UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA

Projeto ITI: Parte 2

Joíson Oliveira Pereira - 20170068553.
Rebeca Raab Bias Ramos - 20170070453.

Professor: Derzu Omaia
Disciplina: Introdução a Teoria da Informação.

João Pessoa - PB
Novembro 2021

Sumário

Introdução

O reconhecimento de sons/fala é totalmente comum atualmente na vida das pessoas. Desde a Siri, da Apple, à Alexa, da Amazon, temos lidado com plataformas que interpretam o que dizemos e executam funções como tocar música, enviar mensagem de texto, criar um lembrete ou até analisar o conteúdo de uma conversa.

Como nós, seres humanos, o reconhecimento de voz tem aprimorado sua capacidade de interpretação da linguagem oral e se tornado mais precisa. Este tipo de reconhecimento permite a um software captar um som ou fala de uma pessoa e torná-la um texto. Um programa ou aplicativo com essa função registra o som/fala do usuário e quebra o áudio em partes individuais. Cada uma dessas partes, então, é analisadas por meio de algoritmos para identificar que palavras melhor representam cada som registrado. Então, o áudio é transcrito de forma inteligível ao sistema.

O propósito da segunda parte deste projeto é usar o banco de dados previamente determinado para obter o reconhecimento de padrões baseado em PPM ou LZW. Sendo assim, foi-se escolhido o LZW, tendo em vista que havíamos trabalhado com este algoritmo na primeira parte do projeto. Inicialmente será gerado um dicionário para as etapas de treinamento de cada categoria do banco de dados selecionado, IRMAS.

Dessa forma, utilizando a técnica de validação cruzada para dividir o banco de dados em amostras de treinamento e classificação, ou seja, para cada categoria no banco de dados, ou cada um de um total de 10 instrumentos, selecione todas as amostras (100 por instrumento) - 10 para treinamento (90 para treinamento) e 10 amostras para classificação, a seleção da amostra é aleatória. Para a classificação, o algoritmo K-Nearest Neighbours (KNN, k-Nearest Neighbors) será usado, K = 1, e o tamanho do arquivo compactado será usado como a métrica da distância que é a quantidade de índices utilizado pelo algoritmo.

O treinamento engloba a parte responsável pela geração de um dicionário para cada categoria do banco de dados. Enquanto isso, a classificação/testes coloca em todos os modelos/dicionários 1 amostra comprimida(que não foi utilizada na geração do modelo). No decorrer da compressão da amostra de teste, o dicionário se manterá estático e a amostra de teste será atribuída ao modelo que gerou a melhor compressão.

Metodologia

Utilizando o banco de dados de instrumentos musicais IRMAS Data base, foi necessário primeiramente realizar um tratamento no banco de dados para ser utilizado neste trabalho. Pois, o database contava com 10 pastas que representavam cada uma um tipo diferente de instrumento musical disponível no banco. Entretanto, a nomeação dos arquivos de áudio com extensão .wav possuía nomes que dificultavam a leitura e consequentemente a implementação do reconhecedor de padrões.

Dessa forma, foi-se usado o script (**renomear.py**) escrito em Python para deixá-lo mais fácil para leitura e uso no projeto. Com isso, os arquivos passaram a possuir o seguinte formato: *abreviação_do_nome_do_instrumentoX*, onde x representa uma ordenação começando por 1. Exemplo: cel1.

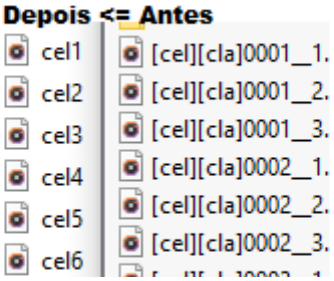

Figura 1: Renomeação dos arquivos

Também foi-se necessário realizar a exclusão dos sons numerados acima de 100, tendo em vista que algumas pastas possuíam bem mais sons do que outras, então com intuito de deixar equivalente as pastas, foi-se determinado o número de 100 sons para cada instrumento.

Além disso, foi-se descartado o cabeçalho do arquivo e utilizada apenas a sequência de pixels. O código foi desenvolvido na linguagem de programação Python na versão 3.8.4 no ambiente Jupyter Notebook, com auxílio das bibliotecas numpy, scykit-learn, time, e matplotlib.

Ademais, o banco de treinamento de teste foi escolhido de forma aleatória com a utilização do numpy.random, com 90 arquivos para treino e 10 para classificação, no total de 100 arquivos por instrumento, em um total de 10 instrumentos no banco de dados. Em relação aos valores de K, foram testados todos os valores no intervalo de K = 9 ao K = 16. Durante a classificação, o dicionário se manteve estático. Vale ressaltar também que as curvas de Taxa de acerto por K e de Tempo de Processamento por K serão apresentados na seção de Resultados deste trabalho. Por fim, a métrica de distância utilizada foi a quantidade de índices do LZW, isto é, a amostras de teste será atribuída ao modelo com o menor número de índices (melhor compressão).

Resultados e discussões

Feitas as importações das bibliotecas, foi-se implementado em seguida a **Função treino**, que tem dois parâmetros: trein que representa a parte do banco de dados que foi separada para o treino e que representa o índice do dicionário com k bits. Ela retorna o dicionário resultante do treinamento utilizando os dois parâmetros.

A **Função de teste** em síntese recebe dois parâmetros: mensagem e dicionário. Sendo mensagem a representação de uma amostra da parte separada do banco de dados para os testes que nesse caso é um som e dicionário a representação de um dicionário que já passou pela fase de treinamento. Sendo assim, a função abaixo retorna o tamanho do som, recebida como parâmetro, comprimida.

Para a implementação do reconhecedor de padrões de som, foi-se declarado a variável K que representa uma lista com valores entre 9 e 16 que são os tamanhos de bits para serem testados nos índices do dicionário, utilizou-se a variável tempo para a medição do tempo de execução do algoritmo e a variável acertos representa a quantidade de vezes de quando a decodificação dos sons testados deram certo com o dicionário feito pela função de treino.

Para avaliar o desempenho do reconhecedor de padrões, deve-se levar em cada a taxa de acertos que neste trabalho correspondeu a figura 1 abaixo:

```
RESULTADO: K = 9 -> acertos: 1
RESULTADO: K = 10 -> acertos: 2
RESULTADO: K = 11 -> acertos: 1
RESULTADO: K = 12 -> acertos: 3
RESULTADO: K = 13 -> acertos: 2
RESULTADO: K = 14 -> acertos: 2
RESULTADO: K = 15 -> acertos: 2
RESULTADO: K = 16 -> acertos: 1
```

Figura 2: Exemplo de saída do reconhecedor de padrões de som, com valor de acertos para cada K de 9 à 16.

Onde K= 13, possui o melhor número de acertos nesse teste. Entretanto, o número de acertos varia a cada vez que é rodado o algoritmo, sendo assim é possível se obter valores diferentes em cada teste, podendo em outra interação outro valor de K possuir mais números de acertos.

Também é possível verificar olhando para o gráfico do tempo pelo K que o K= 13 apresentou o melhor tempo, ou seja, foi mais rápido que os demais valores de K. Lembrando que este também varia a cada vez que se executa o programa.

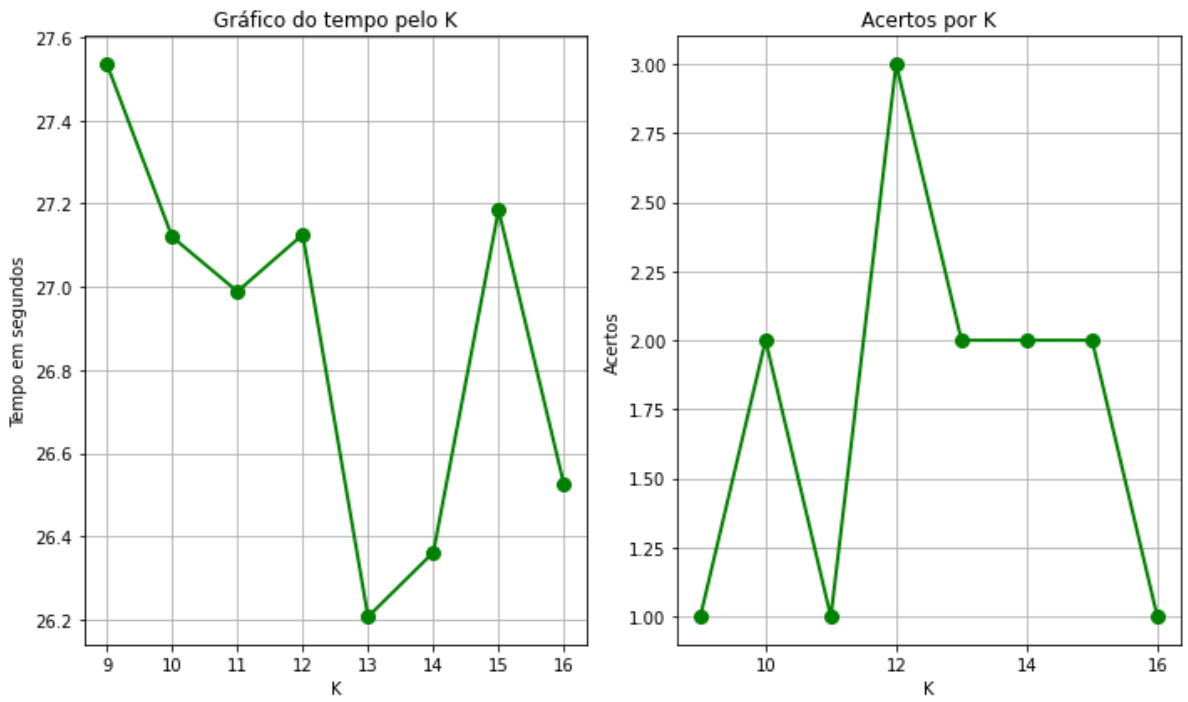

Figura 3: Gráficos do tempo pelo K e de acertos por K

Por fim, foi executada 10 vez o código a fim de se obter um K “médio” e o tempo médio da melhor execução, como mostra a tabela abaixo:

N° da execução	Melhor K da execução	Melhor Tempo da Execução
1	K = 13	t = 26.2s
2	K = 13	t = 25.9s
3	K = 16	t = 26.22s
4	K = 13	t = 26.5s
5	K = 14	t = 27.5s
6	K = 16	t = 26.8s
7	K = 15	t = 26.1s
8	K = 14	t = 26.98s
9	K = 12	t = 26.6s
10	K = 15	t = 26.5s

Tabela 1: Melhores valores de K e de tempo de 10 execuções do código.

Vale ressaltar também que em algumas execuções houve empate no número de acertos de K o fator determinante para o desempate foi o que foi executado mais rápido.

Com base nos dados apresentados foi possível chegar aos seguintes valores médios, o valor de K foi-se escolhido vendo qual valor de K mais se repetiu durante as 10 execuções, ou seja, fazendo a moda. E o valor do tempo foi feito a média aritmética. Resultando em K = 13 e t = 26.53s.

Considerações finais

Tendo em vista que o objetivo do trabalho era aplicar os conhecimentos adquiridos na cadeira de Introdução à teoria da informação fazendo um reconhecedor de padrões com os algoritmos PPM ou LZW, ele foi atingido.

Inicialmente o grupo teve dificuldades em se familiarizar com conceitos que não estavam acostumados, como processos de treinamento e testes, além de utilização de bibliotecas da linguagem de programação escolhida Python que é conhecida por proporcionar uma facilidade na manipulação de dados.

Sendo assim, foi possível implementar o reconhecimento de padrões enquanto era fixado o conteúdo aprendido nas aulas.

Referências

- Slides do Professor
- https://tecnoblog.net/346980/como-funciona-o-reconhecimento-de-voz/
- https://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/