Autovalores/Autovectores
Descomposición en Valores Singulares

Métodos Numéricos

Departamento de Computación
Facultad de Ciencias Exactas y Naturales

26 de septiembre de 2014

Repaso

- Álgebra Lineal
  - Matrices. $A \in \mathbb{R}^{m \times n}$
  - Transformaciones Lineales. $T_A : \mathbb{R}^n \to \mathbb{R}^m, \quad T_A(x) = Ax$
  - Núcleo. $Nu(A) = \{x \in \mathbb{R}^n : Ax = 0\} \subseteq \mathbb{R}^n$
  - Imagen. $Im(A) = \{Ax : x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m = \langle col_1(A), \ldots, col_n(A) \rangle$
  - Rango. $rg(A) =$ cantidad de columnas linealmente independientes.
- Factorización de matrices:
  - $PA = LU$... Eliminación Gausseana. (TP1)
  - $A = LL^T$... Cholesky.
  - $A = QR$... Givens, Householder.
- Autovalores y Autovectores
  - $Av = \lambda v$
  - Polinomio Característico. $P_A(\lambda) = det(A - \lambda I)$
  - Método de la Potencia... TP2!!!

Menú del día

1. Matrices Ortogonales en $\mathbb{R}^n$.
2. Problema de la vida real: Ocultar Información.
3. Descomposición en Valores Singulares.
4. Ejercicio!!

Matrices Ortogonales

- El lunes vieron matrices ortogonales:
  - $Q \in \mathbb{R}^{n \times n}$
  - $Q^T Q = Q Q^T = I$
  - $\|Qx\|_2 = \|x\|_2$
- Vieron casos particulares:
  - Rotaciones: $G = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$, $G^T = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$
  - Reflexiones: $H = I - 2uu^t$ con $u \in \mathbb{R}^n$, $\|u\|_2 = 1$.

- ¿Se podrán extender las rotaciones a $\mathbb{R}^n$?

Matrices Ortogonales en $\mathbb{R}^n$

- Una matriz de rotación en $\mathbb{R}^n$ se puede pensar como una concatenación de rotaciones alrededor de cada uno de los ejes. Por ejemplo:

$$R = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & \sin(\beta) \\ 0 & -\sin(\beta) & \cos(\beta) \end{pmatrix}}_{\text{Rotación de } x} \underbrace{\begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}}_{\text{Rotación de } y} \underbrace{\begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Rotación de } z}$$

- n ángulos de rotación (uno por cada eje).
- Esta no es la única manera de pensarlo...
- Pero... ¿Cuándo una matriz ortogonal es de rotación?
- **Teorema:** Una matriz ortogonal $Q \in \mathbb{R}^{n \times n}$ se puede descomponer como el producto de matrices de rotación y reflexión.
  $Q = G_1 G_2 G_3 H_1 H_2 G_4 H_3 \ldots$
- Es decir que cualquier matriz ortogonal la podemos pensar como una rotación (salvo reflexiones intermedias).

Problema de los Prisioneros (G. Simmons, 1983)

En una cárcel, dos prisioneros quieren planear un escape.

Están en celdas distantes y la única manera de comunicarse es mediante un guardia que lleva el mensaje de una celda a la otra.

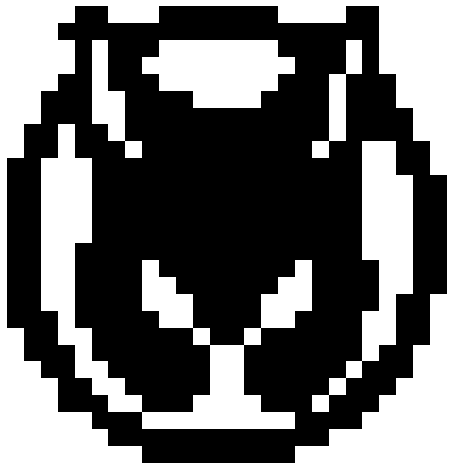
¿Cómo pueden hacerlo?

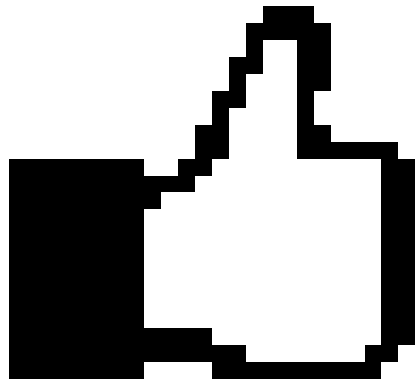
Esteganografía

- "Es el arte o práctica de ocultar un mensaje u objeto, dentro de otro mensaje u objeto."
- No es criptografía: No estamos *protegiendo* un mensaje, sino que estamos *ocultando* un mensaje.
  - En Esteganografía: Gorgory puede mirar el canal de comunicación y no darse cuenta de que se está transmitiendo un mensaje.
  - En Criptografía: Gorgory *SABE* que un mensaje está siendo transmitido pero no entiende lo que dice.
- La esteganografía tiene como objetivo ocultar la *existencia* de un mensaje.

Esteganografía en Imágenes

- Consiste en ocultar una marca de agua en una imagen portadora.
- Imagen Portadora: Es una imagen RGB o escala de grises. También se pueden usar videos (secuencia de imágenes).
- Marcas de Agua:
  - Son imágenes binarias (blanco y negro).
  - Cada píxel es un bit (1 para el blanco, 0 para el negro).
  - Podemos usarlas para representar logos o firmas.
  - Las vamos a representar con un vector $WM \in \{0,1\}^K$.





Esteganografía en Imágenes

- ¿Para qué sirve? Por ejemplo para evitar que un usuario cargue a un sitio web una imagen o un video con derechos de Copyright.

- ¿Cómo ocultar bits en una imágen?
  1. Introducirlos sin que se vea la alteración en la imágen.
  2. Extraerlos... La máquina tiene que poder "verlos".

¿Cómo ocultar bits en una imágen?

Tenemos muchos bits (uno por cada píxel de la marca de agua) y tenemos que esconderlos en la imagen portadora. ¿Qué se les ocurre?

- Esconder la marca de agua en algún bloque de la portadora.
  - Se puede observar a simple vista.
  - Hay que elegir un bloque apropiado, que no se pueda remover sin dañar la imágen.
- Codificar cada bit en un pixel elegido al azar.
  - Se puede observar a simple vista.
  - Para extraerlos necesitamos generar la misma secuencia de pixels aleatorios.
- Codificar cada bit en un bloque de pixels, elegido al azar.
  - Para extraerlos necesitamos generar la misma secuencia de bloques aleatorios.
  - ¿Cómo codificamos un bit en un bloque de pixels?

Bloque de pixels

- Formalmente, un bloque de $n \times n$ pixels es una matriz cuadrada $B \in \mathbb{Z}^{n \times n}$, tal que $0 \leq a_{ij} \leq 255$.
- Por simplicidad vamos a asumir que un bloque es una matriz $B \in \mathbb{R}^{n \times n}$.
- Notación:
  - B el bloque original.
  - $\overline{B}$ el bloque con el bit codificado.

- Tenemos que modificar la matriz B de tal manera que:
  1. Los valores de b_{ij} se modifiquen lo menos posible ($b_{ij} \approx \overline{b}_{ij}$).
  2. La representación en escala de grises de B debe ser similar a la de $\overline{B}$.
  3. Dada $\overline{B}$ debemos poder extraer el mismo bit que habíamos codificado.

Codificar un bit en B

- Ya vimos que modificar la matriz a secas no es una buena idea.
- **Propiedad:** El subespacio generado por las columnas de B es igual a la imagen de B.

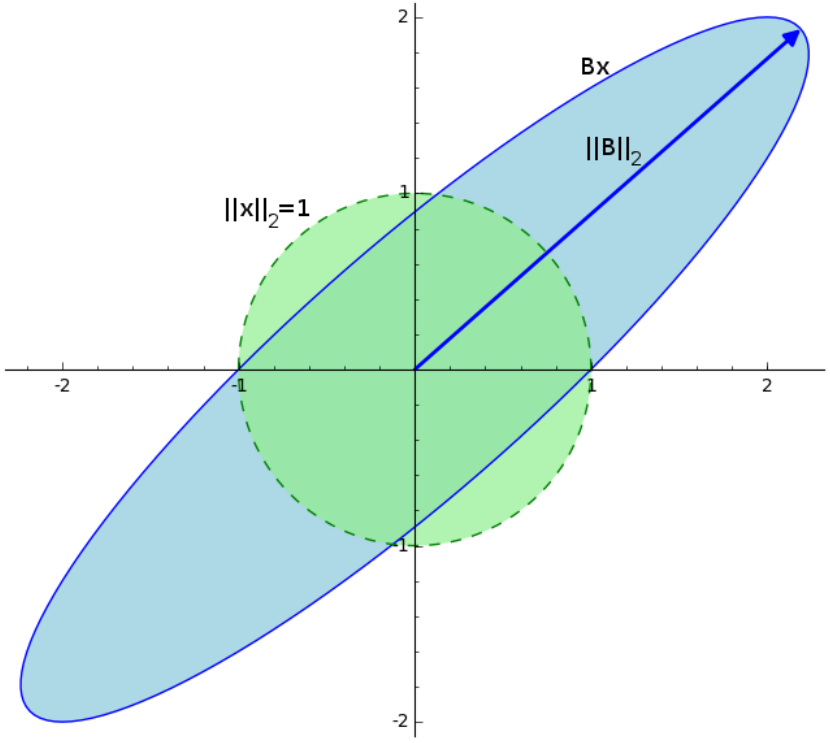
$$Im(B) = \langle col_1(B), \ldots, col_n(B) \rangle$$

- **Idea:** Modificar $Im(B)$. Codificar un bit en las características de $Im(B)$ de tal manera que las nuevas columnas sean "parecidas" a las originales.
  - Invisible al ojo humano: Si las columnas no cambian demasiado esperamos que la representación de $\overline{B}$ en escala de grises sea similar a la de B.
  - Recuperable: Analizando las características de $Im(\overline{B})$ tendríamos que poder determinar cuál es el bit.
- ¿Cuáles son esas "características" de $Im(B)$?
- El subespacio $Im(B)$ está caracterizado por la manera en que B **transforma** a $\mathbb{R}^n$.
- ¿De qué manera transforma B a $\mathbb{R}^n$?

¿De qué manera transforma B a $\mathbb{R}^n$?

Recordemos...

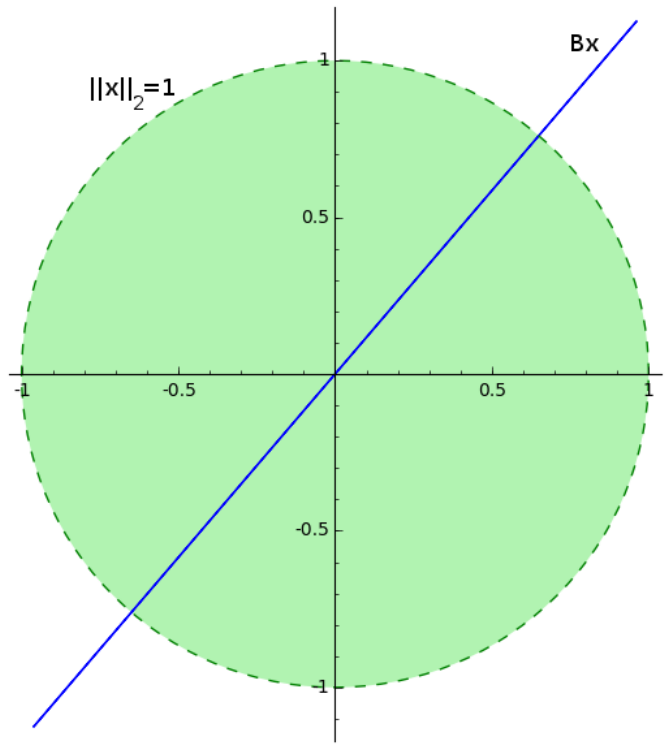
- $\|B\|_2 = \max_{\|x\|_2=1} \|Bx\|_2$
  - Es la mayor deformación a la bola de radio 1.
  - Nos da una idea del máximo escalado que la matriz B produce en $\mathbb{R}^n$.



  - No sabemos en qué dirección se produce este escalado.
  - Si n es muy grande, nos da muy poca información de la deformación que produce B en $\mathbb{R}^n$.

¿De qué manera transforma B a $\mathbb{R}^n$?

- $\kappa_2(B) = \|B\|_2\|B^{-1}\|_2$
  - Nos dice que tan "singular" es una matriz desde el punto de vista numérico.
  - Cuanto mayor sea este número, la deformación de la bola de radio 1 parece que perdiera dimensiones. Por ejemplo, si $\kappa_2(B) >> 1$:



  - Nos da una idea global de la deformación, pero no es precisa.

Necesitamos generalizar...

Descomposición en Valores Singulares (a.k.a. SVD)

Teorema: Sea $A \in \mathbb{R}^{m \times n}$ de rango r, existen $U \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times n}$ y $V \in \mathbb{R}^{n \times n}$, tales que U y V son ortogonales, $\Sigma = diag\{\sigma_1, \ldots, \sigma_r\}$ $(\sigma_1 \geq \cdots \geq \sigma_r > 0)$ y $A = U\Sigma V^T$.

Definición: σ_i son los valores singulares de A, las columnas de U $\{u_1, \ldots, u_m\}$ son los vectores singulares a izquierda de A y las columnas de V $\{v_1, \ldots, v_n\}$ son los vectores singulares a derecha de A.

Propiedades:

- $\sigma_1^2, \ldots, \sigma_r^2$ son los autovalores no nulos de $A^T A$ y de AA^T.
- $\{v_1, \ldots, v_n\}$ son los autovectores de $A^T A$.
- $\{u_1, \ldots, u_m\}$ son los autovectores de AA^T.

- $\|A\|_2 = \sigma_1$
- $\kappa_2(A) = \sigma_1/\sigma_n$

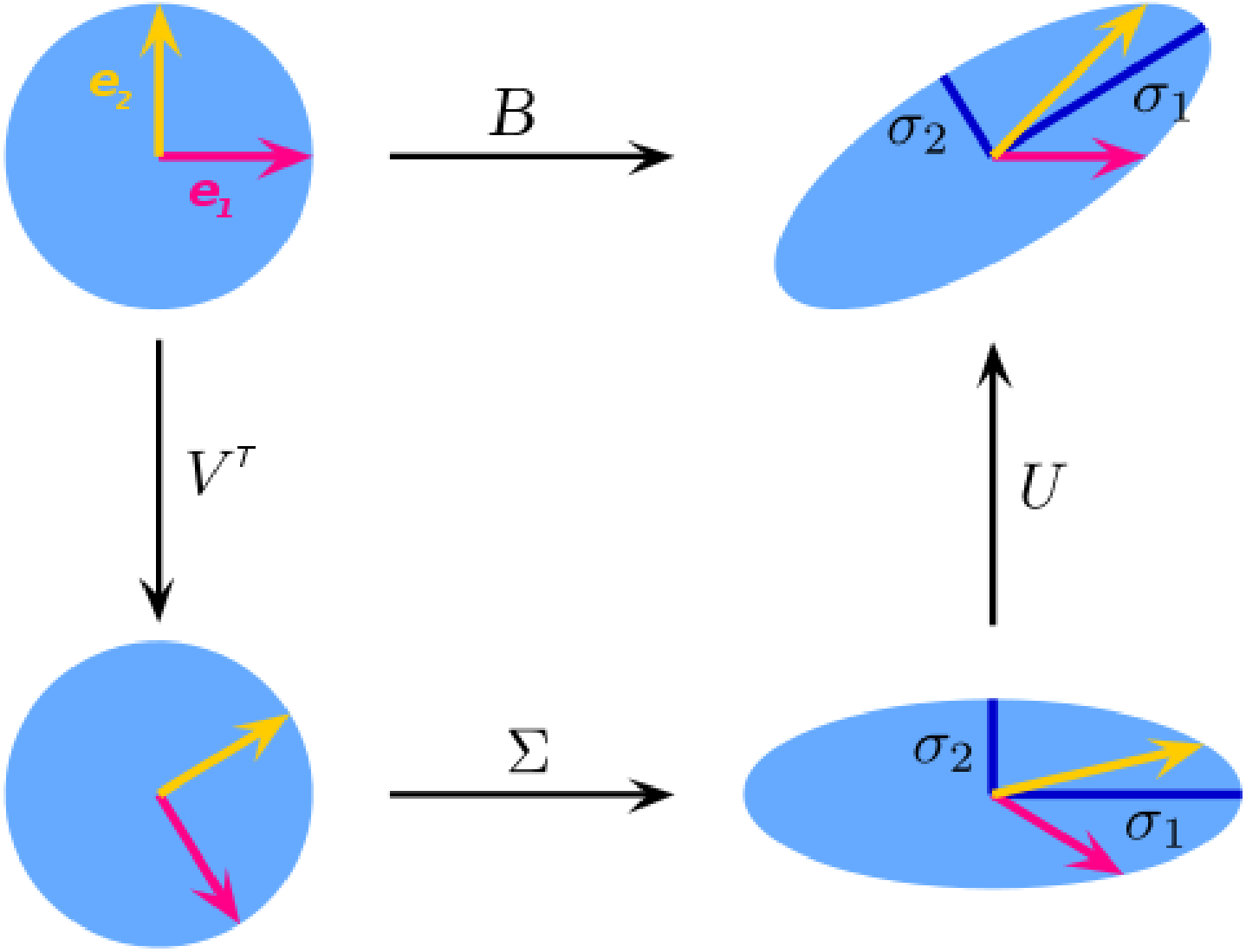
Interpretación geométrica de SVD

- Recordemos que una matriz ortogonal es el producto de rotaciones y reflexiones en el espacio.
- Cualquier matriz se puede escribir como $A = U\Sigma V^T$.
- Si pensamos en una matriz cuadrada $B \in \mathbb{R}^{n\times n}$,

$$Bx = U\Sigma V^T x = \underbrace{U \underbrace{(\Sigma \underbrace{(V^T x)}_{\text{Rotación}})}_{\text{Escalado}}}_{\text{Rotación}}$$

- Es decir que B únicamente realiza rotaciones/reflexiones y escala el espacio.
- Si modificamos alguna de estas matrices, podemos controlar cómo B transforma a $\mathbb{R}^n$.
  - Podemos controlar $Im(B)$...

Interpretación geométrica de SVD



$$B = U \cdot \Sigma \cdot V^T$$

Impacto de SVD en los pixels

- Modificando U, Σ y V podemos cambiar la manera en que B escala o rota el espacio.
- ¿Cómo afectan estas modificaciones a las columnas de B?

$$B = U\Sigma V^T = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t + \cdots + \sigma_n u_n v_n^t$$

- Recordemos que $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.
- Luego el término $\sigma_1 u_1 v_1^t$ es el que más contribuye al "aspecto visual" de B.
- Por ejemplo:

$$B = \begin{pmatrix} 1{,}01 & 0{,}99 \\ 0{,}99 & 1{,}01 \end{pmatrix} = \underbrace{\begin{pmatrix} -0{,}7071 & -0{,}7071 \\ -0{,}7071 & 0{,}7071 \end{pmatrix}}_{\begin{pmatrix} u_1 & u_2 \end{pmatrix}} \begin{pmatrix} 2 & 0 \\ 0 & 0{,}02 \end{pmatrix} \underbrace{\begin{pmatrix} -0{,}7071 & -0{,}7071 \\ -0{,}7071 & 0{,}7071 \end{pmatrix}}_{\begin{pmatrix} v_1^t \\ v_2^t \end{pmatrix}}$$

$$= \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}}_{\sigma_1 u_1 v_1^t} + \underbrace{\begin{pmatrix} 0{,}01 & -0{,}01 \\ -0{,}01 & 0{,}01 \end{pmatrix}}_{\sigma_2 u_2 v_2^t}$$

Recapitulando...

1. Queremos ocultar una secuencia de bits en una imagen.
2. Elegimos aleatoriamente bloques $B_i \in \mathbb{R}^{n \times n}$ para codificar cada bit.
   - ¿De qué tamaño?
   - n es un parámetro para ajustar con experimentos.
   - Por simplicidad asumamos que ya realizamos la experimentación y tomamos $n = 4$.
3. Por cada B_i calculamos su descomposición en valores singulares y codificamos el bit.
4. Reconstruimos $\overline{B_i}$ usando la SVD modificada.

Tengamos en cuenta

$$B = U\Sigma V^T = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t + \sigma_3 u_3 v_3^t + \sigma_4 u_4 v_4^t$$

- Los valores singulares más grandes son los que más afectan el aspecto de B.
- Modificar "poco" los valores singulares más pequeños deberían alterar "poco" el aspecto de B.
- "poco": Los valores singulares más pequeños no deberían pasar a ser los más grandes.
- Modificar valores singulares de B cambia la manera en que B escala $\mathbb{R}^n$.
- **Idea:** Codificar un bit en B, modificando sus valores singulares.

Usar valores singulares para codificar/decodificar

Sea $B \in \mathbb{R}^{4\times 4}$.
Supongamos que queremos codificar $bit \in \{0,1\}$ en $B = U\Sigma V^T$:

- No conviene modificar σ_1 porque es el que más afecta a B.
- No conviene modificar σ_4 porque en general va a ser cercano a cero y susceptible a error.
- Entonces modificamos σ_2 y σ_3.
- Elegimos una constante $T > 0$ y armamos $\overline{\Sigma} = diag\{\overline{\sigma}_1, ..., \overline{\sigma}_4\}$:

$$\overbrace{\overline{\Sigma} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 + T & & \\ & & \sigma_2 & \\ & & & \sigma_4 \end{pmatrix}}^{\text{Si } bit==1} \qquad \overbrace{\overline{\Sigma} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_2 & \\ & & & \sigma_4 \end{pmatrix}}^{\text{Si } bit==0}$$

- De esta manera, $bit == 1 \quad \Leftrightarrow \quad \overline{\sigma}_2 - \overline{\sigma}_3 > T/2$.
- Sutileza: Si $\sigma_1 < \sigma_2 + T$, tomamos $\overline{\sigma}_1 = \sigma_2 + T$.
  Si T es pequeño, esto no afecta la calidad (dado que $\sigma_1 \geq \sigma_2$).

Los algoritmos

Codificación

Dada una imágen $A \in \mathbb{R}^{m \times n}$, una marca de agua $WM \in \{0,1\}^K$ y una semilla s:

1. Generar una secuencia aleatoria (con semilla s) de K bloques $B_i \in \mathbb{R}^{4 \times 4}$ de A.
2. Codificar los bits de WM en los bloques B_i de la secuencia.
   - 2.1 Hallar SVD de $B_i = U_i \Sigma_i V_i^T$.
   - 2.2 Codificar el bit en $\overline{\Sigma_i}$.
   - 2.3 Construir $\overline{B_i} = U_i \overline{\Sigma_i} V_i^T$.
   - 2.4 Reemplazar B_i por $\overline{B_i}$ en A.
3. Devolver A.

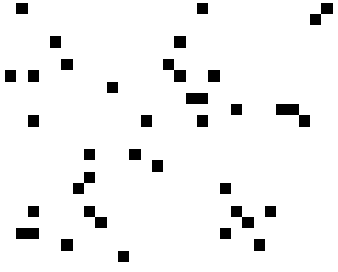
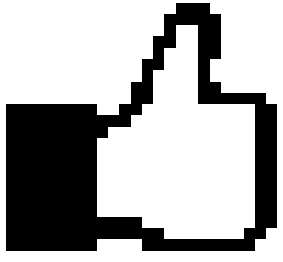
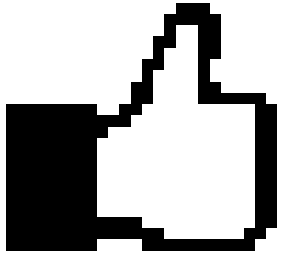
Los algoritmos

Decodificación

Dada la imágen alterada $A \in \mathbb{R}^{m \times n}$, el tamaño de la marca de agua K y la *misma* semilla s:

1. Generar una secuencia aleatoria (con semilla s) de K bloques $B_i \in \mathbb{R}^{4 \times 4}$ de A.
2. Armar WM decodificando de cada bloque B_i un bit.
   2.1 Hallar SVD de $\overline{B_i} = U_i \overline{\Sigma_i} V_i^T$.
   2.2 Decidir si el bit es 0 ó 1 y guardarlo en WM.
3. Devolver WM.

Algunos resultados variando T

$T = 10^{-3}$	$T = 4 \times 10^{-3}$	$T = 5 \times 10^{-2}$	$T = 1{,}7$
			
			
$Error = 26\,\%$	$Error = 16{,}93\,\%$	$Error = 0\,\%$	$Error = 0\,\%$

¿Cómo hallar SVD de B?

- $B \in \mathbb{R}^{4\times 4}$.
- Buscamos U, V ortogonales y Σ diagonal tales que $B = U\Sigma V^T$.
- Propiedades:
  - $\sigma_1^2, \ldots, \sigma_4^2$ son los autovalores (nulos y no nulos) de $B^T B$.
  - $\{v_1, \ldots, v_4\}$ son los autovectores de $B^T B$.
- Si buscamos los 4 autovalores (λ_i) y autovectores (v_i) de $B^T B$:
  - Tomamos $\sigma_i = \sqrt{\lambda_i}$. ¿Siempre está bien definido en $\mathbb{R}$?
  - $\Sigma = diag\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$
  - $B^T B$ es simétrica $\Rightarrow$ $\exists$ b.o.n. de autovectores.
    Supongamos que obtuvimos dicha base, entonces $V = (v_1, \ldots, v_n)$.
- ¿Cómo armamos U? ¿Se pueden usar los autovectores de BB^T?
- Despejamos:

$$B = U\Sigma V^T \quad \Leftrightarrow \quad BV = U\Sigma \quad \Leftrightarrow \quad Bv_i = \sigma_i u_i \ \forall i$$

$$\Rightarrow u_i = \begin{cases} Bv_i/\sigma_i & \text{si } \sigma_i \neq 0 \\ ?? & \text{si no} \end{cases}$$

¿Cómo hallar SVD de B?

- Supongamos que $rg(B) = r < 4$.
- Por definición, $\sigma_1 \geq ... \geq \sigma_r > 0$.
- Según el cálculo anterior, podemos obtener $\{u_1, ..., u_r\}$.
- ¿Cómo obtenemos $\{u_{r+1}, ..., u_4\}$?
- Necesitamos que $\{u_1, ..., u_4\}$ sea una base ortonormal de $\mathbb{R}^4$.
- Primero completemos la base y luego ortonormalizamos.

Completando una base

- Propiedad: Sea $\{u_1, ..., u_r\} \subseteq \mathbb{R}^n$ $(r < n)$ un conjunto de vectores linealmente independientes. Sea $\{e_1, ..., e_n\}$ la base canónica de $\mathbb{R}^n$. Existe $1 \leq i \leq n$ tal que $\{u_1, ..., u_r, e_i\}$ es l.i..
  - Demo:... ejercicio![1]
- Si iterativamente agregamos canónicos al conjunto $\{u_1, ..., u_r\}$, eventualmente obtendremos una base de $\mathbb{R}^n$.
- Tendremos n vectores l.i. de la forma $\{u_1, ..., u_r, e_{i_{r+1}}, ..., e_{i_n}\}$.
- Los primeros r vectores ya son ortogonales entre sí y están normalizados (demostración en la teórica).
- Tenemos que ortogonalizar los demás. ¿Siempre es posible?

[1] Piensen qué pasaría si todos los canónicos fuesen una combinación lineal de u_i's.

Proceso de ortogonalización de Gram-Schmidt

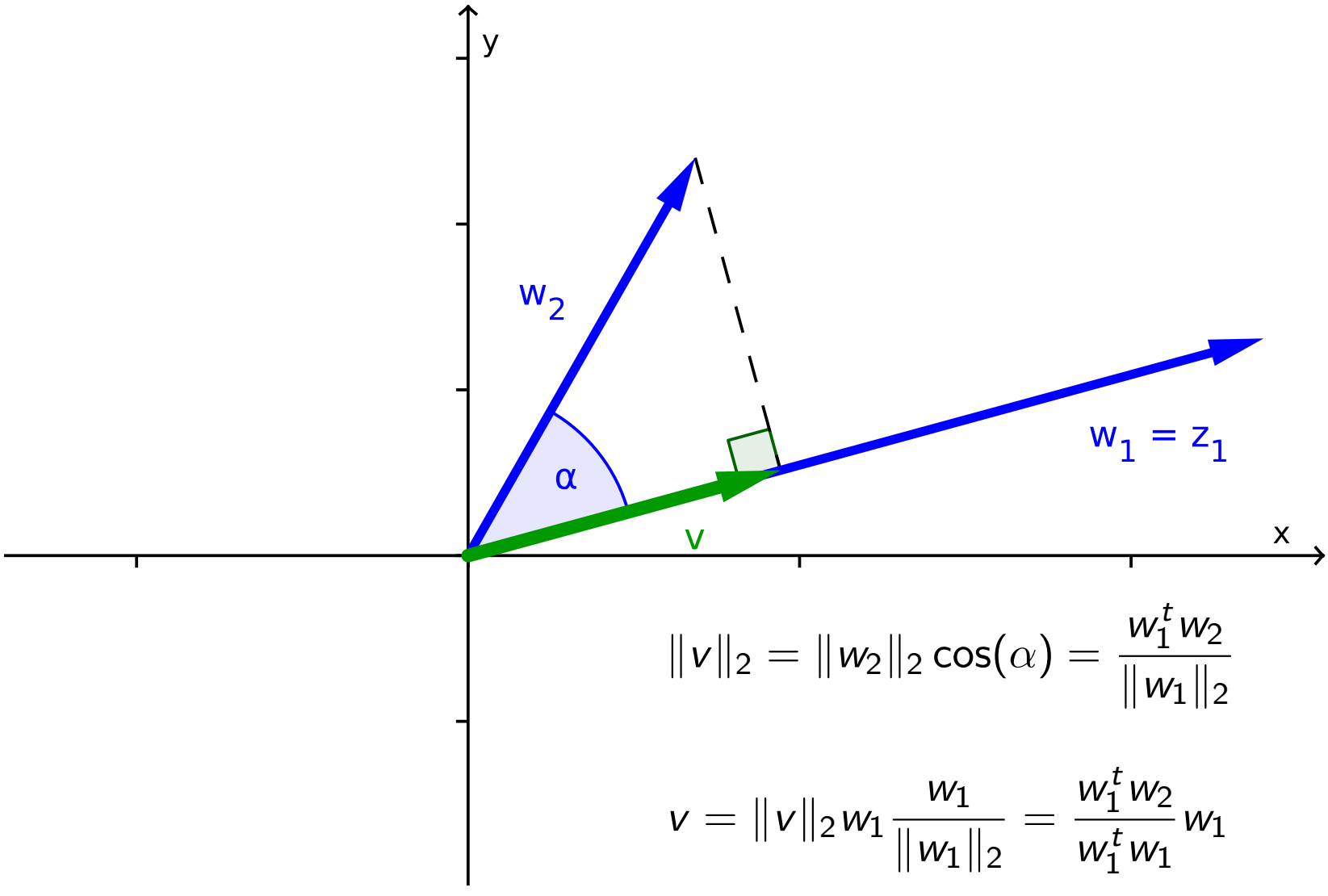
- Dada una base de un subespacio, construye una base ortonormal del mismo subespacio.
- Veamos un ejemplo en $\mathbb{R}^2$.
- Sea $\{w_1, ..., w_n\}$ una base.

$$z_1 = w_1$$
$$z_i = w_i - \sum_{j=1}^{i-1} \frac{z_j^t w_i}{z_j^t z_j} z_j \qquad \forall i \neq 1$$

Finalmente, tomar $u_i = \dfrac{z_i}{\|z_i\|_2} \quad \forall i$.

- El vector i lo calcula usando los vectores $1...(i-1)$.
- Dado que nosotros ya tenemos $\{u_1, ..., u_r\}$ ortonormalizados, arrancamos el proceso en el vector $r+1$.

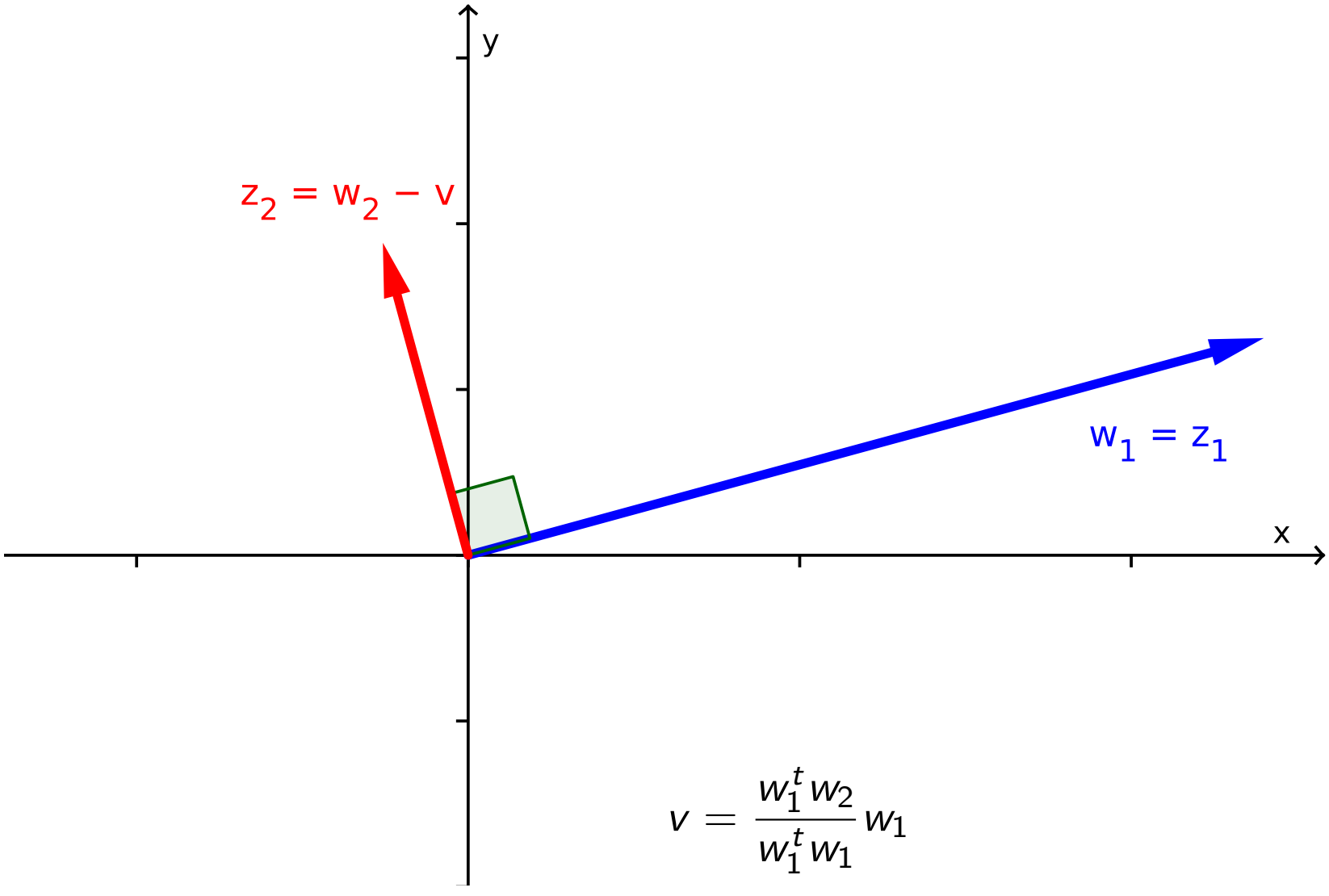
Proceso de ortogonalización de Gram-Schmidt



$$\|v\|_2 = \|w_2\|_2 \cos(\alpha) = \frac{w_1^t w_2}{\|w_1\|_2}$$

$$v = \|v\|_2 w_1 \frac{w_1}{\|w_1\|_2} = \frac{w_1^t w_2}{w_1^t w_1} w_1$$

Proceso de ortogonalización de Gram-Schmidt



$$v = \frac{w_1^t w_2}{w_1^t w_1} w_1$$

Ejercicio

En el archivo `hallarSVD.m`, completar:

1. El armado de V y $\Sigma(S)$ de la descomposición en valores singulares de un bloque B.
2. Calcular las primeras columnas de U que se desprenden de los valores singulares no nulos.
3. Calcular el rango de la matriz B.

Algunos tips:

- Respetar los nombres provistos para las variables (`V`, `S`, `U`, `rango`).
- `U` debe tener dimensión 4×4. Completar las columnas usando `U(:,i)=...`

Bibliografía

- *An SVD Oriented Watermark Embedding Scheme With High Qualities For The Restored Images*. Chang, Ling, Hu. ICIC International, 2007, ISSN 1349-4198, p. 609-620.
- *Linear Algebra and its applications*. David C. Lay. Secciones 6.2, 6.3, 6.4 y 7.4.
- *Moving Steganography and Steganalysis from the Laboratory into the Real World*. Ker, Bas, Böhme. 2013. ACM 978-1-4503-2081-8/13/06.