



UNIVERSIDADE DE AVEIRO

Departamento de Eletrónica, Telecomunicações e Informática

Mestrado em Engenharia Informática

Teoria Algorítmica da Informação

Trabalho Prático nº3

Rafael da Fonseca Fernandes | 95319

Gonçalo Junqueira | 95314

Aveiro | 31 de janeiro de 2022

Índice Geral

1. Desenvolvimento do Programa.....	3
1.1. Estrutura do Programa.....	3
1.2. Base de Dados	3
1.3. Compressores	4
1.4. Samples	4
1.5. Tipos de Ruído	4
2. Análise de Resultados.....	7
2.1. Análise à Duração das Samples	7
2.2. Análise aos Compressores.....	8
2.3. Análise ao Volume do Ruído	8
2.4. Análise ao Tipo de Ruído.....	9
3. Conclusões.....	11
4. Referências.....	12

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*

O programa está dividido em três subprogramas com funcionalidades diferentes, sendo eles, o SetUp, Compressor e, por fim, o project. O project é o “main” do programa, o SetUp tem como funcionalidade gerar os ficheiros de frequência, concatenar os ficheiros e guardar os resultados em pastas específicas e, por fim, o Compressor é responsável por comprimir ficheiros consoante a opção escolhida pelo utilizador e, a partir disso verificar o tamanho do ficheiro para utilizar no futuro.

Resumidamente, inicialmente lê os parâmetros de entrada, de seguida adiciona ruído ao ficheiro target e, assim é criada a frequência. A partir disso é comprimido os targets e os samples e são guardados em diferentes ficheiros (zippedSamples / zippedTarget). O próximo passo é concatenar a frequência target com as frequências dos samples e comprimir o resultado. Por fim, com todos os ficheiros necessários criados é possível calcular e verificar qual das músicas disponíveis é a mais provável de ser a correta e para isso, o programa começa por ler o tamanho dos vários ficheiros e coloca os valores respetivos na fórmula. O resultado desse cálculo é sucessivamente guardado num dicionário (a chave é o nome da música e o valor é o resultado). No fim é visualizado o dicionário ordenadamente.

O programa permite também adicionar ruído à sample que é indicada nos parâmetros e, para isso, é utilizado o programa SoX. Para adicionar ruído à sample, é gerado primeiro um ficheiro com 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 aleatório 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 uma verificação para permitir identificar qual o sistema em que o programa está a ser executado.

1.2. *Base de Dados*

A base de dados consiste num diretório, que está predefinido no programa, que contém os ficheiros de áudio ou das frequências. Para um correto funcionamento do programa, os ficheiros wav devem estar no diretório “samples”. Para adicionar novas músicas à base de dados, é necessário colocar o ficheiro de áudio no diretório da base de dados que está predefinida pelo programa.

Por fim, a base dados deste projeto possui um total de 105 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 5 compressores, nomeadamente o gzip, 7Zip, zlib, bzip2 e lzma.

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 utilizar no futuro. É importante realçar que cada vez que o programa é inicializado remove os ficheiros criados anteriormente.

1.4. *Samples*

De forma a obter as samples necessárias para executar os testes, foi criado um programa auxiliar que permite gerar 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 no meio da música, uma vez que caso fosse retirado no início, por exemplo, poderia existir períodos de silêncio muito longos que impossibilitaria a identificação da sample.

No programa inicial, 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 deve ser em formato .wav.

1.5. *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 [12].

Neste projeto foram utilizados três tipos de ruídos, sendo eles:

- White noise: O ruído está amplamente distribuído por todo o espectro sonoro, incluindo sons de baixa frequência, médios e de alta frequência [12].
- Pink noise: É semelhante ao White noise, mas com as frequências mais altas reduzidas, isto é, o ruído é mais alto no segmento de baixa frequência, e mais suave no segmento de alta frequência [12].
- Brown 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 [12].

As figuras 1, 2 e 3 mostram o espectro sonoro do White noise, pink noise e Brown noise, respetivamente.

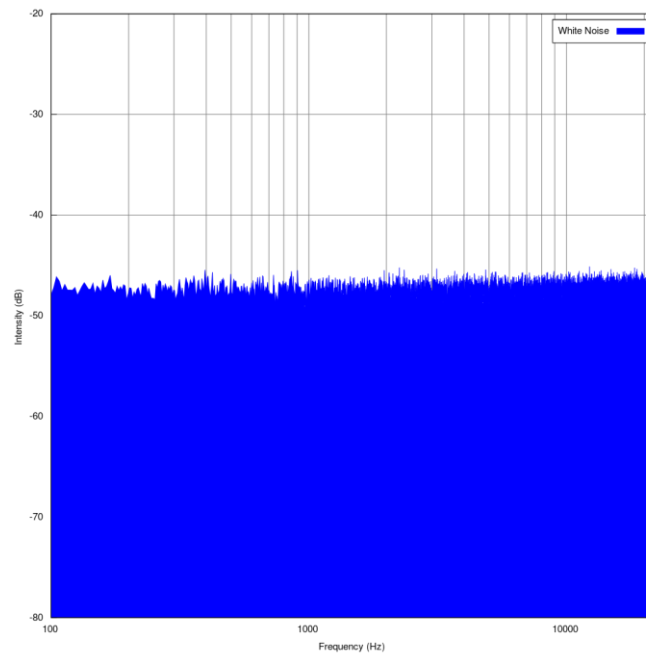


Figura 1 - Espectro do Pink noise

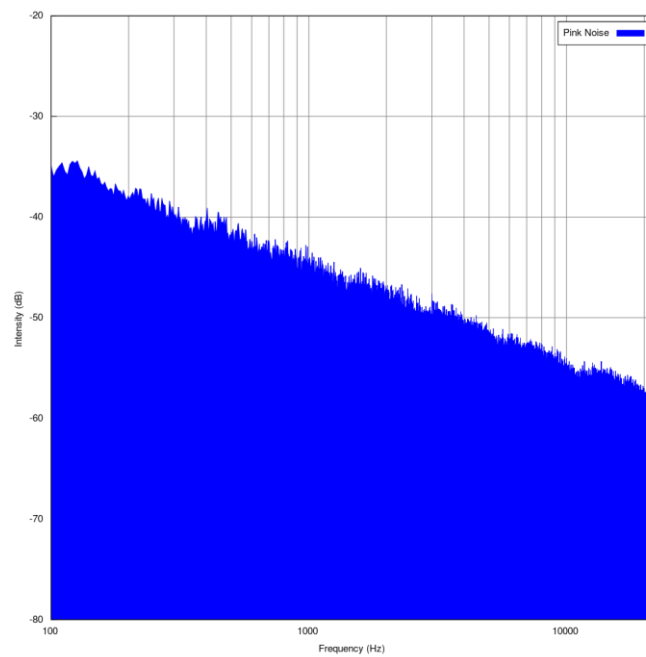


Figura 2 - Espectro do white noise

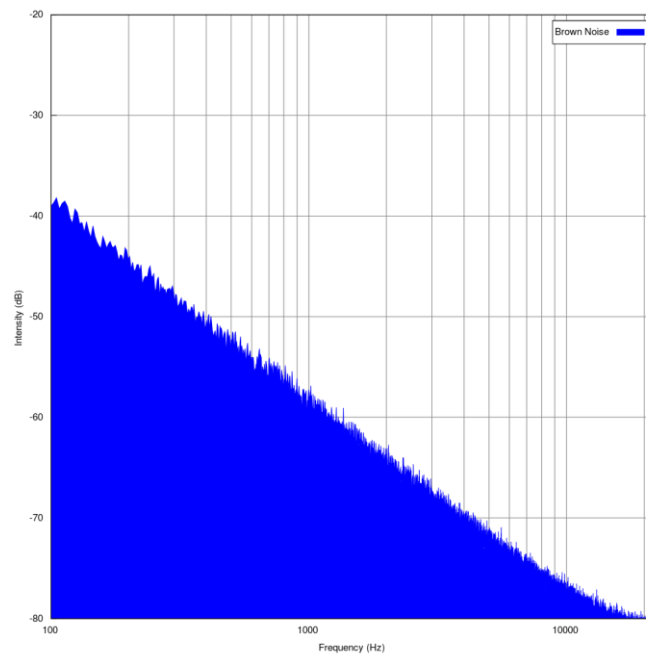


Figura 3 - Espectro do Brown noise

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, o volume e o tipo de ruído que se irá adicionar à sample. Os testes foram realizados com total de 105 samples, cada uma correspondente a uma das 105 músicas da base de dados.

Para calcular a precisão foi desenvolvido um programa auxiliar, 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, sem ruído. A tabela 1 apresenta os resultados obtidos com a variação da duração utilizando o compressor lzma.

Tabela 1 – Resultados obtidos para o compressor lzma com a variação da duração das samples

Duração da sample	Precisão	Tempo de Execução
5 segundos	100%	511.0 segundos
10 segundos	99%	531.3 segundos
15 segundos	99%	586.9 segundos

Analisando a tabela, é possível verificar que os resultados são bastante parecidos e que os melhores resultados para a precisão são obtidos com samples de 5 segundos, enquanto os piores são obtidos para samples de 10 e 15 segundos. Em relação ao tempo de execução, a duração da sample parece ter 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 gzip. Os resultados obtidos com este compressor são apresentados na tabela 2.

Tabela 2 - Resultados obtidos para o compressor gzip com a variação da duração das samples

Duração da sample	Precisão	Tempo de Execução
5 segundos	93.8%	446 segundos
10 segundos	93,8%	500 segundos
15 segundos	94%	554.4 segundos

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, porem, de uma forma geral, a precisão aumenta com a duração da sample. Em relação ao tempo de execução, verifica-se que aumenta consoante o aumento da duração da sample.

2.2. Análise aos Compressores

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

Foram realizados testes para os 5 compressores disponíveis no programa, utilizando samples com duração de 15 segundos e sem ruído. A tabela 3 apresenta os resultados obtidos com a variação do compressor utilizado.

Tabela 3 - Resultados obtidos com a variação do compressor

Compressor	Precisão	Tempo de Execução
gzip	94%	549.4 segundos
7Zip	93.8%	734.9 segundos
zlib	93.8%	560.0 segundos
bzip2	93.9%	547.3 segundos
lzma	99%	586.9 segundos

Analisando a tabela, é possível concluir que os compressores que apresentam os piores resultados são o 7Zip, gzip, zlib e bzip2. Por outro lado, o compressor que apresenta o melhor resultado é o lzma. Destes compressores, o 7Zip e o lzma são os que demoram mais tempo e, ao contrário destes, o bzip2 é o que demora menos tempo. Posto isto, verifica-se que o compressor que apresenta uma melhor relação precisão/tempo de execução é o lzma, obtendo resultados surpreendentemente bons para o seu 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 duração de 15 segundos e com o tipo de ruído White noise. O compressor utilizado foi o lzma, por ter sido o compressor que apresenta a melhor relação precisão/tempo de execução. A tabela 4 apresenta os resultados obtidos com a variação do volume do ruído.

Tabela 4 - Resultados obtidos com a variação do ruído

Volume do Ruído	Precisão	Tempo de Execução
0	99%	586.9 segundos
0.2	93.9%	602.7 segundos
0.4	93.8%	650.6 segundos
0.6	92.7%	599.8 segundos
0.8	92.5%	583.9 segundos
1	90%	778.1 segundos

Analisando a tabela, é possível concluir que quanto maior o volume do ruído menor será a precisão. Para o compressor lzma, um volume de ruído baixo não aparenta ter impacto na precisão, porém quando o volume do ruído começa a ser mais significativo, a precisão diminui. É de realçar que mesmo com bastante ruído na sample, a precisão ainda conseguiu valores bastantes satisfatórios e surpreendentes. Em relação ao tempo de execução, o volume do ruído parece ter impacto na variação deste valor, isto é, quanto mais ruído maior é o tempo de execução.

2.4. Análise ao Tipo de Ruído

Foram realizados testes para os três tipos de ruído disponíveis no programa, utilizando samples com a duração de 15 segundos e com o volume de ruído de 0,6. O compressor utilizado foi o lzma, por ter sido o compressor que apresenta a melhor relação precisão/tempo de execução. A tabela 5 apresenta os resultados obtidos com a variação do tipo de ruído.

Tabela 5 - Resultados obtidos com variação do tipo de ruído

Tipo de Ruído	Precisão	Tempo de Execução
White noise	93.8%	591.8 segundos
Pink noise	89%	600 segundos
Brown noise	87%	580 segundos

Através da análise à tabela, é possível verificar que o White noise é o tipo de ruído que tem menos impacto na precisão, ao contrário o Brown noise, que apresenta um impacto significativo no valor da precisão, tornando mais difícil a identificação da música correta. 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 detalhadamente as características dos vários tipos de ruídos. Tal como é referido anteriormente, 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. 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.

4. Referências

https://en.wikipedia.org/wiki/Colors_of_noise

<https://www.soundofsleep.com/white-pink-brown-noise-whats-difference/>