New approximate nearest neighbor benchmarks

2018-06-17

As some of you may know, one of my side interests is approximate nearest neighbor algorithms. I'm the author of Annoy, a library with 3,500+ stars on Github as of today. It offers fast approximate search for nearest neighbors with the additional benefit that you can load data super fast from disk using mmap. I built it at Spotify to use for music recommendations where it's still used to power millions (maybe billions) of music recommendations every day.

Approximate nearest neighbor search is very useful when you have a large dataset of millions of datapoints and you learn some kind of vector representation of these items. Word2vec might be the most well known example of this, but there's plenty of other examples. For an introduction of this topic, check out an older series of blog posts: Nearest neighbor methods and vector models.

Anyway, at some point I got a bit tired of reading papers of various algorithms claiming to be the fastest and most accurate, so I built a benchmark suite called ann-benchmarks. It pits a number of algorithms in a brutal showdown. I recently Dockerized it and wrote about it previously on this blog. So why am I blogging about it just three months later? Well...there's a lot of water under the bridge in the world of approximate nearest neighbors, so I decided to re-run the benchmarks and publish new results. I will probably do this a few times every year, at my own questionable discretion.

Changes

There were several new libraries added to this benchmark:

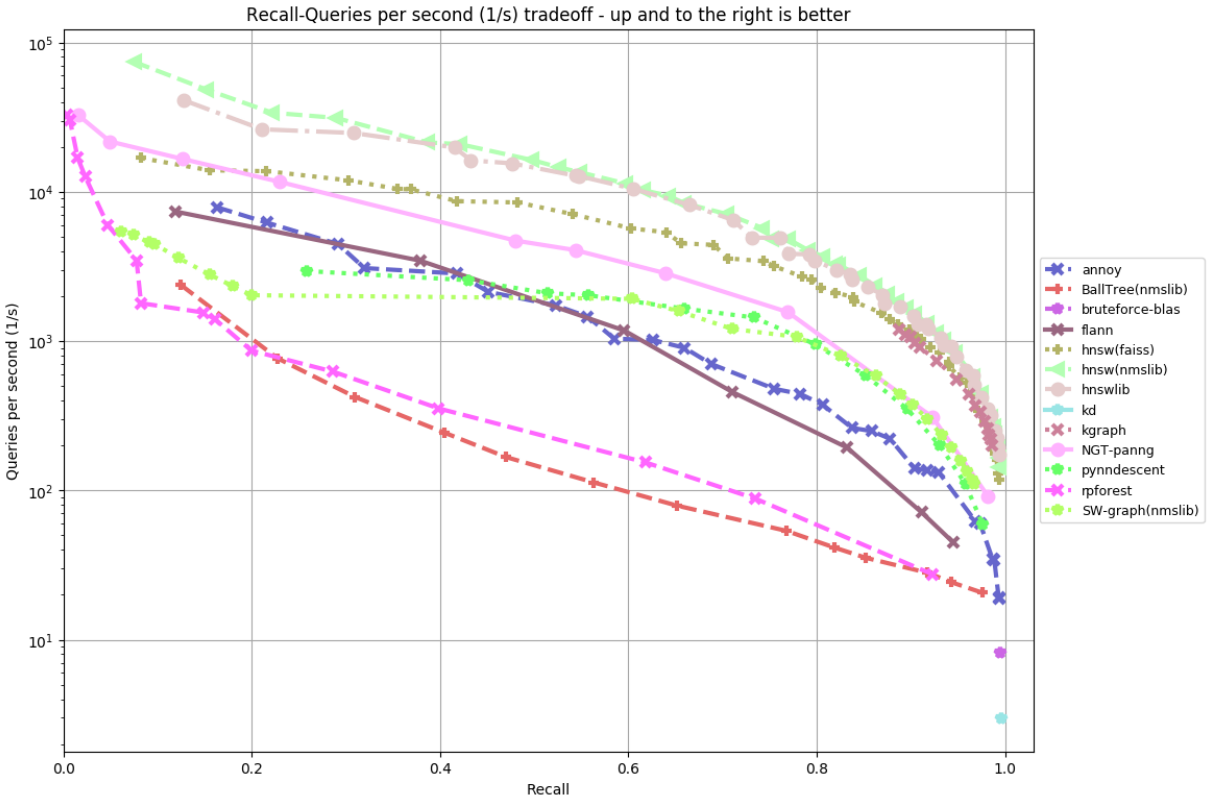
- NGT-Panng from Yahoo! Japan, a graph-based search structure
- pynndescent which is also a graph-based search algorithm, in fact based on the same paper as k-graph
- MRPT which is based on random projects, like Annoy.

On top of that, hnsw are included in three different flavor, one as a part of NMSLIB, one as a part of FAISS (from Facebook) and one as a part of hnswlib. I also dropped a few slow or semi-broken algorithms.

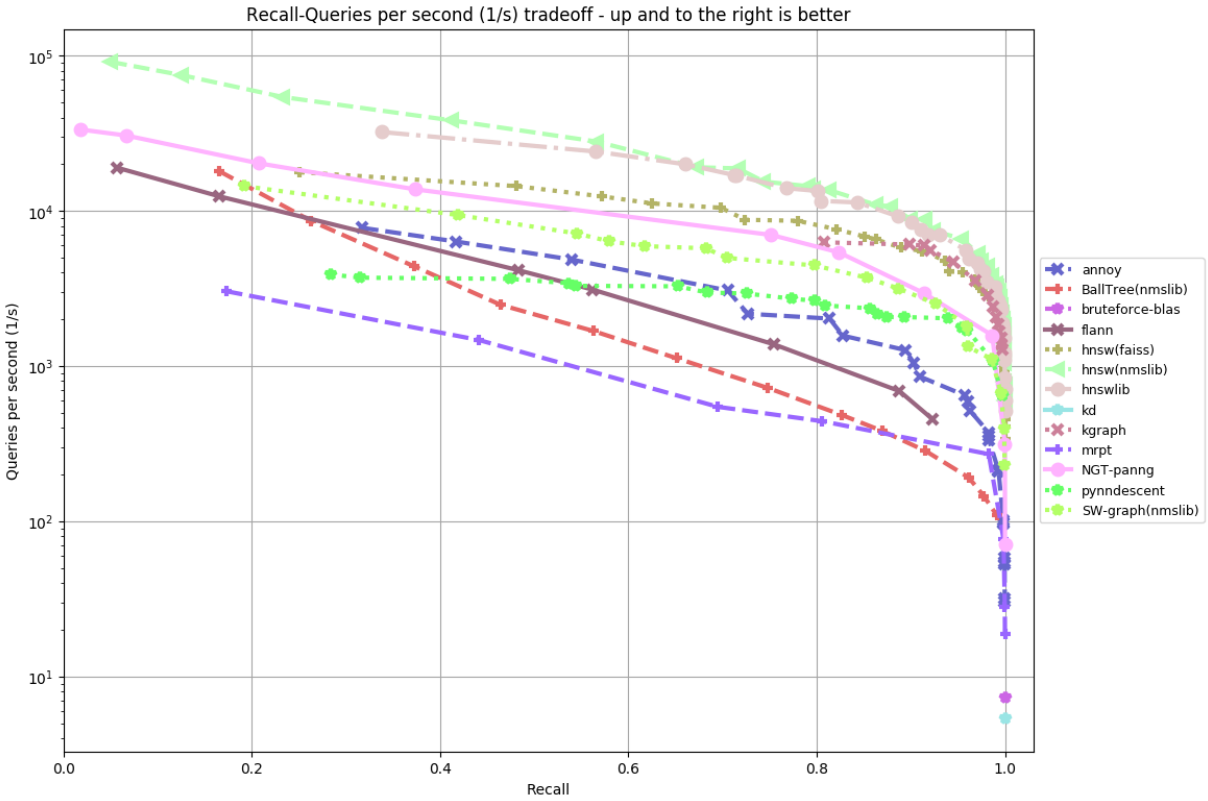
Another change this time was that I'm enforcing single-CPU queries. This m
benchmarks marginally slower, but I think it's the most "fair" way to compare. I think
batching is not always applicable for real world application. Previously, I used a thread pool to saturate all CPUs on the instance, but there was some concern that this might affect certain algorithms in different ways. So I used Docker's ability to tie the container to a single CPU.

Results

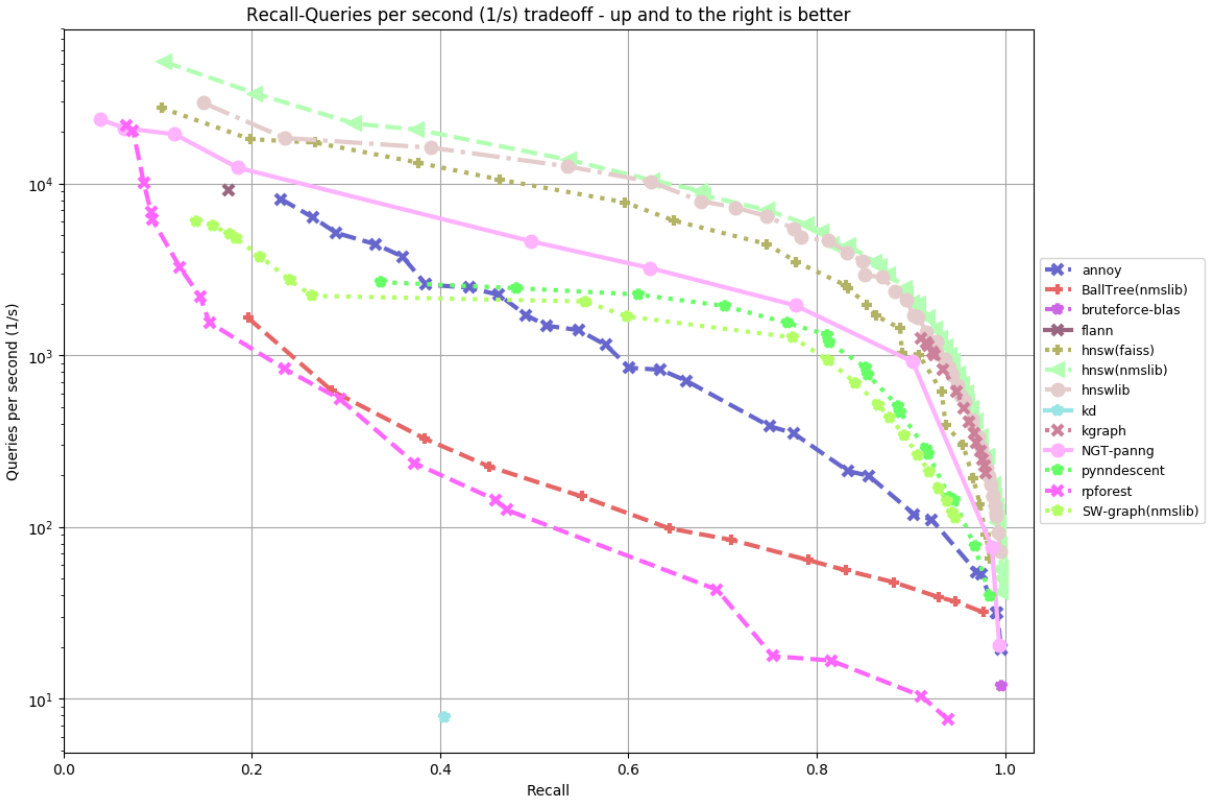
Without further ado, here's the results for the latest run. For the glove-100-angular dataset:



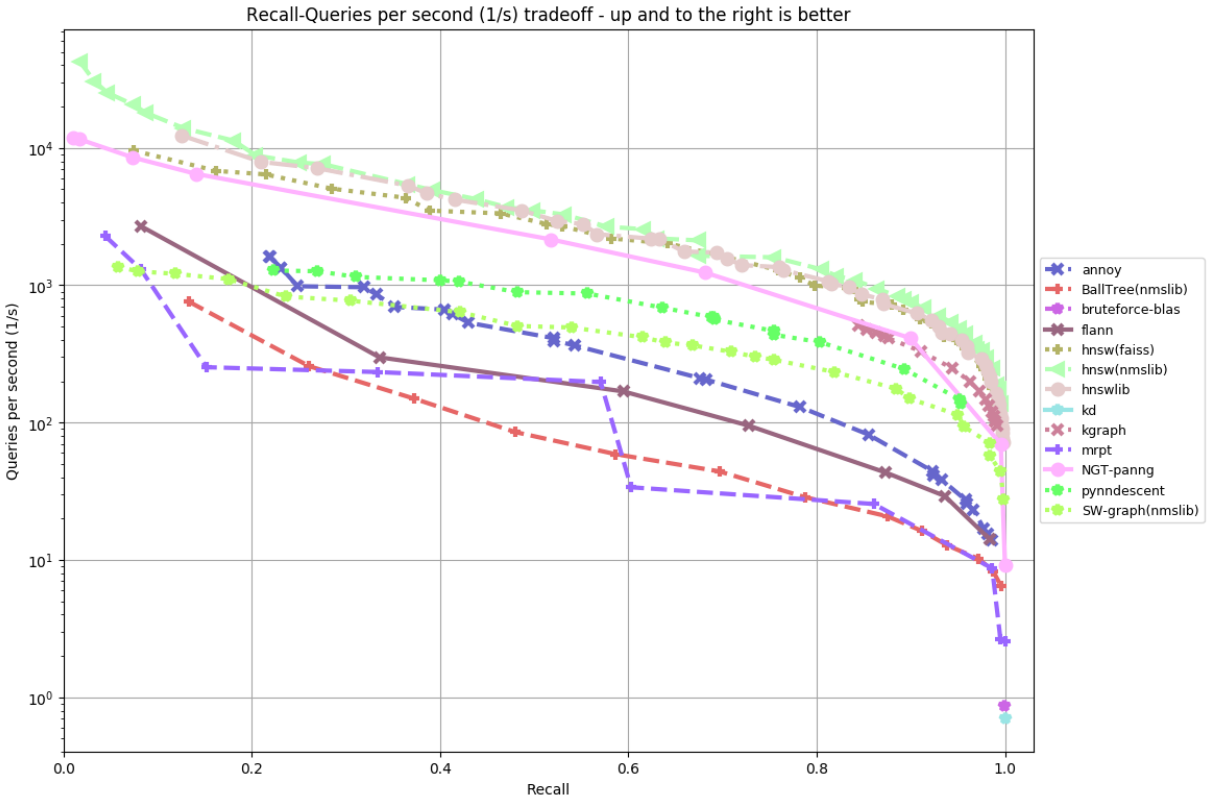
sift-128-euclidean



nytimes-256-angular



gist-960-euclidean



Results: summarized

By now, you're probably squinting at charts to figure out which library is the best. To save you the pain, I'm just going to summarize it into a somewhat subjective list:

1. hnsw(nmbslib)
2. hnswlib
3. hnsw(faiss)
4. kgraph
5. NGT-panng
6. pynndescent
7. SW-graph(nmslib)
8. annoy
9. flann
10. BallTree(nmslib)
11. mrpt
12. rpforest

The various flavors of hnsw are all at the top, but that's partly because they were all built by the same person, Yury Malkov with a paper describing the approach.

pynndescent and kgraph are both based on the same paper so it's not surprising their performance is fairly similar.

For some reason, MRPT would crash when I ran it on angular data, and I gave up after some time investigating it. Hopefully next benchmark will feature MRPT for angular data as well.

There's more goodies! Martin Aumüller and Alexander Faithfull have contributed export all the results to a website. I put it up on a temporary URL for you to enjoy.

That's it! ann-benchmarks ~~currently has almost 500 stars on Github, so I'd love it if you can pay it a visit and who knows... starring a repo just takes a second. Just saying!~~ just passed 500 stars on Github, meaning it's a legitimate project now! 🎉

Want to get blog posts over email?

Enter your email address and get weekly emails with new articles!

Your email address Subscribe!

Related posts

Interviewing is a noisy prediction problem 2018-05-02
Nearest neighbor methods and vector models – part 1 2015-09-24
The hacker's guide to uncertainty estimates 2018-10-08
New benchmarks for approximate nearest neighbors 2018-02-15
Data architecture vs backend architecture 2019-01-10
Conversion rates – you are (most likely) computing them wrong 2017-05-23
Interview with a Data Scientist: Erik Bernhardsson 2015-10-28

Erik Bernhardsson

... is the CTO at Better, which is a startup changing how mortgages are done. I write a lot of code, some of which ends up being open sourced, such as Luigi and Annoy. I also co-organize NYC Machine Learning meetup. You can follow me on Twitter or see some more facts about me.