

# Separación de Fuentes por Análisis Estadístico

# Objetivo Principal

- A diferencia de las técnicas de Beamforming, estas técnicas no asumen mucho del ambiente.
  - Varias, si no es que todas, ni siquiera asumen saber el número de fuentes existentes.
    - Aunque su desempeño puede mejorar al saberlo.

# Objetivo Principal

- La idea es encontrar a  $W$ :

$$\hat{S} = W X$$

Donde:

$s_x$ : es la estimación de las señales de origen

$X$ : son las señales capturadas (en los micrófonos); cada renglón representa un micrófono

**$W$ : es la matriz de separación (*demixing matrix*)**

# Donde...

$$\mathbf{A} = \begin{bmatrix} e^{-i2\pi f T_{1:1}} & e^{-i2\pi f T_{1:2}} & \dots & e^{-i2\pi f T_{1:M}} \\ e^{-i2\pi f T_{2:1}} & e^{-i2\pi f T_{2:2}} & \dots & e^{-i2\pi f T_{2:M}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-i2\pi f T_{D:1}} & e^{-i2\pi f T_{D:2}} & \dots & e^{-i2\pi f T_{D:M}} \end{bmatrix} \quad \mathbf{s} = \begin{bmatrix} s_1(1) & s_1(2) & \dots & s_1(N) \\ s_2(1) & s_2(2) & \dots & s_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ s_M(1) & s_M(2) & \dots & s_M(N) \end{bmatrix}$$

$$\mathbf{X} = \mathbf{S} \mathbf{A}$$

Donde:

$s_x$ : es una señal de origen

N: tamaño de la señal (o de la ventana de la señal)

$T_{d:m}$ : es el retraso recibido de la señal  $s_m$  en el micrófono d

**A: es la matriz que contiene los vectores de dirección (*direction vectors*)**

X: son las señales capturadas (en los micrófonos); cada renglón representa un micrófono

# Técnicas Populares

- Principal Component Analysis
- Independent Component Analysis
- Non-negative Matrix Factorization

# Principal Component Analysis (PCA)

# Principal Component Analysis

- Se asume que las señales de origen no están correlacionadas.
  - Visto de otra manera: que casi no se empalman.
    - Primer luz roja.
- Con esta suposición, propone a la  $W$  como aquella que pueda desmezclar a  $X$  en señales que son ortogonales.

# Principal Component Analysis

- Ortogonalidad:
  - Los eigenvectores de una matriz son ortogonales entre sí.
- PCA propone utilizar los eigenvectores de la matriz de covariancia como  $W$ .
  - Como los utilizados para MUSIC.
- El resultado es la descomposición de  $X$  en sus *componentes principales*.
  - Que se asume **algunas** de ellas son las señales de origen.



# Principal Component Analysis

Centramos las señales de entrada  $X$ , y así  $R$  sería la matriz de covariancia.

$$R = XX^T$$

$$R = V\Lambda V^{-1} \quad V = [v_1 \quad v_2 \quad \cdots \quad v_K] \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_K \end{bmatrix}$$

$$\hat{S} = V' X$$

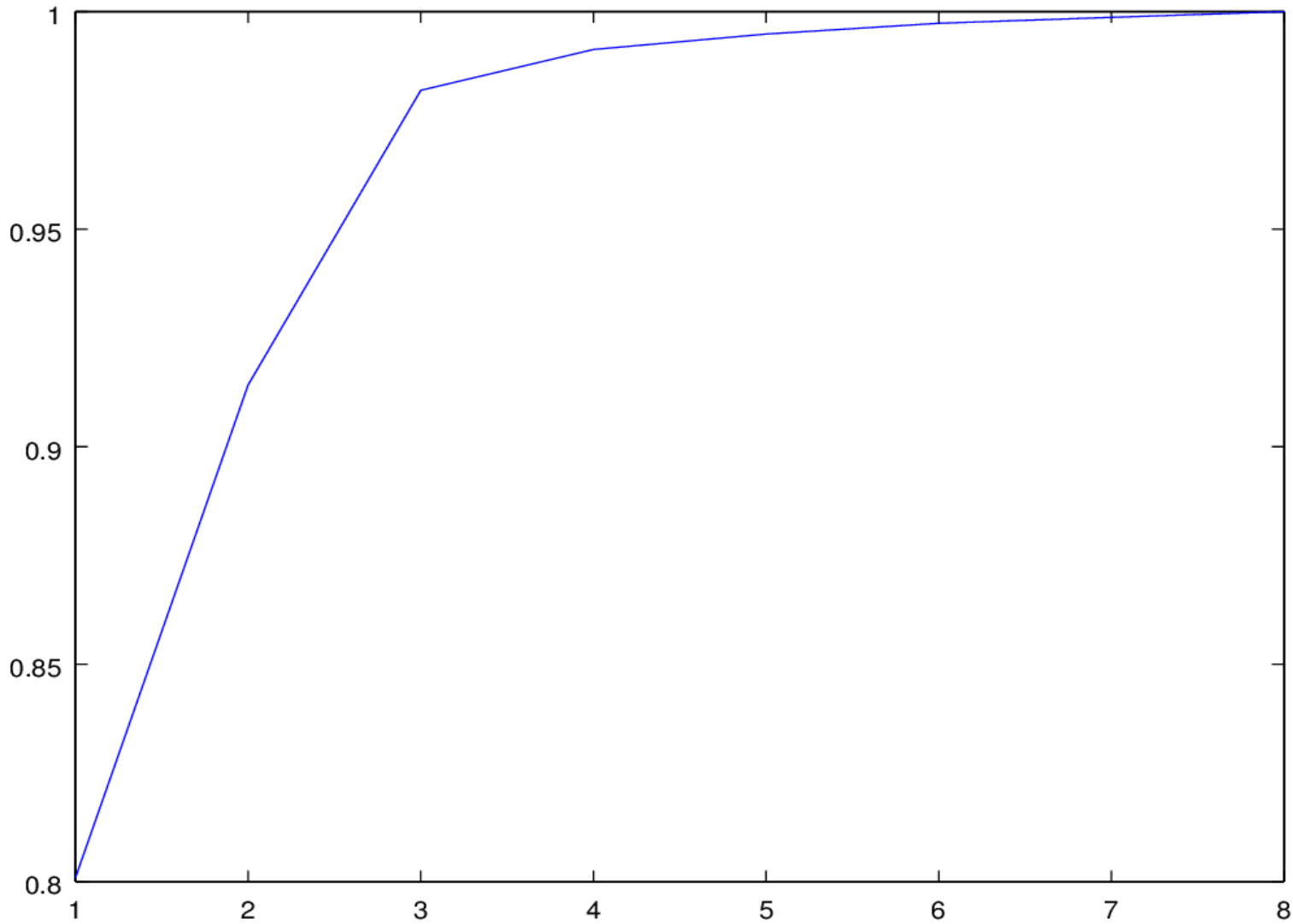
$\hat{S}$ : los componentes principales de  $X$ ; un renglón por componente.

**Tendrá tantos componentes como señales capturadas.**

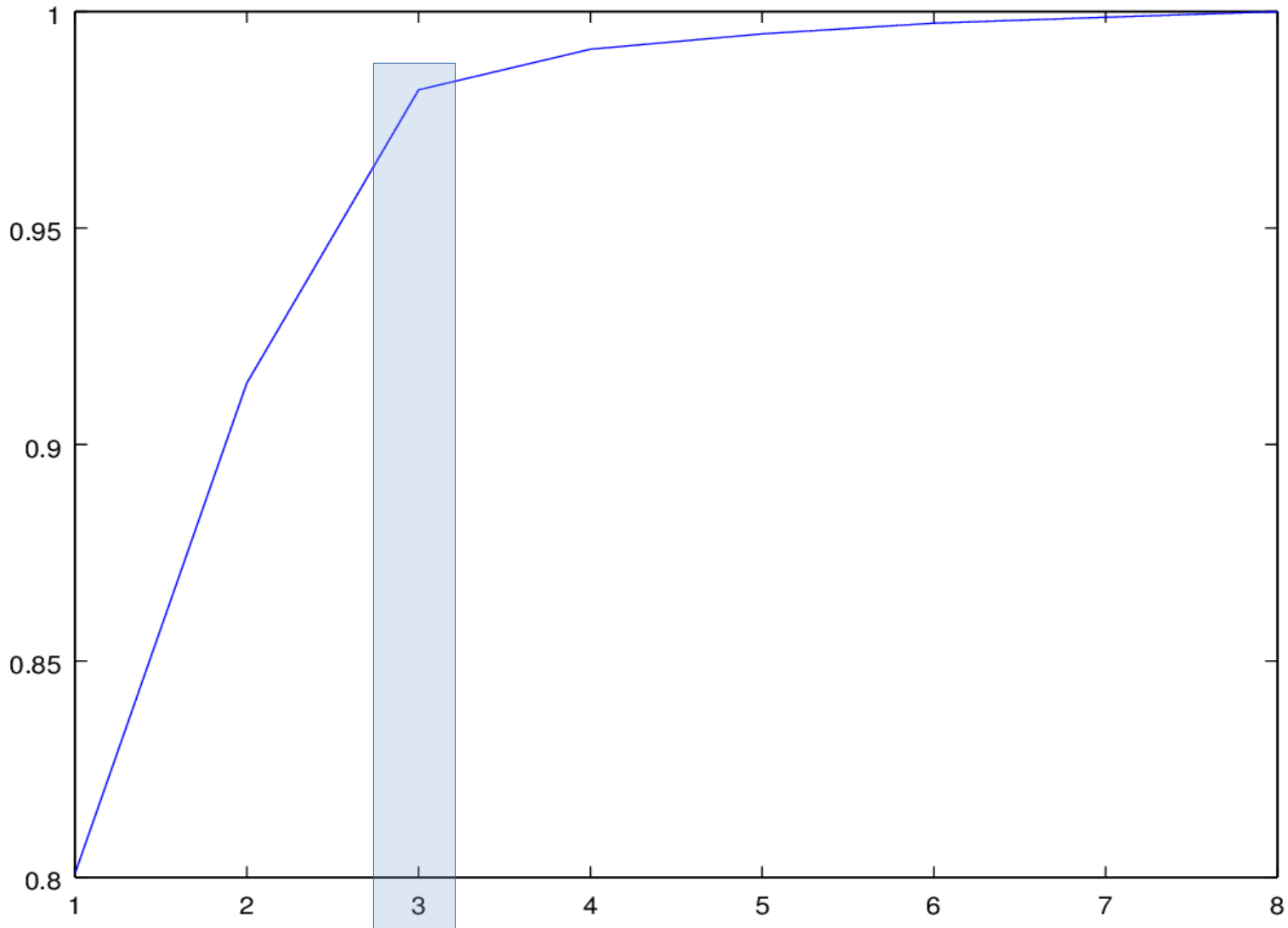
# Tantos componentes como señales capturadas...

- ¿Cómo sabemos cuáles componentes son las estimaciones de las señales de origen?
  - Los eigenvalores es un número que representa la cantidad de información que contiene el eigenvector de la matriz.
  - Por lo tanto, normalmente los eigenvectores se ordenan en orden de su eigenvalor (como lo hicimos con MUSIC).
  - Y luego se presentan de manera acumulativa, divididos por su suma:
    - Si  $D$  es el arreglo de eigenvalores ordenados:  
`plot(sumcum(D)/sum(D))`

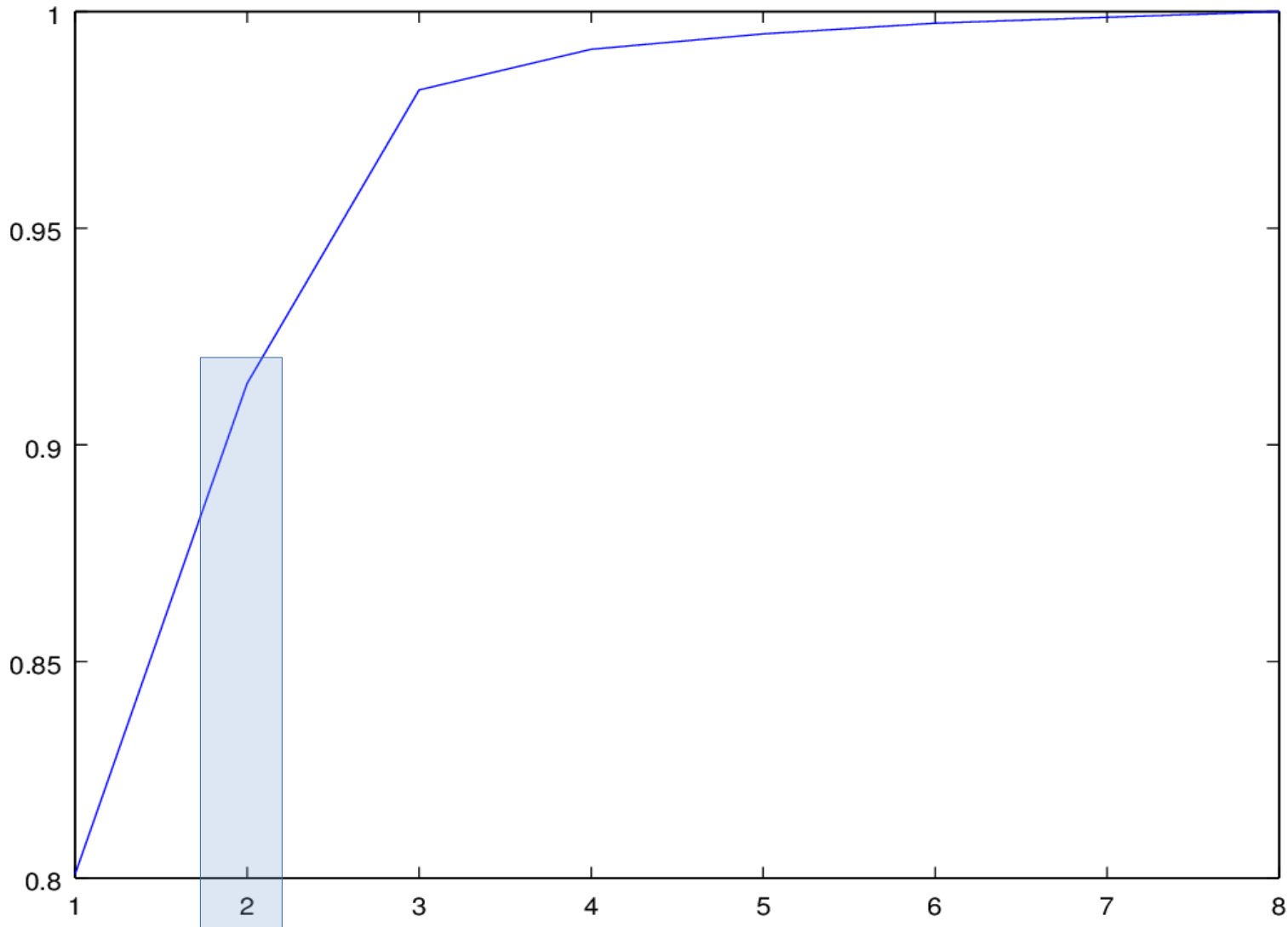
# Presentación de Eigenvalores



# Presentación de Eigenvalores



# Presentación de Eigenvalores



# ¿En serio?

- Desgraciadamente, la decisión de cuántos eigenvectores utilizar es un tanto abierto.
- Para nuestros propósitos dicha decisión se podría simplificar:
  - Si estamos estimando los DOAs anteriores a este paso, también podemos saber cuántas señales de origen hay en el ambiente.
  - Desgraciadamente...

# Ejercicio #1

- Descarguen:  
pca.m
- Contiene una implementación de PCA:
  - Figura 1: las señales de origen
  - Figura 2: las señales de origen superpuestas con el **primer** y **tercer** componente.
  - Figura 3: una señal de origen superpuestas con el **segundo** componente.

# Ejercicio #1

- ¿Qué problema ven?



# Ejercicio #1

- ¿Qué problema ven?
- Los componentes no están completamente bien.
  - Parecen distorsionados, sin haber presencia de ruido.
  - Esto es porque hay algo de empalme entre ellos.
  - Pero no está mal.
    - Es importante mencionar que esta implementación magnifica los componentes al final, para mejor visualización.
- *Los componentes que son las estimaciones no son los dos con el mayor eigenvalor.*

# Principal Component Analysis

- La selección de componentes a señal de origen estimada es el mayor problema de PCA:
  - Recordemos que normalmente no vamos a tener las señales de origen como referencia.
  - Por lo que no podemos saber si un componente es una estimación de una señal de origen, o sólo un componente de algún subespacio escondido.

# Independent Component Analysis (ICA)

# Independent Component Analysis

- Es el “primo lejano” de PCA.
- Su suposición es que los componentes son **independientes** entre sí.

# Independencia

- Ejemplo: hay una gran correlación entre el siguiente grupo de personas:
  - Aquellas que han estado en un accidente de carro.
  - Aquellas que se lavan los dientes.

# Independencia

- ¿Si no me lavo los dientes tendré menos posibilidades de estar en un accidente de carro?
  - No, porque estos dos grupos no son **dependientes**.
    - No hay una *liga de causalidad* entre el acto de lavarse los dientes y estar en un accidente de carro.
  - Es posible que haya un factor que las ligue, y por ende estén correlacionados:
    - Lavarse los dientes es un acto popular en los humanos.
    - Los humanos a veces son dueños de carros.
    - Los carros algunas veces son objeto de accidentes.
  - Pero es coincidencia, no causalidad.

# Independent Component Analysis

- La idea entonces es obtener una matriz de separación  $W$  tal que al aplicar a  $X$ , los componentes que resulten sean lo más independientes posibles.
  - Llamados Componentes Independientes.

# Independent Component Analysis

- Independencia implica no-correlación:
  - Los componentes siguen teniendo que ser poco correlacionados.
    - Por lo que tampoco le gusta el empalme.
- Pero ese no es su mayor problema...



# Medir la Independencia

- Hay varias formas de hacerlo, pero una muy popular es:
  - La reducción de **información mutua** entre las señales
  - La cual a su vez se puede medir por medio de la diferencia entre la *entropía de una señal dado la otra* y la *entropía conjunta de las dos señales*.
    - También conocida como la **entropía diferencial**.

# ¡¿Qué?!

- Fue exactamente lo que yo dije cuando leí ese enunciado.
- Sinceramente, podría entregarles toda una explicación de cómo implementarla, pero...

# Ejercicio #2

- Descarguen:  
ica.m
- Es una implementación de ICA:
  - Figura 1: señales de origen.
  - Figura 2: suponiendo que son sólo dos componentes independientes, aquí los muestra.

# Ejercicio #2

- Dura un rato en correr:
  - Busca la matriz de separación de una manera iterativa.
- Hay una variable que pueden modificar:  
do\_not\_delay
  - Si tiene valor 1: no desfasa a las señales de origen, sólo les modifica su magnitud.
  - Si tiene valor 0: desfasa las señales.

# Ejercicio #2

- Correrlo con:  
do\_not\_delay = 1

# Ejercicio #2

- Correrlo con:  
`do_not_delay = 1`
- Aunque lento, hace mucho mejor trabajo que PCA.
- Requiere correrse varias veces para obtener resultados apropiados.
  - Para eso la línea “`rand('seed',0.0005)`”.

# Ejercicio #2

- Correrlo con:  
do\_not\_delay = 0

# Ejercicio #2

- Correrlo con:  
do\_not\_delay = 0
  
- ????????
- Ni siquiera se acerca...



# Independent Component Analysis

- Por eso, mejor ni se acerquen a esta técnica para estos propósitos.

# Non-negative Matrix Factorization

## (NNMF)

# Non-negative Matrix Factorization

- La suposición es:
  - No hay valores negativos.
  - En ningún lado.
  - Ni en los componentes separados, ni en la matriz de separación  $W$ .

# Non-negative Matrix Factorization

- Hacemos que  $X$  no tenga valores negativos:

$$X = X - \min(X)$$

- Proponemos inicialmente  $\hat{S}$  con valores al azar.
- Luego, repetimos los siguientes pasos *ad nauseam* o hasta que el error se minimize:

$$1) \mathbf{W} = \mathbf{X} \hat{\mathbf{S}}^T (\hat{\mathbf{S}} \hat{\mathbf{S}}^T)^{-1} \quad 3) \hat{\mathbf{S}} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{X}$$

$$2) \mathbf{W} (W < 0) = 0 \quad 4) \hat{\mathbf{S}} (\hat{S} < 0) = 0$$

- El error lo medimos por medio de:

$$error = \|\mathbf{X} - \mathbf{W} \hat{\mathbf{S}}\|$$

# Ejercicio #3

- Descarguen:  
    nnmf.m
- Es una implementación de NNMF, entregando el mismo tipo de figuras que ica.m

# Ejercicio #3

- Correrlo con:  
do\_not\_delay = 1

# Ejercicio #3

- Correrlo con:  
`do_not_delay = 1`
- Sólo entrega una buena estimación de una señal.
  - Ni así entrega buenos resultados.
    - Es una técnica que es muy sensible al empalme.

# Ejercicio #3

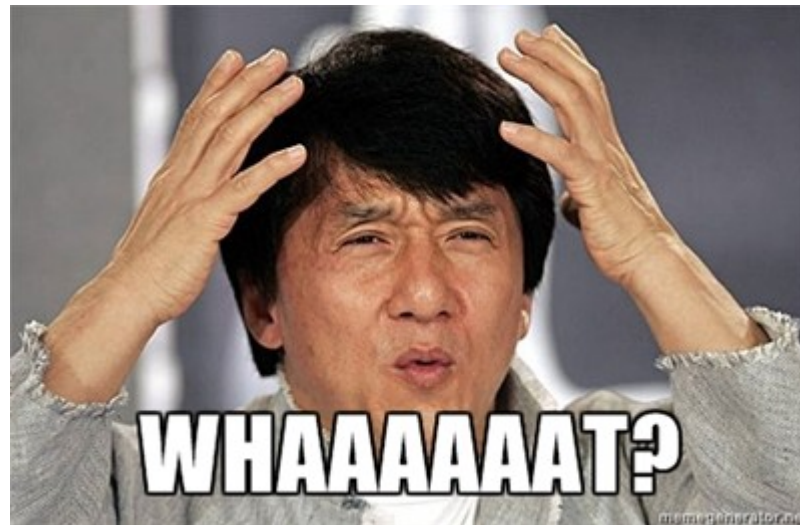
- Correrlo con:  
do\_not\_delay = 0



# Ejercicio #3

- Correrlo con:  
do\_not\_delay = 0
- ????????
- Peor aún...

¿Que está sucediendo?



# No están funcionando...

- Estas técnicas toman a cada renglón de  $X$  como una observación en forma de vector.
  - Y cada columna representa a **una dimensión de dichos vectores.**
- Suponen que las variaciones con las que van a lidiar, son sólo de magnitud en la misma dimensión entre señales.

# No están funcionando...

- En nuestro caso, estamos lidiando mayormente con desfasamiento.
- Esto implica que se están comparando magnitudes que no pertenecen a la misma dimensión.
  - Esto rompe desde la raíz a estas técnicas.
  - Inconsistencias del tipo magnitud-a-dimensión.

# Solución Tentadora: Versión en Frecuencia

- Las inconsistencias en el dominio del tiempo que vemos al aplicar un desfase **deberían** desaparecer en el dominio de la frecuencia.
  - ¿No?
- ¿Qué sucede si aplicamos estas técnicas en el dominio de la frecuencia?

# PCA en Frecuencia

- Descarguen:  
    pca\_fft.m
- Contiene una implementación de PCA con FFT:
  - Figura 1: las señales de origen
  - Figura 2: las señales de origen superpuestas con el **primer** y **tercer** componente.
  - Figura 3: una señal de origen superpuestas con el **segundo** componente.

# PCA en Frecuencia

- ¿Esto se ve conocido?
- Compárenlo con PCA en tiempo...

# PCA en Frecuencia

- Es igual.
  - Excepto por el segundo componente.
- ¿Nos podremos escapar del problema de desfase?
  - Probemos otra técnica en FFT...



# ICA en Frecuencia

- Descarguen:  
ica\_fft.m
- Contiene una implementación de ICA con FFT:
  - Figura 1: señales de origen.
  - Figura 2: suponiendo que son sólo dos componentes independientes, aquí los muestra.

# ICA en Frecuencia

## do\_not\_delay = 1

- Parece ser más rápido.
  - No se la crean...
- Sólo encontró un componente, y la “tendencia” del otro.
- Diferentes resultados que con ICA en el tiempo.

# ICA en Frecuencia

- La *entropía diferencial* se comporta diferente en el dominio del tiempo que en el de la frecuencia dadas las mismas señales.
  - El espacio de búsqueda es diferente entre los dominios.
  - Por lo tanto, la trayectoria al minimizarla es diferente.
- Seguramente se quedó atascado en un óptimo local.
  - Por eso pareció terminar más rápido.

# ICA en Frecuencia

## do\_not\_delay = 0

- Siguen estando muy mal separados.
- Compárenlo con ICA en tiempo...

# ICA en Frecuencia

- Es igual, de nuevo.
- Ya van dos, pero todavía tenemos otra técnica a probar...

# NNMF en Frecuencia

- Descarguen:  
`nnmf_fft.m`
- Contiene una implementación de NNMF con FFT:
  - Figura 1: señales de origen.
  - Figura 2: suponiendo que son sólo dos componentes, aquí los muestra.

# NNMF en Frecuencia

## do\_not\_delay = 1

- ???
  - ¿Ni así?
- Para ser justos, estamos violando la suposición principal de NNMF:
  - No valores negativos.
  - Esto no tiene sentido para números complejos.

# NNMF en Frecuencia sólo Magnitud

- NNMF se podría utilizar utilizando sólo la magnitud:
  - Descarguen `nmmf_fft_mag.m`
- Contiene una implementación de NNMF con FFT utilizando sólo magnitud:
  - Figura 1: señales de origen en FFT.
  - Figura 2: suponiendo que son sólo dos componentes en FFT, aquí los muestra.



# NNMF en Frecuencia sólo Magnitud

- Al correrlo con `do_not_delay = 1`:
  - Hay una separación entre los picos.
  - Desgraciadamente, por sólo utilizar la magnitud, hemos hecho al lado la fase, que es esencial para regresar al dominio del tiempo.
- Aún así, es sensible al estado inicial de las variables.
  - Para eso la línea de `“rand('seed',0.00026)”`

# NNMF en Frecuencia sólo Magnitud

- Se podría incorporar el concepto de números complejos, pero otro supuesto fuera de no-negatividad se tendría que utilizar.
- Una variación en el dominio complejo de NNMF se puede encontrar en:

H. Kameoka, Nobutaka Ono, Kunio Kashino and Shigeki Sagayama, "Complex NMF: A new sparse representation for acoustic signals," 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, 2009, pp. 3437-3440.

- Antes de explorar otras alternativas de NNMF, es importante que lo probemos con `do_not_delay = 0` ...

# NNMF en Frecuencia sólo Magnitud

- Al correrlo con `do_not_delay = 0`:
  - Ningún componente es separado.
- Aún en el dominio de la frecuencia, el desfase impacta a la separación.
- Por lo tanto, se puede intuir que la variación compleja de NNMF también va a ser impactada por el desfase.

¿Entonces no nos podemos escapar del desfase?

# Desfase en Frecuencia

- El desfase también modifica a la señal en el dominio de la frecuencia.
  - Pero esto no se ve cuando sólo vemos la magnitud de las frecuencias.
- En donde se modifica es en la fase.

# Desfase en Frecuencia

- Descarguen:  
`test_delay_in_fft.m`
- Contiene una de las señales de origen y una versión desfasada:
  - Figura 1: señales en magnitud de frecuencia.
  - Figura 2: señales en fase de frecuencia.

# Desfase en Frecuencia

- En la magnitud no hay diferencia.
- Pero sí la hay en la fase.
  - Ésta es la razón por la que NNMF funciona bien si sólo utilizamos la magnitud.
- Entonces: no nos podemos escapar de la fase.

# Es importante recordar...

- Ninguna de estas técnicas usa la información de dirección de arribo.
  - De hecho, por su nulo uso de información extra se les conoce como técnicas de **separación ciega** (*Blind Source Separation*).
- ¿No sería posible utilizar esa información de desfase para nuestro beneficio en vez de nuestro perjuicio?



Solución Híbrida:  
Geometric Source Separation  
(GSS)

# Geometric Source Separation

- Busca una matriz de separación  $W$ , pero en su búsqueda, aplica las siguientes dos restricciones:
  - Las fuentes tienen que ser poco correlacionadas.
    - Como en el caso de PCA e ICA.
  - El producto punto entre  $W$  y el direction vector  $A$  (de la estimación Multi-DOA hecho a-priori) debe ser 1.
    - El  $W$  es, a la vez, una matriz de separación y un steering vector de un beamforming múltiple.

# Geometric Source Separation

- Ambas restricciones son aplicadas de una manera “suave”.
  - Se utiliza un algoritmo tipo LMS.
    - Como en GSC.
    - Las restricciones se aplican como funciones de costo.
- La estimación final de  $W$  no necesariamente cumple con estas restricciones.
  - Pero es cercano a hacerlo.

# Geometric Source Separation

## Búsqueda de W

W se actualiza en cada ventana de tiempo por medio de:

$$W_{t+1} = W_t - \mu (\|X\| \nabla J_1 + \nabla J_2)$$

Donde:

$$\nabla J_1 = 4 E W_t R_{xx} \quad \nabla J_2 = 2 (W_t A - I) A$$

$$R_{xx} = X X^T$$

$$E = R_{ss} - \text{diag}(R_{ss})$$

$$R_{ss} = S S^T$$

$$S = W_t X$$

**Este proceso se tiene que llevar a cabo para cada frecuencia f.**

También:

S: es la salida del sistema (GSS)

X: son las señales capturadas

A: es el direction vector que indica en qué direcciones están las señales de origen

$W_t$ : es la matriz de separación utilizada en la ventana actual

$W_{t+1}$ : es la matriz de separación a ser utilizada en la siguiente ventana

$\mu$ : constante de adaptación

$\text{diag}()$ : función que regresa en un vector los valores en la diagonal de una matriz:  $\text{diag}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$

$$\text{Por cierto: } \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

# Desgraciadamente...

- Como pueden ver, su implementación es compleja.
- Y esta es la versión básica del algoritmo.
- Los proyectos ManyEars y HARK utilizan una variación aún mas compleja.
- Por lo tanto:
  - Su implementación va más allá del nivel de complejidad apropiado para este curso.

# Espera...

- ¿Esto significa que los algoritmos de análisis estadístico puros no son útiles para separación de fuentes defasadas en línea?
  - Éste es un tema todavía no completamente resuelto en la literatura de análisis estadístico.
  - Por lo tanto, considerando sólo las versiones presentadas en este curso (las originales):
    - *Estas técnicas no son útiles para nuestros propósitos.*

# ¿Entonces? ¿Proyecto final?

- La intención de las sesiones de *Separación de Fuentes* es proporcionar el campo completo de todas las alternativas que se han considerado.
  - Que las versiones presentadas aquí no sean viables, no significa que no haya alguna versión que sí lo sea.
    - Hay decenas de ellas: sería interesante explicar/explorar cada una, pero no para ámbitos pedagógicos como las de este curso.
- **Entonces:** es bastante claro que, por ahora, las opciones más convenientes para nuestros propósitos son del tipo de Beamforming.

# Excepto...

- Si alguno de ustedes implementa exitosamente en JACK la versión presentada aquí de Geometric Source Separation:
  - En la página del curso está un artículo que lo explica detalladamente aplicado en un robot de servicio.
    - Es la versión utilizada en ManyEars.
- O alguna versión más reciente.
- Definitivamente merecen el 100 en este curso.
  - Reto presentado...



Gracias por atender el curso.

Nos vemos en las presentaciones de proyectos  
finales.