



**UNIVERSIDADE FEDERAL DA BAHIA**  
**INSTITUTO DE MATEMÁTICA**  
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**Alexandre Tachard Passos**

**Modelos Ocultos de Markov para Análise Harmônica  
Funcional de Música Tonal**

Salvador

2008

**Alexandre Tachard Passos**

# **Modelos Ocultos de Markov para Análise Harmônica Funcional de Música Tonal**

**Monografia apresentada ao Curso de graduação em Ciência da Computação, Departamento de Ciência da Computação, Instituto de Matemática, Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.**

**Orientador: Pedro Ribeiro Kröger Júnior**

Salvador

2008

# ***AGRADECIMENTOS***

À Universidade Federal da Bahia, por tornar possível esse trabalho.

À FAPESB e ao CNPq pelas bolsas de iniciação científica que culminaram nesse trabalho.

Ao, meu orientador, Pedro Kröger, por ter tido a idéia, coordenado o projeto, implementado boa parte do código do Rameau e discutido comigo por email regularmente. Sem essas discussões, quem sabe, eu perderia a prática, e poderia ser péssimo em *flames* hoje em dia.

Aos membros do grupo de pesquisa genos que trabalharam diretamente com o Rameau. Marcos Sampaio, por me corrigir muitos erros de harmonia e explicar vários detalhes confusos. Givaldo de Cidra, Nathanael de Oliveira e Wallace dos Santos, por terem feito vários gabaritos e me ajudado muito a processar e entender os dados todos produzidos pelo Rameau.

A meus pais, por todo o apoio que me deram durante todo esse tempo.

A Rodrigo Souto por ler esse texto com a perspectiva de alguém não-envolvido e me apontar alguns problemas importantes.

A Iara Malbouisson que, mesmo não revisando esse texto, revisou todas as outras coisas feitas por mim para essa pesquisa, e aguentou o peso de todas as ressacas morais, confusões mentais e maluquices que isso criou na minha cabeça.

Finalmente, gostaria de agradecer ao Roteirista, cujo senso de humor foi fundamental em toda a minha vida.

# ***RESUMO***

Em teoria musical, a atividade de identificar a tonalidade de uma música tonal e, no processo, determinar qual a função harmônica de cada acorde é chamada de análise harmônica funcional. Esse problema é ambíguo, subjetivo, possui várias interpretações discordantes e existem quase tantas opiniões e atitudes quanto a ele quanto existem praticantes. Por outro lado, a maior parte dos teóricos no assunto enxergam na análise harmônica uma estrutura objetiva e coerente.

Nesse trabalho analisamos a performance de um modelo de Markov oculto, um modelo probabilístico muito utilizado em problemas de reconhecimento de voz e modelagem de linguagem natural, como forma de automatizar esse processo. Assim, buscamos representar computacionalmente a estrutura objetiva, se existir, do problema. Além disso comparamos esse modelo com algoritmos heurísticos, redes neurais e um classificador k-nearest-neighbors. Os dados de performance são computados usando corais de Bach.

# ***ABSTRACT***

In tonal music theory, functional harmonic analysis is the process of, given a piece of tonal music, identifying its tonality and the tonal function of each chord in the piece. This problem is ambiguous, subjective, has many different accepted interpretations, and there are as many opinions and attitudes to its solution as there are practitioners. On the other hand, most theorists do see in it an objective structure that is commonly perceived as coherent.

In this study we analyze a Hidden Markov Model, a probabilistic model commonly used as a tool for automatic speech recognition and natural language modeling, as a means of automating this task. Thus, we represent computationally some of the structure of the problem. We also compare the analysis of this algorithm with the analyses obtained using heuristic algorithms, neural networks, and a K-nearest-neighbors classifier. The data used in this study is obtained from Bach chorales.

# ***LISTA DE FIGURAS***

2.1	Exemplos dos tipos de acordes considerados nesse trabalho. . . . .	21
2.2	Um exemplo de tonalidade. . . . .	22
3.1	Performance por volume de dados de treinamento. . . . .	29
3.2	Exemplo trivial. . . . .	29
5.1	Screenshots da interface do Rameau Web. . . . .	40
6.1	Análise do coral 130. . . . .	45

# ***LISTA DE TABELAS***

2.1	Notação de inteiros para classes de notas. . . . .	18
2.2	Intervalos. . . . .	18
2.3	Tipos de acordes considerados nesse trabalho. . . . .	20
2.4	Funções harmônicas da escala maior. . . . .	23
4.1	Modelos usados pelo algoritmo de Pardo. . . . .	33
6.1	Intervalo de 95% de confiança para taxa de acertos (em percentual). . . . .	43
6.2	Intervalo de 95% de confiança para a performance da rede neural nos dados de treinamento (em percentual). . . . .	43
6.3	Intervalo de 95% de confiança para performance na detecção de tonalidade (em percentual). . . . .	44
6.4	Intervalo de 95% de confiança para performance na detecção de acorde (em percentual). . . . .	44

# ***LISTA DE ALGORITMOS***

1	Algoritmo de Viterbi. . . . .	26
---	-------------------------------	----

# ***LISTA DE EXEMPLOS***

5.1	Código exemplo para definir um algoritmo. . . . .	41
5.2	Exemplo de código para definir um comando. . . . .	41
5.3	Exemplo de definição de uma busca musicológica. . . . .	42

# *LISTA DE NOTAÇÃO*

lá, si $\sharp$ , dó $\flat\flat$  são notas musicais,

A, Cm7, B $\flat$ °7 são acordes,

I, iv, vii° são funções tonais,

$A$  é uma variável aleatória,

$P(A)$  é a distribuição de probabilidade da variável aleatória  $A$ ,

$P(A = a)$  é a probabilidade da variável aleatória  $A$  ter o valor  $a$ ,

$P(A = a|B = b)$  é probabilidade de  $A$  assumir o valor  $a$  dado que  $B$  assumiu o valor  $b$ ,

$f_X(x)$  é a função de densidade probabilística da variável aleatória  $X$ ,

$f_X(x|Y = y)$  é a função de densidade probabilística da variável aleatória  $X$  dado que  $Y$  assume o valor  $y$ ,

$\max_x F(x)$  é o valor máximo de  $F(x)$ ,

$\arg \max_x F(x)$  é o valor de  $x$  que maximiza  $F(x)$

# ***LISTA DE TERMOS***

Um **segmento mínimo** é um conjunto de notas musicais soando ao mesmo tempo;

um **segmento** é uma sequência de segmentos mínimos com algum significado musical interessante;

uma **nota** é uma unidade de som produzida por um instrumento ou pessoa cantando;

um **acorde** são várias notas soando ao mesmo tempo;

as **probabilidades de emissão** em um modelo de Markov oculto para análise harmônica funcional são as probabilidades de que cada nota apareça enquanto a música está em um certo acorde;

as **probabilidades de transição** em um modelo de Markov oculto para análise funcional são as probabilidades de que um certo acorde soe em um instante, dado o acorde soando no instante anterior;

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Objetivo . . . . .	14
1.2	Metodologia . . . . .	15
1.3	Trabalhos relacionados . . . . .	15
<b>2</b>	<b>Harmonia Tonal</b>	<b>17</b>
2.1	Notas . . . . .	17
2.2	Acorde . . . . .	20
2.3	Tonalidade . . . . .	21
2.4	Análise harmônica funcional . . . . .	23
<b>3</b>	<b>Modelos de Markov ocultos</b>	<b>25</b>
3.1	O modelo <i>per se</i> . . . . .	25
3.2	Estimação de parâmetros . . . . .	28
3.3	Limitações . . . . .	30
<b>4</b>	<b>Outras técnicas analisadas</b>	<b>32</b>
4.1	Algoritmo de Pardo e Birmingham . . . . .	32
4.2	Algoritmo de Tsui . . . . .	33
4.3	Algoritmo de k vizinhos mais próximos . . . . .	34
4.4	Especialistas de harmonia . . . . .	35
<b>5</b>	<b>Rameau</b>	<b>37</b>

5.1	Softwares de apoio . . . . .	37
5.2	A Interface . . . . .	39
5.3	Estrutura interna . . . . .	40
5.4	Planos futuros . . . . .	41
<b>6</b>	<b>Resultados</b>	<b>43</b>
<b>7</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>46</b>
	<b>Referências Bibliográficas</b>	<b>48</b>
	<b>Apêndice A – Probabilidade básica</b>	<b>51</b>
	<b>Apêndice B – O canal <math>n</math>-ário simétrico</b>	<b>53</b>
B.1	Decodificação no canal $n$ -ário simétrico . . . . .	54
B.2	Inferência de $p$ . . . . .	56
B.3	Intervalos de confiança condicionais a outros canais . . . . .	57

# 1 INTRODUÇÃO

Estudar fenômenos culturais com o auxílio do computador é uma tarefa interessante, que sugere novas maneiras de se olhar para esses fenômenos e ajuda a separar, às vezes de forma não-usual, o essencial do accidental. Música tonal como a conhecemos hoje é um fenômeno particularmente interessante de ser estudado por vários motivos. Enquanto, por um lado, é algo fortemente cultural, e conceitos foram a princípio criticados, depois usados cuidadosamente, até finalmente seu uso ser praticamente livre de restrições<sup>1</sup>; por outro lado vários praticantes abordam o fenômeno como algo praticamente formal, derivável a partir de regras simples (LERDAHL; JACKENDOFF, 1983). Em ambos esses pontos de vista, no entanto, há a idéia de harmonia tonal como algo distinto, estudável e interessante. Nesse trabalho, estudamos modelos para analisar automaticamente a harmonia de músicas tonais.

Esse atividade, estudar harmonia de músicas tonais, é uma das várias possíveis em computação musical. De acordo com (MOORE, 1998), computação musical pode a grosso modo ser dividida em duas grandes sub-áreas: análise e síntese. A área de síntese sempre recebeu muita atenção; alguns dos primeiros programas de computadores foram da família MUSIC-N, iniciada em 1957 (TAUBE, 2004). A área de análise vem recebendo crescente atenção nos tempos recentes, com o surgimento da área de Music Information Retrieval (MIR) (BYRD; CRAWFORD, 2002). Esse trabalho se preocupa com estudar e analisar soluções para problemas específicos na área de análise musical automática.

A identificação da tonalidade de uma peça musical e da função tonal de cada trecho, o foco desse trabalho, é um subproblema de MIR. Esse problema é naturalmente ambíguo, o espaço de soluções dele possui grande dimensionalidade, cada solução pode ser interpretada de diversas formas e é possível extrair de alguma solução um enorme volume de dados, tanto para caracterizar melhor o processo cognitivo de apreciar uma música quanto para caracterizar melhor as músicas em si. Uma discussão com mais detalhes do que exatamente é harmonia tonal, tonalidade e análise harmônica funcional está no capítulo 2 desse trabalho.

---

<sup>1</sup>A descrição dos usos do acorde diminuto com sétima diminuta em (SCHOENBERG, 1983) é um bom exemplo disso.

Uma técnica muito popular na área de processamento de linguagem natural para modelar situações muito parecidas com uma análise harmônica funcional, como *part-of-speech tagging*, é a utilização de modelos de Markov ocultos. Eles são usados quando existe uma sequência diretamente observável de eventos no tempo e uma outra sequência não diretamente observável tal que os elementos da primeira sequência são unicamente determinados pelos correspondentes dessa última, e o que se deseja descobrir é qual foi a sequência correta não observável dado uma sequência observável qualquer. Um exemplo disso é querer descobrir quais palavras na frase “Maria foi para a sua casa” são substantivos, preposições, verbos ou adjetivos. Nesse caso, a sequência observável é “Maria foi para a sua casa”, enquanto a sequência não-observável é “Substantivo verbo preposição artigo substantivo”. Um modelo de Markov oculto modela a sequência de não-observáveis como uma cadeia de Markov e associa a cada possível elemento dessa cadeia uma distribuição de probabilidades de que elementos na sequência observável correspondem a ele. O capítulo 3 aborda esses modelos matematicamente e com mais detalhes e exemplos.

Nesse trabalho fazemos, primeiramente, uma descrição básica do que é harmonia tonal, no capítulo 2. O capítulo 3 descreve os modelos de Markov ocultos estudados nesse trabalho, e o capítulo 4 as outras técnicas usadas para comparação. Toda a implementação faz parte de um projeto maior, Rameau, descrito em detalhes no capítulo 5. Os resultados do estudo são analisados no capítulo 6, e o significado disso, conclusões e possíveis linhas futuras de trabalho são discutidos no capítulo 7. As próximas seções discutem os objetivos desse trabalho e os trabalhos relacionados.

## 1.1 OBJETIVO

Nesse trabalho pretendemos estudar e modelar computacionalmente o processo de análise harmônica funcional de música tonal, quantificando-o o máximo possível, e comparando soluções propostas e preexistentes com critérios objetivos. Para fazer isso, estudamos como pode ser aplicada a técnica de modelos de Markov ocultos nesse problema e avaliamos sua performance comparando-a com a do algoritmo de Pardo e Birmingham (PARDO; BIRMINGHAM, 1999), o algoritmo de Tsui (TSUI, 2002) e um classificador K vizinhos mais próximos.

## 1.2 METODOLOGIA

Para avaliar a corretude e utilidade desses métodos tentamos adotar critérios rigorosos. Como algumas das técnicas utilizadas dependem de dados de treinamento, e a performance nesses dados é uma amostra viciada da performance esperada da técnica em todo o problema. Para isso, inicialmente construímos gabaritos para duas amostras independentes, uma para treinamento e outra para testes. Ainda assim, é natural otimizar e ajustar alguns parâmetros baseados nos dados de teste. Por isso os dados finais apresentados na seção 6 são medidos em uma terceira amostra independente, e, por isso, são tão representativos quanto possível da performance final.

A principal métrica de performance utilizada é um intervalo de confiança de corretude na detecção de tonalidade, calculado de forma bayesiana assumindo uma distribuição de probabilidade anterior uniforme para a taxa de acerto. Além disso, apresentamos valores estimados dessa métrica para um músico profissional especializado em análise harmônica, para prover uma base objetiva para comparação. A justificativa para esse modelo está no apêndice B.

Os dados de avaliação utilizados nesse trabalho são corais de Bach, que, por serem musicalmente simples e harmonicamente ricos, são tratados como exemplos canônicos de harmonia tonal, estudados em aulas de harmonia e usados como exemplos em outros artigos dessa área. Nesse trabalho, adotamos os corais da edição (RIEMENSCHNEIDER, 1941), sendo 1, 2, 3, 6 e 42 usados para treinamento, 42 e 54 para teste e 72, 94, 95, 130 e 149 para os dados finais apresentados no artigo.

## 1.3 TRABALHOS RELACIONADOS

A primeira referência a análise harmônica funcional automatizada é (WINOGRAD, 1968), que modela música tonal como uma linguagem livre de contexto na qual uma análise harmônica é facilmente derivada da árvore sintática. Infelizmente, era uma gramática muito complexa, e difícil de adaptar para estilos musicais diferentes. Em (ULRICH, 1977) um modelo parecido foi proposto, onde o parsing era feito com backtracking implícito, para análise de Jazz.

Uma outra abordagem relevante para análise harmônica funcional é o sistema especialista descrito em (MAXWELL, 1992). Cerca de 60 regras de inferência lógicas são usadas para restringir as possíveis análises até encontrar uma correta. Uma implementação desse algoritmo dentro do Rameau está em andamento.

Três técnicas parecidas, baseadas em modelos, são descritas em (PARDO; BIRMINGHAM,

1999; BARTHÉLEMY; BONARDI, 2001; TAUBE, 1999). Todas elas são variações sobre a idéia de representar acordes como modelos, segmentar a música escolhendo os segmentos que melhor se encaixam nos modelos e, no caso de (BARTHÉLEMY; BONARDI, 2001) e (TAUBE, 1999), detectar tonalidade com modelos de notas possíveis em cada tonalidade, também. Implementamos nesse trabalho o algoritmo descrito em (PARDO; BIRMINGHAM, 1999), exposto com maiores detalhes na seção 4.1.

Um uso possível de redes neurais para análise harmônica funcional foi explorado em (TSUI, 2002). Essa técnica também foi implementada para esse trabalho, e está descrita com mais detalhes na seção 4.2.

Por último, (RAPHAEL; STODDARD, 2003) serviu de inspiração para a idéia explorada em detalhes nesse trabalho de usar modelos de Markov ocultos para se fazer análise harmônica funcional. Um dos objetivos desse trabalho é medir objetivamente como essa abordagem se comporta quando comparada com outras preexistentes na literatura.

## 2 *HARMONIA TONAL*

Música como fenômeno sócio-cultural é algo extremamente complexo e fora do escopo desse trabalho. Mesmo assim, algumas considerações básicas sobre a história e prática de música ocidental são necessárias para entender o que é harmonia tonal, qual sua razão de ser e por que é interessante estudar isso computacionalmente. Uma boa referência para entender melhor esse capítulo é (SCHOENBERG, 1983). A harmonia considerada nesse trabalho é a produzida por compositores ocidentais do período comum, ou seja, entre aproximadamente 1600 e 1900.

Uma peça musical é composta de dois elementos entrelaçados: sua melodia e sua harmonia. Nesse trabalho, melodia se refere às relações entre notas tocadas (ou cantadas) em sequência, e harmonia às relações entre notas tocadas (ou cantadas) simultaneamente. Tanto harmonias quanto melodias são conjuntos de notas, que são os elementos básicos da música ocidental do período comum.

### 2.1 NOTAS

Musicalmente, a palavra nota tem vários significados. Acusticamente, uma nota é uma onda sonora com uma frequência, um timbre, um envelope e uma duração. As notas são normalmente distinguidas pelas suas frequências, chamadas de alturas. A altura é a característica da nota que governa a sua relação harmônica com as outras notas em uma dada peça. As notas musicais são, por várias razões fora do escopo desse trabalho, normalmente divididas em **oitavas**. No sistema de afinação temperada, usado por instrumentos como violão e piano, uma oitava é dividida em 12 notas distintas. Cada nota, por sua vez, pode ter mais de um nome. As notas em uma dada oitava estão na tabela 2.1, com seus vários nomes. Duas notas com nomes diferentes mas representadas pela mesma classe de notas são ditas **notas enarmônicas**. Notas enarmônicas, apesar de soarem idênticas em instrumentos com afinação temperada, possuem interpretações harmônicas bem diferentes.

CLASSE DE NOTA	NOTAS
0	dó si $\sharp$ ré $\flat\flat$
1	dó $\sharp$ ré $\flat$
2	ré dó $\sharp\sharp$ mi $\flat\flat$
3	ré $\sharp$ mi $\flat$
4	mi ré $\sharp\sharp$ fá $\flat$
5	fá mi $\sharp$ sol $\flat\flat$
6	fá $\sharp$ sol $\flat$
7	sol fá $\sharp\sharp$ lá $\flat\flat$
8	sol $\sharp$ lá $\flat$
9	lá sol $\sharp\sharp$ si $\flat\flat$
10	lá $\sharp$ si $\flat$
11	si lá $\sharp\sharp$ dó $\flat$

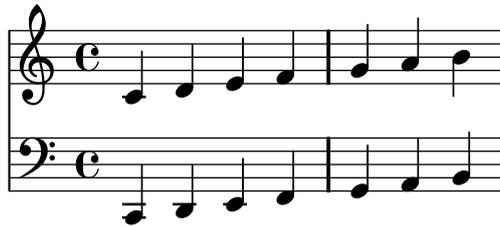
Tabela 2.1: Notação de inteiros para classes de notas.

NOME	NOTAS	CLASSE DE NOTAS
primeira justa	<i>c e c</i>	0
segunda menor	<i>c e db</i>	1
segunda maior	<i>c e d</i>	2
terça menor	<i>c e eb</i>	3
terça maior	<i>c e e</i>	4
quarta justa	<i>c e f</i>	5
quinta justa	<i>c e g</i>	7
sexta menor	<i>c e ab</i>	8
sexta maior	<i>c e a</i>	9
sétima menor	<i>c e bb</i>	10
sétima maior	<i>c e b</i>	11

Tabela 2.2: Intervalos.

Os nomes das notas são derivados da **escala maior natural**, que é a sequência de sete notas dó, ré, mi, fá, sol, lá, si. Essas notas são chamadas **naturais**. As outras notas, chamadas de **acidentes**, podem ser construídas colocando **sustenidos** (representados pelo símbolo  $\sharp$ ) ou **bemóis** (representados pelo símbolo  $\flat$ ). Um  $do\sharp$  soa mais agudo que um dó, e um  $reb$  soa mais grave que um ré, por exemplo. A tabela 2.1 também mostra as alturas dos acidentes.

Os intervalos entre as notas têm importância para a harmonia tonal igual ou maior à das notas. A tabela 2.2 mostra os intervalos mais comuns. Os intervalos são numerados de acordo com a distância entre as letras das notas na escala maior (assumindo que ela é um ciclo). O intervalo entre mi e fá é um intervalo de segunda, entre si e ré de terceira, etc. Além disso, os intervalos de segunda, terça, sexta e sétima podem ser de dois tipos, maior ou menor. O intervalo entre dó e mi se chama terça maior, já o entre mi e sol se chama terça menor. Essa distinção, quando for relevante, será mencionada. Nesse trabalho, a não ser quando especificamente mencionado, os



(a) Exemplo de notação musical



(b) Representação de durações em notação musical

intervalos se referem aos da escala maior natural contando a partir do dó. Então primeira é o intervalo entre dó e dó, segunda é entre dó e ré, terça entre dó e mi, etc. Um intervalo entre dó e ré $\sharp$  é dito uma segunda aumentada. Um intervalo entre dó e sol $\flat$  é dito uma quinta diminuta. A operação de substituir todas as notas de uma música por outras um dado intervalo acima (ou abaixo) se chama **transposição**.

As notas normalmente são representadas em uma notação musical específica. A figura 2.1a mostra um exemplo básico dessa notação. Notas em uma partitura são representadas pelas **cabeças de notas** (as “bolas” na figura), normalmente decoradas. O tempo corre horizontalmente, então notas que estão na mesma linha vertical são tocadas ao mesmo tempo. Que nota está representada por que linha é definido pelo primeiro símbolo na linha, a **clave**. Nesse trabalho usamos duas claves, e ambas estão representadas na figura. As notas mostradas são, nas duas claves, a escala maior natural, ou dó, ré, mi, fá, sol, lá e si. A sequência continua, com a próxima nota na linha de cima e a anterior na linha de baixo. Para colocar o sustenido de uma nota, coloca-se  $\sharp$  antes da cabeça de nota correspondente; para o bemol, coloca-se  $\flat$ . Após um acidente, todas as notas seguintes na mesma linha possuem o acidente implicitamente, até aparecer um  $\natural$ . As notas podem ter durações diferentes, que são indicadas pelo formato da cabeça de nota e da haste que a representam. A figura 2.1b mostra as diferentes durações, sendo que uma nota nessa figura sempre tem o dobro da duração da nota à sua direita. Notas com um “.” do lado têm sua duração multiplicada por  $3/2$ . Todas as durações são relativas a um valor escolhido na hora de reproduzir a partitura. Existem muitos outros detalhes nessa notação, como silêncios, pausas, marcas de velocidade, e articulação, mas eles não são considerados nesse trabalho.

Enquanto notas são realmente a unidade fundamental sobre a qual a música tonal é construída, a teoria de harmonia se preocupa principalmente com acordes, que são grupos de notas soando ao mesmo tempo.

ABREVIACÃO	EXEMPLO	NOME	NOTAS
M	C	maior	fundamental, terça maior, quinta justa
m	Cm	menor	fundamental, terça menor, quinta justa
°7	C°7	diminuto com sétima	fundamental, terça menor, quinta diminuta, sétima diminuta
ø	Cø	meio diminuto	fundamental, terça menor, quinta diminuta, sétima menor
+	C+	aumentado	fundamental, terça maior, quinta aumentada
Al+6	Al+6	sexta alemã	baixo, terça maior, quinta justa, sexta aumentada
It+6	It+6	sexta italiana	baixo, terça maior, sexta aumentada
Fr+6	Fr+6	sexta francesa	baixo, terça maior, quarta aumentada, sexta aumentada

Tabela 2.3: Tipos de acordes considerados nesse trabalho.

## 2.2 ACORDE

Um acorde, a princípio, pode ser qualquer conjunto de notas tocadas ao mesmo tempo, embora, na prática, seja uma dentre poucas alternativas. O acorde básico da música do common period é o acorde maior natural, composto por uma nota (chamada de fundamental), uma outra nota uma terça maior acima da fundamental e uma outra nota uma quinta justa acima da fundamental. Essas notas não precisam estar todas presentes, podem estar repetidas e não precisam aparecer nessa ordem.

Nesse trabalho consideramos apenas alguns tipos de acordes, descritos na tabela 2.3. Os acordes de sexta aumentada são diferentes dos outros e não faz sentido falar da fundamental de um acorde de sexta aumentada, então estão mostrados em relação à nota do baixo. Nesse trabalho, os acordes com maior e menor com sétima são considerados variações dos acordes maior e menor naturais.

Os acordes usados nesse trabalho estão ilustrados na figura 2.1.

Normalmente os corais de Bach são compostos de quatro vozes distintas cantando melodias diferentes. Os acordes, nesse tipo de música, surgem em alguns instantes quando as notas certas são tocadas ou sugeridas de alguma outra forma. Nem todas as notas, no entanto, fazem parte de um acorde; as que não pertencem ao conjunto de notas padrão para o acorde (se existir) que

**Tipo** C   Cm   C°7   Cø   C+   A1+6   It+6   Fr+6

Figura 2.1: Exemplos dos tipos de acordes considerados nesse trabalho.

está soando em um dado instante são chamadas de notas melódicas.

Uma peça musical harmônica, no entanto, não é apenas uma sequência de acordes que soam bem juntos. Os acordes em uma peça que segue as regras da harmonia tonal têm funções específicas, que giram em torno de um elemento implícito na música chamado de tonalidade.

## 2.3 TONALIDADE

De todos os conceitos básicos de harmonia tonal, tonalidade é o mais elusivo. Fundamentalmente, tonalidade é o conceito mais cultural de harmonia. Uma tonalidade não é uma nota específica, ou um acorde específico, mas sim um conjunto de expectativas comuns a ouvintes e compositores sobre como se organiza uma peça musical e quais os significados de cada parte dessa organização. Existem várias teorias de harmonia tonal que definem tonalidade de forma diferente; fundamentalmente, no entanto, tonalidade é algo subjetivo, é uma relação entre as expectativas do compositor, as expectativas do ouvinte e as regras tradicionais da harmonia.

Uma tonalidade  $T$  é uma tupla  $(n, m)$ , onde  $n$  é uma nota chamada de “centro tonal” e  $m$  é um modo (normalmente “maior” ou “menor”). Cada combinação possível de  $n$  e  $m$  associa significados aos acordes que soam naquele contexto. Esses significados são chamados de **funções harmônicas**. A relação entre os acordes e a tonalidade, no entanto, é complexa: por um lado os acordes são os responsáveis por estabelecer e reforçar a tonalidade; por outro, a tonalidade predetermina como cada acorde soará, quais as progressões de acordes mais agradáveis, em que contexto cada acorde soará e vários outros elementos mais sutis. É por causa da dificuldade de caracterizar essa relação que o problema de análise harmônica não foi resolvido ainda.

Um exemplo simples que expressa uma tonalidade é a figura 2.2. Alguns teóricos musi-

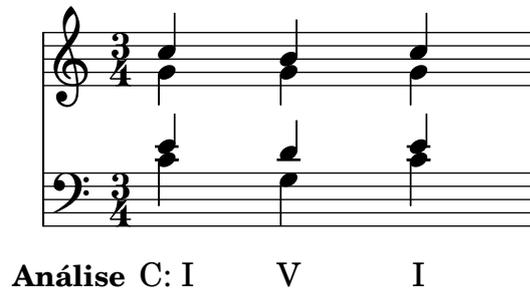


Figura 2.2: Um exemplo de tonalidade.

cais descrevem tonalidade como uma relação hierárquica entre as notas de uma composição, com notas dominando notas, notas antecipando notas, etc (SCHOENBERG, 1983). A figura representa, de forma resumida, uma dessas possíveis relações, a de tônica e dominante. Mesmo sendo muito simples, estruturas como essa estão presentes em vários dos corais de Bach, e conhecendo essa estrutura apenas um modelo de Markov oculto consegue fazer análise funcional com precisão em torno de 40%.

As funções tonais convencionais do modo maior estão descritas na tabela 2.4. Letras maiúsculas significam acordes maiores, letras minúsculas acordes menores. O nome da função é o numeral romano que representa o grau do acorde na escala que começa com o centro tonal. Por exemplo, em uma tonalidade  $T = (c, maior)$ , a função tonal ii representa um acorde de ré menor, ou Dm. A escala menor introduz outras funções harmônicas, mas seguindo a mesma forma, e a característica de cada uma delas não é diretamente relevante para esse trabalho. Os acordes de sexta aumentada (Al+6, It+6 e Fr+6) também possuem funções harmônicas específicas, cujas descrições estão fora do escopo desse trabalho. Para o modo menor outras funções adicionais existem, mas elas seguem uma estrutura parecida com essas e não precisam ser mencionadas especificamente.

Além de possuir uma tonalidade principal, uma peça musical harmônica pode, ocasionalmente, ter períodos que reforçam outro centro tonal. Se esses períodos são curtos eles são comumente chamados de tonicização; se são longos, de modulação. Nesse trabalho não fazemos essa distinção.

Determinar qual a tonalidade principal de uma música, qual é a tônica, qual é a dominante e onde estão as modulações e tonicizações é o objetivo da análise harmônica funcional, que é o problema estudado nesse trabalho.

FUNÇÃO	DESCRIÇÃO
I	é a tônica. Toda a harmonia começa e acaba na tônica.
ii	é a supertônica. A supertônica resolve na dominante.
iii	é a mediante.
IV	é a subdominante. A tônica tenta a ir para a subdominante.
V	é a dominante. A dominante precede a tônica e a fortalece.
vi	é a submediante. A submediante prepara a supertônica ou serve como menor relativa para a tônica.
viiø	é a subtônica. A subtônica ou resolve na mediante ou precede a tônica.

Tabela 2.4: Funções harmônicas da escala maior.

## 2.4 ANÁLISE HARMÔNICA FUNCIONAL

Entender as estruturas harmônicas usadas por um compositor tem vários usos possíveis. Saber quais as estruturas mais e quais as menos comuns e como cada uma se comporta é importante para poder compor músicas tanto seguindo aquele estilo quanto buscando uma cor diferente. Para improvisar sobre ou rearranjar uma peça musical a estrutura harmônica dela é um dos principais guias do que é e do que não é interessante. Um musicólogo, por sua vez, pode passar uma vida inteira estudando usos diferentes da harmonia tonal em músicas de estilos diferentes.

Fazer a análise harmônica funcional de uma música é encontrar a tonalidade, e os acordes da música, indentificando a função tonal de cada acorde. Um exemplo pode ser visto na figura 2.2.

Trabalhos anteriores decompõem o problema da análise harmônica em três partes aparentemente distintas: determinação de quais notas agrupadas pertencem a que acordes (segmentação), a identificação do nome mais apropriado para cada um desses acordes (nomeação de acordes) e a identificação da função tonal de cada acorde (análise funcional *per se*) (PARDO; BIRMINGHAM, 2000; BARTHÉLEMY; BONARDI, 2001). Um dos objetivos desse trabalho, no entanto, é desmistificar essa noção e mostrar que esses três problemas, na verdade, estão entrelaçados, e que uma boa solução só será alcançada considerando-os simultaneamente. Uma justificativa intuitiva disso é a influência de escolhas feitas na solução de qualquer um desses subproblemas na identificação da solução dos outros. Por exemplo, um acorde identificado com o nome incorreto pode sugerir que certas notas que fazem parte dele sejam colocadas pelo analisador em outros acordes adjacentes, confundindo o processo de segmentação; ou então identificar corretamente a função tonal de um acorde pode ser o melhor meio de corrigir ambiguidades na identificação do seu nome. Um resultado parecido foi encontrado em reconhecimento de

voz (JURAFSKY et al., 2000), onde modelos probabilísticos em níveis diferentes (de som para fonema, de prever quais palavras são mais frequentes em uma frase) são unidos em um modelo hierárquico, e isso aumenta consideravelmente a performance.

## **3    *MODELOS DE MARKOV OCULTOS***

Modelos de Markov ocultos são ferramentas muito utilizadas na área de processamento de linguagem natural para modelar linguagem e, com isso, identificar a classe morfológica de cada palavra e reconhecer automaticamente palavras faladas (CHRISTOPHER; SHUTZE, 1999). A teoria por trás das cadeias de Markov data do artigo (MARKOV, 1913), que utilizou cadeias de Markov para prever as letras das palavras em “Eugene Onegin”, uma novela de Pushkin famosa pelo seu estilo inovador. (SHANNON; WEAVER, 1948) famosamente utilizou cadeias de Markov para estimar a entropia a língua inglesa. Os modelos de Markov ocultos foram inicialmente estudados por (BAUM; PETRIE, 1966) como funções probabilísticas de cadeias de Markov. (VITERBI, 1967) desenvolveu o algoritmo mais famoso para fazer inferência em modelos de Markov ocultos, que é o utilizado nesse trabalho.

Como um modelo de Markov oculto é um modelo probabilístico, ele tem vários usos além do aplicado aqui. Um possível uso é a detecção de estilo musical (CHAI; VERCOE, 2001); para isso, treina-se um modelo de Markov oculto para cada estilo musical e escolhe, para cada música, o estilo cujo modelo lhe dá uma maior probabilidade. Outra aplicação musical de modelos de Markov ocultos é recomposição de músicas seguindo um dado estilo (HOFFMAN; COOK; BLEI, 2008); para isso, treina-se o modelo com exemplos de um estilo, escolhe-se amostras do modelo e resintetiza-se essas amostras seguindo o processo inverso ao do treinamento. Nesse trabalho, no entanto, estamos interessados na sequência de estados prevista pelo modelo como mais provável geradora das notas observadas. Em aprendizado de máquina esse processo é conhecido como classificação de sequência.

### **3.1    *O MODELO PER SE***

Intuitivamente, um modelo de Markov oculto usado para análise harmônica funcional diz, dado uma certa função tonal, quais as notas que normalmente representam essa função e para

---

**Algoritmo 1:** Algoritmo de Viterbi.
 

---

```

M = Matriz(n, m);
T = Matriz(n, m);
for j = 1 to m do
  | T(1, j) =  $P(C = c_j | C_a = c_0) \times P(C = c_j | N = n_0)$ ;
end
for i = 2 to n do
  | for j = 1 to m do
  | |  $T(i, j) = \max_{j'} T(i-1, j') \times P(C = c_j | C_a = c_{j'}) P(C = c_j | N = n_i)$ ;
  | |  $M(i, j) = \arg \max_{j'} T(i-1, j') \times P(C = c_j | C_a = c_{j'}) P(C = c_j | N = n_i)$ ;
  | end
end
Res = Vetor(n);
atual =  $\arg \max_i T(n, i)$ ;
for j = n downto 1 do
  | Res(j) = atual;
  | atual = M(j, atual);
end
return Res;

```

---

qual outras funções uma música normalmente progride. Essa estrutura é muito parecida com descrições feitas por músicos profissionais das características de certos acordes, como as sextas aumentadas e napolitanas descritas em (SCHOENBERG, 1983). Um dos objetivos desse trabalho é testar o quão válida essa intuição é, formalizando-a probabilisticamente, já que, segundo (COX, 1946), qualquer processo racional de decisão pode ser mapeado em probabilidades.

Matematicamente, o problema de análise harmônica em um modelo de Markov oculto consiste em encontrar a sequência de acordes com maior probabilidade de gerar as notas da música, ou

$$\arg \max_{c_1, \dots, c_n} P(C_1 = c_1, \dots, C_n = c_n | N_1 = n_1, \dots, N_n = n_n), \quad (3.1)$$

onde  $C_1, \dots, C_n$  são as variáveis aleatórias representando os acordes que poderiam ter sido observados na música,  $c_1, \dots, c_n$  valores possíveis para essa sequência de acordes,  $N_1, \dots, N_n$  as variáveis aleatórias representando as notas que podem soar numa música e  $n_1, \dots, n_n$  os valores reais das notas que soaram.

Pelo teorema de Bayes, mostrado no apêndice A,  $P(C_1 = c_1, \dots, C_n = c_n | N_1 = n_1, \dots, N_n = n_n)$  é igual a

$$\frac{P(N_1 = n_1, \dots, N_n = n_n | C_1 = c_1, \dots, C_n = c_n) P(C_1 = c_1, \dots, C_n = c_n)}{P(N_1 = n_1, \dots, N_n = n_n)}. \quad (3.2)$$

Como o termo  $P(N_1 = n_1, \dots, N_n = n_n)$  é comum a todas as análises possíveis, a equação a ser maximizada é

$$\arg \max_{c_1, \dots, c_n} P(N_1 = n_1, \dots, N_n = n_n | C_1 = c_1, \dots, C_n = c_n) \times P(C_1 = c_1, \dots, C_n = c_n). \quad (3.3)$$

Essa equação precisa de um modelo probabilístico complexo demais para ser estimado diretamente, já que o número de sequências possíveis de  $n$  acordes é exponencial em função de  $n$  (e, inclusive, é infinito considerando todos os valores possíveis de  $n$ ). Por isso, algumas suposições simplificadoras são feitas. Enquanto elas podem parecer razoáveis, elas apresentam problemas, que são discutidos na seção 3.3. A primeira suposição é que, em um certo instante de tempo, as notas ouvidas dependem unicamente do acorde que está soando naquele instante, ou seja

$$P(N_1 = n_1, \dots, N_n = n_n | C_1 = c_1, \dots, C_n = c_n) = \prod_{i=1}^n P(N = n_i | C = c_i). \quad (3.4)$$

Além disso, assume-se que as notas existentes em um certo espaço de tempo são condicionalmente independentes uma das outras; assim cada nota pode ser tratada separadamente. A equação com as notas individuais fica

$$P(N_1 = n_1, \dots, N_n = n_n | C_1 \dots C_n) = \prod_{i=1}^n \prod_{j=0}^m P(N = n_{i,j} | C = c_i), \quad (3.5)$$

onde  $n_{i,j}$  é a  $j$ -ésima nota tocada no instante de tempo  $i$ . Durante o restante desse documento,  $P(N = n_i | C = c_i)$  se refere a  $\prod_{j=0}^m P(N = n_{i,j} | C = c_i)$ .

A outra suposição importante, chamada **pressuposto de Markov**, é que a sequência de acordes após um certo acorde  $C_i$  depende unicamente de  $C_i$ , e não dos acordes anteriores. Matematicamente falando,

$$P(C_1 = c_1, \dots, C_n = c_n) = \prod_{i=1}^n P(C = c_i | C_a = c_{i-1}), \quad (3.6)$$

onde  $c_0$  representa o espaço de tempo antes da música começar, e  $P(C = c_1 | C_a = c_0)$  representa a probabilidade de uma música começar com o acorde  $c_1$ .

Assim, a equação a ser resolvida para encontrar a análise é

$$\arg \max_{c_1, \dots, c_n} \prod_{i=1}^n P(N = n_i | C = c_i) P(C = c_i | C_a = c_{i-1}). \quad (3.7)$$

Uma consequência interessante de (3.6) é que, dado que  $c_1, \dots, c_{i-1}$  é a melhor sequência de acordes até um dado acorde  $c_i$  na posição  $i$ , a análise final, se contiver  $x_i$  na posição  $i$ , terá  $c_1, \dots, c_{i-1}$  como os acordes anteriores. Ou seja, para cada instante no tempo só é necessário

guardar o histórico da melhor análise possível para cada acorde possível naquele instante. Isso diminui o espaço de busca de exponencial para  $O(nm^2)$ , onde  $n$  é o tamanho da música e  $m$  é o número de acordes possíveis. Isso é feito com o algoritmo 1, um algoritmo clássico de programação dinâmica proposto originalmente em (VITERBI, 1967) e redescoberto independentemente pelo menos sete vezes (JURAFSKY et al., 2000).

Antes de poder rodar esse algoritmo, no entanto, precisa-se de alguma estimativa consistente das probabilidades  $P(C|C_a)$  (chamadas de **probabilidades de transição**), e  $P(N|C)$  (chamadas **probabilidades de emissão**). Obter essas estimativas é um problema à parte, para o qual existem várias soluções. A solução escolhida para esse trabalho é apresentada na próxima seção.

## 3.2 ESTIMAÇÃO DE PARÂMETROS

Um dos maiores atrativos de um modelo de Markov oculto, compartilhado por todas as outras técnicas de aprendizado de máquina, é a capacidade de extrair os parâmetros do modelo diretamente de exemplos do problema em questão. Para fazer isso em um modelo de Markov oculto existem duas abordagens principais: não-supervisionada e supervisionada. Numa abordagem não-supervisionada, as probabilidades  $P(N|C)$  já estão pré-especificadas e o objetivo é estimar as probabilidades  $P(C|C_a)$  a partir dos dados, escolhendo as probabilidades que maximizam a verossimilhança dos dados obtidos. Isso é feito usando o algoritmo Baum-Welch (uma variação do tradicional Expectation-Maximization para encontrar máximos locais de probabilidades) descrito pela primeira vez em (BAUM et al., 1970). Para fazer o treinamento supervisionado estima-se separadamente as probabilidades de emissão e as probabilidades de transição. Como temos dados de treinamento à disposição e a performance de treinamento supervisionado é reconhecidamente superior, essa foi a abordagem adotada nesse trabalho (CHRISTOPHER; SHUTZE, 1999; JURAFSKY et al., 2000; RAPHAEL; STODDARD, 2003).

O problema de treinar um modelo de Markov oculto de forma supervisionada já foi amplamente estudado na literatura de processamento de linguagem natural. A abordagem mais adotada é estimar as probabilidades usando o estimador Good-Turing, desenvolvido por Alan Turing na segunda guerra mundial, em parceria com Jack Good (GOOD, 1953) e descrito com uma derivação simples, testes claros e justificação intuitiva em (GALE; SAMPSON, 1995). Nossa implementação segue as linhas desse último artigo. Essa técnica, baseada na suposição de que as frequências desejadas aproximam as esperadas de uma distribuição binomial com parâmetros desconhecidos, evita problemas comuns no estimador de máxima verossimilhança,

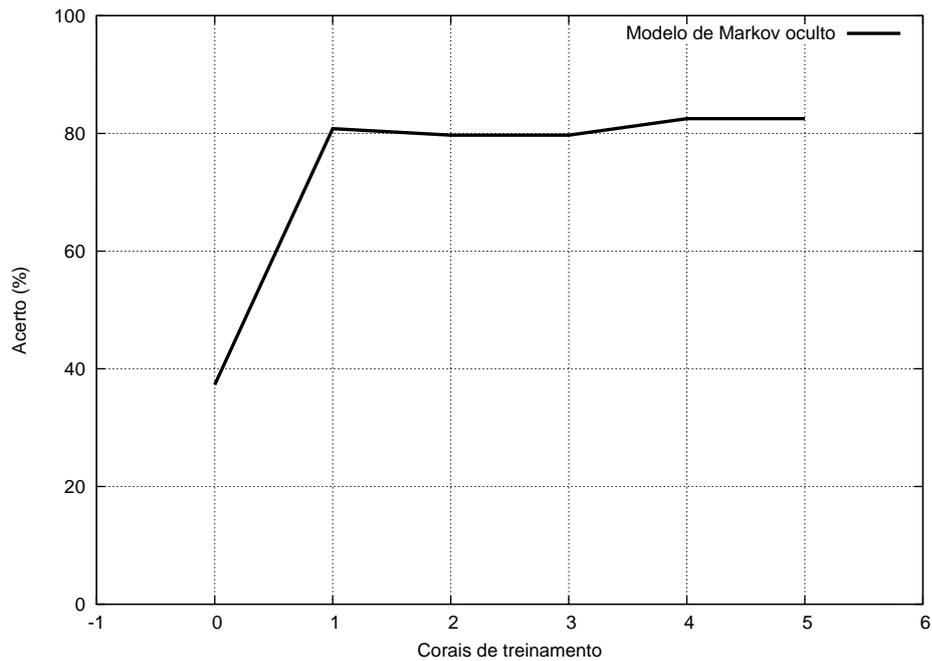


Figura 3.1: Performance por volume de dados de treinamento.

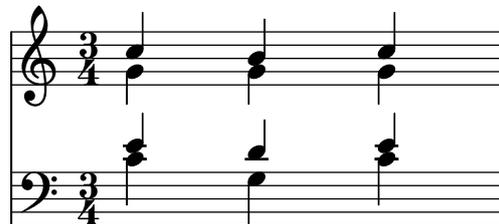


Figura 3.2: Exemplo trivial.

que não lida muito bem com classes não presentes na amostra.

A figura 3.1 mostra a performance do algoritmo analisando os corais 42 e 54 da edição Riemenschneider em função do número de corais nos quais o algoritmo foi treinado. Quando, nessa figura, o eixo horizontal mostra “0 corais” apenas o exemplo simples representado na figura 3.2, que tem como único propósito introduzir a noção mais básica de tonalidade, foi utilizado.

Um problema em treinar um modelo de Markov oculto para análise funcional é a explosão no número de parâmetros. Para esse problema, um acorde  $C_i$  é uma tupla  $(T_c, T_m, F_n, F_a, F_m)$  onde  $T_c$  representa o centro da tonalidade da música naquele instante,  $T_m$  representa o modo dessa tonalidade,  $F_n$  o número associado à função tonal daquelas notas,  $F_a$  o número de aciden-

tes (sustenidos ou bemóis) naquela função e  $F_m$  o modo da função tonal daquelas notas naquele instante. O número de possíveis notas é 96 (usamos, nesse trabalho, a codificação descrita em (KRÖGER et al., 2008)); o número de possíveis modos para uma tonalidade é 2 (maior ou menor); o número de funções tonais é 7 (uma para cada nota na escala); o número de acidentes considerados é 3 (sem acidentes, com um bemol e com um sustenido); e o número de modos para uma função tonal é 8 (maior, menor, aumentado, diminuto, meio diminuto, sexta alemã, sexta italiana e sexta francesa). Isso totaliza  $96 \times 2 \times 7 \times 3 \times 8$ , da ordem de  $10^4$  possíveis análises para cada segmento de uma música, e coloca a tabela de probabilidades de emissão com em torno de  $10^6$  entradas e a de probabilidades de transição com  $10^9$ . Esses números são consideravelmente maiores que o número de acordes em todo o nosso corpus (de treinamento e teste), e provavelmente seriam necessárias mais músicas para estimar sem problemas esses parâmetros do que existem atualmente.

Para contornar o problema da dimensionalidade, seguimos uma variação da abordagem adotada por (RAPHAEL; STODDARD, 2003) assumimos que o modelo deve ser invariante quanto a transposição. Assim, a tabela de probabilidades de emissão precisa apenas de  $(2 \times 7 \times 3 \times 8) \times 96$  (da ordem de  $10^6$ ) valores, enquanto a de transição precisa de  $(2 \times 7 \times 3 \times 8) \times (92 \times 2 \times 7 \times 3 \times 8)$  (da ordem de  $10^7$ ), o que ainda aparenta ser proibitivo. Felizmente várias combinações de modos, funções e acidentes são improváveis demais e não acontecem na música estudada nesse trabalho. Como pode ser visto na seção 3.2, um volume razoavelmente pequeno de dados de treinamento é suficiente para conseguir uma taxa de acertos razoável.

### 3.3 LIMITAÇÕES

Um modelo de Markov oculto, apesar de supor uma estrutura razoavelmente intuitiva para as músicas, falha em alguns pontos importantes. Alguns desses problemas são inerentes ao modelo de Markov oculto em si e não podem ser superados; outros, no entanto, vêm de escolhas particulares desse trabalho e podem ser melhorados em trabalhos futuros.

Os problemas inerentes ao modelo de Markov oculto se devem principalmente às independências condicionais necessárias para o modelo ser aplicado. Enquanto boa parte da característica de um acorde pode ser capturada na forma de uma distribuição de probabilidades sobre notas e uma distribuição de probabilidades sobre próximos acordes, alguns detalhes escapam. Um exemplo é que, não importa qual seja a distribuição de próximos acordes para um dado acorde, a probabilidade de permanecer naquele acorde por  $n$  instantes de tempo é geométrica. Isso não reflete a distribuição da duração de acordes na música tonal de fato. Outro problema é

que as características de uma tonalidade são muito pouco caracterizadas com um modelo perfeitamente local como um modelo de Markov oculto. Uma consequência negativa disso, por exemplo, é que uma música, após modular, tem maior probabilidade de acabar a cada instante que de retornar à tonalidade original; e isso vai contra a característica básica de um centro tonal, que é sempre retornar à tônica. Um problema menor de se usar modelos de Markov ocultos para análise harmônica é a necessidade de prefixar, em tempo de projeto, todos os tipos possíveis de acordes. Isso impossibilita que o modelo aprenda a recombinar notas já existentes e construir novos acordes a partir de blocos menores. É possível que um modelo de Markov oculto como o descrito em (HOFFMAN; COOK; BLEI, 2008) possa resolver esse problema, mas esse ainda é um problema em aberto.

Alguns problemas, no entanto, se devem a escolhas particulares desse trabalho. Nele ignoramos a duração das notas e a posição delas na música. Esse tipo de informação, no entanto, é importante para músicos que estão fazendo a análise harmônica manualmente. Outro problema sério é ignorarmos informação sobre vozes, como marcar qual nota está no baixo, qual nota é uma continuação, etc. Em música tonal, as vozes nas quais as notas estão dispostas codificam várias convenções harmônicas importantes, que poderiam ser extraídas para aumentar a performance, como sugerido em (RAPHAEL; STODDARD, 2003). O detalhe do modelo descrito nesse trabalho, no entanto, que nos parece mais prejudicial é a suposição de que as notas soando em um dado acorde são independentes. Isso não tem nenhuma validação empírica, e um modelo mais preciso dos acordes pode ajudar muito no reconhecimento. Uma desvantagem menor da nossa implementação é um custo computacional grande, devido a ter que processar todas as análises possíveis até encontrar o melhor resultado.

Enquanto as outras técnicas abordadas nesse trabalho não possuem vários desses problemas, o modelo de Markov oculto foi a melhor técnica, como pode ser visto na seção 6.

## 4 OUTRAS TÉCNICAS ANÁLISADAS

Para estudar modelos de Markov ocultos e como eles se comportam para análise harmônica é necessário comparar os resultados da análise com esse tipo de modelo com resultados obtidos por outros métodos. Para isso, além de implementarmos dois outros algoritmos de análise harmônica funcional discutidos na literatura e um proposto aqui fizemos uma pesquisa entre especialistas de harmonia para descobrir, na média, o quanto eles concordam entre si. Neste capítulo discutimos o algoritmo de Pardo e Birmingham (PARDO; BIRMINGHAM, 1999) na seção 4.1 e o algoritmo de Tsui (TSUI, 2002) na seção 4.2. Na seção 4.3 descrevemos o algoritmo de  $k$  vizinhos mais próximos e a nossa adaptação dele para fazer análise harmônica funcional. Por último discutiremos na seção 4.4 nosso experimento com especialistas de harmonia e os resultados. Uma comparação mais precisa dos acertos dos algoritmos com os acertos dos especialistas está no capítulo 6.

### 4.1 ALGORITMO DE PARDO E BIRMINGHAM

Dos algoritmos abordados nesse estudo o mais simples é o proposto por Pardo e Birmingham em (PARDO; BIRMINGHAM, 1999), originalmente proposto para ser um *baseline* para estudo de algoritmos mais complexos e corretos. Surpreendentemente, acrescentado de um simples detector de tonalidade, esse algoritmo se comporta razoavelmente bem, como será visto na seção 6.

O algoritmo consiste em definir, em tempo de projeto, uma lista de possíveis modelos para acordes e, ao analisar uma música, comparar cada segmento com todos os modelos possíveis e escolher o que mais se aproxima. A segmentação da música é feita em um passo de pós-processamento onde é feita a segmentação, que é desnecessários por esse trabalho se concentrar em corais de Bach, que são trivialmente segmentáveis (KRÖGER et al., 2008). Os modelos utilizados nesse trabalho estão especificados, por ordem de preferência (*i.e.*, se dois ou mais mode-

ACORDE	MODELO
M	tônica, terça maior, quinta
M	tônica, terça maior, quinta, sétima menor
m	tônica, terça menor, quinta
m	tônica, terça menor, quinta, sétima menor
°7	tônica, terça menor, quinta diminuta, sétima diminuta
ø	tônica, terça menor, quinta diminuta, sétima menor
M	tônica, terça maior, quinta, sétima maior
+	tônica, terça maior, quinta aumentada
A1+6	baixo, terça maior, quinta justa, sexta aumentada
It+6	baixo, terça maior, sexta aumentada
Fr+6	baixo, terça maior, quarta aumentada, sexta

Tabela 4.1: Modelos usados pelo algoritmo de Pardo.

los se encaixam igualmente bem para uma certa sonoridade, escolha o mais no topo da lista) na tabela 4.1. Esses não são os modelos originais propostos em (PARDO; BIRMINGHAM, 1999), mas são uma simples extensão proposta em (KRÖGER et al., 2008) para melhor descrever os acordes no nosso corpus.

O problema de detecção de tonalidade é completamente ignorado pelo algoritmo original. Sem querer adotar uma abordagem mais complexa, a tonalidade é determinada olhando apenas para o primeiro acorde de um coral. Isso é simplista, mas apresenta resultados razoavelmente bons, como pode ser visto na seção 6, e nos permite usar os resultados desse algoritmo como um baseline razoável. Esse algoritmo é muito inflexível, e precisa de uma certa quantidade de afinação manual explícita para ser utilizado em um estilo de música diferente do original. As regras de comparação propostas no artigo original não são justificadas, e talvez uma função distância diferente melhorasse substancialmente a performance. Ainda assim, é muito útil para análise automática rápida de músicas até então desconhecidas, e identifica razoavelmente bem alguns estruturas harmônicas simples.

## 4.2 ALGORITMO DE TSUI

Uma técnica que consegue tanto manter um baixo tempo de execução quanto uma performance razoável em problemas de reconhecimento de padrões é a rede neural. Redes neurais para análise harmônica funcional foram estudadas por (TSUI, 2002), que mostrou valores de performance na casa dos 90%. O problema analisado por ele, no entanto, parece ser um sub-problema do considerado nesse trabalho, e por isso a arquitetura das redes foi reprojeta para se encaixar melhor com os outros algoritmos e com o método de avaliação.

A principal diferença entre as abordagens é a inexistência, na nossa interpretação, de uma tonalidade secundária de um acorde. Cada acorde está associado apenas a uma tonalidade, e a idéia de tonalidade global de uma música é uma propriedade emergente das suas tonalidades locais. Outra diferença é que, em vez de separar a tarefa de análise funcional em várias redes neurais treinadas independentemente, escolhemos fazer em uma passada só todo o problema de análise funcional. Isso diminui a duração das fases de treinamento e aumenta a performance, já que mais informação está disponível em cada fase do processo. Além disso, consideramos acordes de sexta aumentada e diferenças enarmônicas entre as notas, além de considerarmos possíveis acidentes na função tonal (*i.e.*, uma função de sexto grau é diferente de uma função de sexto grau bemol).

Como foram feitas mudanças no modelo, foi necessário reestimar, com os dados de teste, qual o melhor tamanho de janela contextual para a análise. Uma janela maior possibilita ao algoritmo usar mais informações para chegar à sua decisão, e quanto maior a janela melhor a detecção de tonalidade, já que tonalidade é um fenômeno emergente de uma sequência de acordes. Por outro lado, uma janela maior aumenta o espaço para *overfitting*<sup>1</sup> e aumenta a dimensionalidade do problema, exigindo mais dados de treinamento tempo para estimar os parâmetros que uma janela menor. A janela ótima encontrada para os nossos dados de teste foi de 4 sonoridades precedentes e 4 sonoridades consequentes.

### 4.3 ALGORITMO DE K VIZINHOS MAIS PRÓXIMOS

Outro algoritmo de aprendizado de máquina interessante para fazer classificação em espaços de alta dimensionalidade é o algoritmo dos K vizinhos mais próximos (*K Nearest Neighbors* em inglês, daqui em diante chamado de KNN). Ele é um método não-paramétrico para classificação e regressão. Intuitivamente, treinamento de um KNN é apenas guardar todos os dados de treinamento em uma lista. Na hora de classificar algum exemplo, compara-se ele com todos os elementos da lista, escolhe-se os  $k$  mais próximos (para um  $k$  definido anteriormente ou decidido em fase de treinamento) e a classificação final é a mais comum dentre esses exemplos. Enquanto um KNN é muito simples de se implementar (no código usado nesse trabalho são implementadas três variações do KNN, e o custo em linhas de código por variação é o menor de todos os algoritmos implementados no Rameau), é possível torná-lo eficiente usando árvores Kd (GROTHER; CANDELA; BLUE, 1997). Além disso, no limite quando o número de

<sup>1</sup>*Overfitting* é um termo técnico usado na literatura de aprendizado de máquina para descrever as situações onde um modelo de reconhecimento de padrões começa a reconhecer padrões inexistentes e, por isso, perde a habilidade de generalizar além dos dados de treinamento.

pontos tende a infinito a taxa de erro do KNN é limitada até o dobro da mínima possível (FIX; HODGES, 1989).

Para aplicar um classificador KNN a análise harmônica algumas questões precisam ser consideradas. A mais importante delas é “qual o melhor jeito de definir a distância entre dois pontos em uma música preservando as relações harmônicas?”. Essa pergunta nós não conseguimos ainda responder, e um estudo musicológico mais completo, nas linhas de (MAZZOLA et al., 2002). No entanto, para esse trabalho, encontramos uma representação boa o suficiente. Cada segmento mínimo é representado por um vetor de frequências, onde cada elemento representa a frequência da nota correspondente naquele segmento mínimo. Para incluir informação contextual, os vetores dos segmentos mínimos adjacentes são multiplicados por  $1/\lambda$  e concatenados ao vetor cuja análise deseja-se obter. O valor de  $\lambda$ , assim como o valor de  $k$  e a quantidade ideal de segmentos adjacentes a ser incluída foram otimizados, e o que maximiza a performance nos dados de teste são quatro segmentos de contexto e  $k = 2$ . A função de distância utilizada é a distância euclidiana.

## 4.4 ESPECIALISTAS DE HARMONIA

O problema de análise harmônica é naturalmente ambíguo e as repostas possíveis são muito subjetivas. Uma mesma música, se analisada por especialistas diferentes em harmonia, pode ter interpretações muito diferentes para alguns trechos. Assim, a idéia de medir a corretude de uma técnica de análise harmônica simplesmente comparando-a com um gabarito pré-determinado não parece ser muito interessante. Por outro lado, músicos especialistas conseguem, dado uma certa música, concordar sobre a análise harmônica de longos trechos dessa música.

Nessa seção, procuramos quantificar o quanto músicos especialistas concordam entre si. Isso fornece, a grosso modo, um limite superior à performance de um algoritmo, já que concordar com um especialista mais que o esperado para esse especialista concordar com outros não diz algo significativo sobre a performance geral da técnica analisada. Embora isso não forneça um intervalo de confiança para a performance real do algoritmo<sup>2</sup>, ajuda a passar uma idéia intuitiva de quais diferenças de performance são mais significativas. O apêndice B explora um modelo capaz de quantificar o quanto os especialistas concordam entre si e o quanto concordar com um desses especialistas diz sobre a performance de uma técnica de análise harmônica. Essa técnica, no entanto, é razoavelmente complexa e necessita de um volume maior de dados para dar intervalos de confiança estreitos o suficiente.

---

<sup>2</sup>Para fazer isso é necessário ou um volume de dados muito maior ou algumas suposições restritivas sobre a natureza do problema

Para estimar o quanto os experts concordam entre si primeiro pedimos para 3 especialistas distintos analisarem o coral 19 da edição Riemenschneider. Com essas análises em mãos, contamos o número de segmentos em que cada músico discorda da maioria. Com esse número estimamos, assumindo uma distribuição de probabilidade anterior uniforme para a taxa de acerto dos especialistas, um intervalo de confiança para a sua distribuição de probabilidades de acerto real. A justificativa por trás dessa aproximação está no apêndice B. Com essa aproximação, um intervalo de 95% de confiança para a probabilidade de concordância entre os experts é  $(0.81, 0.92)$ . É com base no centro desse intervalo que os intervalos de confiança reportados no capítulo 6 foram estimados.

## 5 RAMEAU

O uso de harmonia como técnica de composição foi um marco importante da música do período comum, e teóricos musicais consideram que harmonia como a conhecemos surgiu no início do século XVII. O primeiro documento registrando as regras e convenções da harmonia tonal, no entanto, foi o *Traité de l'harmonie*, de Jean-Philippe Rameau, um teórico musical e compositor francês, publicado em 1722 (RAMEAU, 1971). Em homenagem a esse trabalho o projeto de musicologia computacional do qual essa monografia faz parte foi chamado de Rameau.

O objetivo de Rameau é eventualmente ser uma plataforma completa para estudos musicológicos. O Rameau já contém um corpus corrigido, editado, fácil de ler computacionalmente e parcialmente anotado contendo 371 corais de Bach, alguns exemplos de harmonia tonal extraídos de (KOSTKA; PAYNE, 2000), uma sonata de Mozart, um quarteto de Brahms. Além disso, Rameau já possui várias funcionalidades de musicologia computacional implementadas, como detecção de quintas e oitavas consecutivas, análise estatística da resolução de acordes com sétima, detecção de cadências e progressões, detecção de âmbito de corais e verificação estatística das regras de resolução propostas por (SCHOENBERG, 1983). Internamente Rameau é um framework para análise e desenvolvimento de algoritmos de análise harmônica e ferramentas de análise musicológica, e possui código para ler e escrever partituras, processar folhas de respostas preparadas manualmente e analisar estatisticamente os resultados do processo de análise harmônica. Atualmente, Rameau pode ser utilizado pela linha de comando ou via uma interface web. Uma interface gráfica de desktop para Rameau está em fase de projeto.

### 5.1 SOFTWARES DE APOIO

Várias tarefas importantes para uma plataforma de musicologia computacional ou são complexas demais para serem implementadas como parte desse trabalho ou já tinham sido competentemente resolvidas por outras ferramentas.

Possivelmente o software do qual o Rameau mais depende é o GNU Lilypond (NIENHUYS; NIEUWENHUIZEN, 2008). O GNU Lilypond é um software livre para tipografar partituras com qualidade profissional. O GNU Lilypond lê e escreve arquivos de texto puro em um formato razoavelmente fácil de se processar e gerar, além de ser muito prático para um músico, disponibilizando vários pequenos detalhes úteis. Todo o processo de análise harmônica ou musicologia computacional começa com o processamento de uma partitura escrita na sintaxe do GNU Lilypond para extrair as notas, e, possivelmente, termina na geração de uma ou mais partituras para visualização. Existem planos para generalizar essa parte do Rameau, e possibilitar o processamento e a geração de outros formatos de partitura, mas, por enquanto, é em torno do GNU Lilypond que o Rameau funciona.

Outro software de importância vital para o Rameau é o compilador de Common Lisp SBCL (SBCL development team, 2007). Embora o código seja genérico o suficiente para funcionar quase sem modificações em outras implementações de Common Lisp, o Rameau, quando possível, se beneficia de qualidades do SBCL, como um compilador que gera código eficiente, um ambiente de execução dinâmico, e as imagens executáveis fáceis de se gerar. Ser implementado em Common Lisp também é uma grande vantagem para o Rameau, e possibilita experimentar com facilidade mudanças arquiteturais de escopo razoavelmente grande, debugar e modificar trivialmente o Rameau em tempo de execução (criando um ambiente de desenvolvimento dinâmico e persistente) e, quando necessário, estender a linguagem para melhor atender nossos objetivos.

Fora esses dois softwares independentes, o Rameau utiliza algumas bibliotecas de software livre para realizar funções úteis. Utilizamos as bibliotecas Arnesi<sup>1</sup> e Alexandria<sup>2</sup> por fornecerem funções utilitárias genéricas de Lisp. Para fazer laços de forma genérica e eficiente usamos a biblioteca iterate (AMSTERDAM, 1990). Para persistência usamos a biblioteca cl-store<sup>3</sup> como um banco de dados orientado a objetos leve. Para expressões regulares usamos cl-ppcre<sup>4</sup>. Para processar os arquivos GNU Lilypond construímos um parser usando CL-Lexer<sup>5</sup> e CL-Yacc<sup>6</sup>. Para acessar arquivos facilmente usamos CL-FAD<sup>7</sup>. Para criar imagens dinamicamente usamos Cairo<sup>8</sup>. Para implementar os algoritmos com redes neurais usamos a biblioteca Fann (NISSEN, 2003), e para as árvores de decisão usamos código exemplo do livro (MITCHELL,

<sup>1</sup><http://common-lisp.net/project/bese/arnesi.html>

<sup>2</sup><http://common-lisp.net/project/alexandria/>

<sup>3</sup><http://common-lisp.net/project/cl-store/>

<sup>4</sup><http://www.weitz.de/cl-ppcre/>

<sup>5</sup><http://www.geocities.com/mparker762/clawk.html#lexer>

<sup>6</sup><http://www.pps.jussieu.fr/~jch/software/cl-yacc/>

<sup>7</sup><http://www.weitz.de/cl-fad/>

<sup>8</sup><http://www.cairographics.org/>

1997). Para fazer a interface web do Rameau usamos o servidor web Hunchentoot<sup>9</sup>, a biblioteca de geração de HTML CL-WHO<sup>10</sup>. Para testes usamos o framework de testes Fiveam<sup>11</sup>. Para pedaços de infraestrutura usamos biblioteca Swank, desenvolvida como parte do Slime<sup>12</sup> (que, incidentalmente, é a IDE usada para desenvolver o Rameau). Para gerar hashes md5 usamos uma biblioteca específica chamada MD5<sup>13</sup>. O Rameau também gera código  $\text{\LaTeX}$  (LAMPOR, 1994) para alguns relatórios e tabelas, e esse código foi inserido diretamente no capítulo 6.

Além dessas bibliotecas, para desenvolver o Rameau implementamos CL-Utills, uma biblioteca com utilitários comuns para Common Lisp; CL-Music, uma biblioteca para abstrair rotinas que lidam com codificações musicais diferentes, representação textual de notas e intervalos, tonalidade e operações com conjuntos musicais; e CL-Lily, uma biblioteca para processar e gerar programaticamente arquivos no formato do GNU Lilypond. Também escrevemos CL-Fann, um adaptador para a biblioteca Fann ser utilizável em Common Lisp.

## 5.2 A INTERFACE

O Rameau pode ser executado por duas formas principais: pela linha de comando e por uma interface web. A linha de comando foi projetada para ser eficiente, fornecer várias opções e possibilitar o acesso irrestrito a toda a funcionalidade do Rameau, como ferramentas de debug, de trace, etc. A interface web, por sua, foi projetada para ser o mais simples possível de utilizar por um usuário iniciante tanto em computadores quanto no Rameau. A figura 5.1 mostra um exemplo de funcionamento do Rameau Web. A interface web apenas disponibiliza a funcionalidade de análise funcional.

O Rameau, no entanto, possui muito mais comandos que os de análise funcional. Ele pode retrainar os algoritmos de aprendizado de máquina com parâmetros específicos, rodar os testes automáticos de unidade e de regressão, criar automaticamente documentação do código, fazer análise harmônica não funcional (encontrar os nomes dos acordes), gerar relatórios estatísticos sobre o código (as tabelas do capítulo 6 foram geradas automaticamente) e fazer vários testes musicológicos, como detecção de cadências, detecção das progressões segundo Schönberg, análise da resolução de sétimas, análise dos saltos intra-voz, análise de âmbito, análise de cruzamento de vozes, calcular frequência de acordes, calcular frequência de notas por tonalidade, calcular frequência de notas em tipos de acorde, detectar quintas e oitavas paralelas e gerar al-

<sup>9</sup><http://www.weitz.de/hunchentoot/>

<sup>10</sup><http://www.weitz.de/cl-who/>

<sup>11</sup><http://common-lisp.net/project/bese/FiveAM.html>

<sup>12</sup><http://common-lisp.net/project/slime/>

<sup>13</sup><http://www.pmsf.de/resources/lisp/MD5.html>

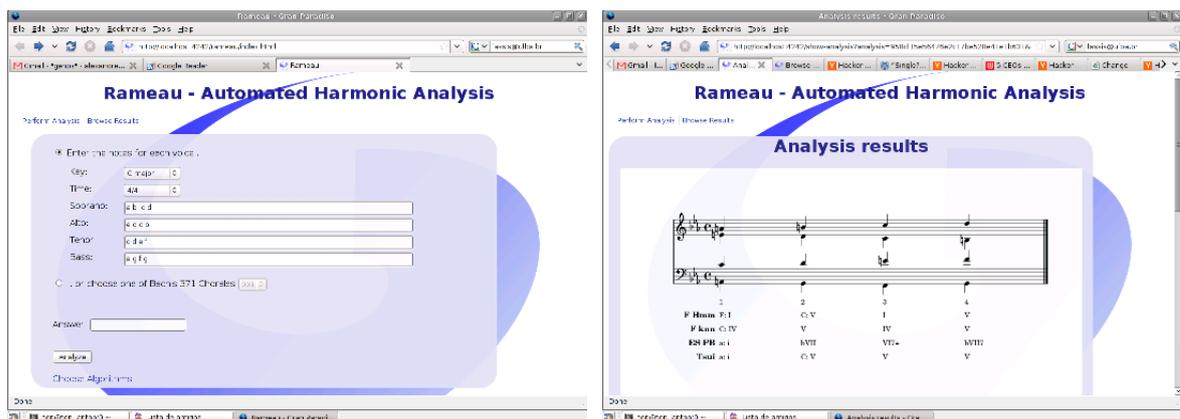


Figura 5.1: Screenshots da interface do Rameau Web.

gumas visualizações. Cada uma dessas funcionalidades é um comando específico da interface de linha de comando.

### 5.3 ESTRUTURA INTERNA

Internamente, o Rameau possui poucas partes:

- infraestrutura para definir algoritmos de análise harmônica;
- infraestrutura para processar argumentos de linha de comando só determinados em tempo de execução;
- infraestrutura para lidar com gabaritos para análise harmônica funcional e não-funcional;
- funções auxiliares para gerar lilypond específico pro corpus do Rameau; e
- infraestrutura para *queries* musicológicas automatizadas.

Todos os algoritmos de análise harmônica e os comandos são definidos externamente, como plugins. Os algoritmos simplesmente são instâncias de alguma subclasse de `rameau-algorithm` que implementam um método de análise harmônica. Para definir um algoritmo basta criar uma subclasse de `rameau-algorithm`, definir o método `functional-analysis` nessa subclasse e registrar uma instância dele com a função `add-falgorithm`. A figura 5.1 mostra o código usado para registrar o modelo de Markov oculto para análise harmônica funcional. Nela, a função `viterbi-decode` se encarrega de calcular a melhor análise usando o algoritmo 1.

Os comandos de linha de comando são estruturas ainda mais simples. Um comando é apenas uma função marcada como tal com uma chamada para a função `register-command`.

```

(defclass functional-hmm (rameau-algorithm)
  ((transitions :accessor trans :initform nil)
   (outputs :accessor out :initform nil)
   (version :accessor version :initform 0)))

(defmethod functional-analysis (segments options (alg functional-hmm))
  (viterbi-decode alg segments))

(add-falgorithm (make-instance 'functional-hmm
                              :name "F-Hmm"
                              :description "A roman numeral
                              functional analysis algorithm based on
                              Raphael and Stoddard.'"))

```

Exemplo 5.1: Código exemplo para definir um algoritmo.

```

(defun funalg (options)
  "Give the command-line options to the functional algorithms."
  (make-functional-training-data options)
  (setf (arg :algorithms options) (%load-algs options :type :func)
        (arg :options options) (process-option-list (arg :options options)))
  (iter (for alg in (arg :algorithms options))
        (format t "Options for ~a~%" (alg-name alg))
        (do-options alg options)
        (save-arg alg))
  (setf (arg :algorithms options) nil
        *training-data* nil))

(register-command :name "funalg"
                 :documentation "Give the specified roman numeral
                 functional analysis algorithms the options passed
                 with -o. Used for training."
                 :action #'funalg)

```

Exemplo 5.2: Exemplo de código para definir um comando.

Para registrar um comando basta definir opções, um nome e escrever uma documentação. A figura 5.2 mostra um exemplo de definição de comando. Para definir um comando musicológico uma infraestrutura adicional está disponível. Em vez de implementar toda a entrada e saída, basta especificar um filtro e o próprio Rameau se encarrega de toda a burocracia necessária. A figura 5.3 mostra como definir um comando simples de musicologia que calcula a frequência de notas em modos menores, relativamente à tonalidade do modo.

## 5.4 PLANOS FUTUROS

O ponto em que o Rameau mais precisa de melhorias é a ampliação do corpus, e subsequente adaptação do framework para melhor lidar com estilos de harmonia diferentes. Isso aparentemente é um problema fácil, mas harmonia é fundamentalmente um fenômeno cultu-

```

(defun count-notes-minor-classifier (context options)
  (declare (ignore options))
  (destructuring-bind ((chorale segno segment answer result &rest ignore))
    context
    (declare (ignore ignore chorale segno result))
    (let ((k (fchord-key answer)))
      (when (eql :minor (tonal-key-mode k))
        (mapcar #'print-event-note
          (first (transpose-segmentos (list segment)
                                     (- (tonal-key-center-pitch k))))))))))

(register-musicology-command :classifier #'count-notes-minor-classifier
  :name "count-minor-notes"
  :show-as :frequency
  :functional t
  :doc "Count the notes in a minor key")

```

Exemplo 5.3: Exemplo de definição de uma busca musicológica.

ral, e regras muito diferentes se aplicam a estilos diferentes. Um mesmo acorde, por exemplo, quando encontrado em uma música de Bach é ignorado ou classificado como uma nota de passagem, enquanto se ele é encontrado em uma música de Wagner ou Schönberg ele é considerado um acorde por si só e uma função tonal é atribuída a ele.

Outra parte do Rameau em desenvolvimento é a criação de uma interface gráfica para desktop que possua toda a funcionalidade da interface de linha de comando. Tal interface seria útil para usuários do Rameau no windows, que não possui um terminal nativo confortável e eficiente. Além disso, uma interface como essa pode facilitar a interação do usuário com alguns processos no Rameau (como o treinamento de algoritmos de aprendizado de máquina) que agora são basicamente automatizadas.

Finalmente, o Rameau ainda pode fornecer mais opções musicológicas interessantes, mais algoritmos de análise harmônica, mais formatos de entrada e saída, pode possuir um código mais rápido e, principalmente, pode ter mais usuários.

## 6 RESULTADOS

ALGORITMO	INFERIOR	SUPERIOR
Modelo de Markov oculto	49.5	61.7
Pardo e Birmingham	38.0	50.2
KNN	37.7	49.9
Rede neural	30.8	42.6

Tabela 6.1: Intervalo de 95% de confiança para taxa de acertos (em percentual).

LIMITE INFERIOR	LIMITE SUPERIOR
70.8	80.4

Tabela 6.2: Intervalo de 95% de confiança para a performance da rede neural nos dados de treinamento (em percentual).

Para avaliar a performance desses algoritmos treinamos os algoritmos de aprendizado de máquina utilizando os corais de Bach de número 1, 2, 3, 6 e 42 da edição Riemenschneider. Os dados obtidos nas tabelas 6.1, 6.3 e 6.4 foram calculados a partir da performance dos algoritmos nos corais de número 72, 94, 95, 130 e 149 da mesma edição. Os dados da tabela 6.2 foram obtidos com os mesmos corais de treinamento.

Os resultados obtidos para performance geral dos algoritmos estão na tabela 6.1. Esses resultados são intervalos de 95% de confiança calculados de acordo com a aproximação por uma distribuição beta descrita no apêndice B. O melhor algoritmo é o modelo de Markov oculto, seguido pelo de Pardo e Birmingham, seguido pelo KNN e por último a rede neural. Como pode ser visto na figura 6.1, o maior problema do modelo de Markov oculto é a detecção de tonalidade. Nos segmentos 3, 7 e 19, por exemplo, o modelo de Markov oculto erra apenas por antecipar ou atrasar em um segmento uma mudança de tonalidade local. No entanto, músicos discutem e admitem que não existe de fato um momento preciso em que a tonalidade muda. Assim, nossas técnicas de avaliação devem ser refinadas para ignorar esse tipo de erro.

A tabela 6.3 mostra o percentual de acerto dos algoritmos para detecção de tonalidade. Como pode ser visto, a performance é aproximadamente equivalente à da tabela 6.1, enquanto

ALGORITMO	INFERIOR	SUPERIOR
Modelo de Markov oculto	56.3	68.6
Pardo e Birmingham	49.7	62.3
KNN	43.5	56.2
Rede neural	42.9	55.6

Tabela 6.3: Intervalo de 95% de confiança para performance na detecção de tonalidade (em percentual).

ALGORITMO	INFERIOR	SUPERIOR
Pardo e Birmingham	65.2	76.7
Modelo de Markov oculto	63.3	75.0
KNN	60.6	72.5
Rede neural	39.1	51.8

Tabela 6.4: Intervalo de 95% de confiança para performance na detecção de acorde (em percentual).

a tabela 6.4, que mostra a performance apenas para identificar corretamente o acorde mostra resultados equivalentes para quase todos os algoritmos, com exceção da rede neural. Isso valida a hipótese de que a principal fonte de erro para todos os algoritmos é uma falta de precisão na detecção de tonalidade. Outra possibilidade é que tonalidades diferentes da original dentro de um coral são inerentemente ambíguas, e nossa metodologia de avaliação deveria ser modificada para lidar melhor com esse fato. Discutimos possíveis implicações disso no capítulo 7.

Ainda outra possível justificativa para esse desnível de performance é a falta de dados de treinamento específicos. Em (KRÖGER et al., 2008) exemplos específicos de estruturas harmônicas mais complexas extraídos de livros-texto foram acrescentados ao corpus de treinamento, que já consistia de mais que 5 corais. Isso não pôde ser avaliado nesse trabalho não só por falta de corais como pela dificuldade de encontrar exemplos significativos de estruturas complexas de harmonia longos o suficiente para não possuírem alguma ambiguidade tonal e assim confundir os algoritmos de aprendizado de máquina.

## 130 - Meine Seel erhebt den Herren

J. S. Bach

1 2 3 4 5 6 7 8 9 10 11 12 13

**Markov** *C: I* G: V I e: V i V **G: IV** V **I** vi ii ii V  
**KNN** *G: vi* V **C: V** e: V i **B: I** **G: IV** V V **D: ii** G: ii ii V  
**Pardo** e: i **bVII III** V7 i V VI **bVII7 III** i7 **iv7 IV7 bV**  
**Tsui** e: iii G: V e: I b: V e: iv **E: V** G: ii A: V g: i ii G: iv V V  
**Gabarito** e: i G: V I e: V i V VI G: V e: III i G: ii ii V

14 15 16 17 18 19 20 21 22 23 24

I I I V V ii e: i i V V i  
I I I V **D: I** V **C: iii** e: i V **E: V** G: vi  
**III III III bVII bVII** iv i i V V7 i  
I **C: I** G: I V V **a: i** **b: I** A: ii e: V V **E: ii**  
I I I V V e: iv i i V V i

Figura 6.1: Análise do coral 130.

## **7 CONCLUSÃO E TRABALHOS FUTUROS**

Nesse trabalho descrevemos o problema de análise harmônica funcional para música tonal do período comum. Resumidamente, fazer a análise harmônica funcional de uma peça consiste em detectar a tonalidade da peça, identificar os acordes e atribuir a cada acorde uma função tonal específica. Esse é um problema razoavelmente ambíguo, e nesse trabalho quantificamos essa ambiguidade usando um modelo baseado na teoria da informação. Aproximadamente, a probabilidade de dois especialistas de harmonia concordarem na análise de um certo segmento de uma peça simples é entre 0.8 e 0.9.

Também exploramos os modelos de Markov ocultos como possíveis soluções para o problema de análise harmônica funcional de música tonal. O modelo aqui utilizado segue o proposto em (RAPHAEL; STODDARD, 2003). Para avaliar esse modelo, um dos objetivos desse trabalho, implementamos também o algoritmo de Pardo e Birmingham (PARDO; BIRMINGHAM, 1999), o algoritmo de Tsui (TSUI, 2002) e um classificador de K vizinhos mais próximos. Para poder avaliarmos esses algoritmos lidando com a ambiguidade do problema exploramos um modelo baseado na teoria da informação para poder estimar um intervalo de confiança para a sua performance dado uma probabilidade de erro de cada segmento de um garito. O melhor algoritmo foi, como esperado, o modelo de Markov oculto. Essa comparação objetiva de vários algoritmos distintos para análise harmônica funcional é algo inexistente na literatura. Implementar mais algoritmos e compará-los com os já existentes é uma das principais direções futuras para esse trabalho.

Neste trabalho também descrevemos a plataforma Rameau, sobre a qual as técnicas aqui discutidas foram implementadas.

Existem várias linhas possíveis de trabalho futuro para essa pesquisa. A mais simples é pesquisar métodos de detecção de tonalidade (que detectem tonicização e modulação) e como incorporá-los nos algoritmos de análise harmônica descritos em (KRÖGER et al., 2008). Outra linha possível de trabalho é refinar a metodologia de avaliação, removendo algumas suposições

de independência do modelo de erros e, por exemplo, lidando diferentemente com erros na detecção de tonalidade e erros na identificação do acorde. Uma outra possibilidade é expandir o método de avaliação para incorporar mais de um gabarito para um mesmo coral, ou então incorporar ambiguidades no próprio formato de gabarito. Como fazer isso sem colocar decisões arbitrárias demais sobre onde exatamente fica a linha entre uma resposta correta e uma resposta incorreta é uma questão em aberto.

Além disso, sempre é possível investigar e implementar novos algoritmos para análise harmônica funcional, o que é muito mais fácil agora com toda a infraestrutura do Rameau à disposição.

# REFERÊNCIAS BIBLIOGRÁFICAS

- AMSTERDAM, J. The Iterate Manual. *AIM-1236*, 1990.
- BARTHÉLEMY, J.; BONARDI, A. Figured bass and tonality recognition. In: DOWNIE, S.; BAINBRIDGE, D. (Ed.). *Proceedings of the 2th International Conference on Music Information Retrieval*. Bloomington, Indiana USA: [s.n.], 2001. p. 129–136. Disponível em: <pdf/barthelemy.pdf>.
- BAUM, L.; PETRIE, T. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, v. 37, n. 6, p. 1554–1563, 1966.
- BAUM, L. et al. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, v. 41, n. 1, p. 164–171, 1970.
- BYRD, D.; CRAWFORD, T. Problems of music information retrieval in the real world. *Information Processing & Management*, v. 38, n. 2, p. 249–272, 2002.
- CHAI, W.; VERCOE, B. Folk music classification using hidden Markov models. In: *Proceedings of International Conference on Artificial Intelligence*. [S.l.: s.n.], 2001. v. 6, n. 6.4.
- CHRISTOPHER, D. M.; SHUTZE, H. *Foundations of Statistical Natural Language Processing*. [S.l.]: Cambridge: MIT Press, 1999.
- COX, R. Probability, Frequency and Reasonable Expectation. *American Journal of Physics*, AAPT, v. 14, p. 1, 1946.
- FIX, E.; HODGES, J. L. An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951). *International Statistical Review*, v. 57, n. 3, p. 233–238, 1989.
- GALE, W.; SAMPSON, G. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, Routledge, v. 2, n. 3, p. 217–237, 1995.
- GOOD, I. J. The population frequencies of species and the estimation of population parameters. *Biometrika*, v. 40, n. 237–264, 1953.
- GROTHER, P.; CANDELA, G.; BLUE, J. Fast implementations of nearest neighbor classifiers. *Pattern Recognition*, Elsevier, v. 30, n. 3, p. 459–465, 1997.
- HOFFMAN, M.; COOK, P.; BLEI, D. Data-driven recomposition using the hierarchical dirichlet process hidden markov model. In: *Proceedings of the 2008 International Computer Music Conference*. [S.l.: s.n.], 2008. p. 695–701.

- JURAFSKY, D. et al. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. [S.l.]: MIT Press, 2000.
- KOSTKA, S. M.; PAYNE, D. *Tonal Harmony, with an introduction to twentieth-century music*. 4th. ed. Boston: McGraw-Hill, 2000.
- KRÖGER, P. et al. Rameau: A system for automatic harmonic analysis. In: *Proceedings of the 2008 International Computer Music Conference*. [S.l.: s.n.], 2008. p. 273–281.
- LAMPOR, L. *LaTeX: a document preparation system*. [S.l.]: Addison Wesley Professional, 1994.
- LERDAHL, F.; JACKENDOFF, R. *A Generative Theory of Tonal Music*. Cambridge: MIT Press, 1983.
- MACKAY, D. *Information Theory, Inference and Learning Algorithms*. [S.l.]: Cambridge University Press, 2003.
- MARKOV, A. An example of statistical investigation in the text of “Eugene Onegin” illustrating coupling of “tests” in chains. *Proceedings of the Academy of Sciences, St. Petersburg*, v. 7, p. 153–162, 1913.
- MAXWELL, H. J. An expert system for harmonizing analysis of tonal music. In: EBCIOGLU, K.; LASKE, O.; BALABAN, M. (Ed.). *Understanding Music with AI: Perspectives on Music Cognition*. [S.l.]: AAAI Press, 1992. p. 335–353.
- MAZZOLA, G. et al. *The topos of music*. [S.l.]: Birkhäuser, 2002.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.
- MOORE, R. *Elements of computer music*. [S.l.]: Pearson Education, 1998.
- NEAL, R.; SCIENCE, D. of C.; TORONTO, U. of. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. [S.l.]: Department of Computer Science, University of Toronto, 1993.
- NIENHUYS, H.-W.; NIEUWENHUIZEN, J. *LilyPond*. Jan 2008. Available at [www.lilypond.org](http://www.lilypond.org).
- NISSEN, S. Implementation of a Fast Artificial Neural Network Library (fann). *Report, Department of Computer Science University of Copenhagen (DIKU)*, v. 31, 2003.
- PARDO, B.; BIRMINGHAM, W. P. *Automated Partitioning of Tonal Music*. [S.l.], 1999.
- PARDO, B.; BIRMINGHAM, W. P. Automated partitioning of tonal music. In: *Proceedings of the 13th International FLAIRS Conference*. [S.l.: s.n.], 2000.
- RAMEAU, J.-P. *Treatise on Harmony*. [S.l.]: Dover, 1971. Traduzido por Philip Gossett.
- RAPHAEL, C.; STODDARD, J. Harmonic analysis with probabilistic graphical models. In: HOOS, H. H.; BAINBRIDGE, D. (Ed.). *Proceedings of the 4th International Conference on Music Information Retrieval*. Baltimore, Maryland, USA: The Johns Hopkins University, 2003. ISBN 2-9746194-0-1. Disponível em: <papers/Raphael.PDF>.

- RIEMENSCHNEIDER, A. (Ed.). *371 Harmonized Chorales and 69 Chorale Melodies with figured bass*. New York: G. Schirmer, 1941.
- SBCL development team. *SBCL 1.0.13.45 User Manual*. 2007. Available at [www.sbcl.org/manual/](http://www.sbcl.org/manual/).
- SCHOENBERG, A. *Theory of Harmony*. [S.l.]: University of California Press, 1983.
- SHANNON, C.; WEAVER, W. A mathematical theory of communications. *Bell System Technical Journal*, v. 27, n. 2, p. 632–656, 1948.
- TAUBE, H. Automatic tonal analysis: Toward the implementation of a music theory workbench. *Computer Music Journal*, MIT Press, Cambridge, MA, USA, v. 23, n. 4, p. 18–32, 1999. ISSN 0148-9267.
- TAUBE, H. *Notes from the Metalevel: An Introduction to Computer Composition*. [S.l.]: Routledge, 2004.
- TSUI, W. S. V. *Harmonic Analysis Using Neural Networks*. Dissertação (Mestrado) — University of Toronto, 2002.
- ULRICH, J. W. The analysis and synthesis of jazz by computer. In: *Proc. of the 5th IJCAI*. Cambridge, MA: [s.n.], 1977. p. 865–872.
- VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, v. 13, n. 2, p. 260–269, 1967.
- WINOGRAD, T. Linguistics and the computer analysis of tonal harmony. *The Journal of Music Theory*, v. 12, p. 2–49, 1968.

## APÊNDICE A – PROBABILIDADE BÁSICA

Nesse trabalho analisamos um modelo probabilístico para análise harmônica e um modelo probabilístico para avaliar os resultados da análise com um gabarito possivelmente incorreto. Para ajudar a entender esses modelos, este apêndice apresenta uma breve, sucinta e incompleta introdução aos conceitos básicos de teoria da probabilidade. Uma introdução mais completa, seguindo uma abordagem prática e bem contextualizada pode ser encontrada em (MACKAY, 2003).

A teoria da probabilidade estuda **variáveis aleatórias**, que são eventos que podem assumir valores de um dado conjunto de possibilidades. Se esse conjunto é enumerável a variável aleatória é **discreta**, senão ela é **contínua**. Uma variável aleatória  $X$  que representa o resultado de um jogo de “cara ou coroa”, por exemplo, pode assumir o valor “cara” ou “coroa”. Uma variável aleatória discreta, no entanto, não assume esses valores indiscriminadamente (senão não haveria como afirmar nada sobre os resultados de experimentos), mas, após um número suficientemente grande de repetições, a frequência com a qual uma variável aleatória  $X$  assume cada valor  $x_i$  no seu domínio é representada por uma função de distribuição de probabilidade  $P(X = x_i)$ . A soma, para todo  $x_i$ , de  $P(X = x_i)$  é sempre igual a 1. Para variáveis aleatórias contínuas, a probabilidade dela assumir qualquer valor  $x_i$  é 0, mas ela pode ter uma probabilidade não-nula de estar em intervalos  $P(x_j < X < x_i)$ . Isso é representado por uma **função de densidade probabilística**  $f(x)$  tal que  $\int_{-\infty}^{+\infty} f(x)dx = 1$ . Assim,  $P(x_j < X < x_i) = \int_{x_j}^{x_i} f(x)dx$ .

As variáveis aleatórias obedecem alguns axiomas. Fora os já mencionados de que a probabilidade de uma variável aleatória assumir qualquer valor é maior que zero e a soma das probabilidades dela assumir cada valor não é maior que um; temos a regra da soma, que diz que, para  $x_i$  e  $x_j$  disjuntos,

$$P(X = x_i \vee X = x_j) = P(X = x_i) + P(X = x_j);$$

e a regra do produto, que diz que, para duas variáveis aleatórias independentes,

$$P(X = x_i, Y = y_i) = P(X = x_i)P(Y = y_i).$$

Para se falar da probabilidade de que  $X$  assumo um dado valor  $x_i$  dado que  $Y$  assume um valor  $y_i$ , falamos de  $P(X = x_i|Y = y_i)$ . Esse valor é definido como

$$P(X = x_i|Y = y_i) = \frac{P(X = x_i, Y = y_i)}{P(Y = y_i)} = \frac{P(Y = y_i, X = x_i)}{P(X = x_i)}.$$

Expandindo isso, temos o teorema de Bayes.

**Teorema A.1** (Teorema de Bayes). *Sejam  $X, Y$  duas variáveis aleatórias discretas,*

$$P(X = x_i|Y = y_i) = \frac{P(Y = y_i|X = x_i)P(X = x_i)}{P(Y = y_i)}.$$

*Ou, sejam  $X$  e  $Y$  duas variáveis aleatórias contínuas, seja  $f_X(r)$  a função de densidade de probabilidade de  $P(X)$ ,  $f_Y(r)$  a de  $P(Y)$  e  $f_{X,Y}(r)$  a de  $P(X, Y)$ ,*

$$f_X(x|Y = y) = \frac{f_Y(y|X = x)f_X(x)}{f_Y(y)}.$$

Esse trabalho assume uma interpretação Bayesiana da probabilidade, e tratamos parâmetros de distribuições de probabilidades como variáveis aleatórias, e fazemos inferência dos possíveis valores desses parâmetros usando o teorema A.1.

## **APÊNDICE B – O CANAL $N$ -ÁRIO SIMÉTRICO**

Para ter uma idéia do que esperar das taxas de sucesso dos algoritmos, que valores considerar como bons e que valores como ruins a partir de poucos gabaritos comparamos a performance desses algoritmos com a performance esperada de músicos especialistas em harmonia tonal analisando os mesmos trechos. Para usar efetivamente essa informação, no entanto, precisamos supor um mínimo de estrutura para o problema. Nessa seção descrevemos um modelo para o problema de análise harmônica baseado na teoria da informação, e como esse modelo pode ser usado para estimar um índice máximo significativo de corretude de uma análise harmônica feita automaticamente.

Inuitivamente, modelamos o processo de análise harmônica como um processo de comunicação entre o compositor e ou um especialista ou um algoritmo de análise harmônica. A mensagem enviada pelo compositor é uma sequência de acordes. Cada acorde é um dentre vários símbolos possíveis. O canal de comunicação, no entanto, não é perfeito, e pode corromper a informação transmitida. Além disso supomos que cada especialista em harmonia, no entanto, interpreta a informação transmitida de forma diferente. Assim, queremos estimar qual a probabilidade que um acorde transmitido do compositor pros músicos seja trocado por outro. Matematicamente, o processo de análise harmônica é modelado como um canal  $n$ -ário simétrico (onde  $n$  é o número de acordes) com uma probabilidade  $p$  de erro fixa e independente dos acordes. As análises dadas por cada músico especialista são então tratadas como um código de repetição, e se deseja, primeiro, uma estimativa do valor de  $p$ . Para esse método fazer sentido,  $p$  precisa ser maior que  $n^{-1}$ , o que é razoável de se supor para um especialista em harmonia. Depois, com esse valor de  $p$ , podemos estimar, para cada algoritmo de análise harmônica automática, dado um conjunto razoável de gabaritos, qual a probabilidade  $p'$  de acerto desse algoritmo. O apêndice B define precisamente o canal  $n$ -ário simétrico e prova alguns resultados importantes sobre ele.

Para estimar a probabilidade  $p$  de acerto dos especialistas em harmonia fazemos algumas suposições adicionais. Primeiro, a de que os erros de um especialista são independentes dos

erros de outro especialista (*i.e.*, que não existem, por exemplo, partes absolutamente mais complicadas em uma música). Segundo, a de que os erros de um especialista para estimar um acorde  $C_i$  são independentes dos erros de estimar um acorde  $C_j$ , para  $i \neq j$ . Além disso, assumimos que, assumindo que um especialista errou ao estimar o acorde  $C_j$ , ele escolhe com probabilidade uniforme uma das outras  $n - 1$  possibilidades para aquele segmento (isso equivale a assumir que não existe ambiguidade natural no problema, ou que, se existe, ela é estatisticamente desprezível).

Nesse apêndice descrevemos o modelo matemático do canal  $n$ -ário simétrico, usado para estimar os dados de experts usados na seção 4.4 e no capítulo 6. Nossa exposição segue a do canal binário simétrico em (MACKAY, 2003).

**Definição (Canal  $n$ -ário simétrico):** Um canal  $n$ -ário simétrico com probabilidade de erro  $p$  é um canal de comunicação usado para transmitir mensagens com um alfabeto  $\Sigma = \{s_1, \dots, s_n\}$  tal que

$$P(\text{obtido} = s_i | \text{enviado} = s_j) = \begin{cases} 1 - p & \text{se } i = j \\ \frac{p}{n-1} & \text{se } i \neq j. \end{cases} \quad (\text{B.1})$$

No uso do canal  $n$ -ário simétrico nesse trabalho, a mensagem a ser comunicada é a análise harmônica uma música e  $\Sigma$  é o conjunto de possíveis análises para um dado segmento.

**Definição (Código de repetição):** Para enviar uma mensagem  $M = \{m_1, \dots, m_n\}$  em um canal  $C$  usando um código de repetição de tamanho  $j$  simplesmente envia-se cada  $m_i$   $j$  vezes.

Por exemplo, a mensagem  $M = \{A, B, A, B, E, L, A, D, A\}$  codificada com um código de repetição de tamanho 3 fica

$$M = \{A, A, A, B, B, B, A, A, A, B, B, B, E, E, E, L, L, L, A, A, A, D, D, D, A, A, A\}.$$

## B.1 DECODIFICAÇÃO NO CANAL $N$ -ÁRIO SIMÉTRICO

Decodificar um código de repetição é um problema de inferência bayesiana. O objetivo é determinar, para cada símbolo  $s_i$ , dado as  $m$  observações  $o_1, \dots, o_m$  do valor enviado pelo canal, qual a probabilidade de  $s_i$  ter sido o símbolo enviado. Como as observações são independentes

e a priori todos os  $s_i$  e os  $o_j$  são igualmente prováveis<sup>1</sup>, pelo teorema de Bayes,

$$P(S = s_i | O = o_1, \dots, o_m) = P(S = s_i) \prod_{j=1}^m \frac{P(O = o_j | S = s_i)}{P(O = o_j)} \quad (\text{B.2})$$

$$= \frac{1}{n} \prod_{j=1}^m \frac{P(O = o_j | S = s_i)}{P(O = o_j)} \quad (\text{B.3})$$

$$= \frac{1}{n} \prod_{j=1}^m \frac{P(O = o_j | S = s_i)}{\frac{1}{n}} \quad (\text{B.4})$$

$$= \frac{1}{n} \prod_{j=1}^m n P(O = o_j | S = s_i) \quad (\text{B.5})$$

$$= n^{m-1} \prod_{j=1}^m P(O = o_j | S = s_i), \quad (\text{B.6})$$

onde

$$P(O = o_j | S = s_i) = \begin{cases} 1 - p & \text{se } s_i = o_j \\ \frac{p}{n-1} & \text{caso contrário} \end{cases} \quad (\text{B.7})$$

**Lema B.1** (Decodificação do código de repetição). *Se, com um código de repetição de tamanho  $m$  sobre um canal  $n$ -ário simétrico de alfabeto  $\Sigma = \{s_1, \dots, s_m\}$  e probabilidade de erro  $p < \frac{1}{2}$ , um dado  $s_i$  é o elemento mais frequente dentre as repetições  $o_1, \dots, o_m$  de um certo símbolo desconhecido,  $s_i$  é o símbolo que tem maior probabilidade de ser o correto.*

**Prova:** Seja  $s_j$  diferente de  $s_i$ . Além disso, podemos assumir, sem perda de generalidade, que  $s_i$  apareceu  $k$  vezes na sequência de observações, e  $s_j$   $l$  vezes, com  $k > l$ . Assim

$$\frac{P(S = s_i | O_1 = o_1, \dots, O_m = o_m)}{P(S = s_j | O_1 = o_1, \dots, O_m = o_m)} = \frac{n^{m-1} \prod_{j=1}^m P(O_j = o_j | S = s_i)}{n^{m-1} \prod_{j=1}^m P(O_j = o_j | S = s_j)} \quad (\text{B.8})$$

$$= \frac{\prod_{j=1}^m P(O_j = o_j | S = s_i)}{\prod_{j=1}^m P(O_j = o_j | S = s_j)} \quad (\text{B.9})$$

$$= \frac{\left(\frac{p}{n-1}\right)^{m-k} (1-p)^k}{\left(\frac{p}{n-1}\right)^{m-l} (1-p)^l} \quad (\text{B.10})$$

$$= \left(\frac{p}{n-1}\right)^{m-k-m+l} (1-p)^{k-l} \quad (\text{B.11})$$

$$= \left(\frac{p}{n-1}\right)^{l-k} (1-p)^{k-l} \quad (\text{B.12})$$

$$(\text{B.13})$$

Para o lema ser verdadeiro, basta que  $\left(\frac{p}{n-1}\right)^{l-k} (1-p)^{k-l} > 1$ , ou seja, basta que  $(1-p)^{k-l} > \left(\frac{p}{n-1}\right)^{k-l}$ , ou  $1-p > \frac{p}{n-1}$ , ou  $(n-1) - (n-1)p > p$ , ou  $p < \frac{n-1}{n}$ , o que é verdade por hipótese, já que  $p < \frac{1}{2}$  e  $n \geq 2$ .  $\square$

<sup>1</sup> Isso é equivalente a afirmar que  $P(S = s_i) = P(O = o_i) = \frac{1}{n}$

Pelo lema B.1, então, a melhor decodificação de um símbolo transmitido por um canal  $n$ -ário simétrico é o símbolo que mais aparece dentre as observações  $o_1, \dots, o_m$ . Isso é equivalente a dizer que escolher o  $s_i$  mais popular dentre  $o_1, \dots, o_m$  é o estimador de máxima verossimilhança para o valor real do símbolo emitido  $e_i$ .

O objetivo de usarmos esse modelo, no entanto, não é ter uma justificativa teórica para estimar qual a análise harmônica real mais provável dado algumas análises harmônicas possivelmente falhas. Para resolver o problema real precisamos estimar qual a taxa de erro  $p$  de um canal  $C$  dado uma sequência de observações de valores  $e_i$  desconhecidos codificadas com um código de repetição. Esse problema, no entanto, também é resolvível com inferência bayesiana.

## B.2 INFERÊNCIA DE $P$

Para descobrir qual a distribuição de probabilidades da frequência de erro  $p$  de um canal  $n$ -ário simétrico  $C$  dado alguns dados  $D = d$  observados no canal quando  $C = c$  foi enviado, podemos usar o teorema de Bayes para variáveis contínuas:

$$f_p(f|D = d) = \frac{P(D = d|p = f)f_p(f)}{P(D = d)}.$$

Usando a regra da soma,

$$P(D = d|p = f) = \sum_{i=1}^n P(D = d|p = f, C = c_i)P(C = c_i),$$

e, assumindo uma distribuição de probabilidade uniforme para as classes corretas<sup>2</sup>, sendo  $c_i$  uma das possíveis classes,

$$P(D = d|p = f) = P(C = c_i) \sum_{i=1}^n P(D = d|p = f, C = c_i).$$

Com outra aplicação da regra de Bayes,

$$P(D = d|p = f, C = c) = \frac{P(C = c|D = d, p = f)P(D = d|p = f)}{P(C = c|p = f)}.$$

Substituindo, as constantes ficam

$$\frac{f_p(f)}{P(D = d)}P(C = c_i) \frac{P(D = d)}{P(C = c_i)} = f_p(f).$$

---

<sup>2</sup>Ou seja:  $P(C = c_i) = \frac{1}{n}$

Assim, a fórmula final é

$$f_p(f|D = d) = f_p(f) \sum_{i=1}^n P(C = c_i|D = d, p = f).$$

Como calcular  $P(E|D, p)$  para um exemplo está descrito na equação (B.6). Para se calcular essa função para  $m$  exemplos, dado que os erros são condicionalmente independentes,

$$f_p(f|D = d) = f_p(f) \prod_{j=1}^m \sum_{i=1}^n P(C = c_i|D = d_j, p = f). \quad (\text{B.14})$$

A equação (B.6), para o caso de uma resposta correta e uma incorreta, se reduz para uma distribuição Beta, cuja densidade é  $P(x) = Kx^\alpha(1-x)^\beta$ , e para o caso de  $n$  respostas possíveis, ela se reduz à distribuição de Dirichlet equivalente, que é aproximável por uma Beta para esse canal. Podemos, assim, se o número de concordâncias entre os experts é alto, aproximar a distribuição calculada na equação (B.14) por uma distribuição  $Beta(\alpha, \beta)$ , onde  $\alpha$  é total de símbolos que pertencem às maiorias definidas no lema B.1, e  $\beta$  o número de símbolos restantes. Essa é a aproximação usada na seção 4.4.

Possuindo essa estimativa do valor de  $p$  para um certo canal  $n$ -ário simétrico  $C$ , pode-se estimar o quão confiável é uma mensagem transmitida por esse canal sem um código de repetição. A probabilidade de cada símbolo transmitido pelo canal  $C$  estar correta é  $1 - p$ . Assim, dado uma mensagem transmitida por esse canal e outra observação da mesma mensagem transmitida por um canal desconhecido, pode-se estimar um intervalo de confiança para a probabilidade de erro daquele outro canal. Isso, nesse trabalho, é usado para, com os gabaritos com ruído feitos por um especialista para vários corais de Bach, podermos estimar um intervalo de confiança para a taxa de acertos da análise harmônica feita por cada algoritmo. Assim podemos comparar objetivamente os algoritmos e fazer testes de hipóteses.

### B.3 INTERVALOS DE CONFIANÇA CONDICIONAIS A OUTROS CANAIS

Para estimar um intervalo de confiança para a frequência de erro  $p'$  de um canal  $n$ -ário simétrico  $C'$  dado a mesma mensagem enviada por  $C'$  e por um  $C$  com frequência de erro conhecida precisamos calcular a distribuição de probabilidades de  $p'$ , ou, dado as observações  $D' = d'$  no canal  $C'$  e  $D = d$  no canal  $C$ ,

$$P(p' = f'|D' = d', D = d, p = f).$$

Para calcular isso podemos usar o teorema de Bayes (descrito no apêndice A. Assim,

$$f_{p'}(f'|D' = d', D = d, p = f) = \frac{P(D' = d'|p' = f', D = d, p = f)f_{p'}(f')}{P(= d')} \quad (\text{B.15})$$

$$= \frac{f_{p'}(f')}{P(D' = d')}P(D' = d'|p' = f', D = d, p = f). \quad (\text{B.16})$$

Para calcular  $P(D' = d'|p' = f', D = d, p = f)$  podemos decompor esse evento usando a regra da soma. Existem dois casos mutuamente exclusivos: ou  $d' = d$  ou não. Assim,

$$P(D' = d'|p' = f', D = d, p = f) = \begin{cases} f'f + (1 - f')(1 - f) & \text{caso } d = d' \\ f'(1 - f) + f(1 - f') & \text{caso contrário.} \end{cases}$$

Intuitivamente,  $f'f + (1 - f')(1 - f)$  é a chance de ambos estarem certos somada com a chance de ambos estarem errados, e  $f'(1 - f) + f(1 - f')$  é a chance de um estar certo e o outro errado. Com essa distribuição pode-se calcular, usando o método de Markov Chain Monte Carlo (NEAL; SCIENCE; TORONTO, 1993), um intervalo de confiança para a probabilidade de erro  $p'$ . Outra opção é aproximar  $f_{p'}(f'|D = d, p = f)$  por uma distribuição  $Beta((1 - f) \times \text{acertos} + f \times \text{erros}, (1 - f) \times \text{erros} + f \times \text{acertos})$  seguindo a mesma lógica que para a equação (B.14). Essa aproximação foi verificada empiricamente e é a usada nos resultados apresentados no capítulo 6.