Reconocimiento Embebido de Habla Aislada Independiente del Hablante en Tiempo Real con HMMs y SVMs

Marufo da Silva, Mariano
mmarufodasilva@est.frba.utn.edu.ar

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Universidad Tecnológica Nacional - Facultad Regional Buenos Aires - Ingeniería Electrónica
Cátedra Proyecto Final: Ing. Sebastian Verrastro, Ing. Pablo Sánchez, Ing. Mariano Vidal

Resumen

Este trabajo describe el diseño de un algoritmo reconocedor de habla utilizando Modelos Ocultos de Markov (HMMs) y Máquinas de Vectores de Soporte (SVMs) como métdos de reconocimiento para su utilización en un sistema de palabras aisladas independiente del hablante.
Se implementó el algoritmo en tiempo real en un microcontrolador de 32 bits ARM Cortex-M4F y se compararon sus resultados con los de un trabajo de investigación previo.

Reconocimiento de Habla Aislada

Representando las características de las palabras pronunciadas a partir de la secuencia de vectores de observaciones O, el objetivo del reconocimiento automático de palabras aisladas es el de resolver:

$$arg \max_{1 \leq i \leq V} \left(P\left(w_i|O\right)\right) \tag{1}$$

donde V es la cantidad de palabras en el diccionario del sistema y cada w_i es la palabra número i del mismo. Aplicando el teorema de Bayes obtenemos:

$$P\left(w_i|O\right) = \frac{P\left(O|w_i\right).P\left(w_i\right)}{P\left(O\right)} \tag{2}$$

Es decir que para un set de probabilidades $P\left(w_i\right)$ iguales, la palabra más probable de haber sido pronunciada depende sólo de $P\left(O|w_i\right)$.

Modelos Ocultos de Markov

El habla puede representarse por un modelo estadístico compuesto por estados con transiciones probabilísticas entre ellos y funciones de densidad de probabilidad de emisión de distintos sonidos. En el reconocimiento de habla basado en HMMs cada modelo acústico (en este trabajo se consideró como modelos acústicos del sistema a las palabras a reconocer) se representa por un HMM.
Un modelo de Markov es una máquina de N estados finita que en cada instante t actualiza su estado, donde la transición del estado i al estado j se da a partir de la probabilidad discreta a_{ij}. Según el estado j en el que se encuentre en cada instante t genera un nuevo vector de observaciones o_t a partir de la densidad de probabilidad de salida $b_j\left(o_t\right)$ (figura 1).
El proceso de reconocimiento consiste en obtener la secuencia de vectores de observaciones O correspondiente a la palabra pronunciada, seguido por un cálculo de las verosimilitudes para cada uno de los modelos $P\left(O|w_i\right)$ para $1 \leq i \leq V$ y finalmente de la selección de la palabra cuyo modelo posea la máxima verosimilitud: $i^* = arg \max_{1 \leq i \leq V} \left(P\left(O|w_i\right)\right)$.

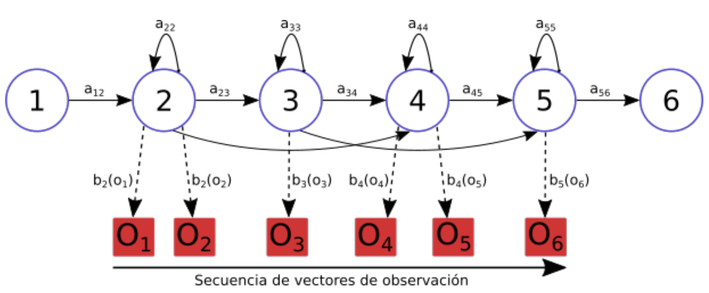

Figure 1: *Ejemplo de representación de un HMM.*

Implementación embebida

Scaling

El algoritmo de reconocimiento está compuesto por productorias de números que representan probabilidades (valores entre 0 y 1), cuyos resultados tienden exponencialmente a 0. Este es un problema que genera *underflow* en muchos de los cálculos (especialmente teniendo en cuenta que el microcontrolador posee una FPU de precisión simple -utiliza sólo 4 bytes para representar variables de punto flotante- en lugar de doble), lo cual tiene como consecuencia una disminución en el grado de reconocimiento. Una mejora consiste en incorporar un procedimiento de escalamiento de las probabilidades a medida que se realizan los cálculos.

Probabilidades logarítmicas

Además del procedimiento de escalamiento, se decidió también implementar el algoritmo completo utilizando solamente probabilidades logarítmicas a lo largo del mismo. El principal inconveniente que se presentó al implementar esta solución, se dio al aplicarlo a una sumatoria de valores. En esos casos, se utilizó la siguiente aproximación:

$$log\left(\sum f\left(x\right)\right) \approx max\left(log\left(f\left(x\right)\right)\right) \tag{3}$$

Máquinas de Vectores de Soporte

Las SVMs son modelos discriminativos que clasifican datos estimando hiperplanos de clasificación o separación de clases, en lugar de modelar una distribución de probabilidad a partir de datos de entrenamiento, como lo hacen los HMMs.
El funcionamiento de las SVMs se basa en maximizar un margen: la distancia entre la frontera de clasificación y las muestras de entrenamiento. El margen indica cuánto ruido (uno de los principales problemas del reconocimiento de habla) agregado a muestras limpias es permitido en el sistema.

Sistema híbrido HMM/SVM

El poder de una representación HMM está en su habilidad de modelar la evolución temporal de la señal a través de un proceso Markov.
Por otro lado, las SVMs son clasificadores estáticos (los vectores con los que se trabaja deben tener todos la misma longitud), que tienen que ser adaptados para lidiar con la duración variable en las pronunciaciones del habla. Es por eso que se mantiene el framework HMM, trabajándose entonces con modelos híbridos HMM/SVM.
La forma de obtener una secuencia de vectores de observación de largo fijo a partir de la secuencia de habla de largo variable utilizando el framework HMM consiste en promediar aquellos vectores asociados (con mayor probabilidad) a los mismos estados (figura 2).

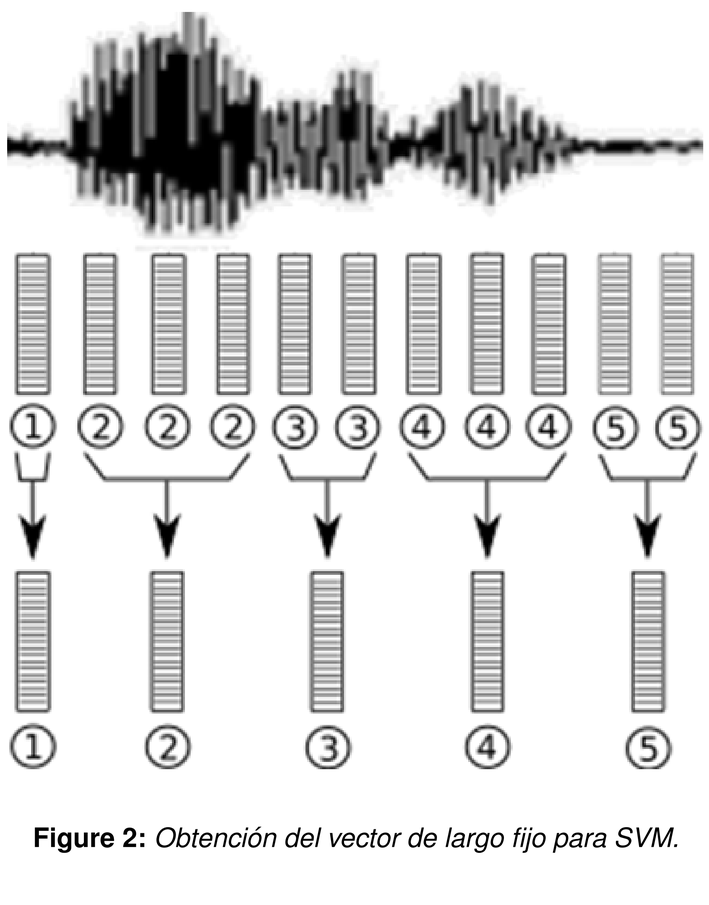

Figure 2: *Obtención del vector de largo fijo para SVM.*

Resultados

Resultados HMM

Los procesos de entrenamiento y testeo de los modelos HMM se realizaron utilizando una base de datos en español de 11 palabras (números 'cero' a 'diez') con 1089 audios para entrenamiento y 264 audios para testeo grabados en una cámara anecoica para trabajar con la mejor relación señal/ruido posible. Se testearon modelos con 8, 16 y 24 estados (tabla 1).

Table 1: *Rendimiento de los modelos HMM [1].*

Estados	Mezclas	Test [%]	WER [%]	Tamaño [KB]
8	2	88.64	11.36	57
	4	94.70	5.30	113
	8	93.56	6.44	226
	16	93.56	6.44	451
16	**2**	**96.21**	**3.79**	**114**
	4	96.59	3.41	228
	8	95.83	4.17	452
24	2	95.08	4.92	171
	4	95.83	4.17	340

Si bien los mejores resultados en términos del Word Error Rate (WER) se obtuvieron utilizando modelos HMM con 16 estados y 4 mezclas cada uno, se eligió como tipo de modelo implementado en el prototipo final al de 16 estados y 2 mezclas, ya que la diferencia en el rendimiento es insignificante, mientras que el requerimiento de memoria es de la mitad.

Comparación HMM vs. DTW

La implementación propuesta fue comparada con un sistema que corre bajo la misma plataforma pero basado en Dynamic Time Warping (DTW) [2]. DTW es una técnica de *template matching* que realiza una alineación temporal entre dos pronunciaciones para obtener luego una comparación que indique sus similitudes.
El rendimiento fue evaluado en términos de grado de reconocimiento (medido con el WER), velocidad (medida usando el RTF[1]) y memoria requerida (tabla 2).

Table 2: *Comparación de rendimientos: HMM vs. DTW [1].*

Algoritmo	Test [%]	WER [%]	RTF	Tamaño [KB]
HMM	96.21	3.79	0.37	114
DTW	65.22	34.78	0.54[2]	111[3]

Resultados HMM/SVM

Los resultados para el modelo híbrido HMM/SVM fueron muy inferiores que para el caso HMM (tienen un WER más de cuatro veces mayor). Entonces, no se justifica su implementación embebida ya que su correspondiente algoritmo de reconocimiento consumiría mayores recursos del microcontrolador (en memoria de código y de datos), tendría un peor RTF y el WER resultante sería mayor.

Conclusiones

El análisis del rendimiento, tanto para el caso de HMM como para el de HMM/SVM, no brinda a primera vista información sobre cuáles podrían ser las razones que generarían mejores resultados con uno u otro clasificador. Sin embargo, una de las posibles razones de la reducción del rendimiento en el modelo híbrido HMM/SVM, respecto al caso HMM, puede haberse encontrado en el método de construcción de los vectores de largo fijo a partir de la secuencia de vectores de observación de largo variable. El hecho de promediar vectores de características elimina información potencialmente útil para el reconocimiento, que no termina siendo utilizada para discriminar entre una clase y otra por parte del clasificador SVM.

Agradecimientos

El autor agradece la colaboración de Diego A. Evin (director de esta tesis de grado) y de miembros del Laboratorio de Investigaciones Sensoriales (INIGEM, CONICET-UBA) por sus enormes aportes a lo largo del desarrollo.

Referencias

[1] Marufo da Silva, M., Evin, D. A., & Verrastro, S. (2016, November). Speaker-independent embedded speech recognition using Hidden Markov Models. In Ciencias de la Informática y Desarrollos de Investigación (CACIDI), IEEE Congreso Argentino de (pp. 1-6). IEEE.

[2] Alvarez, A. G., Evin, D. A., & Verrastro, S. (2016). Implementation of a Speech Recognition System in a DSC. IEEE Latin America Transactions, 14(6), 2657-2662.

[1] El factor de tiempo real (RTF) se define como: $RTF = \frac{\text{Tiempo de reconocimiento}}{\text{Largo de la pronunciación}}$
[2] Valor medio de varias repeticiones medidas.
[3] Para templates de 0.69 segundos de largo.