

Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Mestrado em Engenharia Informática

Ano Letivo 2020/2021

Teoria Algorítmica da Informação

Trabalho Prático nº 3

Aveiro, 18 de janeiro de 2021

António Ramos, 101193

Luís Laranjeira, 81526

Rafael Sá, 104552

Índice

Desenvolvimento do Programa	3
1.1. Estrutura do Programa	3
1.2. Base de Dados	3
1.3. Compressores	4
1.4. Samples	4
1.4.1. Tipos de Ruído	4
Análise de Resultados	6
2.1. Análise à Duração das Samples	6
2.2. Análise aos Compressores	7
2.3. Análise ao Volume do Ruído	8
2.4. Análise ao Tipo de Ruído	8
Conclusões	10
Referências	11

1. Desenvolvimento do Programa

O objetivo do programa é efetuar a identificação de músicas através de *samples* de tamanho reduzido, utilizando o Normalized Compression Distance (NCD). O NCD é calculado com base em compressores, e vai indicar o grau de similaridade entre uma música e a *sample*, sendo que quanto menor o valor, maior o grau de similaridade.

1.1. Estrutura do Programa

A estrutura do programa para identificação de músicas consiste num compressor, que será utilizado no cálculo do NCD, e num *HashMap* que contém as frequências de todas as músicas da base de dados. Para efetuar a identificação da música, o programa calcula o NCD de cada música da base de dados com a *sample*, utilizando as respetivas frequências, e no fim devolve os k melhores resultados, sendo que k é um parâmetro do programa.

O programa permite adicionar ruído à *sample* que é indicada nos parâmetros e, para isso, é utilizado o programa SoX. Para adicionar o ruído à *sample*, é gerado primeiro um ficheiro com o volume e tipo de ruído indicado nos parâmetros e, de seguida, é efetuado um *merge* entre o ficheiro do ruído e o ficheiro da *sample*. Como o ruído é gerado de forma aleatória pelo SoX, o ficheiro resultante poderá variar, o que irá originar valores de NCD ligeiramente diferentes.

Para obter os ficheiros das frequências dos ficheiros de áudio é utilizado o programa GetMaxFreqs, fornecido previamente.

De forma a que os programas SoX e GetMaxFreqs possam ser executados tanto em Windows como em Linux, foi desenvolvido um método que permite identificar qual o sistema em que o programa está a ser executado. Este método permite fazer chamadas dinâmicas ao sistema consoante o seu tipo.

1.2. Base de Dados

A base de dados consiste num diretório, que é indicado como parâmetro do programa, que contém os ficheiros de áudio ou das frequências. Para um correto funcionamento do programa, dentro os ficheiros das frequências devem estar num diretório “freqs” dentro do diretório indicado como parâmetro.

Para adicionar novas músicas à base de dados, é necessário colocar o ficheiro de áudio no diretório da base de dados que será indicado ao programa e, ao executar o programa, deve ser colocada a *flag* “-g” nos parâmetros. Esta *flag* fará com que os ficheiros da frequência das novas músicas sejam adicionados aos ficheiros já existentes. Após ser gerado o ficheiro da frequência, o ficheiro de áudio pode ser removido do diretório. Os ficheiros de áudio devem ser .wav ou .flac, stereo, amostrados a 44100 Hz, 16 bits.

A base de dados deste projeto possui um total de 115 músicas.

1.3. Compressores

Os compressores são utilizados pelo programa para calcular o NCD, sendo que o resultado e tempo de execução varia consoante o compressor utilizado. O programa disponibiliza um total de 8 compressores, nomeadamente o gzip, bzip2, lzma, deflate, lz4, snappy, xz e Zstandard (ou zstd).

A escolha do compressor que se pretende utilizar deve ser indicada nos parâmetros do programa. Durante a execução, o programa utiliza esse compressor para comprimir a informação necessária para um ficheiro temporário que, no final, é eliminado.

1.4. *Samples*

De forma a obter as *samples* necessárias para executar os testes, foi criado um programa auxiliar que permite gerar as *samples* de todos os ficheiros de áudio de um determinado diretório, com a duração pretendida (em segundos). A geração da *sample* é feita recorrendo ao programa SoX, sendo que o excerto é retirado aproximadamente do excerto da música, uma vez que se fosse retirado no início, por exemplo, poderia existir períodos de silêncio muito longos que impossibilitaria a identificação da *sample*.

No programa principal, onde se pretende efetuar a identificação das músicas, a *sample* é passada como parâmetro, e é possível adicionar ruído de vários tipos e com várias intensidades. O volume do ruído é um valor que varia entre 0 e 1, e o ruído poderá ser do tipo *white noise*, *pink noise* e *brown noise*. O ficheiro de áudio tal como já mencionado no capítulo da base de dados deve ser .wav ou .flac, stereo, amostrado a 44100 Hz, 16 bits.

1.4.1. Tipos de Ruído

As cores associadas a cada tipo de ruído são derivadas de uma analogia entre o espectro de frequências de ondas sonoras e o espectro equivalente de frequências de ondas de luz. Por exemplo, se o padrão da onda sonora do *white noise* fosse traduzido em ondas de luz, a luz resultante seria de cor branca [1].

Os tipos de ruído disponibilizados no programa são:

- *White noise*: O ruído está amplamente distribuído por todo o espectro sonoro, incluindo sons de baixa frequência, médios e de alta frequência [2].
- *Pink noise*: É semelhante ao *white noise*, mas com as frequências mais altas reduzidas, ou seja, o ruído é mais alto no segmento de baixa frequência, e mais suave no segmento de alta frequência [2].
- *Brown noise* (ou *Brownian noise*): É um pouco mais "áspero" do que o *pink noise*, pois reduz ainda mais as frequências mais altas, ou seja, é ainda mais intenso na baixa frequência e sem os sons de alta frequência do *white* e *pink noise* [2].

As figuras 1.1, 1.2 e 1.3 mostram o espectro sonoro do *white noise*, *pink noise* e *brown noise*, respetivamente.

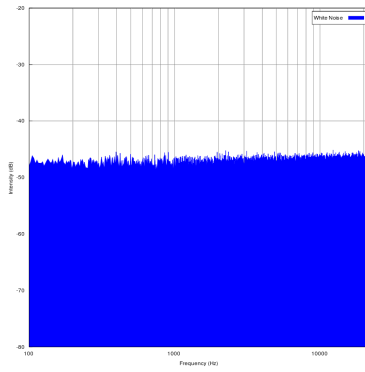


Figura 1.1. Espectro do white noise [1]

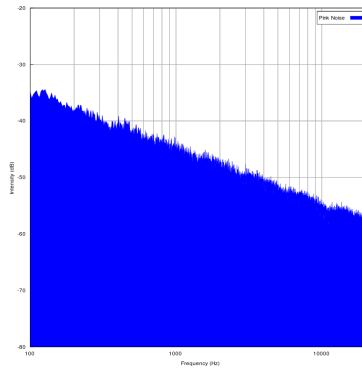


Figura 1.2. Espectro do pink noise [1]

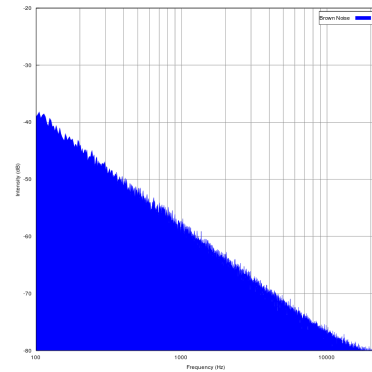


Figura 1.3. Espectro do brown noise [1]

2. Análise de Resultados

De forma a testar o programa desenvolvido, e a sua robustez, foi realizada uma sequência de testes que avalia a precisão do programa na identificação das músicas, variando a duração das *samples*, o compressor utilizado e o volume e tipo do ruído que se irá adicionar à *sample*. Os testes foram realizados com total de 115 *samples*, cada uma correspondente a uma das 115 músicas da base de dados.

Para calcular a precisão foi desenvolvido um programa auxiliar, *precision*, que executa todas as *samples* de um determinado diretório e, no fim, apresenta a precisão e o tempo de execução, em segundos. Para cada *sample* o programa compara o nome da música identificada com o nome da *sample*, sendo que o cálculo da precisão consiste no número de vezes que a identificação foi a correta em relação ao total de *samples* identificadas.

Os tempos de execução apresentados nos resultados variam consoante a máquina em que foram executados os testes, sendo que apenas será relevante na análise aos vários compressores.

2.1. Análise à Duração das *Samples*

Esta análise tem como objetivo compreender de que forma a duração das *samples* influencia a correta identificação da música e, consequentemente, a precisão do programa.

Foram realizados testes para *samples* com 5, 10 e 15 segundos, não considerando a existência de ruído. A tabela 2.1. apresenta os resultados obtidos com a variação da duração utilizando o compressor *gzip*.

Duração da <i>sample</i>	Precisão	Tempo de Execução
5 segundos	30.43%	140 segundos
10 segundos	21.74%	138 segundos
15 segundos	45.22%	143 segundos

Tabela 2.1. Resultados obtidos para o compressor *gzip* com a variação da duração das *samples*

Analisando a tabela, é possível verificar que os melhores resultados para a precisão são obtidos com *samples* de 15 segundos, enquanto que os piores resultados são obtidos para *samples* de 10 segundos. Em relação ao tempo de execução, a duração da *sample* não parece ter qualquer influência neste valor.

Os resultados relativos à precisão não eram os esperados, uma vez que se esperava que a precisão aumentasse com a duração das *samples*, e o que se verifica entre os 5 e os 10 segundos é o oposto. Partindo do pressuposto que esta estranha tendência seria devido ao compressor que se utilizou, decidiu-se efetuar os mesmos testes utilizando outro compressor, neste caso o compressor *zstd*. Os resultados obtidos com este compressor são apresentados na tabela 2.2.

Duração da <i>sample</i>	Precisão	Tempo de Execução
5 segundos	97.39%	103 segundos
10 segundos	99.13%	104 segundos
15 segundos	100.0%	106 segundos

Tabela 2.2. Resultados obtidos para o compressor *zstd* com a variação da duração das *samples*

Analisando a tabela, é possível concluir que, no caso deste compressor, quanto maior a duração da *sample*, maior será a precisão. Com isto, verifica-se que a influência da duração da *sample* varia consoante o compressor, porém, de uma forma geral, a precisão aumenta com a duração da *sample*. Em relação ao tempo de execução, não se verifica qualquer influência da duração, tal como nos resultados do teste anterior.

2.2. Análise aos Compressores

Esta análise tem como objetivo compreender de que forma é que a escolha do compressor irá influenciar a precisão e o tempo de execução do programa.

Foram realizados testes para os 8 compressores disponíveis no programa, utilizando *samples* com a duração de 10 segundos e não considerando a existência de ruído. A tabela 2.3. apresenta os resultados obtidos com a variação do compressor utilizado.

Compressor	Precisão	Tempo de Execução
gzip	21.74%	138 segundos
bzip2	100%	2037 segundos
lzma	69.56%	1677 segundos
deflate	20.87%	132 segundos
lz4	100%	20618 segundos
snappy	86.08%	221 segundos
xz	62.60%	1101 segundos
zstd	99.13%	104 segundos

Tabela 2.3. Resultados obtidos com a variação do compressor

Analisando a tabela, é possível concluir que os compressores que apresentam os piores resultados são o *gzip* e o *deflate*. Apesar de apresentarem resultados na ordem dos 20%, são também dos compressores mais rápidos. Por outro lado, os compressores que apresentam os melhores resultados são o *bzip2*, o *lz4* e o *zstd*. Destes compressores, o

bzip2 e o *lz4* são os que demoram mais tempo, chegando mesmo a demorar quase 6 horas no caso do *lz4*, e, ao contrário destes, o *zstd* é o que demora menos tempo. Posto isto, verifica-se que o compressor que apresenta uma melhor relação precisão/tempo de execução é o *zstd*, obtendo resultados surpreendentemente bons para o seu tempo de execução.

Em relação aos outros compressores analisados, tem-se o *xz* e o *lzma*, que apresentam resultados medíocres para o tempo de execução que demoram, e tem-se *snappy* que apresenta resultados decentes com um baixo tempo de execução comparativamente aos outros, tendo por isso também uma boa relação precisão/tempo de execução.

2.3. Análise ao Volume do Ruído

Esta análise tem como objetivo compreender de que forma é que o volume do ruído que será adicionado à *sample* influencia a precisão da sua identificação.

Foram realizados testes variando o volume entre 0 e 1, utilizando *samples* com a duração de 10 segundos e com o tipo de ruído *white noise*. O compressor utilizado foi o *zstd*, por ser o compressor que apresenta a melhor relação precisão/tempo de execução. A tabela 2.4. apresenta os resultados obtidos com a variação do volume do ruído.

Volume do Ruído	Precisão	Tempo de Execução
0.0	99.13%	104 segundos
0.2	99.13%	109 segundos
0.4	95.65%	114 segundos
0.6	87.82%	105 segundos
0.8	83.48%	116 segundos
1	74.78%	110 segundos

Tabela 2.4. Resultados obtidos com a variação do volume do ruído

Analisando a tabela, é possível concluir que quanto maior o volume do ruído menor será a precisão da identificação. Para o compressor *zstd*, um volume de ruído baixo não aparenta ter impacto na precisão, porém quando o volume do ruído começa a ser mais significativo, a precisão vai diminuindo. De realçar que mesmo com bastante ruído na *sample*, a precisão da identificação ainda conseguiu valores bastante satisfatórios e surpreendentes. Em relação ao tempo de execução, o volume do ruído não tem impacto na variação deste valor.

2.4. Análise ao Tipo de Ruído

Esta análise tem como objetivo compreender de que forma é que o tipo do ruído que será adicionado à *sample* influencia a precisão da sua identificação.

Foram realizados testes para os 3 tipos de ruído disponíveis no programa, utilizando *samples* com a duração de 10 segundos e com o volume de ruído de 0.5. O compressor utilizado foi o *zstd*, por ser o compressor que apresenta a melhor relação precisão/tempo de execução. A tabela 2.5. apresenta os resultados obtidos com a variação do tipo de ruído.

Tipo de Ruído	Precisão	Tempo de Execução
<i>White noise</i>	92.17%	116 segundos
<i>Pink noise</i>	60.87%	107 segundos
<i>Brown noise</i>	3.48%	106 segundos

Tabela 2.5. Resultados obtidos com a variação do tipo de ruído

Através da análise à tabela, é possível verificar que o *white noise* é o tipo de ruído que tem menos impacto na precisão da identificação, ao contrário do *brown noise*, que apresenta um impacto significativo no valor da precisão, tornando muito difícil a correta identificação da música. Em relação ao tempo de execução, o tipo de ruído não tem impacto na variação deste valor.

De forma a compreender os resultados obtidos em relação à precisão, é necessário analisar mais detalhadamente as características dos vários tipos de ruído. Tal como é referido no ponto 1.4.1., o *white noise* é um tipo de ruído que está espalhado por todo o espectro sonoro, desde *low-frequencies* a *high-frequencies*, ao contrário do *pink noise*, que reduz as *high-frequencies*, e do *brown noise* que reduz as *high-frequencies* ainda mais. Como a assinatura da *sample* é obtida através das frequências mais significativas, a redução das *high-frequencies* que é feita no *pink* e *brown noise* vai fazer com que a assinatura seja alterada de forma significativa, resultando em *samples* mais difíceis de se identificar.

3. Conclusões

Com base nos resultados obtidos, conclui-se que é possível efetuar a identificação de músicas através de *samples* utilizando o NCD, porém, existem alguns fatores que contribuem significativamente para a precisão da identificação, como o ruído, a duração da *sample* ou o compressor utilizado.

O compressor assume um papel fundamental para que o programa possua uma boa precisão, visto que é com base nele que o NCD é calculado. Os testes executados aos compressores disponibilizados apresentam uma grande variedade de resultados, desde compressores rápidos com baixa precisão, até compressores muito demorados, mas que oferecem um elevado grau de precisão. Posto isto, deve ser escolhido um compressor que apresente uma boa relação precisão/tempo de execução, de forma a que se possa obter resultados com o grau de precisão pretendido em tempo útil.

Em relação aos outros fatores, a duração da *sample* apresenta influência na precisão da identificação, sendo que, geralmente, quanto maior a duração da *sample*, maior a precisão. Já o ruído também tem impacto na precisão, sendo que o tipo de ruído aparenta ter uma maior influência que o volume do mesmo.

Referências

[1] Wikipedia. (2021). *Colors of noise*. https://en.wikipedia.org/wiki/Colors_of_noise

[2] Sound Of Sleep. (2021). *White, Pink, and Brown Noise: What's the difference?*
<https://www.soundofsleep.com/white-pink-brown-noise-whats-difference/>