Efficient Similarity Search by Reducing I/O with Compressed Sketches

Arnoldo Müller-Molina and Takeshi Shinohara

Department of Artificial Intelligence
Kyushu Institute of Technology (Iizuka, Japan)

SISAP 2009

Table of Contents

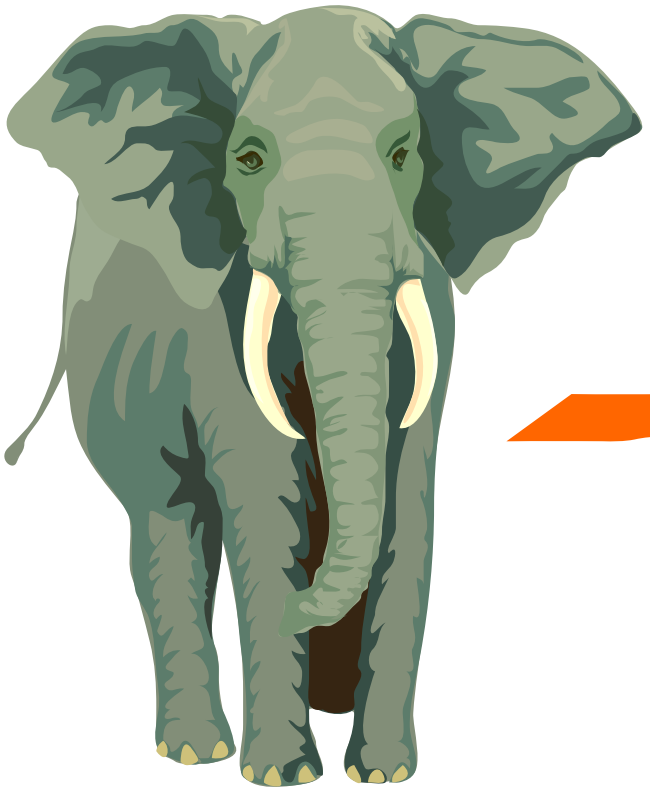
Outline

Sequential Search

- Dimensionality increases:
  - For certain distributions and queries
    - Hierarchical indexes performance decreases
    - Sequential search is better
- Sequential Indexes:
  - VA-file [Weber 1998], IQ-Tree [Berchtold 2000]
  - LCluster [Chávez *et al.* 2005]
  - Distance Permutations [Chávez *et al.* 2008]
  - Sketches [Lv 2004] [Wang 2007] [Dong 2008]

What is a Sketch?

Object $\longrightarrow$ binary string



10101...01

Sketches: cheap sequential search

Compact representations, cheap distance estimators

- Same sketch holds similar objects
- Hamming distance
  - Native in hardware: XOR + bit pop. count
- Bucket access order determined by hamming distance
- Introduced by Lv, Charikar and Li in 2004
  - Only for L_2, L_1 spaces

Contributions of the Paper

- Sketches for general metric spaces
  - Simple mapping, pivot selection strategy
- Speedup over AESA: up to 10x
- Speedup over Slim Tree: 100x - 1000x
- Sketch compression is possible
  - Up to 1000x smaller than original data

Outline

Proposed Sketch Definition

Generalized Hyperplane Sketch (GHS)

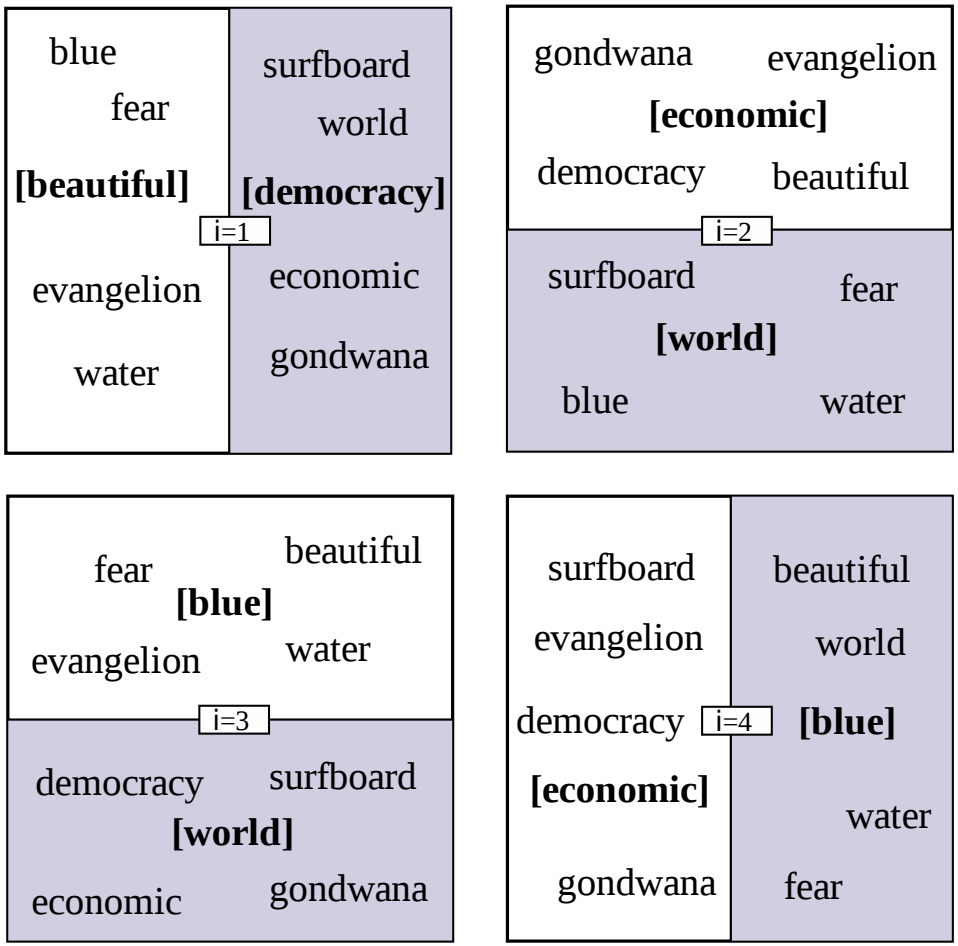
Sketch for object $x \in \mathcal{D}$, is a bit vector $\sigma(x) \in \{0,1\}^m$, where each bit $\sigma_i(x)$ is:

$$\sigma_i(x) = \begin{cases} 0 & \text{if } d(p_{i0}, x) \leq d(p_{i1}, x) \\ 1 & \text{if } d(p_{i0}, x) > d(p_{i1}, x) \end{cases} \quad \forall i = 1, 2, \ldots, m$$

where p_{i0}, $p_{i1} \in \mathcal{D}$ are pivots.

Example

Partitioning in string (Levenshtein) spaces



- “fear” : “0101”

Pivot selection algorithm (rf01)

Prefer balanced partitions

$P, Q \subseteq \mathcal{D}$: pivot sets
Returns true if P is better than Q, otherwise false.

1: **function** $rf01(P = \{p_0, p_1\}, Q = \{q_0, q_1\})$
▷ Get the difference of partition sizes:
2: $st_p \leftarrow ||S_{p0}| - |S_{p1}||$
3: $st_q \leftarrow ||S_{q0}| - |S_{q1}||$
4: **if** $st_p = st_q$ **then** ▷ Equally balanced partitions
▷ Greater inter pivot distance is better
5: return $d(p_0, p_1) > d(q_0, q_1)$
6: **else**
7: return $st_p < st_q$ ▷ S better balanced with P?
8: **end if**
9: **end function**

k-NN Search

Find j closest sketches, search those buckets

- Filter and refine:
  - Find the closest j sketches
  - Search each bucket (sketch associated with bucket)
- How to find j?
  - In this paper: sampling (see annex)
  - Dong *et al.*: $k \times 20$

Compression

Sketches can be efficiently compressed

- Sketches are positive integers
- Inverted index compression:
  - *d*-gaps, Gamma and Delta run length encodings
- Bitmap index compression:
  - Word aligned hybrid (WAH)

Outline

Experiments

Compression and Performance

- Performance:
  - Improvement Efficiency *IE* (speedup)
  - *IE* over Slim-Tree [Traina 2000] and AESA [Vidal 1986]
  - Comparison against L_2 sketch[Dong 2008], distance permutations [Chávez *et al.* 2008]
- Compression (4 compression methods):
  - GAMMA (*d*-gaps)
  - DELTA (*d*-gaps)
  - Bitmap
  - Word Aligned Hybrid (WAH)

Evaluation

EP error position and different *IE*

- Error position:

$$EP = \frac{\sum_{i=1}^{|S^A|}(OX(o_i^A) - S^A(o_i^A))}{|S^A| \times |X|}.$$

- Improvement in efficiency *IE* (over Slim-Tree or AESA):
  - IE_{acc}: disk access count.
  - IE_{obj}: objects read from secondary storage.
  - IE_{dist}: distance computations

Datasets

SISAP datasets and a synthetic dataset

Table: Summary of datasets.

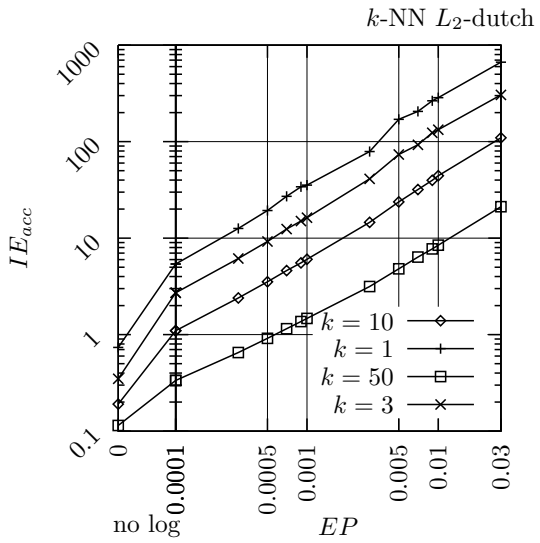
Dataset	DB size	Size	m
dutch	200000	2MB	64
dict	800000	8MB	64
trees	100000	5MB	64
trees-full	300000	17MB	64
vectors	1 billion	223GB	30

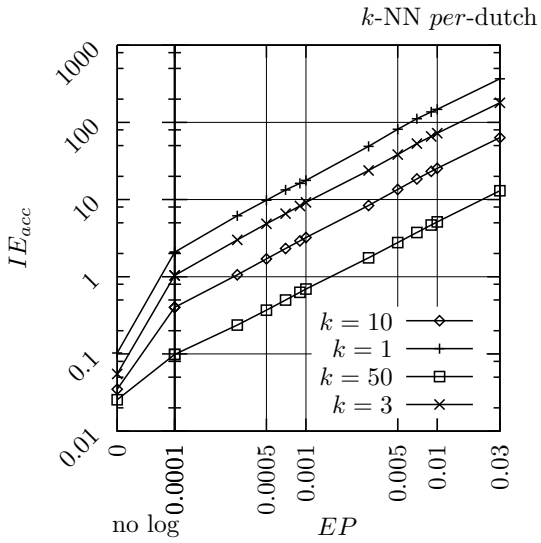
Compressing sketches

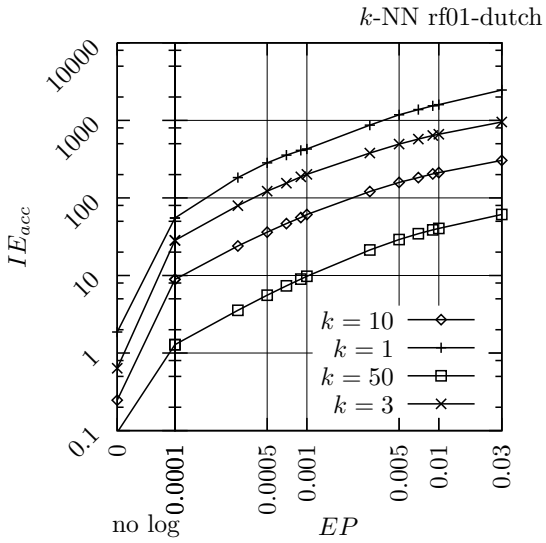
Table: Compressed sketch size, search time

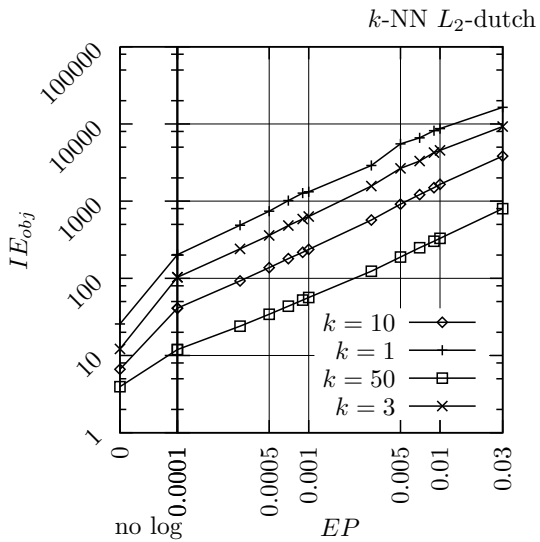
Data Set	Method	Size	Milliseconds
dict(2.4MB)	BitSet	134MB	151.73
	WAH	4MB	67.79
	Delta	1.1MB	23.97
	Gamma	1.2MB	28.67
trees-full(187Kb)	BitSet	134MB	151.73
	WAH	264Kb	5
	Delta	79Kb	2
	Gamma	92Kb	.72
vectors(1.7GB)	BitSet	134MB	14174
	WAH	131MB	20917
	Delta	251MB	10534
	Gamma	204MB	10890

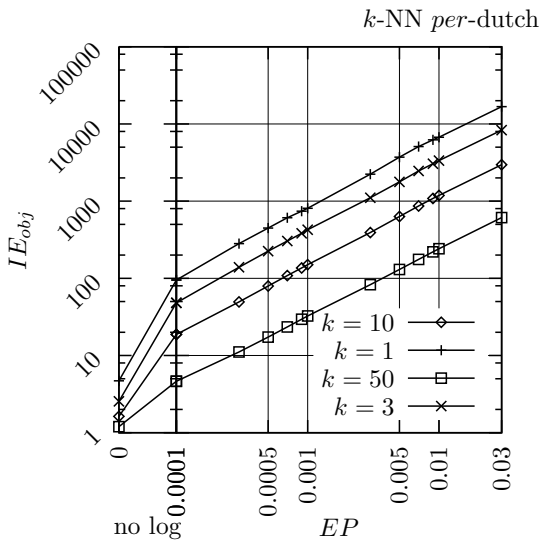
rf01 *IE* over Slim Tree (Dutch)

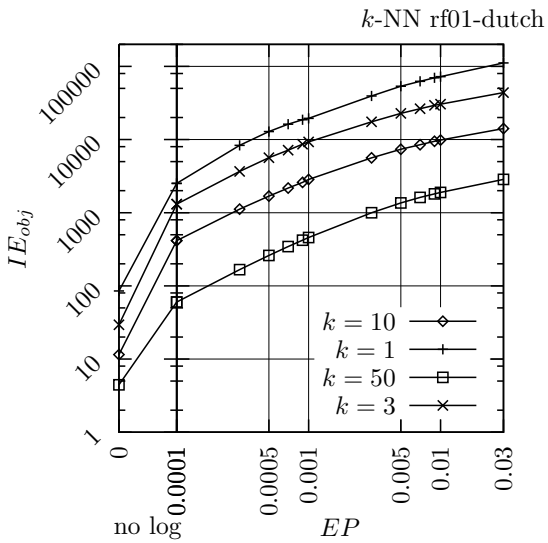




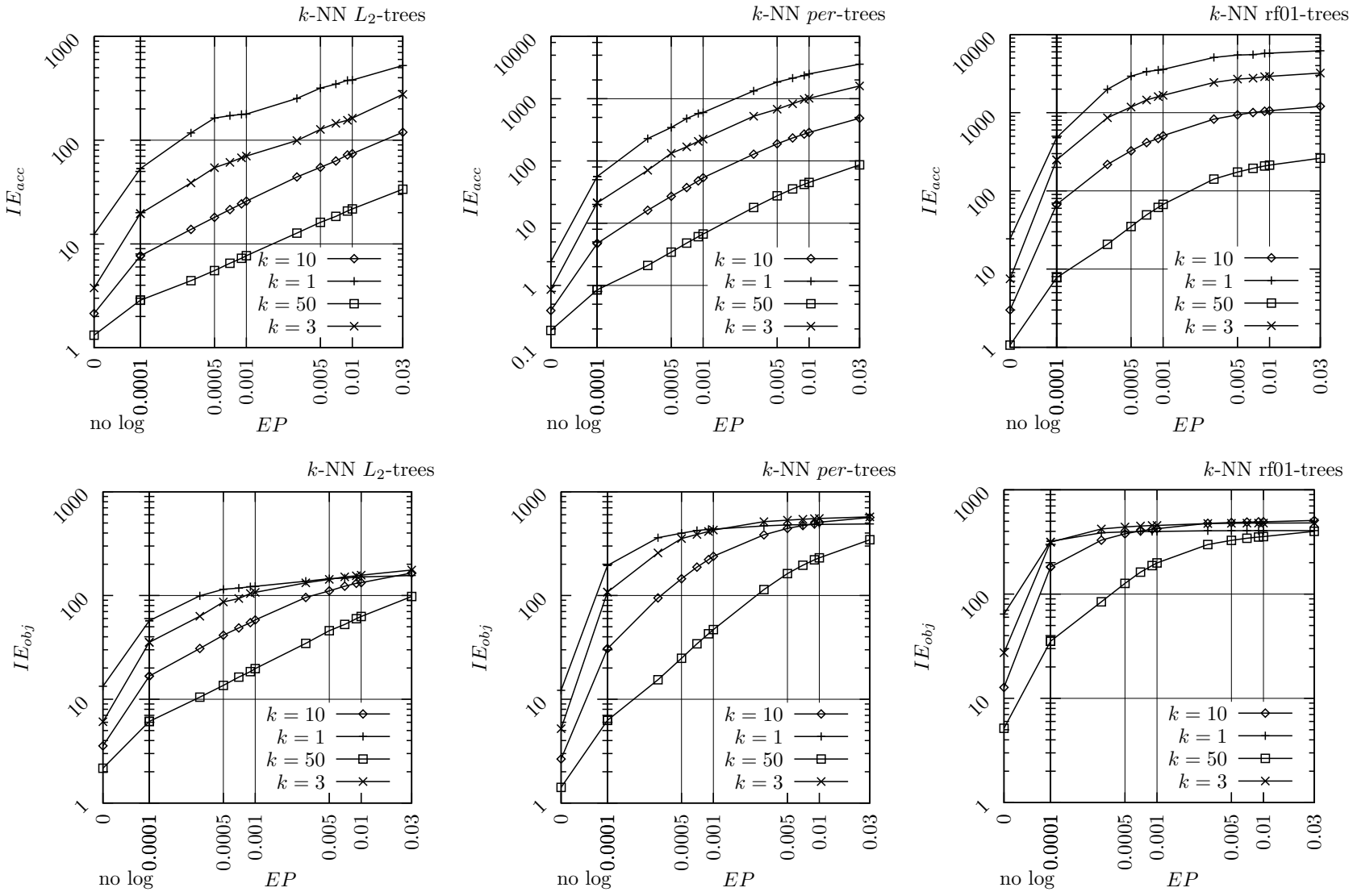






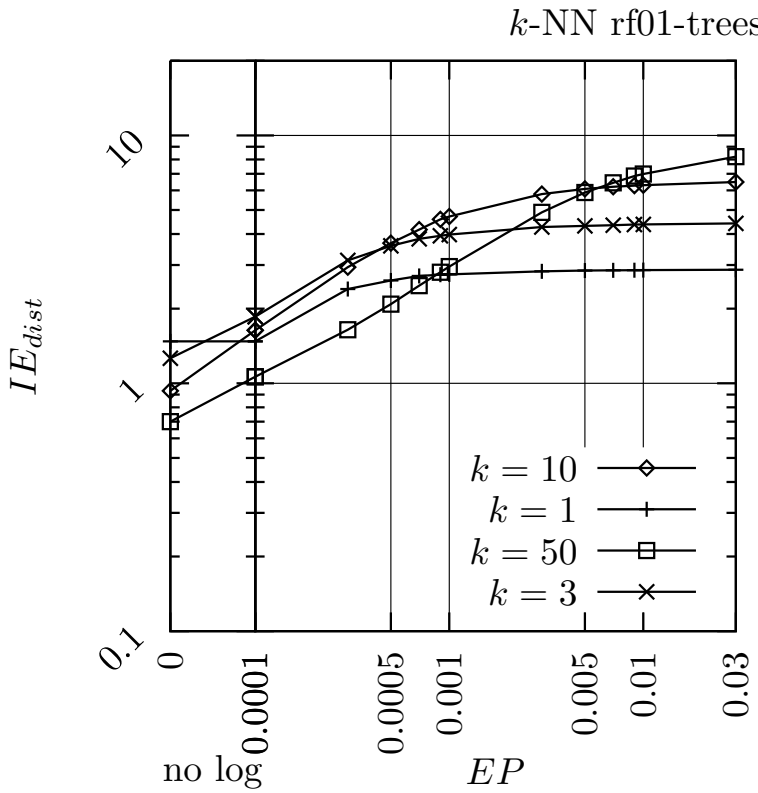


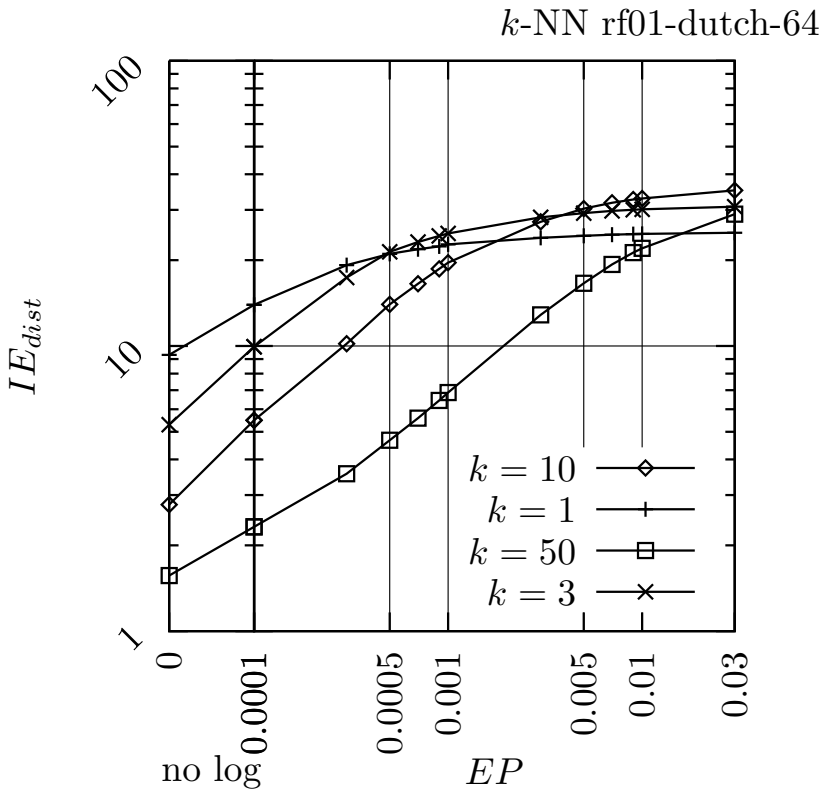
rf01 *IE* over Slim Tree (Trees)



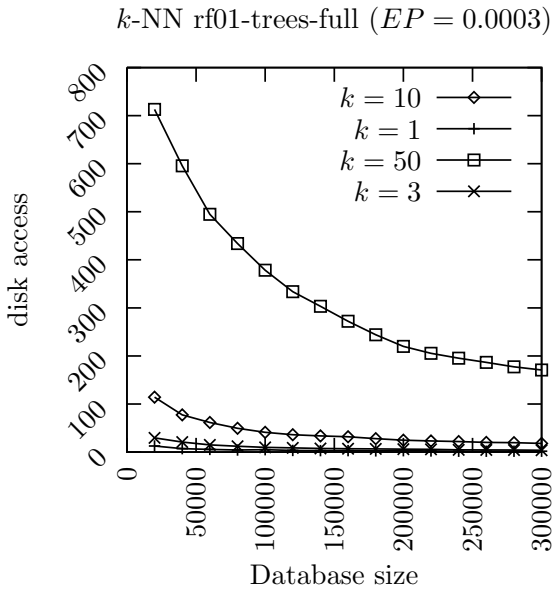
Our Technique (rf01) *IE* over AESA

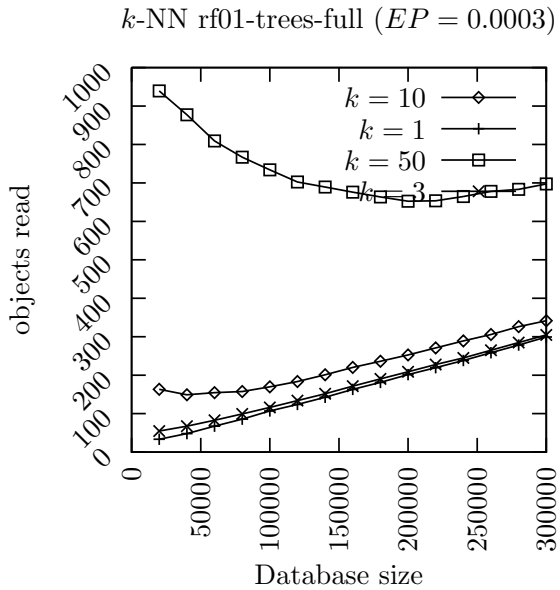
Datasets: trees, dutch

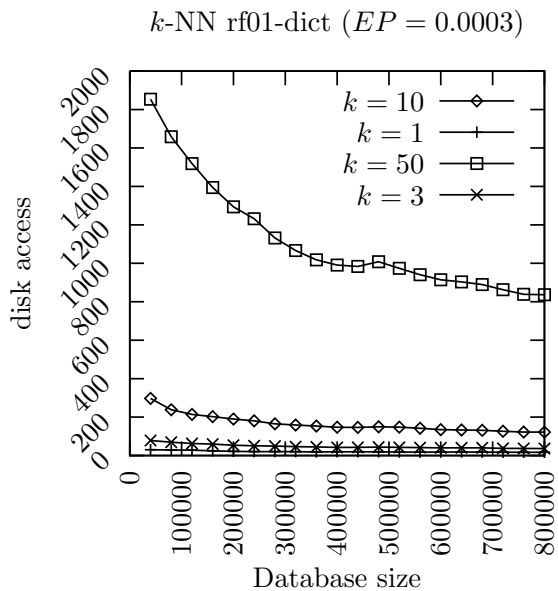


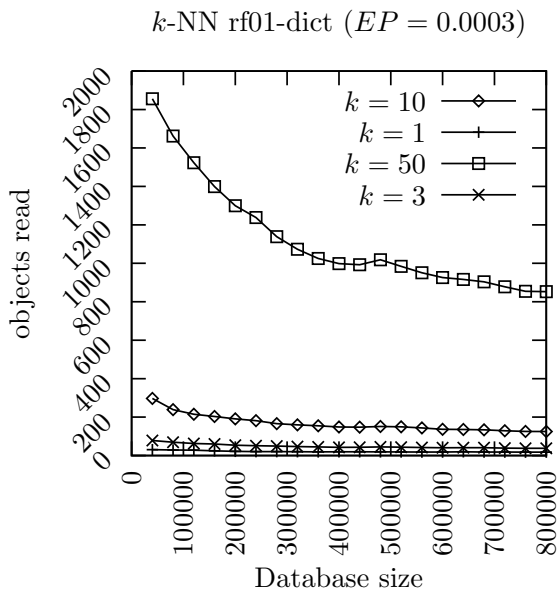


Data Growth (trees, dutch)









Conclusions

- Up to 10x improvement over:
  - L_2 sketch[Dong 2008]
  - Distance permutations [Chávez *et al.* 2008]
- Up to 10x improvement over AESA [Vidal 1986]
- 10x-1000x over Slim-Tree [Traina 2000]
- Compression: 10x - 1000x smaller than original data