BÀI TẬP
RECOMMENDATION SYSTEM

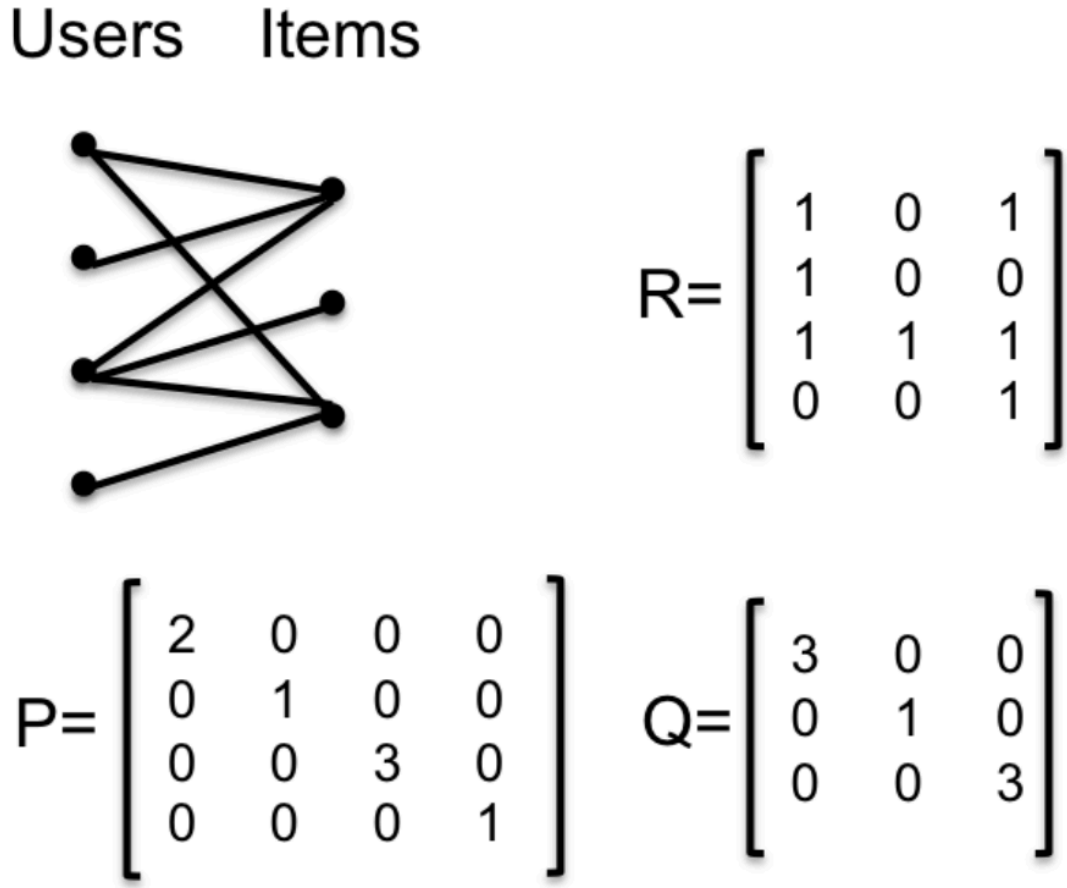
Lưu ý: Bài này dùng python thuần không cần dùng Spark, nếu bị vấn đề bộ nhớ khi làm việc trên ma trận lớn bạn hãy chắc chắn rằng bạn đang sử dụng python 64 bit.

Chúng ta xem quan hệ giữa user và item là đồ thị lưỡng cực (đồ thị hai phía) trong đó mỗi cạnh giữa user U và item I thể hiện cho biết user U thích item I. Chúng ta cũng biểu diễn ma trận đánh giá (ratings) cho các user và item là R, trong đó mỗi dòng trong R tương ứng với một user, mỗi cột tương ứng với một item. Nếu user *i* thích item *j* thì $R_{i,j} = 1$ ngược lại $R_{i,j} = 0$. Giả sử chúng ta có *m* user và *n* item thì ma trận R là $m \times n$.

Ta định nghĩa mà trận P, $m \times m$ là ma trận đường chéo trong đó vị trí thứ *i* trên đường chéo thể hiện độ của user *i* (số lượng item mà user *i* thích). Tương tự ma trận Q, $n \times n$ là ma trận đường chéo trong đó vị trí thứ *i* trên đường chéo thể hiện độ của item *i* (số lượng user thích item *i*). Xem ví dụ trong hình dưới.

Độ đo tương tự cosine: Độ đo tương tự cosine giữa hai vector *u* và *v* được định nghĩa là:

$$\cos\_sim(u, v) = \frac{u.v}{\|u\|\|v\|}$$



Hình 1. Đồ thị hai chiều user-item

a) Ta định nghĩa ma trận user tương tự không chuẩn hoá $T = R * R^T$ (nhân của R và chuyển vị của nó). Hãy giải thích ý nghĩa của T_{ii} và T_{ij} ($j \neq i$) trên cơ sở cấu trúc đồ thị hai phía.

b) Ta định nghĩa ma trận item tương tự (item similarity matrix) S_I, $n \times n$ cho nên mỗi phần tử ở dòng *i* và cột *j* là độ đo cosine giữa item *i* và item *j* tương ứng với cột *i* và cột *j* của ma trận R. Chứng minh rằng $S_I = Q^{-1/2}R^TRQ^{-1/2}$ trong đó $Q^{-1/2}$ được định nghĩa $Q_{rc}^{-1/2} = 1/\sqrt{Q_{rc}}$ cho mọi vị trí khác 0 trong ma trận và là 0 cho các vị trí còn lại.
Câu hỏi tương tự cho ma trận user tương tự (user similarity matrix) S_U, $m \times m$ với mỗi phần tử ở dòng *i* và cột *j* là độ đo cosine giữa user *i* và user *j* tương ứng với dòng *i* và dòng *j* của ma trận R. Tức là bạn cần xác định biểu thức tính S_U theo R, P và Q. Câu trả lời của bạn cần thể hiện dạng ma trận, không nên thể hiện từng thành phần riêng lẻ của S_U.
Câu trả lời cần thể hiện quá trình sinh ra biểu thức cuối cùng.

c) Phương pháp khuyến nghị (recommendation method) sử dụng lọc cộng tác user-user (collaborative filtering) cho user u được mô tả như sau: với mọi item s, tính $r_{u,s} = \sum_{x \in users} \text{cos\_sim}(x, u) * R_{xs}$ và khuyến nghị *k* items có $r_{u,s}$ lớn nhất.
Tương tự, phương pháp khuyến nghị sử dụng lọc cộng tác item-item cho user u được mô tả như sau: với mọi item s, tính $r_{u,s} = \sum_{x \in items} R_{ux} * \text{cos\_sim}(x, s)$ và khuyến nghị *k* items có $r_{u,s}$ lớn nhất.
Hãy định nghĩa ma trận khuyến nghị (recommendation matrix) Γ, $m \times n$ sao cho Γ(i,j) = $r_{i,j}$. Tìm Γ trong cả hai phương pháp lọc cộng tác item-item và user-user theo R, P và Q. Câu trả lời của bạn cần thực hiện phép toán trên ma trận.
Gợi ý: Trường hợp item-item thì $\Gamma = RQ^{-1/2}R^TRQ^{-1/2}$
Câu trả lời của bạn cần thể hiện quá trình sinh ra biểu thức cuối cùng kể cả biểu thức item-item đã cho ở trên.

d) Trong câu hỏi này bạn sẽ áp dụng các phương pháp này vào tập dữ liệu thật. Dữ liệu chứa thông tin về các chương trình TV. Cụ thể có 9985 user và 563 chương trình TV, chúng ta biết nếu user đã xem chương trình TV nào trong thời gian 3 tháng.
Sử dụng tập dữ liệu trong thư mục **data**. Thư mục chứa:
- File user-shows.txt là ma trận đánh giá (ratings matrix) R, trong đó mỗi dòng tương ứng với một user và mỗi cột tương ứng với một chương trình TV. R_{ij} = 1 nếu user *i* đã xem chương trình *j* trong vòng 3 tháng. Các cột được phân cách bởi khoảng trắng.
- File shows.txt chứa tên của các chương trình TV cùng thứ tự với cột của R, tức là tên thứ *i* là tên của chương trình ở cột *i* của R.

Chúng ta sẽ so sánh hai kết quả khuyến nghị user-user và item-item cho user thứ 500 trong tập dữ liệu. Hãy gọi người này là Thạch (tương ứng với vị trí 499 trong mảng của python Thạch=users[499]).

Để làm được việc này tôi đã xoá đi 100 phần tử đầu tiên ở dòng của Thạch trong ma trận và thay nó bằng 0. Có nghĩa là chúng ta không biết 100 chương trình đầu tiên Thạch có xem hay không. Dựa vào hành vi của Thạch trên các chương trình khác chúng ta cần khuyến nghị các chương trình TV cho Thạch trong 100 chương trình đầu tiên. Chúng ta sẽ so sánh kết quả khuyến nghị của chúng ta với những gì Thạch đã xem.
- Tính ma trận P và Q
- Sử dụng công thức trong phần (c) tính Γ cho phương pháp user-user. Đặt S là tập 100 chương trình TV đầu tiên (ứng với 100 cột đầu tiên của ma trận). Với mọi chương trình trong S, Hãy cho biết 5 chương trình nào có độ đo tương tự cao nhất cho Thạch? Trong trường hợp hai chương trình có điểm bằng nhau thì lấy chương trình có số thứ

tự nhỏ hơn. Không xuất chỉ số của chương trình mà xuất tên của chương trình trong file shows.txt.

- Tính ma trận Γ cho movie-movie (là dạng cụ thể của loại item-item). Với mỗi chương trình TV trong S, Hãy cho biết 5 chương trình nào có độ đo tương tự cao nhất cho Thạch? Trong trường hợp hai chương trình có điểm bằng nhau thì lấy chương trình có số thứ tự nhỏ hơn. Không xuất chỉ số của chương trình mà xuất tên của chương trình.

Vì mục đích kiểm tra kết quả của bạn: độ đo tương tự lớn nhất cho user-user cần lớn hơn 900, độ đo tương tự lớn nhất cho movie-movie cần lớn hơn 31.

Kết quả cần nộp:

1. Kết quả giải thích câu a.
2. Biểu thức tính S_I và S_U theo R, P, Q và giải thích ở câu b.
3. Biểu thức tính Γ theo R, P, Q và giải thích ở câu c.
4. Source code và kết quả chạy của câu d, kết quả thể hiện gồm
    a. Tên của 5 chương trình TV có độ đo tương tự lớn nhất cho Thạch theo phương pháp lọc cộng tác user-user
    b. Tên của 5 chương trình TV có độ đo tương tự lớn nhất cho Thạch theo phương pháp lọc cộng tác movie-movie
    c. File colab lưu quá trình chạy và kết quả chạy

➔ Tóm lại cần nộp 2 file: file docx chứa kết quả 1, 2, 3, 4 và file colab chứa code chạy của 4.