TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP GIỮA KÌ MÔN XỬ LÝ DỮ LIỆU LỚN**

**Locality Sensitive Hashing**

*Người hướng dẫn*: **PhGS LÊ ANH CƯỜNG**

*Người thực hiện*: **TRẦN PHƯƠNG NGỌC ANH – 51702061**

**HUỲNH ANH TÀI – 51702171**

**NGUYỄN TẤN RU BY – 51703050**

Khoá **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP GIỮA KÌ MÔN XỬ LÝ DỮ LIỆU LỚN**

**Locality Sensitive Hashing**

Người hướng dẫn: **PhGS LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN PHƯƠNG NGỌC ANH – 51702061**

**HUỲNH ANH TÀI – 51702171**

**NGUYỄN TẤN RU BY – 51703050**

**Khoá : 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

Trong quá trình hoàn thành bài tiểu luận, bên cạnh sự nỗ lực và cố gắng của bản thân không thể không nhắc đến sự quan tâm và hướng dẫn tận tình của quý thầy cô cố vấn. Chúng em xin gửi lời cảm ơn chân thành đến thầy Lê Anh Cường, người đã dùng vốn tri thức cùng những kinh nghiệm quý báu của mình để giúp đỡ chúng em trong suốt quá trình học tập. Nhờ những lời hướng dẫn và dạy bảo tận tình đó, chúng em đã tích lũy thêm cho mình không ít kiến thức cùng sự hiểu biết để hoàn thành bài báo cáo một cách hiệu quả nhất.

Do vốn kiến thức vẫn còn hạn chế nên trong quá trình thực hiện bài báo cáo vẫn không thể tránh khỏi có những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp của thầy để bài báo cáo được hoàn thiện hơn.

Một lần nữa chúng em xin chân thành cảm ơn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của PhGS Lê Anh Cường;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 14 tháng 03 năm 2021*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Phương Ngọc Anh*

*Huỳnh Anh Tài*

*Nguyễn Tấn Ru By*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Ngày nay, chúng ta đang sống trong một thế giới được điều khiển bởi kỹ thuật số, do đó mọi doanh nghiệp đều nghiên cứu dữ liệu lớn để thu được những hiểu biết có giá trị từ lượng dữ liệu thô khổng lồ. Khối lượng của các dữ liệu này là vô cùng lớn và đa dạng. Công cụ phân tích dữ liệu lớn cần phải hiểu được khối lượng dữ liệu khổng lồ này và chuyển nó thành thông tin chi tiết có giá trị. Kích thước của dữ liệu càng lớn thì vấn đề càng lớn, do đó nhu cầu khai thác dữ liệu đang đóng một vai trò rất quan trọng trong việc quản lý các kho dữ liệu nói chung và xử lý các văn bản dưới dạng văn bản nói riêng.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc66688733)

[TÓM TẮT iv](#_Toc66688734)

[MỤC LỤC 1](#_Toc66688735)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 2](#_Toc66688736)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc66688737)

[CHƯƠNG 1 – GIỚI THIỆU 4](#_Toc66688738)

[CHƯƠNG 2 – LOCALITY SENSITIVE HASHING 6](#_Toc66688739)

[2.1 Shingling 7](#_Toc66688740)

[2.2 Jaccard Index 8](#_Toc66688741)

[2.3 Min Hashing 9](#_Toc66688742)

[2.4 Locality Sensitive Hashing 12](#_Toc66688743)

[CHƯƠNG 3 – NGHIÊN CỨU THỰC NGHIỆM 15](#_Toc66688744)

[TÀI LIỆU THAM KHẢO 16](#_Toc66688745)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

LSH Locality Sensitive Hashing

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1.1: Hàm băm (Hash function) 4](#_Toc66687676)

[Hình 2.1: Locality Sensitive Hashing 6](#_Toc66687680)

[Hình 2.2: Các bước của Locality Sensitive Hashing 6](#_Toc66687681)

[Hình 2.3: Ma trận Shingles x Document 8](#_Toc66687682)

[Hình 2. 4: Công thức Jaccard Index 8](#_Toc66687683)

[Hình 2.5: Min Hashing với ma trận 1 cột 10](#_Toc66687684)

[Hình 2.6: Công thức hàm băm 10](#_Toc66687685)

[Hình 2.7: Chuyển đổi ma trận chữ ký với tập hoán vị 1 cột 11](#_Toc66687686)

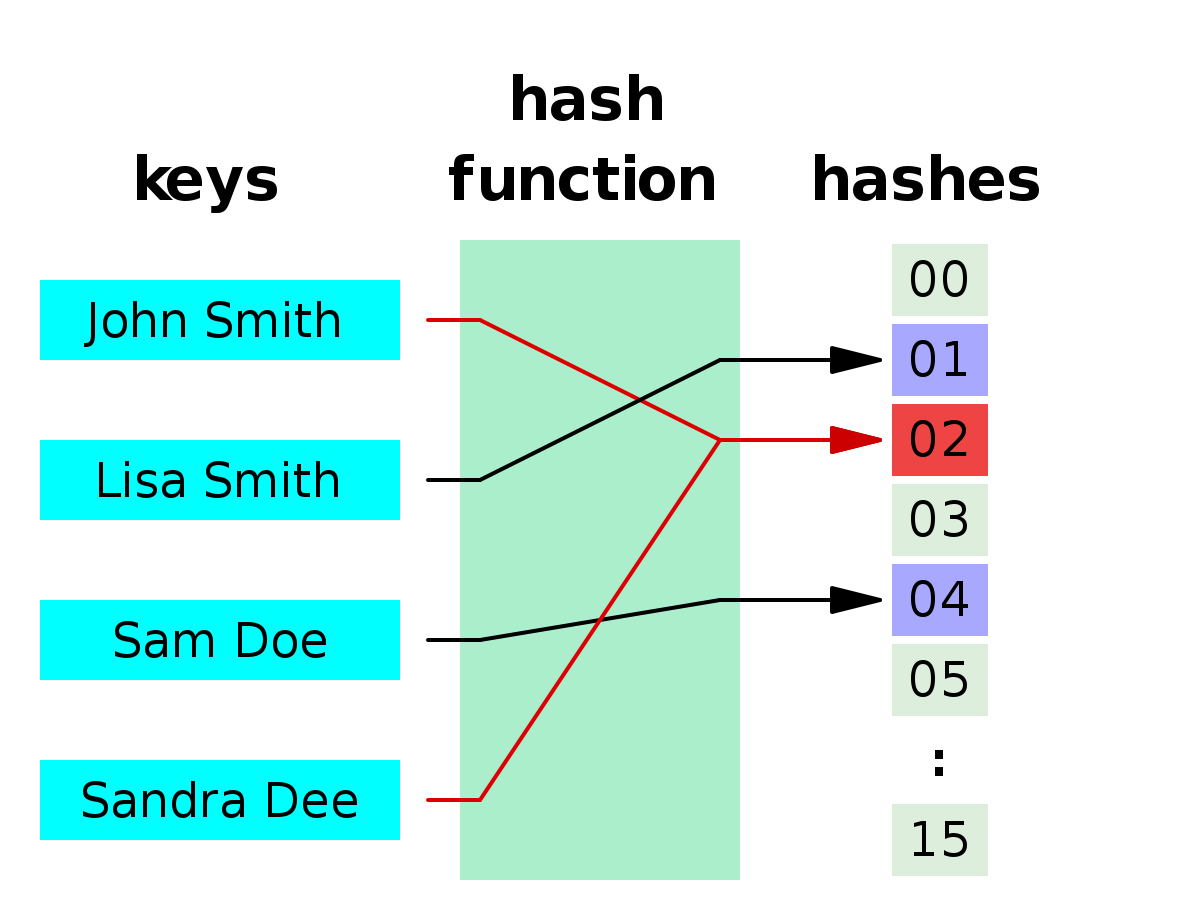
[Hình 2.8: Chuyển đổi ma trận chữ ký với tập hoán vị nhiều cột 12](#_Toc66687687)

[Hình 2.9: Ma trận các dải băng 14](#_Toc66687688)

CHƯƠNG 1 – GIỚI THIỆU

Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật, cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể, thống kê và phân tích dữ liệu. Học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Đối với các dữ liệu lớn, việc tìm ra cách giải quyết cho bài toán là một vấn đề vô cùng quan trọng.

Hàm băm (hash function) là giải thuật nhằm sinh ra các giá trị băm tương ứng với mỗi khối dữ liệu (có thể là một chuỗi ký tự, một đối tượng trong lập trình hướng đối tượng, v.v…). Hàm băm thường được dùng trong bảng băm nhằm giảm chi phí tính toán khi tìm một khối dữ liệu trong một tập hợp (nhờ việc so sánh các giá trị băm nhanh hơn việc so sánh những khối dữ liệu có kích thước lớn), ứng dụng rộng rãi trong việc tìm kiếm, truy xuất dữ liệu trên database có kích thước lớn. Giá trị băm đóng vai trò phân biệt các khối dữ liệu, tuy nhiên, các giá trị này không tránh khỏi những sự trùng lặp, đụng độ. [1]



Hình 1.1: Hàm băm (Hash function)

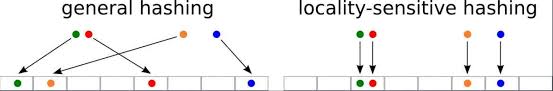
(Nguồn: https://en.wikipedia.org/wiki/Hash\_function)

Trong khoa học máy tính, Locality Sensitive Hashing (LSH) là một kỹ thuật thuật toán băm các giá trị tương tự vào cùng một "nhóm" với xác suất cao. Vì các giá trị tương tự kết thúc trong cùng một nhóm nên kỹ thuật này có thể được sử dụng để phân cụm dữ liệu và tìm kiếm hàng xóm gần nhất. Nó khác với các kỹ thuật băm thông thường ở chỗ các xung đột băm được tối đa hóa chứ không phải giảm thiểu. Ngoài ra, kỹ thuật này có thể được coi là một cách để giảm kích thước của dữ liệu; các giá trị đầu vào có thể được giảm xuống thành các phiên bản có nhỏ gọn hơn trong khi vẫn giữ được khoảng cách tương đối giữa các chúng.

CHƯƠNG 2 – LOCALITY SENSITIVE HASHING

Bài toán tìm kiếm những phần tử gần nhất hoặc giống nhau là một bài toán phổ biến hiện nay như là tìm kiếm những đoạn âm thanh, video hoặc đoạn văn mà giống nhau. Chúng ta có thể thực hiện việc đi kiểm tra từng cái với nhau nhưng điều đó sẽ gây ra mất thời gian rất lớn nếu tập dữ liệu của ta lớn. Chính vì vậy Locality Sensitive Hashing được ra đời để giúp ta tìm ra những phần tử gần giống nhau nhất.

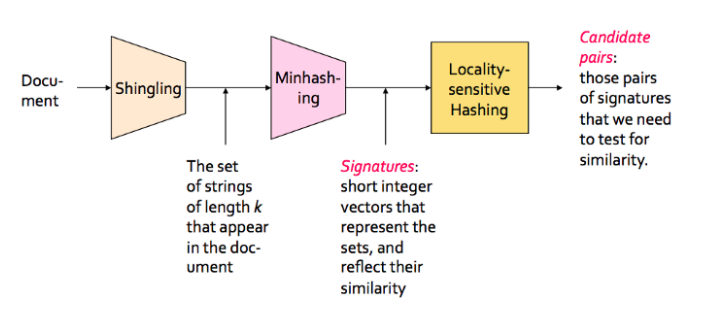
Ý tưởng của Locality Sensitive Hashing như sau: thuật toán sẽ băm (hash) các điểm dữ liệu và các điểm dữ liệu giống nhau sẽ được gom nhóm lại với nhau. Điều này sẽ giúp dễ dàng xác định những tập dữ liệu giống nhau.



Hình 2.1: Locality Sensitive Hashing

(Nguồn: https://whytin.github.io/cs584\_s16/slides/lsh-11.pdf)

Các bước của Locality Sensitive Hashing:



Hình 2.2: Các bước của Locality Sensitive Hashing

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

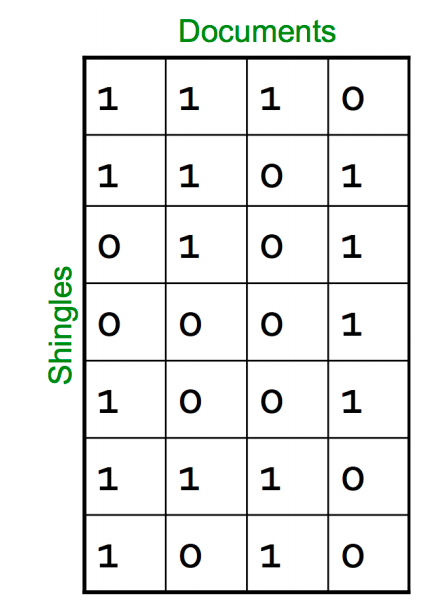
1. Shingling

Cách hiệu quả nhất để biểu diễn tài liệu dưới một dạng tập hợp cho việc xác định các tài liệu tương tự nhau thì chúng ta nên làm một việc là xây dựng các chuỗi ngắn xuất hiện bên trong các document đó để xác định các cụm được chia ra có nằm trong các document khác không. Cho nên ở bước này chúng ta sẽ chuyển đổi các document thành các tập ký tự có độ dài là k. Và ý nghĩa của bước này là chúng ta sẽ biến đổi từng document trong tập dataset của ta trở một tập có k-shingles.

Ví dụ: Với từ “Anh” và k = 2 thì ta sẽ tách thành {An, Ah, nh}

Thông thường ta sẽ chọn k nằm từ giá trị 8-10 vì nếu quá nhỏ thì nó sẽ tạo ra những cụm từ ngắn khó biểu diễn thành một dạng document.

Và để thuận tiện cho việc so sánh giữa các document, ta sẽ chuyển thành dạng ma trận với giá trị 0 hoặc 1. Các cột (column) tượng trưng cho tập các document, còn các hàng (row) tượng trưng cho các tập k-shingles được tạo ra từ các document đó. Nếu có 1 tập k-shingles nằm trong document nào thì ở hành k-shingles và cột document đó được biểu diễn là 1 còn ngược lại là 0.

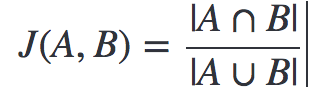


Hình 2.3: Ma trận Shingles x Document

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

1. Jaccard Index

Sau khi trình bày document dưới các dạng ma trận trên thì chúng ta cần một công thức để tính độ giống nhau giữa các document. Jaccard Index là công thức được áp dụng để thực hiện việc đó.



Hình 2. 4: Công thức Jaccard Index

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

Với A, B là các document.

Ví dụ ta có A: “Nadal” và B: “Nadia” và ta sẽ tách thành tập 2-shingles là {Na, ad, da, al} và {Na, ad, di, ia}. Từ đó ta có thể thấy phần A hợp với B = 6 và phần A giao với B là 2 nên J(A,B) = 2/6. Nếu J(A,B) càng lớn thì độ giống nhau giữa các document càng lớn và ngược lại.

Ở bước này ta đã xác định được các document giống nhau tuy nhiên vẫn còn hai vấn đề như sau:

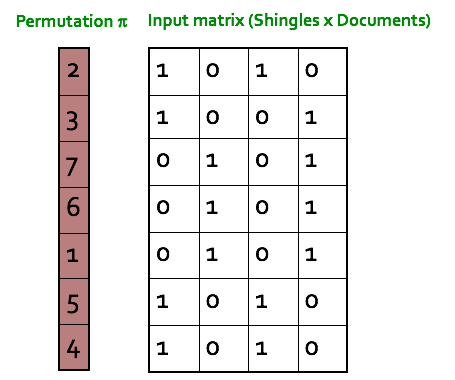
* Mất nhiều thời gian thực thi nếu như tập dataset lớn.
* Nếu lưu trữ một ma trận ở trên với dạng dataset lớn thì sẽ gây ra hao phí.

Để tránh hai vấn đề trên ta sẽ dùng hashing (hàm băm). Ý nghĩa hashing của trường hợp này như sau: ta sẽ chuyển đổi dạng tài liệu ma trận trên thành các chữ ký bằng cách băm, ta không so sánh từng document với nhau nữa mà so sánh từng chữ ký của các document với nhau, điều đó sẽ giúp thời gian chạy nhanh hơn và giảm được tổn thất hao phí.

1. Min Hashing

Các bước để thực hiện Min Hashing:

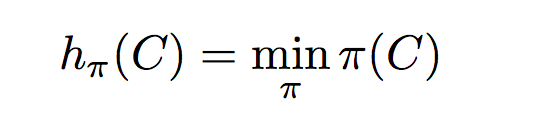
Bước 1: Chọn ra một tập hoán vị ngẫu nhiên (π).



Hình 2.5: Min Hashing với ma trận 1 cột

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

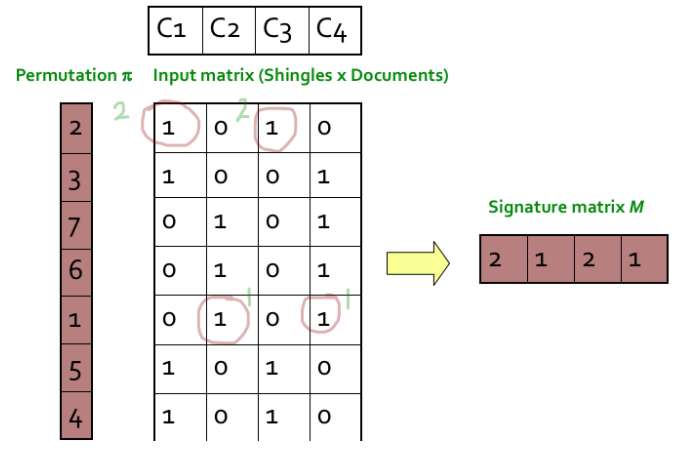
Bước 2: Thực hiện hàm hash bằng cách so sánh các giá trị của mỗi ô (chỉ tính ô có giá trị là 1) trong một cột sau khi map đối xứng với tập hoán vị ngẫu nhiên và chọn ra số nhỏ nhất. Và cứ làm vậy cho đến khi hết từng cả các cột thì ta sẽ tạo ra tập ký hiệu (chữ ký) cho mỗi cột document.



Hình 2.6: Công thức hàm băm

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

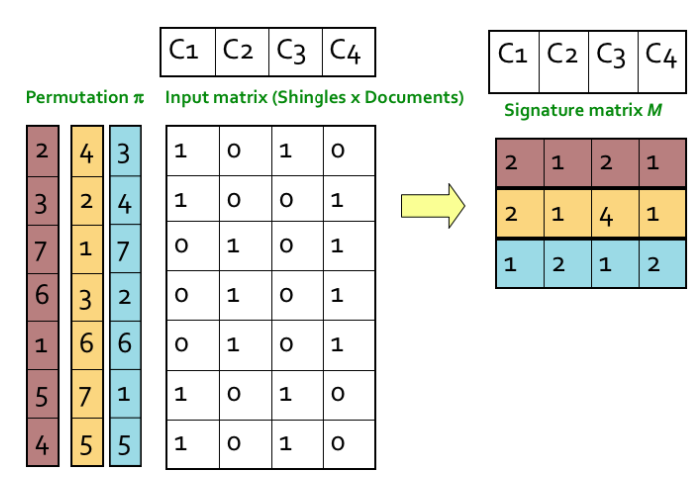
Với công thức trên, ta sẽ có tập ma trận ký hiệu lần lượt cho mỗi cột là 2, 1, 2, 1. Như vậy, từ ma trận document rất dài đã được thu gọn lại thành các ký hiệu với kích thước nhỏ hơn và dễ nhìn hơn.



Hình 2.7: Chuyển đổi ma trận chữ ký với tập hoán vị 1 cột

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

Cách làm tương tự với trường hợp nếu ta chọn ra một tập hoán vị với ngẫu nhiên với nhiều cột.



Hình 2.8: Chuyển đổi ma trận chữ ký với tập hoán vị nhiều cột

(Nguồn: https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134)

Khi đó ta có thể áp dụng Jaccard Index để so sánh tỉ lệ giống nhau giữa các document thông qua ký hiệu (chữ ký) của chúng. Như ở ví dụ trên, ta tính được kết quả của document C2 và C4 là 1 và ở ma trận đầu kết quả là 3/4. Từ đó có thể thấy sau khi biến đổi sang ma trận ký hiệu thì tính tương tự trên mỗi document vẫn không thay đổi nhiều dựa trên hai kết quả đã tính và từ ma trận ký hiệu ta cũng có thể thấy ta đã loại bỏ đi những khoảng trống dư thừa ở ma trận ban đầu (những ô có giá trị 0), điều đó giúp ta loại bỏ những không gian dư thừa đi rất nhiều.

1. Locality Sensitive Hashing

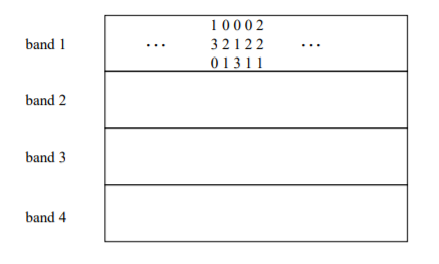
Mặc dù chúng ta có thể dùng phương pháp min hashing để nén các document lớn thành các ký hiệu nhỏ và bảo toàn độ giống nhau nhưng vẫn chưa thể tìm thấy các cặp giống nhau một cách hiệu quả nhất bởi vì có thể lượng số cặp các document rất lớn sẽ làm hao phí rất nhiều tài nguyên.

Mục tiêu của chúng ta là tính toán xem sự tương đồng ở mỗi cặp, thực chất, ta chỉ quan tâm đến những cặp giống nhau nhất hoặc các cặp có mức độ tương tự nhau trên một ngưỡng nào đó. Vì vậy, ta cần tập trung vào những cặp gần như giống nhau thay vì rà soát từng cặp một. Locality Sensitive Hashing (LSH), hay near-neighbor search chính là một giả thuyết cho vấn đề đó.

Cách tiếp cận chung đối với LSH là việc băm các items nhiều lần do đó các items tương tự nhau có khả năng được băm vào cùng một nhóm (bucket), cũng như mong muốn rằng những cặp đôi không giống nhau sẽ không bao giờ được băm vào một nhóm và không cần phải kiểm tra. Chúng ta gọi các cặp được băm vào cùng nhóm là candidate pair (cặp ứng cử viên) và kiểm tra sự giống nhau của chúng. Các cặp khác nhau được phân vào cùng một nhóm được gọi là false positives và hy vọng rằng chúng sẽ chỉ chiếm một phần nhỏ trong các cặp. Các cặp tương tự nhau nhưng không được xếp chung nhóm được gọi là false negatives, ta cũng mong muốn đây chỉ là phần nhỏ giữa các cặp đôi giống nhau.

Đối với các chữ ký minhash, một cách hiệu quả chính là chia ma trận ký hiệu thành b băng gồm r dòng mỗi băng. Đối với mỗi băng sẽ có một hàm băm với các vector của r số nguyên (phần trong một cột của băng đó) và băm chúng thành các số để xếp vào nhóm. Ta có thể sử dụng cùng hàm băm cho tất cả các băng, vì vậy các cột có cùng vector nhưng ở khác băng sẽ không được xếp vào cùng một nhóm.

Hình 2.9 biểu diễn một ma trận các ký hiệu gồm 12 dòng được chia thành 4 băng với 3 dòng mỗi băng. Cột 2 và 4 ở băng đầu tiên đều có vector [0, 2, 1], do đó chúng sẽ được băm vào cùng một nhóm, cũng vì vậy, cho dù các nhóm còn lại có các cột như thế nào thì 2 cột này vẫn sẽ là một candidate pair. Các cột 1 và 2 cũng được băm vào cũng một nhóm, tuy nhiên, vì vector của chúng khác nhau: [1, 3, 0] và [0, 2, 1], tiếp đó còn có rất nhiều nhóm cho mỗi lần băm nên ta có thể thấy rằng xác suất gặp nhau ngẫu nhiên giữa chúng cũng là rất nhỏ. 2 cột không đồng nhất này sẽ có 3 cơ hội khác tương ứng với 3 băng còn lại để tìm kiếm nửa kia của chúng.



Hình 2.9: Ma trận các dải băng

(Nguồn: http://infolab.stanford.edu/~ullman/mmds/ch3.pdf)

Nhìn chung, cách phân chia thành các dải băng này có khả năng tìm ra được các candidate pair khi chúng được băm vào cùng một nhóm với ít nhất 1 băng.

Tóm lại, ý tưởng của LSH là sử dụng một hàm băm để băm các điểm dữ liệu vào từng nhóm sao cho các điểm dữ liệu gần nhau sẽ có xác suất cao rơi vào cùng một nhóm, ngược lại chúng sẽ bị băm vào hai nhóm khác nhau. Do đó, việc tìm kiếm các văn bản tương tự trong tập văn bản cũng dựa trên phương pháp này. [2]

CHƯƠNG 3 – NGHIÊN CỨU THỰC NGHIỆM

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Wikipedia (2020), *Hàm Băm*.

https://vi.wikipedia.org/wiki/H%C3%A0m\_b%C4%83m

**Tiếng Anh**

1. Stanford University (2021), [Ch3: Finding Similar Items](http://infolab.stanford.edu/~ullman/mmds/ch3.pdf) (Sect. 3.1-3.4), *Locality-Sensitive Hashing I*

http://infolab.stanford.edu/~ullman/mmds/ch3.pdf