FUNDAÇÃO EDSON QUEIROZ UNIVERSIDADE DE FORTALEZA – MIA MESTRADO EM INFORMÁTICA APLICADA

FILTRAGEM COLABORATIVA APLICADA À RECOMENDAÇÃO MUSICAL

Guillem Jorba Cabré

Fortaleza - Ceará
2011

Guillem Jorba Cabré

FILTRAGEM COLABORATIVA APLICADA À RECOMENDAÇÃO MUSICAL

Monografia apresentada para obtenção dos créditos da disciplina Trabalho de Conclusão do Curso do Mestrado em Informática Aplicada da Universidade de Fortaleza, como parte das exigências para graduação no Curso de Engenharia de Telecomunicações.

Orientador: Prof. Cícero Nogueira dos Santos

Fortaleza - Ceará
2011

FILTRAGEM COLABORATIVA APLICADA À RECOMENDAÇÃO MUSICAL

Guillem Jorba Cabré

PARECER: _____

DATA: ___/___/____

BANCA EXAMINADORA:

_____

Prof. Cícero Nogueira dos Santos, D.Sc.
(Prof. Orientador – UNIFOR)

_____

Prof. Raimir Holanda Filho, Dr.
(Membro da Banca Examinadora – UNIFOR)

RESUMO

Não há tantos anos algumas músicas eram realmente raridades quase impossíveis de achar, e o acesso a músicas não comerciais tinha limitações demográficas muito grandes. As pessoas tinham uma quantidade limitada de músicas a sua disposição, e para uma pessoa não era muito difícil saber as músicas das que ele gostava dentre as que ele podia escutar. Atualmente, com o desenvolvimento das novas tecnologias nas últimas décadas, é possível obter uma grande diversidade de novas músicas em apenas minutos, sem necessidade de sair de casa. Diante deste amplo leque de possibilidades, torna-se uma árdua tarefa para os usuários procurar as músicas do mundo que melhor se adaptem ao seu gosto. Mas as novas tecnologias também permitem ajudar aos usuários nesta pesquisa. Este trabalho apresenta um estudo de diferentes métodos baseados em filtragem colaborativa para fornecer recomendações personalizadas aos usuários. A principal informação usada na estratégia é frequência das músicas já escutadas pelo usuário e por outros usuários com gostos semelhantes. Os experimentos realizados focam no uso do algoritmo kNN para predizer a frequência com a qual um determinado usuário escutará uma dada música. Os resultados da predição com o kNN são comparados com os resultados da predição com preditores básicos, baseados em médias.

RESUM

No fa gaires anys, algunes músiques eren realment rareses quasi impossibles de trobar, i l'accés a músiques no comercials tenia limitacions demogràfiques molt grans. Les persones tenien una quantitat limitada de músiques a la seva disposició, i no era gaire difícil per una persona saber les cançons que li agradaven d'entre les que podia escoltar. Actualment, amb el desenvolupament de les noves tecnologies en les darreres dècades, és possible obtenir una gran diversitat de cançons noves en pocs minuts, sense necessitat de sortir de casa. Davant d'aquest ventall de possibilitats, esdevé una feina feixuga per als usuaris cercar les músiques del món que millor s'adapten als seus gustos. Però les noves tecnologies també permeten ajudar els usuaris en aquesta recerca. Aquest treball presenta un estudi de diferents mètodes basats en el filtrat col·laboratiu per proporcionar recomanacions personalitzades als usuaris. La principal informació utilitzada en l'estratègia és la freqüència de les cançons ja escoltades pels usuaris i per altres usuaris semblants. Els experiments realitzats focalitzen en l'ús de l'algoritme kNN per predir la freqüència amb la qual un determinat usuari escoltarà una cançó en particular. Els resultats de la predicció amb el kNN són comparats amb els resultats de la predicció amb predictors bàsics, basats en les mitjanes.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Com o advento da computação pessoal e, principalmente, da Internet, os usuários têm à disposição uma quantidade de músicas maior do que podem chegar a ouvir durante toda a sua vida. Muitas pessoas têm nos seus discos rígidos uma grande quantidade de músicas que nunca escutaram ou que ouviram apenas uma vez. Este trabalho aborda a tarefa de recomendação automática de musicas que, feita de forma eficaz, pode agregar valor a toda esta informação que hoje está armazenada, mas que é subutilizada.

O ser humano é regido por diversos fatores e variáveis que estão além da lógica. Nem todas as pessoas gostam das mesmas músicas e não existe um critério objetivo para saber se alguém vai gostar de alguma música em particular. Isso não acontece apenas com relação a músicas, algo semelhante acontece também com filmes e livros.

A recomendação também desempenha um papel importante no comércio eletrônico. De acordo com Greg Linden (2007), que programou o primeiro sistema de recomendação para Amazon, a recomendação gera um número de vendas várias ordens de grandeza maior do que mostrar apenas os mais vendidos. Um elemento que ilustra essa situação é o Netflix Prize, que a partir de 2 de outubro de 2006 prometeu um prêmio de um milhão de dólares para o primeiro time que conseguisse uma melhoria de 10% no sistema de recomendação da companhia (Netflix, 2006). Este prêmio incentivou a busca e melhoria de algoritmos na área de recomendação de filmes. Finalmente, no dia 21 de setembro de 2009, um time internacional composto por sete membros ganhou o prêmio.

Atualmente existem vários aplicativos disponíveis para recomendação de músicas, tais como Last.fm, Amazon ou Spotify. As recomendações são baseadas em informações implícitas (como o número de vezes que o usuário ouve um artista) ou explícitas (como uma avaliação do usuário) que os usuários fornecem.

O presente trabalho, através do uso de aprendizado de máquina, objetiva oferecer soluções para o problema da recomendação musical. A estratégia de recomendação proposta usa informações implícitas, mais especificamente, o número de vezes que a música foi ouvida pelo usuário. A abordagem é então baseada na predição da frequência de escuta usando a estratégia de aprendizado *k*-vizinhos mais próximos (*kNN - k Nearest Neighbors*). Mais especificamente, dada uma música *m*, para saber se um usuário *u* gostaria de escutar *m*, os *k* usuários mais semelhantes a *u* e que já escutaram a música *m* são usados para calcular uma estimativa do número de vezes que o usuário *u* escutaria *m*. As músicas recomendadas são, por tanto, aquelas que obtiverem um maior valor para esta estimativa. Esta versão do *kNN* foi implementada por alguns times participantes no Netflix Prize tais como (Hong e Tsamis, 2006). Neste trabalho o kNN é avaliado em diferentes variações e comparado com os preditores básicos *média do usuário* e *média da música*.

O restante deste trabalho está dividido em mais quatro capítulos. O segundo capítulo trata sobre o problema de recomendação musical e apresenta uma descrição detalhada do problema, bem como lista alguns trabalhos relacionados.

O terceiro capítulo descreve a abordagem de recomendação musical proposta neste trabalho. Esse capítulo detalha: como se pode fazer recomendação musical usando predições da frequência de escuta; o dataset usado nos experimentos e o preprocessamento que foi realizado; e as estratégias usadas para efetuar a predição da frequência com a qual um usuário escutaria uma determinada música.

O quarto capítulo mostra os experimentos realizados e os resultados obtidos. Para isso, descreve as métricas utilizadas para avaliar os resultados, mostra e explica os diferentes experimentos realizados, analisa com tabelas e figuras os resultados obtidos e tece explicações e conclusões referentes aos resultados de cada um dos experimentos.

Finalmente, o quinto capítulo apresenta as conclusões do trabalho e sugestões para trabalhos futuros.

2. RECOMENDAÇÃO MUSICAL

Este capítulo apresenta o problema da recomendação musical. Na seção 2.1, a tarefa de recomendação musical é descrita. Na seção 2.2, alguns trabalhos relacionados e estratégias que já foram usadas para recomendação musical são apresentados.

2.1. A tarefa de recomendação musical

Devido ao imenso número de músicas atualmente disponíveis e ao fato de que este número cresce a cada dia, a tarefa de selecionar novas músicas e/ou artitas para se escutar pode ser uma tarefa árdua. O principal objetivo da recomendação musical é apresentar, para os usuários, músicas que sejam do seu interesse. Experiências pessoais, condicionamentos culturais, pressões sociais e a estrutura cerebral provavelmente ainda desempenham papéis importantes em influenciar a resposta de um indivíduo a uma peça musical. Apesar desta variedade, parecem existir padrões de preferências musicais que, através do uso de aprendizado de máquina, poderiam ser usados para fazer recomendações.

2.1.1. Formulação matemática do problema de recomendação:

O problema pode ser formalizado como segue (Sarwar et al., 2001):

Há uma lista de m usuários $U = \{u_1, u_2, \dots, u_m\}$ e uma lista de n itens $I = \{i_1, i_2, \dots, i_n\}$. Cada usuário u_i tem uma lista de itens I_{u_i} dos quais temos alguma informação. Esta informação pode ser implícita ou explícita. Note que $I_{u_i} \subseteq I$ e é possível que I_{u_i} seja nulo. Dado um usuário $u_a \in U$, a tarefa de recomendação consiste em encontrar uma lista de N itens $I_r \subset I$ que o usuário u_a ainda não consumiu, $I_r \cap I_{u_a} = \Phi$, mas que provavelmente teria grande interesse em consumir. Nesse processo, é necessário se fazer a predição de quanto o usuário u_a vai gostar de um determinado item $i_j \notin I_{u_a}$. Tal predição geralmente é um valor numérico calculado a partir dos dados históricos dos usuários e dos itens.

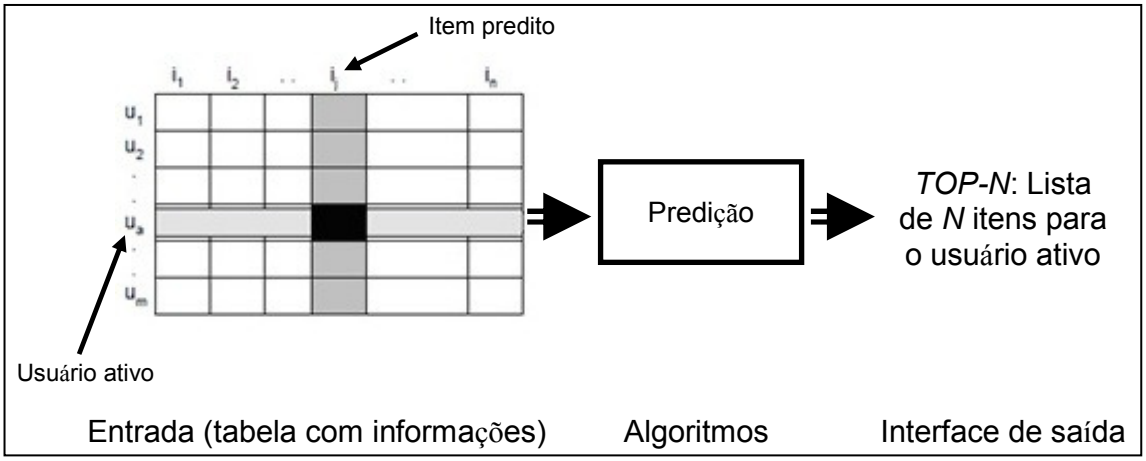

Figura 1: O processo de recomendação.

2.1.2. Tarefas dos sistemas de recomendação:

Para avaliar adequadamente um sistema de recomendação, é importante compreender as metas e tarefas para as quais ele está sendo usado pelo usuário. Herlocker et al. (2004) identificam alguns usos comuns de um recomendador:

- **Encontrar bons itens**: O objetivo deste uso é fornecer uma lista ordenada de itens, juntamente com uma predição de quanto o usuário gostaria de cada item. Idealmente, um usuário poderia esperar alguns itens novos que são desconhecidos pelo usuário, mais também alguns que são familiares.

- **Encontrar todos os bons itens**: A diferença desse caso em relação ao anterior é no que diz respeito à cobertura. Neste caso, a taxa de falsos positivos deve ser mais baixa, apresentando os itens com maior precisão.

- **Recomendar uma sequência**: Visa trazer ao usuário uma sequência ordenada de itens que é agradável como um todo. Um exemplo é a geração de uma lista de recomendação automática.

- **Só navegar**: Neste caso os usuários acham agradável navegar no sistema, mesmo se eles não vão comprar nenhum item. É simplesmente como um entretenimento.

- **Encontrar um recomendador credível**: Os usuários não confiam automaticamente em um recomendador. Eles vão brincar com o sistema para saber se ele faz bem o trabalho (por exemplo procurando um dos seus artistas favoritos e verificando a saída: artistas parecidos, listas geradas, etc.)

- **Exprimir-se**: Para alguns usuários é importante expressar suas opiniões. Um recomendador que oferece uma forma de se comunicar e interagir com outros usuários (através de fóruns, blogs, etc.) permite a autoexpressão dos usuários. Estas expressões permitem que outros usuários possam ter mais informações a partir deles.

- **Influenciar os outros**: Este caso é o mais negativo dos apresentados. Alguns usuários podem querer influenciar os outros em ver ou comprar um item especifico (por exemplo gravadoras tentando promover seus artistas).

2.1.3. Modelo geral:

Os principais elementos de um recomendador são os usuários e os itens. Os usuários precisam ser modelados de maneira que o recomendador possa explorar seus perfis e preferências. Além disso, uma descrição exata dos itens também pode ser útil para obter bons resultados ao recomendar itens para os usuários.

A Figura 2 descreve os principais processos e entidades envolvidos no problema de recomendação. O primeiro passo é modelar os usuários e os itens. Depois disso, dois tipos de recomendações podem ser processados. Uma consiste em apresentar itens recomendados para o usuário (os *N* itens com maior valor predito). A outra consiste em apresentar outros usuários que tenham gostos similares aos do usuário (os *N* vizinhos com maior similaridade). Depois de receber as listas, o usuário pode fornecer um feedback para que o sistema de recomendação possa atualizar seu perfil em conformidade.

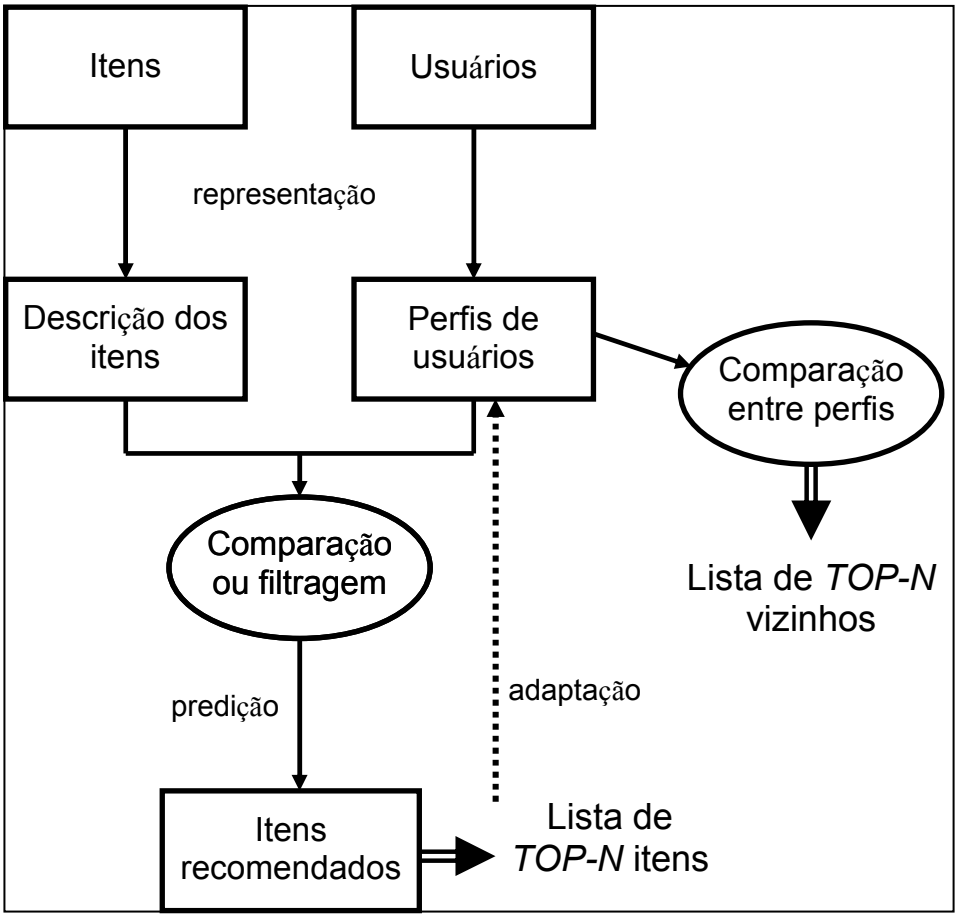

Figura 2: Modelo geral do problema de recomendação.

2.1.4. Métodos de recomendação:

Para fornecer as recomendações interessantes para cada usuário temos que explorar suas preferências. A exploração feita está intimamente relacionada com o método que será usado para filtrar as informações. Os sistemas de recomendação são classificados de acordo com o método adotado para a filtragem da informação, isto é: filtragem demográfica, filtragem colaborativa, filtragem baseada em conteúdo, filtragem baseada em contexto e métodos híbridos. Convém mencionar outro tipo de recomendadores existentes baseados no fornecimento de recomendações para um grupo de usuários, tentando maximizar a satisfação geral do grupo (McCarthy et al., 2006). Os parágrafos seguintes descrevem brevemente os tipos de métodos de recomendação.

- **Filtragem demográfica (*demographic filtering*)**: A filtragem demográfica visa identificar que tipo de usuário gosta de determinada coisa. Para isso reúne os diferentes

usuários em clusters usando informações tais como a idade, o sexo, a cidade onde mora, os estudos feitos ou interesses pessoais. Um exemplo desse tipo de abordagem é usada no sistema LifeStyle Finder (Krulwich, 1997), que tenta identificar em qual dos 62 clusters preexistentes um usuário pertence, e adequa as recomendações para o usuário com base nas informações sobre os outros usuários no cluster. A obtenção de informações demográficas pode ser difícil. LifeStyle Finder entra em diálogo com o usuário para ajudar a classificar o usuário. Outras abordagens tentam obter informações de páginas Web do usuário, blogs, redes sociais, etc. Esta segunda abordagem para a situação faz possível obter informações sem perturbar o usuário.

- **Filtragem colaborativa (*collaborative filtering*)**: Filtragem colaborativa é o processo de filtragem de informação ou de padrões usando técnicas que envolvem a colaboração entre diversos agentes, pontos de vista, fontes de dados, etc. Aplicações da filtragem colaborativa normalmente envolvem conjuntos de dados muito grandes. Este método tem sido utilizado para diversos tipos de dados.

Filtragem colaborativa é um método de fazer predições automáticas (*filtragem*) sobre os interesses de um usuário, coletando informações sobre o gosto de muitos usuários (*colaboradores*). O pressuposto dessa abordagem é que aqueles usuários que concordaram no passado tendem a concordar novamente no futuro: se João gosta de A e B, e Gabriel gosta de A, B e C, então é provável que João goste de C. É por isso que em filtragem colaborativa são calculadas similaridades entre os usuários ou os itens.

Existem duas maneiras principais de obter informações para a filtragem colaborativa:

Feedback explícito (explicit feedback): Os usuários indicam a relevância do item em questão. Este tipo de feedback é definido como explícita apenas quando os avaliadores (ou outros usuários do sistema) sabem que o feedback fornecido é interpretado como tal. A relevância pode ser indicada usando um sistema binário ou uma classificação tal como escala com números, letras ou descrições (por exemplo “não relevante”, “algo relevante”, ”relevante” ou “muito relevante”).

Feedback implícito (implicit feedback): A informação é inferida a partir do comportamento do usuário, sem necessariamente ter sido informado de que será usado como feedback. Portanto o usuário não está avaliando a relevância pelo sistema, mas

está apenas satisfazendo as suas necessidades. Oard e Kim (1998) identificam alguns dos comportamentos observáveis: a seleção de um item, a repetição de um item ou o fato de salvar um item.

- **Filtragem baseada em conteúdo (*content-based filtering*)**: Esses sistemas de recomendação são baseados na similaridade entre os itens. Por exemplo, para recomendação musical, o sistema tem que computar similaridades no áudio considerando características relacionadas com o timbre (como a frequência), com o ritmo (como batidas por minuto, o tipo de métrica) ou com a tonalidade (chave) entre outras.

A similaridade pode ser computada de forma automática com computadores ou de forma manual como a rádio on-line Pandora, que funciona desde o ano 2000. Posteriormente são apresentados para o usuário músicas ou artistas que soam como as do seu perfil.

- **Filtragem baseada em contexto (*context-based filtering*)**: Esta filtragem usa informações diferentes do conteúdo que podem ser usadas para descrever os itens e obter similaridades. No caso da recomendação musical os elementos podem ser tais como tags ou coincidências na mesma sessão. Mobasher et al. (2000) usam as páginas Web mais recentemente visitadas pelo usuário para recomendar as seguintes que poderia visitar, agrupando os usuários em clusters.

- **Métodos híbridos (*hybrid methods*)**: Os métodos híbridos podem aliviar algumas das desvantagens associadas ao fato de usar apenas uma técnica. O mais comum é combinar a filtragem colaborativa com outras técnicas. Burke (2002) identifica sete formas de obter um resultado usando métodos híbridos tais como alternar os resultados fornecidos por vários métodos, misturar os resultados, ponderar os resultados ou aplicar vários métodos em cascata.

2.2. Trabalhos relacionados

Nesta seção serão descritos alguns trabalhos relacionados e algumas estratégias que já foram utilizadas para resolver o problema da recomendação.

2.2.1. Algoritmos de filtragem (Leite, 2002):

O trabalho trata sobre a recomendação de filmes, e considera duas abordagens possíveis para o problema: filtragem baseada em conteúdo e filtragem colaborativa. A primeira analisa a correlação do conteúdo dos itens com o perfil do usuário a fim de sugerir os itens relevantes e descartar os itens insignificantes. O segundo método baseia-se na correlação entre os perfis de usuários. A ideia básica é selecionar os itens preferidos pelos usuários cujas preferências mais se assemelham ao gosto do usuário ativo. O trabalho apresenta uma nova abordagem de filtragem baseada em conteúdo, chamada de Filtragem Baseada em Meta-Protótipos (FMP) para melhorar o desempenho do método do kNN. Finalmente são comparados os desempenhos dos resultados obtidos com diferentes métodos.

2.2.2. OL-RadioUJA. Radio Colaborativa bajo Licencia Creative Commons. (Espinilla et al., 2009):

O trabalho de Espinilla et al. (2009) é baseado na recomendação de músicas para a rádio na Internet OL-RadioUJA. O sistema inclui apenas 197 músicas. A similaridade é computada usando o coeficiente cosseno (os itens são representados por vetores no espaço e a semelhança entre eles é dado pelo cosseno do ângulo) para obter a similaridade entre as músicas. Em seguida, o algoritmo kNN é utilizado para formar o grupo de usuários mais semelhantes para cada um dos usuários do banco de dados. O último passo é calcular a predição com o método da soma ponderada: este método computa a predição de uma música i para um usuário u_a como a soma das notas do usuário u_a para as músicas semelhantes à música i. Cada uma dessas notas é ponderada pela similaridade correspondente, chamada de $s(i,j)$, entre as músicas i e j.

2.2.3. A Probabilistic Model for Music Recommendation Considering Audio Features (Li et al., 2005):

O problema da recomendação musical é abordado através de um método híbrido baseado em filtragem colaborativa e filtragem baseada em conteúdo. O trabalho tenta resolver três problemas do uso de clusters em filtragem colaborativa:

- Dois itens semelhantes que nunca foram ouvidos pelo mesmo usuário não podem ser classificados na mesma comunidade.
- Se um usuário gosta especialmente de um gênero musical, a filtragem colaborativa não dá prioridade a este gênero quando faz recomendações.
- Não há informação suficiente sobre as novas músicas no sistema para fazer recomendações.

Para enfrentar estes problemas, o trabalho visa a obtenção de informações sobre o conteúdo tais como o timbre, o ritmo, a melodia e a harmonia.

2.2.4. Improved Neighborhood-based Collaborative Filtering (Bell e Koren, 2007):

Bell e Koren usaram o conjunto de dados disponibilizado pela Netfix para o NetflixPrize e investigaram o uso do kNN para a recomendação de filmes. Eles conseguiram obter predição mais precisas, sem um aumento significativo no tempo de execução. O sistema não exige o treinamento de muitos parâmetros nem de um pré-processamento longo. Isso faz dele uma opção prática para aplicações de grande escala. Bell e Koren tentam evitar efeitos negativos tais como:

- Filmes avaliados como ruins com vizinhos com uma boa avaliação tendem a ter uma estimação alta.
- Alguns filmes foram avaliados principalmente por usuários que pontuam melhor. Também existe o caso contrário.
- Alguns usuários gostam de filmes conhecidos, outros preferem filmes mais especializados.
- Ao longo do tempo as opiniões dos usuários podem mudar.

2.2.5. Music Recommendation and Discovery in the Long Tail (Celma, 2008):

A preocupação principal deste projeto é a recomendação de músicas novas ou pouco conhecidas, pois o autor demonstra que sistemas de recomendação como o usado pelo last.fm tendem a reforçar artistas populares em detrimento do descarte de músicas menos conhecidas. Outro aspecto ao que o autor dedica especial atenção é o usuário. Alguns usuários não querem apenas receber as recomendações, mas também querem saber por que essa música é recomendada para eles. Ele também busca uma interação com os usuários para determinar se gostam das recomendações recebidas ou não, para depois adaptar o sistema dependendo do feedback fornecido pelo usuário. Depois de avaliar e comparar alguns sistemas de recomendação já existentes no mercado, três algoritmos de recomendação são usados: filtragem colaborativa, filtragem baseada em conteúdo e uma abordagem híbrida. Esta última abordagem combina artistas relacionados (obtidos na Internet) com a similaridade no som: dada uma música o sistema procura os artistas relacionados. Em seguida, classifica as músicas destes artistas com um sistema baseado na similaridade do áudio.

3. FILTRAGEM COLABORATIVA APLICADA À RECOMENDAÇÃO MUSICAL

Este capítulo apresenta a estratégia de recomendação musical proposta neste trabalho, bem como os algoritmos de aprendizado de máquina utilizados. A seção 3.1 mostra como a tarefa da recomendação pode ser realizada com o uso de predição da frequência de escuta. A seção 3.2 descreve os passos aplicados na formatação do conjunto de dados usado e especifica a criação dos conjuntos de treinamento e teste. Em seguida, a seção 3.3 expõe as estratégias de predição utilizadas.

3.1. Abordagem de recomendação proposta

O presente trabalho usa uma abordagem baseada em filtragem colaborativa para o problema de recomendação musical. Como é utilizado um conjunto de dados em que não existe um *feedback* explícito, a repetição de itens será a principal informação com relação à preferencia de músicas pelos usuários. Mais especificamente, utilizaremos a frequência com a qual os usuários escutam as músicas como elemento chave para a recomendação.

O sistema de recomendação proposto no trabalho usa estratégias de aprendizado de máquina para fazer a predição da frequência com a qual o usuário escutaria uma nova música dada. Aqui explicamos como se podem usar estas predições a fim de recomendar músicas. Cabe mencionar que, na teoria da recomendação, as músicas com uma predição de frequência mais elevada são as melhores candidatas, pois se espera sejam as que o usuário escutará com maior frequência.

O sistema segue os seguintes passos:

1 – **modelar usuários e música**: A seguinte seção, 3.2, mostra o processo seguido para a obtenção da matriz usuários x músicas, similar ao que aparece na Figura 1. Cada linha da matriz representa um vetor que modela um usuário. Da mesma forma, cada coluna modela uma música. Cada célula da matriz contém a frequência com a qual um usuário

(linha) escutou uma determinada música (coluna). É importante salientar que esta matriz é extremamente esparsa. Ou seja, a grande maioria das células não estão preenchidas.

2 – **obtenção da similaridade entre os usuários**: O método da correlação de Pearson é aplicado aos vetores dos usuários para se obter a similaridade entre estes. A subseção 3.3.2 explica esse procedimento.

3 – **seleção das músicas candidatas**: Existem diversos métodos para a seleção das músicas candidatas. O requisito principal é que as músicas têm que ser novas, ou seja, que ainda não foram escutadas pelo usuário ativo. Logo, as candidatas podem ser (1) todas as músicas não escutadas, (2) selecionadas aleatoriamente entre todas as músicas não escutadas até encontrar as TOP-N músicas necessárias, ou (3) pode-se restringir a busca apenas às músicas ouvidas pelos *k* usuários mais semelhantes (*k* vizinhos) e que ainda não foram escutadas pelo usuário ativo.

4 – **predição de músicas candidatas**: A seção 3.3 mostra distintos métodos usados neste trabalho para a predição da frequência (médias, kNN, híbridos). Estes métodos têm por objetivo preencher células vazias da matriz usuários-músicas com as predições obtidas pelas músicas candidatas.

5 – **recomendação das músicas com maior valor de frequência predito**: De acordo com o método utilizado na seleção das candidatas, existem critérios diferentes para decidir se recomendar uma música ou não. No caso da seleção aleatoria, o critério pode ser definido em termos absolutos "músicas com uma frequência estimada superior a um valor *x* serão recomendadas", ou relativos "músicas com uma frequência estimada superior à média do usuário serão recomendadas". O processo deve ser repetido até obter o número desejado de recomendações. Nos outros dois casos as músicas que serão recomendadas são as *TOP-N*, isto é, as *N* músicas com maior frequência predita.

Neste trabalho, restringimos o nosso foco na elaboração dos quatro primeiros passos do sistema de recomendação. Uma grande atenção foi dada no passo quatro, que consiste na predição da frequência de escuta. O último passo, a criação de uma lista com as recomendações para os usuários, não é executado neste trabalho.

3.2. Conjunto de dados utilizado

O dataset utilizado para fazer as estimações foi criado por Oscar Celma (2009) com o método user.getRecentTracks do Last.fm. O formato dos dados é de uma música ouvida por linha (separados por tabulador, "\t"). A Tabela 1 apresenta o formato e alguns exemplos de linhas do dataset.

id-do-usuario selo-de-tempo id-do-artista nome-do-artista id-da-musica nome-da-musica
user_000005 2009-04-28T10:25:31Z 9fa0e4be-4cd9-43fb-8b20-ad07c15b3b97 Guru Josh Project Live At United Respect Essen-01- 18-Sat-2009
user_000005 2009-04-28T09:29:08Z b09b5127-c62e-4bb2-b790-1e4aa18749ed Armand Van Helden Live At United Respect (Essen)-Sat-01-18-2009
user_000005 2009-04-28T09:24:37Z c98d40fd-f6cf-4b26-883e-eaa515ee2851 The Cranberries c181fb92-5699-43ee-9f6e-c7ddc338aea3 Linger
user_000005 2009-04-28T09:19:55Z 265f242e-cf4e-4fbe-a3fe-43112387172f TÃ©lÃ©popmusik 7bcf8f24-38bb-4e27-87a8-8e036854f8cc Breathe
user_000005 2009-04-27T22:40:54Z 2d44d331-e622-4242-9e5e-0146dbfc328e Martijn Ten Velden & Lucien Foort d45d917e-d4a1-4cc2-9fea-66be069b1eba Bleeep! (Original Mix)
user_000005 2009-04-27T22:36:41Z 3de58089-b9f0-4999-b405-ed17237b5cf2 Something Brothers Martin Solveig
user_000005 2009-04-27T22:32:41Z 79239441-bfd5-4981-a70c-55c3f15c1287 Madonna a54de956-3d99-4a91-be34-96a75a8b64b3 Get Together
user_000005 2009-04-27T22:26:39Z 79239441-bfd5-4981-a70c-55c3f15c1287 Madonna b1cfa88d-7678-4936-9db9-a1d97aba225f Sorry (Man With Guitar Mix)
user_000005 2009-04-27T22:22:49Z 664a8867-2e24-4d05-9bae-de5f7b16f17b M.A.N.D.Y. Vs. Booka Shade a6f039df-62ca-4e81-a41c-bb1b2a207e59 Body Language
user_000005 2009-04-27T22:10:58Z 1577f48a-dc2d-45f1-b7ac-64c25b44bca8 Kurd Maverick 56c5f699-ecfa-453a-beff-2c0915a789b3 The Rub (I Never Rock)

Tabela 1: Exemplo do conjunto de dados.

Os dados usados neste trabalho são apenas o usuário, o nome do artista e o nome da música, porque algumas das músicas e alguns dos artistas não têm id. A Tabela 2 apresenta algumas estatísticas do conjunto de dados.

Total de linhas	19.150.868
Total de diferentes usuários	992
Total de diferentes artistas	174.091
Total de nomes de músicas diferentes	1.084.873
Total de músicas diferentes	1.500.661

Total de grupos usuário-artista-música	4.618.291
Media de vezes que uma música é ouvida	12,76
Número médio de vezes que um usuário ouve cada música	3,078
Número médio de músicas diferentes ouvidas por usuário	4655,5
Número médio de usuários que escutam cada música	4,147

Tabela 2: Estadísticas dos dados.

O dataset fornecido por Celma (2009) é um arquivo muito grande (Figura 3). Isso faz dele um arquivo que teria um tempo de computação muito longo e que exigiria muitos recursos do computador. A primeira parte do trabalho se concentrou, portanto, na redução do tamanho do arquivo. Em seguida, foi realizada a separação do dataset em dois conjuntos de dados: o conjunto de treino para fazer o aprendizado dos modelos e o conjunto de teste para a avaliação dos resultados.

3.2.1. Formatação do dataset:

A formatação dos dados é baseada em três principais aspectos que permitem uma grande redução no tamanho do arquivo:

- **Excesso de informação**: O arquivo de origem para este trabalho tem muitas informações disponíveis, mas apenas uma parte delas é usada para a estimativa da frequência. Por exemplo, o selo de tempo (o dia e hora que a música foi ouvida) não é usado.

- **Codificação dos dados**: Algumas das informações estão codificadas de uma maneira muito pesada. Com uma nova codificação pode-se conseguir uma boa redução no seu tamanho. Por exemplo, o id de um usuário tal como *user_000639* pode se codificar em um número com três dígitos (há 992 usuários). Essa redução também permite ao programa processar as informações dos usuários (em Java) como *int* (32 bits) em vez de *string char* (16 bits x 11 caracteres = 176 bits).

- **Informações redundantes**: O dataset contém informações redundantes. Por exemplo o nome da música e o nome do artista estão bastante relacionados. Quase sempre é possível saber o nome do artista a partir do nome da música, a menos que existam diferentes versões ou coincidências. Adicionalmente, depois de aplicar as alterações

acima, o arquivo resultante contém linhas repetidas (para cada vez que um mesmo usuário ouve uma mesma música). Portanto, o último passo da formatação é agrupar essas linhas em uma única.

O arquivo fornecido (userid-timestamp-artid-artname-traid-traname.tsv) tem o formato seguente:

```
id-do-usuario selo-de-tempo id-do-artista nome-do-artista id-da-musica nome-da-musica
```

Três programas, os quais são relacionados na Figura 3, são usados para fazer a formatação:

- **PrimeiraReduccNova.java**: O arquivo codifica a id do usuário, o nome do artista e o nome da música, cria um arquivo de saída (1_new_dataset.tsv) com o dataset reduzido e três arquivos com as codificações. A estrutura dos arquivos é:

1_new_dataset.tsv:
#usuario \t #artista \t #nome-da-musica
#usuario \t #artista \t #nome-da-musica
...
1_users_table.tsv:
id-do-usuario \t #usuario
id-do-usuario \t #usuario
...
1_artists_table.tsv:
nome-do-artista \t #artista
nome-do-artista \t #artista
...
1_songs_table.tsv:
nome-da-musica \t #nome-da-musica
nome-da-musica \t #nome-damusica
...

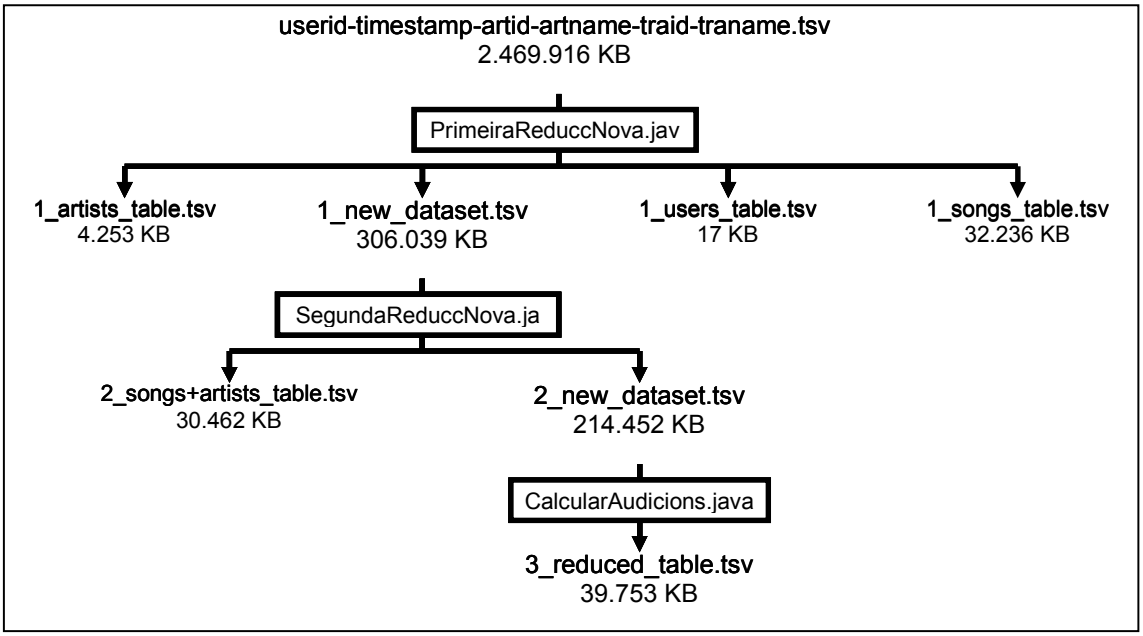

Figura 3: Programas usados para a formatação do dataset.

- **SegundaReduccNova.java**: O arquivo codifica as duplas artista/nome-da-musica com um único número (representa uma única música) e cria um arquivo com o novo dataset reduzido e outro com a tabela usada. A estrutura dos arquivos resultantes é a seguinte:

2_new_dataset.tsv:

#usuario \t #musica

#usuario \t #musica

...

2_songs+artists_table.tsv:

#artista \t #nome-da-musica \t #musica

#artista \t #nome-da-musica \t #musica

...

- **CalcularAudicions.java**: O último dos três arquivos conta quantas vezes cada usuário escutou cada música (a frequência usada para as estimações) e o número de musicas diferentes que cada usuário escutou. Em seguida, cria o arquivo resultante de todo o processo de formatação:

3_reduced_table.tsv:

#usuario1:#musicas-ouvidas,#musica:#reproduções,#musica:# reproduções,...

#usuario2:#musicas-ouvidas,#musica:#reproduções,#musica:# reproduções,...

...

Após da formatação do conjunto de dados, o arquivo inicial de tamanho 2.412MB foi reduzido a um arquivo útil muito menor de tamanho 38,8 MB. Ou seja, o arquivo final possui 1,61% do tamanho inicial (um pouco mais de 62 vezes menor). Além disso, os dados agora podem ser lidos como *int*, o que ocupa menos espaço de memória ao usar a linguagem Java.

3.2.2. Criação dos conjuntos de treino e teste:

Após da formatação do dataset e antes de começar o cálculo de similaridades, a criação de dois conjuntos é necessária:

- **Conjunto de treino**: Contém 90% dos dados e é usado para treinar os algoritmos de aprendizado de predição. Tais algoritmos envolvem o calculo das similaridades entre os usuários.

- **Conjunto de teste**: É composto pelos 10% dos dados restantes. Esse conjunto de dados é utilizado para a aplicação dos modelos de predição aprendidos com o uso do conjunto de treino. Em cada experimento, cada par (usuário, música) do conjunto de teste tem a sua frequência de escuta predita. Em seguida, as frequências preditas são comparadas com os valores das frequências reais possibilitando a avaliação da qualidade do preditor.

A Figura 4. ilustra a separação do conjunto de dados em dados de treino e de teste.

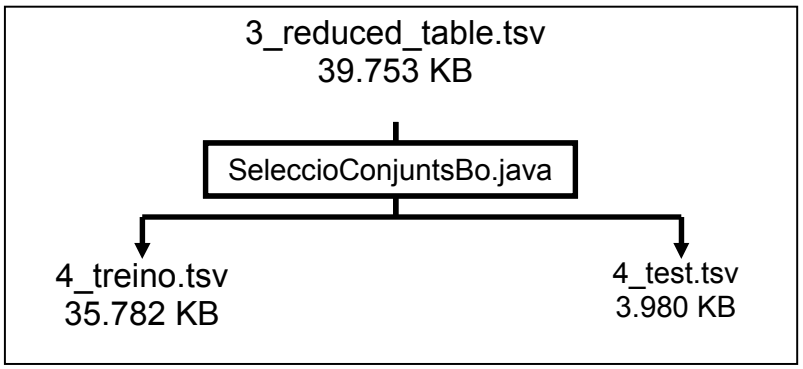

Figura 4: Programas usados para a criação dos conjuntos.

A Figura 5 mostra o algoritmo utilizado para separar os dados do dataset em dois conjuntos no programa SeleccioConjuntsBo.java.

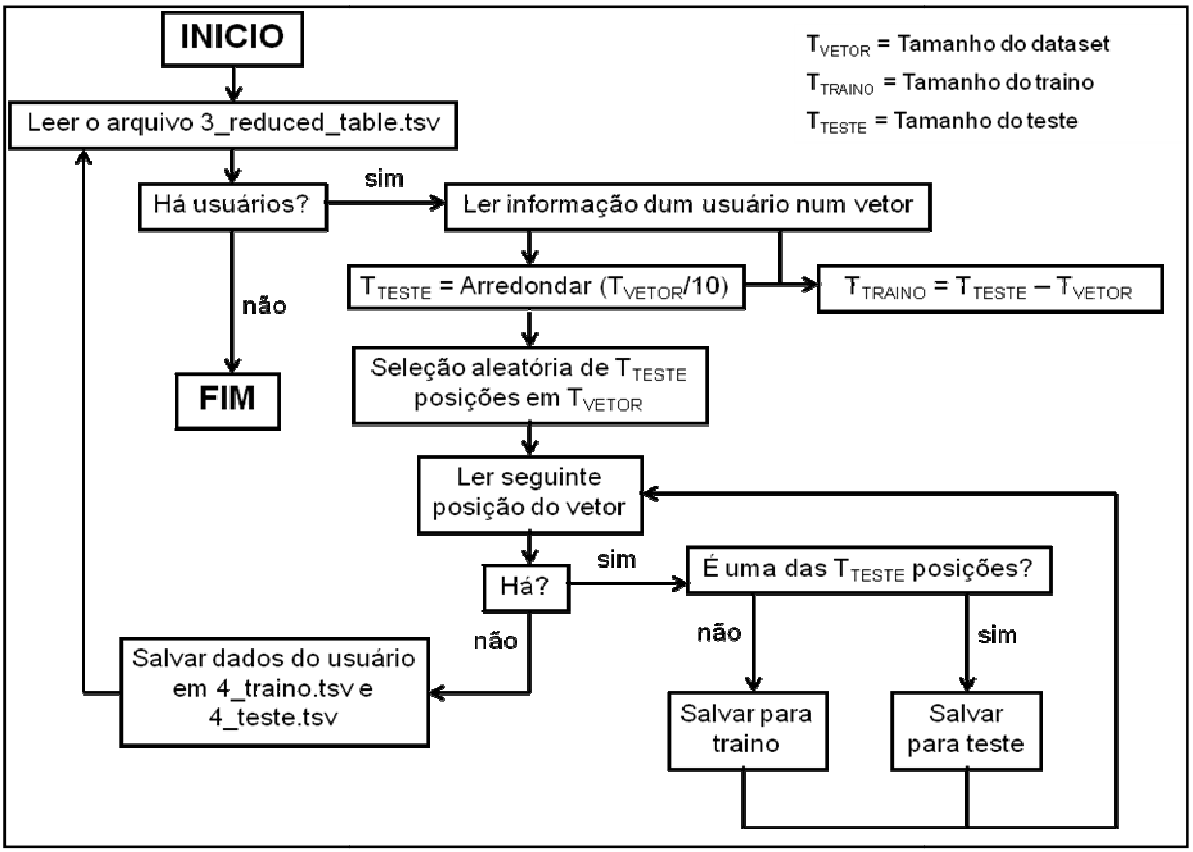

Figura 5: Algoritmo do programa SeleccioConjuntsBo.java.

Algumas estatísticas dos conjuntos de treino e teste são mostradas na Tabela 3.

Dados no conjunto de treinamento (em %)	90
Dados no conjunto de teste (em %)	10
Tamanho do arquivo do conjunto de treinamento	35.782
Tamanho do arquivo do conjunto de teste	3.980
Total de grupos usuário-artista-música no conjunto de treinamento	4.156.892
Total de grupos usuário-artista-música no conjunto de teste	461.399
Total de grupos usuário-artista-música que aparecen só no teste	103.060
Músicas que aparecem só no teste (em %)	22,34

Tabela 3: Estatísticas dos dados de treinamento e de teste.

O detalhe mais notável dos dados é que existem poucos usuários e muitas músicas. Portanto, a matriz usuários-músicas (cada elemento com o número de reproduções dum usuário para uma música) é uma matriz com poucas linhas, muitas colunas e uma imensa quantidade de células vazias. Outro elemento que vale ressaltar é o fato que 22,34% das músicas aparecem só no conjunto de teste (selecionado aleatoriamente), e muitas aparecem apenas uma vez. Extrapolando, pode-se supor que mais de 20% das músicas aparecem apenas uma vez, ou seja, apenas um usuário escutou essa música. Isso faz delas canções para as quais é difícil fazer uma predição. Este é um dos principais problemas que este trabalho enfrenta.

3.3. Estratégias de predição utilizadas

Esta seção mostra as principais estratégias utilizadas para fazer as predições. No próximo capítulo serão avaliados os resultados da aplicação dos diferentes métodos, e também serão feitos algums experimentos e modificações com os mesmos.

3.3.1. Médias:

A maneira mais fácil e intuitiva de fazer uma predição é através do cálculo da média. Se um usuário ouve as músicas muitas vezes, o esperado para uma nova música é que ele também escute essa nova música muitas vezes. Igualmente, se uma música é ouvida muitas vezes por aqueles usuários que a conhecem é provável que os outros usuários também gostariam de ouvi-la muitas vezes.

Esta maneira de fazer predições consegue, as vezes, boms resultados. A principal desventagem é que as predições obtidas não permitem fazer boas recomendações (nem recomendações personalizadas), já que o preditor média do usuário atribui um mesmo valor para tudas as novas músicas e o preditor média da música atribui para todos os usuários o mesmo valor para uma música. Este último método recomenda apenas as músicas mais populares para todos.

A principal utilidade da predição pelas médias é obter resultados que possam ser utilizados para comparação com outros métodos. Se o método proposto não é melhor do que a média, então o método não generaliza bem. As médias calculadas e suas fórmulas matemáticas são:

- **Média do usuário**: A média $\widehat{f_i}$ de vezes que o usuário i ouve as k músicas da sua coleção é calculada com suas frequências f_{ik} para essas músicas:

$$\widehat{f_i} = \frac{\sum_k f_{ik}}{k}$$

- **Média da música**: A média $\widehat{f_j}$ de vezes que a música j é ouvida pelos k usuários de sua lista é calculada com as frequências f_{kj} dos usuários para essa música:

$$\widehat{f_j} = \frac{\sum_k f_{kj}}{k}$$

- **Média das médias**: A média das médias é calculada para ter em conta os dois fatores anteriores. A média do usuário é a mesma para todas as músicas, a média da música é a mesma para todos os usuários. A média $\widehat{f_{ij}}$ das médias é específica para cada música j e cada usuário i:

$$\widehat{f_{ij}} = \frac{\widehat{f_i}.\widehat{f_j}}{2}$$

- **Média do artista**: A média $\widehat{f_{AR_j}}$ de vezes que as músicas do artista da música j são escutadas é calculada com as frequências $\widehat{f_{ij}}$ dos usuários para essas músicas:

$$\widehat{f_{AR_j}} = \frac{\sum_i \sum_{j \in ARTISTA} f_{ij}}{\sum_i \sum_{j \in ARTISTA} 1}$$

3.3.2. kNN:

Métodos baseados em kNN (*k Nearest Neighbors*, k Vizinhos Mais Próximos) são as estratégias mais simples e com mais sucesso na resolução de problemas de reconhecimento de padrões. Apesar de sua idade, eles são particularmente competitivos com algoritmos mais modernos e sofisticados. É por isso que, de acordo com Mackey (2009), kNN é o método de filtragem colaborativa mais utilizado.

O método pode estar baseado no usuário ou na música. Neste trabalho o kNN é baseado no usuário. Cada usuário é representado por um vetor incompleto com suas frequências para as músicas. A tarefa da recomendação visa completar o vetor com estimações e apresentar para o usuário as músicas com maior frequência esperada. A Figura 6 mostra a estratégia usada no kNN para completar cada elemento do vetor (predição de uma música para um usuário).

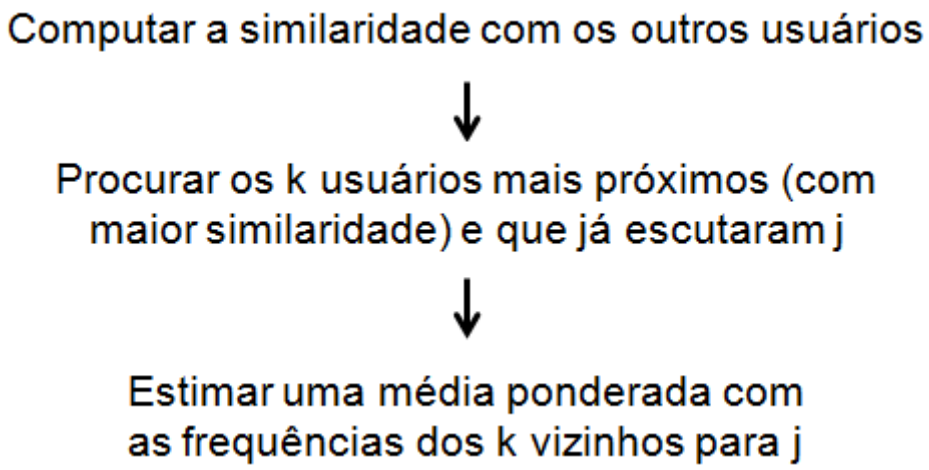

Figura 6: Estratégia do kNN para uma música *j*.

Os seguintes pontos descrevem os métodos utilizados para calcular a similaridade entre usuários e a média ponderada.

- kNN sem padronizar: Calcula-se o coeficiente de correlação de Pearson $\widehat{\rho_{xy}}$ seguindo a seguinte fórmula:

$$\widehat{\rho_{xy}} = \frac{\sum_k (f_{xk} - \mu_x).(f_{yk} - \mu_y)}{\sqrt{\sum_k (f_{xk} - \mu_x)^2} . \sqrt{\sum_k (f_{yk} - \mu_y)^2}}$$

Onde a variável k percorre as músicas que têm em comum os usuários x e y, f_{xk} é a frequência da música k pelo usuário x, e μ_x é a média aritméticas geral do usuário x para todas as N músicas que ele tem escutadas:

$$\mu_x = \frac{1}{N} \sum_{n=1}^{N} f_{xn}$$

O valor obtido está no intervalo $[-1,1]$. Para obter um valor da similaridade no intervalo $[0,1]$ aplica-se a seguinte correção:

$$sim(x,y) = \frac{\widehat{\rho_{xy}} + 1}{2}$$

Neste ponto, para fazer a estimação $\widehat{f_{ij}}$ da música j para o usuário i, são procurados os k vizinhos mais próximos e que já escutaram a música j. Com as frequências f_{kj} dos vizinhos para essa música calcula-se a predição:

$$\widehat{f_{ij}} = \frac{\sum_k sim(i,k).(f_{kj} - \mu_k)}{\sum_k sim(i,k)} + \mu_i$$

- **kNN padronizado**: Neste caso, antes do cálculo da similaridade, os dados da frequência dos usuários são padronizados da seguinte forma:

$$F_{ij} = \frac{f_{ij} - \mu_i}{\sigma_i}$$

Onde F_{ij} é a frequência padronizada e σ_i o desvio padrão geral do usuário i para as N músicas da sua coleção:

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(f_{in} - \mu_i)^2}$$

O coeficiente de correlação de Pearson se calcula então como:

$$\widehat{\rho_{xy}} = \frac{\sum_k (F_{xk} - \nu_x).(F_{yk} - \nu_y)}{\sqrt{\sum_k (F_{xk} - \nu_x)^2} . \sqrt{\sum_k (F_{yk} - \nu_y)^2}}$$

Onde ν_x é a media aritméticas geral das frequências padronizados para o usuário x:

$$\nu_x = \frac{1}{N}\sum_{n=1}^{N} F_{xn}$$

Neste ponto, a mesma correção da seção anterior é feita para obter um valor da similaridade entre 0 e 1. Após, cálcula-se a predição padronizada usando os k vizinhos mais próximos pelo seguinte método:

$$\widehat{F_{ij}} = \frac{\sum_k sim(i,k).F_{kj}}{\sum_k sim(i,k)}$$

Finalmente, a padronização é desfeita para obter un valor estimado da frequência da música *j* para o usuário *i*:

$$\widehat{f_{ij}} = \widehat{F_{ij}}.\sigma_i + \mu_i$$

3.3.3. Métodos híbridos:

O conceito principal da abordagem é que métodos distintos podem ser combinados de alguma forma para obter resultados melhores. Neste trabalho os métodos híbridos são aplicados como combinação ponderada das estimações finais obtidas por distintos métodos pelas músicas do conjunto de teste. A predição híbrida $\widehat{freq_{est}}$ da frequência para uma música é calculada da seguinte forma:

$$\widehat{freq_{est}} = \sum_{i \in METODOS} \widehat{freq_i}.w_i$$

Com as seguintes condições:

$$1 = \sum_{i \in METODOS} w_i$$

e

$$0 \leq w_i \leq 1$$

E onde $\widehat{freq_i}$ é a estimação da frequência pela música obtida pelo método i. Na prática, são combinados um máximo de três métodos, com $w_i = 0$ para os outros métodos.

4. EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os experimentos realizados e comenta os resultados obtidos com os diferentes métodos utilizados. Na seção 4.1, a métrica utilizada para a avaliação dos resultados é descrita. Nas seguintes seções são avaliados os resultados obtidos pelos distintos métodos: médias na seção 4.2, o kNN na seção 4.3 e finalmente métodos híbridos na seção 4.4.

4.1. Métrica utilizada: RMSE

A métrica utilizada neste trabalho para a avaliação dos resultados obtidos nas predições é o *root mean square error* (RMSE). Esta métrica fornece informação de quão longe dois objetos se encontram. Neste caso se deseja saber quão longe as predições obtidas se encontram das frequências reais presentes no conjunto de teste.

Esta métrica é a mesma utilizada no Netflix Prize (2006), que prometeu um prêmio de um milhão de dólares para o primeiro que conseguisse um sistema de recomendação cujo RMSE fosse 10% menor do que o RMSE do sistema de recomendação da companhia. Este fato demonstra o quão aceita é essa métrica de avaliação de sistemas de recomendação. O RMSE é calculado da seguinte forma:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}\sum_{j \in TESTE}\left(f_{ij} - \widehat{f_{ij}}\right)^2}$$

Onde a variável *i* percorre todos os usuários e a variável *j* percorre as músicas de cada usuário no conjunto de teste (desta maneira os dois somatórios percorrem todas os *N* casos do conjunto de teste).

Na avaliação dos resultados, quanto maior o RMSE pior são as estimações (mais longe as predições estão do valor que tentam estimar), portanto o método usado para calcular as estimações é pior. Ao contrario, quanto menor é o RMSE melhor são as estimações e melhor é o método usado.

4.2. Médias:

Os erros obtidos para as estimações das médias são os seguintes:

$RMSE_{MEDIA\ USUARIO} = 9{,}586825$
$RMSE_{MEDIA\ MUSICAS} = 10{,}627651$
$RMSE_{MEDIA\ MEDIAS} = 9{,}7921505$
$RMSE_{MEDIA\ ARTISTA} = 10{,}580656$

O maior erro acontece para a predição com a média da música. Dois fatores causam este fato. O primeiro é que muitas músicas (22,34%) aparecem apenas no conjunto de teste. O segundo é que, em média, cada música foi ouvida apenas por 4,15 usuários. Portanto a média da música muitas vezes não aporta uma informação boa.

O menor erro ocorre para a média do usuário. Este fato acontece porque alguns usuários escutam as músicas muitas vezes (por exemplo usuários velhos que escutam muito um tipo de música) e outros usuários escutam as músicas poucas vezes (por exemplo os novos usuários). Nestes casos a média do usuário é um bom método para obter predições.

Conforme mencionado no capítulo anterior, as estimações computadas com estes métodos não permitem fazer recomendações personalizadas, mas os erros obtidos são uma referência para a avaliação dos erros obtidos com outros métodos.

4.3. kNN:

4.3.1. Comparação dos distintos métodos para calcular o kNN:

- **kNN sem a padronização das frequências**: A Tabela 4 e as Figuras 7 e 8 apresentam os resultados obtidos com o kNN sem a padronização das frequências dependendo do valor de k (número de vizinhos utilizados para as predições).

K	RMSE
1	13,381697
2	11,497281
3	10,817388
4	10,484548
5	10,301241
6	10,2024355
7	10,13098
8	10,083312
9	10,047913
10	10,028874
11	10,01511
12	10,004413
13	9,99861
14	9,990335
15	9,983076
16	9,976025
17	9,973182
18	9,968956
19	9,96553
20	9,962474
21	9,961437
22	9,961573
23	9,960375
24	9,959433
25	9,959143
26	9,959985
27	9,960046
28	9,959513
29	9,959339
30	9,960554
31	9,961654

Tabela 4: Resultados kNN sem padronizar.

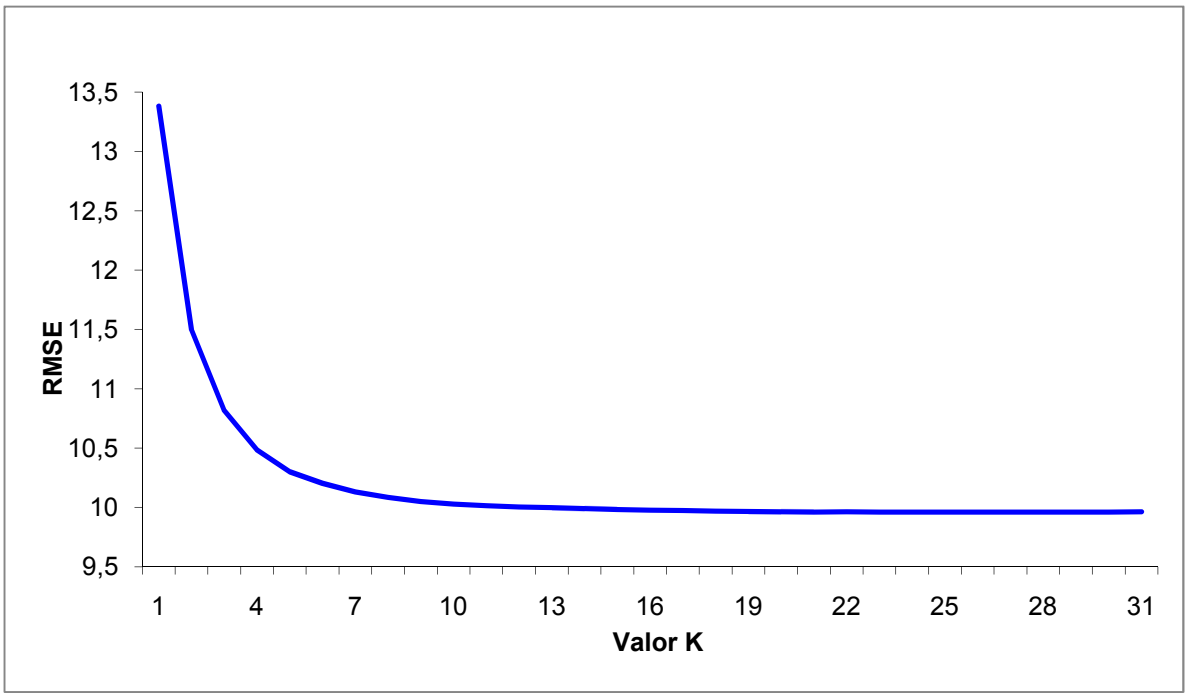

Figura 7: Evolução do RMSE, kNN sem padronizar.

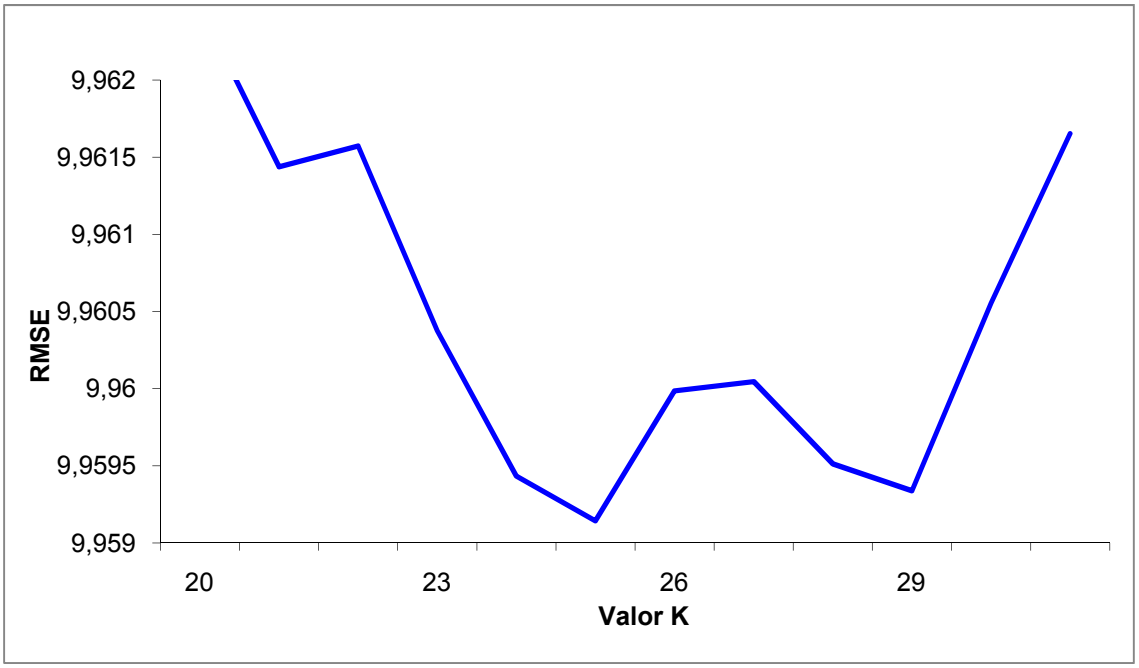

Figura 8: Obtenção do menor RMSE, kNN sem padronizar.

O menor erro obtido é 9,96 para o valor k=25. Este erro é melhor do que o valor obtido com a média da música, mas pior do que o valor obtido com a média do usuário.

- **kNN com a padronização das frequências**: A Tabela 5 e as Figuras 9 e 10 apresentam os resultados obtidos com este método.

K	RMSE 1
1	12,278099
2	10,575425
3	10,035293
4	9,794621
5	9,760661
6	9,726558
7	9,717081
8	9,74484
9	9,745129
10	9,736197
11	9,754358
12	9,758627
13	9,766373
14	9,7676525

Tabela 5: Resultados kNN padronizado.

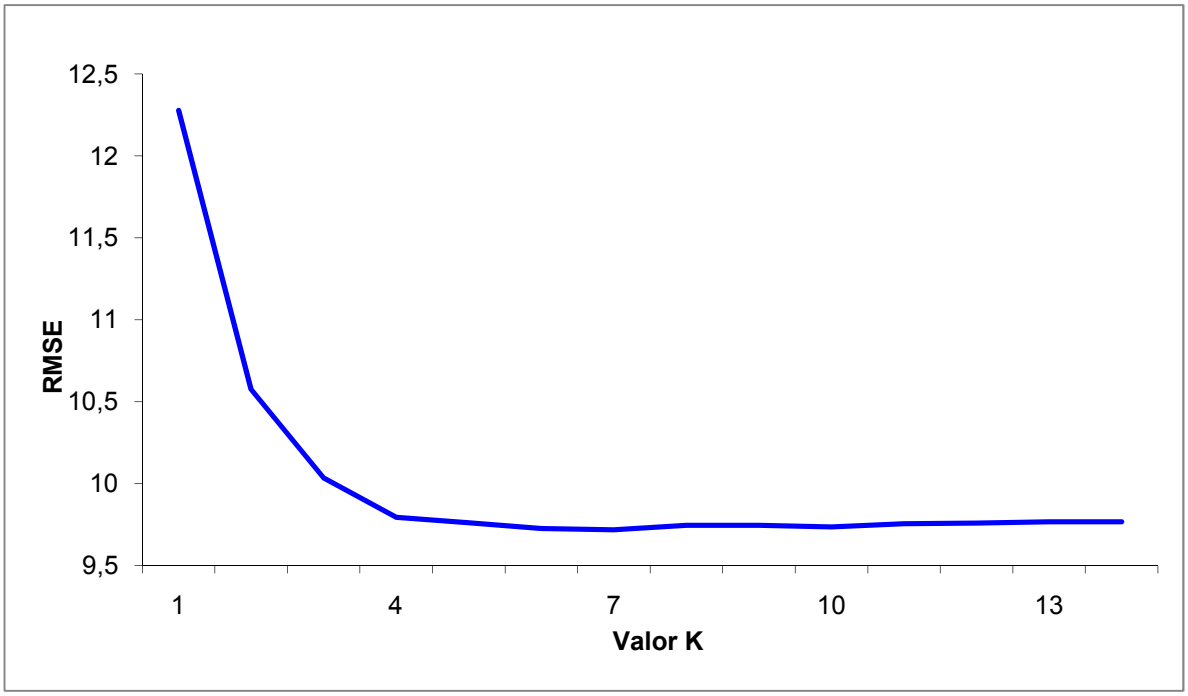

Figura 9: Evolução do RMSE kNN padronizado.

Figura 10: Obtenção do menor RMSE, kNN padronizado.

Neste caso o menor RMSE obtido é 9,72 para k=7. Este erro é menor do que o erro obtido sem a padronização dos dados. Além disso, essa taxa de erro menor é obtida com um menor número de vizinhos (o que permite uma computação mais rápida).
Como o erro obtido ainda é pior do que o erro da média do usuário, investigamos várias estratégias para melhorar os resultados do kNN. As seguintes subseções apresentam tais estratégias.

4.3.2. Remoção de predições inviáveis do kNN padronizado:

Analisando as estimativas com o kNN padronizado, percebe-se que algumas das predições obtidas têm um valor inferior a 1, apesar de saber que pelo fato da presença no conjunto de teste o usuário escutou pelo menos uma vez a música em questão. Portanto, uma melhoria intuitiva viável é a de trocar estas predições inviáveis pelo valor 1. A Tabela 6 e as Figuras 11 e 12 mostram os resultados obtidos com este método em comparação com os do ponto anterior.

K	kNN	kNN com melhoria
1	12,278099	12,117245
2	10,575425	10,490648
3	10,035293	9,969337
4	9,794621	9,734439
5	9,760661	9,708577
6	9,726558	9,679554
7	9,717081	9,672088
8	9,74484	9,701634
9	9,745129	9,703709
10	9,736197	9,696103
11	9,754358	9,71495
12	9,758627	9,719463
13	9,766373	9,727581
14	9,7676525	9,728815

Tabela 6: Resultados com remoção de predições inviáveis.

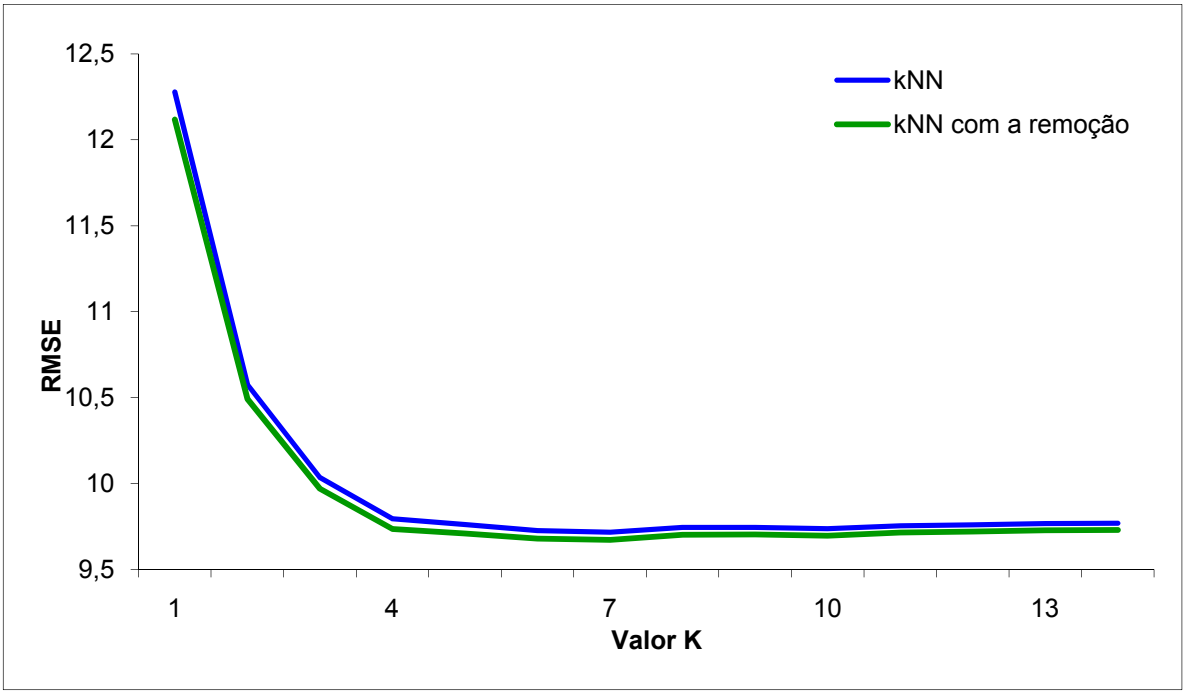

Figura 11: Evolução do RMSE com a remoção de predições inviáveis.

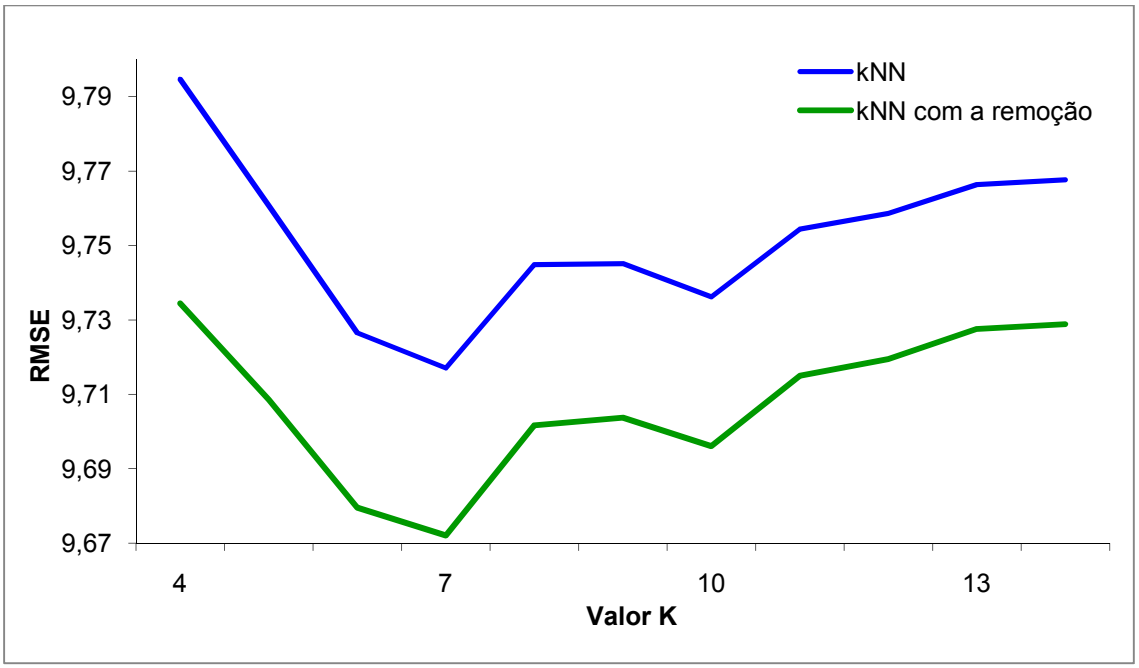

Figura 12: Obtenção do menor RMSE, remoção de predições inviáveis.

Como esperado, o fato de trocar por 1 os valores estimados com um valor inferior a 1 causa uma redução do erro. Por conseguinte, a remoção de predições inviáveis é feita para todas as estimações, antes do cálculo do RMSE, a partir deste ponto.

4.3.3. Tratamento das músicas que estão apenas no teste:

Em outras seções já foi mencionado que 22,34% das músicas aparecem apenas no conjunto de teste (estão ausentes no conjunto de treinamento). A estimação do kNN para estas músicas atribui o valor da média do usuário. O que acontece se tentarmos estimar essas músicas com outros valores, tais como números inteiros ou a média de reproduções no conjunto de teste para essa música? A Tabela 7 e as Figuras 13 e 14 apresentam os resultados dessa estratégia.

	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
K	kNN	PREDIÇÃO AUSENTES=1	PREDIÇÃO AUSENTES=2	PREDIÇÃO AUSENTES=3	PREDIÇÃO AUSENTES=4	PREDIÇÃO AUSENTES= MÉDIA
1	12,117245	12,190165	12,154776	12,143162	12,144409	12,140823
2	10,490648	10,575253	10,5348215	10,520939	10,522834	10,519353
3	9,969337	10,058525	10,0166	10,001438	10,003962	10,001571
4	9,734439	9,826157	9,783264	9,767593	9,770262	9,768
5	9,708577	9,80053	9,757531	9,74181	9,744493	9,742212

K	RMSE kNN	RMSE PREDIÇÃO AUSENTES=1	RMSE PREDIÇÃO AUSENTES=2	RMSE PREDIÇÃO AUSENTES=3	RMSE PREDIÇÃO AUSENTES=4	RMSE PREDIÇÃO AUSENTES= MÉDIA
6	9,679554	9,771785	9,728695	9,712934	9,715667	9,713419
7	9,672088	9,764382	9,721268	9,705496	9,708234	9,70599
8	9,701634	9,793654	9,750651	9,734914	9,737604	9,735336
9	9,703709	9,795751	9,752797	9,737011	9,739755	9,737431
10	9,696103	9,788193	9,7452	9,729394	9,732143	9,729828
11	9,71495	9,806816	9,763941	9,748165	9,750907	9,748595
12	9,719463	9,81122	9,768336	9,7526045	9,75532	9,753009
13	9,727581	9,8192625	9,776402	9,760712	9,763412	9,761106
14	9,728815	9,82049	9,777615	9,761934	9,764636	9,762327

Tabela 7: Resultados com valores para músicas ausentes.

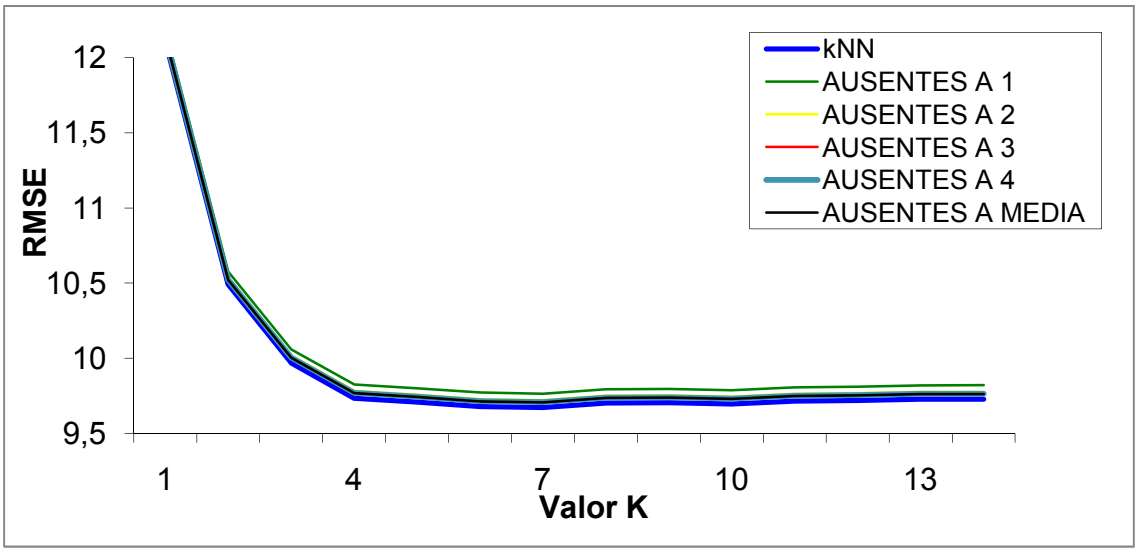

Figura 13: Evolução do RMSE com valores para ausentes.

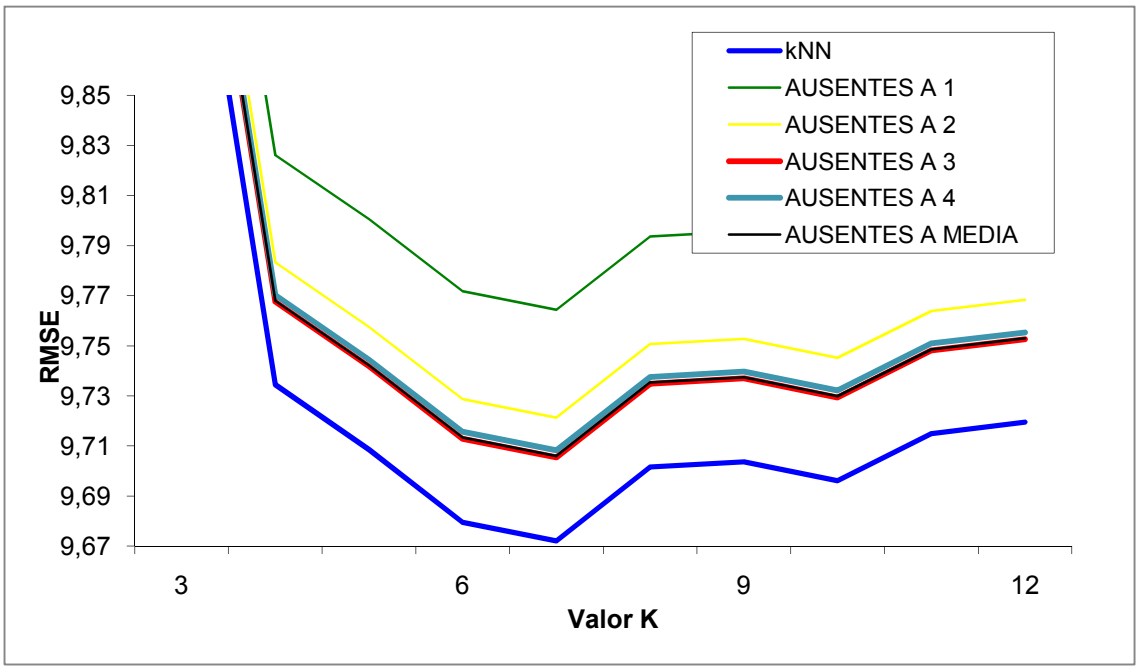

Figura 14: Obtenção do menor RMSE, valores para ausentes.

Pode-se observar que os melhores resultados com este método são os obtidos com a média das músicas ausentes e com os números inteiros mais próximos à média. O pior resultado é obtido com o número mais distante da média, 1. Entanto nenhum dos resultados consegue melhorar os resultados obtidos com o kNN, reafirmando que é um método robusto.

4.3.4. Usar somente vizinhos realmente próximos:

Devido ao fato de que muitas músicas aparecem poucas vezes, algumas vezes os "k vizinhos mais próximos" não são realmente próximos. Neste ponto vão ser usados para o cálculo das estimações somente os usuários com uma similaridade superior a um limiar (chamado de *minsim* neste trabalho). A Tabela 8 e as Figuras 15 e 16 apresentam os resultados dessa estratégia.

	RMSE	RMSE	RMSE	RMSE
K	kNN	minsim = 0,3	minsim = 0,5	minsim = 0,7
1	12,117245	12,114202	11,999105	10,384004
2	10,490648	10,487579	10,477269	9,923125
3	9,969337	9,966372	10,000841	9,756043
4	9,734439	9,738708	9,780458	9,679638
5	9,708577	9,711088	9,742547	9,712373

	RMSE	RMSE	RMSE	RMSE
K	kNN	minsim = 0,3	minsim = 0,5	minsim = 0,7
6	9,679554	9,682041	9,716833	9,729442
7	9,672088	9,674342	9,713502	9,732422
8	9,701634	9,703976	9,742972	9,732923
9	9,703709	9,7060175	9,746863	9,731805
10	9,696103	9,698401	9,740661	9,732173
11	9,71495	9,717113	9,759267	9,733617
12	9,719463	9,721652	9,761802	9,733154

Tabela 8: Resultados com valores para minsim.

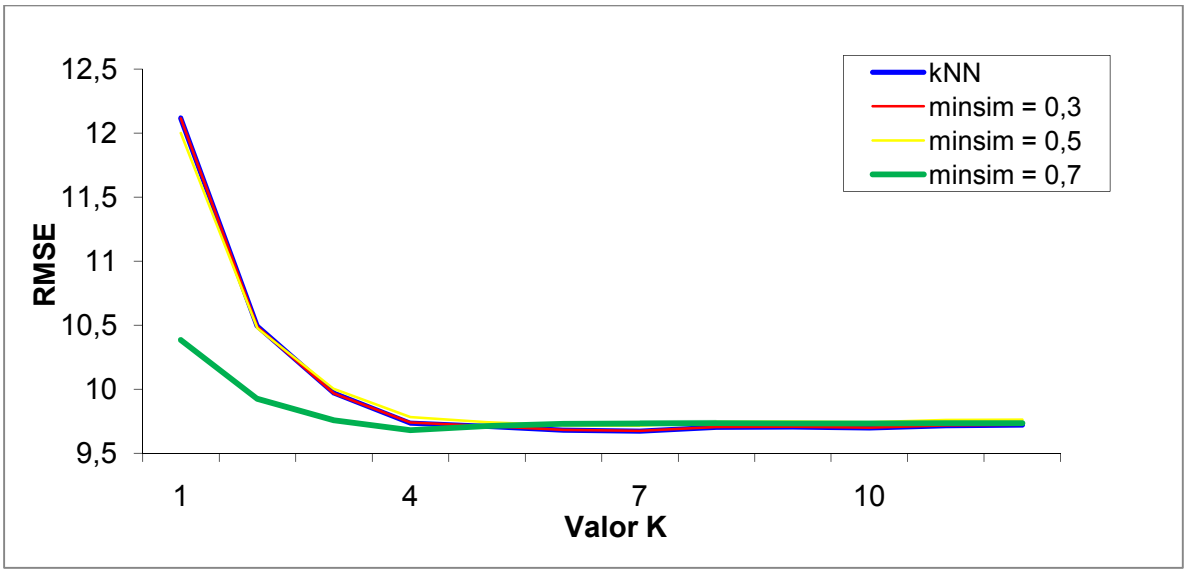

Figura 15: Evolução do RMSE com valores para minsim.

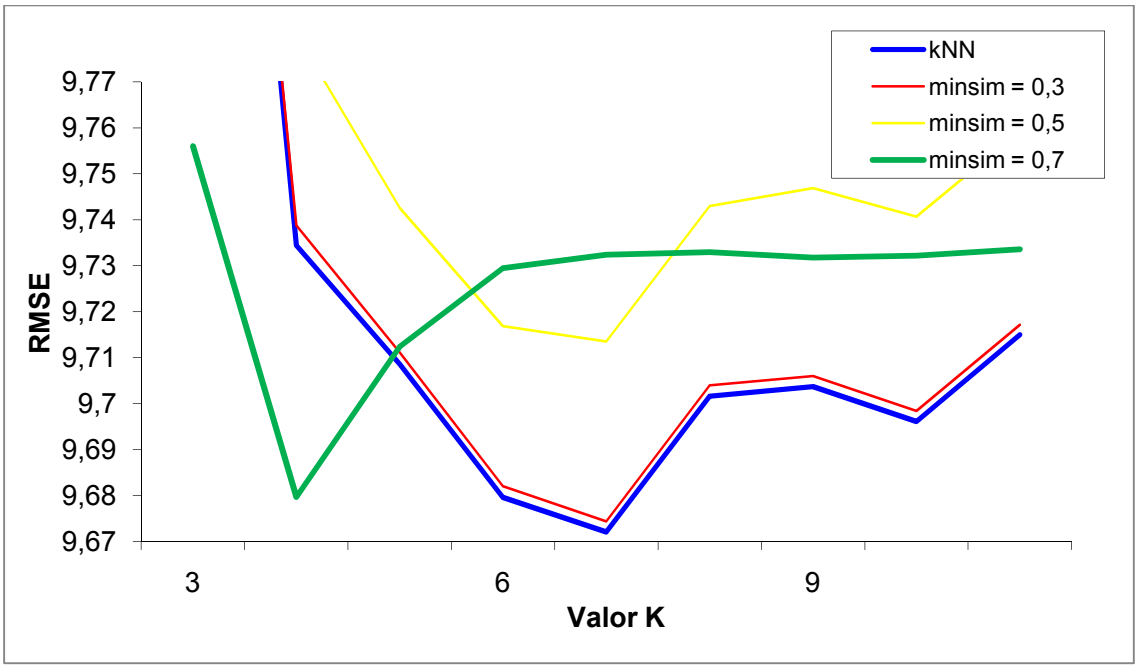

Figura 16: Obtenção do menor RMSE, valores para minsim.

Dos resultados obtidos podem ser tiradas várias conclusões:

- O kNN é um bom método, continua fornecendo os melhores resultados.
- Dos resultados para minsim=0,3 se observa que o kNN já contempla dar pouco peso as frequências dos usuários pouco semelhantes (os resultados são quase os mesmos que para o kNN).
- Usando apenas os usuários mais semelhantes (minsim=0,7) se obtém estimações piores do que com o kNN, mais o menor RMSE é obtido precisando menos vezinhos (permitindo uma computação mais rápida).

4.3.5. Usar somente vizinhos confirmados:

Algumas vezes, dois usuários que escutaram muitas músicas têm apenas uma música (ou poucas músicas) em comum no conjunto de treino, mais eles a ouviram quase o mesmo número de vezes (por exemplo apenas uma vez). O resultado obtido do cálculo da similaridade para estes pares de usuários é elevado (são então para o sistema usuários semelhantes), quando eles não são realmente semelhantes.

Neste ponto são usados pelo cálculo das frequências apenas os usuários para os quais a similaridade é calculada com um número de músicas em comum superior ou

igual a um limiar (chamado de *mincom*). A Tabela 9 e as Figuras 17 e 18 apresentam os resultados obtidos.

	RMSE	RMSE	RMSE	RMSE
K	kNN	mincom=2	mincom=3	mincom=5
1	12,117245	11,920053	11,8941765	11,89945
2	10,490648	10,278	10,278483	10,27894
3	9,969337	9,851222	9,85902	9,855251
4	9,734439	9,733735	9,745398	9,745851
5	9,708577	9,681791	9,690395	9,704775
6	9,679554	9,663812	9,665848	9,731963
7	9,672088	9,6744	9,675901	9,7448
8	9,701634	9,678599	9,682686	9,742279
9	9,703709	9,716428	9,718501	9,745405
10	9,696103	9,724986	9,724782	9,739402
11	9,71495	9,727715	9,726688	9,742791
12	9,719463	9,728662	9,728097	9,742064
13	9,727581	9,72793	9,728762	9,740671
14	9,728815	9,732	9,731431	9,750256

Tabela 9: Resultados com valores para mincom.

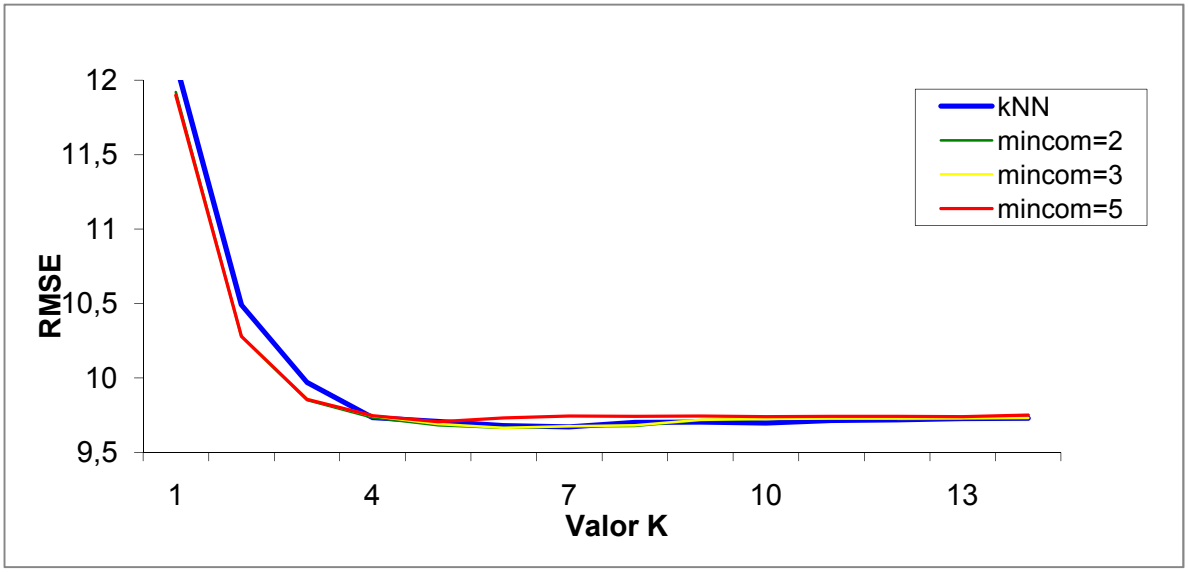

Figura 17: Evolução do RMSE com valores para minsim.

Figura 18: Obtenção do menor RMSE, valores para minsim.

Finalmente os resultados obtidos são um pouco melhores que os que foram obtidos com o kNN, e precisando dum menor número de vizinhos (um a menos), para os valores mincom=2 e mincom=3. Para valores maiores (mincom=5) precisam-se ainda menos vizinhos, mais o menor RMSE é pior. No entanto, nenhuma das estimativas obtidas nesta seção é melhor do que a média do usuário.

4.3.6. Resultados por categorias:

Neste ponto são avaliados os resultados obtidos pelo melhor kNN (k=7, RMSE=9,67) em função de duas variáveis:

- **Dependendo do número de músicas ouvidas pelo usuário**: A Tabela 10 apresenta o RMSE do conjunto de teste calculado de várias maneiras distintas. Se o usuário ouviu mais do que LIMIT músicas os resultados são usados para o cálculo de RMSE_SUP (erro dos usuários que escutaram mais do que LIMIT músicas), no caso contrario para RMSE_INF. O número de estimações (células da matriz usuários-músicas) usadas para cada caso é respectivamente COUNT_INF e COUNT_SUP. Estes valores informam do peso dos RMSE obtidos.

LIMIT	COUNT_SUP	RMSE_SUP	COUNT_INF	RMSE_INF
1.000	450.419	9,390503	10.980	17,723742
5.000	298.634	8,569188	162.765	11,425837
10.000	151.043	5,995204	310.356	11,0286665

Tabela 10: Resultados dependendo do número de músicas.

Os resultados obtidos são melhores enquanto os usuários escutam mais músicas. Para os usuários com poucas músicas (menos de 1.000) o RMSE obtido é o maior deste trabalho (acima de 17). Para usuários com muitas músicas (mais de 10.000) o RMSE obtido é o menor (abaixo de 6). Estes resultados mostram outra das qualidades do kNN como sistema de recomendação. Para o kNN, quanto mais informações do usuário estão disponíveis, melhor vão ser as recomendações que lhe serão feitas.

- Dependendo do número de usuários que escutaram a música: A Tabela 11 também apresenta o RMSE do conjunto de teste calculado de várias maneiras distintas. Se a música foi escutada por mais de LIMIT usuários os dados são usados para calcular o RMSE_SUP, no caso contrario para RMSE_INF. Da mesma forma que no ponto anterior, COUNT_INF e COUNT_SUP informam do peso.

LIMIT	COUNT_SUP	RMSE_SUP	COUNT_INF	RMSE_INF
1	317.028	10,262812	144.371	8,233643
3	269.200	10,450122	192.199	8,467859
5	239.013	10,482611	222.386	8,721709
7	216.610	10,719827	244.789	8,643338
9	198.948	10,669483	262.451	8,8444395
11	183.935	10,856938	277.464	8,801997
13	171.347	10,973659	290.052	8,816021
15	160.331	10,922813	301.068	8,937312
17	150.386	11,039071	311.013	8,938589
19	141.557	11,136441	319.842	8,949797
21	133.757	11,179282	327.642	8,986346
23	126.431	11,27647	334.968	8,994251
25	119.630	11,418448	341.769	8,982703

Tabela 11: Resultados dependendo do número de usuários.

Os resultados são contrários ao que era intuitivamente esperado: as músicas que foram ouvidas por muitos usuários são as que têm piores predições, enquanto as músicas que foram escutadas por poucos usuários levam a resultados melhores. Isto

pode ser causado pelo fato do ponto 4.3.5 (falsos vizinhos): as músicas que foram ouvidas por poucas pessoas foram realmente escutadas por usuários semelhantes que têm coisas em comum, em consequência as predições são melhores para estas músicas.

4.4. Métodos híbridos:

Neste ponto são combinados os resultados obtidos com distintos métodos conforme explicado na seção 3.3.3. As predições do melhor kNN (com valor k=7) são usadas para fazer as novas estimações, com as predições obtidas por algumas médias.

4.4.1. kNN e Média do usuário:

As frequências obtidas com o kNN são combinadas com as obtidas com a média do usuário da seguinte forma:

$$\widehat{freq_{est}} = \widehat{freq_{kNN7}} . w + \widehat{freq_{mUSER}} . (1 - w)$$

A Tabela 12 e a Figura 19 apresentam os resultados desse método.

w	RMSE
0	9,586825
0,1	9,484578
0,2	9,407125
0,3	9,407125
0,4	9,329675
0,5	9,328634
0,6	9,351531
0,7	9,398477
0,8	9,468384
0,9	9,56004
1	9,672088

Tabela 12: Resultados kNN com a média do usuário.

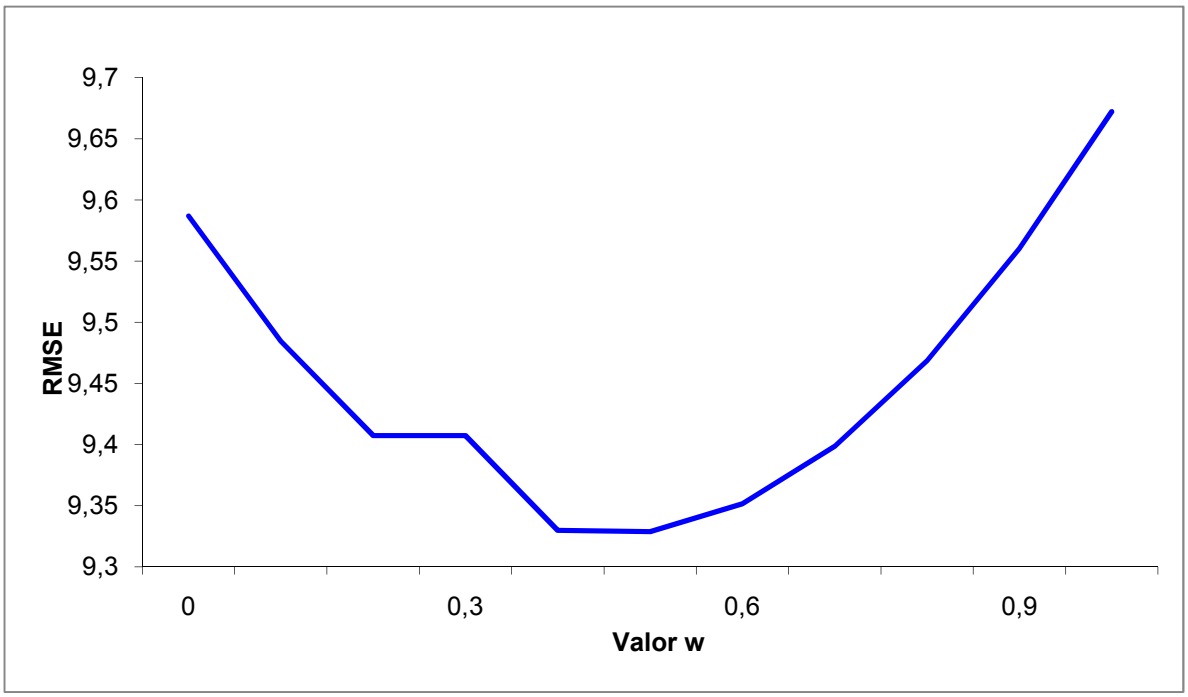

Figura 19: Evolução do RMSE do kNN com a média do usuário.

Os valores extremos são os do kNN (na direita da Figura 19) e os da média do usuário (esquerda). A evolução no gráfico mostra como os resultados obtidos pelo método híbrido combinando o melhor kNN com a melhor média são melhores do que qualquer um dos dois separadamente, obtendo desta maneira os melhores resultados neste trabalho (RMSE=9,33) para w=0,5.

4.4.2. Combinação de três métodos:

Neste experimento, são combinados três métodos: o kNN, a média do usuário e a média da música. Devido ao fato que muitas músicas aparecem apenas no conjunto de teste (e assim não existe para as mesmas uma estimativa de sua média) as predições deste ponto são ser calculadas da seguinte maneira:

$$\widehat{freq_{est}} = \widehat{freq_{kNN7}}.w_{kNN7} + \widehat{freq_{mUSER}}.w_{mUSER} + freq_{mMUSICA}.(\widehat{1 - w_{kNN7}} - w_{mUSER})$$

Se a música se encontra no teste, no caso contrário:

$$\widehat{freq_{est}} = \frac{\widehat{freq_{kNN7}}.w_{kNN7}}{w_{kNN7} + w_{mUSER}} + \frac{\widehat{freq_{mUSER}}.w_{mUSER}}{w_{kNN7} + w_{mUSER}}$$

Para manter os mesmos percentuais.

A Tabela 13 e a Figura 20 apresentam os resultados obtidos usando este método.

		w_{kNN7}			
		0,3	0,4	0,5	0,6
w_{mUSER}	0	NC	NC	NC	9,617021
	0,1	NC	NC	9,575689	9,521787
	0,2	NC	9,555698	9,48371	9,444826
	0,3	9,55831	9,468186	9,411162	9,387904
	0,4	9,476427	9,401143	9,359374	9,351531
	0,5	9,41476	9,354957	9,328634	NE
	0,6	9,37447	9,329675	NE	NE
	0,7	9,355571	NE	NE	NE

Tabela 13: Resultados com a combinação de três métodos.

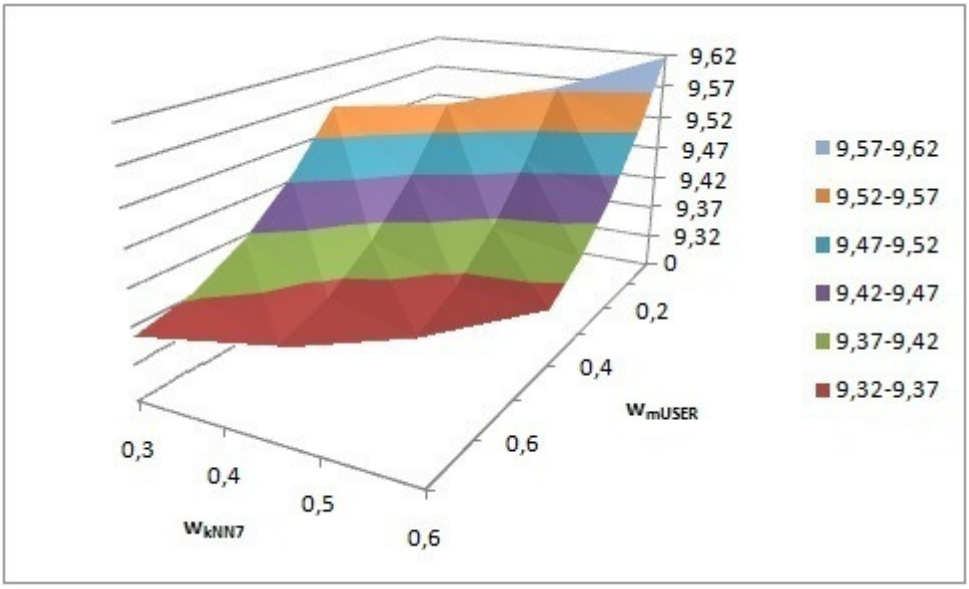

Figura 20: Evolução do RMSE com a combinação de três métodos.

Na Tabela 13 os resultados marcados como NC não foram calculados porque o erro já era grande. Os resultados marcados como NE não existem (o peso obtido para a frequência com a média das músicas seria negativo nesses casos). Pode-se observar que os melhores resultados são obtidos no caso limite em que não é usada a média das

músicas (inferior do gráfico), obtendo em consequência o mesmo RMSE que no ponto anterior.

Uma conclusão errada seria dizer que usar um sistema híbrido baseado em três métodos é equivalente a um sistema híbrido baseado em dois métodos. A conclusão certa neste caso é que (como já foi comentado em outros pontos) a média das músicas não é uma boa estimativa para o conjunto de dados usado neste trabalho, em consequência é melhor não usar esta estimativa num sistema híbrido.

5. CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho propõe uma abordagem colaborativa para resolver o problema da recomendação musical. Os experimentos realizados focam no uso do algoritmo kNN para predizer a frequência com a qual um determinado usuário escutará uma dada música. Os resultados da predição com o kNN foram comparados com os resultados da predição usando médias dos usuários e das músicas. O resultado do kNN foi pior do que a média dos usuários, o que indica que essa estratégia, pelo menos nos formato em que foi testada, não é a mais adequada para o conjunto de dados em questão. Algumas variações para tentar obter melhorias nos resultados foram propostas. Porém, o melhor resultado foi obtido com o uso de uma estratégia híbrida, que usa kNN e média dos usuários.

Uma das características do conjunto de dados que foi responsável pelo insucesso do kNN é a presença de poucos usuários. Muitas músicas foram escutadas por apenas um usuário. Além disso, muitos usuários não têm nenhum usuário realmente semelhante nos dados utilizados (ou têm usuários que *não são semelhantes* mas são identificados como *semelhantes*). Em consequência, a primeira opção para melhorar este trabalho seria aumentar o número de usuários utilizados nos experimentos.

Outros experimentos que poderiam ser feitos no futuro para a melhoria das conclusões obtidas são a criação de novos subconjuntos de dados (teste e treino) com os mesmos dados utilizados para a confirmação das conclusões obtidas, a utilização de outros métodos para fazer as predições, tais como o *Singular Value Decomposition* (SVD) ou a utilização de outros métodos para a avaliação dos resultados obtidos tais como o *Normalized RMSE* (NRMSE).

Finalmente, a última etapa da recomendação não foi feita neste trabalho. Tal etapa consiste na criação de uma lista de recomendações para cada usuário. Em seguida, deveria ser realizada uma avaliação da lista fornecida (com alguma métrica que o permita), e uma interface gráfica para apresentar as listas criadas aos usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

ROBERT M. BELL e YEHUDA KOREN. Improved Neighborhood-based Collaborative Filtering. 2007.

ROBIN BURKE. California State University, Fullerton. Hybrid Recommender Systems: Survey and Experiments. 2002. Em User Modeling and User-Adapted Interaction. v. 12, n. 4. p.331-370.

ÒSCAR CELMA HERRADA. Last.fm Dataset - 1K users. 2010. http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html. Acessado em 01/03/2011.

ÒSCAR CELMA HERRADA. Music Recommendation Datasets for Research. 2009. http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/. Acessado em 01/03/2011.

ÒSCAR CELMA HERRADA. Tese dectoral UPF. Music Recommendation and Discovery in the Long Tail. 2008.

MACARENA ESPINILLA, IVÁN PALOMARES, ROSA MARÍA RODRÍGUEZ e LUIS MARTÍNEZ. Universidad de Jaén. Departamento de Informática. OL-RadioUJA. Radio Colaborativa bajo Licencia Creative Commons. Em I Jornadas Andaluzas de Informática, JAI2009. Jaén. 2009.

JONATHAN L. HERLOCKER, JOSEPH A. KONSTAN, LOREN G. TEREVEEN e JOHN T. RIEDL. Oregon State University e University of Minnesota. Evaluating Collaborative Filtering Recommender Systems. Janeiro 2004. p.8-12.

TED HONG e DIMITRIS TSAMIS. Stanford University. Use of KNN for the Netflix Prize. 2006. http://www.stanford.edu/class/cs229/proj2006/HongTsamis-KNNForNetflix.pdf.

BRUCE KRULWICH. LifeStyle Finder, Intelligent User Profiling Using Large-Scale Demographic Data. Em AI Magazine Volume 18 Number 2. 1997.

Last.fm. API Methods. http://www.lastfm.es/api. Acessado em 01/03/2011.

Last.fm. user.getRecentTracks. http://www.lastfm.es/api/show?service=278. Acessado em 01/03/2011.

BYRON LEITE DATNAS BEZERRA. Universidade Federal de Pernambuco. Centro de Informática. Estudo de Algoritmos de Filtragem de Informação Baseados em Conteúdo. Em Trabalho de Graduação em Inteligência Artificial. Recife. 2002.

LESTER MACKEY. Collaborative Filtering. Practical Machine Learning, CS 294-34. Based on slides by Aleksandr Simma. 2009.

QING LI, SUNG HYON MYAENG, DONG HAI GUAN e BYEONG MAN KIM. A Probabilistic Model for Music Recommendation Considering Audio Features. 2005.

GREG LINDEN. Google News Personalization paper. 2007. http://glinden.blogspot.com/2007/05/google-news-personalization-paper.html.

KEVIN MCCARTHY, MARIA SALAMÓ, LORCAN COYLE, LORRAINE MCGINTY, BARRY SMYTH e PADDY NIXON. Group recommender systems: a critiquing based approach. Em Proceedings of the 11th international conference on Intelligent User Interfaces. New York, NY, USA. 2006. p.267–269.

BAMSHAD MOBASHER, ROBERT COOLEY e JAIDEEP SRIVASTAVA. Automatic Personalization Based on Web Usage Mining. 2000. Em Communications of the ACM. v. 43, n. 8. p.142-151.

The Netflix Prize Rules. 2006. http://www.netflixprize.com//rules. Acessado em 14/01/2011.

DOUGLAS W. OARD e JINMOOK KIM. University of Maryland. Implicit Feedback for Recommender Systems. Em AAAI Workshop on Recommender Systems. Madison, WI. 1998. p.81-83.

About Pandora. http://www.pandora.com/corporate/index.shtml. Acessado em 24/02/2011.

MICHAEL J. PAZZANI. Department of Information and Computer Science. University of California, Irvine. A Framework for Collaborative, Content-Based and Demographic Filtering. Irvine, CA, USA. 1999.

BADRUL SARWAR, GEORGE KARYPIS, JOSEPH KONSTAN, e JOHN RIEDL. Department of Computer Science and Engineering. University of Minnesota, Minneapolis. Item-based Collaborative Filtering Recommendation Algorithms. 2001. Em WWW10 Conference. p.3-4.