Datasets for approximate nearest neighbor search

Overview: This page provides several evaluation sets to evaluate the quality of approximate nearest neighbors search algorithm on different kinds of data and varying database sizes. In particular, we provide a very large set of **1 billion vectors**, to our knowledge this is the largest set provided to evaluate ANN methods.

Each comprises 3 subsets of vectors:
- base vectors: the vectors in which the search is performed
- *query vectors*
- *learning vectors*: to find the parameters involved in a particular method

In addition, we provide the groundtruth for each set, in the form of the pre-computed k nearest neighbors and their square Euclidean distance.

We use three different file formats:
- The vector files are stored in *.bvecs* or *.fvecs* format,
- The groundtruth file in is *.ivecs* format.

.bvecs, .fvecs and .ivecs vector file formats:

The vectors are stored in raw little endian.
Each vector takes 4+*d**4 bytes for *.fvecs* and *.ivecs* formats, and 4+*d* bytes for *.bvecs* formats, where *d* is the dimensionality of the vector, as shown below.

field	field type	description
d	int	the vector dimension
components	(unsigned char|float | int)*d	the vector components

The only difference between *.bvecs*, *.fvecs* and *.ivecs* files is the base type for the vector components, which is unsigned char, float or int, respectively.

In the Input/Output section below, we provide two functions to read such files in matlab.

Details and Download

MD5 sums are available here.

Vector set	Download	descriptor	dimension	nb base vectors	nb query vectors	nb learn vectors	file format
ANN_SIFT10K	siftsmall.tar.gz (5.1MB)	SIFT *(1)*	128	10,000	100	25,000	fvecs
ANN_SIFT1M	sift.tar.gz (161MB)	SIFT *(1)*	128	1,000,000	10,000	100,000	fvecs
ANN_GIST1M	gist.tar.gz (2.6GB)	GIST *(2)*	960	1,000,000	1,000	500,000	fvecs
ANN_SIFT1B	Base set (92 GB) Learning set (9.1 GB) Query set (964 KB) Groundtruth (512 MB)	SIFT *(3)*	128	1,000,000,000	10,000	100,000,000	bvecs

(1) SIFT descriptors, Mikolajczyk implementation of Hessian-affine detector
(2) GIST descriptors, INRIA C implementation
(3) SIFT descriptors Lowe's implementation (DoG)

The groundtruth files contain, for each query, the identifiers (vector number, starting at 0) of its k nearest neighbors, ordered by increasing (squared euclidean) distance.
- k=100 for the dataset ANN_SIFT10K, ANN_SIFT1M and ANN_GIST1M
- k=1000 for the big ANN_SIFT1B dataset

Therefore, the first element of each integer vector is the nearest neighbor identifier associated with the query.

For the largest set (ANN_SIFT1B), the groundtruth is provided for the whole set, but also for subsets of varying size. These subsets are the n first vectors of the bigann_base.bvecs file (n=1M,2M,5M,10M,20M,50M,100M,200M,500M,1B).

The performance measure is recall@R, that is, for varying values of R, the average rate of queries for which the 1-nearest neighbor is ranked in the top R positions. Please use this measure to allow a direct comparison of your system with most of the results reported in the literature.

Matlab resources

We provide several matlab functions to read the different files formats mentioned above, and the matlab functions used to computed the ground-truth for the ANN_SIFT1B dataset.

Function	Download	description
fvecs_read	fvecs_read.m	Read a .fvecs file. Each vector is stored in a column of the output matrix.
ivecs_read	ivecs_read.m	Read a .ivecs file. Each vector is stored in a column of the output matrix.
bvecs_read	bvecs_read.m b2fvecs_read.m	Read a .bvecs file. Each vector is stored in a column of the output matrix. The difference between bvecs_read and b2fvecs is the output type (byte for bvecs_read, single for b2fvecs_read).

In you are interested in efficient k-means or exhaustive nearest neighbors
search in Matlab/C/Python, let take a look at the Yael library.
It uses BLAS3 operations and is multi-threaded, and is required to execute
the groundtruth scripts provided above.

References and results

References: if you use these datasets, please cite the paper
where the dataset you used was formally introduced:
- ANN_SIFT1M and ANN_GIST1M were introduced in [2],
- the big ANN_SIFT1B was introduced in [5].

Note: if you have a paper that use these datasets, do not hesitate to tell us.
We may include you reference and results in this (forthcoming) section.

Groundtruth: exhaustive search using exhaustive Euclidean squared distance calculation

The package used to produce the groundtruth of ANN_SIFT1B and the timings for
the exact exhaustive search will be available "soon". It is based on the Yael library,
which provides optimized function for exact nearest neighbor search.

The timings for the exhaustive search were measured by the time linux program
on a Xeon 2.8 Ghz machine, using only 1 core (efficiency measure=user time).
There are given below for the whole set of queries.

Dataset	real	user	sys
ANN_SIFT1M	212 s	211 s	0.28 s
ANN_GIST1M	138 s	136 s	1.34 s
ANN_SIFT1B	260862 s	248011 s	136 s

Remarks:
- the user timings best reflects the processor activity
  (sys is mainly due to I/O access).
- the timings highly depend one the optimization level of BLAS/LAPACK.
  The timings we measured are probably not the best possible, as we observe
  large variability depending on implementation variables, cache phenomenons
  and concurrent jobs. For ANN_SIFT1B, a more realistic estimation (measured
  on 100M vectors with no concurrent job) is 215360 s.
- In our experiments, multi-threading significantly improves the efficiency
  but increases the total "user" time. Moreover timings are less reproducible.

Papers using these datasets

[1] Searching with quantization: approximate nearest neighbor search using short codes and distance estimators,
Hervé Jégou, Matthijs Douze and Cordelia Schmid, INRIA Technical report 7020, August 2009.
sets: ANN_SIFT1M and ANN_GIST1M

[2] Product quantization for nearest neighbor search,
Hervé Jégou, Matthijs Douze and Cordelia Schmid, IEEE Trans. PAMI, January 2011.
This is the journal version of the tech report above ("Searching with quantization").
sets: ANN_SIFT1M and ANN_GIST1M

[3] Searching with expectations,
Sandhwawalia and Jegou, ICASSP, March 2010.
brief: Transform coding approach for approximate nearest neighbors search.
sets: ANN_SIFT1M and ANN_GIST1M

[4] Transform coding for fast approximate nearest neighbor search in high dimensions,
Jonathan Brandt, CVPR'2010, June 2010.
sets: ANN_SIFT1M and ANN_GIST1M

[5] Searching in one billion vectors: re-rank with source coding,
Hervé Jégou, Romain Tavenard, Matthijs Douze and Laurent Amsaleg, ICASSP'2011, May 2011.
set: ANN_SIFT1B

[6] Locality sensitive hashing: a comparison of hash function types and querying mechanisms,
Loïc Paulevé, Hervé Jégou and Laurent Amsaleg, Pattern Recognition Letter, August 2010.
brief: Evaluation of several hash functions (random projection, lattices, k-means, hierarchical k-means)
and querying scheme (standard, multi-probe and query-adaptive) in LSH.
set: ANN_SIFT1M

Contact & History

Laurent Amsaleg CNRS/IRISA Linkmedia project-team
Hervé Jégou Facebook AI Research

July, 2010 release of ANN_SIFT10K, ANN_SIFT1M and ANN_GIST1M
September, 2010 dataset ANN_SIFT1B released