

# APPLIED LINEAR ALGEBRA FOR IT

## ĐẠI SỐ TUYẾN TÍNH ỨNG DỤNG CHO CNTT

501032

Anh H. Vo

## 1 Giới thiệu

Trước diễn biến dịch bệnh Covid-19 phức tạp, nhiều biến thể mới ra đời với sức lây lan nhanh, cũng như những di chứng mà nó để lại ảnh hưởng rất lớn đến sức khỏe cộng đồng. Thêm vào đó, nhiều biện pháp cách ly, phong tỏa làm gián đoạn quá trình sản xuất, gây đứt gãy chuỗi cung ứng toàn cầu. Những biến động địa chính trị, cùng nhiều chính sách và các lệnh trừng phạt kinh tế giữa các quốc gia góp phần không nhỏ làm giá cả các mặt hàng tăng cao, thị trường tài chính bất ổn, gây khó khăn cho quá trình phục hồi và phát triển kinh tế. Trước những khó khăn và thách thức hiện tại, để giúp các cửa hàng nhỏ lẻ có thể cải thiện doanh thu và tồn tại, đòi hỏi cần phải xây dựng một *hệ thống quản lý cửa hàng thông minh*. Hệ thống gồm các chức năng cơ bản được trình bày trong phần 2.2 và dữ liệu mô tả được trình bày trong phần 2.1.

## 2 Yêu cầu

### 2.1 Mô tả dữ liệu

- *transactions*: danh sách các giao dịch tương ứng các sản phẩm được bán ra. Thành phần đầu tiên chứa mã giao dịch (dạng ký tự), thành phần tiếp theo là danh sách các mặt hàng được bán theo mã giao dịch (dạng ký tự).
- *products*: danh sách thông tin các mặt hàng trong cửa hàng được thể hiện thông qua một ma trận chứa thông tin bao gồm: mã mặt hàng ( $Id\_product$ ) dạng ký tự, giá thành (*price*), lượng tồn kho (*inventory*) dạng số, nhóm mặt hàng (*category*)  $\in \{1, 2, 3\}$  tương ứng bình dân, trung cấp và cao cấp.  
Ví dụ:

$$P = \begin{pmatrix} Id\_product & price & inventory & category \\ I_1 & 10 & 2 & 1 \\ I_2 & 1 & 1 & 3 \\ I_3 & 2 & 0 & 2 \\ \vdots & \vdots & \vdots & \vdots \\ I_n & 20 & 13 & 3 \end{pmatrix}$$

- *history*: lịch sử giao dịch của khách hàng, bao gồm thông tin mã khách hàng (dạng ký tự) và mã giao dịch (dạng ký tự) tương ứng với mã khách hàng.

### 2.2 Yêu cầu thực hiện

Ví dụ giả sử rằng thông tin dữ liệu cửa hàng được lưu trữ theo mô tả trong tập tin *data.mat* (Sinh viên đọc tập tin *data.mat* để so sánh ví dụ kết quả cho từng câu hỏi bên dưới, khi chấm bài nội dung dữ liệu trong

tập tin này sẽ thay đổi). Lưu ý, không được thay đổi định nghĩa của các hàm cho trước (tên và tham số truyền vào). Với những câu hỏi yêu cầu trả về danh sách (nhiều giá trị) nếu trường hợp tính toán có giá trị bằng nhau thì kết quả được sắp xếp theo thứ tự tương ứng tăng dần (dựa trên giá trị số đi kèm với chuỗi mã) của mã mặt hàng/ mã khách hàng.

Ví dụ: kết quả trả về các khách hàng có lần lượt số lần giao dịch  $U_2:4$ ,  $U_1:2$ ,  $U_3:4$ ,  $U_4:2$  thì danh sách trả về có thứ tự như sau:  $U_2, U_3, U_1, U_4$ . (Tương tự với trường hợp mã mặt hàng). Những trường hợp giá trị đầu vào không tồn tại trả về danh sách rỗng.

**Yêu cầu 1:** Viết hàm **req1(.)** để tìm mặt hàng bán chạy nhất và bán chậm nhất trong cửa hàng. Kết quả trả về 2 danh sách tương ứng. (0.5 điểm)

**Chú ý:**

- Sinh viên chỉ được tính điểm nếu làm đúng hết tất cả các yêu cầu.
- Trường hợp số lượng sản phẩm bán ra là như nhau cho tất cả các mặt hàng, kết quả trả về hai danh sách như nhau.

**Gợi ý:**

- Đầu vào: danh sách giao dịch.
- Đầu ra: 2 danh sách (vectors) mặt hàng bán chạy và danh sách mặt hàng bán chậm.
- Ví dụ:  $\vec{v}_{max} = [I_2]$ ,  $\vec{v}_{min} = [I_4, I_5]$

**Yêu cầu 2:** Viết hàm **req2(.)** tìm những sản phẩm có lượng tồn kho nhiều nhất và ít nhất. Kết quả trả về 2 danh sách tương ứng. (1 điểm)

**Chú ý:**

- Sinh viên chỉ được tính điểm nếu làm đúng hết tất cả các yêu cầu.
- Trường hợp số lượng tồn kho là như nhau, kết quả trả về hai danh sách như nhau.

**Gợi ý:**

- Đầu vào: danh sách thông tin các mặt hàng trong cửa hàng.
- Đầu ra: 2 danh sách (vectors) mặt hàng tồn kho nhiều nhất và ít nhất.
- Ví dụ:  $\vec{v}_{max} = [I_3, I_5]$ ,  $\vec{v}_{min} = [I_4]$

**Yêu cầu 3:** Viết hàm **req3(.)** tính tổng doanh thu của cửa hàng. (0.5 điểm)

Gợi ý:

$$total\_revenue = \sum_i^N q_i \times p_i$$

$q_i$  số lượng sản phẩm  $i$  được bán ra và  $p_i$  giá của sản phẩm  $i$ , và  $N$  tương ứng số lượng sản phẩm trong cửa hàng.

- Đầu vào: danh sách thông tin các mặt hàng trong cửa hàng và danh sách các giao dịch.
- Đầu ra: tổng doanh thu của cửa hàng  $\in R$ .
- Ví dụ:  $total\_revenue = 450$

**Yêu cầu 4:** Viết hàm **req4(.)** tìm mặt hàng có doanh thu cao nhất. (0.5 điểm)

Gợi ý:

- Đầu vào: danh sách thông tin các mặt hàng trong cửa hàng và danh sách các giao dịch.
- Đầu ra: danh sách (vector) các mặt hàng có doanh thu cao nhất.
- Ví dụ:  $\vec{g}_{max} = [I_2]$

**Yêu cầu 5:** Viết hàm **req5(.)** tìm  $k$  khách hàng có nhiều giao dịch với cửa hàng. Lưu ý:  $k \leq |users|$ . (1 điểm)

Gợi ý:

- Đầu vào: danh sách các giao dịch và  $k \in N^+$  khách hàng cần tìm.
- Đầu ra: danh sách (vector) có  $k$  thành phần chứa các mã khách hàng có nhiều giao dịch với cửa hàng (thứ tự mã khách hàng được sắp xếp tương ứng với số lượng giao dịch nhiều nhất đến ít nhất).
- Ví dụ:  $k = 3$ ,  $\vec{u} = [U_2, U_3, U_1]$

**Yêu cầu 6:** Viết hàm **req6(.)** tìm sản phẩm mà khách hàng có mã  $k$  thường mua nhất. (0.5 điểm)

Gợi ý:

- Đầu vào: danh sách lịch sử giao dịch của khách hàng và mã khách hàng  $k$ .
- Đầu ra: danh sách (vector) sản phẩm khách hàng  $k$  thường mua nhất
- Ví dụ:  $k = U_2$ ,  $\vec{i}_{max} = [I_2, I_3]$

**Yêu cầu 7:** Viết hàm **req7(.)** cho biết những sản phẩm nào mà khách hàng có ít giao dịch nhất với cửa hàng hay mua. (1.5 điểm)

**Gợi ý:**

- Đầu vào: danh sách giao dịch và lịch sử mua hàng.
- Đầu ra: danh sách (vector) các sản phẩm mà khách hàng có ít giao dịch với cửa hàng hay mua.
- Ví dụ:  $\vec{u} = [I_1, I_2, I_3]$

**Yêu cầu 8:** Viết hàm **req8(.)** tìm danh sách khách hàng hay mua sản phẩm giống khách hàng  $k$ . (1.5 điểm)

**Gợi ý:**

- Sử dụng độ đo tương tự  $\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|_2 \|\mathbf{u}\|_2}$
- $\mathbf{u}, \mathbf{v}$  vector thể hiện số lượng sản phẩm tương ứng mà khách hàng  $\mathbf{u}$  và khách hàng  $\mathbf{v}$  đã mua.
- Đầu vào: danh sách giao dịch và lịch sử mua hàng.
- Đầu ra: danh sách mã khách hàng hay mua sản phẩm giống khách hàng có mã  $k$  nhất.
- Ví dụ:  $k = U_5, \vec{u}_k = [U_3]$

**Yêu cầu 9:** Viết hàm **req9(.)** tìm danh sách các mặt hàng chưa bán được bất kỳ sản phẩm nào. Trường hợp tất cả các sản phẩm đều được bán kết quả trả về danh sách rỗng (1 điểm)

**Gợi ý:**

- Đầu vào: danh sách giao dịch, danh sách thông tin các mặt hàng trong cửa hàng và lịch sử mua hàng.
- Đầu ra: danh sách (vector) các mặt hàng chưa bán được sản phẩm.
- Ví dụ:  $\vec{b} = [I_6]$

**Yêu cầu 10:** Viết hàm **req10(.)** cho biết khách hàng  $k$  thuộc phân khúc khách hàng bình dân, trung cấp hay cao cấp. Kết quả trả về là một số nguyên  $\in \{1, 2, 3\}$  (2 điểm)

**Gợi ý:**

- Khách hàng có mã  $k$  được xếp vào phân khúc khách hàng  $i$  khi số lượng sản phẩm thuộc phân khúc  $i$  được mua nhiều nhất.
- $r = \operatorname{argmax}_i \mathbf{W}$ ,  $\forall r \in \{1, 2, 3\}$ , trong đó  $\mathbf{W}_{m \times c}$  thể hiện số lượng sản phẩm tương ứng từng phân khúc sản phẩm đã được các khách hàng mua,  $c$  tương ứng số lượng phân khúc khách hàng và  $m$  thể hiện số lượng khách hàng. Khách hàng được phân loại vào phân khúc nào tùy thuộc vào số lượng sản phẩm thuộc về phân khúc cụ thể mà họ đã mua.
- Đầu vào: danh sách giao dịch, danh sách thông tin các mặt hàng trong cửa hàng, lịch sử mua hàng, và mã của khách hàng  $k$ .
- Đầu ra: giá trị số nguyên  $\in \{1, 2, 3\}$  thể hiện cho phân khúc khách hàng tương ứng.
- Ví dụ:  $k = U_2$ ,  $r = 1$

### 3 Nộp bài

- Đặt lại tên tập tin bài làm của mình *StudentID.py* với StudentID tương ứng với Mã số sinh viên và nộp theo link quy định.
- **Deadline:** 20/4/2022-7/5/2022 (11h55 p.m)

#### Chú ý:

- Sinh viên không được phép thêm bất kỳ thư viện nào khác trong tập tin Student.py bài làm đã được cung cấp.
- Sinh viên mặc định sẽ nhận điểm 0 nếu chương trình không chạy được (tất cả các trường hợp thêm thư viện làm cho bài làm không chạy bằng chương trình chấm sinh viên mặc định nhận điểm 0, không gọi các hàm  $req_k(.)$  ( $\forall k \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ ) trong chương trình khi nộp bài, không để lệnh nhập, xuất trong bài làm).
- Sinh viên phải tự làm bài của mình. Tất cả những trường hợp tham khảo mã nguồn từ sinh viên khác hoặc từ các nguồn khác đã được công bố trước đó thì sinh viên sẽ nhận 0 điểm giữa kỳ.
- Sinh viên chỉ được phép giống bài mức tối đa 25%, nếu vượt mức từ 26-55% sinh viên làm bài bổ sung, trường hợp trên 55% mặc định 0 điểm.
- Với những trường hợp kết quả trả về là số thực sinh viên phải sử dụng hàm  $round(z,1)$  hoặc  $np.round(z,1)$  để làm tròn số thực  $z$ .
- Đề có thể được cập nhật thường xuyên trong quá trình làm vì vậy sinh viên nên xem online để đảm bảo quyền lợi.
- Để load được file testcase mẫu Sinh viên sử dụng thư viện: **scipy.io** và lệnh **scipy.io.loadmat(tên file cần mở)**. Tuy nhiên, lưu ý rằng khi nộp bài Sinh viên chỉ giữ lại phần bài làm của mình không bao gồm phần mở file này nếu vẫn giữ làm cho việc không thay đổi được testcase dẫn đến 0 điểm. Sinh viên phải tự chịu trách nhiệm.

### 4 Tài liệu tham khảo

1. Ma Siu Lun, [2012], Linear Algebra: Concepts and Techniques on Euclidean Spaces, McGrawHill, Singapore.
2. Howard Anton, Chris Rorres, [2005], Elementary Linear Algebra: Applications Version Tenth Edition, John Wiley & Son, Inc, USA.