Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Carlos Augusto Motta de Lima

Reconhecimento Automatizado de Acordes

São Paulo
Novembro de 2017

Reconhecimento Automatizado de Acordes

Monografia parcial da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Marcelo Queiroz

São Paulo
Novembro de 2017

Resumo

O reconhecimento automatizado de acordes é um processo no qual, dado um sinal de áudio de uma música, se produz uma sequência de etiquetas, cada uma representando um acorde, acompanhadas dos respectivos tempos de início e fim de cada acorde que é tocado na música. Neste trabalho, se descreverá o problema de forma detalhada, passando pelos conceitos teóricos envolvidos, e se estudará e implementará uma solução particular, proposta em Müller (2015). Também se discutirá um método de avaliação do algoritmo, através da obtenção de estatísticas comparativas entre dados rotulados automaticamente e dados de treinamento previamente rotulados, usando estratégias como validação cruzada e a medida de precisão, típica de recuperação de informação. A melhor precisão média obtida nos experimentos foi de 55%, usando CQT para extração de cromagramas e suavização temporal dos cromas antes da etapa de casamento de padrões do algoritmo apresentado.

Palavras-chave: processamento de sinais digitais, recuperação de informação musical, reconhecimento de acordes.

Abstract

Chord recognition is the process where, given an audio signal representation of a song, it is produced a label sequence, each one representing a chord, and its respective onset and offset times where the chords sound on the song. In this work, we describe the problem of chord recognition in a detailed way, going through the involved concepts, and study and implement a particular solution presented in Müller (2015). We also discuss a method for evaluating such algorithm, through the computation of statistics of comparison between automatically labeled data and ground-truth data, using strategies such as k-fold cross-validation and the measure of precision, typical in the information retrieval field. The best average precision obtained in the experiments were 55%, using CQT for chromagram extraction and temporal smoothing of chromas before the pattern matching step of the presented algorithm.

Keywords: digital signal processing, musical information retrieval, chord recognition.

Sumário

Capítulo 1

Introdução

Esta é uma monografia parcial. Em algumas partes deste texto, estão descritos trechos que serão redigidos na monografia final no sentido de torná-lo mais completo e autocontido. Além das mudanças já planejadas, pretende-se aperfeiçoar o texto em geral, acrescentando ou removendo detalhes conforme for desejável em cada trecho.

A seção de experimentos desta versão do texto, em particular, apenas descreve a metodologia usada nos experimentos deste trabalho, e apresenta os dados dos resultados de forma bruta. Tais resultados serão discutidos de forma explicativa na versão final da monografia, onde também se disponibilizarão os dados agregados e com melhores visualizações.

Enfim, diversos itens serão revisados até a monografia final - espera-se, porém, que esta versão passe uma visão geral do trabalho e dos planos para a versão final, que conterá um relato completo.

1.1 Motivação e objetivos

Dos diversos elementos que compõem a música tonal, a harmonia - cujo componente elementar, ao menos dentro da tradição da música ocidental, é o acorde - é um de suma importância. O reconhecimento de acordes é um problema bastante estudado dentro da área de Recuperação da Informação Musical (MIR) e diversos trabalhos já foram apresentados na academia.

Algoritmos de reconhecimento de acordes num sinal de áudio podem ser úteis em diversos problemas, como em classificação de gêneros, por exemplo, ou, de forma particularmente relevante para a escolha do tema deste trabalho, na produção automática de cifras.

Hoje em dia já existem aplicativos em produção capazes de fazer o reconhecimento de acordes de forma eficiente, como o Chordata, do projeto CLAM, ou o Chordify.

Neste cenário, mostrou-se desejável familiarizar-se com o problema, e conhecer possibilidades de solução do mesmo. Como referência para o estudo, utilizou-se o trabalho introdutório de Müller (2015), que apresenta duas soluções diferentes. Restringiu-se, no contexto deste trabalho, a uma das soluções: o algoritmo de classificação de acordes baseado em templates, acompanhado de técnicas para aperfeiçoamento do mesmo.

1.2 Estrutura da monografia

Além do capítulo introdutório e conclusivo, o conteúdo deste monografia está dividido em dois capítulos: Fundamentação Teórica e Desenvolvimento.

No capítulo de Fundamentação Teórica se discorrerá brevemente sobre os conceitos necessários para o entendimento do problema e da solução apresentada neste trabalho. Esses conceitos vêm tanto da teoria musical quanto da ciência da computação. Em especial, se definirá o conceito de acorde, que é fundamental neste contexto, e diversos conceitos secundários necessários para a definição do mesmo. Se apresentarão também conceitos da área de Processamento de Sinais Digitais como sinal de áudio, Transformada de Fourier, espectrograma e cromagrama.

No capítulo de Desenvolvimento, será proposto um algoritmo para a resolução do problema de reconhecimento de acordes, e se discorrerá sobre limitações do mesmo e técnicas para melhorar sua eficácia. Também será discutida uma metodologia de avaliação do algoritmo e apresentada uma base de anotações de referência. Por último, serão relatados os experimentos feitos durante este trabalho e os resultados e conclusões decorrentes deles.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão introduzidos conceitos necessários para a compreensão do problema e da solução estudada, passando pelas áreas de teoria musical (como a definição de um acorde à partir de outros conceitos como nota e intervalo) e computação musical (como sinais de áudio, transformada de Fourier e cromagrama).

2.1 Acordes

De forma pouco rigorosa, um **acorde** pode ser definido como diversas notas musicais que soam ao mesmo tempo. Para entender melhor esse conceito e também o conceito de nota musical, é interessante relembrar as características do som em si.

O som é uma onda mecânica - em particular, é a propagação de compressões e descompressões de um meio material, como o ar. Ele se propaga a partir de uma fonte sonora, como um instrumento musical, até atingir um receptor, como o ouvido humano ou um microfone. Em geral, ele é representado como a variação da pressão do ar num dado ponto do espaço em função do tempo.

É chamada **senoide** uma onda sonora cuja forma é perfeitamente senoidal. As senoides são os sons mais simples possíveis, e a variação da pressão que as caracteriza ocorre com uma determinada **frequência**, medida em Hertz (Hz). Os sons que escutamos na natureza, e também aqueles produzidos por instrumentos musicais, são, por sua vez, mais complexos, e podem ser descritos como uma composição (soma) de diversas senoides com amplitudes e frequências diferentes.

O ser humano é capaz de identificar a frequência dos sons através do correlato psicoacústico chamado **altura**, que é a propriedade que diferencia um som grave, de baixa frequência, de um som agudo, de alta frequência.

A alguns sons, não podemos associar uma altura determinada, pois possuem energia em muitas frequências diferentes: o som de um pandeiro, por exemplo. Aos outros sons, aos quais conseguimos associar uma altura determinada, chamamos de **notas musicais**.

Define-se como **intervalo** a diferença de altura entre duas notas musicais.

A percepção humana de altura possui uma propriedade interessante: duas notas, quando a frequência de uma é o dobro da frequência da outra, são entendidas como possuindo equivalência sonora. Notas que respeitam essa condição têm entre si um intervalo de **uma oitava**. Se a razão entre frequências for 4, então diz-se que as notas têm um intervalo de **duas oitavas**, e assim sucessivamente.

A música ocidental, ao longo da história, desenvolveu a chamada **escala diatônica**, que divide uma oitava em 12 intervalos entendidos perceptualmente como iguais (nota-se que a percepção humana de altura varia logaritmicamente em relação à variação de frequências).

Nota (sustenido)	**Símbolo (♯)**	**Símbolo (♭)**	**Nota (bemol)**
Dó	C	C	Dó
Dó sustenido	C♯	D♭	Ré bemol
Ré	D	D	Ré
Ré sustenido	D♯	E♭	Mi bemol
Mi	E	E	Mi
Fá	F	F	Fá
Fá sustenido	F♯	G♭	Sol bemol
Sol	G	G	Sol
Sol sustenido	G♯	A♭	Lá bemol
Lá	A	A	Lá
Lá sustenido	A♯	B♭	Si bemol
Si	B	B	Si

Tabela 2.1: *Notas musicais da escala diatônica e seus nomes*

Tal intervalo é chamado de **um semitom**. Nessa escala, entre duas notas que possuem um intervalo de uma oitava, há uma sequência de outras dez notas.

A escala diatônica é cíclica. Partindo-se de uma nota e aumentando-a em um semitom 12 vezes, se alcança a oitava dessa nota, e esse processo pode ser repetido até que se obtenha uma nota tão aguda que chega a ser inaudível (o intervalo de frequências sonoras audíveis pelo ser humano vai de 20 Hz a 20.000 Hz).

A música ocidental é construída apoiando-se no semitom como o menor intervalo que distingue duas notas. Dessa forma, nomeiam-se 7 notas musicais de forma cíclica: algumas dessas notas possuem entre si um intervalo de um semitom e outras de dois semitons (ou **um tom**). A cada uma das 7 notas, é atribuída uma letra como forma simplificada de representação. Para representar as outras 5 notas não contempladas com os 7 nomes, usam-se as palavras **sustenido** (que acrescenta uma nota de um semitom) e **bemol** (que diminui uma nota de um semitom). Para a representação simplificada, sustenido e bemol possuem os símbolos ♯ e ♭, respectivamente.

A tabela 2.1 mostra todas as notas da escala diatônica, começando da nota dó, e cada linha tem a nota exatamente um semitom acima da nota anterior.

Já que podemos classificar todas as notas musicais como uma entre as 12 existentes na escala diatônica, pode-se definir o conceito de **classes de altura**. Uma classe de altura é um conjunto de notas que possuem intervalos de um número inteiro de oitavas entre si. Por exemplo, a classe de altura **C** é o conjunto de todas as notas dó, não importando em qual oitava se encontram.

Como anteriormente definido, um acorde é um conjunto de notas que soam ao mesmo tempo. Na ótica da escala diatônica, podemos simplificar essa definição como um conjunto de classes de altura que soam ao mesmo tempo. Com essa nova definição, um acorde pode conter até 12 elementos, que são as classes de altura existentes. Dois acordes são iguais se soam as mesmas classes de altura, não importando se as oitavas de cada nota dos dois acordes são iguais ou não.

Na versão final desta monografia, esta seção ainda explicará alguns conceitos que serão posteriormente necessários para o entendimento do algoritmo apresentado na seção 3.1:

- O que é uma tríade (uso na música ocidental folclórica e popular é ubíquo)
- Tipos de tríades: maiores e menores, 5ª aumentada, 5ª diminuta

- Outros tipos de acordes, notas acrescentadas etc.
- Notações de acordes
- Ambiguidades de representação

2.2 Representação de sinais de áudio

Conforme já visto, o som pode ser descrito como a variação da pressão (em algum ponto do espaço) em função do tempo.

Para representar essa função no domínio digital, é necessário discretizá-la, tanto no eixo do tempo como no da pressão. Assim, o som é representado em computadores como uma sequência de valores reais (amostras) que correspondem à variação da pressão em pontos espaçados uniformemente no tempo. O número de amostras presentes em um segundo de áudio é chamado **taxa de amostragem**. A discretização do sinal analógico introduz certas limitações - como o erro de quantização ou a impossibilidade de representar sons com frequências maiores que metade da taxa de amostragem (ver Broughton e Bryan, 2011, seções 1.3.2 e 1.6).

Quando lidamos com reconhecimento de acordes, é útil identificar quais frequências estão soando num som. No entanto, esta tarefa não é trivial quando representamos o som como a variação da pressão em função do tempo. Para realizá-la, pode-se usar a **transformada discreta de Fourier** (ou **DFT**) - uma transformada que pode ser entendida como uma decomposição de um sinal sonoro qualquer em senoides.

A transformada de Fourier de um sinal é uma representação em função da frequência. Esse tipo de representação, porém, não contempla informações temporais, isto é: apesar de se saber através dela quais frequências (e em que intensidade) estão presentes num sinal sonoro, não se sabe em quais intervalos de tempo essas frequências soaram.

Como alternativa à DFT, existe a **STFT**, ou **transformada de Fourier de tempo curto**, que divide o sinal sonoro em janelas temporais de mesma duração e obtém a DFT de cada janela, possibilitando a extração de informações espectrais e temporais de um mesmo de áudio simultaneamente.

Vale ressaltar que quanto mais curta é uma janela temporal, menor será a resolução de frequências que a DFT é capaz de representar (ver Broughton e Bryan, 2011, seções 2.1 a 2.3). Dessa forma, faz-se necessário encontrar um equilíbrio entre a resolução temporal, que é privilegiada por janelas temporais mais curtas, e a resolução das frequências, que é privilegiada por janelas temporais mais longas.

Uma das características da STFT é que ela oferece uma resolução igual para frequências graves e agudas. Ou seja, o número de pontos contemplados pela STFT entre 100 Hz e 200 Hz é o mesmo número de pontos contemplados entre 1100 Hz e 1200 Hz. Porém, devido à percepção humana de altura, que varia de forma logarítmica em função da variação de frequências, notas agudas terão uma resolução maior na STFT do que notas graves.

Para obter uma resolução igual para diferentes notas musicais, pode-se usar uma outra transformada: a transformada Q-constante ou **CQT**. Ela também tem a função de levar um sinal sonoro da representação temporal para a espectral; contudo, aumenta a resolução das frequências graves e diminui a resolução das frequências agudas, igualando, assim, a resolução de cada nota musical. Ou seja, todas as notas musicais terão um mesmo número de representantes no espectrograma, o que não penaliza a amplitude associada às notas graves em relação às agudas (ver Müller, 2015, seção 3.4.1).

2.3 Características harmônicas

Define-se ***croma*** como um vetor real de dimensão 12 cujas entradas representam a intensidade com que uma classe de altura aparece num dado sinal de áudio. Tal vetor é indexado de 0 a 11, onde 0 é a classe dó (C) e 11 a classe si (B).

Se dividirmos um sinal de áudio longo em janelas e associarmos a cada quadro uma *chroma*, obteremos uma matriz chamada **cromagrama**. Cromagramas são análogos a espectrogramas, com a diferença de que um apresenta as intensidades de frequências presentes no sinal, enquanto outro apresenta intensidades de classes altura.

Esta seção entrará em maior profundidade em relação às características do cromagrama, em especial:

- Como construir o cromagrama
- Que informações se perdem no cromagrama

Capítulo 3

Desenvolvimento

3.1 Classificação baseada em templates

Para reconhecer quais acordes estão sendo tocados numa música a partir de uma gravação, Müller (2015) propõe um algoritmo que se divide em duas partes: extração de *features* e casamento de padrões.

A primeira consiste na construção, usando a STFT, do cromagrama do sinal. O croma é escolhido como *feature* porque captura informações tonais da música, que compõem sua harmonia, da qual os acordes são elementos. Assim, se define $X = (x_1, x_2, \ldots, x_n)$ como sequência de *features*, onde cada elemento é um croma $x_i \in R^{12}$ de uma janela do sinal.

A segunda etapa do algoritmo consiste em etiquetar cada janela da etapa anterior com um acorde. Para isso, define-se o conjunto Λ de acordes a serem levados em consideração durante a classificação. No escopo deste trabalho, tomou-se como Λ o conjunto de todas as tríades maiores e menores:

$$\Lambda = \{\mathrm{C}, \mathrm{C\sharp}, \ldots, \mathrm{A\sharp}, \mathrm{B}, \mathrm{Cm}, \mathrm{C\sharp m}, \ldots, \mathrm{A\sharp m}, \mathrm{Bm}\} \tag{3.1}$$

Define-se, também, um conjunto $T \subset R^{12}$ de *templates de croma*, de forma que cada acorde considerado na classificação possua um representante no conjunto de templates de croma, ou seja:

$$\exists t_\lambda \in T, \forall \lambda \in \Lambda$$

Esse conjunto é construído de forma que cada template se pareça com o croma de uma janela de áudio onde seu respectivo acorde soa. O algoritmo pressupõe que esse conjunto já foi pré-computado.

O conjunto de templates de croma mais simples é o de *templates binários*. Sua construção consiste em definir para cada $\lambda \in \Lambda$ um $t_\lambda \in R^{12}$ tal que:

$$(t_\lambda)_j = \begin{cases} 1, & \text{se } j \in notas(\lambda) \\ 0, & \text{caso contrário.} \end{cases}$$

onde $notas(\lambda)$ é o conjunto de índices das notas presentes no acorde λ na sequência de notas de dó a si.

Se define, enfim, a correlação entre um croma x_i e um template t_λ, que é um valor real que mede quão parecidos eles são. Por simplicidade, usamos o produto interno como medida de correlação. É importante que ambos os vetores estejam normalizados, para que se possa comparar correlações entre pares diferentes de vetores:

$$C(x_i, t_\lambda) = \frac{\langle x_i, t_\lambda \rangle}{||x_i|| \cdot ||t_\lambda||}$$

Definidos todos os itens acima, o algoritmo de classificação de acordes baseado em templates é descrito em dois passos:

1. Se extrai a sequência X de cromagramas do sinal;
2. Para cada $i = 1, 2, \ldots, n$, a i-ésima janela do sinal é classificada com o acorde λ_i definido por:

$$\lambda_i = \arg\max_{\lambda \in \Lambda} \{C(x_i, t_\lambda)\}$$

O algoritmo apresentado é uma das possíveis soluções para o problema. Porém, alguns fatores limitam a acurácia da sua classificação. Por exemplo: o conjunto Λ usado possui apenas 24 acordes, quando o conjunto de todos os acordes existentes é muito maior.

Outro fator que diminui a acurácia do algoritmo é o conjunto T. Apesar de modelarem com simplicidade os cromas de acordes, os templates binários refletem pouco da realidade, o que será discutido em mais detalhes posteriormente.

Por essas e outras razões, algumas técnicas de aperfeiçoamento do algoritmo foram experimentadas para melhorar sua acurácia. Nas subseções seguintes se discutirão essas técnicas.

3.1.1 Templates aprendidos

Templates binários são modelos simples para cromas em que soam determinado acorde. Contudo, num sinal de áudio gravado em uma performance musical do mundo real, muito raramente - provavelmente nunca - se encontrará um croma igual a um template binário.

O motivo é que nenhum som produzido por instrumentos musicais do mundo real pode ser decomposto em apenas uma senoide: sempre haverá a presença de ruído, harmônicos e outros sons que agregam energia em diferentes frequências do espectro sonoro.

Além disso, a própria digitalização do som e extração de cromagrama são processos que pressupõem discretizações e acréscimo de harmônicos no sinal sonoro.

Por essas razões, o template binário não é a representação mais fiel para nosso objetivo.

Diante disso, é proposto que o conjunto de templates seja aprendido a partir de dados previamente anotados (*ground-truth*). Se sabemos previamente quais acordes soam em cada janela temporal de um determinado conjunto de áudios, podemos agregar os cromas das janelas agrupando-as por acorde e produzir um template para cada acorde que aparece na amostra.

A forma mais simples de aprender o template para um acorde é calculando a média simples de cada componente das cromas cujas janelas estão anotadas com ele. Tomando o conjunto T_λ de todos os cromas extraídos dos áudios e cujas janelas estão anotados com o acorde λ, podemos definir cada j-ésima componente do template médio t^a_λ como:

$$(t^a_\lambda)_j = \frac{1}{|T_\lambda|} \sum_{x \in T_\lambda} x_j$$

3.1.2 CQT em vez de STFT

CQT e DFT possuem funções parecidas: ambas transformam a representação de um sinal do domínio do tempo para o de frequências. Por isso, ambas são adequadas para a construção de um cromagrama, mas não igualmente efetivas.

O domínio de um sinal transformado por uma STFT tem pontos que representam frequências espaçadas linearmente no intervalo de frequências considerado. Isto significa que o número de *bins* entre 20 Hz e 40 Hz será o mesmo que entre 2.000 Hz e 2.020 Hz. No entanto, o intervalo percebido entre uma nota cuja frequência é 20 Hz e outra cuja frequência é 40 Hz é de uma oitava, enquanto no caso seguinte (2.000 Hz e 2.020 Hz), há um intervalo menor que um semitom. Em outras palavras, a STFT proporciona uma resolução por nota variável: menor para notas graves e maior para notas agudas.

Dependendo da resolução em frequência da STFT, essa propriedade pode fazer com que, para algum conjunto de notas graves, um mesmo *bin* corresponda a mais de uma nota ao mesmo tempo.

No caso da CQT, a lógica é diferente: um mesmo intervalo musical (digamos, uma oitava, por exemplo) terá a mesma resolução em *bins* independentemente da altura na qual as notas que o definem são tomadas. Em particular, a identificação de notas graves não é penalizada no agrupamento por classes de altura.

Por essa razão, o uso da CQT em vez da STFT pode aperfeiçoar a classificação de acordes baseada em templates.

3.1.3 Compressão espectral

Esta subseção explicará a melhoria que se pode obter na etapa de casamento de padrões quando se faz previamente uma compressão logarítmica dos cromas. Na monografia final se explicará como a compressão enfatiza as classes de altura com menor quantidade de energia em detrimento das com maior quantidade de energia, que podem descaracterizar um acorde num seu cromagrama (dar muita ênfase a determinada nota no casamento de padrões).

3.1.4 Suavização temporal

Em muitos fonogramas é comum que, dentro de um período de tempo de algumas janelas em que um mesmo acorde é tocado, hajam variações locais irrelevantes nos cromas.

Isso pode acontecer por diferentes motivos. Um deles é a presença de "acordes quebrados", que são aqueles cujas notas não são tocadas simultaneamente, mas sim uma de cada vez. Por exemplo, um dó maior (que é formado pelas notas dó, mi e sol) pode ser tocado arpejando-se as três notas que o compõem. Nesse caso, ainda que cada nota continue soando pelas janelas subsequentes à qual foi tocada, é provável que elas contenham menos energia provinda dessa nota e mais das outras.

Variações locais irrelevantes nos cromas podem desencadear, neste algoritmo, variações locais equivocadas na classificação.

Com essa motivação, uma técnica que pode apresentar melhoria expressiva na acurácia da classificação é a *suavização temporal* dos cromas antes da etapa de casamento de padrões. Essa prática espalha a energia presente em cada nota de um croma para os cromas vizinhos e funciona como um filtro passa-baixas para cada classe de altura da sequência de cromas.

Para aplicar essa técnica, se define uma nova sequência de cromas $\hat{X}$ a partir de X. O i-ésimo croma de $\hat{X}$ será igual à média componente a componente dele com os L cromas vizinhos anteriores e posteriores:

$$(\hat{x}_i)_j = \frac{(x_i)_j + \sum_{k=1}^{L}((x_{i-k})_j + (x_{i+k})_j)}{2L+1}$$

Na etapa de casamento de padrões, a sequência de cromas a ser considerada será $\hat{X}$ em vez de X.

3.1.5 Estimativa de afinação

Agrupar a energia do espectrograma de um sinal de áudio por classes de altura envolve, necessariamente, a definição de uma base de afinação.

Em geral, a afinação utilizada como referência na música ocidental nos dias atuais é a nota chamada **Lá 440**, com frequência 440 Hz. Essa convenção foi definida pela Organização Internacional para Padronização como a norma ISO 16.

Apesar de ser uma convenção, há exceções em que a afinação em Lá 440 não é utilizada. Num algoritmo de reconhecimento de acordes para uso geral (sem restrição da afinação da música), seria interessante se saber previamente qual afinação é utilizada, de forma que a construção do cromagrama seja o mais precisa possível.

Sem assumir que esse dado é de conhecimento prévio, pode-se utilizar alguma técnica de estimativa de afinação. No contexto deste trabalho utilizou-se interpolação parabólica para essa estimativa, que é o método padrão da biblioteca escolhida para processamento de áudio.

O algoritmo de estimativa recebe como parâmetro um valor de resolução, que é um número racional que determina a precisão da interpolação. O valor padrão deste parâmetro é de 1e−1, mas pode ser sobrescrito com um valor arbitrariamente pequeno (dentro das limitações de representação). Valores de resolução menores podem ser mais eficientes em alguns casos, porém podem trazer maior ineficiência em outros.

As decisões em relação à estimativa de afinação serão discutidas em mais detalhes na seção de experimentos.

3.1.6 Pós-filtragem

Esta subseção explicará a melhoria que se pode obter acrescentando uma etapa ao algoritmo: após o casamento de padrões, pode-se aplicar a chamada "pós-filtragem", que visa eliminar acordes aparentemente aleatórios que aparecem em algumas classificações entre sequências de acordes iguais. Por exemplo:

$$(\ldots A, A, A, F\sharp m, A, A, A \ldots)$$

Na classificação apresentada, o acorde F♯m aparenta ser um erro, pois é muito improvável que em uma música haja uma troca de acorde que dure apenas o equivalente a uma janela temporal (considerando que as janelas, no escopo deste trabalho, têm, no máximo, perto de um décimo de segundo).

Um forma de tentar corrigir esse tipo de erro é construindo uma nova classificação a partir da classificação original. Nesta nova classificação, cada item será calculado a partir da classificação original através de um voto de maioria que considera o item original e seus vizinhos à esquerda e à direita. É necessário definir quantos vizinhos serão considerados.

Os experimentos feitos neste trabalho mostraram que a pós-filtragem conforme definida acima traz certa melhoria na classificação. No entanto, essa melhoria não se acumula com a melhoria trazida pela técnica de suavização temporal. Isso se dá porque a aleatoriedade

com que os acordes aparecem numa classificação diminui consideravelmente quando se aplica previamente uma suavização temporal nos cromas.

3.2 Implementação

O objetivo deste trabalho não é a implementação de um sistema de reconhecimento de acordes, mas sim o estudo detalhado e completo de um algoritmo que resolve este problema. As escolhas das tecnologias e arquitetura para a implementação do algoritmo observou tal consideração.

A linguagem escolhida para implementação foi Ruby, devido à facilidade para construção de um código-fonte legível e à forte familiaridade do autor com ela.

Para as funções clássicas de processamento de áudio - como construção de espectrograma, cromagrama e estimativa de afinação - utilizou-se o Librosa (2017), um pacote escrito em Python e de simples uso.

Para o uso de um pacote escrito em Python num software escrito em Ruby, utilizou-se a *gem* PyCall, que possibilitou a adição de uma interface para o Librosa de forma direta.

O código foi estruturado de forma que se pudesse rodar experimentos de forma automatizada. Portanto, foi necessário permitir que a seleção das técnicas de aperfeiçoamento do algoritmo fosse feita via passagem de parâmetros.

Para organizar os experimentos, utilizou-se armazenamento de resultados em disco (com identificadores únicos definidos pelos parâmetros) e carregamento preguiçoso dos resultados, de forma que os experimentos rodados alguma vez em uma certa máquina não precisassem ser rodados novamente caso se desejasse rever os resultados.

A geração de gráficos para análise de dados durante os experimentos foi feita com a *gem* gruff.

3.3 Metodologia de avaliação

Numa aplicação prática, deseja-se que o algoritmo seja capaz de reconhecer os acordes tocados num fonograma com a maior acurácia possível. Para isso, é preciso decidir qual versão do algoritmo usar, onde uma versão é o algoritmo básico acrescentado de algum subconjunto das técnicas de aperfeiçoamento descritas na seção anterior.

Avaliar uma versão do algoritmo de reconhecimento de acordes significa, em geral, comparar as sequências de acordes produzidas por ele com uma base de anotações de referência - também chamada de *ground-truth* -, que é um conjunto de fonogramas com anotações feitas manualmente do acorde que é tocado em cada janela temporal de cada fonograma.

Para que se possa fazer essa comparação, é necessário, além de contar com uma base de anotações de referência, definir um critério de comparação de acordes, que, dadas uma classificação feita pelo algoritmo e uma anotação da base de referência, decide se o acorde foi classificado corretamente. É preciso também definir uma (ou mais) medida de avaliação, que atribui um valor à classificação feita em uma música baseada na comparação de todos os acordes classificados com os respectivos acordes de referência. Por fim, pode-se calcular o valor de tal medida para um subconjunto dos fonogramas da base de anotações de referência, obtendo, assim, uma avaliação de uma versão do algoritmo.

Nas subseções seguintes serão discutidas as escolhas feitas em relação à esses três elementos: *ground-truth*, *critério comparação de acordes* e *medidas de avaliação*. Também se discutirá a escolha do subconjunto da base de *ground-truth* no qual são feitas as medidas de avaliação quando o algoritmo envolve algum processo de aprendizado.

3.3.1 Ground-truth

Para o escopo deste trabalho, utilizou-se a base de anotações de referência apresentada em Harte (2010), cujos fonogramas são a discografia de estúdio da banda The Beatles, que consiste em 180 faixas distribuídas em 13 CDs com um total de 8 horas, 8 minutos e 53 segundos de áudio.

Os arquivos dessa base seguem o formato .lab (compatível com programas como Sonic Visualiser e Wavesurfer), que é um arquivo de texto ASCII cujas linhas são compostas de três itens separados por espaço:

$$\langle \text{início} \rangle \ \langle \text{fim} \rangle \ \langle \text{etiqueta} \rangle$$

Início e *fim* são números de ponto flutuante que indicam em que momento no tempo (em segundos) o acorde começou e terminou de soar, respectivamente. *Etiqueta* é uma cadeia de caracteres que identifica tal acorde.

As etiquetas seguem uma notação definida detalhadamente em Harte (2010), onde cada etiqueta possível determina apenas um acorde.

3.3.2 Comparação de acordes

Um critério de comparação entre um acorde classificado e um acorde de referência é uma função binária $E : (\Lambda, \hat{\Lambda}) \to \{0, 1\}$, onde $\hat{\Lambda}$ é o conjunto de todos os acordes existentes e $\Lambda \subset \hat{\Lambda}$ é o conjunto 3.1. Dizemos que um acorde λ_i *passou* no critério de comparação com $\hat{\lambda_i}$ se $E(\lambda_i, \hat{\lambda_i}) = 1$, onde $\hat{\lambda_i}$ é o respectivo acorde da base de anotações de referência.

Um possível critério é aquele em que $E(\lambda_i, \hat{\lambda}_i) = 1 \Leftrightarrow notas(\lambda_i) = notas(\hat{\lambda}_i)$. Esse critério é intuitivamente válido, porém, se Λ não é muito grande, ele pode não capturar casos em que gostaríamos de considerar a classificação correta.

Tomemos como exemplo o acorde Am7 $\notin \Lambda$. Se Am7 é o acorde de referência para uma janela de sinal e o algoritmo classificar tal janela com Am $\in \Lambda$, se poderia considerar essa classificação como um acerto, já que esses acordes são iguais exceto por uma nota acrescentada (sol) em Am7. De fato, isso poderia se aplicar a todos os acordes que são tríades com notas acrescentadas.

Por essa razão, o critério de comparação de acordes escolhido foi E_0 tal que:

$$E_0(\lambda_i, \hat{\lambda}_i) = 1 \Leftrightarrow notas(\lambda_i) \subseteq notas(\hat{\lambda}_i)$$

Essa escolha tem ainda outras consequências que valem atenção. Em alguns casos, acordes possuem um conjunto de notas que são subconjuntos do de outros acordes cujas tônicas não são a mesma. Um exemplo desse caso são os acordes G $\in \Lambda$ e Em7 $\notin \Lambda$. Se observa que $notas(Em7) = \{2, 4, 7, 11\}$ e $notas(G) = \{2, 7, 11\}$, e, por isso, $notas(G) \subset notas(Em7)$, o que configura um acerto no critério E_0. No entanto a tônica de G é a nota sol, enquanto que a tônica de Em7 é a nota mi.

Essa consequência de E_0 não foi considerada como inadequada no escopo deste trabalho. A condição $notas(\lambda_i) \subseteq notas(\hat{\lambda_i})$ é suficiente para que ambos acordes possuam certa semelhança sonora. Considerando o conjunto Λ escolhido, é necessário certa tolerância em relação à consideração de acertos, para que se possa obter números efetivamente comparáveis quando se experimente uma técnica de aperfeiçoamento do algoritmo.

3.3.3 Precisão

É chamado de *classificação* o resultado do algoritmo para algum fonograma específico. Um *verdadeiro positivo* é uma janela de uma classificação cujo acorde passou no critério de comparação com o respectivo acorde da base de anotações de referência. Por sua vez, é chamada de *falso positivo* a janela cujo acorde classificado, analogamente, *não passou* no critério de comparação.

Definem-se os conjuntos VP e FP de todos os verdadeiros positivos e falsos positivos, respectivamente, de uma classificação. Então, a precisão P de uma classificação é calculada da seguinte maneira:

$$P = \frac{\#VP}{\#VP + \#FP}$$

3.3.4 Validação cruzada K-fold

Algumas versões do algoritmo de classificação de acordes baseada em templates podem depender de um processo de aprendizado. Por exemplo, quando se utilizam templates aprendidos em vez de binários, é necessário que algum subconjunto da base de *ground-truth* seja usado para a computação dos templates.

Nesses casos, é necessária atenção no momento de avaliar o algoritmo. Se levarmos em consideração a precisão da classificação de músicas que foram usadas no processo de aprendizado dos acordes, teremos uma situação conhecida no contexto de aprendizagem computacional como *overlapping*. Avaliações com *overlapping* não podem ser consideradas válidas, pois utilizam dados iguais para construção e validação do algoritmo.

Uma possibilidade para a avaliação de tais versões é o uso de um processo chamado *k-fold cross validation* ou *validação cruzada k-fold*. Ele consiste na divisão da base de *ground-truth* em k partes iguais (ou *dobras*), aleatoriamente. O processo de avaliação é feito k vezes, cada uma delas usando uma das dobras como conjunto de validação (e as outras $k - 1$ como conjunto de aprendizagem).

Quando avaliada dessa forma, uma versão do algoritmo terá na verdade k avaliações, e pode-se escolher uma delas de acordo com algum critério - em geral, utilizou-se a melhor avaliação obtida.

3.4 Experimentos

Esta seção descreverá os experimentos feitos para tentar obter a melhor versão do algoritmo de classificação de acordes baseada em templates, além de estudar as razões que diferenciam os resultados entre distintas versões do algoritmo.

Como são muitas as técnicas de aperfeiçoamento do algoritmo, foi necessário implementá-lo de uma forma geral, que permitisse a definição da versão através de passagem de parâmetros, além de um esquema de armazenamento de resultados que permitisse fácil identificação e comparação de resultados.

Os experimentos foram feitos seguindo este esquema:

1. Definição de uma versão do algoritmo
2. Construção dos templates para essa versão
3. Classificação das faixas no(s) conjunto(s) de fonogramas de validação

4. Avaliação das classificações (cálculo de precisão)

5. Avaliação da versão (em geral definida como média das classificações)

6. Análise de resultados

A etapa de análise de resultados envolve comparação da avaliação entre diferentes versões e verificações de precisões específicas dentro da amostra de precisões obtidas.

A distribuição das precisões obtidas nos experimentos não se mostrou satisfatória num primeiro momento. Além de estar longe de uma distribuição uniforme - o que seria desejável, pois traria uma maior previsibilidade quanto à qualidade do reconhecimento do algoritmo de forma geral - houve também ocorrências de precisões muito próximas de zero.

Algumas intuições foram obtidas através das análises individuais dos resultados. Se fará a descrição desses casos na monografia final (como a canção Lovely Rita que está afinada com Lá em 430 Hz).

Para comparar os experimentos, se calculou a média das precisões obtidas e, em alguns casos, se observou a distribuição das precisões. Na monografia final se descreverão análises de como as técnicas de aperfeiçoamento influenciaram na alteração dos resultados. Nesta versão, nos restringiremos à apresentação, no final desta seção, de algumas figuras que resumem alguns dos resultados, com breves descrições nas respectivas legendas.

Em particular, se começará com a partição das versões em duas: as que fazem o uso da STFT e as que fazem o uso da CQT.

Em seguida, se analisará a diferença entre o uso de templates binários e templates aprendidos.

Depois, se dará foco à suavização temporal. Logo, à suavização espectral e, por último, à pós-filtragem e estimativa de afinação, que não apresentaram resultados tão significativos quanto as outras técnicas.

Os gráficos a seguir trazem dados brutos dos experimentos realizados, que serão melhor apresentados e discutidos na versão final.

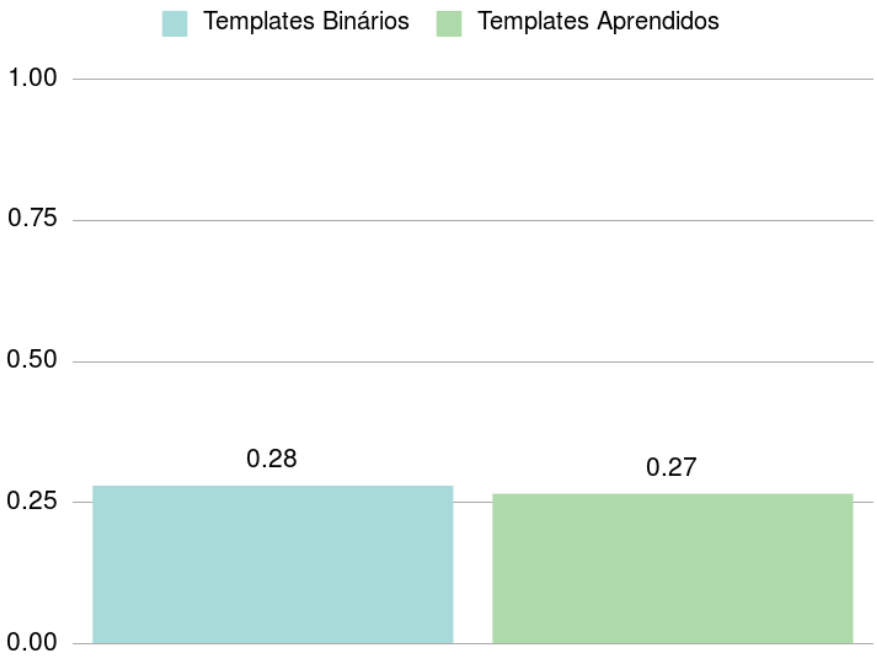

Figura 3.1: *Precisões médias obtidas com uso de templates binários e aprendidos.*

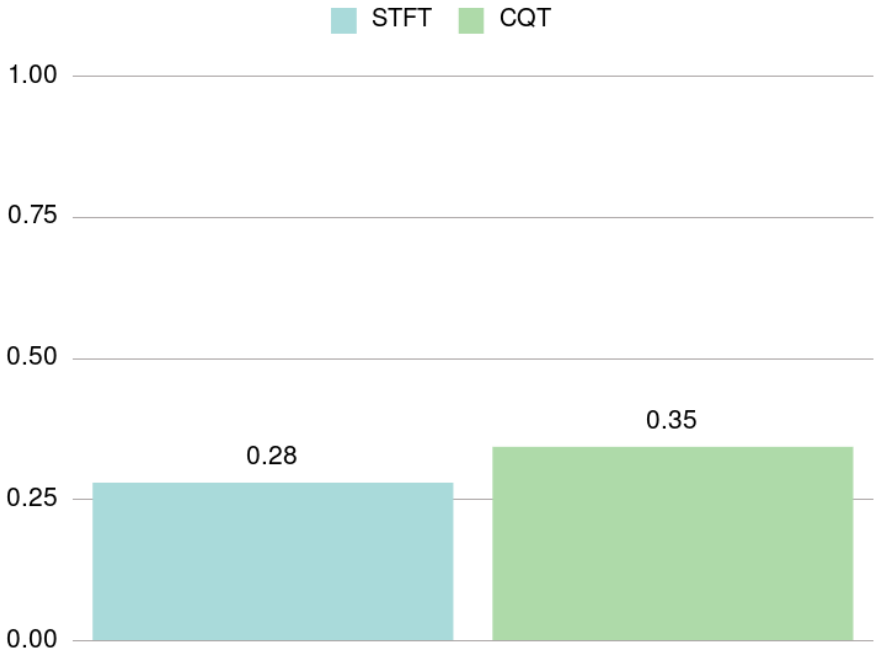

Figura 3.2: *Precisões médias obtidas com uso de STFT e CQT para construção do cromagrama.*

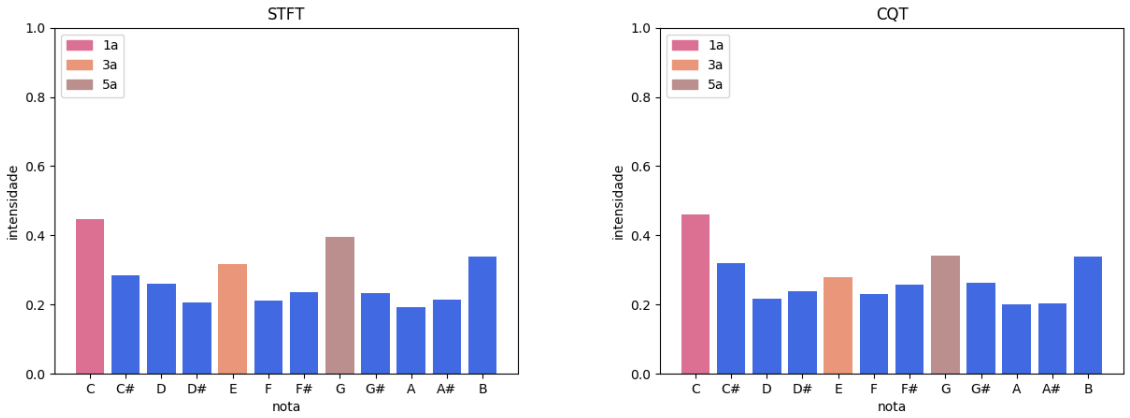

Figura 3.3: *Visualização do cromagrama template construído a partir de cromas STFT e CQT do acorde* ***dó maior*** *(C). As notas marcadas com cores diferentes indicam as notas presentes no acorde. Nota-se que algumas delas possuem menos energia que outras notas, o que se concluiu que acontece por causa da presença de muitos harmônicos nos sinais de áudio usados para treinamento.*

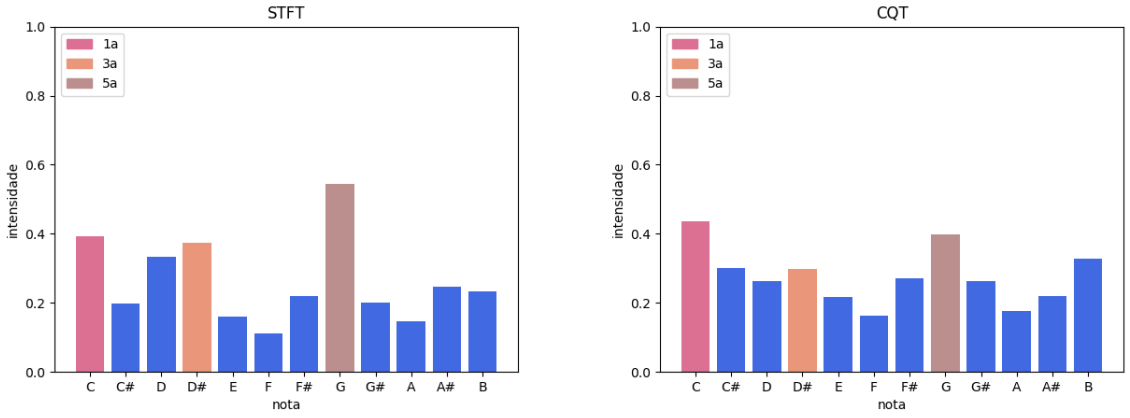

Figura 3.4: *Visualização do cromagrama template construído a partir de cromas STFT e CQT do acorde* ***dó menor*** *(Cm). As notas marcadas com cores diferentes indicam as notas presentes no acorde.*

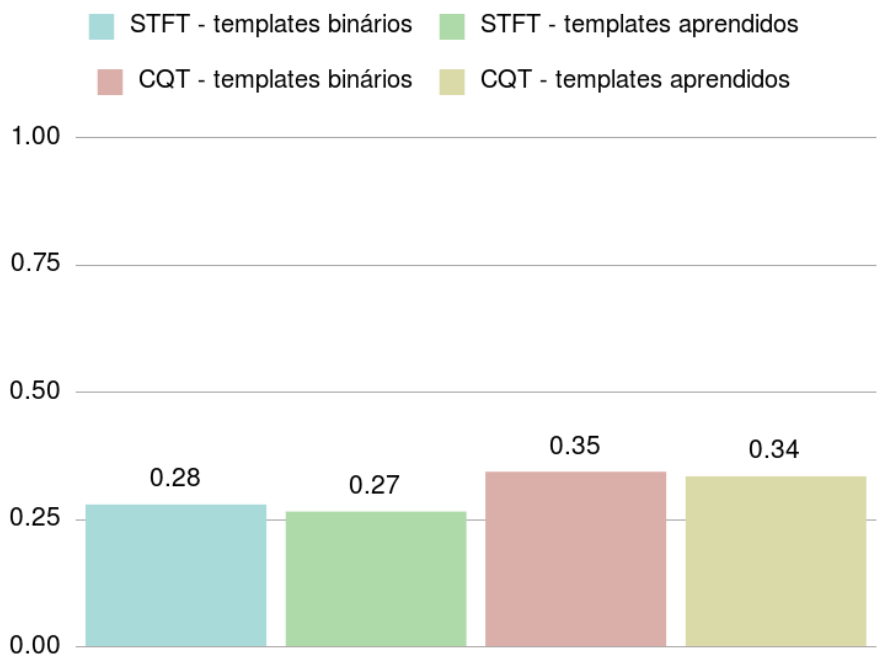

Figura 3.5: *Comparação entre a precisão média obtida por cromagramas construídos usando STFT, CQT e templates binários e aprendidos.*

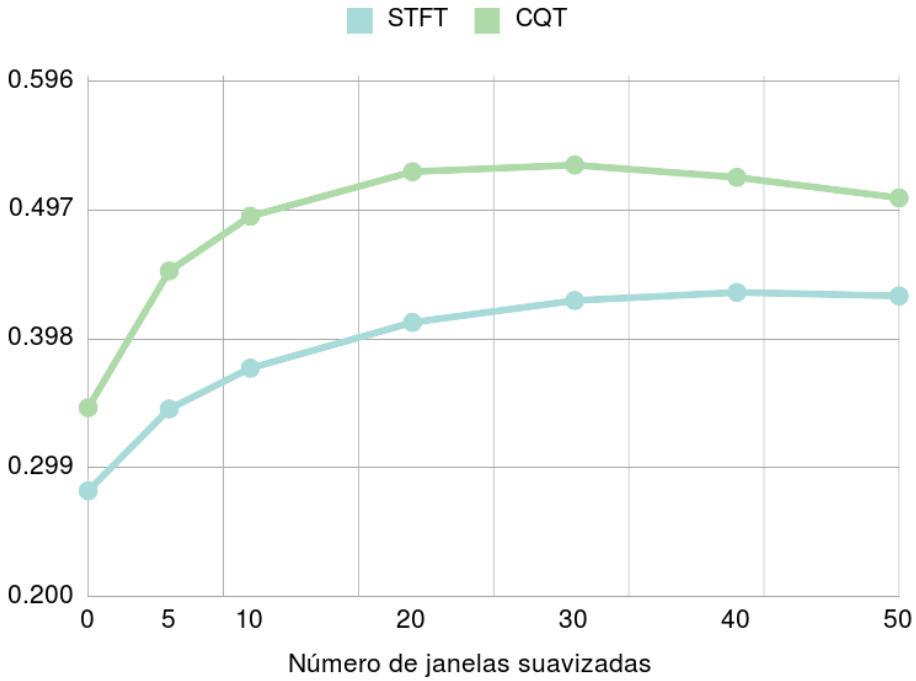

Figura 3.6: *Impacto da suavização temporal com diferentes parâmetros sobre a precisão média do algoritmo.*

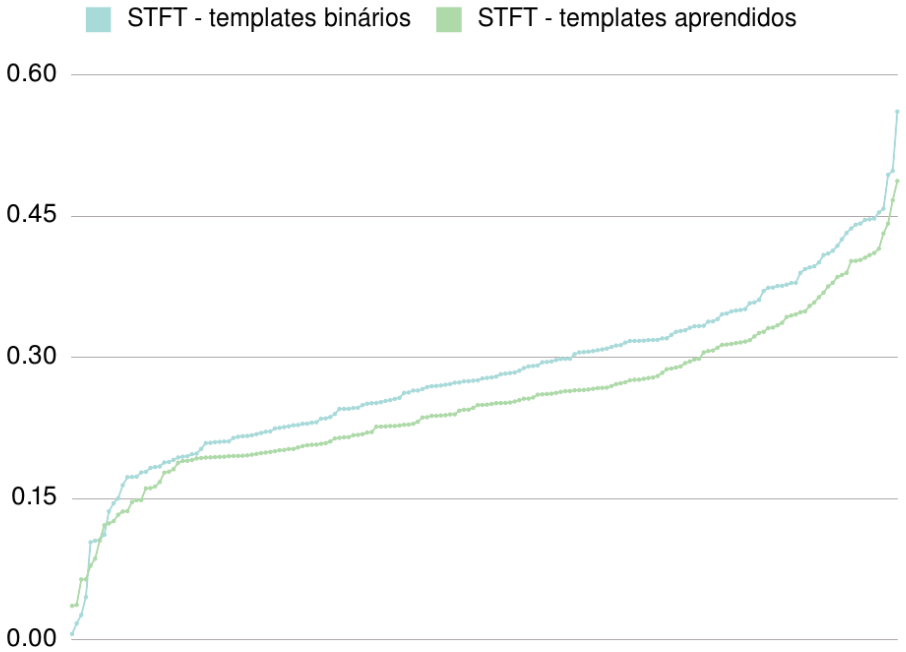

Figura 3.7: *Distribuição das precisões obtidas com STFT.*

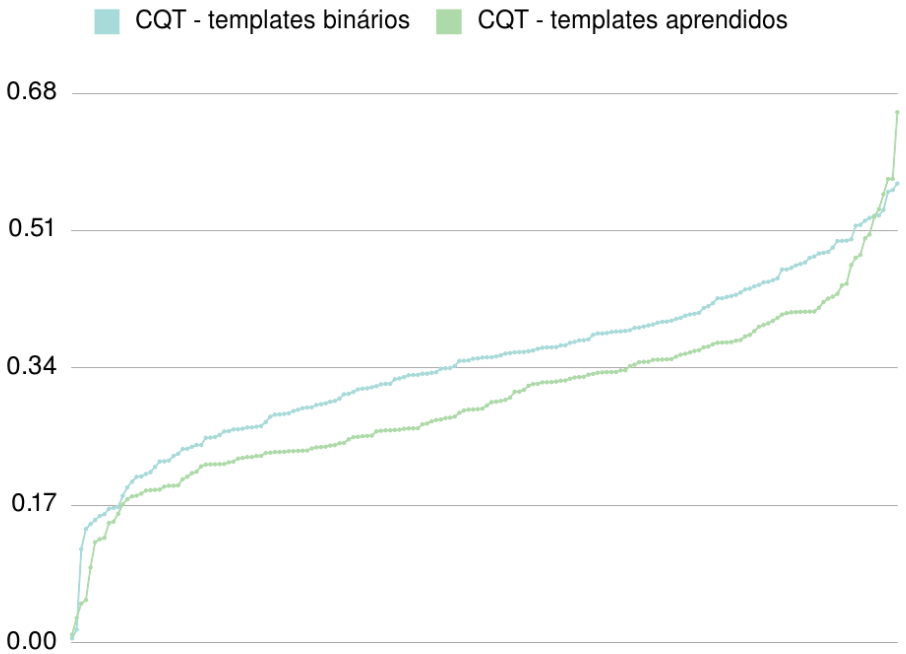

Figura 3.8: *Distribuição das precisões obtidas com CQT. Observe que as precisões mais baixas estão muito próximas de zero. Esse padrão se repete em todas as parametrizações do algoritmo, o que motivou uma análise caso a caso.*

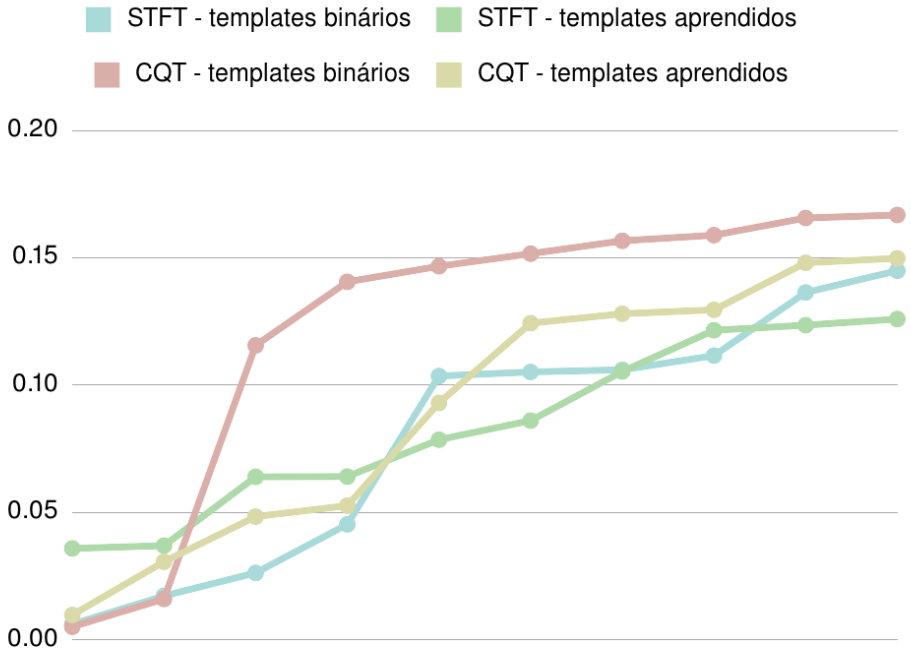

Figura 3.9: *Distribuição das 10 piores precisões obtidas com diferentes versões do algoritmo. Nenhuma das técnicas de aperfeiçoamento implementadas nessas versões trouxe uma distribuição mais equilibrada: em todas, se observaram precisões muito próximas de zero.*

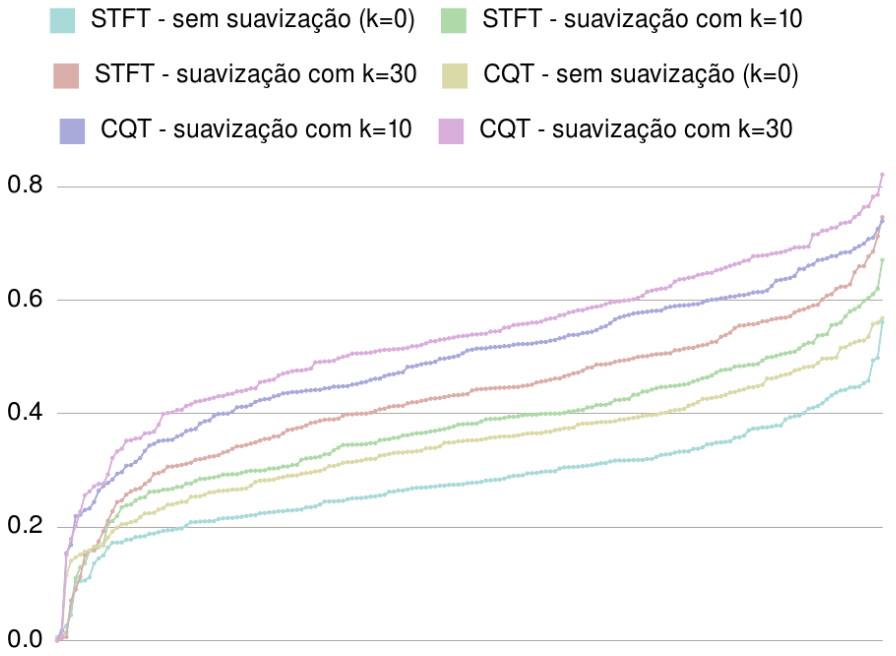

Figura 3.10: *Impacto da suavização temporal na distribuição das preciões obtidas nos experimentos.*

Capítulo 4

Conclusões

As conclusões deste trabalho serão escritas na monografia final, onde se certificará de relembrar o contexto geral do problema, comparar os objetivos iniciais e os resultados obtidos e apresentar perspectivas de trabalhos futuros, com melhorias e caminhos não explorados que o leitor interessado poderá seguir.

Referências Bibliográficas

Broughton e Bryan(2011) S Allen Broughton e Kurt M Bryan. *Discrete Fourier analysis and wavelets: applications to signal and image processing*. John Wiley & Sons. Citado na pág. 5

Harte(2010) Christopher Harte. *Towards automatic extraction of harmony information from music signals*. Tese de Doutorado, Queen Mary, University of London. Citado na pág. 12

Librosa(2017) Librosa. A python library for audio signal processing and music analysis. https://librosa.github.io/, 2017. Citado na pág. 11

Müller(2015) Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer. Citado na pág. i, iii, 1, 5, 7