

Môn học: Thực hành toán ứng dụng thống kê

Báo cáo lab 2

Họ và tên: Bùi Thị Thanh Ngân

MSSV: 21120505

Bài 1: Tính tích vô hướng của hai vector

Hàm `innerproduct(v1, v2)`:

Ý tưởng thực hiện:

- Dùng vòng lặp for để duyệt qua các phần tử của hai vector và tính tổng tích lũy của các tích của hai phần tử tại chỉ số tương ứng

Mô tả hàm:

- input: `v1, v2` là hai vector cùng kích thước
- output: tích vô hướng của hai vector

```
def innerproduct(v1, v2):  
    sum = 0  
    for i in range(len(v1)):  
        sum += v1[i] * v2[i]  
    return sum
```

Bài 2: Phân rã ma trận A thành hai ma trận Q và R

Hàm `QR_factorization(A)`:

Ý tưởng thực hiện:

- Chuyển các cột của ma trận A thành vector và lưu vào một list các vector B $\{b_1, b_2, \dots, b_n\}$
- Tìm cơ sở trực giao của B $\{b_1, b_2, \dots, b_n\}$ và lưu vào một list các vector U $\{u_1, u_2, \dots, u_n\}$
- Tìm cơ sở trực chuẩn của B $\{b_1, b_2, \dots, b_n\}$ và lưu vào một list các vector V $\{v_1, v_2, \dots, v_n\}$
- Ma trận Q là ma trận của các vector trực chuẩn
- Ma trận R là tích của ma trận chuyển vị của ma trận Q và ma trận A

Mô tả hàm:

- input: A là ma trận có m dòng n cột với các phần tử là các số thực
- output: ma trận Q và ma trận R

- Các hàm nhỏ hỗ trợ:
 - Hàm `transpose(A)` : trả về ma trận chuyển vị của ma trận A
 - Hàm `NhanMaTran(A,B)` : trả về tích của hai ma trận A và B
 - Hàm `subV(v1, v2)` : trả về vector $v_1 - v_2$
 - Hàm `multiplyV(v, k)` : nhân vector v với một số k
 - Hàm `divide(v, k)` : chia vector v cho một số k

```
def QR_factorization(A) :
    # số hàng:
    nrow = len(A)
    # số cột:
    ncol = len(A[0])

    # đưa ma trận về mảng các vector cột {b1, b2, ..., bn}
    B = []
    for i in range(ncol) :
        B.append([A[j][i] for j in range(nrow)])

    # tìm một cơ sở trực giao {u1, u2, ..., un}
    U = []
    for i in range(len(B)):
        Ai = B[i]
        Ui = Ai
        for j in range(len(U)):
            Ui = subV(Ui, multiplyV(U[j], innerproduct(Ai, U[j]) /
innerproduct(U[j], U[j])))
        U.append(Ui)

    # tìm cơ sở trực chuẩn {v1, v2, ..., vn}
    V = []
    for i in range(len(U)):
        Vi = divideV(U[i], lenV(U[i]))
        V.append(Vi)

    # Khởi tạo ma trận Q
    Q = []
    for i in range(nrow):
        Q.append([0 for _ in range(ncol)])

    # ma trận Q = [v1, v2, ..., vn]
    for i in range(nrow):
        for j in range(ncol):
            Q[i][j] = V[j][i]

    # ma trận R = Q chuyển vị * A
    T = transpose(Q)
    R = NhanMaTran(T, A)

    return Q,R
```

Input, Output:

- **Input:** file input.txt
 - Dòng đầu tiên chứa số nguyên n là số ma trận A
 - Dòng tiếp theo chứa hai số m n là số dòng và số cột của ma trận A
 - m dòng tiếp theo mỗi dòng chứa n số là các phần tử của ma trận A
 - Tương tự cho các ma trận sau
- **Output:** file output.txt
 - Dòng đầu tiên ghi thông tin thứ tự ma trận A
 - Ma trận Q
 - Ma trận R
 - Tương tự cho các ma trận sau