

Técnicas de Análisis Estadístico Multivariado Basadas en Estadísticos de Orden Superior

Tratamiento Avanzado de Señal en Comunicaciones

Curso 2009-2010



Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 Métodos Kernel
- 5 Resumen

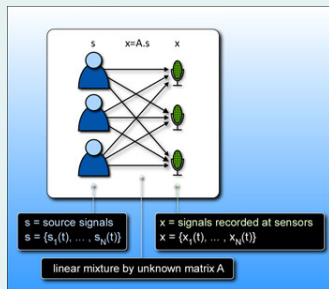
Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 Métodos Kernel
- 5 Resumen

Motivación

● Cocktail Party Problem

- Varias personas hablan de manera simultánea
- Un conjunto de micrófonos recoge mezclas de todas las voces



Separación Ciega de Fuentes

- No conocemos las **fuentes** (s) ni la **matriz de mezcla** (A)
- Nuestro objetivo consiste en extraer las voces de los hablantes

Separación Ciega de Fuentes

- Tipos de Mezclas:
 - Mezclas Instantáneas:**

$$\underbrace{\mathbf{x}}_{n \times 1} = \underbrace{\mathbf{A}}_{n \times p} \underbrace{\mathbf{s}}_{p \times 1}$$

- Mezclas Convolutivas:**

$$\underbrace{\mathbf{x}(\omega)}_{n \times 1} = \underbrace{\mathbf{A}(\omega)}_{n \times p} \underbrace{\mathbf{s}(\omega)}_{p \times 1}$$

$$\underbrace{\mathbf{x}[n]}_{n \times 1} = \sum_{k=0}^{\infty} \underbrace{\mathbf{A}[n-k]}_{n \times p} \underbrace{\mathbf{s}[n-k]}_{p \times 1}$$

- Según el número de sensores:
 - Caso Determinado:** $n \geq p$
 - Caso Indeterminado:** $n < p$

Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 Métodos Kernel
- 5 Resumen

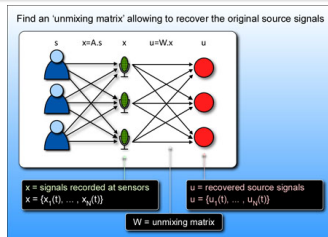
Separación Ciega de Fuentes

- Consideramos **Mezclas Instantáneas con** $n \geq p$

$$\underbrace{\mathbf{x}}_{n \times 1} = \underbrace{\mathbf{A}}_{n \times p} \underbrace{\mathbf{s}}_{p \times 1}$$

- Buscamos una matriz $\mathbf{W} \in \mathbb{R}^{p \times n}$ tal que

$$\mathbf{u} = \mathbf{W}^T \mathbf{x} \simeq \mathbf{s}$$



¿Como encontrar W ?

- En principio, existe una **ambigüedad**:

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \underbrace{\mathbf{A}\mathbf{B}^{-1}}_{\hat{\mathbf{A}}} \underbrace{\mathbf{B}\mathbf{s}}_{\hat{\mathbf{s}}}$$

- ¿Que **información** podemos explotar?
 - Fuentes Independientes** (ICA)
 - Conocimiento (total o parcial) de la **fdp** de las fuentes
 - Fuentes con **Alfabeto Finito** (CMA, ACMA)
 - Fuentes **Dispersas**
 - Estructura Temporal** de las fuentes (Ej: Voz)
 - Suposiciones sobre la **Matriz de Mezcla** $\mathbf{A}$

Análisis de Componentes Independientes

- Dada la Mezcla $\mathbf{x} = \mathbf{A}\mathbf{s}$, se busca la transformación

$$\mathbf{u} = \mathbf{W}^T \mathbf{x}$$

tal que los elementos de $\mathbf{u}$ sean **independientes**

- Si los elementos de $\mathbf{u}$ son independientes,^a habremos conseguido

$$\mathbf{u} = \mathbf{P}\Sigma\mathbf{s}$$

con:

- $\mathbf{P}$: Matriz de **permutación**
- Σ : Matriz diagonal (**escalado** de las fuentes originales)
- Por lo tanto, asumiremos

$$\mathbf{R}_{\mathbf{s},\mathbf{s}} = E \left[\mathbf{s}\mathbf{s}^T \right] = \mathbf{I}$$

^aAdemás se requiere que no más de un elemento de $\mathbf{s}$ sea Gaussiano

¿Como conseguir que los elementos de $\mathbf{u}$ sean independientes?

- Condición Necesaria:

$$\text{Independencia} \begin{matrix} \Rightarrow \\ \nRightarrow \end{matrix} \text{Incorrelación}$$

- En una primera etapa (**preblanqueo**) se forzará la incorrelación de las salidas

$$\mathbf{R}_{\mathbf{u},\mathbf{u}} = E[\mathbf{u}\mathbf{u}^T] = \mathbf{I}$$

proyectando (y reescalando) sobre las p direcciones principales (**PCA**)

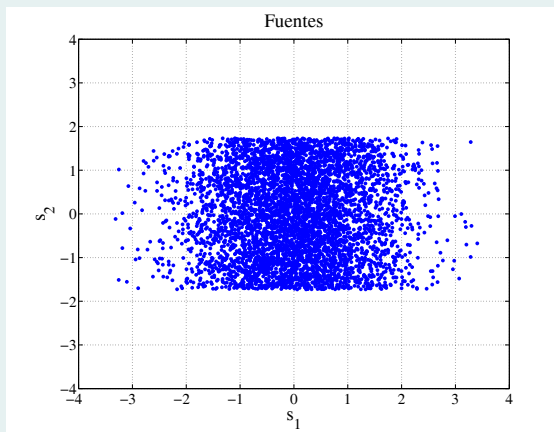
$$\mathbf{y} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{V}^T \mathbf{x} \quad \Rightarrow \quad \mathbf{R}_{\mathbf{y},\mathbf{y}} = \mathbf{I}$$

- $\mathbf{V}$: p autovectores principales de $\mathbf{R}_{\mathbf{x},\mathbf{x}}$
- $\mathbf{\Lambda}$: Matriz diagonal con los p autovalores
- **Ambigüedad residual**: Tras el preblanqueo, el problema se reduce a

$$\mathbf{u} = \mathbf{Q}\mathbf{y} = \underbrace{\mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{V}^T}_{\mathbf{W}^T} \mathbf{x} \quad \text{con} \quad \mathbf{Q}^T \mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$$

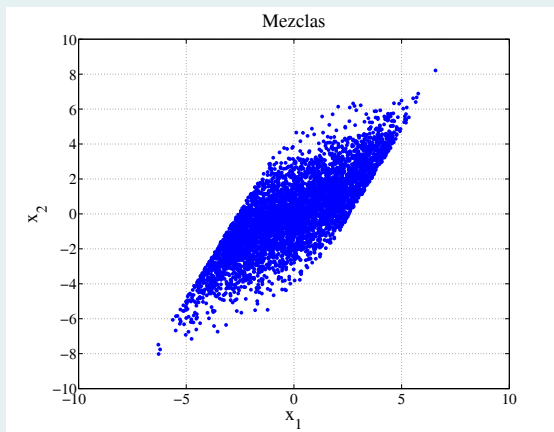
Preblanqueo - Ejemplo con datos bidimensionales

$$\mathbf{x} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{s} \quad s_1 \sim \mathcal{N}(0, 1) \quad s_2 \sim \mathcal{U}(-\sqrt{3}, \sqrt{3})$$



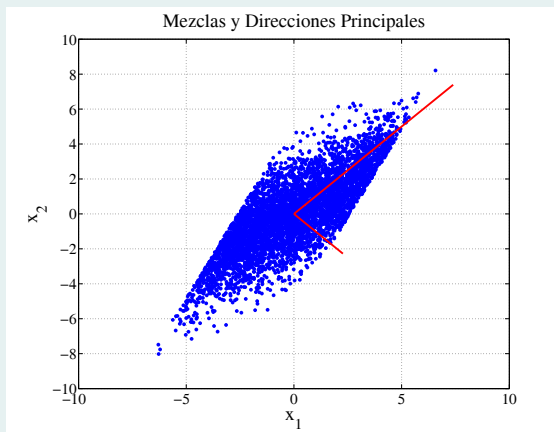
Preblanqueo - Ejemplo con datos bidimensionales

$$\mathbf{x} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{s} \quad s_1 \sim \mathcal{N}(0, 1) \quad s_2 \sim \mathcal{U}(-\sqrt{3}, \sqrt{3})$$



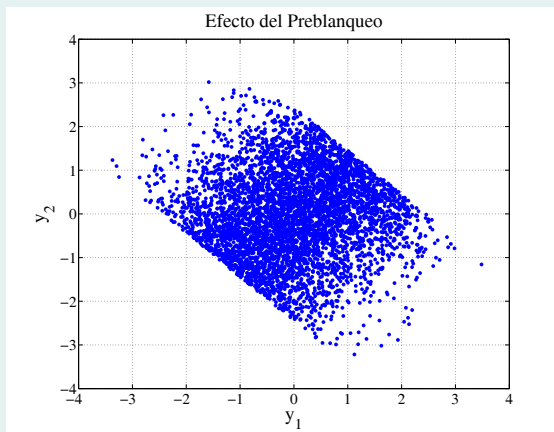
Preblanqueo - Ejemplo con datos bidimensionales

$$\mathbf{x} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{s} \quad s_1 \sim \mathcal{N}(0, 1) \quad s_2 \sim \mathcal{U}(-\sqrt{3}, \sqrt{3})$$



Preblanqueo - Ejemplo con datos bidimensionales

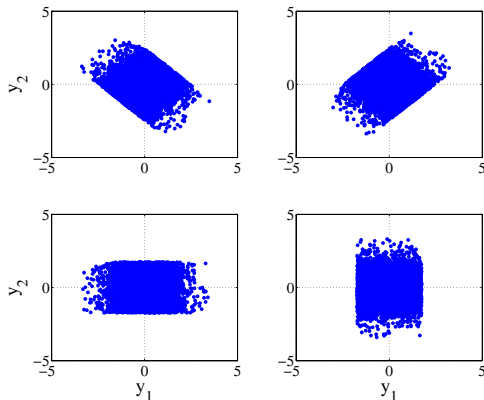
$$\mathbf{x} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{s} \quad s_1 \sim \mathcal{N}(0, 1) \quad s_2 \sim \mathcal{U}(-\sqrt{3}, \sqrt{3})$$



Preblanqueo - Ejemplo con datos bidimensionales

$$\mathbf{x} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{s} \quad s_1 \sim \mathcal{N}(0, 1) \quad s_2 \sim \mathcal{U}(-\sqrt{3}, \sqrt{3})$$

Ambigüedad consistente en una rotación (matriz $\mathbf{Q}$ unitaria)



Análisis de Componentes Independientes

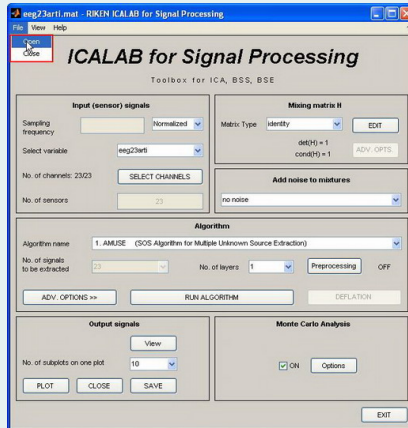
- Tras el preblanqueo, el problema se reduce a

$$\arg \min_{\mathbf{Q}} f(\mathbf{y}[0], \dots, \mathbf{y}[N-1], \mathbf{Q}) \quad \text{sujeto a} \quad \mathbf{Q}^T \mathbf{Q} = \mathbf{I}$$

- $\mathbf{Q}$ es una **matriz unitaria** (recordar que $\mathbf{W} = \mathbf{V} \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{Q}^T$)
- $f(\cdot)$ es una **función de coste (contraste)** que proporciona una medida de la dependencia entre los elementos de $\mathbf{u} = \mathbf{Q}\mathbf{y}$
- $f(\cdot)$ se puede basar en diferentes ideas:
 - Medida de la **Información Mutua** (Divergencia Kullback-Leibler)
 - Medida de la **No-Gaussianidad** (Negentropía)
 - Medida de la **Correlación de Transformaciones No-Lineales**

Análisis de Componentes Independientes

- Toolbox para Matlab: **ICALab^a**



^aA. Cichocki, S. Amari, K. Siwek, T. Tanaka, Anh Huy Phan et al., ICALAB Toolboxes, <http://www.bsp.brain.riken.jp/ICALAB>.

Separación Ciega de Fuentes

- Extensión a **Mezclas Convolutivas con $n \geq p$**
 - Una primera aproximación consiste en trabajar en el dominio de la frecuencia

$$\mathbf{x}(\omega) = \mathbf{A}(\omega)\mathbf{s}(\omega)$$

- Tratando cada frecuencia de manera independiente se obtiene un conjunto de problemas de separación ciega de fuentes de mezclas instantáneas
- **Problema: Permutaciones y Escalados**
 - Si en cada frecuencia se obtiene una permutación y escalado diferente, las señales de salida estarán compuestas por contribuciones de todas las fuentes
 - Para resolver las permutaciones y escalados deberemos recurrir a propiedades adicionales de las fuentes o la matriz de mezclas

Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 Métodos Kernel
- 5 Resumen

Separación Ciega de Fuentes

- ¿Como resolver el problema cuando tenemos **más fuentes que sensores** ($n < p$)?
- Deberemos recurrir a propiedades adicionales de las fuentes o de la matriz de mezclas
- Una de las suposiciones más frecuentes consiste en asumir que las fuentes son **dispersas**
 - Una fuente es dispersa cuando la probabilidad de que esté **activa** es relativamente baja

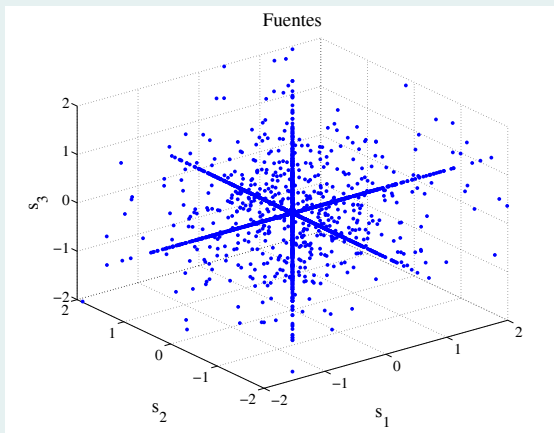
$$P(s \neq 0) \ll 1$$

- Ejemplos:
 - Señales **Audio** en el dominio de la frecuencia
 - Representaciones de **Imágenes** mediante Wavelets

Ejemplo: Problema Indeterminado

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \end{bmatrix} \mathbf{s}$$

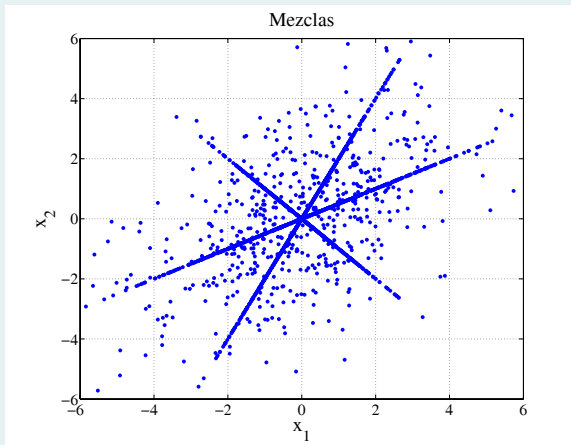
$$P(s_1 \neq 0) = P(s_2 \neq 0) = P(s_3 \neq 0) = 0.2$$



Ejemplo: Problema Indeterminado

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \end{bmatrix} \mathbf{s}$$

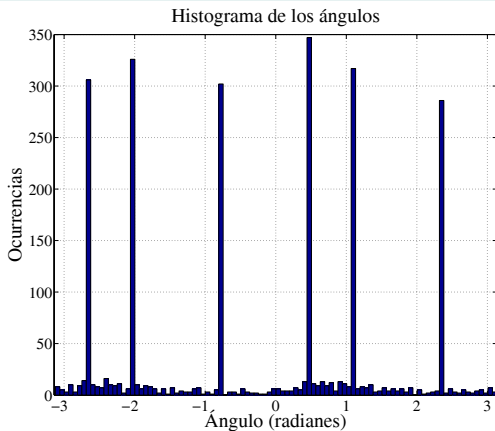
$$P(s_1 \neq 0) = P(s_2 \neq 0) = P(s_3 \neq 0) = 0.2$$



Ejemplo: Problema Indeterminado

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \end{bmatrix} \mathbf{s}$$

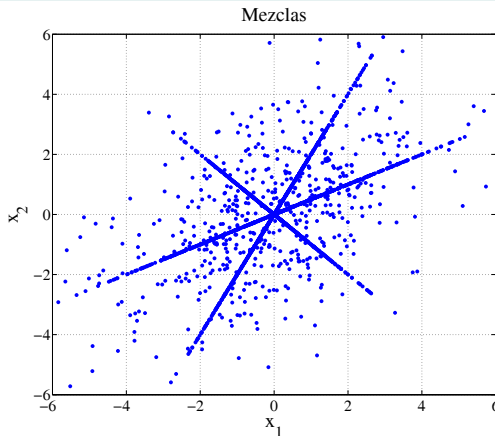
$$P(s_1 \neq 0) = P(s_2 \neq 0) = P(s_3 \neq 0) = 0.2$$



Separación Ciega de Fuentes

- ¿Como recuperar las fuentes una vez conocida la matriz $\mathbf{A}$?

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad \mathbf{A} \in \mathbb{R}^{n \times p} \quad p > n$$



Separación de Mezclas Indeterminadas y Compressive Sensing

- ¿Como recuperar las fuentes una vez conocida la matriz $\mathbf{A}$?

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad \mathbf{A} \in \mathbb{R}^{n \times p} \quad p > n$$

- Un problema relacionado: Compressive Sensing^a**
 - Dados $\mathbf{A} \in \mathbb{R}^{n \times p}$ y $\mathbf{x} \in \mathbb{R}^{n \times 1}$, se trata de recuperar el vector $\mathbf{s}$ sabiendo que es q -disperso
 - Un vector es q -disperso si no tiene más de q componentes no nulas
- La solución al problema se puede obtener de manera sencilla (**programación lineal**) si

$$n = \mathcal{O} \left(q \log \left(\frac{p}{q} \right) \right)$$

- Compressive sensing se puede interpretar como una generalización del **Teorema de Muestreo de Nyquist**

^a<http://www.dsp.ece.rice.edu/cs/>

Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 **Métodos Kernel**
- 5 Resumen

Métodos Kernel

- Los **Métodos Kernel** proporcionan una **extensión no-lineal** elegante de un gran número de **técnicas lineales**
 - Supongamos que transformamos nuestro datos $\mathbf{x} \in \mathbb{R}^{n \times 1}$ a un espacio de dimensión mayor (incluso infinita)

$$\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{n} \times 1} \quad \tilde{\mathbf{x}} = \Phi(\mathbf{x})$$

donde $\Phi(\mathbf{x})$ representa un mapeo no-lineal, por ejemplo

$$\Phi(\mathbf{x}) = \underbrace{[x_1, x_2, \dots, x_n, x_1^2, x_2^2, \dots, x_n^2, x_1^3, x_2^3, \dots, x_n^3]^T}_{3n \times 1}$$

- Una vez que tenemos los datos en el **espacio de características**, podremos aplicar cualquier técnica lineal a los datos transformados

Kernel Trick

- **Problema:** El mapeo $\Phi(\mathbf{x})$ puede resultar computacionalmente muy costoso, sobre todo si $\tilde{n} \gg n$
- **Solución:** Los productos escalares en el **espacio de características** se pueden calcular mediante una función (**Kernel**) en el **espacio de entrada**

$$\tilde{\mathbf{x}}^T \tilde{\mathbf{y}} = \kappa(\mathbf{x}, \mathbf{y})$$

- De esta manera, no necesitamos calcular $\tilde{\mathbf{x}}$ explícitamente
- El mapeo $\Phi(\mathbf{x})$ quedará definido por la función Kernel seleccionada
- Algunos ejemplos de funciones Kernel son
 - **Kernel lineal:** $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$ (técnicas lineales originales)
 - **Kernel polinómico:** $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^p$
 - **Kernel Gaussiano:** $\kappa(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$

Representer Theorem

- El **representer theorem** es uno de los resultados básicos relacionados con métodos kernel
- Asegura que las transformaciones lineales ($\tilde{\mathbf{w}}$) óptimas en el **espacio de características** se pueden representar como

$$\tilde{\mathbf{w}} = \sum_{n=1}^N \alpha_n \tilde{\mathbf{x}}[n],$$

es decir, $\tilde{\mathbf{w}}$ es una combinación lineal de los datos transformados

- Teniendo esto en cuenta, las proyecciones de los datos son

$$u[m] = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}[m] = \sum_{n=1}^N \alpha_n \tilde{\mathbf{x}}^T[n] \tilde{\mathbf{x}}[m] = \sum_{n=1}^N \alpha_n \kappa(\mathbf{x}[n], \mathbf{x}[m])$$

Ejemplo

- **Kernel-PCA (KPCA)**
- Proyección de los datos ($\kappa_{m,n} = \kappa(\mathbf{x}[m], \mathbf{x}[n])$)

$$\underbrace{\begin{bmatrix} u[1] \\ \vdots \\ u[N] \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} \kappa_{1,1} & \cdots & \kappa_{1,N} \\ \vdots & \ddots & \vdots \\ \kappa_{N,1} & \cdots & \kappa_{N,N} \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}}_{\boldsymbol{\alpha}}$$

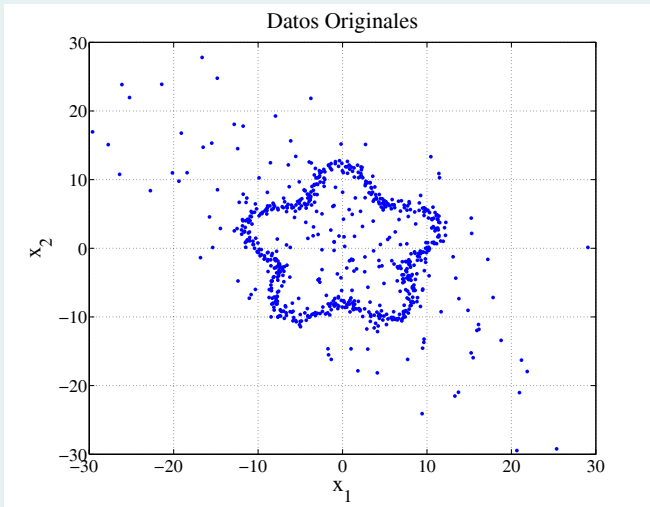
- Energía de la proyección:

$$\mathbf{u}^T \mathbf{u} = \boldsymbol{\alpha}^T \mathbf{K}^2 \boldsymbol{\alpha}$$

- Proyección ($\boldsymbol{\alpha}$) de máxima energía: Autovector principal de $\mathbf{K}$

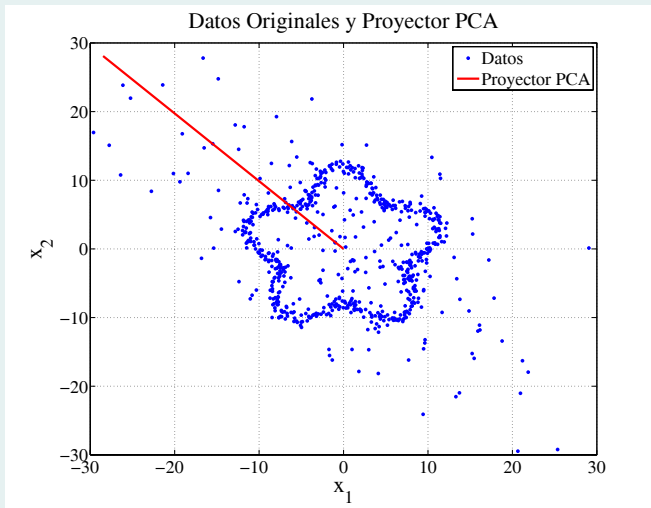
Ejemplo KPCA

- Datos Bidimensionales



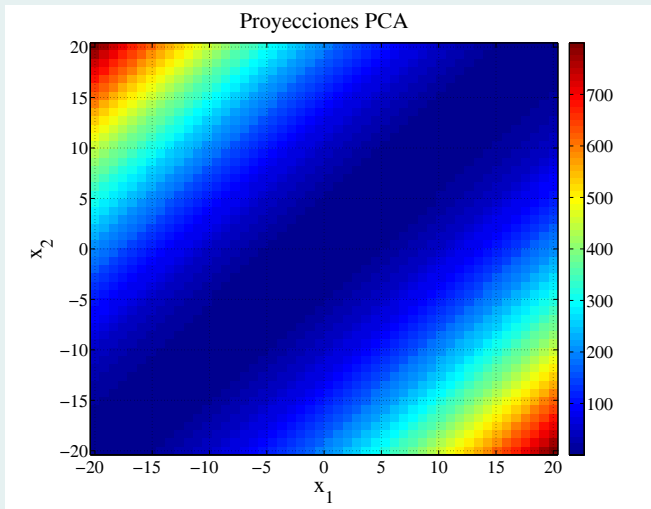
Ejemplo KPCA

- Datos Bidimensionales y PCA Lineal



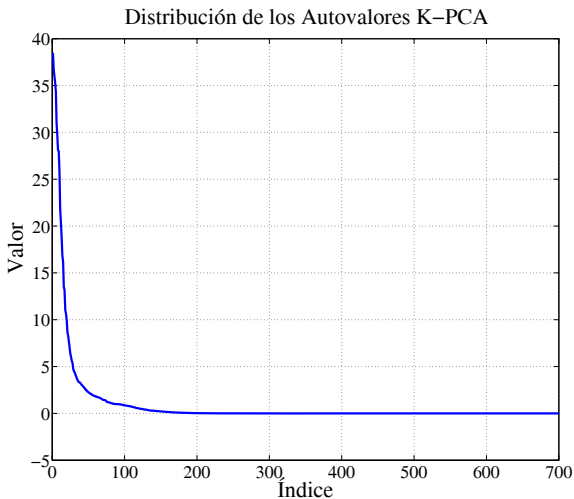
Ejemplo KPCA

- Datos Bidimensionales y PCA Lineal



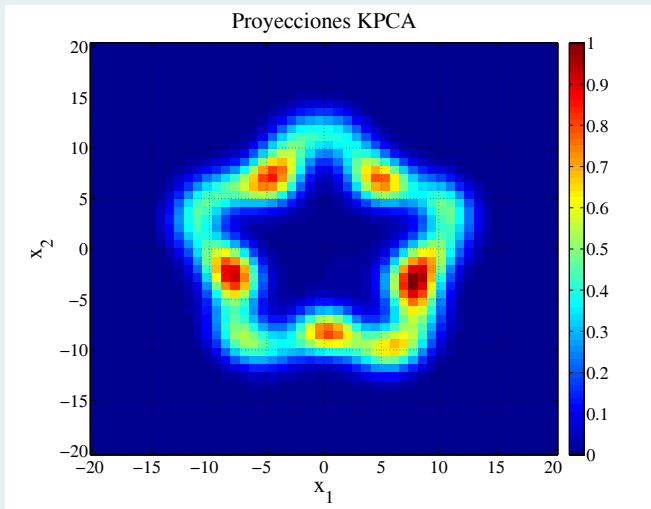
Ejemplo KPCA

- Kernel-PCA con Kernel Gaussiano ($\sigma = 2$)



Ejemplo KPCA

- Kernel Gaussiano ($\sigma = 2$) y $p = 50$ Autovectores Principales



Contenido

- 1 Introducción
- 2 Separación Ciega de Fuentes - Caso Determinado
 - Análisis de Componentes Independientes (ICA)
- 3 Separación Ciega de Fuentes - Caso Indeterminado
 - Compressive Sensing
- 4 Métodos Kernel
- 5 **Resumen**

Resumen

- Un gran número de problemas requiere más información que la proporcionada por los estadísticos de segundo orden (SOS)
- **Ejemplo:** Separación Ciega de Fuentes
 - Caso Determinado:
 - **Idea:** Tendremos en cuenta que las fuentes son independientes
 - **Análisis de Componentes Independientes (ICA)**
 - Caso Indeterminado:
 - **Idea:** Tendremos en cuenta que las fuentes son dispersas
 - Una vez estimada la matriz de mezclas aun hay que recuperar las fuentes
 - Problema análogo a **Compressed Sensing**
- **Más Ejemplos:** Estimación, Clustering, Clasificación, Detección, ...
- Extensiones No-Lineales de Técnicas Lineales: **Métodos Kernel**
 - **Ejemplos:** **KPCA**, **KPLS**, **KMLR**, **KCCA**
 - Aplicación en ICA, Clustering, Clasificación, ...