Universidad Nacional Autónoma de México

Facultad de Ciencias

APLICACIÓN DE ALGORITMOS ENTRÓPICOS DE COMPRESIÓN DE DATOS PARA ENCONTRAR SIMILITUDES EN ARCHIVOS DE SONIDO

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
FÍSICO

PRESENTA:
PÁVEL VÁZQUEZ FACI

DIRECTOR DE TESIS:
DR. JESÚS ANTONIO DEL RÍO PORTILLA

2012

1. Datos del alumno
Vázquez
Faci
Pável
56 61 12 70
Universidad Nacional Autónoma de México
Facultad de Ciencias
Física
404037150

2. Datos del tutor
Dr
Jesús Antonio
del Río
Portilla

3. Datos del sinodal 1
Dr
Jorge Humberto
Arce
Rincón

4. Datos del sinodal 2
Dr
Jesús
Flores
Mijangos

5. Datos del sinodal 3
Dr
Sergio
Cuervas
García

6. Datos del sinodal 4
Dra
María Elena
Lárraga
Ramírez

7. Datos del trabajo escrito
Aplicación de algoritmos entrópicos de
compresión de datos para encontrar similitudes
en archivos de sonido
55 p
2012

Índice general

Índice de figuras

Índice de tablas

Agradecimientos

Agradezco enormemente al Dr. Jesús Antonio del Río Portilla quien ha sido una gran maestro y amigo, brindandome todo su apoyo en cada momento. Debo agradecer al M. En C. Helxine Fuentes Moreno por conceder los sonogramas, sin olvidar su constante aportación de conocimientos. Agradezco a Héctor Daniel Cortés Gonzáles por permitirme usar sus aplicaciones para realizar esta tesis. Además agradezco a mis sinodales el Dr. Jorge Humberto Arce Rincón, el Dr. Jesús Flores Mijangos, el Dr. Sergio Cuevas García y la Dra. María Elena Lárraga Ramírez por sus aportaciones y dedicación. Finalmente quisiera agradecer a Conacyt por otorgarme el estímulo económico para ayudantes de investigador SNI nivel III, con número 8547-7521.

Capítulo 1

Introducción

Este trabajo consiste en una propuesta muy sencilla. Plantea una solución computacional al problema de encontrar similitudes entre archivos de sonido. En particular, para identificar la similaridad de grabaciones provenientes de murciélagos.

Para la investigación sobre poblaciones con murciélagos proponer métodos que simplifiquen el ejercicio de identificar especies es de gran utilidad. Los investigadores de campo en esta área tienen la necesidad de realizar estimaciones en la diversidad, hábitos reproductivos, alimentación y otras características para una descripción completa.

Los biólogos han ideado maneras de realizar estos estudios, que implican la captura de murciélagos, por lo que perturban el medio. Un método que se ha vuelto muy popular por su simplicidad es el muestreo acústico, que estudia las vocalizaciones o también llamadas ecolocalizaciones, ya que los murciélagos usan el sonido como forma para localizarse en el espacio.

Similar al caso de las aves, los murciélagos poseen patrones de emisión de sonido que prácticamente pueden identificar a cada especie. Un inconveniente es que sus vocalizaciones son en frecuencias ultrasónicas, no perceptibles por el oído humano, por lo que se necesita de un equipo de grabación especializado. Uno de estos equipos comerciales es el *Anabat*©, que proporciona herramientas para crear horas y horas de grabaciones para su posterior analisis en una computadora. El investigador entonces tiene la necesidad de observar con un programa como son los sonidos previamente grabados, para así identificar que especie grabó durante la noche. Para hacer esto, se debe saber con trabajos realizados anteriormente como se caracteriza una vocacalización. Esta tarea necesita experiencia y también perícia, pues puede llevar

demasiado tiempo si no se está familiarizado con la misma.

Nosotros proponemos ahorrar tiempo de análisis para esta forma de identificar murciélagos y a la vez aplicar conceptos que se derivan de la teoría de la información propuesta por Claude E. Shannon, usando herramientas de la física estadística. Esto implica que el trabajo aquí presentado es un compuesto de varias disciplinas.

En este trabajo se explora la posibilidad de extender un método para medir la similitud en textos a archivos de sonido, utilizando la definición de entropía de Kolmogorov, que a su vez se usa en la ref. [24] para definir una distancia entrópica entre dos archivos de texto. El artículo explica como se puede utilizar un algoritmo de compresión como el *LZ77* para encontrar la entropía y posteriormente definir la distancia entrópica. Ejemplifica como se usa este concepto para reconocer lenguajes y proponer similaridades entre ellos. El Dr. Jesús Antonio del Río Portilla y el Lic. Hector Daniel Cortés propusieron una aplicación web [26] que implementa estos conceptos para minería de citas, por lo que resulto una gran herramienta para calcular las distancias entrópicas.

El planteamiento de extender todas estas nociones fue una idea a la cual desconociamos el resultado, consistió en que era posible identificar un archivo de sonido desconocido al extender la distancia entrópica para encontrar si/mi/la/ri/da/des. El acceso a las muestras de grabaciones de murciélagos, gracias al M. en C. Helxine Fuentes, nos dió la posibiladad de dar una aplicación real.

Con estos archivos de murcielagos y sus distancias entrópicas respectivas nos remitimos a otra forma de representar similaridades mediante los dendogramas, ya que es una práctica común usada por biólogos. Sin que demostraran que se pueda realizar identifiación. Por lo tanto, nos planteamos el objetivo de crear un método que pueda verificar nuestras suposiciones. La forma más sencilla de lograrlo fue hacer uso de la definición de distancia, mientras más cercanos sean dos archivos entonces más parecidos son. Propusimos formar bibliotecas con archivos ideales muy cercanos, que al compararlos con uno desconocido nos dijeran que tanta similaridad existía. Para lograrlo fue necesario aprender como tratar los datos computacionalmente para que el proceso fuera lo más automatizado posible. En el proceso notamos disparidades ante la falta de identifiación, por lo que fuimos adaptando los programas creados para mejorar los resultados y así finalmente dar una propuesta de lo que encontramos ser la manera más eficaz de lograr el objetivo.

Así, el presente trabajo se organiza en el primer capítulo donde se introduce una noción de conceptos necesarios que se desarrollan posteriormente. El siguiente capítulo consiste en el desarrollo del método, el cual se conforma en tener una primer sección en donde se explican los métodos preliminares y despues los concernientes a nuestra propuesta. Como fuimos adaptando el método entonces la relación con los resultados fue muy estrecha. El método y los resultados se deben comprender como un proceso retroalimentado pero con una secuencia temporal. Para terminar el trabajo se observa que la identifiación no es de la eficacia esperada, por lo que se propone modificar la estrategias y un plan alternativo.

Capítulo 2

Antecedentes

Para el entendimiento del trabajo aquí mostrado, se proporcionan en este capítulo los conocimientos y herramientas previos que se utilizaron durante el desarrollo del mismo.

Se propone dividir el capítulo en secciones que involucran los temás de mayor relevancia. Tratando de incluir otros asuntos que puedan ayudar a la comprensión de esta tesis. La estructura tiene la intensión de ser sucesiva comentando primero acerca de los murciélagos y su ecolocalización, seguido de como se graba el sonido y como se puede manipular. Dar una mirada a las formas de codicficación del audio y la compresión de información mediante *ZIP*. Finalmente explica el funcionamiento de los métodos que usamos para manipular y representar datos.

2.1. Murciélagos y ecolocalización

Daddo que tuvimos acceso a grabaciones de sonido de murciélagos y trabajamos con estos archivos, provenientes de la zona de La Venta en Oaxaca [1]. Nos proveen muestras para realizar experimentos de acuerdo al objetivo descrito, por lo que esta primer sección se da una introducción elemental a los murciélagos.

En el capítulo donde se muestran los resultados utilizamos nombres científicos de las especies empleadas. Es importante explicar un poco en que consiste esta clasificación que se usa en biología desde 1758 inventada por Linneo, que llaman nomenglatura binomial, en el cual el nombre científico se forma con dos vocablos latinos, el primero es el *género* y el segundo la *especie*. Así los

géneros se agrupan en *familias*, las familias en *órdenes*, luego *clases*, *phyla* y para abarcar a todos con los *reinos*. [2] Los murciélagos al ser tan diversos se les divide en dos órdenes, los del viejo mundo llamados *Megachiroptera* y los del nuevo (todo menos Europa, las islas oceánicas y los polos) como *Microchiroptera*. En los micropteros hay 759 especies, de los cuales en México hay 137, siendo 91 insectívoros [4].

Utilizamos especies de murciélagos cuyos nombres reducidos son:

Dicalb= Diclidurus albus
Cynmex= Cynomops mexicanus
Eumops= Eumops sp.
Eptfur= Eptesicus furinalis
Lasega= Lasiurus ega
Lasint= Lasiurus intermedius
Molruf= Molossus rufus
Molmol= Molossus molossus
Mormeg= Mormoops megalophylla
Permac= Peropteryx macrotis
Ptedav= Pteronotus davyi
Pteper= Pteronotus personatus

A diferencia de otros mamíferos que sólo planean, los murciélagos son los únicos capaces de realmente volar. También se diferencian por tener la capacidad de ubicarse en el espacio y encontrar su alimento mediante el uso de la ecolocalización, que explicaremos más adelante.

Los murciélagos poseen caulidades morfológicas únicas, pues parecen roedores con alas. Tienen un pelaje que suele ser de color pardo, gris, amarillo, rojo o negro. Su tamaño es muy variado, ya que pueden medir desde los 25 milímetros hasta los 2 metros, a lo que curiosamente se les llama zorros voladores. Algunas especies pueden vivir hasta 30 años.

Su cabeza es diferente entre una especie y otra, ya que cada especie es fácilmente identificable con solo observar las características del rostro. Suelen tener en común que su nariz es prominente y puede parecer una lámina, la cual funciona como potenciador de las ondas sonoras que emiten a partir de contracciones en la laringe. También sus orejas suelen ser de gran tamaño y con surcos que aumentan su capacidad de ecolocalización.

Su alimentación es muy diversa, teniendo características morfológicas especializadas para cada alimento.

Los hay quienes comen frutos (frugívoros), los cuales son de gran importancia en la selvas húmedas y en la reforestación de bosques, ya que son capaces de dispersar semillas al alimentarse del fruto y soltar los restos consumidos. También existen murciélagos que se alimentan de néctar y polen (nectarívoros), siendo causantes de la polinización de plantas.

Los que se alimentan de insectos (insectívoros) son los más importantes en ecolocalización, pueden diferenciarse en dos grupos, los que lo hacen al vuelo y los que seleccionan del suelo o de la vegetación. Los primeros son los más comunes en zonas templadas, siendo los que se pueden ver al atardecer volando en los alrededores y lanzandose en picada. Estos murciélagos poseen muy desarrollado el sistema de ecolocalización, ya que buscan pequeños insectos volando entre múltiples obstáculos, una vez identificada la presa cambian la manera de emitir sus sonidos y su velocidad de vuelo se incrementa enormemente. Los murciélagos que recolectan insectos inmóviles tienen características distintas, ya que vuelan bajo y lento, con alas grandes y de gran manibrabilidad.

Los muriciélagos que se alimentan de carne (carnívoros), como aves, lagartigas o incluso otros murciélagos, suelen ser de gran tamaño, con alas anchas y dientes mortíferos. También existen especies que se alimentan de peces (pscívoros), los cuales cazan peces que se encuentran cerca de la superficie descendiendo en picada. En cuanto a ecolocalización son muy especiales al ser aptos de detectar pequeñas ondulaciones en el agua.

Finalmente se encuetran los que se alimentan de sangre (hematófagos), aunque sólo se conocen tres especies, los llamados vampiros suelen ser muy famosos. Se distinguen en su habilidad de desplazarse por el suelo, poseen colmillos capaces de penetrar la piel de animales y una saliva que evita la coagulación. [3]

2.1.1. La ecolocalización

Identificar la especie de un murciélago es una labor complicada, pues sus hábitos nocturnos y su vuelo a gran velocidad provocan que no se pueda utilizar métodos visuales, como por ejemplo en las aves. Una de las formas más comunes es capturandolos usando redes de niebla o las trampas arpa, la cuales consisten en trampas que se colocan en zonas donde se espera encontrar poblaciones de murciélagos y así al pasar cerca quedan atrapados. Esta labor suele ser tediosa y menos efectiva [4], ya que implica montar un dispositivo aparatoso y perturbar al murciélago al interrumpir su actividad. La cantidad

de especímenes colectados tampoco suele ser muy grande.

Las vocalizaciones de murciélagos, es otra forma muy útil para su identificación. El método es muy similar al usado en la ornitología (estudio de aves) en donde se ecucha el canto para identificar su especie.

Los murciélagos no cantan como las aves, pero utilizan el sonido para localizarse en el espacio. Estos sonidos los emiten generalemente entre los 14 y 100 kHz, por lo que se encuentran en rangos ultrasónicos. El ser humano únicamente puede escuchar hasta los 20 kHz. El sonido emitido en alta frecuencia viaja en una dirección y al encontrar una superficie rebota. Como el sonido viaja a una velocidad conocida (340 metros por segundo) el murciélago mide instintivamente el tiempo transcurrido entre la emisión y la recepción del sonido, así le es posible conocer la distancia con un objeto [4] [5]. En la figura 2.1.1 se muestra un ejemplo del funcionamiento de la ecolocalización.

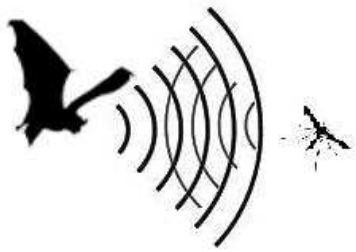

Figura 2.1: Ejemplo de un murciélago insectívoro localizando una presa

De esta forma los murciélagos pueden ubicarse y alimentarse, pero es importante mencionar que los llamados de murciélagos insectívoros, que son los únicos utilizados por este trabajo, poseen tres fase cruciales, la de detectar, identificar y localizar a la presa. A cada llamado se le define como un pulso vocal continuo que se separa de otros por un silencio. Cada llamado posee tres partes, un inicio, una parte media llamada cuerpo y un final [4]. Al grabar un sonido mediante un sistema de detección ultrasónico, que graba los sonidos emitidos por los murciélago (seccion 2.2) se observa que cada especie tiene llamados distintos para cada fase y en rangos de frecuencias diferentes, figura 2.2. Por lo que es posible identificar una especie de muriciélago al conocer previamente sus llamados típicos.

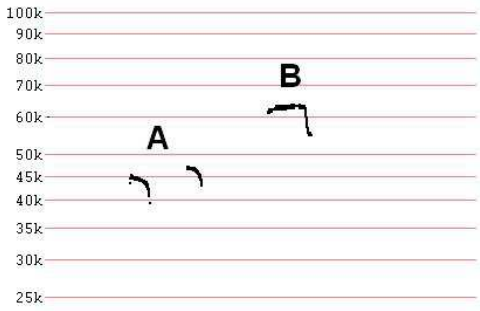

Figura 2.2: A:*Pteronotus parnellii*, B:*Molossus molossus*. En el eje X se muestra el tiempo, mientras que en el eje Y la frecuencia en kHz

Este hecho es muy importante ya que el método, de observar sonogramas y ver la forma de sus llamados, es el método más usual utilizado en la identificación de murciélagos.

A lo largo de esta tesis se espera que se pueda mostrar la utilidad de crear otro método de identificar especies de murciélagos a partir únicamente de grabaciones de sus llamados, dicha grabación se lleva a cabo con el uso de los llamados detectores de murciélagos como por ejemplo el Anabat, que describiremos a continuación.

2.2. Anabat

Como se menciona en la sección anterior los investigadores de campo que necesitan estudiar poblaciones de muerciélagos, se han enfrentado al problema de tener de utilizar formas de captura que suelen ser caros, pesados y lentos, por ello se han utilizado formas menos invasivas, prácticas y baratas. Uno de los sistemas que se usan con regularidad para realizar las grabaciones y así análizar las ecolocalizaciones de murciélagos en el campo, es el sistema Anabat.

El llamado muestreo acústico consiste en colocar un aparato, en nuestro caso el Anabat SD1©(Titley Electronics), el cual graba tanto de manera activa como pasiva ciertos espectros de frecuencias (de 4 kHz a 200 kHz), visualizándose en un espectrograma similar a la figura 2.2.

La forma de grabación pasiva consiste en dejar el aparato con su micrófono en zonas estratégicas como rutas de vuelo, entre vegetación, etc, simplemente donde se haya observado que pasen murciélagos de acuerdo a sus hábitos [1]. Los tiempos de grabación son atomatizados, se activa exclusivamente cuando detecta un ruido en la frecuencias seleccionadas, por lo que se pueden grabar desde segundos hasta minutos; sin embargo como un murciélago insectívoro suele pasar muy rapidamente se graban sólo segundos.

Una vez obtenidas las grabaciones viene la parte laboriosa, concretamente es la que esperamos evitar con este trabajo, que consta de observar cada espectrograma y compararlo con una biblioteca de espectrogramas previamente creada. Obviamente los investigadores que han realizado múltiples análisis con éste método poseen una gran destreza y no necesitan las comparaciones pues conocen cada vocalización de cada especie, pero a pesar de ello deben sentarse frente a un monitor de computadora para observar durante largos periodos las distintas grabaciones hechas durante el muestreo acústico.

En el capítulo dedicado al método se muestran algunas de las propiedades que nos interesan del software *AnalookW*©version 3.8b [6], que se usa para visualizar las grabaciones del Anabat como se ejemplifica en la figura 2.3

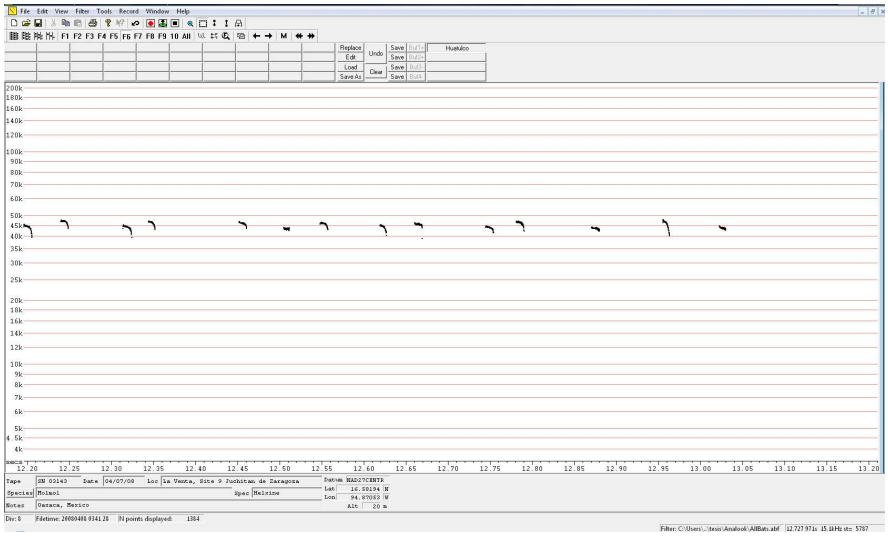

Figura 2.3: Captura de pantalla del software *AnalookW*©version 3.8b con llamados de la especie *Molossus molossus*

El software fue creado por Chris Corben que simplifica la manipulación de las grabaciones creadas por Anabat, para que sencillo visualizar los sonidos usa el dominio del tiempo y frecuencia en el codominio.

Es importante mostrar que en la sección llamada "Filter" es posible insertar un filtro para seleccionar frecuencias que sean exclusivas de los murciélagos. Estos filtros pueden ser creados por el usuario según su conveniencia y conociendo previamente que especies espera seleccionar.

En "Tools" existe la posibilidad de exportar el archivo guardado en la codificación de Anabat, observamos que la información proveniente de un microfono es procesada mediante su propio método, patentado y confidencial. Desconocemos como es dicho funcionamiento y no nos es posible realizar modificaciones a la forma en que se almacenan los datos. Sin embargo, como vemos en la figura 2.4, el software *AnalookW*©nos permite exportar a formato *wav* el sonido grabado. También proporciona herramientas para modificar ciertas características del audio, como la razón de división de la frecuencia, que nos cambia la presición el que se mide la frecuencia. Permite expandir o comprimir el tiempo, cambiar la forma sinusoidal a cuadrada, así como escoger los valores máximos y mínimos en la frecuencia.

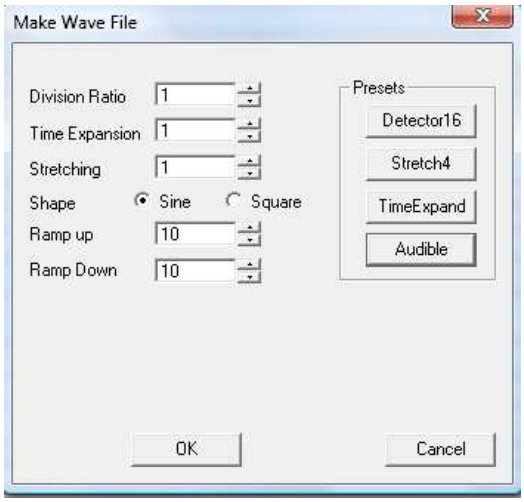

Figura 2.4: Ventana del sistema Anabat para exportar audio

2.3. Formatos de sonido

El *AnalookW*©exporta a formato *WAV* las grabaciones de sonido provenientes del Anabat, que no son audibles mientras que el *WAV* si lo es. En esta parte presentamos una reseña de las características más importantse de tres distintos formatos de sonido que utilizamos.

El sonido es una vibración de un medio físico. En este caso es el movimiento de la laringe de un murciélago que produce una vibración que se propaga por el aire y que finalmente es registrado por un microfono.

Las ondas poseen propiedades para describirla como la frecuencia (f), que mide las repeticiones por unidad de tiempo y se suele expresar en Hertz ($Hz = 1/segundo$). La frecuencia se relaciona con el periodo, que es la cantidad de tiempo que transcurre para repetirse un ciclo de la onda ($T = 1/f$), se suele medir en segundos. A la distancia vertical entre un punto máximo (cresta) y el cero se le llama amplitud. La amplitud puede ser variable a lo largo del tiempo [8].

Un ejemplo sencillo es una onda sinusoidal como se muestra en la figura 2.5

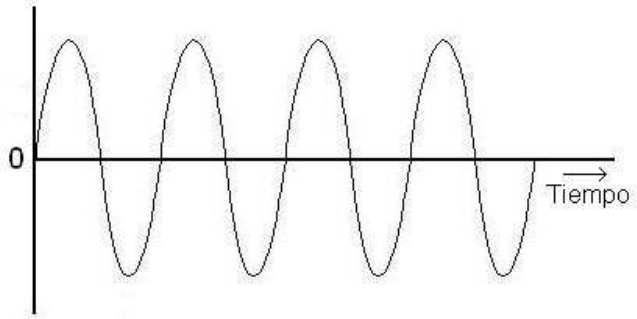

Figura 2.5: Onda sinusoidal en el tiempo.

Es posible almacenar la información de una onda de manera analogica,

es decir continua y sin perder información. Sin embargo los sistemas informáticos solamente pueden guardar valores discretos. Por esto almacenar el sonido en formatos digitales implica discretizar la onda tomando valores instantaneos de la amplitud cada cierto tiempo. A este proceso se le llama *muestreo* o en inglés *sampling* [9]

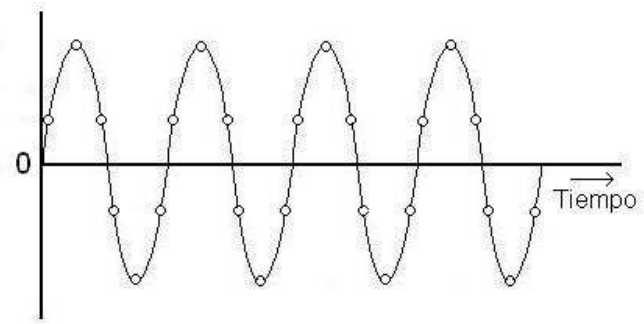

Figura 2.6: Ejemplo de discretización de una onda sinusoidal

En el ejemplo de la figura 2.6 los puntos indican donde se ha almacenado un valor de la amplitud, espaciado en el tiempo regularmente. Mientras más cantidad de puntos se creen para un archivo de sonido perdera menos información en el proceso y a conseciencia tendra una mejor calidad.

Este proceso lo realiza un programa computacional llamado *codec* (coder-decoder), el cual se encarga de almacenar la señal codificándola y luego reproducirla decodificándola [10]. El término de *formato* especifica un único tipo de datos codificado por un *codec*, aunque existen formatos que codifican de múltiples formas no entraremos en detalles.

En la sección 2.4 explicaremos con mayor detalle en que consiste la compresión de los datos, pero consideramos importante listar tres de los métodos que usaremos:

- *Uncompressed*(sin compresión): *WAV*, no posee ningún método de compresión.

- *Lossless*(sin pérdida): *LZ77*, utiliza métodos estadísticos y diccionarios.
- *Lossy*(con pérdida): *MP3, OGG*, elimina información no necesaria y usa métodos de predicción.

Existen gran variedad de formatos [11], explicaremos a continuación a grandes rasgos tres de los más importantes.

2.3.1. *WAV*

También llamado *WAVE*, fue inicialmente desarrollado para los sistemas operativos Windows, aunque actualmente es compatible en muchos otros sistemas y casi cualquier programa de música es capaz de reproducir o guardar [12]. Consiste en un algoritmo sencillo sin compresión, es decir, es una representación fiel de la onda sonora original. Por lo que el dominio de la función es el tiempo.

Al ser un análogo a la forma de discretización regular de una onda, su representación es similar a la presentada en la figura 2.6. A cada valor de la amplitud en el tiempo, les asigna series de símbolos en un código binario [13].

Se almacena mediante la especificación *RIFF*, el cual crea fragmentos (*chunks*) con pedazos de toda la información discretizada de la onda[14].

2.3.2. *MP3*

Es la teccer version del formato llamado *MPEG Audio Layer*, ambos patentados y sin acceso libre a su forma de codificación. Tiene la propiedad de usar algoritmos similares a filtros[17] para reducir en gran medida la longitud en *bits* de sus archivos al eliminar las frecuencias que no se pueden percibir por el ser humano, quedándose únicamente con rangos de entre 2 kHz a 5 kHz. También usa una forma de eliminar sonidos "enmascarados", ya que si se tiene un sonido muy fuerte en cierta frecuencia y otro al mismo tiempo en otra frecuencia, pero debil, entonces el oido humano no escucha el debil [15] y no es necesario alamcenar ambos sonidos.

Se sabe que mediante el uso de un modelo *psicoacústico* analiza la señal de audio para separar y codificar la señal al dominio de frecuencias y así crear un formato independiente del tiempo [16].

2.3.3. *OGG*

Este formato es similar al *MP3* usando el método con pérdida, pero utiliza el *codec* de uso libre llamado *Vorbis* [18]. Sin embargo no elimina rangos de frecuencias que no sean audibles por el hombre, utiliza un modelo de probabilidad de Huffman [19].

Al igual que *MP3* codifica la onda sonora a un dominio lineal de la frecuencia y a una imagen logarítmica de la amplitud [20].

Un archivo en *wav* lo podemos convertir a otros formatos con el uso del software GNU SoundConverter 1.5.4[36]. El siguiente comando nos codifica para *mp3*:

$ soundconverter -b -m audio/mpeg -s .mp3 *.wav

Mientra que para *ogg*;

$ soundconverter -b -m audio/x-vorbis -s .ogg *.wav

De esta manera se covierte un archivo generado con Anabat en *wav* a *mp3* o a *ogg*, el cual usa la codificacion *MPEG* y *Vorbis* para cada uno.

2.4. El algoritmo LZ77

Como se mencionó anteriormente el método de compresión *Lempel-Ziv 77* es un algoritmo convencional y muy famoso (*WinZip*, *GZip* de compresión sin perdida o *lossless data compression*.

Los métodos *lossless*, son métodos que usan diccionarios como una lista de frases. Se basan en el hecho de que un diccionario estático (que no cambia) posee frases de cierto tamaño que se repiten frecuentemente [21]. La idea es cambiar partes del archivo original y remplazarlos por referencias (más pequeñas) a un diccionario que contiene estas frases [22]

Los diccionarios pueden ser de varias formas, por ejemplo *ASCII* o incluso binarios. Pero los algoritmos *Lempel-Ziv* construyen el diccionario a partir del mismo archivo, por lo que no necesitan un diccionario explícito, se puede decir que se adaptan.

La mejor forma de entender el proceso por el cual se comprimen los datos es con un ejemplo. Consideremos una secuencia con cuatro posibles caracteres *a,b, c y d* como en la figura 2.7 El algoritmo lee de izquierda a derecha letra por letra. Supongamos que ya hemos iniciado el proceso y que ya ha leido las primeras nueve. La cadena leída entonces es $E = aabbdadbb$, mientras que la

Figura 2.7: Ejemplo de una secuencia siendo codificada por el algoritmo *LZ77*

cadena todavía por codificar es $S = abcbdacab$. Lo que hace el algoritmo es buscar la cadena más larga en E que se iguale a una de S, que en nuestro caso es *bda* en la tercer posición en S.

Para indicar donde se encuentra la repetición el algoritmo usa un codigo de tres palabras donde se muestra la posición, longitud y el primer símbolo fuera de la repetición. En el ejemplo sería $33c$ más las cadena repetida en cuestión *bda* [22]. Así, si la repetición es más larga entonces tendremos una compresión mayor. Lo mismo ocurriría si la cadena *bda* se repite en múltiples partes del texto.

El algoritmo *LZ77* posee un principio llamado *sliding window* como en la figura 2.8

La ventan se divide en el búfer de busqueda que funciona como diccionario y que tiene los datos ya leídos, el búfer de busqueda hacia adelante posee los datos que se van a codificar. Ambos son de un tamaño predifinido.

En *unix* es muy sencillo realizar esta operación con el siguiente comando:

$ zip test *

El cual crea el archivo test.zip conteniendo todos los archivos del directorio en el que se encuentre.

En la siguiente sección daremos seguimiento a la forma de tratar comparaciones entre archivos, esto nos servirá como herrmienta para comprender en el capítulo referente al método, como es que se maneja el concepto de distancia informática y como se representa visualmente.

Figura 2.8: Principio *sliding window*

2.5. Matrices de similaridad y dendogramas

En esta sección se da una breve presentación de lo que es una matriz de similaridad y un dendograma, que se usaran con regularidad en los próximos capítulos. Ambos se utilizan en el análisis de agrupamientos (*clustering* en inglés) el cual consiste en agrupar individuos cualquiera, que posean alguna medida de distancia.

Dependiendo del contexto una matriz de similaridad también se le puede llamar una matriz de distancias o contrariamente de disimilaridad, que finalmente proporcionan la misma información. Únicamente se diferencian en que una muestra que tan "cercanos" son los individuos y la otra que tan "lejanos". Ambas son matrices de tamaño nxn simétricas, con la primera teniendo su diagonal de ceros y la segunda de unos.

Como se menciona en la ref. [7] la matriz de similaridad proporciona cercania entre dos individuos que se comparen, un valor pequeño muestra que se encuentran proximos, mientras que un valor grande estan apartados. El cero indica que son lo más parecido posible. Por lo tanto, es crucial que los dos individuos tengan una medida de distancia entre ellos $\Delta_{AB} \geq 0$ que cumpla con la desigualdad del triángulo

$$\Delta_{AB} + \Delta_{AC} \geq \Delta_{BC} \tag{2.1}$$

Con $\Delta_{ii} = 0$ si se trata del mismo individuo comparado consigo mismo.
También se debe cumplir que $\Delta_{ij} = \Delta_{ji}$
Un ejemplo muy conocido es la *distancia Euclidiana*

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

Donde x y y son vectores eucledianos a los cuales se les puede asociar una distancia física, aunque no siempre es el caso, como veremos para la *distancia entrópica*.

Una matriz de similaridad se propone como en el siguiente ejemplo.

$$D = \begin{pmatrix} 0 & 1 & 2 & 0{,}5 \\ 1 & 0 & 1 & 0{,}8 \\ 2 & 1 & 0 & 1 \\ 0{,}5 & 0{,}8 & 1 & 0 \end{pmatrix} \tag{2.3}$$

Esta matriz nos indica que tan similares son 4 individuos. La distancia entre un mismo individuo es cero ($D[1,1] = 0, D[2,2] = 0$,etc) por ello la diagonal es sólo ceros. Mientras que las distancias entre distintos es diferente de cero, por ejemplo $D[2,3] = 1$ y también $D[3,2] = 1$.

Podemos notar con ver la matriz que los archivos más cercanos son el 1 y el 4, ya que $D[1,4] = 0{,}5$. Mientras que los más lejanos son el 1 y el 3, pues $D[1,3] = 2$. El resultado es una manera de mostrar como son las distancias entre individuos y así decir quien es cercano a quien. Uno de los problemas que esto implica es que al tener grandes cantidades de individuos para comparar resulta difícil obtener información unicamente "observando" la matriz de similaridad. En la seccion 3.3 proponemos un algoritmo sencillo para crear listas que muestren que archivo es similar a otro de acuerdo a definir una distancia mínima para considerarlos cercanos.

Por el momento explicaremos otra forma visual de representar esa cercania mediante el uso de clasificaciones jerárquicas aglomerativas, que producen el llamado dendrograma o árbol filogenético. También es posible hacerlo mediante métodos divisivos, pero nos limitaremos a las primeras.

Los dendogramas se forman a partir de la matriz de similiraridad, los dendogramas crean las jerarquías para clasificar formando grupos a partir de las distancias entre individuos. Existen tres formas populares de formar estos grupos, la primera es la de *single linkage* o del vecino más cercano, que

utiliza la distancia más pequeña entre dos individuos para formar el grupo. El siguiente ejemplo ilustra como se realiza el método.

Supongamos que tenemos una matriz 4x4 que al ser simétrica hemos eliminado la parte superior de la matriz, simplemente para que sea más sencillo de entender.

$$D_1 = \begin{pmatrix} 0 & & & \\ 0{,}5 & 0 & & \\ 2 & 1 & 0 & \\ 1 & 0{,}8 & 3 & 0 \end{pmatrix} \tag{2.4}$$

La distancia más pequeña distinta de cero se encuentra entre los individuos 1 y 2, por lo estos forman el primer grupo (12). Seguido, se calculan las distancias del grupo (12) con el resto.

$$d_{(12)3} = min[d_{13}, d_{23}] = d_{23} = 1 d_{(12)4} = min[d_{14}, d_{24}] = d_{24} = 0{,}8 \tag{2.5}$$

De forma que se crea otra matriz 3x3 con las nuevas distancias como sigue

$$D_2 = \begin{pmatrix} 0 & & \\ 1 & 0 & \\ 0{,}8 & 1 & 0 \end{pmatrix} \tag{2.6}$$

La siguiente distancia menor es (12) con 4 que forman un nuevo grupo (124). Se vuelven a calcular las distancias de (124) con el que queda

$$d_{(124)3} = min[d_{13}] = 0{,}8 \tag{2.7}$$

Entonces el último grupo es (1234).

Este ejemplo fácil, muestra que la primer separación se da con (12), la segunda con (124) y finalmente tenemos todo el grupo (1234). En la figura 2.5 se muestra el dendograma resultante. El segundo método es llamado *complete linkage* o vecino más lejano, se diferencía de no tomar la distacia mínima sino la máxima. Finalmente el tercero consiste en combinar los elementos en promedio más cercanos y formar un nuevo grupo, así se compara con otro midiendo el menor de las distancias medias. Se le llama *average linkage* o *UPGMA*.

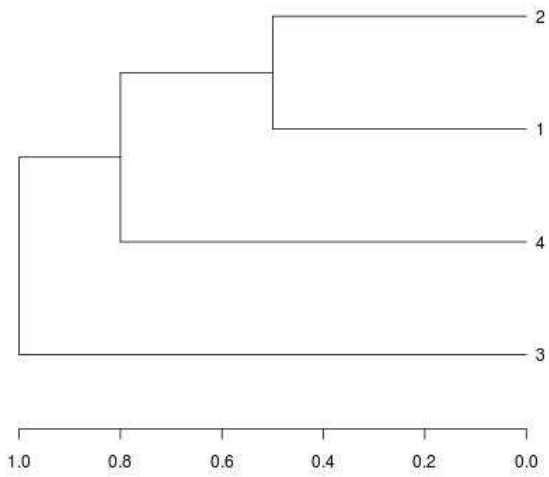

Figura 2.9: Ejemplo de dendograma usando el vecino más cercano. El eje X es distancia, con el valor mínimo 0.8

Capítulo 3

Método para encontrar similitudes en archivos de sonido

Debido a que el M. en C. Helxine Fuentes[1] realizó durante su tesis gran cantidad de grabaciones de murciélagos con el sistema Anabat, decidimos utilizar dichas grabaciones para desarrollar el método que presentamos a continuación.

Describimos la distancia entrópica dentro de la Teoría de la Información y como se calcula de acuerdo a la propuesta de la ref. [24]. Después presentamos como creamos el método detallado por pasos. Comprendido como un proceso para crear un criterio y un procedimiento para poder demostrar la validez del uso de la distancia entrópica como una cantidad de comparación entre archivos de sonido.

Con los archivos en formato de Anabat, el primer paso consistió en exportar dicho audio a *WAV*, para luego codificarlo a los otros formatos. Seleccionamos archivos que pudiesemos considerar patrón y seguidamente conseguir la identificación de un sonograma desconocido.

3.1. Distancias entrópicas

De acuerdo a la Teoría de la Información podemos calcular el contenido informático de un sistema no físico usando la entropía como una medida de incertidumbre [23]. Estos sistemas no físicos pueden ser representados

mediante cadenas de caracteres binarios a los que se les puede calcular su entropía mediante la definición de Kolmogorov.

Esta definición dice que la *Entropía Algorítmica* de una cadena de caracteres es la longitud (en *bits*) del programa más pequeño que pueda reproducir esa cadena de carácteres [25].

Por ejemplo, supongamos las siguientes cadenas de cáracteres binarios:

01010101010101010101

y

11010111010100100010

La primer cadena se puede describir como 10 veces "01" repetidos. Mientras que la segunda, al ser aleatoria, no se puede describir de la misma forma. La entropía de la primer cadena es entonces menor que la entropía de la segunda.

Como vimos el algoritmo *LZ77* está concebido precisamente para crear el programa (archivo *.zip*) más pequeño que reproduce integramente la cadena original. Por lo que a un archivo que se le comprima con *ZIP* se reducirá su longitud original dependiendo de que tan repetitivo sea, o expresado de otra forma que tan entrópico es.

El problema es que esta medida de entropía es relativa. No existe una cantidad absoluta de entropía en un archivo, sino que más bien sirve como una comparación entre dos archivos.

A esta comparación se le pueden asignar las propiedades de distancia indicadas en la sección 2.5. Por lo que en la ref. [24] proponen calcular una distancia entrópica entre pares como sigue:

$$S_{AB} = (\Delta_{AB} - \Delta_{BB})/\Delta_{BB} + (\Delta_{BA} - \Delta_{AA})/\Delta_{AA}. \tag{3.1}$$

Con:

$$\Delta_{BA} = L_{B+A} - L_B, \tag{3.2}$$

donde L es el tamaño en *bits* del archivo en cuestión, comprimido usando la función *zip*.

Esta distancia nos indica que tan parecidos o diferentes, de acuerdo al contexto, son dos archivos cualesquiera.

Con el archivo en *zip* es posible calcular estas distancias entrópicas utilizando el programa Relative Entropy[31], programado por Héctor D. Cortés en *perl*, el cual posee una aplicación web:

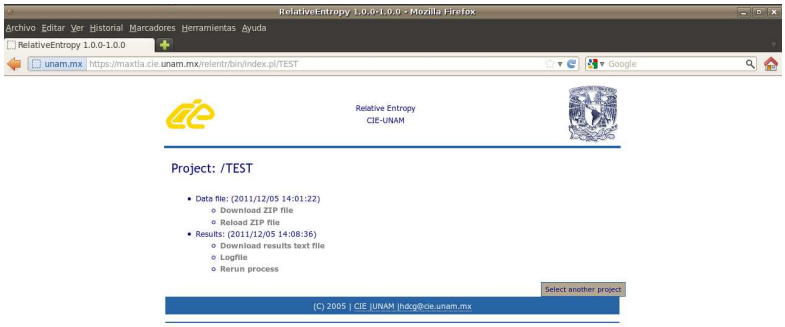

Figura 3.1: Aplicación en web del programa Relantive Entropy

El programa permite la introdcción de archivos en *zip* y calcular todas las distancias entrópicas relativas entre ellos, las cuales presenta en forma de lista de texto, por ejemplo:

FileNumber	FileName
000000	cynmex07.ogg
000001	cynmex24.ogg
000002	cynmex56.ogg
000003	lasega10.ogg
000004	lasega18.ogg

A	B	Entropy(AB)	Entropy(BA)	EntropicDistance(AB)
000000	000001	0.0000	0.0005	0.000557
000000	000002	0.0002	0.0005	0.000775
000000	000003	-0.0004	0.0012	0.001588
000000	000004	0.0001	0.0026	0.002643
....				
000003	000004	-0.0000	0.0003	0.000297

La primer sección del ejemplo muestra la asignación de un número identificador para cada archivo con su nombre respectivo. Así, en la siguiente sección se presenta en las dos primeras columnas el número de los archivos A y B que se comparan. Las dos columnas posteriores indican la entropía entre AB y entre BA, ec. (3.2). Para finalmente en la última columna mostrar la distancia entrópica, ec. (3.1), siendo ésta la que nos interesa para los cálculos

posteriores.

Una vez obtenidas todas las distancias entrópicas nos fue posible formar matricies de similaridad, en donde se muestra en una matriz simétrica cuales son las distancias entre archivos, siendo cero para un archivo consigo mismo y cumple con las propiedades decritas en la sección 2.5.

En la tabla 3.1 presentamos un ejemplo.

	Archivo1	Archivo2	Archivo3	Archivo4	
Archivo1	0	0.001016	0.000061	0.000279	....
Archivo2	0.001016	0	0.000388	0.000451	
Archivo3	0.000061	0.000388	0	0.000139	
Archivo4	0.000279	0.000451	0.000139	0	
	....				

Tabla 3.1: Ejemplo de una matriz de similaridad para archivos con distancias entrópicas

Para transformar la lista del software Relative Entropy en una matriz de similaridad, establecimos un algoritmo que transforma y reacomoda los datos. Para ésta tarea utilizamos el intérprete de comandos para Unix llamado KSH o Korn Shell[32], el cual usamos para crear un programa o script. El número de datos total ingresados se calcula fácilmente contando el número de líneas ($count$) y usando la parte entera de la siguiente fórmula $n = (1 + sqrt(1 + 8 * count))/2 - 2$, donde n es el número buscado de datos ingresados.

El siguiente comando corre el scirpt creado y almacena la salida en el archivo de texto ejemplo.txt

$ matrizR1B.sh > ejemplo.txt

Un resumen del algoritmo para formar la matriz de similaridad se puede mostrar con el siguiente ejemplo.

Supongamos que hemos calculado todas las distancias entre cuatro archivos de audio, el número de combinaciones que podemos formar es

$$C(n,k) = \frac{n!}{k!(n-k)!} \tag{3.3}$$

Entonces, $C(4,2) = \frac{4!}{2!(4-2)!} = 6$. Obtendremos una lista de distancias entrópicas con seis elementos, por ejemplo:

A	B	EntropicDistance(AB)
000001	000002	0.000557
000001	000003	0.000775
000001	000004	0.001588
000002	000003	0.002643
000002	000004	0.002643
000003	000004	0.000297

Como $n = 4$ la matriz que formaremos es de tamañano 4x4, el algoritmo toma como un vector la lista de la columna de "EntropicDistance(AB)".

Inicia escribiendo $i = n-3 = 1$ ceros, seguido de sus primeros $j = n-1 = 3$ elementos como renglón divididos por un espacio.

0.000000 0.000557 0.000775 0.001588

Como siguiente paso escribe otro renglón con $i = n - 2 = 2$ ceros, luego a partir de la posición $k = n = 4$ los siguientes $j = n - 2 = 2$ valores.

0.000000 0.000000 0.002643 0.002643

El siguiente renglón se forma escbribiendo $i = n - 1 = 5$ ceros con el siguiente $j = n - 3 = 1$ elementos a partir de $k = n + 1 = 5$

0.000000 0.000000 0.000000 0.000297

Finalmente se escriben $i = n = 4$ ceros, $j = n - 4 = 0$ elementos y $k = n + 2 = 6$

0.000000 0.000000 0.000000 0.000000

La salida entonces queda como sigue

0.000000 0.000557 0.000775 0.001588
0.000000 0.000000 0.002643 0.002643
0.000000 0.000000 0.000000 0.000297
0.000000 0.000000 0.000000 0.000000

La cual si la sumamos con su transpuesta nos proporciona una matriz de similaridad. Podemos notar que los enteros auxiliares *i, j, k* toman los siguientes valores, $i = 1..n$, $j = n - 1 \ldots 0$, $k = 2 \ldots 4..C(n, k)$ si n es par y $k = 1 \ldots 3..C(n, k)$ si n es impar.

La salida de este algoritmo es en formato de texto simple. La matriz formada ahora se puede utilizar con facilidad para análisis posteriores como los que se presentan en las secciones siguientes.

Con esta explicación ya sabemos como definir distancias, ahora es necesario escoger el formato de sonido que permita un mejor desempeño de la metodología.

3.2. Selección del formato

Analizamos archivos de sonido *controlados* y determinamos sus distancias con el método propuesto para obtener distancias entrópicas.

Estos archivos controlados los realizamos a través de distintas grabaciones a partir de un micrófono convencional conectado a una PC. Los grabamos originalmente en *WAV* y posteriormente los codificamos en *MP3* y *OGG*). En las grabaciones participaron seis voluntarios humanos, leyendo cada uno los siguientes textos:

- Dos textos distintos con tamaño de un párrafo en español.
- Una repetición de los textos anteriores leídos por las mismas dos personas.
- Una copia del archivo de uno de los textos en español.
- Un audio de radiodifusión en inglés.
- Un audio musical.

En total resultaron sesenta archivos de audio.

Los resultados de calcular las distancias entrópicas entre todos estos ejemplos lo analizamos mediante dendogramas, que muestran visualmente el grado de similitud entre dos sujetos cualquiera, en nuestro caso de dos archivos grabados por seres humanos.

Por lo tanto, debido a que su programación es sencilla, usamos el software libre *R* [34], el cual nos lee la matriz de similaridad que formamos con *KSH* y nos produce un dendograma de acuerdo al método *single linkage* [33].

Vimos que se formaron árboles para cada uno, exceptuando los archivos en inglés y la música. Por lo que hay una identificación de los distintos formatos de grabación.

La figura 3.2 muestra el dendograma que obtuvimos.

El audio duplicado mantiene una similaridad completa, ya que la distancia entrópica obtenida entre ellos fue de cero. Mientras que los archivos de repetición por el mismo individuo da resultados dispares, se observó la similitud entre ambos. En *mp3* solo 1/3 fue exitoso, en *ogg* 2/3 y en *wav* 0. Para los textos distintos leídos por los mismos individuos observamos similitud en solo 1/6 para *mp3*, 2/6 para *ogg* y 0 para *wav*.

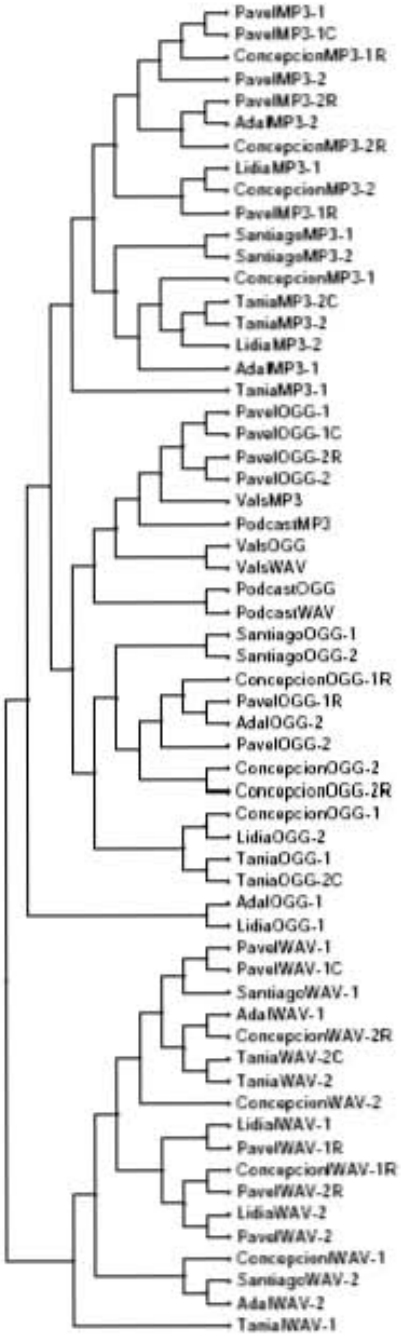

Figura 3.2: Dendrograma de las distancias de los archivos analizados.

El formato que aparenta mayor precisión en identificar a un mismo individuo es el *ogg*, siendo el que mayor relaciones exitosas presentó en el dendograma. Por ello en lo sucesivo decidimos utilizar exclusivamente éste formato y descartar los otros dos.

Dada la investigación que presentamos respecto a las diferencias entre los formatos, éste resultado es esperado, recordando que *ogg* es un formato sin compresión y posee un dominio en las frecuencias. A pesar de estas diferencias cruciales entre formatos, no esperábamos que *wav* no relacionara identificaciones, debido a que no tiene ningún tipo de compresión, pero su dominio se mantiene en el tiempo, lo cual parece ser la cuausa en la disparidad entre los archivos.

El hecho de que los archivos en inglés y la música se hayan localizado en ramales separadas del resto, diferenciándose de los textos en español, indica que la distancia entrópica entre estos archivos es grande.

El uso de dendogramas nos crea el problema de tener un archivo muy cercano a dos grupos y que puede resultar en una idenficación errónea, formando grupos falsos. Al ser una respresentación visual de la cercanía entre archivos es necesaria su inspección a detalle y si el número de archivos que se desea analizar es grande puede ser un procedimiento complicado. En la siguiente sección propondremos un método nuevo y conveniente para proporcionar otra herramienta y así analizar las similaridades.

3.3. Archivos patrón y criterio para la identifición de archivos desconocidos

Ya que tenemos la matriz de similidaridad formada, también podemos calcular valores estádisticos de las distancias entre los archivos, como la media y la desviación estandar. Como queremos crear un criterio para indicar cuales archivos son los más parecidos. Sabemos que tenemos una distribución de valores aleatorio, la desviación estándar al ser un valor propio de la muestra, nos indica como es la distribución respecto al promedio.

La desviación estándar puede indicarnos que tan dispersa se encuentra la muestra, por lo que podemos usarla como un valor estádistico para dividir en intervalos. Dicho de otra manera, deseamos proponer un valor en el cual sea posible decir que una distancia entrópica es pequeña, así poder decir que un archivo es lo suficientemente cercano a otro como para enunciarlo en un

mismo grupo.

El promedio o media se calcula:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.4}$$

Por lo que la desviación estándar se calcucla usando la media:

$$S_X^2 = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n} \tag{3.5}$$

La razón de usar una fracción de la desviación estándar para subdividir los archivos y así formar grupos (*cúmulos*), es una propuesta propia, que se retribuye a partir de la distribución observada en los resultados que se presentan en el capítulo 4.

Como los resultados muestran que a cierta distancia cercana al cero e/xis/ten archivos con distancias entrópicas muy pequeñas, quisimos usar este hecho para mostrar que podemos formar ciertos grupos de archivos que se encuentran muy cercanos unos con los otros y que también existen otros que son parte de más de un grupo que no se pueden aglomerar.

La figura 3.3 muestra el criterio por el cual se dividen en grupos los archivos.

Como era de esperarse al crear estas listas un mismo archivo puede estar igualmente cercano a dos grupos, precisamente uno de los inconvenientes que se manifestó al usar dendogramas. Un archivo cercano a dos grupos distintos se repitió en más de una lista y fue eliminado manualmente de la misma. Si una lista se componía en su mayoria de archivos repetidos entonces la lista era eliminada.

De esta forma pudimos formar grupos con archivos patrón esperando que los integrantes fueran de una especie en particular. Por ejemplo en la sección 3.4 mostraremos que al usar el algoritmo, los archivos de sonido con la especie *Molossus Molossus* que se hallaron cercanos, se los agrupó y así se les consideró patrón.

3.4. Elección de los archivos patrón

Como siguiente experimento introdujimos 269 archivos de sonido (únicamente en *ogg*) provenientes de grabaciones de murciélagos. Con el objetivo

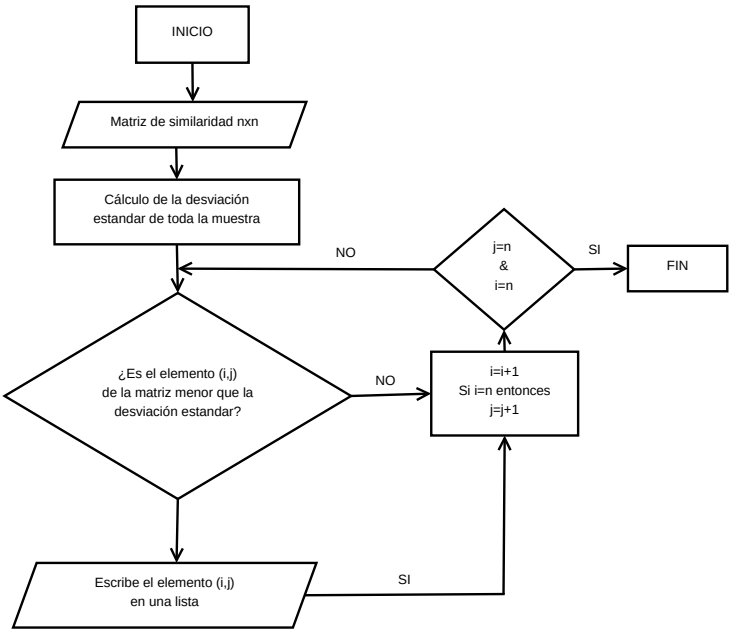

Figura 3.3: Diagrama para formar listas de archivos cercanos

como indica el método, de conocer cuales son los archivos más cercanos o ideales.

Utilizando el sotware para elaborar las distancias entrópicas obtuvimos una matriz de similaridad correspondiente. Los valores estadísticos de la muestra fueron:

Intervalo(valor mínimo y valor máximo): 0.000022 y 1.704161
Desviación Estándar: 0.322148 ± 0.001697
Media: 1.130946

La distribución de frecuencias de las distancias entrópicas entre los 269, convenientemente usando 1/4096 de la desviación estándar como intervalo, resulto como la figura 3.4

La mayoría de las distancias entre archivos se encuentran cercanos a la media; debido a que la desviación estándar es grande, por consiguiente la distribución es diversa. Lo que señala que muchos de ellos se hallan a una

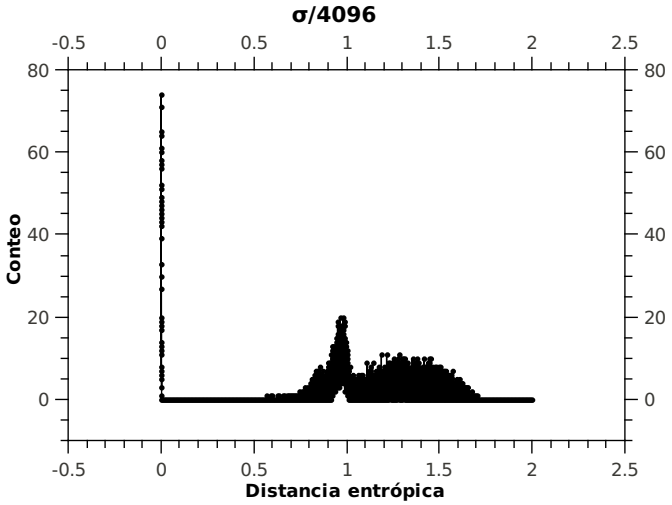

Figura 3.4: Distribución de distancias en formato *ogg*.

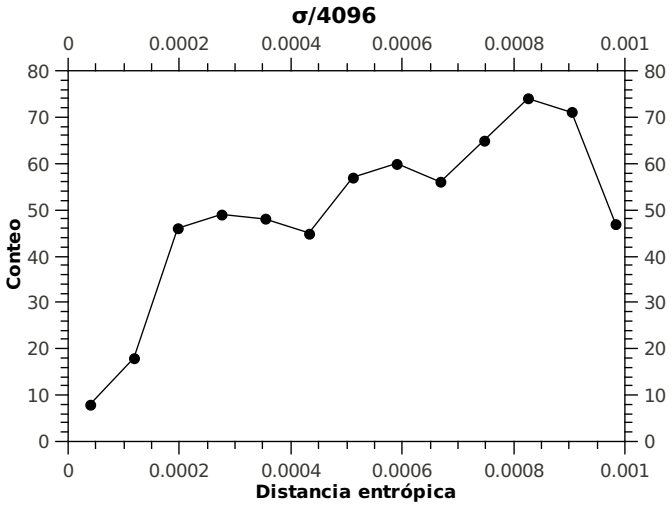

Figura 3.5: Distancias cercanas al cero en formato *ogg*.

distancia entrópica mayor que uno, que podemos considerar como una distancia lejana. Pero también existen archivos con una distancia muy cercana al cero, notorio en el pico izquierdo de la figura 3.4, éstas distancias son un indicio de lo que podemos considerar como una distancia cercana.

La discusión para la determinación del uso de un múltiplo de fracción de desviación estándar como intervalo para indicar cercanía entre archivos se encuentra en la sección 3.3.2

Es importante destacar que fue posible usar distintos intervalos de fracción de la desviación estandar. Determinar cuáles archivos son lejanos y cuáles no lo son, resulta ser una decisión previa que depende de poder excluir la mayor cantidad de archivos, para así los restantes considerarlos cercanos.

Ante esta situación probamos múltiples resultados usando fracciones pares, observando una mayor formación de grupos muy cercanos entre ellos dividiendo entre 2048 la desviación estandar.

Entonces los archivos que distan con otros menos que ésta fracción fueron:

```
lasega10.ogg lasega18.ogg lasega21.ogg

balpli14.ogg cynmex24.ogg cynmex56.ogg molmol092.ogg

balpli14.ogg cynmex07.ogg cynmex24.ogg cynmex56.ogg molmol13.ogg

lasega10.ogg lasega21.ogg molmol58.ogg

cynmex24.ogg molmol09.ogg molmol13.ogg molmol28.ogg

balpli14.ogg molmol09.ogg molmol13.ogg molruf102.ogg

cynmex24.ogg molmol13.ogg molruf10.ogg ptedav19.ogg

lasega21.ogg molmol13.ogg molruf55.ogg permac33.ogg

cynmex24.ogg molmol13.ogg molmol28.ogg

lasega10.ogg lasega18.ogg permac27.ogg
```

ptepar19.ogg ptepar24.ogg ptepar29.ogg

ptepar002.ogg ptepar122.ogg ptepar24.ogg

ptedav28.ogg ptepar12.ogg ptepar19.ogg ptepar24.ogg ptepar29.ogg ptepar39.ogg ptepar48.ogg

Eliminando archivos repetidos en distintos grupos y juntando grupos de una misma especie, obtuvimos:

lasega10.ogg lasega18.ogg lasega21.ogg

cynmex24.ogg cynmex56.ogg cynmex07.ogg

molmol09.ogg molmol13.ogg molmol28.ogg

ptepar002.ogg ptepar122.ogg ptepar12.ogg ptepar19.ogg ptepar24.ogg ptepar29.ogg ptepar39.ogg ptepar48.ogg

Únicamente mantuvimos 17 archivos divididos en cuatro especies que pudimos considerar patrón. Corresponden a 3 archivos para Lasiurus ega, 3 para Cynomops mexicanus, 3 para Molossus molossus y 8 para Pteronotus personatus. Los valores estadísticos para cada grupo fueron los siguientes:

Cynomops mexicanus (3 archivos)
Media: 0.000605
Desviación estandar: 0.000152 $\pm$ 0.000009

Lasiurus ega (3 archivos)
Media: 0.000539
Desviación estandar: 0.000315 $\pm$ 0.000182

Molossus molossus (3 archivos)
Media: 0.000148
Desviación estandar: 0.000105 $\pm$ 0.000006

Pteronotus personatus (8 archivos)
Media: 0.000706

`Desviación estandar: 0.000623 ± 0.000118`

Estos valores para la media de cada grupo fueron los utlizados como radios de los archivos patrón y así poder continuar con la comparación de de archivos desconocidos.

3.5. Identificación de archivos desconocidos

El siguiente paso fue implementar un proceso por el cual al introducir un archivo desconocido poder establecer si es lo suficientemente cercano para poderlo considerar parte de la especie, parte del grupo o ninguno de los dos.

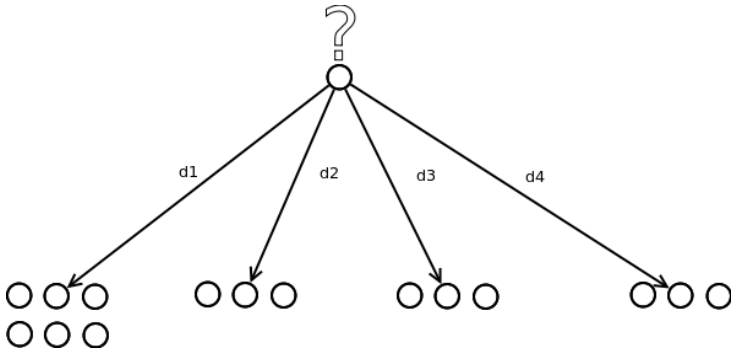

Figura 3.6: Distancia entre un ejemplo y grupos de especies identificadas.

Creamos un algoritmo, 3.5, que determina que tan cercana es la muestra de los patrones a partir de que a cada grupo consituido de archivos patrón se les puede medir su distancia media como

$$\bar{d}_m = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{3.6}$$

Donde $\bar{d}_m$ es la media, n es el número de elementos de la muestra y a_i es el valor del iésimo elemento.

Podemos multiplicar la media por un factor $f > 1$, de manera que

$$B = f \cdot \bar{d}_m \tag{3.7}$$

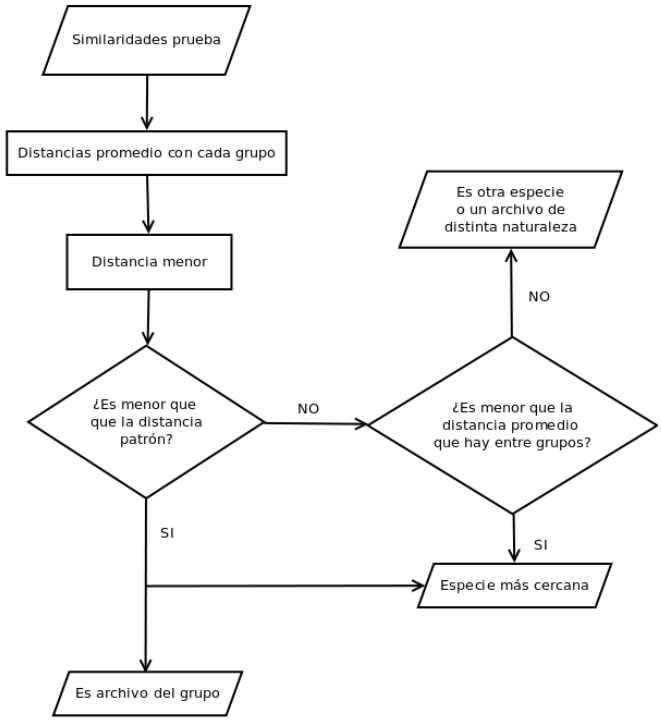

Figura 3.7: Algoritmo de selección

Como $B > \bar{d}_m$ nos representa un radio o "canasta" como criterio para observar si un archivo desconocido es menor que B.

Si calculamos ahora la media del archivo desconocido como

$$\bar{d}_d = \frac{1}{n} \sum_{i=1}^{n} b_i \tag{3.8}$$

donde b_i es la distancia entrópica entre el archivo desconocido y cada archivo patrón de una misma especie. Entonces si $\bar{d}_d < \bar{d}_m$ podemos considerar al archivo desconocido como parte de la misma especie.

En la figura 3.5 se ejemplifica la propuesta de "canasta"

El método aquí mostrado propone crear "bibliotecas" con archivos patrón previamente seleccionados, y con éstas poder comparar la cantidad de

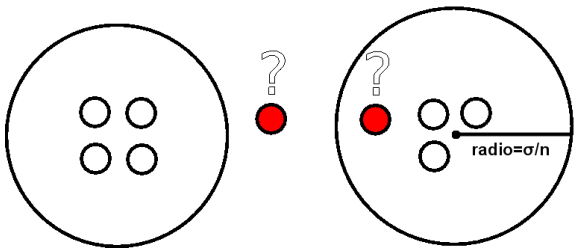

Figura 3.8: Radios de cercania usando la media

archivos desconocidos que se desee. En el siguiente capítulo enunciamos los resultados que se lograron a lo largo del proceso para crear el método.

Capítulo 4

Resultados

En este capítulo se presentan los distintos experimentos ordenados temporalmente, ya que al analizar el resultado inmediato anterior de un ensayo se propusieron nuevas formas de resolver el objetivo expuesto y así poder crear un criterio refinado. Dicho de otra forma, los métodos para analizar los archivos fueron evolucionando de manera que nos enfrentamos a los problemas para buscar si es viable la comparación de archivos sonoros mediante el uso de la distancia entrópica.

Primeramente dictaminamos simplificar el trabajo computacional seleccionando un único formato, como es práctica común en biología utilizamos dendogramas para ello, pero notamos la dificultad de exponer resultados de esta forma. Por ello nos basamos solamente en obtener resultados no de forma gráfica, sino como listas con los nombres de archivos. Buscamos los archivos patrón, de acuerdo al método propuesto para formar la regla de comparación entre archivos y así con las pruebas posteriores ir perfeccionando éste método.

4.1. Aplicaciones del método

Reingresando los 252 archivos restantes identificados por métodos usuales y utilizando distintos valores para los radios, obtuvimos los resultados de la tabla 4.1.

El mayor número de aciertos respecto al total de archivos identíficados (aciertos/total) fue con el radio en 1.5 veces la media, obteniendo un 70 % de éxitos máximo.

Radio(veces la media)	Total de reconocidos	Identifiaciones correctas	Porcentaje correcto
1.0	5	3	60 %
1.1	5	3	60 %
1.2	5	3	60 %
1.3	8	5	62.5 %
1.4	8	5	62.5 %
1.5	10	7	70 %
1.6	11	7	63.6 %
1.7	12	7	58.3 %
1.8	12	7	58.3 %
1.9	14	7	50 %
2.0	14	7	50 %

Tabla 4.1: Primera aplicación de archivos reintroducidos

La figura 4.1 presenta los distintos radios utilizados y su porcentaje de éxito en el reconocimiento.

Al introducir los 500 archivos desconocidos obtuvimos los siguientes resultados. Se identificó exitosamente un único archivo, obteniendo un éxito de un 20 % cuando se usó 1.5 a 1.6 veces el radio. Este resultado no es conveniente debido a que solo se identificó menos del 10 % del total y muy pocos correctos.

Dados estos resultados desiguales, en el que se identifico menos del 10 % del total y muy pocos correctos, decidimos buscar las causas. Nos remitimos al M. en C. Helxine Fuentes, para corroborar los resultados. Encontramos que los ejemplares desconocidos poseían una gran cantidad de ruido de fondo e incluso más de una especie en un mismo sonograma, además notamos longitudes de tiempo distintas a las de sonogramas patrones.

El tamaño de los archivos en kilobytes posee una relación proporcional a su duración en segundos, por lo que se pueden usar ambos términos para referirnos a lo mismo. Mientras que un sonograma con ruido y con un traslape de otra especie se observa ejemplificado en la siguiente figura.

En el ejemplo, la especie Molossus molossus se halla en el rango de frecuencias entre 25 y 40 Khz mientras que Pteronotus parnelli cercano a los

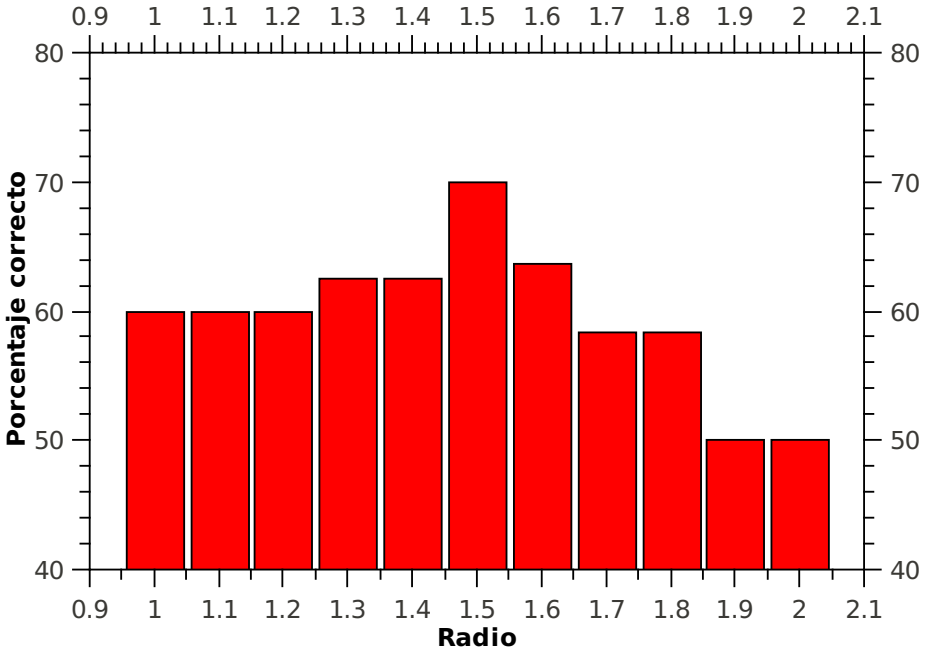

Figura 4.1: Porcentaje de identificaciones correctas con distintos radios.

60 Khz, también se realizó la grabación accidental de un insecto que se encontraba en la zona. Ambas especies se encuentran cortadas y con patrones sonoros poco definidos.

Por estos motivos resolvimos usar otra muestra nueva, identificada por métodos usuales, eligiendo sonogramas limpios y que mantuvieran una longitud similar (35-75 kilobytes).

De esta selección resultaron 150 sonogramas, en donde hubo una identificación de dos archivos correctos, dando un 100 % de éxito a partir de 1.1 veces el radio. No se obtuvieron falsos positivos hasta aumentar el radio a 135 veces.

Este resultado nos indica que al seleccionar archivos de muestra que tengan una longitud parecida y que no se encuentren sujetas a anomalias, nos proporciona una identificación de 100 %, lo cual es muy favorable pues otorga una herramienta más para desarrollar el método. Utilizamos el radio indis-

Radio(veces la media)	Total de reconocidos	Identifiaciones correctas	Porcentaje correcto
1.0	0	0	0 %
1.2	3	0	0 %
1.3	3	0	0 %
1.4	4	0	0 %
1.5	5	1	20 %
1.6	5	1	20 %
1.7	6	1	16.6 %
1.8	6	1	16.6 %
1.9	9	1	11.1 %
2.0	9	1	11.1 %

Tabla 4.2: Segunda aplicación para sonogramas desconocidos

tintamente mayor que 1.1 y menos que 100.0, debido a que la tasa de éxito fue la misma.

4.2. Método extendido

Los experimentos de las secciones anteriores, dieron constancia de que es posible proponer un método combinado, en el cual se tenga la posibilidad de obtener la mayor cantidad de archivos identificados exitosamente. El método consiste en utilizar los resultados de las distintas aplicaciones anteriores. Proponemos realizar este experimento para poder concluir la forma más eficiente para identificiar los sonogromas de murciélagos.

Volvimos a evaluar todos los sonogramas muestra disponibles (902 archivos), para posteriormente en otro experimento seleccionar únicamente los archivos con longitudes entre 35 y 75 kilobytes (116 archivos).

De la anterior tabla se aprecia que el hecho de tomar sonogramas con tamaños semejantes a los patrones el porcentaje de éxitos aumentó, ya que el total de reconocidos disminuyo, pero las identificaciones correctas permanecieron similares. Cabe mencionar, que el archivo identificado exitosamente faltante en la selección correspondió al proveniente de la muestra de desconocidos, debido a que su longitud fue menor al criterio tomado.

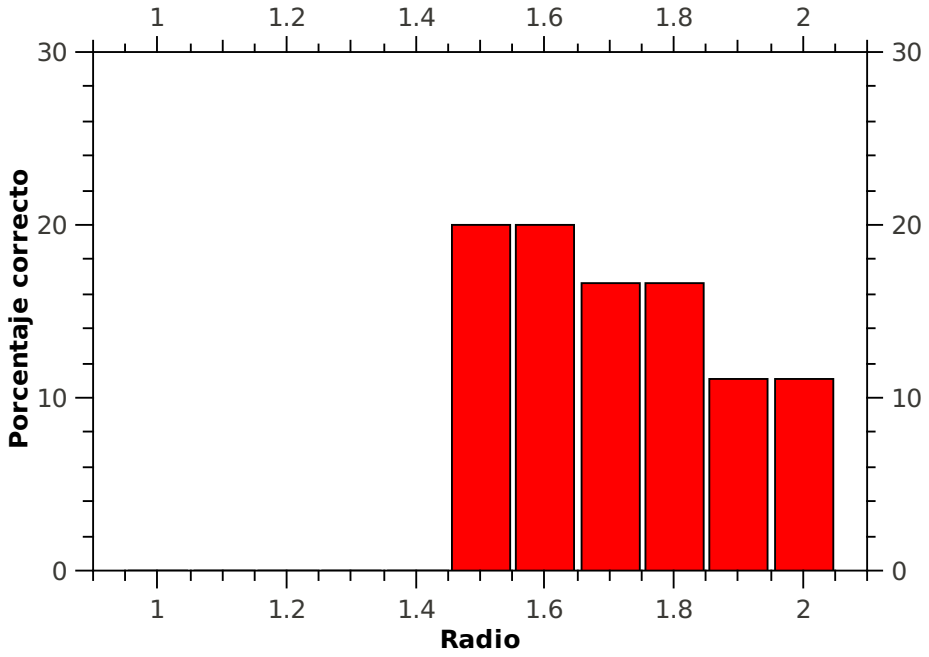

Figura 4.2: Porcentaje de identificaciones correctas con distintos radios para archivos desconocidos

La especie que exhibe una mayor cantidad de reconocimiento fue Pteronotus parnellii (Ptepar), ya que para la misma no se mostraron falsos positivos. Mientras que la especie *Lasiurus ega* (Lasega) produjo la mayor cantidad de archivos falsamente identificados. En todos los casos el radio de 1.5 veces la media resultó ser el más adecuado para obtener una mayor cantidad de reconocimientos exitosos. Mientras que la identificación de archivos fue entre 1 % y 4 % del total de archivos ingresados.

Desconocemos la razón por la cual la especie *Lasiurus ega* produce errores. No es posible evitar que esa especie sea grabada y tampoco que se grabe otro tipo de ruido, por lo que no refinamos esa parte.

Como ejercicio importante mostraremos por pasos la implementación propuesta del método para identificar un sonograma desconocido.

1. Grabación con Anabat

Radio(veces la media)	Total de reconocidos	Identifiaciones correctas	Porcentaje correcto
1.0	0	0	0 %
1.1	1	1	100 %
1.2	1	1	100 %
1.3	1	1	100 %
1.4	1	1	100 %
1.5	1	1	100 %
1.6	1	1	100 %
1.7	2	2	100 %
1.8	2	2	100 %
1.9	2	2	100 %
2.0	2	2	100 %
3.0	2	2	100 %
10.0	2	2	100 %
50.0	2	2	100 %
100.0	2	2	100 %
125.0	2	2	100 %
135.0	5	2	40 %
140.0	19	2	10.5 %

Tabla 4.3: Tercer aplicación para sonogramas seleccionados

2. Codificación *WAV*

3. Codificación *OGG*

4. Selección de archivos con longitudes similares a las bibliotecas.

5. Cálculo de sus distancias entrópicas

6. Comparación con las bibliotecas (1.5 veces la media).

El proceso es un resumen de lo que resulta ser el método más eficiente, las implicaciones y otras cuestiones que nos faltaron se discutirán en las conclusiones.

	Total de reconocidos	Identifiaciones correctas	Porcentaje correcto
Muestra total	16	9	56.2 %
Selección	12	8	66.6 %

Tabla 4.4: Pocentaje de identificaciones correctas para la muestra total y sólo con archivos seleccionados.

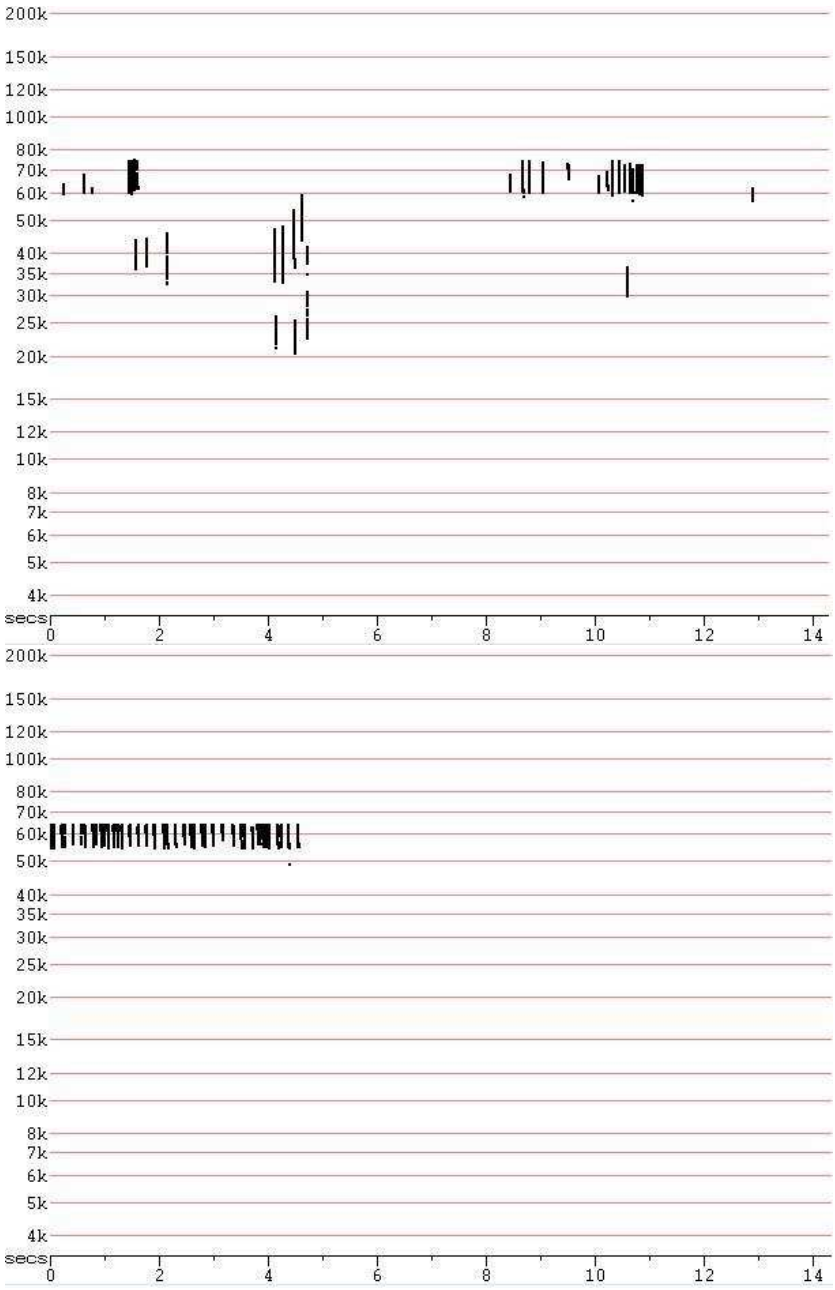

Figura 4.3: El primer sonograma ejemplifica un caso con ruido por parte de un insecto y dos especies diferentes juntas. Para realizar una comparación la segunda figura es un ejemplo de sonograma típico de la especie Pteronotus parnellii.

Capítulo 5

Conclusiones

La identificación de especies de murciélagos es clave en la biología para poder estudiar poblaciones de organismos y la dinámica de los sistemas ecológicos. Para realizar esta identificación existen muchos métodos que en ocasiones implican visualizar los individuos. Frecuentemente dicha observación visual no es posible sin realizar una captura. En el caso de las aves existe la posibilidad de escuchar vocalizaciones, ya que cada especie posee llamados típicos.

En el caso de murciélagos puede realizarse la misma tarea, sin embargo es más complicado, ya que estos llamados acústicos típicos no se encuentran en el espectro audible por el ser humano. Para grabar el sonido emitido por murciélagos es inevitable utilizar equipamento especial que grabe en dichas frecuencias y con el uso de una computadora observar una representación del audio. Al ser necesario el conocimiento previo de las vocalizaciones de cada especie puede llegar a ser una actividad complicada y tediosa.

En este trabajo propusimos una metodología para automatizar la labor de identificar especies de murciélagos utilizando la entropía algorítmica de Kolmogorov a un archivo binario, específicamente a una grabación de audio. Esto se pudo lograr al asociar una métrica propuesta usada anteriormente en archivos de texto, en donde se calcula una distancia entrópica que indica que tan cercanos son dos de estos archivos.

Primeramente valoramos nuestra propuesta y determinamos el mejor formato de sonido, vimos que se logra diferenciar de un texto leído por varias personas con audio en inglés y de un archivo de música. Utilizando algoritmos para la formación de dendogramas notamos que la creación de grupos no fue la adecuada para el español. Así diferenciamos que la distancia entrópica

puede ser aplicada a archivos de sonido guardados en formato *OGG*.

Planteamos una forma alternativa para abordar el problema. Para ello fue necesaria la formulación de un método nuevo. Quisimos que esta nueva forma de tratar los archivos involucrara tanto elementos estadísticos como computacionales y que no presentara dificultades en su implementación.

Mediante estadística descriptiva usual, notamos que las distancias entrópicas se agrupan muy cercanas al cero. Pudimos formar grupos de una misma especie y que fueran muy cercanos entre si. Creamos un criterio para definir esta cercanía utilizando la desviación estándar, que describe el grado de centralización o dispersión de las distancias entre archivos.

Gracias a la definición de distancia entrópica usada se formalizó una metodología que puede asociar sonogramas a especies de murciélagos dentro de un banco de archivos patrón.

Continuamos tomando un archivo que fuera de especie desconocida e indicamos otro criterio para indicar la mayor cercanía a un grupo patrón para cada especie. Propusimos que otra medida estadística de estos ideales con el desconocido nos podría dar cuenta de su cercanía. Utilizamos la media de cada grupo y la comparamos con el archivo en cuestión. Encontramos que a 1.5 veces la media existía un mayor acierto para archivos con una especie conocida. Mientras que para los archivos desconocidos la identifición fue casi nula.

Para resolver este problema, una inspección a los sonogramas desconocidos demostró que son muy "ruidosos", tienen especies combinadas, insectos y todo tipo de perturbaciones. De igual forma una distancia entrópica implica que se comparan dos archivos de distintas longitudes en bits, obtuvimos que una longitud similar produce mejores identificaciones.

Es posible seleccionar archivos que tengan longitudes similares. Sin embargo, evitar que se grabe ruido es prácticamente imposible. Emplear filtros en el sistema Anabat reduce el ruido, pero queda como trabajo posterior su uso.

Además con un método extendido, que consiste en preseleccionar archivos que tengan la misma longitud y poco ruido, lo que nos dió una mejora en las identificaciones correctas. Con un 66 % de identificaciones correctas, con un algoritmo automático y con un tiempo de proceso pequeño. Esto hace que el proceso de la información sea muy rápido en comparación con el métdo tradicional de inspección visual de los sonogramas.

También encontramos que la selección de los archivos "patrón" es un proceso que influye en el resultado final en el momento de identificar descono-

cidos. Proponer bibliotecas con mayor número de archivos puede resultar en una mejora en la tasa de identificación.

Finalmente proponemos que esta aplicación de la distancia entrópica expresa un método novedoso que merece ser explorado a otras formas de codificar información.

Bibliografía

[1] Helxine Fuentes Moreno, Estructura del ensamble de murciélagos de La Venta, Oaxaca, México, Tesis de Maestría, IPN, 2010.

[2] Randall T. Schuh, Biological systematics : principles and aplications, Cornell University Press, 2000, p. 38.

[3] D. E. Wilson, Murciélagos, Respuestas al vuelo, Universidad Veracruzana, 2002, p. 1-21 y 55-72.

[4] Aida Trejo Ortíz, Caracterización acústica de los murciélagos insectívoros del Parque Nacional Huatulco, Oaxaca. Tesis de Maestría, IPN, 2010.

[5] Thomas H. Kunz y M. Brock Fenton, Bat ecology, University of Chicago Press, 2003. Cap. 12.

[6] http://users.lmi.net/corben/ (mayo 2012).

[7] Brian S. Everitt, Sabine Landau, Morven Leese, Cluster analysis, Oxford University Press, 2001. Cap. 3 y 4.

[8] John W. Jewett y Raymond A. Serway, Physics for Scientists and Engineers with Modern Physics, Cengage Learning 2010, 469-471.

[9] http://www.joelstrait.com/blog/2009/10/12/a_digital_audio_primer (junio 2012).

[10] http://windows.microsoft.com/en-US/windows7/Codecs-frequently-asked-questions (junio 2012).

[11] http://www.iana.org/assignments/media-types/audio/index.html (junio 2012).

[12] http://www.sonicspot.com/guide/wavefiles.html (junio 2012).

[13] N.S Jayant y Peter Noll, Digital Coding of Waveforms, Prentice Hall, 1984.

[14] https://ccrma.stanford.edu/courses/422/projects/WaveFormat/ (junio 2012).

[15] http://www.mp3-tech.org/ (junio 2012).

[16] Pan Davis, A Tutorial on MPEG/Audio Compression, IEEE Multimedia Journal, 5, 1995.

[17] Karlheinz Brandenburg,MP3 and AAC Explained, Fraunhofer Institute for Integrated Circuits FhG-IIS A, Germany, 2005, 4-5.

[18] http://www.vorbis.com/ (mayo 2012).

[19] Thomas H. Cormen, Introduction to Algorithms, The MIT Press. , 2001, 385-392.

[20] Vorbis I specification, Xiph.Org Foundation, http://xiph.org/vorbis/doc/Vorbis_I_spec.pdf (2012).

[21] Darrel Hankerson, Greg A. Harris, Peter D. Johnson, Jr., Introduction to information theory and data compression, CRC Press, 2003, Cap. 2 y 9

[22] Christina Zeeh, The Lempel Ziv Algorithm, Seminar: Famous Algorithms, , January 16 2003.

[23] Montrol, E.W, About the Physics of no physical systems. J. Stat Phys, Vol. 42, 1986, 647.

[24] Benedetto, D. Cagliotu, E. Loreto, V. Language Trees and Zipping, Phys. Rev. Lett. 88, 048702, 2002.

[25] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, and W. Zurek, Information Distance, IEEE Trans. Information Theory, 44:4, 1998, 1407–1423

[26] Cortes HD, Del Rio JA, Garcia EO, Web Implementation of Entropy-like Algorithms for Citation Mining, WSEAS Transactions on Information Science and Applications, 2(9): (2005) 1430-1437 .

[27] Kostoff RN, Johnson D, del Rio JA, Bloomfield LA, Shlesinger MF, Malpohl G, Cortes HD, Duplicate publication and 'paper inflation' in the fractals literature, SCI. ENG. ETHICS 12, (2006) 543-554.

[28] Russell J., del Río JA, Cortés HD, Highly visible science: a look at three decades of research froma Argentina, Brazil, Mexico and Spain, Interciencia, 32: (9), (2007), 629-634 .

[29] Lipschutz, Seymuour, Theory and problems of probability, New York, McGraw-Hill, 1974, 9, 23 y 55.

[30] Brian S. Everitt, Torsten Hothorn, A Handbook of Statistical Analyses Using R, Chapman and Hall, 2006, Cap. 18.

[31] https://maxtla.cie.unam.mx/relentr/ (abril 2012)

[32] http://kornshell.com/ (abril 2012)

[33] http://www.mkssoftware.com/docs/ (abril 2012)

[34] http://www.r-project.org/ (mayo 2012)

[35] http://cran.r-project.org/manuals.html (enero 2012)

[36] http://soundconverter.org/ (abril 2012)