Algoritmos de Compensación de Características Cepstrales para Reconocimiento Automático del Habla Robusto

Oscar Saz, Luis Buera, Eduardo Lleida, Antonio Miguel, Alfonso Ortega

Departamento de Ingeniería Electrónica y Comunicaciones
Centro Politécnico Superior
Universidad de Zaragoza
{oskarsaz,lbuera,lleida,amiguel,ortega}@unizar.es

Resumen

En este artículo se va a realizar una revisión de varios algoritmos de compensación de características cepstrales desarrollados hasta el momento. Se verá como mediante una estimación de error cuadrático medio mínimo (MMSE) se pueden obtener diferentes estrategias de compensación de la variabilidad existente entre la señal de voz limpia y la señal de voz contaminada por el ruido. Todo el conjunto de algoritmos estudiados se aplicará a la tarea de proporcionar robustez frente al ambiente a un sistema de reconocimiento automático del habla ante la base de datos SpeechDat-Car; y se podrán estudiar y comparar los resultados obtenidos por cada algoritmo en términos de mejora de la tasa de error.

1. Introducción

Es bien sabido el problema que suponen para las prestaciones de los sistemas de reconocimiento automático del habla la variabilidad en las condiciones acústicas entre el entrenamiento del sistema y las condiciones en reconocimiento [1]. La aparición de ruido ambiental y otros elementos como stress en el locutor pueden llegar a degradar las propiedades de la señal de voz, dificultando en gran medida la tarea de los sistemas reconocedores del habla.

La compensación de características cepstrales a partir de una estimación de error cuadrático medio mínimo (MMSE) se ha aplicado de forma exitosa en diferentes algoritmos como los que se van a tratar en este artículo (RATZ, SPLICE, MMCN, POF, RATZ Interpolado, SPLIC-ME y MEMLIN). Todos ellos constan de dos fases: En una primera fase de entrenamiento, se estiman un conjunto de parámetros que representan la modificación que se produce entre el cepstrum de las señales ruidosas y el cepstrum de las señales sin contaminar; para esta fase se requiere una base de datos con señales estéreo. En la fase de test, se aplica el estimador MMSE a una determinada función de compensación para obtener a partir de una señal ruidosa una aproximación a la señal limpia original, dados los parámetros que modelan el ambiente calculados en el entrenamiento.

Este artículo está organizado de la siguiente manera: En la sección 2, se verá cómo se aplica el estimador MMSE a la tarea de la compensación de características cepstrales. Posteriormente, en la sección 3, presentaremos la descripción teórica de los siete algoritmos que van a ser objeto de estudio en este trabajo. En la sección 4, expondremos los resultados en términos de mejora de la tasa de error obtenidos por los algoritmos presentados en la tarea de reconocimiento automático del habla ante la base de datos SpeechDat-Car. Estos resultados serán discutidos en la sección 5, para finalmente extraer las conclusiones a este trabajo en la sección 6.

2. Estimador MMSE

2.1. Aproximaciones realizadas

En todos los algoritmos que aquí se someten a estudio, se realizan un conjunto de aproximaciones que vamos a comentar a continuación. En primer lugar, el vector de características cepstrales de la señal limpia, x, se modela con una distribución de probabilidad consistente en una mezcla de gaussianas (s_x) de vector de medias μ_{s_x}, matriz de covarianzas Σ_{s_x} y probabilidades a priori $p(s_x)$ (1)(2).

$$p(x) = \sum_{s_x} p(x|s_x)p(s_x) \tag{1}$$

$$p(x|s_x) = N(x; \mu_{s_x}; \Sigma_{s_x}) \tag{2}$$

De la misma forma, la señal ruidosa, y, se modela también como una mezcla de gaussianas (3)(4) de forma similar a como se hace con la señal limpia. Cuando la señal ruidosa puede haber sido generada por un conjunto de posibles entornos acústicos se puede obtener un modelo para toda la señal ruidosa, s_y, independientemente del entorno en el que se haya producido; o se puede obtener un modelo de señal para cada uno de los entornos, s_y^e. Veremos como esta última situación, aprovechada por algunos de los algoritmos aquí presentados, permite mejorar la estimación de la señal a realizar.

$$p_e(y) = \sum_{s_y^e} p(y|s_y^e)p(s_y^e) \tag{3}$$

$$p(y|s_y^e) = N(y; \mu_{s_y^e}; \Sigma_{s_y^e}) \tag{4}$$

Por último, también se considerará que el vector de características de la señal limpia, x, es una función (5) del vector de características de la señal ruidosa, y, el conjunto de gaussianas que modelan la señal limpia, s_x, y el conjunto de gaussianas que modelan la señal ruidosa, s_y^e.

$$x \simeq f(y, s_x, s_y^e) \tag{5}$$

2.2. Estimación MMSE

Dado el vector de características de la señal ruidosa, y, la formulación teórica del estimador MMSE (6) nos dice que la estimación del vector cepstral de la señal limpia, x, será la esperanza de la señal limpia condicionada por la señal ruidosa.

$$\hat{x} = E[x|y] = \int_x x p(x|y) dx \quad (6)$$

El estimador MMSE así presentado es imposible de resolver, dado que no se puede conocer $p(x|y)$, la función densidad de probabilidad de x condicionada por y. Pero, asumiendo las aproximaciones vistas en la subsección anterior, podemos obtener una aproximación (7) a la solución exacta.

$$\hat{x} \simeq \int_x \sum_e \sum_{s_x} \sum_{s_y^e} f(y, s_x, s_y^e) p(x, s_x, s_y^e|y) dx \quad (7)$$

En este caso, nos queda que el término más complejo es $p(x, s_x, s_y^e|y)$, que es la probabilidad de x, s_x y s_y^e dado y. Por la aplicación del teorema de Bayes, tenemos que $p(x, s_x, s_y^e|y) = p(x|s_x, s_y^e)p(s_y^e|y)p(s_x|s_y^e, y)$; donde $p(x|s_x, s_y^e)$ es la probabilidad del vector cepstral de la señal limpia dada una pareja de gaussianas limpia-ruidosa; $p(s_x|s_y^e, y)$ es la probabilidad de la gaussiana limpia s_x dada la gaussiana sucia s_y^e y el vector de características de la señal ruidosa; y $p(s_y^e|y)$ es la probabilidad de la gaussiana del espacio de señal ruidosa dada la señal ruidosa y. Este último término se puede sustituir por Bayes, $p(s_y^e|y) = p(e|y)p(s_y^e|e, y)$, donde $p(e|y)$ es la probabilidad de que la señal ruidosa se haya generado en el entorno e y $p(s_y^e|e, y)$ es la probabilidad de la gaussiana ruidosa s_y^e dado el entorno y la señal ruidosa.

De esta forma, se obtiene la solución definitiva (8) al problema MMSE que estabamos buscando resolver para realizar la compensación de características cepstrales.

$$\hat{x} \simeq \sum_e \sum_{s_x} \sum_{s_y^e} f(y, s_x, s_y^e) p(s_x|s_y^e, y) p(s_y^e|e, y) p(e|y) \quad (8)$$

3. Algoritmos a estudio

3.1. Algoritmo RATZ

El algoritmo multivaRiate-gAussian-based cepsTral normaliZation (RATZ) [2][3], considera al vector de características de la señal limpia función (9) del vector cepstral de la señal ruidosa y del modelo de la propia señal limpia (1)(2).

$$x \simeq f_{RATZ}(y, s_x) = y - r_{s_x} \quad (9)$$

Aplicando esta función a la ecuación del estimador MMSE hallada (8), se obtiene la solución propuesta por el algoritmo RATZ (10); cuyas prestaciones descansan sobre el término de transformación r_{s_x}, que relaciona los vectores de características de las señales limpia y sucia para cada gaussiana limpia.

$$\hat{x}_t \simeq y_t - \sum_{s_x} p(s_x|y_t) r_{s_x} \quad (10)$$

Para obtener este resultado, debemos obtener previamente la probabilidad $p(s_x|y_t)$, que es la probabilidad de que el vector de características de la señal ruidosa de entrada, y_t, corresponda con la gaussiana limpia s_x. Este valor se obtiene fácilmente mediante el teorema de Bayes (11).

$$p(s_x|y_t) = \frac{p(y_t|s_x)p(s_x)}{\sum_{s_x} p(y_t|s_x)p(s_x)} \quad (11)$$

Los vectores de transformación, r_{s_x} (13), se obtienen en entrenamiento minimizando el error cuadrático (12) entre la señal limpia real, x, y la estimación de RATZ, $\hat{x}$, con la ayuda del algoritmo de Expectation-Maximization (EM) [4].

$$E_{s_x} = \sum_{t_e} p(s_x|x_{t_e})(x_{t_e} - y_{t_e} + r_{s_x})^2 \quad (12)$$

$$r_{s_x} = \frac{\sum_{t_e} p(s_x|x_{t_e}^e)(y_{t_e}^e - x_{t_e}^e)}{\sum_{t_e} p(s_x|x_{t_e}^e)} \quad (13)$$

Para el entrenamiento se debe conocer la probabilidad de la gaussiana limpia s_x dada la trama limpia de entrada $x_{t_e}^e$, lo cual se puede obtener por Bayes (11).

3.2. Algoritmo SPLICE

El siguiente algoritmo a estudio es el Stereo based Piecewise LInear Compensation for Environments (SPLICE) [5]; este algoritmo trabaja considerando los vectores cepstrales de la señal limpia función (14) de la señal contaminada por el entorno acústico y del modelo de la señal ruidosa (3)(4) sin considerar divisiones entre diversos entornos.

$$x \simeq f_{SPLICE}(y, s_y) = y - r_{s_y} \quad (14)$$

La solución a la ecuación de error cuadrático medio mínimo que se obtiene aplicando la función anterior nos da la solución (15) que propone SPLICE para compensar el vector de características cepstrales de la señal de voz.

$$\hat{x}_t \simeq y_t - \sum_{s_y} p(s_y|y_t) r_{s_y} \quad (15)$$

Hay que hallar un vector de transformación r_{s_y} por cada gaussiana del espacio de señal ruidosa; aplicando un proceso de minimización (16) al igual que en el caso de RATZ se obtiene este conjunto de vectores (17).

$$E_{s_y} = \sum_{t_e} p(s_y|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_y})^2 \quad (16)$$

$$r_{s_y} = \frac{\sum_{t_e} p(s_y|y_{t_e}^e)(y_{t_e}^e - x_{t_e}^e)}{\sum_{t_e} p(s_y|y_{t_e}^e)} \quad (17)$$

En SPLICE, tanto en entrenamiento (17) como en test (15) es necesario obtener mediante Bayes (11) la probabilidad de la gaussiana ruidosa s_y condicionada al vector cepstral de la señal ruidosa y_t, $p(s_y|y_t)$.

3.3. Algoritmo MMCN

El algoritmo Multivariate Model based Cepstral Normalization (MMCN) [6] propone que el vector de características cepstrales de la señal limpia sea función (18) de la señal sucia, del espacio de señal limpia (1)(2) y del espacio de señal de la señal ruidosa (3)(4), sin considerar división entre los diferentes entornos de trabajo.

$$x \simeq f_{MMCN}(y, s_x, s_y) = y - r_{s_x, s_y} \quad (18)$$

Con esta función, la solución (19) que se obtiene al problema MMSE queda en función de ambos espacios de señal, de la señal limpia y de la señal ruidosa.

$$\hat{x}_t \simeq y_t - \sum_{s_x} \sum_{s_y} p(s_x|s_y) p(s_y|y_t) r_{s_x, s_y} \quad (19)$$

En este caso, se deben minimizar los errores asociados a cada pareja de gaussianas limpia-ruidosa (20) para obtener en entrenamiento (21) el conjunto de vectores de transformación asociados a la pareja de gaussianas (s_x,s_y).

$$E_{s_x,s_y} = \sum_{t_e} p(s_x|x_{t_e})p(s_y|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_x,s_y})^2 \quad (20)$$

$$r_{s_x,s_y} = \frac{\sum_{t_e} p(s_x|x^e_{t_e})p(s_y|y^e_{t_e})(y^e_{t_e} - x^e_{t_e})}{\sum_{t_e} p(s_x|x^e_{t_e})p(s_y|y^e_{t_e})} \quad (21)$$

Para este algoritmo, se requiere también conocer la probabilidad condicional entre los dos espacios de señal, $p(s_x|s_y)$, es decir, la probabilidad de la gaussiana limpia s_x dada la gaussiana ruidosa s_y. Este cálculo sólo se puede llevar a cabo en entrenamiento, cuando se dispone de señales estéreo; y no existe una forma analítica de llevarlo a cabo, por lo que se realiza obteniendo las frecuencias relativas de las gaussianas de los espacios de señal (22). Para su cálculo se debe obtener N, que es el número de veces que la gaussiana más probable para el vector de características ruidoso de entrada es s_y; y también se debe obtener $C_N(s_x|s_y)$ que es el número de veces que la pareja (s_x, s_y) es la más probable para cada pareja de vectores cepstrales limpio-ruidoso.

$$p(s_x|s_y) \simeq \frac{C_N(s_x|s_y)}{N} \quad (22)$$

También este algoritmo necesita conocer dos probabilidades condicionales, la de la gaussiana limpia s_x dado el vector de características de la señal limpia y la de la gaussiana ruidosa s_y dado el vector de características de la señal contaminada. Estos valores, necesarios tanto para el entrenamiento (21) como para el test (19) se obtienen por el teorema de Bayes (11).

3.4. Algoritmo POF

Una estrategia diferente a las que se han presentado hasta ahora es el algoritmo Probabilistic Optimum Filtering (POF) [7]. Este algoritmo plantea al vector cepstral limpio, x, función del vector cepstral ruidoso y del espacio de señal ruidosa (3)(4); pero lo hace por medio un filtrado lineal (23), por lo que no sólo usa el vector de características del instante actual, sino de los circundantes.

$$x \simeq f_{POF}(y, s_y) = y * h_{s_y} \quad (23)$$

Por lo que ahora la solución al problema MMSE resulta ser bastante más compleja (24), tanto en cuanto se utiliza una ventana sobre la señal ruidosa de la que no sólo se usa la trama actual. Algunos estudios [8] han mostrado que las mejores prestaciones se obtienen utilizando un filtro de tamaño cinco, por lo que presentaremos el algoritmo para este tamaño concreto de filtro.

$$\hat{x}_t \simeq \sum_{s_y} \left(\sum_{i=-2}^{2} y_{t-i} A_{s_y,i} + B_{s_y} \right) p(s_y|y_t) \quad (24)$$

El conjunto de filtros utilizados en POF (26) se obtienen en entrenamiento tras minimizar el error cuadrático entre la señal limpia real y la salida de los filtros obtenidos (25), y resulta ser una solución muy parecida al filtro de Wiener clásico.

$$E_{s_y} = \sum_{t_e} p(s_y|y_{t_e})(x_{t_e} - y_{t_e} * h_{s_y})^2 \quad (25)$$

$$\begin{aligned} W^T_{s_y} &= \left[A_{s_y,-2}, A_{s_y,-1}, A_{s_y,0}, A_{s_y,1}, A_{s_y,2}, B_{s_y}\right] = \\ &= R^{-1}_{s_y} r_{s_y} \end{aligned} \quad (26)$$

con:

$$R_{s_y} = \sum_{t_e} Y_{t_e} Y^T_{t_e} p(s_y|y^e_{t_e}) \quad (27)$$

$$r_{s_y} = \sum_{t_e} Y_{t_e} x^T_{t_e} p(s_y|y^e_{t_e}) \quad (28)$$

donde:

$$Y^T_{t_e} = \left[y^T_{t_e-2}, y^T_{t_e-1}, y^T_{t_e}, y^T_{t_e+1}, y^T_{t_e+2}, 1\right] \quad (29)$$

La probabilidad $p(s_y|y_t)$ se calcula de la forma que ya hemos visto mediante el teorema de Bayes (11).

3.5. Algoritmo RATZ Interpolado

El algoritmo RATZ interpolado [3] es una variación del algoritmo RATZ, que tiene en cuenta que existen varios entornos que contaminan la señal de voz; para ello, utiliza la misma función que relaciona la señal limpia con la señal ruidosa que RATZ (9), pero con un conjunto de vectores de transformación distintos para cada entorno posible de trabajo (30).

$$x \simeq f_{RATZ-I}(y, s_x) = y - r^e_{s_x} \quad (30)$$

Por lo que la solución que se obtiene con el estimador MMSE (31) tiene en cuenta las transformaciones asociadas a todos los posibles entornos.

$$\hat{x}_t \simeq y_t - \sum_e \sum_{s_x} p(e|y_t)p(s_x|y_t)r^e_{s_x} \quad (31)$$

La forma de obtener los vectores de transformación es igual que en el algoritmo RATZ (13), pero realizando el cálculo para todos los entornos de trabajo con los que se pueda contaminar la señal de voz (33), ya que estamos buscando minimizar un error donde los diferentes modelos de ambiente tienen influencia (32).

$$E_{s_x} = \sum_{t_e} p(s_x|x_{t_e})(x_{t_e} - y_{t_e} + r^e_{s_x})^2 \quad (32)$$

$$r^e_{s_x} = \frac{\sum_{t_e} p(s_x|x^e_{t_e})(y^e_{t_e} - x^e_{t_e})}{\sum_{t_e} p(s_x|x^e_{t_e})} \quad (33)$$

También es necesario encontrar una forma de calcular la probabilidad de que la trama ruidosa a compensar, y_t, haya sido contaminada por el entorno e del conjunto de E entornos posibles. Este cálculo se va a realizar de forma iterativa, de forma que la probabilidad $p(e|y_t)$ se convierte en el parámetro $\alpha_{e,t}$; que para cada instante t se va actualizando (34) con la trama ruidosa y_t con un factor de memoria β.

$$p(e|y_t) \simeq \alpha_{e,t} = \beta \cdot \alpha_{e,t-1} + (1-\beta)\frac{p_e(y_t)}{\sum_e p_e y_t} \quad (34)$$

La probabilidad de la gaussiana s_x dado el vector de características ruidoso no cambia a lo visto hasta ahora, y se calcula mediante Bayes (11).

3.6. Algoritmo SPLIC-ME

El algoritmo SPLIC Multientornos (SPLIC-ME) [6] es una variación del algoritmo SPLICE que trabaja utilizando modelos de señal distintos para los diferentes entornos acústicos con los que se pueda contaminar la señal de voz (3)(4) De esta forma, el vector de características de la señal limpia queda en función (35) de la señal ruidosa y de los modelos de señal contaminada en cada entorno, s_y^e.

$$x \simeq f_{SPLIC-ME}(y, s_y^e) = y - r_{s_y^e} \tag{35}$$

Con esta función, la solución al problema de error cuadrático mínimo queda (36).

$$\hat{x}_t \simeq y_t - \sum_e \sum_{s_y^e} p(e|y_t) p(s_y^e|y_t) r_{s_y^e} \tag{36}$$

Con un conjunto de vectores de transformación (38) para todas las gaussianas de cada entorno acústico posible, a partir de un proceso de minimización (37) entre las señales limpia y la limpia estimada.

$$E_{s_y^e} = \sum_{t_e} p(s_y^e|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_y^e})^2 \tag{37}$$

$$r_{s_y^e} = \frac{\sum_{t_e} p(s_y|y_{t_e}^e)(y_{t_e}^e - x_{t_e}^e)}{\sum_{t_e} p(s_y|y_{t_e}^e)} \tag{38}$$

La probabilidad de entorno se obtiene de forma iterativa (34), y la probabilidad de la gaussiana ruidosa s_y^e dado el vector de características contaminado mediante Bayes (11).

3.7. Algoritmo MEMLIN

El último algoritmo que se va a presentar en este artículo es el Multi-Environment Models based LInear Normalization (MEMLIN) [6]. Este algoritmo usa los modelos más complejos a la hora de obtener una función (39) que relacione la señal limpia, x, con la señal ruidosa ruidosa, y, el modelo de señal limpia (1)(2) y el modelo de señal ruidosa (3)(4), considerando un modelo distinto para cada entorno posible de contaminación acústica.

$$x \simeq f_{MEMLIN}(y, s_x, s_y^e) = y - r_{s_x, s_y^e} \tag{39}$$

La solución al problema MMSE va acumulando complejidad (40), debido a que hay que tener en cuenta muchos factores.

$$\hat{x}_t \simeq y_t - \sum_e \sum_{s_x} \sum_{s_y^e} p(e|y_t) p(s_x|s_y^e) p(s_y^e|y_t) r_{s_x, s_y^e} \tag{40}$$

Tras el proceso de minimización del error cuadrático entre la señal limpia real y la señal estimada del modo planteado por MEMLIN (41) se llega a la obtención de los términos de transformación (42) que relacionan la señal limpia y la ruidosa para cada entorno acústico existente.

$$E_{s_x, s_y^e} = \sum_{t_e} p(s_x|x_{t_e}) p(s_y^e|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_x, s_y^e})^2 \tag{41}$$

$$r_{s_x, s_y^e} = \frac{\sum_{t_e} p(s_x|x_{t_e}^e) p(s_y^e|y_{t_e}^e)(y_{t_e}^e - x_{t_e}^e)}{\sum_{t_e} p(s_x|x_{t_e}^e) p(s_y^e|y_{t_e}^e)} \tag{42}$$

Para la aplicación de este algoritmo es necesario conocer todos los factores probabilísticos que hemos visto a lo largo de este artículo; la probabilidad de entorno $p(e|y_t)$ (34), la probabilidad cruzada entre las gaussianas del modelo de señal limpia y del modelo de señal ruidosa $p(s_x|s_y)$ (22) y las probabilidades de las gaussianas limpia y ruidosa dados los vectores de características limpio y ruidoso correspondientes que se obtienen por Bayes (11).

4. Resultados en reconocimiento de voz

Todos los algoritmos vistos fueron enfrentados a la tarea de proporcionar robustez frente al ambiente a un sistema de reconocimiento automático del habla bajo la base de datos SpeechDat-Car.

En la base de datos SpeechDat-Car [9] se definen siete entornos: coche parado con el motor encendido (E1), tráfico de ciudad, ventanas cerradas y climatizador apagado (condiciones silenciosas) (E2), tráfico de ciudad y condiciones ruidosas: ventanas abiertas y/o climatizador encendido (E3), baja velocidad por carretera de asfalto rugoso y condiciones silenciosas (E4), baja velocidad por carretera rugosa y condiciones ruidosas (E5), alta velocidad por carretera lisa y condiciones silenciosas (E6), y alta velocidad por carretera lisa y condiciones ruidosas (E7).

La base de datos SpeechDat-Car se compone de 200 frases a reconocer para el entorno E1, 223 para el entorno E2, 136 para el entorno E3, 152 para el entorno E4, 120 para el entorno E6 y 56 para el entorno E7. Para la fase de entrenamiento de cada algoritmo se usa el corpus de entrenamiento de la propia base de datos SpeechDat-Car (16108 frases entre los siete entornos de más de 300 locutores diferentes), de tal forma que el material de test se utiliza exclusivamente para la fase de reconocimiento.

Se utiliza la tarea de reconocimiento de dígitos aislados y continuos. Las señales limpias, a las que llamaremos C0, se graban con un micrófono close-talk Shune SM-10A; y las señales ruidosas, denominadas C2, se registran con un micrófono Peiker ME15/V520-1 situado en el techo del automóvil enfrente del conductor. El rango de relación señal a ruido para las señales limpias es de 20 a 30 dB, mientras que las señales ruidosas presentan unas relaciones señal a ruido de 5 a 20 dB.

La señal para reconocimiento se muestrea a una tasa de 16 kHz; y se divide en tramas cada 10 msg mediante una ventana de Hamming de 25 msg de longitud. El vector de características cepstrales empleado se compone de 12 MFCC normalizados, la primera y segunda derivada de dichos coeficientes y de la derivada de la energía normalizada; quedando finalmente con 37 coeficientes. Los modelos fonético-acústicos se componen de 25 modelos de Markov (HMM) continuos de tres estados para modelar los fonemas de la lengua española, y de dos modelos más para los silencios largos y para los silencios cortos entre palabras.

Los resultados del baseline de la base de datos SpeechDat-Car se presentan en la Tabla 1. C0-C0 representa los resultados obtenidos entrenando el reconocedor con C0 y testeando con C0, C0-C2 representa entrenamiento con C0 y test con C2, C2-C2 son los resultados testeando con C2 el reconocedor entrenado con todos los entornos de C2;y, por último, C2†-C2 indica las tasa de error cuando se reconoce las frases ruidosas para cada entorno con un entrenamiento realizado con frases ruidosas de ese mismo entorno. Se presentan las tasas de error obtenidas para cada entorno (E1...E7) y MWER es la tasa de error media a través de todos los entornos. En todos los casos se aplica la sustracción de la media del cepstrum a los vectores de características previamente al reconocimiento, tanto para el baseline

	E1	E2	E3	E4	E5	E6	E7	MWER
C0-C0	1.90	2.64	1.81	1.75	1.62	0.64	0.35	1.75
C0-C2	5.91	14.49	14.55	20.17	21.07	16.19	35.71	16.21
C2-C2	10.39	19.38	16.78	16.41	17.73	13.65	9.86	15.56
C2†-C2	2.86	7.12	4.34	4.39	7.63	4.60	4.76	5.30

Tabla 1: Resultados del baseline. C2†indica re-entrenamiento específico para cada entorno

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
RATZ 8	4.10	12.69	9.79	16.92	18.49	12.06	20.74	12.79	26.10
RATZ 16	4.19	11.24	8.67	15.04	16.68	10.95	17.69	11.46	33.68
RATZ 32	3.24	10.63	7.41	12.41	13.06	10.16	15.65	10.14	44.71
SPLICE 8	4.79	10.10	7.83	11.15	15.82	11.75	15.64	10.51	37.24
SPLICE 16	4.75	9.86	7.83	8.90	14.11	8.25	13.50	9.33	43.96
SPLICE 32	4.70	9.50	6.29	8.77	11.44	7.46	12.92	8.42	49.60
MMCN 8-8	4.79	8.06	8.39	11.53	14.11	11.11	15.31	9.81	42.06
MMCN 16-16	3.64	8.49	7.55	8.27	11.15	8.57	13.50	8.21	54.74
MMCN 32-32	3.35	8.74	6.57	7.64	9.89	7.94	12.58	7.65	58.85
POF 8	3.91	11.32	7.13	8.52	13.73	8.57	17.69	9.51	45.83
POF 16	3.81	10.03	6.71	8.27	11.63	7.46	15.65	8.52	52.20
POF 32	3.24	8.75	7.13	8.15	11.34	6.03	12.24	7.81	58.50

Tabla 2: Resultados con los algoritmos RATZ, SPLICE, MMCN y POF

como en los resultados obtenidos por los algoritmos.

En la Tabla 2 se presentan los resultados obtenidos en el reconocedor por los algoritmos RATZ, SPLICE, MMCN y POF para un número variable de gaussianas con que se modelan los vectores de características de la señal de voz (en el caso de MMCN, los dos valores indican el número de gaussianas usadas para el modelo limpio y para el modelo sucio respectivamente). Además de las tasas de error para cada entorno y de la tasa de error media, MIMP representa la mejora media obtenida en la tasa de error por cada algoritmo. Por último, en la Tabla 3 se muestran los resultados obtenidos por los algoritmos que utilizan modelos multientornos, RATZ Interpolado, SPLIC-ME y MEMLIN, también para un número variable de gaussianas en los modelos de señal.

5. Discusión

Los resultados expuestos muestran claramente la tendencia de que cuanto más complejo se hace el modelo que se aplica a los vectores de características de las señales limpia y ruidosa, mejores resultados se obtienen en reconocimiento. También se puede ver que, dentro de cada algoritmo, cuantas más gaussianas se usan para los modelos de señal mejores son los resultados. Esto es esperable, ya que la modelización de los vectores de características de la señal de voz como un conjunto de gaussianas es sólo una aproximación a la distribución real de los mismos; y cuanto más complejo sea el modelo más próxima será a la realidad y más exacta será la estimación MMSE obtenida.

De esta forma, el algoritmo más básico es el algoritmo RATZ, que utiliza el modelo de señal limpia sobre las tramas de señal ruidosa (10)(13), siendo el que obtiene los peores resultados, 44.71 % de mejora de la tasa de error. El siguiente paso es el algoritmo SPLICE, que utiliza un modelo de señal ruidosa, más ajustado para calcular la probabilidad condicional con la trama ruidosa a compensar (15); obteniendo un resultado del 49.60 % de mejora media. Un gran salto se produce con el algoritmo MMCN, que utiliza un doble modelado de las señales limpia y sucia (19), lo cual incrementa la complejidad y coste computacional del algoritmo, pero sube su capacidad de eliminación de errores en reconocimiento hasta el 58.85 %.

El algoritmo POF supone una variación dentro de los algoritmos aquí presentados, aunque sigue utilizando una estrategia MMSE; su planteamiento de realizar un filtrado (24) del vector de características de la señal ruidosa a compensar le permite llegar al 58.50 % de mejora, aunque es mucho más complicado de implementar que el resto de algoritmos revisados en este artículo. De hecho, trabajos sobre el algoritmo POF [8] muestran que el coste computacional de este algoritmo basado en el filtrado de los vectores de características de la señal ruidosa es bastante mayor que el de los algoritmos basados en vectores de transformación que se sustraen de los vectores cepstrales de la señal ruidosa como RATZ.

El gran avance se obtiene cuando se empieza a tener en cuenta que las señales ruidosas pueden haber sido contaminadas por ambientes diferentes, y que interesa obtener un modelo de la distorsión generada por el ruido independiente para cada entorno. Esa estrategia la utiliza el algoritmo RATZ interpolado y alcanza una mejora del 62.43 %; un 17.72 % más de mejora que el RATZ simple. También la aplica el algoritmo SPLIC-ME, variación multientornos del SPLICE, alcanzando unos resultados del 65.56 % de mejora en tasa de error, 15.96 % más de mejora que SPLICE.

Por último, los mejores resultados los obtiene el que es tam-

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
RATZ-I 8	3.81	8.75	6.85	8.15	12.39	9.68	12.59	8.49	52.52
RATZ-I 16	3.24	8.32	6.99	6.89	10.87	8.41	11.22	7.65	59.24
RATZ-I 32	3.15	8.15	6.72	6.64	9.34	7.94	9.53	7.11	62.43
SPLIC-ME 8	3.16	8.74	6.15	9.27	12.58	9.21	15.65	8.58	54.99
SPLIC-ME 16	3.45	8.23	5.87	7.77	10.77	7.78	13.95	7.71	58.94
SPLIC-ME 32	2.59	7.98	6.15	7.52	9.34	6.67	12.59	7.04	65.56
MEMLIN 8-8	3.16	8.49	6.43	9.27	11.91	9.05	14.97	8.39	56.00
MEMLIN 16-16	3.26	8.06	5.45	7.64	10.01	7.78	12.92	7.37	61.49
MEMLIN 32-32	2.49	7.80	5.03	6.64	9.25	6.51	11.22	6.62	68.50

Tabla 3: Resultados con los algoritmos multientornos: RATZ-Interpolado, SPLIC-ME y MEMLIN

bién el algoritmo más elaborado de los vistos, MEMLIN, que alcanzando una tasa de mejora del 68.50 % obtiene un 10 % de mejora que su versión sin utilizar multientornos, el algoritmo MMCN. Los resultados de tasa de error obtenidos por este algoritmo se acercan en la mayoría de los entornos al mejor resultado posible, que es el obtenido cuando se realiza un entrenamiento específico con las señales ruidosas para cada entorno (C2†-C2 de la Tabla 1). Incluso para el entorno menos ruidoso (Entorno 1), MEMLIN obtiene menos tasa de error que el reentrenamiento específico; y sólo en el entorno más ruidoso (Entorno 7) los resultados de MEMLIN son mucho peores que los dados con el entrenamiento específico.

6. Conclusiones

En este artículo se ha hecho un profundo repaso a todas las técnicas de compensación de características cepstrales basadas en el estimador MMSE. Se ha visto cómo según se aumenta la complejidad de los algoritmos y se usan aproximaciones más precisas en la modelización del cepstrum de la señal de voz se mejoraban los resultados en la tarea de reconocimiento del habla robusto. Uno de los pasos más importantes, como así lo denotan los resultados, es el uso de algoritmos multi-entornos, que tienen en cuenta la modelización de las señales ruidosas en función del entorno acústico en que se han generado. Los algoritmos presentados obtienen su techo en la mejora de la tarea de reconocimiento en un 68.50 % de recuperación de errores, que se obtiene con el algoritmo MEMLIN usando 32 gaussianas para modelar los espacios de señal limpia y ruidosa.

Sin embargo, aún quedan posibilidades de mejora con este tipo de algoritmos; todos los presentados trabajan modelando el ruido mediante desviaciones en la media de los modelos de señal, pero no tienen en cuenta las desviaciones de la varianza; por este motivo, los resultados no son tan buenos cuanto más nivel de ruido hay presente en la señal ruidosa. Consiguiendo conocer y compensar las desviaciones en la varianza de los modelos se podrían mejorar las prestaciones de los sistemas de normalización cepstral. También se podrían obtener nuevas mejoras si se modelan los vectores de características semilimpios obtenidos a la salida de los diferentes algoritmos, de la misma forma que se modelan los vectores cepstrales de las señales limpia y ruidosa, para aplicar sobre la señal semilimpia nuevamente los algoritmos y aumentar la tasa de reconocimiento.

7. Referencias

[1] Sagayama S., Shimoda K., Nakai M., and Shimodaira H., "Analytic methods for acoustic model adaptation: a review", in Proc. Isca ITR-Workshop2001, pp. 67–76, Agosto 2001, Sophia-Antipolis.

[2] Moreno P.J., Raj B., Gouvêa E. and Stern R.M., "Multivariate-gaussian-based cepstral normalization for robust speech recognition", in Proc. ICASSP, Mayo 1995.

[3] Moreno P.J., "Speech recognition in noisy environments", Ph. D. Thesis, Electical and Computer Engineering Department, Carnegie-Mellon University, Abril 1996.

[4] Bilmes, J., "A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models", University of Berkeley, ICSI-TR-97-021, 1997.

[5] Droppo J., Deng L., and Acero A., "Evaluation of the splice algorithm on the aurora2 database", in Proc. Eurospeech, vol. 1, Septiembre 2001.

[6] Buera L., Lleida E., Miguel A., and Ortega A., "Multi-environment models based linear normalization for speech recognition in car conditions", in Proc. ICASSP, Mayo 2004.

[7] Neumeyer L., and Weintraub M., "Probabilistic optimum filtering for robust speech recognition", in Proc. ICASSP, Abril 1994.

[8] Saz, O., "Algoritmos de adaptación al ambiente en sistemas de reconocimiento automático del habla para el automóvil", Proyecto Fin de Carrera, Departamento de Ingeniería Electrónica y Comunicaciones, Centro Politécnico Superior, Universidad de Zaragoza, Febrero 2004.

[9] Moreno A., Noguiera A., and Sesma A., "SpeechDat-Car: Spanish", Technical Report SpeechDat.