ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Process 2 Essay

Subject: Mining Massive Datasets

Group: 11

Instructor: MSc Nguyễn Thành An

Introduction to LSH

Locality Sensitive Hashing (LSH):

- LSH is fuzzy hashing technique
- Hashes similar input items into the same "buckets" with high probability
- This technique can be used for data clustering and nearest neighbor search
- It differs because it hash collisions are maximized, not minimized
- A way to reduce the dimensionality of high-dimensional data

Introduction to LSH

Upside:

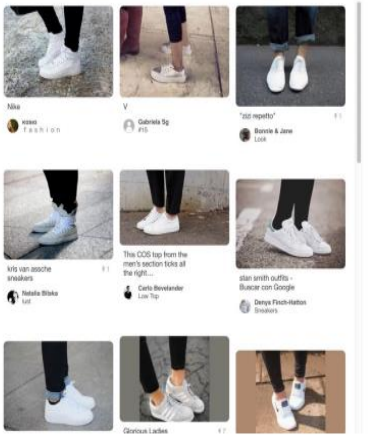
- Designed correctly, only a small fraction of points are ever examined.
- Performs well on large datasets with high dimensions.
- reducing computational complexity in high-dimensional problems.

Downside:

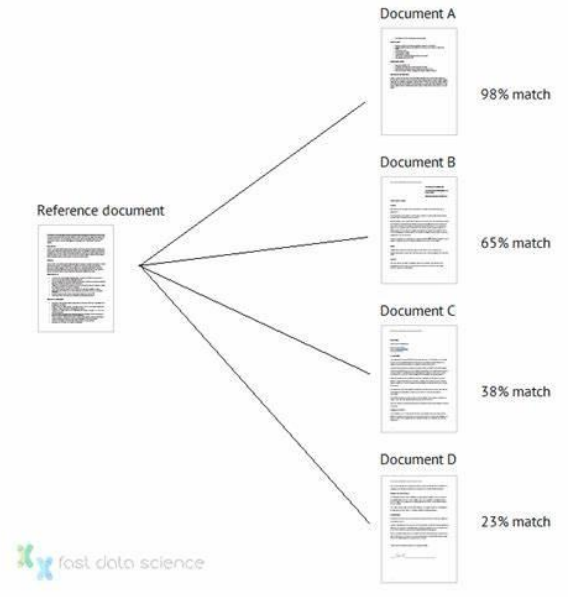
- There are false negatives – there might be similar items that get missed
- selecting hash functions and other parameters
- LSH is not a perfect solution for all problems, experimentation and tuning to fit specific scenarios.

Application

Q



Searching for Similar Images



Finding similar documents

Solve the requirements

Dataset: WebOfScience-5736.txt

The WebOfScience-5736 dataset contains 5736 corresponding documents, one per line.

Request: Implement the MinHashLSH algorithm:

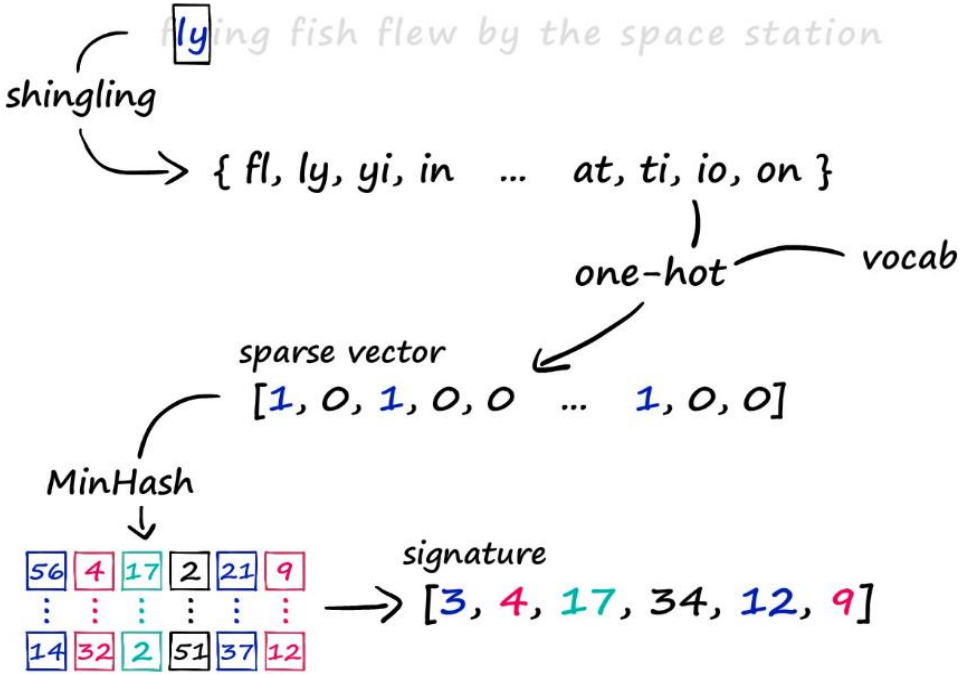
+ In-memory MinHashLSH

+ LargDataMinHashLSH

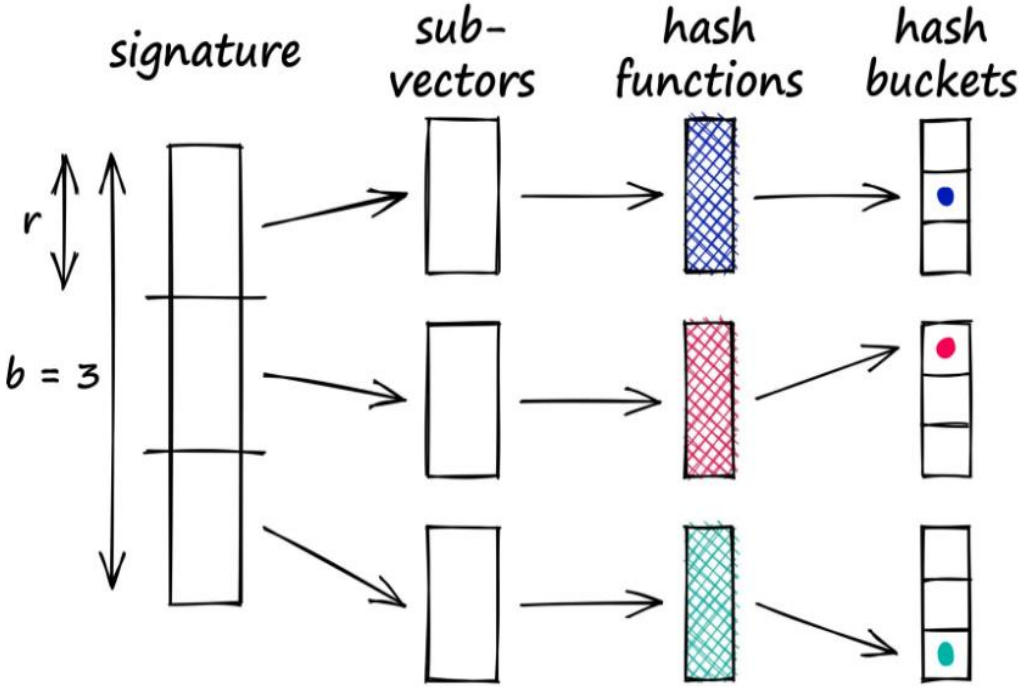
The first paragraphs of WebOfScience-5736.txt:

Phytoplasmas are insect-vectored bacteria that cause disease in a wide range of plant species. The increasing availability of molecular DNA analyses, expertise and additional methods in recent years has led to a proliferation of discoveries of phytoplasma-plant host associations and in the numbers of taxonomic groupings for phytoplasmas. The widespread use of common names based on the diseases with which they are associated, as well as separate phenetic and taxonomic systems for classifying phytoplasmas based on variation at the 16S rRNA-encoding gene, complicates interpretation of the literature. We explore this issue and related trends through a focus on Australian pathosystems, providing the first comprehensive compilation of information for this continent, covering the phytoplasmas, host plants, vectors and diseases. Of the 33 16Sr groups reported internationally, only groups I, II, III, X, XI and XII have been recorded in Australia and this highlights the need for ongoing biosecurity measures to prevent the introduction of additional pathogen groups. Many of the phytoplasmas reported in Australia have not been sufficiently well studied to assign them to 16Sr groups so it is likely that unrecognized groups and

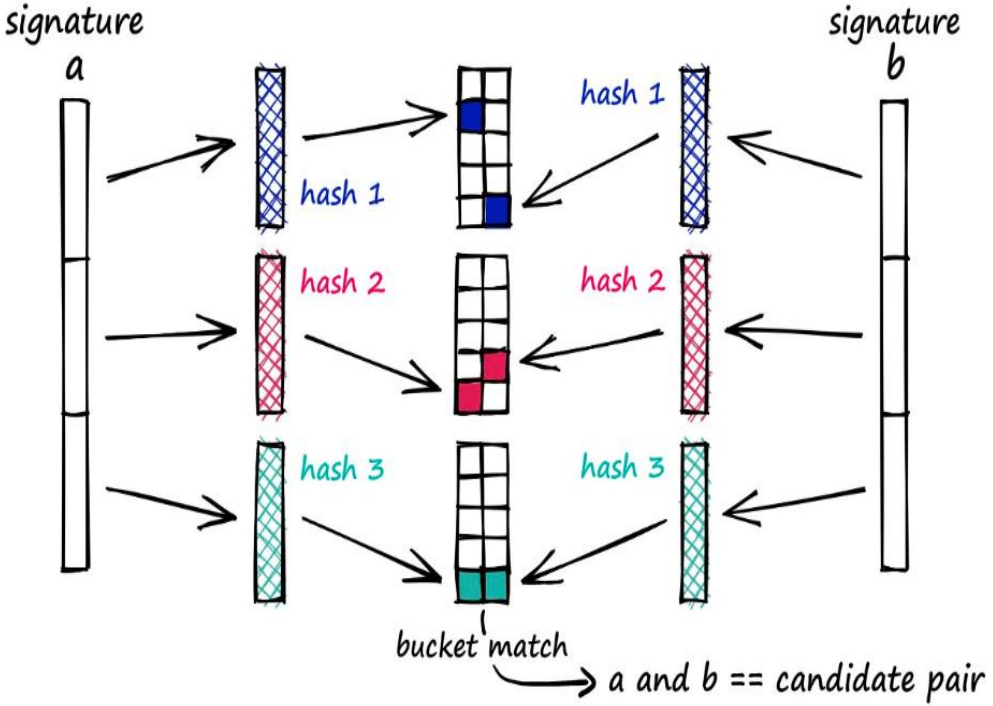
Shingling and Minhash



locality_sensity_hashing



approxNearestNeighbors



ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Pseudocode

Pseudocode for Shingling

```
1 Function shingling(self, documents : pd.DataFrame):
2       shingles_set <- an empty set
3       for each doc in documents['text'] do
4           for i from 0 to len(doc) - self.shingle_size + 1 do
5               shingle <- substring of doc from index i to index i+self.shingle_size
6               add shingle to shingles_set
7       bool_vectors <- an empty list
8       for each doc in documents['text'] do
9           document_vector <- an empty list
10          for each shingle in shingles_set do
11              if shingle exists in doc then
12                  add 1 to document_vector
13              else
14                  add 0 to document_vector
15          add document_vector to bool_vectors
16      returns bool_vectors
```

Pseudocode for minhashing

```
1 Function minhashing(self, bool_vectors):
2       num_docs <- length of bool_vectors
3       num_features <- length of bool_vectors[0]
4       signatures <- a 2D array of zeros with dimensions (self.num_hashes, num_docs)
5
6       for i from 0 to self.num_hashes-1 do
7          if length of self.permutations <= i then
8             append create_permutation(num_features) to self.permutations
9          permutation <- self.permutations[i]
10
11          for j from 0 to num_docs-1 do
12             signatures[i, j] <- minhash(bool_vectors[j], permutation)
13
14      return signatures
```

Pseudocode for jaccard_similarity

```
1 Function jaccard_similarity(self, sequence1, sequence2):
2       intersection_count <- 0
3       union_count <- length of sequence1
4       for each element1, element2 in zip(sequence1, sequence2) do
5           if element1 equals element2 then
6               increment intersection_count by 1
7       similarity <- intersection_count / union_count
8       return similarity
```

Pseudocode for locality_sensity_hashing

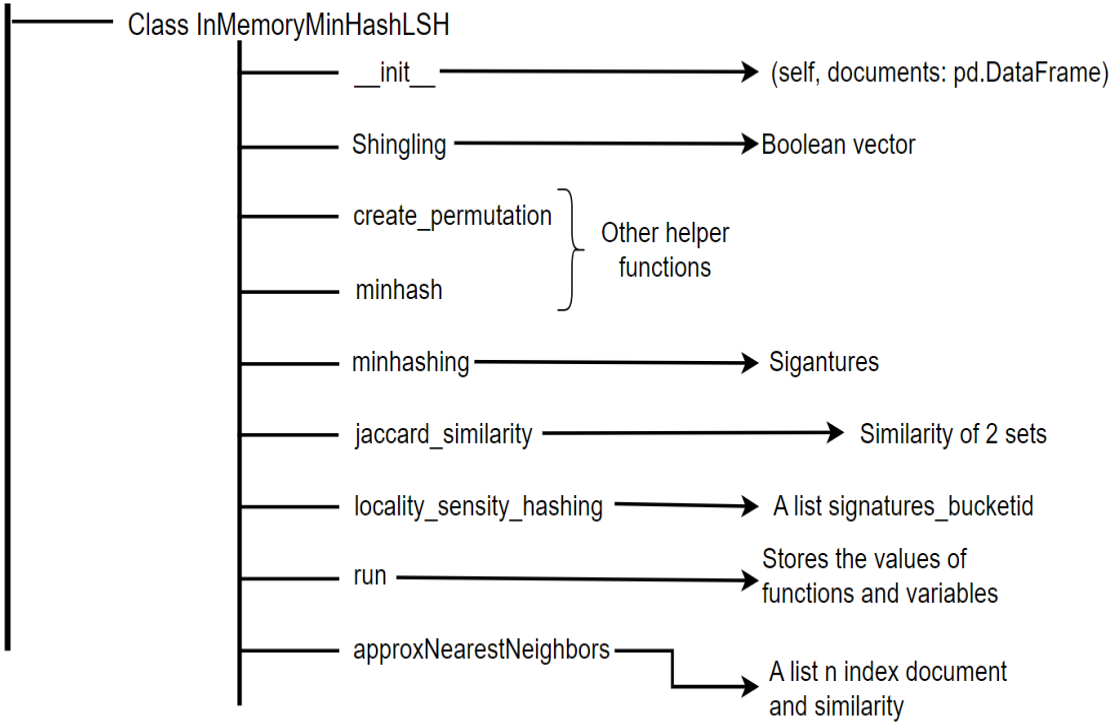
```
1 Function locality_sensity_hashing(self, signatures):
2     num_buckets <- 102233
3     r <- self.num_hashes divided by self.num_bands
4     num_docs <- number of columns in signatures
5     results <- an empty dictionary
6
7     for each band from 0 to self.num_bands-1 do
8         start_row <- band * r
9         end_row <- (band + 1) * r
10
11        for each column from 0 to num_docs-1 do
12            band_signature <- slice of signatures from start_row to end_row, column-wise
13            hash_value <- hash of the tuple formed by band_signature modulo num_buckets
14            signature_str <- concatenate all elements in band_signature
15            signature_hash_str <- concatenate signature_str and hash_value
16
17            if column not in results then
18                create an empty list in results for column
19
20            append signature_hash_str to results[column]
21
22     return results
```

Pseudocode for approxNearestNeighbors

```
1 Function approxNearestNeighbors(self, key, n):
2       textqueried_df <- DataFrame containing the key as a single text document
3       bool_vectors_queried <- shingling(textqueried_df)
4       signature_queried <- 2D array of zeros with dimensions (self.num_hashes, 1)
5
6       for i from 0 to self.num_hashes-1 do
7          permutation <- self.permutations[i]
8          signature_queried[i, 0] <- minhash(bool_vectors_queried[0], permutation)
9
10      lsh_signature_queried <- locality_sensity_hashing(signature_queried)
11      list_query_bucket_id <- Extract bucket IDs from lsh_signature_queried
12      candidate_docs <- empty list
13
14      for each doc_id, lsh_signatures in self.lsh_buckets.items() do
15         current_bucket_ids <- Extract bucket IDs from lsh_signatures
16         if any bucket_id in list_query_bucket_id matches current_bucket_ids then
17            append doc_id to candidate_docs
18
19      similarities <- empty list
20
21      for each doc_id in candidate_docs do
22         signature_set_queried <- Flatten signature_queried
23         lsh_signature_existing <- Flatten self.signatures[doc_id]
24         jaccard_similarity <- jaccard_similarity(signature_set_queried, lsh_signature_existing)
25         if jaccard_similarity >= self.similarity_threshold then
26            append (doc_id, jaccard_similarity) to similarities
27
28      Sort similarities in descending order based on similarity value
29      Return the top n similarities
```

In-memory MinHashLSH

Summary of how to solve the requirements of In-memory MinHashLSH



Output

```
[ ]  query = "Many recent publications highlight the large role of the pivotal eukary
     n = 3 # số lượng tài liệu tương tự
     document_similarity = minhash_lsh.approxNearestNeighbors(query, 3) #5720
     print(document_similarity)
```

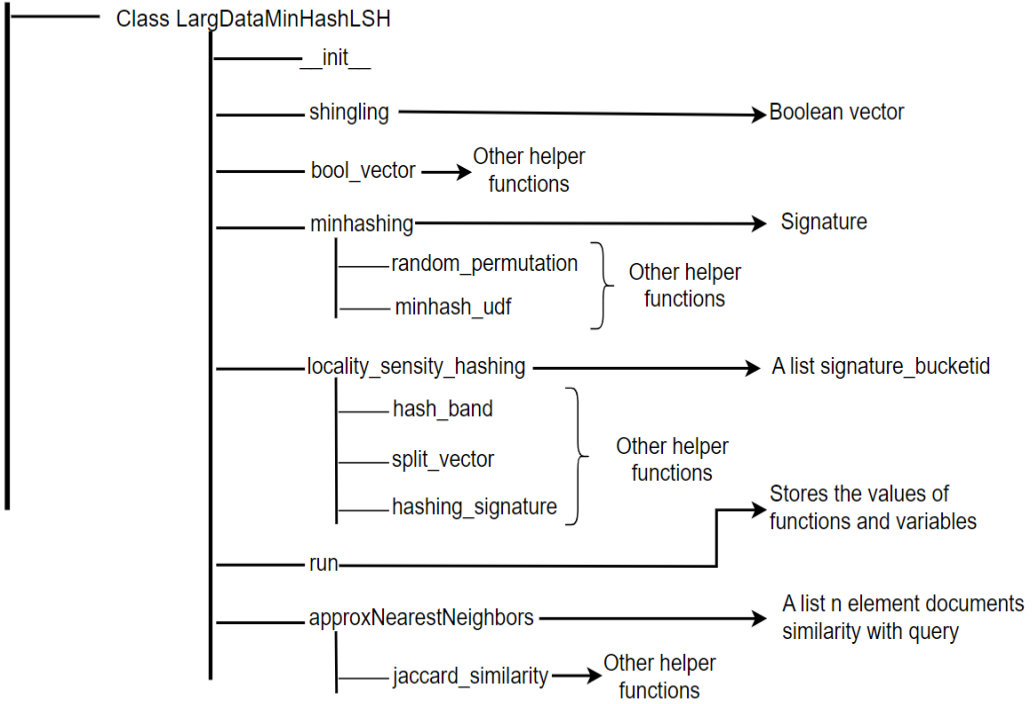
```
[(5720, 1.0), (1556, 0.66), (5678, 0.66)]
```

```
[ ]  query_1 = "Background: (-)-alpha-Bisabolol, also known as levomenol, is an unsat
     query_20 = "3D printing has shown promise for neural regeneration by providing c
     query = query_1+query_20
     n = 5 # số lượng tài liệu tương tự
     document_similarity = minhash_lsh.approxNearestNeighbors(query, 5) #1, 20
     print(document_similarity)
```

```
[(1, 0.89), (20, 0.66), (3885, 0.65), (5238, 0.6), (710, 0.59)]
```

LargDataMinHashLSH

Summary of how to solve the requirements of LargDataMinHashLSH



Output

```
query_document = "Many recent publications highlight the large role of the pivotal eu
df_with_similarity = large_data_minhash_LSH.approxNearestNeighbors(query_document, 5)
```

```
+--------------------+-----------------+
|            document|jaccard_similarity|
+--------------------+-----------------+
|Many recent publi...|              1.0|
|The integration o...|             0.66|
|Six different com...|             0.65|
|Protein-protein i...|             0.64|
|Quantitative PCR(...|             0.64|
+--------------------+-----------------+
```

Output

```
query_1 = "Background: (-)-alpha-Bisabolol, also known as levomenol, is an unsat
query_20 = "3D printing has shown promise for neural regeneration by providing c
key = query_1+query_20
df_with_similarity = large_data_minhash_LSH.approxNearestNeighbors(key, 3)
```

```
+--------------------+------------------+
|            document|jaccard_similarity|
+--------------------+------------------+
|Background: (-)-a...|              0.84|
|3D printing has s...|               0.6|
+--------------------+------------------+
```

Advantages versus disadvantages

Advantages:

- Rich documentations
- Diverse perspectives
- Collaboration and synergy
- Reduced workload

Disadvantages:

- Coordination and communication challenges
- Differences in work styles and commitment levels
- Information disorder

REFERENCES

[1] "Locality Sensitive Hashing (LSH): The Illustrated Guide | Pinecone," *www.pinecone.io*. [Online]. Available: https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/.

[2] "Locality-sensitive hashing," *Wikipedia*, Nov. 11, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Locality-sensitive_hashing.

[3] "Finding similar documents," *Fast Data Science*, Nov. 09, 2021. [Online]. Available: https://fastdatascience.com/finding-similar-documents-nlp/ (accessed Apr. 27, 2024).

[4] A. R. J. U. Jure Leskovec, "Stanford University," 10 March 2024. [Online]. Available: http://www.mmds.org.

[5] A. R. J. U. Jure Leskovec, *Mining of Massive Datasets*, California: Cambridge University Press, 2014.

[6] J. a. R. A. a. U. J. D. Leskovec, *Mining of Massive Datasets*, USA: Cambridge University Press, 2014.

Task assigments

ID	FULL NAME	EMAIL	TASK	COMPLETE
521H0290	Do Minh Quan	521h0290@student.tdtu.edu.vn	1, 3	100%
521H0489	Ho Huu An	521h0489@student.tdtu.edu.vn	2, 3	100%
521H0287	Van Cong Nguyen Phong	521h0287@student.tdtu.edu.vn	2, 3	100%
521H0514	Nguyen Le Phuoc Tien	521h0514@student.tdtu.edu.vn	1, 3	100%

Self-assesment

TASKS	COMPLETE PERCENTAGE
Task 1	100%
Task 2	100%
Task 3	100%

Thanks for listening!