Estimación de estructuras acentuales

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

A. J. Rubio Ayuso
Departamento de Electrónica y Tecnología de Computadores,
Universidad de Granada, España

En estudios anteriores se ha demostrado que no existe un método directo para obtener la acentuación de las palabras de una frase a partir de los rasgos prosódicos (energía, frecuencia fundamental y duración del núcleo vocálico). Debido a que en el discurso continuo se pierde de manera significativa la relación entre acentuación y rasgos prosódicos, es necesario utilizar técnicas más sofisticadas que puedan extraer relaciones complejas entre los datos. En este sentido es necesario cubrir dos aspectos importantes del problema: las características locales de los segmentos de voz y sus dinámicas a lo largo de una frase.
En la primera parte de este trabajo se describirán distintas alternativas para la clasificación de estructuras acentuales. Se utilizan métodos de cuantización vectorial con aprendizaje y árboles de redes neuronales para encontrar las estructuras acentuales a partir de los rasgos prosódicos, sobre la base de una segmentación silábica conocida. En la segunda parte del trabajo se describe un método que ataca los dos aspectos del problema en forma conjunta. Este método, basado en modelos ocultos de Markov, realiza la segmentación y clasificación de estructuras acentuales simultáneamente.
Los resultados de clasificación obtenidos con árboles de redes neuronales superan ampliamente a los alcanzados con cuantización vectorial con aprendizaje. Los mejores resultados obtenidos con modelos ocultos de Markov se lograron modelando explícitamente las tonicidades silábicas (atónica o tónica) y teniendo en cuenta las palabras que se consideran inacentuadas en el discurso continuo. Los experimentos en donde se hizo distinción entre los diferentes núcleos vocálicos arrojaron resultados algo inferiores pero debe considerarse que estos modelos proporcionan más información para etapas posteriores de clasificación.

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & A. J. Rubio Ayuso; "Estimación de estructuras acentuales"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 70, 2003.

1 Introducción

En esta sección se hará una introducción a los sistemas de clasificación estática que pueden utilizarse para estimar la estructura acentual (EA) de una palabra a partir de la información prosódica de cada sílaba.

Los *árboles de decisión* (AD) y las *redes neuronales artificiales* (RNA) son dos técnicas ampliamente utilizadas para la clasificación de patrones. Los AD generan un conjunto de particiones en el espacio de entrada basándose en una estructura jerárquica de nodos en los que se realizan comparaciones sobre alguna componente del vector de características. Las redes neuronales están formadas por un conjunto de unidades de procesamiento no lineal altamente interconectadas, que procesan en paralelo un conjunto de datos para extraer información. Existen diferentes modelos neuronales para la implementación de clasificadores supervisados y no supervisados. El perceptrón multicapa es un ejemplo clásico de clasificador simple supervisado, mientras que los *mapas autoorganizativos* (MAO) son ejemplos de clasificadores simples no supervisados [1][6].

1.1 Mapas autoorganizativos

Las áreas del cerebro, especialmente de la corteza cerebral, se hallan organizadas según diferentes modalidades sensoriales. Esta organización de la actividad cortical del cerebro puede describirse mediante mapas ordenados. Por ejemplo, se encuentran los mapas retinoscópicos de la corteza visual, los mapas tonotópicos de la corteza auditiva, los mapas somatotópicos de la corteza somatosensorial y los mapas de retardo interaural. Inspirado en el mapeo ordenado del cerebro, Kohonen introdujo en 1982 un algoritmo de autoorganización para producir mapas ordenados que simulan cortezas biológicas simplificadas, con el objeto de resolver problemas prácticos de clasificación y reconocimiento de patrones [7]. Los MAO presentan la propiedad de preservación de la vecindad, que los distingue de otros paradigmas de RNA. Estas arquitecturas son entrenadas mediante aprendizaje

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

competitivo, es decir, las neuronas compiten entre ellas para ser activadas, dando como resultado la activación de una sola a la vez. Esta neurona es llamada *neurona ganadora* y a diferencia de otras RNA donde sólo se permite que aprenda la unidad ganadora, en los MAO todas las unidades vecinas a la ganadora reciben una realimentación procedente de la misma, participando de esta manera en el proceso de aprendizaje.

En la configuración básica de un MAO se observan las neuronas de entrada e_i y una red bidimensional de neuronas de salida s_j. Un peso sináptico w_{ij} conecta a la neurona e_i con la s_j. A cada neurona de entrada e_i se le presenta el i-ésimo elemento de cada patrón de entrada $\mathbf{x}(n) \in \mathbb{R}^D$, siendo n la ocurrencia temporal de este patrón. El arreglo bidimensional de neuronas de salida incluye conexiones entre las neuronas vecinas simulando la realimentación lateral. Si G es una neurona ganadora durante el entrenamiento de un MAO, las neuronas vecinas que también serán actualizadas quedan en una región determinada por una función de vecindad $\Lambda_G(n)$. Esta región puede tener diferentes formas y es variable con n. El área cubierta comienza siendo máxima y se reduce a medida que avanza el entrenamiento hasta no incluir ninguna neurona vecina a la ganadora. La ecuación básica de adaptación es:

$$\mathbf{w}_j(n+1) = \begin{cases} \mathbf{w}_j(n) + \eta(n)[\mathbf{x}(n) - \mathbf{w}_j(n)] & \text{si } s_j \in \Lambda_G(n) \\ \mathbf{w}_j(n) & \text{en otro caso} \end{cases}$$

Para más detalles véase [5].

1.2 Cuantización vectorial con aprendizaje

La cuantización vectorial surge originalmente como un método de compresión, pero también puede ser interpretada como un proceso de clasificación. En la cuantización vectorial se intenta extraer la estructura subyacente a un grupo de patrones para dividir el espacio de entrada en un número finito de regiones y asociar a cada una de ellas un vector característico o centroide. Cada uno de estos centroides está asociado a una etiqueta o número de índice y de esta forma se *cuantiza* la información contenida en los vectores de entrada. En particular, la cuantización vectorial con aprendizaje (CVA) es una técnica que se puede utilizar para ajustar la posición de los centroides y mejorar el rendimiento de un clasificador en las fronteras de las regiones de decisión.

Existen diferentes versiones del algoritmo de CVA en base a una misma idea central. A partir de una apropiada configuración inicial, el algoritmo CVA1 consiste simplemente en acercar o alejar un centroide al patrón de entrada de acuerdo a si fue bien o mal clasificado, respectivamente. La ecuación de adaptación que utiliza el algoritmo es: $\mathbf{w}_c(n+1) = \mathbf{w}_c(n) + s(n)\eta(n)[\mathbf{x}(n) - \mathbf{w}_c(n)]$, donde se define:

$$s(n) = \begin{cases} +1 & \text{si } x^c(n) = c \\ -1 & \text{si } x^c(n) \neq c \end{cases}$$

Una optimización para este algoritmo consiste en la adecuada selección de la función de variación para la velocidad de aprendizaje mediante:

$$\eta_c(n) = \frac{\eta_c(n-1)}{1 + s(n)\eta_c(n-1)}$$

El método resultante se denomina CVA1 optimizado (CVA1-O) [6].

1.3 Inducción de reglas mediante árboles de decisión

En este paradigma el algoritmo de aprendizaje busca una colección de reglas que clasifican “mejor” los ejemplos de entrenamiento y se puedan representar como un AD. Estas estructuras pueden pensarse como diagramas de flujo en donde cada nodo representa una prueba y cada rama que sale del nodo representa un resultado posible a dicha prueba. Para una revisión más detallada se puede consultar [2].

Existen AD binarios y n-arios, de acuerdo a la cantidad de particiones realizadas en cada nodo. Dependiendo de las características de la función del nodo y del tamaño del árbol, la frontera final de decisión puede ser muy compleja. Una de las funciones más empleadas es la prueba mediante un cierto umbral para cada atributo, teniendo como resultado la partición del espacio de atributos por medio de hiperplanos paralelos u ortogonales a los ejes coordenados del espacio de atributos.

Dos de los algoritmos de aprendizaje más utilizados son ID3 y CART [12]. El algoritmo ID3 genera AD n-arios debido a que particiona el conjunto de ejemplos de entrenamiento en función del mejor atributo. La función heurística que utiliza ID3 para determinar el mejor atributo es una medida de la entropía para cada atributo. El algoritmo CART genera AD binarios ya que para particionar el conjunto de ejemplos en un nodo elige el mejor par atributo-valor de acuerdo con el denominado *criterio de Gini* [15].

Aunque los AD son intuitivamente atractivos y han tenido aplicaciones exitosas, existen algunos problemas que pueden obstaculizar su empleo en casos reales. Entre estos problemas se pueden mencionar la presencia de datos inconclusos, incompletos o ruidosos y el hecho de que raramente se aprovechan en simultáneo todos atributos de los vectores de entrada.

2 Estimación mediante árboles de redes neuronales autoorganizativas

Para solucionar algunos de los problemas que se presentan con los AD, una alternativa consiste en la implementación híbrida de AD y RNA. Este tipo de enfoque permite aprovechar las ventajas de la clasificación jerárquica y crear fronteras de decisión más complejas con menos nodos, minimizando los problemas de ruido y de estructuras intrincadas. Los árboles de redes neuronales (ARN) son AD que implementan la tarea de decisión en los nodos mediante una red neuronal. De esta manera la decisión que se toma en cada nodo se basa en reglas más complejas, lo que permite aproximar mejor las fronteras a costa de perder claridad en la interpretación de las reglas resultantes.

La cantidad de particiones que se producen en cada nodo puede ser fija o variable. Cuando la cantidad de clases generadas puede variar para cada nodo, el ARN tiene la posibilidad de adoptar una configuración más adecuada para el problema a resolver. Los ARN realizan una clasificación basada en una combinación de los métodos de clasificación simple y jerárquica. Si se utiliza un MAO en cada uno de los nodos del ARN se aprovecha también el hecho de que estas redes de entrenamiento no supervisado pueden separar los patrones de acuerdo a su distribución natural.

El algoritmo propuesto permite que en las primeras capas o nodos se separen los grupos de patrones más alejados entre sí (o más fácilmente separables) y en las capas finales se haga una separación más fina de los patrones (es decir, los más difícilmente separables). En el caso de árboles n-arios, un problema importante es cómo decidir acerca de la cantidad de particiones a realizar en cada nodo. Para atacar este problema se establecieron criterios basados en los coeficientes de clasificación que se describen a continuación.

2.1 Coeficientes de clasificación

Dado un clasificador general se define el conjunto de patrones de entrada como $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P\}$ con $\mathbf{x}_i \in \mathbb{R}^D$. Los patrones de X pueden ser agrupados en M clases de entrada C_i^I. De la misma forma en que los patrones de entrada se agrupan según las clases a las que pertenecen realmente, también se pueden agrupar de acuerdo a las clases C_j^O en que son separados por el clasificador. Estas últimas forman el conjunto de clases de salida $C^O = \{C_1^O, C_2^O, \ldots, C_N^O\}$.

Debe observarse que, en el caso más general, la cantidad de clases de entrada M no necesariamente debe ser igual a la cantidad de clases de salida N. Esta generalización resulta muy útil cuando el proceso de clasificación se realiza mediante clasificaciones sucesivas. En estas etapas intermedias en general $M \geq N$. No obstante, en el clasificador visto como un solo conjunto generalmente se tiene $M \leq N$.

Una definición importante para el desarrollo posterior es la *matriz de intersección de entrada-salida*:

$$N_{ij}^{IO} = \left|(C_i^I \cap C_j^O)\right|; \quad 1 \leq i \leq M; \quad 1 \leq j \leq N.$$

donde $|\cdot|$ es el operador de cardinalidad. Esta matriz contiene en su i,j-ésima celda la cantidad de patrones de la clase de entrada C_i^I clasificados como pertenecientes a la clase de salida C_j^O.

A continuación se analizan las limitaciones en la utilización del coeficiente de reconocimiento clásico como criterio para el desarrollo de la topología de un ARN.

2.1.1 Coeficiente de reconocimiento clásico

El coeficiente de reconocimiento que se utiliza generalmente para medir el rendimiento de un clasificador en el reconocimiento de patrones se puede definir según:

$$cr = \frac{\sum_{i=1}^{M} \max_{j=1}^{N} \left(N_{ij}^{IO}\right)}{|X|}$$

Este coeficiente tiene algunas propiedades que suelen hacer confusa su interpretación. El máximo que puede alcanzar cr es 1 cuando $\forall i \exists j / N_{ij}^{IO} = |C_i^I|$. Sin embargo, el mínimo que puede alcanzar cr no es cero ya que éste depende del número de clases de salida M. Cuando el clasificador se encuentra en un máximo de confusión, distribuye igualmente cada clase de entrada en las clases de salida. Por lo tanto el mínimo para cr es $1/M$, ya que

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & A. J. Rubio Ayuso; "Estimación de estructuras acentuales"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 70, 2003.

el máximo en cualquier clase de salida es $|C_i^I|/M$. Esto es particularmente confuso ya que un clasificador con dos clases de salida no podría tener nunca un $cr \leq 0{,}5$ (rendimiento menor al 50%).

Por otro lado, este coeficiente de reconocimiento no es aplicable cuando $M \neq N$. Además, tampoco lo es cuando se hacen agrupaciones intermedias de varias clases de entrada en una clase de salida para ser luego separadas por otro clasificador. El coeficiente no permite discernir en qué medida patrones de las mismas clases de entrada son concentrados en la misma clase de salida y patrones de distintas clases de entrada son distribuidos en distintas clases de salida. Para poder eliminar estas restricciones se definen dos coeficientes que miden estas concentraciones y dispersiones por separado.

2.1.2 Coeficiente de concentración interclase

Para medir en qué grado un clasificador agrupa patrones pertenecientes a una clase de entrada en una misma clase de salida se define el coeficiente de concentración interclase:

$$cc = \frac{\sum_{i=1}^{M} N \max_{j=1}^{N}\left(N_{ij}^{IO}\right) - \sum_{i=1}^{M}\sum_{j=1}^{N} N_{ij}^{IO}}{(N-1)\sum_{i=1}^{M}\sum_{j=1}^{N} N_{ij}^{IO}}$$

donde $\sum_i \sum_j N_{ij}^{IO} = |X| \neq 0$.

Se puede deducir que $cr = cc(M-1)/M + 1/M$ y así, cuando los máximos de cada clase de entrada se encuentran en distintas clases de salida y la cantidad de clases de salida es igual a la de clases de entrada, el coeficiente de reconocimiento puede expresarse como una versión escalada y desplazada del coeficiente de concentración intraclase.

Hay que destacar que, si bien el coeficiente de concentración mide la capacidad con que un clasificador agrupa patrones de una misma clase de entrada en una única clase de salida, no es capaz de detectar cuando todos los patrones de entrada son llevados a una misma clase de salida. Para esto se define a continuación el coeficiente de dispersión intraclase.

2.1.3 Coeficiente de dispersión intraclase

Para medir la capacidad que posee un clasificador para llevar patrones de distintas clases de entrada a distintas clases de salida se define el coeficiente de dispersión intraclase:

$$cd = \frac{\sum_{j=1}^{N} M \max_{i=1}^{M}\left(N_{ij}^{IO}\right) - \sum_{j=1}^{j=N}\sum_{i=1}^{i=M} N_{ij}^{IO}}{(M-1)\sum_{j=1}^{j=N}\sum_{i=1}^{i=M} N_{ij}^{IO}}$$

donde nuevamente $\sum_j \sum_i N_{ij}^{IO} = |X|$.

2.2 Algoritmo de entrenamiento

El algoritmo de entrenamiento tiene por finalidad encontrar la estructura del AD y entrenar cada uno de los nodos clasificadores. La totalidad de los patrones de entrenamiento se presenta inicialmente al nodo que se encuentra en la raíz del árbol y a los nodos de los niveles siguientes les llega un subconjunto de patrones que ha sido derivado jerárquicamente de abajo (raíz) hacia arriba (hojas).

Considerando un nodo en particular se debe decidir, en primer lugar, si se justifica o no realizar una tarea de clasificación. Así se distingue entre dos tipos de nodos: nodos clasificadores y nodos terminales. Para declarar que un nodo es terminal o clasificador se deben tener en cuenta dos características de su conjunto de patrones de entrada: el grado de homogeneidad en clases y el número de patrones que posee. Si bien esta última característica no presenta ninguna dificultad en cuanto a su medición objetiva la medida de la homogeneidad en clases no es tan trivial. Por esta razón se define el coeficiente de concentración para el conjunto de patrones de entrada como:

$$pc = \frac{M \max_{i=1}^{M}(|C_i|) - |X|}{(M-1)|X|}$$

Para determinar el tipo de nodo en base a las características mencionadas se comparan sus medidas con dos umbrales: el umbral de concentración mínima de patrones de entrada (u_{pc}) y el umbral de cantidad mínima de

patrones de entrada (u_X). Si se encuentra un nodo clasificador entonces debe entrenarse el MAO correspondiente. La dimensión de entrada en esta red está determinada por la dimensión de los patrones y es la misma para todo el árbol. La dimensión o cantidad de clases de salida junto con los nodos terminales definen la topología final del árbol.

Para determinar la cantidad apropiada de clases de salida se utiliza un proceso de crecimiento de nodo basado en los coeficientes cc y cd y dos umbrales de capacidad de clasificación mínima u_{cc} y u_{cd}. Se adopta inicialmente una configuración con dos clases de salida ($N = 2$), se entrena la red y se evalúa su rendimiento en la clasificación. En el caso en que no se supere alguno de los umbrales se incrementa N en uno y se repite el entrenamiento y prueba. Este proceso culmina cuando ambos coeficientes superan sus correspondientes umbrales o cuando N alcanza el máximo permitido N_{max}. En este último caso, se elige la mejor de todas las configuraciones entre 2 y N_{max} y se considera concluido el entrenamiento de ese nodo. Este algoritmo de crecimiento de nodo se repite para todos los nodos de cada nivel del árbol.

Cuando el árbol ha sido entrenado se procede al etiquetado de los nodos terminales. La elección de la etiqueta asignada a cada nodo terminal se realiza en base al máximo de la matriz N_{ij}^{IO} del nodo clasificador que le dio origen. Luego, los nodos terminales se unen —de acuerdo a su etiqueta— en otro nivel de nodos artificiales que poseen las etiquetas de todas las clases. De esta forma en el ARN en su conjunto cumple $M = N$.

2.3 Funcionamiento del ARN entrenado

Para realizar la clasificación de un patrón se necesita propagarlo a través del ARN. La propagación del patrón puede realizarse en forma secuencial o en forma paralela. Cuando se propaga un patrón en forma secuencial se describe un camino a través del árbol mediante un simple algoritmo: se comienza por el nodo raíz, se miden las distancias del patrón a cada uno de los centroides del MAO correspondiente y se elige como nodo siguiente aquel indicado por el centroide que está más cerca del patrón. Los dos últimos pasos se repiten hasta que se llega a un nodo terminal y se clasifica al patrón según la etiqueta de este último nodo.

En la propagación paralela se miden las distancias entre el patrón y todos los centroides del ARN simultáneamente y luego se sigue el camino formado por los nodos activados a partir del nodo raíz, hasta llegar a un nodo terminal. En [8] se pueden encontrar más detalles acerca de los ARN y un conjunto de experimentos con baterías de prueba de dominio público. Todos estos experimentos se contrastan con otros clasificadores mostrando las ventajas del método. En [9] se presentan pruebas para el reconocimiento de fonemas.

3 Estimación mediante modelos ocultos de Markov

Para buscar una solución integrada, que permita obtener tanto la segmentación como la clasificación, se realizaron diferentes pruebas de estimación de secuencias de estructuras acentuales (SEA) mediante modelos ocultos de Markov (MOM). Las alternativas investigadas se implementan mediante cambios en los tres niveles de un MOM: procesamiento de la señal, modelado acústico y modelado del lenguaje.

3.1 Alternativas en el procesamiento de la señal

Para el procesamiento de la señal es necesario redefinir el vector $\mathbf{x}_t$, resultado del 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. Estos vectores forman las evidencias acústicas que el MOM modela mediante las mezclas de N_c gaussianas en $\mathbb{R}^{N_x}$. En esta sección se describen algunas de las alternativas evaluadas para $\mathcal{T}(k)$.

3.1.1 Energía y frecuencia fundamental

En este caso se define $\mathbf{x}_t = [e(t), F_0(t)]$, donde:

$$e(t) = \log \sum_{n=1}^{N_v} v(t;n)^2$$

La $F_0(t)$ se calcula en base al cepstrum real, en base al algoritmo de Noll [10]. En el caso de completarse el vector con coeficientes delta y aceleración:

$$\mathbf{x}_t = [e(t), F_0(t), \Delta e(t), \Delta F_0(t), \Delta^2 e(t), \Delta^2 F_0(t)].$$

3.1.2 *Curvas de diferencia por ajuste*

Se incorporaron diversos procesamientos alternativos para la F_0. El primer paso fue considerar un ajuste de la curva de entonación mediante polinomios de grado variable entre 3 y 15. Los coeficientes para estos polinomios fueron calculados en base al método de cuadrados mínimos generalizado, resuelto por descomposición en valores singulares [11]. Una vez obtenido el polinomio de interpolación, se resta a la curva de entonación original y se utiliza la curva resultante como otra evidencia para los MOM. Esta curva resultante fue denominada *diferencia de entonación por ajuste* (difF_0). En este caso el vector de evidencias acústicas para los MOM queda definido como: $\mathbf{x}_t = [\epsilon(t), \mathrm{dif}F_0(t)]$.

Este análisis de diferencia por ajuste se hizo extensivo a la curva de energía y se realizaron pruebas con *diferencia de energía por ajuste* (difϵ). También se probaron polinomios con grados que iban desde 3 hasta 15. Para completar la descripción, el vector de evidencias acústicas para los MOM queda definido según: $\mathbf{x}_t = [\mathrm{dif}\epsilon(t), \mathrm{dif}F_0(t)]$ aunque también se realizaron experimentos con $\mathbf{x}_t = [\mathrm{dif}\epsilon(t), F_0(t)]$.

3.1.3 *Otras alternativas evaluadas*

Resta por mencionar la utilización de coeficientes cepstrales en escala de mel (CCEM), tal como se utilizan normalmente en el contexto del reconocimiento automático del habla (RAH) [3]. Dado que las unidades elementales a reconocer en esta aplicación de MOM tienen una longitud mayor (generalmente la de una sílaba), también se experimentó con la variación del ancho (T_ω) y el paso (T_d) de la ventana de análisis. El ancho de ventana fue extendido desde 25 hasta 40, 64 y 100 ms. El paso de análisis fue extendido desde 10 hasta 20, 25 y 50 ms.

3.2 Alternativas en el modelado acústico

En el caso del modelo acústico (MA) se probaron diversas alternativas que pueden separarse en dos grupos: las relacionadas con lo que se modela y las relacionadas con cómo se modela.

3.2.1 *Alternativas en el objeto de modelado*

En los MOM utilizados para el RAH, las unidades elementales eran los modelos de fonemas ${}^{F}\Theta_{\varphi}$ y con éstos se construían por concatenación los modelos palabras ${}^{W}\Theta_{w}$. Para utilizar los MOM en la estimación de SEA es necesario reemplazar el modelo de fonema por un modelo de tonicidad silábica. De la concatenación de modelos de tonicidad silábica se obtienen las EA que, en la organización estructural del habla, poseen un nivel jerárquico equivalente al de las palabras.

Las dos alternativas básicas para el modelo de tonicidad silábica son la /A/ para las sílabas átonas y la /T/ para las sílabas tónicas: $\mathcal{F}_\Theta = \{{}^{F}\Theta_{\mathrm{A}}, {}^{F}\Theta_{\mathrm{T}}\}$. En base a estos dos modelos se pueden construir todas las EA del corpus de habla utilizado. Por ejemplo: ${}^{W}\Theta_{ATA} = {}^{F}\Theta_{\mathrm{A}}{}^{F}\Theta_{\mathrm{T}}{}^{F}\Theta_{\mathrm{A}}$, donde se ha obviado la definición de un diccionario ya que su estructura es trivial. Adicionalmente, para algunos experimentos se definió un modelo especial para las palabras monosilábicas ${}^{W}\Theta_{\mathrm{M}} = {}^{F}\Theta_{\mathrm{M}}$ (modelos TAM). Para otros se clasificaron las palabras como acentuadas e inacentuadas, utilizando todos los modelos /A/ en estas últimas [13] (modelos TA-Q).

En busca de ampliar la estructura de los MA también se realizaron pruebas en donde se formaron modelos para cada una de las vocales y diptongos del núcleo silábico con cada tonicidad. Para estos experimentos se formaron 31 modelos elementales distinguiendo en cada caso:

- la vocal o diptongo que forma el núcleo: /a/, /e/, . . ., /ai/, /ie/, . . .
- su tonicidad: /T/ y /A/.

Los mejores resultados para estos experimentos se resumen luego, en la sección de resultados, bajo la denominación TA-v.

3.2.2 *Alternativas en los parámetros del modelo*

En relación a la estructura de los MOM se probaron diferentes configuraciones que incluían variaciones en:

- Cantidad de estados: entre 3 y 15.

- Tipo de MOM: continuos, semicontinuos o discretos[1] .
- Cantidad de gaussianas (N_c) en la mezclas que modelan las observaciones en cada estado: 1, 2 y 4.

3.3 Alternativas en el modelo de lenguaje

Una vez formadas las EA, con cualquiera de las alternativas mencionadas en la sección anterior, se pueden formar modelos de n-gramáticas e incorporar estas probabilidades en el modelo compuesto. En los experimentos realizados se utilizaron siempre modelos de bi-gramáticas con probabilidades estimadas por el método de *back-off* [4].

En torno a esta estructura básica se consideraron dos variantes:

- Modelos de tonicidad silábica con distinción entre vocales (como en TA-v) pero sin formar EA (a nivel de palabras). En este caso no se concatenan sílabas para formar palabras sino que se trata a las frases como una secuencia de sílabas y a partir de la cual se construye una secuencia de modelos independientes. En estos experimentos, que luego se denominan TA*-v, las probabilidades del modelo de lenguaje (ML) se incorporan directamente a nivel de sílabas.

- Diferentes pesos relativos para las probabilidades de los MA y ML. En el momento de incorporar las probabilidades del ML en la búsqueda por el algoritmo de Viterbi, se multiplica la probabilidad del modelo de lenguaje ($G^{(2)}_{mn}$) por una constante que, en estos experimentos, tomó los valores: 0.01, 0.5, 1.0 y 5.0.

Dado que las restricciones de tiempo no son importantes, todos los experimentos reportados se realizaron sin utilizar el método de podado.

4 Resultados

Para generar los patrones de entrenamiento y prueba se utilizó un subconjunto de frases del corpus de habla Albayzin. Con este subconjunto de 1000 frases se entrenó un sistema de RAH basado en MOM y con las mismas frases se realizó la segmentación[2] buscando la secuencia más probable mediante el algoritmo de Viterbi. Los modelos del sistema de RAH fueron MOM semicontinuos, con 3 estados para los fonemas y el silencio y 1 estado para una pausa corta al final de cada palabra. Las características de la voz utilizadas en este sistema de RAH fueron CCEM con coeficientes de energía y delta (un total de 26 elementos). La ventana de análisis fue de 25 ms y el paso del análisis de 10 ms, con ventana de Hamming [14].

Para cada una de las frases se obtuvieron las EA y las curvas de energía, frecuencia fundamental (F_0) y duración del núcleo vocálico en cada sílaba. A partir de estas frases se generaron 6860 patrones de entrenamiento y 4570 para las pruebas de validación. Cada patrón de entrada se corresponde con una palabra y consiste en un vector con los valores de los rasgos prosódicos para cada una de las sílabas. Dado que los patrones de entrada deben tener dimensión fija, los elementos que están más allá de la cantidad de sílabas de la palabra se hacen cero. Como clase de salida se asigna un código que representa a la EA correcta de cada palabra.

4.1 Resultados con CVA1-O

Debido a que los métodos CVA poseen una topología fija que es definida antes de comenzar el entrenamiento, se han evaluado diversas alternativas con el objetivo de encontrar la estructura con el número de centroides (N_C) más apropiado. Un parámetro que también debe considerarse en el entrenamiento es la cantidad de veces que se ajustan los centroides a partir de los patrones (N_I). La elección de estos parámetros de entrenamiento no es obvia, principalmente porque son muy dependientes de la estructura de los patrones de entrada y la complejidad del problema de clasificación. Sin embargo, se pueden considerar algunos casos extremos como referencia. Por ejemplo, no parece apropiado tener tantos centroides como patrones de entrenamiento. En el otro extremo, salvo para problemas muy simples, no es suficiente con poseer tantos centroides como clases a discriminar. A partir de estos criterios empíricos se ha realizado la búsqueda de los parámetros óptimos y los resultados se muestran en la Tabla 1. Los porcentajes que se observan son el resultado de la utilización de los centroides obtenidos para la clasificación de los patrones del conjunto de prueba.

[1] Sólo para los primeros se reportan resultados.

[2] Solamente necesarios para las pruebas con CVA (sección 4.1) y ARN (sección 4.2)

% $N_I \rightarrow$	100	500	1000	2000	4000	8000	10000
N_C 32	61.82	61.86	61.79	61.79	61.79	61.82	61.84
↓ 64	60.90	64.05	64.79	64.73	64.75	65.38	65.21
128	66.81	67.57	67.64	68.40	68.60	69.10	69.65
256	71.23	71.05	71.42	71.73	73.17	73.13	73.54
512	72.84	72.08	71.36	72.47	73.15	73.41	73.85
1024	72.74	71.29	72.14	72.10	72.84	73.17	73.22
2048	74.25	74.00	73.11	73.00	74.07	74.46	74.49

Tabla 1: Resultados de clasificación de estructuras acentuales mediante cuantización vectorial con aprendizaje. En las columnas se muestran los resultados con diferentes números de iteraciones N_I en el entrenamiento. Las distintas filas indican la cantidad de centroides N_C utilizados en el clasificador.

4.2 Resultados con ARN

En el caso de los ARN la topología se optimiza en el mismo algoritmo de entrenamiento (tanto la estructura interna de cada nodo como la del árbol en su conjunto). Sin embargo, como se vio anteriormente, existen algunos parámetros que regulan el crecimiento del árbol. La forma en que estos parámetros deben variar durante el crecimiento del árbol ha sido analizada y verificada experimentalmente en [8]. Por lo tanto sólo se realizaron dos experimentos con diferente cantidad de iteraciones en el entrenamiento de cada nodo. Los resultados fueron muy similares debido a que en el ARN el resultado final no es tan dependiente del entrenamiento de los nodos como de la estructura de árbol generada. En la Tabla 2 se muestran los parámetros de configuración del algoritmo de entrenamiento y los resultados obtenidos con los datos de prueba. También se muestra en esta tabla el resultado para un experimento donde se incluyó la duración del núcleo vocálico en cada sílaba.

Máxima dimensión de salida	6
Número de iteraciones de entrenamiento de cada nodo	750
Umbral de concentración de patrones (u_{pc}) inicial	0.7
Umbral de concentración de patrones (u_{cc}) final	0.9
Umbral de capacidad de concentración (u_{cc}) inicial	0.3
Umbral de capacidad de concentración (u_{cc}) final	0.9
Umbral de capacidad de dispersión (u_{cd}) inicial	0.8
Umbral de capacidad de dispersión (u_{cc}) final	0.2
Resultado con energía y F_0 (cr)	85.65%
Resultado con energía, F_0 y duración (cr)	89.98%

Tabla 2: Resultados de clasificación de estructuras acentuales mediante árboles de redes neuronales. Se incluyen en esta tabla los parámetros con que fue entrenado el árbol de redes neuronales y los resultados de clasificación sobre el conjunto de prueba. Estos resultados se muestran para un entrenamiento con energía y frecuencia fundamental solamente y con los tres rasgos prosódicos juntos.

4.3 Resultados con MOM

De la amplia lista de combinaciones posibles para las configuraciones presentadas en la Sección 3, se han seleccionado en la Tabla 3 los experimentos con resultados de reconocimiento de EA mayor al 40%. En estos experimentos las frases se separaron en dos grupos, uno para el entrenamiento y el otro para las pruebas de validación (80 y 20% respectivamente). Para ilustrar estos resultados, se transcribe a continuación un ejemplo de la estimación de SEA realizada por el modelo TA-Q:

- Frase: *Ríos de la Comunidad Autónoma Gallega*.
- SEA correcta: /TA A A AAAT ATAA ATA/
- SEA estimada: /T TA A AAAT ATAA A A/

5 Discusión y conclusiones

Cuando se compara el ARN con el mejor caso de CVA1-O se encuentra una diferencia realmente importante a favor del ARN. Además, hay que considerar que habiendo 4570 patrones de prueba el número de 2048 centroides para el CVA1-O es algo excesivo. Si se considera que para el ARN se han utilizado solamente 750 iteraciones en el entrenamiento, sería más razonable compararlo con clasificadores de la región central de la Tabla 1, donde las diferencias a favor del ARN son aún más significativas.

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & A. J. Rubio Ayuso; "Estimación de estructuras acentuales"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 70, 2003.

Una consideración muy importante a la hora de realizar comparaciones entre diferentes arquitecturas es el hecho de que mientras el ARN adapta su topología al problema en cuestión, otros métodos necesitan que se especifique una configuración inicial, generalmente basada en la experiencia del usuario y refinada mediante prueba y error. El ARN adapta su topología localmente a través de múltiples pruebas, como se desprende del algoritmo de crecimiento. Estas pruebas se realizan de manera jerárquica y automática en cada nodo lo que da lugar a un ahorro importante del costo computacional. Hay que destacar que el resultado del algoritmo de crecimiento no es sensible a los umbrales que deben fijarse de antemano. Como regla general, es suficiente con seguir simplemente los alineamientos de la Tabla 2 para asignar el inicio y el fin de cada umbral a lo largo de los niveles.

Procesamiento ($\mathbf{x}_t$)	MA	$\|\mathcal{Q}\|$	N_c	GP	T_d	T_ω	Rendimiento
$[\epsilon, F_0, \Delta, \Delta^2]$	TAM	5	–	–	20	64	45.56%
$[\epsilon, F_0, \Delta, \Delta^2]$	TAM	7	–	–	20	64	53.31%
$[\epsilon, F_0, \Delta, \Delta^2]$	TAM	7	4	–	20	64	55.16%
$[\epsilon, \mathrm{dif}F_0]$	TAM	7	–	11	20	64	56.82%
$[\epsilon, \mathrm{dif}F_0, \Delta]$	TAM	7	–	11	20	64	50.56%
$[\epsilon, \mathrm{dif}F_0, \Delta, \Delta^2]$	TAM	7	–	11	20	64	50.49%
$[\mathrm{dif}\epsilon, \mathrm{dif}F_0]$	TAM	7	–	13	20	64	44.59%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TAM	7	–	–	10	25	53.08%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TAM	4	4	–	10	25	54.88%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TAM	15	4	–	10	25	43.39%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TAM	5	4	–	50	100	53.09%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TAM	7	4	–	25	100	56.94%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TA-Q	7	4	–	25	100	54.41%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TA-v	5	4	–	10	25	50.69%
$[\epsilon, \mathbf{a}]$	TA-v	7	4	–	10	25	50.74%
$[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$	TA*-v	7	4	–	25	100	52.58%

Tabla 3: Resumen de los mejores resultados obtenidos para la estimación de estructuras acentuales con modelos ocultos de Markov. En las columnas se indican: MA: modelos acústicos elementales; $|\mathcal{Q}|$: estados por modelo; N_c: cantidad de gaussianas en la mezcla; GP: grado de los polinomios con que se obtuvo el resultado; T_d: paso en la ventana de análisis en ms; T_ω: ancho en la ventana de análisis en ms; Rendimiento: medido como EA correctamente estimadas en relación a las obtenidas desde la transcripción mediante reglas ortográficas (salvo en el caso TA*-v, donde se cuentan directamente las tonicidades silábicas). En relación con el procesamiento se ha simplificado la notación suprimiendo la t de tiempo: ϵ: energía; F_0: frecuencia fundamental; Δ: coeficientes delta; Δ^2: coeficientes de aceleración; dif: diferencia por ajuste con polinomios de grado 3 a 15; $\mathbf{c}_{mel}$: vector de coeficientes cepstrales en escala de mel; $\mathbf{a}$: vector de coeficientes de predicción lineal; Para los modelos acústicos se ha abreviado: TAM: modelos /T/, /A/ y /M/; TA-Q: modelos /T/ y /A/ con palabras inacentuadas; TA-v: modelos /T/ y /A/ por cada vocal y diptongo; TA*-v: tonicidades silábicas por separado (sin formar EA).

Dado que los cómputos realizados para la generación de un ARN son sencillos, esta arquitectura es considerablemente más veloz que otras estructuras neuronales. La forma jerárquica en que se organiza la información permite que la clasificación de cada patrón de prueba sea sustancialmente más rápida. No se necesitan más de 6 medidas de distancias por nivel[3], mientras que en el método de CVA se requieren tantas medidas de distancia como centroides existan.

La principal fuente de las ventajas de este método está en la combinación de diferentes paradigmas de clasificación. El algoritmo planteado combina las ventajas del aprendizaje no supervisado con las del aprendizaje supervisado. Por un lado, durante el crecimiento y definición de la topología del árbol se utiliza información acerca de la identidad de los patrones. En cambio, para la tarea de clasificación en cada nodo el MAO no usa información acerca de la identidad de los patrones de entrenamiento. Otra de las combinaciones de paradigmas de clasificación que se encuentran en este algoritmo es la de los clasificadores simples y los jerarquizados. Mientras que la estructura general responde a los métodos de clasificación jerarquizada, en cada nodo se utiliza un típico clasificador simple. Debe destacarse el hecho de que estos clasificadores son estáticos y no pueden modelar la información temporal contenida en la señal. De hecho, la segmentación silábica correcta siempre se ha supuesto conocida a priori.

Pero la segmentación automática no es una tarea simple. Sin embargo la principal ventaja del método por MOM es que se realiza en forma conjunta la segmentación y clasificación. Si bien las tasas de reconocimiento son notablemente inferiores a las de los métodos anteriores, el MOM no requiere de una segmentación previa (y por tanto de las transcripciones de las frases).

En la estimación por MOM las consideraciones realizadas en cuanto a la longitud de los segmentos han mostrado sus beneficios con las modificaciones realizadas tanto en el procesamiento de la señal como en los parámetros del modelo. Los mejores resultados se han alcanzado para modelos de 7 estados con T_d y T_ω algo superiores al procesamiento estándar en RAH. Es importante destacar que con un procesamiento sencillo como el de $[\epsilon, \mathrm{dif}F_0]$

[3]Para el ARN utilizado en los experimentos aquí descriptos.

se han logrado rendimientos comparables al de $[\epsilon, \mathbf{c}_{mel}, \Delta, \Delta^2]$, que cuenta con mucha más información en el vector de evidencias acústicas e implica MOM más complejos, con más parámetros y mayor costo computacional. Evidentemente, la eliminación de la función distintiva de F_0 a nivel de frases ha permitido una mejor extracción de la información relativa a la acentuación. Sin embargo, como se podía esperar, no ocurrió lo mismo para el caso de la energía.

Los dos mejores resultados corresponden a los modelos que consideran a los monosílabos por separado, pero el rendimiento del modelo TA-Q no se encuentra muy lejos de estos. Hay que considerar que los resultados del modelo TA-Q proveen más información acerca de las SEA, ya que no sólo se clasifican los monosílabos como tales sino que además se explicita su tonicidad silábica /A/ o /T/ y se contemplan las palabras inacentuadas. En este mismo sentido, el resultado obtenido con los modelos TA-v también proporciona más información útil para una etapa posterior ya que se modelan por separado los diferentes núcleos vocálicos. Sin embargo, en este punto se mezcla en parte el MA tradicional a nivel fonético con el nuevo nivel suprasegmental que se incorpora en este trabajo.

6 Referencias

[1] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press.

[2] Breiman, L., Friedman, J. H., Olshen, R. A., y Stone, C. J. *Classification and Regression Trees*. Wadsworth Int, 1984.

[3] Deller, J. R., Proakis, J. G., y Hansen, J. H. *Discrete-Time Processing of Speech Signals*. Macmillan Publishing, NewYork, 1993.

[4] Jelinek, F. *Statistical Methods for Speech Recognition*. MIT Press, Cambrige, Massachussets, 1999.

[5] Kohonen, T. "The self-organizing map". *Proceedings of the IEEE*, volumen 78, número 9, páginas 1464–1480, 1990.

[6] Kohonen, T. *The Self-Organizing Map*. Springer-Verlag, 1995.

[7] Kohonen, T., Makisara, K., y Saramaki, T. "Phonotopics maps - insightful representation of phonological features for speech recognition". En *Proceedings of the IEEE 7th International Conference on Pattern Recognition*, páginas 182–185, Montreal, Canada, 1984.

[8] Milone, D. H., Sáez, J. C., Simón, G., y Rufiner, H. L. "Árboles de redes neuronales autoorganizativas". *Revista Mexicana de Ingeniería Biomédica*, volumen 19, número 4, páginas 13–26, 1998.

[9] Milone, D. H., Sáez, J. C., Simón, G., y Rufiner, H. L. "Self–organizing neural tree networks". En *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volumen 3, páginas 1348–1351, Hong Kong, 1998.

[10] Noll, A. M. "Cepstrum pitch determination". *Journal of the Acoustic Society of America*, volumen 41, páginas 293–309, 1967.

[11] Press, W., Teukolsky, S., Vetterling, W., y Flannery, B. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2º edición.

[12] Quinlan, J. R. *C4.5: Programs for Machine Learning*. The Morgan Kaufmann Series in Machine Learning, 1993.

[13] Quilis, A. *Tratado de Fonología y Fonética Españolas.* Biblioteca Románica Hispánica. Editorial Gredos, Madrid, 1993.

[14] Rabiner, L. R. y Juang, B. H. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[15] Sestito y Dillon. *Automated Knowledge Acquisition*. Prentice may, 1994.

sinc(*i*) Laboratory for Signals and Computational Intelligence (http://fich.unl.edu.ar/sinc)
D. H. Milone & A. J. Rubio Ayuso; "Estimación de estructuras acentuales"
Memorias del XIV Congreso Argentino de Bioingeniería. pp. 70, 2003.