

Improvise

Criação de playlists tendo em conta a actividade do utilizador

Ricardo Antunes Lopes da Cunha

Dissertação para a obtenção de Grau de Mestre em
Engenharia Informática e de Computadores

Orientador: Prof. Manuel João Caneira Monteiro da Fonseca

Júri

Presidente: Prof. Nuno João Neves Mamede

Orientador: Prof. Manuel João Caneira Monteiro da Fonseca

Vogal: Prof. Bruno Emanuel da Graça Martins

Mai 2014

Agradecimentos

Esta dissertação representa o fim de uma parte importante da minha vida, no entanto o trabalho realizado não teria sido possível sem o apoio dos meus familiares e amigos. Desta forma quero agradecer em especial à minha mãe e ao meu pai, cujo apoio e a ajuda fornecida durante a realização desta dissertação foi muito importante.

Também gostaria de agradecer a todas as pessoas que dispensaram do seu tempo para me ajudar durante os testes realizados para testar a solução aqui apresentada. Destas pessoas gostaria de agradecer em especial aos meus amigos e familiares Ana Cláudia, Diogo Henriques, Duarte Barbosa, Inês Pereira, João Costa, Miguel Cunha, Miguel Torres, Pedro Borges, Ricardo Sancho e Rui Ribeiro.

Ao Prof. Manuel João Caneira Monteiro da Fonseca e ao Ricardo Dias que me orientaram durante a realização desta tese, quero deixar um especial agradecimento por todo o apoio e ajuda que me deram durante este longo processo, sem eles o trabalho apresentado neste documento não teria sido possível.

Resumo

O número de músicas disponíveis na Internet tem crescido continuamente desde o aparecimento da mesma. Como resultado deste crescimento, começou a ser cada vez mais difícil para os utilizadores encontrarem músicas adequadas às suas necessidades.

De forma a permitir aos utilizadores ouvirem músicas que gostam, dois tipos de técnicas são bastante usadas, a recomendação e a procura. Enquanto que os sistemas de recomendação procuram mais activamente pelas necessidades dos utilizadores, os sistemas de procura recebem as necessidades passivamente. Isto é, enquanto que os sistemas de pesquisa perguntam aos utilizadores para submeterem a informação que necessitam em forma de interrogação, os sistemas de recomendação usam a informação que têm sobre o utilizador para recomendar novos itens.

No nosso trabalho apresentamos uma solução de recomendação musical, cujo objectivo é o de gerar e recomendar músicas que estejam em sintonia com a actividade que o utilizador está a realizar.

Para isso, desenvolvemos uma abordagem híbrida que reflecte a integração de um sistema de recomendação baseado no contexto e no conteúdo. Em relação ao contexto, é utilizada a informação sobre a actividade que o utilizador está a realizar, quanto ao conteúdo é tida em conta a informação sobre as características musicais acústicas de cada música.

O algoritmo de recomendação, no início, começa por sugerir músicas iguais para todos os utilizadores, na medida em que usa um modelo genérico. Depois, e à medida que o utilizador vai interagindo com o sistema, o modelo genérico vai-se ajustando aos gostos do utilizador, e torna-se num modelo personalizado. Após algumas iterações com o sistema, cada utilizador terá o seu próprio modelo de recomendação, que reflecte de melhor forma os seus gostos.

Abstract

The number of songs available in Internet has grown steadily since its appearance. As a result of this growth, it is extremely difficult for users to find the appropriate music that suit their needs. To allow users to listen to the music they like, two techniques are quite used, searching and recommendation. While recommender systems rely more actively on the users needs, the search engines receive their needs more passively.

In the present work we developed a solution for music recommendation, whose objective is to search and recommend songs in tune with the activity that the user is performing.

With this objective in mind, we developed a hybrid approach reflecting the integration of a recommendation system based on context and content. Regarding context, we use information about the user's activity. In what concerns the content, we took into account the information about the acoustic features of each song.

The recommendation algorithm starts by recommending the same songs for all users, we call this the generic model. The user's interaction with the system leads to the adjustment of the generic model, to its own musical preferences changing it into a personalized model. A personalized recommendation model for each user will then be obtained after some interactions with the system, which will better reflect their tastes.

Palavras Chave

Palavras Chave

Sistema de recomendação musical
Recomendação híbrida
Recomendação baseada no contexto
Actividade do utilizador
Associação de músicas a actividades
Modelo genérico e personalizado

Keywords

Music recommendation system
Hybrid recommendation
Context-based recommendation
User activity
Association between songs and activities
Generic and personalized model

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objectivos	1
1.3	Solução desenvolvida	2
1.4	Contribuições e Resultados Atingidos	3
1.5	Organização da Dissertação	3
2	Trabalho Relacionado	4
2.1	Tipos de Recomendação	4
2.1.1	Recomendação usando filtros colaborativos	4
2.1.2	Recomendação baseada no conteúdo	6
2.1.3	Recomendação usando filtros demográficos	7
2.1.4	Recomendação baseada no contexto	7
2.1.5	Abordagens de recomendação híbridas	8
2.1.6	Discussão	8
2.2	Trabalhos de recomendação de música	9
2.2.1	Context-Aware Mobile Music Recommendation for Daily Activities	9
2.2.2	Nextone Player: A Music Recommendation System Based On User Behavior	11
2.2.3	Lifetrak: Music In Tune With Your Life	12
2.2.4	Foxtrot: A Soundtrack for Where You Are	14
2.2.5	Music Playlist Recommendation Based on User Heartbeat and Music Preference	15
2.2.6	InCarMusic: Context-Aware Music Recommendations in a Car . . .	16
2.2.7	Context-Aware Recommendation by Aggregating User Context . . .	17
2.2.8	Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns	18
2.2.9	Music for My Mood: A Music Recommendation System Based on Context Reasoning	19
2.2.10	Content-based music recommendation based on user preference examples	21

2.2.11	Discussão	22
3	Associação de músicas a actividades	25
3.1	Procedimento	25
3.2	Restrições	26
3.3	Aplicação de recolha de dados	27
3.3.1	Fluxo da aplicação	27
3.3.2	Arquitectura	29
3.4	Síntese	31
4	Modelo de recomendação	32
4.1	Recolha de dados	32
4.2	Seleccção de características	33
4.2.1	Características disponíveis	33
4.2.2	Seleccção de características relevantes	34
4.2.3	Qualidade das características relevantes	36
4.2.4	Refinamento das actividades	37
4.3	Modelo genérico de recomendação	38
4.3.1	Intervalos das características	38
4.3.2	Determinação dos intervalos das características	39
4.4	Síntese	43
5	Avaliação Experimental	45
5.1	Aplicação de avaliação	45
5.2	Procedimento de avaliação	46
5.3	Resultados da avaliação do modelo genérico	47
5.4	Resultados da avaliação do modelo personalizado	50
5.5	Síntese	53
6	Conclusões e Trabalho Futuro	55
6.1	Sumário da Dissertação	55
6.2	Conclusões Finais e Contribuições	56
6.3	Trabalho Futuro	57

Lista de Figuras

2.1	Matriz típica usada nos filtro colaborativos.	5
3.1	Representação da recolha de músicas para a criação do modelo genérico. . .	26
3.2	Seleccção de géneros.	28
3.3	Seleccção de artistas.	28
3.4	Seleccção de músicas.	29
4.1	Resultados dos boxplots	35
4.2	Total de data sets calculados.	40
4.3	Testes efectuados para determinar o melhor método de calculo dos intervalos.	41
4.4	Resultados da determinação dos intervalos das características	42
4.5	Método para o cálculo dos intervalos seleccionado.	43
5.1	Seleccção da actividade.	46
5.2	Seleccção de músicas.	46
5.3	Média de músicas escolhidas por actividade.	48
5.4	Resultados online da satisfação da recomendação.	49
5.5	Satisfação da recomendação do modelo genérico presencial.	49
5.6	Percentagem do feedback positivo, neutro e negativo da recomendação do modelo genérico.	50
5.7	Média de músicas escolhidas por actividade.	51
5.8	Evolução do número de músicas escolhidas utilizando o modelo personalizado.	52
5.9	Evolução do número de músicas escolhidas utilizando o modelo personalizado.	52
5.10	Satisfação da recomendação do modelo personalizado dos 10 utilizadores. .	53
6.1	Novo cálculo de intervalos.	57

Lista de Tabelas

2.1	Comparação entre artigos de recomendação de música	22
4.1	Matriz de confusão.	36
4.2	Matriz de confusão com 5 actividades.	37
4.3	Matriz de confusão com média $\pm 15\%$ desvio padrão usando o classificador Random Forest.	42
4.4	Matriz de confusão com mediana $\pm 20\%$ desvio padrão usando o classifica- dor Random Forest.	43

Capítulo 1

Introdução

1.1 Motivação

Hoje em dia, temos cada vez mais recursos à disposição para o consumo de música em formato digital, quer através de lojas online para compra de músicas, quer de serviços de *streaming*. Assim, temos à nossa disposição milhares de músicas que podem ser ouvidas em qualquer lugar e a qualquer hora, em diversos dispositivos (ex. leitores MP3, telemóveis, *tablets* etc.).

No entanto, com o acesso a uma colecção de músicas tão vasta, o principal problema que surge é o de escolher que músicas pretendemos ouvir. Este problema é ainda agravado quando queremos criar uma lista de músicas para ouvir durante a realização de uma determinada tarefa, como por exemplo, exercício físico, condução ou estudo. Assim, a criação destas listas de músicas torna-se um processo moroso e frustrante para os utilizadores.

Para resolver este problema nos últimos anos têm surgido várias propostas para a criação e geração de *playlists*. Neste documento descrevemos e analisamos as principais técnicas usadas, das quais destacamos as técnicas de recomendação híbridas visto conseguirem colmatar alguns dos problemas presentes nas outras técnicas ao combinar algumas das diferentes abordagens.

Apesar dos bons resultados que se conseguem obter com este tipo de técnicas, elas funcionam bem num contexto genérico, não se adaptando à actividade específica que o utilizador está a realizar. De facto, este é um tópico pouco abordado não existindo ainda uma solução eficaz para esse tipo de recomendação.

1.2 Objectivos

A quantidade enorme de músicas que podemos escolher para ouvir como podemos comprovar cada vez que pesquisamos um catálogo de músicas não personalizado, como é o caso do Myspace ou iTunes, por vezes deixa-nos paralisados. Esta situação em que fica-

mos paralisados com dúvidas quando enfrentamos uma quantidade enorme de escolhas foi apelidado por Schwartz[1] como o "paradoxo da escolha". Existe assim a necessidade de eliminar algumas dessas escolhas, o que pode ser conseguido fornecendo filtros personalizados ou fazendo recomendações para facilitar as decisões dos utilizadores.

Os sistemas de recomendação são ferramentas de *software* compostas por um conjunto de técnicas que fornecem sugestões de itens a serem consumidos pelos utilizadores. O uso do contexto nestes sistemas é um tópico de pesquisa que tem crescido consideravelmente nos últimos anos. A solução aqui apresentada pretende explorar este tópico, tendo como objectivo principal recomendar músicas de acordo com a actividade que cada pessoa está a realizar.

Com este problema em vista pretendemos conceber uma solução para a recomendação de playlists de forma automática tendo em conta a actividade que o utilizador está a realizar (contexto). Para além da actividade a ser realizada pelo utilizador é nosso objectivo usar informação referente às músicas para nos ajudar no processo de recomendação, de forma a detectar um possível paralelismo entre características musicais e as diferentes actividades.

Para além de recomendar as músicas preferidas do utilizador é nosso objectivo fornecer também uma solução que permita ao utilizador conhecer novas músicas, que estejam de acordo com as suas preferências musicais. Pretendemos também que a solução desenvolvida seja capaz de ultrapassar os maiores problemas presentes nos sistemas de recomendação nomeadamente o *cold start Problem* e o *popularity bias*.

1.3 Solução desenvolvida

Tendo em conta os trabalhos desenvolvidos na área de recomendação de música, a nossa solução pode-se considerar um sistema de recomendação híbrido que tem em consideração o contexto em que a música está a ser ouvida e as características musicais de cada uma delas.

A nossa solução começa por recomendar um conjunto de músicas iguais para todos os utilizadores, com base num modelo inicial genérico. Estas músicas são recomendadas tendo em conta testes feitos a utilizadores de forma a perceber que músicas, na sua maioria, as pessoas preferem para cada actividade que realizam. As músicas são recomendadas tendo em conta um conjunto de valores para as suas características acústicas, que definem assim que músicas devem ser recomendadas para cada actividade.

Em futuras interacções dos utilizadores com o sistema, estes valores vão-se ajustando tendo para tal em conta as músicas anteriormente escolhidas por cada utilizador. Desta forma as músicas recomendadas vão estando cada vez mais alinhadas com os gostos de cada utilizador.

1.4 Contribuições e Resultados Atingidos

Com este trabalho desenvolvemos uma solução para a recomendação de música utilizando um sistema de recomendação híbrido. Na nossa solução é utilizado um modelo genérico, que é usado inicialmente por todos os utilizadores, que após várias interações dos utilizadores se transforma num modelo personalizado. Foram também detectadas algumas das características que melhor distinguem as músicas das várias actividades

Os resultados obtidos foram positivos, e no global a utilização do modelo genérico verificou-se capaz de recomendar músicas aos diferentes utilizadores. Apesar de por vezes o número de músicas consideradas como adequadas pelos utilizadores não ser muito elevado, a sua satisfação com as músicas recomendadas foi positiva.

Quando utilizado o modelo personalizado, foi possível verificar um aumento tanto no número de músicas consideradas como adequadas como também no grau de satisfação da recomendação para cada actividade.

1.5 Organização da Dissertação

O resto do documento está dividido da seguinte forma: no capítulo dois são apresentadas um conjunto de técnicas e trabalhos realizados na área da recomendação de música. De seguida no capítulo três são descritos os passos efectuados para conseguir associar as músicas às actividades, e no capítulo quatro é descrita a nossa solução.

O capítulo cinco por sua vez constituiu a análise dos dados obtidos da avaliação efectuada com utilizadores bem como uma descrição de como foram conduzidos os testes realizados. Para finalizar no capítulo seis é apresentado um sumário do documento, são tiradas conclusões referentes ao trabalho realizado e é também proposto um conjunto de experiências a seguir em trabalhos futuros.

Capítulo 2

Trabalho Relacionado

Nesta secção vamos começar por descrever os principais tipos de recomendação, tais como a recomendação baseada no conteúdo, baseada no contexto, filtros colaborativos, filtragem demográfica e soluções mistas. Depois de uma visão geral sobre os diferentes tipos de recomendação é feita uma discussão apresentando os pontos positivos e negativos de cada uma das abordagens.

Na secção 3.2 apresentamos trabalhos desenvolvidos na área de sistemas de recomendação de música. No final é apresentada uma tabela comparativa que compara cada um dos trabalhos relacionados descritos neste documento.

2.1 Tipos de Recomendação

2.1.1 Recomendação usando filtros colaborativos

Os filtros colaborativos constituem a abordagem mais comum não só para a recomendação de música mas também para outros tipos de sistemas de recomendação. Esta técnica baseia-se em conteúdo gerado pelo utilizador (p.ex: *ratings* ou *feedback* implícito) onde os itens são recomendados a um utilizador se estes forem apreciados por utilizadores similares.

Os filtros colaborativos funcionam construindo uma matriz M com n itens e m utilizadores (ver figura 3.1), que contém a interacção dos utilizadores com os itens (p.ex *ratings*, número de visualizações, etc.). Cada linha representa um perfil de um utilizador e as colunas representam itens (músicas). O valor M_{u_a, i_j} é o *rating* que o utilizador u_a atribuiu ao item i_j .

Algoritmos que usem a técnica dos filtros colaborativos podem efectuar recomendações a partir de dois métodos conhecidos como *item-based neighbourhood* e *user-based neighbourhood*.

O primeiro método explora a similaridade entre os diversos itens, calculando a semelhança entre os itens classificados pelo utilizador e o item a ser recomendado com o

	i1	i2		ij		in
u1				4		
u2				Φ		
				4		
ua				?		
				2		
				1		
um				Φ		

Figura 2.1: Matriz típica usada nos filtro colaborativos.

objectivo de decidir se o novo item deve ou não ser recomendado. A similaridade entre dois itens pode ser calculada utilizando, a similaridade do co-seno (eq. 2.1), o coeficiente da correlação de pearson ou através de uma probabilidade condicionada.

$$sim(i, j) = cos(\mathbf{i}, \mathbf{j}) = \frac{\mathbf{i} \cdot \mathbf{j}}{\|\mathbf{i}\| * \|\mathbf{j}\|} = \frac{\sum_{u \in U} r_{u,i} r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \sqrt{\sum_{u \in U} r_{u,j}^2}} \quad (2.1)$$

O segundo método por sua vez faz a recomendação tendo em conta os utilizadores que são semelhantes ao utilizador ao qual se quer recomendar um dado item. A equação 2.2 calcula o valor de predição do item i para o utilizador u . \bar{r}_u é a média de *ratings* atribuídos pelo utilizador u e $r_{u,i}$ corresponde ao *rating* atribuído pelo utilizador u ao item i . $S^k(u)$ é o conjunto de k vizinhos do utilizador u , desta forma a recomendação do item i ao utilizador u é expressa como:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in S(u)^k} sim(u, v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in S(u)^k} sim(u, v)} \quad (2.2)$$

Embora este método seja um dos mais usados para a recomendação de itens, este apresenta alguns problemas. Um deles acontece quando temos um número relativamente elevado de utilizadores e itens. Neste caso a possibilidade de a matriz ser esparsa é bastante elevada, fazendo com que se tenha uma baixa cobertura de *ratings* entre os vários itens.

Outro problema deve-se ao facto de os filtros colaborativos serem baseados em avaliações dos utilizadores. Assim a entrada de novos utilizadores com poucas avaliações efectuadas torna difícil a sua categorização. O mesmo problema acontece com novos itens, devido ao facto de inicialmente não terem nenhuma avaliação quando adicionados à colecção. Assim não podem ser recomendados até os utilizadores atribuírem alguma avaliação. Este problema é conhecido como *early-rater* ou *cold-start problem*[2].

2.1.2 Recomendação baseada no conteúdo

Enquanto que as técnicas baseadas em filtros colaborativos foram das primeiras a serem usadas na recomendação de música, as recomendações baseadas em conteúdo no domínio da música têm sido consideravelmente menos usadas. A razão de tal acontecer pode prender-se com o facto de as técnicas baseadas no conteúdo requererem conhecimento sobre os dados, e a música é notoriamente difícil de descrever e classificar.

Os sistemas de recomendação de música baseados em conteúdo caracterizam-se por explorar técnicas tradicionais de recuperação de informação musical como é o caso da impressão digital acústica [3] ou a detecção por género [4]. Numa abordagem de recomendação baseada no conteúdo, o sistema coleciona informação que descreve os itens, depois baseando-se nas preferências do utilizador ele prevê que itens é que o utilizador é capaz de gostar. Esta abordagem não se baseia em avaliações efectuadas pelos utilizadores, mas sim na descrição dos itens.

Neste sistemas a similaridade entre dois itens é calculada usando a distância entre dois itens. Alguns dos métodos mais comuns para o cálculo destas distâncias são: distância euclidiana, distância de Manhattan (eq. 2.3), distância de Chebyshev ou distância de Mahalanobis.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2.3)$$

Algoritmos de aprendizagem são também muitas vezes utilizados neste tipo de sistemas de recomendação, sendo o algoritmo *Nearest Neighbor* um dos mais usados.

Os itens são normalmente representados por vectores de características de n dimensões. As características que descrevem os itens podem ser recolhidas automaticamente (p.ex usando a análise do sinal acústico) ou atribuídas manualmente (p.ex por especialistas do domínio). O passo fundamental deste método é a extracção dessas características.

Tal como no método dos filtros colaborativos, a recomendação baseada no conteúdo sofre de alguns problemas. Um dos principais problemas desta abordagem é a falta de novidade que pode existir nas músicas recomendadas. Isto acontece porque o sistema tenta recomendar itens que são similares aqueles que contribuem para definir o perfil do utilizador. Este tipo de recomendação apresenta outro problema que consiste no facto de estarem focados em encontrar semelhanças entre vários itens usando apenas características da própria música (ritmo, timbre e género). O que significa que a opinião pessoal do utilizador não está a ser tida em conta quando a recomendação está a ser efectuada.

Apesar das suas limitações as abordagens baseadas no conteúdo conseguem ultrapassar alguns dos problemas identificados nos filtros colaborativos. Com o uso deste método não existe o problema de novos itens não serem recomendados, visto que não existe nenhuma classificação e a selecção dos itens está apenas dependente das características dos mesmos.

2.1.3 Recomendação usando filtros demográficos

A filtragem demográfica é um dos métodos mais básicos que permite identificar o tipo de utilizadores que apreciam um certo item [5]. Com este método, pode-se saber o tipo de pessoa que gosta de um certo cantor (por exemplo, encontrar o estereótipo de utilizador que ouve a banda Muse).

Esta técnica classifica os perfis de utilizadores em grupos de acordo com alguns dados pessoais (idade, estado civil, género, etc), dados geográficos (cidade, país) e dados psicográficos (interesses, estilo de vida, etc.). O principal problema deste método reside no facto de o sistema recomendar os mesmos itens para pessoas com perfis demográficos similares, assim as recomendações são muito gerais ou pelo menos pouco específicas para um dado perfil de utilizador. Outra desvantagem é a geração do perfil que necessita de algum esforço por parte do utilizador.

2.1.4 Recomendação baseada no contexto

A ideia de usar informação sobre o contexto em aplicações de computador foi introduzida a meio dos anos 90 por Dey, que definiu o contexto como "qualquer informação que pode caracterizar a situação de uma entidade. Onde uma entidade pode ser uma pessoa, lugar ou objecto que é considerado relevante para a interacção entre o utilizador e a aplicação, incluindo também o próprio utilizador e aplicação" [6].

No domínio de sistemas de recomendação, o contexto pode ser a situação do utilizador enquanto procura por uma recomendação como por exemplo o tempo, actividade que está a realizar ou estado de espírito.

Tecnologias avançadas como *smart devices* e *wearable computers*, criaram a oportunidade aos investigadores de recolher e usar dados sobre o contexto para enriquecer a interacção entre o utilizador e os computadores. A informação situacional como o tempo, estado emocional, presença de outras pessoas, eventos passados e futuros podem ajudar o sistema a perceber melhor as necessidades correntes do utilizador.

Graças aos dispositivos móveis, os utilizadores podem aceder a informação e serviços em diferentes ambientes e situações. As necessidades dos utilizadores podem variar dependendo do contexto, e estes sistemas de recomendação podem adaptar-se para fornecer os serviços mais relevantes.

Podemos dividir a informação referente ao contexto em três grupos: Relacionados com o contexto de ambiente (informação sobre a localização do utilizador, temperatura, condições atmosféricas, hora actual, etc.); contexto relacionado com o utilizador (informação sobre a actividade que o utilizador está a realizar, informação demográfica do utilizador e estado emocional) e contexto multimédia (outros tipos de informação a que o utilizador está a ser exposto para além da música, ex. texto ou imagens).

A motivação de estudar o contexto relacionado com o ambiente, é baseada no facto

de o ambiente em que estamos envolvidos influenciar o estado de espírito ou estado emocional do utilizador, o que conseqüentemente pode influenciar indirectamente as nossas preferências músicas[7].

2.1.5 Abordagens de recomendação híbridas

Os maiores problemas das abordagens que usam filtros colaborativos e das baseadas no conteúdo são os novos utilizadores ou itens e a modelação das preferências dos utilizadores, respectivamente. O principal propósito de um método híbrido é conseguir melhores recomendações combinando algumas das técnicas apresentadas anteriormente.

De acordo com Burke[8] existem vários métodos para a integração das diferentes abordagens num sistema de recomendação híbrido:

Weighted - Um método híbrido que combina o resultado de diferentes abordagens, utilizando por exemplo uma combinação linear das pontuações de cada uma das técnicas de recomendação usadas.

Switching - O sistema utiliza um critério para alternar entre as diferentes técnicas de recomendação. Uma possível solução é o sistema usar uma técnica e se os resultados não forem suficientemente confiantes, o sistema muda de técnica com o objectivo de melhorar a recomendação.

Mixed - As recomendações produzidas pelas diferentes técnicas são apresentadas em conjunto (p.ex numa lista ou matriz)

Cascade - O *output* de uma técnica de recomendação é refinado por outra técnica. Por exemplo, a técnica de filtragem colaborativa pode ser usada para produzir uma classificação dos itens, mais tarde uma técnica de filtragem baseada no conteúdo pode ser usada para resolver possíveis empates.

2.1.6 Discussão

Todos os tipos de recomendação discutidos anteriormente têm as suas vantagens e os seus problemas. Devido à subjectividade existente na música, a assumpção de que os utilizadores com comportamentos semelhantes devem ter os mesmos gostos ainda não foi bem estudada. Assim, embora os filtros colaborativos funcionem bem na maior parte dos casos, problemas como o *cold start* e o *popularity bias* não conseguem ser superados.

O *cold star problem* conhecido também como o problema dos dados esparsos, significa que num estado inicial o sistema de recomendação apresenta poucos *ratings* fornecidos. Como os filtros colaborativos dependem muito destes *ratings*, a recomendação torna-se imprecisa.

O *popularity bias* é um problema que está relacionado com a popularidade das músicas. Usualmente músicas populares conseguem ter mais *ratings*. A música menos conhecida por sua vez raramente obtêm algum *rating*. Como resultado os sistemas de recomendação

recomendam maioritariamente as músicas mais populares. Isto é um problema porque não torna a recomendação interessante sendo que apenas se recomendam músicas já conhecidas do utilizador.

Para além dos problemas apresentados nos filtros colaborativos, os sistemas de recomendação baseados no contexto na área de recomendação musical ainda não foram muito estudados e um dos seus problemas reside no facto de entender como é que as condições do contexto influenciam a nossa percepção sobre a música.

Os sistemas de recomendação baseados no conteúdo procuram resolver os problemas apresentados pelos filtros colaborativos e pelos sistemas de recomendação baseados no contexto. No entanto sendo este método baseado em técnicas de similaridade, o resultado da recomendação deste tipo de sistemas pode tornar-se demasiado restritivo, não permitindo descobrir músicas novas, sugerindo apenas músicas muito iguais.

Por sua vez não existe dúvida de que os sistemas híbridos apresentam um desempenho melhor do que as restantes abordagens apresentadas, uma vez que conseguem colmatar os problemas apresentados por cada tipo de técnica individualmente, integrando no seu funcionamento as diferentes características dessas técnicas.

2.2 Trabalhos de recomendação de música

2.2.1 Context-Aware Mobile Music Recommendation for Daily Activities

Com a popularidade dos *smartphones* e com o largo número de sensores e capacidades de comunicação sem fios que estes dispõem, os autores deste trabalho apresentam uma nova abordagem que incorpora informação do contexto do utilizador recolhida através de um dispositivo móvel. Com esta abordagem os autores acreditam ser mais precisa a recomendação de modo a satisfazer as necessidades dos utilizadores a curto prazo [9].

A ideia apresentada consiste em criar um sistema de música que seja capaz de detectar de forma automática e em tempo real, as actividades que o utilizador está a realizar e tocar músicas adequadas a essas actividades.

Neste trabalho os autores consideram 6 actividades (correr, andar, dormir, estudar, trabalhar e fazer compras) e assumem que cada utilizador pode apenas estar presente numa actividade. De modo a conseguir detectar a actividade que um utilizador está a realizar, o sistema guarda um conjunto de dados provenientes dos sensores. Para o modelo apresentado, os dados dos sensores incluem informação do tempo (horas), dados do acelerómetro e áudio proveniente do microfone.

O problema da recomendação é formulado em dois passos: (1) infere o contexto actual do utilizador usando os dados fornecidos através dos sensores e (2) encontra uma música adequada a essa actividade.

Para além de utilizar informação sobre o contexto o modelo apresentado utiliza também *feedback* implícito do utilizador, de modo a fazer uma recomendação mais personalizada. O modo de recolher este *feedback* é bastante simples. Se um utilizador ouvir uma música até ao fim então para o contexto corrente é atribuído um *feedback* positivo. No caso do utilizador mudar de música passados poucos segundos da mesma ter começado, então é atribuído um *feedback* negativo.

Para modelar a relação entre a música e o contexto os autores utilizam o conteúdo do áudio. Dado que uma música pode estar associada a mais do que um contexto, o problema de classificar as músicas num dado contexto pode ser visto como o método de *tagging* musical conhecido como *Autotager*. Com isto é possível estimar a probabilidade de uma música ser adequada a um dado contexto para todos os utilizadores.

Em alguns testes efectuados, os autores encontraram dificuldade em diferenciar algumas das actividades. Apesar de poderem ser parecidas, as actividades tinham de ser diferenciadas de alguma maneira. Para conseguir capturar esta diferença em vez de obter exactamente o contexto corrente a partir do conjunto de dados \mathbf{f} provenientes dos sensores, é obtida uma distribuição de probabilidade $p(c_i|\mathbf{f})$ para todas as situações de contexto distintas. Aplicando a regra de Bayes's a probabilidade $p(c|\mathbf{f})$ pode ser vista como:

$$p(c, \mathbf{f}) = \frac{p(\mathbf{f}|c)p(c)}{p(\mathbf{f})} \propto p(\mathbf{f}|c)p(c) \quad (2.4)$$

Utilizando esta probabilidade verificou-se um melhoramento em relação ao contexto inferido pelo sistema.

Para modelar quando é que uma música s é adequada para uma categoria de contexto c , os autores introduziram uma variável aleatória $R \in \{0,1\}$, onde $R = 1$ significa que a música s é adequada para o contexto c . A probabilidade $p(R = 1|c, s)$ é usada para indicar o grau de satisfação do utilizador para a música s quando este está no contexto c .

Com esta solução os autores conseguiram mostrar resultados bastante positivos, no entanto para conseguirem uma recomendação mais eficaz poderiam ter usado filtros colaborativos. Estes poderiam ser usados para melhor entender as preferências de cada um dos utilizadores, no entanto é necessário ter cuidado pois é importante que o esforço do utilizador enquanto classifica as músicas seja baixo.

Um aspecto negativo que devemos ter em conta é o facto de nesta solução os autores não terem considerado o factor de frescura das músicas, o que significa que a mesma música pode ser recomendada várias vezes num curto espaço de tempo. Com a utilização deste conceito seriam capazes de obter uma recomendação mais agradável.

2.2.2 Nextone Player: A Music Recommendation System Based On User Behavior

Neste trabalho [10] os autores conceberam um sistema que permite recomendar músicas que são favorecidas pelo utilizador, que são novas para o seu ouvido e que correspondem aos padrões de gosto do mesmo. Para tal os autores usam o comportamento do utilizador como *feedback* para decidir qual a música que deve ser tocada de seguida.

Reduzir a quantidade de *feedback* explícito é um ponto importante no desenho de um sistema de recomendação visto os utilizadores serem normalmente preguiçosos. Para ultrapassar este problema os autores inferem o gosto que um utilizador tem sobre uma dada música. Em particular se uma música é tocada desde o início ao fim, os autores inferem que o utilizador gosta da música e por isso é uma recomendação satisfatória. Por outro lado se a música for mudada passados apenas alguns segundos, é assumido que o utilizador não gosta da música que está a tocar nesse momento.

De forma a manter o tempo de recomendação curto, o método apresentado é baseado nos metadados e no comportamento do utilizador em vez da análise do conteúdo da música. A escolha da música que deve ser tocada de seguida é determinada com base no género, ano, frescura, padrão do tempo e preferência.

Para conseguir prever os atributos de género e ano da próxima música a ser tocada os autores fazem uma análise usando séries temporais em vez de recorrerem ao tradicional método de escolher a música com o género e ano mais semelhantes ao da música actual. A razão é porque alguns utilizadores gostam de ouvir músicas similares de acordo com género e ano enquanto outros preferem misturar músicas com variância em género e ano.

Uma característica nova introduzida pelos autores foi a de considerar a frescura de uma música em relação a um utilizador. A frescura de uma música pode ser considerada como a força de estranheza ou a quantidade de experiência esquecida. Esta característica é calculada pelos autores aplicando um curva de esquecimento[?] para avaliar a frescura de uma música para um dado utilizador. A curva de esquecimento é calculada segundo a equação 2.5.

$$R = e^{-\frac{t}{S}} \quad (2.5)$$

Quanto menor a quantidade de retenção de memória R de uma música na mente do utilizador, mais fresca é a música para o mesmo. Neste trabalho a variável S é definida como o número de vezes que uma música foi tocada e t é a distância entre o tempo actual e a última vez em que a música foi recomendada.

Esta métrica contribui para seleccionar músicas frescas como resultado da recomendação em vez de recomendar um conjunto pequeno de músicas repetidamente.

A força de preferência de uma música desempenha um papel importante na recomendação. Ao escolher músicas o sistema deve dar prioridade às músicas favoritas do

utilizador. A preferência de uma música é calculada com base em quatro factores: número de vezes que a música foi escolhida activamente, número de vezes que foi escolhida passivamente, número de *skips* e número de remoções.

Tendo em conta que os utilizadores têm hábitos ou gostos diferentes para diferentes períodos do dia ou semana, o sistema apresentado toma em consideração o padrão do tempo baseando-se em *logs* do utilizador. O sistema regista a hora do dia e da semana em que as músicas foram tocadas. Na altura da recomendação é usado um modelo para estimar a probabilidade de uma dada música ser tocada no tempo corrente.

Na altura da recomendação os resultados obtidos pelas diferentes perspectivas apresentadas anteriormente são integrados num resultado final. Tendo em conta que diferentes utilizadores têm diferentes gostos, atribuem-se diferentes pesos aos cinco factores na altura da integração dos resultados obtidos.

Este sistema apresenta várias técnicas bastante interessantes. Em primeiro lugar devemos referir o conceito de frescura que é bastante importante para manter motivados os utilizadores a usarem a aplicação, podendo assim despertar o interesse numa música até então desconhecida. Outro dos aspectos interessantes é o uso da interacção dos utilizadores com o sistema para descobrir qual o ano e género da próxima música. No entanto esta abordagem não tem em conta o contexto em que o utilizador está inserido, aspecto esse relevante para uma recomendação mais eficaz.

2.2.3 Lifetrak: Music In Tune With Your Life

O Lifetrak [11] tem como objectivo libertar o utilizador da escolha da música que este pretende ouvir, procurando criar uma experiência musical que esteja em sintonia com ele e com o ambiente que o rodeia.

Para atingir tal objectivo o Lifetrak usa informação sobre o contexto que está relacionada com a localização do utilizador, o tempo de operação, a velocidade e informação do ambiente que o rodeia tal como o tráfego, as condições climatéricas e o som. O contexto é ajustado implementando um modelo de aprendizagem que é baseado no *feedback* do utilizador sobre quando uma música é apropriada para um dado contexto.

A ideia que os autores deste artigo pretendem provar é a de que diferentes sensações influenciam a música que está a ser tocada. Um aspecto que deve ser enfatizado é a maneira pela qual a música é seleccionada. O sistema não analisa autonomamente as músicas, conectando-as aos respectivos contextos, pois os autores consideram que esta pode ser uma má prática visto que diferentes utilizadores podem preferir ouvir estilos de música opostos no mesmo contexto. Em vez disso o Lifetrak baseia-se nos utilizadores para marcarem as músicas da base de dados com informação básica do contexto. O sistema posteriormente descobre o contexto actual do utilizador e escolhe as músicas da base de dados consoante a informação recolhida. O resultado é a lista das músicas mais

apropriadas para aquela situação.

Existem basicamente três componentes principais que envolvem a arquitectura de *software* do Lifetrack. Em primeiro lugar está o documento que contém a lista de todas as músicas e os contextos que são aplicados a essas músicas. Depois existe o motor de contexto que automaticamente recolhe informação dos diversos sensores e que a classifica usando *tags* específicas. Existe também o gerador de *ratings* que toma em conta o contexto actual e a base de dados das músicas do utilizador, que contém *tags* para cada música e que depois gera uma lista de classificações.

O espaço ou localização onde o utilizador está inserido pode ter uma influência significativa na escolha de música. Por exemplo, ambientes urbanos podem evocar emoções mais diferenciadas quando comparadas com ambientes rurais. Para além disso, o conhecimento espacial é necessário para receber informação contextual como o clima, ou para saber se o utilizador está fora (recebe sinal GPS) ou dentro de casa (não recebe sinal GPS).

Os utilizadores normalmente preferem ouvir determinado tipo de música em diferentes períodos do dia, e podem associar actividades a certos períodos ao longo do dia, tais como fazer uma corrida matinal ou voltar à noite do trabalho.

A velocidade do utilizador é outra informação sobre o contexto que é importante porque, um dado utilizador pode preferir música motivacional enquanto faz uma corrida ou música pacífica numa viagem possivelmente stressante entre o trabalho e a sua casa.

O estado do ambiente é obtido de duas maneiras distintas. Se o utilizador não estiver a conduzir, o Lifetrack calcula a média do valor em decibéis de uma amostra de áudio com cinco segundos. O nível em decibéis dá uma indicação da ocupação do ambiente em que o utilizador está inserido, o resultado obtido é depois mapeado em calmo, moderado ou caótico. No caso do utilizador estar a conduzir o sistema obtém as condições do tráfico referente à localização do mesmo através de um *feeds* RSS.

A última informação do contexto utilizada pelos autores refere-se ao clima, neste caso o sistema vai buscar informação meteorológica de *feeds* RSS e divide esta informação em *tags* que representam as condições climáticas. A temperatura pode assim ser categorizada por frio, moderado, morno ou quente.

Como último aspecto devemos referir como é tratado o *feedback* do utilizador. Uma das soluções apresentadas é a inclusão de uma espécie de equalizador que permite especificar o grau de importância de cada um dos tipos de contexto. Podendo assim o utilizador decidir que a informação retirada da sua localização é mais importante que as condições climáticas aquando da decisão da próxima música a tocar. O segundo mecanismo de *feedback* utilizado foi a inclusão de dois botões, o botão "*love it*" serve para o utilizador afirmar que gosta de ouvir a música actual no contexto em que está presente. Por sua vez o botão "*hate it*", serve para o utilizador mostrar o seu desagrado com a recomendação efectuada.

Quando o *feedback* fornecido é negativo o valor das *tags* presentes na música referentes

ao contexto actual decresce, por sua vez se o *feedback* é positivo o valor dessas *tags* aumenta.

Para melhorar a experiência do Lifetrack, seria interessante os utilizadores poderem atribuir emoções aos diversos contextos existentes. Assim seria possível associar mais facilmente as *tags* presentes na música com as *tags* presentes nos contextos. Visto que existem muitos serviços que já têm *tags* associadas a músicas, seria interessante usá-los para melhorar o sistema fazendo assim uma espécie de *tagging* colaborativo.

Finalmente outra característica que não está presente nesta solução e que seria importante é a questão da frescura das músicas.

2.2.4 Foxtrot: A Soundtrack for Where You Are

Neste artigo [12], os autores propõem uma aplicação móvel de nome Foxtrot, que permite às pessoas partilharem a música que estas associam a uma dada localização. Para proporcionar uma experiência atraente, mas ao mesmo tempo discreta, o Foxtrot mistura as *tags* atribuídas pelos utilizadores, a localização geográfica e a música, para criar uma espécie de canal de rádio onde as pessoas podem ouvir a música associada a uma dada localização.

Um dos principais objectivos do Foxtrot é aumentar a sensação de se estar num lugar criando uma atmosfera emocional sobre esse lugar. Esta aplicação funciona como um canal de rádio que dinamicamente selecciona o conteúdo a tocar para o utilizador baseando-se na sua localização, nas suas preferências e na popularidade das músicas para a actual localização.

Para os autores uma das suas maiores preocupações durante a concepção do Foxtrot foi a de como mapear as músicas em localizações. Para tal pediram aos utilizadores que especificassem precisamente as coordenadas geográficas para cada item. A vantagem de usar as coordenadas exactas reside no facto de ser mais simples para o sistema calcular as músicas mais próximas de uma dada localização.

No entanto, alguns utilizadores podem achar difícil compreender porque é que uma música está associada a uma dada localização. Para minimizar este efeito os autores permitiram aos utilizadores associarem uma distância para a música que atribuíram à localização. Isto define assim uma região circular centrada no local da música, que representa a área dentro da qual a música é relevante.

O Foxtrot gira em volta de dois objectos primários, os utilizadores e as músicas. Os utilizadores têm a sua localização corrente, velocidade, perfil de audição, histórico e lista de amigos. Cada música está centrada numa localização geográfica, tem uma distância de visibilidade, um tipo e um conjunto de *tags* especificadas pelos utilizadores.

A abordagem apresentada para criar uma *playlist* consiste em três passos. Num primeiro passo, a aplicação recolhe músicas que estão perto da localização do utilizador,

ordenadas por ordem crescente de distância.

A cada música está associada uma pontuação que indica a sua relevância para o utilizador. Esta pontuação é uma combinação de três factores, uma pontuação de preferência, pontuação de relevância geográfica e uma pontuação de frescura.

Uma vez que no Foxtrot o utilizador pode estar a andar a pé ou de automóvel, a distância ao ponto onde está localizada a música não é usada directamente para calcular a música a tocar. Em vez disso é usada a velocidade do utilizador para calcular o tempo expectável que o utilizador demora a chegar à posição onde se encontra a música. Isto significa que, se o utilizador viajar a grande velocidade (p.ex, num carro), o sistema procura músicas que se encontrem a uma distância superior à sua posição actual.

O valor correspondente à frescura da música tem em conta factores como a quantidade de vezes em que a música foi tocada para o utilizador e qual a última vez em que isso aconteceu. Quantas mais vezes e quanto mais recentemente o utilizador ouviu uma dada música, menor o valor de frescura para a mesma. Isto faz sentido visto que se quer fornecer ao utilizador um conjunto de músicas novas e interessantes que este ainda não conheça.

Apesar do conceito ser interessante a proposta apresentada sofre de alguns problemas. O primeiro problema que podemos verificar nesta solução está relacionado com o contexto. Qualquer pessoa pode atribuir uma música a uma dada localização, no entanto esta atribuição pode ter sido feita num contexto completamente diferente daquele no qual uma pessoa vai ouvir essa música, o que pode levar a pensar que a atribuição feita não faz qualquer sentido.

O segundo problema que podemos detectar tem a ver com a utilização do sistema enquanto o utilizador está a conduzir, visto que este está mais preocupado com o que se passa na estrada, nestas condições o sistema não deve produzir grandes resultados.

2.2.5 Music Playlist Recommendation Based on User Heartbeat and Music Preference

Neste trabalho, os autores apresentam um novo sistema de música baseado na web que tira partido do *biofeedback* do utilizador [13]. Este integra os conceitos de sistemas adaptativos, preferência musical do utilizador e métodos de utilização da música para ajustar a frequência cardíaca do utilizador.

O sistema não recomenda apenas músicas baseando-se na preferência musical do utilizador, mas também recomenda músicas tendo em conta o batimento cardíaco do utilizador para o manter dentro do intervalo normal, que para pessoas saudáveis com idade superior a 18 anos está entre os 60 e os 100 batimentos por minuto.

Se o batimento cardíaco for superior ou inferior em relação ao batimento cardíaco normal, o sistema escolhe uma música utilizando um processo de decisão de Markov para baixar ou aumentar o ritmo cardíaco de modo a que este volte aos níveis normais.

Para compreender a relação que a música tem com a nossa frequência cardíaca, Peter Sleight descobriu que ouvir música com um ritmo lento ou médio tem um efeito relaxante sobre as pessoas, conseguindo desacelerar a respiração e a frequência cardíaca [14]. Também Knight and Rickard revelaram que a música relaxante atenua a pressão arterial e a frequência cardíaca depois de uma tarefa stressante [15]. Podemos assim perceber que o ritmo da música é um parâmetro importante aquando da escolha da próxima música a tocar.

Para este sistema foram modelados três estados de frequência cardíaca: taquicardia (>100), normal (60-100) e bradicardia (<60). Com a definição destes estados é possível definir qual a próxima música que deve ser escolhida.

Inicialmente é definida a frequência cardíaca normal para o utilizador. De seguida o sistema começa por observar a frequência cardíaca actual do utilizador criando com esta observação uma representação da situação actual do mesmo. Finalmente dependendo da diferença entre a frequência actual e a frequência alvo, a componente de inferência presente na *framework* determina (1) quando é que o utilizador está ou não no estado pretendido e (2) se o utilizador não estiver no estado pretendido então uma música é recomendada utilizando as preferências do utilizador e a informação para transferir o utilizador do seu estado actual para o estado pretendido.

Neste sistema o utilizador funciona como um sistema adaptativo, pois é a sua percepção que cria uma representação interna da música. Para além da informação recolhida através do utilizador, o sistema utiliza um conjunto de metadados que descrevem as músicas.

Com esta solução podemos verificar que o ritmo da música está directamente associado à nossa frequência cardíaca, sendo este um factor que deve pesar aquando da recomendação de uma música. No entanto os métodos para extrair tal informação são algo intrusivos e podem causar desconforto quando usados durante muito tempo. Para além disso nem todos as pessoas têm a possibilidade de andar com um medidor de frequência cardíaca durante a realização das suas actividades.

2.2.6 InCarMusic: Context-Aware Music Recommendations in a Car

Na metodologia apresentada neste artigo os autores elaboraram um desenho de sistema com uma metodologia onde é assumido que os utilizadores podem ser solicitados para julgar duas coisas: 1) se um dado contexto é um factor relevante para a descrição da sua tarefa e 2) como é que eles classificam um dado item assumindo que estão num dado contexto [16].

InCarMusic é uma aplicação móvel que oferece recomendação de músicas a passageiros de um carro depois destes terem atribuído *ratings* a alguns itens, usando para tal efeito uma aplicação web. Nesta solução são utilizados filtros colaborativos e o contexto em que

o utilizador está inserido. Se um utilizador não tiver atribuído nenhum *rating* então é apenas usada a informação do contexto em que o utilizador está inserido.

O conceito de canal é introduzido pelos autores como sendo uma ferramenta que fornece um dado tipo de música ao utilizador. Na especificação de um canal o utilizador pode detalhar por exemplo, que o canal "*Happy-Classic*" é apropriado quando está contente e deve tocar música clássica e jazz. Por omissão é fornecido um canal para recomendação não sendo necessário o utilizador criar um.

Enquanto o utilizador ouve uma música ele pode atribuir uma classificação. Esta classificação é atribuída dentro do contexto em que está a ser ouvida a música, guardando o sistema o *rating* atribuído a uma música num dado contexto.

Sempre que necessário o utilizador pode mudar a situação do seu contexto actual bem como as suas preferências por géneros musicais.

A fim de perceber a influência do contexto nas preferências musicais de um passageiro num carro, o contexto foi modelado como um conjunto independente de factores contextuais. Foram assim encontrados alguns factores contextuais para os quais estão associadas algumas condições como por exemplo: estilo de condução (relaxada ou desportiva), paisagem (urbana, montanhosa ou campo), condições atmosféricas (sol, chuva ou nublado).

Os autores concluíram que a utilização dos filtros colaborativos, juntamente com a informação sobre o contexto pode melhorar o processo de recomendação de músicas. Um dos aspectos negativos da solução apresentada está relacionado com os filtros colaborativos em que o utilizador tem de dispensar algum tempo para criar o seu perfil atribuindo *ratings* a várias músicas e para seleccionar que tipo de factores contextuais influenciam cada um dos géneros de música.

2.2.7 Context-Aware Recommendation by Aggregating User Context

Nos sistemas de recomendação tradicionais, as alterações das preferências dos utilizadores de acordo com o contexto em que este está inserido não são tomadas em atenção. No entanto neste artigo [17] os autores têm em atenção não apenas as preferências dos utilizadores, mas também a informação do contexto em que a música está a ser ouvida. Assim, juntando a informação do contexto, a qualidade de recomendação é melhorada e mais precisa.

A proposta apresentada consiste em três partes: 1) Abstracção do contexto corrente, 2) a definição do contexto do utilizador, e 3) a recomendação atribuindo itens apropriados. Na camada de abstracção do contexto, um contexto do tipo (p.ex., 21:30, 25°C) é abstraído num conjunto de conceitos (p.ex., noite, quente).

A importância de cada contexto depende do tipo de utilizadores. Por exemplo se um

utilizador é sensível ao contexto do mês então os conceitos relacionados com meses contribuem com maior peso que os outros. O peso adequado para cada conceito é determinado na camada do contexto.

Na camada dos itens, cada item é originalmente representado por valores de atributos tais como o género e cantor. Como os valores não são mais do que palavras chave, é difícil representar a possível semântica de cada uma das palavras chave.

Os autores usam então o *Open Directory Project*¹ (ODP) para representar a semântica como um conjunto de conceitos que são apresentados como os conceitos do domínio. Este domínio é um conjunto de conceitos (um conjunto de categorias) retiradas através do ODP para representar a semântica de um item.

O cálculo para prever as preferências do utilizador sobre um item consiste em dois passos: 1) calcular a similaridade entre o contexto corrente do utilizador e o histórico dos seus contextos passados, e 2) calcular a correlação entre o histórico dos contextos passados e de um item. O item com maior valor é o apropriado para recomendar ao utilizador. Para calcular o passo 2, o histórico do contexto e os itens podem ser representados como um conjunto de conceitos de domínio. Assim a similaridade entre eles pode ser calculada usando a similaridade do co-seno.

A proposta apresentada neste artigo assume que existem padrões específicos entre o contexto do utilizador e os itens acedidos. Isto é, existem características contextuais preferidas pelos utilizadores. Estas características podem ser encontradas juntando contextos.

Neste artigo os autores apresentaram uma proposta de um sistema de recomendação com base em informação do contexto através de uma abordagem em camadas. Foi proposta uma técnica para abstrair informação de contexto para um valor semântico mais elevado que permite ser aplicado ao sistema de recomendação. No entanto tendo em conta os nossos objectivos existem algumas características que aqui não se encontram presentes como é o caso da actividade que o utilizador está a realizar e o histórico do utilizador.

2.2.8 Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns

Em algumas aplicações de recomendação, o facto de não incluírem a situação actual do utilizador pode resultar numa quebra significativa de performance em termos de uma recomendação efectiva pois os utilizadores podem ter diferentes preferências em diferentes contextos.

Os autores deste artigo [18] introduzem um método para inferir o contexto do utilizador em sistemas de recomendação, cuja informação sobre o contexto do utilizador não é completamente observável. Ou seja não é utilizada qualquer tipo de informação sobre a localização do utilizador, condições climatéricas, estação do ano ou tempo. Neste caso o

¹<http://www.dmoz.org/docs/en/help/geninfo.html>

contexto é reflectido na sequência de músicas ouvidas pelo utilizador durante a corrente interacção com o sistema. Dado um conjunto de músicas em que o utilizador mostra interesse durante a interacção com o sistema, o sistema de recomendação deve ser capaz de recomendar músicas adequadas para o contexto actual do utilizador.

O sistema apercebe-se de mudanças nas preferências do utilizador e dinamicamente adapta-se a essas mudanças. Para isso os autores usam uma abordagem de modelação por tópicos para mapear a sequência de interacções do utilizador numa sequência de tópicos latentes que capturem tendências mais gerais nos interesses dos utilizadores.

Os tópicos latentes são gerados a partir das *tags* mais frequentes associadas às músicas, obtidas num *website* de *tagging* social tal como é o caso do last.fm. De forma a capturar mudanças no estado do contexto actual ao longo do tempo os autores usam um algoritmo de *sequential patter mining*.

Dada a interacção corrente do utilizador com o sistema como uma sequência das últimas w músicas, os padrões descobertos são usados para prever o contexto para a próxima música.

Para os autores ter reconhecimento de padrões baseados nos tópicos latentes em vez de padrões das músicas pode ser útil no tratamento do *cold start problem*. Para além do objectivo de recomendação de músicas, esta proposta pode ser usada em várias aplicações onde a ordem das músicas e as transições entre elas é importante.

Com esta abordagem é-nos introduzido um método diferente de recomendação baseada no contexto, apesar de não usarem informação específica sobre o estado actual do utilizador. Os resultados obtidos mostram que este método consegue produzir bons resultados em comparação com sistemas de recomendação baseados no conteúdo. Ainda assim poderia ser mais benéfico juntar informação específica sobre o contexto do utilizador.

2.2.9 Music for My Mood: A Music Recommendation System Based on Context Reasoning

Os autores deste artigo [19] desenvolveram um sistema de recomendação que infere o contexto do utilizador utilizando para esse efeito a técnica de raciocínio baseada em casos bem como o histórico do utilizador. Inicialmente o contexto do utilizador é inferido usando técnicas de *data mining*. Posteriormente a recomendação é feita usando o histórico do utilizador.

O sistema desenvolvido tem o nome de M^3 (Music for My Mood). Este sistema consiste em três camadas: camada de interface, camada aplicação e camada de repositório. A camada de interface tem como objectivo coleccionar dados ambientais referentes ao contexto do utilizador. A camada aplicação é responsável por inferir o contexto do utilizador e seleccionar a próxima música a ser tocada. Esta é constituída por três módulos: o módulo de intenção, de estado de espírito e de recomendação. A camada de repositório é

composta por um conjunto de base de dados que contem as músicas que podem ser tocadas bem como o histórico do utilizador. Para perceber melhor como é feita a recomendação ao utilizador, vamos dividir este processo em quatro passos:

Passo 1 - Uma vez que o M^3 percebe a presença do utilizador, a camada de interface colecciona os dados do contexto de ambiente e envia-os para o módulo de intenção. Neste sistema os dados de ambiente incluem, a estação do ano, mês, dia da semana, condições atmosféricas e temperatura.

Passo 2 - Depois de receber os dados do contexto de ambiente, o módulo de intenção infere quando é que um utilizador quer ou não ouvir música, usando para esse efeito os dados de contexto recebidos anteriormente e o histórico do utilizador. Se este módulo concluir que o utilizador quer ouvir música, então este envia os dados do contexto para o módulo de estado de espírito. Neste caso é usada uma função de similaridade para detectar qual dos casos antigos é mais parecido com o actual, a equação 2.6 representa o cálculo efectuado para detectar essa semelhança.

$$Similarity(N, C) = \frac{\sum_{i=1}^n f(N_i, C_i) \times W_i}{\sum_{i=1}^n W_i} \quad (2.6)$$

Na formula 2.6 N_i representa o valor da i^{th} feature do novo caso e C_i o valor da i^{th} feature do novo antigo. A variável n corresponde ao número de features, W_i é o peso da feature i e $f(N_i, C_i)$ é a função de distância entre N_i e C_i .

Passo 3 - O módulo de estado de espírito infere o estilo de música mais adequado para o utilizador, usando os dados do contexto de ambiente e o histórico do utilizador. O estado de espírito inferido por esta camada é depois enviado para o módulo de recomendação, o valor calculado nesta camada pode corresponder a um de três estados possíveis sendo eles música lenta (Baladas e R&B), qualquer música e música rápida (*Rock/Metal* e *Dance*). O estado de espírito do utilizador num determinado dia d é calculado a partir da equação 2.7.

$$M(d) = \frac{\sum_i n_{id} \times G_i}{L_d}, d = 1, \dots, K \quad (2.7)$$

Nesta equação, a variável i é representada por um dos géneros (*Ballad*, *R&B*, *Rock/Metal*, *Dance*), a variável n_{id} corresponde ao número de músicas do género i ouvidas no dia d , por sua vez a variável G_i é um valor constante referente à actividade i , L_d é o número de músicas ouvidas no dia d , por último K representa o número de dias.

Passo 4 - O módulo de recomendação gera uma lista de músicas adequadas ao estado de espírito do utilizador e envia-as para o sistema de áudio através da camada de interface.

Tanto o módulo de intenção como o de estado de espírito foram desenvolvidos usando a técnica de raciocínio baseado em casos.

Com este artigo podemos verificar que a utilização da técnica de raciocínio baseado em casos é uma boa abordagem quando temos um grande conjunto de treino, só assim

é vantajoso usar esta técnica sendo que quando não se encontram casos semelhantes àquele em que o utilizador se encontra, a recomendação torna-se imprecisa. Uma das características que inicialmente os autores tinham considerado usar eram as horas do dia, no entanto esta não foi usada. Para conseguir uma melhor predição sobre o estado de espírito do utilizador, esta seria uma característica que os autores deveriam pensar em incluir.

2.2.10 Content-based music recommendation based on user preference examples

Começando por um conjunto de músicas fornecidas pelo utilizador que evidenciam as suas preferências musicais, a solução aqui apresentada infere descritores semânticos de alto nível que cobrem diferentes facetas musicais, tais como género, cultura, ritmo e duração da música [20].

Partindo desta ideia foram propostas três abordagens baseadas no conteúdo. Duas das abordagens aqui apresentadas usam uma medida de similaridade semântica para a geração de recomendações. A terceira abordagem por sua vez cria um modelo probabilístico das preferências do utilizador.

Neste trabalho os autores baseiam-se na recomendação baseada no conteúdo, tendo em atenção aspectos como a relevância e a novidade da música a ser recomendada. Para a concepção desta solução os autores começaram por pedir aos utilizadores para escolherem um conjunto de músicas que transmitissem as suas preferências musicais.

Para obter os descritores de cada uma das músicas que fazem parte das preferências dos utilizadores, foram calculadas um conjunto de características de baixo nível para cada uma dessas músicas usando uma ferramenta de análise de áudio desenvolvida pelos autores. Com esta ferramenta foi possível obter propriedades de cada uma das faixas tais como características referentes ao timbre, tempo, tonalidade entre outras.

Estas características de baixo nível são depois usadas para inferir categorias de descritores semânticos tais como o género, estado de espírito (*mood*), instrumentos e ritmo. Depois de recolhidos os descritores semânticos para cada uma das músicas é altura de descrever os três métodos de recomendação usados pelos autores.

A primeira abordagem testada pelos autores tem o nome de *Semantic distance from the mean (SEM-MEAN)*. Esta é a mais simples das abordagens apresentadas, sendo que aqui o utilizador é representado como um ponto no espaço semântico do descritor. Para calcular essa posição no espaço é calculada a média para o conjunto de preferências do utilizador. Com esta abordagem, dada uma coleção de músicas, estas são organizadas de acordo com a sua distância semântica ao ponto onde se encontra o utilizador, sendo depois retornadas como recomendação as 20 músicas mais próximas ao utilizador.

Na segunda abordagem com o nome de *Semantic distance from all tracks (SEM-*

ALL), o utilizador não é representado como um único ponto no espaço, em vez disso são consideradas todas as músicas pertencentes ao conjunto de preferência do utilizador sendo estes os pontos usados para a escolha da música a recomendar. A mesma distância utilizada anteriormente para calcular a distância de uma música a um ponto no espaço é usada nesta abordagem.

A última abordagem experimentada pelos autores deste artigo têm o nome de *Semantic Gaussian mixture model (SEM-GMM)*, aqui o utilizador é representado como um modelo probabilístico das suas preferências no espaço de dados semântico. A probabilidade de cada música de um dado conjunto é calculada, sendo escolhidas as 20 com maior probabilidade. Como vantagem esta abordagem consegue retirar a relevância dos atributos semânticos que correspondem às preferências do utilizador.

Com os resultados obtidos das suas experiências os autores concluem que descritores semânticos de alto nível produzem melhores resultados que os descritores de baixo nível (p.ex. timbre) na tarefa de recomendação de música. Com a solução apresentada esperam que esta seja adequada para descobrir músicas que sejam pouco conhecidas.

Para conseguir melhorar a recomendação seria importante para os autores no futuro usarem um perfil de utilizador que contenha os artistas favoritos, álbuns, géneros e até actividades bem como informação sobre o os seus hábitos de ouvir música.

2.2.11 Discussão

	Tipo de recomendação	Perfil de utilizador	Informação Geográfica	Informação Actividade	Histórico do Utilizador	Conceito de Frescura
[9](3.2.1)	Baseada no Contexto Baseada no Conteúdo	Não	Sim	Sim	Não	Não
[10](3.2.2)	Baseada no Contexto	Não	Não	Não	Sim	Sim
[11](3.2.3)	Baseada no Contexto	Sim	Sim	Sim	Não	Não
[12](3.2.4)	Baseada no Contexto	Sim	Sim	Não	Sim	Sim
[13](3.2.5)	Baseada no Contexto	Sim	Não	Não	Sim	Não
[16](3.2.6)	Baseada no Contexto Filtros Colaborativos	Sim	Não	Não	Não	Não
[17](3.2.7)	Baseada no Contexto	Não	Não	Não	Sim	Não
[18](3.2.8)	Baseada no Contexto	Não	Não	Não	Sim	Não
[19](3.2.9)	Baseada no Contexto	Não	Sim	Não	Sim	Não
[20](3.2.10)	Baseada no Conteúdo	Sim	Não	Não	Não	Sim

Tabela 2.1: Comparação entre artigos de recomendação de música

Para fazer uma comparação entre os diferentes artigos estudados, usámos seis características que consideramos importantes para a nossa proposta de solução e desta forma tentamos perceber em que medida os trabalhos estudados podem ajudar a resolver os nossos problemas.

Sendo que no nosso trabalho queremos apresentar uma proposta de solução para a recomendação personalizada de *playlists*, achamos importante incluir a característica de

perfil de utilizador na nossa tabela. A inclusão da informação geográfica na tabela deve-se ao facto de na nossa solução querermos utilizar esta informação para recolher os *tops* musicais referentes ao país onde o utilizador se encontra daí ser importante verificar como é que esta informação é utilizada por outras soluções.

Uma das nossas principais motivações é a da utilização da actividade que o utilizador está a realizar para a recomendação de música, por isso achamos importante incluir esta característica na nossa tabela para podermos perceber se esta é ou não muito usada nos sistemas de recomendação musical.

Como a nossa proposta de solução pretende efectuar recomendações personalizadas, decidimos introduzir a característica do histórico do utilizador para perceber até que ponto esta tem uma influência positiva no processo de recomendação.

O conceito de frescura é importante pois acreditamos que irá permitir que o processo de recomendação não seja repetitivo, decidimos assim incluir esta característica na tabela para perceber se este é um conceito muito usado e como é que ele é utilizado.

Através da tabela 3.1 conseguimos verificar que quase todos os artigos estudados usam o tipo de recomendação baseada no contexto, existindo dois casos onde são utilizadas abordagens híbridas e apenas um que utiliza unicamente recomendação baseada no conteúdo.

Como já foi referido anteriormente um dos aspectos em que a nossa proposta de solução se foca é na informação da actividade que o utilizador está a realizar. Através da tabela conseguimos perceber que esta informação não é muito utilizada nos sistemas de recomendação musical pois apenas dois trabalhos usam essa informação durante o processo de recomendação.

Por sua vez a característica do histórico do utilizador é a mais comum entre os trabalhos estudados o que mostra consenso sobre a sua importância num sistema de recomendação.

A informação geográfica e o perfil do utilizador são duas características igualmente utilizadas por vários trabalhos, no entanto a utilização da informação geográfica apresentada nestes trabalhos é diferente da utilização que lhe queremos dar.

Já o conceito de frescura é pouco usado nos trabalhos de recomendação musical, existindo apenas três que consideram esta característica interessante. Através da tabela conseguimos verificar que nos dois trabalhos que utilizam a informação da actividade não é usado o conceito de frescura. No entanto na nossa proposta de solução queremos integrar ambas as características.

Olhando para a tabela conseguimos perceber que o Foxtrot [12] é o trabalho que apresenta o maior número de características que queremos incluir na nossa proposta de solução. No entanto a solução proposta neste artigo tem um conceito bastante diferente daquele que propomos. A informação geográfica é utilizada para atribuir músicas a localizações geográficas e são as pessoas que constroem a aplicação sendo elas responsáveis por atribuir músicas às diferentes localizações. Para além disso em vez de usar um leque muito mais vasto de músicas, a personalização deste sistema apenas escolhe entre as músicas

que estão disponíveis numa dada localização o que limita o processo de recomendação.

Com os artigos analisados conseguimos perceber que o problema que aqui apresentámos ainda não está resolvido, para além disso percebemos como é que as características que pretendemos usar são utilizadas.

Capítulo 3

Associação de músicas a actividades

Para conseguir uma recomendação de música eficaz de acordo com a actividade que cada utilizador está a realizar é necessário perceber de que maneira poderá ser possível captar os gostos dos utilizadores de acordo com a actividade que este está a realizar. Com este objectivo em mente pedimos a um conjunto de utilizadores para associarem músicas a actividades, assim o total de músicas associadas para cada actividade são utilizadas para perceber os gostos mais genéricos dos possíveis utilizadores do sistema para cada uma das actividades presentes na nossa solução.

Desta forma, nesta secção descrevemos o método utilizado para captar estes gostos bem como as decisões de implementação tomadas durante este processo. Apresentamos também a aplicação de recolha de dados utilizada, explicando o seu fluxo e a forma como foi desenvolvida.

3.1 Procedimento

Na primeira fase do processo de recomendação presente na nossa solução, são recomendadas um conjunto de músicas por actividade que são iguais para todos os utilizadores. Este procedimento corresponde à interação dos mesmos com o modelo genérico que representa o modelo inicial de recomendação a ser utilizado por todos os utilizadores e cujo objectivo é o de ser moldado tendo em conta a interação do utilizador com o sistema de forma a recomendar músicas que estejam de acordo com os seus gostos musicais para cada uma das actividades. Dado que é a partir deste modelo que se vão gerar os vários modelos personalizados, é importante por isso que este consiga captar gostos gerais a todos os possíveis utilizadores. Desta forma será possível fornecer uma primeira recomendação onde as escolhas dos utilizadores efectuadas durante essa recomendação sejam tidas em conta para começar a moldar o modelo personalizado para cada um deles.

A criação deste modelo foi efectuada passando por diferentes etapas, desde a recolha da informação das músicas que os utilizadores mais gostam para cada actividade, até aos testes efectuados com esses dados.

Para isso, começámos por escolher 6 actividades para as quais a nossa solução deve recomendar músicas. Estas actividades foram escolhidas com base no estudo de Xinxin Wang, David Rosenblum e Ye Wang [9], que identificaram as actividades andar, correr, dormir, relaxar, ir às compras e trabalhar, como as mais importantes.

De forma a conseguirmos recolher as músicas que cada utilizador prefere para cada actividade, elaborámos uma aplicação web que vai ser explicada em mais detalhe ao longo desta secção. Para conseguirmos um equilíbrio no número de respostas por actividades, nesta aplicação é escolhida de forma aleatória uma das 6 actividades que o nosso sistema suporta e para a qual o utilizador ainda não tenha escolhido músicas.

Devido ao paradoxo da escolha de Schwartz [1] que nos indica que a dificuldade de uma escolha é maior, quanto maior o leque de opções que temos à nossa disposição, foi importante arranjar uma forma de filtrar a grande quantidade de músicas e de artistas disponíveis. Assim, o utilizador começa por escolher um ou mais géneros musicais para a actividade actual, de seguida com base nas escolhas anteriores é apresentado um leque de artistas. Destes o utilizador escolhe os que mais lhe agradam, sendo posteriormente apresentadas 100 músicas divididas por todos os artistas seleccionados no passo anterior.

A recolha destes dados por parte dos utilizadores para a criação do modelo genérico pode ser vista na figura 3.1.

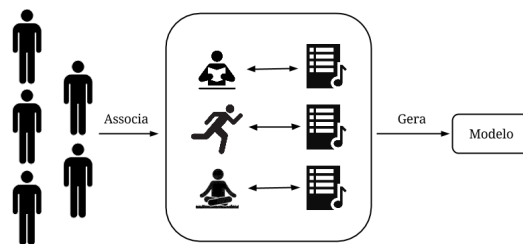


Figura 3.1: Representação da recolha de músicas para a criação do modelo genérico.

3.2 Restrições

De forma a fazermos uma ligação entre as músicas e as diferentes actividades, tivemos que tomar algumas decisões que nos ajudaram no processo de identificação deste paralelismo. Estas decisões têm em conta os diferentes géneros musicais e artistas de forma a conseguirmos apresentar um conjunto de músicas suficientemente alargado, no qual o utilizador tenha várias hipóteses de escolha.

Actualmente existem dezenas de géneros musicais e a sua taxonomia ainda não está bem definida. Devido a esta dificuldade de identificação de géneros musicais, foi importante guiarmo-nos pelo que é utilizado actualmente pelos maiores serviços de música digital, apesar destes utilizarem diferentes géneros para fazerem uma distinção entre as músicas existentes. Assim sendo escolhemos 15 géneros distintos para representar as nos-

sas músicas. O número e os géneros escolhidos tiveram em conta a divisão de géneros efectuada por alguns sites e serviços conhecidos na área da música digital de onde se destacam o last.fm, musicbox e o musicoverly.com.

Para além da identificação dos principais géneros musicais, foi necessário utilizarmos uma filtragem por artista. Esta filtragem tem em conta três factores; o número de géneros disponíveis, os géneros escolhidos para cada actividade durante o processo de associação e a popularidade actual de cada um dos artistas existentes. O número de artistas apresentados é de 30, sendo este número o dobro do número de géneros existentes. Esta é a quantidade de artistas ideal visto não ser um número muito elevado e que garante no pior caso pelo menos duas escolhas possíveis para cada um dos géneros, sendo que isto acontece quando o utilizador escolhe todos os 15 géneros para uma dada actividade.

Uma vez que existem milhões de músicas à nossa disposição, foi necessário fazer uma filtragem no número de músicas a apresentar. Para atingir esse objectivo as filtrações feitas anteriormente são de extrema importância isto porque a filtragem por género, permite escolher o género da música que melhor se adapta aos gostos do utilizador consoante a actividade actual e a filtragem por artistas por sua vez é aquela que permite escolher os artistas que o utilizador mais gosta reduzindo desta forma o número de músicas possíveis a apresentar ao utilizador. Ainda assim para cada actividade limitamos o número de músicas a apresentar para 100 músicas no total.

3.3 Aplicação de recolha de dados

Tomadas as decisões apresentadas anteriormente, começámos por desenvolver a aplicação web cujo objectivo é o de recolher informação sobre as músicas preferidas para cada uma das diferentes actividades.

Nesta secção explicaremos a arquitectura da aplicação desenvolvida, o seu fluxo de funcionamento e o processo da recolha de dados efectuado a partir da mesma.

3.3.1 Fluxo da aplicação

A aplicação web para a criação do modelo genérico pode ser dividida em três grandes fases. Numa primeira é escolhida uma actividade de forma aleatória entre aquelas para as quais o utilizador ainda não escolheu qualquer música. Ao termos a actividade actual, são apresentados um conjunto de géneros ao utilizador dos quais este irá ter que escolher pelo menos um que esteja de acordo com os seus interesses para a actividade atribuída (figura 3.2).

Na segunda fase são apresentados artistas que espelham as escolhas efectuadas no passo anterior. Estes artistas são obtidos através da API do serviço echonest¹, sendo

¹echonest.com

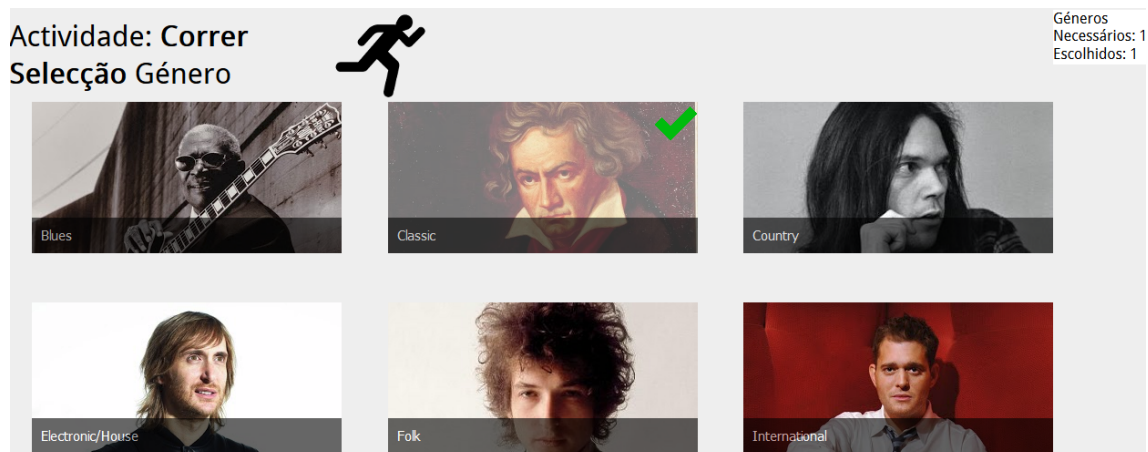


Figura 3.2: Selecção de géneros.

estes os artistas mais relevantes para cada um dos géneros escolhidos. Aqui o utilizador deverá seleccionar pelo menos um para seguir para a última fase (figura 3.3).

Este serviço web disponibiliza um conjunto de ferramentas através da sua API, permitindo entre outras coisas fazer pesquisa por artistas e músicas.

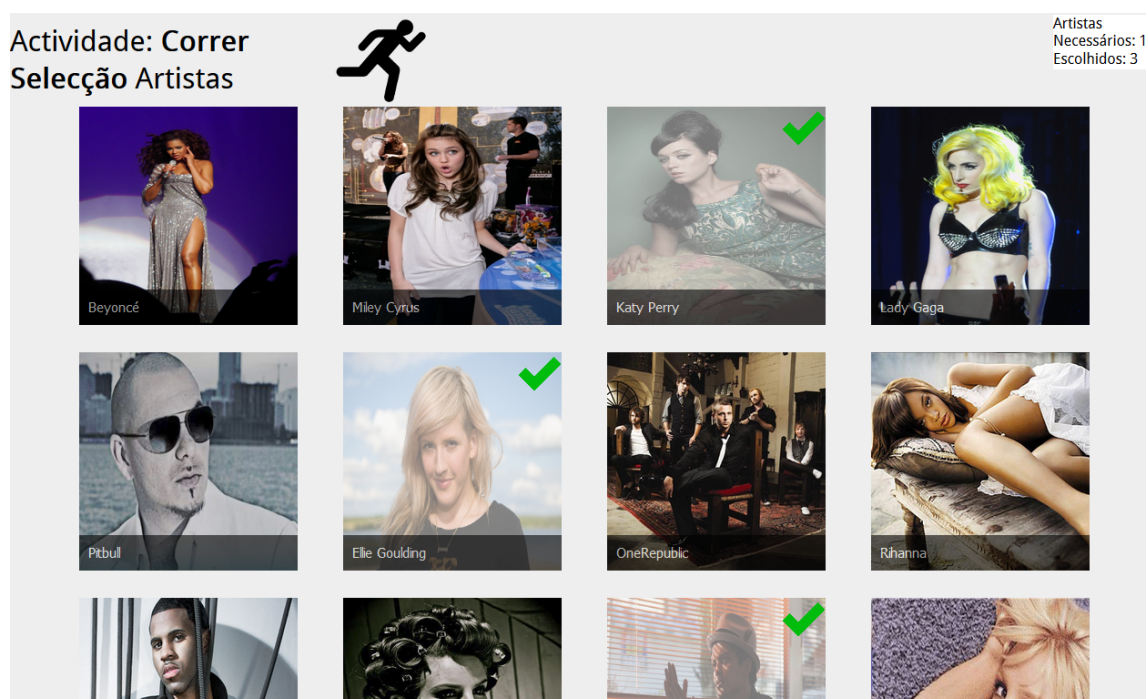


Figura 3.3: Selecção de artistas.

A última fase apresenta um conjunto de 100 músicas referentes aos artistas escolhidos anteriormente. Das músicas apresentadas o utilizador deverá escolher pelo menos 20 que considera adequadas para a actividade em questão.

A construção da aplicação web teve como base as tecnologias HTML, CSS e JavaScript. Para a recolha de artistas e músicas foram utilizadas as API's fornecidas pelo serviço echo-

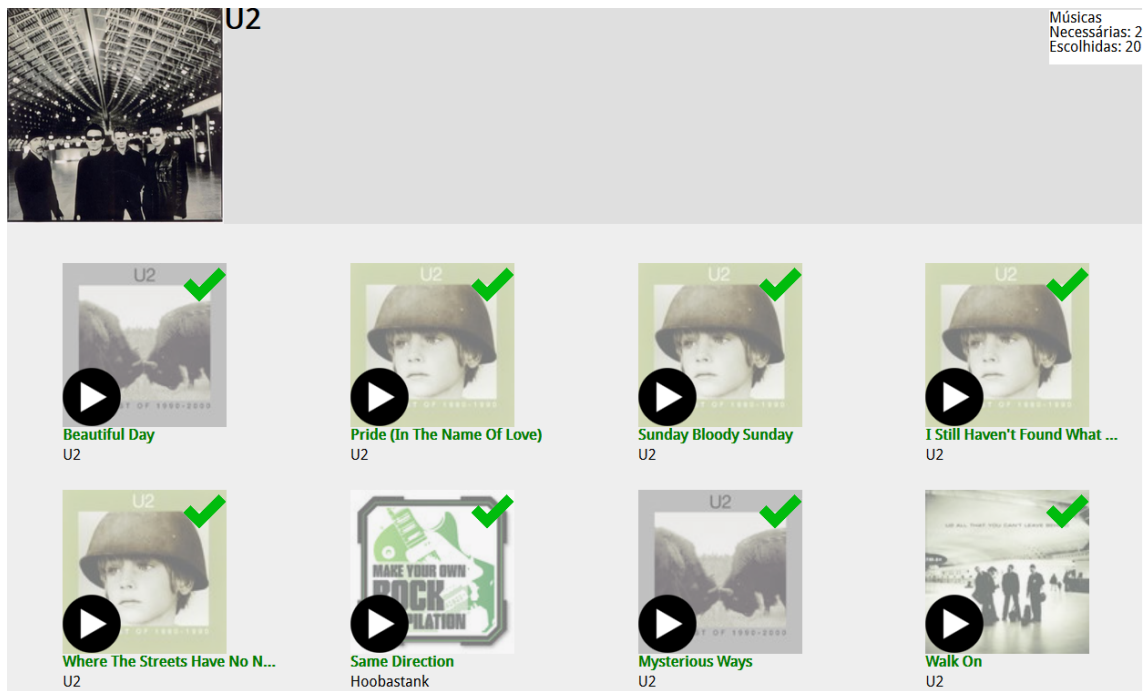


Figura 3.4: Selecção de músicas.

nest, assim como também o serviço fornecido pelo deezer² que permite tocar as músicas previamente escolhidas pelo serviço echonest. De forma a guardar a informação fornecida pelos utilizadores utilizamos a tecnologia PHP que serviu para guardar em diferentes ficheiros de texto, as músicas seleccionadas para cada uma das actividades.

3.3.2 Arquitectura

A nossa aplicação começa com um pequeno formulário com quatro perguntas simples onde o utilizador terá que indicar o seu género, a sua faixa etária, quantas horas por dia ouve música e se ouve música diferente consoante a actividade que está a realizar.

As respostas a este pequeno questionário são posteriormente enviadas para o servidor que está encarregue de guardar esta informação num ficheiro de texto, o qual também incluirá as músicas escolhidas para cada uma das diferentes actividades.

Depois de ter respondido ao questionário, é apresentado um conjunto de 15 géneros, onde o utilizador tem a possibilidade de optar por mais do que uma opção. Na página de escolha de géneros bem como nas restantes páginas que se seguem é mostrado um contador do lado superior direito que mostra o número de itens que já foram seleccionados.

Uma vez escolhidas as opções do formulário de selecção de géneros, essa informação é passada para a página seguinte.

Na aplicação para recolha de dados e na nossa solução utilizamos o serviço web echonest, que nos permitiu fazer a pesquisa de artistas, músicas e reproduzir as músicas

²deezer.com

encontradas.

Tendo em conta as decisões tomadas na secção 3.2, são apresentados 30 artistas tendo em conta os géneros escolhidos anteriormente. Estes por sua vez são apresentados por ordem de popularidade decrescente.

Para apresentar as fotografias para cada artista obtido, fazemos uma pesquisa usando a API do serviço echonest, desta feita esta é uma pesquisa por imagens de cada um dos artistas. O formulário dos artistas é idêntico ao formulário dos géneros diferindo apenas no número de itens apresentados que neste caso é de 30 artistas, o resultado deste é enviado tal como no caso anterior para a página seguinte que é responsável por apresentar as músicas para os artistas seleccionados.

Por último são apresentadas um conjunto de 100 ou menos músicas para os artistas escolhidos. Idealmente seriam apresentadas sempre 100 músicas para o total de artistas seleccionados no entanto pode acontecer ser impossível adquirir o total de músicas para um dado artista e desse modo o número de músicas apresentadas poder ser inferior a 100.

Na fase de recolha de músicas, utilizamos a API do echonest para recolher as 100 mais populares para cada um dos artistas escolhidos. Com a utilização da API do echonest a cada música está associado o seu título, artista e foto do álbum a que pertence sendo esta informação usada para apresentar as músicas aos utilizadores.

A cada música está também associado um conjunto de ID's referentes à representação de cada música num outro serviço web permitindo a integração da API presente nesses serviços com a API do echonest. Estes ID's representam cada música em serviços como o deezer, rdio³ ou spotify⁴.

Na nossa aplicação usámos o ID referente ao serviço deezer que nos permite fazer *streaming* de uma amostra de 30 segundos para as músicas recolhidas anteriormente pela API do echonest. A escolha deste serviço de *streaming* teve por base a facilidade da sua integração com a nossa aplicação. No caso normal o utilizador tem que escolher pelo menos 20 músicas que reflectam os seus interesses (figura 3.4). Quando não existam pelo menos 20 músicas no total, o utilizador deve escolher pelo menos 1 satisfça os seus gostos.

Finalmente depois de escolhidas as músicas esta informação é passada mais uma vez para o servidor que é responsável por guardar a informação referente às músicas escolhidas num ficheiro de texto com o nome da actividade para a qual o utilizador esteve a escolher músicas, juntamente com o número do utilizador que foi guardado aquando da resposta ao primeiro formulário.

³rdio.com

⁴spotify.com

3.4 Síntese

Nesta secção, definimos as actividades para as quais a nossa solução deve recomendar músicas, sendo estas a actividade de andar, correr, dormir, relaxar, fazer compras e trabalhar. No processo de recomendação inicial necessitamos de perceber no geral os gostos dos possíveis utilizadores. Aqui apresentámos uma aplicação web para a recolha destes dados e um conjunto de restrições para ajudar na escolha de músicas por parte dos utilizadores.

Estas restrições são referentes ao número de géneros, artistas e músicas e com elas conseguimos facilitar e guiar de uma melhor forma o processo de escolha de músicas por parte dos utilizadores.

Capítulo 4

Modelo de recomendação

Depois do desenvolvimento da aplicação web responsável pela obtenção da informação sobre os gostos musicais de diferentes utilizadores, para cada actividade foi necessário arranjar utilizadores para nos ajudarem a obter dados. Neste capítulo explicamos o procedimento efectuado para a recolha de dados, como foi efectuada a sua validação, bem como as conclusões a que chegámos para a criação do modelo genérico.

4.1 Recolha de dados

Sendo o modelo genérico um modelo bastante abrangente e visto ser nossa intenção que este seja igual para todos os utilizadores, (pelo menos no início) foi necessário arranjar o contributo de um conjunto grande de pessoas de forma a recolher um conjunto alargado de gostos musicais para cada uma das actividades que o nosso sistema suporta. Assim conseguimos determinar se existem algumas características dominantes a quando da escolha de uma determinada música para uma dada actividade.

Para a recolha dos dados distribuimos o URL da nossa aplicação através das redes sociais facebook e google+, bem como por email, com o objectivo de chegarmos ao maior número de pessoas possíveis. Através desta forma conseguimos obter respostas de 98 utilizadores, com um total de 289 respostas válidas para as várias actividades.

O número de respostas por actividade foi uniforme, sendo que para a actividade *andar* foram obtidas 55 respostas, para a actividade *correr* obtivemos 53, para a actividade *dormir* 47, para a actividade *relaxar* conseguimos 48, para a actividade *fazer compras* atingimos as 48 respostas e por último para a actividade *trabalhar* conseguimos 44, o que equivale a um total de 289 respostas válidas.

A partir destas respostas, identificamos as 100 músicas mais escolhidas para cada uma das actividades. O passo seguinte foi usar a API do echonest para extrair informação do conteúdo das músicas.

4.2 Selecção de características

Como já referimos anteriormente, a nossa solução baseia-se num sistema de recomendação híbrido, que se baseia num sistema de recomendação baseado no contexto e num sistema de recomendação baseado no conteúdo. Nesta secção descrevemos o processo desenvolvido para recolher e escolher as características que melhor classificam as músicas e as actividades a que estas pertencem para cada utilizador.

4.2.1 Características disponíveis

Como referido no capítulo 3, o *echonest* é um serviço web que disponibiliza uma API para a pesquisa de músicas e artistas. Este serviço web permite extrair informação do conteúdo das músicas, sendo que na nossa solução pretendemos tirar partido dessa funcionalidade, nesta secção iremos descrever as diferentes características musicais oferecidas por este serviço.

O *tempo* (andamento) de uma música pode ser entendido como a velocidade de uma música. Quanto maior o *tempo*, maior o número de batidas que devem ser tocadas num minuto e por conseguinte mais rápida a música é tocada. A música influencia o modo como agimos durante a execução de uma actividade e o *tempo* é um bom indicador que nos permite decidir qual a música mais adequada para a actividade que estamos a realizar. Enquanto corremos é preferível uma música com um *tempo* que esteja de acordo com o nosso batimento cardíaco, enquanto estudamos poderá ser mais indicada uma música com um *tempo* lento. No *echonest* esta característica é representado por um número decimal que varia entre 0.0 e 500.0.

O atributo *acousticness* estima o quanto acústica uma música é. Músicas com o valor de *acousticness* elevado vão consistir em grande parte em sons acústicos naturais como (guitarra acústica, piano, orquestra e voz humana não alterada), por outro lado músicas com o valor de acústica baixo vão consistir na sua maioria em sons eléctricos tais como (guitarras eléctricas, sintetizadores, *auto-tuned* entre outros). Com a utilização deste atributo podemos perceber para que actividades o utilizador prefere música mais electrónica ou mais acústica, ao mesmo tempo permitirá fazer uma melhor relação entre os géneros e as actividades. Este atributo é representado pelo *echonest* como um valor decimal que varia entre 0.0 e 1.0.

Por sua vez a *energy* é um atributo que indica o quanto energética é uma dada música. Este atributo vai permitir escolher a música mais adequada para cada uma das actividades que o utilizador está a realizar, uma vez que para a actividade *running* será normal o utilizador preferir uma música com um nível maior de *energy* do que durante a actividade *sleeping*, onde nesses casos poderá preferir músicas com um menor valor de *energy*. Este parâmetro pode estar também ele relacionado com o estado de espírito do utilizador. Este atributo é representado por um valor decimal que varia entre 0.0 e 1.0.

O *loudness* é um atributo do som que pode encontrar-se numa escala que se estende desde um som tranquilo a um som alto. Pelas mesmas razões apresentadas no atributo *energy* este atributo pode ser adequado para escolher músicas entre actividades que exigem um maior esforço físico como por exemplo a actividade *running* e uma actividade mais relaxante como a actividade *relaxing*. Este atributo é representado pelo echonest como um número decimal que pode variar entre os -100.0 e os 100.0 (dB).

O *liveness* é um atributo do echonest que indica o quão viva é uma música. Quanto maior este valor, maior a probabilidade de esta música ter sido gravada ao vivo. O valor deste atributo tem uma gama de valores compreendida entre 0.0 e 1.0.

O atributo *speechiness* permite identificar se uma música é mais ou menos falada, ou seja conseguimos fazer um paralelismo com este atributo e o atributo *acousticness*, sendo que se uma dada música tem um valor de *speechiness* elevado então deve ser de esperar que o seu valor de *acousticness* seja baixo. Tal como sucede com o atributo *acousticness*, este atributo é representado por um valor decimal que varia entre 0.0 e 1.0.

A *danceability* de uma música é um atributo subjectivo, mas que é definido pelo echonest como uma combinação de vários factores da música, alguns destes factores são a força da batida, a estabilidade do tempo, o valor do tempo em geral entre outros factores. Com a utilização deste atributo é possível detectar por exemplo que os *remixes* das músicas tem usualmente o valor de *danceability* mais elevado. Este atributo é definido pelo echoest como um valor decimal com valores que variam entre 0.0 e 1.0.

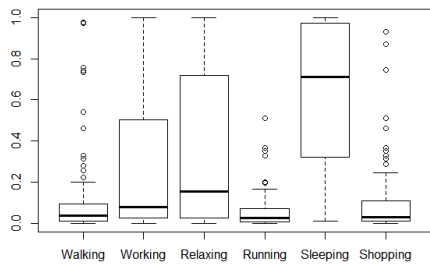
4.2.2 Seleção de características relevantes

Uma vez percebidas as diferentes características recolhidas através do echonest para cada uma das músicas mais escolhidas por parte dos utilizadores para cada actividade, foi necessário verificar se existia alguma relação entre o valor dessas características e a actividade escolhida. Desta forma será possível entender que características têm o maior impacto na escolha de uma determinada música para uma dada actividade.

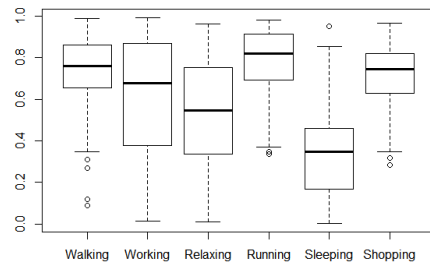
Para termos uma melhor percepção dos valores das características e em que gama é que estes valores se encontram para cada uma das actividades criámos boxplots para os valores das características de cada uma das actividades (ver figura 4.1).

Através de alguns dos boxplots criados conseguimos perceber que é fácil distinguir algumas das actividades, como por exemplo a actividade dormir e correr, no entanto algumas delas não são de fácil distinção.

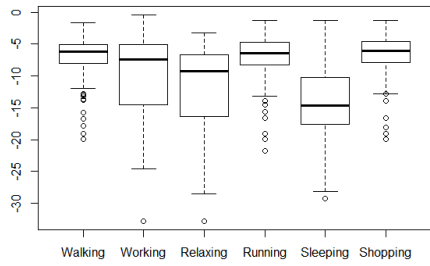
Com os boxplots conseguimos perceber também que algumas das características têm uma gama de valores bastante semelhantes para todas as seis actividades existentes na nossa proposta de solução, o que indicia que a utilização dessas características é irrelevante para a escolha de músicas para uma dada actividade. Os resultados dos boxplots por cada actividade podem ser vistos na figura 4.1.



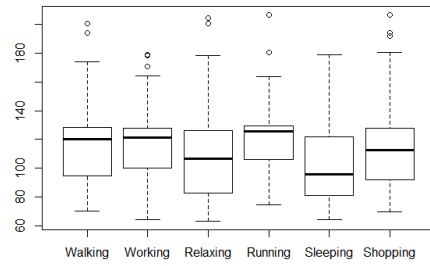
Accousticness



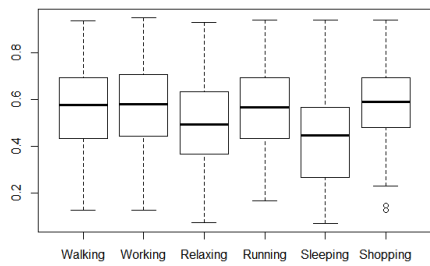
Energy



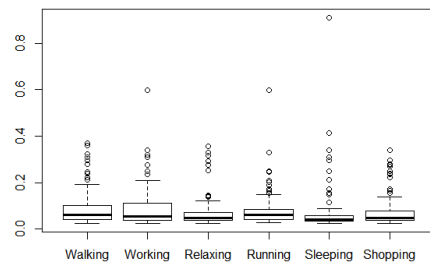
Loudness



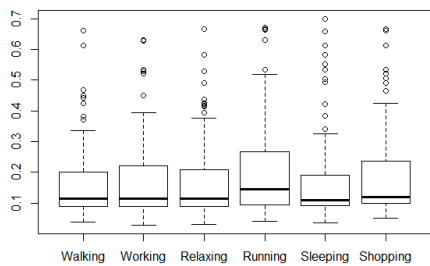
Tempo



Danceability



Speechiness



Liveness

Figura 4.1: Resultados dos boxplots

De forma a percebermos melhor qual a influência de cada uma das características para a selecção de uma determinada música para uma dada actividade, recorreremos ao Weka[21]. O Weka¹ permite usar vários algoritmos de aprendizagem tais como árvores de decisão C4.5 e algoritmos de *clustering* como é o caso do algoritmo *Expectation Maximization*. Para além destas capacidades o Weka pode ser utilizado para avaliar os atributos presentes num conjunto de dados de modo a perceber quais os mais discriminativos.

Esta última funcionalidade foi usada no nosso conjunto de dados para identificarmos as características mais importantes. Para tal usamos o avaliador de atributos *CfsSubsetEval* juntamente com o método de procura *bestfit*. Com esta avaliação conseguimos concluir que das sete características presentes no *echonest* as mais importantes para discriminar a relação entre as músicas escolhidas pelos utilizadores e as actividades são o *acousticness*, *energy*, *loudness* e o *tempo*, o que está de acordo com os boxplots da figura 4.1.

Com base nos resultados deste teste usámos apenas estas quatro características para representar a relação entre as actividades e as músicas, excluindo assim os atributos, *danceability*, *liveness* e *speechiness*.

4.2.3 Qualidade das características relevantes

Depois de escolhidas as características mais importantes, utilizámos o Weka para testar vários tipos de classificadores com o objectivo de perceber se seria fácil distinguir as músicas usando estas características. Dos classificadores testados os classificadores *Random Tree* e *Random Forest*[22] foram os que produziram melhores resultados, tendo sido estes os escolhidos para verificar a qualidade das características.

Para treinar o classificador, usámos um conjunto de treino correspondendo às 20 músicas mais escolhidas por parte dos utilizadores para cada actividade, das quais extraímos os atributos que foram considerados como os mais importantes. O classificador foi testado usando as mesmas músicas do conjunto de treino.

Com os resultados obtidos, conseguimos perceber que no geral as características escolhidas são boas, sendo estas suficientemente discriminativas de forma a prever que música deve pertencer a cada actividade.

	a	b	c	d	e	f
a = Walking	70	5	0	0	5	20
b = Working	20	40	5	20	0	15
c = Relaxing	5	10	65	0	0	20
d = Running	15	15	0	60	0	10
e = Sleeping	5	0	0	0	95	0
f = Shopping	20	10	0	5	0	65

Tabela 4.1: Matriz de confusão.

¹<http://www.cs.waikato.ac.nz/ml/weka/>

Contudo através da matriz de confusão apresentada na tabela 4.1, conseguimos verificar que a actividade *working* é a única onde não foram detectadas pelo menos 50% das músicas pertencentes a essa actividade. Uma das razões para tal suceder prende-se com o facto de esta actividade ser a mais genérica de entre as seis actividades presentes na nossa solução, uma vez que o trabalho pode exigir um esforço tanto físico como mental bastante diferente entre o trabalho que cada pessoa está a executar. Um dos exemplos poderá ser o de uma pessoa que trabalha como *personal trainer* ou na construção civil, profissões que requerem maior esforço físico ao contrário de profissões como a de contabilista ou escritor cujo o esforço físico é menor mas que necessitam de um maior esforço mental. Esta é a razão que encontramos para o facto de ser difícil distinguir esta actividade das restantes.

4.2.4 Refinamento das actividades

Com os dados recolhidos através do classificador, conseguimos verificar que existiu alguma dificuldade em distinguir a que actividade é que algumas das músicas deviam pertencer.

A actividade que mais sofre deste problema é a actividade *working*, uma vez que se confunde na sua maior parte com as actividades *walking* e *running* (ver tabela 4.1).

Devido aos resultados obtidos e ao facto de a actividade *working* ser a mais subjectiva, decidimos excluir esta actividade da nossa solução. Desta forma voltámos a criar o classificador da mesma forma explicada anteriormente mas com apenas 100 músicas em vez das 120 que incluíam as 20 músicas pertencentes à actividade *working*. A matriz de confusão resultante da remoção da actividade *working* pode ser encontrada na tabela 4.2.

	a	b	c	d	e
a = Walking	60	5	5	10	20
b = Relaxing	10	80	0	0	10
c = Running	10	0	75	0	15
d = Sleeping	0	0	0	100	0
e = Shopping	15	10	5	0	70

Tabela 4.2: Matriz de confusão com 5 actividades.

Com a remoção desta actividade, conseguimos um melhoramento no número de músicas correctamente classificadas em todas as actividades, com a excepção da actividade *walking*, que verificou um decréscimo no número de músicas classificadas correctamente. No entanto globalmente o número total de músicas classificadas correctamente foi superior ao caso anterior.

Desta forma daqui para a frente todo o trabalho desenvolvido não tem em conta a actividade *working*, por ser a mais genérica e a mais difícil de distinguir de entre todas as actividades presentes na nossa solução. Com esta decisão acreditamos também que uma melhor recomendação será feita ao utilizador.

4.3 Modelo genérico de recomendação

Depois de termos à nossa disposição as características mais importantes, bem como as músicas mais escolhidas para cada uma das actividades, deparámo-nos com o problema de como utilizar essa informação de forma a conseguir uma recomendação eficaz.

A resposta a esta pergunta pode ser encontrada nas secções que se seguem onde explicaremos o processo de recomendação que vamos seguir, bem como os testes realizados e as conclusões que foram retiradas dos mesmos.

Um sistema de recomendação é muitas vezes visto como a sugestão de itens semelhantes a um vector de características que representa os gostos do utilizador. Isto é, a recomendação é baseada num ponto do espaço n-dimensional onde n representa o conjunto de dimensões/características.

Uma das hipóteses para recomendar através de um ponto no espaço seria calcular o centroíde das músicas mais escolhidas para cada uma das actividades e assim definir um conjunto de seis *clusters* onde a recomendação de músicas seria efectuada tendo em conta o centróide dos *clusters* criados.

No entanto, a utilização do serviço web echonest não permite dado um ponto no espaço obter um conjunto de n músicas mais perto desse ponto. Para ultrapassar esta barreira, optámos pela utilização não de um único ponto no espaço mas sim de um hiper rectângulo que é definido por um conjunto de intervalos que delimitam o valor que cada característica pode tomar.

O hiper rectângulo é assim uma figura geométrica de n dimensões onde cada dimensão representa uma das características presentes nas músicas. Desta forma tendo em conta as decisões tomadas anteriormente estes hiper rectângulos são constituídos por 4 dimensões sendo elas o *accouticness*, *energy*, *loudness* e *tempo*.

Estes são definidos pelos intervalos de cada característica que são calculados a partir dos valores máximo e mínimo que cada uma pode tomar. Assim o tamanho destes rectângulos difere entre actividades e utilizadores. Como temos cinco actividades, iremos ter não um mas sim cinco hiper rectângulos, um para cada actividade.

Nas seguintes sub-secções explicaremos o procedimento e os métodos seguidos para o cálculo dos intervalos dos hiper rectângulos.

4.3.1 Intervalos das características

Depois de recolhidas as músicas mais escolhidas para cada uma das actividades e de termos recolhido os valores dos quatro atributos mais significativos fornecidos pelo echonest (*accousticness*, *energy*, *loudness* e *tempo*), fizemos um conjunto de *queries* através de uma página web desenvolvida utilizando a API do echonest que recebe como valores de entrada, valores referentes aos máximos e mínimos de cada um dos atributos musicais. Estes valores máximos e mínimos referem-se assim aos intervalos de cada uma das características

musicais cujas músicas de uma dada actividade não devem ultrapassar.

Para o cálculo dos intervalos começámos por testar dois métodos. O primeiro era calculado usando a média menos o desvio padrão para achar o valor mínimo do intervalo e a média mais o desvio padrão para achar o seu máximo. O segundo método usado para definir os intervalos foi o de usar o percentil de 10% como o valor mínimo do intervalo e o percentil 90% para o máximo.

Os resultados obtidos referentes aos diferentes métodos utilizados na criação dos intervalos, foram analisados de forma a perceber qual dos métodos era o mais adequado. Para tal foram efectuadas um conjunto de *queries* para cada actividade, cada *query* retornou 100 músicas que foram posteriormente usadas em diferentes algoritmos de classificação presentes no Weka, onde tentámos detectar quantas das músicas resultantes eram reconhecidas como pertencentes à actividade a que os intervalos correspondiam.

Depois de calculados os intervalos através destes dois métodos e de termos recolhido 100 músicas para cada uma das actividades, conseguimos perceber com os testes efectuados que estes métodos iniciais não eram os mais adequados uma vez que os intervalos gerados eram demasiado grandes existindo uma elevada sobreposição entre os mesmos. O que resultava por sua vez numa recomendação pouco diversificada, isto é, bastante semelhante para todas as actividades.

Uma vez que os dois métodos iniciais que testámos não serviam para distinguir a que actividade cada música devia pertencer, utilizámos dois novos métodos que se basearam no primeiro, no entanto desta vez decidimos não utilizar todo o valor do desvio padrão mas apenas uma percentagem do mesmo. As percentagens escolhidas foram de 15, 20, 25 e 30%. Para além da variante resultante do primeiro método, testámos um outro método parecido com o anterior mas com a diferença de usarmos a mediana em detrimento da média.

Com estes novos métodos tínhamos como objectivo obter uma menor sobreposição dos intervalos, reflectindo essa menor sobreposição num melhoramento das músicas consideradas como correctas para cada uma das actividades.

4.3.2 Determinação dos intervalos das características

Como explicado na sub-secção anterior, os dois primeiros métodos devolveram resultados bastante disparees isto porque os intervalos definidos por estes métodos eram demasiado grandes, nesta secção vamos apresentar e analisar os resultados obtidos para os testes efectuados com os métodos média e mediana mais ou menos 15, 20, 25 e 30% do desvio padrão.

Para testarmos o melhor algoritmo, usámos o mesmo classificador referido na secção 4.2.3, que neste caso serviu para verificar qual o melhor método para o cálculo dos intervalos do hiper rectângulo. No entanto desta vez para testar o classificador usámos um

conjunto de *data sets* compostos pelas 100 músicas sugeridas por cada um dos diferentes métodos de cálculo dos limites para cada uma das actividades.

As escolhas das percentagens do desvio padrão a serem utilizadas foram feitas tendo em conta o número de músicas devolvidas pelo *echonest* para o intervalo fornecido. Uma vez verificado que para percentagens inferiores a 15%, o número de músicas apresentadas era demasiado pequeno (<20), excluímos a hipótese de usar percentagens do desvio padrão inferiores a 15% aquando da criação dos limites para as diferentes características musicais.

Para determinar o melhor método para o cálculo dos limites elaborámos um conjunto de *data sets* para teste, onde cada um utiliza os intervalos das características calculadas para as 20 ou 100 músicas mais escolhidas para cada actividade.

No total foram calculados 16 limites distintos, estes resultam do facto de utilizarmos dois métodos de cálculo distintos (média e mediana) e das quatro percentagens distintas utilizadas com o valor do desvio padrão (15%, 20%, 25% e 30%), sendo que estes métodos foram usados para as 20 e as 100 músicas mais escolhidas por cada utilizador, temos um total de 16 *data sets* de teste ($2 \times 2 \times 4$) figura 4.2.

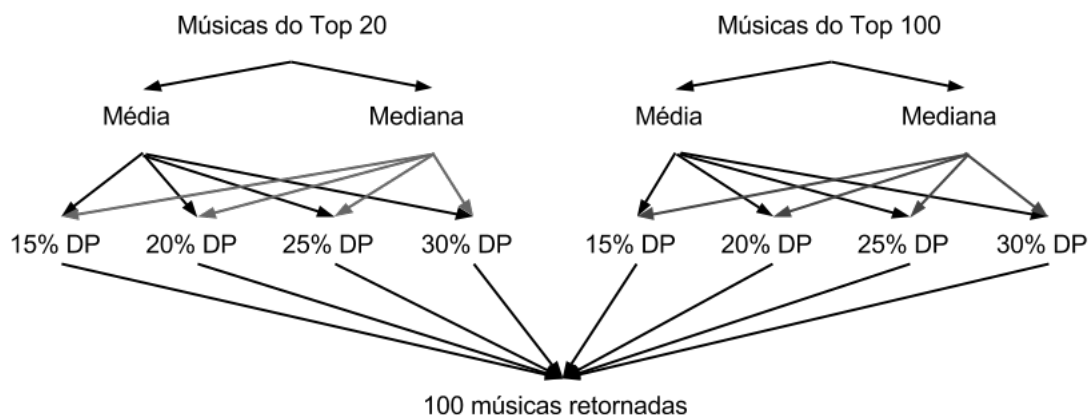


Figura 4.2: Total de data sets calculados.

No processo de determinação do melhor método para o cálculo dos intervalos usámos os classificadores *Random Forest* e *Random Tree*, usando as 20 e as 100 músicas mais escolhidas por actividade como as instâncias de treino do classificador. A utilização destes classificadores tem por base os testes efectuados na secção 4.2.3, uma vez que foi com eles que se conseguiram melhores resultados aquando dos testes efectuados para verificar a qualidade das características musicais usadas na nossa solução.

Cada *data set* de teste foi testado com os dois classificadores (*Random Tree* e *Random Forest*) e foi também testado usando os dois conjuntos de 20 e 100 músicas mais escolhidas por parte dos utilizadores como as instância utilizadas para treinar o classificador. Desta forma foram efectuados um total de 64 testes (ver figura 4.3).

A utilização de 20 e 100 músicas para a criação dos limites prende-se com a necessidade de perceber se será mais benéfico ter ou não um histórico alargado de músicas para



Figura 4.3: Testes efectuados para determinar o melhor método de calculo dos intervalos.

o cálculo dos limites (20 ou 100), de forma a obter a maior percentagem de músicas consideradas correctas para cada uma das actividades.

Assim cada um dos testes realizados utilizam as 20 ou 100 músicas mais ouvidas para cada uma das actividades para criar o classificador e usam 500 músicas (100 por actividade) que representam as músicas obtidas para cada um dos métodos de cálculo dos intervalos das características musicais para testar o classificador.

O gráfico presente na figura 4.4 representa os resultados obtidos para todos os testes efectuados. A nomenclatura usada para descrever cada um dos testes presentes no gráfico da figura 4.3 tem o seguinte significado; O primeiro valor C1 e C2 refere-se à utilização dos dados com os quais o classificador é treinado, o valor 1 corresponde à utilização das 20 músicas mais escolhidas e o valor 2 corresponde às 100 mais escolhidas. Por sua vez o segundo valor S1 e S2 está ligado ao conjunto de músicas utilizadas para calcular os intervalos das características musicais. Neste caso o valor 1 e 2 têm o mesmo significado do valor anterior, assim S1 significa que os intervalos foram calculados usando as 20 músicas mais escolhidas pelos utilizadores e por sua vez S2 representa a utilização das 100 mais escolhidas para o mesmo cálculo. Por sua vez os acrónimos RF e RT retratam o tipo de classificador utilizado no teste, sendo que RF significa *Random Forest* e RT *Random Tree*. Uma vez que se verificou uma igualdade nos resultados obtidos por estes classificadores aquando da detecção da qualidade das características musicais estes foram ambos testados de forma a verificar qual o melhor. O último valor corresponde então ao método utilizado para o cálculo dos intervalos, desta feita o valor AVG representa a utilização da média das características musicais e o valor MEDN representa por sua vez a utilização da mediana.

Através dos resultados obtidos conseguimos perceber que nos métodos onde utilizamos a média das músicas, o número de músicas consideradas correctas para cada actividade tende a decrescer com o aumento da percentagem utilizada do desvio padrão. Por sua vez quando utilizada a mediana, o número de músicas consideradas como correctas para cada actividade parece aumentar na sua maioria com o aumento da percentagem utilizada do desvio padrão.

Os melhores resultados foram obtidos quando utilizamos as 20 músicas mais escolhidas por actividade pelos utilizadores como as instâncias de treino do classificador.

Pelo gráfico conseguimos perceber que o resultado mais elevado no número de músicas consideradas como correctas acontece quando utilizamos o método que é composto pela

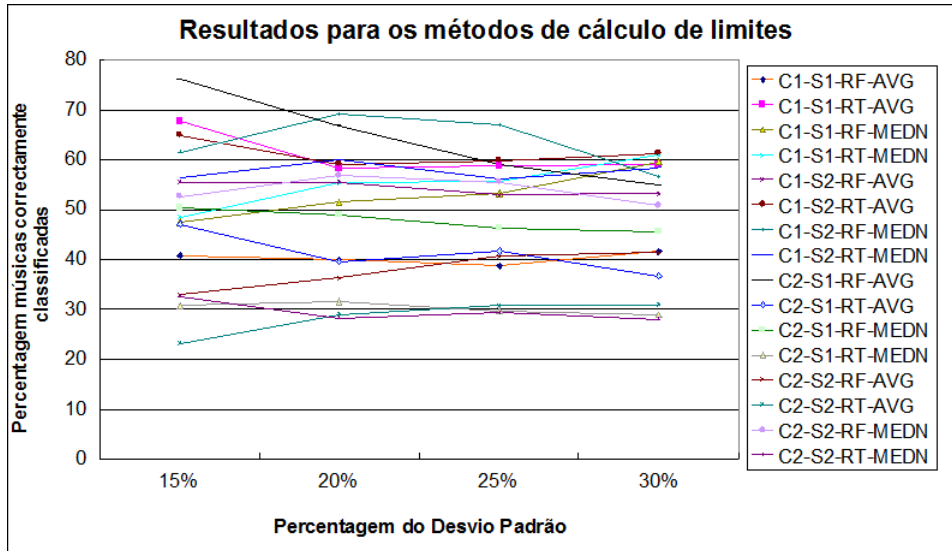


Figura 4.4: Resultados da determinação dos intervalos das características

média \pm 15% do desvio padrão. No entanto é de ter em conta que neste caso o número de músicas usadas para teste (ou seja, recomendadas) foi apenas de 331 músicas ao contrário das 500 utilizadas na maioria dos restantes testes.

Dado o número de músicas recolhidas com o método referido anteriormente ser baixo, decidimos optar pelo segundo melhor método que corresponde à utilização da mediana \pm 20% do desvio padrão. Neste caso a mediana e o desvio padrão foram calculados a partir das 100 músicas mais escolhidas para cada actividade por parte dos utilizadores. Como instâncias de treino do classificador foram utilizadas as 20 músicas mais escolhidas pelos utilizadores.

As tabelas 4.3 e 4.4 correspondem às matrizes de confusão para os dois testes referidos anteriormente (média \pm 15% do desvio padrão e mediana \pm 20% do desvio padrão).

	a	b	c	d	e
a = Walking	88.5	11.5	0	0	0
b = Relaxing	0	100	0	0	0
c = Running	40	0	57	0	3
d = Sleeping	0	5	0	95	0
e = Shopping	19	81	0	0	0

Tabela 4.3: Matriz de confusão com média \pm 15% desvio padrão usando o classificador Random Forest.

Para além do problema referido anteriormente sobre o número de músicas retornadas para o método média \pm 15% do desvio padrão, analisando a matriz de confusão apresentada na tabela 4.3 podemos verificar que das música que deviam pertencer à actividade *shopping*, nenhuma delas foi considerada como pertencente a essa mesma actividade.

Na tabela 4.4 conseguimos verificar uma maior confusão do que aquela apresentada na tabela 4.3, no entanto neste caso foram recolhidas 493 músicas valor bastante próximo do

	a	b	c	d	e
a = Walking	53	11	30	0	6
b = Relaxing	0	85	0	0	15
c = Running	2	0	95	0	3
d = Sleeping	0	0	0	100	0
e = Shopping	52	30	3	0	15

Tabela 4.4: Matriz de confusão com mediana $\pm 20\%$ desvio padrão usando o classificador Random Forest.

desejável (500 músicas), nesta matriz todas as actividades à excepção da actividade *shopping* conseguiram atingir uma taxa de sucesso superior a 50%. Uma vez que comparando estes resultados, com os obtidos nos restantes testes não se detectou um número muito superior em relação ao número de músicas consideradas como correctas para a actividade *shopping*, decidimos usar o método que utiliza 100 músicas para o cálculo da mediana $\pm 20\%$ do desvio padrão para a determinação dos intervalos dos hiper rectângulos (figura 4.5).

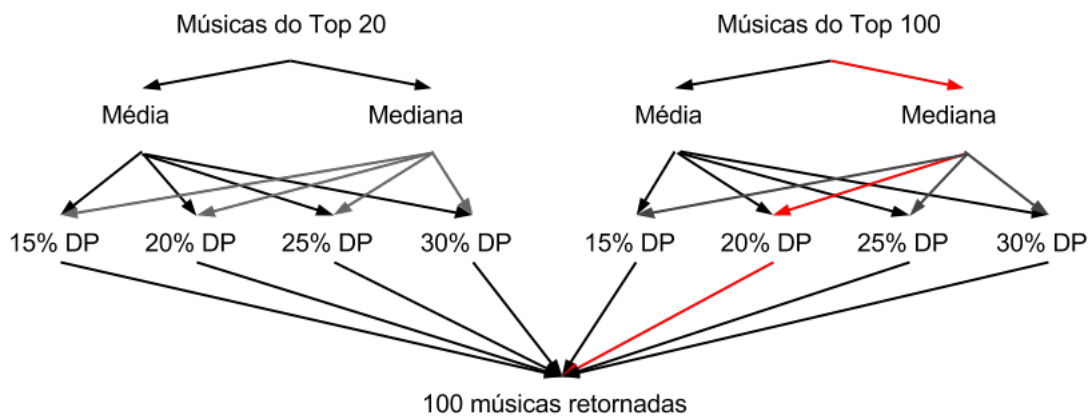


Figura 4.5: Método para o cálculo dos intervalos seleccionado.

4.4 Síntese

Nesta secção apresentámos os resultados obtidos da recolha das músicas escolhidas pelos utilizadores para cada uma das actividades. Com estes identificamos as características mais discriminativas: *acousticness*, *energy*, *loudness* e *tempo*.

Apercebemo-nos que das actividades presentes na nossa solução, o *working* é demasiado genérica, o que a torna mais difícil de recomendar músicas. Como consequência essa actividade foi excluída da nossa solução.

Depois de encontradas as características musicais mais importantes e de termos feito o refinamento das actividades, apresentámos um conjunto de métodos para o cálculo dos limites das características musicais que no processo de recomendação são utilizados para

definir as músicas a recomendar para cada actividade.

Dos resultados analisados chegámos à conclusão que o melhor método para o cálculo dos limites das características musicais é o de utilizar a mediana $\pm 20\%$ do desvio padrão das 100 músicas escolhidas para cada actividade. Os limites calculados a partir deste método serão então usados para calcular as músicas a serem recomendadas pelo modelo genérico. Por sua vez o método utilizado para o cálculo destes intervalos será o mesmo a ser utilizado para o cálculo dos novos intervalos correspondentes à interacção dos utilizadores com o modelo genérico resultando assim num modelo personalizado.

Capítulo 5

Avaliação Experimental

Depois de decidido o melhor método para o cálculo dos intervalos, utilizámos essa informação para criar o modelo de recomendação genérico que fará recomendações iguais para todos os utilizadores. De forma a validar este modelo, elaborámos uma aplicação web onde os utilizadores podem escolher as músicas que consideram mais adequadas para cada uma das actividades. Alguns dos resultados recolhidos durante os testes feitos com o modelo genérico, foram depois usados para treinar o modelo personalizado, que será construído utilizando as músicas escolhidas no teste anterior. A aplicação web construída para testar os modelos personalizados foi a mesma que foi utilizada para testar o modelo genérico.

Nesta secção explicamos o procedimento seguido na construção desta aplicação e os resultados obtidos nos testes efectuados para o modelo genérico e para o modelo personalizado.

5.1 Aplicação de avaliação

A aplicação de avaliação foi construída tendo como objectivo recomendar um conjunto de músicas de acordo com a actividade que cada utilizador esteja a realizar. Os utilizadores começam por responder a um pequeno formulário onde indicam informação demográfica. Depois é apresentado o conjunto de 5 actividades suportadas pelo sistema das quais o utilizador deverá escolher aquela que está a executar (figura 5.1).

Para testar a qualidade do modelo genérico mostramos aos utilizadores 50 músicas que estejam dentro dos respectivos intervalos calculados de acordo com o que foi decidido na secção anterior. Embora sejam apresentadas aos utilizadores um total de 50 músicas, em cada instante apenas uma música está visível para o utilizador.

De modo a que os utilizadores tenham uma ideia da música sugerida, tocamos uma amostra de 30 segundos para cada música, em detrimento do total de cada música para diminuir o tempo de escolha dos utilizadores.



Figura 5.1: Selecção da actividade.

Na aplicação desenvolvida o utilizador deverá apenas seleccionar as músicas que considera adequadas para cada uma das actividades. Sempre que uma música não lhe agradar, este deverá pressionar o botão *next* para passar para a próxima música (ver figura 5.2).

De forma a obrigar os utilizadores a ouvirem todas as músicas, estes apenas podem submeter as músicas que consideraram adequadas para cada uma das actividades depois de terem passado por todas as 50 músicas recomendadas. Um exemplo de uma interacção com a aplicação de recolha de dados do modelo genérico pode ser vista na figura 5.2.



Figura 5.2: Selecção de músicas.

5.2 Procedimento de avaliação

A nossa solução baseia-se na construção de um modelo genérico que é inicialmente igual para todos os utilizadores. Depois de uma interacção do utilizador com o modelo genérico este adapta-se às escolhas efectuadas passando assim o modelo genérico para um modelo personalizado.

Desta forma a avaliação foi dividida em dois passos. Numa primeira fase avaliámos o modelo genérico com o intuito de perceber se este é suficientemente flexível para recomendar músicas que estejam de acordo com algumas das preferências dos possíveis utilizadores do sistema, resolvendo o problema do *Cold Start*.

Na segunda fase, pedimos a alguns utilizadores que tinham previamente interagido com as músicas recomendadas pelo modelo genérico que usassem novamente o sistema. No entanto desta vez as novas músicas recomendadas tiveram em conta as escolhas efectuadas pelos utilizadores aquando da sua interacção com o modelo genérico.

As novas músicas apresentadas foram calculadas utilizando o mesmo método de cálculo utilizado na criação dos hiper rectângulos usados no modelo genérico. Aqui o nosso objectivo foi o de tentar perceber se o modelo se ajustava de acordo com as preferências do utilizador sendo assim de esperar um aumento no número de músicas que este consideraria adequadas para cada actividade.

5.3 Resultados da avaliação do modelo genérico

A avaliação do modelo genérico foi efectuada de duas formas distintas. Foi feita uma avaliação online na qual divulgámos a página web descrita na secção 5.1 através das redes sociais facebook, google+ bem como através de email. A segunda avaliação consistiu numa avaliação presencial onde nos encontrámos com os utilizadores para estes testarem o modelo genérico.

No total obtivemos 21 respostas através da avaliação online e 10 com os testes presenciais. Na sua maioria os utilizadores que testaram o modelo genérico, eram do género masculino. Das 21 pessoas dos testes online 15 são do género masculino sendo as restantes do género feminino. No caso dos testes presenciais apenas 2 utilizadores representam o género feminino e os restantes 8 o género masculino.

A idade da maioria dos utilizadores está compreendida entre os 22 e 29 anos, estando esta faixa etária representada com 81% dos utilizadores. Entre os 16-21 anos temos 5 utilizadores o que equivale a 16% do total de utilizadores, com apenas 1 utilizador representando uma percentagem de 3% temos uma pessoa com a idade entre os 30 e os 39 anos.

Apesar de nos testes online nem todas os utilizadores terem respondido a todas as actividades, não existe uma grande diferença no número de respostas por actividade: para a actividade *walking* foram conseguidas 20 respostas, a actividade *relaxing* conseguiu 16 enquanto que as actividades *running* e *shopping* obtiveram 14 respostas, finalmente temos a actividade *sleeping* que adquiriu 15 respostas. No caso dos testes efectuados presencialmente todas as actividades tiveram o mesmo número de respostas uma vez que todos os utilizadores responderam a todas as actividades.

Comparando os testes efectuados presencialmente e online conseguimos verificar uma

grande diferença no número de músicas escolhidas em média por cada utilizador. Em média os utilizadores que responderam online consideraram apenas entre 4 a 5 músicas como correctas para cada actividade. Por sua vez nos testes presenciais este valor subiu para uma média de 11 a 12 músicas seleccionadas por actividade. Esta diferença pode estar relacionada com o facto de nos testes presenciais as pessoas sentirem-se observadas e tenderem a responder de forma mais atenta e concentrada. Os resultados obtidos para a média do número de músicas seleccionadas por actividade podem ser vistos na figura 5.3.

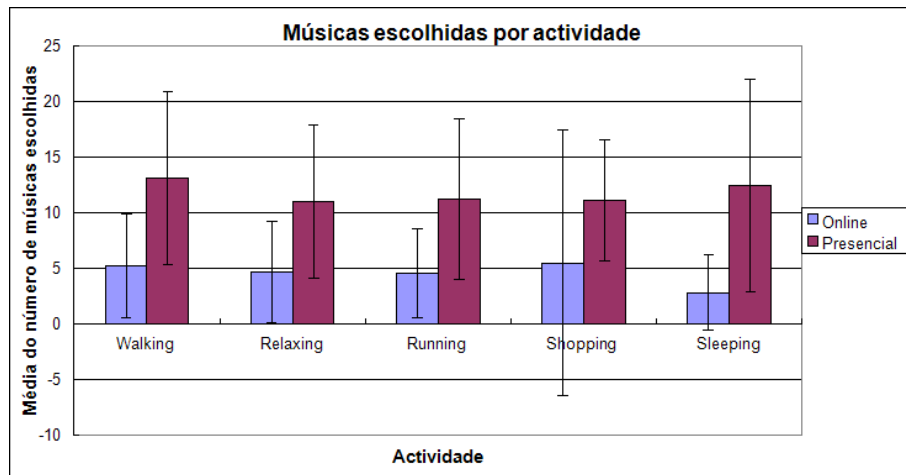


Figura 5.3: Média de músicas escolhidas por actividade.

Retirando o caso da actividade *sleeping* para os utilizadores que fizeram o teste online, conseguimos verificar que a recomendação efectuada pelo modelo genérico está a recomendar em média um número de músicas adequadas para cada utilizador de forma uniforme para todas as actividades. Isto demonstra que o método utilizado para o cálculo dos intervalos de cada característica que permite escolher as músicas a recomendar está a ser constante para todas as actividades.

Depois de seleccionadas as músicas para cada uma das actividades, cada utilizador teve que responder dizendo se concordava ou não com a seguinte afirmação; "As músicas sugeridas estavam ajustadas para a actividade Y." onde 'Y' correspondia à actividade para a qual o utilizador acabou de seleccionar músicas. Para responder a esta afirmação cada utilizador tinha à sua disposição 5 hipóteses; discordo completamente, discordo, não concordo nem discordo, concordo e concordo completamente. Apesar desta ser uma avaliação subjectiva o resultado da mesma permite verificar o grau de satisfação da recomendação efectuada para cada utilizador. Os resultados obtidos tanto para os testes realizados online como para os realizados presencialmente podem ser encontrados nas figuras 5.4 e 5.5 respectivamente.

Através dos dados apresentados conseguimos perceber que em ambos os testes efectuados a actividade *walking* foi aquela que obteve um menor grau de satisfação no seu



Figura 5.4: Resultados online da satisfação da recomendação.

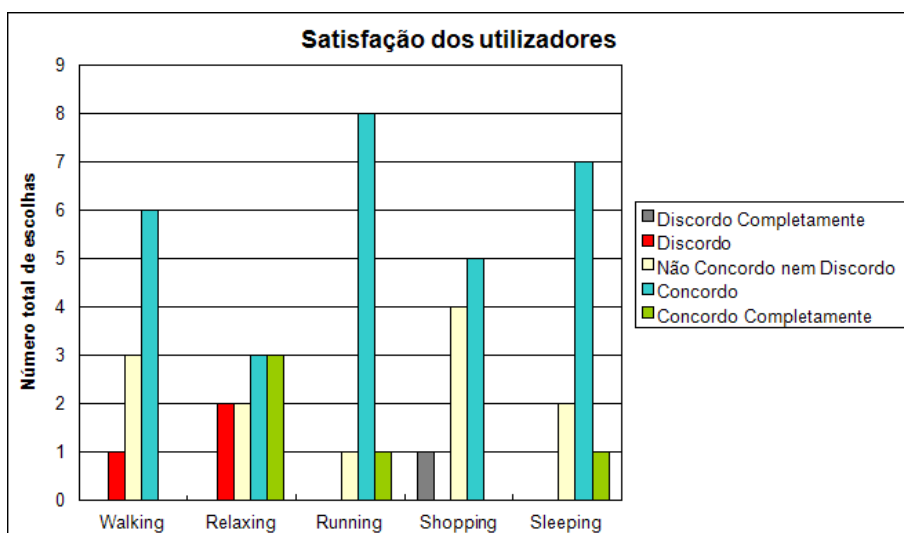


Figura 5.5: Satisfação da recomendação do modelo genérico presencial.

processo de recomendação. Em contrapartida a actividade *running* foi a que conseguiu um maior grau de satisfação.

Utilizando a escala usada para classificar o grau de satisfação de cada utilizador para cada uma das recomendações efectuadas, podemos classificar as opções discordo completamente e discordo como *feedback* negativo por parte do utilizador. A opção não concordo nem discordo é vista como *feedback* neutro, e as opções concordo e concordo completamente como *feedback* positivo. Desta forma usando esta separação verificamos um aumento de 13% no *feedback* positivo fornecido pelos utilizadores que realizaram o teste presencialmente. As percentagens referentes ao *feedback* fornecido pelos utilizadores estão apresentadas na figura 5.6.

Apesar de em média o número de músicas escolhidas pelos utilizadores que efectuaram o teste online ser pequeno, conseguimos verificar que no geral a sua satisfação com as músicas recomendadas é positiva. O mesmo é verificado com os utilizadores que rea-

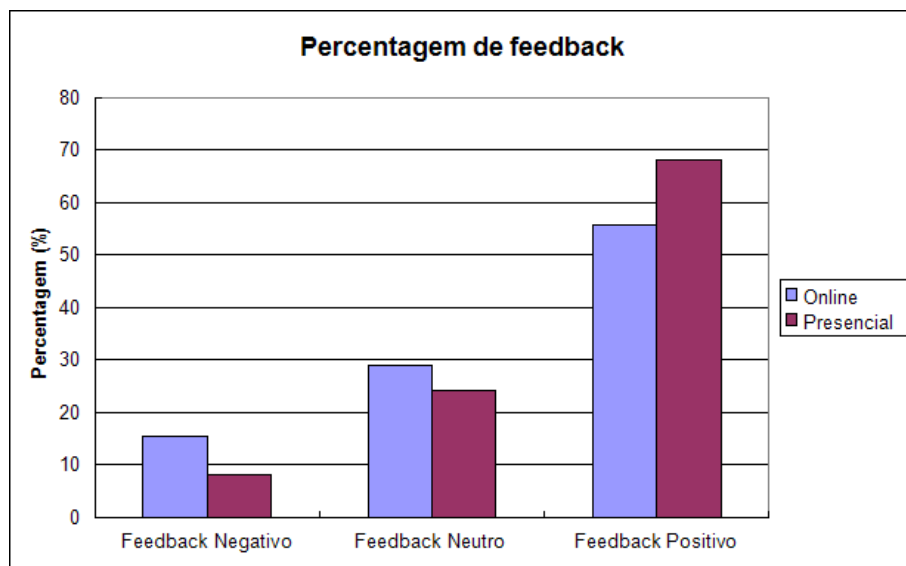


Figura 5.6: Percentagem do feedback positivo, neutro e negativo da recomendação do modelo genérico.

lizaram os testes presencialmente. Isto demonstra que no geral a recomendação satisfaz os interesses dos utilizadores no entanto por alguma razão os utilizadores online escolheram em média um número de músicas bastante mais reduzido. A razão apontada pelos utilizadores sobre o sucedido, está relacionada com o facto de estes concordarem com a recomendação diferenciada de músicas mais calmas ou mais agitadas para cada uma das actividades mas não gostarem das músicas em si.

5.4 Resultados da avaliação do modelo personalizado

Com o objectivo de verificarmos se o modelo genérico se consegue moldar para um modelo personalizado, depois dos testes efectuados anteriormente para o modelo genérico foi testado o modelo personalizado. Estes testes foram efectuados com os 10 utilizadores que responderam ao teste anterior de forma presencial.

Os testes referentes ao modelo personalizado são efectuados da mesma forma que os testes anteriores, no entanto os limites das diferentes características musicais foram modificados tendo em conta as escolhas feitas por cada utilizador durante a sua interacção com o modelo genérico.

O cálculo dos intervalos para o modelo personalizado é efectuado da mesma maneira que o cálculo utilizado no modelo genérico, fazendo uso da mediana do valor das características musicais $\pm 20\%$ do desvio padrão. No entanto as 100 músicas utilizadas para definir os intervalos são diferentes das músicas utilizadas no modelo genérico. As músicas escolhidas durante os testes pelos utilizadores vão ser utilizadas para substituir aquelas presentes no final da lista das 100 músicas mais escolhidas pelos utilizadores, aquando da associação de músicas a actividades.

Assim, inicialmente as músicas escolhidas pelos utilizadores vão sendo inseridas no fundo da lista das 100 músicas utilizadas para o cálculo dos limites. Estas são inseridas no fim da lista uma vez que as últimas músicas representam as menos escolhidas pelos utilizadores de entre o top 100 de músicas seleccionadas pelos utilizadores. Quando todas as músicas que tiverem sido utilizadas na criação do modelo genérico não pertencerem às 100 músicas utilizadas no cálculo dos intervalos então o processo de "reciclagem" das músicas segue uma ordem FIFO.

A retirada das músicas utilizando uma ordem FIFO é importante uma vez que é garantido que as músicas mais recentes, que melhor representam os gostos do utilizador no momento actual, são aquelas que permanecem mais tempo para o cálculo dos novos intervalos. Isto permite fornecer uma recomendação mais actual uma vez que tem em conta os gostos mais recentes dos utilizadores.

Os resultados obtidos durante os testes efectuados utilizando o modelo personalizado podem ser observados na figura 5.7.

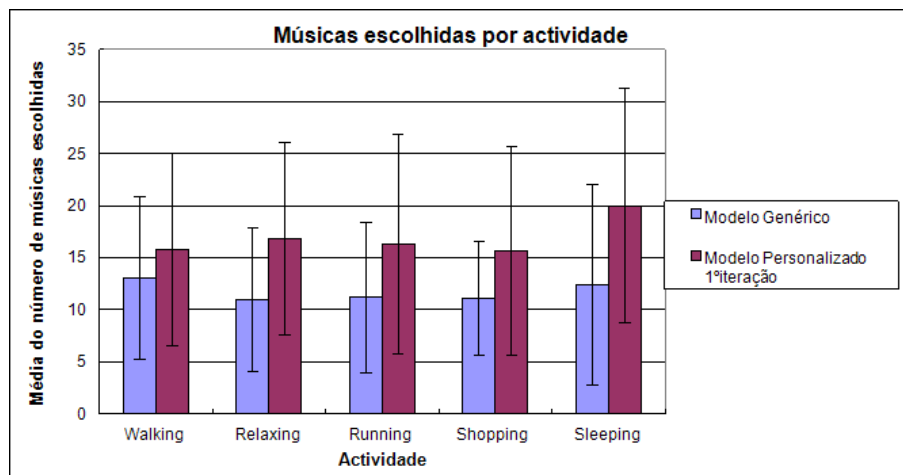


Figura 5.7: Média de músicas escolhidas por actividade.

No gráfico apresentado na figura 5.7 percebemos que existiu em média uma melhoria na recomendação quando utilizado o modelo personalizado. Este consegue em média recomendar mais de 15 músicas por actividade.

Depois de efectuada uma avaliação a todos os utilizadores usando o modelo personalizado de cada um, voltámos a fazer uma nova iteração com as músicas escolhidas durante a primeira interacção com o modelo personalizado. Este teste teve como objectivo verificar se existe uma melhoria no número de músicas consideradas como adequadas para cada utilizador com a utilização do modelo personalizado. Infelizmente não foi possível efectuar estes testes com todos os 10 utilizadores que experimentaram o seu modelo personalizado, tendo esta 2ª iteração sido realizada por apenas 5 dos 10 utilizadores.

Na figura 5.8 está presente a evolução no número de músicas escolhidas por cada um dos utilizadores que utilizaram o modelo personalizado pela segunda vez.

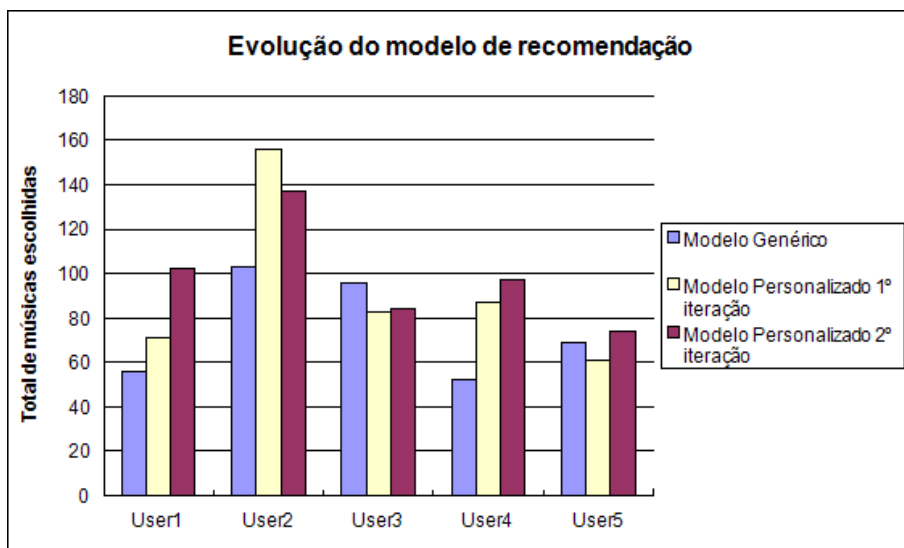


Figura 5.8: Evolução do número de músicas escolhidas utilizando o modelo personalizado.

A partir destes dados verificamos que nos dois casos onde se verificou um menor número de músicas escolhidas durante a utilização do modelo genérico, existiu um aumento contínuo na recomendação durante o uso do modelo personalizado. Nos restantes casos essa evolução não foi contínua, no entanto apenas num caso o modelo personalizado não conseguiu recomendar mais músicas que o modelo genérico.

Os resultados obtidos permitem verificar também que no geral existiu uma melhoria no processo de recomendação existindo um aumento do número total de músicas consideradas como correctas passando de 376 músicas no modelo genérico para 458 na 1ª iteração com o modelo personalizado e 494 na 2ª iteração com o modelo personalizado (figura 5.9). Esta melhoria reflectiu-se num aumento de 28% no número de músicas consideradas como correctas desde a utilização do modelo genérico até à 2ª iteração do modelo personalizado.

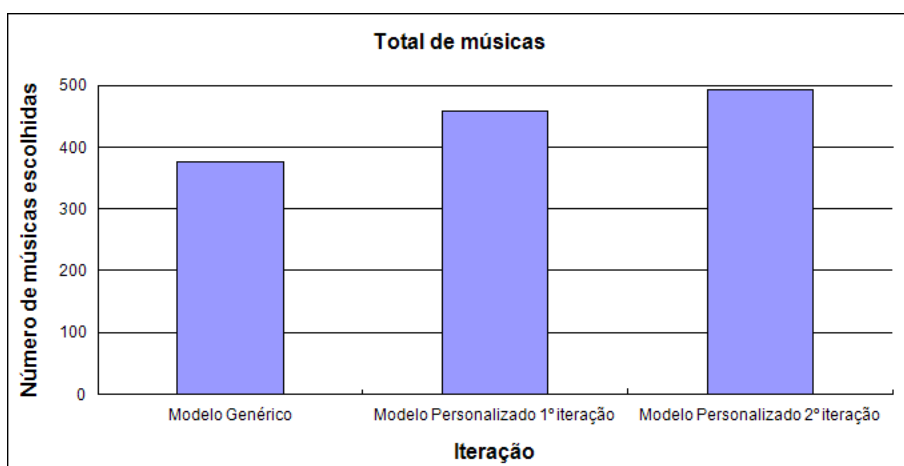


Figura 5.9: Evolução do número de músicas escolhidas utilizando o modelo personalizado.

Em relação à satisfação da recomendação, na figura 5.10 conseguimos verificar a existência de uma diminuição da insatisfação da recomendação comparando com aquela

presente no modelo genérico (figura 5.5). Em relação ao *feedback* positivo verificamos um decréscimo de apenas um voto. O aumento verificou-se no *feedback* neutro que foi escolhido em 15 ocasiões de entre 50. Em comparação com o modelo genérico, no geral os resultados presentes na figura 5.10 demonstram a capacidade do modelo personalizado satisfazer de melhor forma os gostos dos utilizadores.



Figura 5.10: Satisfação da recomendação do modelo personalizado dos 10 utilizadores.

5.5 Síntese

Os resultados obtidos dos testes realizados demonstraram que o modelo genérico em alguns casos é capaz de recolher um conjunto de músicas que reflectem os interesses dos utilizadores e com as escolhas efectuadas pelos mesmos é possível personalizar esse mesmo modelo de forma a aumentar o número de músicas consideradas como correctas para cada actividade.

Os testes online demonstraram uma menor performance no modelo genérico. Acreditamos que esta quebra em relação aos resultados obtidos dos testes realizados presencialmente está relacionada com o facto de os testes necessitarem de algum tempo despendido pelo utilizador de forma a ouvir as músicas para decidir se gosta ou não das mesmas. O tempo despendido pelos utilizadores online para decidir que músicas gostam em cada actividade é em média de 4 minutos por actividade. Por sua vez os utilizadores que responderam presencialmente demoraram uma média de 7 minutos por actividade o que corresponde a um aumento de 47% no tempo despendido durante a realização dos testes.

Foi também possível verificar que o modelo personalizado consegue efectuar uma recomendação mais eficaz que a efectuada pelo modelo genérico, este aumento verificou-se tanto no número de músicas consideradas como correctas para cada utilizador bem como no grau da sua satisfação com a recomendação.

No entanto estamos cientes que o número de testes deveria ser maior para conseguir tirar melhores conclusões. No entanto devido à dificuldade de arranjar utilizadores para efectuarem os testes presencialmente não foi possível efectuar mais testes.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Sumário da Dissertação

Com a chegada das tecnologias da música digital e da Internet, uma grande quantidade de conteúdo musical está disponível para milhões de utilizadores à volta do mundo. Com os milhões de artistas e músicas existentes no mercado, começa a ser difícil para os utilizadores procurarem músicas que lhes agradem. Além disso, a grande quantidade de música disponível abriu oportunidade para o surgimento de novas técnicas de recuperação e recomendação musical.

Motivados por estes problemas e pelo crescente interesse nesta área, desenvolvemos uma solução para a recomendação de músicas que tem em conta a actividade que o utilizador está a realizar. Para atingir este objectivo começámos por pesquisar o que se tem feito nesta área, tentando perceber quais são os problemas existentes e como é que os podemos resolver.

Identificámos o *cold start problem* e o *popularity bias* como os principais problemas dos sistemas de recomendação e foi com base neles que desenvolvemos a nossa solução. A solução aqui apresentada baseia-se num sistema de recomendação híbrido em cascata que usa a actividade que o utilizador está a realizar e o conteúdo das músicas com o objectivo de fazer uma recomendação eficaz.

Para identificar um possível paralelismo entre as músicas e as diferentes actividades fizemos um conjunto de testes para recolher informação sobre as preferências musicais de diferentes utilizadores para cada uma das actividades. Estes testes iniciais permitiram fazer um paralelismo entre as músicas e as actividades o que resultou na criação de um modelo genérico, que é utilizado inicialmente para recomendar músicas a todos os utilizadores.

A recomendação utiliza um conjunto de características musicais que foram escolhidas tendo em conta a sua capacidade para distinguir as diferentes actividades. As características utilizadas foram a *energy*, *accousticness*, *loudness* e o *tempo*. Para decidir as

músicas a recomendar em cada actividade são utilizados intervalos representativos dos valores que cada uma das características musicais pode tomar dentro de cada actividade.

O cálculo destes intervalos foi objecto de estudo tendo sido testados vários métodos para o seu cálculos. Os primeiros métodos testados foram a utilização da média \pm desvio padrão e os percentis de 10 e 90% para a definição dos valores mínimos e máximos dos intervalos. No entanto verificou-se que estes não eram adequados e por isso os métodos média e mediana \pm uma percentagem do desvio padrão foram utilizados. As percentagens testadas para o valor do desvio padrão foram de 15, 20, 25, e 30%. No final o método utilizado no processo de recomendação e o que revelou os melhores resultados foi a utilização da mediana \pm 20% do desvio padrão, utilizando as características musicais das últimas 100 músicas ouvidas para cada actividade.

Posteriormente foram elaborados um conjunto de testes. Os primeiros para verificar se a recomendação do modelo genérico era suficiente para uma recomendação inicial e por último foram feitos dois testes para verificar se o modelo personalizado conseguia obter melhores resultados que o modelo genérico.

Obtidos os resultados verificámos que o modelo genérico consegue recomendar um conjunto de músicas satisfatório que permite a criação do modelo personalizado que por sua vez apresentou uma melhoria em relação ao modelo genérico.

6.2 Conclusões Finais e Contribuições

Com este trabalho foi possível perceber os desafios e a complexidade inerente aos sistemas de recomendação. Foram adquiridos vários conhecimento referentes a esta área tais como os diferentes tipos de recomendação e os desafios que esta disciplina apresenta. Ao mesmo tempo foi importante para perceber o que existe hoje em dia nos sistemas de recomendação com foco na música.

Os resultados obtidos foram satisfatórios, pois conseguimos verificar que a utilização de um modelo genérico para todos os utilizador pode ser benéfico para uma recomendação inicial. A capacidade do modelo genérico para uma recomendação inicial é importante porque permite colmatar um dos grandes problemas da recomendação conhecido como *Cold Start Problem*.

A utilização dos intervalos para distinguir as músicas a recomendar em cada actividade mostrou-se um método capaz de se moldar aos gostos dos diferentes utilizadores. Isto foi visível com a melhoria no número de músicas recomendadas durante as passagens do modelo genérico para o personalizado e através da segunda iteração feita utilizando as músicas recomendadas pelo modelo personalizado.

Estes intervalos permitem assim recomendar músicas tendo por base as suas características musicais o que permite ultrapassar outro dos problemas da recomendação conhecido como *popularity bias*. Desta forma conseguimos atingir os objectivos a que nos

propusemos, apresentando uma solução que não apresenta dificuldade em recomendar músicas a novos utilizadores e que consegue recomendar qualquer tipo de música seja ela conhecida ou não. Este último aspecto vai de encontro a outro objectivo proposto que se refere à recomendação de músicas desconhecidas do utilizador mas que vão ao encontro dos seus gostos. Este aspecto acabou por ser evidenciado nos testes efectuados com os utilizadores. No entanto, o objectivo de incluir o conceito de frescura não está presente na nossa solução uma vez que não está a ser tido em conta o número de vezes que uma música já foi recomendada a cada utilizador, bem como a última vez que a recomendação foi feita.

6.3 Trabalho Futuro

Tendo em conta que não conseguimos concretizar a utilização do conceito de frescura das músicas na nossa solução, a inclusão deste conceito seria um aspecto importante a considerar no futuro. Com este conceito presente na solução seria possível atribuir uma pontuação de frescura que serviria para ter uma recomendação mais diversificada, e promover a descoberta de novas músicas.

Actualmente o processo de recomendação está a ser feito utilizando um hiper rectângulo que é representado pelos intervalos que cada característica musical pode tomar. Para captar mais rapidamente os gostos dos utilizadores durante o processo de recomendação, poderia ser interessante escolher um ponto dentro do hiper rectângulo que seria utilizado como novo centro dessa "figura geométrica". Com um novo centro calculado existiria uma translação do hiper rectângulo, em que no pior caso os intervalos iriam ter 50% de novos valores. A figura 6.1 representa a ideia aqui recomendada para uma dimensão (característica).

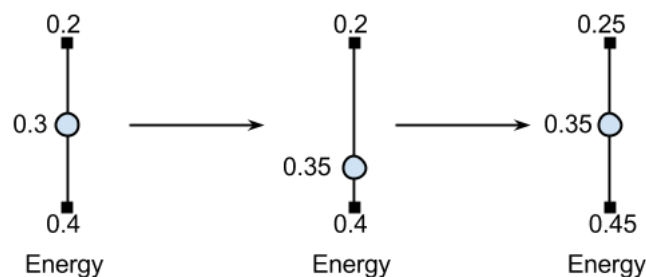


Figura 6.1: Novo cálculo de intervalos.

Este método poderia ser utilizado esporadicamente (p.ex em cada 5 iterações com o sistema) para detectar novos interesses. Durante a recomendação, no caso de muitas das músicas aceites por parte do utilizador, terem as suas características musicais presentes na nova área coberta pela translação imposta pelos intervalos, então os intervalos resultantes destas escolhas iriam reflectir a deslocação do hiper rectângulo.

Em vez de calcular um novo centro para cada intervalo, poderia também ser interessante escolher aleatoriamente dois novos pontos dentro do intervalo actual para representarem um novo máximo e mínimo. Estes novos pontos estariam sempre dentro do intervalo actual, a restrição de alguns valores dentro do intervalo tal como no caso anterior permitiriam verificar mais rapidamente em que sentido apontam os gostos musicais dos utilizadores.

A informação relativa ao nome dos artistas das músicas mais escolhidas pelos utilizadores poderia ser também usada no processo de recomendação. Com estes artistas seria interessante recomendar novas músicas dos artistas mais escolhidos mas que ainda não tivessem sido recomendadas. Com os artistas mais escolhidos seria também possível procurar por artistas semelhantes o que seria interessante para o utilizador conhecer novas músicas. Tal como o nome dos artistas, a informação dos géneros das músicas mais escolhidas para cada actividade também poderia ser utilizada.

No entanto em relação aos artistas é necessário ter cuidado quando utilizada essa informação de forma a não cair no problema do *popularity bias*. Uma vez que seriam recomendadas mais músicas dos artistas que conhece, a apresentação de músicas ainda não conhecidas pelo utilizador poderia sofrer com isso.

Bibliografia

- [1] Antti Oulasvirta, Janne P. Hukkinen, and Barry Schwartz. When more is less: The paradox of choice in search engine use. pages 516–523, 2009. [cited at p. 2, 26]
- [2] Christopher Avery and Richard Zeckhauser. Recommender systems for evaluating computer messages. *Commun. ACM*, pages 88–89, 1997. [cited at p. 5]
- [3] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitisma. A review of audio fingerprinting. *J. VLSI Signal Process. Syst.*, pages 271–284, 2005. [cited at p. 6]
- [4] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, pages 133–141, 2006. [cited at p. 6]
- [5] Elaine Rich. Readings in intelligent user interfaces. pages 329–342, 1998. [cited at p. 7]
- [6] Anind Kumar Dey. Providing architectural support for building context-aware applications. 2000. AAI9994400. [cited at p. 7]
- [7] Marius Kaminskis and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, pages 89–119, 2012. [cited at p. 8]
- [8] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002. [cited at p. 8]
- [9] Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. *Proceedings of the 20th ACM international conference on Multimedia - MM '12*, page 10, 2012. [cited at p. 9, 22, 26]
- [10] Yajie Hu and Mitsunori Ogihara. Nexttone player: A music recommendation system based on user behavior. *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 103–108, 2011. [cited at p. 11, 22]
- [11] Jeff Mascia. Lifetrak : Music In Tune With Your Life Categories and Subject Descriptors. *1st ACM International Workshop on Human-Centered Multimedia*, pages 25–34, 2006. [cited at p. 12, 22]
- [12] Anupriya Ankolekar and Thomas Sandholm. Foxtrot: a soundtrack for where you are. *Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications*, pages 26–31, 2011. [cited at p. 14, 22, 23]
- [13] Hao Liu, Jun Hu, and Matthias Rauterberg. Music Playlist Recommendation Based on User Heartbeat and Music Preference. *2009 International Conference on Computer Technology and Development*, pages 545–549, 2009. [cited at p. 15, 22]

- [14] L Bernardi, C Porta, and P Sleight. Cardiovascular, cerebrovascular, and respiratory changes induced by different types of music in musicians and non-musicians: the importance of silence. *Heart (British Cardiac Society)*, pages 445–52, 2006. [cited at p. 16]
- [15] W E Knight and N S Rickard PhD. Relaxing music prevents stress-induced increases in subjective anxiety, systolic blood pressure, and heart rate in healthy males and females. *Journal of music therapy*, pages 254–72, 2001. [cited at p. 16]
- [16] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Karl-Heinz Lüke, and Roland Schwaiger. Incarmusic: Context-aware music recommendations in a car. *EC-Web 2011 - International Conference on Electronic Commerce and Web Technologies*, 2011. [cited at p. 16, 22]
- [17] Dongmin Shin, Jae-won Lee, Jongheum Yeon, and Sang-goo Lee. Context-Aware Recommendation by Aggregating User Context. *2009 IEEE Conference on Commerce and Enterprise Computing*, pages 423–430, 2009. [cited at p. 17, 22]
- [18] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latenttopic sequential patterns. pages 131–138, 2012. [cited at p. 18, 22]
- [19] J Lee. Music for my mood: A music recommendation system based on context reasoning. *Smart Sensing and Context*, pages 190–203, 2006. [cited at p. 19, 22]
- [20] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Emilia Gómez, and Perfecto Herrera. Content-based music recommendation based on user preference examples Categories and Subject Descriptors. 2010. [cited at p. 21, 22]
- [21] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software: An update; sigkdd explorations, volume 11, issue 1. 2009. [cited at p. 36]
- [22] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. [cited at p. 36]