Modelagem de Usuário Baseada em Grafos

André C. Silva[1], **Marcel L. Oliveira**[1], **Aloísio de M. Vilas-Bôas**[1], **Hendrik T. Macedo**[1]

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

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

Abstract. *In order to provide personalization to computer systems, the user's most representative aspects must be identified. This work proposes an original user's model, which catches relations between items according to user's preferences. Such model is empirically validated considering the comparison with manually-designed models provided by volunteers. Its applicability is evaluated by precision metrics within a developed music recommendation system. The results have shown that the learned user's model is close to the real one and its practical use in the information filtering domain is promising.*

Resumo. *Para a personalização de sistemas computacionais, é necessária a identificação dos aspectos mais representativos do usuário. Este trabalho apresenta uma proposta de modelagem de usuário original, que captura relações entre itens de acordo com a preferência do usuário. O modelo é validado empiricamente através da comparação com modelos supridos por usuários através de entrevistas. Sua praticidade é avaliada com seu uso em um sistema de recomendação, utilizando métricas de precisão. Os resultados mostram que o modelo capturado é próximo ao real e que o uso prático no domínio da filtragem é promissor.*

1. Introdução

Em muitos sistemas computacionais, a comunidade de usuários é heterogênea o bastante de forma que o uso de apenas um modelo de usuário representado toda a comunidade é insuficiente com relação a interação com o sistema [Rich 1983]. Modelagem do usuário torna o sistema mais útil, usável e torna a experiência de interação com o sistema melhor de acordo com o conhecimento do usuário [Rich 1983] [Fischer 2001]. Em geral um modelo de usuário procura discernir as seguintes informações [Webb et al. 2001]: o processo cognitivo base para as ações do usuário; as diferenças entre as habilidades de um especialista e do usuário; o padrão de comportamento do usuário ou suas preferências; as características do usuário. Algumas características desejadas para modelagem de usuários são apontadas em [Kobsa 2001]: capacidade de generalização, independência de domínio, capacidade de inferência expressiva e forte e suporte a rápida adaptação.

Modelagem de usuário é um campo com uma forte relação com classificação [Allen 1990] e o campo de aprendizado de máquina(*Machine Learning*) é bastante utilizado em modelagem de usuários[Webb et al. 2001]. Alguns problemas apontados para modelagem de usuário são: a necessidade de uma grande quantidade de dados - em alguns modelos uma precisão aceitável exige uma grande quantidade de dados; a necessidade de dados rotulados - a rotulação muitas vezes não é explícita a partir da observação do comportamento do usuário; complexidade computacional - um algoritmo que produz 78% de

precisão e complexidade computacional menor é preferível, em alguns domínios, do que outro algoritmo que produz 80% de precisão e complexidade muito maior; *concept drift* - modelagem de usuário é uma tarefa dinâmica pois o que caracteriza um usuário pode mudar com o tempo. É interessante então que os algoritmos se ajustem a essas mudanças, este é um problema conhecido como *concept drift* no domínio de aprendizado de máquina. Estes problemas são discutidos com mais detalhes em [Webb et al. 2001].

Exemplos de sistemas que utilizam um perfil do usuário personalizado são sistemas de recomendação (filtragem de informação) [Herlocker 2000]. Duas abordagens são bastante utilizadas na literatura: baseada em conteúdo (*Item Based*) e colaborativa (*User Based*) [Adomavicius and Tuzhilin 2005]. A abordagem colaborativa é bastante utilizada na literatura pelo fato de ser genérica a qualquer tipo de item e por ser pouco custosa comparada a baseada em conteúdo. Esta última requer em muitos casos um maior processamento para construção de modelos, uma representação do item mais detalhada para melhor capturar a natureza do item e possivelmente algoritmos especializados [Adomavicius and Tuzhilin 2005]. O algoritmo mais comum utilizado para ambas abordagens é o KNN (*K-Nearest Neighbors*)[Adomavicius and Tuzhilin 2005]. O KNN utiliza a classificação dos vizinhos (sejam usuários ou itens) mais "próximos"para prever a classificação do item em questão. Essa proximidade, considerando um espaço amostral ω, pode ser aferida de uma medida de similaridade $s : \omega \times \omega \rightarrow \mathbb{R}$ [Santini and Jain 1999] que não segue os rígidos axiomas de distância. Usar similaridade é útil quando a representação do item não se encaixa bem em um modelo matemático de espaços lineares, como espaços euclidianos (e.g. informações textuais como gênero para uma música).

Informações como compositores e gêneros musicais carregam uma semelhança subjetiva. Infelizmente esta semelhança é difícil de ser percebida pela máquina e muitas vezes há discrepâncias de pessoas diferentes quanto a estas semelhanças. Este tipo de informação pode ser utilizada como medida de similaridade entre itens, e pode ser aferida do comportamento do usuário.

Este trabalho tenta atacar a problemática utilizando uma modelagem das preferências dos usuários nas relações entre os itens existentes. Na seção 2 é apontado como funciona o modelo e como se dá sua representação. Na seção 3 é abordada sua construção e como é possível inferir relações não diretamente aferidas. Hipóteses levantadas sobre características do modelo são validadas através de experimentação empírica descrita na seção 4. Finalmente algumas conclusões são apresentadas na seção 5.

2. Modelagem de Usuário Baseada em Grafos

A proposta desse modelo é capturar as relações existentes entre categorias cuja relação não é clara. Aplicamos o modelo apresentado em um sistema de filtragem de informação como estudo de caso. Neste sistema, é capturada a relação de similaridade entre gêneros musicais e artistas/compositores. No decorrer do artigo será usado o modelo no domínio da filtragem como exemplificação para facilitar o processo cognitivo.

Grafos são estruturas que carregam informações de relações entre seus vértices, portanto é um candidato ideal para modelar as relações. 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.

2.1. Induzindo Relações

Uma das características desejáveis para um modelo do usuário é uma forte capacidade de inferência[Kobsa 2001]. Capturar a relação de todo par de categorias seria exaustivo e necessitaria de uma grande quantidade de dados. Como as relações expressam similaridade, podemos assumir uma transitividade entre categorias não diretamente relacionadas. Seja $adj(v)$ o conjunto de vértices adjacentes no grafo G, dado que $v \in adj(w), w \in adj(x), x \notin adj(v)$ e $v, w, x \in V$, podemos inferir uma relação $\nu_i(v, x)$ através da transitividade: $\nu_i(v, x) = (\nu(v, w) + \nu(w, x))/2$. Formalizando para uma transitividade entre n vértices: seja um caminho entre vértices $c = \{a_1, a_2, ..., a_n\}, \forall i, j \bullet (a_i, a_j) \in E$, definimos uma outra função $\nu_i : V \times V \rightarrow \mathbb{R}$:

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

Esta função induz uma relação entre duas categorias não diretamente relacionadas utilizando os caminhos no grafo. Todavia é natural que existam vários (ou nenhum) caminhos entre dois vértices no grafo. A escolha de um caminho pode ser condicionada a aplicações/domínios específicos, podendo existir caminhos mais interessantes a aplicação; o caminho, por exemplo, pode ser associado à confiabilidade da modelagem das relações em aplicações onde a confiabilidade é uma característica de relativa importância. Os dois candidatos intuitivamente naturais, por serem extremos com relação a função ν_i no modelo, para a escolha de um caminho c entre os demais são: o caminho c_{max} que maximiza $\nu_i(a_1, a_n)$ e c_{min} que minimiza $\nu_i(a_1, a_n)$. Capturemos esta escolha em uma hipótese, relevando a escolha de que caminho melhor funciona na modelagem relativa ao domínio:

Hipótese 1: c_{max} deve ser considerado em vez de c_{min}

Mais adiante será apresentada uma maneira de verificar esta e outras hipóteses apresentadas adiante. É importante observar que para induzir uma relação entre dois vértices a_1 e a_n é necessário que haja um caminho entre os dois vértices. Logo a capacidade de indução do modelo depende da conectividade do grafo construído.

3. Construindo Relações de Preferência

Definido como funciona a estrutura e como deve ser interpretada, é necessário definir uma forma de capturar as informações do usuário para as relações do modelo. Podem existir várias maneiras de construir o modelo, será apresentado nesta seção uma maneira para construir as relações $\nu(v, w)$ baseando-se na preferência do usuário. As relações baseadas em preferência podem ser entendidas semanticamente como um valor

de troca. Por exemplo, no modelo utilizado no estudo de caso, imaginando uma relação $\nu(v, w) = 0.8$, para o usuário significa que uma nota z atribuída ao gênero v valeria 80% caso v fosse "trocado"pelo gênero w. Como as relações estão modeladas em um grafo as relações são bidirecionais (i.e. $\nu(v, w) = \nu(w, v)$), porém nada impede que seja utilizado um dígrafo tornando as relações direcionais.

A construção das relações ocorre incrementalmente de forma que o modelo seja modificado gradualmente e para que seja adaptável a mudanças na preferência do usuário; a adaptação é uma característica desejada como apontado em [Kobsa 2001]. Definimos então uma **sessão** como uma interação do usuário com o sistema. Nessa sessão o usuário expressa sua preferência com relação às categorias que serão modeladas. Essas preferências são capturadas em uma relação $p : V \rightarrow \mathbb{R}$, o modo como essa preferência é capturada depende do domínio/aplicação. No estudo de caso o usuário avalia itens das categorias modeladas (gêneros) com uma nota. A preferência $p(v)$ de uma categoria é dadas pela média das notas dos itens que se enquadram naquela categoria(gênero). Sendo uma sessão $S = \{p(v_1), p(v_2), ..., p(v_k)\}$, podemos construir uma nova relação para cada par de categorias $\forall(v, w) \bullet p(v), p(w) \in S$ da seguinte maneira:

$$\nu_n(v, w) = 1 - |p(v) - p(w)| \tag{2}$$

É importante notar que as relações construídas dependem das preferências expressas em uma sessão, logo preferências expressas em sessões diferentes não serão capturadas em uma relação. Supondo que em uma sessão recente seja capturada uma nova relação $\nu(v, w)$, caso a relação $\nu(v, w)$ exista anteriormente(e.g. capturada a partir de outra sessão) simplesmente definir $\nu(v, w) = \nu_n(v, w)$ causará a perda da informação capturada anteriormente. Por outro lado, desconsiderar a nova informação capturada faria o modelo perder a capacidade de adaptação, caracterizando o problema de *concept drift* [Webb et al. 2001]. É necessário então ponderar sobre a importância da nova informação para o modelo. Podemos então definir uma simples média ponderada para a atualização da relação:

$$\nu(v, w) = \frac{\nu_n(v, w)w_n + \nu(v, w)w_e}{w_n + w_e} \tag{3}$$

Os pesos w_n e w_e expressam a flexibilidade de adaptação do modelo, com w_n representando o peso da relação construída mais recentemente e w_e representando o peso da relação anteriormente construída. Caso $w_n > w_e$ o modelo priorizará informações novas sobre a relação ficando mais flexível, porém se o modelo ficar flexível demais a capacidade de generalização do modelo pode ser afetada. Se $w_e > w_n$ o modelo será mais resistente a mudanças, entretanto quanto mais resistente o modelo for, ele se adaptará com uma menor velocidade. Estes pesos podem ser ajustados ou podem depender da aplicação/domínio. A dúvida sobre estes pesos é capturada em uma hipótese:

Hipótese 2: w_n deve ser maior do que w_e

O grafo então é construído da seguinte forma: as relações então são construídas com a equação 2 e atualizadas com a equação 3, e as categorias nas sessões são incluídas como vértices que são devidamente rotulados com números naturais de forma que não

haja ambiguidade. Descrito e construído o grafo $G(V, E, \mu, \nu)$, é esperado que as relações descrevam bem as relações subjetivas de um usuário. Supondo que exista um grafo G^* descrevendo as relações de forma optimal, é esperado que G aproxime-se de G^*, com isso, temos uma terceira hipótese:

Hipótese 3: G construído segundo as equações 3 e 2 é próximo de G^*

4. Resultados e Experimentação

Para validar as hipóteses apresentadas necessitamos comparar um G construído segundo as equações 3 e 2 com um grafo ótimo G^*. Sendo que G^* contêm a preferência do usuário de maneira correta. Dessa forma G^* só poderá ser indicado explicitamente pelo próprio usuário. Em forma de entrevista, foi solicitada a construção de G^* considerando gêneros musicais como as categorias das relações montadas por algumas pessoas. G foi montado segundo avaliações explícitas (i.e. notas de 0-10) de músicas solicitadas aos mesmos usuários. A seleção de itens ocorreu de forma aleatória com músicas conhecidas pelos próprios usuários. Esta validação de hipóteses pode ser encarada como uma forma de decisão quanto as escolhas capturadas nas mesmas, apontando para a melhor configuração do modelo para um determinado domínio e sua utilidade.

Cada usuário considerado tem um conjunto de avaliações. Em uma sessão são agrupadas um determinado número de avaliações, as médias das notas dos itens avaliados (músicas neste caso) que contem uma determinada categoria (gêneros, neste experimento) v indicam a preferência $p(v)$ do usuário quanto a categoria e assim são montadas as relações de acordo com as equações 2 e 3. Dessa forma as relações são construídas/atualizadas apenas entre categorias que estejam em uma mesma sessão. Logo a ordem das avaliações pode modificar a estrutura do grafo G montado de acordo com as preferências do usuário.

Uma forma de comparação entre G^* e G que foi aplicada nestes experimentos é o problema de Isomorfismo de Grafos com Correção de Erro (*Error Correcting Subgraph Isomorphism*). Este tema descreve o problema de comparação de grafos rotulados; para isso é calculada uma distância de edição (i.e. edições para transformar um grafo em outro) de um grafo de origem P a um subgrafo S do grafo destino D. A distância é calculada segundo a soma dos custos das transformações mínimas necessárias para achar um isomorfismo entre P e $S \subseteq D$ [Wang et al. 1995] [Wong et al. 1990][Messmer 1995], de modo a P ser transformado em um subgrafo S de D. O problema foi considerado por utilizar uma medida de distância entre grafos, necessária para comparar G e G^*. O problema de achar o isomorfismo para subgrafos foi considerado, ao invés de um isomorfismo para grafos(neste caso $S = D$), pelo fato de que as avaliações podem ter informações insuficientes para conseguir montar G^* completo, sendo o suficiente apenas para a captura de uma parte (subgrafo) de G^*.

4.1. Definição do Problema de Error Correcting Subgraph Isomorphism

Dado um grafo G(V,E,μ,ν), as transformações atômicas (δ) consideradas em um grafo, segundo definidas em [Wong et al. 1990], [Messmer 1995] e [Wang et al. 1995] são:

- $\mu(v) \rightarrow l, v \in V, l \in L_V$: a substituição de um rótulo, ou renomeação, de um vértice.

- $\nu(e) \rightarrow l', e \in E, l' \in L_E$: a substituição de um rótulo, ou renomeação, de uma aresta.
- $e \rightarrow \$, e \in E$: a eliminação de uma aresta.
- $v \rightarrow \$, v \in V$: a eliminação de um vértice.
- $\$ \rightarrow e = (v_1, v_2), v_1, v_2 \in V$: a criação de uma aresta entre os vértices v_1 e v_2.
- $\$ \rightarrow v, v \in V'$: a criação de um vértice de um conjunto V'.

Onde \$ denota o elemento nulo, ou seja, que não existe. A última operação não é considerada em problemas de isomorfismo de subgrafos com correção de erro, apenas no problema de achar um isomorfismo entre grafos completos, já que a criação de vértices é uma edição utilizada para aumentar a estrutura básica de G e o isomorfismo é calculado para um subconjunto do grafo destino, ou seja, queremos "casar"o máximo de P possível em D. Cada uma dessas transformações gera um grafo G' derivado de G, pois estas operações modificam G com relação a sua estrutura (i.e. vértices e arestas) e rotulação. Essas transformações estão associadas a um custo através de uma função $c : \delta \rightarrow \mathbb{R}$, como o custo associado ao se eliminar uma aresta por exemplo.

A composição dessas transformações formam um mapeamento Δ (*Mapping*) como mostrado em [Wang et al. 1995][Messmer 1995], de forma a transformar P em um grafo induzido pelo mapeamento $\Delta(P)$. O mapeamento Δ é composto por tuplas $\{(v, w)|v \in P, w \in D$ ou $w = \$\}$, que representam a edição de substituição de v por w. Cada tupla presente em Δ implica em várias transformações, de forma que o custo inferido por cada tupla seria[Wong et al. 1990]:

$$C((p, q_p)) = c(p, q_p) + \sum_{j=1}^{p-1} c((j, p), (q_j, q_p)) \tag{4}$$

Quando um vértice $p \in P$ é mapeado a outro vértice $q_p \in D$ além do custo $c(p, q_p)$ de mapeamento entre vértices, uma aresta $(q_j, q_p) \in D$ deve ter seu equivalente $(j, p) \in P$, logo os custos de adição, modificação ou eliminação de arestas relativos aos outros vértices já mapeados também são contabilizados.

Formalmente, o problema consiste em achar uma função $f = (\Delta, f_\Delta)$, sendo Δ um mapeamento, representando um isomorfismo de subgrafos com correção de erro entre P e D [Messmer 1995][Wang et al. 1995], de forma que:

- Δ é uma mapeamento que origina $\Delta(P)$ um grafo induzido pelo mapeamento .
- f_Δ é um isomorfismo de subgrafos de $\Delta(P)$ para D.

Um isomorfismo é função bijetora $f : V \rightarrow V'$ para dois grafos $P(V, E, \mu, \nu)$ e $D(V', E', \mu', \nu')$ que satisfaz as seguintes condições, definidas segundo [Messmer 1995]:

- $\forall v \in V, \mu(v) = \mu'(f(v))$
- Para qualquer aresta $e = (v_1, v_2) \in E$ existe uma aresta $e' = (f(v_1), f(v_2)) \in E'$ de forma que $\nu(e) = \nu'(e')$, e para qualquer $e' = (v'_1, v'_2) \in E'$ existe uma aresta $e = (f^{-1}(v'_1), f^{-1}(v'_2)) \in E$ de forma que $\nu(e) = \nu'(e')$

Uma função de isomorfismo para subgrafos de P para D é uma função $f : V \rightarrow V'$ injetora, onde existe um subgrafo $S \subseteq D$ de forma que f é um isomorphismo de P para S. A distância $d(P, D)$ seria definida então como apresentado em [Messmer 1995]:

$$d(P, D) = Min_\Delta C(\Delta)| \text{ existe um } f = (\Delta, f_\Delta) \text{ de } P \text{ para } D$$

Achar $d(P, D)$ é um problema NP-Difícil [Messmer 1995][Wang et al. 1995]. Este problema pode ser modelado como uma busca de estados utilizando heurísticas (algoritmo A^*) [Wang et al. 1995][Messmer 1995][Wong et al. 1990]. A literatura mostra que esta é a maneira clássica de resolvê-lo. Entretanto, existem outras soluções que em geral, funcionam melhor para um número grande de vértices[Messmer 1995]. Diferentes trabalhos costumam aplicar heurísticas diferentes para uma busca mais eficiente.

O estado da busca pode ser descrito como $S = \Delta$ um mapeamento parcial pela aplicação de sucessivas transformações atômicas. Os próximos estados são gerados com a adição de mais uma transformação atômica $S_i = \Delta \cup \delta_i$. Um estado solução representa uma mapeamento completo para todos os vértices de P, mas não necessariamente de D. A busca termina quando não é possível mais expandir estados. Para melhorar a busca, é aplicada uma poda de acordo com o estado solução com o menor custo encontrado até então. Os estados que ultrapassam esse custo são eliminados.

4.2. Metodologia e Validação de Hipóteses

Cada usuário considerado tem um conjunto de avaliações (i.e. notas dadas a músicas). Em uma sessão são agrupadas um determinado número de avaliações, as médias das notas dos itens avaliados (músicas neste caso) que contem uma determinada categoria (gênero, neste experimento) v indicam a preferência $p(v)$ do usuário quanto a categoria e assim são montadas as relações de acordo com as equações 2 e 3. Sendo assim as relações são construídas/atualizadas apenas entre categorias que estejam em uma mesma sessão. Logo a ordem das avaliações pode modificar a estrutura do grafo G montado de acordo com as preferências do usuário.

Para cada usuário foram montados 9 possíveis G. Foram consideradas 3 situações com relação a ordem das avaliações apresentadas. Os grafos 1-3 mantém a ordem das avaliações apresentadas, 4-6 e 7-9 embaralham de forma diferente as avaliações. O embaralhamento se refere a uma ordem aleatória diferente da apresentada pelo usuário quanto as avaliações. Em cada uma dessas situações, os três grafos consideram os pesos w_e e w_n de forma diferente: $w_e = 0.5 = w_n = 0.5$, $w_e = 0.75 > w_n = 0.25$ e $w_e = 0.25 < w_n = 0.75$. Um relatório contendo as relações $\{\forall v, w \in V | \nu(v, w), n_i(v, w)\}$ foi gerado para cada G montado, utilizando tanto c_{max} quanto c_{min}.

As funções de custo utilizada para a comparação de G e G^* são:

- $\mu(v) \rightarrow l, v \in V, l \in L_V$: constante, 2.0.
- $\nu(e) \rightarrow l', e \in E, l' \in L_E$: $3.0 + |\nu(e) - l'|$
- $v \rightarrow \$, v \in V$: constante, 5.0.
- $e \rightarrow \$, e \in E$: $7.0 + \nu(e)$.
- $\$ \rightarrow e = (v_1, v_2), v_1, v_2 \in V$: $5.0 + r(e)$.

Com relação a hipótese 1, foram induzidas todas as relações $\nu_i(v, w)$ de G^* e os G montados. A diferença absoluta entre as relações induzidas $\nu_i(v, w)$ com c_{max} de G e de G^* foi : 0.33, enquanto que para c_{min}: 0.30. O erros alcançados utilizando ambos c_{max} e c_{min} são próximos o suficiente de forma que a escolha de qualquer uma das abordagens não tem muito impacto na precisão do modelo no domínio de gêneros. Para a hipótese 3 avaliamos a média absoluta de similaridade encontrada utilizando a distância de edição entre todos os G e G^*. A média de distância foi de 120.25, com o desvio padrão de 29.31.

Considerando o valor absoluto dos pesos e o tamanho dos grafos, a média mostra que G e G^* são relativamente próximos, dessa forma validando a hipótese 3.

A tabela 1 mostra a concentração dos grafos que obtiveram a mínima distância de todos os usuários com relação a variação dos pesos w_e e w_n. A tabela mostra que houve uma maior concentração dos grafos G, construídos com menores distâncias ao optimais G^* quando $w_e < w_n$, ou seja, quando o modelo considera mais importante as relações novas do que as existentes, esta informação é utilizada para a hipótese 2. Outro fato observado é que a quase totalidade dos mínimos estão concentrados nos grafos que foram montados com um embaralhamento das avaliações (5-6,7-9). É possível que a distância possa diminuir, implicando em uma melhor precisão no modelo, caso as relações fosse construídas de maneira inteligente, de forma a deixar o grafo conexo por exemplo. Algoritmos de seleção de avaliações, chamados de avaliação ativa, existem no domínio da filtragem de informação.

Tabela 1. Números de mínimos segundo variação de w_n e w_e

#	$w_e = w_n$	$w_e > w_n$	$w_e < w_n$
1-3	12,5%	0%	0%
2-6	0%	0%	12,5%
7-9	0%	0%	75%

4.3. Uso no Domínio da Filtragem de Informação

O sistema utilizado para testes filtra os itens através do KNN, utilizado para prever a relevância de um item. Para este sistema foram consideradas 3 características que representa a música m : (Gênero, Artista/Compositor,Ano). A medida de similaridade entre anos está expressa na equação 7, onde $dec(a)$ é a década do ano a. Para gênero e artista/compositor duas relações foram utilizadas para comparação: uma similaridade binária descrita pela equação 6 e uma medida de similaridade expressa pela equação 5 a partir das relações montada de modelos construídos para gêneros e artistas/compositores. Na equação 5 uma penalidade de 0.1 é aplicada quando a relação induzida se mostra máxima(i.e. alcança valor 1) variando os valores da relação $\nu_i(v, w)$ entre [0..0.9], isto é feito para valorizar os vizinhos selecionados pela similaridade com os valores da relação direta em detrimento da induzida. A similaridade entre duas músicas seria uma relação de similaridade $s : M \times M \rightarrow \mathbb{R}$, sendo M o conjunto das músicas, obtida pela média ponderada das similaridades utilizadas em cada característica. O método colaborativo utiliza Correlação de Pearson para similaridade entre usuários e a previsão é calculada através de uma média ponderada dos desvios das médias dos vizinhos como mostrado em [Herlocker 2000].

$$s_a(v,w) = \begin{cases} v \in adj(w) & \nu(v,w) \\ v \notin adj(w) & \nu_i(v,w) \end{cases} \qquad s_b(v,w) = \begin{cases} v = w & 1 \\ v \neq w & 0 \end{cases} \tag{5,6}$$

$$s_{ano}(a_i, a_j) = 1 - \frac{|dec(a_i) - dec(a_j)|}{100} \tag{7}$$

A tabela 2 resume os valores encontrados segundo as métricas aplicadas, escolhidas de acordo com as recomendações em [Herlocker 2000]. A métrica MAE (*Mean*

Absolute Error) avalia o erro do valor absoluto da nota(variando de 0-10), o valor apresentado indica o erro médio das notas previstas pelo sistema. A correlação de Kendall Tau analisa a capacidade do sistema de ordenação das recomendações do sistema, seu valor varia de 0-1, que indicam a concordância da ordenação dos itens dada pelo sistema com relação a ordenação do usuário. As métricas de precisão foram aplicadas duas vezes em duas divisões diferentes para o conjunto de avaliações de cada usuário. Uma das divisões considera 80% para treinamento e 20% para testes, a outra considera 50% para teste e treinamento. A divisão foi feita de maneira aleatória e repetida 100 vezes. As métricas foram aplicadas a conjuntos diferentes de características de forma a analisar sua contribuição na qualidade.

Tabela 2. Métricas aplicadas X Conjunto de Características

Características	MAE (0.8/0.2)	Kendall (0.8/0.2)	MAE (0.5/0.5)	Kendall (0.5/0.5)
(GêneroA,ArtistaA)	2,08±1,35	0,36±0,36	2,28±1,18	0,44±0,20
(GêneroB,ArtistaB)	4,48±2,26	0,36±0,36	4,88±1,81	0,41±0,20
(GêneroA,ArtistaA,Ano)	2,12±1,34	0,40±0,37	2,28±1,15	0,48±0,20
(GêneroB,ArtistaB,Ano)	2,30±1,48	0,40±0,37	2,38±1,18	0,50±0,20
(Ano)	2,33±1,46	0,44±0,37	2,34±1,18	0,48±0,19
(Colaborativa)	1,66±1,41	0,28±0,33	2,23±1,48	0,32±0,19

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. Ainda assim, é interessante observar que nessa base de dados a precisão do método colaborativo quando o número de avaliações é menor se aproxima dos resultados obtidos utilizando-se o modelo apresentado.

A natureza da similaridade binária (GêneroB,ArtistaB) 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 Ano), a precisão tende a melhorar. A modelagem utilizada (GêneroA,ArtistaA) trabalha em uma medida de similaridade entre gêneros e artistas mais distribuída utilizando a preferência do usuário, alcançando uma melhor precisão no cálculo da vizinhança e portanto uma melhor precisão geral. O alto desvio padrão do erro (com relação a média) 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. A relevância dos resultados está na comparação entre os métodos e sua clara diferença com relação a precisão, pois são utilizados em uma mesma base de dados.

5. Conclusões

Este trabalho apresentou uma alternativa de modelagem de usuário baseada em grafos. Esta alternativa tenta capturar as relações entre itens baseado em uma indicação de preferência, seja explícita ou não. As hipóteses apresentadas foram validadas empiricamente com a comparação com modelos construídos por usuários voluntários considerados optimais. Os resultados mostram que o modelo captura bem as preferências dos usuários.

A modelagem foi avaliada quanto a sua praticidade e utilidade com relação ao domínio de filtragem de informação. Ela foi utilizada para capturar as preferência dos usuários relativas a cada par de gêneros e artista no domínio da música, sendo esta preferência utilizada no processo de similaridade.

Os resultados no domínio da filtragem apresentados mostraram-se satisfatórios quanto ao uso do modelo apresentado. Quando as relações do modelo foram utilizadas em detrimento da similaridade binária obteve-se precisão superior em relação as métricas MAE e Kendall, principalmente quando utilizado de forma isolada das outras características. Em comparação ao método colaborativo, o modelo apresentou precisão similar quando o número de avaliações consideradas para treinamento foi menor, mas obteve desempenho inferior quando o conjunto de treinamento considerado aumentou.

Referências

Adomavicius, G. and Tuzhilin, A. (2005). "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):734–749.

Allen, R. B. (1990). "User models: theory, method and practice". *International Journal of Man-Machine Studies*.

Fischer, G. (2001). "User modeling in user computer interaction". *User Modeling and User-Adapted Interaction*.

Herlocker, J. L. (2000). *Understanding and improving automated collaborative filtering systems*. PhD thesis, Universisity of Minnesota. Adviser-Joseph A. Konstan.

Kobsa, A. (2001). "Generic user modeling systems". *User Modeling and User-Adapted Interaction*, 11(1-2):49–63.

Messmer, B. T. (1995). *Efficient Graph Matching Algorithms*. PhD thesis, University of Bern, Switzerland.

Rich, E. (1983). "Users are individuals:- individualizing user models". *International Journal of Man-Machine Studies*.

Santini, S. and Jain, R. (1999). "Similarity measures". *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):871–883.

Wang, J. T. L., Zhang, K., and Chirn, G.-W. (1995). "Algorithms for approximate graph matching". *Inf. Sci. Inf. Comput. Sci.*, 82(1-2):45–74.

Webb, G. I., Pazzani, M. J., and Billsus, D. (2001). "Machine learning for user modeling". *User Modeling and User-Adapted Interaction*.

Wong, A., You, M., and Chan, S. (1990). "An algorithm for graph optimal monomorphism". *IEEE Transactions on Systems, Man and Cybernetics*, 20(3):757–768.