universidade
de aveiro

Music recognition using compression

Gonçalo Machado | 98359

Grupo 1

Teoria Algorítmica da Informação, 2023/2024

NCD

NCD is an approximation, through the use of compression, of the Normalized Information Distance (NID), with the following formula:

$$\mathrm{NCD}(x, y) = \frac{C(x, y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}},$$

where C(x) is the number of bits required by compressor C to represent x and C(x, y) is the number of bits needed to compress x and y together (usually, the two strings are concatenated).

An NCD closer to 1 means the files are not similar, while closer to 0 means they are similar.

Methodology - Data Processing and Preparation

The whole process was divided into smaller programs to better test and obtain results:

- fresh_start.sh - Delete all files from previous experiments.
- original_to_standard.sh - Transforms original songs to standardized songs
- get_test_samples.sh - Creates samples from 30 random songs with different durations.
- get_noise_samples.sh - Creates new samples with white noise added.
- create_database_dataset.sh - Uses GetMaxFreqs to generate frequency files for the songs, as well as calculate the bits needed to compress them using different compression methods.
- create_database_test.sh - Same function as the previous one, but for the samples.

Methodology - offbrand_shazam.sh

This is the program that is used to identify the song a sample is from using the NCD.

Input parameters:

- f - The frequency file of the sample.
- c - The compression method we want to use.
- n - The number of results to present.

Methodology - Results

To obtain results three programs were developed:

- get_results.sh - Obtains results for every combination of samples and variables.
- parse_results.py - Parse results into a JSON that will be used for analysis.
- graphs.py - Generate graphs used for analysis.

Methodology - Results

```
{
       "File": "Inertia-Interval-11_21_Noise_0.32.txt",                                  ##Name of test sample
       "Noise": 0.32,                                                                    ##Volume of whitenoise
       "Duration": 10,                                                                   ##Duration of sample (s)
       "BaseFile": "Inertia",                                                            ##Original Song
       "Compression": "lzma",                                                            ##Compression method
       "Top": {                                                                          ##Top 10 results
          "Inertia": 0.9751857035202928,                                                 ##Song and NCD value
          "AJR_-_Birthday_Party_(Official_Audio)": 0.9911755817467512,
          "AJR_-_BANG!_(Official_Video)": 0.991313386294454,
          "AJR_-_Adventure_Is_Out_There_(Official_Audio)": 0.9913798567535019,
          "AJR_-_Growing_Old_On_Bleecker_Street_(Official_Audio)": 0.9918896481256785,
          "AJR_-_Dear_Winter_(Official_Video)": 0.991957311886165,
          "AJR_-_Woody_Allen_(Official_Audio)": 0.9920832242868359,
          "AJR_-_Yes_I'm_A_Mess_(Official_Visualizer)": 0.9922881102126385,
          "AJR_-_Big_Idea_(Official_Audio)": 0.9924664264657713,
          "Steve_Aoki_-_Pretender_feat._Lil_Yachty_&_AJR_[Ultra_Music]": 0.9925499492041991
       }
       “CorrectGuessPlace”: 1                                                            ##Place of correct guess
   }
```

JSON object created to represent the result of a test with the variables associated

Results

Database songs:

- 79 songs by AJR
- From 2:30 to 5:00 minutes of duration

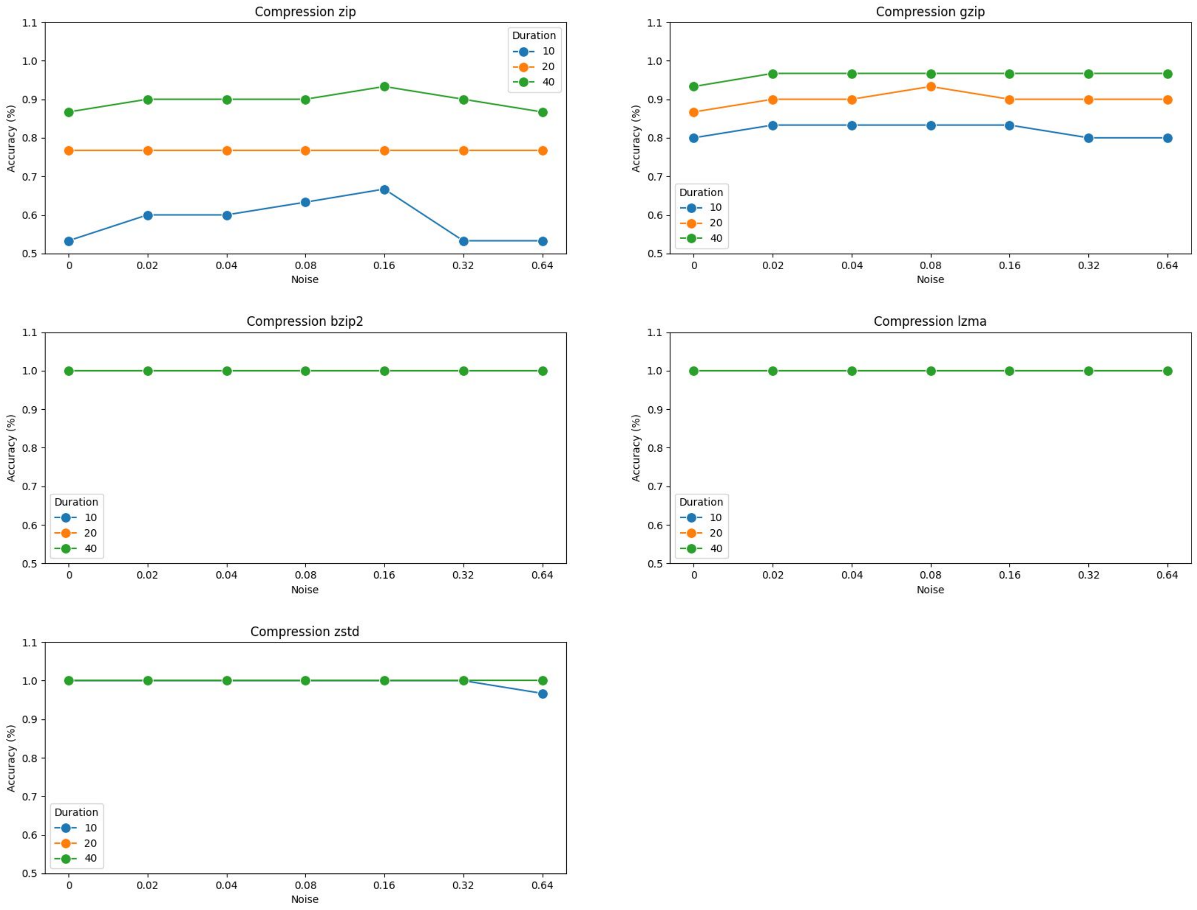
Range of variables used:

- White noise volume - {0, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64}
- Compression method- {zip, gzip, bzip2, lzma, zstd}
- Duration (secs) - {10, 20, 40}

Top	Number of correct guesses	Accuracy (%)
1	2923	93.11
3	3004	95.70
5	3037	96.75
10	3063	97.58
Total	3139	100

Table 1 - Accuracy of program

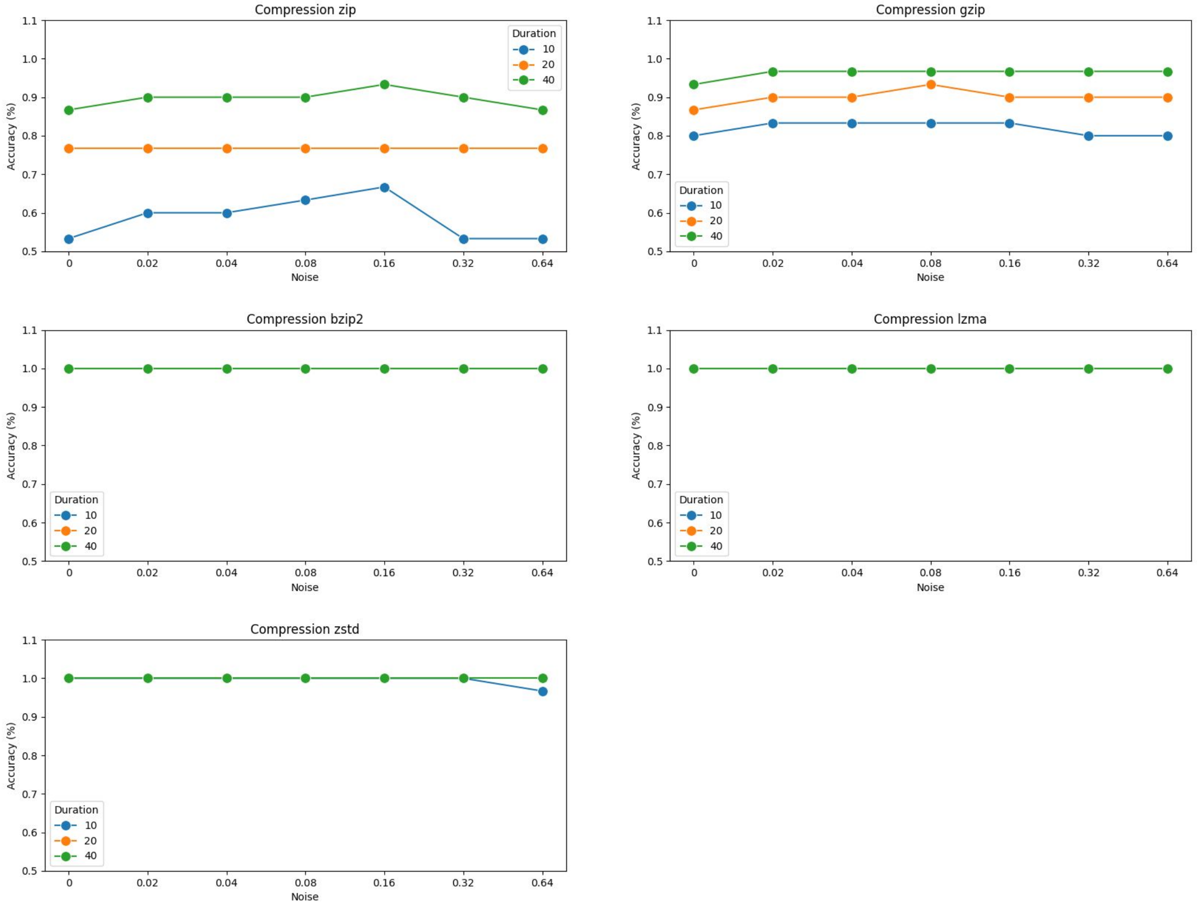
Results



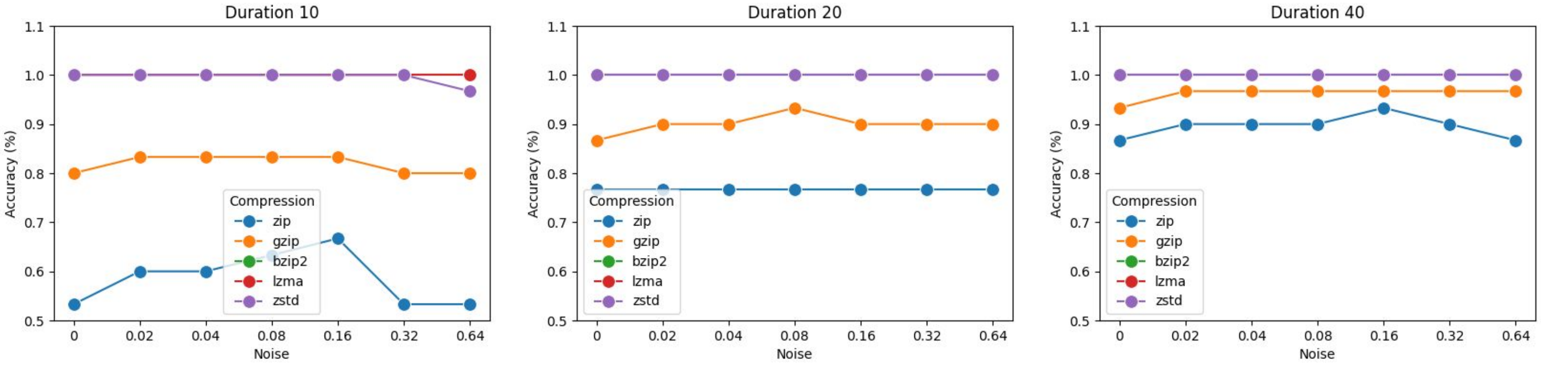
lzma, bzip2 and zstd with a near 100% accuracy -> more complex algorithms than zip and gzip

Results

The longer the sample used, the better the accuracy on zip and gzip -> DEFLATE benefits from repeated sections

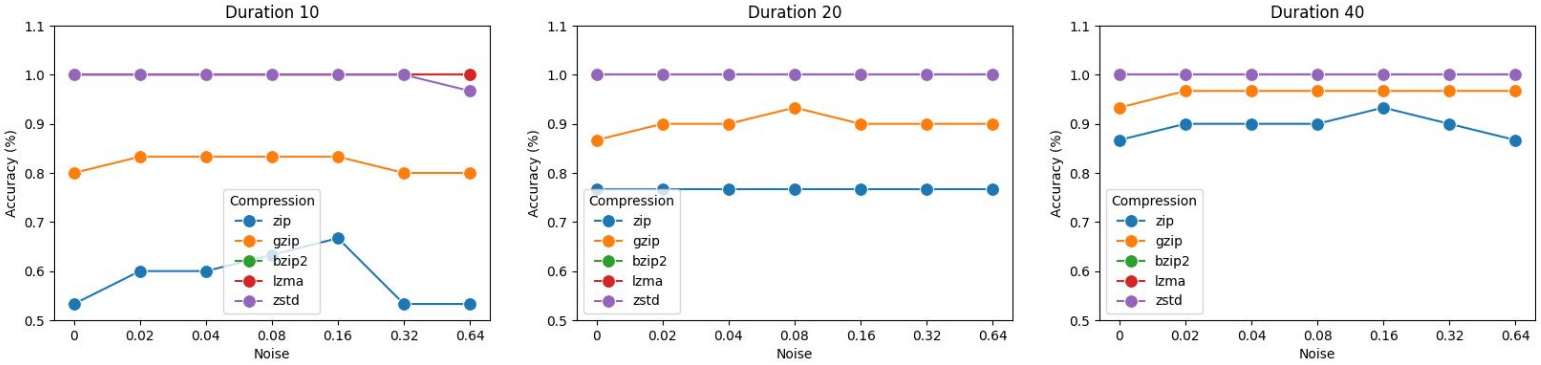


Results



White noise volume does not negatively impact accuracy in most cases, and improves it in some cases

Results



More duration -> less spread results -> longer samples less affected by other variables

Conclusion

- The results obtained showed that the overall accuracy of the developed program was 93.1% for identifying the correct song using a sample, which we take as proof that NCD can be used to identify songs using samples.
- The duration of the sample is directly related to the accuracy of the program, with the longer the sample the better the accuracy.
- White noise volume, we unexpectedly found out that white noise does not have an impact on accuracy.
- Compression methods impact accuracy, with lzma,zstd and bzip2 outperforming gzip and zip by a considerable amount.

Future Work

- Use songs from different genres, with different instruments and with/without vocals.
- Analyze different metrics for better understanding quality of results and performance of programs.
- Use other programming languages to possibly increase efficiency, like C++ or CUDA
- Make code more readable, documented and automation-friendly.