Segmentación evolutiva de la voz

D. H. Milone [*]
Laboratorio de Cibernética, Departamento de Bioingeniería,
Facultad de Ingeniería, Universidad Nacional de Entre Ríos
Ruta 11 Km 10 ½, (CP 3100), Paraná, Entre Ríos, Argentina

J. J. Merelo
Departamento de Arquitectura de Computadores,
Universidad de Granada, España

La segmentación de la voz consiste en dividir una emisión en diferentes trozos de acuerdo con algún criterio. Es común que se segmente la voz para separarla en fonemas pero también suele ser de interés la segmentación según sílabas o unidades de nivel superior, como la palabra. Para la segmentación de voz se han utilizado varias técnicas. En primer lugar está la segmentación manual, en la que generalmente un experto lingüista genera la segmentación en base a espectrogramas, curvas de energía, entonación y otros estudios utilizados para el análisis de la voz. Esta técnica posee la ventaja de que la experiencia del lingüista asegura un muy buen resultado en la segmentación. Sin embargo los costos en tiempo y recursos que lleva este proceso manual son altísimos, lo que lo hace sólo aplicable a estudios muy especializados. La segunda técnica aplicable a la segmentación viene de la mano de los sistemas de reconocimiento automático del habla basados en modelos ocultos de Markov. En este caso, se entrena el sistema y mediante el algoritmo de Viterbi se puede obtener la secuencia más probable de estados que determina la segmentación. Sin embargo, para realizar esta operación es necesario contar con la transcripción correcta de la emisión de voz.
Los diferentes métodos de computación evolutiva han brindado en la última década una solución a muchos problemas, principalmente en la búsqueda y optimización de soluciones. En este trabajo se propone utilizar éstas técnicas para la segmentación de la voz. En primer lugar se define cómo se representa la solución del problema mediante cromosomas. Luego se define la función de aptitud: una medida de la capacidad de supervivencia de un individuo. En el dominio de las soluciones, se debe poder medir qué tan buena es cada solución en relación a las demás. Finalmente se aplican los operadores de selección, reproducción y variación.
Las pruebas que se realizaron se dividen en tres partes. En primer lugar se presenta un ejemplo que tiende a mostrar las características más importantes del algoritmo de segmentación evolutiva. Este experimento se realiza en base a una señal creada artificialmente con información que resulta en una segmentación obvia. Los segundos experimentos se realizaron en un archivo de voz y se comparan los resultados con la segmentación realizada por modelos ocultos de Markov. En los últimos experimentos se segmentaron 600 frases de un corpus de habla en español. Para comparar ambas segmentaciones se contabilizaron las veces en que la segmentación resultante del método propuesto coincidía con la realizada por modelos ocultos de Markov, con relación al total de marcadores en cada frase. Para el caso de la segmentación silábica se obtuvo una tasa de error del 32.36%. Si se acepta un marcador desalineado por cada palabra el error se reduce al 7.59%.
Una de las principales ventajas de los métodos propuestos es que no existe un proceso de entrenamiento ni parámetros almacenados para su posterior utilización durante la segmentación. Esto, si bien hace que los métodos trabajen con muy poca información de la tarea a realizar, también les da robustez, flexibilidad y aprovecha al máximo su capacidad de autoadaptación.

1 Introducción

En el caso más simple, el problema de la segmentación de voz consiste en encontrar los límites precisos que definen a cada segmento o unidad fonética [16][18][7]. Cada segmento presenta dos límites o marcadores que miden el tiempo, a partir del inicio de la emisión, en que se encuentran el principio y el final del segmento en cuestión. Una emisión puede tener muchos segmentos y así la ubicación correcta de todos sus límites puede ser un problema complejo. Más aún si se consideran todas las variaciones asociadas con los distintos lenguajes, como generalmente ocurre en los problemas relacionados con el habla.

Se han utilizado diversos métodos para la segmentación de voz, entre ellos: la segmentación manual por expertos lingüistas, la segmentación basada en modelos ocultos de Markov (MOM) [3], en redes neuronales artificiales (RNA) [10][19][8], el modelado estadístico [5][15] y el filtrado paramétrico [11]. En cualquier caso el problema de la segmentación automática aún sigue sin ser resuelto totalmente y menos aún en aplicaciones de tiempo real.

[*] Correspondencia: d.milone@ieee.org

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & J. J. Merelo; "Segmentación evolutiva de la voz"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 71, 2003.

La computación evolutiva se ha aplicado con buenos resultados en la segmentación de imágenes [2]. La analogía en que se basa la computación evolutiva estriba en reconocer el mecanismo esencial del proceso evolutivo en la naturaleza e imitarlo para el diseño y optimización de sistemas artificiales. La computación evolutiva abarca un número cada vez mayor de métodos basados en la misma idea original. Entre muchos otros se destacan: los algoritmos genéticos [6], la programación genética [9] y la programación evolutiva [13]. Una revisión y comparativa de éstos y otros métodos de computación evolutiva puede verse en [1]. Los componentes fundamentales del mecanismo de la evolución biológica son los cromosomas —material genético de un individuo biológico—, donde se guardan sus características únicas. Los cambios en el material genético de las especies permiten el proceso de adaptación. El proceso de evolución se ve afectado por: la selección natural, la recombinación de material genético y la mutación; fenómenos que se presentan durante la reproducción de las especies. La competencia entre los individuos por los recursos naturales limitados y por la posibilidad de procreación o reproducción permite que sólo los mejor adaptados sobrevivan. Esto significa que, en términos generales, el material genético de los mejores individuos sobrevive y se reproduce.

Los métodos de computación evolutiva manipulan una población de soluciones potenciales codificadas en cadenas o vectores que las representan. Los operadores artificiales de selección, cruza y mutación son aplicados para buscar los mejores individuos (mejores soluciones) a través de la simulación del proceso evolutivo natural. Cada solución potencial se asocia con un valor de aptitud, que mide qué tan buena es comparada con las otras soluciones de la población. Este valor de aptitud es la simulación del papel que juega el ambiente en la evolución natural darwiniana. Este paradigma se resume en:

```
Crear Población
Evaluar Población
Mientras MejorAptitud < AptitudRequerida
  Seleccionar Progenitores
  Reproducir Progenitores
  Evaluar nueva Población
FinMientras
```

Para comenzar se crea la población completamente al azar. En la configuración inicial hay que tener en cuenta que la distribución de valores debe ser uniforme para cada rango representado por los cromosomas. Luego se decodifica el genotipo en el fenotipo de esta población inicial y se evalúa la aptitud de cada individuo: se le asigna un valor numérico a su "capacidad de supervivencia" o bien, en el espacio de soluciones del problema, se mide qué tan bien resuelve el problema cada individuo. A continuación se entra en el bucle de optimización o búsqueda. Este ciclo termina cuando se encuentra una solución adecuada para el problema —cuando la aptitud para el mejor determina que su fenotipo es suficientemente bueno como solución— o se cumple un número máximo de iteraciones.

2 Algoritmo evolutivo para la segmentación de voz

2.1 Marcadores de segmentación

Los vectores de características $\mathbf{x}_t$ se obtienen a partir de un análisis por tramos de la señal de voz: $x(t;k) = \mathcal{T}(k)\{v(t;n)\}, \quad 0 < k \leq N_x$, donde $\mathcal{T}(k)$ es un operador para la transformación de dominio y $v(t;n)$ los tramos de voz en el tiempo. La segmentación da como resultado un conjunto $\Phi = \{E_m\}$ donde cada segmento E_m contiene vectores de características $\mathbf{x}_t$ con determinado grado de pertenencia. Sobre esta definición general se harán dos restricciones. La primera es considerar que la segmentación es totalmente exclusiva, es decir, cada vector de características puede pertenecer a sólo un segmento $x(t;k) \in E_{j_1} \Leftrightarrow x(t;k) \notin E_{j_2} \forall j_2 \neq j_1$. Esto permite describir la pertenencia sin un *grado* de pertenencia asociado a cada vector. La segunda restricción está en que el orden temporal según el que aparecen los vectores de características en los segmentos no puede ser invertido. Las dos restricciones se pueden expresar conjuntamente mediante $x(t_1;k) \in E_{j_1} \wedge x(t_2;k) \in E_{j_2} \Leftrightarrow t_1 < t_2 \forall j_1 < j_2$.

Dadas estas restricciones, se puede representar la segmentación mediante el vector de los marcadores del primer elemento de cada segmento $\boldsymbol{\phi} = [M_1, M_2, \ldots, M_{N_\phi}]$ con $N_\phi = |\Phi| + 1$ ya que se incluyen los marcadores inicial y final y además $1 \leq M_1 < M_2 < \ldots < M_{N_\phi} \leq T + 1$. Como se verá luego, es conveniente dejar abierta la posibilidad de que la primera marca sea mayor a 1 y la última menor que $T + 1$. Estrictamente $\mathbf{x}_t$ no está definido en $t = T + 1$ pero sí será válido el marcador para la definición de la función de aptitud.

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & J. J. Merelo; "Segmentación evolutiva de la voz"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 71, 2003.

2.2 Representación de los individuos

El primer aspecto a resolver en el diseño del algoritmo de computación evolutiva es la codificación del problema en un alfabeto finito. Tradicionalmente se han empleado cadenas binarias —los denominado algoritmos genéticos puros— pero actualmente se están empleando esquemas más flexibles [13][12].

En el material genético de cada individuo de la población se deberá codificar un conjunto de marcadores de segmentación. Esta codificación tomará como punto de partida la segmentación lineal de la emisión de voz. En principio se trabajará en la base de que se conoce el número de segmentos $|\Phi|$. Luego se discutirá un método para eliminar esta restricción. La partición lineal consiste en asignar los marcadores de cada segmento según $M_j = M_1 + \frac{M_{N_\Phi} - M_1}{N_\Phi - 1}(j-1)$ con $1 < j < N_\Phi$, donde los marcadores inicial y final pueden no necesariamente ser 1 y T. De hecho, se implementó un detector de inicio y finalización de la emisión basado en el análisis por ventanas de la energía, lo cual permite reducir el espacio de búsqueda para la segmentación.

A partir de esta segmentación lineal se pueden definir los desplazamientos de los marcadores como $\Delta\phi = [\Delta M_2, \Delta M_3, \ldots, \Delta M_{N_\phi - 1}]$, que será un vector más conveniente para la evolución (ver Figura 1). El vector de desplazamientos $\Delta\phi$ no incluye al desplazamiento para el primer y último marcador debido a que quedan fijos. Los desplazamientos para los marcadores ΔM_j son números enteros que están en un rango determinado por las máximas longitudes posibles para los segmentos. En el caso de la segmentación de fonemas es suficiente que este rango permita hasta 50 ms de desplazamiento. Sin embargo, para la segmentación de sílabas, el rango puede llegar a los 200 ms.

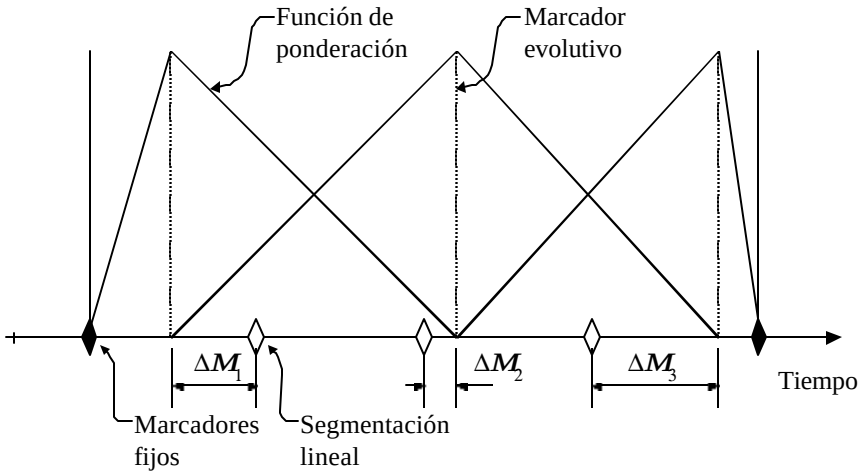

Figura 1: Marcadores de segmentación y funciones de ponderación. En este ejemplo se pueden observar los marcadores evolutivos (líneas de punto) y la segmentación lineal (◊). A partir de los marcadores se esquematiza también la función de ponderación $\alpha(\cdot)$.

De esta forma queda definida la codificación del material genético de cada individuo como un vector de enteros, con rango acotado y conocido, que posee los desplazamientos que deben realizarse a partir de los marcadores de la segmentación lineal, sin incluir el primero y el último. El método para obtener los marcadores a partir de la información codificada en el material genético de cada individuo es $M_j = M_1 + \frac{M_{N_\Phi} - M_1}{N_\Phi - 1}(j-1) + \frac{\Delta M_j}{T_d}$ con $1 < j < N_\Phi$. En esta ecuación aparece el paso de las ventanas de análisis T_d para convertir el tiempo de los desplazamientos de cada marcador en índices de tiempo en el análisis por tramos.

Quedan por resolver algunas cuestiones relacionadas con el proceso mismo de evolución. Dado que evoluciona una codificación de las soluciones del problema y no las soluciones en sí mismas, es posible que durante la evolución el material genético dé como resultado fenotipos no válidos (soluciones incoherentes). En este problema en particular y dada la codificación elegida, existen dos casos en que las soluciones no son válidas. El primero es cuando al decodificar los marcadores no se respeta su orden natural y se producen solapamientos. El segundo caso es cuando uno o más marcadores están fuera de los límites de tiempo de la emisión, posibilidad que existe independientemente del primer caso dado que los marcadores inicial y final no forman parte de la evolución.

El problema se puede resolver de muchas formas [13]. Por ejemplo, se podría elegir una codificación que no permita estos errores genéticos luego de la aplicación de los diferentes operadores. También se podrían diseñar

operadores que no permitan la generación de cromosomas erróneos a partir de cromosomas válidos. En cualquier caso se trata de adaptaciones del algoritmo de computación evolutiva al problema en mano.

Una técnica más sencilla que no implica una modificación importante en la idea de la computación evolutiva es la operación de verificación y reparación combinada con la penalización de aptitud. Para realizar la verificación del solapamiento simplemente se debe comprobar la inecuación $M_{j_1} < M_{j_2} \forall j_1 < j_2$ con $1 < j_1, j_2 < N_\phi$. La verificación se completa comprobando que ningún marcador se encuentra fuera de los límites determinados por los marcadores inicial y final. Todo se puede resumir ampliando los rangos en la expresión anterior a $1 \leq j_1, j_2 \leq N_\phi$.

2.3 Función de aptitud

Es necesario obtener una medida de qué tan buena es la solución que ofrece cada individuo. La función de aptitud trabaja en el dominio del problema, sobre el fenotipo de cada individuo.

Se define el vector propio de un segmento como:

$$\varphi_j(k) = \frac{1}{A_j(\cdot)} \sum_{t=M_j}^{M_{t+1}-1} \alpha(\cdot) x(t;k)$$

siendo $A_j(\cdot) = \sum_{t=M_j}^{M_{j+1}-1} \alpha(\cdot)$ con $0 < j < N_\phi - 1$. El vector propio cumple la función de representar a todo el segmento ya que se obtiene mediante un promedio ponderado de todos sus vectores de características.

La función de ponderación $\alpha(\cdot)$ tiene por objetivo asignar diferentes pesos a los vectores de características según se encuentren más cerca o más lejos del límite del segmento. Como función de ponderación se puede definir, por ejemplo, $\alpha(d,N) = e^{-\frac{d}{N}}$ o bien una relación lineal $\alpha(d,N) = 1 - \frac{d}{N}$, siendo d la distancia al marcador y N el número total de muestras a ponderar. En el caso de que se adopte la relación lineal y $1 \leq d \leq N$, se puede demostrar que $A(d,N) = \sum_{d=1}^{N} 1 - \frac{d}{N} = \frac{1}{2}(N+1)$.

Para distinguir entre el vector propio de un segmento ponderado como anterior o posterior a un marcador, se utilizarán los superíndices '–' y '+', respectivamente. A continuación se presentan las ecuaciones de los vectores propios de un segmento según su posición relativa al marcador:

$$\varphi_j^-(k) = \frac{\sum_{t=M_j}^{M_{j+1}-1} \alpha(M_{j+1} - t, N_{M_{j+1}}) x(t;k)}{\sum_{t=M_j}^{M_{j+1}-1} \alpha(M_{j+1} - t, N_{M_{j+1}})}$$

y

$$\varphi_j^+(k) = \frac{\sum_{t=M_j}^{M_{j+1}-1} \alpha(t - M_j + 1, N_{M_{j+1}}) x(t;k)}{\sum_{t=M_j}^{M_{j+1}-1} \alpha(t - M_j + 1, N_{M_{j+1}})}$$

con $N_{M_j} = M_j - M_{j-1} + 1$.

La distancia euclídea entre dos vectores propios en torno al marcador M_j es:

$$\delta_j^E = \sum_{k=1}^{N_x} \left(\varphi_{j-1}^-(k) - \varphi_j^+(k) \right)^2 ; \quad 1 < j < N_\phi - 1 \tag{1}$$

A partir de esta expresión se define la función de aptitud como el promedio $\Gamma_\phi = \frac{1}{N_\phi - 2} \sum_{j=2}^{N_\phi - 1} \delta_j^E$. Reemplazando según las consideraciones tomadas hasta el momento se obtiene:

$$\Gamma_\phi = \frac{1}{N_\phi - 2} \sum_{j=2}^{N_\phi - 1} \sum_{k=1}^{N_x} \left[\frac{2}{N_{M_{j-1}} + 1} \sum_{t=M_{j-1}}^{M_j - 1} \left(1 - \frac{M_j - t}{N_{M_{j-1}}} \right) x(t;k) - \right.$$
$$\left. - \frac{2}{N_{M_j} + 1} \sum_{t=M_j}^{M_{j+1}-1} \left(1 - \frac{t - M_j + 1}{N_{M_j}} \right) x(t;k) \right]^2 \tag{2}$$

2.4 Selección

Existen varias formas de realizar la selección de los progenitores. Al igual que en la naturaleza, la selección no está relacionada directamente con la aptitud de un individuo sino a través de operadores probabilísticos. Desde el punto de vista del algoritmo de búsqueda, la selección lleva a cabo la tarea de concentrar el esfuerzo computacional en las regiones del espacio de soluciones que se presentan como más prometedoras [17]. Los operadores de selección utilizados en la computación evolutiva generalmente encuentran un compromiso entre estos dos extremos. Tres operadores elementales de selección son: la rueda de ruleta, la selección por ventanas y la competencia [6]. En los experimentos siguientes se utilizó el método de competencias, según el cual se eligen completamente al azar $v > 1$ individuos, se los hace competir por aptitud y queda seleccionado el ganador. Generalmente se utilizan valores de v entre 2 y 5 dependiendo del tamaño de la población. Este método es uno de los más utilizados debido a lo simple y eficiente de su implementación.

2.5 Reproducción

La reproducción es el proceso mediante el cual se obtiene la nueva población a partir de los individuos seleccionados y los operadores de variación. Existen varias alternativas para realizar la reproducción, en el caso más sencillo se obtienen todos los individuos de la nueva población a partir de variaciones (cruzas y mutaciones) de los progenitores. Es posible también transferir directamente a la población nueva los padres seleccionados en la población anterior y completar los individuos faltantes mediante variaciones.

Una variante adicional en la reproducción que no se extrae directamente de la evolución biológica pero que es utilizada con muy buenos resultados es el *elitismo*. En esta estrategia se busca el mejor individuo de la población anterior e independientemente de la selección y variación se lo copia exactamente en la nueva población. De esta manera se resguarda la mejor solución a través de las generaciones.

2.6 Operadores de variación

La *mutación* trabaja alterando alelos de genes con una probabilidad p_m muy baja, por ejemplo $p_m = 0{,}001$. Las mutaciones son típicamente realizadas con una probabilidad uniforme en toda la población y el número de mutaciones por individuo puede ser fijado de acuerdo a esta probabilidad y la cantidad de individuos. En los casos más simples se da la posibilidad de mutar sólo un alelo por individuo o se distribuye uniformemente sobre todo el cromosoma. Cuando se utiliza elitismo es posible asegurar la mejor solución de cada generación lo que permite utilizar probabilidades de mutación más altas. Una revisión comparativa y combinación de diferentes métodos de mutación puede verse en [4].

En el algoritmo de segmentación evolutiva se elige al azar un gen y se lo muta mediante $\Delta M_{j^*}(G+1) = \Delta M_{j^*}(G) + R \sqcup (-1,1)$, donde j^* es el gen elegido para la mutación, G es el número de la generación actual y R es el rango en que se produce la alteración. La función $\sqcup(a,b)$ devuelve un número real al azar entre a y b con una distribución uniforme. Existe un control para que el resultado no salga del rango previsto para los desplazamientos de los marcadores.

La *cruza* es un operador que actúa sobre dos cromosomas para obtener otros dos. Existen dos tipos de cruzas: cruzas simples y cruzas múltiples. En las cruzas simples se elige un punto de cruza al azar y se intercambia el material genético correspondiente a las partes del cromosoma que separa este punto. En la cruza múltiple puede cortarse el cromosoma en más de dos partes para realizar el intercambio. También en este caso los puntos son elegidos al azar. Para el problema de segmentación de voz se utiliza la cruza simple. El punto de cruza se elige al azar pero los dos cromosomas se cortan en el mismo lugar. Esto asegura que la longitud de los cromosomas se mantenga después de la cruza. Sin embargo, sería de interés para aplicaciones de tiempo real poder tener cromosomas con diferentes números de segmentos y así elegir un punto de cruza diferente para cada uno de los dos cromosomas que intervienen. Se presentan más detalles de este algoritmo evolutivo en [14].

3 Algoritmo de segmentación con detector de máximos

La ecuación (1), que mide la distancia euclídea entre dos vectores propios, puede utilizarse como una medida del cambio en los vectores de características a cada lado de un marcador. Si estas distancias no se integran sobre toda la frase como se hizo en la función de aptitud (2), entonces pueden utilizarse como medida de los cambios a nivel *local* para cada tramo de análisis. Se puede esperar que en las posiciones de la frase en donde esta medida sea máxima se encuentren los límites que separan dos estructuras acústicas relevantes. En base a esta idea se desarrolla a continuación un método de segmentación ciega de voz. En este caso no existe un conjunto de

marcadores predefinidos ni se necesita medir la aptitud como en el caso de la segmentación evolutiva. Ahora el conjunto de marcadores surgirá a través de un proceso iterativo de optimización.

3.1 Redefinición de la distancia entre segmentos

Es necesario realizar unos cambios en la definición original, ya que ahora no se poseen marcadores. Para independizar la distancia (1) del contexto es necesario fijar la cantidad de vectores de características que se consideran a cada lado de un tramo de voz dado. Así, se redefinen los nuevos vectores propios para cada t:

$$\varphi_t^{-\Delta M}(k) = \frac{\sum_{\tau=t-1}^{t-\Delta M} \alpha(t-\tau, \Delta M) x(\tau;k)}{\sum_{\tau=t-1}^{t-\Delta M} \alpha(t-\tau, \Delta M)}$$

y

$$\varphi_t^{+\Delta M}(k) = \frac{\sum_{\tau=t}^{t+\Delta M-1} \alpha(\tau-t+1, \Delta M) x(\tau;k)}{\sum_{\tau=t}^{t+\Delta M-1} \alpha(\tau-t+1, \Delta M)}$$

Considerando una relación lineal para $\alpha(\cdot)$, se define la distancia euclídea entre los segmentos en torno al tiempo t y con ancho ΔM:

$$\delta_t^E(\Delta M) = \frac{2}{\Delta M+1} \sum_{k=1}^{N_x} \left[\sum_{\tau=t-1}^{t-\Delta M} \left(1 - \frac{t-\tau}{\Delta M}\right) x(\tau;k) - \right. \\ \left. - \sum_{\tau=t}^{t+\Delta M-1} \left(1 - \frac{\tau-t+1}{\Delta M}\right) x(\tau;k) \right]^2$$

con $\Delta M < t \leq T - \Delta M$.

3.2 Búsqueda de los picos de segmentación

Para segmentar resta definir un algoritmo que detecte los picos de la función $\delta_t^E(\Delta M)$, es decir, aquellos instantes de tiempo en donde se realizan mayores cambios en los vectores de características. La detección de estos máximos se realiza en dos pasos: búsqueda de los candidatos por caída de gradiente y selección de los mejores máximos. El algoritmo consiste en acumular los gradientes que se encuentran a cada lado de un pico y de esta forma medir su importancia relativa. Se comienza considerando que existe un candidato en cada instante de tiempo t de la curva $\delta_t^E(\Delta M)$ y en cada paso elimina aquellos candidatos para los que no se cumpla $\delta_{t-1}^E(\Delta M) < \delta_t^E(\Delta M) > \delta_{t+1}^E(\Delta M)$. Cada vez que un candidato no supera esta prueba se elimina de la lista y se acumula su diferencia con el que sea mayor de los que están a su lado. En la Figura 2 se resume este algoritmo de detección de picos.

```
Comienzo: δ²_t = δ^E_t(ΔM)²; pk_t = 1 ∀t
Repetir
  Para cada t Si pk_t ≠ 0
    Si (δ²_{t+1} ≥ δ²_t) ∧ (δ²_{t+1} − δ²_t > δ²_{t−1} − δ²_t)
      pk_{t+1} = pk_{t+1} + δ²_{t+1} − δ²_t
      pk_t = 0
    Si δ²_{t−1} ≥ δ_t
      pk_{t−1} = pk_{t−1} + δ²_{t−1} − δ²_t
      pk_t = 0
  FinPara
Hasta no observar cambios en pk_t
```

Figura 2: Algoritmo detector de picos de segmentación.

Los candidatos quedan indicados en los elementos $pk_t \neq 0$. La selección definitiva se realiza en dos etapas de filtrado con diferentes tamaños de ventana. En la primera se consideran ventanas de ancho $W_f = 10T_d$ y se eliminan los máximos menores a un 10% del máximo en la ventana. En la segunda etapa se consideran ventanas de una ancho menor ($W_f/2$) y se deja un único máximo por ventana.

4 Resultados

Las pruebas que se realizaron se dividen en tres partes. En primer lugar se presenta un ejemplo que tiende a mostrar las características más importantes del algoritmo de segmentación evolutiva. Este experimento se realiza en base a una señal creada artificialmente con información que resulta en una segmentación obvia. Los segundos experimentos se realizaron en un archivo de voz y se comparan los resultados con la segmentación realizada por MOM. En los últimos experimentos se segmentaron 600 frases.

Para las primeras pruebas se generó un archivo de 1 segundo con las siguientes señales: silencio [0,166) ms; ruido blanco [166, 250) ms; silencio [250,750) ms; seno de 1000 Hz [750,833) ms y silencio [833,1000] ms. En esta señal los segmentos del ruido y la señal senoidal son fácilmente detectables. El algoritmo de segmentación evolutiva se aplicó con los parámetros que se muestran en la Tabla 1, donde se utilizaron vectores de características con coeficientes cepstrales en escala de mel (CCEM). En la Figura 3 se muestra la superficie de aptitud y en la Figura 4 el resultado de la segmentación.

Individuos en la población	10
Rango de alelos en ms	400
Probabilidad de cruzas	0.5
Probabilidad de mutaciones	0.5
Generaciones	500
Elitismo	si
Ancho de la ventana de análisis en ms	8
Paso de la ventana de análisis en ms	8
Tipo de análisis	CCEM

Tabla 1: Parámetros utilizados en el ejemplo de ruido y senoidal.

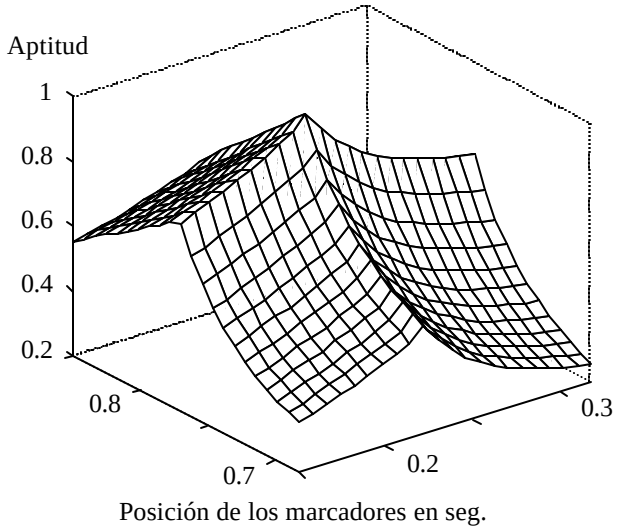

Figura 3: Superficie de aptitud para el ejemplo de ruido y senoidal. Se evoluciona la posición de los dos marcadores centrales (los marcadores de inicio y fin se encuentran fijos).

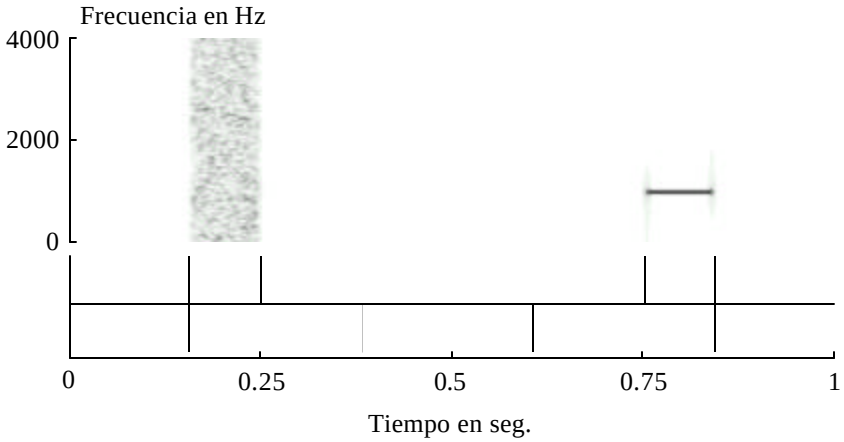

Figura 4: Segmentación obtenida en el ejemplo de ruido y senoidal. En la parte superior se observa el espectrograma de la señal del ejemplo. Las líneas de abajo indican la segmentación lineal, a partir de la cual evolucionan los marcadores. Las líneas de arriba (evol.) indican la segmentación realizada por el algoritmo evolutivo.

En el ejemplo de segmentación de voz se realizaron diversas pruebas con un archivo del corpus de voz Albayzin. Para el primer caso en la segmentación de voz se utilizaron los parámetros de la Tabla 2 y se exigieron tantos segmentos como sílabas tenía la frase.

En la Figura 5, con etiqueta 'evol.1', se observa el resultado de la segmentación por sílabas. Se han realizado varias pruebas en las que la convergencia se obtuvo antes de las 100 generaciones. En la parte inferior de la misma gráfica se indica como referencia la segmentación obtenida con MOM. Esta segmentación se obtiene buscando la secuencia más probable mediante el algoritmo de Viterbi, a partir de la transcripción completa de cada frase.

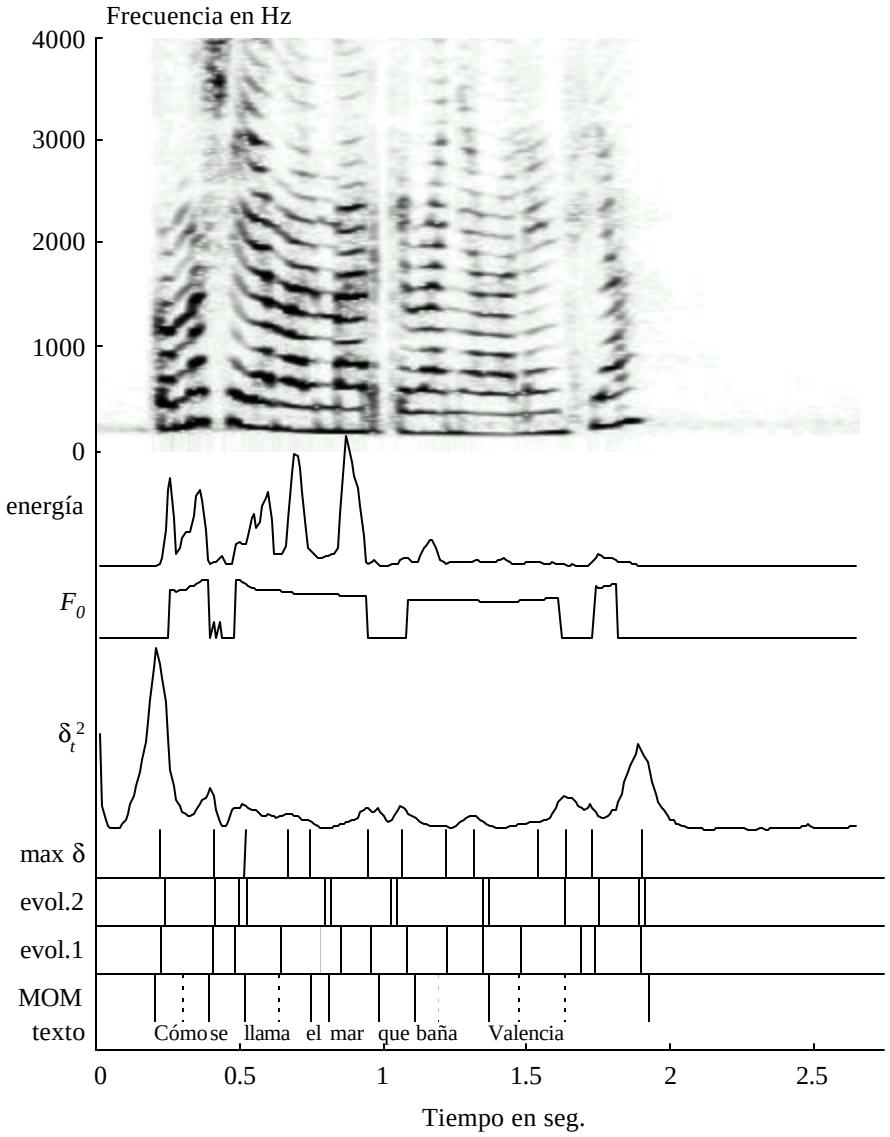

Figura 5: Segmentación de la frase *¿Cómo se llama el mar que baña Valencia?* mediante los diferentes métodos evaluados. En la parte superior se observa el espectrograma y las curvas de energía y frecuencia fundamental. A continuación se puede ver la curva δ_t^2, a partir de la cual se obtiene la segmentación silábica por el algoritmo de detección de máximos, indicada a como 'max δ'. Las etiquetas 'evol.2' y 'evol.1' indican las segmentaciones por palabras y por sílabas obtenidas con el algoritmo evolutivo. En la parte inferior se observa la segmentación obtenida mediante MOM y el etiquetado en palabras correspondiente (con líneas de puntos la segmentación silábica).

Para el segundo caso de segmentación mediante el algoritmo evolutivo se modificó únicamente el rango de los alelos, que se amplió a 250 ms para poder incluir palabras. Sin embargo, se mantuvo la exigencia de cantidad de segmentos de acuerdo con una segmentación silábica. En la Figura 5, con etiqueta 'evol.2', se observa claramente que el método tiende a realizar una segmentación por palabras.

Para las pruebas de segmentación local con detección de máximos se utilizó un subconjunto del corpus de habla Albayzin. Los parámetros utilizados en el algoritmo fueron $\Delta M = 21$ y $W_f = 10T_d$. En la Figura 5 se puede apreciar la curva de δ_t^2 y la segmentación realizada por el algoritmo (con etiqueta 'max δ'). Luego se midió el error sobre las 600 frases, contando las veces que la segmentación resultante coincidía con la segmentación realizada mediante los MOM. Para la segmentación silábica el error promedio fue de 32.36% y para la segmentación de palabras 47.57%. Si se acepta el error de una sílaba por palabra el error promedio se reduce a 7.59%.

Individuos en la población	200
Rango de alelos en ms	100

Probabilidad de cruzas	0.5
Probabilidad de mutaciones	0.5
Generaciones	500
Elitismo	si
Ancho de la ventana de análisis en ms	16
Paso de la ventana de análisis en ms	16
Tipo de análisis	CCEM

Tabla 2: Parámetros utilizados en el primer ejemplo con una señal de voz.

5 Discusión y conclusiones

Los primeros resultados muestran que el silencio y la senoidal son segmentados fácilmente, con muy poca carga computacional, y una población mínima que hasta es inusual en métodos de computación evolutiva. La utilización del elitismo permitió elegir una alta probabilidad de mutaciones acelerando la convergencia (sin llegar a una búsqueda al azar). También se puede observar en la curva de evolución que el tiempo total podría ser reducido casi a la mitad. Vale destacar que se podría reducir el análisis de la señal a simplemente el cálculo de la energía por ventanas. El rango de los alelos (400 ms) fue fijado en base a cuánto se tiene que poder desviar el marcador de la segmentación lineal para poder realizar la segmentación ideal.

En el caso de la segmentación por sílabas los marcadores encontrados por el método de segmentación evolutiva coinciden casi exactamente (considerando las ventanas de análisis utilizadas) con los marcadores de la segmentación por MOM (Figura 5, etiqueta 'evol.1'). Sin embargo, se puede ver que existe un error por omisión en la primera sílaba y uno por inserción en la penúltima. El primer error puede ser debido a que la emisión de la palabra *cómo* tiene el mismo fonema /o/ en cada sílaba. Además está separado por una /m/, que ofrece una transición suave de las formantes de los sonidos vocálicos de su entorno, que en este caso son iguales. El error en el anteúltimo marcador puede responder a varias causas. En primer lugar debe considerarse que el rango elegido para los alelos apenas alcanza para abarcar a la sílaba /cia/. Por otro lado se puede observar que dado el error de omisión en la primera sílaba el método queda forzado a insertar un marcador (ya que la cantidad total de marcadores es fija). En la misma línea de razonamiento, se puede ver que la pausa que se produce entre /len/ y /cia/ determina una fuerte diferencia entre estas regiones y el método encuentra que la función de aptitud se maximiza haciendo esta separación a costa de unir la palabra *cómo*. En la segmentación 'evol.2', si bien el rango de los alelos es mucho mayor (250 ms), aún se observa el error en *Valencia*.

En la segmentación por palabras (Figura 5, etiqueta 'evol.2') se puede ver la forma en que la elección del rango de los alelos condiciona fuertemente los resultados. Esto permitiría seleccionar el rango de los alelos a partir de las longitudes típicas de las unidades a segmentar. Sin embargo, puede que esto no sea tan obvio en el caso del habla. Existen palabras que pueden tener la longitud de tan sólo una sílaba o fonema y, de la misma forma, algunas sílabas pueden tener la longitud de toda una palabra. Este puede ser el punto más débil del método dado que no utiliza otra información relativa al contexto o a la gramática, como en el caso de los MOM. Otro aspecto que puede constituir una desventaja es el tiempo total necesario para realizar la segmentación. Para dar una idea de estos tiempos, para segmentar un frase de 3.5 segundos en un procesador Pentium Celeron 366 MHz se necesitaron 17.2 segundos. Esta podría ser una limitación importante para un sistema de tiempo real, pero no invalida la aplicación del método a la segmentación de corpus de habla.

Al igual que el rango de los alelos controla el tipo de segmentación en el algoritmo evolutivo, los parámetros ΔM y W_f lo hacen con el método por detección de máximos. En este caso (Figura 5, etiqueta 'max δ') se puede observar que nuevamente no se detecta la separación silábica de la palabra *cómo* y se agrega un marcador extra en la palabra *Valencia*. Hay que destacar que en este método no es necesario conocer a priori la cantidad total de marcadores. De los 12 marcadores de la segmentación por MOM, el método por detección de máximos ha encontrado 11 (sin contar el primero y el último de la frase). Esto abre la posibilidad de combinar ambos métodos, uno para la detección de los extremos de la frase y la cantidad de sílabas, y el otro para la segmentación propiamente dicha.

Cuando se realizaron pruebas con los coeficientes espectrales (CE) se observó que la energía condicionaba fuertemente la posición de los marcadores. En este caso las marcas se ubicaron en las máximas variaciones de energía, no segmentando sílabas sino más bien vocales. Esta influencia de la energía también se observa, aunque en menor medida, para los CCEM. Esto último podría dar lugar a una revisión del algoritmo para obtener una normalización por energías que anule este efecto indeseado. Las pruebas realizadas con coeficientes de predicción lineal (CPL) no difieren mucho de las realizadas con CCEM pero el cálculo de los CPL es algo más lento.

Por último, cabe mencionar una particularidad de los métodos propuestos: en ningún caso hay un proceso de entrenamiento ni parámetros almacenados para su posterior utilización durante la segmentación. Esto, si bien hace que los métodos trabajen con muy poca información de la tarea a realizar, también les da robustez, flexibilidad y aprovecha al máximo su capacidad de autoadaptación.

6 Referencias

[1] Bäck, T., Hammel, U., y Schewfel, H.-F. "Evolutionary computation: Comments on history and current state". *IEEE Trans. on Evolutionary Computation*, volumen 1, número 1, páginas 3–17, 1997.

[2] Bhandarkar, S. M. y Zhang, H. "Image segmentation using evolutionary computation". *IEEE Trans. on Evolutionary Computation*, volumen 3, número 1, 1999.

[3] Brugnara, F., Falavigna, D., y Omologo, M. "Automatic segmentation and labeling of speech based on hidden Markov models". *Speech Communication*, volumen 12, número 4, páginas 357–370, 1993.

[4] Chellapilla, K. "Combining mutation operators in evolutionary programming". *IEEE Trans. on Evolutionary Computation*, volumen 2, número 3, 1998.

[5] Gallwitz, F., Batliner, A., Buckow, J., Huber, R., Niemann, H., y Nöth, E. "Integrated recognition of words and phrase boundaries". En *Proceedings of 5th International Conference on Spoken Language Processsing*, páginas 328–331, Sydney, 1998.

[6] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1997.

[7] Hemert, J. V. "Automatic segmentation of speech". *IEEE Trans. on Signal Processing*, volumen 39, número 4, páginas 1008–1012, 1991.

[8] Jeong, C. y Jeong, H. "Automatic phone segmentation and labelling of continuous speech". *Speech Communication*, volumen 20, páginas 291–311, 1996.

[9] Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[10] Lee, T. y Ching, P. C. "Cantonese syllable recognition using neural networks". *IEEE Trans. on Speech and Audio Processing*, volumen 7, número 4, páginas 466–472, 1999.

[11] Li, T.-H. y Gibson, J. D. "Speech analysis and segmentation by parametric filtering". *IEEE Trans. on Speech and Audio Processing*, volumen 4, número 3, 1996.

[12] Merelo, J. J., Carpio, J., Castillo, P., Rivas, V. M., Romero, G., y Schoenauer, M. "Evolving objects". En *Third International Workshop on Frontiers in Evolutionary Algorithms*, Atlantic City, 2000.

[13] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.

[14] Milone, D. H., Merelo, J. J., y Rufiner, H. L. "Evolutionary algorithm for speech segmentation". En *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, páginas 741–744, Honolulu, HI. Paper No. 7270, 2002.

[15] Pauws, S., Kamp, Y., y Willens, L. "A hierarchical method of automatic segmentation for synthesis applications". *Speech Communication*, volumen 19, páginas 207–220, 1996.

[16] Reddy, D. R. "An approach to computer speech recognition by direct analysis of the speech wave". Reporte técnico CS59, Computer Science Department, Stanford University, 1966.

[17] Salomon, R. "Evolutionary algorithms and gradient search: Similarities and differences". *IEEE Trans. on Evolutionary Computation*, volumen 2, número 2, páginas 45–55, 1998.

[18] Svendsen, T. y Soong, F. K. "On the automatic segmentation of speech signals". En *Proceedings of the IEEE International Conference on Acoustic and Signal Processing*, volumen 1, páginas 77–80, Dallas, Texas, 1987.

[19] Vorstermans, A., Martens, J.-P., y Van Coile, B. "Automatic segmentation and labelling of multi-lingual speech data". *Speech Communication*, volumen 19, páginas 271–293, 1996.