**Principal Component Analysis (PCA)**

1. Giới thiệu:

Dimentionality Reduction (giảm chiều dữ liệu) là một trong những phương pháp để giảm số chiều dữ liệu, đây là một dạng của phương pháp nén dữ liệu nhằm giải quyết được vấn đề lưu trữ và tốc độ tính toán.

Dimentionality Reduction: là việc đi tìm một hàm số, hàm số này lấy đầu vào là một điểm dữ liệu ban đầu **x** ∈ với **D** rất lớn, và tạo ra một điểm dữ liệu mới **x** ∈ có số chiều **K < D**

1. Nguyên nhân:

Các features vector trong bài toán thực tế có thể có số chiều rất lớn, tới vài nghìn. Ngoài ra, các điểm dữ liệu cũng thường rất lớn, nếu thực lưu trữ và tính toán trực tiếp trên dữ liệu có số chiều lớn như thế sẽ gặp khó khăn trong việc lưu trữ và tốc độ tính toán.

**Singular Value Decomposition (SVD)**

1. Giới Thiệu:

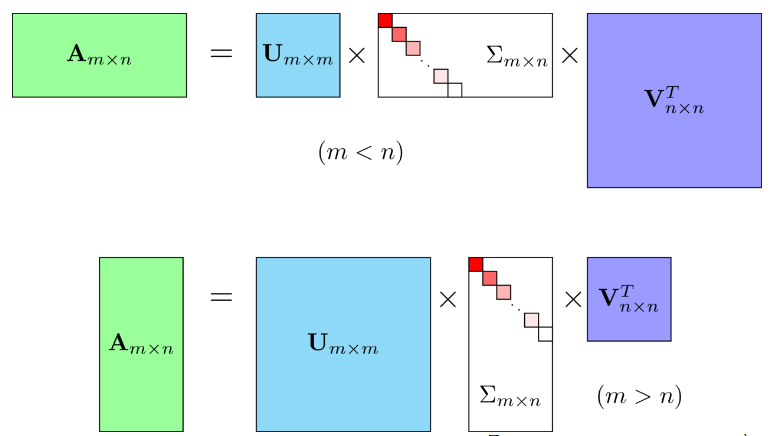
Việc phân tích ma trận ra thành tích của nhiều ma trận khác (Matrix Factorization hoặc Matrix Decomposition) mang lại nhiều lợi ích quan trọng như: giảm số chiều dữ liệu (dimentionality reduction), nén dữ liệu, tìm hiểu các đặc tính của dữ liệu, giải các hệ phương trình tuyến tính, clustering, ngoài ra recomendation system cũng là một trong những ứng dụng điển hình của Matrix Factorization. Phương pháp đó có tên là Singular Value Decomposition (SVD), với mọi ma trận, không nhất thiết là vuông, đều có thể được phân tích thành tích của ba ma trận đặc biệt.

1. Phân Tích

Một ma trận bất kì đều có thể phân tích thành dạng:

= (I)

Trong đó: **U**,**V** các ma trận trực giao, là ma trạn đường chéo không vuông với các phần tử trên đường chéo, và **r** là rank của ma trận **A**.Lưu ý rằng mặc dù Σ không phải ma trận vuông, ta vẫn có thể coi nó là ma trận chéo nếu các thành phần khác không của nó chỉ nằm ở vị trí đường chéo, tức tại các vị trí có chỉ số hàng và chỉ số cột là như nhau.



Hình 1: SVD cho ma trận **A** khi: **m < n** (hình trên), và **m** **>** **n** (hình dưới). **Σ** là một ma trận đường chéo với các phần tử trên đó giảm dần và không âm. Màu đỏ càng đậm thể hiện giá trị càng cao. Các ô màu trắng trên ma trận này thể hiện giá trị 0.

*#Thêm thư viện  
#numpy.linalg là thư viện cho phép tính SVD***from** numpy **import** linalg **as** LA  
*#Tạo ma trận 2x2*A = [[4,0],[3,-5]]  
*#Tách ra 3 ma trận U, S, V riêng biệt*U, S, V = LA.svd(A)  
*#Đảo ngược ma trận V bằng biến V\_T*V\_T=V.T  
print(**"Kiểm tra kích thước của 3 ma trận U,S,V"**)  
print(**"\n U shape= {}\n S shape= {}\n V shape= {}\n"**.format(U.shape,S.shape,V.shape))  
print(**"In ra 3 ma trận U, S, V tìm được:"**)  
print (**'\n U= {0}\n\n S = {1}\n\n V={2}\n'**.format(U,S,V))  
print(**"Đây là ma trận Singular value decomposition(SVD) tìm được:**

**A = {}"**.format((U\*S\*(V\_T))))

Kết Quả:

Kiểm tra kích thước của 3 ma trận U,S,V

U shape= (2, 2)

S shape= (2,)

V shape= (2, 2)

In ra 3 ma trận U, S, V tìm được:

U= [[-0.4472136 -0.89442719] [-0.89442719 0.4472136 ]]

S = [ 6.32455532 3.16227766]

V=[[-0.70710678 0.70710678][-0.70710678 -0.70710678]]

Đây là ma trận Singular value decomposition tìm được:

A = [[ 2. 2.] [-4. -1.]]