Trần Ngọc Uyên Nhi - 050608200529

Nguyễn Thị Uyển Nhi - 050608200527

**BÀI TẬP LỚN CUỐI KỲ MÔN PHÂN TÍCH DỮ LIỆU LỚN\_L14**

**CÂU 1:**

Để giải quyết vấn đề này, sử dụng Spark để phân phối tính toán, tạo các cặp người dùng-bạn bè và bạn bè chung, xử lý và sắp xếp các cặp một cách hiệu quả, đồng thời tạo danh sách bạn bè chung cho mỗi người dùng. Tệp đầu vào được đọc dưới dạng Spark RDD (Resilient Distributed Dataset - Bộ dữ liệu phân tán đàn hồi). Một hàm được xác định để tạo các cặp người dùng và bạn chung từ mỗi dòng của hàm RDD (generate\_pairs(line)). Sử dụng các cặp đã tạo, một danh sách người dùng và bạn bè được đề xuất của họ được tạo bằng cách sắp xếp các cặp dựa trên thứ tự giảm dần của bạn chung và thứ tự tăng dần của tên bạn bè. Trích xuất và xuất danh sách ứng viên bạn bè cho từng người dùng.

**CÂU 2:**

**a. Một nhược điểm của việc sử dụng độ tin cậy là nó bỏ qua Pr(B). Tại sao đây là một nhược điểm? Giải thích tại sao mức tăng và niềm tin không bị nhược điểm này?**

**Giải câu a:**

Sử dụng độ tin cậy trong phân tích giỏ hàng có một nhược điểm là nó bỏ qua xác suất xuất hiện của tập B độc lập với tập A, Pr(B). Tức là độ tin cậy lúc này chỉ cung cấp thông tin về xác suất xuất hiện của B nếu A đã xuất hiện, mà không cung cấp thông tin về xác suất xuất hiện của B nếu A không xuất hiện.

Việc này có thể dẫn đến một số luật kết hợp được đưa ra dựa trên độ tin cậy, nhưng chúng không có giá trị thực tiễn vì không thể dự đoán được xác suất xuất hiện của B nếu A không xuất hiện như đã nói ở trên

Mức tăng và Niềm tin không có nhược điểm này vì cả hai chỉ số Mức tăng và Niềm tin đều được tính toán dựa trên xác suất xuất hiện của B, không phụ thuộc vào sự xuất hiện A. Chỉ số Mức tăng dùng để tính toán sự tăng lên của xác suất xuất hiện của B khi A xuất hiện, so với xác suất xuất hiện của B nếu A không xuất hiện. Trong khi đó, chỉ số Niềm tin dùng để tính toán xác suất xuất hiện của A mà không có B so sánh với tần suất thực tế của sự xuất hiện của A mà không có B. Tuy nhiên, chỉ số này có thể bị giảm giá trị nếu tập mục A hoặc B xuất hiện rất ít trong dữ liệu.

Vì vậy, cần phải lựa chọn các chỉ số phù hợp và kết hợp chúng để có thể đánh giá luật kết hợp trong phân tích giỏ hàng cũng như đưa ra khuyến nghị kinh doanh có giá trị thực tế.

**b.** **Một đại lượng được gọi là cân đối khi đo lường(A → B) = đo lường(B → A). Trong số các đại lượng được đưa ra, đại lượng nào cân đối? Đối với mỗi đại lượng, đưa ra dẫn chứng cho thấy đại lượng đó là cân đối hoặc một ví dụ phản chứng cho thấy đại lượng đó không cân đối.**

**Giải câu b:**

Trong số các đại lượng được đưa ra, Niềm tin (conviction) và Độ tin cậy (confidence) không cân đối, còn Mức tăng (lift)  cân đối.

*Ví dụ chứng minh Conviction và Confidence không cân đối:*

Cho các giỏ hàng sau:

* Giỏ hàng 1: A, B
* Giỏ hàng 2: A, C, D, E
* Giỏ hàng 3: B, D
* Giỏ hàng 4: A, C, E
* Giỏ hàng 5: A, B, C, D, E

Ta muốn tính conv(A → B) và conv(B → A). Ta sẽ thực hiện như sau:

* Tính support của A, B
  + S(A) = 4
  + S(B) = 3
* Tính confidence của quy tắc A → B và quy tắc B → A:
  + Conf(A → B) = Pr(B|A) = S(A,B) / S(A) = 2/4 = 1/2
  + Conf(B → A) =Pr(A|B)= S(A,B) / S(B) = 2/3

Tính lift của quy tắc A → B và quy tắc B → A:

* Lift(A → B) = Conf(A → B) / Support(B) = (1/2)/3 =  0.167
* Lift(B → A) = Conf(B → A) / Support(A) = (2/3)/4 = 0.167
* Tính conviction của quy tắc A → B và quy tắc B → A:
  + Conv(A → B) = (1 - S(B)/N) / (1 - Conf(A → B)) = (1 - 3/5) / (1 - 1/2) =  0.8
  + Conv(B → A) = (1 - S(A)/N) / (1 - Conf(B → A)) = (1 - 4/5) / (1 - 2/3) =  0.6

**Kết luận:** Như vậy, conv(A → B) ≠ conv(B → A) và conf((A → B) ≠ conf(B → A) . Do đó, conviction và confidence không cân đối.

**CÂU 3:**

Trong chương 3.3.5 của sách "Mining of Massive Datasets", tác giả đề cập đến việc mô phỏng một hoán vị ngẫu nhiên của các hàng trong ma trận, bằng cách chọn ngẫu nhiên một tập con k hàng trong n hàng và thực hiện minhashing trên tập con này. Tuy nhiên, nếu trong tập con k hàng này không có bất kỳ hàng nào chứa giá trị 1 ở một cột nhất định, thì kết quả của minhashing sẽ là "không biết" hoặc "don't know". Trong trường hợp này, không nên giả định rằng hai cột mà cả hai đều có kết quả minhash là "không biết" thì có khả năng giống nhau. Tuy nhiên, nếu xác suất để có kết quả minhash là "không biết" là nhỏ, chúng ta có thể chấp nhận tình huống này và đơn giản bỏ qua các giá trị minhash này khi tính tỷ lệ các minhash mà hai cột trùng nhau.

Trong phần (a), chúng ta cần chứng minh rằng xác suất để có kết quả minhash là "không biết" khi chỉ xem xét một tập con k hàng ngẫu nhiên trong n hàng, là không quá (n-k/n)^m. Trong phần (b), chúng ta sử dụng giới hạn này để xác định lựa chọn phù hợp cho k, dựa trên sự chấp nhận của chúng ta đối với xác suất này.

**a. Đề bài đưa ra vấn đề về việc tìm xác suất để thuật toán minhash trả về giá trị "don't know" khi chỉ xét một tập con ngẫu nhiên gồm k hàng của ma trận. Với tập con này, nếu không có phần tử nào trong một cột nào đó của ma trận có giá trị 1, thì kết quả trả về sẽ là "don't know".**

**Câu (a) yêu cầu chúng ta chứng minh rằng xác suất trên là không quá (n-k/n)^m, trong đó n là số hàng của ma trận, m là số lượng giá trị 1 trong cột đang xét.**

**Giải câu a:**

- Đầu tiên, ta cần tính xác suất để thuật toán minhash trả về giá trị "don't know" khi chỉ xét một tập con gồm k hàng của ma trận. Để tính xác suất này, ta cần xác định số lượng cột có giá trị 1 trong cột đang xét, và số lượng cột không có giá trị 1 trong tập con k hàng được chọn. Sau đó, ta sử dụng phương pháp xác suất để tính xác suất cột đó sẽ trả về "don't know".

- Số lượng cột có giá trị 1 trong cột đang xét là m, và số lượng cột không có giá trị 1 trong tập con k hàng được chọn là (n-k)^m. Để tính xác suất cột đó sẽ trả về "don't know", ta cần tính tỷ lệ giữa số lượng cột không có giá trị 1 trong tập con k hàng được chọn và tổng số lượng cột có giá trị 1 trong ma trận.

- Số lượng cột có giá trị 1 trong ma trận là (n choose m), tức là số cách chọn m phần tử từ n phần tử. Tỷ lệ giữa số lượng cột không có giá trị 1 trong tập con k hàng được chọn và số lượng cột có giá trị 1 trong ma trận là: (n-k)^m / ()

Ta muốn chứng minh rằng tỷ lệ này không vượt quá (n-k/n)^m. Để chứng minh điều này, ta sử dụng công thức đa thức Newton:

(n choose m) = n! / (m! \* (n-m)!)

Chia tử và mẫu của tỷ lệ trên cho m!, ta có:

(n-k)^m / (n choose m) = (n-k)^m \* m! / (n! / ((n-m)! \* m!)) = (n-k)^m \* (n-m)! / n!

Lưu ý rằng (n-m)! / n! = 1 / (n \* (n-1) \* ... \* (n-m+1)) và 0 <= m <= n. Do đó, ta có thể viết lại tỷ lệ trên dưới dạng:

(n-k)^m / (n choose m) = (n-k)/n \* (n-k-1)/(n-1) \* ... \* (n-k-m+1)/(n-m+1)

Với mỗi thành phần trong tích này, ta có (n-k-i+1) <= n-k. Do đó, ta có thể ước lượng tỷ lệ trên như sau:

(n-k)^m / (n choose m) <= (n-k)/n \* (n-k)/n \* ... \* (n-k)/n = (n-k/n)^m

Do đó, ta đã chứng minh được rằng tỷ lệ giữa số lượng cột không có giá trị 1 trong k hàng được chọn và tổng số cột có đúng m giá trị 1 là không vượt quá (n-k)/n)^m.

Với công thức này, ta có thể tính được xác suất để có được "don't know" trong quá trình minhashing khi chỉ xét đến một tập con k hàng được chọn ngẫu nhiên từ n hàng.

Giả sử ta muốn chấp nhận một mức độ sai lệch trong việc đánh giá sự tương đồng của các cặp cột trong khoảng [0,epsilon], với epsilon là một số thực dương nhỏ. Điều này có nghĩa là ta có thể bỏ qua tất cả các cặp cột mà các minhash của chúng đều cho giá trị "don't know". Ta muốn tính toán giá trị tối đa của k, sao cho xác suất để có được "don't know" nhỏ hơn epsilon.

Để làm điều này, ta giải phương trình (n-k)/n)^m = epsilon, để tìm giá trị của k. Khi giải phương trình, ta được k = n(1-epsilon)^(1/m). Điều này có nghĩa là nếu ta chọn k hàng ngẫu nhiên từ n hàng, thì xác suất để có được "don't know" là nhỏ hơn epsilon.

Vì vậy, khi tính toán minhash, nếu chúng ta chỉ xem xét một tập con ngẫu nhiên của k hàng thay vì toàn bộ n hàng, ta có thể lượng giảm thời gian tính toán. Điều này đặc biệt hữu ích khi xử lý các tập dữ liệu lớn. Tuy nhiên, ta cần chọn giá trị k phù hợp để đảm bảo rằng xác suất để có được "don't know" là không quá lớn.

**b. Ở câu (b), yêu cầu của đề bài là tìm giá trị nhỏ nhất của k để đảm bảo xác suất để cho ra "don't know" là tối đa e^(-10) trong trường hợp n và m là rất lớn (nhưng n lớn hơn rất nhiều so với m và k).**

**Giải câu b:**

Câu (b) yêu cầu tính giá trị nhỏ nhất của k để đảm bảo rằng xác suất "don't know" nhỏ hơn hoặc bằng e^(-10). Ta bắt đầu với biểu thức đã được tính toán trong câu (a), tức là xác suất "don't know" của một cột khi chỉ xem xét k hàng được chọn ngẫu nhiên trong n hàng:

(n-k/n)^m

Ta cần tìm giá trị nhỏ nhất của k sao cho biểu thức trên nhỏ hơn hoặc bằng e^(-10). Ta bắt đầu bằng cách lấy logarit tự nhiên cả hai vế của bất phương trình này, và đổi dấu:

ln((n-k/n)^m) ≤ -10

m\*ln(n-k/n) ≤ -10

ln((1-k/n)^m) ≤ -10

Dựa trên gợi ý thứ hai, ta có thể xấp xỉ (1 - k/n)^n/k bằng 1/e khi n lớn. Do đó, ta có thể xấp xỉ ln((1 - k/n)^m) bằng -m/k. Vì vậy, ta có thể viết lại bất phương trình ở trên như sau:

-m/k ≤ -10

mk/n ≥ 10

Từ đó, ta suy ra:

k ≥ 10n/m

Do đó, giá trị nhỏ nhất của k để đảm bảo rằng xác suất "don't know" nhỏ hơn hoặc bằng e^(-10) là 10n/m.

**c. Đề bài yêu cầu cho một ví dụ về hai cột (tương ứng với hai tập hợp S1 và S2) trong đó xác suất để giá trị minhash của hai cột bằng nhau không giống với độ tương đồng Jaccard của chúng khi chỉ xét các hoán vòng.**

**Giải câu c:**

Giả sử có hai cột (tương ứng với hai tập hợp A và B) như sau:

A = {1, 2, 3, 4, 5}

B = {2, 4, 6, 8, 10}

*Jaccard similarity:*

J(S1, S2) = |S1 ∩ S2| / |S1 ∪ S2| = {2,4} / {1,2,3,4,5,6,7,10} = 2/7

Để tính minhash cho hai cột này, ta cần tạo ra các bộ hash functions và sử dụng chúng để tạo ra signature matrix.

|  |  |
| --- | --- |
| **Cột 1** | **Cột 2** |
| 1 | 1 |
| 0 | 0 |
| 1 | 0 |
| 0 | 1 |
| 1 | 0 |

Giả sử ta bắt đầu với hàng đầu tiên của ma trận và thực hiện các hoán vị tuần hoàn theo thứ tự. Giá trị minhash của cột 1 được tính bằng việc đếm số lượng hàng đầu tiên mà giá trị là 1, sau đó chuyển sang hàng tiếp theo và tiếp tục đếm số lượng hàng mà giá trị là 1, và cứ tiếp tục như vậy cho đến khi đến hàng cuối cùng của ma trận. Với cột 2, chúng ta thực hiện quá trình này tương tự.

Do đó, xác suất để hai cột có giá trị minhash giống nhau khi chỉ sử dụng các hoán vị tuần hoàn của ví dụ trên là 2/6, tức là 1/3. Do chỉ có hai hoán vị (1, 2, 3) và (2, 3, 1) cho phép hai cột có giá trị minhash giống nhau trong các hoán vị tuần hoàn.

Tổng hợp kết quả trên, ta có thể thấy rằng giá trị minhash của S1 và S2 không phải là một ước lượng tốt cho độ tương đồng Jaccard của chúng, khi chỉ sử dụng các hoán vị tuần hoàn. Điều này cho thấy rằng không phải tất cả các hoán vị tuần hoàn đều có khả năng đưa ra ước lượng chính xác cho Jaccard similarity. Thay vào đó, ta cần sử dụng một số lượng lớn các hoán vị khác nhau để đạt được ước lượng chính xác hơn cho Jaccard similarity của hai tập hợp.

**Kết luận:** Kết quả minhash này không thể thay thế cho Jaccard similarity. Ta sẽ không thể đưa ra kết quả chính xác cho Jaccard similarity nếu chỉ sử dụng các hoán vị tuần hoàn để tính minhash.