PA2 Report

Syth Ryan

10/22/16

COM S 435/535

MinHash

My term Document Matrix was of the following type: Map<String, Map<String, Integer>>. where the first map is Map<Document, Map of Terms in that Document>. The encased map is, Map <Term, Term Frequency>. If a key (term) was not in a document, it was considered to have 0 frequency saving space by not having to make an all term list for each document. Multiple random hash functions did the hashing for the terms. Random a and b were generated at the beginning of the MinHash and stored for later hashing, along with 1 prime number that is the next prime greater than the number of all terms.

MinHashAccuracy

	400	600	800
0.04	28272	492	95
0.07	4	0	0
0.09	0	0	0

It appears that the greater number of permutations the more accurate minhash becomes. Increasing the accuracy required also appears to weed out most of any that may still be around for lower permutation counts.

MinHashTime

for 600 permutations:

Computing exact Jaccard took about 1242 seconds.

Computing MinHash matrix took about 352 seconds.

Computing approximate Jaccard took about 359 seconds.

Storing b hash tables could cause a problem as memory could grow large. I decided to use nested array lists (Bands<Signature<Document>>). I use the first element of each list to be the Signature so that I can access elements easier and only add as many as I need.

My hash function was simple. I stored my signature bands as the String value. For example, tuple [5, 31, 15, 16] would be hashed to string “[5, 31, 15, 16]”.

```
nearDuplicatesOf:

For each table (band)
    For each signature
        For each Document
            If that document == Doc being searched for.
                Copy all docs in that collection
Remove duplicates
```

```
100, .9, "space-0.txt"
```

```
[space-0.txt, space-0.txt.copy1, space-0.txt.copy2, space-0.txt.copy3, space-
0.txt.copy5, space-0.txt.copy6, space-0.txt.copy7, space-0.txt.copy4]
```

```
50, .9, “space-0.txt”
```

```
[space-0.txt, space-0.txt.copy1, space-0.txt.copy3, space-0.txt.copy4, space-
0.txt.copy5, space-0.txt.copy6, space-0.txt.copy7]
```

```
100, .5, "space-0.txt"
```

```
[space-0.txt, space-0.txt.copy1, space-0.txt.copy2, space-0.txt.copy3, space-
0.txt.copy4, space-0.txt.copy5, space-0.txt.copy6, space-0.txt.copy7, space-531.txt,
space-531.txt.copy1, space-531.txt.copy2, space-531.txt.copy3, space-531.txt.copy4,
space-531.txt.copy5, space-531.txt.copy6, space-531.txt.copy7]
```

```
50, .5, “space-0.txt”
```

```
[space-0.txt, space-0.txt.copy1, space-0.txt.copy2, space-0.txt.copy3, space-
0.txt.copy4, space-0.txt.copy5, space-0.txt.copy6, space-0.txt.copy7, space-531.txt,
space-531.txt.copy1, space-531.txt.copy3, space-531.txt.copy4, space-531.txt.copy5,
space-531.txt.copy6, space-531.txt.copy7, space-531.txt.copy2]
```

```
50, .9, “baseball0.txt”
```

```
[baseball0.txt, baseball0.txt.copy1, baseball0.txt.copy2, baseball0.txt.copy3,
baseball0.txt.copy4, baseball0.txt.copy5, baseball0.txt.copy6, baseball0.txt.copy7]
```

```
100, .9, “baseball0.txt”
```

```
[baseball0.txt, baseball0.txt.copy1, baseball0.txt.copy2, baseball0.txt.copy3,
baseball0.txt.copy4, baseball0.txt.copy5, baseball0.txt.copy6, baseball0.txt.copy7]
```

```
50, .5 “baseball0.txt”

[baseball0.txt, baseball0.txt.copy1, baseball0.txt.copy2, baseball0.txt.copy3,
baseball0.txt.copy4, baseball0.txt.copy5, baseball0.txt.copy6, baseball0.txt.copy7]

100, .5 “baseball0.txt”

[baseball0.txt, baseball0.txt.copy1, baseball0.txt.copy2, baseball0.txt.copy3,
baseball0.txt.copy4, baseball0.txt.copy5, baseball0.txt.copy6, baseball0.txt.copy7]

50, .9 “baseball11.txt”

[baseball11.txt, baseball11.txt.copy1, baseball11.txt.copy3, baseball11.txt.copy5,
baseball11.txt.copy6, baseball11.txt.copy7]

100, .9, “baseball11.txt”

[baseball11.txt, baseball11.txt.copy1, baseball11.txt.copy3, baseball11.txt.copy5,
baseball11.txt.copy6, baseball11.txt.copy7, baseball11.txt.copy4,
baseball11.txt.copy2]
```