que rotula arestas. Para o modelo apresentado é considerado que $L_E = [0..1] \subset \mathbb{R}$ e $L_V = \mathbb{N}$. O valor $k = \nu(v, w)$ indica a similaridade/proximidade entre os vértices v e w ; se $k = 1$ a relação pode ser considerada máxima, nesse caso os itens têm uma forte relação de proximidade/similaridade, se $k = 0$ os itens tem uma baixa similaridade. Este caso é diferente de $\nu(v, w)$ não estar definida (i.e. a relação não existe), pois neste caso a informação pode não ter sido modelada. No estudo de caso cada gênero é representado como um inteiro, por simplificação, e cada aresta indicaria a relações entre os gêneros indicando proximidade/similaridade.

Estas relações são construídas a partir das avaliações do usuário em questão pela equação 4. Onde $p(v)$ e $p(w)$ indicam a preferência do usuário quanto uma categoria de gênero ou artista. Em sistemas de recomendação a preferência por uma categoria pode ser expressa pela média das notas de um conjunto de itens pertencentes a uma mesma categoria. Imaginando um conjunto de avaliações de músicas oriundas de uma interação com o sistema a média das músicas com o mesmo gênero v resultaria em $p(v)$. Uma vez que a preferência do usuário quanto a categorias tende a mudar com o tempo as relações são atualizadas de acordo com a equação 5, onde w_n e w_e regulam o pesos da nova relação com o da existente no grafo.

$$\nu_n(v, w) = 1 - |p(v) - p(w)| \quad \nu(v, w) = \frac{\nu_n(v, w)w_n + \nu(v, w)w_e}{w_n + w_e} \quad (4,5)$$

$$\nu_i(a_1, a_n) = \frac{\sum_{i=1}^{n-1} \nu(a_i, a_{i+1})}{n} \quad (6)$$

Uma vez que são definidas relações entre itens que expressam uma proximidade/similaridade podemos então inferir relações entre itens não diretamente relacionados, ou seja, $x \notin adj(v)$. Seja $v \in adj(w)$, $w \in adj(x)$, $x \notin adj(v)$ podemos inferir uma relação $\nu_i(v, x)$ através da transitividade: $\nu_i(v, x) = (\nu(v, w) + \nu(w, x))/2$. Generalizando para uma transitividade entre n vértices, temos um caminho $c = \{a_1, a_2, \dots, a_n\}$, $a_1 = v$, $a_n = x$, temos a equação 6.

3 Avaliando a Qualidade da Recomendação

Sistemas de recomendação diferentes possuem objetivos e tarefas diferentes. Como exemplo alguns sistemas de recomendação preocupam-se apenas em entregar recomendações que estejam acima de um certo limiar de relevância enquanto outros buscam entregar uma lista ordenada de itens para consumo. Como diferentes métricas analisam diferentes aspectos do sistema de recomendação é necessário então identificar as métricas condizentes com as tarefas do sistema [9].

Em [10] é apontado que apesar das atuais métricas consideradas capturarem a precisão de um sistema, apenas a precisão não garante a utilidade do sistema. É preciso considerar uma visão mais voltada ao usuário para avaliar a utilidade de uma filtragem. Uma das abordagens desse problema é mostrado em [11], onde o autor propõe uma métrica chamada *Intra-List Similarity Metric*.

3.1 Métricas de Precisão

Métricas de precisão analisam o desempenho do sistema baseado nas avaliações do usuário. Uma parte das reais avaliações são utilizadas para montar o modelo de filtragem e outra parte é utilizada para calcular o erro efetivamente. É necessário adotar uma estratégia quanto a divisão do conjunto para treinamento e teste. Utilizar todas as avaliações para montar o modelo do usuário é enviesado de maneira otimista, enquanto que utilizar apenas metade para treinamento é considerado pessimista[12]. Algumas estratégias geralmente utilizadas em aprendizado de máquina, como bootstrap [12], podem ser utilizadas para diminuir o viés ou a variância do erro. Uma das métricas mais utilizadas é o Erro Médio Absoluto (*Mean Absolute Error*)[2] que é definida segundo:

$$|\bar{E}| = \frac{\sum_{i=1}^N |\hat{p}(i, u) - p(i, u)|}{N} \quad (7)$$

Onde $\hat{p}(i, u)$ são previsões feitas para a filtragem e $p(i, u)$ a real avaliação do usuário u . Esta métrica indica o desvio médio das notas do usuário em relação as notas previstas pelo sistema e costuma ser aplicado para cada usuário do sistema, obtendo-se a média do erro do sistema. É um interessante indicador da precisão das previsões de um sistema em geral.

Outra tipo de métrica utilizada são as correlações entre as avaliações dos usuários e as previsões do sistema. A correlação de Kendall Tau[2] verifica a correlação de ordenação entre duas listas:

$$c = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \quad (8)$$

Onde n_c e n_d são os pares concordantes e discordantes respectivamente, de acordo com a ordenação das listas. Esta métrica é útil para verificar se o sistema está conseguindo ordenar as recomendações de acordo com a preferência do usuário.

3.2 Métricas de Diversificação

Estas métricas medem a diversificação de uma lista de recomendação gerada pelo sistema. Uma lista diversificada pode ser mais útil para o usuário em alguns domínios, pois a recomendação consegue recomendar itens nunca antes vistos pelo usuário. Estes fenômenos são chamados de *serendipity* e *novelty* [3] [10]. Imagine no domínio de supermercado, onde existe um usuário que sempre compra bananas, recomendar a compra de bananas a este usuário é inútil apesar de preciso, pois o mesmo sempre compra bananas.

Intra-List Similarity mede a similaridade dentre os itens em uma lista [11], e é expressa pela equação 9. Onde R é um conjunto contendo os itens da lista, e $s(i_k, i_l)$ é uma função de similaridade que segue algum critério específico, como uma categoria por exemplo (gêneros musicais).

$$ILS(R) = \frac{\sum_{i_k \in R} \sum_{i_l \in R, i_k \neq i_l} s(i_k, i_l)}{2} \quad (9)$$

Para evitar a criação de uma função de similaridade que pudesse favorecer uma medida de similaridade no algoritmo, podemos simplesmente analisar a concentração de uma determinada categoria através de um histograma. Para isso calcula-se a participação porcentual p_w de cada categoria distinta w na lista R . Aplicamos então a equação 10. Quanto mais próximo de zero for c , menos concentrada a lista estará em uma categoria. Este cálculo é útil para domínios que contem categorias bem separadas.

$$c = \sum_w p_w^2 \quad (10)$$

4 Composição do Sistema de Recomendação de Músicas

O sistema desenvolvido para este trabalho envolve músicas como item da recomendação. As avaliações do sistema são explícitas e as notas estão no intervalo [0..10]. O sistema gera recomendações utilizando o KNN para predições da nota, tanto de forma colaborativa quanto baseada em conteúdo, como mostrado nas equações 2 e 1. A similaridade entre usuário é dada pela Correlação de Pearson, descrita na equação 3. A similaridade entre itens é uma média ponderada das similaridades de cada uma das características consideradas, que serão apresentadas adiante.

O sistema utiliza uma representação semi-estruturada de músicas. Esta estrutura contem metadados e dados extraídos da composição da mesma (contidas no formato MIDI). Os metadados contem informações sobre a música, e são: título - título representativo da música; gênero - gênero musical ao qual a música pertence; artista/compositor - compositor ou representativo da música; ano - ano de publicação da mesma.

Uma das características extraídas é o conjunto de instrumentos. Uma hipótese é que esta característica (instrumentos) seja representativa, uma vez que algumas pessoas são mais inclinadas a gostar de uma música se ela é tocada com

instrumentos que elas apreciem. Para a característica consideramos a quantidade de notas por instrumento de uma música, de forma relativa. O algoritmo para extração percorre as mensagens da MIDI na ordem temporal. Se a mensagem for uma nota, a contagem de notas do instrumento ativo do canal ao qual foi enviada a mensagem é incrementada, assim como o total de notas lidas. Se a mensagem for uma mudança de instrumento, a referência ao instrumento ativo daquele canal é alterada. Por fim, todas as contagens de notas são divididas pelo total, de modo a obter o valor relativo de cada instrumento. O resultado é um conjunto de duplas $I = \{i_1 = (p_1, r_1), i_2 = (p_2, r_2), \dots, i_n = (p_n, r_n)\}$ contendo o instrumento e sua devida participação na música com relação a quantidade de notas.

O método de similaridade consiste na paridade dos instrumentos de duas músicas distintas, com a posterior seleção desses pares. Esta paridade tenta relacionar os instrumentos das músicas distintas segundo sua participação. Os pares são definidos pela relação $r : I_A \times I_B \rightarrow \mathbb{R}$ entre o conjunto de instrumentos das músicas A e B . A similaridade então é calculada incrementalmente seguindo o conjunto de equações 6. Onde $\max(R_i)$ é a relação cujo $r.v$ é máximo no conjunto R_i . A cada iteração i a relação de maior valor é selecionada, então o novo conjunto R_{i+1} é criado. O fator f corresponde ao fator de grupo, que tem um valor variante de 0 a 1, onde instrumentos iguais recebem 1, diferentes 0, e do mesmo grupo um valor intermediário. Este grupos são previamente definidos pelo padrão MIDI mas podem ser redefinidos.

$$\forall r(i_A, i_B) \in R, \quad r.v = \frac{\min(i_A, i_B)}{\max(i_A, i_B)} * f \quad (4)$$

$$S(A, B) = \frac{S_1 + S_2 + \dots + S_n}{n} \quad S_i = \max(R_i) \quad R_{i+1} = R_i - E_i \quad (5)$$

$$E_i = \{r(x, y) \in R_i | \exists z, w, \max(R_i) = r(z, w), x = z \vee y = w\} \quad (6)$$

As medidas de similaridade utilizadas no sistema para cada características são explicitadas abaixo:

1. GêneroA/ArtistaA ((s_{ge}^a, s_{ar}^a)) - Utiliza o modelo de grafos indicado neste trabalho. A similaridades são oriundas da relação $\nu(a_i, a_j)$ e $\nu_i(a_i, a_j)$ definidas e construídas nas equações 4, 5 e 6. Para estes experimentos uma penalidade de 0.1 foi aplicada para relações induzidas com valor máximo, para valorizar as relações diretas.
2. GêneroB/ArtistaB ((s_{ge}^b, s_{ar}^b)) - Uma relação binária onde se $a_i = a_j$ então $s(a_i, a_j) = 1$, caso contrário $s(a_i, a_j) = 0$. É uma medida simples.
3. Ano (s_{an}) - é utilizada uma distância entre décadas, resumida na equação 4. Onde $dec(k)$ é a década do ano k .
4. Instrumentos s_{ins} - utiliza o conjunto de equações 6, no processo descrito anteriormente.

$$s(a_i, a_j) = 1 - \frac{|dec(a_i) - dec(a_j)|}{100} \quad (7)$$

5 Experimentos e Resultados

Seguindo a definição em [9] o sistema está apropriado para a tarefa *Find Good Items*, ou seja, o importante é achar uma devida quantidade de itens relevantes ao usuário. Logo foram aplicadas as métricas de precisão MAE, Kendall Tau, segundo as recomendações[2], e as métricas de diversificação Topic Diversification e análises de concentração em histograma. As métricas de precisão foram aplicadas a divisões aleatórias do conjunto de avaliações do usuário. Duas divisões foram utilizadas nesta análise: 80% de treinamento e 20% para teste e 50% para treinamento e teste. O erro foi calculado 100 vezes, o suficiente para a maioria das comparações com significância estatística. Na tabela 1 para cada conjunto de característica considerado é mostrado a média e o desvio padrão médio das 100 amostras, logo o desvio padrão apresentado representa o desvio entre usuários e não da média amostral. As métricas foram aplicadas a conjuntos diferentes de características de forma a analisar sua contribuição na qualidade.

Para a análise da diversificação as métricas foram aplicadas a uma lista de 10 recomendações selecionando-se os itens com a maior previsão para cada usuário. A tabela 1 apresenta a média para todos os usuários. A métrica de *Intra-List Similarity* foi aplicada com o mesmo método de similaridade utilizado para gerar a recomendação, exceto no colaborativo, que foi utilizada a similaridade binária para gênero e artistas. Os resultados estão resumidos na tabela 1, onde a numeração indica as características acima nas medidas de similaridade e (*Col*) é o método colaborativo aplicando-se equações 2 com a correlação de pearson (equação 3) para similaridade de usuários.

A base utilizada para a experimentação contem um total de 17 usuários, 202 itens e um total de 337 avaliações gerando uma média de 19 avaliações por usuário. Todos os usuários avaliados tem pelo menos 12 avaliações. Por avaliação geral 162 itens foram avaliados por pelo menos um usuário e 40 não foram avaliados por nenhum usuário. Como os itens a serem avaliados eram escolhidos de forma aleatória a distribuição das avaliações por itens é aproximadamente igual. A distribuição de gêneros na base é também aproximadamente equivalente. A base de dados pode ser considerada difícil de trabalhar, visto que o *Movielens* pede um mínimo de 20 avaliações para gerar recomendações[3].

O experimentos consideram $K = 3$, ou seja o uso de três vizinhos para a previsão, exceto para o método colaborativo que considera $K = 5$ pelo fato deste apresentar melhora com o aumento do número de vizinhos [2]. As similaridades que utilizam o modelo consideram 6 avaliações por interação do usuário com o sistema para montar as relações.

Os conjuntos que continham a característica instrumentos obtiveram uma boa diversificação sem sacrifício de precisão, principalmente quando combinadas. Esses resultados apontam a significativa melhora que características mais representativas, mesmo simples, de um item podem melhorar bastante o algoritmo de recomendação, concordando com a observação com relação ao uso de informações

Tabela 1. Métricas aplicadas X Conjunto de Características

Carac.	MAE (0.8/0.2)	Kendall (0.8/0.2)	MAE (0.5/0.5)	Kendall (0.5/0.5)	Hist. (Gênero)	Hist. (Ar- tista)	ILS
(1)	1,91±1,26	0,39±0,36	2,10±1,07	0,42±0,16	0,78±0,24	0,67±0,30	3,32±1,01
(2)	4,48±2,26	0,36±0,36	4,88±1,81	0,41±0,20	0,80±0,23	0,73±0,28	3,33±1,09
(1,3)	2,12±1,34	0,40±0,37	2,28±1,15	0,48±0,20	0,80±0,23	0,73±0,28	3,33±1,09
(2,3)	2,30±1,48	0,40±0,37	2,38±1,18	0,50±0,20	0,51±0,24	0,46±0,23	2,75±0,73
(1,4)	1,89±1,15	0,41±0,36	2,10±1,05	0,45±0,15	0,48±0,25	0,43±0,25	1,33±0,86
(2,4)	2,09±1,30	0,38±0,36	2,30±1,14	0,46±0,19	0,68±0,28	0,58±0,33	1,77±0,97
(1,3,4)	1,95±1,23	0,41±0,36	2,23±1,11	0,48±0,15	0,48±0,25	0,43±0,25	1,33±0,86
(2,3,4)	2,19±1,34	0,46±0,36	2,28±1,15	0,48±0,14	0,30±0,16	0,24±0,11	1,95±0,30
(4)	2,25±1,34	0,40±0,36	2,37±1,16	0,46±0,18	0,19±0,05	0,16±0,04	1,72±0,52
(3)	2,33±1,46	0,44±0,37	2,34±1,18	0,48±0,19	0,36±0,12	0,31±0,08	4,48±0,02
(Col)	0,58±0,48	0,14±0,23	0,86±0,75	0,13±0,11	0,16±0,04	0,14±0,03	0,68±0,15

adicionais sobre o item [3]. A característica ano se mostrou um pouco aleatória, já que com a mudança de tamanho de conjunto não houve muitas mudanças.

Como era esperado a abordagem colaborativa teve um ótimo desempenho, com o melhor aprendizado, ou seja, com a maior diferença na precisão entre as duas divisões de conjunto (80%/20% e 50%/50%). Isso se deve ao fato de que com mais avaliações cada usuário ganha uma medida mais precisa da similaridade entre usuários e consegue achar mais vizinhos.

A natureza da similaridade binária (i.e. conjuntos que utilizavam a similaridade 2) mostra que esta necessita de um número de avaliações variada para conseguir boa precisão. Isso se deve ao fato de que esta similaridade só conseguirá achar vizinhos para um item com o mesmo gênero/artista, limitando bastante o espaço de busca mas alcançando boa precisão na previsão destes casos. Logo quando combinada com outras características que conseguem mais vizinhos (como a 4), a precisão tende a melhorar. A modelagem utilizada (similaridade 1) trabalha em uma medida de similaridade entre gêneros e artistas mais distribuída, utilizando a preferência do usuário, conseguindo uma melhor precisão no cálculo da vizinhança e portanto uma melhor precisão geral. O alto desvio padrão do erro (com relação aos usuários) da métrica MAE indica que o erro varia bastante entre extremos, com alguns usuários conseguindo erros baixos e outros com erros altos. Esta variação é causada pela baixa uniformidade das avaliações dos usuários, com usuários com menos avaliações alcançando erros maiores.

6 Conclusões

Este trabalho apresentou o desenvolvimento de um sistema de recomendação de músicas com extração de características musicais. Foram utilizados modelos de usuário baseados em grafos para a similaridade do domínio de gêneros musicais e artistas/compositores, criando relações de preferência. Foi apresentando uma característica extraída da música (instrumentos utilizados) e um método

de similaridade da mesma para ser usado no processo de recomendação baseado em conteúdo.

Os resultados apontam que o uso de algoritmos mais especializados, neste caso uma modelagem para a similaridade entre características e a extração dos instrumentos que compõem a música, ajuda bastante tanto na precisão da filtragem quanto na diversificação. Tanto a modelagem quanto a característica extraída mostraram-se alternativas viáveis para uso, a primeira por alcançar boa precisão com pouca informação e a segunda por conseguir bons resultados gerais quando combinada.

Referências

1. Sheth, B.D.: A learning approach to personalized information filtering. Master's thesis, Massachusetts Institute of Technology (January 1994)
2. Herlocker, J.L.: Understanding and improving automated collaborative filtering systems. PhD thesis, University of Minnesota (2000) Adviser-Joseph A. Konstan.
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.* **17**(6) (2005) 734–749
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998) 43–52
5. Perny, P., Zucker, J.D.: Preference-based search and machine learning for collaborative filtering: the "film-conseil" movie recommendation system (2001)
6. Pazzani, M.J., Billsus, D. In: *The Adaptive Web*. Springer Berlin / Heidelberg (2007) 325–341
7. Chen, Y., Garcia, E.K., Gupta, M.R., Cazzanti, L., Rahimi, A.: Similarity-based classification: Concepts and algorithms. Technical Report UWEEETR-2008-0005, Department of Electrical Engineering, University of Washington, Seattle, WA 98195 (December 2008)
8. Silva, A.C., Oliveira, M.L., de M. Vilas-Bôas, A., Macedo, H.T.: Modelagem de usuário baseada em grafos. In: *VII Encontro Nacional de Inteligência Artificial*. (2009)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1) (2004) 5–53
10. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, New York, NY, USA, ACM (2006) 1097–1101
11. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: *WWW '05: Proceedings of the 14th international conference on World Wide Web*, New York, NY, USA, ACM (2005) 22–32
12. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000)