



Recuperación de Información Multimedia

Búsquedas por Similitud

CC5213 – Recuperación de Información Multimedia

Departamento de Ciencias de la Computación

Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2019



Descripción de Contenido

- Fase previa: extracción de atributos numéricos de imágenes, audio, videos, textos
 - Descriptor de Contenido, Vector Característico, *“feature vector”*
- Similitud entre objetos corresponde a la cercanía entre puntos en el espacio vectorial
- El problema de búsqueda por similitud en el espacio original se convierte en buscar puntos cercanos en un espacio vectorial



Definiciones

- Sea \mathcal{D} el espacio de los objetos, el dominio
- Sea d una función que compara objetos:

$$d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^+ \cup \{0\}$$

- Función de Similitud: Mide el grado de “parecido” entre dos objetos
- Función de Disimilitud o Distancia: Mide el grado de “diferencia” entre dos objetos



Similitud y Disimilitud

- Transformación de similitud (s) a disimilitud (d)

- Lineal: $d = 1 - s, \quad d = \frac{\max s - s}{\max s - \min s}$

- Inverso: $d = \frac{1}{s} - 1, \quad d = \frac{1}{s - \min s + \varepsilon} - \frac{1}{\max s - \min s + \varepsilon}$

- Exponencial: $d = e^{-s}$

- En general, sirve cualquier función monótona decreciente



Propiedades

■ Propiedades de funciones de disimilitud:

□ No-negatividad

$$d(x, y) \geq 0$$

□ Reflexividad

$$d(x, y) = 0 \Leftrightarrow x = y$$

□ Simetría

$$d(x, y) = d(y, x)$$

□ Desigualdad triangular $d(x, z) \leq d(x, y) + d(y, z)$

■ Una función que cumpla con estas cuatro propiedades se denomina **Métrica**



Funciones no-métricas

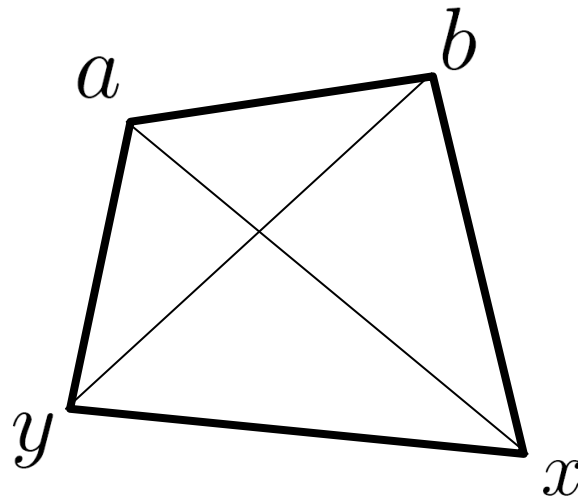
- Cuando una función no cumple alguna de las propiedades anteriores se le denomina **No-Métrica**
- En particular, las no-métricas se definen como:
 - **Pseudométrica**: no cumple reflexividad
 - **Quasimétrica**: no cumple simetría
 - **Semimétrica**: no cumple desigualdad triangular

Otras propiedades

- **Desigualdad Ptolemaica:** Para cualquier cuadrilátero se cumple que:

$$d(a, x) \cdot d(b, y) \leq d(a, b) \cdot d(x, y) + d(a, y) \cdot d(b, x)$$

- La igualdad se alcanza en cuadriláteros circunscritos





Otras propiedades

- Se llama **Ultramétrica** cuando la función cumple una desigualdad triangular más fuerte:

$$\forall x, y, z \quad d(x, z) \leq \max \{d(x, y), d(y, z)\}$$

- Notar que la desigualdad ultramétrica obliga que $d(x, y) \leq d(x, z) = d(y, z)$
- Ej: distancia entre strings es $(1/2)^n$ donde n es la posición donde difieren sus prefijos

Distancias entre vectores

■ Distancias de Minkowski (L_p)

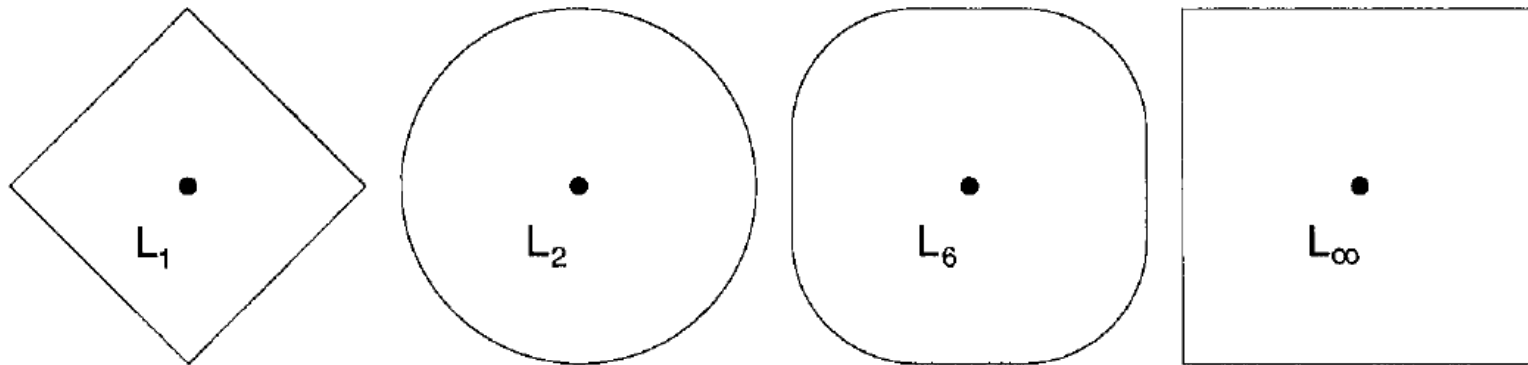
$$L_p(\vec{x}, \vec{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad p \geq 1$$

- $p=1$: Manhattan, taxicab, city block
- $p=2$: Euclidiana
- $p=\infty$: Máximo, chessboard, Chebyshev

$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^n |x_i - y_i| \quad L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad L_{\max}(\vec{x}, \vec{y}) = \max_{1 \leq i \leq n} \{|x_i - y_i|\}$$

Distancias entre vectores

- Distancias de Minkowski:
 - Costo de evaluación: $O(n)$
 - Cumple las propiedades métricas

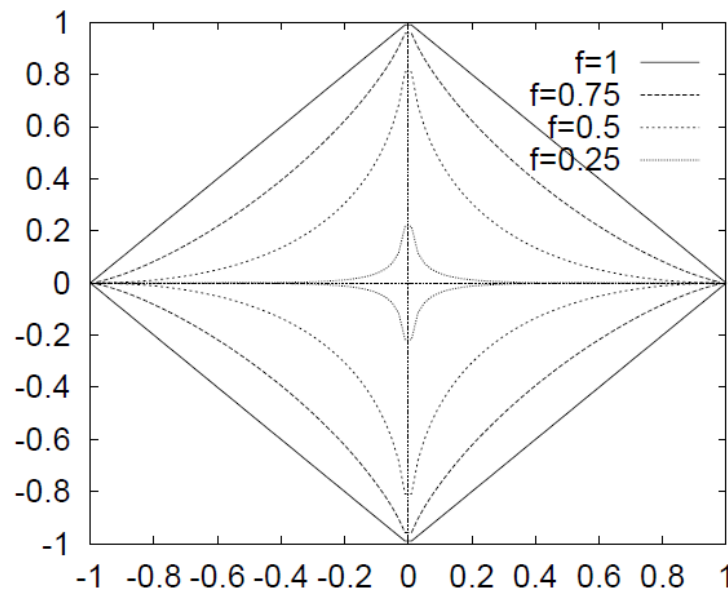


Círculos según para algunas L_p (puntos a una misma distancia del centro)

Distancias entre vectores

■ Distancias Fraccionales:

- L_p con $0 < p < 1$
- No cumplen la desigualdad triangular



Círculos radio 1



Distancias entre vectores

■ Distancias Fraccionales:

- Notar como las distancias fraccionales prefieren objetos con cambios concentrados en pocas dimensiones
- Por ejemplo:
 - Dado $\mathbf{q}=(9, 9, 9, 9)$ $\mathbf{A}=(5, 5, 5, 5)$ $\mathbf{B}=(1, 1, 9, 9)$
 - Usando L_2 : \mathbf{A} es el más cercano a \mathbf{q}
 - Usando L_1 : \mathbf{A} y \mathbf{B} están a igual distancia de \mathbf{q}
 - Usando $L_{0.5}$: \mathbf{B} es el más cercano a \mathbf{q}



Distancias entre vectores

- DPF (Dynamic Partial Function)
 - L_p comparando sólo un subconjunto de las dimensiones.
 - Δ_m es el conjunto de las m dimensiones donde hay menor diferencia.
 - Las dimensiones comparadas dependen de los vectores a comparar.

$$\text{DPF}(\vec{x}, \vec{y}) = \left(\sum_{i \in \Delta_m} |x_i - y_i|^p \right)^{\frac{1}{p}} \quad p \geq 1$$



Distancia cuadrática

- Formas cuadráticas $QFD(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \cdot A \cdot (\vec{x} - \vec{y})}$
 - A : matriz de similitud
 - Costo de evaluación: $O(n^2)$
- Ejemplos
 - Euclidiana (A =matriz identidad)
 - Weighted Euclidean Distance (A =diagonal)
 - Mahalanobis (A =inv. matriz de covarianza)
- Cumple propiedades métrica y ptolemaica



Distancia entre conjuntos

- Coeficiente de Jaccard:

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

- ☐ Usado para comparar conjuntos de respuestas o comparar bits.
- ☐ Cumple las propiedades métricas.

- Distancia de Tanimoto (generalización a vectores):

$$d_{TS}(\vec{x}, \vec{y}) = 1 - \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|^2 + \|\vec{y}\|^2 - \vec{x} \cdot \vec{y}}$$



Distancia entre nube de puntos

- Distancia Hausdorff:

$$d(A, B) = \max\{d_s(A, B), d_s(B, A)\}$$

$$\begin{aligned} d_s(A, B) &= \sup_{x \in A} d_p(x, B) & d_p(x, B) &= \inf_{y \in B} d_e(x, y) \\ d_s(B, A) &= \sup_{y \in B} d_p(A, y) & d_p(A, y) &= \inf_{x \in A} d_e(x, y) \end{aligned}$$

- Es la “máxima distancia mínima” entre dos nubes de puntos
- d_e es la ground-distance entre pares de puntos (e.g. L_2)
- Se usa para comparar el grado de coincidencia entre dos formas o figuras



Búsquedas por similitud

- Sea $\mathcal{R} \subseteq \mathcal{D}$ un conjunto de objetos, el espacio de búsqueda
- Sea $q \in \mathcal{D}$ un objeto de consulta
- Se definen las búsquedas:
 - Búsqueda exacta
 - Búsqueda por rango (range query)
 - Búsqueda del vecino más cercano (nearest neighbors query)



Búsqueda exacta

- Determinar si el objeto q está en \mathcal{R}
- Algoritmo secuencial (linear scan):

```
foreach  $u_i \in \mathcal{R}$  do
    | if  $u_i = q$  then
    | | return true ;
    | end
end
return false ;
```

- Se puede hacer más eficiente si los objetos se pueden ordenar de menor a mayor (ver TDA Diccionario)



Búsqueda por rango

- Recuperar los objetos del espacio de búsqueda a distancia menor o igual a r de q :

- $r \in \mathbb{R}$ es el radio de tolerancia

$$\mathcal{R}(q, r) = \{u \in \mathcal{R}, d(u, q) \leq r\}$$

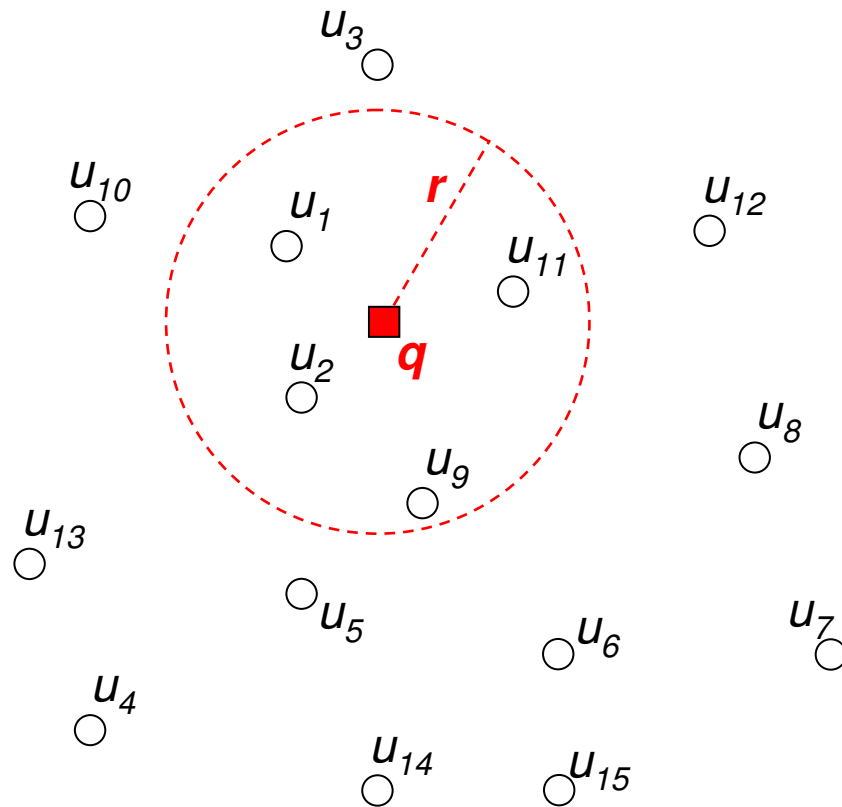
- Se llama “Bola de Consulta” al subespacio:

$$\mathcal{B}(q, r) = \{x \in \mathcal{D}, d(x, q) \leq r\}$$

- La búsqueda por rango es recuperar los objetos de \mathcal{R} que están dentro de la bola de consulta:

$$\mathcal{R}(q, r) = \mathcal{R} \cap \mathcal{B}(q, r)$$

Búsqueda por rango



$$R(q, r) = \{ u_1, u_2, u_9, u_{11} \}$$



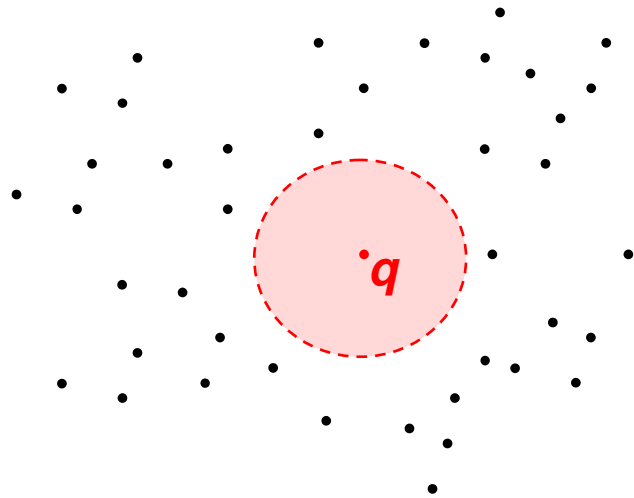
Búsqueda por rango

- Algoritmo secuencial (linear scan):

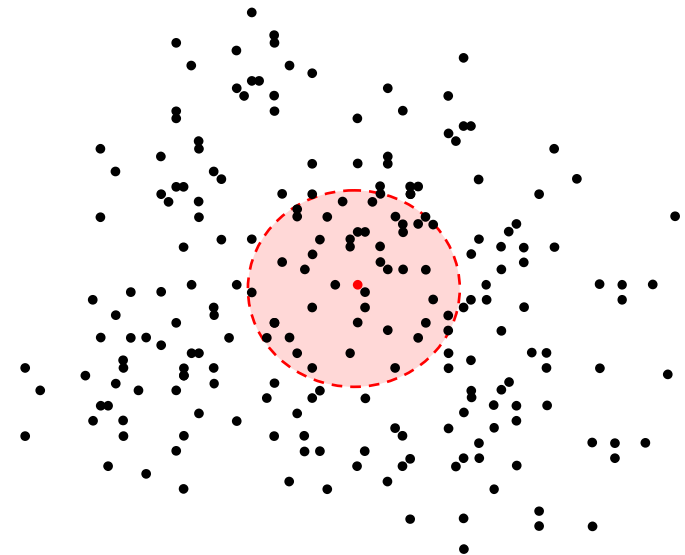
```
queue  $\leftarrow \emptyset$  ;  
foreach  $u_i \in \mathcal{R}$  do  
    if  $d(u_i, q) \leq r$  then  
        | queue.Add( $u_i$ ) ;  
    end  
end  
Print(queue);
```

Búsqueda por rango

- ¿Cómo fijar el radio de tolerancia?



Muy pequeño: no retorna nada



Muy grande: retorna demasiado



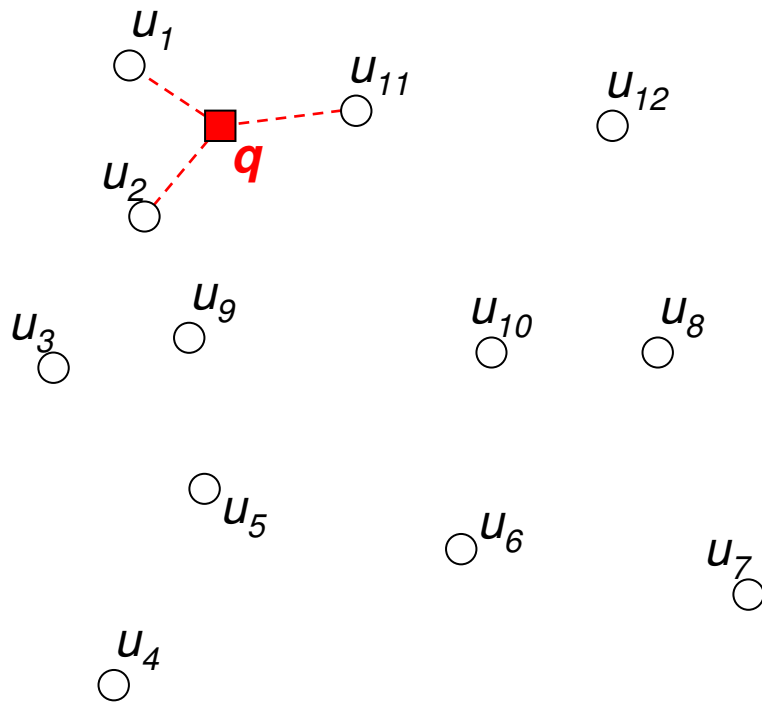
Búsqueda del vecino más cercano

- *k-Nearest Neighbors*: Recuperar los k objetos del espacio de búsqueda con menor distancia a q

$$k\text{NN}(q) = \{ \mathcal{C} \subseteq \mathcal{R}, |\mathcal{C}| = k \wedge \\ \forall x \in \mathcal{C}, y \in \mathcal{R} - \mathcal{C}, d(x, q) \leq d(y, q) \}$$

- Parámetro: $k \in \mathbb{N}$
- Existe más de una respuesta válida cuando hay varios objetos a la misma distancia de q

Búsqueda del vecino más cercano



$$3NN(q) = \{ u_1, u_2, u_{11} \}$$



Búsqueda del vecino más cercano

■ Algoritmo secuencial 1-NN (linear scan):

```
candidate  $\leftarrow$  null ;  
candidate_dist  $\leftarrow +\infty$  ;  
foreach  $u_i \in \mathcal{R}$  do  
    | dist  $\leftarrow d(u_i, q)$  ;  
    | if dist < candidate_dist then  
    | | candidate  $\leftarrow u_i$  ;  
    | | candidate_dist  $\leftarrow$  dist;  
    | end  
end  
Print(candidate);
```



Búsqueda k vecinos más cercanos

■ Algoritmo secuencial k -NN (linear scan):

```
candidates  $\leftarrow \emptyset$  ; // Priority Queue (Max-Heap)
foreach  $u_i \in \mathcal{R}$  do
    | dist  $\leftarrow d(u_i, q)$  ;
    | if candidates.Size()  $< k$  then
    | | candidates.Add(dist,  $u_i$ ) ;
    | else if candidates.Get-Max().priority  $>$  dist then
    | | candidates.Remove-Max() ;
    | | candidates.Add(dist,  $u_i$ ) ;
    | end
end
Print(candidates);
```



Otros tipos de búsquedas

- Combinaciones de criterios:
 - Consulta por rango + k -NN
 - Consulta k -NN con región de búsqueda (constrained k -NN)
- Reverse Nearest Neighbor:
 - Dado q recuperar todos los objetos de R para los cuales q es uno de sus k -NN
- Similarity Join:
 - Dado dos conjuntos Q y R resolver una búsqueda k -NN o por rango para cada q_i en Q , es decir, obtener los k -NN de todos los objetos en Q (All-Nearest Neighbors) o todos los pares donde $d(q_i, u_j) \leq r$
- Self Similarity Join:
 - Resolver un Similarity Join cuando $Q=R$



Tipos de Índices

■ Índices Multidimensionales

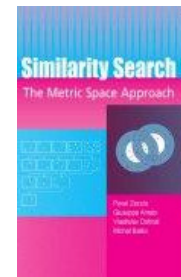
- Asumen que los datos son vectores y usan los valores de las coordenadas para agruparlos
- **Árboles**: Agrupan vectores en regiones espaciales ordenadas jerárquicamente
- **Hashing**: Asignan vectores a una o más tablas de tamaño fijo
- **Filling Curves**: Convierten el espacio multi-dimensional en un espacio unidimensional

■ Índices Métricos

- Pueden indexar cualquier tipo de objeto mientras la función de distancia pueda compararlos
- Usan las propiedades métricas de la función de distancia
- Seleccionan objetos de referencia y los comparan con el resto

Bibliografía

- **Similarity Search: The Metric Space Approach.** Zezula et al. 2006.
 - Capítulo 1, Secciones 1-4.





Papers

- Aggarwal et al. **On the Surprising Behavior of Distance Metrics in High Dimensional Space.** 2001.
- Rubner et al. **Empirical Evaluation of Dissimilarity Measures for Color and Texture.** 2001.
- Huttenlocher et al. **Comparing images using the Hausdorff distance.** 1993.
- Meng et al. **Enhancing DPF for near-replica image recognition.** 2003.
- Skopal et al. **On Nonmetric Similarity Search Problems in Complex Domains.** 2011.