Reconocimiento/ Clasificación de vocales Naive Bayes

Tratamiento digital del sonido 3º de GISAM

Grupo 7: David Santa Cruz Del Moral, Hao Zhou, Yuriy Alejandro Moreno Salomón

ÍNDICE

1. Introducción y objetivos del trabajo.

El proyecto se centra en el reconocimiento y clasificación de vocales mediante el uso del clasificador Naive Bayes. El objetivo principal es desarrollar un sistema capaz de identificar la vocal presente en un archivo de audio dado. Para lograr esto, se lleva a cabo la búsqueda o grabación de archivos de audio que contengan las cinco vocales (a, e, i, o, u), dividiendo las grabaciones en conjuntos de señales de entrenamiento y test. Las señales de entrenamiento sirven para tener una amplia base de datos de diferentes tipos de características de cada audio, con la que se entrenará el algoritmo de Naive Bayes. Por otro lado, las señales de prueba serán diferentes de las de entrenamiento, para poner a mejor a prueba el algoritmo.

La representación de los archivos de audio se abordará tanto en el dominio temporal como en el dominio frecuencial. Se realizará una cuidadosa selección y extracción de características representativas de cada vocal. En este caso en concreto, se verá en dos enfoques para la representación de características: los Coeficientes Cepstrales de Mel (MFCC) y los Coeficientes de Predicción Lineal (LPC):

- MFCC: Este método se basa en la emulación del comportamiento del oído humano al utilizar un banco de filtros Mel para extraer características perceptualmente relevantes de las señales de audio. Este enfoque nos permite capturar eficientemente propiedades acústicas y formantes específicos de las vocales. Al aplicar la Transformada de Fourier de corto tiempo (STFT) y el banco de filtros Mel, obtendremos los coeficientes MFCC, destacando las frecuencias fundamentales y las características esenciales de la pronunciación vocal.

- LPC: Por otro lado, los coeficientes LPC ofrecen una representación lineal que modela las propiedades de resonancia y forma del tracto vocal humano. La descomposición de la señal en componentes lineales nos permite capturar la envolvente espectral y la ubicación de los formantes. Este enfoque, al predecir la señal en instantes dados a partir de valores anteriores, revela la estructura de resonancia de la cavidad bucal durante la articulación de vocales.

Posteriormente, evaluaremos el rendimiento del sistema utilizando métricas estándar. La visualización detallada de los resultados proporcionará una comprensión profunda de la capacidad del modelo para reconocer y clasificar las vocales, permitiéndonos ajustar y mejorar el sistema de manera iterativa al aumentar el número de muestras.

2. Metodología

Grabación/Obtención de señales.

Para los audios de las vocales usadas en el entrenamiento del clasificador Naive Bayes se ha optado por usar las proporcionadas por la profesora en el aula virtual. Mientras, los audios de las vocales usadas en la prueba se han grabado en una cámara anecoica situada en el laboratorio de Ingeniería Acústica del laboratorio III, los audios fueron grabados por cada uno de nosotros dividiendo cada vocal en una pronunciación corta y otra extendiendo dicha vocal, se tuvo que cortar la grabación para quitar los silencios y para guardar dichos audios en mono para poder sacar los LPC.

Cómo se ha realizado la extracción de características y razones por las que se han escogido esas características.

Se han extraído dos características para la realización del proyecto:

- Los coeficientes de Mel (MFCC)
- LPC

Por un lado, se usa el análisis cepstral MFCC ya que ofrecen una capacidad de modelar eficientemente las propiedades perceptuales y acústicas específicas de las vocales, debido a que aplica un banco de filtros que imita la forma en que percibimos las frecuencias, dando importancia a ciertas bandas de frecuencias perceptualmente relevante (como a las bajas frecuencias). Los primeros coeficientes poseen la mayor parte de la información como la envolvente espectral y la estructura de formantes.

Por otro lado, la predicción lineal permite modelar de forma simplificada el mecanismo de producción de voz y extraer características relevantes de la señal: envolvente del espectro o formantes. Esta técnica permite parametrizar una señal con un número pequeño de parámetros y no requiere demasiado tiempo de procesamiento, por lo que es perfecta para el reconocimiento de fonemas o palabras.

En conclusión, la razón principal por la que se han elegido estos parámetros para extraer es porque en el fondo van a tener la misma información, es decir, ambos coeficientes contienen información sobre las formantes o de forma general sobre la envolvente espectral. Entonces, la idea es comparar los resultados del clasificador cuando se usan unas características frente a otras.

En el apartado MFCC se pueden observar dos funciones, *mfcc_train* y *mfcc_test*. Ambas funciones reciben una lista de audios obtenida de distintas carpetas, *vocals* (vocales del AV) y *Audios* (vocales grabadas) respectivamente. Luego, se sacan los coeficientes de Mel de cada audio (13 porque los primeros coeficientes poseen la mayor parte de la información como la envolvente espectral o las formantes), a los que se aplican una media a cada coeficiente debido a que cada coeficiente está compuesto por varias ventanas que se sacan por defecto al aplicar la función de extracción de coeficientes y solamente interesa tener un valor. El resultado de la media de los

coeficientes se va añadiendo a una lista llamada *mel* y, una vez estén todos los coeficientes del audio en dicha lista, esta se introduce en otra llamada *mfcc_train*, donde se guardarán los MFCC de todos los audios. Además, se crea otra lista en ambas funciones llamada *vocals* en la que se añaden las vocales extraídas del nombre de los audios para saber a qué vocal pertenece cada conjunto de coeficientes de Mel. Finalmente, ambas funciones devuelven las listas mencionadas (*mfcc_train()* → *mfcc_train* y *vocals_train*. *mfcc_test()* → *mfcc_test* y *vocals_test*) en forma de lista *NumPy*, ya que *librosa* solo trabaja con dicha esto.

En el apartado LPC también se observan dos funciones, una de entrenamiento, *lpc_train*, y una de prueba, *lpc_test*, que hacen lo mismo, pero para audios diferentes (los de entrenamiento y los de prueba). Ambas empiezan con el mismo funcionamiento, extrayendo de los audios f_s y la señal para su utilización posterior. Para obtener los coeficientes de predicción lineal, se usa la función *predlin* (extraída de *tds_utils*) y que usa como parámetros de entrada un frame de la señal de audio, un orden de coeficientes y una ventana de Hamming. El orden para los LPC, p, se ha elegido a diez por dos razones: la primera, porque la ganancia de predicción aumenta con el orden de predicción, pero satura aproximadamente para p = 10 – 12; y la segunda, porque se ha probado con diferentes valores para el orden y se puede concluir en que p = 10 es el orden óptimo para encontrar el conjunto de parámetros que minimicen la energía del error de predicción. Por otro lado, la ventana de Hamming se ha elegido de este tamaño de forma empírica, se ha comprobado que con 30, 40 y 50 ms dan los resultados más altos en Naive Bayes. Finalmente, los coeficientes sacados se introducen a una lista, para que guarde los coeficientes de cada audio; y luego, se convierte a una lista *NumPy*, por la misma razón que con los MFCC.

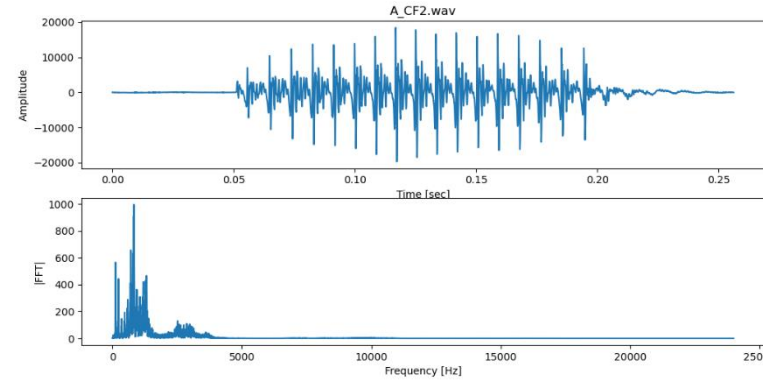
- Cómo se han formado las estructuras de datos que se han pasado como entrada a los clasificadores.

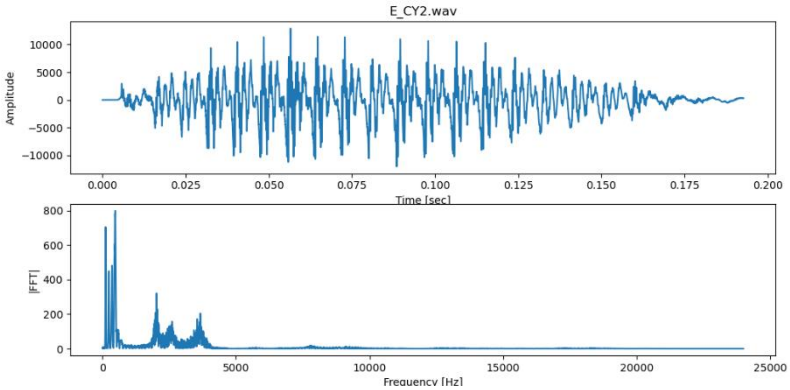
Como se menciana anteriormente, cada función del código devuelve dos listas, los valores de los coeficientes y las vocales a las que corresponde cada coeficiente. Estas listas son las que usaremos como base de datos para Naive Bayes.

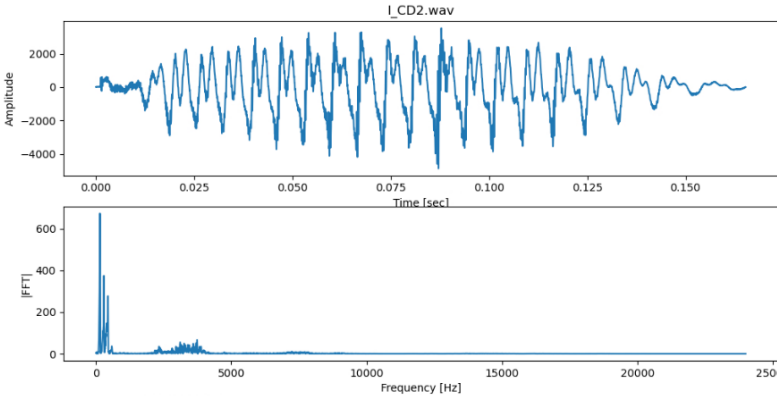
- Cómo se han dividido los datos para entrenamiento y prueba.

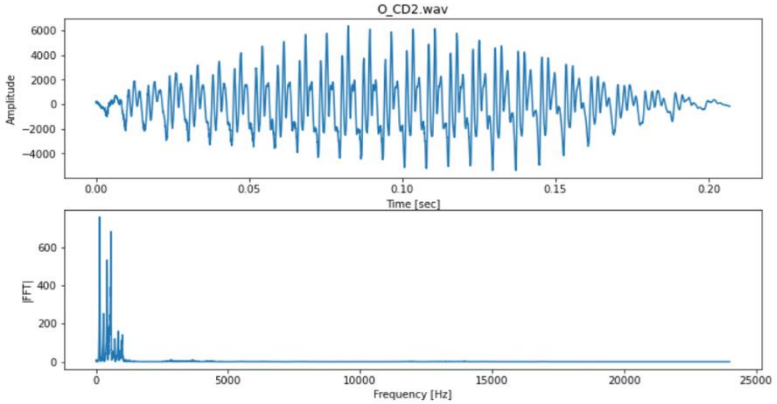
Con las listas que componen la base de datos, *def mfcc_train()* → *mfcc_train* y *vocals_train*; *def mfcc_test()* → *mfcc_test* y *vocals_test*, se trasladan al programa de Naive Bayes. En dicho programa, las listas de la base de datos se igualan a sus respectivas variables dentro del programa, es decir, *X_train_mfcc = mfcc_train*, y_train = *vocals_train_mfcc*, *X_test_mfcc = mfcc_test* y *y_test_mfcc = vocals_test* y lo mismo para los LPC. Por tanto, los datos que se usan como entrenamiento son las características extraídas de los audios que se nos proporcionan en el AV (MFCC y LPC) y los datos de prueba son las características extraídas de los audios propios grabados.

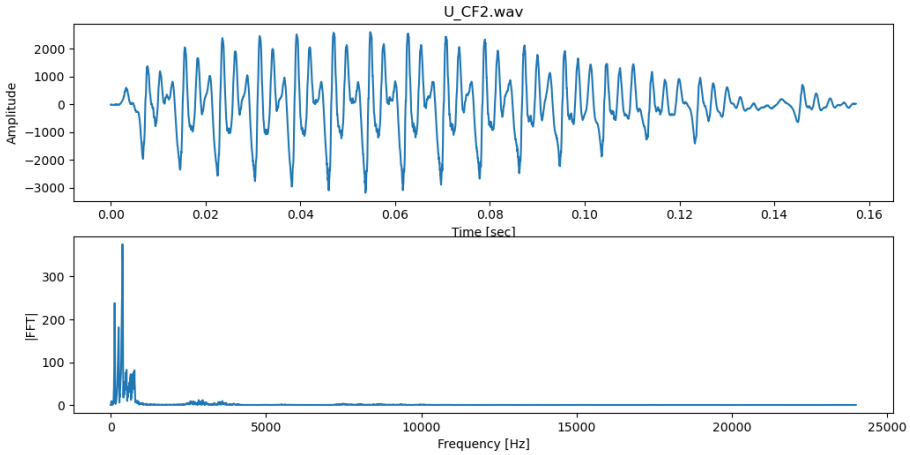
- Representaciones de señales en tiempo y frecuencia.











3. Pruebas y resultados

Como la idea es comparar los resultados del clasificador cuando se usan unas características frente u otras, se han usado dos modelos de Naive Bayes diferentes. Es decir, un modelo que se entrena y se predice mediante los coeficientes de Mel y otro con los coeficientes LPC.

Con los datos que se han mencionado a lo largo del trabajo, el resultado del clasificador Naive Bayes según que coeficientes se usen da lo que se ve en las siguientes tablas:

MFCC
Porcentaje de acierto: 76.66666666666667 %

Predict Vocals	i	u	e	u	e	o	a	u	i	u	i	a	u	a	e	i	u	u	e	u	a	i	u	i	i	i	a	a	u	u
Original Vocals	i	u	e	u	e	o	a	o	e	u	i	a	o	a	e	i	o	u	e	o	a	e	u	i	i	i	a	a	u	o

LPC
Porcentaje de acierto: 33.33333333333333 %

Predict Vocals	o	o	e	a	o	o	a	o	a	a	a	a	o	e	o	o	o	a	o	a	i	e	o	o	o	o	a	o	a	o
Original Vocals	i	u	e	u	e	o	a	o	e	u	i	a	o	a	e	i	o	u	e	o	a	e	u	i	i	i	a	a	u	o

Como se observa, utilizar los coeficientes Mel para el reconocimiento y clasificación de vocales es notoriamente mejor que usar los coeficientes LPC.

4. Conclusiones y líneas de memoria

En el proceso del trabajo se encontraron dos problemas en los códigos facilitados, siendo una de ellas la obtención de los MFCC y LPC, y el otro problema fue el funcionamiento de dichas características con Naive Bayes.

En el primer caso se encontraron problemas en importar a los programas MFCC y LPC todos los audios de una carpeta. Para solventar dicho problema se utilizaron un ajuste en cómo se extraía los audios de las carpetas y una serie de cambios en el guardado de las variables de nuestro programa. Luego, en el caso del programa MFCC se realizaron cambios en su guardado de características ya que se obtuvieron múltiples valores por coeficiente debido a que la función que extrae los coeficientes utiliza un tamaño de ventana por defecto y el resultado de un coeficiente era una lista de valores (un por ventana); por tanto, como se quería un solo valor por coeficiente se optó por hacer un promediado del conjunto obtenido. Por otro lado, la extracción de LPC en dichos audios solamente se pueden obtener atreves de archivos .wav que sean en modo mono y no en estéreo por lo que se tuvieron que cambiar los de estéreo a mono. También, se tuvo que ajustar la frecuencia de muestreo de los audios grabados (48000) para que coincidiera con los del aula virtual (8000 Hz) y así poder realizar la comparación en Naive Bayes.

En el segundo caso para poder ejecutar Naive Bayes con los MFCC y LPC se requería que los coeficientes obtenidos se guardaran en matrices NumPy, debido al funcionamiento de *librosa*. Se solucionó modificando en los programas de MFCC y LPC cambiando las listas que guardan las características a dicho formato.

En el caso de mejora en la precisión de identificación de las vocales, se han logrado resultados óptimos de los LPC como se menciona en el apartado de metodología. Para el caso de los coeficientes de Mel se obtuvieron resultados óptimos la precisión en el caso experimental (N.º de coeficientes de Mel = 19), que son mayores al número de coeficientes propuesto en la teoría (N.º coeficientes de Mel = 13). Entonces, para mejorar los resultados para los MFCC, habría que aumentar el número de coeficientes extraídos a 19.

MFCC Porcentaje de acierto: 93.33333333333333 %																												
Predict Vocals	i	u	e	u	e	o	a	o	e	u	i	a	o	a	e	i	o	u	e	o	a	i	u	i	i	i	a	a	u	a
Original Vocals	i	u	e	u	e	o	a	o	e	u	i	a	o	a	e	i	o	u	e	o	a	e	u	i	i	i	a	a	u	o

Para mejorar el porcentaje de aciertos, se podría aumentar el número de archivos de audio que se usen para entrenar, que haría que se aumentara el número de características en la base de datos. De esta forma, las características de los audios de prueba tendrían más datos con los que cotejarse, aumentando la probabilidad de acertar el reconocimiento de la vocal mediante Naive Bayes. En el caso de no tener límite de tiempo, se aumentaría el número de audios de entrenamiento de forma que tendieran a infinito, ya que el coste computacional no sería un problema por el tiempo ilimitado.

5. Bibliografía

- Departamento de matemática aplicada, Universidad Politécnica de Madrid (2021). Introducción al Aprendizaje Automático, Clasificación Naive-Bayes.
https://dcain.etsin.upm.es/~carlos/bookAA/02.1_MetodosdeClasificacion-Naive-Bayes.html
- Rebeca G.E. Escuela Técnica Superior de Ingeniería de Telecomunicaciones Universidad Rey Juan Carlos (2023). Naturaleza y percepción de la señal acústica.
https://www.aulavirtual.urjc.es/moodle/pluginfile.php/11912555/mod_resource/content/1/TDS.pdf
https://dcain.etsin.upm.es/~carlos/bookAA/02.1_MetodosdeClasificacion-Naive-Bayes.html
- Mario C.M. Escola Tècnica Superior d'Enginyeria Informàtica, Universitat Politècnica de València (2021). Clasificación de textos basada en redes neuronales.
https://riunet.upv.es/bitstream/handle/10251/172276/Campos%20-%20Clasificacion%20de%20textos%20basada%20en%20redes%20neuronales.pdf?sequ
- Rebeca G.E. Escuela Técnica Superior de Ingeniería de Telecomunicaciones Universidad Rey Juan Carlos (2023). Tratamiento Digital de Sonido.
https://www.aulavirtual.urjc.es/moodle/pluginfile.php/11912559/mod_resource/content/0/tema5TDS.pdf
- Rebeca G.E. Escuela Técnica Superior de Ingeniería de Telecomunicaciones Universidad Rey Juan Carlos (2023). Tecnologías del Habla.
https://www.aulavirtual.urjc.es/moodle/pluginfile.php/11912564/mod_resource/content/0/TDStema6_1.pdf