FIT2010@Kyushu university

SketchSort: An Efficient Nearest Neighbor Graph Construction Method

Yasuo Tabei
JST Minato Project, Sapporo, Japan

Outline

- Motivation
- Method
- Experiments and Results

Data represented as vector

Text

Vector

x_t=(1, 0, 1, 0, 0, …)

Image

x_i=(0.2, -0.3, -1.3, 1.2, 2.2, …)

Chemical Compound, Protein, DNA/RNA etc

Locality Sensitive Hashing
(Gionis et al,99)

- Mapping vector to binary string (sketch)
- Conserve the distance in the original space

- Enable to store gigascale data in main memory

- Speed up learning algorithms

x=(0.2, -0.3, -1.3, 1.2, 2.2, …)

s=10101011101010101

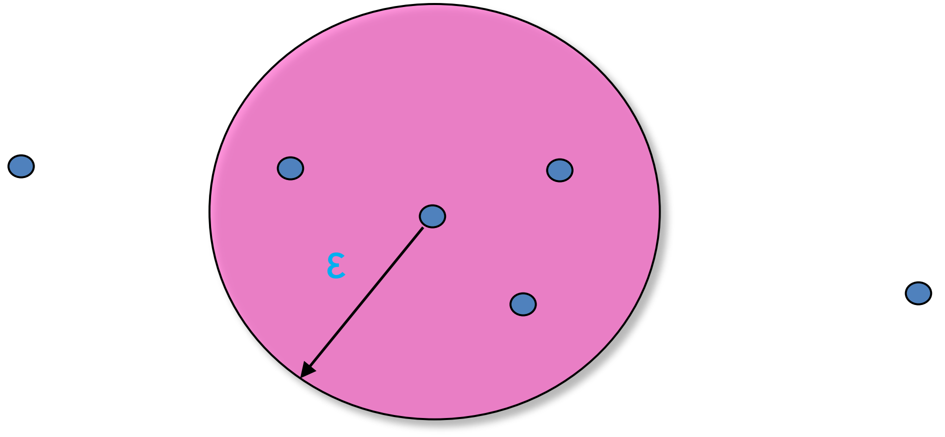
All Pairs Similarity Search

- Finding all neighbor pairs from sketches

  - Find all pairs (i, j), i < j, $\Delta(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \epsilon$

- Enable to build a neighborhood graph

  - semi-supervised learning, spectral clustering, ROI detection in images, retrieval of protein sequences



Single Sorting Method (SSM)

- Find neighbors by sorting sketches

 - Various applications ex) google news

(a)Input data	(b) Sort	(c) Scan neighbors	
1:101111	7:000000	7:000000	
2:110101	4:010000	4:010000	
3:110010	8:010110	8:010110	w
4:010000	10:100100	10:100100	
5:101000	5:101000	5:101000	
6:111100	1:101111	1:101111	
7:000000	3:110010	3:110010	w
8:010110	2:110101	2:110101	
9:110110	9:110110	9:110110	
10:100100	6:111100	6:111100	

Drawbacks of Single Sorting

- Need a large number of distance calculation for achieving reasonable accuracy.
- Can not derive an analytic estimate of the fraction of missing neighbors.

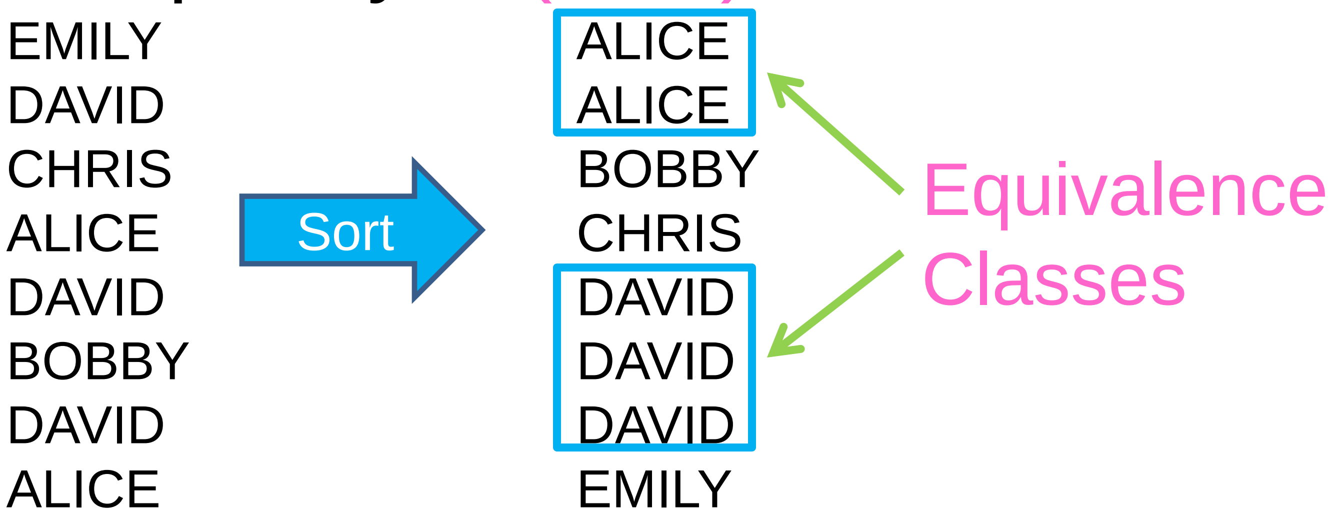
Overview of SketchSort

- Employ the multiple sorting method (MSM) as a building block

  - Enumerate all pairs within Hamming distance d from a string pool $S=\{s_1,\ldots,s_n\}$
- A number of distance calculation is significantly reduced
- A bound of the expected fraction of missing neighbors can be obtained.

Special case: Finding identical strings(d=0)

- Radix sort, and partition the strings into equivalence classes: O(n)
- Build edges between all pairs in equivalent classes: O(m)
- Complexity: O(n+m)



Multiple sorting method (d > 0)

- Mask d characters in all possible ways
- Perform radix sort $\binom{l}{d}$ times
- Time exponential to d, polynomial to the string lengh l
- Still linear to the number of strings!!
- Ex) d=2

```
 7:000   0001 0011 11 0
 4:010   0001 1101 11 0
 8:010   1001 0111 10 0
10:100   0011 1001 01 1
 5:101   0010 1110 10 0
 1:101   1111 0011 11 0
 2:110   0111 0111 00 1
 3:110   1000 1101 11 0
 9:110   1000 1101 11 0
 6:111   0011 1001 01 1
```

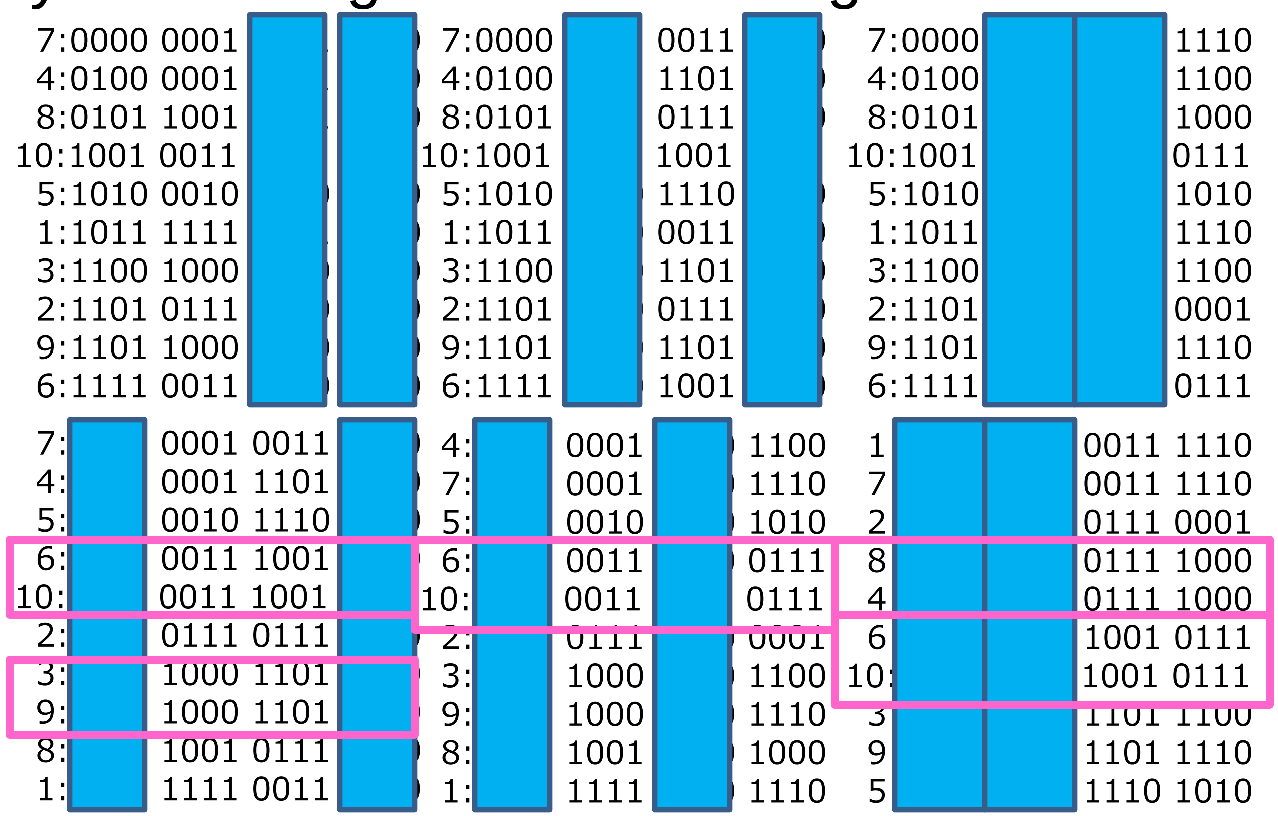
```
 7:0   0 0001 0011 1110
 4:0   0 0001 1101 1100
 8:0   1 1001 0111 1000
 5:1   0 0010 1110 1010
 3:1   0 1000 1101 1100
 6:1   1 0011 1001 0111
10:1   1 0011 1001 0111
 2:1   1 0111 0111 0001
 9:1   1 1000 1101 1110
 1:1   1 1111 0011 1110
```

Blockwise masking

- Mask d blocks in all possible ways
- The number of sotring operations reduced
- Non-neighbors might be detected

- Filtered out by calculating actual Hamming distances

▪ Ex)d=2



Recursive Algorithm

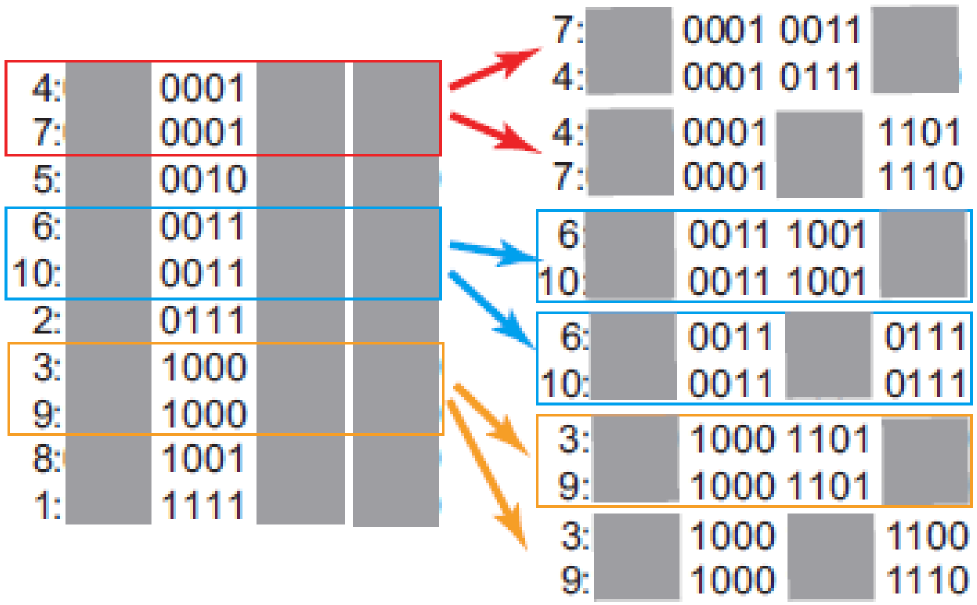


Figure 5: Updating equivalence classes in block concatenation. Strings in a block are sorted and equivalence classes (shown as square frames) are detected. A next block is concatenated to each equivalence class and sorted again.

SketchSort

- Basic idea: Map vectors to strings and apply MSM
- Not good: Create long strings and apply MSM at once
- Replication:

 - Create Q independent string pools of length l

 - apply MSM to each string pool

- Report the pairs less than a threshold ε

$$\Delta(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \epsilon$$

Duplication Checks

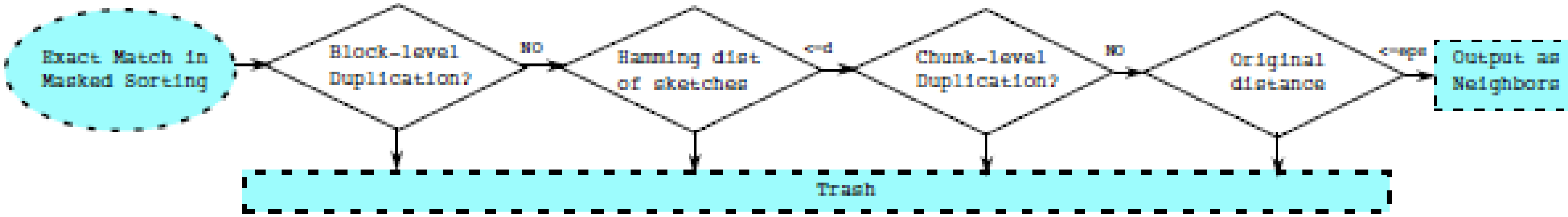


Figure 2: Global flow of our approach.

- Block-level duplication check

  - Define dictionary order of blocks, and take only minimum combinations of blocks.

  ex) d=2

  (1,2)<(1,3)<(1,4)<(2,3)<(2,4)<(3,4)

- Chunk-level duplication check

  - Take only minimum chunks.

Two types of errors

- True edges E*, Our results E
- Type-I error (false positive): A non-neighbor pair has a Hamming distance within d in at least one replicate

$$F_1 = \{(i,j) \mid (i,j) \in E, (i,j) \notin E^*\}.$$

- Type II-error (false negative): A neighbor pair has a Hamming distance larger than d in all replicates

$$F_2 = \{(i,j) \mid (i,j) \notin E, (i,j) \in E^*\}.$$

Bound of type-II error: Missing edge ratio

- Basically, type-II error is more crucial

 - type-I errors are filtered out by distance calculations

- Missing edge ratio (type-II error) is bounded as

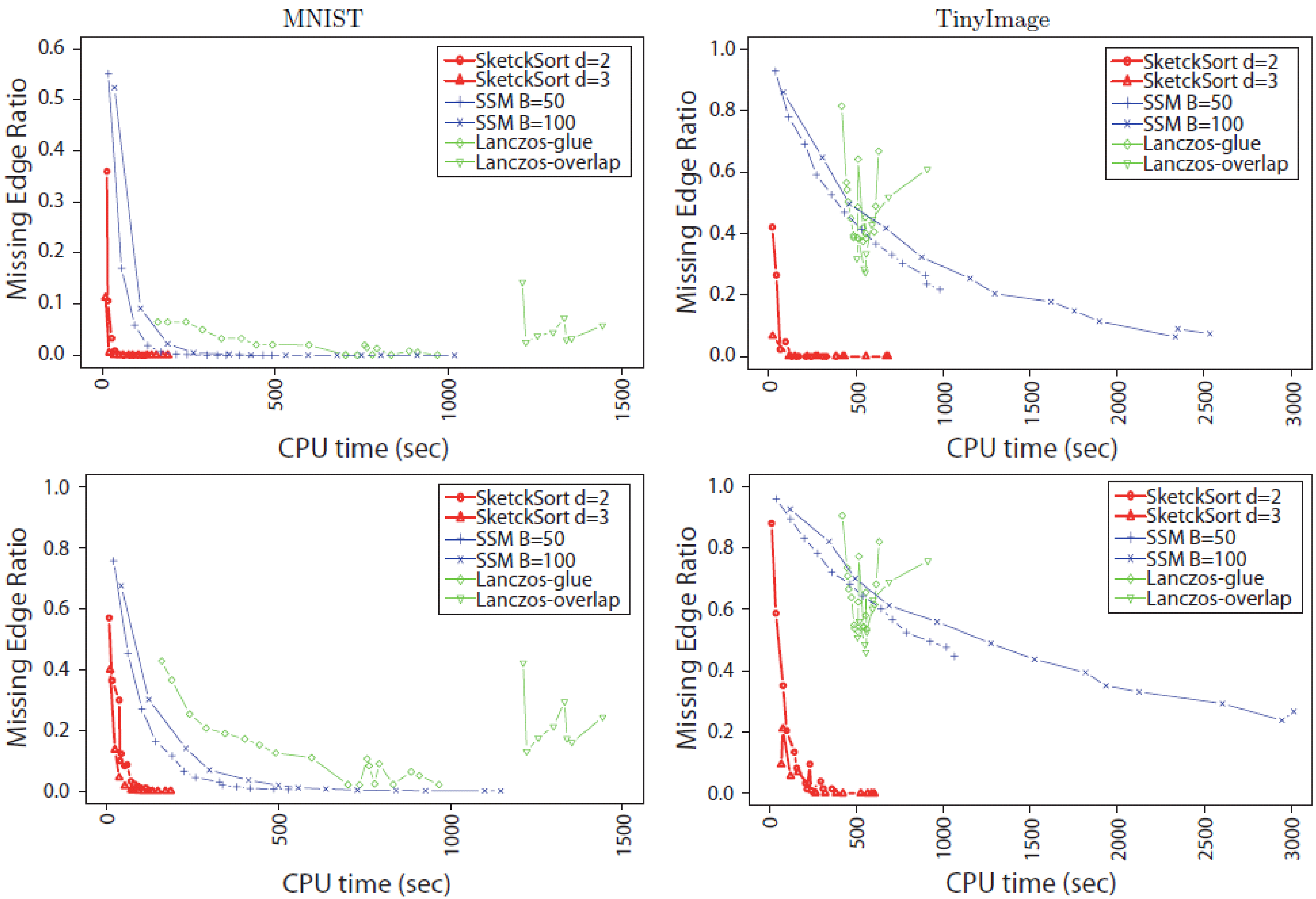
$$E\left[\frac{|F_2|}{|E^*|}\right] \leq \left(1 - \sum_{k=0}^{\lfloor d \rfloor} \binom{\ell}{k} p^k (1-p)^{\ell-k}\right)^Q,$$

where p is an upper bound of the non-collision probability of neighbors

$$p = \frac{\arccos(1-\epsilon)}{\pi}.$$

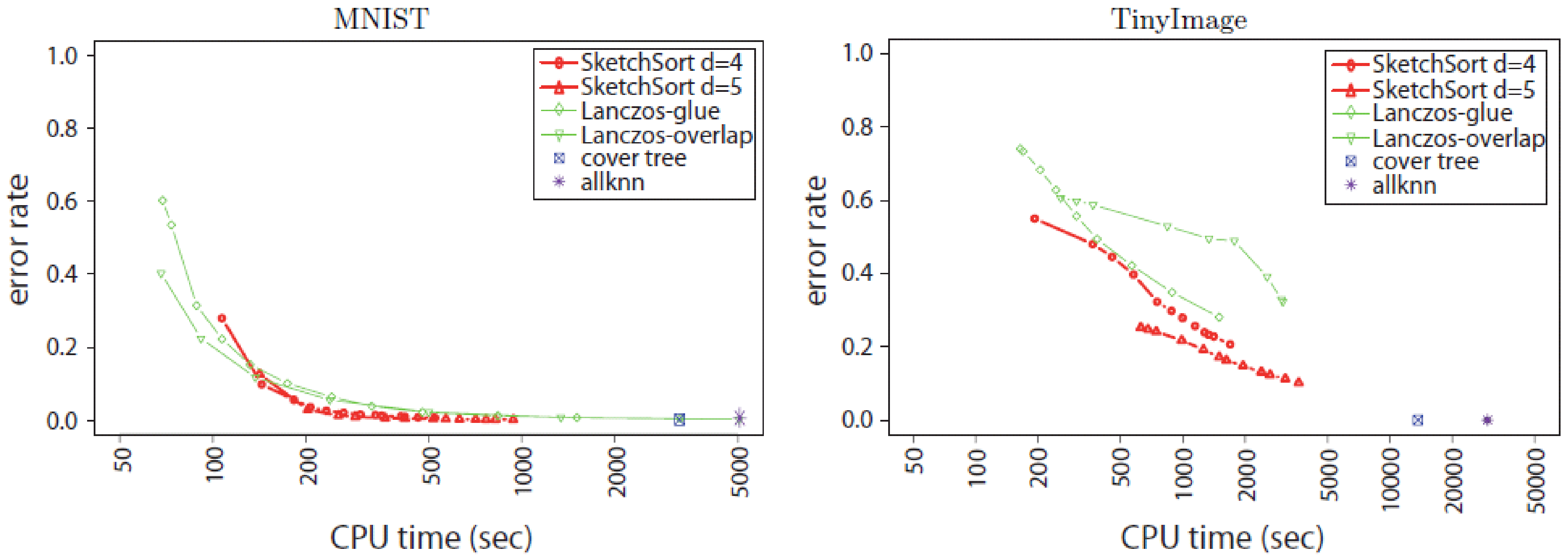
Results for All Pairs Similarity Search

Faster and more accurate than recent methods



All pairs similarity search on MNIST and TinyImage datasets for cosine distance thresholds 0.10Π (top) and 0.15Π (bottom).

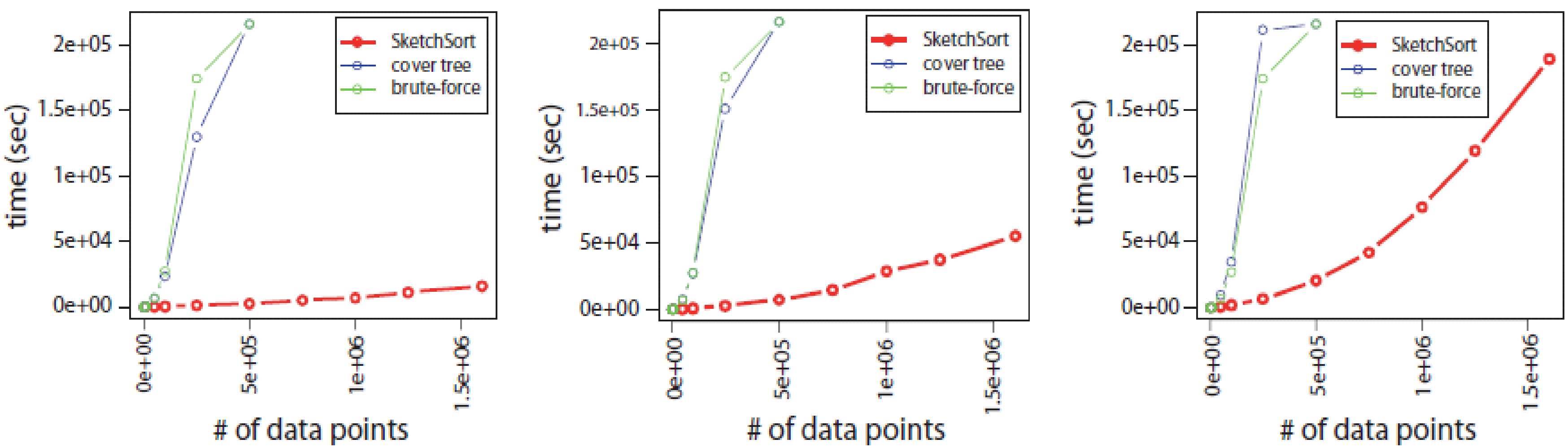
Results for 5-nearest neighbor search



Error rate for 5-nearest neighbor search on MNIST and TinyImage datasets

All Pairs Similarity Search in 1.6 Million Images

- Set parameters so as to keep missing edge ratio no more than 1.0×10^{-6}
- Enable to detect similar pairs nearly exactly
- Take only 4.3 hours for 1.6 million images



Near duplication detection in up to 1.6 million images at threshold 0.05Π (left), 0.10Π (middle) and 0.15Π (right)

A C++ implementation of SketchSort is available from http://code.google.com/p/sketchsort/

sketchsort

software for all pairs similarity search

Project Home | Downloads | Wiki | Issues | Source | Administer

Summary | Updates | People

Tip: Project owners, see our Getting Started guide for steps to configure your project. hide

This page is under construction.

Introduction

SketchSort?(1) is a software for all pairs similarity search. It takes as an input data points and outputs approximate nearest neighbor pairs within a distance. First, the input data points are mapped to binary bit strings by locality sensitive hashing, and then nearest neighbor pairs of strings within a Hamming distance are enumerated by the *multiple sorting method* (2). Finally, the cosine distances for such nearest neighbor pairs are calculated. If the cosine distance for a nearest neighbor pair is no more than a user-specified threshold , the nearest neighbor pair is outputted. One might worry about missed nearest neighbor pairs by our method. A theoretical bound of the expectation of missing edge ratio is derived. It enables us to set parameters so as to limit the empirical missing edge ratio as small as possible.

Quick Start

To compile SketchSort?, please type the followings:

```
tar -xvzf sketchsort-0.0.1.tar.gz
cd sketchsort-0.0.1/src
make
./sketchsort -hamdist 3 -numblocks 6 -cosdist 0.01 -numchunks 10 sample.txt outputfile
```

Usage

```
./sketchsort [options] input-file output-file
Option:
  -hamdist [d]: set the Hamming distance threshold (default: 2)
  -numblocks [b]: set the number of blocks (default: 3). That b is set to d + 3 is recommended.
  -cosdist [¥epsilon]:  set the cosine distance threshold (default: 0.01)
  -numchunks [Q]: set the number of chuncks (default: 3)
```

A reader who would like to know the meanings of these options can see our original paper (2).

Format of input file

First of all, data in the input file need to be centered at 0. Each line in the input file is a feature vector in which each element is separated by a space. Elements in lines need to be the same number. Here is an example.

```
0.3 1.2 0.3 0.1 0.3
0.2 -0.2 0.3 0.4 0.2
0.1 -0.3 0.3 -0.5 0.3
0.2 0.4 0.5 0.6 0.7
```

Star this project

Code license: New BSD License

Feeds: Project feeds

Project owners: People details

Yasuo.Tabei

Project contributors:

koji.tsuda