Reconocimiento Automático del Habla

Preprocesamiento y Parametrización de la Voz

María José Castro

mcastro@dsic.upv.es

Objetivos del preprocesamiento y la parametrización

Con el **preprocesamiento** se pretende *acondicionar* la señal:

- eliminar ruido,
- realzar la señal.

Con la **parametrización** se pretende obtener una *buena representación* de la señal

- compacta (poco redundante),

  (Téngase en cuenta que la señal muestreada requiere entre 50.000 y 125.000 bits por segundo, cuando la información transmitida como fonemas apenas llega a 50 bits por segundo.)
- y que recoge (y realza nuevamente) toda la información importante.

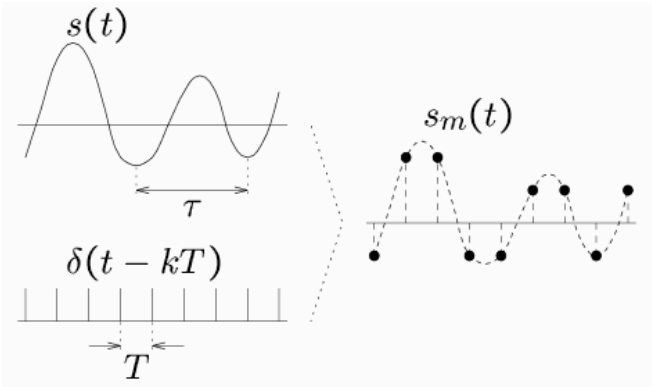
Adquisición de la señal

Micrófono

Adquirimos la voz mediante un micrófono que transforma **ondas de presión** en **señal eléctrica** mediante un transductor.

Conversión A/D

La señal proveniente del micrófono es continua. Una etapa conversora A/D (analógico/digital) **discretiza** la señal a una **frecuencia** dada y con determinados niveles de **cuantificación**.

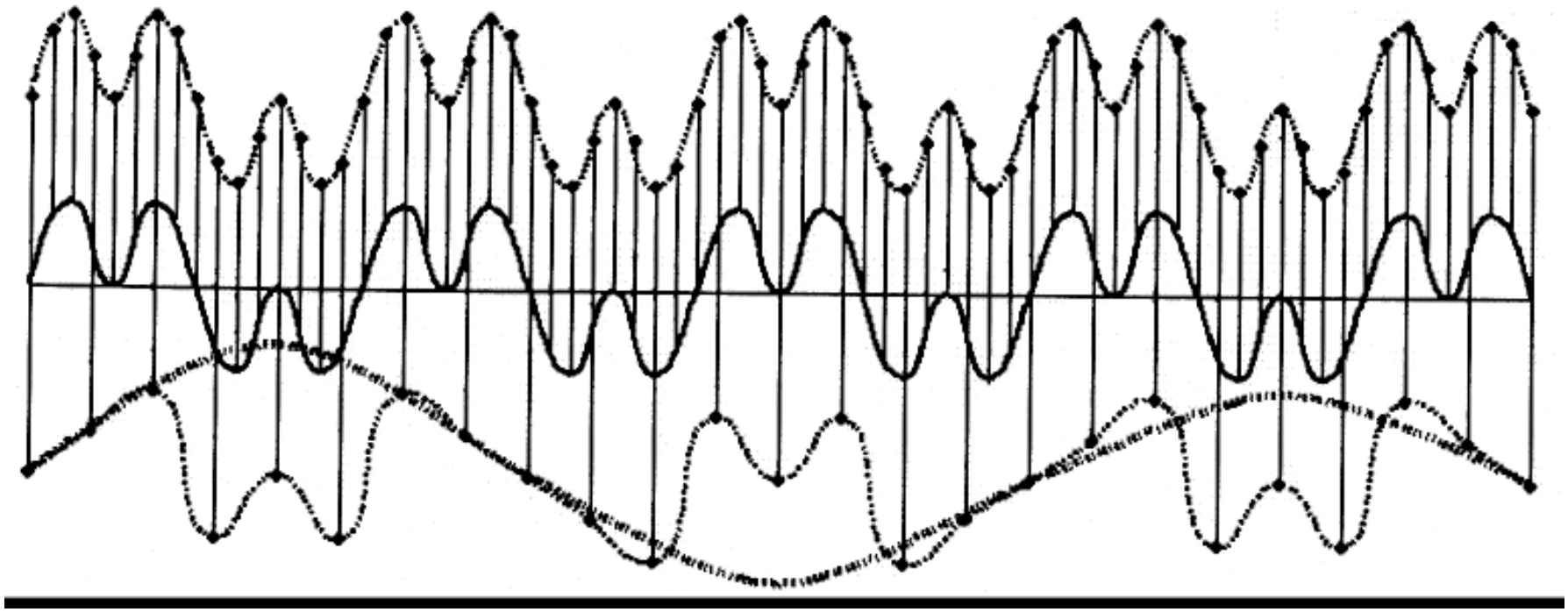


Discretización de la señal.

La señal muestreada es un vector de valores:

$$(s_0, s_1, s_2, \ldots).$$

Frecuencia La señal acústica es muestreada a una frecuencia F_s (*sampling frequency*). Por el teorema de Nyquist, F_s ha de ser al menos el doble de la máxima frecuencia presente en la señal a muestrear o aparecen fenómenos de "aliasing".



Fenómeno de "aliasing".

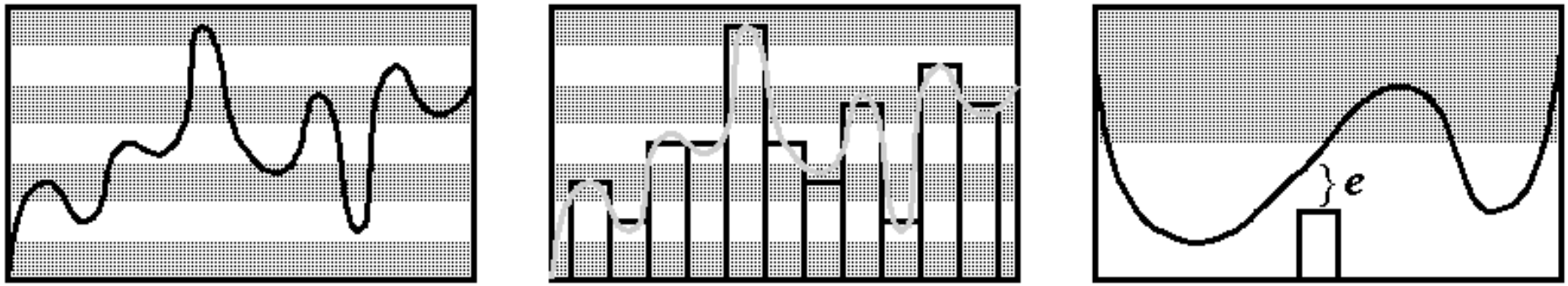
Si se escoge un valor de T (y por tanto una frecuencia F_s) que solapa los espectros repetidos, aparece el fenómeno de "aliasing".

La mayor parte de la energía en el habla se encuentra por debajo de 7 kHz.

- Típicamente $F_s = 16$ kHz en aplicaciones de reconocimiento de voz normales.
- En aplicaciones de reconocimiento de voz transportada sobre línea telefónica, la frecuencia de muestreo típica es de $F_s = 8$ kHz. (La energía está en la banda 300–3400 Hz.)

Cuantificación La cuantificación es el proceso de discretización de la señal continua.

Añade cierto ruido e a la señal s: hay errores de cuantificación.



Señal continua. Cuantificación. Error de la cuantificación.

El rango dinámico del oído es de aproximadamente 20 bits.

El número de bits por muestra es, típicamente, 16 (aunque 12 son suficientes para recoger el rango dinámico de la señal oral).

La señal telefónica tiene un rango dinámico de 12 bits, aunque cuantificados en 8 bits con una función de compresión no-lineal.

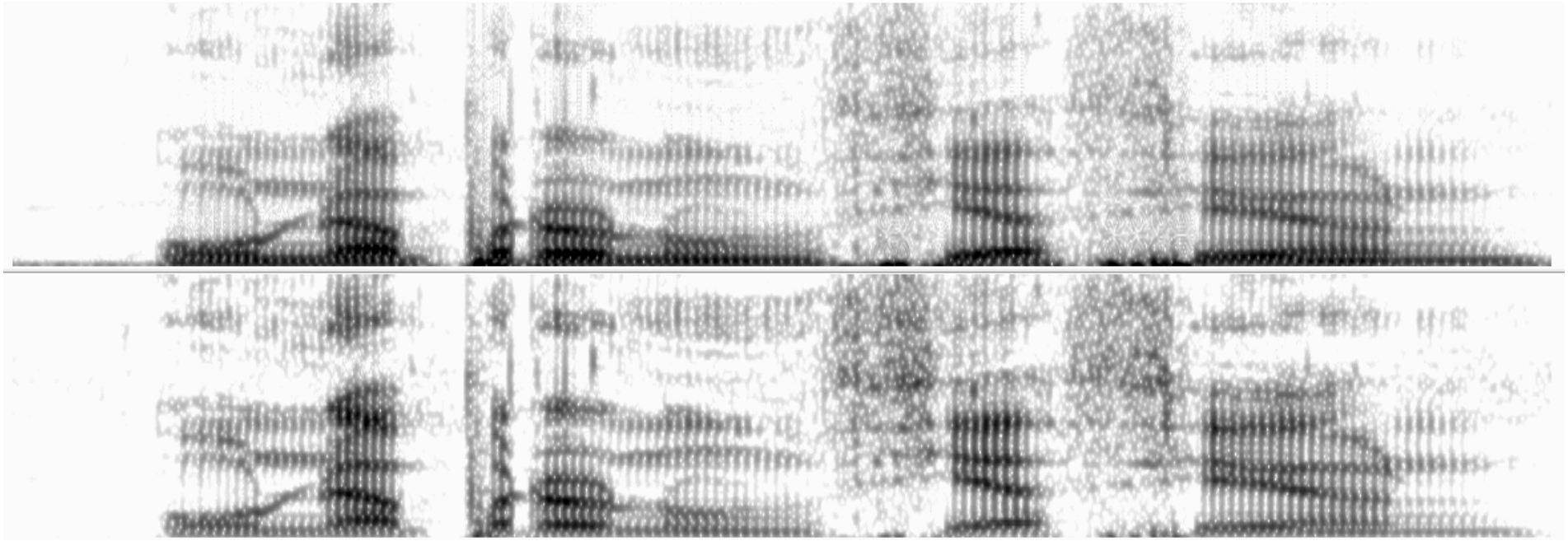
Preprocesamiento

Preénfasis

En el dominio de la frecuencia hemos observado que las características de la voz se manifiestan como *formantes* (picos debidos a resonancias del tracto vocal).

Los formantes de alta frecuencia tienen una amplitud menor que la de los formantes de baja frecuencia, aunque poseen información importante.

Es conveniente realzar las altas frecuencias con un proceso de *preénfasis*.



Sonograma de "una pronunciación". Arriba: sin preénfasis. Abajo: con preénfasis (factor $a = 0.97$).

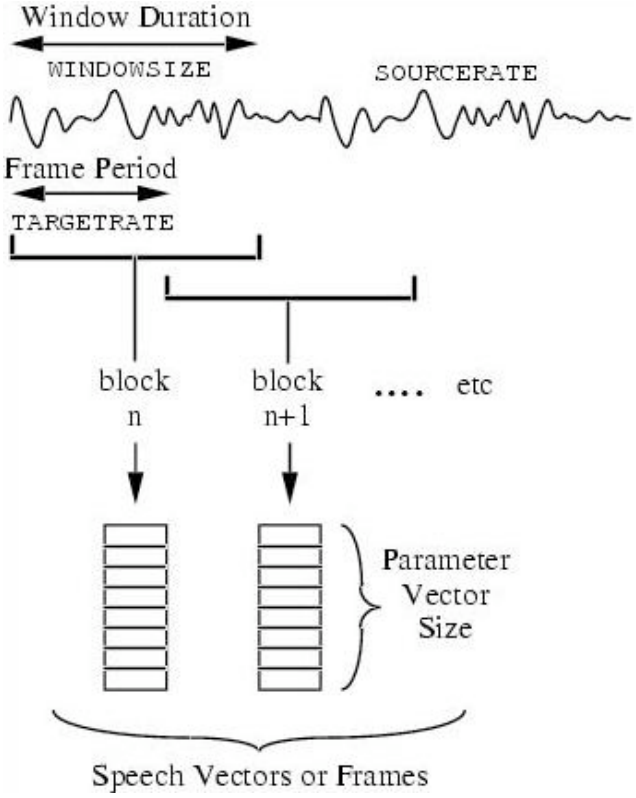
Otros preprocesamientos

Puede ser deseable efectuar otros tipos de preprocesamiento, como

- la cancelación de ruidos cuando estos pueden caracterizarse,
  - ruido ambiente,
  - ruido de motores,
  - ruido inducido por instalaciones eléctricas,
  - ...
- eliminación de componentes continuas en la señal,
- ...

Parametrización

Análisis en bloques

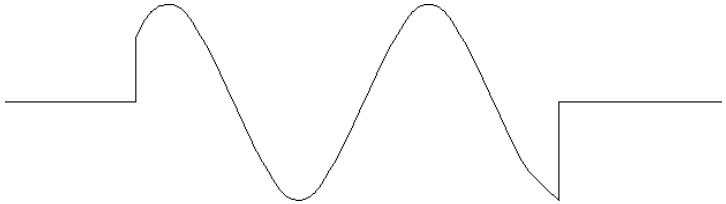


Fragmentación en bloques

Se "trocea" la señal en bloques contiguos con ciertos solapamiento. A la frecuencia con obtenemos los bloques se le denomina **frecuencia de submuestreo**. Con este proceso se convierte una señal en una **secuencia de vectores** (*frames*).

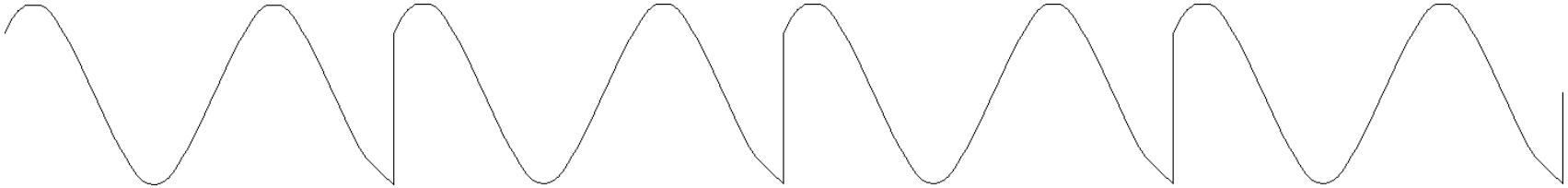
Aplicación de ventanas de suavizado espectral

La forma abrupta con que empieza y acaba cada bloque se traduce en una fuerte distorsión de los espectros resultantes.



Señal recortada.

Recuerda que se asume que la onda es periódica:

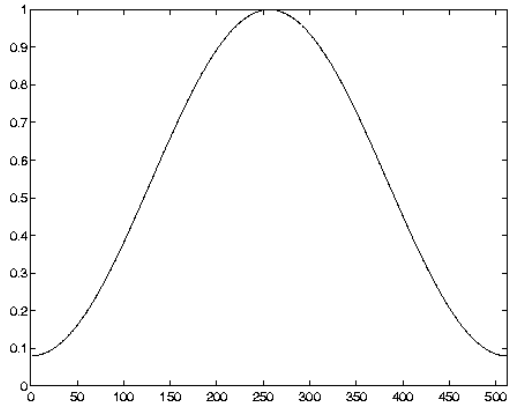


Visión de la señal recortada como función periódica.

Los "saltos" verticales se modelan espectralmente introduciendo un ruido que interfiere con el "espectro real".

El espectro se puede suavizar si aplicamos una *ventana* a cada bloque. Una ventana es una función de valor nulo fuera de cierto intervalo. Una familia de ventanas utilizada es la de Hamming generalizada:

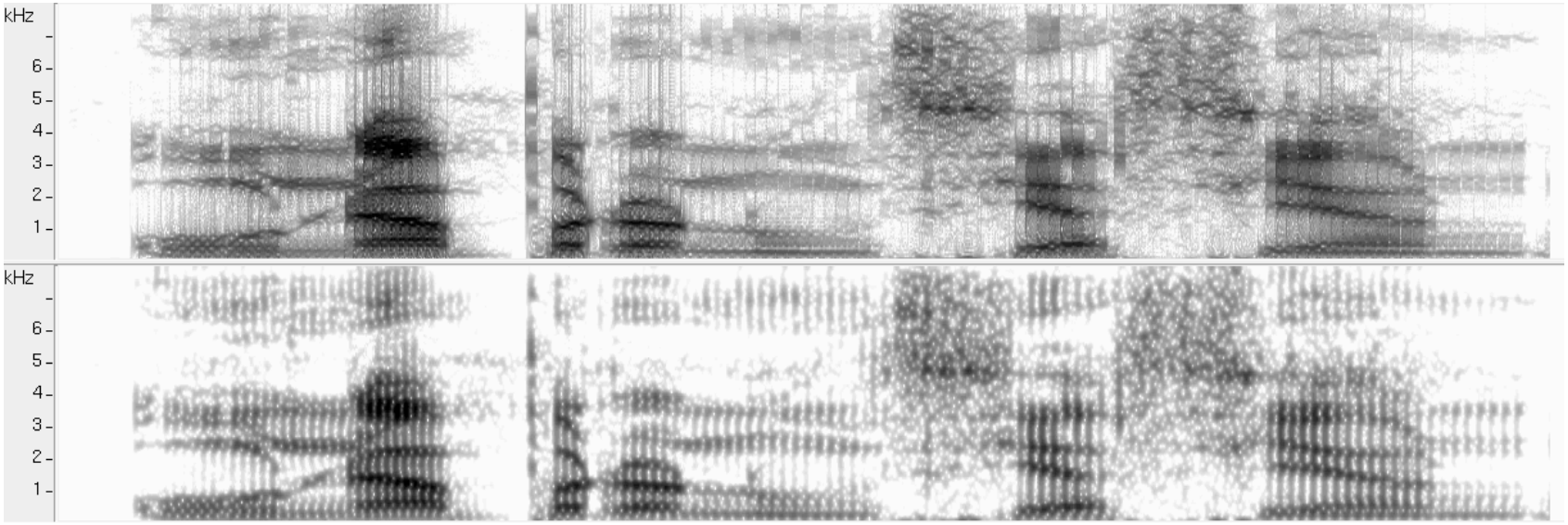
- Ventana de Hamming: si $\alpha = 0.54$. Es la utilizada normalmente.



Ventana de Hamming

- Ventana de Hanning (o de Hann): si $\alpha = 0.5$.

Ambas ventanas aplican pesos menores a los extremos del bloque.



Senonograma de "una pronunciación". Arriba: sin ventana de suavizado. Abajo: con ventana de Hamming

La ventana se aplica entre la fragmentación en bloques y el cálculo de la FFT.

Algunas características espaciales

Características espaciales son las que se obtienen de la señal en el dominio del tiempo. Algunas tareas de clasificación simple pueden resolverse considerando únicamente características espaciales.

- **Amplitud** (no muy útil) y envolvente, que es correlacionada con la **energía**. La energía permite discriminar voz y silencio.
- Tasa de **cruces por cero**: permite distinguir sonido débil de silencio. Sonidos como la /s/, por ejemplo, presenta una alta tasa de cruces por cero.
- **Autocorrelación**: Permite distinguir señales sordas de sonoras (que están muy autocorrelacionadas).
- Secuencia de **pendientes para los cruces por cero**.

Análisis en el dominio de la frecuencia: transformada discreta de Fourier

Se ha demostrado que conviene trabajar en el dominio *frecuencial* en lugar de en el dominio *temporal*.

La voz (función periódica) es una superposición de ondas senoidales.

Se puede calcular la **transformada discreta de Fourier** de cada fragmento "ventaneado" así:

$$s(t) = \frac{1}{N} \sum_{n=0}^{N-1} S(e^{i\omega n}) e^{i\omega n}.$$

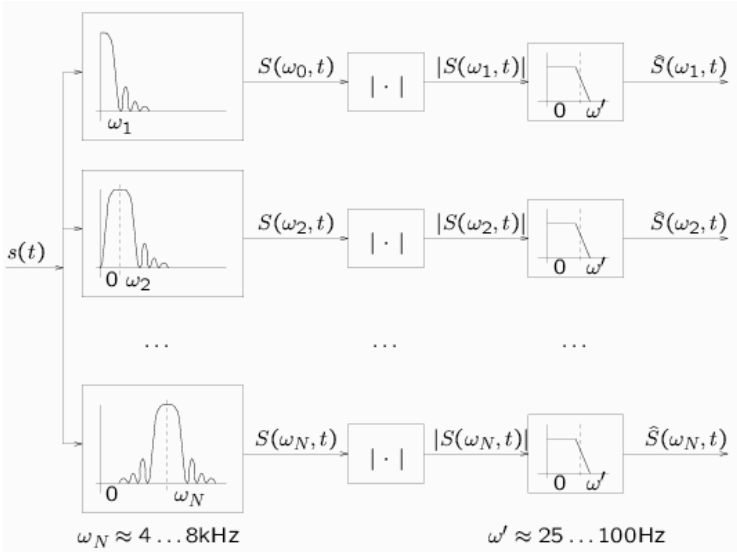
Los valores complejos

$$S(e^{0 \cdot i \frac{2\pi}{N}}), S(e^{i \frac{2\pi}{N}}), S(e^{2i \frac{2\pi}{N}}), \ldots, S(e^{(N-1)i \frac{2\pi}{N}}).$$

son los denominados coeficientes de Fourier o la transformada discreta de Fourier (DFT) de la señal $s(t)$. Los coeficientes de Fourier nos proporcionan una descripción de la señal en términos de funciones elementales (senos y cosenos).

El algoritmo FFT

El cálculo de TDF podría efectuarse mediante técnicas analógicas:



La TDF es la salida de un banco de filtros.

Dado que disponemos de una resolución de N frecuencias para ω, el cálculo trivial de un espectro en un computador es $O(N^2)$.

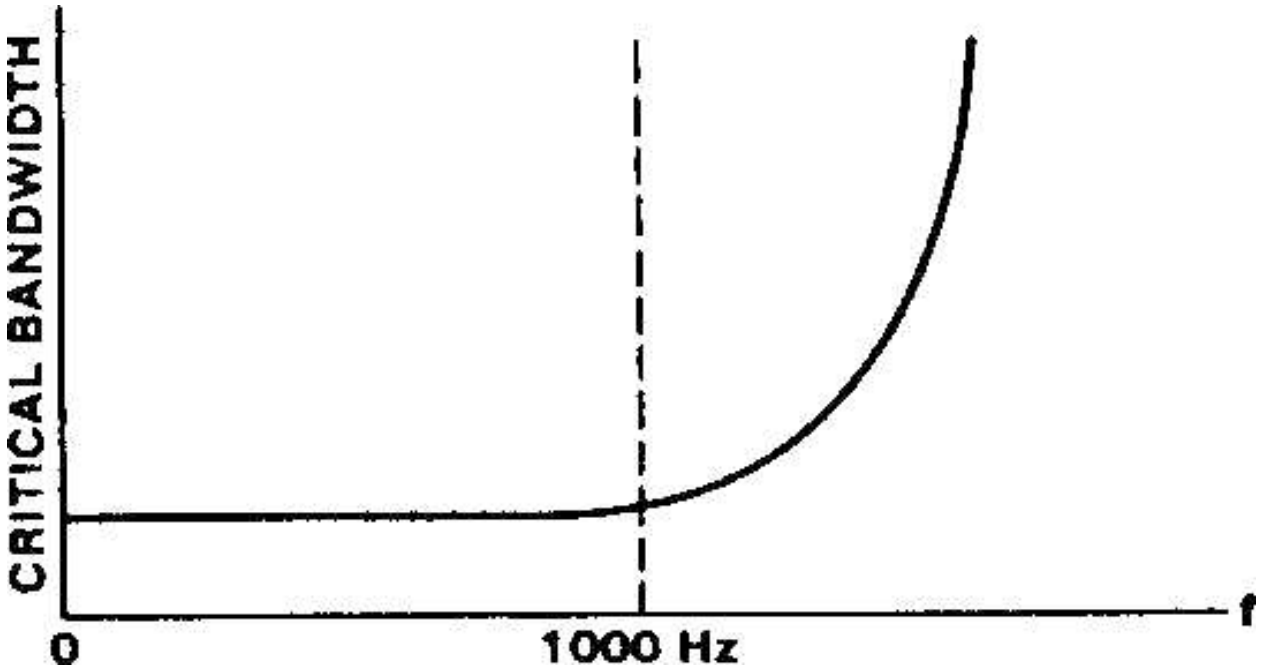
El algoritmo FFT (Fast Fourier Transform) sigue una estrategia "Divide y Vencerás" para calcular la transformada de Fourier en $O(N \log(N))$ Las versiones eficientes se basan en una transformación iterativa del algoritmo recursivo.

La FFT es susceptible de una implementación eficiente en hardware. La implementación de la FFT no es trivial. Afortunadamente hay muchas implementaciones eficientes disponibles (por ejemplo, FFTPACK).

Análisis con banco de filtros

La FFT proporciona un **banco de filtros uniforme**. El oído humano no discrimina todos los rangos de frecuencias por igual. Un sistema que reproduzca la resolución no lineal de frecuencias del oído humano mejorará la calidad del reconocimiento:

Se usa un **bancos filtros en escala logarítmica** (similar a la del oído humano):



Anchos de banda en la escala perceptual.

La escala es lineal a 1 KHz y logarítmica a partir de ella. La escala de Mel se asemeja a este

modelo perceptual.

Cepstrum

Los valores proporcionados por el banco de filtros se hallan fuertemente correlacionados. Esto da problemas cuando trabajamos con modelos estadísticos.

Un modo de descorrelacionarlos es calcular la transformada inversa sobre el logaritmo de la salida del banco de filtros.

Los coeficientes de esta transformación ya no presentan la correlación con el filtro. Son los denominados **coeficientes cepstrales** o **cepstrum**.

El resultado de todo el proceso se conoce por **MFCC** (Mel Frequency Cepstrum Coefficients) o **cepstrum**. La señales caracterizadas por combinaciones de armónicos se analizan mejor con el cepstrum. Por ejemplo, el cepstrum enfatiza los formantes vocálicos incluso en presencia de ruido.

El coeficiente c_0 es prácticamente la energía-log de la trama. Este coeficiente se suele calcular directamente sobre la señal. Los coeficientes cepstrales proporcionan un suavizado del espectro.

- Los coeficientes de la "parte baja" representan la macroestructura del espectro.
- Los coeficientes de la "parte alta" representan la microestructura del espectro.

Nota: hará falta cierta normalización de los coeficientes para igualar su "peso" en la descripción de la señal. La modificación de pesos en el **ceps**trum se denomina **lif**tering.

Típicamente se trabaja con, a lo sumo, 15 coeficientes cepstrales (normalmente, 10 o 12).

LPC

Hay un *modelo distinto* de parametrización. Se basa en la idea de que la muestra s_n se puede aproximar con una combinación lineal de las p últimas muestras: Linear Prediction Coefficients (LPC).

$$s_n = a_0 + a_1 s_{n-1} + a_2 s_{n-2} + \cdots + a_p s_{n-p}.$$

La motivación es que la función de transferencia de un tubo sin pérdidas puede representarse con un modelos todo-polos, pero hay que tener en cuenta que:

- El tracto vocal no está compuesto por cilindros y no es un sistema de tubos sin pérdidas.
- Hay una "fuga": la cavidad nasal.
- Las fricativas se producen al final del sistema de tubos (en los dientes/labios).

Aun así, con suficientes parámetros, el modelo LPC aproxima razonablemente la envolvente espectral de los sonidos.

- Los LPC son un buen modelo de la señal vocal, especialmente en las zonas cuasi-estacionarias del habla.
- Proporciona una buena separación de la señal en fuente y tracto vocal.

- Es sencillo de implementar y menos costoso que la aproximación basada en el banco de filtros.
- Los reconocedores basados en LPC proporcionan resultados comparables a los basados en el modelo del banco de filtros.

Conversión de los LPC a LPC cepstrales

De nuevo es necesario descorrelacionar los coeficientes obtenidos, especialmente si vamos a recurrir a modelos estadísticos a la hora de efectuar reconocimiento.

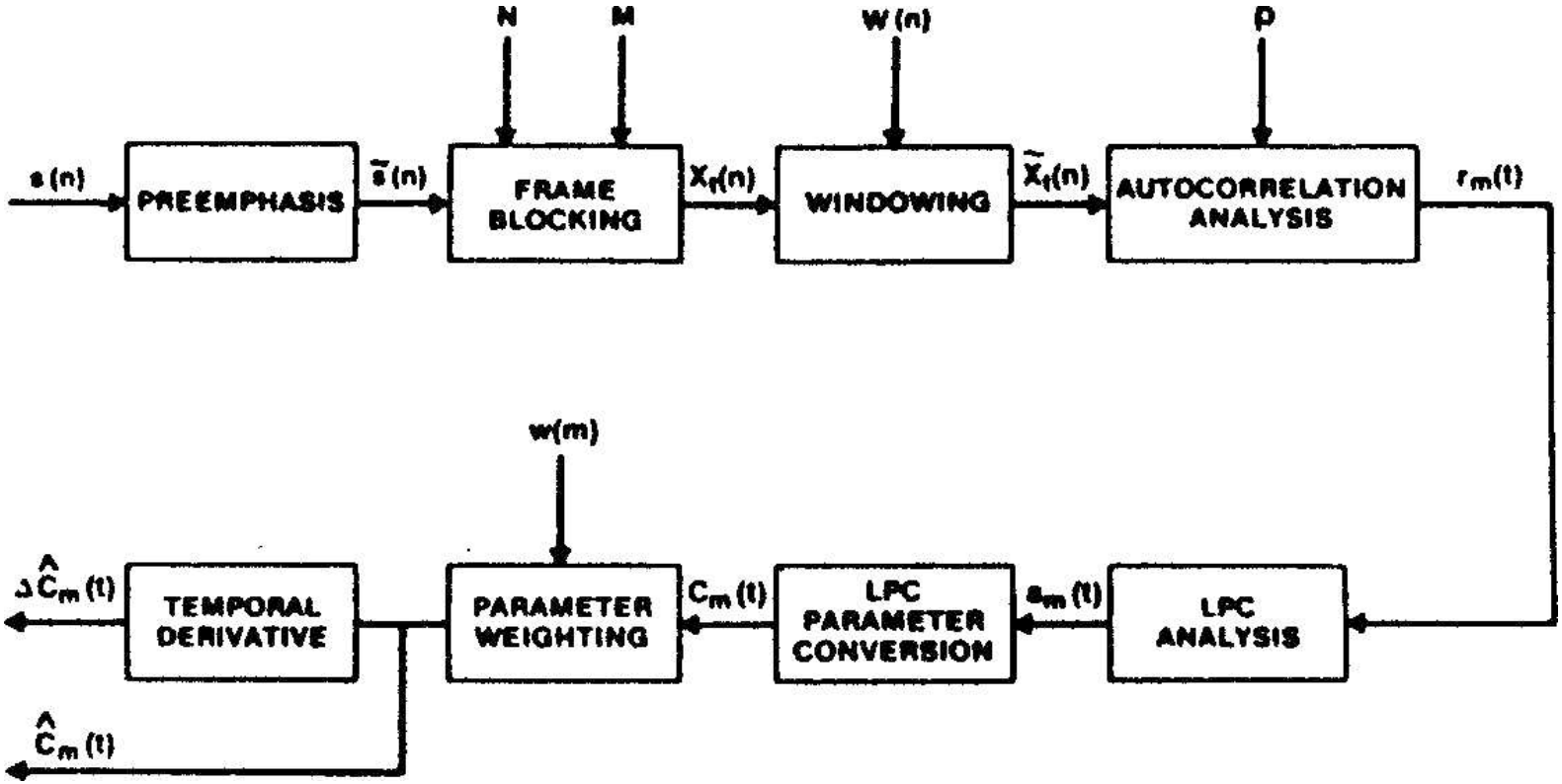
Coeficientes Delta y de Aceleración

Tanto si hemos seguido la aproximación del banco de filtros como la del cálculo LPC, las prestaciones de los sistemas de reconocimiento mejoran sustancialmente si se añaden las derivadas temporales de los parámetros considerados.

La derivada se calcula tomando las diferencias de los valores en una ventana.

Típicamente $K = 3$ para derivadas de primer orden. Si hemos obtenido Q parámetros, el resultado final es, pues, un vector con $3Q$ elementos:

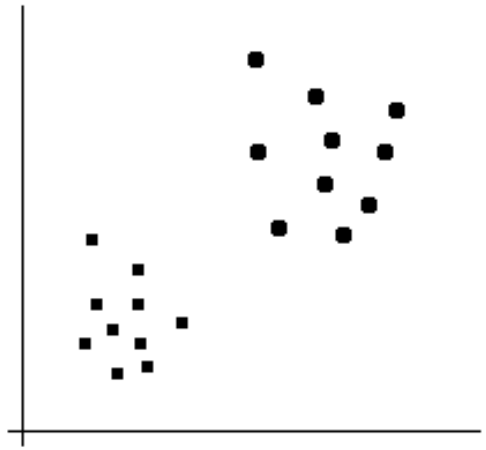
El proceso completo



Proceso completo con LPC

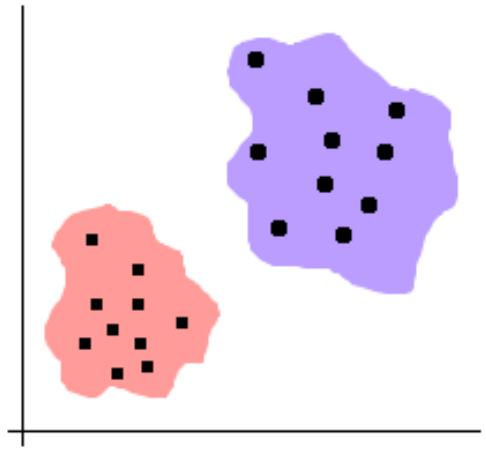
Cuantificación vectorial

Es posible reducir aún más la cantidad de información. Es de esperar que sonidos similares queden descritos por vectores similares.



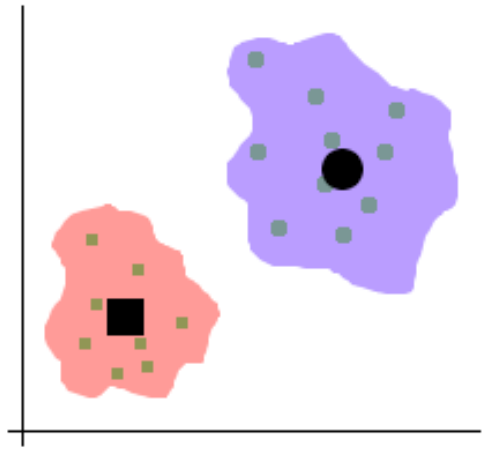
Cada cuadrado y cada círculo es un vector de parámetros (bidimensional, por simplificar).

Ya que tenemos una sucesión de vectores, podríamos tratar de detectar grupos de vectores similares...



Los vectores forman grupos homogéneos por su relación de vecindad.

... y codificar la voz como una secuencia de etiquetas que identifican a unos **prototipos**.



El cuadrado grande y el círculo grande pasan a representar a cualquier vector de su "área de vecindad". Cuando vemos un nuevo vector, lo clasificamos como perteneciente al área de uno de los prototipos y se sustituye por la etiqueta correspondiente ("cuadrado" o "círculo").

Si, por ejemplo, encontramos 1024 grupos de vectores similares (el 1024 lo podemos fijar nosotros), estaremos reduciendo la cantidad de información a 1000 bps.

Ventajas:

- Se necesita **menos espacio** para almacenar la voz.
- Se **reduce** la necesidad de **cálculo** a la hora de comparar vectores (se puede reducir a una búsqueda en una matriz precalculada).

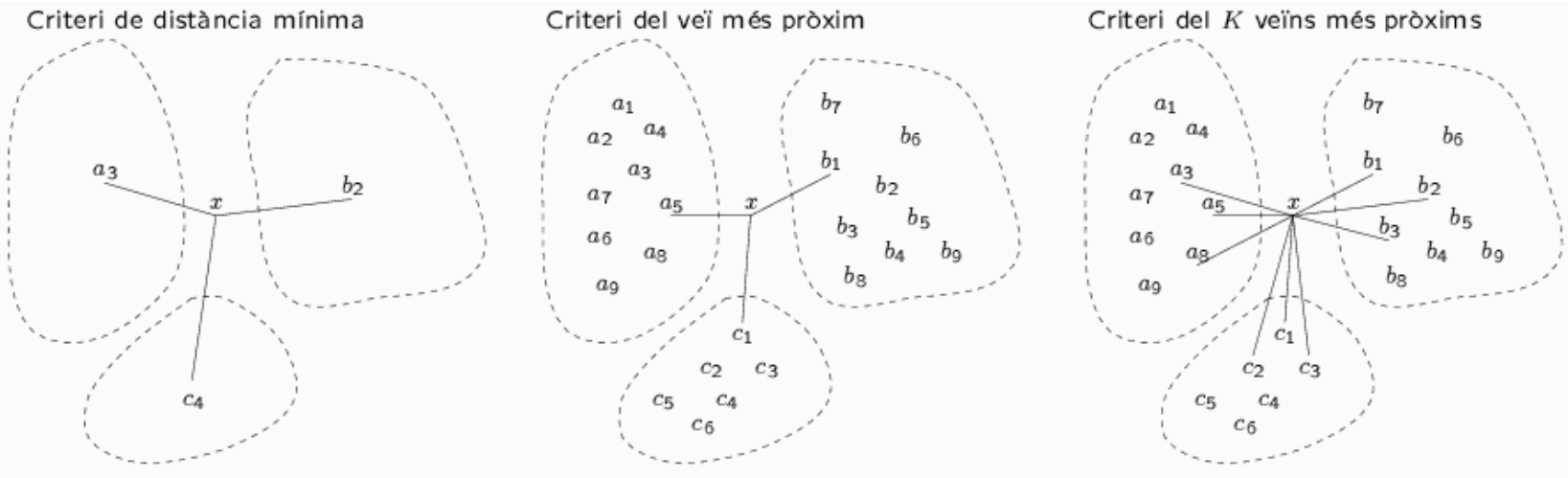
Desventajas:

- Ha de haber una **fase de entrenamiento** en la que se determina el diccionario de prototipos (*codebook*).
- Se introducen **errores en la cuantificación** y, por tanto, se pierde "resolución".
- La **memoria necesaria** para almacenar los vectores prototipo y la matriz de distancias **puede ser considerable**.

Podemos sustituir cada vector por la etiqueta del prototipo en cuya región lo clasifiquemos:

- ¿Hay técnicas que permitan clasificar rápidamente en la región correspondiente?
- ¿Qué criterio de distancia usar?
- ¿Cómo encontrar "buenos" prototipos?

Se pueden mejorar las prestaciones representando cada clase (región) con un *conjunto* de prototipos y buscar los k vecinos más próximos.



Criterios de clasificación basados en vecindad.

Hay abundante literatura sobre técnicas de clasificación rápida.

Algunos criterios de distancia

Distancia entre vectores:

- Distancia de Manhattan:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|.$$

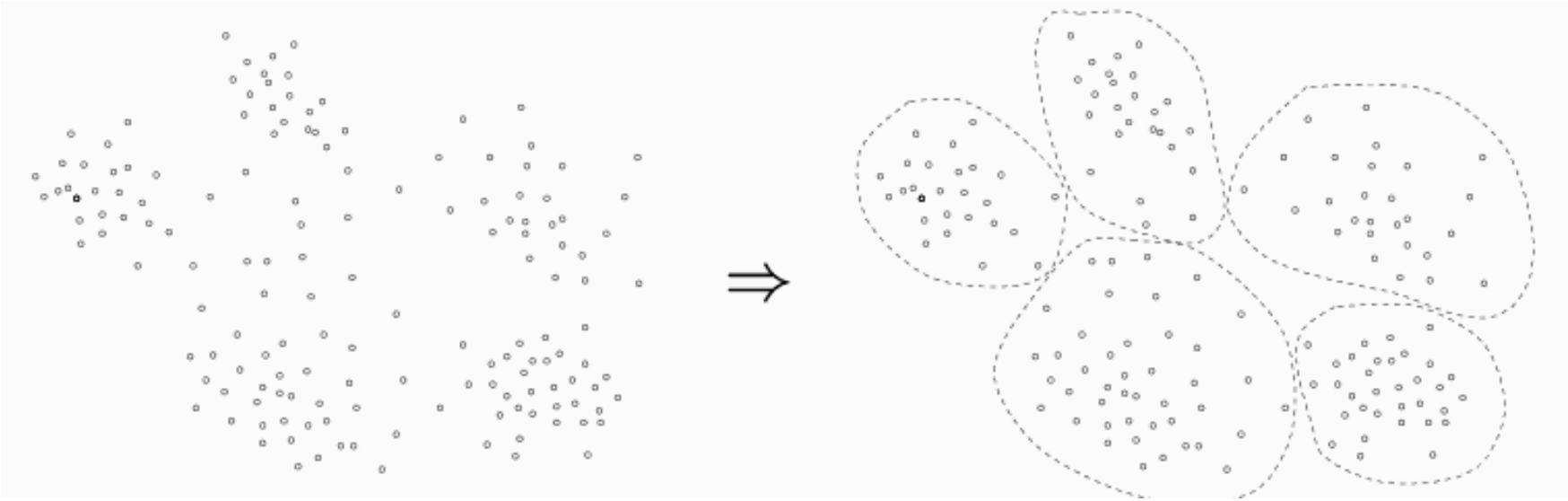
- Distancia de Hamming (asimétrica):

$$d(x, y) = \sum_{i=1}^{n} x_i - y_i.$$

- Distancia Euclídea:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

El algoritmo de las K-medias ¿Cómo seleccionamos el conjunto de prototipos? El algoritmo de las K-medias encuentra automáticamente un conjunto de K prototipos.



Agrupamiento automático.

Tenemos L vectores que deseamos agrupar en K grupos representados por sendos prototipos.

1. **Inicialización**. Escoger K vectores al azar de entre los L como *codebook* inicial.
2. **Búsqueda del vecino más próximo**. Para cada vector, encontrar el prototipo *más próximo* (en términos de distancia espectral).
3. **Actualización de centroide**. Actualizar el prototipo de cada conjunto de vectores clasificados en una misma clase con el vector que minimiza la suma de distancias a los demás (el centroide).
4. **Iteración**. Repetir los pasos 2 y 3 hasta que la distancia promedio caiga por debajo de algún umbral predeterminado.

Complejidad: $O(IKLn)$, donde I es el número de iteraciones y n es la dimensión de los vectores.

Hay otras técnicas en las que no se precisa determinar L a priori.

Clasificación con mixturas de Gaussianas

Otra posibilidad es asumir que cada conjunto de prototipos sigue una distribución Gaussiana ($\mathcal{N}(\mu, \Sigma)$). Entonces podemos caracterizar cada clase por la media μ y la matriz de covarianzas Σ.

Distancia de un vector a una clase modelada por una distribución Gaussiana:

- Densidad Gaussiana (probabilidad de que x pertenezca a la clase):

$$d(x, \mathcal{N}(\mu, \Sigma)) = \frac{1}{\sqrt{(2\pi)^n \cdot |\Sigma|}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}$$

- Distancia de Mahalanobis (distancia normalizada por la varianza en cada dimensión):

$$d(x, \mathcal{N}(\mu, \Sigma)) = (x-\mu)'\Sigma^{-1}(x-\mu).$$

Otra posibilidad aún mejor es asumir que cada conjunto se puede modelar con una **mixtura de Gaussianas** (una combinación lineal de Gaussianas). Las mixturas pueden modelar distribuciones multimodales.

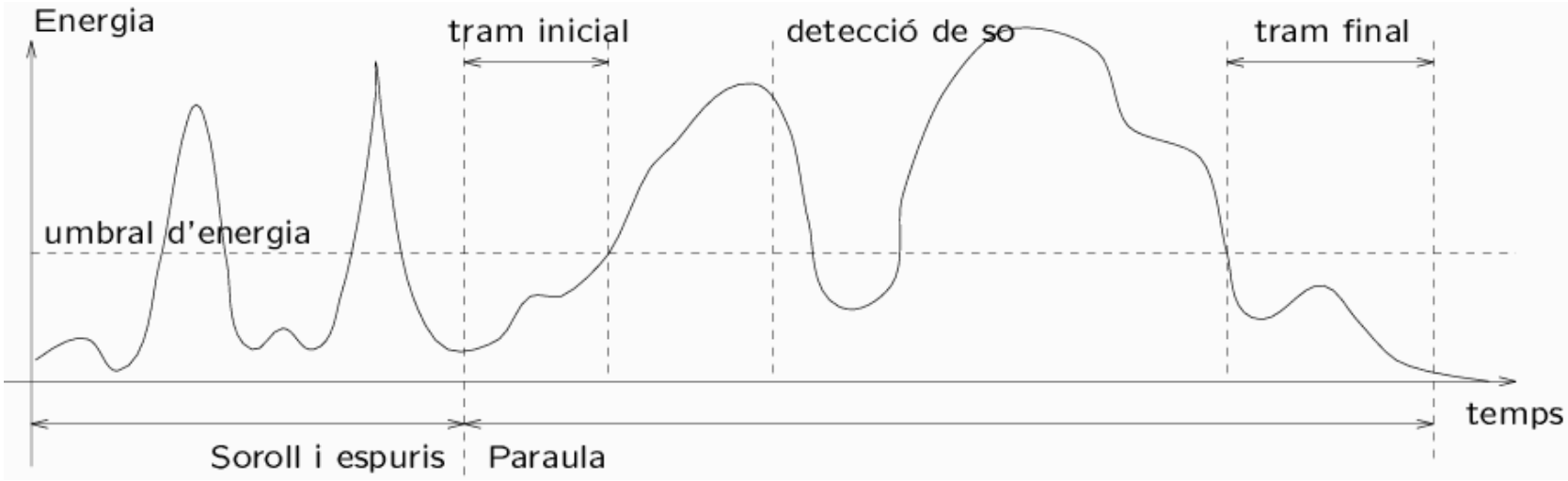
Detección de voz/silencio

Hasta el momento hemos estudiado como describir la voz, pero ¿cómo saber dónde empieza y acaba ésta? Un sistema de reconocimiento debe "escuchar" permanentemente la entrada y determinar cuándo hay voz que reconocer.

Estudiando únicamente la energía podemos decidir si hay voz o silencio. Pero no basta con analizar una trama:

- Un valor alto de energía en unas pocas tramas podría deberse a un ruido espúreo.
- Un valor bajo de energía en unas pocas tramas podría deberse a la pronunciación de una explosiva o a una breve pausa en la pronunciación.

Suele implementarse un autómata que determina que hay voz cuando un número n de tramas presenta una energía superior a un umbral α y que una pronunciación ha finalizado cuando m tramas presentan una energía inferior a un umbral β.



Detección de silencio/señal.

¡Ojo! Cuando detectamos que hay voz, ésta ya ha empezado hace n tramas. Es necesario disponer de un buffer que almacene las tramas "perdidas" para que el reconocedor las trate.

Bibliografía

- Lawrence Rabiner, Biing-Hwang Juang: *Fundamentals of speech recognition*. Prentice Hall. 1993.
- Tony Robinson: *Speech Analysis*. `http://svr-www.eng.cam.ac.uk/~ajr/SpeechAnalys`
- Francisco Casacuberta, Enrique Vidal: *Reconocimiento automático del habla*. Marcombo. 1987.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: *Introduction to algorithms* (2nd ed.). The Massachussetts Institute of Technology. 2001.
- Alan V. Oppenheim, Ronald W. Schaffer: `Discrete-time Signal Processing`. Prentice-hall. 1989.
- William H. Press, et al.: *Numerical Reciopes in C*. Cambridge University Press. 1993. (Disponible en `http://www.nr.com`.)