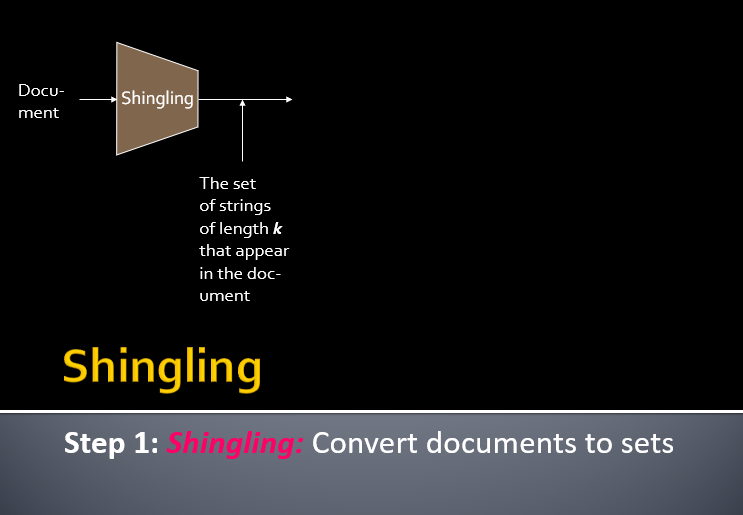
**MINING OF MASSIVE DATASET**



**STEP 1: SHINGLING: CONVERT DOCUMENTS TO SETS**

**TÀI LIỆU DƯỚI DẠNG DỮ LIỆU HIGH-DIM**

Các cách tiếp cận đơn giản:

* Tài liệu = tập hợp các từ xuất hiện trong tài liệu
* Tài liệu = tập hợp các từ "quan trọng"
* Không hoạt động tốt cho ứng dụng này. Tại sao?

Chú ý: Cần phải tính đến thứ tự của các từ!

Một cách khác: **Shingles!**

**ĐỊNH NGHĨA: SHINGLES**

K-shingle (hoặc k-gram) cho 1 tài liệu là một chuỗi các tokens k xuất hiện trong tài liệu

* Token có thể là ký tự, từ ngữ hoặc thứ gì đó khác, tùy thuộc vào ứng dụng
* Giả sử tokens = ký tự cho ví dụ

**Ví dụ:** k =2; tài liệu D1 = abcab  
Tập hợp 2-shingles: S(D1) = {ab, bc, ca}

Tùy chọn: shingles dưới dạng 1 cái túi (nhiều bộ), đếm ab hai lần: S'(D1) = {ab, bc, ca, ab}

**NÉN SHINGLES**

Để **nén shingles dài**, chúng ta có thể **băm** chúng thành 4 bytes

Đại diện cho một tài liệu bằng tập các giá trị băm của k-shingles

* Ý tưởng: Hai tài liệu có thể (hiếm khi) có shingles chung, trong khi trên thực tế chỉ có các giá trị băm được chia sẻ

**Ví dụ:** k=2; tài liệu D1= abcab

Tập của 2-shingles: S(D1) = {ab, bc, ca}

Băm singles: h(D1) = {1, 5, 7}

**METRIC TƯƠNG TỰ CHO SHINGLES**

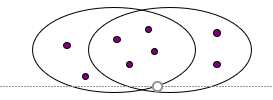
Tài liệu D1 là một tập các k-shingles của nó C1= S(D1)

Tương đương, mỗi tài liệu là một vectơ 0/1 trong không gian k-shingles

* Với shingle độc nhất là một chiều không gian
* Các vectơ rất thưa thớt

**Một biện pháp tự nhiên tương tự là sự tương đồng Jaccard:**

sim(D1,D2) -| C1∩C2|/| C1∪C2 |



**GIẢ ĐỊNH CÔNG VIỆC**

Các Tài liệu có nhiều shingles chung thì có văn bản tương tự, ngay cả khi thứ tự văn bản xuất hiện theo cách khác nhau

Cảnh báo: Bạn phải chọn **k** đủ lớn, hoặc hầu hết các tài liệu sẽ có shingles

* k = 5 là OK “ổn” cho các tài liệu ngắn
* k = 10 là better “tốt hơn” cho các tài liệu dài

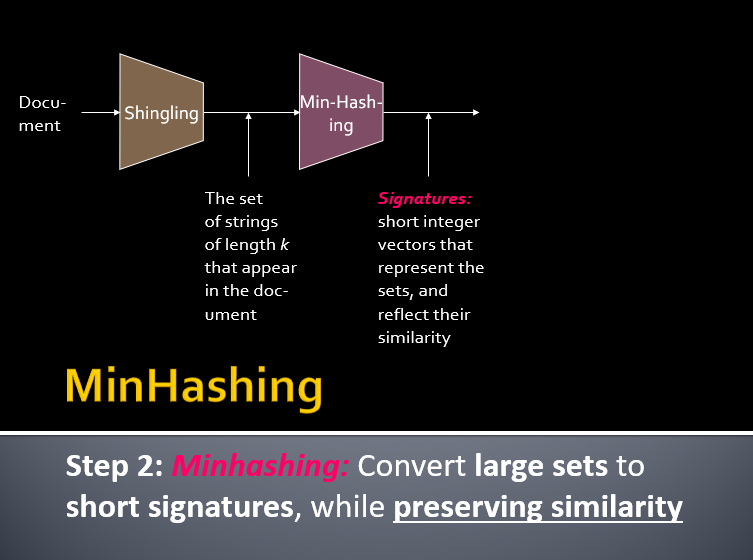
**ĐỘNG LỰC CHO MINHASH / LSH**

Giả sử chúng ta cần tìm các tài liệu gần trùng lặp trong số hàng triệu tài liệu

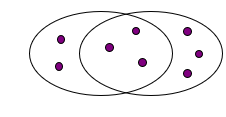
Chúng ta sẽ phải tính toán **pairwise Jaccard similarities (các cặp tương đồng Jaccard)** cho mỗi cặp tài liệu

* ≈ 5\*1011 so sánh
* Với tốc độ 105 giây/ngày và 106 so sánh/giây, nó sẽ mất **5 ngày**

Đối với **N = 10** hàng triệu, phải mất hơn một năm



**MÃ HÓA SETS DƯỚI DẠNG BIT VECTORS**

Nhiều vấn đề tương tự có thể được chính thức hóa như **tìm các tập hợp con có giao điểm quan trọng**

Bộ mã hóa sử dụng vectơ 0/1 (bit, boolean)

* Một kích thước trên mỗi phần tử trong tập hợp phổ quát

Chứng minh đặt giao điểm (intersection) theo bitwise AND và đặt phép hợp (union) theo bitwise OR

**Ví dụ:** C1 = 10111; C2 = 10011

* Kích thước intersection = 3; kích thước của union = 4,
* **Jaccard** **similarity** (không phải khoảng cách) = 3/4
* **Khoảng cách**: d(C1,C2) = 1 – (**Jaccard** **similarity**) = 1/4

**TỪ SETS ĐẾN BOOLEAN MATRICES**

Hàng = phần tử (shingles)

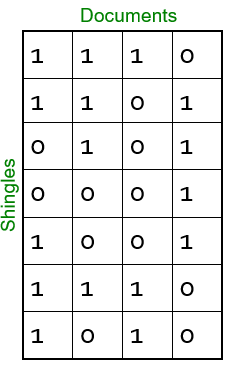
Cột = tập hợp (tài liệu)

* 1 trong hàng e và cột s nếu và chỉ khi e là thành viên của s
* Tương tự cột là Jaccard similarity của các tập hợp tương ứng (hàng có giá trị 1)
* Ma trận điển hình là ma trận thưa thớt!

**Mỗi tài liệu là một cột:**

**Ví dụ:** sim(C1, C2) = ?

* Kích thước intersection = 3; kích thước của union = 6, **Jaccard** **similarity** (không phải khoảng cách) = 3/6
* d(C1,C2) =1 – (**Jaccard** **similarity**) = 3/6



**OUTLINE: TÌM CÁC CỘT TƯƠNG TỰ**

**Cho đến nay:**

* Tài liệu → Tập hợp cỉa shingles
* Đại diện cho các tập hợp dưới dạng vectơ boolean trong ma trận

**Mục tiêu tiếp theo**: Tìm các cột tương tự trong khi tính toán chữ ký nhỏ

* Sự giống nhau của cột == sự giống nhau của chữ ký

**Mục tiêu tiếp theo**: Tìm các cột tương tự, Chữ ký nhỏ

Cách tiếp cận **Naïve**:

1) Chữ ký của các cột: tóm tắt nhỏ các cột

2) Kiểm tra các cặp chữ ký để tìm các cột tương tự

*Điều cần thiết*: Điểm tương đồng của chữ ký và cột có liên quan

3) Tùy chọn: Kiểm tra xem các cột có chữ ký tương tự có thực sự giống nhau không

Cảnh báo:

So sánh tất cả các cặp có thể tốn quá nhiều thời gian: **Công việc cho LSH**

* Các phương pháp này có thể tạo ra phủ định sai và thậm chí là khẳng định sai (nếu kiểm tra tùy chọn không được thực hiện)

**CỘT BĂM (CHỮ KÝ)**

**Ý tưởng chính**: "băm" mỗi cột **C** đến một **chữ ký nhỏ** **h(C),** sao cho:

(1) h(C) đủ nhỏ để chữ ký phù hợp với RAM

(2) sim(C1, C2) giống như "sự tương đồng" của chữ ký h(C1) và h(C2)

**Mục tiêu**: Tìm hàm băm h(·) như vậy:

* Nếu sim (C1,C2) cao, thì với prob cao**. h(C1) = h(C2)**
* Nếu sim (C1,C2) thấp, thì với prob cao. **h(C1) ≠ h(C2)**

Băm docs vào buckets. Mong đợi rằng "hầu hết" các cặp tài liệu gần trùng lặp băm vào cùng một bucket!

**MIN-HASHING**

**Mục tiêu**: Tìm hàm băm h(·) như vậy:

* Nếu sim (C1,C2) cao, thì với prob cao. **h(C1) = h(C2)**
* Nếu sim (C1,C2) thấp, thì với prob cao**. h(C1) ≠ h(C2)**

Rõ ràng, hàm băm phụ thuộc vào metric tương tự:

Không phải tất cả các chỉ số tương tự đều có hàm băm phù hợp

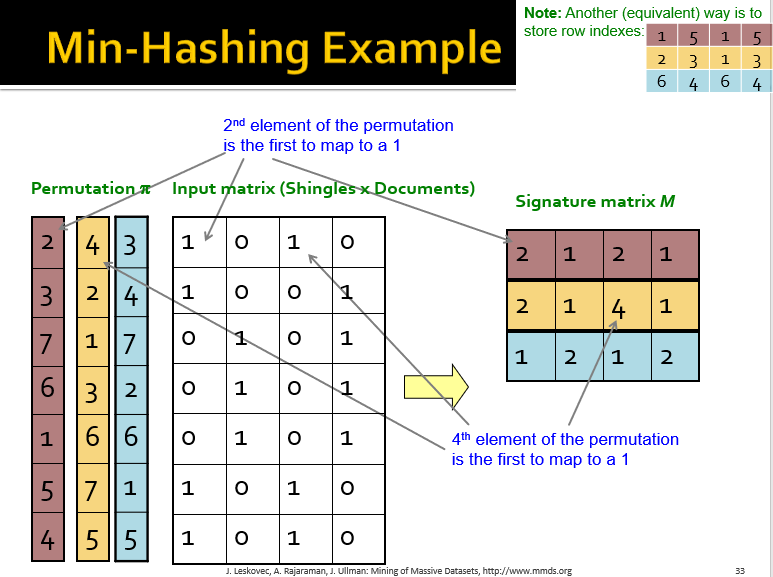
Có một hàm băm phù hợp cho **Jaccard similarity**: Nó được gọi là **Min-Hashing**

Hãy tưởng tượng các hàng của ma trận boolean được hoán vị dưới dạng hoán vị ngẫu nhiên π

Xác định hàm "băm" hπ(C) = chỉ mục của hàng đầu tiên (theo thứ tự hoán vị π) trong đó cột C có giá trị là 1:

hπ(C) = minππ(C)

Sử dụng một số hàm băm độc lập (nghĩa là hoán vị) để tạo chữ ký của một cột



Chọn hoán vị ngẫu nhiên π

Yêu cầu bồi thường: Pr[hπ(C1)- hπ(C2)]- sim(C1,C2)

Tại sao?

Hãy để X là một doc (bộ bệnh zona), y X là một∈ shingle

Sau đó: Pr[(y) = min((X))] = 1/| X|ππ

Cũng có khả năng bất kỳ y X nào ∈ được ánh xạ đến phần tử min

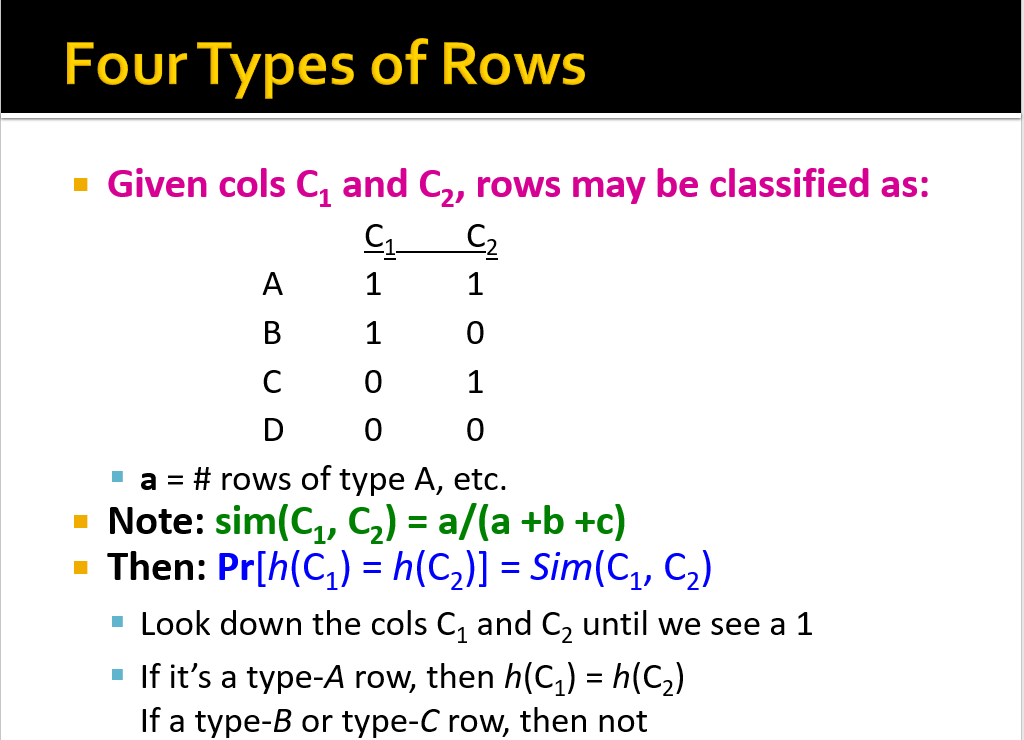
Hãy để y được s.t. (y) = min(π(C1∪C2))

Sau đó: π (y) = min((Cπ1))nếu y C∈1 , hoặc

π (y) s min ((Cπ2))nếu và C∈2

Vì vậy, prob. rằng cả hai đều đúng là prob. y ∈ C1 ∩ C2

Pr[min(π(C1))=min(π(C2))]=|C1∩C2|/|C1∪C2|= sim(C1, C2)



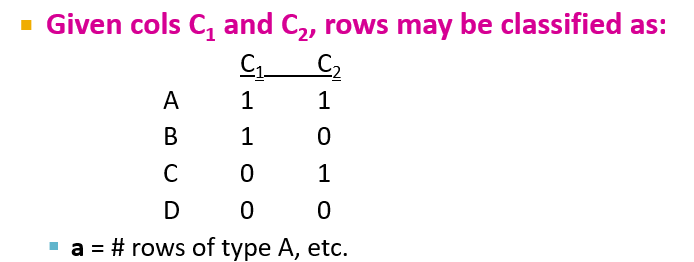
Nhìn vào cột C1 và C2 cho tới khi thấy số 1

Nếu là hàng loại A, thì h(C1) = h(C2)

Nếu là hàng loại B hoặc C thì không

Chữ kí tương tự:

Ta biết : Pr[hπ(C1) = hπ(C2)] = sim(C1, C2)



Ghi chú : sim(C1,C2) =a/(a+b+c)

Sau đó : Pr[h(C1) = h(C2)] = Sim(C1, C2)

o Nhìn xuống cols C1 và C2 cho đến khi chúng ta thấy A1

o Nếu đó là hàng loại A thì h(C1)=h(C2)

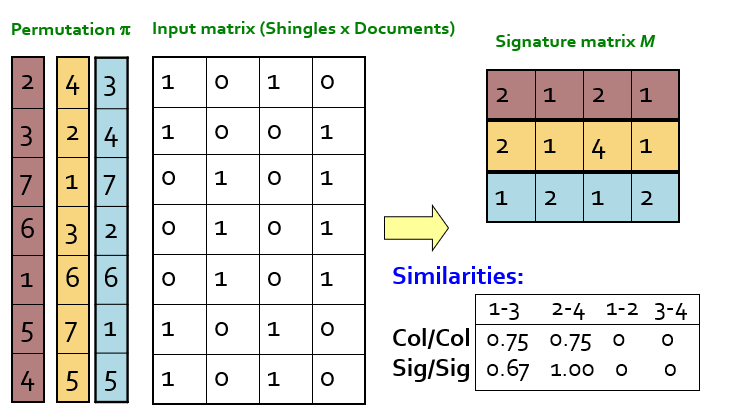
o Nếu là hàng loại B hoặc loại C thì không

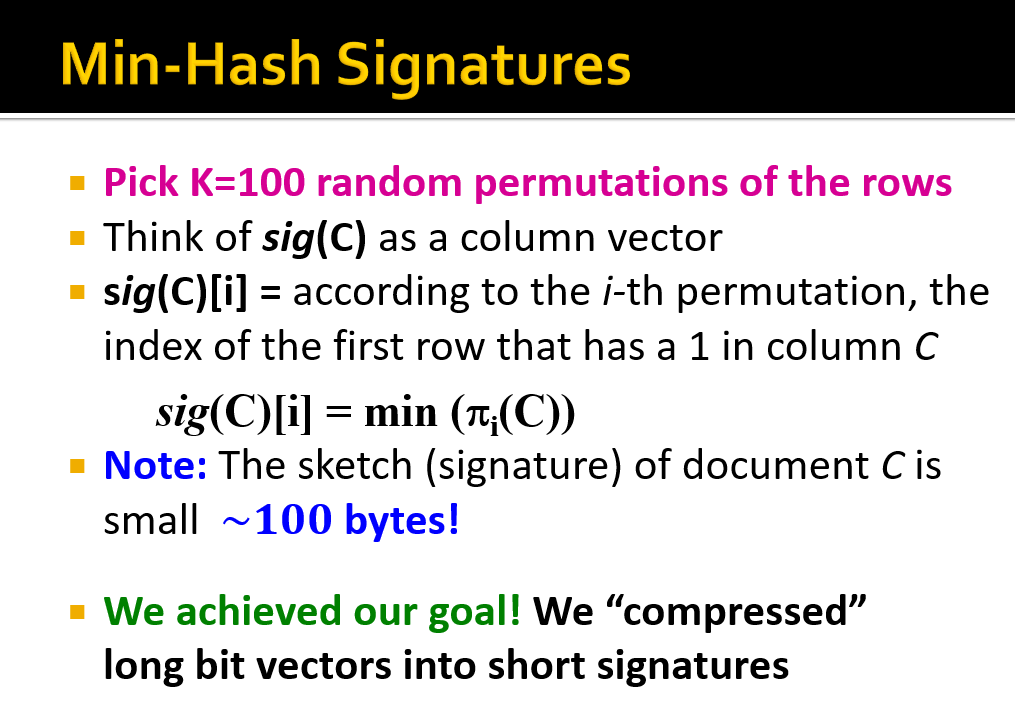
Chúng ta có : Pr[hπ(C1) = hπ(C2)] = sim(C1, C2)

• Sau đó chúng ta tổng quát hóa thành nhiều hàm băm

• Sự giống nhau của hai chữ ký là phần nhỏ của các hàm băm mà chúng đồng ý

Ghi chú : Do thuộc tính Min-Hash, sự giống nhau của các cột cũng giống như sự giống nhau mong đợi của các chữ ký của chúng.





Chọn K = 100 hoán vị ngẫu nhiên của các hàng

• Sig (C) như một vectơ cột

• sig (C) [i] = theo hoán vị thứ i, chỉ số của hàng đầu tiên có số 1 trong cột C

sig(C)[i] = min (πi(C))

Ghi chú : Bản phác thảo (chữ ký) của tài liệu C có kích thước nhỏ gần bằng 100 Byte

**STEP 3: LOCALITY – SENSITIVE HASHING – TẬP TRUNG VÀO CÁC CHỮ KÝ TỪ CÁC TÀI LIỆU TƯƠNG TỰ**

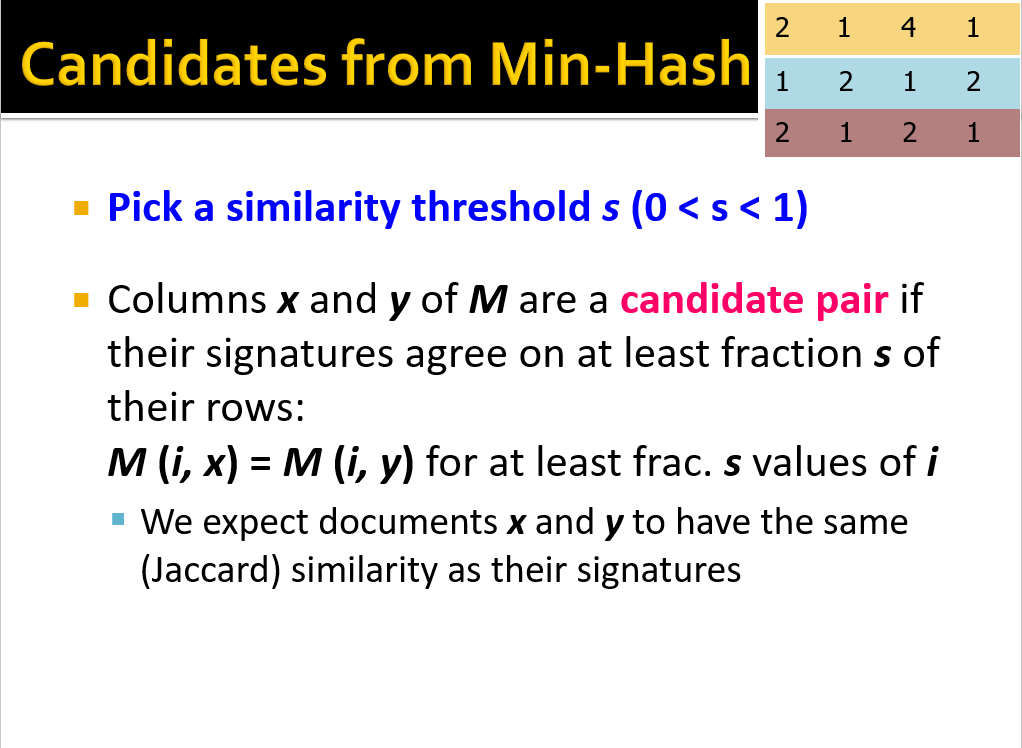
Mục tiêu: Tìm tài liệu có độ tương tự ít nhất s(đối với một số ngưỡng tương tự, ví dụ: s=0.8)

LSH - Ý tưởng chung: Sử dụng một hàm f (x, y) cho biết x và y có phải là một cặp ứng cử viên hay không: một cặp phần tử có độ giống nhau phải được đánh giá

Đối với ma trận Min-Hash:

Băm các cột của ma trận chữ ký M thành nhiều nhóm

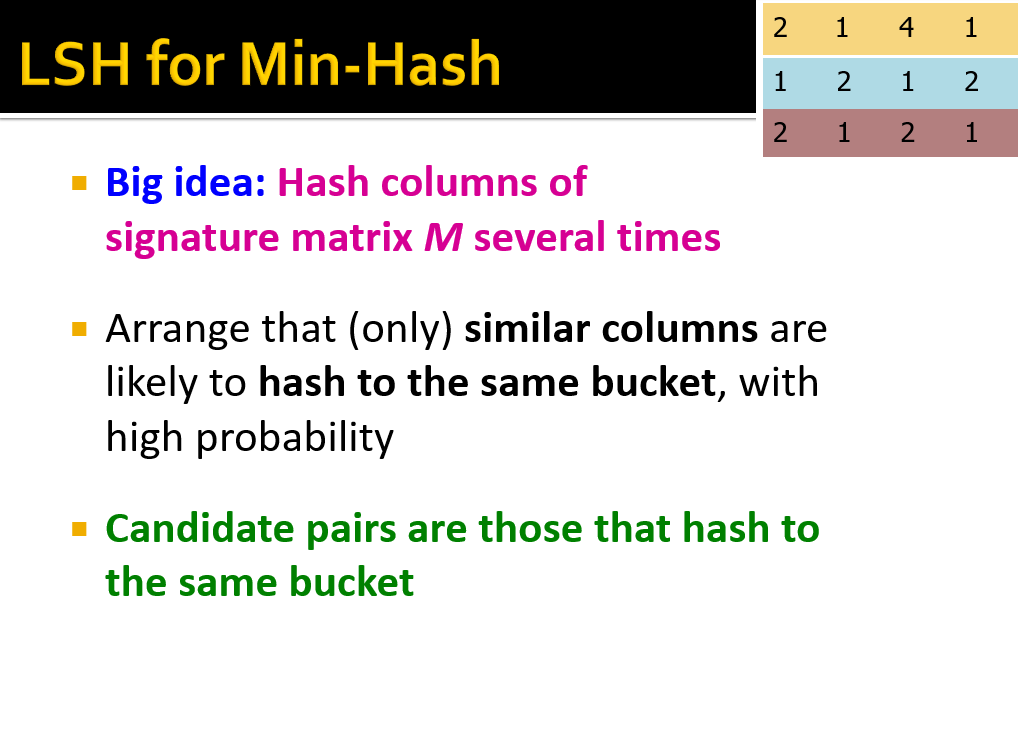
Mỗi cặp tài liệu được băm vào cùng một nhóm là một cặp ứng viên



Chọn ngưỡng tương tự s (0<s<1)

Các cột x và y của M là một cặp ứng cử viên nếu chữ ký của chúng đồng ý trên ít nhất phần s trong các hàng của chúng: M (i, x) = M (i, y) cho ít nhất là phân số. Giá trị s của i

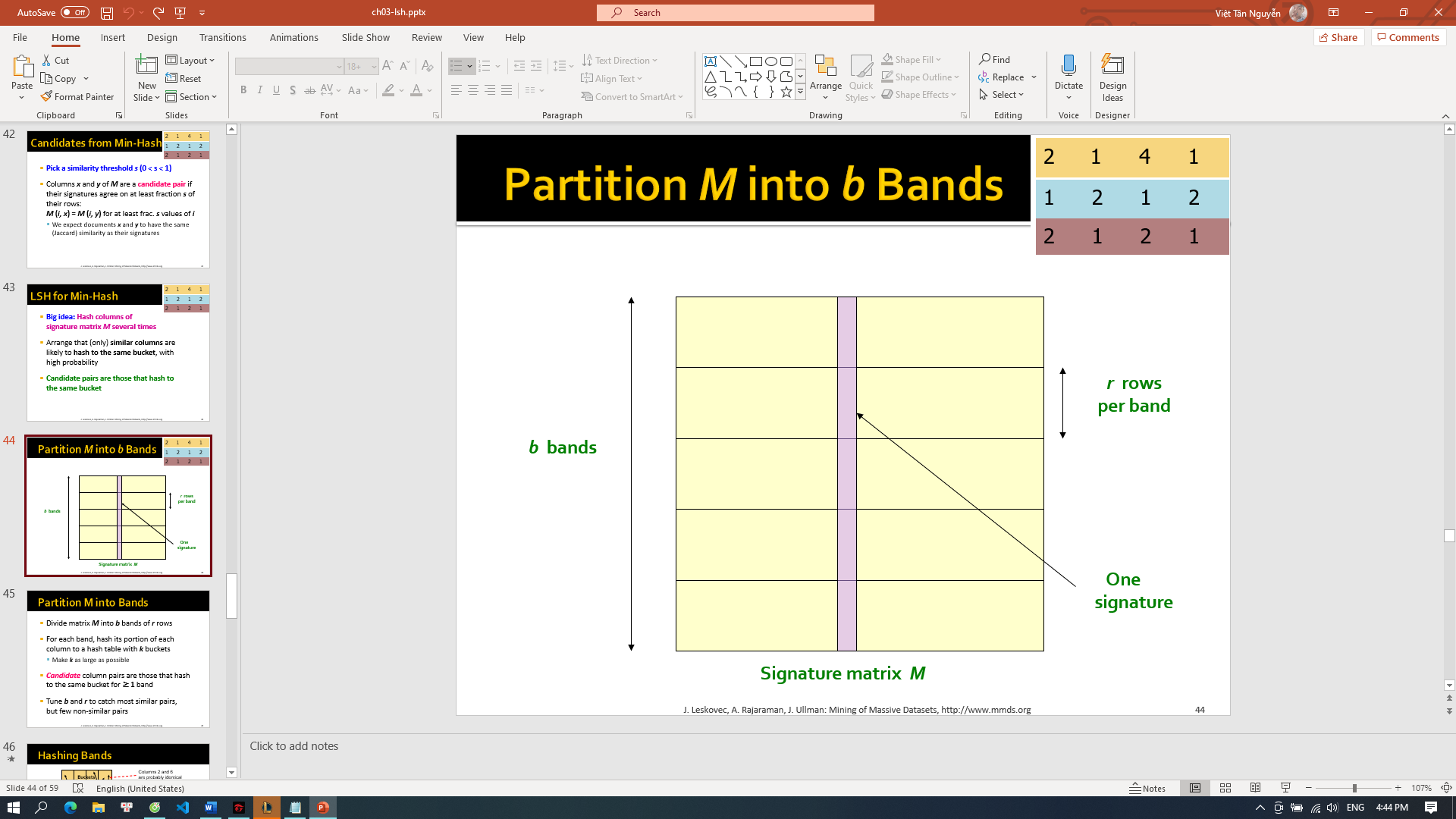
Mong đợi các tài liệu x và y có cùng (Jaccard) giống nhau như chữ ký của chúng



Ý tưởng: Băm các cột của ma trận chữ ký M nhiều lần

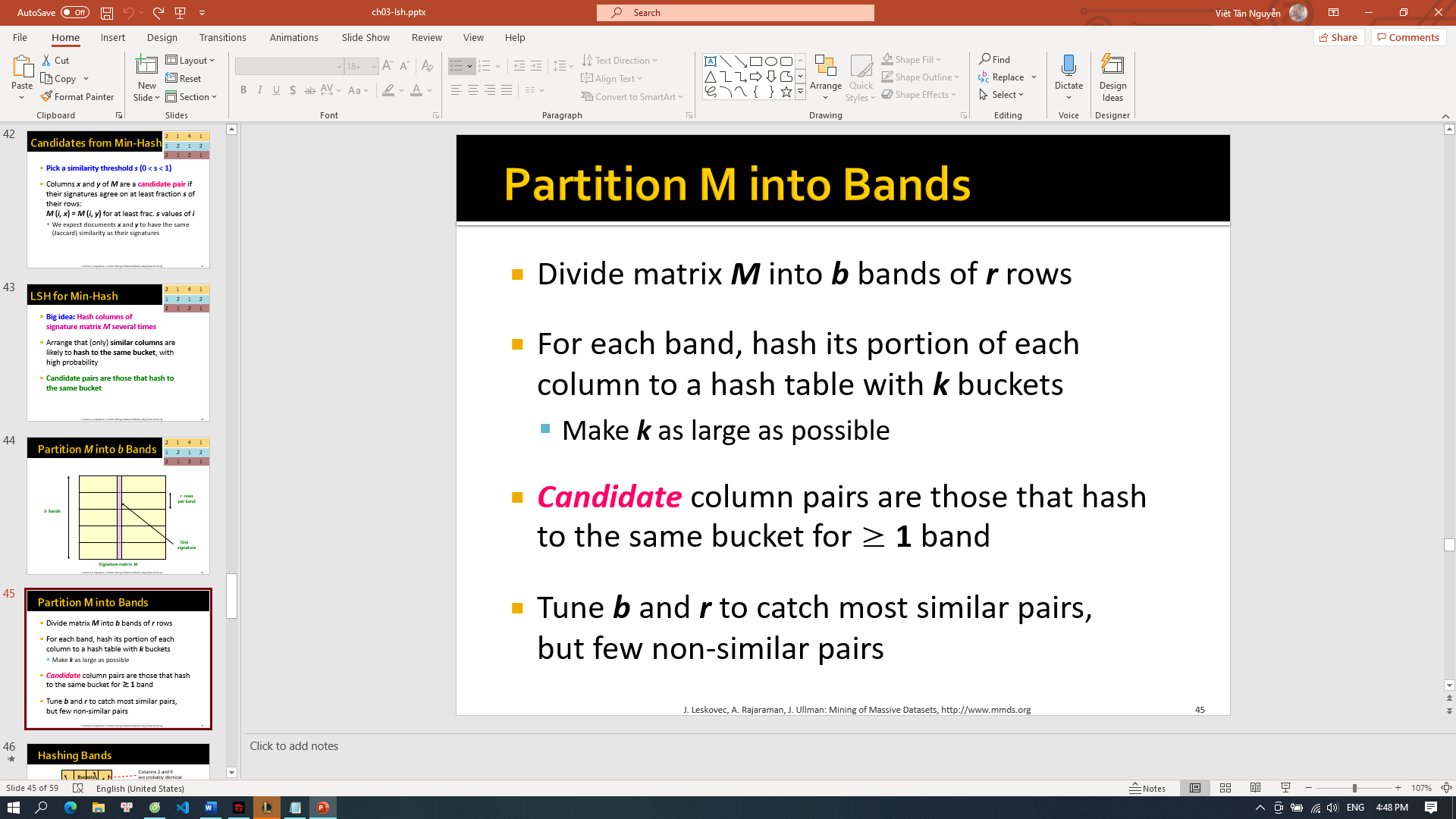
Sắp xếp để các cột tương tự có khả năng băm vào cùng một nhóm, với xác suất cao

Các cặp ứng cử viên là những cặp được băm vào cùng một nhóm



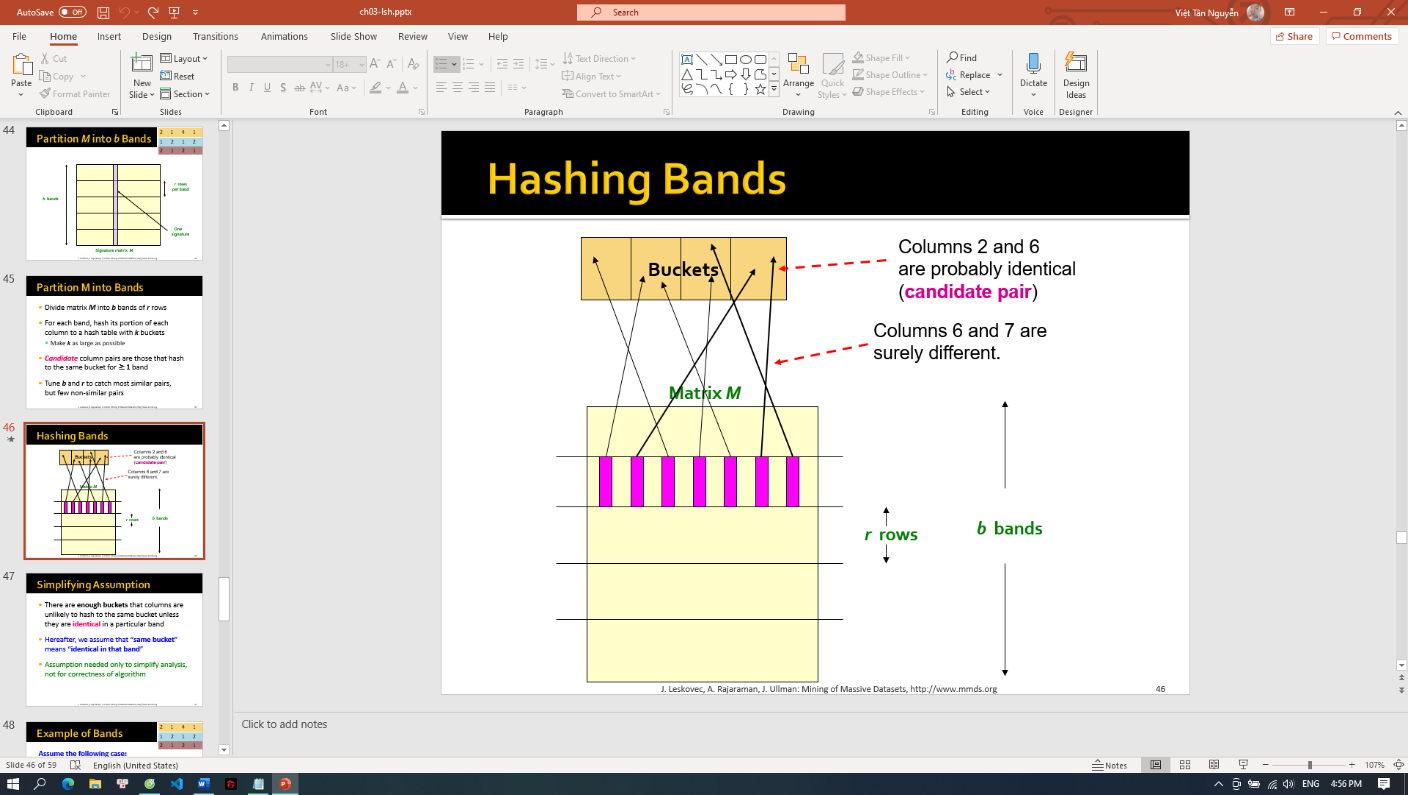
Vùng màu vàng là ma trận chữ ký M. Mỗi cột tương ứng với một chữ ký và mỗi hàng là một trong những thành phần của tất cả các chữ ký

Chia các hàng thành các dải b cho một số b. Kết quả là có r hàng trên mỗi dải, trong đó b nhân với r là tổng chiều dài của các chữ ký. Đó là, số lượng các hàm băm chính mà chúng ta sử dụng để tạo chữ ký. Chúng ta sẽ tạo một hàm băm từ mỗi băng tần.



Chúng ta đã chia ma trận chữ ký M thành b dải r hàng mỗi dải.

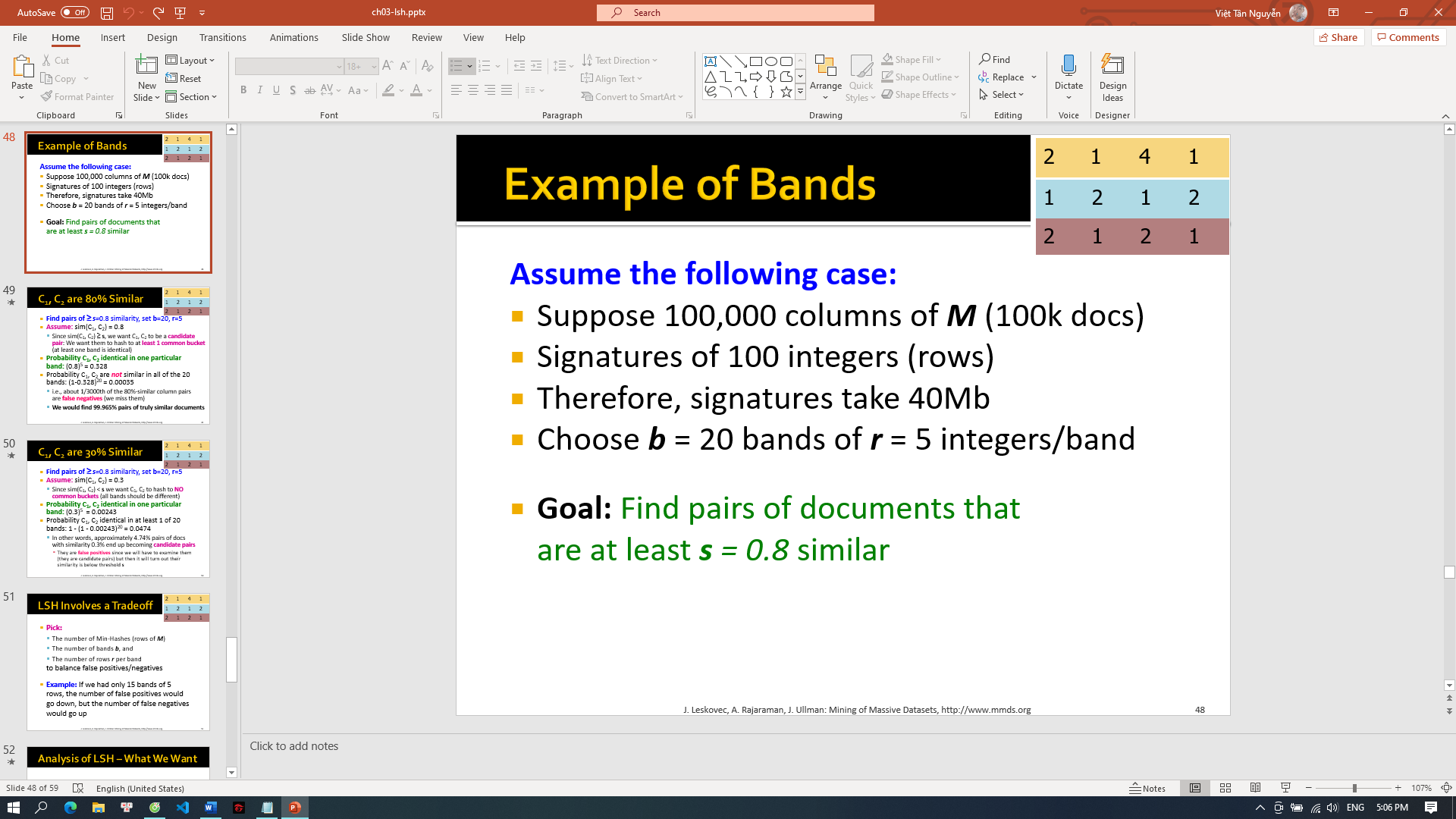
Từ mỗi dải, chúng ta tạo một hàm băm. Hàm băm này chỉ băm các giá trị mà một cột nhất định có trong dải đó. Chúng ta sẽ tạo một nhóm cho mỗi vectơ có thể có giá trị b mà một cột có thể có trong dải đó. Đó là, chúng ta muốn có quá nhiều nhóm để hàm băm thực sự là hàm nhận dạng, nhưng đó có lẽ là quá nhiều nhóm



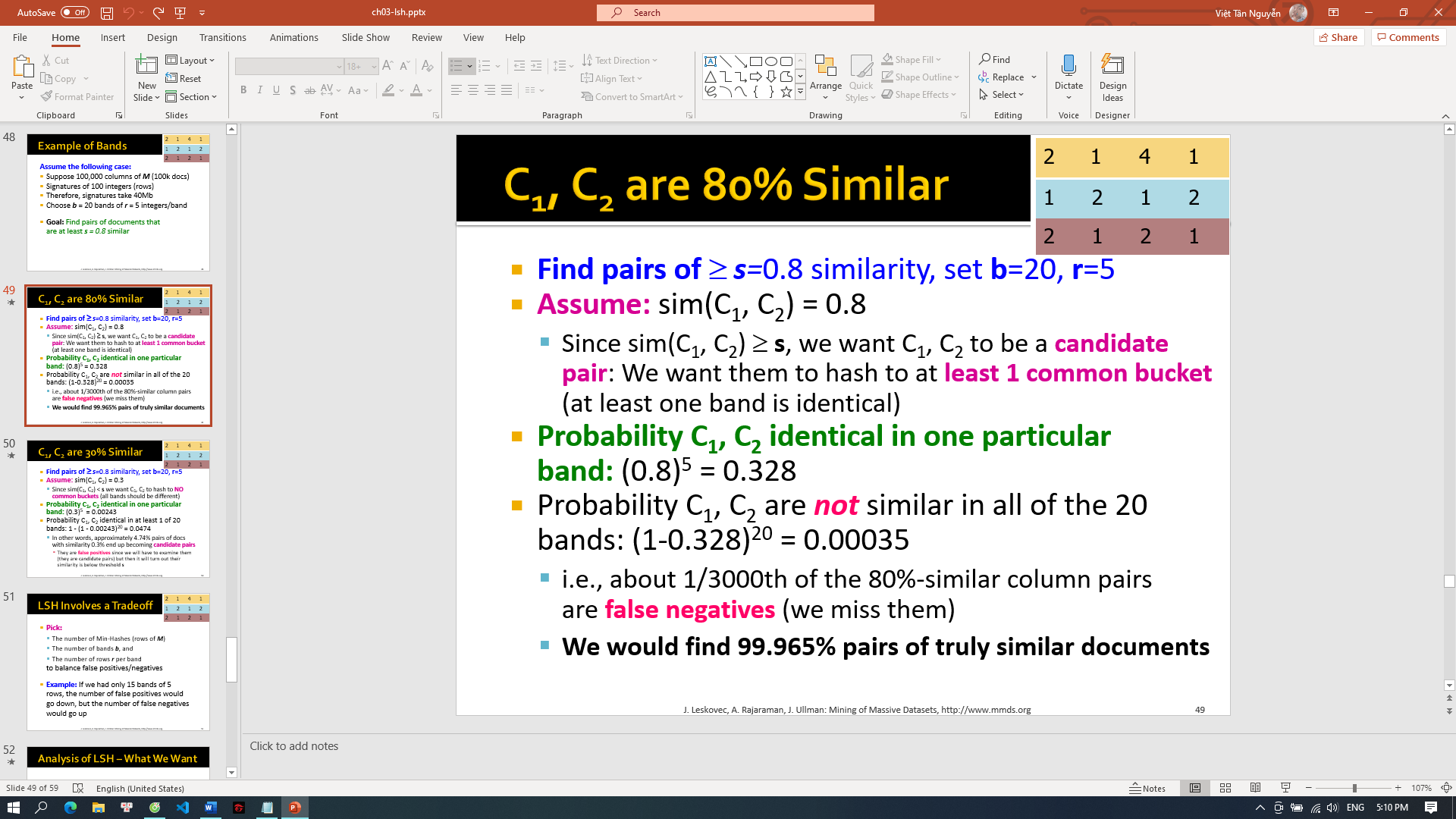
Đây là hình ảnh về những gì mà hàm băm cho LSH trên ma trận chữ ký. Chúng ta thấy một trong những dải b, tất nhiên là dải bao gồm r hàng. Nó cũng hiển thị ma trận bao gồm một số, trong số bảy cột hoặc chữ ký.

Và mỗi đại diện màu hồng, hình chữ nhật đại diện cho phần cột trong 1 dải mà chúng ta tập trung vào. Cột 6 và cột 7 băm thành các nhóm khác nhau. Do đó chúng chắc chắn khác nhau. Cặp 6 và 7 không được tạo thành 1 cặp ứng cử viên bởi hàm băm LSH này.

Mặt khác, cột 2 và 6 thực hiện việc băm cho cùng một nhóm. Vì vậy, cột 2 và 6 là một cặp ứng cử viên bất kể điều gì xảy ra trong các nhóm khác



Giả sử có 100.000 cột. Chúng ta đang tìm kiếm các tài liệu tương tự trong số 100.000 tài liệu. Chữ ký có độ dài là 100, chúng ta sử dụng các hàm minhash 100 để tạo chữ ký, do đó ma trận chữ ký M là 100 hàng x 100.000 cột. Chữ ký yêu cầu lấy 40Mb

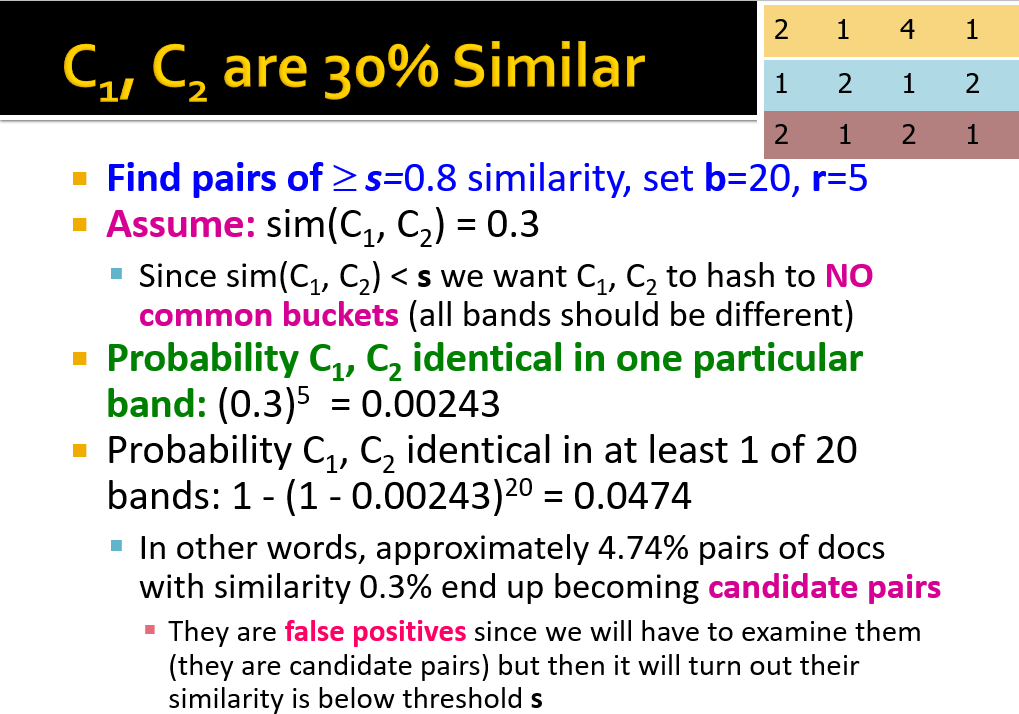


Đầu tiên xem xét 2 cột C1 và C2, đại diện cho các bộ có độ tương tự Jaccard 0.8. Do tính ngẫu nhiên liên quan đến băm nhỏ, các cột C1 và C2 có thể đồng ý trong nhiều hơn hoặc ít hơn 80 hàng của họ, nhưng rất có thể chúng sẽ có khoảng 80 hàng bằng nhau

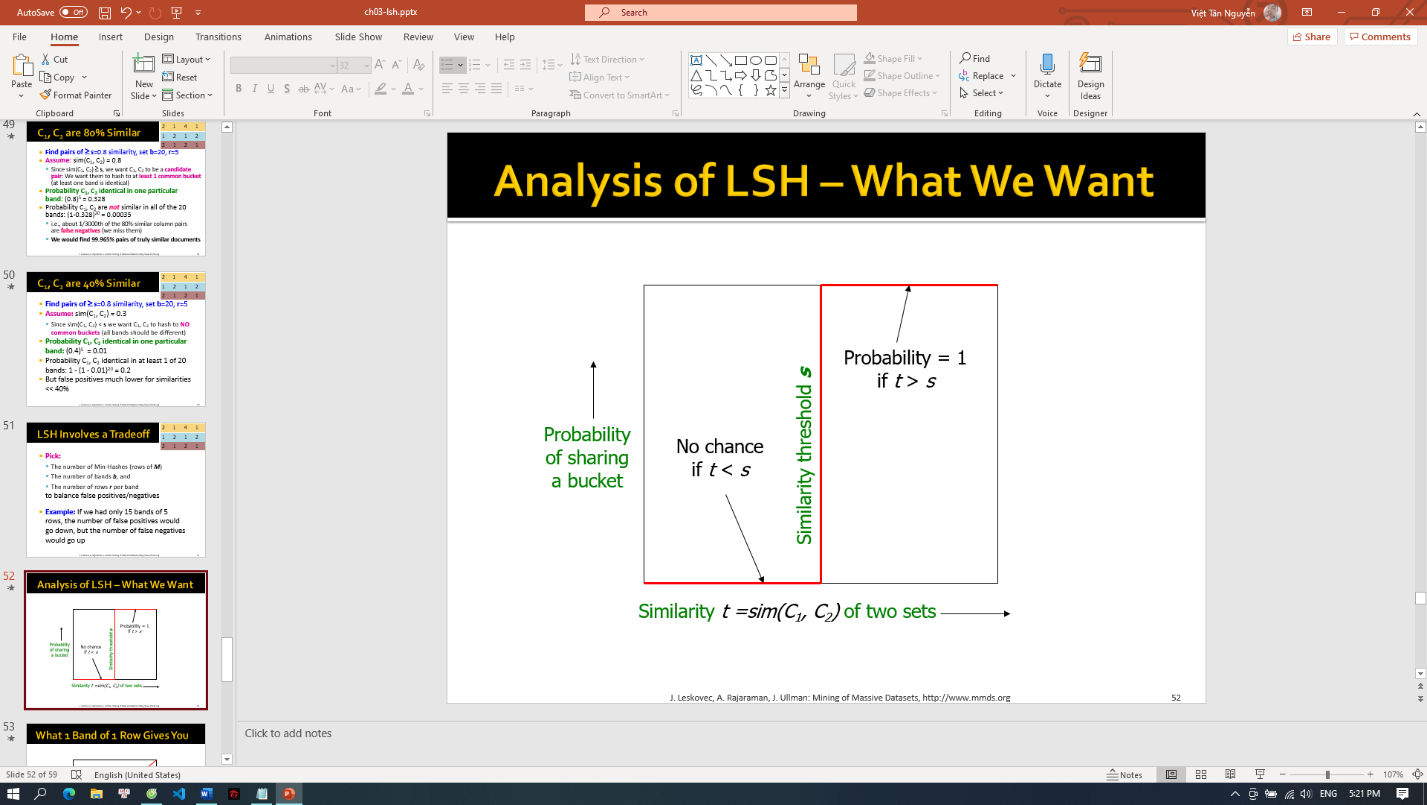
Xác suất để chúng đồng ý trong một hàng bất kỳ là 0.8, xác suất để hai cột đồng ý ở tất cả năm hàng của một dải được nâng lên 0,8 lên lũy thừa thứ năm, hoặc xấp xỉ 0,328.

Chúng ta có 20 cơ hội để biến cặp cột trở thành cặp ứng cử viên. Xác suất để chúng không băm vào cùng một nhóm trong một dải là 1 - 0,328 hoặc 0,672

Xác suất các cột không thể băm thành cùng 1 nhóm cho bất kỳ dải nào trong số 20 dải là giá trị 0.672 được nâng lên thành lũy thừa 20, đó là một con số nhỏ, là 0.00035. Cơ hội mà cặp C1 và C2 sẽ là 1 cặp ứng cử viên là 1-0.00035, là 0.99965. Xác suất âm tính giả, một cặp bộ có độ giống Jaccard 80%, nhưng những chữ ký không trở thành một cặp ứng cử viên là 0,00035, hoặc khoảng 1/3.000.



Bây giờ , đến với 1 cặp có độ giống nhau của Jaccard là 0.3. Xác suất để chữ ký của chúng băm vào cùng một nhóm trong ít nhất 1 trong số 20 dải chắc chắn không nhiều hơn 20 lần.



Một cách để xem xét vấn đề thiết kế một lược đồ LSH từ ma trận minhash là cái này. Chúng ta muốn xác suất của hai cột chia sẻ một nhóm là một hàm bước với ngưỡng t bằng giá trị mà tại đó chúng ta coi các tập cơ bản tương tự nhau. Tức là, nếu độ tương tự Jaccard của các tập cơ bản nhỏ hơn t, chúng ta muốn không có cơ hội các chữ ký sẽ chia sẻ một nhóm cho một trong các băm và do đó trở thành một cặp ứng cử viên. Tuy nhiên, nếu mức tương tự Jaccard cơ bản vượt quá 2, chúng ta muốn cặp chữ ký chắc chắn trở thành một cặp ứng cử viên.

