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

Carlos Augusto Motta de Lima

Reconhecimento Automatizado de Acordes em Sinais de Áudio

São Paulo
Fevereiro de 2018

Reconhecimento Automatizado de Acordes em Sinais de Áudio

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Marcelo Queiroz

São Paulo
Fevereiro de 2018

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 52%, 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 representing of a song, a label sequence is produced, where each label represents a chord and its respective onset and offset times. In this work, we describe the problem of chord recognition in a detailed way, going through the involved concepts, and discuss and implement a particular solution presented in Müller (2015). We also discuss a method for evaluating such an algorithm, through the computation of comparison statistics 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 52%, 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

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 Harte (2010); Müller (2015).

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.

Atualmente já existem aplicações capazes de fazer o reconhecimento de acordes de forma eficiente, como o Chordata[1], do projeto CLAM[2], ou o Chordify[3].

Neste cenário, mostrou-se desejável familiarizar-se com o problema e conhecer suas possíveis soluções. 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 dos capítulos introdutório e conclusivo, o conteúdo desta 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 sua definição. 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 suas limitações e técnicas para melhorar sua eficácia. Também será discutida uma metodologia de avaliação do algoritmo e

[1] http://clam-project.org/wiki/Chordata_tutorial
[2] http://clam-project.org/index.html
[3] https://chordify.net/

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 estudados, passando pelas áreas de teoria musical (como a definição de um acorde a 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 importante retomar 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 periódicos 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 infinitas senoides com diferentes amplitudes e frequências (ver Müller, 2015, seção 2.1).

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 não trivialmente relacionadas: é o caso do som de um pandeiro, por exemplo. Outros sons, aos quais conseguimos associar uma altura determinada, denominamos de **notas musicais** às alturas correspondentes. Assim a um som de um piano que se repete de forma aproximadamente igual 440 vezes por segundo associamos a nota Lá (ou A4).

Define-se como **intervalo** a diferença de altura entre duas notas musicais. Sabe-se que essa diferença no espaço perceptual de alturas corresponde a uma razão entre as respectivas frequências (em Hz), ou seja, diferenças iguais de alturas correspondem a razões iguais entre as frequências das notas correspondentes. Diz-se portanto que a relação entre alturas e frequências é logarítmica.

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

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.*

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. 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 espaçadas igualmente, como se pode observar na Tabela 2.1. A razão entre as frequências de duas notas sucessivas nessa escala é $\sqrt[12]{2}$, de tal forma que 12 passos aditivos na altura (multiplicativos na frequência) perfazem uma oitava (relação $2:1$). Considerando-se a equivalência das oitavas, pode-se dizer que a escala diatônica é cíclica: partindo-se de uma nota de frequência f e aumentando-a em um semitom 12 vezes, se alcança a oitava dessa nota, de frequência $2f$. 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. Por razões históricas, usam-se 7 nomes de notas 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 o **sustenido** (ou ♯, que acrescenta a uma nota um semitom) e o **bemol** (ou ♭, que diminui uma nota de um semitom).

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. Considerando-se a equivalência de oitavas, podemos simplificar essa definição como um conjunto de classes de altura que soam ao mesmo tempo. Dentro dessa simplificação, um acorde pode conter no máximo 12 elementos, que são as classes de altura existentes. Dessa forma, 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.

Com tal restrição, nota-se que o número de acordes existentes é finito, porém não é pe-

queno. No entanto, um subconjunto dos acordes existentes teve destaque no contexto deste trabalho: as **tríades**, cujo uso na música ocidental folclórica e popular é ubíquo. Tríades são acordes de três notas construídos por superposição de intervalos de terças maiores e menores (respectivamente 4 e 5 semitons). Existem seis principais tipos de tríades, identificados conforme a disposição dos intervalos entre suas notas. No contexto deste trabalho, se restringiu a dois tipos principais: as tríades **maiores** e as tríades **menores**.

Uma forma de nomear um acorde que é uma tríade é identificando duas características: em primeiro lugar, o tipo de tríade que o acorde caracteriza; e, em segundo, a nota a partir da qual se começa a contagem dos intervalos que configuram tal tipo de tríade. Por exemplo, se os intervalos de um acorde configuram uma tríade menor (caracterizada por uma terça menor seguida de uma terça maior) e a contagem dos mesmos começaram da nota ré, então este acorde se chama **ré menor**. Por outro lado, se os intervalos de outro acorde, contados a partir da nota sol, configuram uma tríade maior (caracterizada por uma terça maior seguida de uma terça menor), então o acorde tem o nome de **sol maior**.

Existem outras formas de nomear um acorde, além de algumas convenções de notação para representá-lo. Aqui, nos restringiremos às nomenclaturas exemplificadas acima.

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 ao valor da pressão em pontos espaçados uniformemente no tempo. O número de amostras presentes em um segundo de áudio é chamado de **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 presentes em 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 dado sinal sonoro em componentes senoidais.

A transformada de Fourier de um sinal é uma representação de pressão (amplitude) 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 sinal 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 outra transformada: a 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 desde uma perspectiva perceptual. Ou seja, todas as notas musicais terão um mesmo número de representantes no espectrograma, o que não penaliza a amplitude total 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 um croma, obteremos uma matriz chamada **cromagrama**. Cromagramas são análogos a espectrogramas, com a diferença de que um apresenta as intensidades de componentes senoidais presentes no sinal, enquanto o outro apresenta intensidades de classes de altura. Cromagramas são de fundamental importância no contexto de reconhecimento de acordes em sinais de áudio porque capturam informações harmônicas do sinal. A construção de um croma, em alto nível, segue os seguintes passos:

1. Obtenção da DFT do sinal;
2. Identificação dos *bins* da DFT cuja frequência corresponda a uma nota da classe de altura C;
3. Soma dos valores desses *bins*;
4. Se repete o passo 2 e 3 para as outras 11 classes de altura.

Alternativamente, se pode usar outras transformadas além da DFT para a construção do croma de um sinal, como a CQT, por exemplo.

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 características (*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 $x_i \in R^{12}$ é um croma 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:

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

onde classes(λ) é o conjunto de índices das classes de altura presentes no acorde λ na sequência de classes que começa com a classe dó e segue aumentando cada elemento em um semitom até a classe si. Formalmente, se λ é a tríade maior construída a partir da classe de altura $n \in \{0, \ldots, 11\}$ então classes(λ) $= \{n, (n+5)\%12, (n+7)\%12\}$, enquanto a tríade menor a partir de n corresponde a classes(λ) $= \{n, (n+4)\%12, (n+7)\%12\}$.

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 cromas 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. Num sinal de áudio que representa uma música, pode haver uma quantidade qualquer de acordes que não estão presentes em Λ.

Outro fator que diminui a robustez 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 na seção seguinte.

Por essas e outras razões, algumas técnicas de aperfeiçoamento do algoritmo foram experimentadas para melhorar sua acurácia. Essas técnicas serão descritas nas subseções seguintes.

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, 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 representado por apenas uma senoide. Em primeiro lugar, sempre haverá a presença de ruído, que, por menos intenso que seja, trará energia a frequências que não pertencem à formação do acorde tocado. Além disso, haverá também a presença dos harmônicos, que agregam energia (em diversos casos, em quantidade considerável) a diferentes frequências do espectro sonoro. Por fim, a própria digitalização do som e extração do cromagrama são processos que pressupõem discretizações e acréscimo de ruídos ao sinal sonoro original.

Por essas razões, o template binário não é a representação mais fiel para nosso objetivo. Diante disso, uma alternativa natural é 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 sinais de áudio, podemos agregar os cromas das janelas agrupando-as por acorde e produzir um template para cada acorde que aparece no conjunto analisado.

Usando esse tipo de abordagem, é esperado que os templates de croma estejam mais próximos de cromas extraídos de sinais sonoros do mundo real, o que melhoraria as correlações obtidas na etapa de casamento de padrões do algoritmo de classificação.

A forma mais simples de aprender o template para um acorde é calculando a média simples de cada componente dos cromas cujas janelas estão anotadas com esse acorde. Tomando

o conjunto T_λ de todos os cromas extraídos dos sinais de áudio cujas janelas estão anotadas com o acorde λ, podemos definir a 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

As transformadas CQT e DFT possuem efeitos parecidos: 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 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 logarítmica

É comum que determinadas classes de altura presentes num acorde possuam menos energia em seu croma que outras classes ausentes no mesmo. Isso ocorre, por exemplo, devido à grande quantidade de energia que alguns instrumentos produzem em harmônicos das notas do acorde.

Para que a distribuição da energia num croma seja mais uniforme, pode-se aplicar uma técnica conhecida como *compressão espectral*. Essa técnica consiste na aplicação de uma função de compressão Γ_c em todos os elementos do croma, onde c é o *fator de compressão*, que é proporcional a quão uniforme será o croma após a compressão. O ganho esperado com essa técnica é a enfatização de classes de altura que possuem pouca energia, porém são perceptíveis na harmonia da música. A função de compressão usada, conforme Müller (2015), foi:

$$(\Gamma_c(x))_i = \log(1 + cx_i).$$

Encontrar uma constante c adequada é de fundamental importância: valores muito pequenos de c farão com que a compressão não interfira muito no resultado da classificação, enquanto que valores muito grandes descaracterizarão a harmonia presente nos cromas, tornando-o demasiadamente uniformes.

3.1.4 Suavização temporal

Em muitos sinais de áudio 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 acorde de dó maior (que é formado pelas notas dó, mi e sol) pode ser tocado de forma sequencial, "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 se aplicar essa técnica, define-se 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: uma frequência de referência para uma nota específica, a partir da qual se calculará a frequência correta de todas as notas.

Em geral, a afinação utilizada como referência na música ocidental nos dias atuais é a nota chamada **Lá 4** 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), é importante se saber previamente qual afinação é utilizada, de forma que a construção do cromagrama seja a mais precisa possível.

Sem pressupor 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 fazer essa estimativa, que é o método padrão da biblioteca escolhida (vide Seção 3.2) para processamento de áudio.

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 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 dois décimos de segundo).

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

3.2 Implementação

O objetivo deste trabalho não é a implementação de um sistema de reconhecimento de acordes, mas sim o estudo 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 a integração do pacote escrito em Python num software escrito em Ruby, utilizou-se a *gem* PyCall[1], que possibilitou a adição de uma interface para o Librosa de forma direta.

O código[2] 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].

3.3 Metodologia de avaliação

Numa aplicação prática, deseja-se que o algoritmo seja capaz de reconhecer os acordes tocados num sinal de áudio 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 3.1.

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 sinais com anotações feitas manualmente do acorde que é tocado em cada janela temporal de cada sinal.

[1] https://rubygems.org/gems/pycall/versions/1.0.3
[2] https://github.com/gutomotta/chors
[3] https://github.com/topfunky/gruff

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 sinais 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 a esses três elementos: *ground-truth*, *critério de 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 (registros sonoros em sinais de áudio) formam 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:

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

onde *início* e *fim* são números de ponto flutuante que indicam em que momento no tempo (em segundos) o acorde começou e terminou, respectivamente, e *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 definido na Equação 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 $E(\lambda_i, \hat{\lambda}_i) = 1 \Leftrightarrow \text{classes}(\lambda_i) = \text{classes}(\hat{\lambda}_i)$. Este 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:

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

Essa escolha tem ainda outras consequências que valem atenção. Em alguns casos, acordes possuem um conjunto de classes de altura 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 classes$(Em7) = \{2, 4, 7, 11\}$ e classes$(G) = \{2, 7, 11\}$, e, por isso, classes$(G) \subset$ classes$(Em7)$, 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 classes$(\lambda_i) \subseteq$ classes$(\hat{\lambda_i})$ 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 sinal específico. Um *verdadeiro positivo* é uma janela de uma classificação cujo acorde classificado 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.

Na base de anotações de referência, algumas janelas são marcadas como *sem acorde*. Isso acontece quando o trecho de áudio não possui conteúdo harmônico significativo (por exemplo, em um solo de bateria ou em um trecho onde só se podem ouvir aplausos). Janelas sem acorde foram descartadas nas avaliações feitas nos experimentos deste trabalho.

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:

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

O cálculo da precisão poderia, alternativamente, atribuir um peso a cada verdadeiro positivo e falso positivo proporcionais à duração de suas respectivas janelas, somando os pesos dos verdadeiros positivos e finalmente dividindo pela duração total do sinal. No contexto deste trabalho, as janelas têm duração fixa e, por isso, não se utilizaram pesos distintos.

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. No experimentos realizados neste trabalho, sendo necessária a produção de uma avaliação única, tomou-se sempre a média simples das k avaliações.

3.4 Experimentos

Esta seção descreve os experimentos feitos para analisar os resultados das diferentes versões do algoritmo de classificação de acordes baseada em templates, procurando justificar as eficácias obtidas em cada versão.

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 (definida como média das precisões das classificações);
6. Análise de resultados.

A etapa de análise de resultados consistiu na comparação da avaliação entre diferentes versões e na verificação de precisões específicas dentro do conjunto de precisões obtido. Para comparar versões entre si, se calculou a média das precisões obtidas e, em alguns casos, se observou a distribuição das precisões. Em todas as versões analisadas, a distribuição das precisões obtidas nos experimentos não foi uniforme - o que seria desejável, pois traria uma maior previsibilidade quanto à qualidade do reconhecimento do algoritmo de forma geral. Além disso, se observaram ocorrências de precisões muito próximas de zero.

Nas subseções seguintes, são descritas análises de como as técnicas de aperfeiçoamento influenciaram na alteração dos resultados. Os experimentos, por padrão, foram feitos com uma taxa de amostragem de 22.050 Hz, janelas de 4096 amostras e saltos entre janelas de 2048 amostras.

3.4.1 Estimativa de afinação

Conforme explicado anteriormente, a extração do cromagrama de um sinal de áudio pressupõe o uso de um valor de frequência como referência de afinação da nota Lá 4. A qualidade do cromagrama obtido do sinal é impactada pelo valor de afinação utilizado como referência.

Nos experimentos feitos, observou-se casos em que a precisão obtida no reconhecimento de acordes foi muito próxima de zero. Um caso particular de resultado pouco satisfatório foi o da canção Lovely Rita, que, no fonograma processado, está afinada com Lá 4 em 425 Hz.

Tal afinação fez com que a maioria das janelas fosse classificada com acordes um semitom abaixo dos corretos, em todas as versões do algoritmo testadas.

Num experimento isolado, com uso de CQT para extração de cromagramas, foi possível obter uma precisão de 31.63% utilizando a afinação correta como base da extração de cromas. Utilizando a afinação estimada, obteve-se uma precisão de apenas 0.5%.

Em outros casos de afinação desviada, este cenário não se repetiu. Por exemplo, na canção Wild Honey Pie, cujo fonograma teve sua afinação estimada de Lá 4 em aproximadamente 426 Hz (ver Harte, 2010, tabela 9.8), se obteve uma classificação com precisão de 14.66% utilizando algoritmo de estimativa de afinação, contra 14.39% com a afinação fornecida como parâmetro. Neste caso, se observou que a estimativa algorítmica da afinação foi mais precisa que a estimativa manual.

3.4.2 Transformadas para extração do cromagrama

Nos experimentos deste trabalho, foram testados dois tipos de transformada na construção de cromagramas: STFT e CQT.

Conforme esperado, os resultados obtidos com a transformada Q-constante foram melhores, mas não muito significativamente, devido ao tamanho da janela usado (4096), que fez com que a resolução da STFT para notas graves não fosse tão baixa. Num primeiro momento, considerando apenas classificações com uso de templates de croma binários, a precisão média obtida com uso de STFT foi de 32.89%, contra 34.28% com uso de CQT.

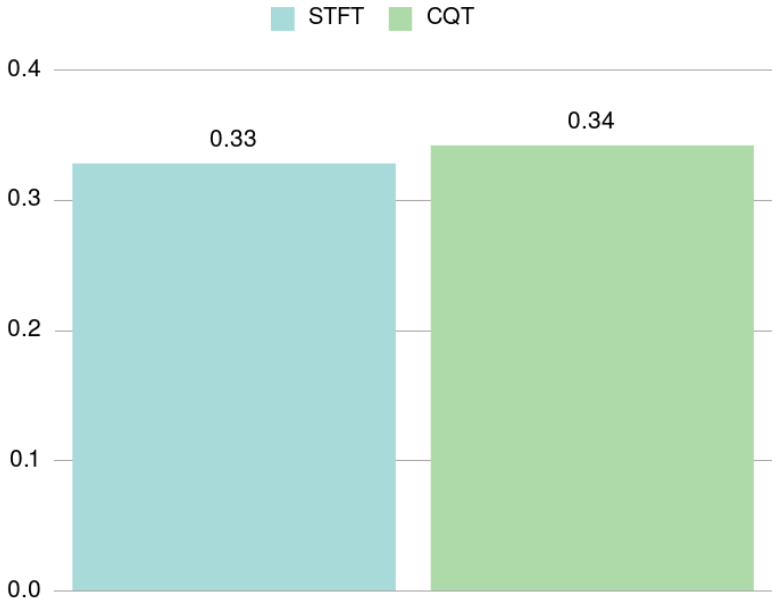

Figura 3.1: *Precisões médias obtidas com uso de STFT e CQT para construção do cromagrama.*

Os desvios-padrão obtidos em ambos os casos foram próximos, porém, com uso da CQT, o valor foi ligeiramente menor, o que indica que uma previsibilidade um pouco maior nesse caso. Os resultados completos podem ser vistos na tabela 3.1.

Transformada	Média	Desvio Padrão
STFT	32.89%	10.95%
CQT	**34.28%**	**10.12%**

Tabela 3.1: *Comparação das precisões obtidas com uso de duas transformadas diferentes na extração de cromagramas.*

3.4.3 Templates binários e templates aprendidos

Devido ao fato de que templates de croma binários não refletem as intensidades reais da distribuição de energia em classes de altura de acordes, espera-se, intuitivamente, que existam templates mais adequados para o reconhecimento de acordes.

No entanto, experimentalmente, se observou que a técnica apresentada de aprendizado de templates - a partir do cálculo da média dos cromas de janelas de áudio previamente rotuladas - não trouxe melhoria nas precisões obtidas.

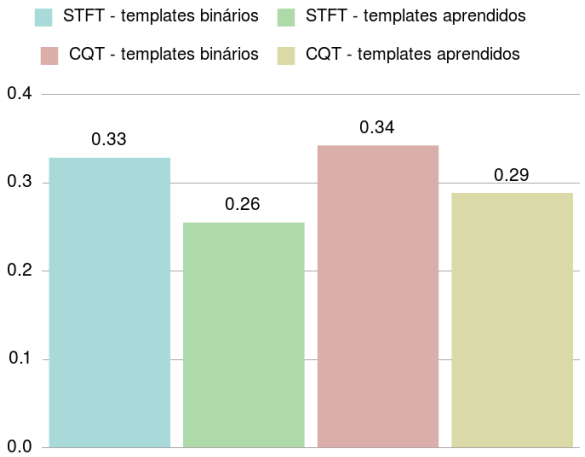

Figura 3.2: *Comparação entre a precisão média obtida com templates binários e aprendidos.*

Uma possível explicação para tal fato seria um erro de estimativa de afinação dos áudios (leva-se em conta que não se possui uma base de anotações de referência das afinações de cada áudio utilizado nos experimentos). Isso implicaria no agrupamento de cromas que foram extraídos com desvios de afinação e na consequente obtenção de um template médio de má qualidade.

Na figura 3.3, pode-se visualizar dois dos templates de croma do acorde dó maior, calculados através da técnica de aprendizado descrita. Três quartos da base de anotações de referência foram usados para o treinamento, e um quarto para a validação.

Nessa figura, percebe-se que as classes de altura de notas que fazem parte da formação do acorde (C, E e G) estão entre as que mais possuem energia no template produzido. Contudo, a intensidade de outras classes de altura, que não fazem parte do acorde mas pertencem aos harmônicos das notas do acorde, também são consideravelmente altas.

Na versão do algoritmo que faz uso de STFT, nota-se uma maior ênfase das classes destacadas, notando-se apenas uma classe (B) com intensidade próxima à de uma classe destacada (C). No versão de CQT, a distribuição da energia é mais uniforme, e diversas classes possuem energia próxima à presente nas classes destacadas, em especial E e G.

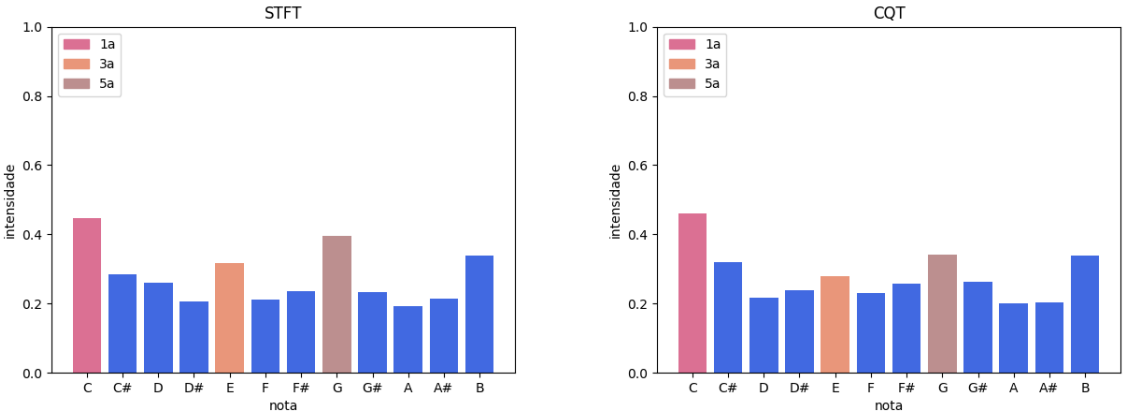

Figura 3.3: *Visualização do cromagrama template construído a partir de cromas STFT e CQT do acorde* ***dó maior*** *(C). As classes de altura destacadas com cores diferentes refletem as notas presentes no acorde. Nota-se que algumas classes ausentes no acorde (como B, harmônico tanto de G quanto de E) possuem mais energia que outras presentes.*

Tanto com STFT como CQT, os resultados pioraram quando se passou do uso de templates binários para o de templates aprendidos. Os resultados estão expostos na tabela 3.2.

Transformada	Templates	Média	Desvio Padrão
STFT	**binários**	**32.89%**	10.95%
STFT	aprendidos	25.55%	8.87%
CQT	**binários**	**34.28%**	10.12%
CQT	aprendidos	28.89%	10.3%

Tabela 3.2: *Resultados dos experimentos que usam templates binários contra templates aprendidos na etapa de casamento de padrões.*

3.4.4 Suavização temporal e pós-filtragem

A suavização temporal é uma técnica de aperfeiçoamento que tem como intenção a captura de harmonias cujos componentes soam ao longo de várias janelas e não necessariamente de forma simultânea. Além disso, se supõe que tal técnica mitigue erros de classificação que ocorrem em janelas isoladas, onde as janelas vizinhas são classificadas corretamente.

Conforme explicado anteriormente, a suavização temporal possui um parâmetro L que corresponde ao número de vizinhos (posteriores e anteriores) usados na suavização. Experimentou-se diversos valores distintos de L, e se observou que a melhoria gerada chega a um ponto ótimo, e depois as precisões voltam a cair. Intuitivamente, utilizar um número muito grande de janelas no cálculo dos cromas suavizados aumenta o risco de misturar o conteúdo de cromas que capturam o conteúdo harmônico de dois ou mais acordes, e consequentemente a probabilidade de erro.

O gráfico da figura 3.4 mostra a evolução das precisões médias obtidas com diferentes valores de L, comparando os resultados do uso das transformadas STFT e CQT.

Com a taxa de amostragem e tamanho de janela usados, cada janela temporal tem aproximadamente 180 ms. Os valores ótimos, em função do parâmetro L, observados nos experimentos, ocorreram para $L = 10$, o que implica que o número total de janelas suavizadas é 21, o que equivale a aproximadamente 3.9 s de áudio.

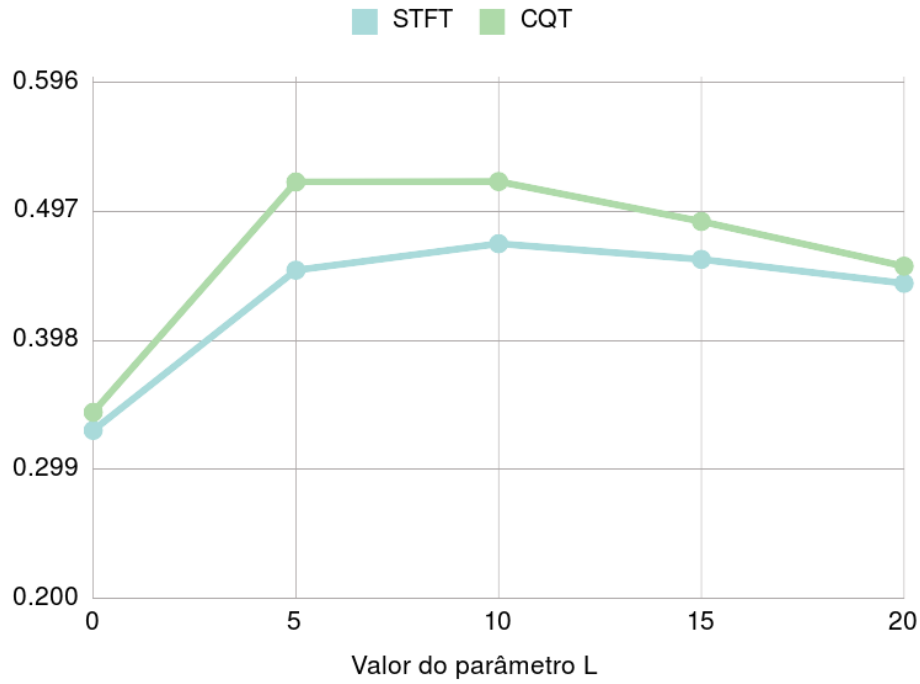

Figura 3.4: *Impacto da suavização temporal com diferentes parâmetros.*

É importante notar que os fonogramas usados nos experimentos são todos de uma mesma banda e que, ainda que sejam bastante diversos, o valor ótimo de L pode ser distinto em outros contextos; vale, no entanto, como uma estimativa.

A tabela 3.3 traz a precisão média observada para distintos valores de L, considerando o uso de STFT e CQT.

Valores de L	Precisão Média STFT	Precisão Média CQT
0	32.89%	34.28%
5	45.19%	51.96%
10	**47.22%**	**52.0%**
15	46.02%	48.93%
20	44.18%	45.5%
25	42.34%	42.39%

Tabela 3.3: *Resultados dos experimentos com templates binários, uso de STFT e CQT e distintos valores do parâmetro L para suavização temporal.*

Os experimentos feitos neste trabalho mostraram que a pós-filtragem conforme definida na subseção 3.1.6 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.

3.4.5 Compressão espectral

Durante a experimentação da técnica de compressão espectral logarítmica, se voltou a comparar templates aprendidos e binários. Essa comparação foi motivada pelos resultados

expostos em Müller (2015), figura 5.22, que sugerem que a melhoria no reconhecimento trazida pelo uso de templates aprendidos se apresenta apenas quando combinado com o uso de compressão de cromas.

Dessa forma, os experimentos desta técnica foram divididos em duas etapas: uma fazendo uso de templates binários, e outra, de templates aprendidos.

Os resultados obtidos com uso de templates binários não foram melhorados quando combinados com a etapa de compressão espectral logarítmica. Tal resultado era esperado, pois ao aumentar-se a compressão dos cromas extraídos do sinal de áudio, eles estarão cada vez mais próximos de um croma uniforme, e, portanto, menos similares aos templates binários.

No gráfico da figura 3.5, observa-se essa tendência. Ainda que haja uma pequena variação positiva da precisão obtida pelo algoritmo quando se usa STFT com compressão, essa variação é inexpressiva - de aproximadamente 0.25% no melhor caso (ou seja, com fator de compressão 5). Analogamente, pode-se dizer que as variações negativas trazidas pelo aumento do fator de compressão também são inexpressivas, o que indica uma provável injustificabilidade do uso de tal técnica (que possui um custo computacional) combinado ao de templates binários.

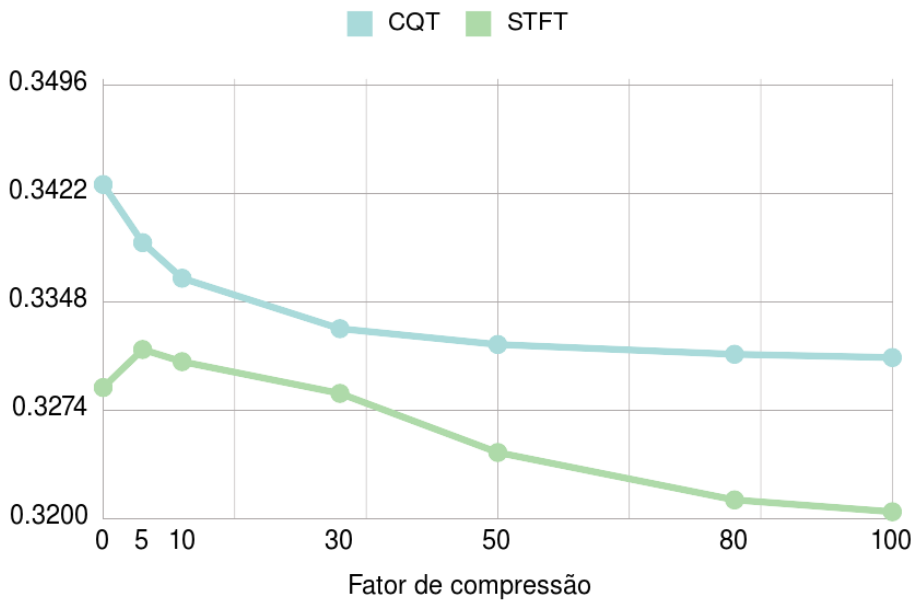

Figura 3.5: *Efeito da compressão espectral logarítmica nas versões do algoritmo que usam templates binários. Não se observou ganho obtido por esta técnica nestes casos.*

No caso de templates aprendidos com CQT, não se observou uma tendência consistentemente positiva nos resultados obtidos com uso de compressão espectral. Em realidade, o impacto causado pela compressão nesse caso foi bastante pouco significativo.

Já no caso de templates aprendidos com STFT, houve uma melhoria consistente, que chega num ápice com fator de compressão igual a 30 e diminui um pouco com fatores maiores. Tal efeito pode ser observado na figura 3.6.

Também se observou que a consistente melhoria trazida pela compressão quando combinada com templates aprendidos e STFT persiste quando é aplicada, posteriormente, a suavização temporal de cromas. No entanto, ainda que a compressão logarítmica melhore os resultados obtidos com uso de templates aprendidos, não foi possível superar os resultados

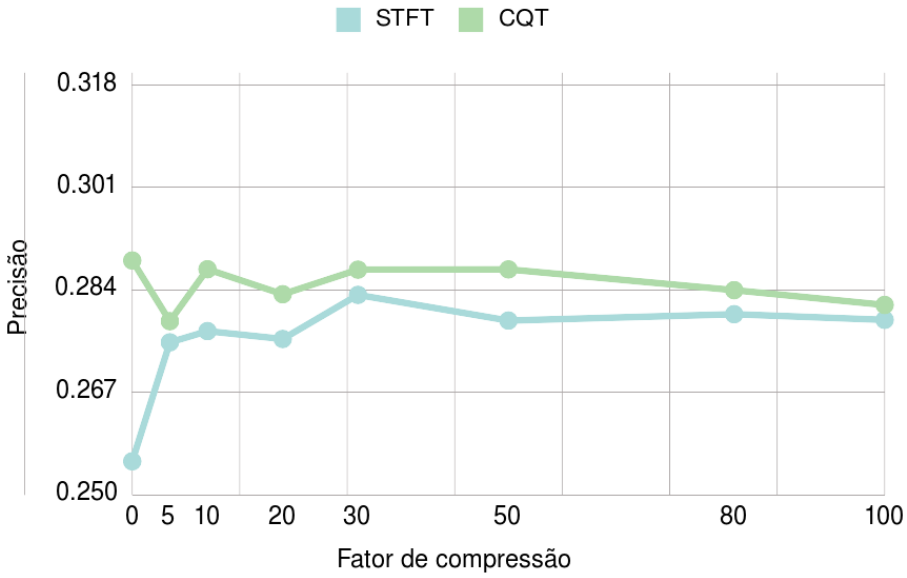

Figura 3.6: *Efeito da compressão espectral logarítmica nas versões do algoritmo que usam templates aprendidos. Note que o impacto da compressão é pouco significativo quando combinada uso de CQT, porém positivo quando combinada ao uso de STFT.*

dos templates binários (que, conforme mostrado anteriormente, não se beneficiam do uso de compressão). O gráfico 3.7 expõe tais resultados.

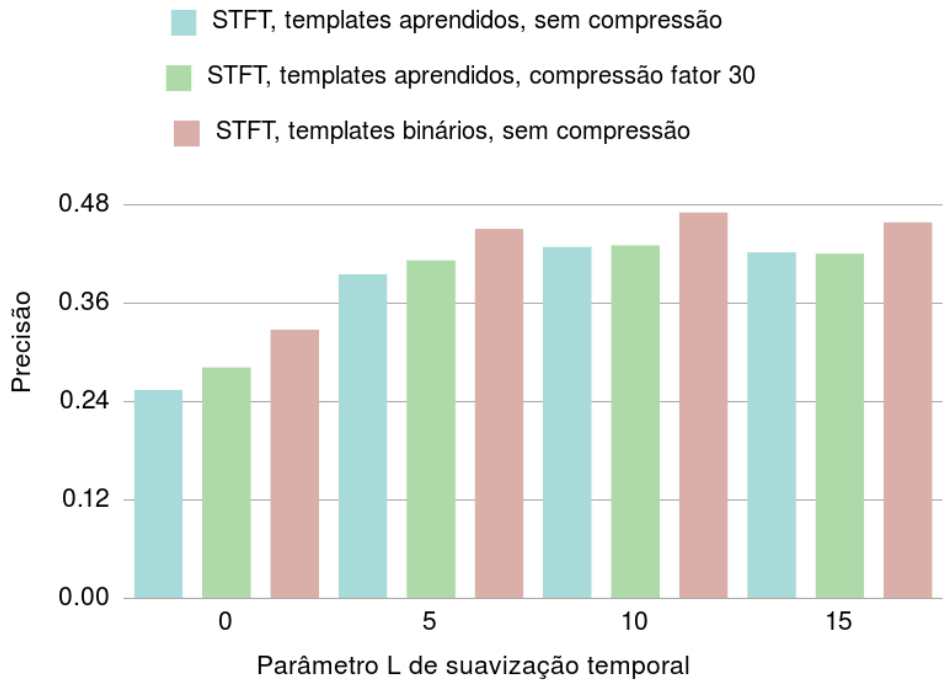

Figura 3.7: *Comparação entre as precisões obtidas em três cenários distintos que usam STFT: i) uso de templates aprendidos, sem compressão logarítmica; ii) uso de templates aprendidos, com compressão logarítmica (fator de compressão 30) e iii) uso de templates binários, sem compressão logarítmica. Os três cenários foram testados combinados com suavizações temporais com quatro valores diferentes para o parâmetro L: 0, 5, 10 e 15. Notou-se que a compressão beneficia os resultados obtidos com uso de STFT e templates aprendidos, mas não é suficiente para superar os resultados obtidos com templates binários.*

Capítulo 4

Conclusões

Neste trabalho, se estudou o problema de reconhecimento de acordes e uma solução particular para ele: a classificação baseada em templates. Se discutiram pontos do algoritmo em que se poderia experimentar técnicas para aperfeiçoá-lo. Se descreveu um método de avaliação do algoritmo que foi usado experimentalmente para decidir se as técnicas de aperfeiçoamento discutidas apresentariam ou não melhorias na prática.

Dadas as simplificações adotadas, consideraram-se satisfatórios os resultados obtidos, que alcançaram uma precisão média de 52% numa base de 180 fonogramas. Para algumas das possíveis técnicas de aperfeiçoamento descritas, não se pode observar experimentalmente uma melhoria nos resultados, como no caso do uso de templates aprendidos a partir de anotações de referência; para outras, pode-se observar um ganho significativo: como no caso da suavização temporal de cromas e no uso da transformada Q-constante em vez da transformada de Fourier, técnicas que se conclui serem essenciais para construir um algoritmo classificador de acordes baseado em templates.

Muitos caminhos ainda podem ser explorados no estudo de reconhecimento de acordes. O algoritmo apresentado não classifica intervalos do sinal sem acordes, e usa um conjunto pequeno de acordes passíveis de classificação - diversos acordes poderiam ser incluídos. Além disso, se poderia estudar algoritmos que fazem uso de técnicas de aprendizado de máquina para aperfeiçoar a classificação.

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. 1, 12, 15

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, 3, 6, 7, 9, 19