Speeding up Similarity Search by Sketches

Vladimir Mic   David Novak   Pavel Zezula

Faculty of Informatics
Masaryk University
Brno, Czech Republic

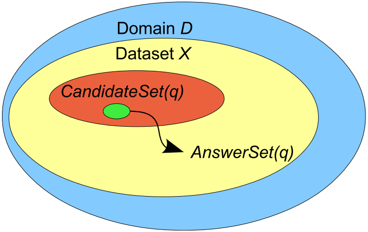
October 26, 2016

Motivation and Context

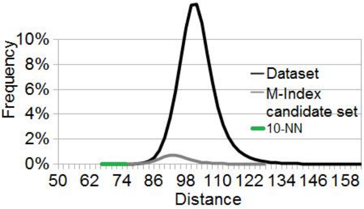
- Similarity search in Metric space (D, d)
  - Dataset $X \subseteq D$

- Techniques (*indexes*) for similarity search usually:
  - decompose dataset X into disjoint partitions $P_1, \ldots, P_z$
  - for query object $q \in D$ evaluate similarity query:
    - determine partitions which likely contain similar objects
    - union these partitions into $CandidateSet(q)$
    - for each object $o \in CandidateSet(q)$ evaluate distance $d(q, o)$ to determine $AnswerSet(q) \subseteq CandidateSet(q)$ (refinement)

Observation



(a) Relationship of D, X, CandidateSet(q) and AnswerSet(q)



(b) Distance densities of X and CandidateSet(q_1) to selected query q_1

- Beside similar objects, $CandidateSet(q)$ usually contains many dissimilar objects as well (especially in complex metric spaces)
- The size of the candidate set determines the query processing time:
  - number of evaluations of function d,
  - I/O cost if dataset X is stored on hard-drives

Objectives and Approach

- We propose to add a small piece of information to each object to
  - significantly reduce the size of $CandidateSet(q)$
  - but preserve objects o from $AnswerSet(q)$

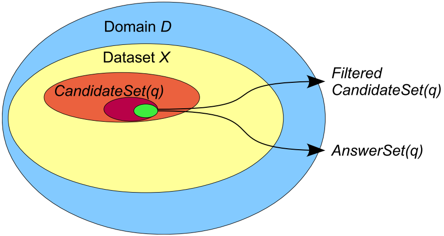


Figure: Filtered candidate set with almost preserved answer set

- The additional information is represented by an object *sketch*

Sketch (of object $o \in D$)

- Short bit string representation of object o

$Sk(o)$:

1	0	1	1	0	0	0	0

- Distance of two sketches is measured by Hamming distance h
- Each bit value is set by partitioning of the domain D into two parts using *Generalized hyperplane partitioning* (GHP)

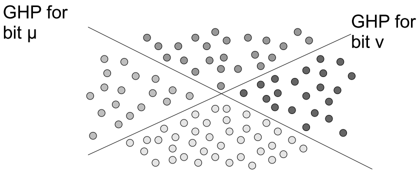


Figure: Two instances of GHP used to set bits μ and ν

Sketch (of object $o \in D$)

- Short bit string representation of object o

$Sk(o)$:

1	0	1	1	0	0	0	0

- Distance of two sketches is measured by Hamming distance h
- Each bit value is set by partitioning of the domain D into two parts using *Generalized hyperplane partitioning* (GHP)

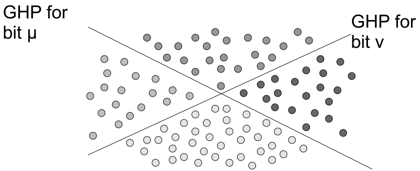


Figure: Two instances of GHP used to set bits μ and ν

- Sketch approximately determines object position in the metric space
- Ideally it would preserve an object ordering with respect to an arbitrary query q: $\forall q \in D : \forall o_1, o_2 \in X$:
$d(q, o_1) < d(q, o_2) \implies h(Sk(q), Sk(o_1)) < h(Sk(q), Sk(o_2))$

How to Produce Good Sketches?

The following properties improve the sketch ability to approximate the ordering on a given dataset X:

- sketches should reflect spatial relationships between objects (achieved by GHP of dataset)
- each bit of sketches should be set to 1 in one half of sketches (balanced bits)

$Sk(o_1)$:	1	0	1	1	0	0	1	1
$Sk(o_2)$:	1	0	0	0	1	1	0	1
$Sk(o_3)$:	0	1	0	1	1	0	0	0
$Sk(o_n)$:	0	1	1	0	0	1	1	0

Example: sketches of four objects with balanced bits

- sketches should have low correlated bits

Candidate Set Filtering with Sketches

- Proposed approach: shrink *CandidateSet*(q) using the Hamming distances of corresponding sketches

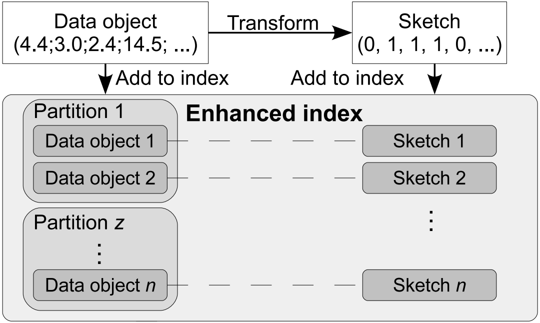


Figure: General concept of index enhancement with sketches

Candidate Set Filtering with Sketches

- Proposed approach: shrink *CandidateSet*(q) using the Hamming distances of corresponding sketches

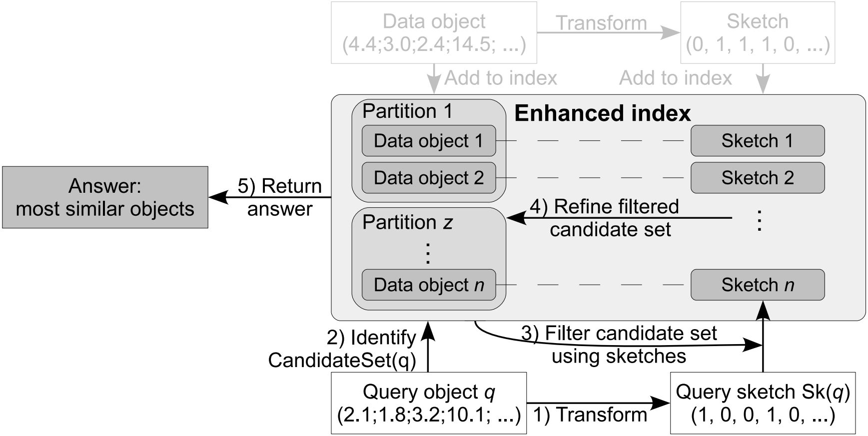


Figure: General concept of index enhancement with sketches

Description of Our Experiments

- Two datasets:
  - DeCAF and CoPhIR, each with 1,000,000 objects
  - Visual descriptors of images
  - Vectors of $4,096$ and 280 floats respectively
- Two indexes:
  - M-Index
  - PPP-codes
- Two lengths of sketches:
  - 64 bits and 32 bits

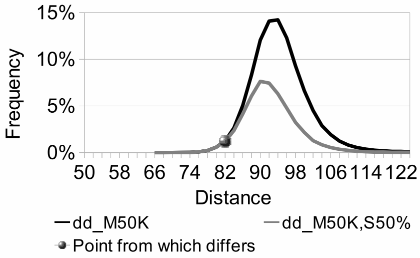
For a whole family of indexes, sketches can be created nearly for free

- Many indexes use a static set of pivots to organize objects and all the object-pivot distances are computed
- We can use this information to create sketches practically for free

Experiments – Distance Density on Candidate Set

A selected query q_1:

1. black curve: distance density of $CandidateSet(q_1)$ of size 50,000 objects with respect to q_1
2. grey curve: 50 % of $CandidateSet(q_1)$ filtered out according to the Hamming distance between corresponding sketches



Objects within the smallest distances are preserved (499 out of 500).

Experiments – Measured Recall

Selected results: DECAF dataset

1. Size of candidate set selected by index
   - M-Index and PPP codes respectively
2. Relative size of candidate set filtered out using 64bit sketches
3. Measured recall on 10-NN queries

M-Index cand. set	Percentage filtered out	Recall	PPP-Codes cand. set	Percentage filtered out	Recall
100,000	0 %	97.6	20,000	0 %	97.3
	50 %	97.4		30 %	97.1
	65 %	97.0		50 %	96.4
40,000	0 %	91.0	10,000	0 %	94.3
	50 %	90.7		30 %	94.0
	56 %	90.1		50 %	92.9

Table: Results – example

Summary

- General enhancement of indexes with sketches
- Negligible memory overhead
- Practically no additional time needed for the whole family of indexes
- Significant reduction of candidate set (30–65 %) with a small recall loss (0.2–0.9 %)
- It promises significant speed-up of query evaluation