



UFPE
Centro de Informática

Graduação em Ciência da Computação

Transcrição Automática de Acordes com Fontes Híbridas

Arthur Lubambo Peixoto Accioly

Trabalho de Graduação

Recife
10 de Julho de 2017

UFPE
Centro de Informática

Arthur Lubambo Peixoto Accioly

Transcrição Automática de Acordes com Fontes Híbridas

*Trabalho apresentado ao Centro de Informática da UFPE
como requisito parcial para obtenção do grau de Bacharel
em Ciência da Computação.*

Orientador: *Prof. Giordano Ribeiro Eulalio Cabral*

Recife
10 de Julho de 2017

Agradecimentos

Agradeço em primeiro lugar a Deus por ter me proporcionado saúde, força e determinação para superar as dificuldades e continuar em busca dos meus objetivos.

Aos meu pais, avós, irmãos, cunhados e demais familiares por entenderem minhas ausências e serem uma base de amor, incentivo e apoio incondicional.

Aos meus amigos por me proporcionarem momentos de alegrias, descontração e compreensão ao longo desse processo.

A minha noiva, com quem eu desejo partilhar todos os momentos da vida, e sua família por terem estado ao meu lado durante todos esses anos, incentivando, apoiando, acreditando e acima de tudo amando.

A Universidade Federal de Pernambuco, ao Centro de Informática e aos professores que cruzaram meu caminho ao longo da vida por proporcionarem um excelente ambiente de aprendizagem e curiosidade e por estarem sempre atentos às necessidades dos alunos.

Ao meu orientador Giordano Ribeiro Eulalio Cabral pelo suporte, paciência e incentivo, também pelo tempo e correções dedicados a este trabalho.

A Pedro Thiago de V. S. R. Araújo por fornecer detalhes adicionais para a implementação deste trabalho.

Aos meus animais de estimação, em especial meu cachorro Dinho, por permanecer ao meu lado dias e noites durante a realização deste trabalho me fazendo sentir o ser humano mais especial do mundo.

E a todos que direta ou indiretamente contribuíram para a minha formação, o meu muito obrigado.

”Nós não temos a chance de fazer muitas coisas, e cada uma deve ser realmente excelente. Porque esta é a nossa vida. A vida é breve, e então você morre, sabe? E todos nós escolhemos o que fazer com as nossas vidas. Então é melhor que seja muito bom. É melhor valer a pena.”

—STEVE JOBS

Resumo

Na computação tem-se estudado diversas formas de fazer a transcrição harmônica de uma música de maneira automática, pois esta é uma tarefa bastante difícil de ser feita manualmente e requer especialistas bem treinados. Os métodos mais comuns usam exclusivamente o áudio para tentar extrair as informações musicais. No entanto, recentemente, um trabalho de graduação do Centro de Informática utilizou de cifras pra fazer o alinhamento entre os acordes e os segmentos de áudio.

O presente trabalho fez uma progressão deste estudo, agora tratando o problema como transcrição, já que muitas cifras não possuem a sequência de acordes compatível com o áudio da música. Para melhorar o desempenho utilizou-se aprendizagem de máquina para aprender uma função de similaridade entre um acorde e um segmento de áudio. A proposta obteve melhorias em vários aspectos, porém precisa melhorar a detecção de tonalidade (uma etapa importante do processo).

Palavras-chave: Computação Musical, Transcrição Automática de Acordes , Aprendizagem de Máquina

Abstract

In computing has been studied several ways to do harmonic transcriptions of a song automatically, because it is a very difficult task to do manually and requires well-trained specialists. The most common methods uses exclusively the audio to attempt to extract the musical information. However, recently, an undergraduate work from the Computer Center of UFPE that used chords transcriptions to make the alignment between the chords and the audio segments.

The present work has made progress in this study, now treating the problem as transcription, since many chord's transcriptions does not have a chord sequence compatible with the audio of the song. To improve performance it has used machine learning to learn a similarity function between a chord and an audio segment. The proposal obtains improvements in several, but it needs improve tone detection.

Keywords: Musical Computing, Automatic Chord Transcription, Machine Learning

Sumário

1	Introdução	1
2	Problema	2
2.1	Transcrição Harmônica	2
2.2	Alinhamento de Acordes	2
2.3	Problema Híbrido	2
2.4	Problemas Intermediários	3
3	Estado da Arte	4
3.1	Pitch Class Profile	4
3.2	Métodos de Transcrição	4
3.2.1	Pré processamento	4
3.2.2	Reconhecimento de Acorde	5
3.2.3	Pós processamento	5
3.3	Alinhamento de Acordes com Fontes Híbridas	5
4	Proposta de Solução	7
4.1	Coleta dos dados	7
4.1.1	Spotify	7
4.1.2	CifraClub	7
4.2	Etapas	7
4.3	Detecção de Tonalidade	8
4.4	Similaridade	8
4.4.1	Produto Interno	8
4.4.2	Aprendizagem de Máquina	9
4.5	Templates	9
4.5.1	Parser	9
4.5.2	Construindo Template	10
4.6	Modelo	10
4.6.1	Hidden Markov Models (HMM)	10
4.6.1.1	Estados Ocultos	11
4.6.1.2	Matriz de Transição	11
4.6.1.3	Matriz de Emissão	12
4.6.1.4	Probabilidade a priori	12
4.6.2	Viterbi	12

5	Avaliação	13
5.1	Detecção de Tonalidade	13
5.2	Experimentos	14
5.3	Similaridade com Multi Layer Perceptron	14
5.3.1	Comparação Similaridade MLP com Produto Interno	15
5.4	Experimentos com o Modelo SA	15
5.4.1	Modelo SA com suavização (SAs)	17
5.5	Experimento com o Modelo AE	17
5.6	Comparação Geral	18
5.7	Considerações	18
6	Conclusões	19
6.1	Resultados	19
6.2	Trabalhos Futuros	19

Lista de Tabelas

5.1	Músicas usadas na avaliação.	13
5.2	Acerto da Tonalidade vs Método	14
5.3	Método vs Taxa de Acerto	14
5.4	Música vs Área abaixo da Curva ROC	15
5.5	Comparação da Precisão MLP e PI	15
5.6	Comparação da Recall MLP e PI	16
5.7	Comparação da Métrica-F MLP e PI	16
5.8	Tabela de comparação entre SAss e SAcS.	17
5.9	Tabela de comparação AEss vs AECS com ambos usando MLP.	17
5.10	Comparação Geral	18

CAPÍTULO 1

Introdução

A transcrição musical automática é uma área de estudo na computação que se baseia em extrair informações musicais a partir do áudio gravado.

Existem vários tipos de transcrição: melódica, harmônica, rítmica. Este trabalho lida com a transcrição harmônica.

A maior parte dos algoritmos usam informações exclusivamente do áudio, processando-as de diversas formas.

Recentemente num trabalho de graduação [dVSRA16] foi feito o uso de outra fonte de dados além do áudio. Foi integrado à análise tradicional o uso de informações de cifras, essa nova abordagem muda a forma de como o problema é visto. [dVSRA16] analisou o problema na ótica de um alinhamento de acordes, isto é, dada uma sequência de acordes vinda da cifra o algoritmo alinha os segmentos de áudio com seus respectivos acordes.

Como muitas cifras não tem a sequência compatível com a música, o trabalho corrente tratou o problema não como alinhamento e sim como transcrição de acordes. Sendo assim, foi proposto um algoritmo mais robusto à sequência de cifras mal formadas.

Foram feitas, então, modificações na abordagem de [dVSRA16] para permitir transições mais flexíveis entre a sequências de acordes. Para evitar que o aumento dessa flexibilidade degrade o modelo, realizou-se uma modificação na função de similaridade entre o acorde e o segmento de áudio. Além disso, aplicou-se o uso de aprendizagem de máquina (mais especificamente o Multi Layer Perceptron) para aprender a função de similaridade.

O uso do MLP ocasionou melhorias nas métricas de desempenho; o uso das transições mais flexíveis melhorou o resultado para algumas músicas com problemas na sequência, mas degradou um pouco o desempenho em músicas com a sequência de cifras bem formadas. Um ponto que deixou a desejar foi a detecção da tonalidade, o que é feito em uma das etapas do algoritmo.

No Capítulo 2 será caracterizado o problema de transcrição harmônica, e problemas secundários no processo. No Capítulo 3 será visto alguns métodos que tratam do problema, inclusive o de [dVSRA16] que serviu de base para o trabalho. No Capítulo 4 serão mostradas as etapas da coleta de dados, o pré processamento e as modificações do método base. O Capítulo 5 contém os resultados e a avaliação do modelo proposto. No capítulo 6 é feita uma análise do resultado do modelo, e são propostas sugestões para trabalhos futuros.

CAPÍTULO 2

Problema

Neste capítulo será feita a caracterização do problema estudado. Será visto que se trata de um problema de natureza híbrida, isto é, uma mistura entre transcrição harmônica de uma música e o alinhamento de acordes com o tempo.

2.1 Transcrição Harmônica

A Transcrição Harmônica de uma música consiste em enumerar a sequência de acordes e seus respectivos tempos. Geralmente tenta-se processar e analisar o áudio ao longo do tempo para estimar os acordes em seu respectivo momento.

2.2 Alinhamento de Acordes

O problema do alinhamento de acordes consiste em relacionar cada acorde de uma sequência dada ao seu respectivo tempo. Por exemplo: Dada uma música com a seguinte sequência de acordes:

"C, F, C, F, C, G, C"

O algoritmo que simplesmente diz a duração que cada acorde terá, pois ele já supõe que a sequência está correta.

2.3 Problema Híbrido

Este trabalho fez a integração de sequências de acordes disponíveis no CifraClub com a análise de áudio da API do Spotify. Desta forma temos dados musicais de duas naturezas. Os dados do CifraClub são textuais de alto nível, já os dados do Spotify são features de cada segmento de áudio.

Visto que o CifraClub é uma plataforma de cifras colaborativa (as músicas são postas por usuários), as transcrições são passíveis de erros. Além disso as músicas tem representações abstratas difíceis de serem interpretadas por uma máquina (Exemplo:...repete refrão 2x intro). Outro problema também comum é a diferença entre versões da mesma música. Isto pode refletir na ordem dos acordes ou trechos da música.

Diante disto é frágil supor que a sequência de acordes obtida será a mesma na música do Spotify. Apesar de em alguns casos um alinhamento simples resolver, o problema de modo geral não é um alinhamento puro, e sim um híbrido entre alinhamento e transcrição. Já que a

sequência obtida no CifraClub tem um bom indicativo de quais acordes tendem a ser executados, existe uma noção mínima de ordem. Sendo assim pode-se juntar os dois mundos: diminuir o espaço de busca para os acordes obtidos no CifraClub e utilizar as características sonoras para estimar o acorde.

Logo, este trabalho investigou um problema de natureza híbrida, algo entre o Alinhamento e a Transcrição Harmônica. Com a caracterização do problema desta forma espera-se obter resultados mais robustos e genéricos, que funcione bem para uma quantidade maior de cifras. Pois até as cifras mal formatadas podem obter bons resultados.

2.4 Problemas Intermediários

Foi apresentada nas seções anteriores o problema principal, entretanto existem outros sub-problemas a serem resolvidos:

- A versão da cifra e da música do Spotify podem estar em tonalidades distintas;

- Os acordes podem ter execuções homófonas;

- As inversões não são detectáveis utilizando o Chroma já que só temos 12 notas compondo a unificação de todas as oitavas, então para os dados oriundos do Spotify acordes originais e suas inversões são equivalentes.

Estado da Arte

Neste Capítulo será feita uma breve descrição dos estudos existentes na área, observando os pontos críticos e restrições de cada método. Basicamente existem duas abordagens principais: Fazer a transcrição somente a partir do áudio da música, e a outra é fazer utilizando uma fonte de áudio e uma de cifras.

3.1 Pitch Class Profile

O Pitch Class Profile é a representação de um sinal numa distribuição de 12 notas da escala cromática. Ele é conhecido como Chroma, e é muito usado em sistemas de Detecção Automática de Acordes (ACE).

3.2 Métodos de Transcrição

A transcrição de um áudio consiste enumerar temporalmente características sonoras de alto nível a partir do áudio. Na música existem transcrições da letra, melódicas, harmônicas, e rítmicas.

Esse trabalho tratou do problema da transcrição harmônica. Então será mostrado alguns outros métodos que abordam o tema.

Os métodos tradicionais, buscam estimar o acorde dentro de uma música analisando somente suas características sonoras no tempo [MMB14]. Esse tipo de tarefa é conhecida como *Automatic Chord Estimation (ACE)*.

O pioneiro no uso do Chroma para o reconhecimento de acordes foi Fujishima (ICMC, 1999) e com o passar do tempo sua abordagem foi sendo aprimorada [PB].

A método consiste em utilizar o chroma e o template do acorde para calcular a similaridade.

O processo de transcrição é feito em 3 etapas: o pré processamento; o reconhecimento de acorde; e o pós processamento.

3.2.1 Pré processamento

O pré processamento visa reduzir o ruído do áudio combinando um segmento com seus vizinhos. O exagero pode dificultar a detecção de fronteira entre um acorde e outro, então isso deve ser bem balanceado. Outra opção é utilizar a mediana ao invés da média pra combinar os segmentos. Veja a Figura 3.1 :

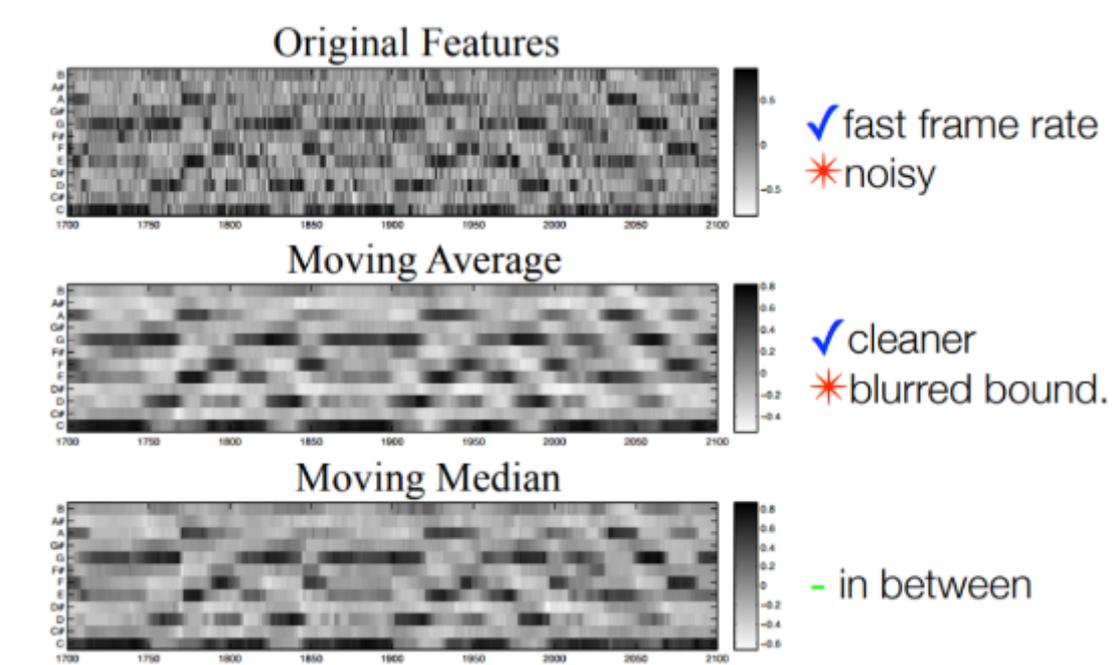


Figura 3.1 Figura de [PB] mostrando as visualizações entre as diferentes formas de pré processar.

3.2.2 Reconhecimento de Acorde

Essa etapa busca encontrar o(s) acorde(s) que maximiza(m) uma função de similaridade (matching). Note que previamente é definido o escopo de acordes (maiores, menores, diminutos, aumentados,...), e nessa etapa testa-se todos os acordes.

O método de matching tem sido feito de diferentes formas.

3.2.3 Pós processamento

A etapa de pós processamento visa gerar a transcrição de acordes final. [PB] usa o algoritmo de Viterbi, buscando encontrar o caminho de transições que maximizam a probabilidade total. (Figura 3.2)

Nesse modelo (Figura 3.2) os acordes seriam i 's e os segmentos seriam os estados observados.

3.3 Alinhamento de Acordes com Fontes Híbridas

No trabalho de graduação [dVSRA16] foi estudada a integração de duas fontes de dados distintas, o Spotify e o CifraClub, com o intuito de fazer o alinhamento de acordes. Foi utilizada a sequência de acordes do CifraClub como base, e os segmentos com o Chroma do Spotify. A junção dessas duas fontes foi uma inovação no estudo de algoritmos da área.

O problema abordado foi o do alinhamento, isto é, dada a sequência de acordes, descobrir

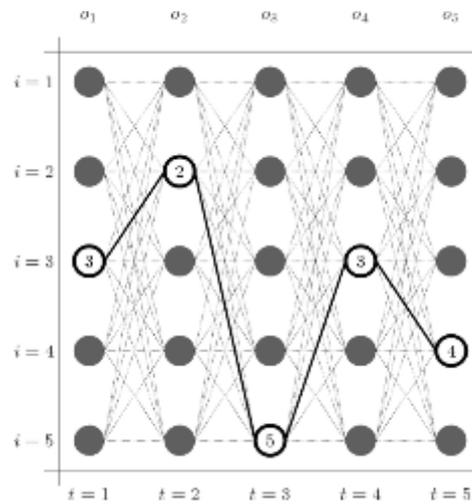


Figura 3.2 Figura mostrando um caminho calculado no algoritmo de Viterbi. (<https://commons.wikimedia.org/wiki/File:Hmm-Viterbi-algorithm-normal.png>)

quais são os momentos das transições. Para isso, [dVSRA16] faz o uso do algoritmo de Viterbi para calcular a sequência mais provável conseqüentemente as transições.

Proposta de Solução

Neste capítulo será explicado o método pesquisado, e como foi feita a junção de dados das duas fontes para fazer a transcrição harmônica automática. Também serão mostradas as estratégias para lidar com os problemas intermediários. A proposta deste trabalho é baseada no trabalho de graduação [dVSRA16] alterando alguns pontos que visam solucionar os problemas encontrados.

A principal vantagem do modelo em relação às abordagens tradicionais é a redução do espaço de busca. Visto que o universo de acordes é reduzido para um número bem pequeno.

Já comparando com o modelo do qual esse trabalho foi baseado [dVSRA16], o principal ponto é tentar ser mais robusto quanto a compatibilidade da Cifra com a Música.

4.1 Coleta dos dados

4.1.1 Spotify

O Spotify é um serviço de streaming de música online. Recentemente ele disponibilizou uma API para outras aplicações coletarem informações das faixas. Para esse trabalho foi utilizado o endpoint chamado "Get Audio Analysis for a Track"[Spo], que retorna características acústicas da faixa. O retorno é em forma de JSON e contém uma lista de segmentos da música, cada segmento possui o Pitch Class Profiles (PCP), também conhecido como Chroma ("chroma features").

O Chroma é constituído de 12 entradas representando cada nota musical, na API do Spotify ele já vem normalizado (O valor máximo das entradas é 1). Neste trabalho focamos no uso dessa feature para fazer a estimativa do acorde.

4.1.2 CifraClub

O CifraClub, um dos sites de cifra mais populares do Brasil, é colaborativo. Isto é, os próprios usuários enviam o conteúdo e a administração simplesmente faz a moderação.

Este site não possui API para outras aplicações acessarem o seu conteúdo, devido a isso foi feito Data Scraping [Wik] para coletar as cifras.

4.2 Etapas

O Método funciona da seguinte forma:

1. Dado um TrackID, coleta-se o Chroma da API do Spotify;
2. Dada a url do CifraClub da Música Correspondente, coleta-se a cifra;
3. Altera-se o tom da cifra para ser compatível com a música do Spotify;
4. Constrói-se uma Matriz de Similaridade entre os Acordes da Música e o Chroma de cada Segmento;
5. Constrói-se uma Matriz de Transição entre os Acordes;
6. Constrói-se a probabilidade a priori de cada Acorde;
7. Executa o algoritmo Viterbi para estimar a transcrição mais provável;

Em algumas etapas foram testadas mais de uma alternativa para o modelo. As seguintes seções irão detalhar melhor as diferentes formas testadas para cada etapa.

4.3 Detecção de Tonalidade

Como foi visto na seção anterior, uma das etapas do método é alterar o tom da cifra pra ser compatível com o da música.

Esse Trabalho usa a seguinte estratégia para estimar o tom, para cada modulação da cifra: Calcula-se o somatório da similaridade máxima de cada segmento da música.

Faz esse procedimento para todos os tons possíveis e assume-se que seja aquele que tem o maior valor de somatório.

4.4 Similaridade

Uma das etapas deste trabalho é estimar o quão similar o chroma de um segmento de um acorde. Para isto foi usado o template do acorde [BP05]. Por exemplo: O template de C (Dó maior) é constituído pelas notas Dó-Mi-Sol que corresponde ao seguinte vetor [1,0,0,0,1,0,0,1,0,0,0,0].

No cálculo da similaridade foram testadas duas formas:

4.4.1 Produto Interno

Uma delas é a mesma proposta num trabalho passado[dVSRA16], que consiste no produto interno entre o Chroma e o Template do acorde equação (4.1).

$$S(c,t) = \vec{c} \bullet \vec{t} \quad (4.1)$$

4.4.2 Aprendizagem de Máquina

A outra forma testada foi o uso de aprendizagem de máquina para calcular a similaridade. Para isso foi criado um conjunto de dados da seguinte forma: Foram rotulados os chromas de um conjunto manualmente, preenchendo o resultado para cada par (chroma,template) um rótulo '1' se for similar e '0' se não for similar. Além disso o conjunto de dados foram enriquecidos com a criação de instâncias oriundas das modulações musicais.

Por Exemplo: Para um dado vetor [1,0,1,0,0,0,1] sua modulação +1 seria [1,1,0,1,0,0,0]

Foram aplicadas todas as modulações para os vetores do Chroma e do template do acorde. Simulando uma coleta de dados em todas as tonalidades, evitando assim um overfitting de tonalidade.

No final desta operação obtivemos uma matriz da seguinte forma

$$\begin{bmatrix} c_1 & t_1 & s_1 \\ c_2 & t_2 & s_2 \\ c_3 & t_3 & s_3 \\ \vdots & \vdots & \vdots \\ c_n & t_n & s_n \end{bmatrix}$$

c_i é o chroma da instância i ; t_i é o template da instância i ; s_i (4.2)

Foi usado o Multi Layer Perceptron(MLP) para aprender a função de Similaridade $S(c,t)$ e em seguida foi usado o modelo aprendido nas etapas posteriores.

4.5 Templates

O método explicado na seção anterior fez o uso de templates de acordes para calcular a similaridade entre acordes e Chromas. Nesta seção será explicada a forma utilizada para obter o template a partir do acorde.

No CifraClub obtemos os dados em forma de texto (semi-estruturada). Os acordes ficam dentro de uma tag "b"(por exemplo: Cm7) então um simples processador de texto foi capaz de buscar por essas tags.

Com isso cada acorde fica na sua forma textual para posteriormente ser processado por um algoritmo gerador de template.

4.5.1 Parser

Para analisar isto foi construído um parser que cria uma árvore sintática de um acorde dado. Com essa árvore constrói-se o template do respectivo acorde.

A gramática construída capaz de reconhecer os acordes é descrita da forma abaixo:

S -> MP Comp | MP

Comp -> Dis | Dis Bass | Bass
 Dis -> DT | DT DComp
 DComp -> OPr DT CPr | OPr DT CPr DComp
 Bass -> Bar P
 DT -> N | N Sig
 MP -> P | P M
 P -> PC | PC SB
 N -> Dig | Dig N
 Bar -> '/'
 OPr -> '('
 CPr -> ')'

PC -> 'C' | 'D' | 'E' | 'F' | 'G' | 'A' | 'B'
 SB -> '' | 'b'
 M -> 'm'
 Sig -> '+' | '-' | 'M'
 Dig -> '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' | '0'

4.5.2 Construindo Template

Com a árvore de um dado acorde em mãos analisamos primeiro a parte principal do acorde, ou seja, verificamos qual o formato da tríade (maior ou menor), e a partir disso criamos a base do template.

Por exemplo: "menor" -> [1,0,0,1,0,0,0,1,0,0,0,0]

Em seguida analisa-se as dissonâncias e para cada uma delas adiciona-se o valor na sua respectiva posição (previamente cadastrada num dicionário), entretanto com um peso menor do que as notas da tríade (foi escolhido 0.4):

Por Exemplo: "menor7" -> [1,0,0,1,0,0,0,1,0,0,0.4,0]

Por último faz a modulação para o acorde correto:

Por Exemplo: "Dm7" -> [0.4,0,1,0,0,1,0,0,0,1,0,0]

Foi encapsulado então todo esse tratamento em uma função que recebe a string do acorde e retorna o seu template.

4.6 Modelo

Nessa seção será abordado o modelo usado para estimar a transcrição dos acordes. Devido a característica probabilística do problema utilizamos o HMM (Hidden Markov Models) para modelá-lo.

4.6.1 Hidden Markov Models (HMM)

O HMM[Rab89] é um modelo de Markov que contém dois tipos de estados, um observável e um oculto. Ele usa as probabilidades condicionais entre os dois tipos de estado e as probabili-

dades de transições entre estados ocultos, para fazer inferências sobre o futuro, ou para calcular a sequência de estados ocultos mais provável dada uma sequência de estados observados.

Nas próximas seções serão mostradas como foram modeladas cada etapa, em alguns casos foram testados mais de uma forma de modelagem.

4.6.1.1 Estados Ocultos

Os Estados Ocultos foram modelados de duas formas, uma utilizando a Sequência de Acordes da Cifra como estado oculto, e a outra usando os Acordes da Música de forma isolada.

Por Exemplo: Uma Música que tem a Seguinte Sequencia de Acordes

[C,F,C,G,C,C,F,G,C]

Na abordagem de Sequência de Acordes (**SA**) como estado os estados ocultos seriam:

[$S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9$]

Tendo um estado para cada ocorrência de acorde.

Já a segunda abordagem Acordes como Estado (**AE**) seriam: [S_1, S_2, S_3]

Onde cada estado é um acorde existente na música

A primeira abordagem foi utilizada no Trabalho de Graduação[dVSRA16]]

4.6.1.2 Matriz de Transição

A Matriz de Transição deve mostrar as probabilidades de transição entre os estados S_i e S_j .

Foram propostas duas alterações no modelo passado[dVSRA16]], a fim de evitar problemas com cifras mal formadas, ou com incompatibilidade entre versões da música.

É neste ponto que as mudanças são essenciais para abordar o problema como transcrição e não alinhamento.

Na abordagem passada[dVSRA16]], foi feita uma matriz de transição onde o modelo só pode ficar no mesmo estado ou ir para o próximo estado. Quaisquer outras transições não são permitidas.

Então essa é a característica crítica do modelo, visto que, grande parte das cifras fazem transições cruzadas, contaminando então o modelo.

A primeira mudança proposta é utilizar uma suavização vista na equação (4.3):

$$M_t = (1 - \alpha) \cdot M_b + \alpha \cdot (1/n) \cdot O \quad (4.3)$$

$$M_b = \begin{bmatrix} P_f & P_m & 0 & 0 & \vdots & 0 \\ 0 & P_f & P_m & 0 & \vdots & 0 \\ 0 & 0 & P_f & P_m & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \vdots & P_m \\ 0 & 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad (4.4)$$

M_b representa a Matriz de Transição Básica; O é a matriz de Uns; n é o número de estados; e α é um fator que vai ponderar a suavização;

Espera-se que a suavização permita ao modelo fazer transições de maneira mais livre, mas ao mesmo tempo levando em consideração a sequência original. Essa Mudança só se aplica ao modelo **SA**.

Além disso, foi testada outra forma de transição, agora para o modelo **AE**. Neste outro modelo foi construída uma matriz de transição básica da seguinte forma:

$$M_b = (1 - P_f) \cdot \text{normalizarLinhas}([Count_{ij}]) + P_f \cdot I \quad (4.5)$$

Onde $Count_{ij}$ é a contagem de quantas transições existem do acorde i para j na sequência da cifra; e P_f é a probabilidade de ficar (parâmetro do modelo).

Então foram feitos os testes com o modelo original **SA** sem suavização, e com os modelos propostos (**SA** com suavização; **AE** sem suavização; **AE** com suavização) comparando os resultados.

4.6.1.3 Matriz de Emissão

No HMM a matriz de Emissão representa a seguinte propriedade:

$$M_e = [p_{ij}] : p_{ij} = P(o_j | s_i)$$

Onde o_j é o estado observado j e s_i é o estado oculto i .

O Modelo deste trabalho supõe que: $P(o_j | s_i) \propto S(c,t)$

Então se torna equivalente escrever a matriz de emissão com as respectivas probabilidades e em seguida aplicar a normalização nas linhas.

Foram feitos experimentos com as duas formas de calcular a Similaridade.

4.6.1.4 Probabilidade a priori

A probabilidade a priori é calculada de diferentes formas, uma para cada tipo de Modelagem de Estado.

No modelo **SA** cuja o estado é a Sequência de Acorde ela é colocada como 1 para o primeiro estado e 0 para os demais.

No outro Modelo (**AE**), cujo o estado é o acorde, a probabilidade é proporcional a quantidade de ocorrências do acorde.

4.6.2 Viterbi

O algoritmo de Viterbi [Rab89] é usado para calcular a sequência de estados ocultos mais provável de acordo com um HMM dado.

Nesse trabalho foi Modelado o HMM das diversas formas explicadas anteriormente e em seguida executado o Algoritmo de Viterbi.

Após isso, foi recuperada a duração de cada segmento original e construída a transcrição de cada acorde com seu instante inicial e duração;

Concluindo assim a última etapa do Método.

CAPÍTULO 5

Avaliação

Este Capítulo explicará os resultados de cada etapa assim como o desempenho. Foram coletadas 10 músicas entre as 12 mais acessadas do CifraClub, e em seguida foram rotuladas manualmente as transcrições de todas.

Música	Track Id	Url CifraClub
Trem Bala	05QeyKGAAn4TZrv41tMiD1A	/ana-vilela/trem-bala/
Despacito	5CtI0qwDJkDQGwXD1H1cLb	/luis-fonsi/despacito/
Vidinha de Balada	4FHw39Ke6pVt5tNKvLr4r3	/henrique-e-juliano/vidinha-de-balada/
Eu sei de Cor	45wqPwIH4A5cr2IyyD4WKf	/marilia-mendonca/eu-sei-de-cor/
Tu (Trevo)	6bgQIw80xWC3MPZX4kVYGw	/anavitoria/trevo-tu/
De quem é a culpa	4YhPcVIxIAJrLhFVPGbJkE	/marilia-mendonca/de-quem-a-culpa/
Flor e Beija Flor	0phVJ23NJlrwtt1XCm94N	/henrique-e-juliano/flor-e-o-beija-flor/
Medo Bobo	2CW04trIoYmBroZWDzPAjs	/maiara-maraisa/medo-bobo/
Shape of You	7qiZfU4dY1lWllzX7mPBI3	/ed-sheeran/shape-of-you/
tempo perdido	7MnT7msJZg3XBAS0OTfGrB	/legiao-urbana/tempo-perdido/

Tabela 5.1 Músicas usadas na avaliação.

5.1 Detecção de Tonalidade

Uma das primeiras etapas do modelo é a detecção do tom. Essa é uma etapa essencial, dado que a falha nessa etapa é propagada até o final.

Como essa etapa depende de como a similaridade é calculada foram feitos dois experimentos um para cada método: Produto Interno (PI) e o Multi Layer Perceptron (MLP)

O método proposto para essa etapa não foi bem sucedido, pois obteve uma baixa acurácia. Visto que ele só leva em consideração as características de similaridade da música, muitas melhorias podem ser feitas nessa etapa.

Assim as etapas seguintes receberam os dados com a tonalidade corrigida, para suas métricas serem avaliadas adequadamente.

Música	PI	MLP
Trem Bala	Sim	Sim
Despacito	Não	Não
Vidinha de Balada	Não	Não
Eu sei de Cor	Sim	Sim
Tu (Trevo)	Não	Não
De quem é a culpa	Não	Não
Flor e Beija Flor	Não	Não
Medo Bobo	Não	Não
Shape of You	Sim	Não
Tempo perdido	Não	Não

Tabela 5.2 Acerto da Tonalidade vs Método

Método	Taxa de Acerto
Produto Interno	3/10
Multi Layer Perceptron	2/10

Tabela 5.3 Método vs Taxa de Acerto

5.2 Experimentos

Para todos os experimentos seguintes foram utilizadas as seguintes métricas: Precisão (Prec.), Cobertura(Rec.) e Métrica-F (F-M).

Em cada música temos essas métricas para cada acorde, sendo assim os resultados nas tabelas mostram a média de cada respectiva métrica por acorde.

5.3 Similaridade com Multi Layer Perceptron

O MLP foi treinado para estimar a similaridade entre um Chroma e um Template.

Para avaliar o desempenho do MLP foram feitas 10 iterações dividindo o conjunto em 9 músicas para treinar e 1 para testar. Assim cada música foi testada uma vez.

Como os dados são desbalanceados com a proporção 1:12 foi feito um balanceamento selecionando todos os similares fazendo uma amostragem do conjunto de dissimilares.

Foi usado o MLP do SciKitLearn de python utilizando os parâmetros padrão e variando o número de neurônios na camada escondida [50,65,90,120].

Foi analisada a Área abaixo da Curva ROC como métrica.

Os resultados observados na Tabela 5.4 foram bons, entretanto para algumas músicas essa tarefa não é fácil de ser aprendida, pois elas possuem alguns segmentos pouco preenchidos. Por exemplo, 'Shape of You' e 'Despacito' possuem muitos contratempos, logo várias partes das regiões de acordes contêm segmentos de silêncio.

Música	AUC(h=50)	AUC(h=65)	AUC(h=90)	AUC(h=120)
Trem Bala	0,7897	0,8002	0,8022	0,8091
Despacito	0,6892	0,6919	0,6947	0,6931
Vidinha de Balada	0,7499	0,7578	0,75992	0,7611
Eu sei de Cor	0,7047	0,7134	0,7187	0,7161
Tu (Trevo)	0,72082	0,7156	0,7186	0,7148
De quem é a culpa	0,7834	0,7876	0,7891	0,7946
Flor e Beija Flor	0,8971	0,8968	0,9003	0,9050
Medo Bobo	0,7106	0,7139	0,7218	0,7147
Shape of You	0,6513	0,6563	0,6581	0,6695
Tempo perdido	0,7449	0,7517	0,75378	0,7458

Tabela 5.4 Música vs Área abaixo da Curva ROC

Tabela 5.5 Comparação da Precisão MLP e PI

Música	PI	MLP
Trem Bala	0,7069961102	0,7271871650
Despacito	0,7201658989	0,6914781187
Vidinha de Balada	0,4040250316	0,4991086362
Eu sei de Cor	0,7485617199	0,8360598253
Tu (Trevo)	0,1948058233	0,2606958927
De quem é a culpa	0,5015244446	0,6557461458
Flor e Beija Flor	0,8077112040	0,8253217312
Medo Bobo	0,5922637028	0,6410383304
Shape of You	0,4826447434	0,5291122383
Tempo perdido	0,1726053809	0,3069438642

5.3.1 Comparação Similaridade MLP com Produto Interno

Para fazer essa comparação foram feitos dois experimentos com o SAss: um deles usando a similaridade com o PI e outro utilizando o MLP.

A melhoria com o uso da similaridade treinada (MLP) é evidente. De modo geral essa mudança apresentou uma melhoria ao modelo proposto anteriormente.

Como o MLP se mostrou superior, todos os experimentos seguintes utilizaram o MLP como Similaridade.

5.4 Experimentos com o Modelo SA

Foram feitos testes comparando os modelos SA sem suavização (SAss) e o SA com suavização (SAcs).

Note que o SAss é o modelo de [dVSR16].

Tabela 5.6 Comparação da Recall MLP e PI

Música	PI	MLP
Trem Bala	0,7499878829	0,7201370332
Despacito	0,6979102664	0,6905993798
Vidinha de Balada	0,5281768789	0,5695013147
Eu sei de Cor	0,8161056831	0,8511053255
Tu (Trevo)	0,2240507532	0,2729260523
De quem é a culpa	0,5278736222	0,69625139
Flor e Beija Flor	0,8831077464	0,9013117355
Medo Bobo	0,6194929715	0,6709437759
Shape of You	0,4148730583	0,5403932782
Tempo perdido	0,2340388845	0,344254137

Tabela 5.7 Comparação da Métrica-F MLP e PI

Música	PI	MLP
Trem Bala	0,7161046228	0,7225504553
Despacito	0,6978467704	0,6864792417
Vidinha de Balada	0,4491579748	0,5313507851
Eu sei de Cor	0,7500786419	0,841622646
Tu (Trevo)	0,1894984632	0,2606493538
De quem é a culpa	0,4935495981	0,6584790977
Flor e Beija Flor	0,8341591454	0,8514759591
Medo Bobo	0,6028776405	0,6514896786
Shape of You	0,4041525126	0,5314988033
Tempo perdido	0,1959788334	0,3072668966

Tabela 5.8 Tabela de comparação entre SAss e SAcS.

Música	Prec(SAss)	Rec(SAss)	F-M(SAss)	Prec(SAcS)	Rec(SAcS)	F-M(SAcS)
Trem Bala	0,7272	0,7201	0,7226	0,6426	0,5999	0,6124
Despacito	0,6915	0,6906	0,6865	0,6865	0,6732	0,6686
Vidinha de Balada	0,4991	0,5695	0,5314	0,3801	0,3634	0,3194
Eu sei de Cor	0,8361	0,8511	0,8416	0,8361	0,8511	0,8416
Tu (Trevo)	0,2607	0,2729	0,2606	0,2991	0,3400	0,3149
De quem é a culpa	0,6557	0,6963	0,6585	0,5387	0,6161	0,5444
Flor e Beija Flor	0,8253	0,9013	0,8515	0,9151	0,9169	0,9145
Medo Bobo	0,6410	0,6709	0,6515	0,6410	0,6709	0,6515
Shape of You	0,5291	0,5404	0,5315	0,5281	0,5319	0,5239
Tempo perdido	0,3069	0,3443	0,3073	0,4035	0,4310	0,4107

Tabela 5.9 Tabela de comparação AEss vs AEcs com ambos usando MLP.

Música	Prec(AEss)	Rec.(AEss)	F-M(AEss)	Prec.(AEcs)	Rec (AEcs)	F-M (AEcs)
Trem Bala	0,7087	0,7413	0,7152	0,7084	0,7556	0,7215
Despacito	0,6844	0,6407	0,6575	0,6860	0,6273	0,6521
Vidinha de Balada	0,4288	0,5023	0,4623	0,4232	0,4917	0,4542
Eu sei de Cor	0,6786	0,6959	0,6850	0,6786	0,6959	0,6850
Tu (Trevo)	0,2056	0,2847	0,2168	0,2079	0,3017	0,2383
De quem é a culpa	0,3640	0,3803	0,3640	0,3328	0,3438	0,3224
Flor e Beija Flor	0,9056	0,9064	0,9029	0,7934	0,8972	0,8050
Medo Bobo	0,6675	0,6991	0,6771	0,6379	0,6675	0,6495
Shape of You	0,5433	0,5578	0,5487	0,5433	0,5578	0,5487
Tempo perdido	0,3842	0,3199	0,3334	0,4163	0,3667	0,3729

5.4.1 Modelo SA com suavização (SAcs)

Para avaliar essa proposta foi usado o SAcS com o coeficiente de suavização ($\alpha = 0.08$).

Segundo os dados apresentados na Tabela 5.8 nota-se que houve melhoria em algumas músicas, mas que em outras o desempenho foi degradado.

5.5 Experimento com o Modelo AE

Foram feitos também experimentos com o Modelo AE comparando também o uso da suavização (AEss e AEcs). Descritos na Tabela 5.9. (AEcs com $\alpha = 0.1$)

Com o experimento reportado observa-se que o AEss é ligeiramente melhor do que AEcs, pois na maioria dos casos o modelo AEcs não aumenta o valor das métricas. O que é esperado dado que o AEss já contém bastante flexibilidade, não sendo necessária uma suavização.

Tabela 5.10 Comparação Geral

Música	F-M(SAss)	F-M(SAcs)	F-M(AEss)	F-M (AEcs)
Trem Bala	0,7226	0,6124	0,7152	0,7215
Despacito	0,6865	0,6686	0,6575	0,6521
Vidinha de Balada	0,5314	0,3194	0,4623	0,4542
Eu sei de Cor	0,8416	0,8416	0,6850	0,6850
Tu (Trevo)	0,2606	0,3149	0,2168	0,2383
De quem é a culpa	0,6585	0,5444	0,3640	0,3224
Flor e Beija Flor	0,8515	0,9145	0,9029	0,8050
Medo Bobo	0,6515	0,6515	0,6771	0,6495
Shape of You	0,5315	0,5239	0,5487	0,5487
Tempo perdido	0,3073	0,4107	0,3334	0,3729

5.6 Comparação Geral

Como o SAss é o modelo base, é importante compará-lo com o todos os outros na Tabela 5.10.

5.7 Considerações

A mudança principal que contribuiu para a melhoria do método foi o uso de aprendizagem de máquina, que se obteve bons resultados na estimação da similaridade. Já com as outras alterações não fica claro a obtenção de avanços, pois existiu melhora em algumas músicas mas em outras houve uma pequena piora.

CAPÍTULO 6

Conclusões

Neste capítulo será apresentado o que foi inferido neste trabalho, e pontos que valem a pena ser estudados futuramente.

6.1 Resultados

O presente trabalho foi realizado com êxito, no entanto possui ressalvas. Por exemplo, o desempenho da detecção de acorde não foi satisfatório, que é algo crítico, já que esse tipo de erro inviabiliza as próximas etapas.

No que diz respeito aos resultados positivos, o maior destaque se dá com o uso do MLP para aprender a função de similaridade, pois neste caso o avanço foi evidente.

O uso da suavização apontou uma melhora em algumas músicas cuja sequência de acordes do CifraClub era incompatível com a versão do Spotify, mas com o custo de degradar um pouco as músicas já bem formadas.

Além de tudo o que foi citado anteriormente, esse trabalho contribuiu para construção de um parser de acordes que pode ser reusado para uma gama de aplicações.

Acreditamos que esse trabalho contribuiu para a progressão do estudo de Transcrição Automática de Acordes usando fontes híbridas, podendo servir de base para projetos futuros.

6.2 Trabalhos Futuros

Para trabalhos futuros: vale a pena abordar novas formas de detecção de tons; fazer o uso de outros algoritmos de aprendizagem de máquina para aprender a similaridade; pré-processar os segmentos com uma janela deslizante para dar mais robustez à detecção e segmentação dos acordes para músicas com segmentos pouco preenchidos; por fim integrar tudo isso numa API para que outras aplicações possam usar essa ferramenta de maneira criativa.

Referências Bibliográficas

- [BP05] J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. *Proc. of the Int. Symposium on Music Information Retrieval*, pages 304–311, 2005.
- [dVSRA16] Pedro Tiago de V. S. R. Araújo. Transcrição e matching automático de uma música e sua cifra. *Trabalho de Graduação*, CIn - UFPE, 2016.
- [MMB14] Yizhao Ni Matt McVicar, Raúl Santos-Rodríguez and Tijl De Bie. Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 22(2), 2014.
- [PB] J. Pablo Bello. Chroma and tonality. *MPATE-GE 2623 Music Information Retrieval*.
- [Rab89] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 77(2):257–286, 1989.
- [Spo] Spotify. Get audio analysis for a track, <https://developer.spotify.com/web-api/get-audio-analysis/>.
- [Wik] Wikipedia. Data scraping, 'https://en.wikipedia.org/wiki/data_scraping'.