APPROXIMATED DIRECT AND REVERSE NEAREST NEIGHBOR QUERIES, AND THE K-NEAREST NEIGHBOR GRAPH

Karina Figueroa
Universidad Michoacana
México

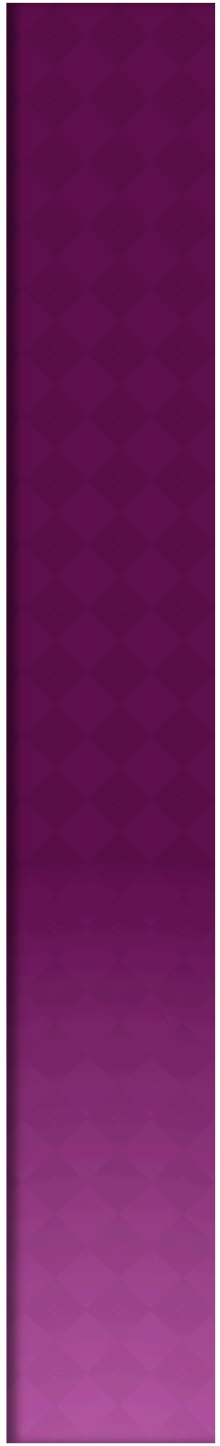
Rodrigo Paredes
Universidad de Talca
Chile

CONTENT

- Introduction (huge databases, similarity searching :)
- Definitions (metric spaces, distance :)
  - Nearest neighbor
  - Reverse nearest neighbor
- Permutation-based algorithm
- Approximated direct and reverse nearest neighbor
- k-nearest neighbor graph
- Experimental results
- Conclusions

INTRODUCTION

- Similarity searching
  - Retrieve alike objects
- Metric space
  - A database
  - Distance between objects
- Queries
  - Range queries
  - k-nearest neighbor queries
  - Reverse k-nearest neighbors



QUERIES

$\mathbb{U}$ database

- Range query

$$(q, r) = \{x \in \mathbb{U}, d(x, q) \leq r\}$$

- k-nearest neighbor

$$\forall\ x \in NN_k(q) \qquad |NN_k(q)| = k$$
$$y \in \mathbb{U} \setminus NN_k(q),\ \ d(q, x) \leq d(q, y)$$

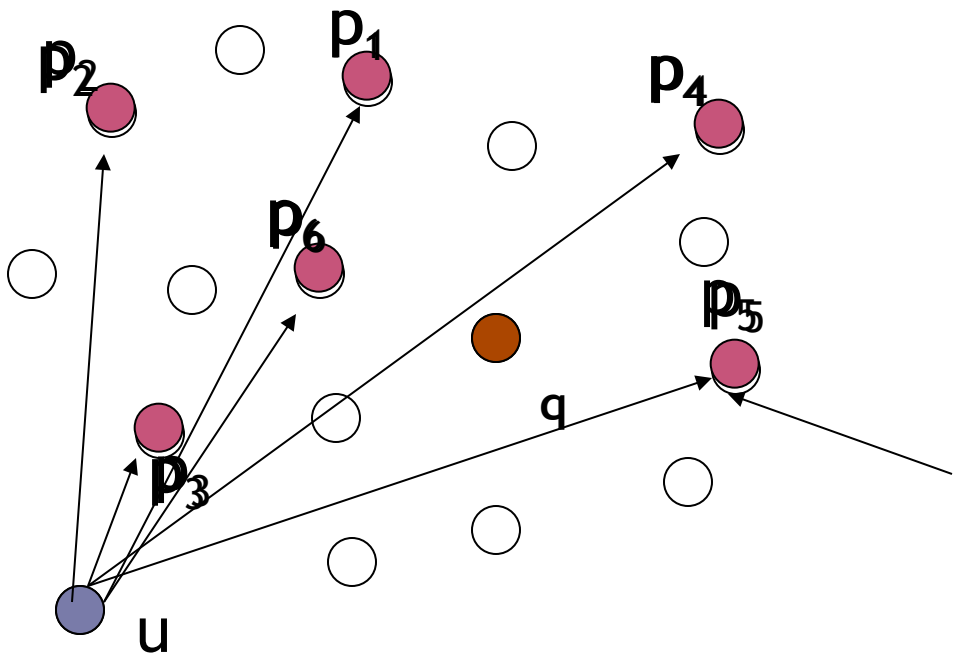
- Reverse k-nearest neigbor

$$\forall\ x \in RNN_k(q) \Rightarrow d(q, x) \leq cr_{x,k}$$

$cr_{q,k}$ Covering radius

PERMUTATIONS BASED ALGORITHMS

IDEA

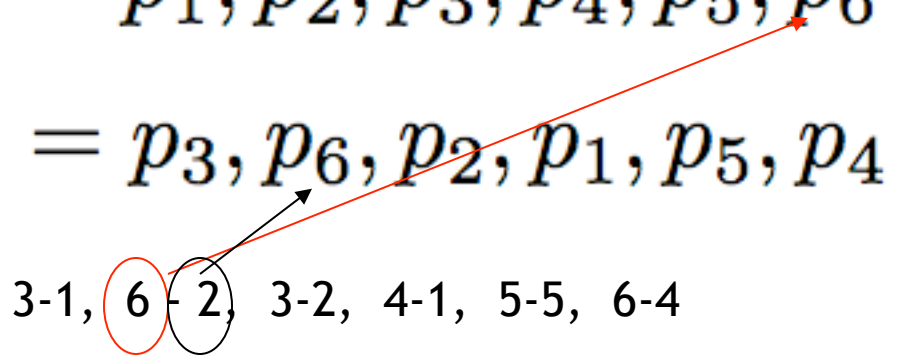


$$\Pi_u = p_3, p_6, p_2, p_1, p_5, p_4$$
$$\Pi_q = p_6, p_5, p_4, p_1, p_3, p_2$$

SPEARMAN FOOTRULE METRIC

$$\Pi_q = p_1, p_2, p_3, p_4, p_5, p_6$$

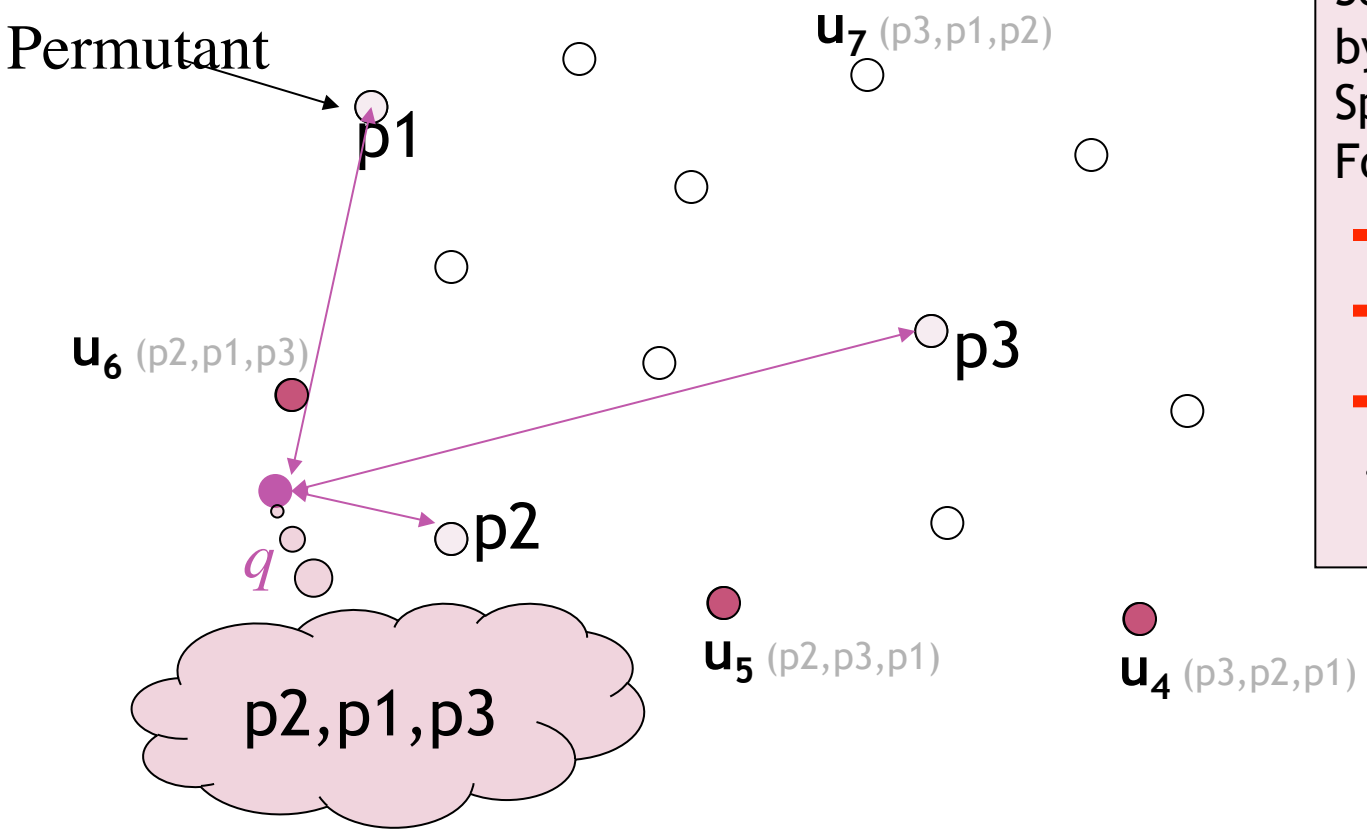
$$\Pi_u = p_3, p_6, p_2, p_1, p_5, p_4$$



Differences between positions

$$F(\Pi_q, \Pi_u) = |1-3| + |2-6| + |3-2| + |4-1| + |5-5| + |6-4| = 12$$

SEARCH PROCESS

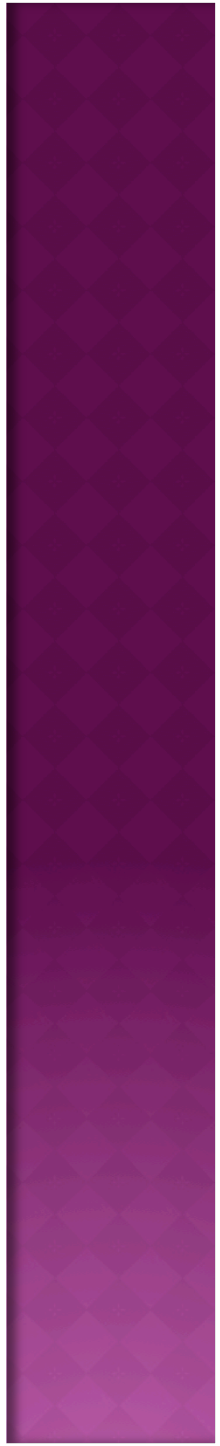


Sorting elements by Spearman Footrule

.....

OUR PROPOSAL

- Use an alternative search order, to solve the k-nearest neighbor
  - It is cheaper to compute
- Use our approximate k-nearest neighbor queries to construct
  - Direct k-nearest neighbor (knn)
  - Reverse k-nearest neighbor (rknn)
  - K-nearest neighbor graph (knng)
  - Dynamic knng



DOMINATING ORDER

- **Definition.** Order $\leq_s$ dominates order $\leq_t$ at level K/k
  - if the first K elements in order $\leq_s$ contain the first k elements in order $\leq_t$

- We use $\leq_{\pi q}$ as an almost dominate order for $\leq_q$

APPROXIMATED DIRECT *k*-NEAREST NEIGHBOR

- To solve the knn
  - Compute the permutation of the query
  - Select the first $|C|$ objects in order $\leq_{\pi q}$
    - Order the database according to and select the first |C| elements as candidates.
  - Select the k-nearest neighbors among the |C| possible candidates.

APPROXIMATED REVERSE *k*-NEAREST NEIGHBOR

- **To Solve R-knn**
  - Compute the permutation of the query Π_q
  - Select the first $|C|$ objects in the order $\leq_{\Pi q}$
  - For each object *u* selected
    - Compute d(q,*u*)
    - Check if *q* is within $NN_k(u)$.

THE *k*-NEAREST NEIGHBOR GRAPH

- Build the knn Graph
  - We solve a k-nearest neighbor query for each element in the database
- Handle dynamic a graph. Problems to solve
  - Deletions
    - Delete the object
    - Find its k-nearest neighbor, notify.
  - Insertions
    - Compute the new permutation
    - Compute the k-nearest neighbor
    - Compute the reverse k-nearest neighbor

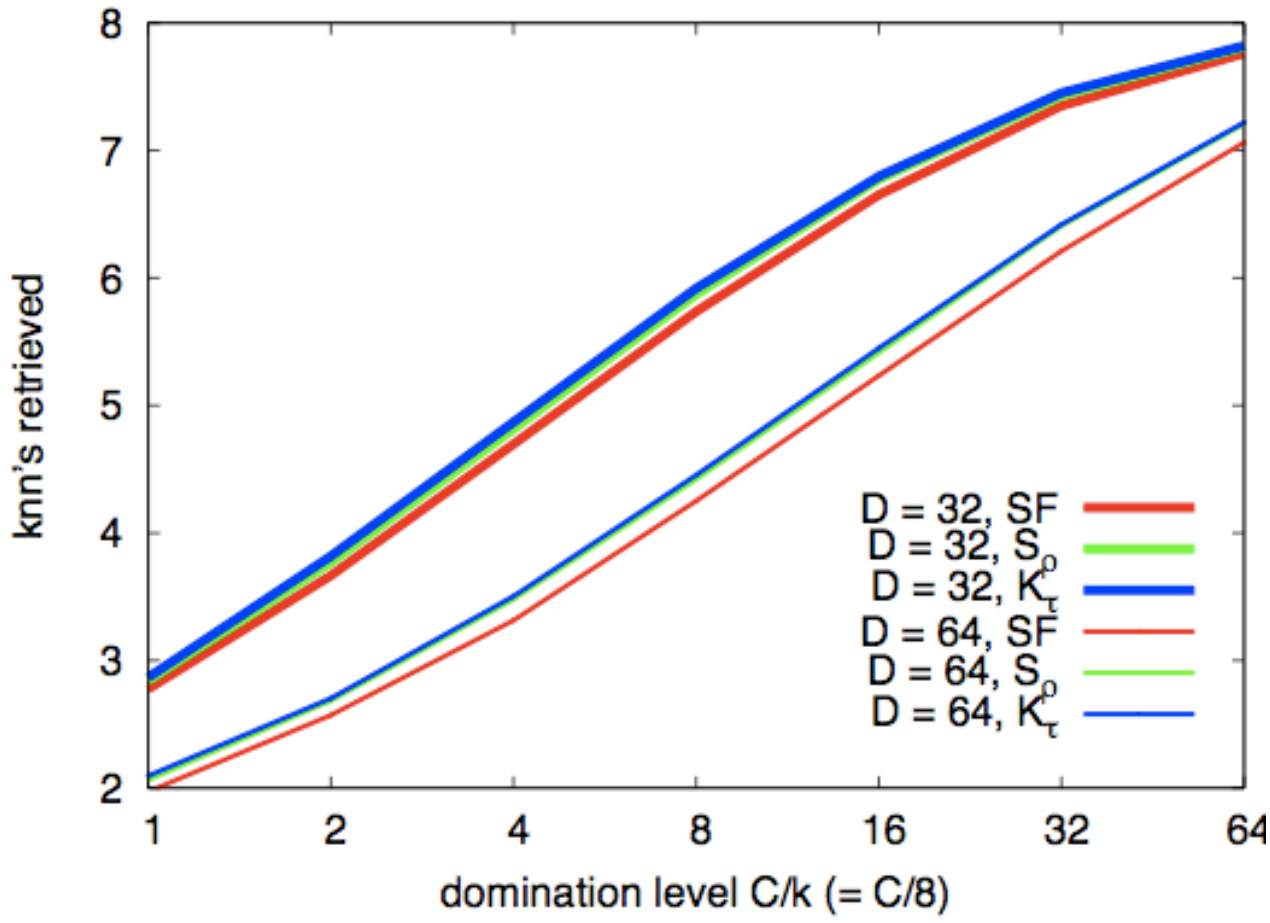
DATABASES

- Synthetic vectors in an unitary cube
  - Dimension 32 and 64
  - 10,000 objects
- Faces (2,152 dimensions)
  - CAS-PEAL
  - Bio-ID
  - Karade
  - 20,000 objects

EXPERIMENTAL RESULTS

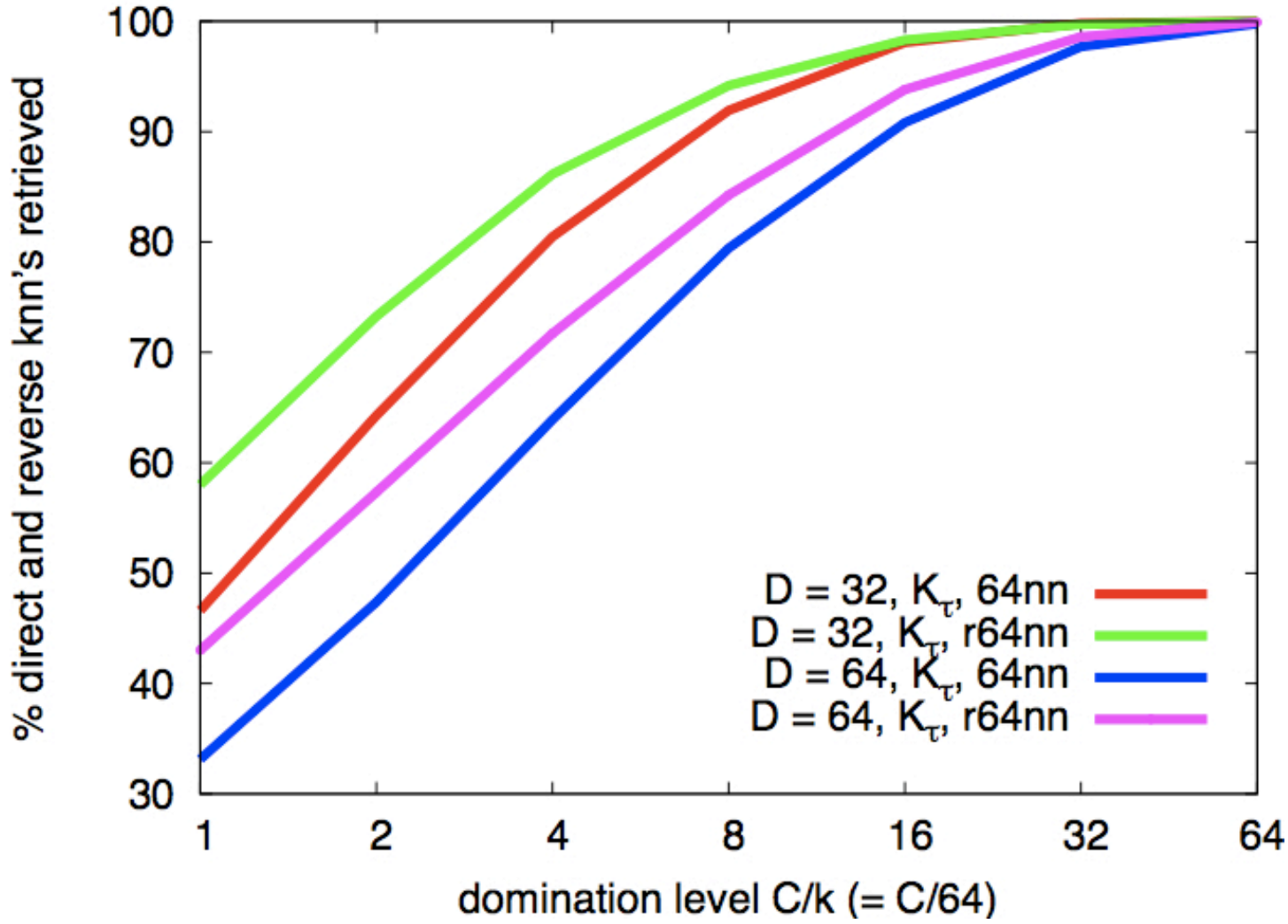
METRIC BETWEEN PERMUTATIONS

Unitary Cube, 8NN, 5000 objects, 64 anchors



DIRECT AND REVERSE KNN

Unitary Cube, 64NN, 64 rNN, 10,000 objects, 128 anchors



KNN GRAPH.

UNITARY CUBE

Table I

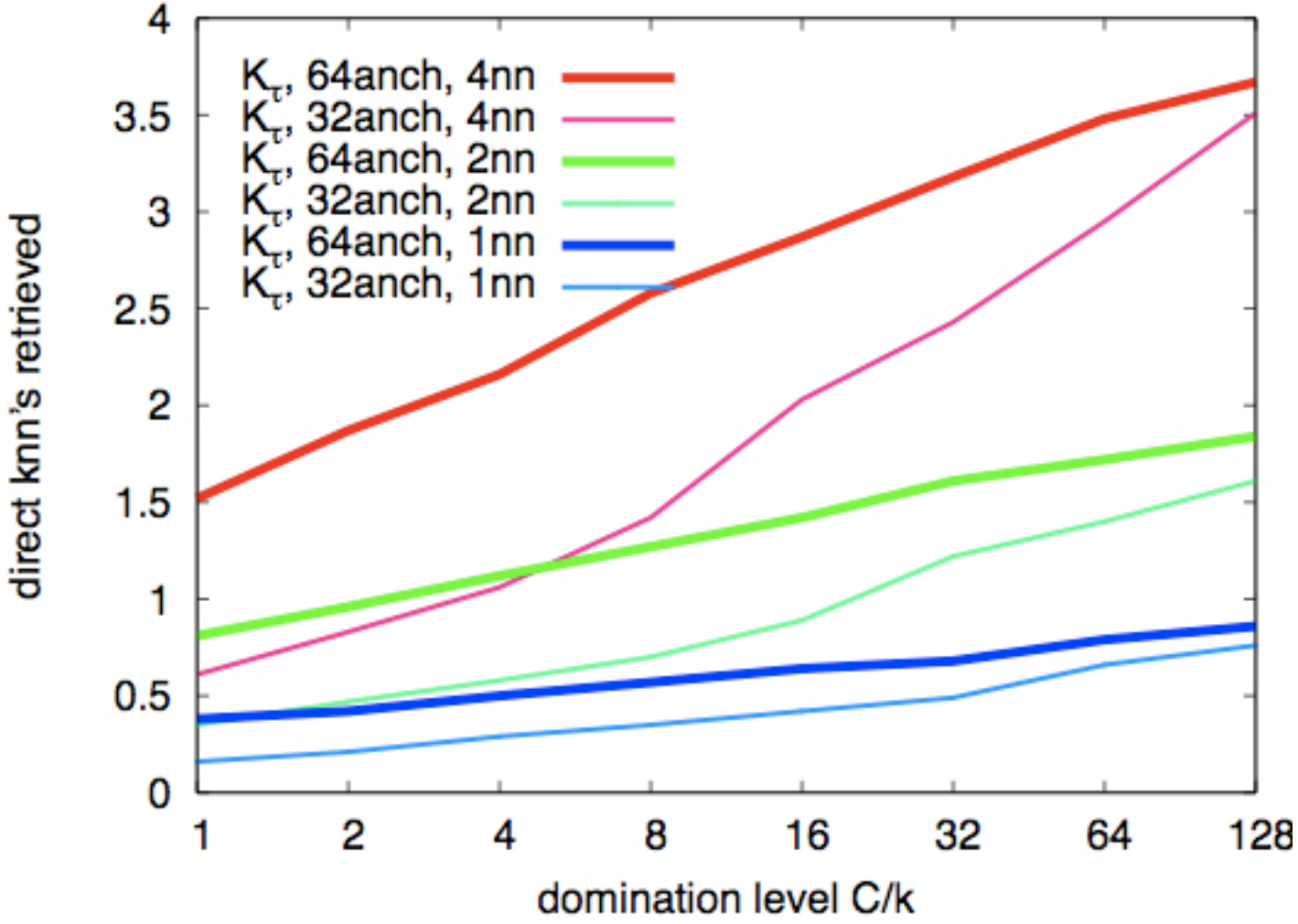
QUALITY PERFORMANCE IN THE CONSTRUCTION OF kNNGS

20,000 objects, 128 anchors

D	$k = 8$, $C/k = 32$		$k = 64$, $C/k = 16$	
	% kNN	cr ratio	% kNN	cr ratio
32	98.1	1.035	98.8	1.052
64	91.1	1.023	92.1	1.083

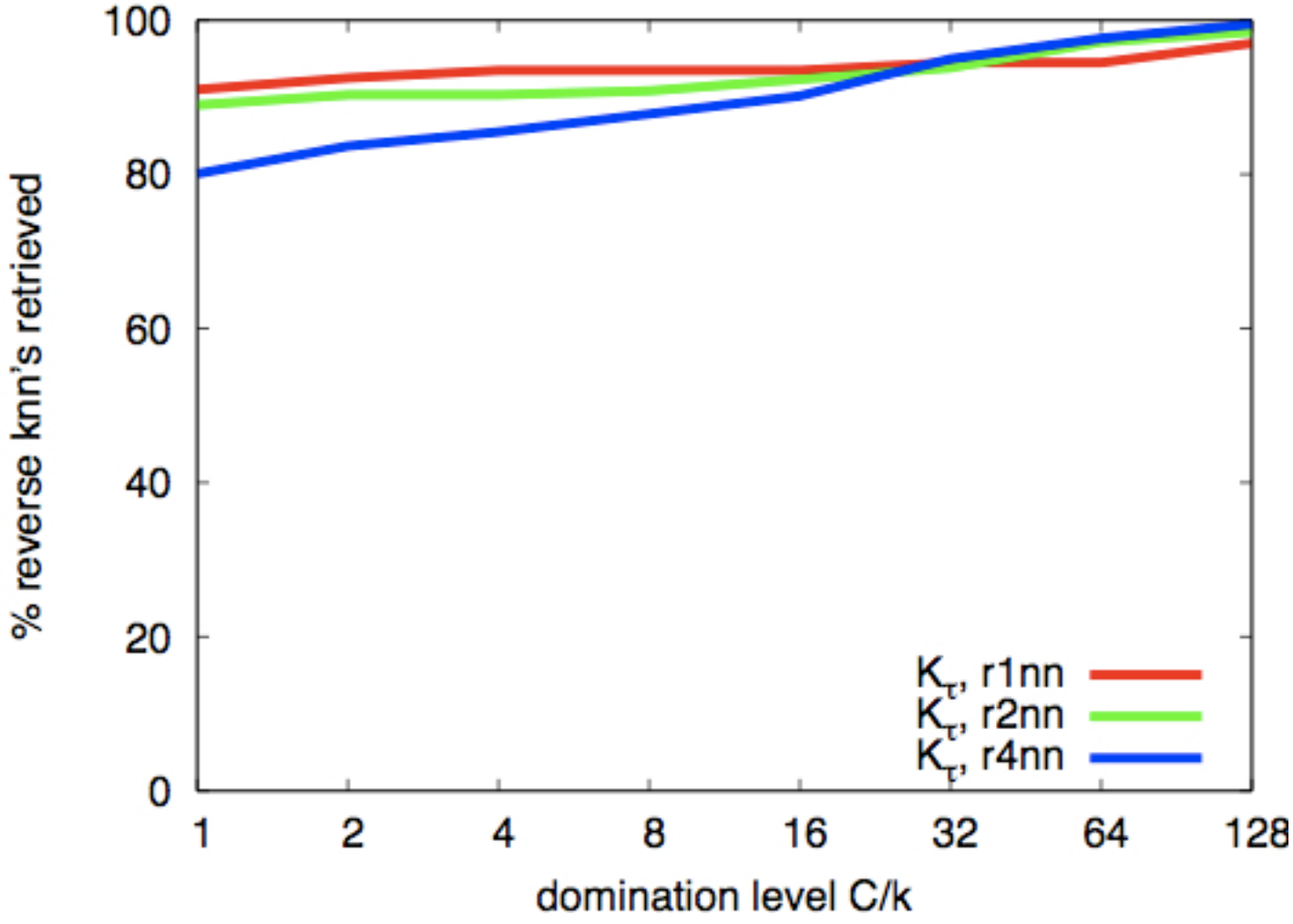
FACES IMAGES

20,000 objects, with 2152 dimension, 32 and 64 anchors



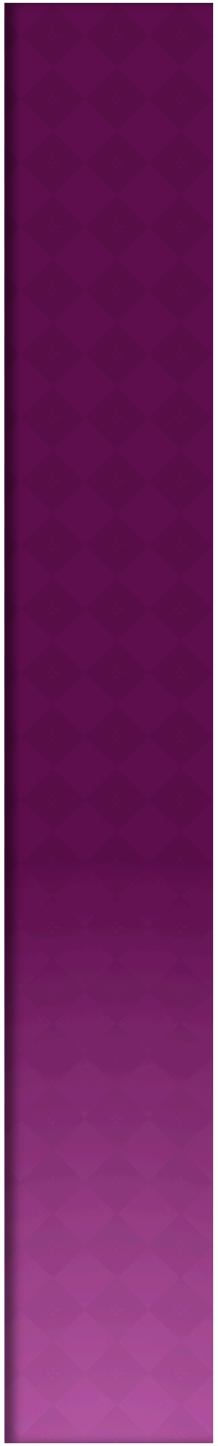
FACES IMAGES

20,000 objects, with 2152 dimension, 64 anchors



CONCLUSIONS

- We presented a new approximated approach to solve direct and the reverse knn problems.
- Our idea is to use an alternative order, which is cheaper to compute, yet it yields a rather similar sequence.



DĚKUJI (THANK YOU)

karina@umich.mx

raparede@dcc.uchile.cl