https://github.com/kaja47/sketches

approximate similarity search

1000000 sets

Jaccard similarity

$$\frac{|A \cap B|}{|A \cup B|}$$

```
val sets: Array[Array[Int]] = loadDataSet()

val sims = Array.fill(sets.length) { new TopKFloatInt(64) }

for {i <- 0 until sets.length; j <- i until sets.length) {
   val sim = jaccardSimilarity(sets(i), sets(j))
   sims(i).insert(sim.toFloat, j)
   sims(j).insert(sim.toFloat, i)
}

// 129400 s ~ 34 hours
```

MinHash

$$P\{ h(A) = h(B) \} = J(A, B)$$

```
val minhash = atrox.sketch.MinHash(sets, hashes)

for (i <- 0 until sets.length; j <- i until sets.length) {
   val sim = minhash.estimateSimilarity(i, j)
   sims(i).insert(sim.toFloat, j)
   sims(j).insert(sim.toFloat, i)
}

// 79000 s ~ 23 hours
```

```
val minhash = MinHash(sets, hashes)

val sims = minhash
  .withConfig(SketchCfg(maxResults = 64))
  .allSimilarItems

// 54500 s ~ 15 hours
```

```
val cfg = LSHBuildCfg(
  maxBucketSize = 950,
  minBucketSize = 2
)

val minhash = MinHash(sets, hashes)
val lsh     = LSH(minhash, bands, cfg)

val sims = lsh
  .withConfig(LSHCfg(maxResults = 64))
  .allSimilarItems()

// 300 s
```

```
val minhash = MinHash(sets, hashes)
val lsh     = LSH(minhash, bands, cfg)

val sims = lsh.withConfig(LSHCfg(
   maxResults = 64,
   compact    = false
)).allSimilarItems()

// 120 s  (1000x)
```

```
val minhash = MinHash(sets, hashes)
val lsh     = LSH(minhash, bands, cfg)

val sims = lsh.withConfig(LSHCfg(
   maxResults = 64,
   compact    = false,
   parallel   = true
)).allSimilarItems()

// 57 s
```

```
MinHash
WeightedMinHash
RandomHyperplanes
RandomProjections
HammingDistance
SimHash

+ p-stable distributions
+ spectral hashing
+ speed
- memory
```

https://github.com/kaja47/sketches

approximate similarity search