Locality Sensitive Hashing

Rebecca C. Steorts (based upon prior work with Andee Kaplan)

2024-09-05

Agenda

- Locality Sensitive Hashing (LSH)
- Hash functions
- Hashed shingles
- Signatures
- Characteristic Matrix
- Minhash (Jaccard Similarity Approximation)
- Back to LSH

```
##   cora_id unique_id
## 1       1         1
## 2       2         1
## 3       3         1
## 4       4         1
## 5       5         1
## 6       6         1
```

```
##      cora_id unique_id
## 1694    1874       135
## 1809    1875       135
## 1695    1876       136
## 1696    1877       136
## 1697    1878       136
## 1810    1879       136
```

```
## [1] 1879    2
```

LSH

Locality sensitive hashing (LSH) is a fast method of blocking for record linkage that orginates from the computer science literature.

- LSH tries to preserve similarity after dimension reduction.
  - What kind of similarity? $\leftrightarrow$ What kind of dimension reduction?

Data set

Consider the cora citation data set.

1. Shingle all records using a shingle size of 3. Then calculate the Jaccard similarity for all record pairs using the shingled records.

```
# get only the columns we want
# number of records
```

```r
n <- nrow(cora)
# create id column
dat <- data.frame(id = seq_len(n))
# get columns we want
dat <- cbind(dat, cora[, c("title", "authors", "journal")])
shingles <- apply(dat, 1, function(x) {
  # tokenize strings
  tokenize_character_shingles(paste(x[-1], collapse=" "), n = 3)[[1]]
})
# empty holder for similarities
jaccard <- expand.grid(record1 = seq_len(n),
                       record2 = seq_len(n))

# don't need to compare the same things twice
jaccard <- jaccard[jaccard$record1 < jaccard$record2,]


time <- Sys.time() # for timing comparison
jaccard$similarity <- apply(jaccard, 1, function(pair) {
  # get jaccard similarity for each record pair
  jaccard_similarity(shingles[[pair[1]]], shingles[[pair[2]]])
})
# timing
time <- difftime(Sys.time(), time, units = "secs")
head(jaccard)
```

```
##      record1 record2 similarity
## 1880       1       2  0.8648649
## 3759       1       3  0.8648649
## 3760       2       3  1.0000000
## 5638       1       4  0.8648649
## 5639       2       4  1.0000000
## 5640       3       4  1.0000000
```

2. Visually plot the Jaccard similarity. What do you observe?

```r
# plot the jaccard similarities for each pair of records
ggplot(jaccard) +
  geom_raster(aes(x = record1, y = record2,
                  fill=similarity)) +
  theme(aspect.ratio = 1) +
  scale_fill_gradient("Jaccard similarity") +
  xlab("Record id") + ylab("Record id")
```

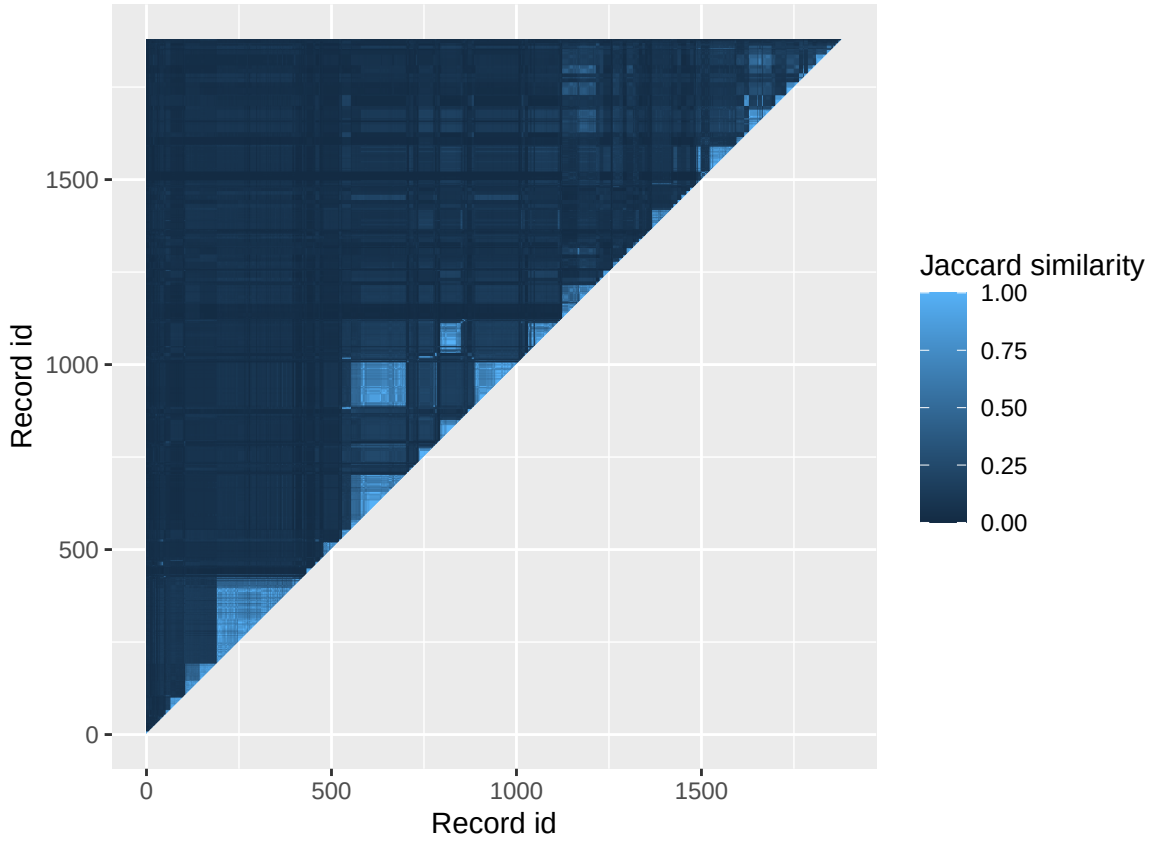


Figure 1: Jaccard similarity for each pair of records. Light blue indicates the two records are more similar and dark blue indicates less similar.

Perform LSH

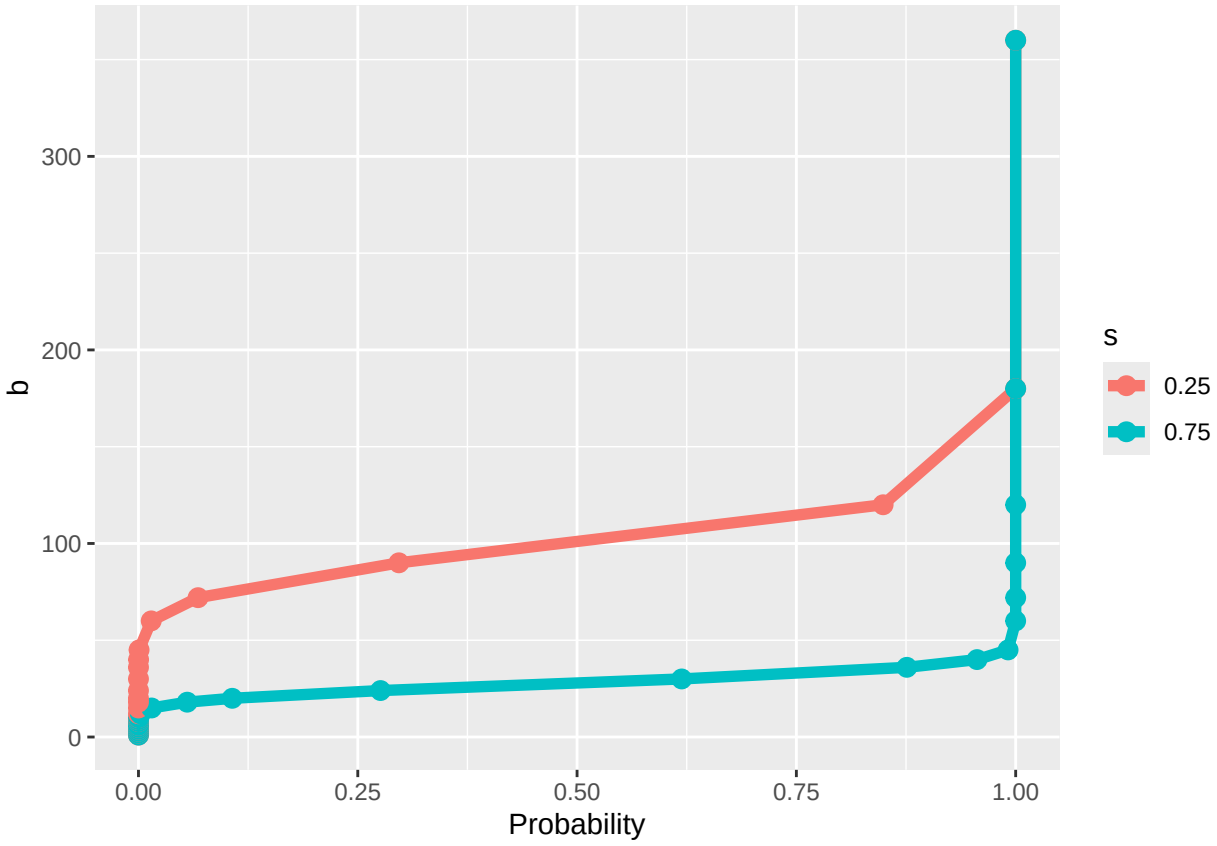
3. To reduce the overall computational complexity, let's use the lsh approximation.

There an easy way to do LSH using the built in functions in the `textreuse` package via the functions `minhash_generator` and `lsh` (so we don't have to perform it by hand):

Find the number of buckets or bands to use

```
library(numbers)
m <- 360
bin_probs <- expand.grid(s = c(.25, .75), h = m, b = divisors(m))
#bin_probs
# choose appropriate num of bands and number of random permutations m (tuning parameters)
bin_probs$prob <- apply(bin_probs, 1, function(x) lsh_probability(x[["h"]], x[["b"]], x[["s"]]))
# plot as curves
ggplot(bin_probs) +
  geom_line(aes(x = prob, y = b, colour = factor(s), group = factor(s)), size = 2) +
  geom_point(aes(x = prob, y = b, colour = factor(s)), size = 3) +
  xlab("Probability") +
  scale_color_discrete("s")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# create the minhash function
minhash <- minhash_generator(n = m, seed = 02082018)
b <- 90
```

Build corpus and perform shingling

```
head(dat)
```

```
##   id           title
## 1  1 Inganas and M.R
## 2  2            <NA>
## 3  3            <NA>
## 4  4            <NA>
## 5  5            <NA>
## 6  6            <NA>
##                                                                                 authors
## 1                                M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O
## 2 M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R.   Andersson
## 3 M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O.  Inganas and M.R.  Andersson
## 4   M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R. Andersson
## 5  M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O.  Inganas and M.R. Andersson
## 6   M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R. Andersson
##                    journal
## 1 Andersson, J Appl. Phys.
## 2             JAppl. Phys.
## 3            J Appl. Phys.
## 4             J Appl.Phys.
## 5            J Appl. Phys.
## 6             J Appl.Phys.
```

```
# build the corpus using textreuse
docs <- apply(dat, 1, function(x) paste(x[-1], collapse = " ")) # get strings
names(docs) <- dat$id # add id as names in vector
corpus <- TextReuseCorpus(text = docs, # dataset
                          tokenizer = tokenize_character_shingles, n = 3,
                          simplify = TRUE, # shingles
                          progress = FALSE, # quietly
                          keep_tokens = TRUE, # store shingles
                          minhash_func = minhash) # use minhash
head(minhashes(corpus[[1]]))
```

```
## [1] -2064207635 -2111669997 -2125164234 -2097867716 -2144942958 -2116065215
```

```
length(minhashes(corpus[[1]]))
```

```
## [1] 360
```

Note that all our records are now represented by 360 randomly selected and hashed shingles. Comparing these shingles are equivalent to finding the Jaccard similarity of all the record pairs. We still have an issue of all the pairwise comparison.

Find buckets, candidate records, and Jaccard similarity

Now, we find the buckets, candidates records, and calculate the Jaccard similarity for the candidate records (in the buckets)

```
# perform lsh to get buckets
buckets <- lsh(corpus, bands = b, progress = FALSE)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## i Please use `gather()` instead.
## i The deprecated feature was likely used in the textreuse package.
##   Please report the issue at <https://github.com/ropensci/textreuse/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# grab candidate pairs
candidates <- lsh_candidates(buckets)

# get Jaccard similarities only for candidates
lsh_jaccard <- lsh_compare(candidates, corpus,
                           jaccard_similarity, progress = FALSE)
head(buckets)
```

```
## # A tibble: 6 x 2
##   doc   buckets
##   <chr> <chr>
## 1 1     fb93d6f4c56666ec8210570af8e8edd0
## 2 1     cf942dbe840d4365cf182ea24d6951c8
## 3 1     e9535be0f24e39103ba1f11442cc170e
## 4 1     52a293069e3920a0f56e38a0f0c6af37
## 5 1     b751d8b2d24bec53a78b3043dc017837
## 6 1     0ce07d00e8c2e811204352b51229ed18
```

```
dim(buckets)
```

```
## [1] 169110      2
```

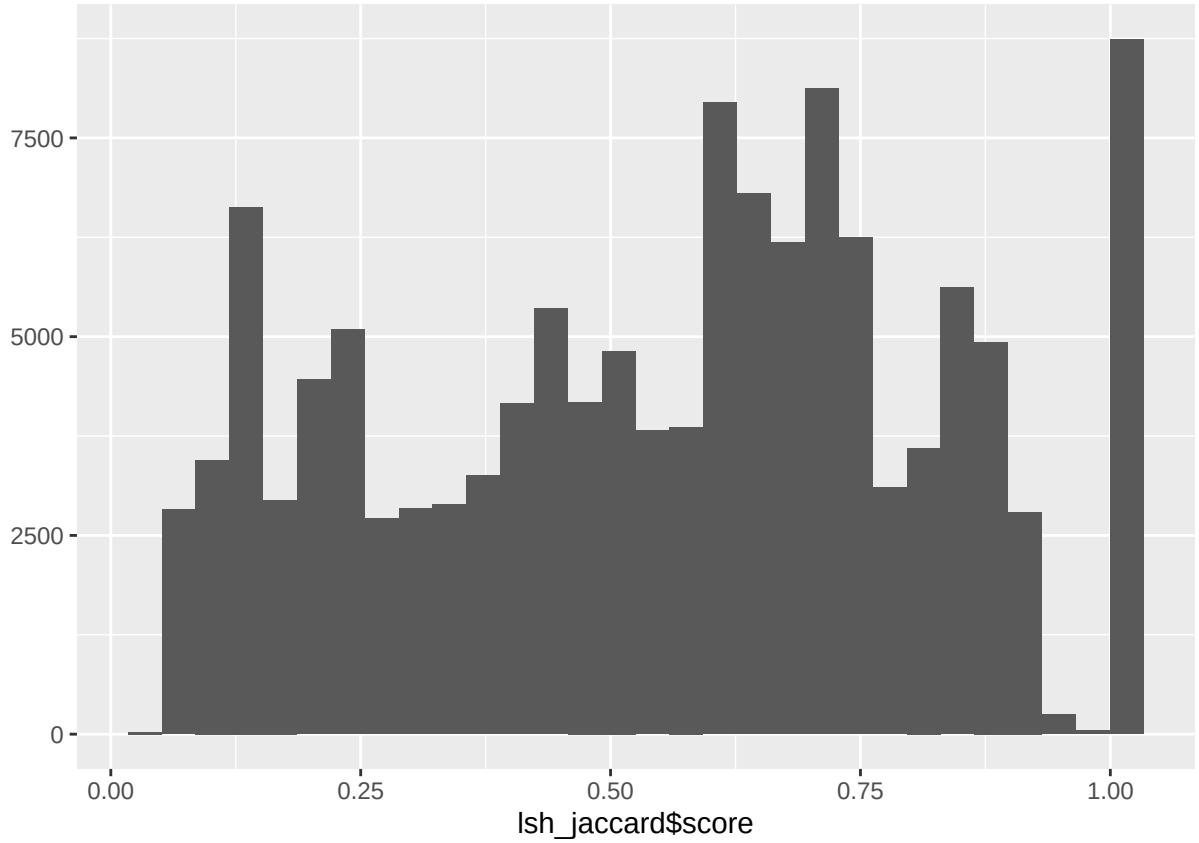
```
length(unique(buckets))
```

```
## [1] 2
```

We now plot the Jaccard similarities that are candidate pairs (under LSH)

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


7500
5000
2500
0
0.00
0.25
0.50
0.75
1.00
lsh_jaccard$score