
SISTEMA DE COMPRESIÓN DE SEÑALES DE AUDIO MEDIANTE AGRUPACIÓN DE PATRONES DE FRECUENCIA

202200135 – Joab Israel Ajsivinac Ajsivinac

Resumen

El “Centro de Investigaciones de la Facultad de Ingeniería” requiere desarrollar una solución para la compresión de señales de audio basada en los parámetros denominados frecuencia y amplitud. Estos parámetros esenciales describen el sonido con relación al tiempo. El centro aborda esta problemática como un desafío combinatorio NP-Hard, con un enfoque basado en agrupamientos.

El programa carga las matrices originales de frecuencia mediante archivos de entrada, además cuenta con la funcionalidad para crear matrices binarias a partir de las originales, las cuales se usan para identificar patrones similares, con el fin de sumar las tuplas originales en base a los patrones similares obtenidos, para luego generar las matrices reducidas de frecuencias. Finalmente, el programa genera diferentes archivos de salida como: graficas en forma de nodos, o archivos con el mismo formato al archivo de entrada.

El sistema busca simplificar el análisis que efectúan los investigadores, cuando se estudie una gran cantidad de muestras de audio.

Palabras clave

Señales, compresión, amplitud, frecuencia, matriz

Abstract

The “Centro de Investigaciones de la Facultad de Ingeniería” requires the development of a solution for audio signal compression based on the parameters known as frequency and amplitude. These essential parameters describe sound in relation to time. The center addresses this problem as an NP-Hard combinatorial challenge, with a clustering-based approach.

The program loads the original frequency matrices through input files and has the functionality to create binary matrices from the originals, which are used to identify similar patterns, to sum the original tuples based on the similar patterns obtained, and then generate the reduced frequency matrices. Finally, the program generates different output files such as: graphs in the form of nodes, or files with the same format as the input file.

The system seeks to simplify the analysis conducted by researchers when studying many audio samples.

Keywords

Signals, compression, amplitude, frequency, matrix.

Introducción

El sistema de compresión de señales de audio es una herramienta que resulta de gran importancia en la industria de la telecomunicación y radiodifusión, ya que facilita la visualización de patrones en las señales de audio así como poder obtener de un gran grupo de datos sus señales reducidas para su futuro análisis.

El software proporciona la forma de poder generar graficas para las diferentes señales tanto originales como las reducidas mediante gráficos generados con el software de diseño de diagramas Graphviz. Con el fin de ayudar a la identificación eficiente de los patrones de las diferentes señales.

Desarrollo del tema

El sistema se basa en la utilización de listas enlazadas para la creación de los diferentes elementos que conforman una señal de audio.

Una lista enlazada tiene como base el almacenar una colección de elementos, donde cada elemento se denomina nodo que contiene dos partes principales: el valor del elemento, y una referencia o enlace al siguiente nodo de la secuencia.

Las matrices utilizadas están creadas a partir de listas enlazadas, de forma que una lista general guarda como elementos a otras listas, de la siguiente forma: la primera lista haciendo referencia a las filas, y la lista contenida en la primera lista hace de referencia a las columnas, dentro de esta segunda lista están guardados los datos de las frecuencias.

El sistema utiliza como lenguaje de programación principal Python, el cual es un lenguaje de alto nivel y orientado a objetos. Se usó dicho lenguaje por su sintaxis legible y clara, que se asemeja al lenguaje humano lo que facilita la comprensión del código. El programa es manejado únicamente mediante la

consola y la interacción con el menú, de modo que el uso del sistema sea más familiar hacia el usuario. El programa se seguirá ejecutando hasta que el usuario se lo indique seleccionando la opción respectiva del menú de opciones.

a. Carga de Datos

Los datos se cargan al sistema mediante el ingreso de la ubicación dentro del sistema del archivo XML, para luego almacenar la información obtenida de la señal de audio en memoria en diferentes listas para completar la lista principal la cual contiene el nombre de la señal de audio junto con su respectiva matriz de frecuencias. Este proceso se puede llegar a realizar n veces con diferentes archivos XML.

La creación de las matrices toma a los tiempos como filas y a las amplitudes como columnas.

El sistema cuenta con una verificación de nombres, ya que no acepta nombres de señales de audio repetidas, por lo que si encuentra un nombre idéntico a uno ya existente en memoria se procede a cambiar el valor actual con el nuevo valor.

b. Procesado de la Información

Posterior a la carga de los datos iniciales si el usuario elige la opción de “procesar archivo” el programa procede a calcular la matriz reducida de todas las señales de audio que estén cargadas en memoria.

A partir de la lista principal de señales, se procede a calcular la matriz binaria o de patrones de cada señal, esto mediante un bucle que itere las diferentes señales agregadas al sistema y utilizando una copia de la matriz original respectiva para cada señal de audio, dicha copia se realiza mediante la iteración de los elementos de las matrices originales para poder copiar los valores y no solo la dirección de memoria

asignada a la matriz original. Se usa una copia para mantener los valores de los archivos iniciales sin ningún tipo de cambio.

Una matriz binaria es una estructura de datos que consiste en una cuadrícula rectangular de elementos que solo pueden tomar dos valores: 0 y 1. Se asigna un 1 si existe un valor diferente de 0 en la matriz original, y si el valor es 0, no se modifica el valor.

La obtención de la matriz binaria se hace mediante la iteración de los elementos, iterando primeramente entre las filas, para luego iterar la lista que contiene la columna y el valor, si el valor de la matriz copia es diferente de 0, se le asigna un 1 en la posición (fila y columna) donde se encuentra el valor, y si es 0, no se hacen modificaciones en la copia.

Con la matriz de patrones se procede a crear los grupos respectivos, para esto se precisan de la matriz original se procede a iterar cada fila para comparar si existen semejanza entre sus valores, la semejanza se debe dar a lo largo de todos los valores de una fila.

El proceso que se sigue para la reducción de las matrices es el siguiente, se obtiene la primera fila (llamada en este caso actual), dentro de un bucle se crea una variable que contenga el siguiente nodo respecto de la fila actual (llamada fila de comparación), y se crea una variable que contenga los valores de la fila actual, además de eso se crean variables para guardar el nombre de los grupos junto con los tiempos en los cuales existen semejanzas, luego dentro del primero bucle se crea un segundo bucle que se encarga de comparar las filas de comparación con la actual en este caso, si los valores de la fila actual y la de comparación son iguales quiere decir que comparten un patrón.

El proceso de comparación de filas hace uso del método `__eq__` que sirve para comparar dos objetos

diferentes, el método para el sistema itera los valores de cada fila, con un valor de la fila que no sea igual se rompe el ciclo y devuelve un valor Falso, si completa el ciclo regresa un valor True.

Luego de recibir la respuesta de parte de la comparación se procede a concatenar los tiempos y sumar las tuplas.

Para la suma de las tuplas se usa el método `__add__` el cual permite agregar dos expresiones y producir una nueva expresión, de modo que se iteran los valores contenidos en la lista de las columnas para poder sumar cada valor y regresar una lista con los datos ya sumados.

Luego de a ver sumado las tuplas se eliminan la fila según el índice (el índice en este contexto hace referencia a la fila donde está la información) ya que dicha fila no puede ser parte de otro grupo, debido a la eliminación se crea una variable temporal que contenga el valor del siguiente nodo de la lista de comparación, ya que la eliminación desenlaza a la fila, y reemplaza el valor del siguiente nodo, lo que perjudica el cambio al siguiente nodo de comparación.

Luego de haber completado la iteración se procede a cambiar la fila de comparación al siguiente nodo, esto hasta completar todas las filas existentes, si la fila es nula se salta dicho bucle para seguir con las siguientes iteraciones. Finalizado el proceso de comparación de la primera fila, se inserta los valores a otra fila, y se cambia el valor de la fila actual a la siguiente, al eliminar las filas que ya cumplen con un patrón se asegura de que los siguientes valores no se repitan, a su vez ayuda a que el sistema no haga iteraciones de más. Luego se vuelve a hacer todo el proceso anterior hasta completar el análisis de todas las filas, si la fila

actual no tiene valor se salta la iteración del primer bucle.

Una vez completados los bucles se procede a crear un objeto que tenga los métodos de una matriz, y se retorna dicho objeto, el cual contienen únicamente las matrices reducidas. Con las matrices reducidas se crea una lista con los mismos atributos que la lista principal pero ahora se guarda la matriz reducida en vez de la matriz original. La lista que almacena los datos reducidos también cuenta con una verificación de nombres repetidos por lo que si existen señales que se procesan y en memoria hay una señal con el mismo nombre se reemplaza el valor anterior por el nuevo. La lista con la información reducida es guardada en memoria al igual que los datos originales.

c. Escritura de archivos de salida

Esta opción contenida dentro del menú principal le pide al usuario que se le ingrese una ruta con el nombre del archivo que se desea guardar, si la extensión ingresada no es XML, se muestra un error en consola. Si la ruta es válida se crea un archivo en formato XML.

El archivo se genera por medio de las librerías `xml.etree.ElementTree` y `xml.dom`, la primera es para manipular archivos XML y la segunda es para agregar la indentación deseada en el archivo resultante.

El proceso de escritura es primero crear una etiqueta llamada “`senalesReducidas`” para contener los valores de las señales, para crear las etiquetas requeridas se hace uso de 3 bucles, el primero se hace cargo de iterar todas las señales, agregando la etiqueta “`senal`” que tiene como parámetros el nombre de la señal junto con su respectiva Amplitud, la segunda

iteración se hace cargo de crear la etiqueta “`grupo`” que tiene como valor el nombre del grupo asignado a la matriz reducida, la etiqueta “`tiempos`” la cual tiene como valor los tiempos que comparten un mismo patrón y la etiqueta “`dato`” que contendrá los valores de la matriz reducida, y el tercer ciclo se hace cargo de escribir los datos de la suma total de las tuplas, dentro de la etiqueta “`dato`”. El proceso se repite hasta terminar con las señales reducidas, y generando un archivo con los datos con indentación e información conforme a cada señal.

d. Mostrar datos del Estudiante

Esta opción tiene el propósito de ayudar al usuario a conocer a la persona que desarrolló el sistema, además de contar con información detallada del alumno.

e. Generar gráfica

En esta opción, se le pide al usuario que elija una de las señales de las cagadas en memoria, para poder generar su gráfica con Graphviz, una vez se defina la gráfica deseada se crea un objeto de tipo `Graph` que contiene los estilos que se le aplican a la gráfica.

La clase `Graph` contiene dos métodos los cuales son `create_original` y `create_reduced`, los cuales se hacen cargo de generar los grafos originales y si en dado caso existen datos ya procesados, se generan las imágenes de las señales reducidas.

Para la creación de imágenes se sigue la siguiente estructura (independientemente si se genera la gráfica de los datos iniciales o de los datos reducidos) Se crea un nodo (en el contexto de las gráficas los nodos son los elementos visuales que podemos ver con una forma en específico) principal que tiene como etiqueta el nombre de la señal, en el caso de ser una

señal reducida se le agrega un salto de línea y un texto que indique que es la señal reducida posteriormente se agrega un nodo de amplitud y tiempo (en el caso se las gráficas reducidas no se agrega este nodo), luego se agrega un subgráfico que contiene los datos de la matriz, dichos datos se muestran en la forma convencional de mostrar matrices, donde se puede identificar de forma visual sus filas y columnas. Para poder imprimir la matriz se usan dos ciclos uno para obtener las filas y el otro para obtener el valor contenido en cada columna. Los datos adicionales que contiene una gráfica reducida son el nombre de los grupos, y los tiempos contenidos en el grupo. Ambos métodos generan un archivo en formato “gv” y en formato “svg”, con la diferencia que la señal reducida agrega un “_reducida” en el nombre para diferenciar las gráficas.

El uso del formato “svg”, es para mejorar la visibilidad de los datos a la par que con este tipo de formatos se pueden llegar a visualizar mayor cantidad de información sin tener limitaciones, lo que si pasa al intentar graficar una imagen en formato png o jpeg.

e. Inicializar el sistema

Esta funcionalidad se encarga de limpiar las listas, y los cálculos realizados para poder volver a ejecutar el sistema, de forma que los datos anteriormente ingresados ya no existen dentro del contexto del programa.

Conclusiones

Con base en el análisis de los datos y la comprensión del software para la reducción de señales de audio, se pueden resaltar la eficacia y las posibles áreas de mejor de esta solución innovadora.

El uso de listas simplemente enlazadas para el procesamiento de señales de audio ha demostrado ser

una eficiente en términos de gestión de memoria, gracias a que permite una asignación dinámica de recursos, lo que resulta valioso en la aplicación. Además, resulta ser muy flexible el uso de las listas ya que se pueden agregar nodos con diferentes atributos.

A pesar de estos aspectos positivos, surgen algunas interrogantes que merecen ser consideradas. ¿Cómo podría optimizarse aún más el rendimiento del software en términos de velocidad de procesamiento? ¿Existen desafíos en la implementación de esta solución en sistemas con recursos limitados?

El uso del software de diseño de diagramas Graphviz permite la visualización de las distintas señales de audio tanto originales como las reducidas, de forma que es posible comprender de un modo más intuitivo el comportamiento de las señales de audio.

El software representa un avance prometedor en el campo del procesamiento de señales. Los resultados obtenidos indican su eficacia y potencial para una amplia gama de aplicaciones. Sin embargo, persisten desafíos y oportunidades para mejorar que invitan a futuras investigaciones y desarrollo.

Referencias bibliográficas

CC30A Algoritmos y Estructuras de Datos: *Tipos de datos abstractos*. (s.f.).
<https://users.dcc.uchile.cl/%7Ebebastos/apuntes/cc30a/TDA/>

L. Salcedo (2020). *Clases Abstractas – Ejemplos Paráticos con Python y ABC*. Pythondiario.
<https://pythondiario.com/2018/08/clases-abstractas-ejemplos-practicos.html>

Anexos

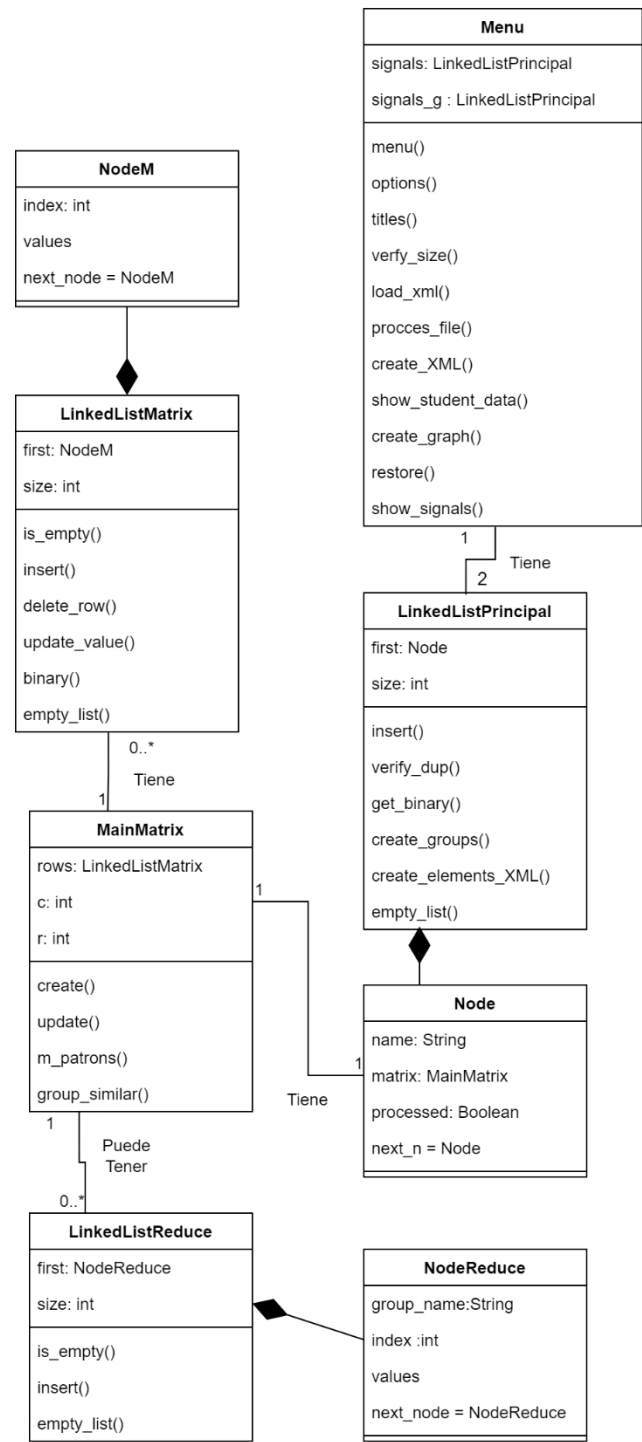


Figura 1. Diagrama de clases de las listas y del menu.

Fuente: elaboración propia

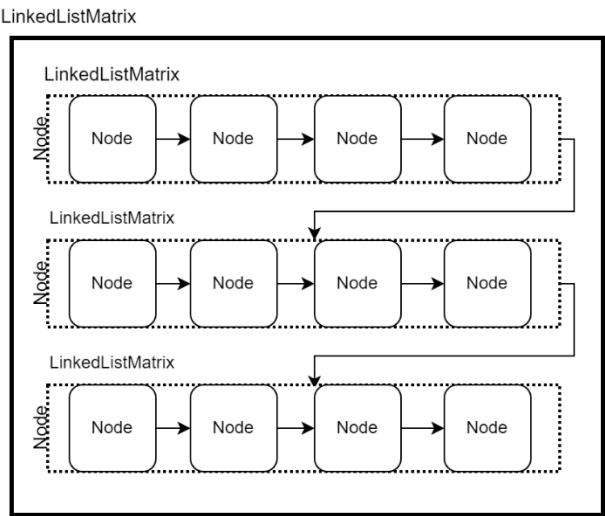


Figura 2. Representación gráfica de la Matriz.

Fuente: elaboración propia

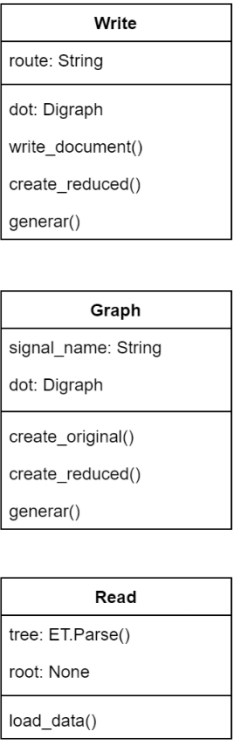


Figura 3. Clases para el manejo de Archivos.

Fuente: elaboración propia