

# UNIDAD 3: BÚSQUEDA DE PARES SIMILARES

## MEDIDAS DE SIMILITUD Y DISTANCIA

---

Gibran Fuentes Pineda

Marzo 2021

# ¿CÓMO ENCONTRAMOS IMÁGENES?

house

Page 2 of about 413,000,000 results (0.08 seconds)

Related searches: [house tv show](#) [greg house](#) [house clipart](#) [cartoon house](#) [house music](#)



Click the Small House In The  
465 x 346 - 58k - jpg  
[supercoloring.com](#)  
Find similar images



The house ...  
600 x 400 - 93k - jpg  
[museumoffloridahistory.com](#)  
Find similar images



This large house ...  
500 x 375 - 43k - jpg  
[glamro.gov.uk](#)  
Find similar images



HouseplanGuys.com, The largest  
500 x 300 - 35k - jpg  
[houseplanGuys.com](#)  
Find similar images

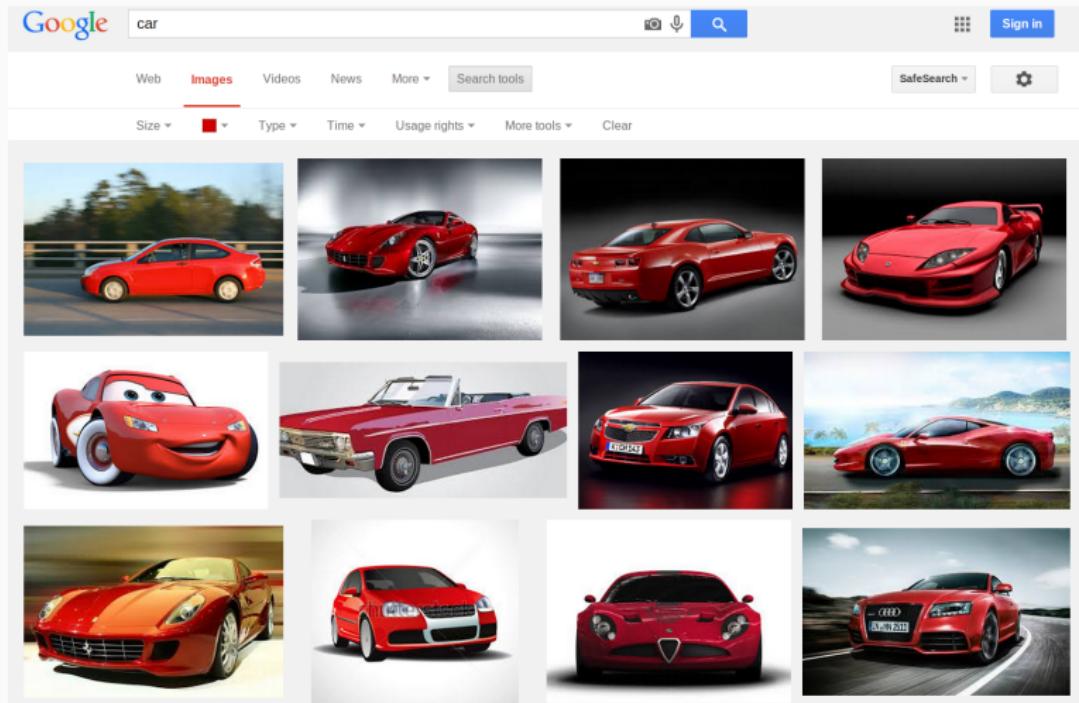


House picture by atkinson\_crystal  
800 x 600 - 299k - jpg  
[s588.photobucket.com](#)  
Find similar Images



Barack & Michelle Obama P.  
622 x 402 - 104k - jpg  
[hiptics.com](#)  
Find similar images

# BÚSQUEDA DE IMÁGENES POR COLOR



# BÚSQUEDA DE IMÁGENES POR ESTILO

Google bicycle

Web Images Videos News More Search tools SafeSearch Sign in

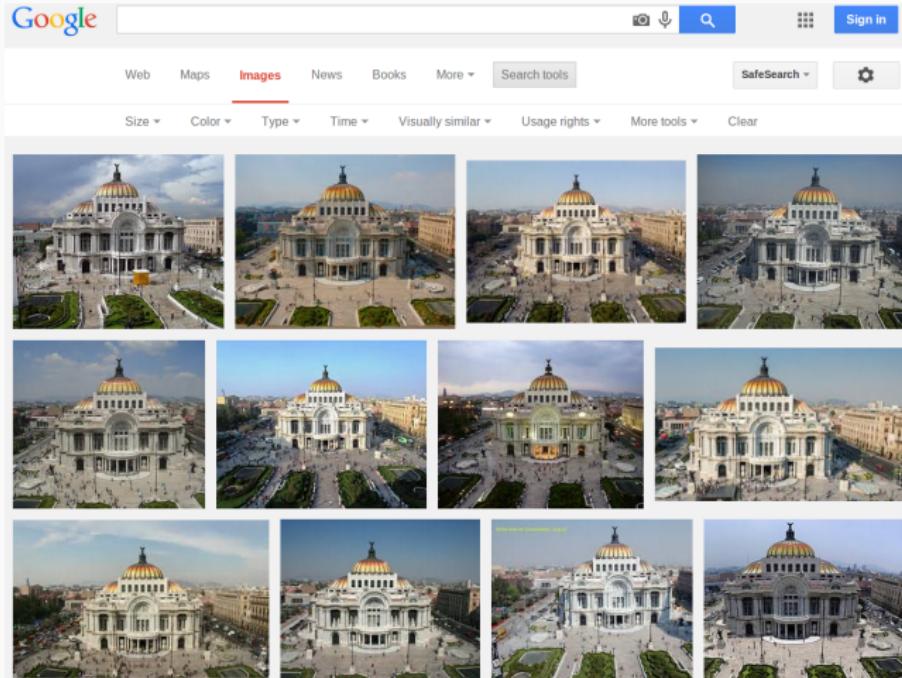
Size Color Line drawing Time Usage rights More tools Clear

d2011

# BÚSQUEDA DE IMÁGENES CON CATEGORÍAS ESPECÍFICAS



# BÚSQUEDA DE IMÁGENES VISUALMENTE SIMILARES



# BÚSQUEDA DE IMÁGENES DE LA MISMA CATEGORÍA



## EL PROBLEMA DEL VECINO MÁS CERCANO (1)

---

- Tarea frecuente en análisis de datos (por ej. agrupamiento de clientes similares, búsqueda de documentos sobre el mismo tema, etc.).

## EL PROBLEMA DEL VECINO MÁS CERCANO (1)

---

- Tarea frecuente en análisis de datos (por ej. agrupamiento de clientes similares, búsqueda de documentos sobre el mismo tema, etc.).
- Usado por algunos métodos no paramétricos de aprendizaje de máquinas (por ej. clasificador de k-vecinos más cercanos.).

## EL PROBLEMA DEL VECINO MÁS CERCANO (2)

---

- El problema es encontrar el par de objetos  $(x^{(1)}, x^{(2)}) \in \mathcal{X}$  que son más similares o que son más cercanos bajo algún criterio de similitud o distancia  $\mathcal{M}(x^{(1)}, x^{(2)})$ .

## EL PROBLEMA DEL VECINO MÁS CERCANO (2)

---

- El problema es encontrar el par de objetos  $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \in \mathcal{X}$  que son más similares o que son más cercanos bajo algún criterio de similitud o distancia  $\mathcal{M}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ .
- Usando fuerza bruta requeriría comparar todos los pares posibles en  $\mathcal{X}$ , lo cual es  $\binom{n}{2} = \Theta(n^2)$ .

- Cuantifica la disimilitud entre 2 objetos
- Debe cumplir las siguientes propiedades:
  1. **No negativo:**  $dist(x, y) \geq 0$
  2. **Identidad de los indiscernibles:**  $dist(x, y) = 0 \Leftrightarrow x = y$
  3. **Simétrico:**  $dist(x, y) = dist(y, x)$
  4. **Desigualdad del triángulo:**  $dist(x, y) \leq dist(x, z) + dist(z, y)$

# DISTANCIAS EN LA PERCEPCIÓN HUMANA

- No siempre se mantienen las propiedades de las distancias en la percepción humana. Por ej., la desigualdad del triángulo:



Imagen tomada de Veltkamp. *Shape matching: similarity measures and algorithms*, 2001.

## DISTANCIA EUCLIDIANA

---

- Representa el tamaño del segmento de línea que conecta dos puntos  $\mathbf{x}, \mathbf{y}$ .

$$dist(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^L (y_k - x_k)^2}$$

## SIMILITUD Y DISTANCIA COSENO

---

- La similitud coseno compara la orientación de 2 vectores mediante el coseno del ángulo entre ellos

$$S_C(\mathbf{x}, \mathbf{y}) = \cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^L x_i \cdot y_i}{\sqrt{\sum_{i=1}^L x_i^2} \cdot \sqrt{\sum_{i=1}^L y_i^2}}$$

- Dos vectores con la misma orientación tienen una similitud de 1
- Comúnmente usada para comparar documentos de texto

## DISTANCIA DE HAMMING

---

- Para un tamaño fijo  $T$ , es el número de elementos distintos de 2 vectores o cadenas
- **Ejercicio:** Calcula la distancia de *Hamming* de los siguientes objetos
  1. ‘asar’ y ‘azar’
  2. 57941137 y 23111431
  3. 10110010 y 01011110

## SIMILITUD Y DISTANCIA DE JACCARD

- Dados 2 conjuntos  $\{\mathcal{C}^{(1)}, \mathcal{C}^{(2)}\}$ , su similitud de Jaccard se define como:

$$J(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}) = \frac{|\mathcal{C}^{(1)} \cap \mathcal{C}^{(2)}|}{|\mathcal{C}^{(1)} \cup \mathcal{C}^{(2)}|} \in [0, 1].$$

- La distancia de Jaccard es

$$dist_J(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}) = 1 - J(\mathcal{C}^{(1)}, \mathcal{C}^{(2)})$$

- **Ejercicio:** Calcula la similitud de Jaccard de los conjuntos  $\mathcal{C}^{(1)} = \{0, 3, 6, 7\}$  y  $\mathcal{C}^{(2)} = \{2, 3, 5, 7\}$ .

## SIMILITUD Y DISTANCIA MINMAX (1)

- Generalización de similitud de Jaccard para bolsas:

$$J_{\mathcal{B}}(\mathcal{B}^{(1)}, \mathcal{B}^{(2)}) = \frac{\sum_{w=1}^D \min(\mathcal{B}_w^{(1)}, \mathcal{B}_w^{(2)})}{\sum_{w=1}^D \max(\mathcal{B}_w^{(1)}, \mathcal{B}_w^{(2)})} \in [0, 1],$$

donde  $\mathcal{B}_w^{(1)}$  y  $\mathcal{B}_w^{(2)}$  son las multiplicidades del elemento  $w$  en la bolsa  $\mathcal{B}^{(i)}$  y  $\mathcal{B}^{(j)}$  respectivamente.

- La distancia MinMax es

$$dist_{J_{\mathcal{B}}}(\mathcal{B}^{(1)}, \mathcal{B}^{(2)}) = 1 - J_{\mathcal{B}}(\mathcal{B}^{(1)}, \mathcal{B}^{(2)})$$

## SIMILITUD Y DISTANCIA MINMAX (2)

---

- **Ejercicio:** Calcula la similitud MinMax de las siguientes bolsas:
  - $\mathcal{B}^{(1)} = \{(0, 2), (3, 1), (6, 1), (7, 3)\}$
  - $\mathcal{B}^{(2)} = \{(2, 1), (3, 2), (5, 3), (7, 1)\}$

## TRASLAPE DE DOS CONJUNTOS

- Número de elementos en común sobre mínimo de elementos de dos conjuntos

$$ovr(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}) = \frac{|\mathcal{C}^{(1)} \cap \mathcal{C}^{(2)}|}{\min(\mathcal{C}^{(1)}, \mathcal{C}^{(2)})}$$

- **Ejercicio:** Calcula el traslape entre los siguientes pares de conjuntos:
  - $\mathcal{C}^{(1)} = \{0, 3, 6, 7\}$  y  $\mathcal{C}^{(2)} = \{2, 3, 5, 7\}$
  - $\mathcal{C}^{(1)} = \{0, 3, 6, 7\}$  y  $\mathcal{C}^{(2)} = \{0, 3, 7\}$
  - $\mathcal{C}^{(1)} = \{2, 3, 7\}$  y  $\mathcal{C}^{(2)} = \{2, 3, 4, 7\}$

- Árbol binario para realizar búsqueda del vecino más cercano de forma eficiente
- Cada nivel del árbol se compara con 1 dimensión
- Para buscar puntos
  1. Se construye el árbol con el conjunto de puntos disponible
  2. Dado un nuevo punto de consulta, se busca el punto más cercano recorriendo el árbol

## CONSTRUCCIÓN DE ÁRBOLES K-D

1. Elige dimensión de forma alternada (por ej. para 2D la raíz usa x, sus hijos y, los nietos x y así sucesivamente).
2. Inserta punto con valor en la mediana<sup>1</sup> de la dimensión seleccionada, puntos menores son descendientes en su rama izquierda y mayores en su derecha
3. Se repite 1 y 2 para los descendientes hasta que no haya más puntos.

---

<sup>1</sup>Es posible usar otros criterios para elegir el punto

## EJEMPLO DE CONSTRUCCIÓN DE UN ÁRBOL K-D

- Crea el árbol binario con los siguientes puntos:  
 $\mathcal{X} = \{(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)\}$ .

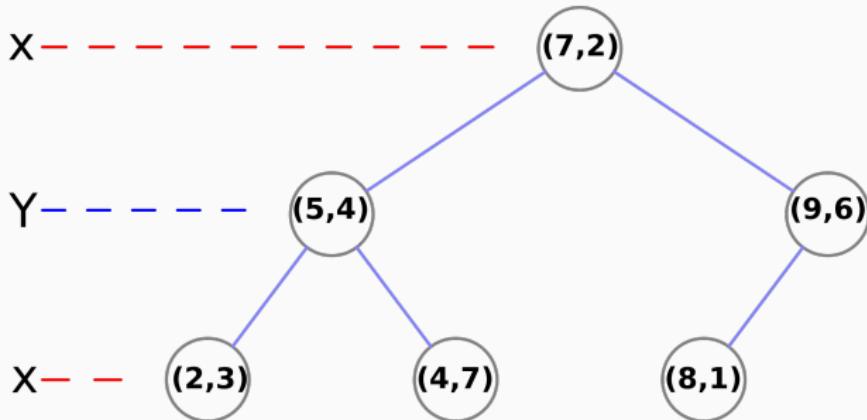


Imagen tomada de Wikipedia (k-d tree)

# PARTICIÓN DEL ESPACIO

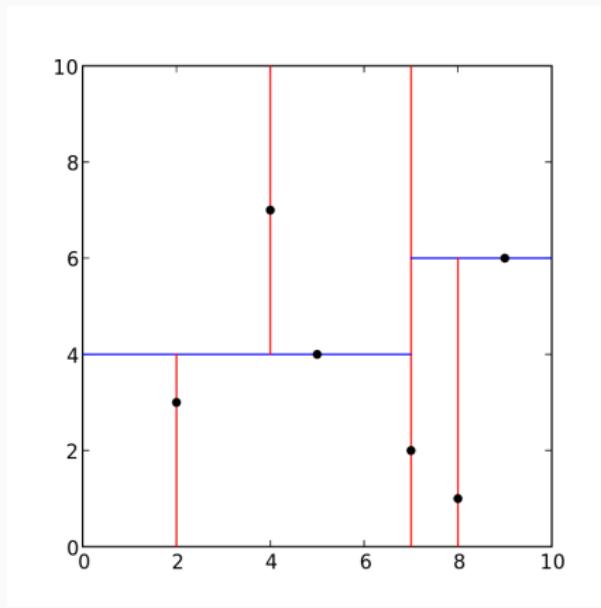


Imagen tomada de Wikipedia (k-d tree)

## INSERCIÓN DE PUNTOS AL ÁRBOL

---

1. Recorre el árbol a partir de la raíz y moviéndose hacia el descendiente correspondiente
2. Cuando se encuentra el nodo padre del punto a insertar, se agrega a la derecha o izquierda dependiendo del valor en la dimensión de partición.
3. En caso de estar desbalanceado, se aplica un algoritmo de re-balanceo para evitar pérdida de rendimiento

## BÚSQUEDA DEL VECINO MÁS CERCANO EN ÁRBOLES K-D

- Recorre el árbol a partir de la raíz y moviéndose hacia el descendiente correspondiente
  1. Mantén el punto más cercano  $c_{min}$  y quita los nodos del árbol que están más alejados a este
  2. Recorre los sub-árboles restantes<sup>2</sup>
- En el peor de los casos el tiempo de búsqueda es  $O(n)$  pero en promedio es  $O(\log(n))$
- Algoritmo sufre por la *maldición de la dimensionalidad*

---

<sup>2</sup>Existen heurísticas para elegir aquel que permita quitar más nodos.

# EJEMPLO DE BÚSQUEDA

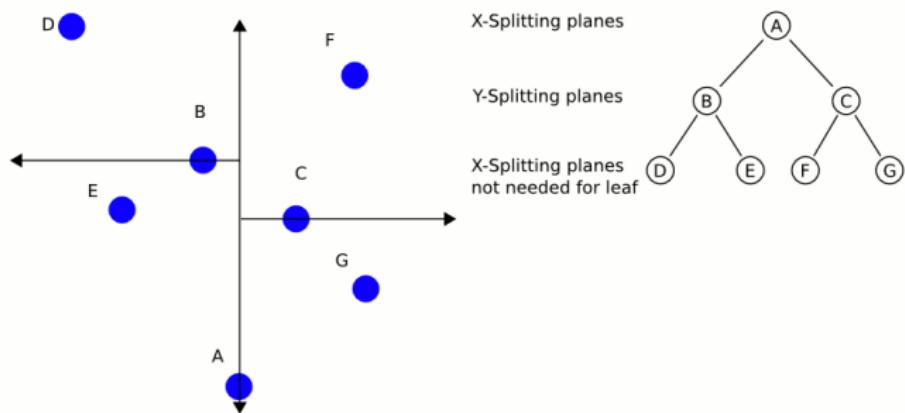


Imagen tomada de Wikipedia (k-d tree)

# EJEMPLO DE BÚSQUEDA

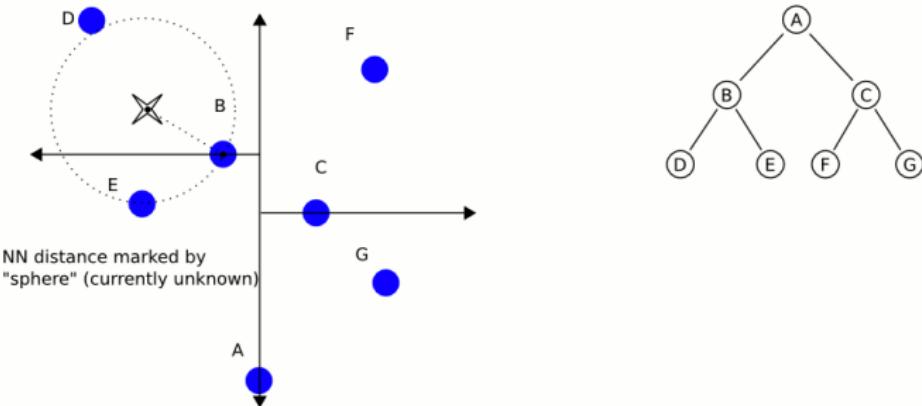


Imagen tomada de Wikipedia (k-d tree)

# EJEMPLO DE BÚSQUEDA

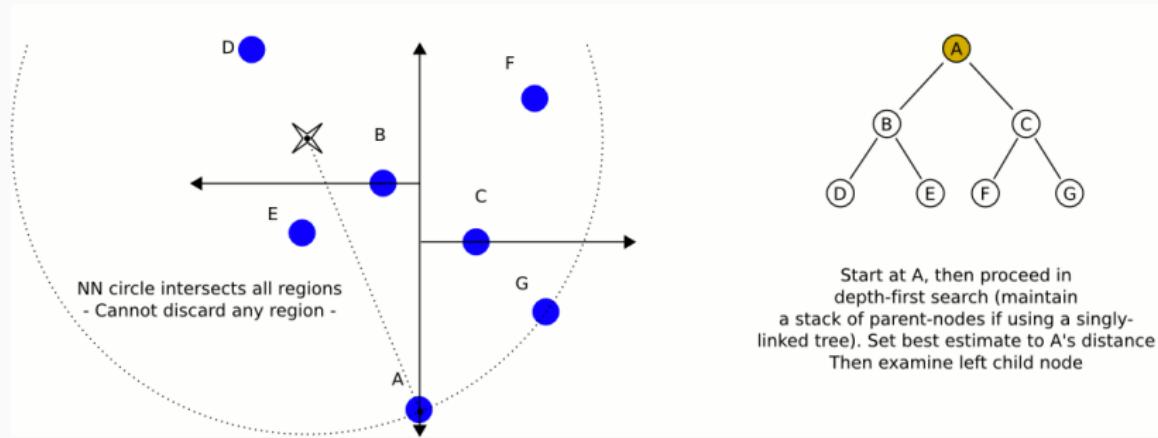
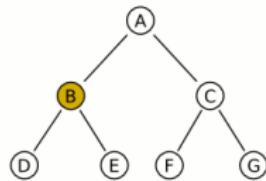
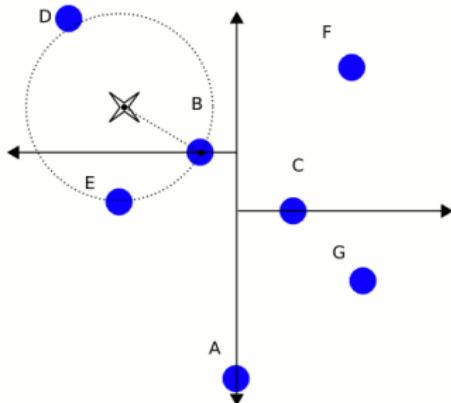


Imagen tomada de Wikipedia (k-d tree)

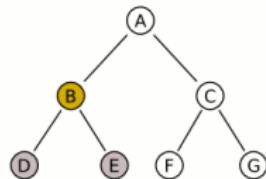
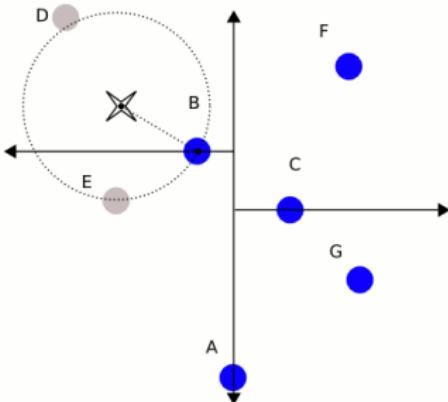
# EJEMPLO DE BÚSQUEDA



Calculate B's distance and compare against best estimate  
- It is smaller distance, so update best estimate. Examine children (left then right)

Imagen tomada de Wikipedia (k-d tree)

# EJEMPLO DE BÚSQUEDA



D & E Discarded as B  
(already visited) is closer.  
B is the best estimate for B's sub-branch  
Proceed back to parent node

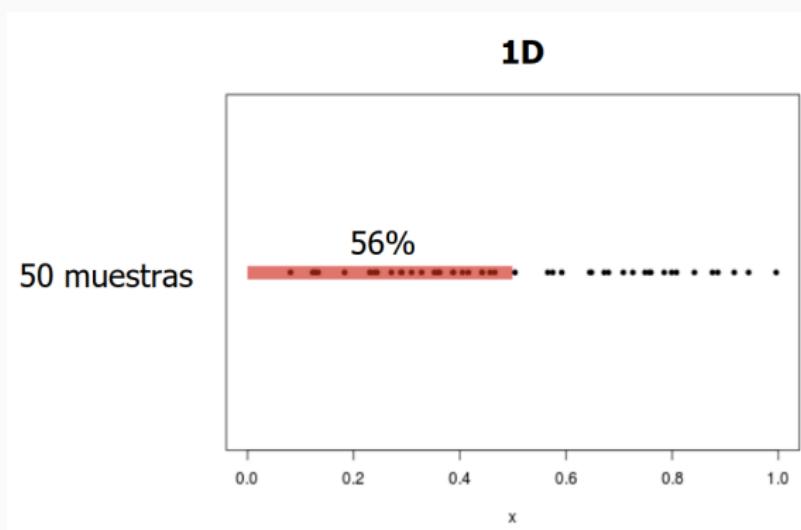
Imagen tomada de Wikipedia (k-d tree)

## EJERCICIO CON ÁRBOLES

- Construye el árbol binario K-D para el siguiente conjunto de puntos en 2D:  $\mathcal{X} = \{(8.3, 3.0), (6.2, 2.4), (0.2, 4.3), (7.5, 1.6), (3.6, 0.2), (2.5, 8.8), (1.7, 5.1)\}$
- Dibuja la partición del plano correspondiente
- Busca los vecinos más cercanos en  $\mathcal{X}$  de los siguientes puntos:  $\{(9.8, 1.7), (3.3, 9.6), (8.1, 1.2)\}$

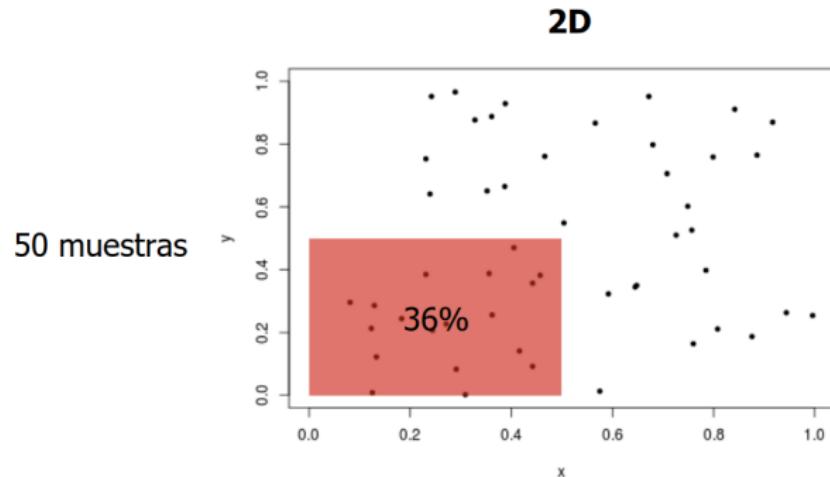
# LA Maldición DE LA DIMENSIONALIDAD

- Objetos cada vez más dispersos conforme aumenta el número de dimensiones



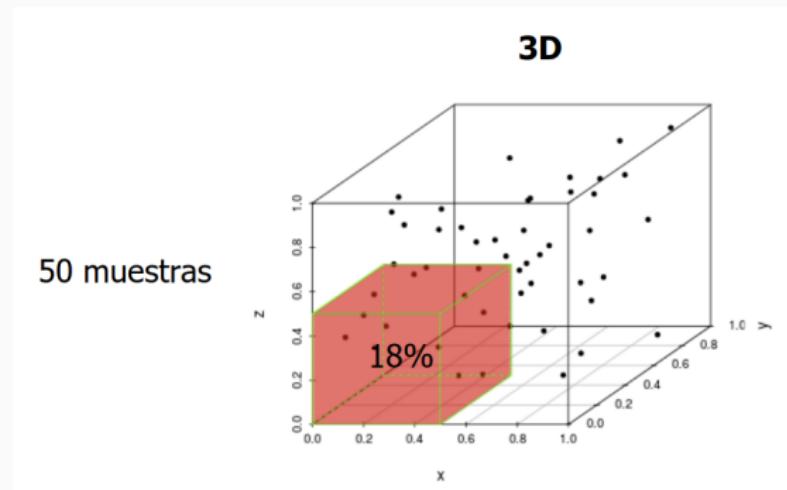
# LA Maldición DE LA DIMENSIONALIDAD

- Objetos cada vez más dispersos conforme aumenta el número de dimensiones



# LA Maldición DE LA DIMENSIONALIDAD

- Objetos cada vez más dispersos conforme aumenta el número de dimensiones



# LA Maldición DE LA DIMENSIONALIDAD

---

- Objetos cada vez más dispersos conforme aumenta el número de dimensiones
- **Ejercicio:** ¿Cuántas muestras necesitaría para cubrir un espacio de 1000 dimensiones con una precisión del 56 %?

# MATRIZ DOCUMENTO-TÉRMINO (1)

- Conjunto de documentos

$d_1$  *Ella toma café y él toma mate*

$d_2$  *Ella toma notas mientras toma café*

- Matriz documento-término

	café	él	ella	mate	mientras	notas	toma	y
$d_1$	1	1	1	1	0	0	1	1
$d_2$	1	0	1	0	1	1	1	0

- Bolsas de palabras

- $d_1 = \{\text{café}, \text{él}, \text{ella}, \text{mate}, \text{toma}, \text{y}\}$

- $d_2 = \{\text{café}, \text{ella}, \text{notas}, \text{mientras}, \text{toma}\}$

## MATRIZ DOCUMENTO-TÉRMINO (2)

- Conjunto de documentos

$d_1$  Ella toma café y él toma mate

$d_2$  Ella toma notas mientras toma café

- Matriz documento-término sin palabras vacías

	café	mate	notas	toma
$d_1$	1	1	0	1
$d_2$	1	0	1	1

- Bolsas de palabras binaria (conjunto)

- $d_1 = \{1, 2, 4\}$

- $d_2 = \{1, 3, 4\}$

## MATRIZ DOCUMENTO-TÉRMINO (3)

- Conjunto de documentos

$d_1$  Ella  $\underbrace{\text{toma}}_{w_4}$   $\underbrace{\text{café}}_{w_1}$  y él  $\underbrace{\text{toma}}_{w_4}$   $\underbrace{\text{mate}}_{w_2}$

$d_2$  Ella  $\underbrace{\text{toma}}_{w_4}$   $\underbrace{\text{notas}}_{w_3}$  mientras  $\underbrace{\text{toma}}_{w_4}$   $\underbrace{\text{café}}_{w_1}$

- Matriz documento-término sin palabras vacías con frecuencias

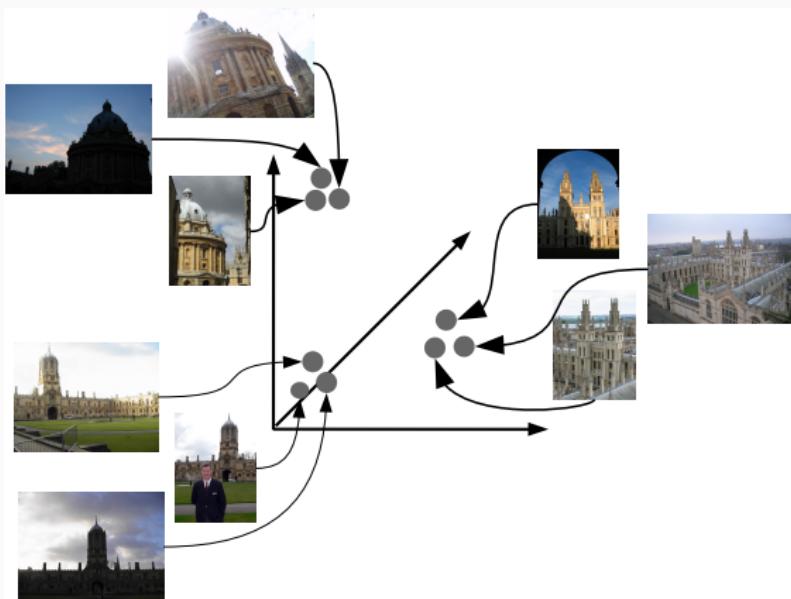
	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	w <sub>4</sub>
d <sub>1</sub>	1	1	0	2
d <sub>2</sub>	1	0	1	2

- Bolsas de palabras con frecuencia

- $d_1 = \{1, 2, 4, 4\}$
- $d_2 = \{1, 3, 4, 4\}$

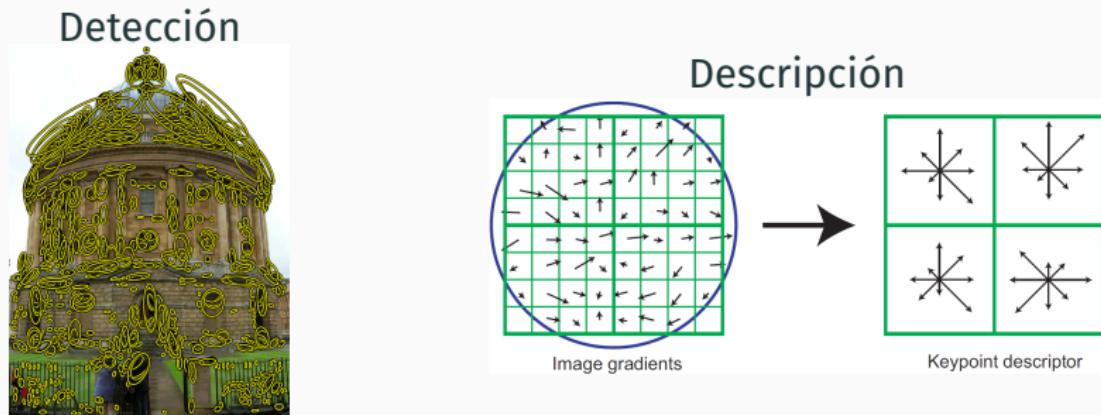
# REPRESENTANDO IMÁGENES (1)

- Buscamos mapear las imágenes a una representación compacta, discriminatoria, descriptiva, robusta y rápida de obtener



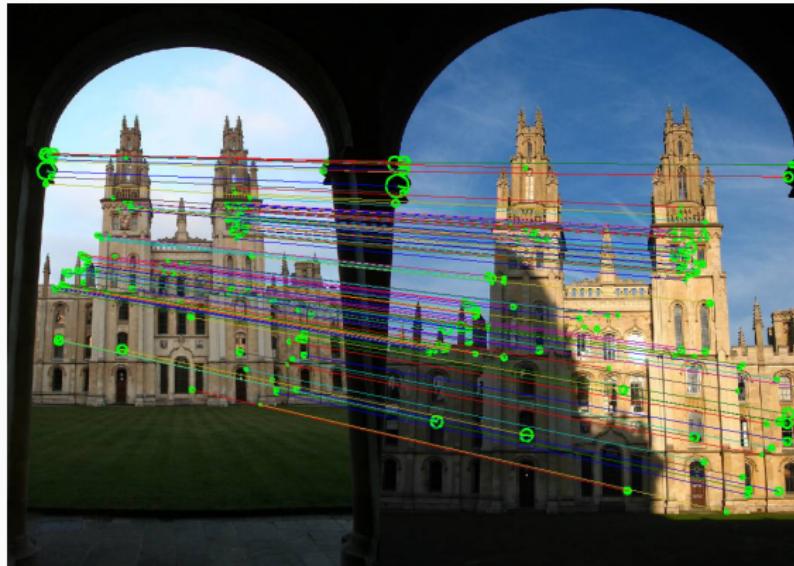
# REPRESENTANDO IMÁGENES (2)

- Dos tareas fundamentales:
  1. Detección de regiones de interés
  2. Descripción de cada región
- Imagen – conjunto de vectores característicos



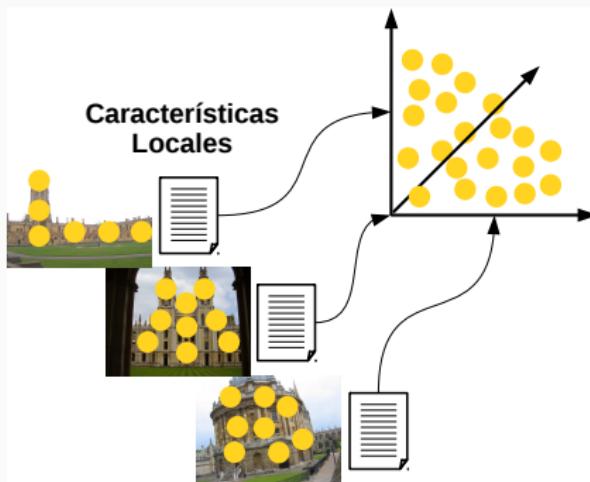
# ¿CÓMO COMPARO LAS IMÁGENES?

- Buscar características similares



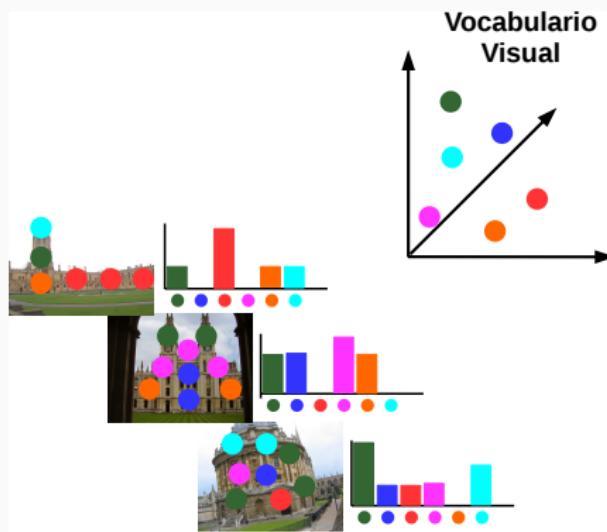
# LA ANALOGÍA CON EL TEXTO – SIVIC Y ZISSELMAN 2003

- Palabras – Características locales



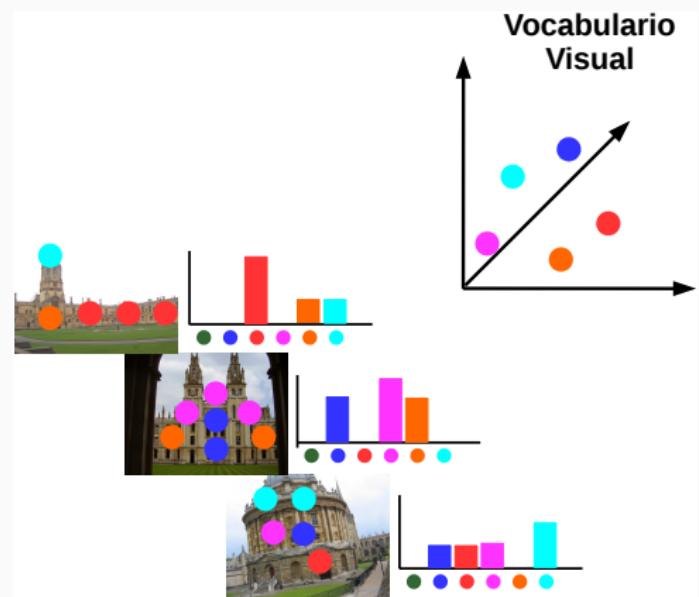
# LA ANALOGÍA CON EL TEXTO – SIVIC Y ZISSELMAN 2003

- Palabras – Características locales
- Stemming – Cuantización (ej. K-Means)



# LA ANALOGÍA CON EL TEXTO – SIVIC Y ZISSELMAN 2003

- Palabras – Características locales
- Stemming – Cuantización (ej. K-Means)
- Stop Words



# ¿CÓMO BUSCO IMÁGENES/DOCUMENTOS SIMILARES?

- Filas y columnas de la matriz documento-término usualmente dispersas y se representan por conjuntos o bolsas
- Compara solo las/los que compartan al menos una característica/palabra
- Ordena por valor de distancia o similitud

Palabra	Ocurre
0	3, 5, 7
1	6, 9
2	2, 5, 12, 20
3	1, 7, 17
4	2, 5, 7
:	:

- Búsqueda por índice inverso
  1. Recupera los conjuntos o bolsas de documentos donde ocurren las palabras en  $D$
  2. Calcula la distancia o similitud entre  $D$  y cada elemento en la lista
  3. Ordena  $d$  de acuerdo a las distancias o similitudes calculadas

- Son una secuencia de  $n$  objetos, que pueden ser símbolos ( $n$ -gramas de símbolos) o palabras ( $n$ -gramas de palabras)
- **Ejercicio:** Genera los 2-gramas y 3-gramas de símbolos y palabras de la siguiente oración *Ella toma café y él toma mate.*