Reconocimiento de voz mediante coeficientes Melcepstrales

Bezdjian, Alejandro
abezdjia@itba.edu.ar
52108

Rivera, Ignacio
riveign@itba.edu.ar
53029

Hirschowitz, Kevin
khirscho@itba.edu.ar
52539

Hanna, Kevin
khanna@itba.edu.ar
52034

November 13, 2015

Métodos Numéricos Avanzados, Trabajo Práctico Número 2.

1 Resumen

En el siguiente informe se describe el proceso y metodología para construir un reconocedor de voz. El trabajo se desarrolló en Octave basado en el paper 'An efficient MFCC extraction method in speech recognition'[1]. En base al reconocedor de voz construido, se evaluará empiricamente su efectividad en base a casos de prueba y entrenamientos del mismo.

2 Palabras claves

Frecuencias de Mel - Frames - vecotr quantization - Pre-énfasis - Filter-Bank(Bancos de filtro) - Coeficientes de Cepstrum - Fourier - Coeficientes Melcepstrales

3 Introducción

A partir de que el ser humano contrae obligaciones y derechos ha surgido la necesidad de poder identificarlo unívocamente. Ya sea con su firma física, digital, huellas dactilares, fotografías, contraseñas, llaves y voz, entre otras. Existen diversas metodologías para poder distinguir en cada una de ellas la autenticidad del individuo en cuestión. Hoy en día ya es posible desbloquear un celular con la huella dactilar, probablemente en un futuro no lejano se lo podrá desbloquear con la voz del usuario si uno lo deseara. Se plantea entonces un desafío tanto interesante como desafiante. ¿Se podrá distinguir la voz unívocamente y cuáles son las variables intervinientes?

El siguiente informe fue dividido en distintas secciones que intentarán responder esta pregunta. Primero se explica la metodología utilizada para realizar un reconocedor de voz. Cada uno de los pasos con sus ventajas e inconvenientes que presentaron darán una idea de la dificultad del problema a resolver. Muchos están sujetos a mejoras y, probablemente, el uso de esta misma metodología en otra tecnología o lenguaje podría resultar muy ineficiente debido al alocamiento de memoria y complejidad computacional. Encontramos en Octave un buen equilibrio entre estas últimas dos que permitió conseguir resultados aceptables. Los mismos son analizados en la sección de resultados.

Para finalizar se presenta en una última sección las conclusiones obtenidas en base a los resultados y metodología empleada, posibles ventajas y mejoras.

4 Metodología

4.1 Obtención de coeficientes Melcepstrales

Para ello fue necesario tomar muestras de voces de personas, las mismas fueron grabadas en una frecuencia de 8000 Hz en un único canal de grabación (Mono).

El programa utilizado para realizar las grabaciones fue Audacity. Se intentó que en el ambiente en el que se grabaron las muestras no haya ruidos molestos que pudieran interferir en el posterior desarrollo del trabajo. Se tomaron cuatro

[1] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun. An ef- ficient mfcc extraction method in speech recognition. In Proceedings IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006., pa- ges 4 pp.–, May 2006.

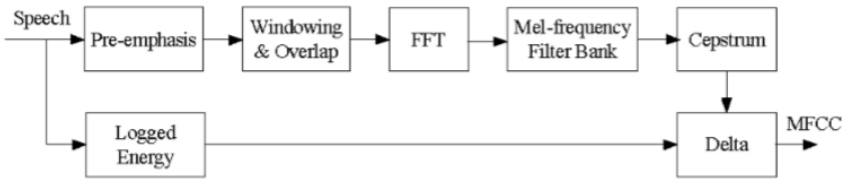

Figure 1: Metodología utilizada para implementar el reconocedor de voz.

muestras a cada persona, una fue destinada al entrenamiento y las restantes a las pruebas mismas del sistema. Las frases elegidas para las grabaciones fueron cortas ya que se deseaba que tengan un intervalo de duración de aproximadamente 2-3 segundos cada una. Dentro de las personas elegidas tomamos hombres y mujeres en un rango de edad amplio, de 20 a 65 años.

4.2 Pre-énfasis

El proceso de pre-énfasis realiza la primera transformación de la señal obtenida. Le realiza un suavizado/aplanamiento, para eliminar todo tipo de efectos sonoros o ruidos no deseados mediante la alteración de las características de amplitud y frecuencia. Eliminando entre otros, los efectos de saturación y distorsión indeseados en la señal dada.

Algorithm 1 Pre-énfasis

1: **procedure** PRE-ENFASIS($speech, a$)
2: $preemphasizedspeech(1) \leftarrow speech(1)$
3: **for** $k \leftarrow 2, rows(speech)$ **do**
4: $preemphasizedspeech(i) \leftarrow speech(i) - a * speech(i-1)$
5: $output \leftarrow preemphasizedspeech$

4.3 Separación en Frames y restricciones de superposición

En esta etapa se busca representar mediante una estructura de datos eficiente la superposición de dos ondas consecutivas. En un principio se pensó realizar esto en una matriz creada en base a dos ciclos anidados.

Empiricamente se comprobó que la eficiencia de Octave realizando este tipo de ciclos no se comparaba en lo absoluto respecto a realizar operaciones entre vectores y matrices del tipo producto o suma, que además llevaban al mismo resultado final. Es por esta última razón que la primer idea se dejó de lado, no significando esto que en otros lenguajes pueda resultar más conveniente.

La estructura pensada es una matriz donde cada columna representa un frame, para el problema se tomó 20ms(160 muestras) y 10ms(80 muestras). Cada columna comienza con un valor múltiplo de 80 mientras que en las filas el valor incrementa unitariamente hasta llegar a 160. Se puede pensar que cada celda de la matriz representa un punto de la onda, una discretización de la onda continua. Se muestra a continuación el pseudo-código asociado a la estructura de datos explicada:

Algorithm 2 Separación en frames

```
1: procedure FRAMES(speech, framelength, framelegthoverlapped)
2:     k ← 1
3:     for i ← 1, framelength do
4:         for j ← 1, quantityoframes do
5:             framematrix(i, j) ← k
6:             k ← k + framelengthoverlapped
7:         k ← i + 1
8:     framematrix ← framematrix
```

El pseudocódigo mostrado a pesar de ser ineficiente es bastante más claro en cuanto a la legibilidad y poder identificar sus partes, es por eso que decidimos de todas formas incluirlo en el informe.

Esta separación en Frames trae apegado grandes saltos de frecuencia, desde cero hasta la señal. Para reducir este efecto se le aplica un ventaneo de Hamming. Octave cuenta con su propia implementación de Hamming que se podría haber usado, sin embargo, decidimos para este trabajo realizar nuestra propia implementación.

Algorithm 3 Hamming

```
1: procedure HAMMING(n)
2:     for i ← 1, n do
3:         Ham(i) ← 0.54 − 0.46 ∗ cos(2 ∗ pi ∗ ((i − 1)/(n − 1)));
4:     Hamming ← Ham′
```

4.4 Transformada rápida de Fourier

Una vez que se tiene la señal dividida en sus distintos frames se le aplica la transformada rápida de Fourier. El paper en el que basamos el trabajo sugiere una FFT de 256 puntos para obtener una buena resolución en frecuencia. Para este paso decidimos utilizar la función implementada por Octave que realiza la transformada rápida de forma eficiente.

Se puede utilizar como una posible mejora la FFTW(Fastest Fourier Transform in the West) en caso de que la versión de Octave que se esté utilizando no lo haga. Algunas versiones de Octave implementan FFT con FFTW mientras que otras con FFTPACK.[2]

4.5 Bancos de filtro

Se tomó como referencia la bibliografía sugerida por la cátedra en el que James Lyons realiza un tutorial de como realizar filterbanks[3]. Los filtros que se crean en esta parte están caracterizados por la función partida que sigue a continuación:

[2]https://www.gnu.org/software/octave/doc/interpreter/Signal-Processing.html. Accesado en Noviembre 2015

[3]James Lyons. Mel frequency cepstral coefficient (mfcc) tutorial. *http* : *//practicalcryptography.com/miscellaneous/machine−learning/guide−mel−frequency−cepstral − coefficients − mfccs/*. Accesado en Noviembre 2015.

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

Figure 2: Fórmula partida para la creación de los bancos de filtros.

4.6 Mapeo a frecuencias de Mel

Existe una pequeña diferencia en las fórmulas que se dan en el paper: 'An efficient mfcc extraction method in speech recognition.' con respecto a las fórmulas presentes en la página sugerida por la cátedra como tutorial citada en la subsección anterior. El cambio en las fórmulas no parecería ser muy significativo debido al comportamiento del logaritmo natural. En el trabajo utilizamos las fórmulas sugeridas por el tutorial.

$$M(f) = 1125 \ln(1 + f/700) \quad (1)$$

Figure 3: Fórmula para traducir de frecuencias a Mel.

$$M^{-1}(m) = 700(\exp(m/1125) - 1) \quad (2)$$

Figure 4: Fórmula para traducir de Mel a frecuencias.

4.7 Coeficientes de Cepstrum

Se calculan los primeros doce coeficientes de Cepstrum con la fórmula sugerida por el paper, Fig 5. Se calculan a partir del resultado obtenido posterior a la aplicación del filtro por bancos. A la salida de cada uno de los filtros se le aplica la función.

4.8 Cálculo de energía

El cálculo de la energía cobra relevancia ya que es el decimo-tercer coeficiente Mel-cepstral. Se lo debe añadir a los otros coeficientes antes de realizar el cálculo de deltas.

4.9 Cálculo de deltas

Existen distintas formas de calcular el valor de los deltas, se siguió la sugerencia del paper que basa el cálculo de los mismos en función de los coeficientes Cepstrum encontrados en la sub-sección anterior, Fig 6. La salida de esto es entonces nuestros coeficientes Mel-Cepstrales, los cuales serán de utilidad para poder identificar individuos.

5 Vector quantization

Una vez obtenidos los coeficientes melcepstrales tanto de las muestras de entrenamiento como de las de prueba, los coeficientes correspondientes a la muestra de entrenamiento se pasarán por el proceso de vector quantization.

En esta etapa se generan clusters que resumen y reducen la información en menos puntos de acuerdo a su distancia realtiva a los otros puntos presentes. Se decidió utilizar la implementación de vector quantization facilitada por la cátedra, sin embargo se podrían haber utilizado otros métodos de aprendizaje no

$$c_n = \sum_{k=1}^{K} (\log S_k) \cos[n(k-0.5)\frac{\pi}{K}]$$

Figure 5: Fórmula para calcular los coeficientes de Cepstrum.

Algorithm 4 Cálculo de energía

```
1: procedure GETSPEECHENERGY(frame, framelength)
2:     energy ← 0
3:     for n ← 1, framelength do
4:         energy ← energy + frame(n) * *2
5:     energy ← log10(energy)
```

$$dc_t = \frac{2(c_{t+2} - c_{t-2}) + (c_{t+1} - c_{t-1})}{10}$$

Figure 6: Fórmula para calcular los deltas en base a los coeficientes Cepstrum.

supervisado para reducir la información en clusters más pequeños. El algoritmo de Hopfield podría ser una opción o una división con Voronoi también podría aplicarse al problema.

Sin descartar los algoritmos de aprendizaje supervisado que también ayudan a resolver este tipo de diferenciaciones. Uno de los más conocidos es Learning Vector Quantization.

6 Comparación y coincidencias

A modo de explicación del proceso, a los vectores salientes de vector quantization(representan el entrenamiento de cada individuo de la población) los llamaremos vq-indiv-N, siendo N el individuo al que se refiere. Los vq-indiv-N son comparados con los coeficientes melcepstrales de cada una de las pruebas Mel-coef-N. Se hizo que cada individuo grabe cuatro(4) fragmentos, de los cuales uno(1) se utilizó para entrenamiento y los restantes para pruebas(por ende habrá 3 pruebas). Por lo tanto el proceso para realizar una prueba es el siguiente:

Se le aplica una función para poder medir las diferencias relativas entre cada vq-indiv-N respecto a cada Mel-coef-N y aquel que arroje el menor resultado se condiderará que es para esa prueba el que corresponde para ese individuo.

7 Ejecución y organización

Para la ejecución del reconocedor de voz deberá abrirse en una consola el Octave o abrir desde GUI Octave, configurar los parámetros con los que desea correr el algoritmo en el archivo "input.txt" ubicado dentro del directorio "algoritmo".

Cada paso del algoritmo está modularizado en una función distinta por si a futuro se deseara implementar alguna de ellas de forma más eficiente sin tener que cambiar la estructura de la aplicación total. Los casos de prueba desarrollados son "fragmento2()", "fragmento3()", "fragmento4()" y "graphs()", los cuales al finalizar su ejecución muestran la cantidad de matcheas sobre los casos de prueba, y qué individuo coincidió con quién. Ninguno de ellos recibe

parámetros y todos se encuentran en la carpeta de "algoritmo".

Bajo las carpetas "Entrenamiento" y "Prueba" se encuentran los audios rotulados de acuerdo al número de frase y al interlocutor al que corresponde. En la carpeta "Papers" se encuentran los documentos que fueron consultados con el fin de poder construir el reconocedor de voz.

8 Resultados y conclusiones

El resultado obtenido fue del 57% de efectividad. Si bien podemos afirmar que el reconocedor identifica a los individuos más de la mitad de las veces, se quiso investigar si las respuestas eran contundentes. Referimos a contundentes a si la mínima distancia que se encontró entre la prueba y el entrenamiento del individuo era significativamente menor a las demás distancias. Para ello se consideró que la mejor alternativa para visualizar esto era graficar las distancias mínimas con respecto a cada uno de los individuos.

Se tomaron algunas pruebas en particular y graficamos la distancia promedio de sus coeficientes mel-cepstrales en función de los individuos en la muestra. En las pruebas que se muestran, se acotó la cantidad de individuos para que el gráfico se aprecie más. Además se tomaron individuos que son familiares entre sí para poder ver si, en caso de que el algoritmo devuelva un error, pueda ser que confundió la voz del padre con la del hijo o entre hermanos, etc.

El algoritmo por como está diseñado siempre dará una respuesta, en base a qué voz es más similar el input que se le da. Por ejemplo, en la Fig 7, la voz que busca buscarle coincidencia es a la de KevinK, como se observa en el eje X, el número 4 es de KevinK y la distancia promedio es la menor de todas. Lo cual es un resultado coherente, es decir el algoritmo acertó para este individuo en esta prueba.

Sin embargo, no hay que confundir que porque en una prueba un individuo dio poca distancia promedio de diferencia, esto no necesariamente implica que su voz sea similar a la del otro para todas las pruebas. Se podría llegar a esta conclusión si el resultado se repite a lo largo de muchas y variadas pruebas. Un ejemplo que muestra lo mencionado es la Fig 8 y Fig 9. En la Fig 9, cuando se le busca una coincidencia a Eduardo, Erika parecería que podría tener una gran coincidencia pero esto queda refutado ya que en otra prueba(Fig 8) la diferencia entre ambos es significativa.

La Fig 10 muestra que ante la búsqueda de una coincidencia para un individuo femenino, la que aparece como resultado es otro individuo femenino. En particular, su relación es de madre e hija. No resulta sorprendente que ante una prueba de un individuo femenino, los resultados a pesar de no coincidir sean de una mujer.

Esto se debe a que sus registros son más agudos que los que puede alcanzar un hombre, esto se ve reflejado en el espectro de frecuencias que utilizan al hablar.

Cuando se realizaron las pruebas se realizó una modificación sobre el algoritmo de vector quantization con el fin de eliminar la aleatoriedad en el mismo. Existían casos de prueba en los que un mismo individuo tenía coincidencia

con A y luego con B. Se consideró que para poder analizar de forma más eficaz la eficiencia del algoritmo, el mismo debía ser determinístico. Es decir para todas las corridas que se realizaran con los mismos parámetros, devolver la misma respuesta.

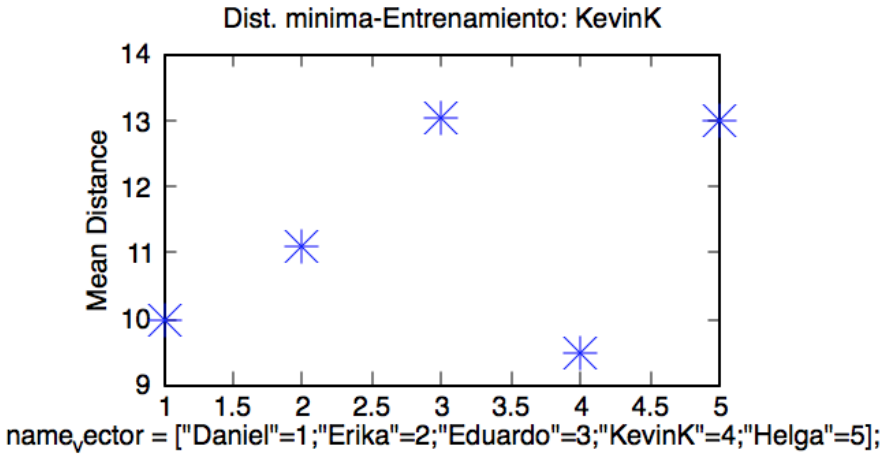

Figure 7: Gráfico del promedio de la distancia de una grabación de KevinK respecto a los demás fragmentos.

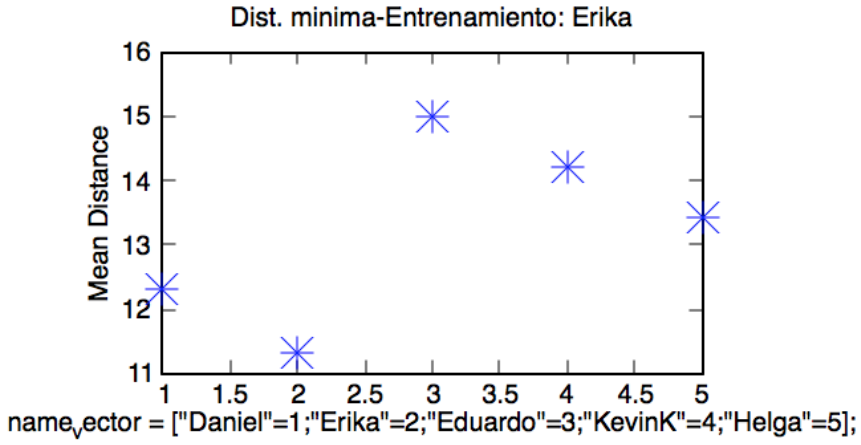

Figure 8: Gráfico del promedio de la distancia de una grabación de Erika respecto a los demás fragmentos.

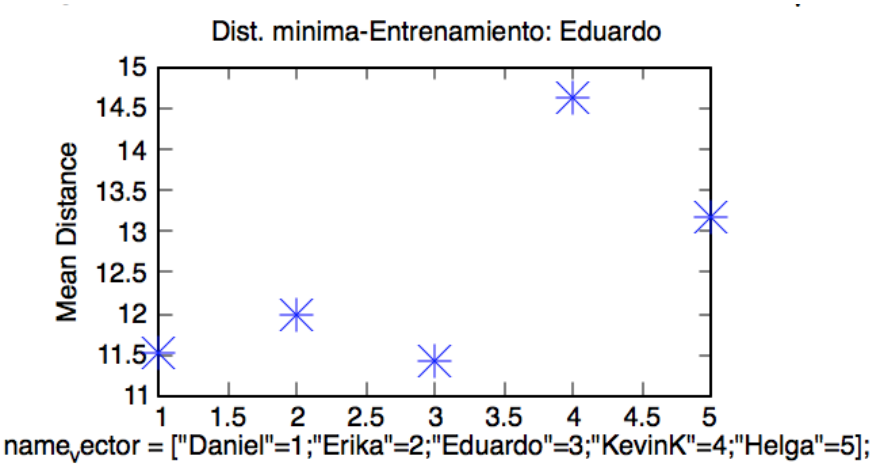

Figure 9: Gráfico del promedio de la distancia de una grabación de Eduardo respecto a los demás fragmentos.

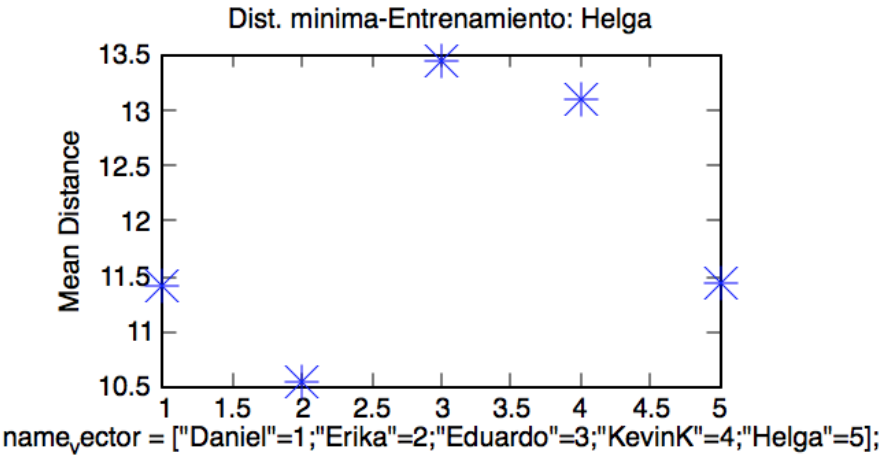

Figure 10: Gráfico del promedio de la distancia de una grabación de Helga respecto a los demás fragmentos.

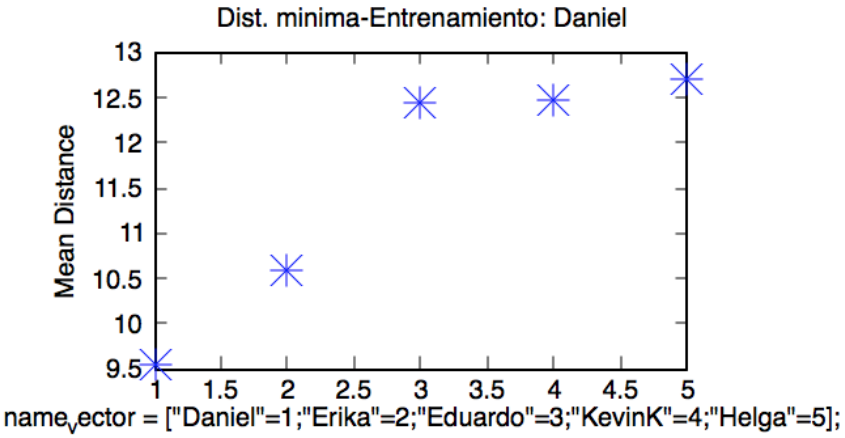

Figure 11: Gráfico del promedio de la distancia de una grabación de Daniel respecto a los demás fragmentos.

Individuo	Según el reconocedor la voz es de:	Correcto
Daniel	Daniel	Si
Erika	Erika	Si
Eduardo	Eduardo	Si
KevinK	KevinK	Si
Helga	Erika	No

Table 1: Tabla de resultados de la prueba 1.

Individuo	Según el reconocedor la voz es de:	Correcto
Daniel	Daniel	Si
Erika	Daniel	No
Eduardo	Eduardo	Si
KevinK	KevinK	Si
Helga	Daniel	No

Table 2: Tabla de resultados de la prueba 2.

Individuo	Según el reconocedor la voz es de:	Correcto
Daniel	Daniel	Si
Erika	Erika	Si
Eduardo	Daniel	No
KevinK	KevinK	Si
Helga	Erika	No

Table 3: Tabla de resultados de la prueba 3.

9 Bibliografía

- Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun. An ef- ficient mfcc extraction method in speech recognition. In Proceedings IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006., pa- ges 4 pp.–, May 2006.

- Md Rashidul Hasan, Mustafa Jamil, Md Golam Rabbani, and Md Saifur Rahman. Speaker identification using mel frequency cepstral coefficients. In 3rd International Conference on Electrical & Computer Engineering ICECE, volume 2004, 2004.

- Y. Linde, A Buzo, and R.M. Gray. An algorithm for vector quantizer design. IEEE Transactions on Communications, 28(1):84–95, Jan 1980.

- James Lyons. Mel frequency cepstral coefficient (mfcc) tutorial. $http : //practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/$. Accesado en Noviembre 2015.