Recuperación de Información Multimedia

Índices Métricos

CC5213 – Recuperación de Información Multimedia
Departamento de Ciencias de la Computación
Universidad de Chile
Juan Manuel Barrios – https://juan.cl/mir/ – 2019

Espacios Métricos

- Definición:
  - Universo de objetos válidos: $\mathcal{D}$
  - Función de distancia: $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$
  - El par $(\mathcal{D}, d)$ es un espacio métrico ssi d cumple con las propiedades métricas:
    - Positividad estricta $\forall x, y \in \mathbb{X},\ x \neq y \Rightarrow \delta(x, y) > 0$
    - Simetría $\forall x, y \in \mathbb{X},\ \delta(x, y) = \delta(y, x)$
    - Reflexividad $\forall x \in \mathbb{X},\ \delta(x, x) = 0$
    - Desigualdad triangular $\forall x, y, z \in \mathbb{X},\ \delta(x, z) \leq \delta(x, y) + \delta(y, z)$

Espacios Métricos

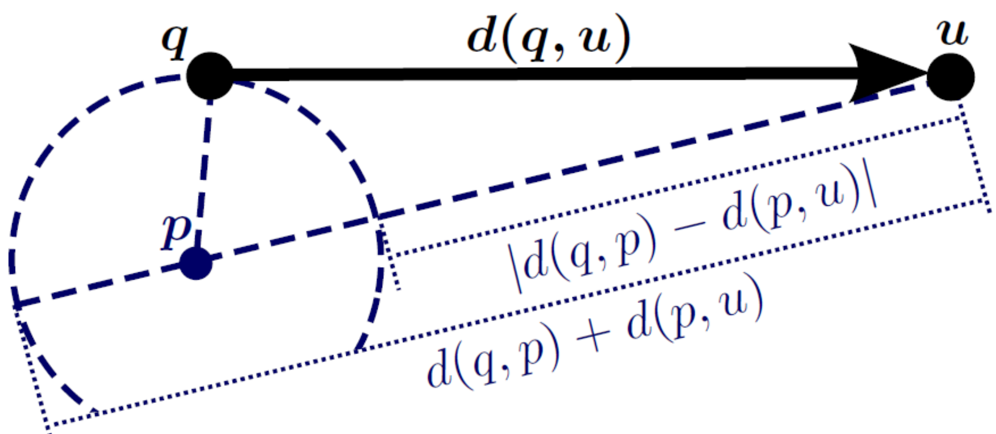
- Objetos de $\mathcal{D}$ son comparados utilizando la función de distancia
- La función d indica el grado de disimilitud entre dos objetos
- Ejemplo de espacio métrico:
  - Strings y distancia de edición
  - Vectores y una distancia de Minkowski Lp
  - Signatures y distancia EMD

Metric Access Methods

- Un Metric Access Method (MAM) es una estructura de datos que intenta reducir el número de veces que se evalúa la función de distancia al resolver búsquedas por similitud
  - Supuesto: la función de distancia es costosa de evaluar

Pivote

- Un pivote es un objeto de la colección fijo
- Permite calcular una cota inferior y superior de *d(q,u)*
  - Con la cota inferior se puede saber si *q* y *u* son lejanos entre sí
  - Los valores *d(q,p)* y *d(p,u)* se deben mantener en memoria



$$|d(q,p) - d(p,u)| \leq d(q,u) \leq d(q,p) + d(p,u)$$

Conjuntos de Pivotes

- El valor de las cotas mejoran (se acercan más al valor de *d*) cuando se usan más pivotes
- Dado un conjunto de pivotes $\mathcal{P}$:
  - Cota inferior:

$$LB_{\mathcal{P}}(q,r) = \max_{p \in \mathcal{P}} \{|d(q,p) - d(r,p)|\}$$

  - LB_P utiliza solamente valores precalculados

Tablas de Pivotes

- AESA (Approximating and Eliminating Search Algorithm, 1986)
  - Todos los objetos de la colección se pueden usar como pivote
  - Requiere mantener una tabla de distancias entre todos los pares de objetos del dataset
    - Memoria $O(n^2)$
- LAESA (Linear AESA, 1994)
  - Seleccionar subconjunto de elementos de la colección como pivotes
  - ¿Cómo resolver una búsqueda eficientemente?
  - ¿Cómo seleccionar el conjunto de pivotes?

Creación de Tabla de Pivotes

- Dados *k* pivotes de la colección
- Construir una tabla con las $k \cdot n$ distancias entre cada pivote y cada objeto

	$\boldsymbol{p_1}$	**…**	$\boldsymbol{p_k}$
$\boldsymbol{u_1}$	$d(p_1,u_1)$	…	$d(p_k,u_1)$
…	…	…	…
$\boldsymbol{u_n}$	$d(p_1,u_n)$	…	$d(p_k,u_n)$

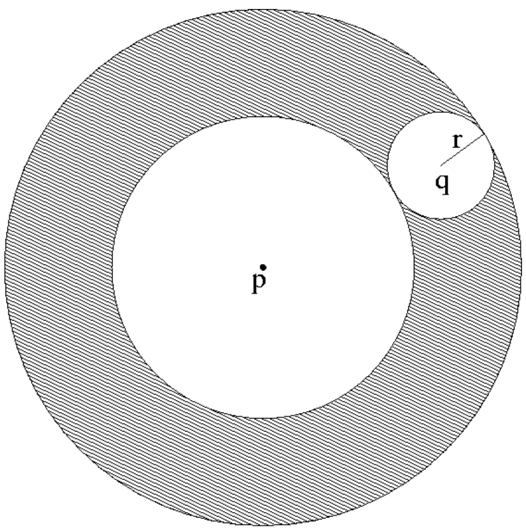
Consulta por rango

- Calcular la distancia entre *q* y cada pivote
- Para cada objeto:
  - Calcular su cota inferior
  - Criterio de exclusión: Si la cota inferior es mayor que *r* el objeto no es relevante (se descarta)
    - Notar que basta un pivote para descartar el objeto, por lo que no es necesario evaluar todos los pivotes
  - Si no pudo ser descartado se evalúa su distancia real y se determina si es relevante o no

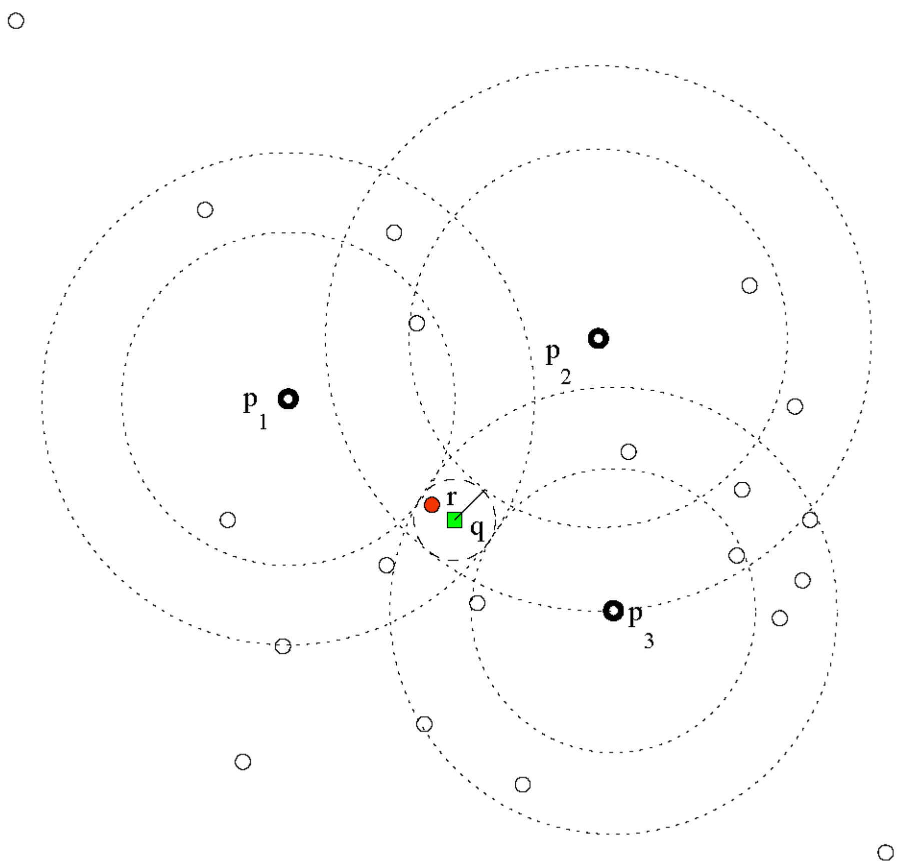
foreach $p_i \in \mathcal{P}$ **do**
  evaluar $d(p_i, q)$ y guardar
end
queue $\leftarrow \emptyset$;
foreach $u_i \in \mathcal{R}$ **do**
  if $LB_{\mathcal{P}}(q, u_i) > r$ **then**
    continue;
  else if $d(q, u_i) \leq r$ **then**
    queue.Add(u_i) ;
  end
end
Print(queue);

Criterio de Exclusión

- El criterio de exclusión descarta todos los objetos *u* que cumplan: $|d(q,p) - d(p,u)| > r$
- Gráficamente en el plano, usando un pivote se descartan todos los objetos que están fuera del anillo



Ejemplo consulta por rango



$d(p_1,u_1)$	$d(p_2,u_1)$	$d(p_3,u_1)$
$d(p_1,u_2)$	$d(p_2,u_2)$	$d(p_3,u_2)$
…	…	…
$d(p_1,u_n)$	$d(p_2,u_n)$	$d(p_3,u_n)$

Criterio de exclusión:

$LB_P(q,u_i) > r$

Consulta k-NN

- Similar a una consulta por rango donde *r* es la distancia al candidato actual
- Calcular la distancia entre *q* y cada pivote.
- Para cada objeto:
  - Calcular su cota inferior
  - Si la cota inferior es mayor que el candidato actual el objeto no es relevante (se descarta)
  - Si no pudo ser descartado se evalúa su distancia real y se determina si es mejor que el candidato actual o no

foreach $p_i \in \mathcal{P}$ **do**
  evaluar $d(p_i, q)$ y guardar
end
candidate $\leftarrow$ **null** ;
candidate_dist $\leftarrow +\infty$;
foreach $u_i \in \mathcal{R}$ **do**
  if $LB_{\mathcal{P}}(q, u_i) \geq$ candidate_dist **then**
    continue;
  end
  dist $\leftarrow d(u_i, q)$;
  if dist $<$ candidate_dist **then**
    candidate $\leftarrow u_i$;
    candidate_dist $\leftarrow$ dist;
  end
end
Print(candidate);

Espacio de pivotes

- Espacio de pivotes es un espacio *k*-dimensional, donde cada coordenada es la distancia entre el objeto y cada pivote

  Convertir *u* en vector k-dimensional:

  $$v_{\mathcal{P}}(u) = (d(p_1, u) \ldots d(p_k, u))^T$$

  Notar que:

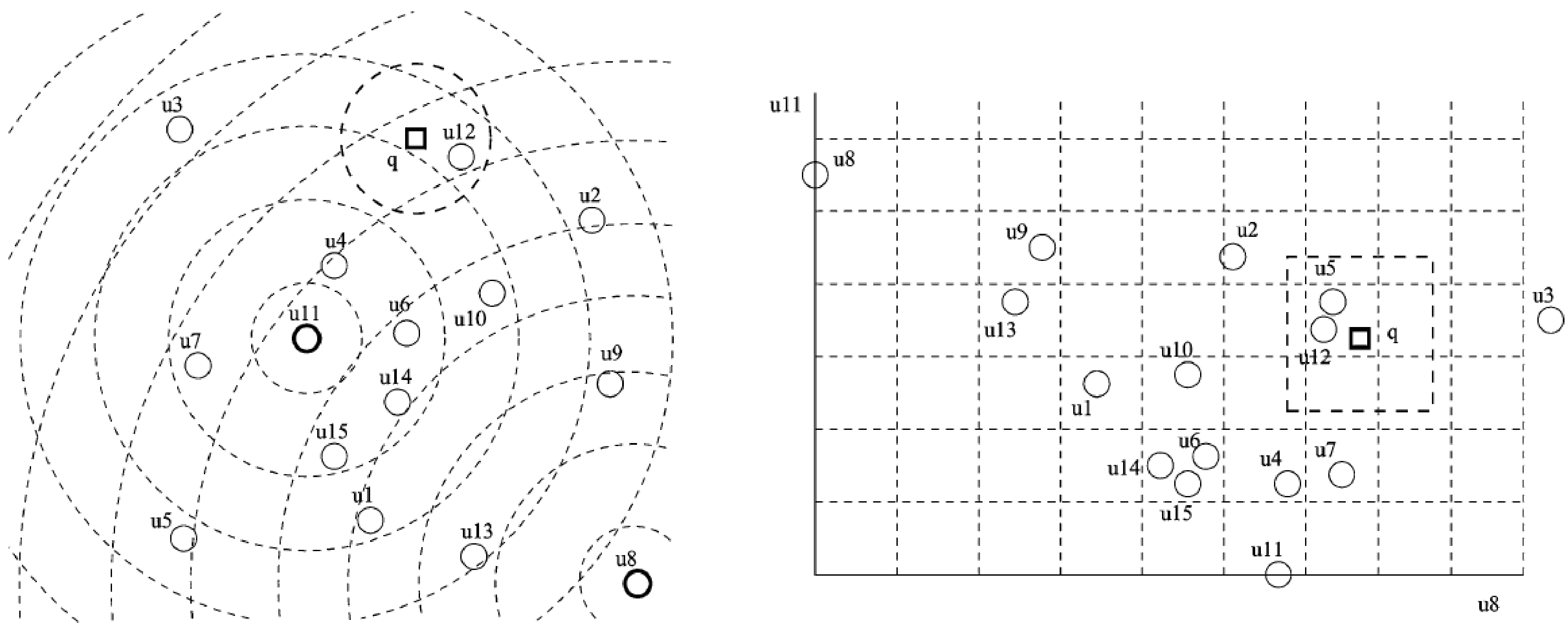
  $$LB_{\mathcal{P}}(q, u_i) = \mathrm{L}_{\max}(v_{\mathcal{P}}(q), v_{\mathcal{P}}(u_i))$$

- Criterio de exclusión en el espacio de pivotes:

  $$\mathrm{L}_{\max}(v_{\mathcal{P}}(q), v_{\mathcal{P}}(u_i)) > r$$

Espacio de pivotes

- Convertir espacio métrico al espacio de pivotes



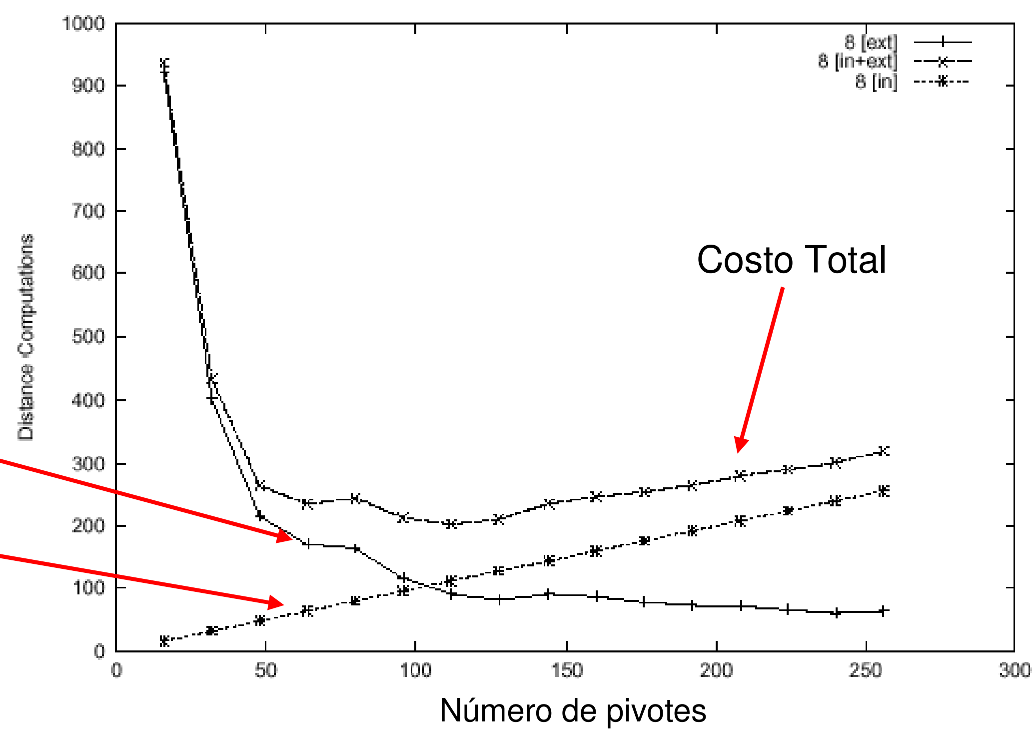
Ver Chavez et al. 2001.

Complejidad Interna y Externa

- Complejidad externa:
  - Cómputos de distancia entre *q* y objetos no descartados
- Complejidad interna:
  - Cómputos de distancia entre *q* y pivotes
  - Cómputos de *LB* entre *q* y todos los objetos
- Al aumentar el número de pivotes:
  - Disminuye la complejidad externa
  - Aumenta la complejidad interna (linealmente)
- Existe un número óptimo de pivotes
  - Comparar performance del óptimo contra no usar índice

Complejidad versus Pivotes

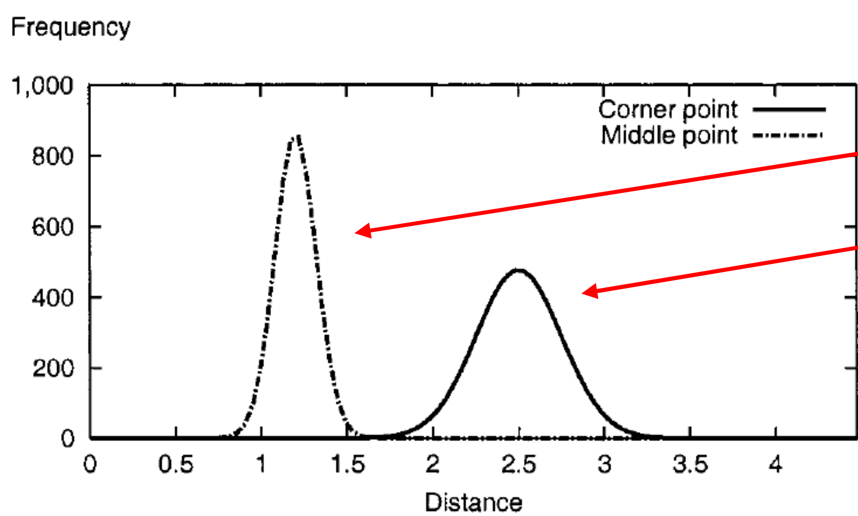
- Al aumentar el número de pivotes:
  - Disminuye la complejidad externa (se descartan más objetos)
  - Aumenta la complejidad interna (crece el trabajo realizado para descartar un objeto)
- Hay un punto óptimo



Ver Chavez et al. 2001.

Selección de pivotes

- Dependiendo del dataset, hay objetos que son mejores pivotes que otros.
  - Mejor pivote → Descarta más distancias → Cotas inferiores lo más altas posible
- Ejemplo: si tenemos datos en un cubo unitario de 20 dimensiones:



Distancias entre los datos y el punto central

Distancias entre los datos y una esquina

Ver Zezula cap 10.5

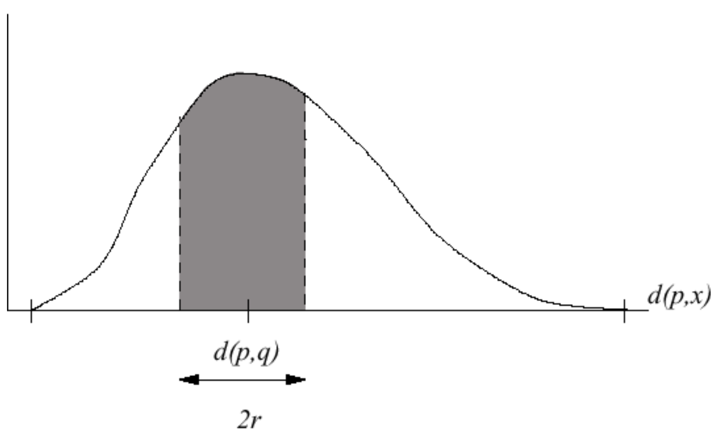
Selección de pivotes

- Una baja varianza en el histograma de distancias implica que al restar la distancia entre dos objetos probablemente será cercano a cero
  - El punto central es un mal pivote porque todos los objetos están casi a una misma distancia de él
  - Puntos en la esquinas obtienen una mayor varianza en las distancias
- Sin embargo, en un espacio métrico genérico no hay geometría!

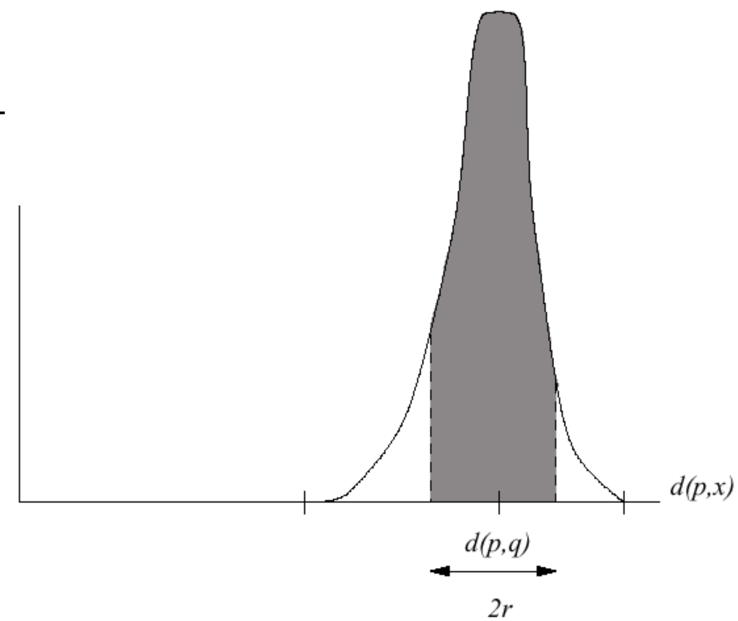
Dimensión Intrínseca

- Concepto de alta dimensión en espacios métricos:

$$\rho = \frac{\mu^2}{2\sigma^2}$$



ρ bajo



ρ alto

Ver Chavez et al. 2001.

Selección de pivotes

- Criterio de selección 1: elegir un conjunto de pivotes en forma aleatoria
  - El rendimiento de la búsqueda depende del conjunto de pivotes elegido
  - Se debe evaluar un conjunto de pivotes
  - Elegir varios conjuntos de pivotes y elegir el que parezca mejor
- Notar que la selección de pivotes es parte de la creación del índice, es decir, cuando aún no se conocen los objetos de consulta

Evaluación de pivotes

- Criterio de evaluación: Un buen conjunto de pivotes debe dar el valor de *LB* más alto posible
- Dados *N* conjuntos de pivotes, escoger el grupo que minimiza la diferencia entre la cota inferior y la distancia real:
  - Calcular μ_Δ para cada grupo (diferencia promedio entre LB_P y *d*)
  - Escoger el grupo que minimiza μ_Δ

Evaluación de pivotes

- Estimación de μ_Δ para un conjunto de pivotes *P*
  - Elegir al azar *m* pares de puntos (a_i,b_i)
  - Para todos los pares calcular $\Delta_i = |LB_P(a_i,b_i)-d(a_i,b_i)|$
  - El valor estimado de μ_Δ es el promedio de los Δ_i
  - Pero el valor de $d(a_i,b_i)$ es el mismo para todos los conjuntos!
  - Por lo tanto, el criterio de evaluación consiste en calcular LB_P promedio de cada conjunto *P* y escoger el conjunto que lo maximice
  - Costo para estimar μ_Δ: *km* cálculos de *d*

Selección de pivotes

- Idea:
  - Dos pivotes muy cercanos entre sí no mejoran mucho el valor de *LB*
  - Se deben evitar pivotes cercanos y preferir los pivotes que están lejos entre sí
- Criterio de selección 2:
  - Dado un parámetro de distancia mínima crear una “zona de exclusión” alrededor de cada pivote

Selección de pivotes

- SSS: Sparse Spatial Selection
  - Realizar un recorrido aleatorio de los objetos de la colección y elegir objetos distantes entre sí.
- Parámetro de exclusión *M*α:
  - *M*: máxima distancia en el espacio
  - α: factor (típicamente 0.4)
- Seleccionar con SSS distintos conjuntos reduciendo *M*α hasta obtener varios conjuntos con el tamaño deseado y luego quedarse con el mejor

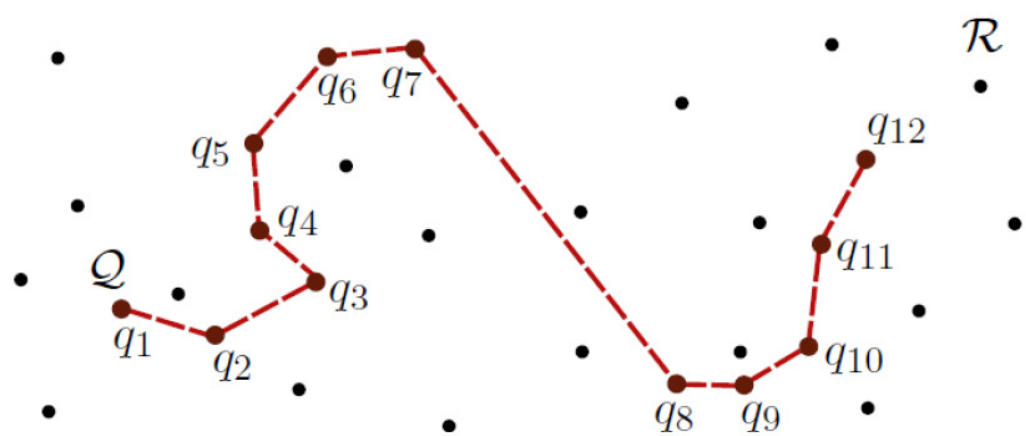
$$
\begin{array}{l}
PIVOTS \leftarrow \{x_1\} \\
\textbf{for all } x_i \in \mathbb{U} \textbf{ do} \\
\quad \textbf{if } \forall\, p \in PIVOTS,\ d(x_i, p) \geq M\alpha \textbf{ then} \\
\quad\quad PIVOTS \leftarrow PIVOTS \cup \{x_i\} \\
\quad \textbf{end if} \\
\textbf{end for}
\end{array}
$$

Selección de pivotes

- Se debe notar que el valor de *LB* finalmente depende de un solo pivote
  - Opción 1: En la tabla de pivotes, se puede ordenar cada fila para probar primero el mejor pivote de cada objeto (el más cercano o más lejano)
  - Opción 2: La tabla de pivotes se puede reducir a una sola columna, dejando sólo el mejor pivote por objeto
  - Reducir el espacio en memoria

Snake Table

- Criterio de selección: Seleccionar pivotes en forma dinámica, según se resuelven consultas
  - Utilizar como pivote el objeto de consulta previo



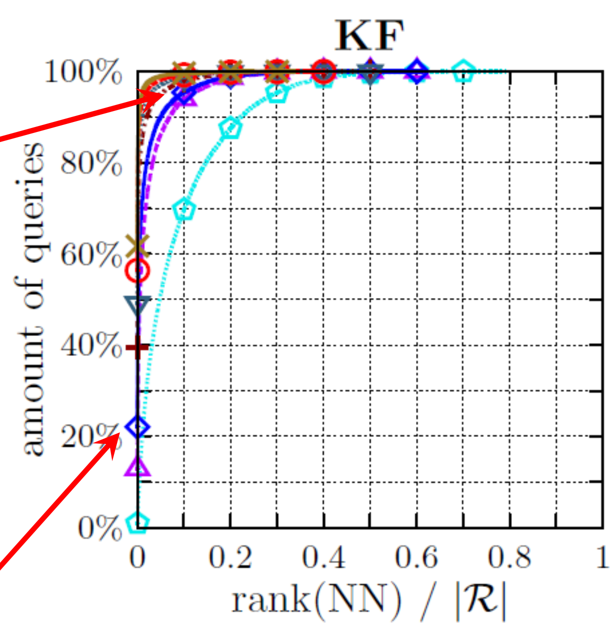
Búsqueda Aproximada con Pivotes

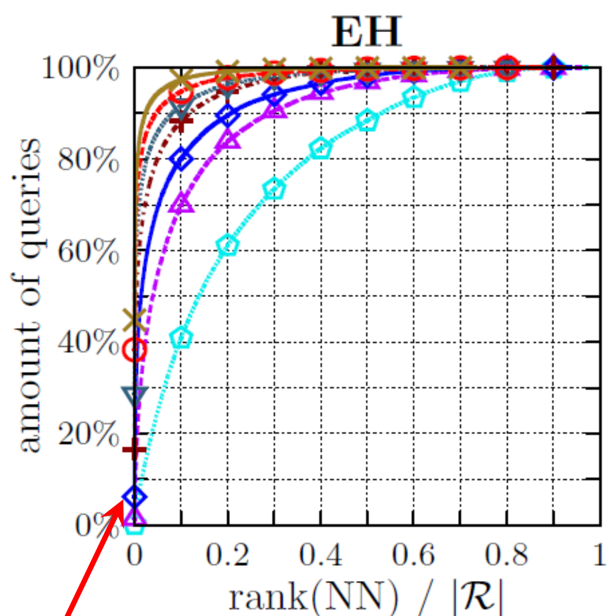
- Idea: La función LB_P puede ser usada como una estimación rápida de la distancia real
- Parámetro de aproximación T (entre 0 y 1):
  - Calcular LB_P para todos los objetos y seleccionar los *T%* menores valores
  - Calcular la búsqueda por rango o los *k*-NN solo entre esos los T% objetos seleccionados
    - La distancia real *d* se evalúa solo para un *T%* de los objetos y los restantes son descartados
    - Para que sea más rápido que la búsqueda lineal, el tiempo de evaluar LB_P debe ser al menos *T* veces más rápido que *d*
    - Requisito: Los objetos con menor *d(q,o)* deben tener un valor bajo de $LB_P(q,o)$

Búsqueda Aproximada con Pivotes

- Distribución del valor de LB_P para los vecinos más cercanos (efectividad):

Para este dataset usando 5 pivotes, en un 92% de las consultas el objeto que era el NN tuvo un valor de LB_P dentro del 10% de menor → Si usamos T=10 un 92% de las veces obtendremos el NN correcto





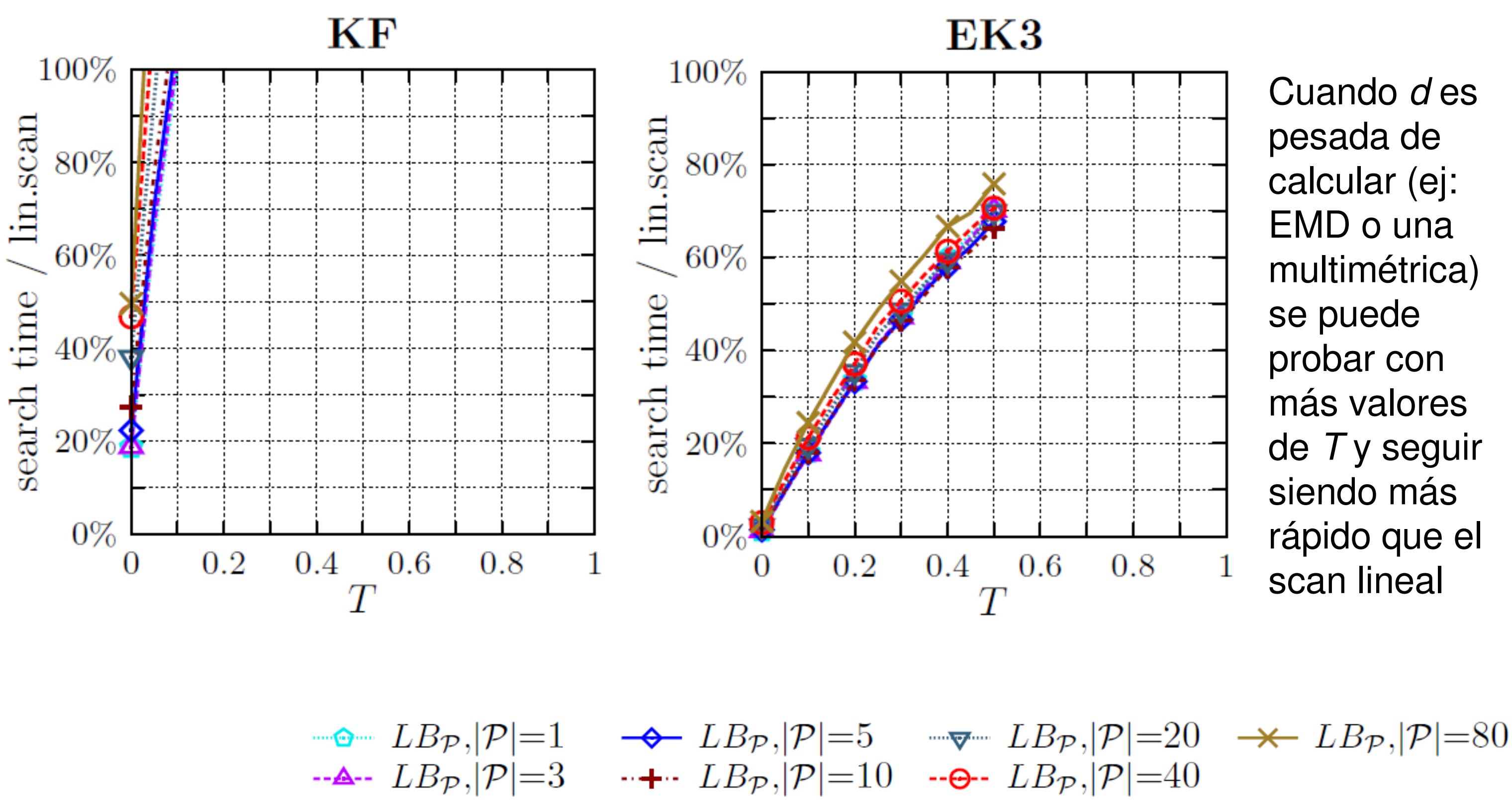
En un 21% de las consultas el objeto que era el NN fue también el de menor LB_P

En este dataset usando 5 pivotes, sólo un 6% de las veces el NN fue también el de menor LB_P y un 80% de las veces estuvo dentro del 10% menor

Búsqueda Aproximada con Pivotes

- Reducción de los tiempos de búsqueda:

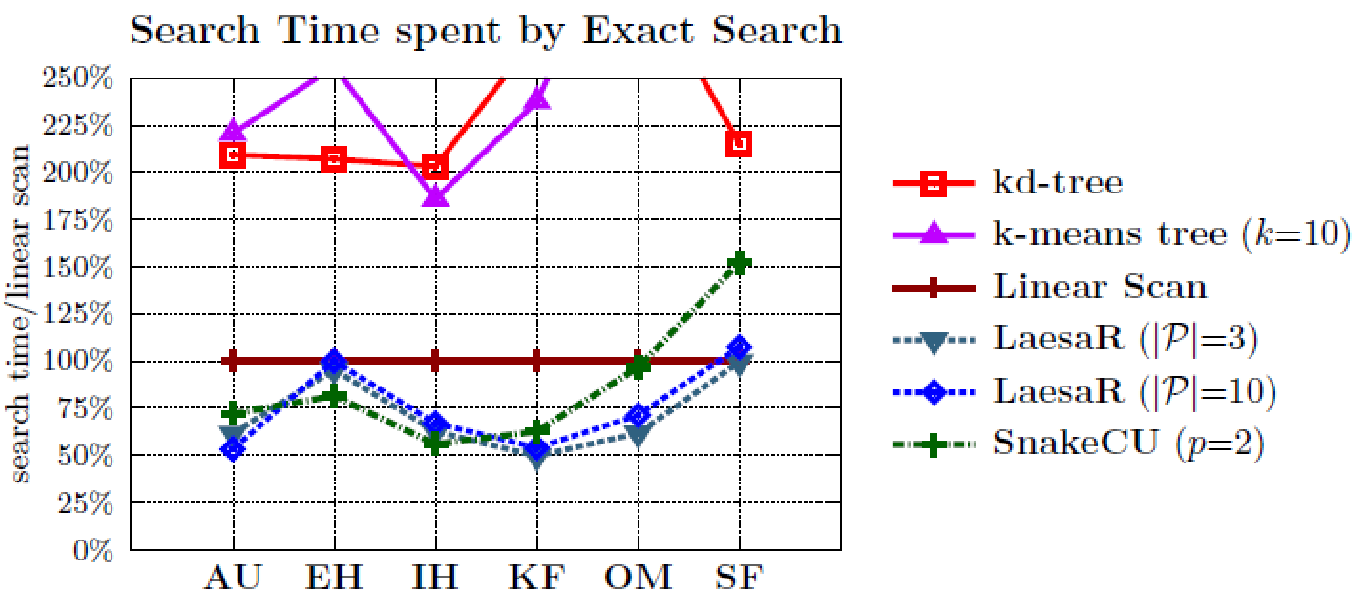
Cuando *d* es muy rápida de calcular (ej: L_1) el valor de *T* no puede ser muy alto si no la búsqueda aproximada se vuelve más lenta que el scan lineal



Cuando *d* es pesada de calcular (ej: EMD o una multimétrica) se puede probar con más valores de *T* y seguir siendo más rápido que el scan lineal

Indices Métricos vs Multidimensional

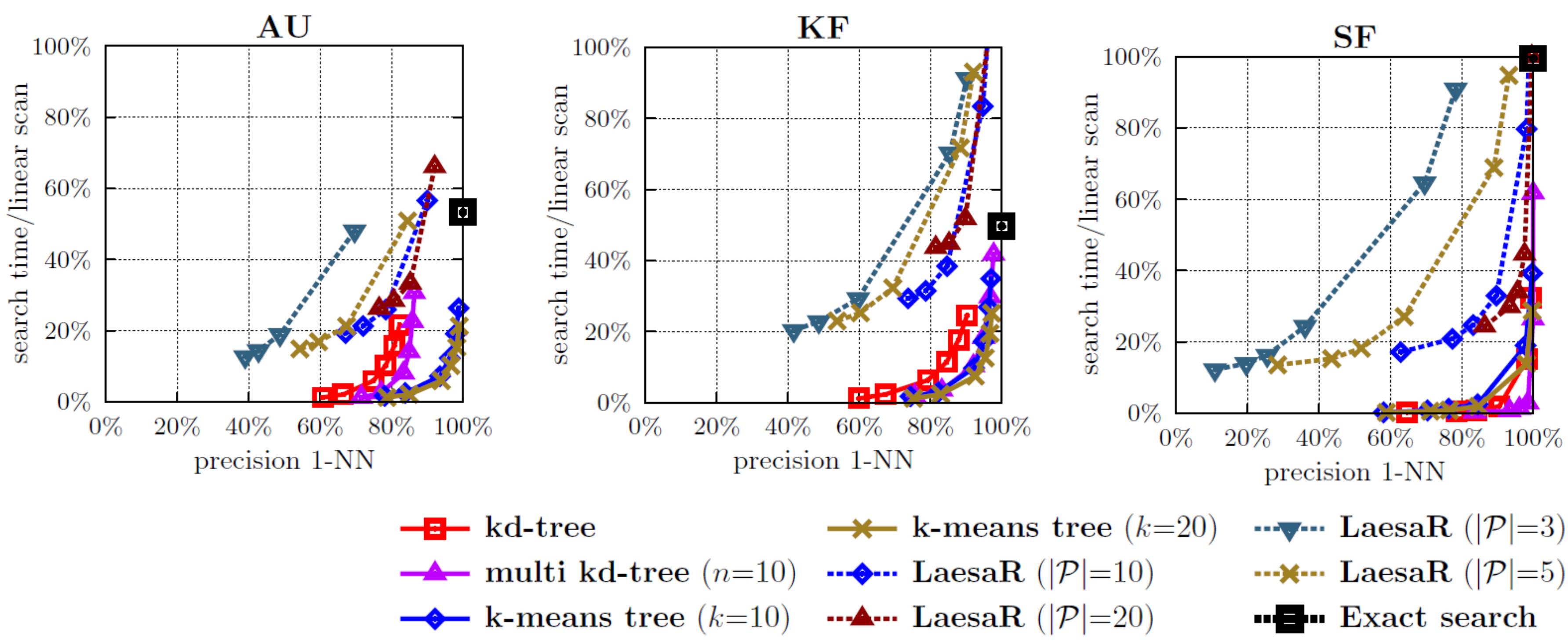
- Búsqueda Exacta:



Para búsqueda exacta, los índices métricos usualmente son más rápidos que la búsqueda lineal y los multidimensionales son usuamente más lentos

Indices Métricos vs Multidimensional

- Búsqueda Aproximada:



Para búsqueda aproximada, los índices multidimensionales logran un mucho mejor balance de efectividad vs tiempo de búsqueda

ESPACIOS MULTIMÉTRICOS

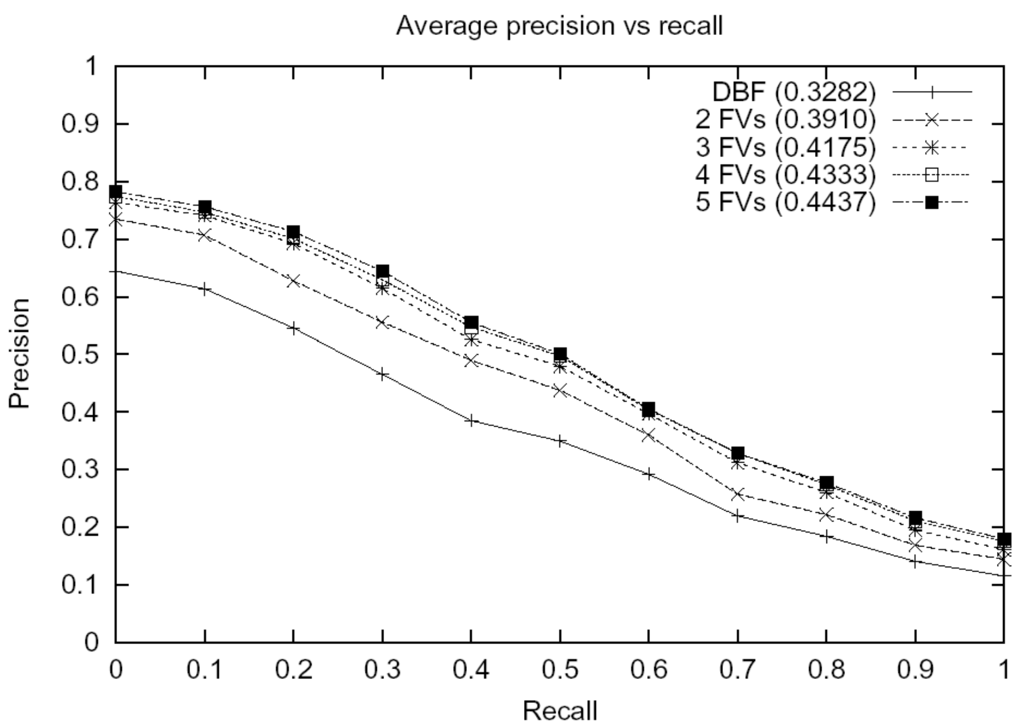
Combinación de distancias

- Sean δ_1, ... δ_m diferentes métricas para un mismo universo de objetos
  - Se puede definir una nueva función de distancia como la combinación lineal de las *m* métricas, es decir, sumando cada distancia multiplicada por un peso w_i:

$$\Delta(q,o) = \sum_{i=1}^{m} w_i \cdot \frac{\delta_i(q,o)}{normFactor_i}$$

Combinación de distancias

- Combinando más distancias (i.e. usando más descriptores) usualmente se mejora la calidad de la respuesta

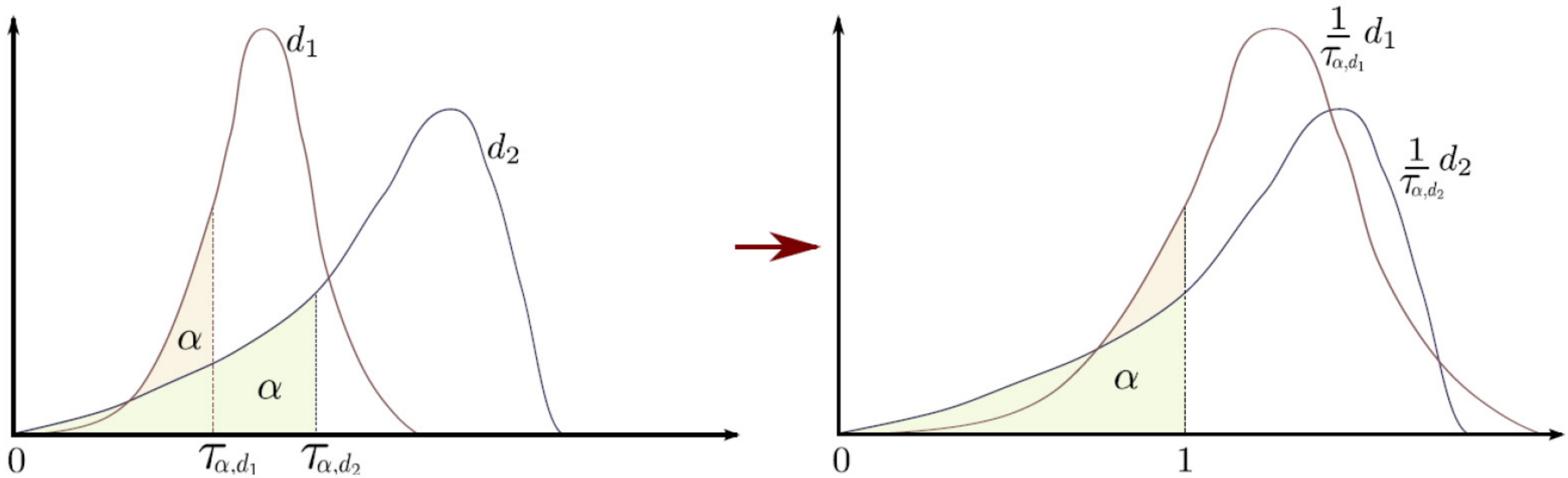


Combinación de distancias

- Al combinar funciones de distancia se construye una nueva función que (usualmente) logra mejor efectividad
- Pesos estáticos: función con pesos fijos
  - La distancia combinada también es métrica
  - Índices pueden indexar la distancia combinada
- Pesos dinámicos: función puede cambiar sus pesos dependiendo del objeto de consulta
  - Usualmente logra mejor efectividad que pesos fijos
  - La distancia combinada no es métrica
  - Se deben indexar las distancias por separado

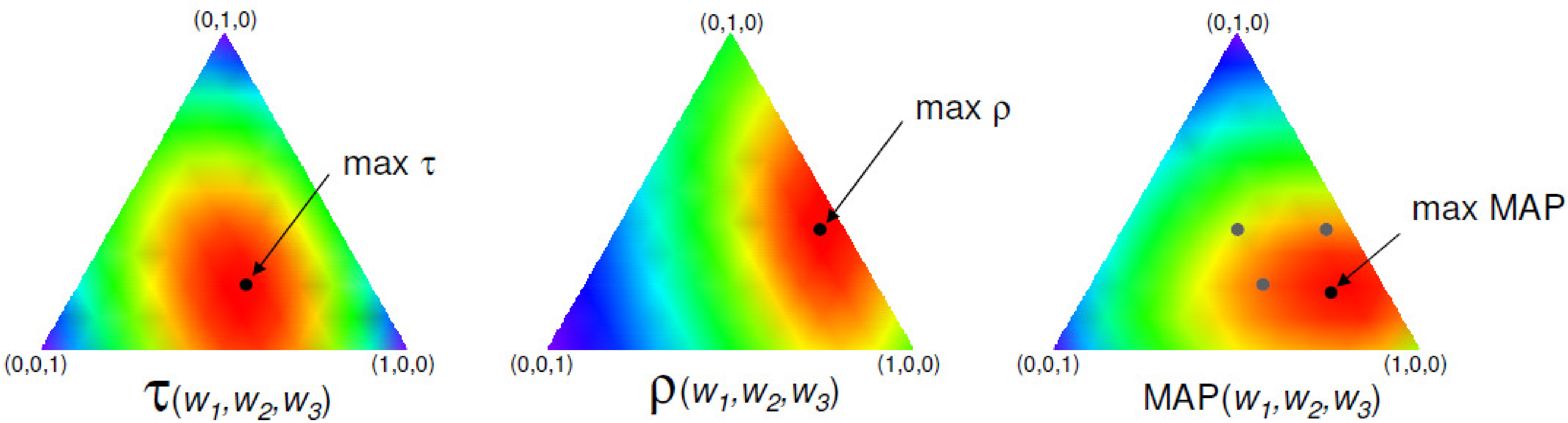
Normalización

- Normalizar por la distancia máxima o por una distancia de probabilidad α



Pesos estáticos

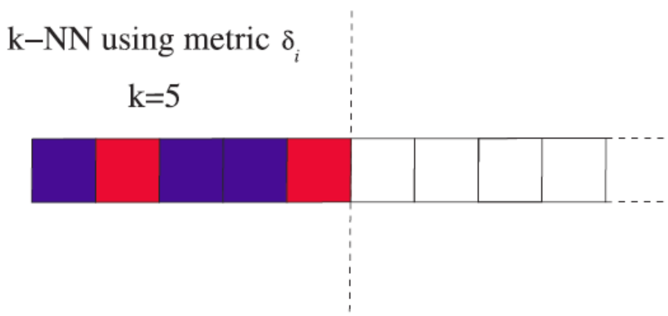
- Cálculo automático de pesos estáticos:



Pesos dinámicos

- Entropy Impurity

I. Perform k-NN in training dataset



Three objects belong to the blue class and two objects belong to the red class.

II. Entropy impurity

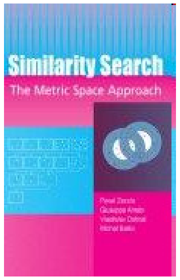
P_{ω_i} : fraction of objects that belong to model class i

$$entropy(\delta_i) = -\sum_{i=1}^{|\#classes|} \begin{cases} P_{\omega_i} \cdot \log_2(P_{\omega_i}) & \text{if } P_{\omega_i} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The entropy impurity of metric δ_i is equal to 0 if all objects belong to the same class, and has a maximum value ($\log(k)$) if each object belongs to a different class.

Bibliografía

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



Papers

- Chávez, Navarro, Marroquín, and Baeza-Yates. **Searching in metric spaces**. ACM Computing Surveys, 2001.
- Pedreira and Brisaboa. **Spatial Selection of Sparse Pivots for Similarity Search in Metric Spaces**. In SOFSEM, 2007.
- Bustos, Keim, Saupe, Schreck, and Vranic. **Automatic selection and combination of descriptors for effective 3D similarity search**. In ISMSE, 2004.
- Barrios, Bustos, and Skopal. **Analyzing and dynamically indexing the query set**. Information Systems, 2014.
- Barrios and Bustos. **Competitive content-based video copy detection using global descriptors**. Multimedia Tools and Applications, 2013.

Librerías

- Metric Space Library
  - http://www.sisap.org/metricspaceslibrary.html
- MetricKnn: Fast Similarity Search using the Metric Space Approach
  - https://juan.cl/metricknn_org/