

RECONOCIMIENTO DEL HABLA CON CUANTIFICADORES VECTORIALES OPTIMOS DISEÑADOS MEDIANTE ALGORITMOS GENETICOS

Jesús Fuster González, Enric Monte Moreno
Departamento de Teoría de Señal y Comunicaciones
Universidad Politécnica de Catalunya
Campus Nord C/Gran Capitán s/n, Edificio D5, 08034 BCN
e-mail: enric@gps.tsc.upc.es

ABSTRAT

A speech recognition system based in Hidden Markov Models (HMM) needs that we give it the minimum information to train the model, but this information must be so precise as possible. In this paper we present a system that makes the training of a codebook using a combination of LVQ [3] and a genetic algorithm [1]. This method perform a parallel search and decrease the minimum distortion of the voice signal. This is a desirable property because minimize errors when training hidden Markov models.

1. INTRODUCCION

En la actualidad existen numerosos sistemas de reconocimiento del habla basados en el empleo de los modelos ocultos de Markov (HMM). Como es bien conocido, estos modelos son procesos estocásticos que nos sirven para caracterizar las palabras a reconocer con el fin de emitir una decisión de reconocimiento. El aspecto crucial del empleo de esta técnica se encuentra en el entrenamiento de los modelos, ya que las prestaciones que el sistema global alcance dependerán fuertemente de la estimación de los parámetros que hacemos en este paso.

La utilización de los modelos ocultos de Markov [2] conlleva un preprocesado previo de la señal de voz, que consta de las siguientes fases: Conversión A/D, Preénfasis, Enventanamiento, Parametrización y Cuantificación Vectorial. La última fase es crítica en todo el proceso ya que es donde se consigue la mayor tasa de comprensión y donde pueden ser mayores las distorsiones de la señal, pensemos que cualquier defecto que introduzcamos en esta etapa repercutirá directamente en el entrenamiento de los modelos y será en detrimento de la tasa de reconocimiento.

En el presente artículo exponemos un método de entrenamiento de cuantificadores vectoriales diseñados mediante algoritmos genéticos, este método pretende encontrar el mínimo global de la distorsión que introduce el cuantificador vectorial, para ello utilizamos la distancia euclídea.

La estructura del artículo es la siguiente: en el apartado 2 hacemos un breve resumen de la técnica de entrenamiento LVQ, destacando sus ventajas e inconvenientes. El apartado 3 describe la utilización de los algoritmos genéticos. En el apartado 4 hacemos una breve descripción del sistema de entrenamiento propuesto.

2. ENTRENAMIENTO BASADO EN LVQ

El método LVQ (Learning Vector Quantization) está descrito con detalle en [3]. Básicamente los pasos que sigue son:

- 1) Seleccionar un punto de la base de datos (x_i)
- 2) Buscar el codeword (c_j) más cercano a x_i , utilizando para ello la distancia euclídea.
- 3) Actualizar la posición del codeword a partir de la siguiente ley de aprendizaje:
$$c_{jnew} = c_{jold} + \alpha (c_{jold} - x_i)$$
- 4) Volver a 1) hasta llegar al último punto de la base de datos.
- 5) épocas = épocas - 1.
- 6) Volver a 1) hasta que épocas = 0.

Como vemos se trata de un algoritmo adaptativo que aprende de manera iterativa la posición de los codewords. Esta proyección genera la propiedad de colindancia, que consiste en que codewords colindantes en el espacio de las características son colindantes en el mapa fonotrópico.

El problema principal de este algoritmo radica en la inicialización, se puede constatar que al cambiar las condiciones iniciales se obtienen codebooks con distintas distorsiones, cada una de estas distorsiones correspondería a un mínimo relativo de la función de distorsión asociada al cuantificador

vectorial. Dependiendo de las condiciones iniciales que tomemos, puede ocurrir que existan codewords que no se actualicen de forma continua, quedando "anclados" en regiones del espacio aisladas. Esta situación la ilustramos con un ejemplo en la figura 1.

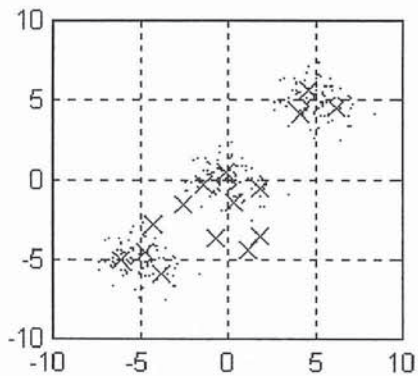


fig 1: Vemos como el método LVQ puede provocar la aparición de codewords aislados.

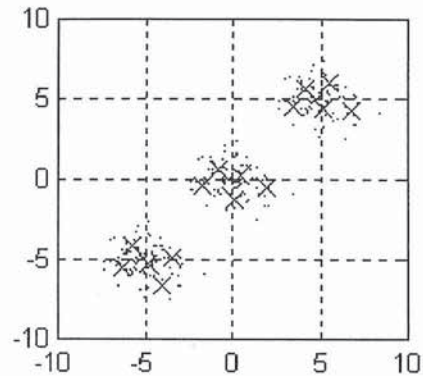


fig 2: Utilizando el algoritmo genético eliminamos el problema de los codewords aislados.

3. ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos [1] son métodos numéricos de búsqueda que se basan en mecanismos de selección y de genética natural. Trabajan sobre una población de individuos, donde cada individuo es una posible solución al problema sobre el que estamos trabajando, en nuestro caso un individuo será un cuantificador vectorial. Sobre una población se aplican los operadores genéticos (selección, cruce y mutación) y se genera otra población cuyos individuos tienen la propiedad de aumentar el valor de la adaptación media. De esta forma, combinando soluciones cada vez más adaptadas conseguimos acercarnos a la solución óptima.

La estructura del procedimiento del Algoritmo genético es la siguiente:

- 1) Generar población inicial $P(t)$.
- 2) Codificar población $P(t)$.
- 3) Calcular la adaptación de $P(t)$.
- 4) $t=t+1$
- 5) Seleccionar $P(t)$ a partir de $P(t-1)$
- 6) Cruzar $P(t)$
- 7) Mutar $P(t)$
- 8) Volver a 2) hasta alcanzar el número de iteraciones deseado.

Sobre este algoritmo cabe destacar los siguientes aspectos:

- La generación de la población inicial se puede realizar de varias formas: inicializando aleatoriamente los cromosomas de los individuos, generando una población de individuos que tengan los codewords dentro de una esfera centrada en el origen, etc.
- La codificación se realiza expresando en binario cada una de las coordenadas de todos los codewords que forman el individuo, tal como se representa con ayuda de la figura 3. Conseguimos una cadena de bits (cromosoma) formado mediante la yuxtaposición de todos los codewords codificados de un individuo.
- La adaptación de cada individuo se calcula a partir de la distorsión que introduce el cuantificador vectorial que representa. De forma que los individuos mejor adaptados son los que introducen menores distorsiones.
- La selección de los individuos de una población, se hace a partir de una regla de selección probabilística, es decir los individuos mejor adaptados tienen mayor probabilidad de ser seleccionados.
- El cruce de los individuos seleccionados, se consigue intercambiando la información codificada de las coordenadas de cada codeword, a partir de una posición de la "cadena" generada aleatoriamente.

- Respecto a la mutación, ésta se implementa generando un número al azar que representa la posición dentro del cromosoma, cuyo bit será cambiado a su complementario.

Empleando el algoritmo genético realizamos una búsqueda en paralelo de puntos de máxima adaptación del espacio de solución, evaluamos diferentes hipótesis descartando las soluciones menos adaptadas y seleccionando las mejores.

Los principales problemas de este método son: La discretización y la saturación del algoritmo. La codificación impone una discretización del espacio de solución, pudiendo ocurrir que el mínimo de la función de distorsión (máximo de adaptación) se encuentre dentro del intervalo de cuantificación, produciendo un efecto de baja precisión. Respecto a la saturación del algoritmo genético, se puede observar que muchas iteraciones del algoritmo genético conduce a poblaciones muy homogéneas, con lo cual se generan individuos muy próximos debido a la falta de diversidad en la población, esto produce que el algoritmo converja hacia un mínimo local y no el global como desearíamos. Estos dos problemas se pueden solventar con la hibridación: utilizando el algoritmo genético y el método LVQ de forma combinada.

El inconveniente primordial que supone la utilización del método híbrido, es el elevado coste computacional que conlleva. El periodo de entrenamiento con este método suele ser de varios días.

En la figura 2 podemos comprobar que el problema de los codewords aislados, que surgía al utilizar el método LVQ, ha desaparecido.

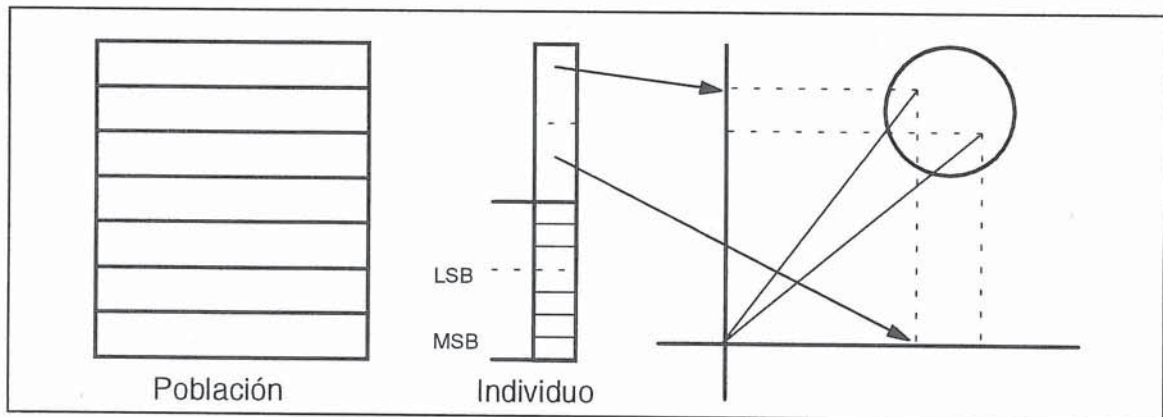


fig 3: Representación esquemática del proceso de codificación.

4. SISTEMA DE ENTRENAMIENTO

La base de datos de entrenamiento (BD_Valencia) está formada por 10 locutores cada uno de los cuales pronunció 50 frases del tipo f y 100 frases del tipo F. Esta base de datos ha sido parametrizada con MFCC (Mel Frequency Cepstrum) de 12 coeficientes. El número total de tramas después de la parametrización es de 391144.

El sistema global de entrenamiento consta de las siguientes fases:

- 1) Inicializamos los individuos de la población a partir de puntos de la base de datos escogidos de forma aleatoria, con probabilidad uniforme.
- 2) Realizamos una primera iteración mediante el método LVQ. (De 10 a 50 épocas con α constante e igual a 0.05) a todos los individuos de la población inicial.
- 3) Aplicamos el algoritmo genético sobre la población de individuos obtenida en 2), con las siguientes particularidades:

- El tamaño de la población es de 50 a 100 individuos.
- Hemos tomado codebooks de 32 y 64 codewords.
- El número de generaciones para cada iteración del algoritmo genético lo hemos escogido entre 10 y 20.
- Utilizamos el modelo elitista para el algoritmo genético, de forma que conservamos los mejores individuos de cada generación (supervivientes). El número de supervivientes lo fijamos a la cuarta parte de la población.
- El cruce de los individuos lo llevamos a cabo sobre cada una de las coordenadas del codeword, de esta forma aumentamos el número de cruces por individuo, como consecuencia la región de búsqueda es mayor que en el caso de un sólo cruce por individuo.

-La probabilidad de mutación la elegimos de forma lineal decreciente con el peso del bit, de forma que bits de mayor peso le corresponden probabilidades de mutación menores. Así protegemos al algoritmo de efectos de inestabilidad.

-Realizamos una ordenación de los codewords de un mismo individuo respecto a los codewords del mejor individuo de la población, con el fin de que el cruce se materialice entre las coordenadas de codewords próximos. De esta forma mejoramos la codificación del problema.

-La función de adaptación tiene la siguiente forma:

$$f_{\text{adp}} = F (1 - (d_m/K))$$

donde F y K son constantes y d_m es la suma de las distorsiones mínimas de cada codeword (c_j) a los puntos más próximos de la base de datos (x_i) utilizando la distancia euclídea.

En cada generación realizamos un escalado del valor de la adaptación para cada individuo con el fin de evitar los fenómenos de saturación y convergencia prematura del algoritmo.

4) Realizamos una nueva iteración del método LVQ, pero con α variando de forma exponencial decreciente y con el mismo número de épocas que en 2).

5) Iteramos 4 o 5 veces la secuencia 3) y 4).

5. CONCLUSIONES

Con la técnica presentada se pueden conseguir mejoras en la tasa de reconocimiento del orden del 10%, ya que la combinación del algoritmo genético con LVQ anula el efecto de codewords aislados manteniendo las buenas propiedades de colindancia que proporciona LVQ. El resultado obtenido mediante este método debe conducir a una solución óptima ya que el algoritmo genético realiza la búsqueda en todo el espacio de solución.

6. REFERENCIAS

- [1] David E. Goldberg, *Genetic Algorithms, in search, optimization and Machine Learning*. Addison Wesley Publishing company. 1989
- [2] Lawrence R. Rabiner, *A Tutorial on hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, vol 77, Feb. 1989.
- [3] R. Schalkoff 1992, *Pattern Recognition. Statical, structural and neural aproaches*. Jhon Wiley & Sons, Inc.
- [4] Linde, Buzo, Gray. 1980. *An Algorithm for Vector Quantizer Design*. IEEE Transactions on Communication Vol. COM-28, N° 1, Jan 1980.